TOWARDS VAULTING THE HURDLE OF SHORT LIFETIMES IN WIRELESS SENSOR NETWORKS: DISTRIBUTED ALGORITHMS AND UWB IMPULSE RADIO

by

ANAND OKA

M.Sc., The Technion - Israel Institute of Technology, 1999
B.E., The University of Pune (Government College of Engineering, Pune), 1995

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

THE FACULTY OF GRADUATE STUDIES

(Electrical and Computer Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

July 2009

Abstract

Wireless Sensor Networks (WSNs) offer a compelling solution for distributed sensing problems because they can be deployed rapidly and inexpensively, and are robust to failures. However, since they operate on batteries, they tend to have short lifetimes. We present several algorithmic techniques for reducing the power consumption of such networks, based on Algorithmic Data Reduction (ADR) and low-power Ultra-Wide-Band Impulse Radio (UWB-IR).

In the ADR approach, we minimize power-hungry communication out of the network via distributed in-situ broadcast 'message-passing' algorithms for filtering, compression and model identification. These algorithms are scalable, power-efficient, stable, and computationally tractable. At the same time their performance is close to the respective ultimate theoretical limits. Specifically, the filter performs close to an optimal Bayesian recursion, the compressor approaches the rate-distortion and channel-capacity bound, and the identification scheme is asymptotically efficient in the Cramer-Rao sense.

The UWB-IR approach exploits a well-known tradeoff predicted by Shannon theory, namely that one can maintain reliable communication at a given data rate at a reduced transmit power provided the transmission bandwidth is requisitely increased. We propose a novel UWB-IR receiver, which is eminently suited to the bursty mode of operation of the WSN physical layer. The receiver is based on the principle of Compressed Sensing and offers a practical alternative to costly high-rate analog-to-digital

conversion. It can tolerate strong inter-symbol interference and can therefore operate at high pulsing rates, which allows us to fully leverage the power-vs-bandwidth tradeoff. It is impervious to poor timing synchronization, which means that the transmitter can avoid sending training headers, thus further saving a significant amount of power. In addition, it is also robust to strong narrow-band interference from licensed systems like WiMAX.

With a synergy of the ADR and UWB-IR techniques, the communication related power consumption of the WSN can be reduced by about 30 dB or more in practical scenarios, which substantially alleviates the handicap of limited lifetimes. We study a practical application of these techniques in the problem of target tracking by interpreting the received signal strength of transmissions from RFID tags.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ADC	Analog to Digital Conversion
ADR	Algorithmic Data Reduction
AWGN	Additive White Gaussian Noise
BBP	Broadcast Belief Propagation
BP	Belief Propagation
bps	Bits per Second
BS	Base Station
CLO	Cross Layer Optimization
CPE	Customer Premise Equipment
CRLB	Cramér-Rao lower bound
CS	Compressed Sensing
DAC	Digital to Analog Conversion
DFC	Digital Fountain Code
DTR	Differential Transmit Reference
ED	Energy Detection
EM	Expectation Maximization
FC	Fusion Center
FF	Fusion First
FFT	Fast Fourier Transform

GMRF	Gauss-Markov Random Field
GPS	Global Positioning System
GS	Gibbs Sampling
HMM	Hidden Markov Model
HMRF	Hidden Markov Random Field
i.i.d.	Independent and Identically Distributed
ICM	Iterated Conditioning of Modes
IF	Inference First
IFFT	Inverse Fast Fourier Transform
IP	Interior Point (method of solving LP and QP optimization problems)
IR	Impulse Radio
ISI	Inter-Symbol Interference
LDPC	Low Density Parity Check
LMS	Least Mean Square
LOS	Line of Sight
LP	Linear Program
MA	Multiple Access
MAC	Medium Access Control
MAP	Maximum A-posteriori Probability
MB-OFDM	Multi-band Orthogonal Frequency Division Multiplexing
MC	Markov Chain
MCMC	Markov Chain Monte-Carlo
MFD	Mean-Field Decoding
MLSE	Maximum Likelihood Sequence Estimation
MMSE	Minimum Mean Square Error

MRC	Maximum Ratio Combining
MRF	Markov Random Field
OFDM	Orthogonal Frequency Division Multiplexing
PAN	Personal Area Network
p.d.f.	Probability Density Function
p.m.f.	Probability Mass Function
PEP	Pairwise Error Probability
PSD	Power Spectral Density
QOS	Quality of Service
QP	Quadratic Program
RF	Radio Frequency
RFID	Radio Frequency Identification
SIR	Signal to Interference Ratio *or* Sampling Importance-Resampling
SIR_{bit}	Signal to Interference Ratio per bit
SNR	Signal to Noise Ratio
SNR_{bit}	Signal to Noise Ratio per bit
SPA	Sum-Product Algorithm
SPR_{bit}	Signal to Perturbation Ratio per bit
TOA	Time of Arrival
TR	Transmit Reference
UEP	Uniform Error Property
UWB	Ultra-Wide-Band
UWB-IR	Ultra-Wide-Band Impulse Radio
WSN	Wireless Sensor Network

Notation

Probability: We will typically use a capital letter like A to denote a random variable, and a small letter like a to denote an instance or a dummy variable drawn from its alphabet. We will make an abuse of notation and denote the p.d.f. or p.m.f. of X by $P(x)$ rather than $P_X(x)$, and similarly the p.d.f. or p.m.f. of X conditioned on Y by $P(x|y)$ etc. $\mathbb{E}_{X\sim p}\left[f(X)\right]$ denotes the expectation of a function $f(X)$, with X distributed according to p. When no confusion arises, we simply write this as $\mathbb{E}\left[f(X)\right]$.

Vectors and Matrices: No particular upper case, lower case, under-bar or over-bar notation is used for vectors and matrices. Unless otherwise specified, vectors and matrices are presumed real and all matrix-vector operations are performed in the real field $\mathbb{R}$. Suppose a, b are column vectors and A a matrix of compatible dimensions. Then a_i denotes the i^{th} component of a, $|a|$ denotes a vector of the absolute values of components of a, and $a + b$ and Aa are usual matrix-vector operations.

Norms: $\|a\|_2$ will denote the L_2-norm (Euclidean length), $\|a\|_1$ the L_1-norm (largest absolute value), and $\|a\|_0$ the number of non-zero elements of a. When A is a square matrix, $\|A\|$ will signify its spectral norm and $\|A\|_F$ the Frobenius norm.

Other miscellaneous notation:

$\mathbb{R}^N$	space of real vectors of dimension N
$\mathbb{R}^{N\times N}$	space of real matrices of dimension $N \times N$
$(\cdot)^*$	complex conjugate

$(\cdot)^T$	transpose
$\text{diag}(x)$	a matrix with the elements of vector x on the main diagonal
$\doteq$	defined equal to
$\Pr\{\cdot\}$	the probability of some event
$h(t) \star \phi(t)$	convolution of $h(t)$ and $\phi(t)$
$\mathcal{F}\{h(t)\}$	Fourier transform of $h(t)$
$H(f), \Phi(f)$ etc	$\mathcal{F}\{h(t)\}, \mathcal{F}\{\phi(t)\}$ (unless otherwise indicated)
$\mathcal{N}(z; \mu, \Sigma)$	Multivariate Gaussian density of *mean* μ and *covariance* Σ
$\mathcal{N}_C(z; \theta, W)$	Multivariate Gaussian density of *bias* θ and *precision* W
$\mathbf{1}$	A matrix or vector with all elements equal to 1
I	An identity matrix
I_M	An identity matrix of dimensions $M \times M$
$I(\sigma)$	Information Content of LLR distributed as $\mathcal{N}(\sigma^2, \sigma^2)$
$D(f_1(x) \Vert f_2(x))$	KL divergence of distribution $f_2(x)$ w.r.t. $f_1(x)$, $\doteq \mathbb{E}_{X \sim f_1}\left[\log \frac{f_1(X)}{f_2(X)}\right]$
$U([a, b])$	A uniform distribution over the interval $[a, b]$, of the real line or integers, depending on the context.

Acknowledgments

I am indebted to my advisor Prof. Lutz Lampe for his guidance and encouragement throughout this thesis research, and I am deeply appreciative of the trust he placed in my abilities. It has indeed been a pleasure to study and collaborate with him.

I would like to express my gratitude to Prof. P. S. Krishnaprasad at the University of Maryland, College Park, for being an inspiring teacher and mentor, and for his infectious enthusiasm for the joys of research. I am grateful to my doctoral committee members, Prof. Vikram Krishnamurthy and Prof. Jane Wang, for their guidance and supervision during my research, as well as the university examination committee members, Prof. Rabab Ward and Prof. Nando De Freitas, for their time and effort. I am also greatly honored to have Prof. H. Vincent Poor of Princeton University as my external examiner, and I would like to thank him for his encouraging appraisal of my work.

I would like to thank my colleagues at UBC - Anna, Chris, Jeebak, Paul, Jan and many others - for their help and support during my studies here. In particular, I would like to thank Anna for patiently indulging my questions, comments, criticisms and armchair philosophy over the last three years, and convincing me through her own brilliant example that hope triumphs cynicism.

Last but not the least, I owe my wife Sheila my heartfelt appreciation and gratitude. I am lucky to have a spouse who understands my innermost aspirations and fears. Without her unflinching support this doctorate could not have been accomplished.

The financial support of NSERC (through its Post-Graduate Scholarship) and the British Columbia Innovation Council (through the British Columbia Industrial Innovation Scholarship in Intelligent Systems) is gratefully acknowledged.

Dedication

To my lovely baby daughter, Anoushka.

1 Introduction and Overview

Wireless Sensors Networks (WSNs) form a special class of *ad-hoc*, *multi-hop*, *application specific*, *mesh* communication networks. In Section 1.1 we motivate why such networks are of great interest to engineers and scientists alike, and discuss some of their uses. Then, in Section 1.2, we examine the principal obstacle to their widespread application, namely a small lifetime due to limited batteries. In Section 1.3 we survey the literature concerning solutions to this problem, which spans several specialties like information theory, statistical signal processing, and communications theory. Section 1.4 presents an overview of our thesis and a statement of the major results. Appendix A lists our publications based on this research.

1.1 The Many Uses of Wireless Sensor Networks

On December 24, 2004 an undersea earthquake off Sumatra created a massive tsunami that washed across the coastal regions around the Indian ocean rim, killing almost a quarter million people and wreaking massive devastation. Unfortunately, due to an absence of suitable monitoring mechanisms, sufficient forewarning was not available, and there was much avoidable loss of life. Following this eye-opening experience, a group of ocean rim nations have set up a Tsunami Warning System - a network of wirelessly connected monitoring stations afloat on buoys - which will likely drastically reduce loss

of life in any future Tsunami events.

Tsunami Warning systems and *Geodetic networks* (for seismic monitoring) are two eye-catching examples of how WSNs can make a significant positive impact on our lives. But lest the reader think that WSNs are limited to geophysical telemetry alone, consider some other examples. In *Building Automation*, a network of small wireless devices can measure the micro-climate and occupancy of buildings and dramatically improve the efficiency of heating, ventilation, air-conditioning (HVAC) and lighting systems. *Security and Surveillance* networks can localize and track people and goods in vast spatial areas (even indoors and in mines), and warn of suspicious activity. *Body Area Networks* can monitor the vital signs of patients in hospitals while allowing them to move around freely, thus considerably improving their quality of life, which is especially important in the long-term care of elderly and Alzheimer patients. In fact, WSNs have applications [1] that range from *habitat monitoring* [2] in the deepest trenches of Earth's oceans to *extraterrestrial exploration* [3].

What is it about WSNs that makes them so versatile? It is the fact that a WSN is an *infrastructure-less* ('ad-hoc') wireless network. Each node of the network, called a *mote*, is a small inexpensive device equipped with a battery, a transceiver, a modest micro-controller and one or more sensors.[1] By an infrastructure-less network, we mean that there is no a-priori architecture for the placement of, or communication among, the motes. This allows the network to be set up quickly and inexpensively. Typically the motes will be randomly scattered in the region of interest, perhaps even from a plane, where they self-organize into an adequate logical structure of clusters and neighborhoods. Then they start measuring environmental data and producing usable inference, which is

[1]How small can these motes be? Practical motes already have a form factor of a match-box, and there are realistic proposals for creating *smart dust* [4], a collection of thousands of motes about 1 mm^3 in size, that can be mixed into concrete, alloys etc so as to enable a comprehensive monitoring of the integrity of structures like bridges and dams.

communicated back to a monitoring station called a *Fusion Center*.

While a few 'anchor' motes may be localized manually or equipped with Global Positioning System (GPS) receivers to give the network an absolute frame of reference, the vast majority of the motes localize themselves via a collaborative effort based on message passing. In fact such collaborative effort is the hall-mark of WSNs which distinguishes them from centralized sensing systems. Additionally, the ad-hoc mesh nature of the network, while being a major design challenge, gives it an intrinsic robustness and self-healing ability in that the failure of a small fraction of the motes has no noticeable impact on the overall performance of the system.

WSNs are said to be turning a new page in engineering as well as science. From an engineering perspective, WSN technology allows us to build *intelligent systems* that can accurately sense distributed phenomena and implement more agile and accurate feedback control, while retaining the ability to scale almost indefinitely in spatial scope. In general this yields a reduction in cost and an improvement in operational efficiency. Scientists, on the other hand, can now access a deluge of detailed, first-hand, real-time data on the phenomena of their interest, be it the formation of tornadoes, migration of the Pacific salmon stock, or the effect of air-pollutants on public health. This fundamentally improves their ability to construct useful theories.

1.2 The Hurdle of Limited Lifetimes

The radio transceiver that sits in each mote untethers it from wired communication links like optic fibers or twisted copper wires, while the battery in the mote frees it from power cords, thus making it a truly portable device. It is however a common observation that a typical mote dies within days of deployment when used at full sensing and transmission capacity, simply because its battery runs out. Since it is prohibitively expensive to

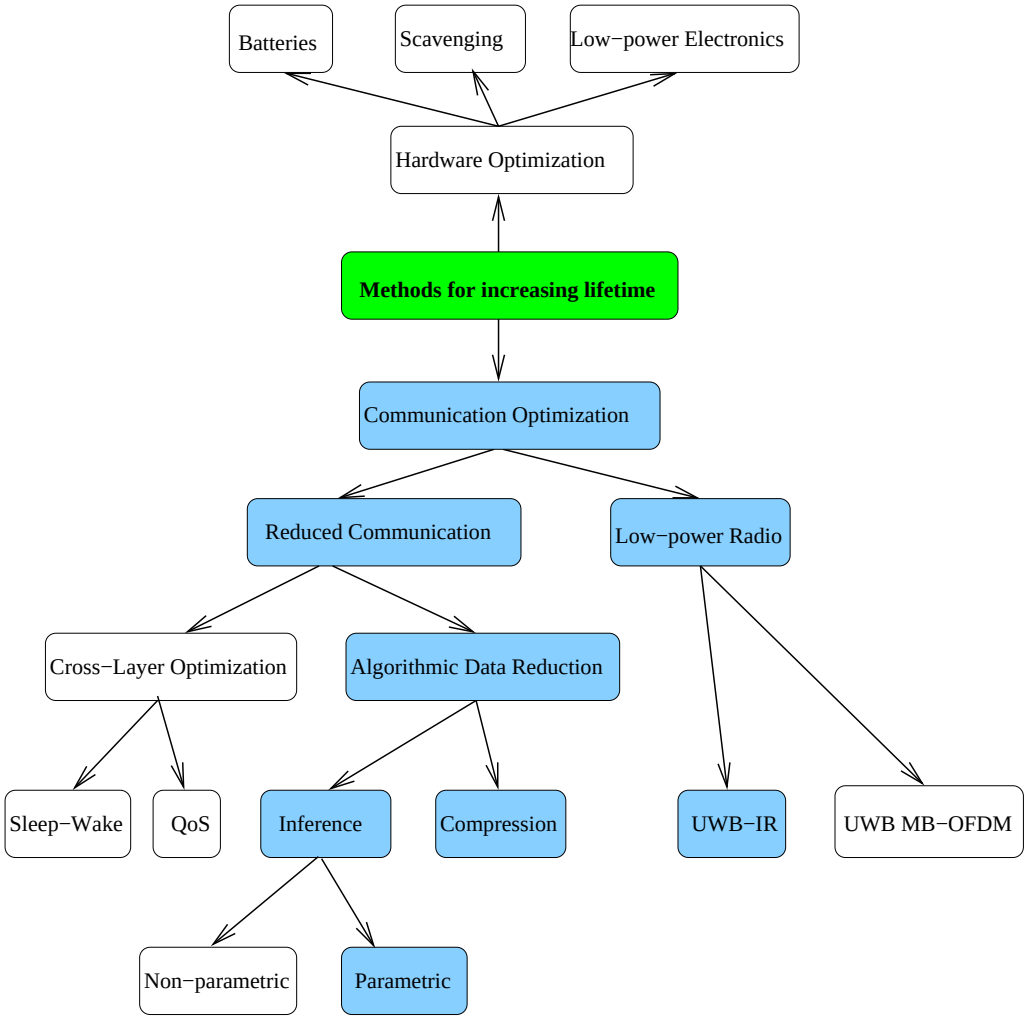

Figure 1.1: An overview of various approaches to improving the lifetime of WSNs. The shaded boxes indicate the methods we have investigated.

manually replace batteries in the field at short intervals, WSNs can become a practically attractive proposition only when we can dramatically increase the lifetime of the motes.

A mote's battery is drained because of two reasons: (i) Power is consumed in the operation of the sensors, micro-controller and supporting electronics, and (ii) Power is physically radiated by the radio transceiver over the air. Hence, as shown in Figure 1.1, the approaches to improving the lifetime of WSNs are principally divided into (i) *hardware optimization* and (ii) *communication optimization*, respectively.

Hardware optimization involve the use of specially fabricated low-power electronic circuits, more efficient and long-lasting batteries, and even the intriguing possibility of *energy scavenging*, where the mote harvests solar, wind, vibrational and thermal energy from its environment. Unfortunately, low-power electronics has had a limited impact because the sensors and the micro-controller consume little power relative to the radio transmitter, especially when the motes are *sparsely* deployed. Similarly, it is highly unlikely that there will be an increase of orders of magnitude in the energy packing efficiency of batteries. Scavenging techniques too can ultimately provide only a limited source of power, as dictated by basic physics.

This leads us to the inevitable conclusion that we must reduce the power spent on *wireless communication*, via communication optimization. This approach involves two distinct and mutually compatible possibilities, namely, *reduced communication* and *low-power physical layer radio*. Information theory [5] tells us that a certain minimum amount of transmit power is necessary to maintain reliable communication at a given data rate over a given distance using a given bandwidth. Since the communication distance is typically a constant dictated by the application, the only variables we can engineer are the data rate and the bandwidth. That is, we must try to decrease the data rates and utilize the maximum allowed transmission bandwidth. This cardinal principle is essentially the subject of our thesis.

1.3 Literature Survey

We will now briefly review the literature on *energy efficient data aggregation algorithms* and *low-power physical layer communication* (in particular, *ultra-wide-band radio*), which form the appropriate context for our research. Note that this constitutes but a small fraction of the vast over-all literature on WSNs, which spans such other diverse

topics as power-efficient hardware and large scale integration, networking, protocols and standards, programming languages, operating systems and visualization tools. A survey of all these areas would far exceed the scope of this thesis.

1.3.1 Algorithmic Data Reduction

Cross-layer Optimization or Algorithmic Data Reduction (or both) ?

As mentioned earlier, one approach to lifetime enhancement through communication optimization is by *reduced communication*, and this can further be divided into two methods that are not necessarily mutually exclusive. One is a *protocol based cross-layer optimization* (CLO) approach [6, 7, 8, 9, 10], a simple example of which is *sleep-wake* scheduling [11]. This is a top-down approach where the end-application queries the network *on demand*, and specifies a quality of service (QOS) [12]. The query causes the sensors to turn on, make pertinent measurements, communicate relevant data to the user, and then relapse into a hibernation mode. The other is a bottom-up *algorithmic data reduction* (ADR) approach, which will be the focus of our thesis. This latter approach is relevant when it is essential to autonomously sample the underlying phenomena (the *field*) at a sufficiently high rate without hiatuses, even in the absence of any explicit queries, so that important transient phenomena are not missed. Since we must avoid massive transport of all this raw sampled data out of the network to conserve batteries, adequate in-network processing needs to be implemented that can allow the WSN to notify the Fusion Center only with informative *meta-data*, as events of interest occur. The ADR techniques can be further classified into the following two categories:

- Distributed Inference (which includes topics like Detection, Filtering, Model Identification, Regression and Learning)
- Distributed Compression (which includes topics like Compression, Networked Stor-

age, Correlated Data Gathering, and Energy Aware Routing)

Distributed Inference

Distributed Detection [13, 14, 15, 16, 17, 18, 19, 20] is a method of data aggregation that is relevant to applications like detection of plumes of contaminants in air, oil slicks in water etc, where the end-user is interested in testing for one out of a small set of hypotheses [21, 22, 23]. For example, suppose that nature presents a binary hypothesis H_0/H_1 corresponding to the presence or absence of a contaminant plume, and the sensors make conditionally independent measurements of the concentration of the contaminant, often in the presence of several unknown parameters (decalibration). We wish to achieve an efficient hypothesis test at the FC, *without exporting all raw measurements*. This is achieved by some form of a collaborative distributed message-passing algorithm implemented within the network, where the messages are *partial* results generated by *localized* rules. It has been shown [19] that even such a constrained system can, in some cases, attain an optimal performance in the Neyman-Pearson sense, and in other cases the loss in performance is not too great. *Distributed Filtering* is a generalization of the detection problem where we are interested in inferring a multitude of hypotheses or even an entire spatio-temporal processes [15, 24, 25, 26, 27, 28, 29, 30, 31].

While [15, 16, 18, 19, 24, 25, 26] consider detection or filtering in a statistical parametric setting, the approach of [17, 30, 31] is a non-parametric one, based on kernel regression and smoothening, also known as *Kriging* [32]. This is actually a classical idea predating sensor networks, originating in *Reproducing Kernel Hilbert Space* (RKHS) regression [33, 34], which has been reinvented in the context of WSNs in the form of *Distributed Learning* [30, 31, 35, 36]. Kernel regression methods are more robust than statistical parametric methods because they do not need an a-priori model or parameters for the statistical behavior of the field (though they do need a suitable kernel,

as well as smoothening and regression hyper-parameters). On the other hand they are significantly less accurate when full model knowledge is available or can be attained. A discussion on the correspondence between these approaches can be found in [37].[2] A popular instance of distributed filtering occurs in the distributed tracking of targets carrying Radio Frequency Identification (RFID) tags [8, 38, 39, 40, 41, 42, 43, 44, 45].

Distributed Compression

The Nyquist sampling theorem [46] gives *sufficient* conditions on spatial or temporal sampling rates of band-limited processes, such that perfect reconstruction in the mean squared error (MSE) sense is assured, provided the samples are of infinite precision. In real life, of course, the samples always have finite precision, and we are interested in minimizing not just the total number of samples but the total number of *bits* needed for adequate representation. [47, 48] have demonstrated a so called *bit conservation principle* for distributed sensing systems which says that, for *band-limited* or *sufficiently smooth* fields, one can trade the sampling precision for the sampling rate. This property can be especially useful in the spatial domain because spatial oversampling can be done naturally by increasing the density of the motes. A very contrasting approach to the above is that of *under-sampling* of physical phenomena, albeit with high precision per sample, provided we have adequate a-priori information of their structure. This is the exciting new field of *Compressed Sensing* [49, 50, 51, 52, 53, 54] whose main principle

[2]There is an ambiguity in literature on the convention for the use of the terms *filtering* and *estimation*. While both imply discovery of hidden quantities from observed data, filtering typically refers to inference of rapidly changing random processes (in time or space), usually in a Bayesian setting, while estimation usually refers to the discovery of quasi-static deterministic parameters with a Maximum-Likelihood approach. However some researchers also use the latter to refer to Bayesian inference of random variables. In the non-parametric regression approach, the separate notions of filtering and estimation are, strictly speaking, not well-defined, tough one can say that estimation corresponds to calculation of the correct regression coefficients while filtering corresponds to interpolation or extrapolation based on these learned coefficients.

is that *random-like decoherent linear measurements* form sufficient statistics with a low overhead that allow perfect reconstruction of *sparse* signals. Recently some regularization methods have also been proposed to attain a modicum of robustness to a low sampling precision or the presence of noise [52, 55].

Bit-conservation as well as compressed sensing fall under the broader gamut of *rate-distortion* techniques [56] from Information Theory. Given a well-defined distortion measure, the rate-distortion characteristics $R(D)$ of a discrete or continuous stochastic source [5, 57, 58] specifies the minimal information needed in bits per sample, R, to achieve a given average distortion level, D, via the use of a suitable source code. The literature on rate-distortion itself is very vast, where some pivotal results are: efficient[3] non-universal loss-less variable-length compression with prefix codes (Huffman codes) [59], efficient universal[4] loss-less compression with Lempel-Ziv codes [60] and recently with Fountain Codes [60], efficient non-universal distributed[5] loss-less compression by Slepian-Wolf Codes [61], of which practical realizations are DISCUS codes [62] and Correlated Data Gathering [63], and efficient non-universal distributed lossy compression with Wyner-Ziv codes [64, 65]. The holy grail of rate-distortion theory, currently an open problem, is an efficient universal distributed lossy compression scheme with linear complexity encoding and decoding.

However, the application of rate-distortion principles in WSNs is *not* necessarily in this most general setting [66, 67]. Firstly, it is not unrealistic to allow some cooperation between motes during encoding (which is not the case in a strictly distributed source code in the Slepian-Wolf or Wyner-Ziv sense). In fact, in addition to compression prior

[3]By *efficient* we mean approaching the $R(D)$ characteristic.

[4]A source code is *universal* if the structure of the encoder is independent of the source statistics.

[5]A multi-terminal source code is said to be *distributed* (in the information theoretic sense) if each terminal encodes its data without looking at the data of any of the other terminals.

to transmission out of the network, the notion of *distributed networked storage* is also of interest [68, 69, 70], where some form of inter-mote cooperation is mandatory. Here the idea is that functionals of the data measured by the motes are dispersed throughout the network, typically using a *rate-less* fountain code [71, 72], in such a way that a mobile agent can query any sufficiently large subset of the motes and reconstruct *all* the data. Secondly, even when considering the traditional compress-and-transmit scenario, the ultimate aim is to reconstruct the data at the FC with a minimal number of *transmissions* from the motes, and hence the exploitation of all the available channel capacity is as important as efficient compression.

At first blush this may seem to be an easy problem since the literature on channel coding is even more abundant than that on rate-distortion, and practical capacity approaching codes now exist like irregular LDPC codes [73, 74, 75, 76], Turbo codes [77, 78, 79] and Raptor codes [80, 81]. However it is known [5, 82] that the principle of source-channel separation is *not necessarily* an optimally efficient procedure in the multi-terminal setting. Hence joint-source channel codes must be considered, on which the literature is relatively scant [83] and there are many open problems. Even in special cases where source-channel separation remains optimal, like a degenerate multiple access (MA) channel consisting of independent mote-to-FC links, it is known that much can be gained in terms of complexity by using a joint source-channel approach [5]. In addition to physical layer joint source-channel coding, an efficient method for collaborative data gathering based on the Medium Access Control (MAC) layer, called *Energy Aware Routing*, has also been suggested in literature [84, 85].

A final twist in the problem of efficient data extraction from WSNs comes from the possibility of *feedback* from the FC to the motes. Since the FC typically has no severe constraints on its power consumption, it can possibly make frequent broadcast transmissions to the motes to direct and assist the data extraction procedure. Again it

is known that while feedback cannot improve channel capacity of discrete memoryless channels (DMCs) [5], it can vastly improve the *complexity and delay* of the data extraction system thanks to a much improved coding exponent (the Burnashev exponent [86, 87, 88, 89, 90]) relative to the usual Gallager exponent [82].

In summary, the question as to what efficiencies are in principle achievable in data extraction from practical sensor networks seems to be, on the whole, still open. However we do know that the answer depends on the interplay of (a) the distortion measure, (b) the amount of inter-node cooperation, (c) joint source-channel techniques, (d) rate-less codes and (e) a judicious use of feedback. A unified view of this interplay is currently missing in literature and could go a long way in reducing the power consumption bottleneck in WSNs.

1.3.2 Ultra-Wide-Band Impulse Radio

Finally, let us turn our attention to the other approach to communication optimization, namely low-power radio, and consider the literature on UWB radio, which is widely regarded to be the ideal physical layer technology for WSNs since it inherently allows us to tradeoff bandwidth for a reduced transmit power. According to FCC requirements [91, 92], UWB transmissions must have a transmission bandwidth that is at least 20% of the center-frequency or greater that 500 MHz, and must satisfy a very stringent absolute transmit power mask (-41.3 dBm/MHz) in an allowed transmission band of 3 to 10 GHz. The reason behind these exacting requirements is that UWB systems are allowed to be *unlicensed*, but are treated as *secondary users* of the spectrum who must not noticeably interfere with primary licensed systems like WiMAX. While this restricts UWB to short-haul multi-hop systems, this category is still very vast and includes very high-rate applications like wireless communication between a desktop computer and

peripherals, as well as low-rate low-power applications like sensor networks, as can be seen from the following example.

Under the FCC spectral mask, the maximum UWB transmit power is limited to about $P_{TX} = 0.5$ milli-Watts. Assuming a link length of 25 meters, geometric spreading with a path-loss exponent $\varrho = 3.0$, and a center frequency of $f_c = 6$ GHz, the total path loss according to the Friis model [93] is ~ 80 dB, and hence the received power is $P_{RX} \sim 5$ pico-Watts. Shannon's celebrated theorem tells us that the capacity of an Additive White Gaussian Noise (AWGN) channel is given by

$$C = B \log_2 \left(1 + \frac{P_{RX}}{N_0 B}\right) \qquad \text{[bits per second]}, \tag{1.1}$$

where B is the transmission bandwidth (Hz) and $N_0 = kT$ is the single-sided thermal noise power spectral density (Watts/Hz)[6]. One can deduce [94] from expression (1.1) that as $B \to \infty$, $C \to \frac{P_{RX}}{N_0} \log_2 e$. However this saturation tends to happen at very high bandwidths (in the excess of several GHz), and up to that point the capacity rises very fast (almost linearly), as evidenced from Figure 1.2. Notice that the asymptotic capacity in the above example is 1.74 Gbps, relatively a very large number that enables high rate applications as mentioned above. In contrast, for a moderate bandwidth like $B \sim 5$ MHz (as used in extant narrow-band PAN physical layer protocols like Zig-Bee/IEEE 802.15.4-2003 [95]) the capacity is only ~ 40 Mbps. The figure further reveals that if such a smaller data rate is acceptable, as in WSN applications, then a UWB system can in fact deliver it even if P_{RX} (hence P_{TX}) is reduced by 16 dB. This demonstrates that the huge bandwidth afforded in UWB communication is a precious resource that can be traded for a lower power consumption.

[6] $k = 1.38 \times 10^{-23}$ Joule/Kelvin is the Boltzmann constant and T is the ambient temperature in degrees Kelvin.

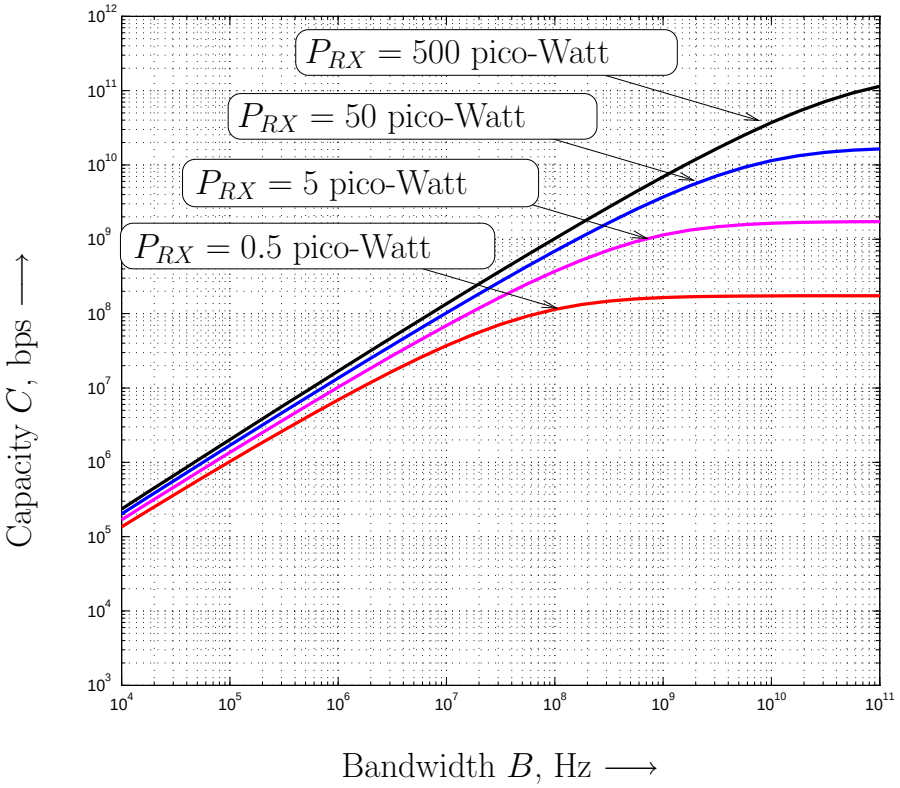

Figure 1.2: The capacity of an AWGN channel as a function of bandwidth, at received powers compatible with UWB transmissions. $N_0 = kT$ where $k = 1.38 \times 10^{-23}$ Joule/Kelvin is the Boltzmann constant, and $T = 300$ degrees Kelvin is the room temperature.

High-rate UWB applications are dominated by the *multi-band Orthogonal Frequency Domain Multiplexing* (OFDM) method of spread-spectrum signal generation [91, 96], whose principal advantage is that it makes it easy to tackle inter-symbol interference (ISI) caused by frequency selective fading, thanks to the fact that the individual sub-carriers are modulated at a relatively low baud-rate. Also, with adequate channel coding and interleaving, the system can be made robust to narrow-band interference. On the flip side, UWB-OFDM requires very accurate frequency and phase synchronization, otherwise there can be significant cross-talk among the sub-carriers. The requirement of accurate RF up and down conversion, and the use of high-speed FFT-IFFT and

DAC-ADC, makes UWB-OFDM systems relatively expensive and unsuited to low-cost low-power low-rate applications like WSNs.

A more attractive alternative for such applications is UWB based on *impulse radio* (IR), also some times called 'carrier-less' radio. In fact UWB-IR is actually the originally conceived form of UWB transmission [97], and involves a very simple low cost transmitter. A very short-lived (and hence ultra-wide-band) impulse of electromagnetic radiation is generated and modulated directly, usually by opto-electronic methods or avalanche-diode based circuits. There is no RF up-conversion or down-conversion and hence there are no stringent requirements of phase/frequency synchronization. The typical indoor UWB channel can have tens or even hundreds of multi-path components and a temporal dispersion of 10 to 100 nanoseconds [98]. Hence the pulse repetition rate is usually kept significantly smaller than the maximum possible rate (Nyquist rate) in order to eliminate inter-symbol interference, which allows a significantly simpler demodulator. This of course implies a reduced data rate which potentially eats into the available power vs bandwidth tradeoff. Moreover, the guiding principle in current practical impulse radio implementations is to avoid high-speed ADC altogether, in order to reduce cost as well as power consumption. Hence the detection of the UWB-IR signal is typically done entirely in analog, e.g. with a rake receiver (in coherent systems) [91], a transmit-reference (TR) [99, 100] or differential transmit-reference (DTR) [100] receiver (in semi-coherent systems), or a simple energy detection receiver [101, 102] (in non-coherent systems). In the case of rake receivers, one needs the knowledge of the total system impulse response, which can be obtained separately, perhaps using pilot symbols [103][104], as well as accurate timing synchronization [105, 106, 107, 108, 109].

UWB-IR systems are intrinsically very robust to multi-path fading [91] because each multi-path component is well-resolved on account of the short impulse duration (there is no 'destructive self-interference' as in narrow-band systems), and with proper techniques

all the incident signal energy can be retrieved. Similarly, UWB-IR transmissions allow the accurate estimation of the 'time of flight', from which one can obtain accurate ranging and localization [110, 111, 112, 113]. These advantages, in addition to the low-cost low-power transmitter, makes UWB-IR an ideal match to WSN type applications.

In summary, while UWB-IR seems to have great potential for WSN applications, extant practical UWB-IR systems suffer from several problems like sensitivity to timing errors, lack of robustness to ISI and costly analog processing like accurate delay lines or fast ADC sampling in the receiver. The currently proposed solutions to these problems seem unsatisfactory. For example, the use of long headers for timing synchronization introduces an unacceptable over-head for bursty transmissions as typically found in WSNs. Similarly, the use of a low pulsing rate gives up a big chunk of the potential power reduction achievable due to the large bandwidth. A solution of these problems could convert UWB-IR into a truly enabling technology for WSNs.

1.4 Overview of This Thesis

We will now present an overview of this thesis and a summary of the major results. (Note that Appendix A lists our publications based on this research.) First, in Chapter 2, we present a parametric statistical model for the behavior of the hidden field of interest and the measurements made thereof. This model is based on the *exponential family* [114] of probability distributions, which is very versatile and subsumes several important practical cases like finite and Gauss-Markov random fields (GMRFs). This source model will form the basis of the algorithmic strategies presented in Chapters 3, 4 and 5. The field model used in the target tracking algorithm of Chapter 6 is slightly different and will be described there in context.

As we discussed in Section 1.3.1, distributed inference is one major tool in the ADR

strategy. In this context, Chapter 3 presents a *distributed filter* [115, 116] for denoising/inferring a random process governed by a spatio-temporal *Hidden Markov Model* (HMM) from conditionally independent observations. The filter is a practical approximate realization of the optimal but intractable HMM-filter (Bayesian recursion) [117]. It performs an approximate marginalization of the sub-states of the joint distribution at every two consecutive discrete time instants. The marginalization is carried out via a distributed message-passing algorithm such as Broadcast Belief Propagation (BBP), Gibbs Sampling (GS), Mean Field Decoding (MFD) or Iterated Conditional Modes (ICM). We demonstrate that MFD is perhaps the best choice considering its complexity, power consumption and performance. We show that the filter possesses requisite properties of computational tractability and scalability, and asymptotically has a performance approaching that of the optimal HMM filter. Finally we prove a lower bound on the energy gain obtained by the in-situ distributed filter relative to a centralized filter at the FC, which demonstrates that a significant power-saving of the order of 10 dB or more is achievable in practical situations.

The other device in the ADR tool-box is source compression. Chapter 4 describes a *joint source-channel coding scheme* that simultaneously approaches the rate-distortion limit of the field as well as the capacity of the channel [118, 119]. The scheme uses a distributed fountain encoder, similar to the one used in *networked storage* [70]. However, crucially, *at the receiver* it also utilizes the a-priori statistical knowledge about the dependencies of the field to ensure that, on an average, a minimum number of transmissions are required to be made from the sensor array per sampling interval. While the scheme is not strictly distributed in the sense of Slepian-Wolf [61] or Wyner-Ziv [64], because it requires some cooperation between the nodes, it is fully distributed in a computational sense since no single mote is responsible for the entire encoding procedure, and the failure of a small fraction of the motes has no noticeable impact on the

performance of the system. We also discuss the connections of our scheme to the topic of compressed sensing of continuous valued fields [120].

The filtering scheme of Chapter 3, as well as the compression scheme of Chapter 4, requires that the statistical model of the field be a-priori known. This may seem to be a difficult condition to meet in practice. However we allay this concern in Chapter 5 where we demonstrate a distributed algorithm that incrementally identifies which particular instance of the model (from the family described in Chapter 2) is actually governing the observations made by the motes. The proposed recursive update is an on-line version [121] of the classical Expectation Maximization (EM) algorithm [122], and is shown to possesses desirable properties like stability, power efficiency, and covariance (Cramer-Rao) efficiency [123, 124]. By implementing the identification algorithm in tandem with the scheme of Chapter 3, we can obtain an *adaptive* filter, and in tandem with scheme of Chapter 4, a universal compression scheme.

In Chapter 6 we consider an application based on the ideas discussed in Chapter 3 and Chapter 5. We consider the problem of tracking targets carrying RFID tags that make periodic low power transmissions that are heard by nearby motes of the WSNs [125, 126]. From the received signal strength indication (RSSI) of these transmissions, the network infers and tracks the location of the targets via a distributed particle filter [43, 44, 127, 128, 129]. The filter exploits not just the temporal dependencies in the motion of each target but also the *inter-target* dependencies. However instead of a complex calculation based on the *joint multi-target probability density* (JMPD) [38, 41, 43, 44] we use a bank of sub-state particle filters that interact only on the basis of marginal statistics [130]. Moreover since radio occlusions and shadowing in the indoor environment can cause serious biases in the estimated positions [112], the algorithm also incrementally learns (identifies) the radio environment parameters and compensates for them when performing the particle filter update.

All the algorithmic solutions presented in Chapters 3 through 6 possess three desirable properties, namely

- *Distributedness and Scalability*: Even if the network scales in spatial area, while keeping the density of the motes approximately constant, the computation and communication load of each mote, and hence its power consumption, remains *invariant*. The lifetime of the network therefore remains unchanged even as it scales up in size.

- *Robustness to failures*: Even if a (small) fraction of the motes fail, the overall system performance is hardly affected.

- *Robustness to poor timing synchronization*: The message passing required by the algorithms can be performed in a fully asynchronous fashion, and fine symbol level timing synchronization among the motes is not necessary.

We saw in Section 1.3.2 that, apart from low-rate communication via ADR, the other stratagem for conserving the batteries in the motes is to trade bandwidth for a reduced transmit power. Furthermore we demonstrated that Ultra-Wide-Band Impulse radio is a well-placed physical layer technology for this purpose, though its extant implementations suffer serious problems with respect to ISI and poor timing synchronization, especially in bursty systems. In Chapter 7 we propose solutions to some of these leading issues based on a novel proposal for reception of bursty ultra-wide-band impulse radio signals [131, 132]. A receiver structure inspired by the idea of compressed sensing (CS) [50, 53] is proposed, that allows the use of sub-Nyquist sampling without serious loss in performance. Hence the receiver does not require high speed D/A conversion. Similarly, it does not need accurate timing synchronization with respect to the transmitter, and we need not waste power on training headers. Thirdly, the transmission time (channel

occupancy) of each mote can be kept quite small because the receiver allows the use of a high pulsing rate, *even if it causes heavy ISI*.

The proposed receiver permits a tractable implementation due to the use of a quadratic program (QP) based demodulator instead of a full fledged MLSE. At the same time it does not sacrifice significantly in performance and compares favorably with extant methods like rake reception [91], energy detection [101] and transmit reference [99]. The proposed receiver is coherent in the sense that it needs to know the total channel response in performing the demodulation. However we also provide a tandem incremental algorithm that identifies the said response *using the same analog observations* that are available for demodulation. Thus we have a totally blind receiver that does not need any external knowledge (like training sequences) for signal acquisition. Finally, in Section 7.6 it is also shown that the receiver can be made very robust to strong narrow-band interference from multiple licensed sources (like WiMAX customer premise equipment and base-stations) *without requiring* analog notch filters tuned in real-time (as is the case in conventional UWB receivers), which saves cost as well as complexity.

Chapter 8 provides a recapitulation of the major results of the thesis, and a discussion on future related research in ADR techniques and UWB-IR radio. In conclusion we offer some remarks on the outlook for WSN technology in the coming decade.

2 Statistical Field Model

As we demonstrated through several examples in Section 1.1, WSNs are often employed in practical applications such as the detection and tracking of plumes of noxious gases in air or oil-slicks on water, measurement of the distribution of temperature or salinity, assessment of pathogen count etc. In the statistical-parametric setting this can be viewed as the problem of estimating a spatio-temporal hidden random process, which we will simply refer to as the 'field'. The physics governing such fields [21, 133]) typically imposes certain *sparse localized Markovian* constraints on their spatio-temporal statistical distribution. Hidden Markov Random Fields (HMRFs), a major class of parametric models based on sparse graphs, are popularly used to describe such processes.

For example, Gauss-Markov random fields (GMRFs) are often used for modeling continuous valued spatio-temporal processes like temperature or salinity [24, 63, 84, 85, 134], with covariance structures chosen from classes such as spherical, power exponential, rational quadratic, Matérn etc, as befits the application [84]. On the other hand, when the WSN is used to simply detect the presence of some phenomena (like an oil slick) and to delineate its boundary, a binary valued MRF, also known as a Boltzmann or Ising model, is typically used [13, 15, 16]. A third application involves the use of Boltzmann/Ising models, already popular in centralized image processing [135, 136, 137, 138], for the *distributed* detection of textures and features in imagery obtained by *imaging* sensors [139, 140]. Lastly, Boltzmann/Ising models have also been proposed as a convenient tool

for distributed configuration management of sensors [141].

Apart from their great versatility, HMRFs are also a desirable choice from an analytical point of view. This is because it is known, thanks to the classical Hammersley-Clifford theorem [142], that *every* MRF has an equivalent representation based on an *exponential family* [142]. In fact, exponential families can even describe the class of bi-partite graphical models called Factor Graphs [143]. Much is known about the information-geometric properties [114] of this family.

2.1 HMRF Model for the Physical Field

Let $X \doteq \{X_s^t\}$ be the *hidden* field drawn from an alphabet $\mathcal{X}$, and $Y \doteq \{Y_s^t\}$ be the process observed by the sensors, drawn from an alphabet $\mathcal{Y}$. When $\mathcal{X}$ is countably finite we say that the field is discrete, otherwise it is continuous. The superscript $t \in \{1, 2, 3 \dots\}$ is the temporal index, and the subscript $s \in \{1, 2, \dots, N\}$ is the spatial index, indicating one out of N spatially scattered sensors. The observation process is allowed to be any kind of probabilistic measurement $Y_s^t(\omega) = f(X_s^t(\omega))$, where $\omega \in \Omega$ is an element drawn from the underlying probability sample space, which implies that the following *conditional independence property* holds:

$$(Y_s^t \perp (X_{s'}^{t'}, Y_{s'}^{t'}))|X_s^t \;\; \forall (s', t') \neq (s, t). \tag{2.1}$$

For example, the observation could be contaminated by an additive clutter,

$$Y_s^t = X_s^t + V_s^t, \;\; \forall s, t, \tag{2.2}$$

where $\{V_s^t\}$ is a Gaussian process statistically independent of the field $\{X_s^t\}$, whose samples in space and time are independently and identically distributed with zero mean

and variance σ^2.

Let us define

$$X^t \doteq [X_1^t, X_2^t, \ldots, X_N^t]^T, \tag{2.3}$$

$$\bar{X}^t \doteq [(X^1)^T, (X^2)^T \ldots, (X^t)^T]^T, \tag{2.4}$$

with analogous definitions for Y^t and $\bar{Y}^t$. We will sometimes denote $\bar{X}^t$ simply as X, and refer to it as a frame (of $N \times t$ random variables).

Now suppose that $\{X^t\}$ forms a homogeneous temporal Markov chain, with transition probabilities $P(x^t|x^{t-1})$. Furthermore, assume that the MC has a fully connected trellis. This is not a serious restriction since most natural models *are* fully connected or can be reasonably modeled to be so by assigning small probabilities to 'prohibited' transitions. The MC is therefore irreducible and aperiodic. Let us further assume that it has a unique stationary distribution $\pi(\cdot)$ (this is guaranteed in case of a finite field by the 'fundamental theorem' of MCs [144]). Let us define the *meta-state* as $Z^t \doteq [(X^t)^T, (X^{t-1})^T]^T$, and indirectly specify the transition probabilities of the HMM as

$$P(x^t|x^{t-1}) \doteq \frac{Q([x^{t^T}, x^{t-1^T}]^T)}{q(x^{t-1})}. \tag{2.5}$$

Here $Q(z)$ is a suitable (non-unique) time-invariant strictly positive exponential distribution [114, 136, 137, 138, 142] on the alphabet of the meta-state $\mathcal{X}^{2N}$ given by

$$Q(z|\gamma) = \exp\left\{\gamma^T b(z) - \Psi(\gamma)\right\}, \tag{2.6}$$

and

$$q(x^{t-1}) \doteq \sum_{x^t} Q([x^{t^T}, x^{t-1^T}]^T) \tag{2.7}$$

is its marginal. In the case of continuous fields $Q(\cdot), q(\cdot)$ are densities and the summation in the RHS of equation (2.7) is interpreted as an integral. In equation (2.6) $\gamma \in \mathbb{R}^M$ is the *parameter* of the model and $b : \mathcal{X}^{2N} \to \mathbb{R}^M$ is a set of M affinely independent *basis functions*. $\Psi(\gamma)$ is a normalization constant called the *log-partition* function, given by

$$\Psi(\gamma) = \log\Big(\sum_z \exp\left\{\gamma^T b(z)\right\}\Big). \tag{2.8}$$

In deference to the origin of such models in statistical physics (the Gibbs distribution), the quantity $-(\gamma^T b(z) - \Psi(\gamma))$ is called the *energy* of the configuration z. Hence the model assigns a low probability to configurations with high energy and vice versa.

Note that each basis function $b_i(z)$, $i = 1, 2, \ldots, M$ is not necessarily explicitly dependent on all the components of the vector z. Similarly, not all the active components correspond to physically distinct sites. This is best clarified by an example. Suppose

$$b_1([x_1^t, x_2^t, x_3^t, x_1^{t-1}, x_2^{t-1}, x_3^{t-1}]^T) = x_1^t x_2^t x_2^{t-1}.$$

Then only the first, second and fifth components are active, and only the first and second sensor sites are active (since both the second and fifth components correspond to site number two). Let $\mathcal{A}_i \subset \{1, 2, \ldots, 2N\}$ denote the subset of *active components* and $\mathcal{B}_i \subset \{1, 2, \ldots, N\}$ denote the subset of *active sites* in basis function number i. Then in the above example $\mathcal{A}_1 = \{1, 2, 5\}$ and $\mathcal{B}_1 = \{1, 2\}$. With this convention in place, now consider the following definition, which plays a crucial role in the design of tractable algorithms:

Definition 1 *The statistical neighborhood of a sensor site s is defined as the subset of sites $\{1, 2, \ldots, N\} \supseteq \Gamma_s = \{\cup_{i:s\in\mathcal{B}_i}\mathcal{B}_i\}$. The radius of interaction r_s of site s, is defined as the Euclidean distance from s to the farthest site in Γ_s. The radius of interaction of*

the model is defined as $r = \max_{s\in\{1,2,\ldots,N\}} r_s$.

2.2 Sparsity and Localization

The model description till now has still been fairly general. In fact there is no loss of generality at all if we are considering finite fields, because it is known that every positive probability mass function on the alphabet $\mathcal{X}^{2N}$ can be written in the form of equation (2.6). However, to facilitate the development of tractable algorithms, we will now assume that the field satisfies two additional constraints, namely:

1. *Sparsity*: The model has only $M = O(N)$ basis functions.

2. *Localization*: The radius of interaction is bounded by a constant ρ, invariant w.r.t. N, the size of the WSN.

Assumption 1 is typically made by default in literature [15, 137, 138], in order to avoid the 'curse of dimensionality'. Assumption 2 needs to be made in distributed applications to ensure that the communication load on each mote remains bounded even as the network scales, which is a prerequisite for scalability. It is a reasonable assumption provided the network scales by increasing the spatial area of deployment while keeping the *spatial density of motes roughly constant*.

In addition to the above restrictions, we may some times also make an assumption of *mild interactions*, namely $\|\gamma\|_\infty \leq \Gamma_{max}$, where Γ_{max} is some small number. This is a more specialized constraint that allows Markov Chain Monte Carlo (MCMC) methods like Gibbs Sampling to be terminated early [136, 137, 138]. However it can be be relaxed when we do not intend to use MCMC techniques, such as in Chapters 3 and 4.

In terms of graphical models like MRFs, the above restrictions are equivalent to saying that the graphical model is sparsely connected, has a topology mirroring the

structure of physical proximities among the motes, and that the edge potentials are bounded and small.

Fortunately, the above restrictions are far from artificial. Many natural phenomena of interest like diffusions and epidemics are indeed governed by sparse models [134]. Similarly, localized interactions are typically the rule rather than the exception, because the power of physical interactions (electromagnetic, acoustic or bio-chemical) falls off rapidly with distance. Finally, mild interactions are also common because un-modeled 'nuisance' processes usually preclude strong couplings [142][145].

2.3 Example: The Boltzmann Field

An important special case of the above described model is the Boltzmann field. In this case $\mathcal{X} = \{+1, -1\}$ and the basis functions are *linear or quadratic* (i.e. polynomials of degree one or two). Hence this model is also called a *pair-wise* Ising model. We can therefore write

$$Q(z|\theta, W) \doteq \exp\left\{z^T\theta + \frac{1}{2}z^T W z - \Psi(\theta, W)\right\}, \tag{2.9}$$

where $\theta \in \mathbb{R}^{2N}$, and $W \in \mathbb{R}^{2N\times 2N}$ is by convention a symmetric matrix with zeros along the diagonal. The imposition of symmetry is consistent since every quadratic form can be written with a symmetric matrix. Similarly, the zero-diagonal is consistent since $a^2 = 1$ identically for any $a \in \mathcal{X}$. This means that the degrees of freedom in the matrix W are not $2N \times 2N$ but limited to a maximum of $2N(2N-1)/2$. To see the correspondence with the general form in equation (2.6), note that $\gamma \equiv (\theta, W)$, the first $2N$ elements of γ are $\theta_1, \ldots, \theta_{2N}$, and the next $M - 2N$ elements are the $M - 2N$ non-zero entries in the upper-triangular portion of matrix W, listed in some convenient order. Similarly the first $2N$ components of $b(z)$ are $z_1, \ldots, z_{2N}$, and the next $M - 2N$ components are the corresponding second order terms of the form $z_i z_j$, $i \neq j$. The constraint $M = O(N)$

implies that W is a sparse matrix. Similarly, the constraint that the basis functions are physically localized implies that W should have a structure that mirrors the topology of the WSN. Note that the parameterization in this example is minimal by construction, hence this exponential family is *regular* [114].

Let us further write the parameters θ, W in partitioned forms as follows:

$$W \doteq \begin{pmatrix} W_s & G \\ G^T & W_s \end{pmatrix}, \qquad \theta \doteq \begin{pmatrix} \theta_s \\ \theta_s \end{pmatrix}, \tag{2.10}$$

where W_s is a symmetric, real $N \times N$ matrix with a zero diagonal, and $\theta_s \in \mathbb{R}^N$. Expanding equation (2.9) under these choices makes it clear that basis functions of the form $x_s^t x_{s'}^{t-1}$ are weighted with coefficients exclusively from G. Thus matrix $G \in \mathbb{R}^{N \times N}$ determines the *temporal* memory of the process while matrix W_s specifies its *spatial memory*. Note that the chain will be i.i.d. if $G = 0$. In general G need not be symmetric. However, if it is symmetric we have the following lemma:

Lemma 1 *If $G = G^T$, then the MC $\{X^t\}$ is time reversible, with $q(x^t)$ (defined in equation (2.7)) as its unique stationary distribution, and the MC $\{Z^t\}$ is time reversible with $Q(z^t|\theta, W)$ (defined in equation (2.9)) as its unique stationary distribution.*

The proof is given in Appendix B.1.

The Boltzmann model is well suited to modeling discrete events like the generation and dispersion of plumes. Let $X_s^t = +1$ ($X_s^t = -1$) indicate that a plume covers (does not cover) the location s at time t. Assuming that $G = G^T$ and that the initial state X^0 is drawn from $q(\cdot)$, the joint distribution of a frame is given by

$$\begin{aligned} \Pr\{X = x\} &= P(x|W_s, \theta_s, G) \\ &= \frac{\prod_{k=1}^{\tau} Q([x^{k^T}, x^{k-1^T}]^T)}{\prod_{k=1}^{\tau-1} q(x^k)}, \end{aligned} \tag{2.11}$$

which cannot be simplified further. However a good approximation is given by

$$P(x|W_s, \theta_s, G) \approx c \exp\left\{x^T\Theta + \frac{1}{2}x^T\mathbf{W}x\right\}, \tag{2.12}$$

where $\Theta = [\theta_s^T, \theta_s^T, \ldots, \theta_s^T]^T$ and

$$\mathbf{W} = \begin{pmatrix} W_s & G & 0 & 0 & \ldots & 0 \\ G^T & W_s & G & 0 & \ldots & 0 \\ 0 & G^T & W_s & G^T & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & G^T & W_s & G \\ 0 & 0 & 0 & 0 & G^T & W_s \end{pmatrix}, \tag{2.13}$$

and c is a generic normalization constant. The accuracy of this approximation improves as $\|G\| \longrightarrow 0$. In our simulations in Chapter 4, the generation of the sensor data is done according to the model of equations (2.5), (2.9), and for decoding an entire frame we will use the approximation in equation (2.12) since it is amenable to efficient inference algorithms like Mean-Field decoding.

2.3.1 Example: Linear Uniform Array Measuring a Boltzmann Field

As a specific example suppose that the N sensors are located at equally spaced points on a straight line, one meter apart. Such a uniform linear array could be used to measure, for example, the presence $(+1)$ or absence (-1) of an effluent released from a chemical plant into a river. Then one must choose W_s to be a *Toeplitz* matrix. In particular, suppose that the first row of W_s is given by $[\ 0 \quad \xi_1 \quad \xi_2 \quad \xi_3 \quad 0 \quad \ldots \quad 0\ ]$, with $\xi_1, \xi_2, \xi_3, \xi_4$ a parameter. Similarly, let the temporal dependency matrix be $G = \xi_4 I$ where I denotes

an identity matrix. This implies that the plumes diffuse without convection (drift). The *statistical interaction neighborhood* of any mote in this model is thus limited to the set of motes who are not more than 3.0 meters away. Also let $\theta_s = 0$, implying that each location of the field is a-priori unbiased. Then, by tuning the parameters $\xi_1, \xi_2, \xi_3, \xi_4$, we can simulate plumes of a variety of sizes and diffusion rates.

2.4 Example: The Gauss-Markov Random Field

Another important special case arises when $\mathcal{X} = \mathbb{R}$ and the basis functions are linear or quadratic. Formally this model is identical to the Boltzmann model of equation (2.9), except that the matrix W does not have a zero diagonal but in fact is required to be negative definite. It is easy to show that the covariance matrix is given by

$$\Sigma = \mathbb{E}_{Q(z)}\left[(Z-\mu)^T(Z-\mu)\right] = -W^{-1} \tag{2.14}$$

where $\mu = \mathbb{E}_{Q(z)}[Z] = -W^{-1}\theta$ is the mean of the model. The partitioning in equation (2.10) remains applicable and so do the remarks about the reversibility of the chain. We will let $\mathcal{N}(z;\mu,\Sigma)$ denote a multivariate Gaussian distribution parameterized in the usual way with the *mean* μ and *covariance* Σ, while $\mathcal{N}_C(z;\theta,W)$ will denote a multivariate Gaussian distribution parameterized as an exponential density with *bias* θ and *precision* W.

2.5 Information Geometry

As we remarked in the introduction, exponential models have a very rich geometric structure. In order to be self-contained for the development in Chapters 3, 4, 5, and 6, below we give a set of useful information-geometric definitions, many of them specialized

to the case of Boltzmann fields. More details on the information geometry of exponential models can be found in [146, 147, 148, 149]. Let $Z \sim Q(x|\gamma)$, where $Q(\cdot|\gamma)$ is formally defined as in equation (2.6), and let

$$U = Z + R \tag{2.15}$$

where $R \sim \mathcal{N}(v|0_N, \sigma^2 I_{2N\times 2N})$. Let

$$\pi(u|\gamma) = \sum_z Q(z|\gamma)P(u|z) \tag{2.16}$$

denote the unconditional distribution of the observation U parameterized by γ. Let

$$\eta_\gamma \doteq \nabla_\gamma \Psi(\gamma), \tag{2.17}$$

$$F_\gamma \doteq \nabla^2_\gamma \Psi(\gamma). \tag{2.18}$$

It is known [114] that η_γ and F_γ are respectively the mean and covariance in the directly observed model,

$$\begin{aligned} \eta_\gamma &= \mathbb{E}_{Q(z|\gamma)}\left[b(Z)\right], \\ F_\gamma &= \mathbb{E}_{Q(z|\gamma)}\left[(b(Z)-\eta_\gamma)(b(Z)-\eta_\gamma)^T\right]. \end{aligned} \tag{2.19}$$

The direct and indirect observation log-likelihoods are respectively defined as

$$l_\gamma(Z) \doteq \log Q(Z|\gamma), \tag{2.20}$$

$$L_\gamma(U) \doteq \log \pi(U|\gamma), \tag{2.21}$$

the respective *scores* are

$$s_\gamma(Z) \doteq \nabla_\gamma l_\gamma(Z), \tag{2.22}$$

$$S_\gamma(U) \doteq \nabla_\gamma L_\gamma(U), \tag{2.23}$$

and the respective Fisher informations are [5, 114, 146]

$$F_\gamma^Z \doteq \mathbb{E}_{Q(z|\gamma)}\left[s_\gamma(Z)s_\gamma(Z)^T\right] = -\mathbb{E}_{Q(z|\gamma)}\left[\nabla_\gamma^2 l_\gamma(Z)\right], \tag{2.24}$$

$$F_\gamma^U \doteq \mathbb{E}_{\pi(u|\gamma)}\left[S_\gamma(U)S_\gamma(U)^T\right] = -\mathbb{E}_{\pi(u|\gamma)}\left[\nabla_\gamma^2 L_\gamma(U)\right]. \tag{2.25}$$

It is straightforward to show that $F_\gamma^Z = F_\gamma$. Also note that when $\sigma^2 = 0$, we have direct observation $U = Z$, and therefore $S_\gamma(U) = s_\gamma(Z)$ and $F_\gamma^U = F_\gamma^Z$. Since F_γ^U is a continuous function of σ^2, we have $\lim_{\sigma^2\to 0} F_\gamma^U = F_\gamma^Z$, where the convergence is element-wise.

An important property of regular (i.e. minimal) exponential families is that $\Psi(\gamma)$ is *strictly* convex [146] in γ, and hence the log-likelihood $l_\gamma(\cdot)$ is strictly concave in γ for any X. This means that $\forall\gamma, F_\gamma^Z = F_\gamma > 0$.

Suppose that $Q(\cdot|\gamma^*)$ is the 'truth model', and we make n *independent* drawings from this distribution, which are packed into $\bar{Z}^n$ (cf. notation of equation (2.3)). Then the *Maximum Likelihood (ML) estimate* of γ^* based on the corresponding n *i.i.d.* observations packed into $\bar{U}^n$ is

$$\hat{\gamma}_{ML}(n) = \max_{\gamma\in\Gamma_{\text{feasible}}} \log P(\bar{U}^n|\gamma). \tag{2.26}$$

It is known [123] that $\hat{\gamma}_{ML}(n)$ is asymptotically (as $n \to \infty$) unbiased and efficient i.e. achieves the Cramér-Rao lower bound (CRLB)

$$\mathbb{E}\left[(\hat{\gamma}_{ML}(n) - \gamma^*)\,(\hat{\gamma}_{ML}(n) - \gamma^*)^T\right] \geq \frac{1}{n}(F_{\gamma^*}^U)^{-1}. \tag{2.27}$$

2.5.1 Projective Geometry of Boltzmann Fields

For Boltzmann fields ($\mathcal{X} = \{+1, -1\}$), we can define an indirect-observation *log-likelihood ratio*

$$h_i(U) = \begin{cases} \frac{1}{2}\log\frac{P(U_i|Z_i=+1)}{P(U_i|Z_i=-1)} = \frac{U_i}{\sigma^2}, & i = 1, 2, \ldots, 2N \\ 0, & i = 2N+1, 2N+2, \ldots, M. \end{cases} \tag{2.28}$$

Then it is easily shown that [114]

$$s_\gamma(Z) = b(Z) - \eta_\gamma \tag{2.29}$$

$$S_\gamma(U) = \mathbb{E}_{P(z|U,\gamma)}\left[s_\gamma(Z)\right] = \eta_{\gamma+h(U)} - \eta_\gamma. \tag{2.30}$$

Several well known algorithms for statistical inference and estimation, like Belief Propagation (BP), Mean Field Decoding (MFD) and Expectation Maximization (EM) can be elegantly described in terms of the projective geometry on statistical manifolds. We will see examples of this in Chapter 3. Hence we will now give a minimal set of definitions and two important theorems which play a key role in information geometry.

Definition 2 *The statistical manifold* $\aleph$ *of the model in equation (2.9) is defined as*

$$\aleph \doteq \left\{ Q(z;\theta,W) \middle| \begin{array}{c} \theta \in \mathbb{R}^{2N}, \\ W \in \mathbb{R}^{2N\times 2N}, \\ W = W^T, \mathrm{diag}(W) = 0 \end{array} \right\}. \tag{2.31}$$

Definition 3 *The sub-manifold* $M_W \subset \aleph$ *corresponding to a particular value of* W *is defined as*

$$M_W \doteq \{Q(z;\theta,W)|\theta \in \mathbb{R}^{2N}\}. \tag{2.32}$$

Note that the special sub-manifold M_0 (corresponding to $W = 0$) consists of distributions on Z such that the components of Z are *independently distributed*. M_0 is an important sub-manifold that plays a crucial role in understanding exact and approximate inference techniques.

Definition 4 *The Kullback-Leibler (KL) divergence of $p(z)$ relative to $Q(z)$, where $Q(z), p(z) \in \aleph$, is defined as*

$$D(Q(z)||p(z)) \doteq \mathbb{E}_{Q(z)}\left[\log\left(\frac{Q(Z)}{p(Z)}\right)\right]. \tag{2.33}$$

Definition 5 *The m-projection from $Q(z) \in \aleph$ to a sub-manifold $M \subset \aleph$ is denoted by $\Pi_M^m \circ Q(z)$ and defined as*

$$\Pi_M^m \circ Q(z) \doteq \underset{p(z)\in M}{\operatorname{argmin}} D(Q(z)||p(z)). \tag{2.34}$$

Definition 6 *The e-projection from $Q(z) \in \aleph$ to a sub-manifold $M \subset \aleph$ is denoted by $\Pi_M^e \circ Q(z)$ and defined as*

$$\Pi_M^e \circ Q(z) \doteq \underset{p(z)\in M}{\operatorname{argmin}} D(p(z)||Q(z)). \tag{2.35}$$

Theorem 1 *[147] Let M be an e-flat sub-manifold in $\aleph$, and let $Q(z) \in \aleph$. Then the m-projection $\Pi_M^m \circ Q(z)$ is unique and given by a point in M such that the m geodesic connecting $Q(z)$ and $\Pi_M^m \circ Q(z)$ is orthogonal to M at this point in the sense of the Riemannian metric due to the Fisher information matrix.*

Theorem 2 *[147] The projection $\Pi^m_{M_0} \circ Q(z;\theta,W)$ constitutes a marginalization of the distribution $Q(z;\theta,W)$. That is,*

$$\mathbb{E}_{\Pi^m_{M_0} \circ Q(z;\theta,W)}[Z] = \eta(\theta,W).$$

Proofs can be found in [147][146].

3 Distributed Filtering

3.1 Introduction

As we discussed in the Chapter 2, the WSN usually remotely monitors a *physical field* that may have considerable spatio-temporal dependencies [150], and communicates relevant data to a distant fusion center (FC) where it is interpreted and used by higher level applications. Consider a WSN where the motes quantize their measurements with a sufficiently large number of bits, compress the resulting bit-streams with a distributed Slepian-Wolf type code [151], and directly communicate the resulting data to the FC. Let us call this the 'Fusion First' (FF) approach to data extraction. When the sensor measurements are contaminated by a substantial amount of clutter[1], the data compression is not very effective, and the communication of the resulting voluminous data stream rapidly exhausts the batteries of the motes, resulting in short network lifetimes. This is simply a consequence of the law of wireless transmission with freely propagating electromagnetic waves, which says that

$$P_{RX} = \alpha l^{-\kappa} P_{TX}, \tag{3.1}$$

[1]Note that such clutter subsumes not just the thermal noise in the sensor electronics but also *physical* sources of perturbation; for example, errors in sea-level measurements due to waves or wakes of ships.

where P_{TX} is the transmitted power, P_{RX} is the received power, l is the distance of the transmission, α is the link 'gain' (which includes the gains of the antennae, if any), and κ is a path loss exponent that can range from 1.6 to 6 [93, Table 4.2]. For error-free data communication, P_{RX} needs to be sufficiently large relative to the thermal noise threshold at the receiver's front-end amplifier, and this determines P_{TX} immutably.

In the light of this discussion, we need to ask a more fundamental question: Is it necessary to communicate all the sensor data to the fusion center? For example, in the application of plume detection, only a few summary statistics are needed by the user, namely the identity and locations of the sensors where the plume has been reliably detected. It is immaterial to the user whether these statistics are calculated by the FC or by the WSN itself. Clearly then, the option of doing inference within the network, which we call the 'Inference First' (IF) paradigm, has the potential to vastly reduce the communication load.

In this chapter we intend to demonstrate an efficient distributed algorithm for statistical parametric inference within the WSN ('in-situ') such that the energy expended on the inference procedure is much smaller than the energy saved by eliminating the need to communicate all sensor data to the fusion center. The problem of distributed detection by statistical parametric methods has been considered recently by other researchers. For example [16, 18] formulate it as an M-ary hypothesis testing problem with conditionally independent observations, while [22] considers a specialized application involving localization and tracking of a diffusive source using a distributed sequential Bayes' estimator. Similarly, [15] considers a spatial process modeled via a Hidden Markov Random Field (HMRF). We will consider a more general model, that subsumes these special cases, where the phenomena of interest is treated as a full-fledged spatio-temporal random process that needs to be estimated without latency.

As we saw in Chapter 2, even for a binary field ($\mathcal{X} = \{+1, -1\}$), in general the full

specification of the hidden Markov model requires $O(2^{2N})$ parameters (all the transition probabilities), where N is the number of motes in the WSN. We avoid this curse of dimensionality by exploiting the fact that natural fields have strong dependencies in space and time. In particular we specify the HMM transition probabilities in terms of a joint distribution on the state of the Markov Chain (MC) at two consecutive time instants. This distribution is allowed only $O(N)$ parameters, and moreover the interactions are required to be localized in space and time (cf. Chapter 2), which enables the development of scalable algorithms.

Our goal is optimal filtering, i.e. optimal inference of the value of the hidden process at each point in space at a given time epoch, based on *all* the history of observations till that epoch. In Section 3.2, we will explain why optimal filtering of the observed process via the well-known forward filter of Baum et al. [117] is infeasible for practical WSNs. Then we present tractable algorithms for approximate filtering based on the idea of iterated decoding (i.e. iterated marginalization) of the model. In this context, we compare and contrast various marginalization techniques suitable for the *broadcast* nature of the WSN, namely, Gibbs Sampling (GS) [127], Mean Field Decoding (MFD) [152], Iterated Conditional Modes (ICM) [15] and Broadcast Belief Propagation (BBP) [153]. We discuss various properties of these algorithms like conditions for approaching optimality, energy efficiency, robustness and scalability. In Section 3.3 we make a head to head comparison of the energy efficiency of the IF and FF paradigms, and derive a lower bound on the achievable energy gain. In Section 3.4 we provide results of extensive simulations of the proposed approximated HMM filter. We provide numerical evidence about the respective regimes of good performance for the various marginalization algorithms like GS, MFD, ICM and BP, and explain the significance of these results in the light of the information geometry of densely cyclic graphical models. We also show, using practical values of network parameters, that when the distance of the sensor array

from the fusion center exceeds a certain threshold the IF approach gives a very significant reduction in energy consumption relative to the FF approach. In Section 3.5 we present a summary and conclusions.

For simplicity, in the rest of this chapter we will specialize to the Boltzmann field (cf. Chapter 2.3), while noting that all the major results remain applicable to the general case of the exponential model described in Chapter 2.1. A discussion of such generalizations is provided in Section 3.4.4.

3.2 Efficient In-situ Inference With Approximate Filtering

The delay-free filtering problem can be defined as the optimal estimation of X_s^t, for every time t and each site s, based on $\bar{Y}^t$. The optimality criterion could be Minimum Error Probability (MEP) or Minimum Mean Squared Error (MMSE). In either case, we need to calculate the a-posteriori marginal of X_s^t, which forms a sufficient statistic.

In Section 3.2.1 we will first demonstrate that the optimal HMM filter is intractable. Then we will derive, in two steps, a tractable approximation of the filter. First, in Section 3.2.2, we will approximate the propagated p.m.f. with a fully factorized product form, having only N parameters. The resulting algorithm, which still involves solving an NP-hard marginalization problem, is analyzed and its properties are discussed. Then in Section 3.2.3 we will further describe how to calculate the marginals approximately with efficient linear complexity algorithms suited to the WSN constraints.

3.2.1 Optimal Filtering is Intractable

Using the Markovian independence properties of the chain, one can write the well-known 'forward' filter equation (Bayesian recursion) [117]:

$$p^t(x^t) \doteq c\, P(x^t, \bar{y}^t) = c\, P(y^t|x^t) \sum_{x^{t-1}} P(x^t|x^{t-1}) p^{t-1}(x^{t-1}), \tag{3.2}$$

where c is a normalization constant[2]. $p^t(x^t)$ is known as the propagated a-posteriori p.m.f., and the finite dimensional update in the above equation is known as the *HMM filter*. The a-posteriori marginal of X_s^t is then given by:

$$\begin{aligned} p_s^t(x_s^t) &\doteq \sum_{x_{\neq s}^t} p^t(x^t) \\ &= c \sum_{x_{\neq s}^t, x^{t-1}} P(y^t|x^t) P(x^t|x^{t-1}) p^{t-1}(x^{t-1}). \end{aligned} \tag{3.3}$$

Then the a-posteriori conditionally most likely value, $\text{argmax}_{x_s^t} p_s^t(x_s^t)$, is the MEP estimate of X_s^t, while the conditional expectation $\mathbb{E}_{p_s^t(x_s^t)}[X_s^t]$ is the MMSE estimate. Although this sounds straightforward, the implementation of the exact HMM filter is in fact infeasible even for a moderately large number of sensors like $N = 100$. This is because the propagated p.m.f. $p^t(x^t)$ has $2^N - 1$ degrees of freedom, and the transition probability matrix $P(x^t|x^{t-1})$ has $2^{2N} - 2^N$ degrees of freedom, so their direct storage and manipulation is impossible for all but small N. Additionally, in sensor networks a great premium is placed on energy usage, and hence on communication. So any solution that requires collection of all raw data Y at a centralized location is undesirable. In the following, we will propose a distributed scalable approximation of the HMM filter. While the approximation is suboptimal, we will demonstrate by analysis and simulations

[2]In the rest of this section, we will let c denote a generic normalization constant.

that the loss in performance w.r.t. to optimal filtering is not serious provided the SCR is not too small.

3.2.2 Approximation I: Product Form Representation of the Propagated P.M.F.

Let us rewrite equation (3.3) in terms of $Q([x^{tT}, x^{t-1T}]^T)$ as

$$p_s^t(x_s^t) = c \sum_{x_{\neq s}^t, x^{t-1}} P(y^t|x^t) Q([x^{tT}, x^{t-1T}]^T) \frac{p^{t-1}(x^{t-1})}{q(x^{t-1})}. \tag{3.4}$$

We interpret this equation as follows: $P(y^t|x^t) = \prod_{s=1}^N P(y_s^t|x_s^t)$ provides a prior for X^t, while $\frac{p^{t-1}(x^{t-1})}{q(x^{t-1})}$ provides a prior for X^{t-1}. Then, optimal inference is obtained by the marginalization of $Q(x^t, x^{t-1})$ with these priors. Let us define

$$q_s(x_s^{t-1}) \doteq \sum_{x_{\neq s}^{t-1}} q(x^{t-1}) \ , \quad \forall s \in \{1, 2, \ldots, N\} \ , \tag{3.5}$$

and, letting $\sim$ indicate equality up to a normalization constant, make the following definitions for canonical parameters (log-likelihood ratios)[3] α_s^t, β_s and h_s^t:

$$\begin{aligned} e^{x_s^{t-1}\alpha_s^{t-1}} &\sim p_s^{t-1}(x_s^{t-1}), \\ e^{x_s^t h_s^t} &\sim P(y_s^t|x_s^t), \\ e^{x_s^{t-1}\beta_s} &\sim q_s(x_s^{t-1}). \end{aligned} \tag{3.6}$$

[3] A distribution $P(x)$ on $x \in \{+1, -1\}$ can always be represented as $P(x) = e^{x\delta - \Psi}$ where $\delta = \frac{1}{2}\log\left(\frac{P(x=+1)}{P(x=-1)}\right)$ and $\Psi = \log(e^{\delta} + e^{-\delta})$. δ is called the 'canonical' parameter of the distribution.

Now, crucially, we make the following approximations:

$$\begin{aligned} p^{t-1}(x^{t-1}) &\approx c \exp\left(\sum_s \alpha_s^{t-1} x_s^{t-1}\right), \\ q(x^{t-1}) &\approx c \exp\left(\sum_s \beta_s x_s^{t-1}\right). \end{aligned} \tag{3.7}$$

In each case, we are approximating a distribution on the state space of the MC by a product of its marginals. (Note that these approximations specialize to the *Mean Field approximation* [154] if the marginals are calculated exactly. As we shall see later in Section 3.2.3, we can also use the Mean Field technique for tractable approximate marginalization. Our proposal has analogies to the procedure of tractable inference of stochastic processes suggested by [155, 156] in that we use factored approximate representations of beliefs on the state space, though at the same time we are also endevor to maintain distributedness and scalability of the algorithm.) Under these approximations, equation (3.4) becomes (recall that $z = [(x^t)^T, (x^{t-1})^T]^T$)

$$p_s^t(x_s^t) \approx c \sum_{x_{\neq s}^t, x^{t-1}} \exp\left\{ \begin{array}{c} z^T \left(\theta + \begin{bmatrix} h^t \\ \alpha^{t-1} - \beta \end{bmatrix}\right) \\ +\frac{1}{2} z^T W z \\ -\Psi\left(\theta + \begin{bmatrix} h^t \\ \alpha^{t-1} - \beta \end{bmatrix}, W\right) \end{array} \right\}. \tag{3.8}$$

Suppose we have access to a subroutine $\mathcal{M}_1(\theta, W)$ that returns a vector in $\mathbb{R}^N$ that contains the canonical parameters of the marginals of the model in equation (2.9) w.r.t. $X_s^t, s = 1, 2, \ldots N$. Similarly, let a subroutine $\mathcal{M}_2(\theta, W)$ return the marginals w.r.t

X_s^{t-1}. Then we can implement the approximation in equation (3.8) via the following algorithm (the text following `//` are comments):

1. $\beta \leftarrow \mathcal{M}_2(\theta, W)$ `// Calculate and store the marginals of` $q(\cdot)$ `as per equation (3.5).`
2. $t \leftarrow 0, \alpha^t \leftarrow \beta$ `// Initialize the propagated p.m.f. marginals.`
3. $t \leftarrow t + 1$
4. $\alpha^t \leftarrow \mathcal{M}_1\left(\theta + \begin{bmatrix} h^t \\ \alpha^{t-1} - \beta \end{bmatrix}, W\right)$ `// Update propagated p.m.f. as per equation (3.8).`
5. Go to step 3 `// Repeat the update for the next time epoch.`

This algorithm is a non-linear discrete time *deterministic* dynamical system in the state variables α^t, starting from the initial state $\alpha^0 = \beta$, and driven by the *random input sequence* h^t. The algorithm has some desirable properties that are stated in the following lemmas.

Lemma 2 *If the chain is reversible and the observation process Y is not available, β is a stationary point of the algorithm. It is locally asymptotically stable provided $||G||$ is sufficiently small.*

Lemma 3 *The algorithm provides the exact a-posteriori marginals, if the chain is initialized for $t = 0$ with its stationary distribution $\pi(\cdot)$, and* any *of the following conditions is true:*

1. *$G = 0$. (Hence the chain X^t is i.i.d.).*
2. *$W_s = 0$ and G is diagonal. (Hence each X_s^t, $s = 1, 2, \ldots, N$ evolves as an independent MC).*

3. $h^t = 0$ *for all* $t \geq 0$*, and the chain is time reversible.*

Proofs are given in Appendices C.1 and C.2. The above lemmas also suggest that when the spatio-temporal interactions are not too strong we can expect to have only a modest degradation in performance relative to the exact HMM filter. Also, since a finite aperiodic irreducible chain is geometrically ergodic, the initial distribution is quickly forgotten and does not affect long term performance. Simulation results given in Section 3.4 confirm these predictions.

3.2.3 Approximation II: Marginalization by Iterated Decoding

Although we have eliminated the need to store and manipulate the entire $O(2^N)$ propagated p.m.f. $p^t(x^t)$ in the approximate filter presented in the previous section, our algorithm still needs to compute the marginals of a distribution $Q([x^{t^T}, x^{t-1^T}]^T; \theta, W)$ of the form defined in equation (2.9), using subroutines $\mathcal{M}_1, \mathcal{M}_2$. We will now discuss exact marginalization and its tractable approximations, and their geometric interrelationships. (Please refer to the definitions made in Section 2.3.)

During the discussion of various marginalization algorithms presented below, it will be useful to refer to Figure 3.1, which demonstrates their information geometry. In the figure, Q is the model we wish to marginalize, and is of the form given by equation (2.9). The sub-manifold M_0 is a special sub-manifold of distributions in $\aleph$ corresponding to the case $W = 0_{2N \times 2N}$. This implies that every point in M_0 is a distribution on $Z^t = (X^t, X^{t-1})$ such that all the components of Z^t are *independent* of each other. We will soon see that all the marginalization algorithms are projections of some sort onto M_0. Note that M_0 can be unambiguously parameterized by the $2N$ canonical parameters (see footnote 3) corresponding to the respective components of Z^t. Thus, each algorithm produces a set of $2N$ canonical parameters, which correspond either to

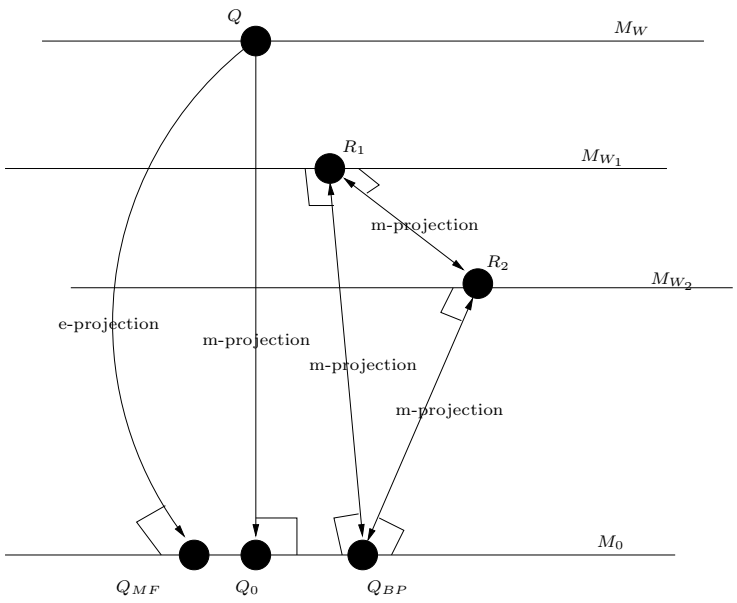

Figure 3.1: The geometry of exact and approximate marginalization.

the exact marginal probabilities or approximations thereof.

True Marginalization

The true marginals can be represented by a point $Q_0 \in M_0$ specified by the canonical parameter vector $a \doteq [\mathcal{M}_1(\theta, W)^T, \mathcal{M}_2(\theta, W)^T]^T$. That is

$$M_0 \ni Q_0(z) \doteq Q(z; a, 0) = c \prod_{i=1}^{2N} e^{z_i a_i}. \tag{3.9}$$

From Theorem 2 we know that [146, 147]

$$Q_0(z) = \Pi^m_{M_0} \circ Q(z; \theta, W). \tag{3.10}$$

That is, exact marginalization is an m-projection from $Q(z; \theta, W)$ to M_0, as illustrated in Figure 3.1.

Exact marginalization is NP-hard for generic cyclical models, since expectations have

to be calculated by performing summations over all the configurations in the space $\{-1,+1\}^{2N}$. Hence we must consider linear complexity distributed algorithms that can calculate a good approximation of the marginals. Furthermore, they should make optimal use of the *broadcast* nature of the wireless network. Four popular algorithms that satisfy these requirements are: Gibbs Sampling (GS) [127], Mean Field Decoding (MFD) [152], Iterated Conditional Modes (ICM) [15] and Broadcast Belief Propagation (BBP) [153].

Approximate Marginalization by Gibbs Sampling (GS)

GS is a *stochastic* method that yields an arbitrarily accurate estimate of the true marginals, provided sufficiently many extensions are made of a specially constructed MC $\{U^t\}$. The state space of $\{U^t\}$ is $\{+1,-1\}^{2N}$. In the simplest construction, from a current state u, outgoing transitions are allowed only by flipping at most one component of u. The probability of making a transition by flipping the i_{th} component is given in terms of the 'full conditionals' derived from $Q(z;\theta,W)$ as follows:

$$\begin{aligned}\Pr\{(u_1,\ldots,u_i,\ldots,u_{2N}) \rightarrow (u_1,\ldots,\tilde{u}_i,\ldots,u_{2N})\} \\ = c\,\exp\left(\tilde{u}_i[\theta_i + \sum_{j\neq i} W_{ij}u_j]\right).\end{aligned} \tag{3.11}$$

It is well known [127, 135] that such a Markov Chain 'Monte Carlo' (MCMC) has the invariant distribution $Q(u|\theta,W)$. Furthermore, an ergodic theorem holds that assures us that the relative frequencies of occurrence converge almost surely to $Q(u)$, and the empirical average at each site converges almost surely to the expectation under the true site marginal. Thus, as long as we make sufficiently many extensions of the chain to allow ergodicity to take effect, we can approach the true marginals and marginal expectations as closely as we desire. Geometrically speaking, with a sufficiently large sample, GS can

produce an estimate that lies in an arbitrarily small neighborhood of Q_0 in M_0.

Distributed Implementation: The Gibbs Sampler update is eminently suited to distributed inference. A mote s merely needs to maintain the state variables U_s and U_{s+N}, which correspond to the field variables X_s^t and X_s^{t-1}. (We will refer to mote s as the 'owner' of variables U_s and U_{s+N}). Once it receives the values of all the other local state variables from its *statistical neighborhood* (see Definition 1), it can compute new realizations for U_s and U_{s+N}, according to the distribution in equation (3.11), and broadcast them to all the sites in its statistical neighborhood. This action is repeated all across the network in a random or pseudo-random order, such that each site performs a resampling sufficiently many times (which we will call the number of iterations, n_{iter}). After sufficiently many iterations the site average of U_s is read off as an approximation of the marginal of X_s^t.

While it is scalable, GS is still relatively expensive in terms of the *communication* load because n_{iter} may be required to be very large. Though the chain is *geometrically ergodic*, we still need a relatively large number of extensions in the stationary phase to obtain an accurate time average. This is especially true when the elements in W have large absolute values, which makes the chain *weakly mixing*. Hence the Gibbs Sampler should be considered as a candidate for inference only if other approximate algorithms are found to give a large degradation as compared to optimal inference.

Approximate Marginalization by Mean Field Decoding (MFD)

Another type of approximate inference is the Mean Field (MF) inference. A first-order MF inference is given by

$$Q_{MF} = \Pi_{M_0}^{e} \circ Q(z; \theta, W), \tag{3.12}$$

that is, an e-projection from $Q(z;\theta,W)$ to M_0, which is also illustrated in Figure 3.1. Since the KL-divergence is *not symmetric*, the MF inference Q_{MF} is in general distinct from the optimal inference Q_0, and will give some degradation in the error rate. Nevertheless it is often used as an approximation in practice because it is very easy to compute, since expectations are done with respect to the tractable distributions in M_0. The quality of approximation is good provided $\|W\|$, and hence $D(Q_{MF}||Q)$, is small enough [154]. This 'soft constraint' condition is likely to be satisfied by natural fields encountered in WSN applications. There also are some other variational methods of MF inference, like *structured MF* [157] and *mixture MF* [152], which aim is to improve the inference by projecting onto a more structured manifold than M_0. However, this also results in a faster than linear dependence of the complexity and communication load as a function of N, which makes such advanced techniques less suited for WSNs. Hence we will limit our investigation to the first-order MF technique.

MFD is an iterative method of solving the optimization problem in equation (3.12), cf. e.g. [158]. Since for any $R(z) \in M_0$, $R(z) = \exp(\zeta^T z - \Psi(\zeta,0))$ with mean $m(\zeta) \doteq \mathbb{E}_R[Z] = \tanh(\zeta)$, we have

$$\begin{aligned} &D(R||Q) = \\ &\mathbb{E}_R\left[\zeta^T z - \Psi(\zeta,0) - \theta^T z - \frac{1}{2} z^T W z + \Psi(\theta,W)\right] \\ &= \zeta^T m - \Psi(\zeta,0) - \theta^T m - \frac{1}{2} m^T W m + \Psi(\theta,W)\ , \end{aligned} \tag{3.13}$$

and hence,

$$\frac{\partial D(R||Q)}{\partial m^T} = \zeta - \theta - Wm, \tag{3.14}$$

where we used the fact that $\mathrm{diag}(W) = 0$, and the moment generating property of the log-partition function. Equating the derivative to zero we have the necessary 'Mean

Field Condition'

$$m = \tanh(\theta + Wm). \tag{3.15}$$

Note that this is a *non-linear* equation in m. It can be solved in several ways and can have multiple solutions, of which, in principle, the best one must be chosen by directly comparing the divergence. In practice however, due to complexity/communication constraints, we will implement an iterative root-finding algorithm and accept its converged solution as the MF inference, disregarding the possibility that it may only be a local optimum. A method of solution particularly suited to distributed processing is 'coordinate-wise' descent where for each i, we update m_i as

$$m_i \;\leftarrow\; \tanh(\theta_i + \sum_{j \neq i} W_{ij} m_j). \tag{3.16}$$

Distributed Implementation: The reader will notice the close similarity of the above recursion to the Gibbs Sampler update in equation (3.11). The only difference is that the 'state' of the MF 'chain' is a (deterministic) real valued mean-vector $m \in [-1, 1]^{2N}$ rather than a random vector $u \in \{+1, -1\}^{2N}$. The components m_s and m_{N+s} are physically stored and manipulated at site s. The site is updated by replacing the old value of m_i (for $i = s, s+N$), by the new *conditional mean* $\tanh(\theta_i + \sum_{j \neq i} W_{ij} m_j)$ of the distribution $\sim \exp\{u_i(\theta_i + \sum_{j \neq i} W_{ij} m_j)\}$ (rather than a *random sample* drawn according to that distribution, as is the case in the Gibbs Sampler). This makes the MFD algorithm deterministic, while GS is a stochastic method. Clearly then, there is no question of ergodicity and large sample size; we merely need to ensure that convergence (in the standard sense) has taken place before we read off the value of m_i as the MF approximation of the true marginal expectation. As we will see in Section 3.4, this typically needs very few (say twenty) iterations, which compares very well with the

thousands of iterations needed for the Gibbs Sampler.

Iterated Conditional Modes (ICM)

MFD has close structural similarities to another algorithm called Iterated Conditional Modes (ICM), which has also been recently proposed for statistical inference in WSNs [15]. The main difference between the two algorithms is that while MFD broadcasts conditional *means*, ICM broadcasts conditional *modes* of the predictive likelihood. For many well-behaved continuous valued fields, the conditional mean and mode tend to be almost the same, and hence the performance of MFD and ICM is indistinguishable, in terms of quality of inference as well as convergence time. However, for a binary (or in general discrete) valued field, broadcasting the conditional mode is equivalent to broadcasting a conditional *decision*. Such a *bit-flipping* approach always shows some loss in performance relative to soft message passing as in MFD. Although the convergence behavior of ICM and MFD is identical, ICM nevertheless has the advantage that its communication cost for inter-sensor message passing is smaller, since only one bit needs to be transmitted per message, as compared to $2-4$ bits for MFD. However, as we will see in Section 3.3, this *does not* translate into a multi-fold improvement in energy efficiency. This is because, as the distance from the WSN to the FC becomes large compared to the inter-sensor distance, the asymptotic energy gain of the IF procedure relative to FF becomes *independent* of the number of bits used for inter-sensor messaging. Still, for the sake of completeness, we will include the ICM algorithm in the simulation results presented in Section 3.4.

Approximate Marginalization by Broadcast Belief Propagation (BBP)

Belief Propagation (BP) was originally proposed in [159] as an efficient procedure for *exact* marginalization of *tree-like* graphical models. But it has since been applied as an

approximate marginalization procedure to a variety of *loopy* models like Turbo Codes and Low Density Parity Check Codes [143, 160], with surprisingly good results whenever the cyclicity is not too severe. This phenomena has been analyzed in terms of variational free energy approximations [161], as well as information geometry [147].

Consider all the non-zero entries in the upper-triangular portion of W. Organize these L entries into a vector in any suitable order, and index them with the positive integer i. Let (r_i, c_i) denote the (row,column) in W corresponding to the i^{th} entry in the vector. Define matrix W_i as the matrix obtained by dropping all the entries in W except those at locations (r_i, c_i) and (c_i, r_i). Then it can be shown [147] that the BP algorithm maintains L distributions $R_i \in M_{W_i}$, $i = 1, 2, \ldots, L$, and the BP update is a discrete dynamical system governing the motion of these distribution in their respective sub-manifolds. The fixed point, Q_{BP}, of the algorithm satisfies the following geometric condition:

$$\begin{aligned} Q_{BP} &= \Pi^m_{M_0} \circ R_i, \, \forall i \in \{1, 2, \ldots L\}, \\ R_j &= \Pi^m_{M_j} \circ \Pi^m_{M_i} \circ R_j, \quad \forall i, j = 1, 2, \ldots, L. \end{aligned} \tag{3.17}$$

That is, the m-projection from all the partial-constraint sub-manifolds M_{W_i} to M_0 are required to coincide. This is illustrated in Figure 3.1. The most important consequence of this property is that *dense cyclicity* in the model (W, θ) leads to a large relative curvature of certain statistical sub-manifolds that *prevents Q_{BP} from approaching Q_0 closely* [162]. Since the WSN field tends to have numerous local interactions in feedback (like convection-diffusion processes governing plume dispersion), the structure in W has many short cycles. Hence BP may not perform as well for WSN inference as it does for decoding channel codes, especially for small SCR.

Another minor difficulty with BP is that traditionally it is formulated as a message

passing algorithm where each node communicates a separate message to each of its neighbors, in each iteration. This is not suitable for the broadcast nature of wireless networks. Fortunately, it has be shown [153] that BP can be formulated in a 'broadcast' form where a node sends the *same* modified message to all its neighbors, and they can recreate the true BP message from this incoming modified message and previous state information. In the following we will reformulate Broadcast-BP (BBP) for the binary pairwise field in terms of canonical parameters, which will help clarify its relation to the GS, MFD and ICM iterations.

Firstly, it is easy to see that $\phi_i(z_i) \sim e^{\theta_i z_i}$ and $\phi_{ij}(z_i, z_j) \sim e^{W_{ij} z_i z_j}$ are the 'potentials' [163] of a node i and an edge (ij) respectively in the Markov Random Field (MRF) of the distribution $Q(z; \theta, W)$, since we have the factorization property

$$Q(z; \theta, W) = \prod_i \phi_i(z_i) \prod_{(i,j)} \phi_{ij}(z_i, z_j). \tag{3.18}$$

In the traditional formulation of BP, at any iteration number n the estimated belief of a node i is given by

$$p_i^{(n)}(z_i) = \phi_i(z_i) \prod_j m_{ji}^{(n)}(z_i), \tag{3.19}$$

where $m_{ij}^{(n)}(z_j)$ denotes a message sent by a node i to node j during iteration n, and is generated by the *sum-product* rule [143]

$$\begin{aligned} m_{ij}^{(n)}(z_j) &= \sum_{z_i} \phi_{ij}(z_i, z_j) \phi_i(z_i) \prod_{v \neq j} m_{vi}^{(n-1)}(z_i) \\ &= \textstyle\sum_{z_i} \phi_{ij}(z_i, z_j) \frac{p^{(n-1)}(z_i)}{m_{ji}^{(n-1)}(z_i)}. \end{aligned} \tag{3.20}$$

After representing these quantities in the exponential form as

$$m_{ij}^{(n)}(z_j) \sim \exp\left\{\beta_{ij}^{(n)} z_j\right\}, \quad p_i^{(n)}(z_i) \sim \exp\left\{\alpha_i^{(n)} z_i\right\}, \tag{3.21}$$

it is easy to show that the message update becomes

$$\beta_{ij}^{(n)} = \frac{1}{2}\log\frac{\cosh(\alpha_i^{(n-1)} - \beta_{ji}^{(n-1)} + W_{ij})}{\cosh(\alpha_i^{(n-1)} - \beta_{ji}^{(n-1)} - W_{ij})}, \tag{3.22}$$

and the belief update takes the form

$$\alpha_i^{(n)} = \theta_i + \sum_j \beta_{ji}^{(n)}. \tag{3.23}$$

Distributed Implementation: Analogous to GS, MFD, and ICM, in BBP each mote s maintains ('owns') the variables α_s and α_{s+N}. Their update is very similar to the GS, MFD and ICM updates. The new value of α_i (for $i = s, s+N$) is calculated as a function of θ_i and the messages β_{ji}, according to equation (3.23). However note that a message β_{ji} is not communicated directly by the owner of j to s, the owner of i. Instead, it is calculated at the mote s as a function of α_j (the value broadcast by owner of j), and β_{ij} (the previous *outgoing* messages from node i to j). Symmetrically, the outgoing message from i to j, β_{ij}, is *never actually sent* individually, only its record is maintained by owner of i, for use in the next iteration. Thus the value broadcast by owner of i is always $\alpha_i^{(n)}$ at the n^{th} iteration. Clearly, BBP is also a deterministic algorithm like MFD, and after n_{iter} iterations, $\tanh(\alpha_s^{(n_{iter})})$ is read off as an approximation of the true conditional expectation for site s.

Summary of Useful Properties Shared by GS, MFD, ICM and BBP

All the four approximate marginalization algorithms discussed above share the following properties which make them well-suited to WSN applications: (i) Messages are sent in a broadcast form. Hence, beam-forming with large or multiple antennae is not required. (ii) Since the statistical neighborhood typically closely mirrors the physical neighbor-

hood, it can usually be fully covered with a single broadcast transmission, without requiring multiple hops. (iii) The order of message passing is not critical. In fact the algorithms work robustly even under mildly non-uniform *frequencies* of updates across the network. In other words, they are robust to poor timing synchronization. (iv) Computations are fully distributed. Since the radius of interaction is assumed to remain bounded irrespective of N, the computation and communication load of a mote *remains invariant even as the network scales*.

3.3 Analysis of Energy Efficiency

We defined a statistical model for the WSN in Section 2.1 and in Section 3.2 devised tractable distributed algorithms to perform in-situ inference based on this model. In this section we compare the energy efficiency of the 'Inference First' (IF) approach with the alternative the 'Fusion First' (FF) approach. In particular, we will demonstrate that the former does give a significant energy advantage for practical values of network parameters.

For simplicity, in this section, we assume that there is no temporal memory in the sampled field ($G = 0$), so X^t and Y^t are i.i.d. processes. Hence, we drop the time subscript t, and refer to X^t, Y^t as X, Y. We would like to emphasize, however, that the argument below can be easily extended to the case of a sensor field with temporal memory by replacing the entropy with the entropy rate. Furthermore, let us postulate an additive white Gaussian noise (AWGN) communication channel, and assume that the same type of transceiver is used by all the sensors such that N_0 is the single-sided thermal noise power spectral density (PSD) in the receiver's front-end. The fusion center may perhaps use a better (low-noise) receiver with a noise PSD of $N_0^F < N_0$. We will also assume isotropic transmission (omni-directional antennae), so the antenna gain is

unity.

Generically, suppose a sensor has to transfer b bits over a distance of a meters in an allocated time interval of T seconds. (T may be defined by a time-slotted multiple access (MA) protocol, or by a fair resource sharing policy in an ALOHA type carrier-sense MA scheme). Let the bandwidth allocated for the WSN operation be W Hertz, and assume passband quadrature signaling at the highest rate possible without inter-symbol interference (Nyquist signaling), so that the symbol time is $T_S = \frac{1}{W}$. As a consequence of the wireless transmission characteristic given in equation (3.1), and Shannon's capacity theorem for AWGN channels [5], the sensor can transmit $n_S = WT$ quadrature symbols in the T second interval with a maximum spectral efficiency of $\eta = \log_2(1 + \frac{P_{TX}}{a^\kappa N_0 W})$ bits per symbol. Thus, in order to be able to communicate b bits the transmitted energy must satisfy the constraint

$$E \doteq T\, P_{TX} \geq a^\kappa f(b/n_S, \sigma_n^2), \tag{3.24}$$

where we defined $\sigma_n^2 = TN_0W$ (similarly $\hat{\sigma}_n^2 = TN_0^F W$) and $f(m,n) \doteq n\,(2^m - 1)$. The inequality becomes an equality when we use a capacity achieving code. Notice that the right-hand side in (3.24) is a monotonically decreasing function of T, implying that it is optimal for a sensor to use all the time T that it is allocated for communication. A similar observation has been made for uncoded modulation in [164]. Note however that these observations hold only if we ignore other sources of power consumption like the sensor electronics (which are relatively minor in typical WSNs).

First consider the FF approach. The aim is to transmit enough information to a fusion center so that it can reconstruct the underlying field with an error probability that is close to the ultimate bound given by maximum a-posteriori (MAP) decoding of unquantized observations. We quantize the sampled observations sufficiently densely and

transfer the resulting bits directly to the fusion center, who will do the MAP decoding (or some approximation thereof). Suppose we have a tolerable degradation when $\delta = \frac{1}{2^{n_{acc}}}$ is the precision of quantization per observation, and the quantizer input range is $[-2^{n_{mag}}, 2^{n_{mag}}]$ (values outside this range are clipped). Let Y^δ be such a quantization of Y. Since the underlying field is correlated, so are the components of Y, and hence of Y^δ. The source coding theorem [5] tells us that we do not need to transfer $N(n_{acc} + n_{mag} + 1)$ bits out of the network per sampling interval, but rather only $H(Y^\delta)$, the entropy of Y^δ. Furthermore, the Slepian-Wolf theorem [61] assures us that that this can be achieved *without* expending any energy in inter-sensor communication. Thus, using a good distributed code [151], sensor number 1 transmits $H(Y_1^\delta)$ bits, sensor number 2 transmits $H(Y_2^\delta|Y_1^\delta)$ bits, etc., and the fusion center is able to decode Y^δ with arbitrarily low error probability. (The order is not important, and can be symmetrized, say by time-sharing). Let d be the distance from the sensor array to the FC. Then, due to the energy bound in equation (3.24), the energy expended by the network as a whole must satisfy

$$\begin{aligned} \mathcal{E}^{FF} &\geq d^\kappa \sum_{i=1}^{N} f\left(\frac{1}{n_S} H(Y_i^\delta|Y_1^\delta,\ldots,Y_{i-1}^\delta), \hat{\sigma}_n^2\right) \\ &\geq N d^\kappa f\left(\frac{n_{acc} + \log_2(\sigma\sqrt{2\pi e})}{n_S}, \hat{\sigma}_n^2\right). \end{aligned} \tag{3.25}$$

In writing the second inequality we have used the property that for any $s \in \{1, 2, \ldots, N\}$

$$H(Y_s^\delta|Y_1^\delta,\ldots,Y_{s-1}^\delta) \geq H(Y_s^\delta|X_s) \geq n_{acc} + \log_2(\sigma\sqrt{2\pi e}),$$

when n_{acc} is sufficiently large [5]. The second inequality approaches an equality as the clutter variance becomes large. Note that the bound in equation (3.25) is typically very loose, i.e. optimistic in terms of the energy required for the FF approach.

Now consider the IF approach, which has two causes of energy drain. The energy

spent in producing in-situ inference by message-passing among the motes satisfies the inequality

$$\mathcal{E}_{INF}^{IF} \leq N r^{\kappa} n_{bcast} \gamma f(n_{msg}/n_S, \sigma_n^2), \tag{3.26}$$

where r is the radius of interaction in the statistical model (see Definition 1), n_{msg} is the number of bits of quantization needed for the broadcast messages, $n_{bcast} = 2n_{iter}$ is the number of broadcast transmissions per sampling interval, and $\gamma > 1$ is a factor that accounts for the fact that the communication of messages *is delay constrained*, since it affects the dynamics of the inference algorithm, and hence long codes that approach capacity cannot be used. The energy spent in communicating the inferred bits to the fusion center, with a capacity achieving channel code, is

$$\begin{aligned} \mathcal{E}_{TX}^{IF} &= d^{\kappa} \sum_{s=1}^{N} f\left(\frac{1}{n_S} H(\hat{X}_s|\hat{X}_1, \ldots, \hat{X}_{s-1}), \hat{\sigma}_n^2\right) \\ &\leq N d^{\kappa} f\left(\frac{1}{n_S}, \hat{\sigma}_n^2\right), \end{aligned} \tag{3.27}$$

where the second inequality follows if we do not compress the inferred bits.

Let $\mathcal{E}^{IF} \doteq \mathcal{E}_{INF}^{IF} + \mathcal{E}_{TX}^{IF}$. Assuming the scenario of uncompressed inferred bits, we then have a lower bound on the energy gain of the FF vs. IF approach,

$$\begin{aligned} \frac{\mathcal{E}^{FF}}{\mathcal{E}^{IF}} &\geq \frac{2^{\frac{1}{n_S}[n_{acc} + \log_2(\sigma\sqrt{2\pi e})]} - 1}{\frac{\sigma_n^2}{\hat{\sigma}_n^2}\left(\frac{r}{d}\right)^{\kappa} n_{bcast}\gamma(2^{\frac{n_{msg}}{n_S}} - 1) + (2^{\frac{1}{n_S}} - 1)} \\ &\rightarrow \frac{2^{\frac{1}{n_S}[n_{acc} + \log_2(\sigma\sqrt{2\pi e})]} - 1}{2^{\frac{1}{n_S}} - 1} \text{ as } d \rightarrow \infty. \end{aligned} \tag{3.28}$$

In Section 3.4.3, we will numerically evaluate this bound for practical values of network parameters.

3.4 Numerical Results, Discussion and Extensions

In this section we will provide simulation results for the quality of inference and the energy efficiency of the approximate filter proposed in Section 3.2. To this end, in Section 3.4.1 we will introduce exemplary statistical models for linear as well as planar sensor arrays, that emulate the generation and dispersion of plumes. Then, in Section 3.4.2 we will compare the quality of the inference of the exact HMM filter and the approximated HMM filter presented in Section 3.2, with the various marginalization engines (GS, MFD, ICM and BP) described earlier. For this comparison we use a small model with $N = 8(9)$ sensors, for the linear (planar) cases respectively, since we cannot simulate the *exact* filter for large N. The proposed approximate filter is of course tractable, and we will investigate its scaling properties as N becomes large (up to $N = 64$). Finally we will display a characteristic of detection versus false alarm probability, when the WSN is used as a *detector*. In Section 3.4.3, we will compare the speed of convergence of the marginalization engines, which determines the energy spent in in-situ inference. We then give numerical results for the over-all energy gain of the IF approach vs the FF approach. Finally, in Section 3.4.4 we will discuss possible extensions of the approximated filter algorithm to non-binary fields, higher order interactions, and the problems of prediction and smoothening.

3.4.1 Simulation Model

Linear sensor array

First consider a regularly spaced linear sensor as in Figure 3.2(a), where the minimum inter-sensor distance is Δ meters. The distance from the array to the FC is d meters (usually $d \gg \Delta$). Suppose that the field is governed by a Boltzmann model (θ, W) of

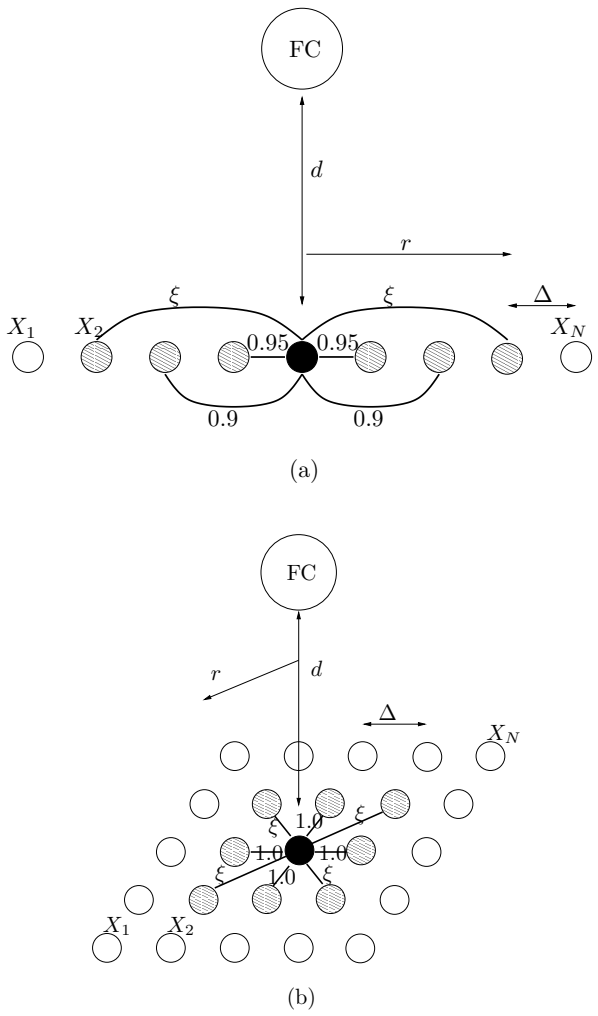

Figure 3.2: Examples of regularly-spaced sensor arrays and their statistical models: (a) linear array (b) planar array.

Section 2.3 such that

$$G = 0.5\ I_{N\times N}, \qquad \theta_s = 0_N, \tag{3.29}$$

and the matrix W_s has the following structure (refer to Figure 3.2(a)): the interaction coefficient for adjacent sensors is 0.95, the interaction coefficient for sensors spaced 2Δ meters apart is 0.9, and the interaction coefficient for sensors spaced 3Δ meters apart is ξ. All other entries in W_s are identically zero. Hence note that the radius of interaction is $r = 3\Delta$. If we label the successive sensors in an increasing order, then the above

specification implies that W_s is an $N \times N$ symmetric Toeplitz matrix whose first row is $[0, 0.95, 0.9, \xi, 0, \ldots, 0]$. The free parameter ξ allows us to tune the spatial correlations in the field. We will use two exemplary values, namely $\xi = -0.7$ and -0.2. For $\xi = -0.7$ the correlation in the hidden field falls off rapidly to zero beyond a distance of about 5Δ, thus simulating plumes of a spatial extent of ten to twenty sensors. $\xi = -0.2$ results in a more slowly decaying spatial correlation, thus modeling larger plumes covering hundreds of sensor sites.

Planar sensor array

Now consider a planar sensor array as in Figure 3.2(b), where the sensors are organized on a square integer lattice. As before, the minimum inter-sensor distance is Δ meters and the distance to the FC is d meters. Similarly, for the statistical model we assume G and θ_s as defined in equation (3.29). The matrix W_s is however defined differently, as follows (refer to Figure 3.2(b)): the interaction coefficient for sensors spaced Δ meters apart is 1.0, the interaction coefficient for sensors spaced $\sqrt{2}\Delta$ meters apart is ξ, and all other entries in W_s are identically zero. Hence note that the radius of interaction is $r = \sqrt{2}\Delta$. Analogous to the case of the linear array, the free parameter ξ allows use to tune the correlation in the planar field. We will consider two exemplary values, namely $\xi = -0.6$ and $\xi = -0.2$, giving rapidly and weakly decaying spatial correlations respectively.

3.4.2 Quality of Inference

Figure 3.3(a) shows, for a linear array of $N = 8$ sensors, the error rate of the exact and the approximated filter as a function of the SCR. Figure 3.3(b) shows similar curves for a planar array of $N = 9$ sensors organized in a 3×3 matrix. In each sub-plot there

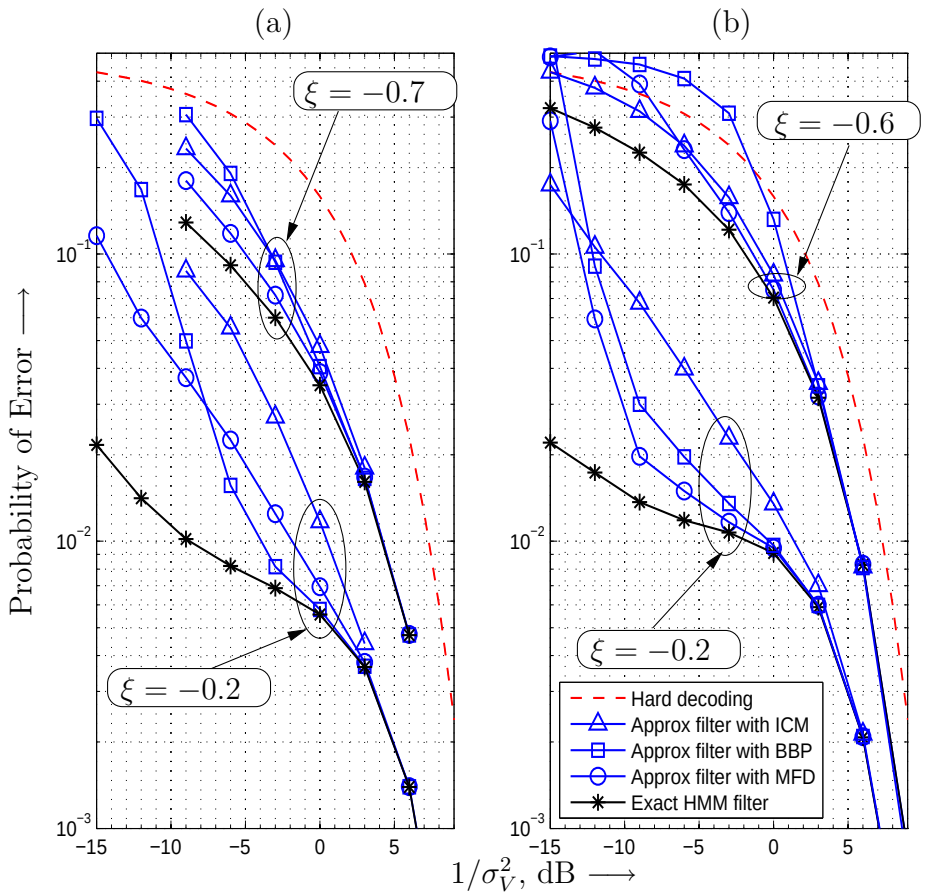

Figure 3.3: Error rate versus SCR $\frac{1}{\sigma^2}$. (a) $N = 8$ sensors, linear array of Section 3.4.1. (b) $N = 9$ sensors, planar array of Section 3.4.1.

are two sets of curves, corresponding to the two values of the parameter ξ. Note that since a 'hard-decoder' makes a decision about X_s^t based only on the local observation Y_s^t (thereby completely ignoring the dependency structure of the field), its quality of inference remains unchanged irrespective of the value of ξ, resulting in only a single curve.

We see that for sufficiently small clutter, the performance of the approximated filter approaches that of the exact HMM filter. This is true for linear as well as planar arrays, and holds for all the three approximate marginalization procedures, namely MFD, ICM

and BBP. For moderate to low SCR however, there are distinct differences in the behavior of the algorithms. Firstly, as predicted in Section 3.2.3, ICM gives significant degradation w.r.t MFD. Thus, although ICM has the smallest communication cost for inter-sensor message-passing, the resulting degradation relative to MFD is perhaps not tolerable. A second interesting feature of the curves is the performance of BBP relative to MFD. For the planar array, BBP shows a significant degradation relative to MFD for all SCR values. For linear arrays, BBP slightly outperforms MFD for moderate SCR, but has a *cross-over* at a certain SCR (that depends on ξ) below which it degrades rapidly. This behavior agrees with the prediction made in Section 3.2.3 that BBP may not perform well on the densely cyclic models typical of WSNs. Note that there is more cyclicity in a planar array than in a linear array, and similarly there is more cyclicity when $|\xi|$ is large. Hence, it is not surprising that BBP has the worst relative degradation for the case of a planar array with $\xi = -0.6$.

In Figure 3.4, we demonstrate the effect of scaling, i.e. increasing the network size N, for the case of the linear array, with $\xi = -0.7$ and several values of the clutter variance. For reasons explained earlier, the exact filter curve has been simulated only till $N = 11$, while the approximate filter is fully simulated up to $N = 64$ (this maximal value has been chosen arbitrarily; much larger networks can also be simulated). We see that MFD scales robustly in all cases, and gives minor degradation irrespective of the network size N. BBP on the other hand scales poorly, especially for low SCR. Although not displayed here due to space constraints, we have observed that the scaling behavior of BBP improves for $\xi = -0.2$ and becomes comparable to MFD. Similarly, we have verified that these scaling properties are also replicated in the case of the planar array, where in fact the scaling behavior of BBP is found to be somewhat worse than in the case of the linear array. Again these results are not surprising since the cyclicity in the model increases more rapidly for the planar case than the linear case, as the network

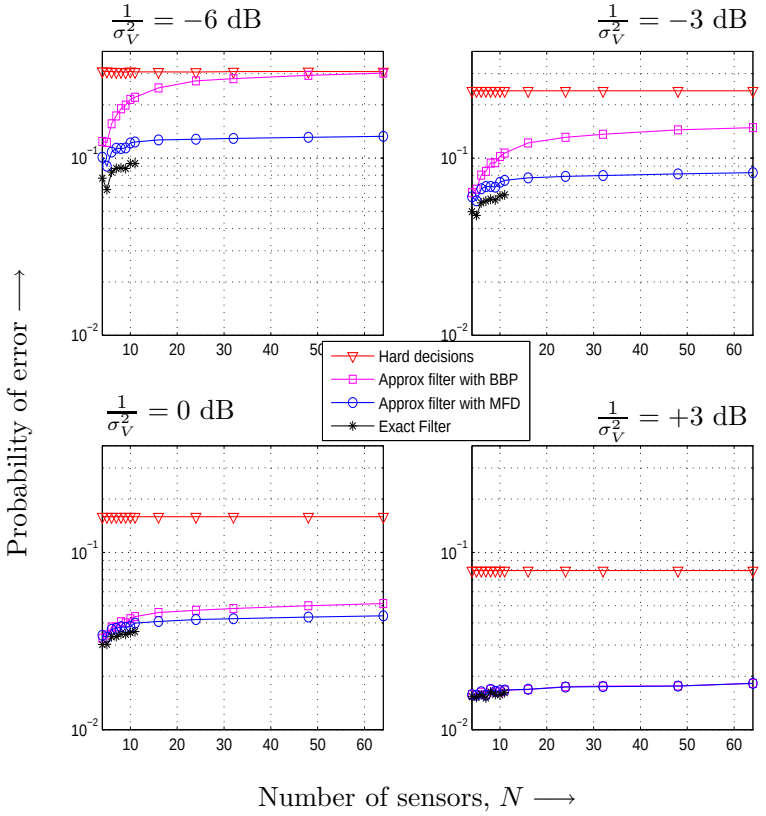

Figure 3.4: Scaling of filtering algorithms with network size N. Linear array of Section 3.4.1 with $\xi = -0.7$.

scales.

Finally, in Figure 3.5 we show the operating characteristic of the WSN used as a *detector*, at two values of SCR, and a linear array with $N = 8$ sensors and $\xi = -0.2$. The detection-probability vs. false-alarm tradeoff is controlled by slicing the a-posteriori mean by a threshold that takes various values from the interval $[-1, 1]$. Again, we see that for moderate to high SCR the approximated filter, with MFD or BBP, gives a close to optimal tradeoff between detection and false-alarm probability, and that it degrades more gracefully under MFD than BBP, as the clutter increases.

In light of the discussion above, MFD seems to be a more robust algorithm than ICM and BBP. Furthermore, as we shall see in the next section, it converges rapidly

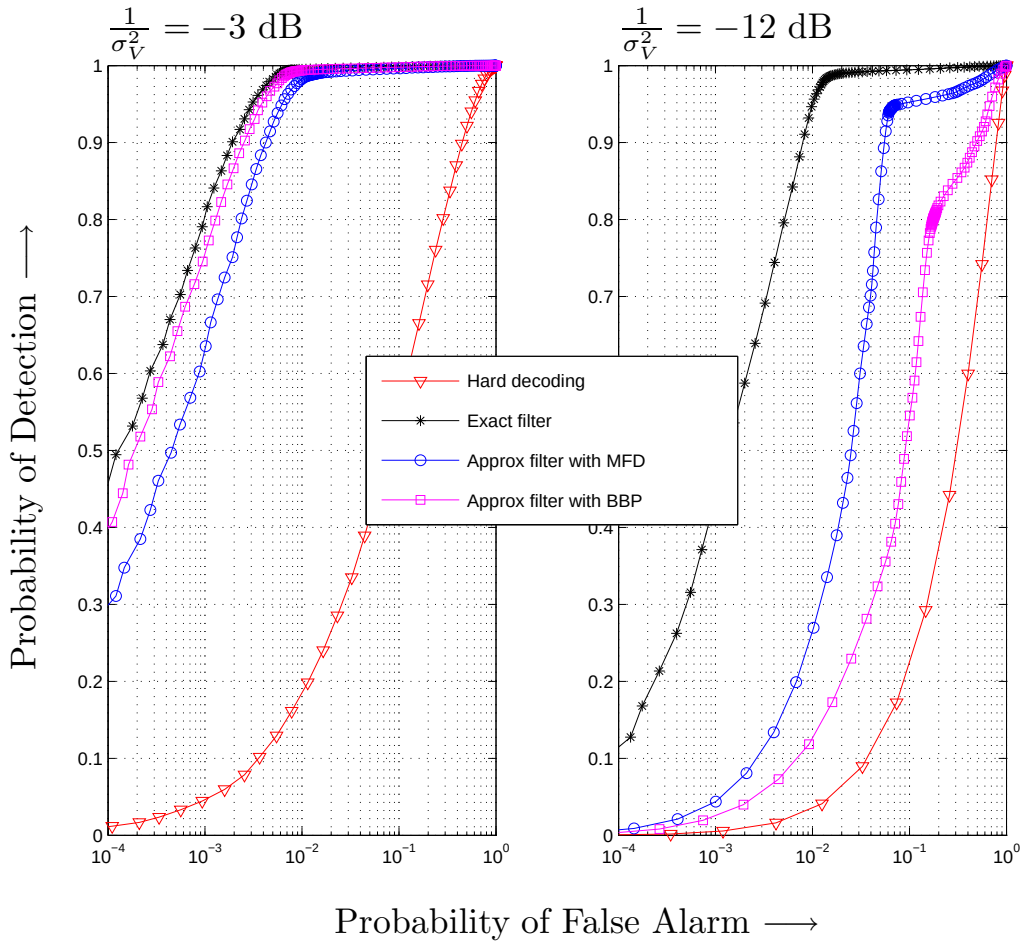

Figure 3.5: Operating characteristic of WSN used as a detector. Number of Sensors $N = 8$, linear array of Section 3.4.1 with $\xi = -0.2$.

as compared to GS and thus has a relatively low communication cost (only marginally larger than ICM). Thus, on the whole, MFD seems to be a good choice for distributed filtering applications.

3.4.3 Energy Efficiency

We will now consider, for GS and MFD marginalization, the tradeoff between energy consumption and the quality of in-situ inference. Figure 3.6 shows the probability of error as a function of the number of iterations allowed for the inference algorithm, for

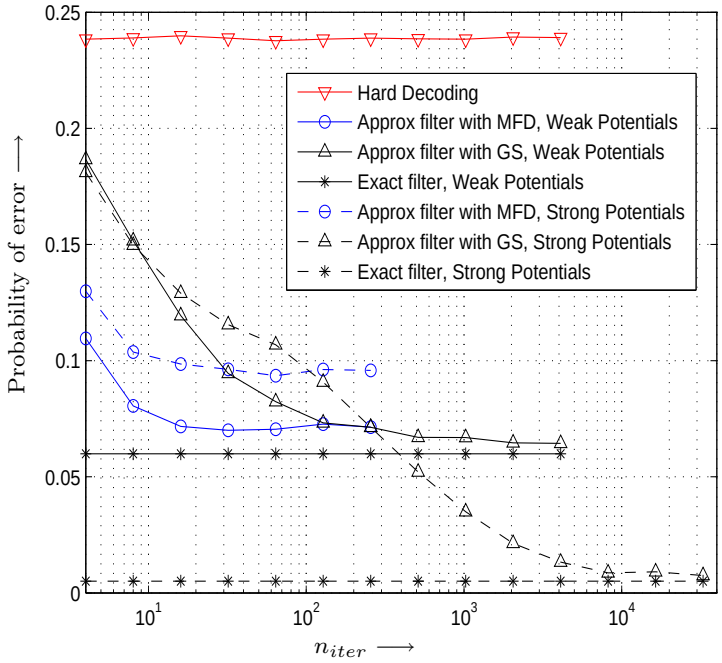

Figure 3.6: Quality of inference as a function of n_{iter}. SCR -3.0 dB. Linear array with $N = 8$ and $\xi = -0.2$. 'Weak potentials' implies the model of Section 3.4.1. 'Strong potentials' implies the model of Section 3.4.1, with entries in W_s scaled up by a factor of 3.0.

two types of linear array models of size $N = 8$. The first model, which we will refer to as the model with *weak potentials*, is exactly as described in Section 3.4.1. The second model, which we will refer to as the model with *strong potentials*, is also the same model as in Section 3.4.1, but with all entries in W_s scaled up by a factor of 3.0. We fix $\xi = -0.7$ for both cases. The exact filter performance is provided as a baseline, against which we compare the approximate filter with MFD or GS marginalization. We see that MFD gives most of the performance gain within 10 to 20 iterations, while GS typically needs thousands of iterations. For weak potentials, MFD gives a performance almost as good GS does asymptotically, and both are close to the exact filter. For strong potentials however, while the asymptotic performance of GS is again close to the exact filter, the performance of MFD has a significant degradation. This suggests that it may

be worthwhile to use GS when the potentials are strong and the energy constraints are not very stringent. However, if the number of iterations is severely constrained, MFD marginalization seems to be a *uniformly good choice* irrespective of the strength of interactions.

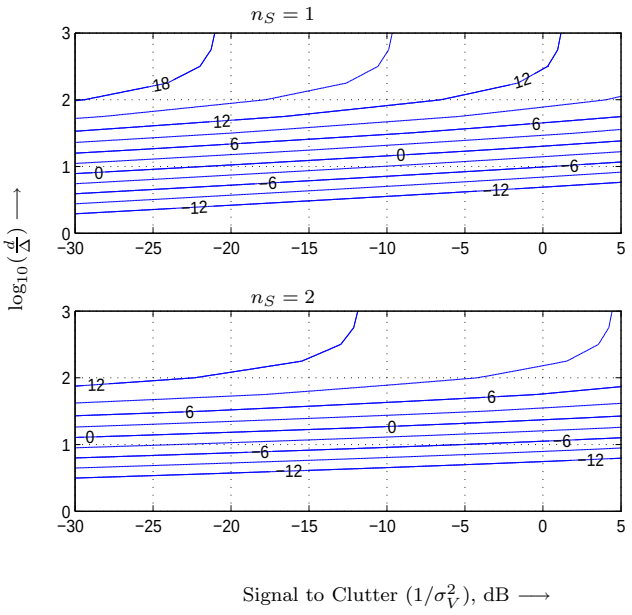

Figure 3.7: A comparison of the energy efficiency of Inference-First and Fusion-first approaches. Figure shows contour plots of a lower bound on $\frac{\mathcal{E}^{FF}}{\mathcal{E}^{IF}}$ in 3 dB intervals, for two values of n_S. In the region above the break-even curve (0 dB contour) IF is more energy efficient than FF. $N_0 = N_0^F$ and $n_{bcast} = 20, \gamma = 5, r = 3\Delta, \kappa = 2.0, n_{acc} = n_{msg} = 4$.

Finally, we will consider over-all energy efficiency. We assume the worst-case scenario where the inferred bits are not compressed. In Figure 3.7 we show the contours of the energy gain of the IF versus the FF paradigm, as given by equation (3.28), where we choose practical parameter values: $n_{acc} = n_{msg} = 4, n_{bcast} = 20, \gamma = 5$, and $n_S = 1, 2$. The radius of interaction is assumed to be $r = 3\Delta$. Since the energy efficiency does not directly depend on the topology of the array once the radius of interaction is specified, it follows that the results shown here are applicable to linear as well planar arrays with this radius of interaction. We assume free space propagation, so $\kappa = 2.0$. The contours are

plotted for the free parameters $\frac{d}{\Delta}$ and σ^2. The contour for 0 dB gain divides the $\frac{d}{\Delta}$ vs. σ^2 plane into two regions where IF is respectively better and worse that FF in terms of energy efficiency. Notice that for large clutter, the *asymptotic energy gain* (as $\frac{d}{\Delta} \to \infty$) can be as large as 12 to 18 dB. A further large factor of improvement is possible if the field has a low entropy and we implement (distributed) compression of the inferred bits. Since the distance to the fusion center is typically hundreds or thousands of times the radius of interaction, one can conclude that the IF approach will be of advantage in most practical WSNs and will give at least an order of magnitude improvement in the energy efficiency, and hence the lifetime, of the network. Note that due to the looseness of the bound in equation (3.25), and the assumption that the inferred bits are not compressed, our estimate of the energy gain is very conservative.

3.4.4 Extensions

While we have considered the 'filtering' problem, generalizations can obviously be made to the closely related problems of 'prediction' and 'smoothening'. For example, suppose we wish to predict the state of a sensor site n_f steps into the future. For this, after *every* sampling epoch t, the filtering algorithm is first implemented to estimate the current state, and then n_f more extensions of the algorithm are made assuming $h^k = 0,\ k = t+1, t+2, \ldots, t+n_f$. Thus the computation and communication load per sensor is increased by a factor of $n_f + 1$, but otherwise remains invariant w.r.t. N, thus maintaining the scalability properties.

Secondly, although we considered a binary valued field, the extension to an M-ary field can be made in a straightforward way by making any suitable choice of $M-1$ sufficient statistics in the exponential model of Q. In particular, if M is a power of two then we can simply use a binary representation for the M-value of the field at each sensor

location, and our algorithm can then be used with minor changes. Each sensor will now need to broadcast $2\log_2 M$ values at each iteration, rather than 2 values as is the case currently, and the size of the marginalization problem will be $2N\log_2 M$ rather than $2N$. Again, scalability is preserved. Similarly, the proposed approximate filtering algorithm can also work for a variety of measurement models other than additive Gaussian clutter. We only require that the conditional independence property in equation (2.1)) be satisfied, and the correct conditional distribution $P(y_s^t|x_s^t)$ be used in equation (3.6).

Thirdly, an extension to the case of ternary and higher order interactions is also easily possible, with no increase in the communication load, and only a minor increase in computation complexity. Such higher order interactions are implemented by using basis functions that simultaneously depend on three or more site-variables, and can be useful in modeling a-priori known 'patterns' or 'textures' in the field.

Finally, extensions to continuous valued fields are also possible. A very special and useful case is that of Gauss-Markov fields (where pairwise interactions suffice to give a complete description). The optimal filter in this case becomes the well-known *Kalman* filter, and, as in the case of the binary valued field, it is intractable for large N (though now the complexity grows only as N^3 rather than the astronomical 2^N). It can be shown that the MF marginalization of the joint density Q for such a field is equivalent to solving the set of 'Normal Equations' that give the best linear estimate in the MSE sense (and in fact the general minimum MSE estimate, since Q is Gaussian). Such positive definite systems can of course be solved very efficiently by conjugate gradient methods. But since we want a physically distributed algorithm we need to use iterative methods based on matrix splitting, even though they do not give the fastest possible convergence. The simplest such methods, called the *Jacobi* and the *Gauss-Seidel* iterations [24, 165], are structurally identical to the MFD iteration we have formulated for the binary field, which thus gives a very satisfying connection between these two methods. Moreover, even BP

is known to give the exact marginals for a Gauss Markov model, provided it converges [29]. We conjecture that the proposed approximate filter can be fruitfully applied even to non-Gaussian continuous valued fields, by using exponential models with carefully chosen higher order basis functions.

3.5 Summary and Conclusions

We outlined two principal approaches to data-gathering in WSNs, which we called the 'Fusion First' and the 'Inference First' methods respectively. The main difference between these schemes is that the former delivers all the raw sensor data to the fusion center, while the later delivers only the few sufficient statistics required by the application. We claimed that the IF method can give a large improvement in the energy expenditure (and hence, the lifetime) of the network, provided efficient distributed algorithms are used to perform the 'in-situ' inference.

To demonstrate this, we proposed a *scalable* distributed filter for estimating the hidden field from sensor observations. The filter is based on three novel ideas: Approximating the propagated p.m.f. by a product of its marginals, viewing the filter update as a marginalization procedure on a joint distribution for the state of the HMM at two consecutive epochs, and then approximating this marginalization using efficient message passing algorithms. Theoretical and simulation results indicate that the filter is stable and robust, and has minor degradation w.r.t. the optimal filter, provided the SCR is not too low. We compared various marginalization engines like GS, MFD, ICM and BBP, and found that for large networks with densely cyclic models and significant clutter, MFD is perhaps the best choice. With a scalable in-situ filtering algorithm at hand, we then showed that the 'Inference First' approach gives a substantial energy gain over the alternative of exporting all the raw sensor data out of the WSN. For typical practical

networks an energy saving of more than 10 dB seems easily achievable.

4 Distributed Compression and Data Extraction

4.1 Introduction

The discussion in Chapter 3 was concerned with inferring or de-noising a hidden field from an observation process using a distributed scalable *in situ* algorithm. While the results of such an algorithm can sometimes be used for local feedback control of actuators in real time, it also often the case that one wishes to recreate the inferred field at a, perhaps distant, Fusion Center (FC) for more elaborate processing and analysis. (It is presumed that the FC itself has no serious constraints on complexity or power consumption.) The reconstruction has to be done with a precision and latency that is satisfactory to the end-application. For example, a low flying unmanned aerial vehicle may may make passes over an array of sensors in inaccessible terrain and gather real-time environmental data. The data transfer from the WSN to the FC has to be done over long-haul wireless channels, perhaps using 'gateway' motes. Since wireless transmissions suffer a path loss that increases quadratically or more with distance, if we wish to maximize the network lifetime we must ensure that the motes minimize their power consumption. Hence they must make a minimal number of transmissions to the FC.

A principled method of achieving this is to exploit the significant spatiotemporal

dependencies that all natural fields exhibit. In particular, when an independent communication channel is available to each mote for communicating with the FC, *source-channel separation* i.e. first compressing the source down to its entropy rate and then using a capacity achieving code, is known to be an optimal procedure [5]. Furthermore, in the case of fields with finite alphabets, optimal compression with perfect reconstruction is possible, at least in principle, without requiring the motes to share their data with each other – a remarkable result demonstrated by Slepian and Wolf [61]. It is noteworthy, however, that such a result does not hold in the general rate-distortion case investigated by Wyner and Ziv [64], where typically some rate-loss has to be tolerated.

The traditional approach to data extraction based on source-channel separation would be to designate one mote as a leader mote, aggregate all the field data at the leader via multi-hop transmissions, and then let the leader encode and transmit the data out of the network. However this approach is obviously not scalable, because as the network size increases the communication load on any typical mote due to data aggregation increases unboundedly. Also, if the leader mote fails, the scheme is completely crippled. A more practical proposal is the one based on Slepian-Wolf compression, using the so called 'Distributed Source Coding Using Syndromes' (DISCUS) method [151], which can approach any corner point of the achievable rate region. However, source-channel separation methods like DISCUS also have several drawbacks from a practical point of view, prominent among which are: (a) The encoder is not universal, i.e. it has to be re-designed every time there is a change in the joint statistical model of the sensor data. Moreover, the encoder also has to be re-designed every time a *new mote* joins the network, and the encoding complexity per-mote increases as the network scales. (b) It is difficult to accommodate fields with *temporal* dependencies because this involves designing channel codes for certain 'virtual channels' possessing arbitrary memory structure [66]. (c) A single compression code cannot give the user the ability to seamlessly trade

fidelity of reconstruction for increased power efficiency in an optimal manner. (d) Like centralized compression, the procedure is still very *fragile* in the sense that the failure of even a single mote disables the entire reconstruction procedure.

With these observations in mind, we propose a new method of data extraction that gives a power efficiency comparable to or even better than practical source-channel separation methods based on distributed source coding, while solving *all* the above problems (universality, temporal memory, rate-distortion trade-off, and robustness to mote failures). The procedure uses a distributed 'rate-less' Digital Fountain Code (DFC) [72, 166], having a light-weight homogeneously distributed encoder. Rather than exploiting the spatiotemporal dependencies in the natural field for direct compression at the WSN, we treat them as an outer code serially concatenated with the DFC, and jointly decode this concatenation *at the FC* using a multi-stage iterative decoder.

To our knowledge, our proposal of probabilistic compression of discrete fields via low density generator matrix Luby Transform (LT) [166] fountain codes is novel. Fountain codes have been used before, in the context of WSNs, for networked data storage by [68, 70, 71], but those works did not exploit the dependencies in the data to achieve compression. Our proposal is more closely allied to the universal loss-less compression scheme developed by [60], which is based on fountain codes and the Burrows-Wheeler transform. However the approach in [60] involves a significantly complex encoding procedure and is not amenable to a distributed implementation. In contrast we use a very simple distributed encoder and an elaborate decoder – an allocation of complexity well suited to the WSN architecture. Another notable issue, relevant to applications with strict latency constraints, is that universal compression codes like [60] need significantly longer block sizes than model based compression codes like our proposal, to achieve similar compression gains [5, Section 12.3]. Our scheme also has striking similarities, in terms of the philosophy of data extraction, to the technique of *Compressed Sensing*

(CS) [50, 53], which has recently generated considerable interest in the signal-processing community as a means of universal compression of *sparse real valued* signals. We use a random generator matrix over GF(2) just as CS uses a random measurement matrix over $\mathbb{R}$, and hence both schemes are universal with respect to the source statistics. We are interested in Shannon-theoretic 'compressibility', while CS is interested in 'sparsity'. We exploit compressibility at the receiver by using an outer field decoder concatenated to the fountain decoder, while CS exploits sparsity at the receiver via a basis pursuit reconstruction i.e. an L_1-minimization with inequality constraints. Both reconstruction methods are suboptimal, but eminently tractable compared to the respective optimal schemes of joint maximum a-posteriori probability (MAP) decoding and an L_0-minimization under inequality constraints, which are prohibitively complex. These analogies suggest that there is a deeper connection that could lead to a powerful general theory of data acquisition.

Outline of this chapter: In Section 4.2 we will delineate the statistical models for the source and the channel of the WSN, and define the goals of the data extraction procedure, as well as the constraints it needs to satisfy. In Section 4.3 we will describe our proposed data extraction scheme, and also present two variants. In Section 4.4 we will present a performance analysis based on Extrinsic Information Transfer (EXIT) charts, and information theoretic bounds and approximations. Section 4.5 presents simulation results, and Section 4.6 concludes the chapter.

4.2 System Model

Consider the system setup displayed in Figure 4.1, where the principal blocks are the WSN, the channel, and the FC. The WSN is an array of N spatially scattered motes that periodically (every T_{samp} seconds) sample an underlying natural field X, which we

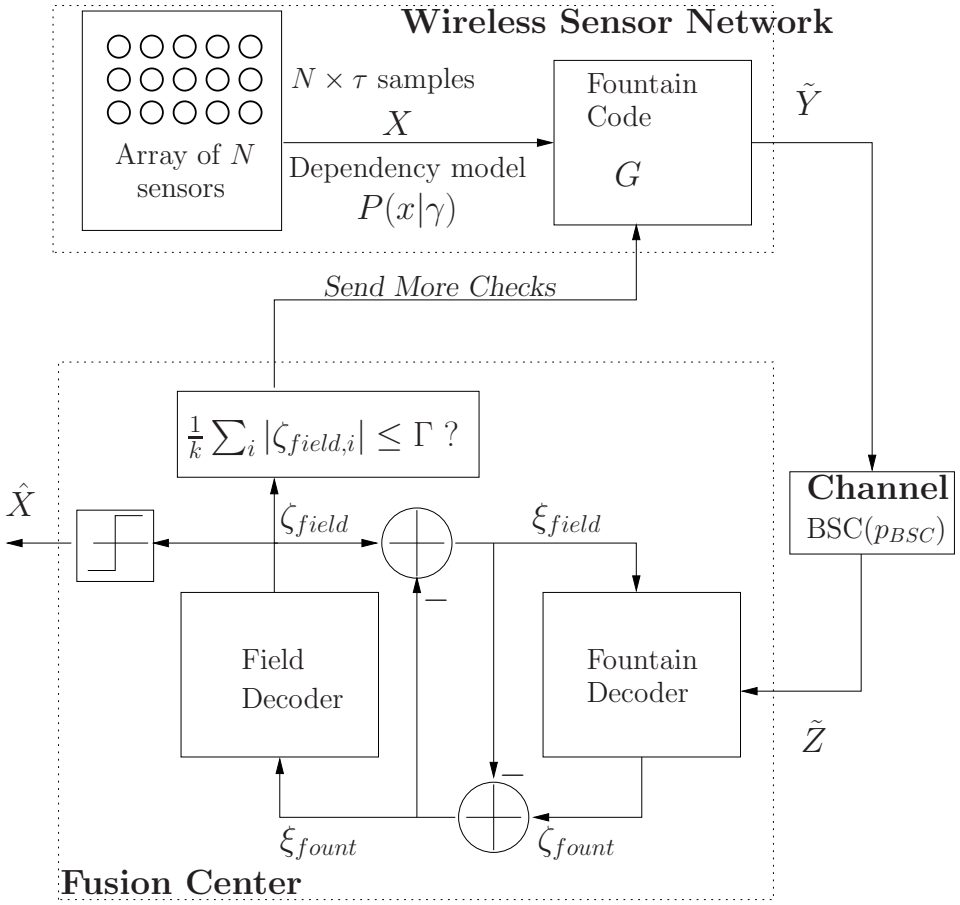

Figure 4.1: Block diagram of the data extraction schema.

view as a random spatiotemporal process. A sufficiently long set of these measurements is encoded within the WSN via a message passing fountain encoder. The resulting output symbols are transmitted by a subset of motes to the FC over noisy channels. The FC infers X from these these noisy symbols using a two stage iterated decoder, with intermittent low-rate feedback to the WSN.

4.2.1 Statistical Model for the Random Field

Since natural fields generally exhibit spatial as well as temporal dependencies, both should be maximally exploited to achieve the best possible compression efficiency. We

will assume in this chapter that the field samples have been filtered (de-noised) using a tractable distributed algorithm like the one discussed in Chapter 3 (see also [15, 30, 31, 115]). Let $\{X_s^t\}$ be the filtered field process, drawn from the discrete alphabet $\mathcal{X} = \{+1, -1\}$. We will discuss generalizations to larger alphabets in Section 4.3.4. Recall the notation in Chapter 2 where we denoted a frame of field realizations from time 1 through τ as $\bar{X}^\tau$ or simply X. Assume that X has a generic parametric joint distribution $\Pr\{X = x\} = P(x|\gamma)$, where γ is a (vector) parameter. We will only require that the distribution be known to the FC and be amenable to approximate marginalization with tractable algorithms. For example, Section 2.3 describes a Markov Chain (MC) model for plume generation and dispersion, which we will use in our simulations.

4.2.2 Statistical Model for the Channel

For clarity of exposition we will focus our attention on a WSN where motes can directly communicate with the FC with long-haul wireless transmissions. Extensions of the data extraction method proposed here to the case where motes use gateways are straightforward. Suppose that each mote communicates with the FC using an independent binary-input-symmetric-output memory-less channel of capacity C bits per use. For simplicity we will assume that the communication *in-between motes* occurs with negligible latency and error probability. This is reasonable because the inter-mote distance is small relative to the WSN-to-FC distance, and the power expended on ensuring error-free inter-mote communication is consequently small, even in a multi-hop scenario, compared to the power used for long-haul transport to the FC. (Note however that *ideal inter-mote communication is not a critical requirement of our proposal.* A single error in collecting the parity-check data can be equivalently viewed as an error in transmitting the parity check to the FC. Two errors cancel each other, and so on. Hence unreliabilities

in inter-mote communication can simply be viewed as an apparent decrease in the capacity of the sensor-to-FC channel, causing no loss of generality. In particular, this implies that moderate error rates in inter-mote communication cause only a marginal reduction in transmission efficiency, and hence our scheme is robust in this respect.) Lastly, since the FC is not power constrained, we will assume that the FC can communicate with the motes using an error-free broadcast channel.

4.2.3 Goals of Data Extraction

We wish to produce at the FC a sufficiently accurate reconstruction $\{\hat{X}_s^t\}$ of the true sensed field $\{X_s^t\}$. To conserve batteries, we wish to achieve this reconstruction with a minimum number of transmissions to the FC. It is also important that no single mote or subset of motes be encumbered with a disproportionately heavy communication load, since that can lead to the formation of 'blind-spots' and make the entire array unusable. The latency of the data extraction, τ samples (τT_{samp} seconds), should also be kept small, to allow applications to implement agile feedback control. Lastly, we need to have the ability to seamlessly trade-off fidelity of reconstruction for power efficiency, thus giving the system designer the ability to do cross-layer optimization (adaptively increasing the fidelity only when events of interest are detected.) Since we are considering a discrete field, a natural fidelity criterion is the *average probability of error*, defined as

$$P_e \doteq \frac{1}{N\tau} \sum_{t=1}^{\tau} \sum_{s=1}^{N} \Pr\{\hat{X}_s^t \neq X_s^t\}. \tag{4.1}$$

To achieve a fidelity level P_e the network needs to pay a certain cost $\rho(P_e)$ which needs to be minimized. We define this cost to be the normalized communication load

$$\rho(P_e) \doteq \frac{\mathbb{E}\left[n_{tot}(P_e)\right]}{N\tau} \quad \text{[transmissions per mote per sample]}, \tag{4.2}$$

where $\mathbb{E}\left[n_{tot}(P_e)\right]$ is the expected total number of transmissions made from the WSN to the FC during the reconstruction of one frame. The rationale behind using $\rho(P_e)$ as the cost function is that the power budget of a mote is dominated by the wireless transceiver, rather than sensing and signal processing operations. Furthermore, as a first order of approximation, we can neglect the cost of inter-mote communication. Note that since the relation between ρ and P_e is one-to-one, we can also equivalently consider the functional dependence $P_e(\rho)$, i.e. the fidelity achieved by paying a given communication cost. We will use these two descriptions interchangeably.

4.3 Data Extraction Scheme

In this section we will first describe the data extraction scheme in two stages. Section 4.3.1 describes a simple distributed encoding scheme that is implemented in the WSN, while Section 4.3.2 describes the corresponding reconstruction scheme implemented at the FC. Then in Section 4.3.3 we will describe two variants of the scheme that can be used to satisfy additional practical constraints that may be imposed in some scenarios. Finally, in Section 4.3.4 we present a discussion of the salient properties of the scheme, and some possible extensions. For the following, please refer to the schema shown in Figure 4.1.

4.3.1 The Encoder

Let $\tilde{X} \in \{0,1\}^k$ be the $\{0,1\}$ representation of X, obtained by the following mapping:

$$\begin{aligned} -1 &\leftrightarrow 1 \\ +1 &\leftrightarrow 0. \end{aligned} \tag{4.3}$$

The fountain code is defined by a generator matrix G, with k rows and potentially an infinite number of columns.[1] Each column of G is chosen *independently* in the following manner: First a random degree d is chosen by sampling the *Soliton/Robust-Soliton distribution* or one of its variants [80]. Then d locations are chosen uniformly, without replacement, within that column and the corresponding elements are set to one, while the rest are set to zero. Each column is then interpreted as a parity check on the elements of $\tilde{X}$ corresponding to the non-zero locations. Note that the resulting G is very *sparse*, irrespective of the value of k. For example, the average degree $\bar{d}$ of the Shokrollahi distribution [80] is only 5.2.

We will assume that every column of G can be synchronously generated by all the motes as needed, by synchronizing the seed of their random number generators. Similarly, the selected G is known to the FC. Each column is assigned an *owner mote*, who can be selected via some convention from among the participating motes. The owner is solely responsible for gathering the necessary data and calculating the parity check defined by its column. The other motes involved in the column are called *slave motes*; they only need to be cognizant of the fact that they have to communicate their relevant data values to the owner, when that parity check is required to be calculated. Given a block of α columns of G, each mote is expected to be an owner of $\frac{\alpha}{N}$ parity checks and a slave to $\bar{d}\frac{\alpha}{N}$ parity checks. Let

$$\tilde{Y} \doteq G^T \tilde{X} \ (mod\ 2) \tag{4.4}$$

be the vector of check bits produced by the fountain code. These check bits are calculated and transmitted only when demanded by the FC, as follows. When a realization of X is

[1]Note that the symbol G has also been used to denote the temporal-dependency sub-matrix in the Boltzmann model introduced in Section 2.3. However this ambiguity in usage is easily resolved from context.

sampled in the WSN, the motes wait till they receive a *Send More Checks* signal from the FC. When such a signal is received, a subset of n_{incr} owner-motes, corresponding to the first n_{incr} columns of G, gather the data they need to calculate their respective parity-check bits. The data gathering is done via inter-mote communication, with multiple hops if necessary. Each owner-mote from the chosen subset then uses its forward channel to transmit its calculated check bit to the FC, after which the WSN awaits further instruction from the FC. If the instruction is again a *Send More Checks* signal, the above service procedure is repeated by a new set of owner motes corresponding to the next n_{incr} columns of G, and so on. Otherwise, if the FC sends a *Frame Decoding Successful* signal, the WSN array flushes the sensed data and remains quiescent till it has again sampled a new realization of X.

4.3.2 The Decoder

The decoder is implemented at the FC, and is relatively complex. (However the decoder is still an algorithm of complexity $O(N)$ and hence retains the important property of tractability and scalability w.r.t. the size of the WSN.) It is assumed that the FC knows the field model $P(x|\gamma)$ (cf. Section 4.2.1). Let the channel outputs $\tilde{Z}_i$ be probabilistic memory-less functions of the inputs, i.e. $\tilde{Z}_i = f_i(\tilde{Y}, \omega)$, where $\omega \in \Omega$ is an element from the underlying probability sample space, such that the following conditional independence conditions hold: $(\tilde{Z}_i \perp \tilde{Y}_j)|\tilde{Y}_i \; \forall j \neq i$. On receiving each increment of n_{incr} components of $\tilde{Z}$, a two-stage iterative decoding procedure is implemented, based on *all* the channel outputs $\tilde{Z}_1, \ldots, \tilde{Z}_n$ observed up to that point. At the start of the iterative decoding procedure the extrinsic information vectors $\xi_{field}, \xi_{fount} \in \mathbb{R}^k$ are initialized to zero. (Please refer to Figure 4.1.)

Fountain Decoder

The first stage is a fountain decoder based on the factor graph [143] of the matrix G. This graph has two disjoint subsets of variable nodes. The first subset corresponds to the k variables in the vector $\tilde{X}$, and its *a-priori information* is given by ξ_{field}. The second subset consists of the n check-sums transmitted from the WSN so far, $(\tilde{Y}_1, \ldots, \tilde{Y}_n)$, and its a-priori information is the *intrinsic* information

$$\xi_{chan,i} = \frac{1}{2} \log \left(\frac{\Pr\{Z_i | Y_i = 1\}}{\Pr\{Z_i | Y_i = -1\}} \right) , \; i = 1, 2, \ldots, n, \tag{4.5}$$

where Z, Y are the $+1/-1$ representations of $\tilde{Z}, \tilde{Y}$ respectively (cf. Section 4.3.1). (**Please note**: We include a factor of $\frac{1}{2}$ in definition (4.5) following the information-geometry literature [162, equation (1)]. We prefer this convention because it allows us to avoid a factor of two in equations (4.6) through (4.13), and yields the consistency condition "mean equals variance" in Section 4.4.1.)

The graph has n factor nodes, corresponding to the n check sums, and their connectivity to the variable nodes is defined by G. The FC implements a standard Sum-Product (SP) [143] message passing algorithm, to efficiently (in $O(k)$ operations) produce approximate marginals of this model[2]. The message from a variable i to a factor $j \in \mathcal{A}(i)$ in its neighborhood is denoted by $\alpha_{i \to j}$, and is initialized to the a-priori information. The message from a factor j to a variable $i \in \mathcal{A}(j)$ in its neighborhood is denoted by $\beta_{j \to i}$ and

[2]The approximation error is known to be small for models with long cycles, as is the case for DFCs.

is initialized to zero. Then the following iterative message passing algorithm is executed:

$$\begin{aligned} \alpha_{i\to j} &= \sum_{l\in\mathcal{A}(i),\, k\neq j} \beta_{l\to i}, \\ \beta_{j\to i} &= \tanh^{-1}\left(\prod_{l\in\mathcal{A}(j),\, l\neq i} \tanh(\alpha_{l\to j})\right). \end{aligned} \tag{4.6}$$

A sufficient number of these internal message-passing iterations are made, after which the fountain decoder produces the a-posteriori log-likelihood ratios (LLRs)

$$\zeta_{fount,i} = \sum_{j\in\mathcal{A}(i)} \beta_{j\to i},\ i = 1, 2, \ldots, k, \tag{4.7}$$

of the field values $\tilde{X}$ (hence X). From this the extrinsic information $\xi_{fount} = \zeta_{fount} - \xi_{field}$ is extracted and passed to the second stage, namely the Field decoder.

Field Decoder

The field decoder is also a message passing algorithm, and is based on the graphical representation of the field model $P(x|\gamma)$. For the example of the Boltzmann field model described in Section 2.3, the graph is a *Markov Random Field* [143] whose nodes are the random variables in X, and whose connectivity is defined by the matrix $\mathbf{W}$ defined in equation (2.13). Due to the tight cyclicity and *soft constraints* typically imposed by $\mathbf{W}$, SP does not perform well as a marginalization algorithm. This phenomenon is well-known in the Turbo decoding literature, and can be rigorously explained in terms of the information geometry of the statistical model [162]. Hence, instead of SP we prefer to use an alternative $O(k)$ procedure for approximate inference, namely the Mean-Field (MF) algorithm [152]. The MF algorithm consists of a sequence of message broadcasts

by the nodes, where the message broadcast by the i^{th} node is given by

$$m_i \;=\; \tanh(\Theta_i + \xi_{fount,i} + \sum_{j \neq i} \mathbf{W}_{ij} m_j). \tag{4.8}$$

(Note that the quantities $\Theta, \mathbf{W}$ are parameters of the field model, and are defined in Section 2.3.) The order in which nodes broadcast their messages is not important and can be chosen pseudo-randomly. An iteration is said to be completed when one round of message broadcasts is completed. After a small number of such internal iterations, the MF algorithm produces the a-posteriori LLRs

$$\zeta_{field,i} = \tanh^{-1}(m_i),\ i = 1, 2, \ldots, k, \tag{4.9}$$

of the field values X. These are converted into extrinsic information $\xi_{field} = \zeta_{field} - \xi_{fount}$, which is then passed back to the fountain decoder.

Terminating an Attempt and Asking For More Check-Bits

The two stages of the decoder described above are iterated several times till ξ_{fount}, ξ_{field} converge. Then the sign of ζ_{field} is the estimate $\hat{X}$ of the field based on the n check-bits observed so far. If

$$\frac{1}{k} \sum_i |\zeta_{field,i}| \geq \Gamma, \tag{4.10}$$

where Γ is a pre-defined threshold, the FC broadcasts a *Frame Decoding Successful* signal to the WSN. Otherwise a *Send More Checks* signal is broadcast, which leads to the reception of a new batch of elements from Z, and the iterated decoding procedure is repeated. The fidelity of reconstruction can thus be controlled via Γ. Since the log-likelihood ratio is related to the a-posteriori error probability via the relation[162]

$\Gamma = \frac{1}{2}\log\frac{1-P_e}{P_e}$ (cf. equation (4.5)), the error probability is nominally given by

$$P_e^{nom}(\Gamma) = \frac{\exp(-\Gamma)}{\exp(\Gamma)+\exp(-\Gamma)} \,. \tag{4.11}$$

4.3.3 Two Variants

Reducing the Inter-Mote Communication

Unlike Slepian-Wolf compression, our data extraction procedure requires a certain amount of data exchange among the motes. A possible concern could be regarding the amount of inter-mote message passing needed in calculating the parity checks. Note that since G is very sparse, a mote communicates its values to only about $\bar{d}\frac{\alpha}{N}$ other (not necessarily distinct) motes, for producing α checks. Nevertheless, in large networks, some of these messages may need to be communicated to distant motes via hopping. To mitigate this problem, we can put further *localization* constraints on G. That is, the active motes in each column must lie in a small bounded physical neighborhood (preferably such that it can be covered without hops). During the generation of G, the columns that don't satisfy this constraint are simply skipped over. As an extreme case we can even forbid all inter-mote communication. Then each column has only one active mote, and the parity check is performed only on local (temporal) values of the field at that mote. One can expect such localization constraints to cause some degradation because there is some loss of 'randomness' in the matrix G. The moot question however is: how pronounced is this effect? We will demonstrate via simulations in Section 4.5 that we can put strong locality constraints on G without suffering a significant loss in performance.

Focused Checking

The proposed scheme gives a close-to-optimal extraction efficiency for moderate values of P_e, as shall see in Section 4.5. For very small P_e, however, the slope of the $P_e(\rho)$ characteristic degrades (floors). This is partly attributed to the fact that, in order to keep the encoding complexity $O(N)$, we used a DFC having a degree distribution with *finite support*. It is known [80] that this degradation can be substantially eliminated by using a *Raptor* code rather than a stand-alone DFC. However, for clarity of exposition we will not consider these extensions in this chapter. A second reason for the floor is the "softness" of the field constraints. That is, even if all the values of the field X in the spatiotemporal neighborhood of a particular location (s,t) become known, the field model still leaves some residual uncertainty about that location[3]. This uncertainty can be eliminated only when a check sum which directly involves that location becomes available without corruption, an event that happens with probability close to one only after k/C transmissions.

If the end application insists on a very low P_e, one can significantly improve the floor by using a technique we call *focused checking*. In this method the *FC* takes over the task of sampling the columns of G, and broadcasts its choices to the motes before the start of each incremental transmission from the WSN. In choosing the columns, the (converged) a-posteriori LLRs $\zeta = \zeta_{field}$ on X available to-date are utilized, in two ways. Firstly, unreliable bits are included more often in the parity checks, by using a selection probability for bit i that is proportional to $e^{-\zeta_i}$ (instead of a uniform selection probability). Secondly, the probability of degree-one parity checks ('dongles') is increased in proportion to $1/k\sum_{i=1}^{k}|\zeta_i|$. Thus in effect the FC "focuses" the parity checks onto unreliable parts of the field. In the first decoding attempt the focus is uniform (i.e.

[3]The level of the floor becomes *smaller* as the field becomes more *tightly correlated* and large compression gains are available.

identical to regular checking), but it gets sharper as more decoding attempts are made. (This has analogies to the *adaptive sampling* technique proposed in [167].) As a result, we do not need to wait for k/C transmissions to achieve very high fidelity. Of course this improvement comes at the expense of some additional system complexity, since we need more descriptive feedback from the FC to the WSN, rather than a simple stop/continue indication. Focused checking can be treated as an extension of Hybrid Automatic Repeat Request (HARQ) systems [168].

4.3.4 Discussion and Extensions

Comparisons with Distributed Source Codes

Unlike distributed source codes like DISCUS [151], our encoding procedure is simple and universal relative to the field or the channel. In our procedure, only the FC needs to know the field model for decoding, while DISCUS codes (hence encoders) have to be designed *a-priori* to 'match' the true model. Our scheme distributes the communication load evenly across the WSN and no special *time-sharing* schedule is needed. Also, in sharp contrast to DISCUS codes, our procedure is intrinsically robust to mote failures since it is not overtly dependent on a single mote or small subset of motes. The FC simply treats unavailable check-bits as erasures and assigns them zero intrinsic information while decoding. As long as the number of such failures is not large, the resulting degradation is negligible. Lastly, there is an important sense in which our scheme is more robust. The reliability and speed of model acquisition algorithms based on maximum-likelihood principles [122, 169][4] depend strongly on the fidelity of $\hat{X}$. A badly mismatched DISCUS code may give such a poor reconstruction fidelity that it may preclude robust acquisition

[4]The acquisition phase is analogous to the codebook generation phase of a universal compressor like Lempel-Ziv [5].

and therefore prevent a timely reprogramming of the motes with a new matched code. In contrast, our proposal has a built-in mechanism to achieve good fidelity *even when the field model is completely unknown*. We simply initialize the adaptive estimator with a maximally uninformative model (in our exemplary case of Section 2.3, by letting $\Theta = 0, \mathbf{W} = 0$), so that the extraction scheme is initially reduced to a *classical DFC*, like the scheme of [70]. Then good reconstruction fidelity is assured even during the acquisition phase, at a temporarily reduced power efficiency (about k/C transmissions per frame [80]), and hence acquisition can proceed robustly. As the acquisition is completed, $\Theta, \mathbf{W}$ are tuned to their correct values, and the power efficiency is progressively restored to its optimal level.

On the flip side, our method does depend upon a low-rate feedback channel that is used sporadically by the FC to direct the data extraction procedure, which is a reasonable assumption in the WSN scenario. We also require that the random number generators in the WSN and the FC be synchronized, and the motes keep track of the service requests and respond appropriately, which involves a small overhead. Finally, as mentioned earlier, we need some degree of inter-mote communication, though it can be kept at small levels via localized parity checks.

Extensions to M-ary or Real Fields, and General Channels

Although we have considered binary valued fields in this chapter, extensions to M-ary fields are straightforward because: (i) LT fountain codes can be formulated on general higher order Galois fields, (ii) graphical dependency models can be easily generalized to non-binary fields, and (iii) decoders based on belief propagation and mean field techniques tractably generalize to such non-binary alphabets. In fact we can even consider extensions to a real valued field as we did in [120], in which case our scheme specializes to a variant of *compressed sensing* [50, 53]. Similarly, while we consider the sensor-to-FC

channel to be a Binary Symmetric Channel (BSC) for numerical results in Section 4.5, our scheme is equally applicable to erasure, additive white Gaussian noise and fading channels, since LT fountain codes achieve good performance for all these cases [80]. The decoder merely needs to use the correct formula for the extrinsic information as a function of the channel outputs; the rest of data extraction schema remains unchanged.

4.4 Performance Analysis and Lower Bound

The purpose of this section is twofold. First, in Section 4.4.1 we will present a semi-analytical Extrinsic Information Transfer (EXIT) analysis [170] to estimate the performance of the proposed compression scheme. Then in Section 4.4.2 we will provide an information theoretic lower bound on the efficiency of the best extraction procedure possible, and also an approximation, which will serve as one benchmark for the proposed distributed extraction scheme.

4.4.1 Extrinsic Information Transfer Analysis

EXIT analysis (see [171] for an accessible introduction) is a well-known technique for estimating the performance of concatenated decoders, in particular serial concatenations as in our case. As usual, the extrinsic LLRs $\xi_{field}, \xi_{fountain}$ are postulated to be i.i.d. and to obey a *consistency condition*, namely that conditioned on X, they are Gaussian r.v.s. with equal magnitude of mean and variance (recall our comment after equation (4.5)). The information content of LLRs having a mean of magnitude σ^2 is given by

$$I(\sigma) = 1 - \mathbb{E}_{\xi \sim \mathcal{N}(\sigma^2, \sigma^2)} \left[\log_2 \left(1 + e^{-2\xi} \right) \right] \quad \text{bits/use.} \tag{4.12}$$

Though $I(\sigma)$ cannot be reduced to a closed form, it is a monotonically increasing invertible function that maps the parameter σ to the information content of ξ. Hence an inverse function $\sigma(I)$ is also well-defined. Let $T_{fount}(I_{field}, r)$ be the information content of the output of the fountain decoder, ξ_{fount}, as a function of the information content of its input, ξ_{field}. The novel aspect in our case is that the instantaneous rate of the fountain code, $r = \frac{k}{n}$, is treated as a parameter which gives rise to a *family* of characteristics. Furthermore, although the field has soft constraints, the field decoder also has a well-defined EXIT characteristic $T_{field}(I_{fount})$. Note that in this case there is no parameterization by r, since the field decoder does not have direct access to the channel outputs. The two EXIT characteristics are estimated by Monte-Carlo simulation of the decoders and plotted on a single graph, with interchanged ordinates and abscissa, as shown in Figure 4.2 (this will be described in more detail in Section 4.5). Noting that $T_{fount}(I_{field} = 0, r) > 0$ for all finite r, we are assured that the iterations can always 'get started' provided we decode the fountain code first, and then iterate between the two decoders. Thus, for each r, the subsequent sequence of I_{fount} and I_{field} can be read off by alternating between the two characteristics. In Figure 4.2 this is shown with a trajectory marked by arrows. The iteration gets stalled at the first intersection point $(I^*_{field}, I^*_{fount})$ of the EXIT curves, which corresponds to an error rate given by[5]

$$P_e(r) = Q\left(\sqrt{\sigma^2(I^*_{field}) + \sigma^2(I^*_{fount})}\right). \tag{4.13}$$

The error-rates can be thus calculated for all the various rates r of the fountain code. Noting that $r = \frac{1}{\rho}$, we obtain a $\rho^{EXIT}(P_e)$ characteristic, which serves as an estimate of the true characteristic of our data extraction procedure.

[5] $Q(\sigma) \doteq (2\pi)^{-1/2} \int_{\sigma}^{\infty} \exp(-s^2/2)\, ds$

4.4.2 Rate-Distortion and Capacity Bound

In Section 4.2.1 we allowed the parametric statistical model of the source, $P(x|\gamma)$, to be quite general, with the only requirement that the model be known to the FC. In this section we wish to calculate the analytical capacity-rate-distortion bound on performance of the data extraction system, which requires that we use some specific instance for the model. In particular, we will assume that the source is governed by the plume generation and dispersion model described in Section 2.3, which will also be used in our simulations. This model postulates that the sequence of field realizations $\{X^t\}$ form an irreducible MC whose state space is $\{+1, -1\}^N$, and whose transition probabilities are specified via an exponential family. Let η be the normalized *entropy rate* of such a source, in bits/sensor/sample, which is defined as [5, Chapter 4]

$$\begin{aligned} \eta &\doteq \frac{1}{N} H(X) \\ &\doteq \frac{1}{N} \lim_{t\to\infty} \frac{1}{t+1} H(X^t, X^{t-1}, X^{t-2}, \ldots, X^0) \\ &= \frac{1}{N} \lim_{t\to\infty} H(X^t | X^{t-1}, X^{t-2}, \ldots, X^0). \end{aligned} \tag{4.14}$$

While η can always be calculated exactly in principle, due to NP-hard complexity such an exact calculation is infeasible for large networks ($N \uparrow \infty$) having cyclic graphical models [143].

Recall that the Slepian-Wolf/Wyner-Ziv setup requires that there be *no cooperation* between the spatially distributed motes while compressing the field data. However our scheme *does not satisfy* this extreme constraint since we do allow some localized cooperation between the motes. Hence, rather that the Wyner-Ziv [64] rate distortion function $R^*(D)$, we must compare our performance with the classical rate-distortion function $R(D)$ of Shannon [56]. Surprisingly, except for a few special cases, $R(D)$ is typically not

known exactly (even among i.i.d. sources). Hence researchers, starting with Shannon, have calculated lower bounds to $R(D)$ [57] using variational principles. In particular, [58] has calculated the lower bound for a finite-alphabet finite-state Markov source X with a *balanced* distortion measure, which reads as

$$R(D) \geq H(X) - \phi(D)\log_2(e). \tag{4.15}$$

Here $H(X)$ (in bits) is the (un-normalized) entropy-rate of the process $\{X^t\}$ as given in equation (4.14), and $\phi(D)$ (in nats) is given by

$$\phi(D) = \max_{\alpha \geq 0} \alpha D - \log f_0(\alpha), \tag{4.16}$$

where

$$f_0(\alpha) = \Big(\sum_x \exp\left(-\alpha d(x,0)\right) \Big)^{-1}, \tag{4.17}$$

and $d(x,x') = d(x',x) \geq 0$ denotes the distortion between any pair of elements x, x' from the source alphabet. By treating α as a parameter, the optimum distortion is $D(\alpha) = \frac{\partial f_0(\alpha)}{\partial \alpha} \Big/ f_0(\alpha)$. [58] has shown that there exists a nonzero distortion $D_{crit} > 0$ such that for all $D < D_{crit}$, the inequality in equation (4.15) actually holds as an *equality*. Unfortunately, evaluation of D_{crit} seems intractable for all but the simplest sources [58]. Moreover, even for the simple example of a symmetric binary MC with Hamming distortion, D_{crit} is a very small number of the order of the cross-over probability of the chain, and the calculation of $R(D)$ for more interesting distortions $D > D_{crit}$ "appears an awesome task" [58].

Our model is a special case of the source considered by [58], with the state of the MC itself being the output of the source. Similarly, since our interest is in the probability of field symbol error P_e as given by equation (4.1), the appropriate distortion measure

between $x, x' \in \{+1, -1\}^N$ is given by

$$d(x, x') \doteq \frac{N - \sum_{i=1}^{N} x_i \, x'_i}{2N} = \frac{w_{\text{Hamming}}(\tilde{x} \oplus \tilde{x'})}{N}, \tag{4.18}$$

where $w_{\text{Hamming}}(\cdot)$ is the Hamming weight of the argument binary vector and $\oplus$ denotes addition modulo 2. (Recall that $\tilde{x}$ denotes the 0/1 representation of the $+1/-1$ spin vector x). Equation (4.18) is indeed a balanced measure as per definition of [58]. Also, it is easily shown that $f_0(\alpha) = \left(1 + e^{-\alpha/N}\right)^{-N}$ and that the optimizing distortion is $D(\alpha) = \left(1 + e^{\alpha/N}\right)^{-1}$. Hence it follows that $\phi(D) = Ng(D)/\log_2(e)$ nats, where $g(p) = -p\log_2(p) - (1-p)\log_2(1-p)$ is the binary entropy function in bits [5, Chapter 2]. Furthermore, due to the assumption of independence of the WSN-to-FC channels, we have a degenerate Multiple Access (MA) channel and hence there is no loss in achievable performance due to source-channel separation [5, 56, 57]. Consequently, the over-all extraction efficiency cannot be better than that obtained by optimally compressing all WSN data, and then transmitting it to the FC at the maximal possible rate. Hence the lower bound of [58] in our notation takes the form

$$\rho^{RD}(P_e) \geq \rho^{lower}(P_e) \doteq \frac{\eta - g(P_e)}{C}. \tag{4.19}$$

where C is the capacity of the sensor-to-FC channel, measured in bits per use, and $\rho^{RD}(P_e)$ is the minimum possible (rate-distortion) communication load. Note that it is merely a coincidence that $\eta - g(P_e)$ is also the rate distortion function of an i.i.d. Bernoulli source of entropy η bits/symbol [5, Chapter 13]. We would again like to stress that $\rho^{lower}(P_e)$ is only a lower bound to $\rho^{RD}(P_e)$ above the critical distortion and hence is not necessarily achievable with any coding scheme in that region (i.e. does not necessarily support a coding theorem). Only in the case of an i.i.d. Bernoulli($\frac{1}{2}$) field

we have $\rho^{RD}(P_e) = \rho^{lower}(P_e) = (1 - g(P_e))/C$, for all P_e.

To get another estimate of $\rho^{RD}(Pe)$ we now construct an approximation via the following thought experiment. We know that a sufficiently long sequence of field realizations, $\Omega_1 = \{X^1, X^2 \ldots, X^\tau\}$, can be *losslessly* compressed to a set Ω_2 of $\eta N \tau$ i.i.d. Bernoulli($\frac{1}{2}$) bits. We further know, from the rate-distortion function of an i.i.d. Bernoulli(p) source, that the bits in Ω_2 can be compressed to a set Ω_3 of $\eta N \tau (1 - g(P_e))$ bits and thereafter reconstructed back into a set Ω_4 such that the error rate between Ω_2 and Ω_4 does not exceed P_e. Now let Ω_5 be the reconstruction of the field symbols from the set Ω_4, and *assume* that the error rate between Ω_1 and Ω_5 is also P_e. This is an optimistic assumption, since the set Ω_4 is smaller than Ω_5 and hence the error rate will get magnified to some extent. On the other hand, the above two step procedure of lossless compression followed by lossy compression is only a subset of all possible schemes allowed by rate distortion theory, so it does not necessarily imply a true lower bound. Nevertheless, we conjecture that the transmission efficiency of the scheme in this thought experiment, which is given by

$$\rho^{approx}(P_e) = \frac{\eta\Big(1 - g(P_e)\Big)}{C}, \tag{4.20}$$

is a more realistic approximation of the ultimate possible transmission efficiency for non-asymptotic P_e, than $\rho^{lower}(P_e)$. This conjecture seems to be borne out in our simulations.

4.5 Simulation Results and Discussion

In this section we present extensive simulation results for the data extraction scheme proposed in Section 4.3 and compare them to the EXIT chart analysis of Section 4.4.1. Also, where feasible, we compare them to the information theoretic lower bound $\rho^{lower}(P_e)$ on

the best efficiency possible, and its approximation $\rho^{approx}(P_e)$, from Section 4.4.2. We will use a fountain code generated via the Shokrollahi degree distribution of [80]. For the field dependencies, we will use the model described in Section 2.3. We choose $G = 0.5I$ (here G denotes the temporal-dependency sub-matrix, *not* the fountain code generator matrix), and $\xi_3 = \xi = -0.7$ and $\xi_3 = \xi = -0.2$, thus respectively modeling a *moderately correlated* (small plumes) and a *strongly correlated* (large plumes) field. On the other hand by choosing $W_s = 0, G = 0, \theta_s = 0$ we also simulate a field whose spatiotemporal samples are *independently identically distributed* (i.i.d.) with equiprobable $+1$ and -1. In general, the weaker the correlations in the field, the larger is η, and the equiprobable i.i.d. field has the maximal value of $\eta = 1.0$ bits per sensor per sample.

For sensor-to-FC communication we will simulate a BSC with a cross-over probability p_{BSC}. The capacity of each such BSC is $C = 1 - g(p_{BSC})$ bits/use. The outputs of the channel are related to the inputs via

$$\tilde{Z}_i = \tilde{Y}_i \oplus \tilde{V}_i, \ i = 1, 2, \ldots,$$

where $\tilde{V}_i$, $i = 1, 2, \ldots$ are i.i.d. Bernoulli-0/1 random variables with $\Pr(1) = p_{BSC}$. The intrinsic information ξ_{chan} is therefore given by

$$\xi_{chan,i} = Z_i \cdot \frac{1}{2} \log \left(\frac{p_{BSC}}{1 - p_{BSC}} \right), \ i = 1, 2, \ldots, n, \tag{4.21}$$

where Z is the $+1/-1$ representation of $\tilde{Z}$ (cf. Section 4.3.1).

First we will consider a small network of $N = 8$ sensors, which allows us to do a complete information theoretic analysis. We can explicitly calculate [5, Chapter 4] $\eta = 0.21$ bits per-sensor per-sample for the moderately correlated field and $\eta = 0.045$ bits per-sensor per-sample for the strongly correlated field. We choose a latency of

$\tau = 256$ samples, implying that the frame size is $k = 2048$ bits.

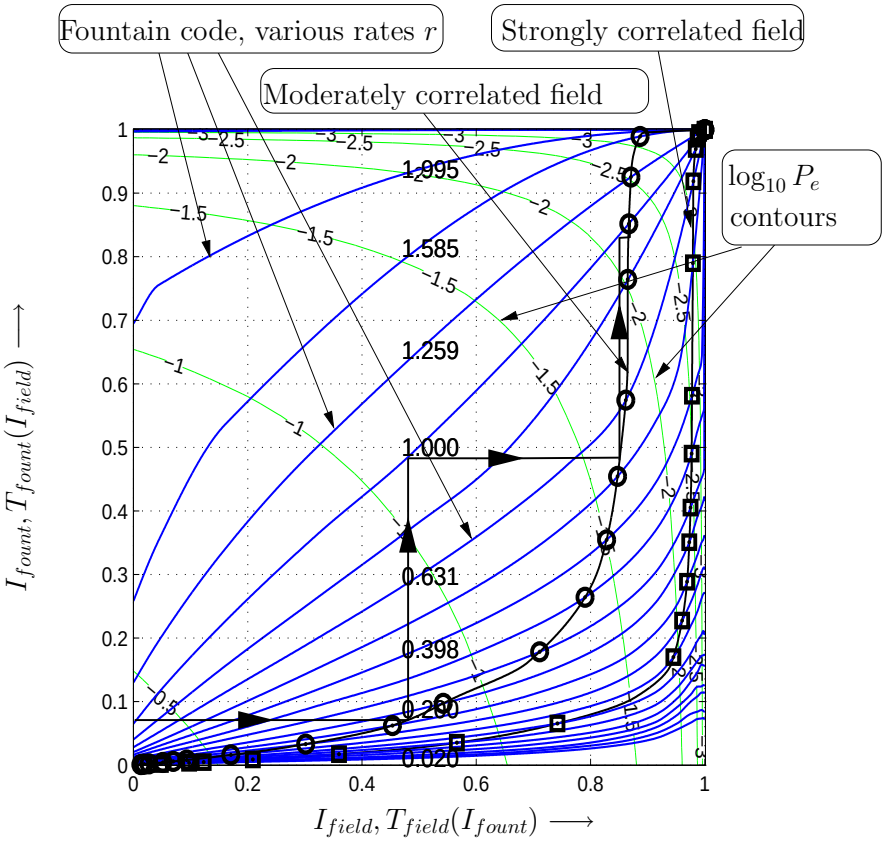

Figure 4.2: EXIT chart analysis for $N = 8, \tau = 256, C = 0.5$ bits/use.

Figure 4.2 displays the EXIT chart analysis for this setup for a communication channel of capacity $C = 0.5$ bits/use (hence $p_{BSC} = 0.110$). For clarity, the trajectory of the iterated decoder is shown only for one exemplary case, namely $r = 1.0$ and a moderately correlated field. In general, circle or square markers are the fixed points of the iterated decoder for the corresponding field type and fountain code rate. In the background we have also shown a few contours of the error probability (green lines), labeled with the logarithm to the base ten. For each type of field (strongly correlated, moderately correlated and i.i.d.), the probability contour level of each fixed point is read off from the chart and then plotted in Figure 4.4. The figure also displays the loose lower

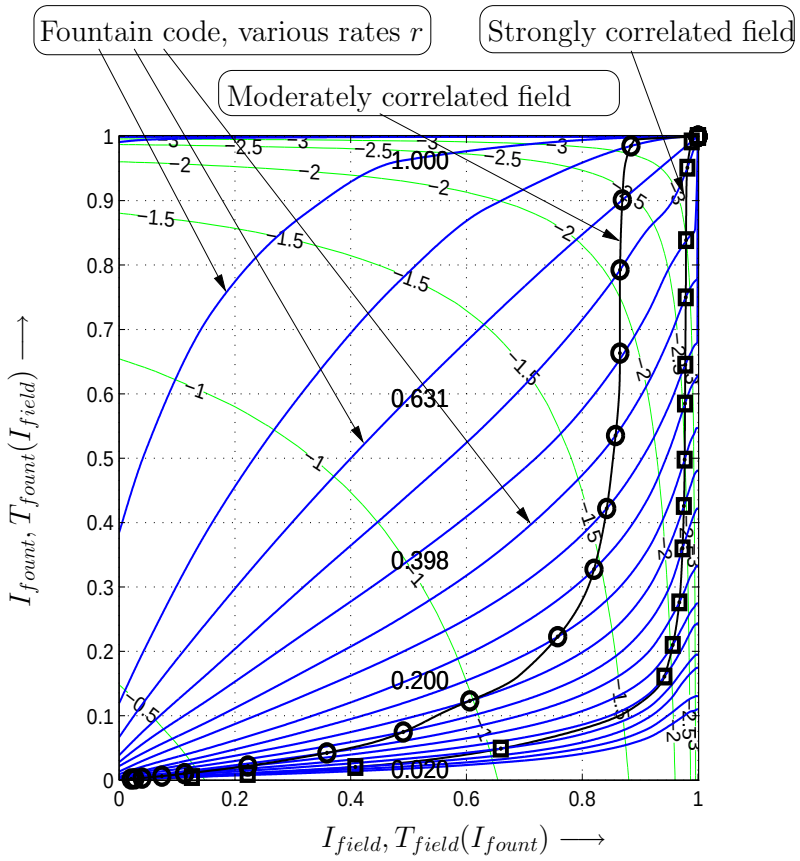

Figure 4.3: EXIT chart analysis for $N = 8, \tau = 256, C = 0.9$ bits/use.

bound $\rho^{lower}(P_e)$ given by equation (4.19), the rate-distortion approximation $\rho^{approx}(P_e)$ of equation (4.20) and curves obtained from simulation, $\rho^{simul}(P_e)$. Notice that we use a *logarithmic* scale on the x-axis, which allows a simultaneous comparison of compression gains under various correlation strengths. Similarly, Figure 4.3 displays the EXIT analysis for a communication channel of capacity $C = 0.9$ bits/use (hence $p_{BSC} = 0.013$), and Figure 4.5 shows the corresponding error probability curves given by analysis and simulations. Note that the simulations presented in Figures 4.4 and 4.5 are made with regular checking and without any localization constraints on the parity check matrix G.

We make the following important observations that highlight the universality and

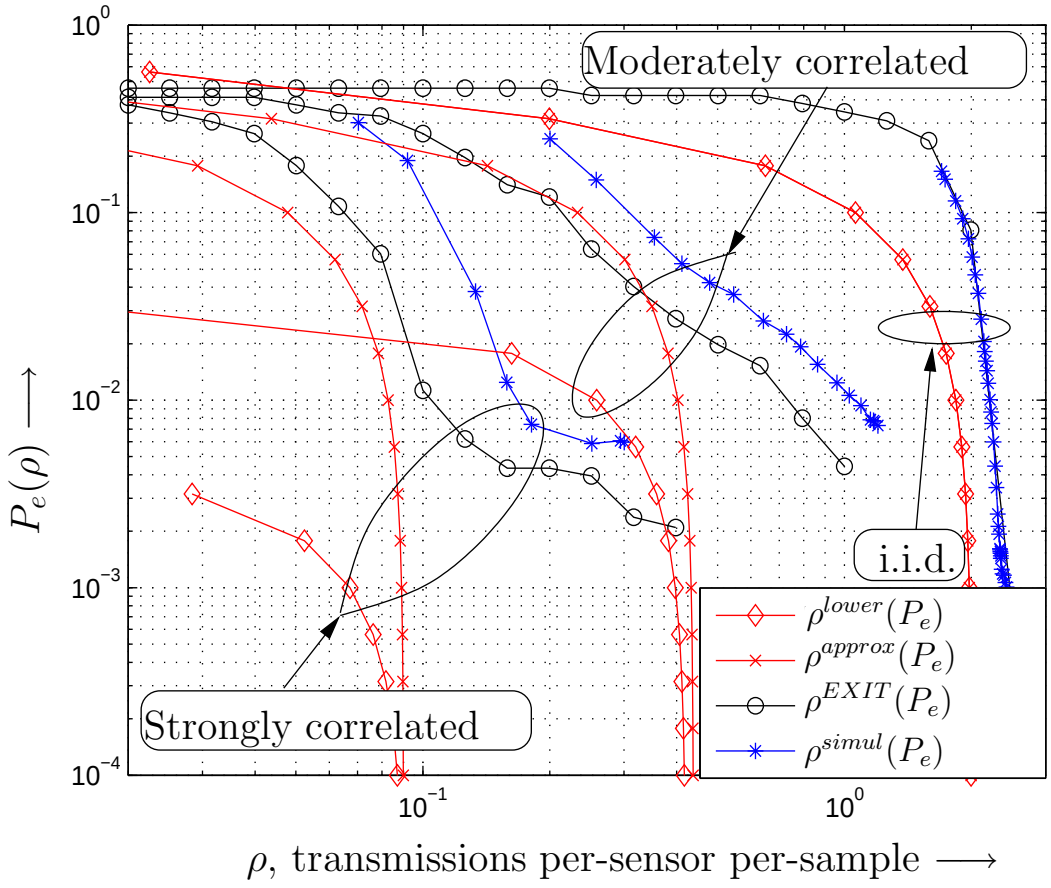

Figure 4.4: $P_e(\rho)$ characteristic for $N = 8, \tau = 256, C = 0.5$ bits/use, and three types of field models: strongly correlated ($\xi = -0.2$), moderately correlated ($\xi = -0.7$), and i.i.d. ($\theta = 0, W = 0$).

efficiency of our scheme: **(a)** For *every* combination of channel and field model, there is good agreement of $P_e(\rho)$ between the simulations and the EXIT analysis. The EXIT charts predict a relatively quick convergence for the iterated decoder (5 to 10 iterations), which was verified to be true in simulations too. **(b)** For *every* combination of channel and field model, and for moderate P_e, the simulations respect the loose lower bound $\rho^{lower}(P_e)$, and are very close to the approximation $\rho^{approx}(P_e)$. This suggests that the proposed scheme is close to being optimally power efficient. For very small P_e, there is a flooring effect which we will discuss shortly. **(c)** When the FC assumes an i.i.d.

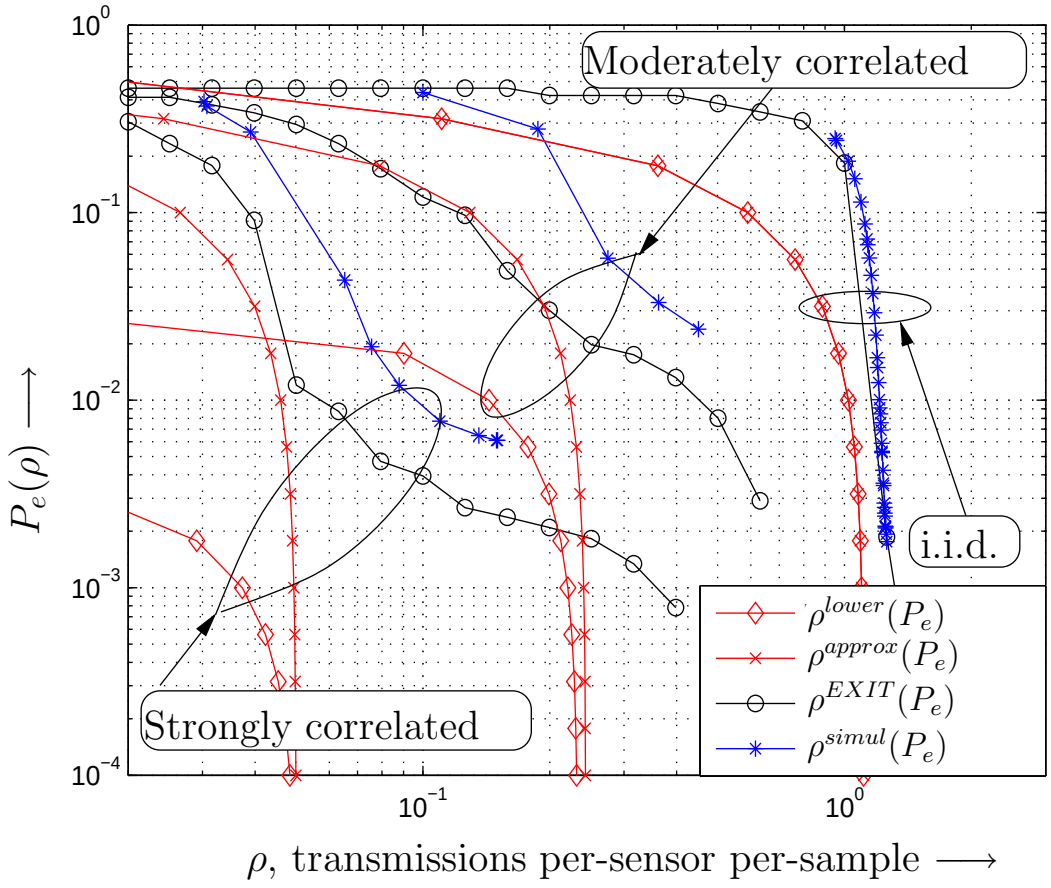

Figure 4.5: $P_e(\rho)$ characteristic for $N = 8, \tau = 256, C = 0.9$ bits/use, and three types of field models: strongly correlated ($\xi = -0.2$), moderately correlated ($\xi = -0.7$), and i.i.d. ($\theta = 0, W = 0$).

field, the field decoder produces identically zero extrinsic information and hence it could even be turned off. In this case we specialize to a stand-alone DFC scheme like [70]. Moreover, the DFC, like all linear codes, has the Uniform Error Property (UEP), so its $P_e(\rho)$ characteristic is invariant to whether the actual encoded field values are i.i.d. or correlated. Thus the 'i.i.d.' curve in Figures 4.4 and 4.5 (and in all later figures) is the performance of a baseline DFC scheme such as [70] simultaneously for a *correlated field* as well as a truly *i.i.d.* field, and hence is the correct benchmark against which to measure compression gains. We observe that large gains, **of the order 6 to 13 dB**,

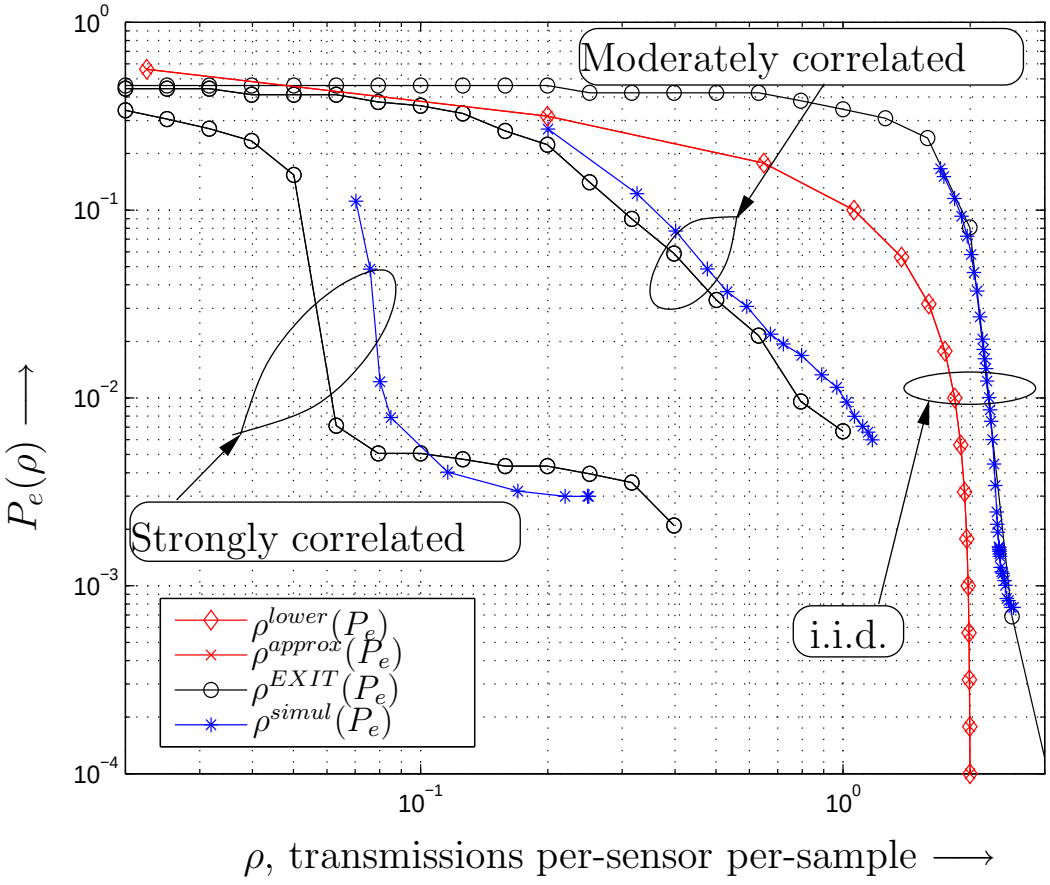

Figure 4.6: $P_e(\rho)$ characteristic for $N = 128, \tau = 16, C = 0.5$ bits/use, and three types of field models: strongly correlated ($\xi = -0.2$), moderately correlated ($\xi = -0.7$), and i.i.d. ($\theta = 0, W = 0$).

are practically achievable at moderate P_e.

Now we turn our attention to a large array of $N = 128$ sensors. We continue to use the *same* field model as before. For ease of comparison (to allow the use of an identical fountain code) we again choose the frame size $k = 2048$, so the latency now is $\tau = 16$. As remarked earlier, in this case η cannot be tractably calculated, so we cannot plot information theoretic bounds or approximations for the moderately and strongly correlated field (but we can still do so for the i.i.d. field). Figures 4.6 and 4.7 show the error probability curves given by bounds, analysis, and simulations for a communication

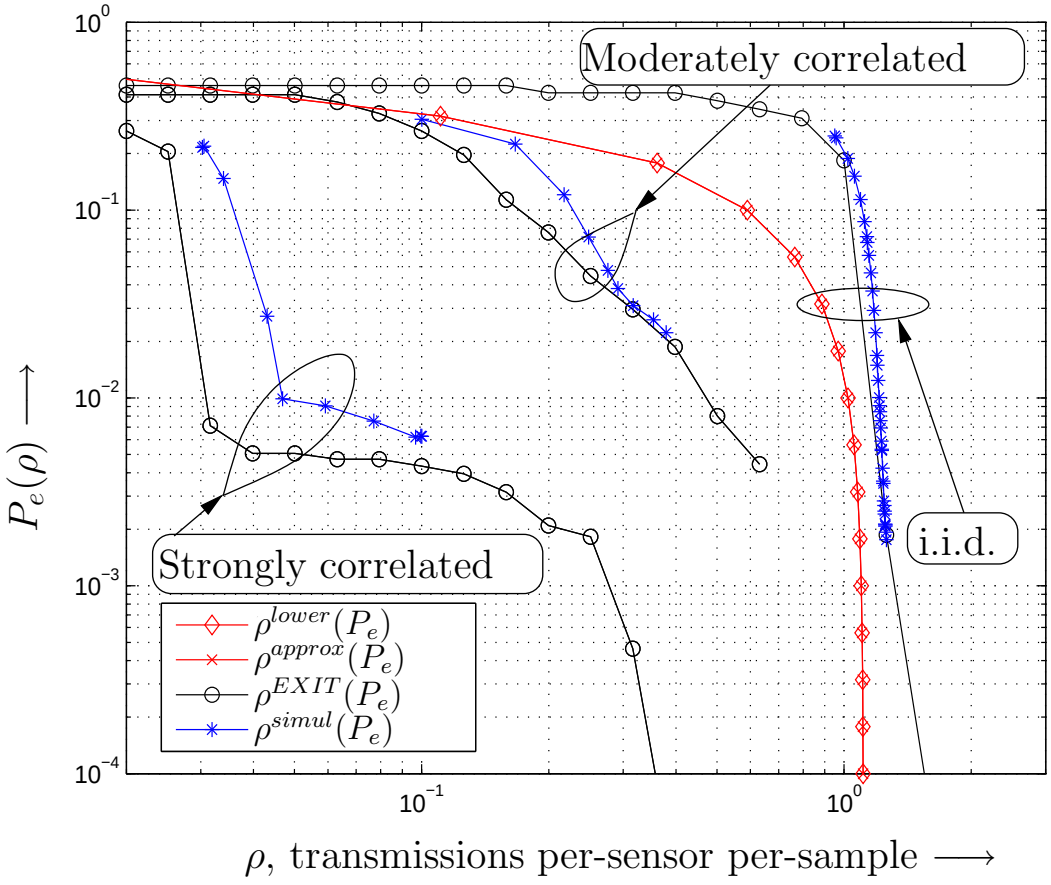

Figure 4.7: $P_e(\rho)$ characteristic for $N = 128, \tau = 16, C = 0.9$ bits/use, and three types of field models: strongly correlated ($\xi = -0.2$), moderately correlated ($\xi = -0.7$), and i.i.d. ($\theta = 0, W = 0$).

channel of capacity $C = 0.5$ bits/use and $C = 0.9$ bits/use respectively. We again observe that there is good agreement between the analysis and the simulations. We also observe that the performance is not qualitatively different from the case of the small array, which implies a crucial scalability property: for *a given homogeneous field model*, the normalized communication load $\rho(P_e)$ is approximately *independent* of the size of the network.

In Figures 4.8 and 4.9 we provide simulative evidence for the utility of the focused checking procedure, proposed in Section 4.3.3, in improving the floor of the $P_e(\rho)$ char-

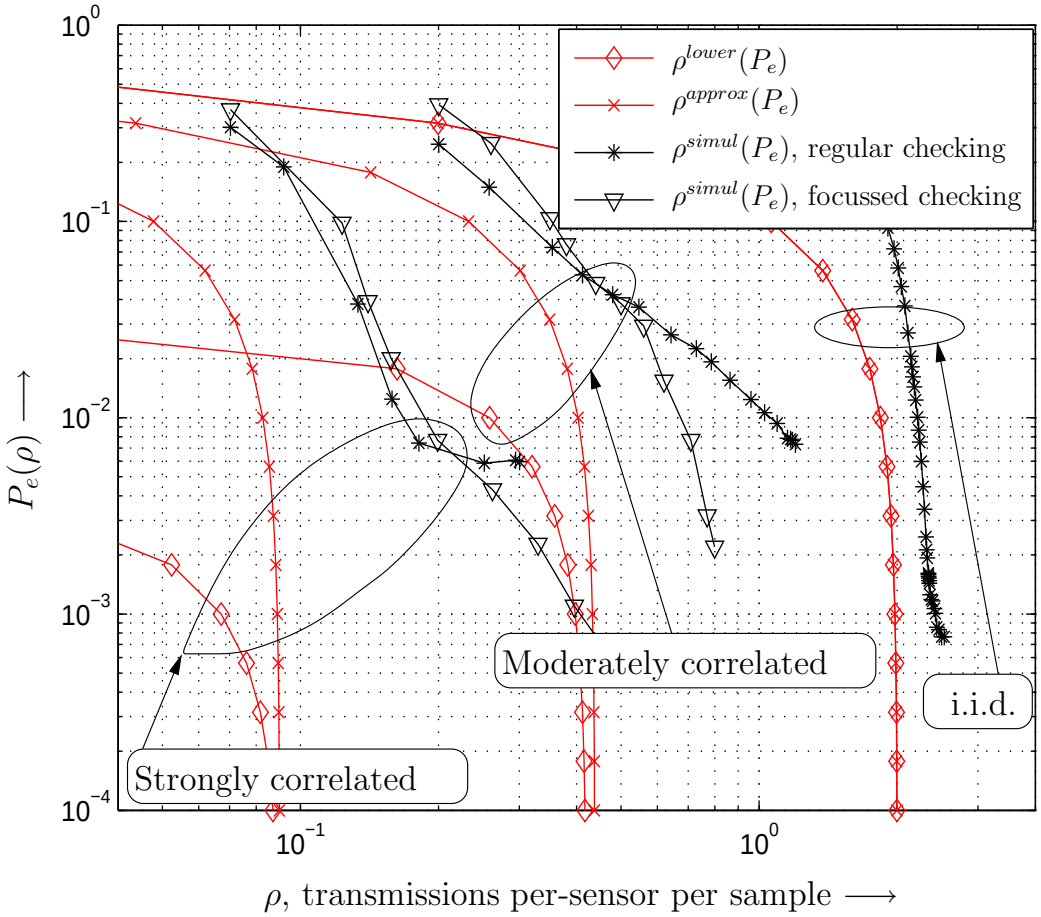

Figure 4.8: Effect on $P_e(\rho)$ of using a focused checking generator matrix G, as compared to regular checking. Network size $N = 8$. Frame size $k = 2048$. Channel capacity $C = 0.5$ bits/use.

acteristic relative to regular checking. The generator matrix G is still selected without localization constraints. For brevity, we limit our attention only to a communication channel of capacity $C = 0.5$ bits/use. We see that in each case, focused checking greatly improves the floor, in line with the predictions made in Section 4.3.3. Thus, by paying the overhead of a descriptive feedback, we can achieve large compression gains even under very stringent fidelity requirements. We note that the slope at low $P_e(\rho)$ can be further improved by the use of Raptor instead of LT fountain codes [80].

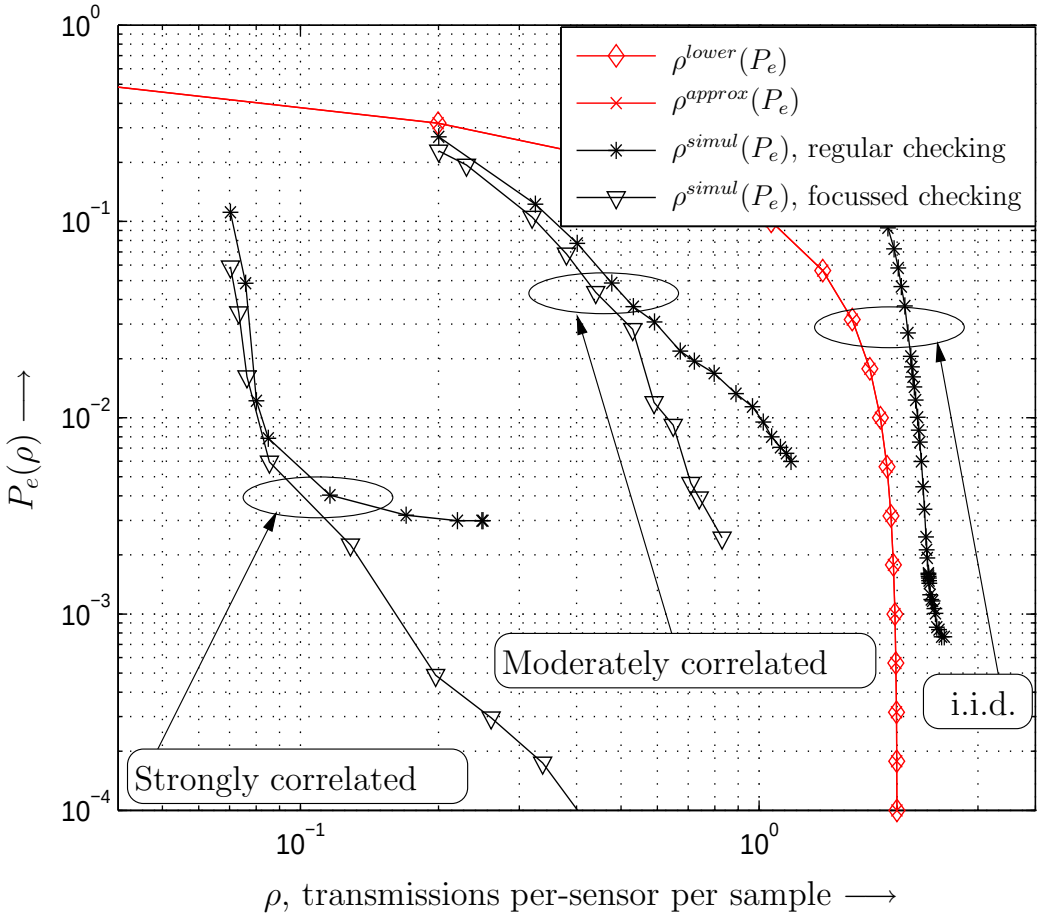

Figure 4.9: Effect on $P_e(\rho)$ of using a focused checking generator matrix G, as compared to regular checking. Network size $N = 128$. Frame size $k = 2048$. Channel capacity $C = 0.5$ bits/use.

Finally we will consider the effect of localization constraints on G, which may be necessary in certain practical settings. We will continue to consider the setup of $N = 128$ sensors, with a frame size of $k = 2048$. The channel capacity is again $C = 0.5$ bits/use. Figure 4.10 shows the simulation of the error probability curves, with several localizations constraints defined via a message passing *locale* l. For a given l, $G(l)$ is chosen such that every mote needs to exchange messages with other motes only in a physical neighborhood of l meters or less. Recall that in the model presented in Section 2.3, the sensors are

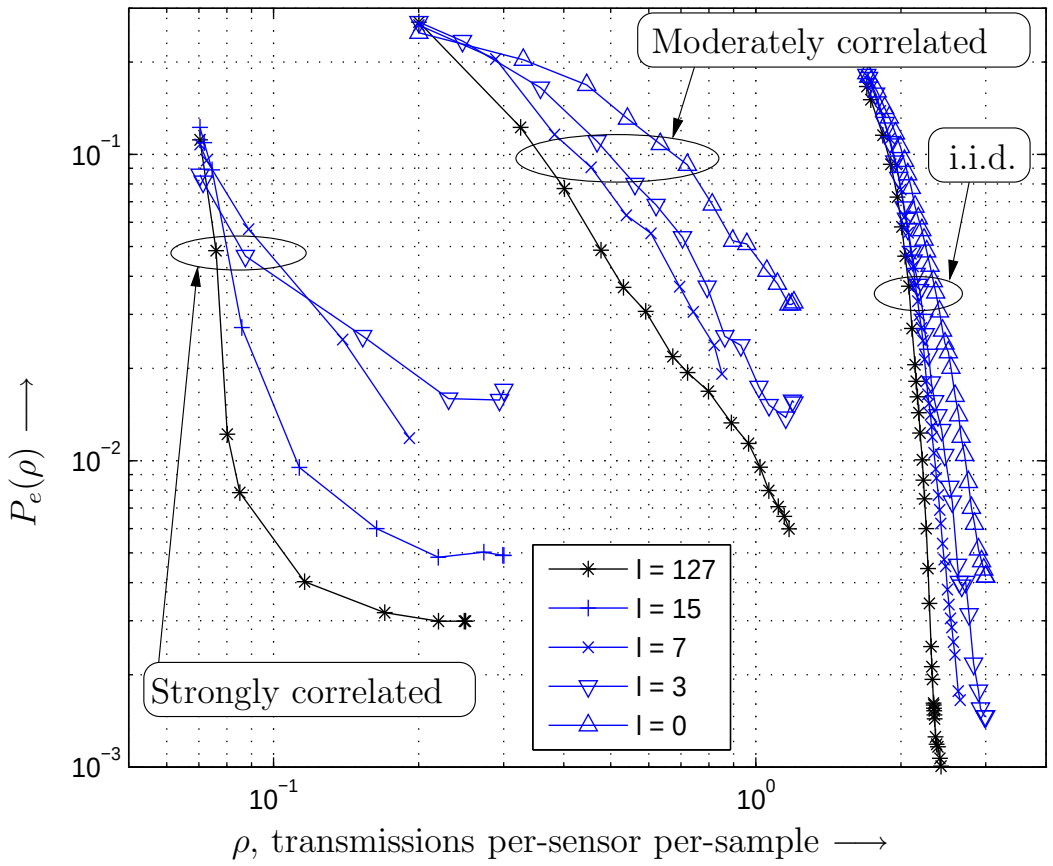

Figure 4.10: Effect on $P_e(\rho)$ of using a generator matrix G satisfying localization constraints on message passing. l is the *locale* (defined in Section 4.5), specified in meters, and the $N = 128$ motes are one meter apart. Channel capacity $C = 0.5$ bits/use.

deemed to be located one meter apart. Thus, in this case, $l = 127$ implies no localization constraints while $l = 0$ implies non-cooperative encoding. We observe that when $l \geq 7$ meters, the degradation in performance w.r.t. $l = 127$ is relatively small for the i.i.d. and the moderately correlated field, while $l \geq 15$ is found to be sufficient for the strongly correlated field. Note that the direct statistical interactions in the field are limited to a neighborhood of 3.0 meters (cf. Section 2.3). This suggests that if we choose the locale to be moderately larger than the *statistical interaction neighborhood*, most of the potential power efficiency can be achieved. It follows that if a network scales while

maintaining a constant statistical interaction neighborhood, the message-passing cost per sensor can be kept *bounded*.

4.6 Conclusions

We have presented a universal scheme for efficient data extraction from a wireless sensor network, based on joint source-channel decoding. It uses a low complexity distributed encoder that is independent of the statistical model of the field or the channel, and still achieves a power efficiency close to the information theoretic limit. The latency of the procedure is very small compared to universal compression codes, and robustness to mote failures is an intrinsic property. The procedure can accommodate strict localization constraints on inter-mote message passing without sacrificing performance significantly. By using a novel technique called focused checking, large compression gains can be achieved even under a high fidelity requirement. We also demonstrated a method to accurately and rapidly predict performance using EXIT charts.

5 Distributed Model Identification

5.1 Introduction

As we discussed in Chapter 1, much research has been recently directed towards the exploitation of the spatiotemporal dependencies in the sensor data to improve the power efficiency of the network via strategies like Distributed Source Coding [62], Correlated Data Gathering [63], Source-Channel Decoding [118], [120], Distributed Detection [13, 15], Distributed Filtering [115], Distributed Learning [30, 31], and Energy Aware Routing [84]. Our own contributions in this respect were presented in Chapters 3 and 4.

A common theme in all these approaches is that some knowledge of the statistical model of the field is a prerequisite, e.g. to design optimal codes [62] or routing tables [84]. Since Maximum Likelihood (ML) identification via the Expectation Maximization (EM) algorithm of [122] is typically intractable for large models, low complexity stochastic recursive algorithms have been proposed in literature, which also have the advantage that they can seamlessly track model variations. For example, [172] has proposed an incremental EM algorithm for finite mixture models, [173] and [174] have proposed stochastic recursive estimators for the parameters of Hidden Markov Chain Models (HMMs), and [136, 137, 138] have proposed stochastic EM algorithms (SAEM) for spatial Hidden Markov Random Fields (HMRFs) in the context of image processing. Moreover, [174] and [136, 137, 138] use a Markov Chain 'Monte-Carlo' (MCMC) approximation [127]

for their stochastic innovations, since an exact calculation is infeasible for large cyclical MRFs. A nice overview of incremental EM algorithms can be found in [121].

However, all the above noted researchers [121, 136, 137, 138, 172, 173, 174] have exclusively considered *centralized* identification schemes, since there is no incentive for distributed algorithms in applications like image processing. In the WSN scenario on the other hand, transmission of all raw observation data to the FC for centralized identification requires energy-intensive communication, which greatly diminishes the *net* energy savings possible from the knowledge of the model. Therefore, it is clearly advantageous if the model identification can be done *in situ*, i.e. within the network itself. Any such proposal, however, must also satisfy the special constraints of the motes, and should be scalable with the size of the network, which makes a *distributed* algorithm mandatory. This novel problem of *distributed in-situ* model identification has received only a limited attention to date, e.g. identification of linear models [23, 175], incremental EM identification of fully observed mixture models [176], and supervised identification of MRF models [15]. To our knowledge, the practical problem of *blind* identification of *hidden nonlinear* probabilistic field models like HMRFs, using a *spatially distributed* algorithm, seems to be largely unexplored.

Contributions and organization of this chapter: In this chapter we address the above mentioned gap in literature by proposing a distributed incremental identification algorithm for a class of probabilistic models often encountered in WSN applications, namely, exponential/HMRF models [15, 24], which we have described in some detail in Chapter 2. We specify our proposed incremental identification algorithm in Section 5.2. While it is similar to the SAEM algorithm of [136, 137, 138] in that it makes a stochastic gradient ascent of the observation log-likelihood using MCMC techniques, we analyze our algorithm w.r.t. four issues crucial in a distributed implementation, namely (i) stability (Section 5.2.1), (ii) covariance efficiency (Section 5.2.2), (iii) scalability and power

efficiency (Section 5.3), and (iv) robustness to early termination of MCMC iterations (Section 5.4). Section 5.6 concludes the chapter.

5.2 Incremental Parameter Estimation, Stability and Covariance Efficiency

Please note that the necessary mathematical notation pertaining to exponential families is collected in Section 2.5, which the reader may wish to consult at this point. For simplicity, **we will initially specialize the spatio-temporal model of Chapter 2 to a strictly spatial model, where the temporal field realizations are i.i.d.**. This means that $Q([x^{t^T}, x^{t-1^T}]^T|\gamma)$ factorizes as

$$Q([x^{t^T}, x^{t-1^T}]^T|\gamma) = q(x^t|\gamma)q(x^{t-1}|\gamma).$$

Clearly $q(\cdot|\gamma)$ consistently satisfies the marginal equation (2.7), and is the common distribution from which each spatial field pattern is drawn. Since $Q([x^{t^T}, x^{t-1^T}]^T|\gamma)$ is presumed to be an exponential distribution, $q(\cdot|\gamma)$ is also exponential. For example, recall from Section 2.3 that for the Boltzmann field (and similarly for the GMRF), $G = 0$ will ensure the temporal i.i.d. property, and in this case

$$q(x|\theta_s, W_s) \doteq \exp\left\{x\theta_s + \frac{1}{2}x^T W_s x - \Psi(\theta_s, W_s)\right\}. \tag{5.1}$$

In the following material till Section 5.4.3, we will use the information-geometric convention and properties discussed in Section 2.5 as applied to the case of $q(x|\gamma)$ rather than the meta-state distribution $Q(z|\gamma)$. Hence in the relevant expressions we must substitute $Z, U, F_\gamma^Z, F_\gamma^U, \theta, W$ etc respectively with $X, Y, F_\gamma^X, F_\gamma^Y, \theta_s, W_s$ etc. Thus for

example, $\pi(\cdot|\gamma)$ from equation (2.16) gets re-defined as

$$\pi(y|\gamma) = \sum_{X} q(x|\gamma)P(y|x).$$

Also, the log-likelihood ratio statistic $h(U)$, a vector of length $2N$ defined by equation (2.28), is replaced by $h(Y)$, a vector of length N.

Structural knowledge of $q(x^t|\gamma)$ means the knowledge of its basis function, and this is often available from physical considerations. However, parametric knowledge, i.e. the actual numerical value of γ, is rarely available a-priori, and needs to be blindly estimated based on the observations. In this respect, the question of *identifiability* is important and needs to be addressed first. Can two (or more) distinct parameter values γ_0, γ_1 lead to an identical distribution, i.e. $\forall x^t,\ q(x^t|\gamma_0) = q(x^t|\gamma_1)$? It is known in the context of exponential families [146] that the family of distributions $\aleph = \{q(x^t|\gamma)|\text{all valid } \gamma\}$ is *uniquely* parameterized *if and only if* the basis functions are affinely independent. Hence, given affine independence (therefore a 'minimal' model), there is an unambiguous parameter value that will maximize the likelihood of an asymptotically large set of realizations from the model $q(\cdot|\gamma)$. We will always assume affine independence is this thesis, which can be easily satisfied by construction.

Our aim is the *maximum-likelihood* (ML) estimation of the parameter of the hidden exponential model/HMRF, because, as noted in Section 2.5, ML estimators are asymptotically unbiased and efficient. However this is typically an intractable problem that requires a full-fledged EM algorithm [122]. Instead, we wish to devise a distributed tractable approximation. Various researchers [136, 137, 138, 172, 173, 174] have proposed, with minor variations, a stochastic EM algorithm (SAEM) as a tractable method to achieve the goal of ML estimation in lieu of a full-fledged EM algorithm. We will, in particular, use a 'partial M-step' variant [121], where we start with some initial estimate

γ^1, which is recursively updated as

$$\gamma^{t+1} \;=\; \gamma^t + \epsilon\, A\, S_{\gamma^t}(Y^t). \tag{5.2}$$

In this recursion, $S_{\gamma^t}(Y^t)$ is the *score* of Y^t, $\epsilon > 0$ is a user-defined *step size*, and A is a *constant* user-defined symmetric positive-definite pre-scaling matrix (hence this recursion is not doubly stochastic like [138]). The rationale behind the choice of ϵ and A will be discussed in Section 5.2.2.

5.2.1 Asymptotic Stability of the Expected Gradient System

Using the Averaged-ODE method [177, 178, 179] it is sometimes possible to provide guarantees of almost-sure convergence of recursion (5.2) assuming a time-invariant truth-model, a suitably decaying sequence of step sizes, and certain strict regularity properties like scale invariance and global asymptotic stability. While realistic models may not necessarily possess all these regularity properties, expected gradient analysis nevertheless gives useful information about the stability and convergence of such algorithms. The basic premise is that, with a small enough step size, the trajectory of the stochastic recursion gets coupled to the trajectory of a *deterministic* recursion driven by an 'averaged' increment, where the averaging is done over all the sources of stochasticity. Denote the stochastic and expected gradients respectively as

$$f(\gamma^t, Y^t) \doteq A\, S_{\gamma^t}(Y^t)$$

and

$$\bar{f}(\gamma^t) \doteq \mathbb{E}_{\pi(y^t|\gamma^*)}\left[f(\gamma^t, Y^t)\right],$$

where note that the expectation is done under the *truth model* with $\pi(\cdot)$. Then we have the following theorem, which appears to be original:

Theorem 3 *There exists a threshold clutter variance $\sigma^2_{thresh} > 0$ and a corresponding threshold step size $\bar{\epsilon} > 0$ such that, when $\sigma^2 < \sigma^2_{thresh}$ and $0 \leq \epsilon \leq \bar{\epsilon}$, γ^* is as an asymptotically stable (A.S.) fixed point of the expected gradient recursion*

$$\bar{\gamma}^{t+1} \;\; = \;\; \bar{\gamma}^t + \epsilon \; \bar{f}(\bar{\gamma}^t). \tag{5.3}$$

The proof of the theorem appears in Appendix D.1.

Remark 1 *While the theorem above states that the expected gradient system is A.S. for a sufficiently large SCR, we conjecture, based on numerical calculations, that $F^Y_{\gamma^*} > 0$ and A.S. holds at all SCRs.*

Remark 2 *While $l_\gamma(\cdot)$ is strictly concave in γ, $L_\gamma(\cdot)$ is not, since numerical evaluation of $\frac{\partial \bar{f}(\gamma)}{\partial \gamma}$ at points sufficiently far away from γ^* yields a sign-indefinite matrix. Thus γ^*, in general, may not be globally A.S.*

(Recall from Section 2.5, equations (2.20) and (2.21), that $l_\gamma(\cdot)$ and $L_\gamma(\cdot)$ are respectively the direct and indirect observation log-likelihoods.)

5.2.2 Efficiency Analysis

Let us define the *error* w.r.t. the averaged gradient recursion to be $\alpha^t \doteq \gamma^t - \bar{\gamma}^t$. Then by subtracting equation (5.3) from equation (5.2), we obtain the *error evolution equation*

$$\alpha^{t+1} = \alpha^t + \epsilon A \left(S_{\gamma^t}(Y^t) - \mathbb{E}_{\pi(y^t|\gamma^*)} \left[S_{\bar{\gamma}^t}(Y^t) \right] \right). \tag{5.4}$$

Recall that we showed in Theorem 3 that $\bar{\gamma}^t \to \gamma^*$ provided γ^0 is in the attractor of γ^*. We would like to analyze the error evolution in the regime of small deflections of γ^t around the putative lock point γ^*. Hence we assume that the error asymptotically converges to a wide-sense stationary phase with zero mean, and *linearize* equation (5.4) around zero error. That is, we expand $S_{\gamma^*}(Y^t)$ and $\mathbb{E}_{\pi(y^t|\gamma^*)}\left[S_{\gamma^*}(Y^t)\right]$ in a Taylor series around γ^* and then ignore second and higher order terms to get

$$\alpha^{t+1} = (I - \epsilon A F^Y_{\gamma^*})\alpha^t + \epsilon A S_{\gamma^*}(Y^t). \tag{5.5}$$

Now we take second order moments on both sides and use the independence property $\alpha^t \perp Y^t$, to obtain

$$\begin{aligned}\mathbb{E}\left[\alpha^{t+1}\alpha^{t+1^T}\right] = \; & (I - \epsilon A F^Y_{\gamma^*})\mathbb{E}\left[\alpha^t \alpha^{t^T}\right](I - \epsilon A F^Y_{\gamma^*}) \\ & + \epsilon^2 A F^Y_{\gamma^*} A.\end{aligned} \tag{5.6}$$

Let the time-invariant covariance matrix of the error in the stationary phase be denoted by Σ_α. Then it must satisfy the equation $\Sigma_\alpha = (I - \epsilon A F^Y_{\gamma^*})\Sigma_\alpha(I - \epsilon A F^Y_{\gamma^*}) + \epsilon^2 A F^Y_{\gamma^*} A$, whose solution is given by[1]

$$\Sigma_\alpha = \epsilon^2 \sum_{i=0}^{\infty} (I - \epsilon A F^Y_{\gamma^*})^i A F^Y_{\gamma^*} A (I - \epsilon A F^Y_{\gamma^*})^i. \tag{5.7}$$

An important special case arises when we choose $A = (F^Y_{\gamma^*})^{-1}$, where the solution simplifies to

$$\Sigma_\alpha = \frac{\epsilon}{2 - \epsilon}(F^Y_{\gamma^*})^{-1}. \tag{5.8}$$

[1]The series converges provided ϵ is chosen small enough.

Lemma 4 *With $A = (F^Y_{\gamma^*})^{-1}$, the incremental estimator (5.2) is asymptotically efficient.*

The lemma is proved in Appendix D.2.

Remark 3 *To achieve the CRLB we need to use the ideal pre-scaling matrix $A = (F^Y_{\gamma^*})^{-1}$. This also has the useful consequence of making recursion (5.2) a "natural gradient" recursion [149], where the spread in the speeds of slow and fast manifolds is small, and convergence is uniform.*

Remark 4 *In practice, of course, we must presume some nominal value γ^{nom} and set $A = (F^Y_{\gamma^{nom}})^{-1}$. It is worth reemphasizing that the choice of a suboptimal $A > 0$ does not cause any loss in stability or unbiasedness (cf. Theorem 3). Hence we may intermittently recalculate $A = (F^Y_{\gamma^{nom}})^{-1}$ where γ^{nom} is the temporal average of the recent record of γ^t, and thus assure an efficiency close to the CRLB.*

Remark 5 *Even if we have a fairly good nominal estimate $\gamma^{nom} \approx \gamma^*$, we may still choose to use a sub-optimal pre-scaling matrix A in order to satisfy certain localization constraints on message passing (c.f. Section 5.3). In this case, we must ensure the normalization $\|A\| = \|(F^Y_{\gamma^{nom}})^{-1}\|$ is maintained, so that the estimator's tracking speed is consistently determined by ϵ alone.*

Remark 6 *Numerical calculations indicate that when $\|\gamma^*\|$ is large the condition number of $F^X_{\gamma^*} = F_{\gamma^*}$ degrades. In light of the stability and efficiency analysis presented earlier, this implies that the relative stochastic stability of the recursion in equation (5.2) degrades. However note that the Kullback-Leibler (KL) divergence[2] of the estimated model*

[2]Recall that the KL-divergence of a distribution f_2 relative to a distribution f_1 is defined as $D(f_1(x)\|f_2(x)) \doteq \mathbb{E}_{f_1(x)}\left[\log \frac{f_1(X)}{f_2(X)}\right]$.

w.r.t. the truth model is given (in nats) by

$$\begin{aligned} D(\gamma^*\|\gamma^t) &\doteq D(Q(x|\gamma^*)\|Q(x|\gamma^t)) \\ &= \frac{1}{2}(\gamma^* - \gamma^t)^T F^X_{\gamma^*}(\gamma^* - \gamma^t) + \text{H.O.T.} \end{aligned} \tag{5.9}$$

It is known [5] that the KL-divergence, rather than the mean squared error (MSE), is the key information theoretic measure for quantifying the mismatch loss in hypothesis testing. Hence, depending on $F^X_{\gamma^*}$, a large MSE need not necessarily translate into a proportionately large KL-divergence. Secondly, though a small $F^X_{\gamma^*}$ leads to a large MSE via the CRLB, its presence in equation (5.9) mitigates the final effect on $D(\gamma^*\|\gamma^t)$.

5.3 Distributed Implementation, Scalability, Power Efficiency

In Section 5.2 we showed that the proposed estimator has desirable properties of stability and covariance efficiency. Now we will tackle the question of distributed implementation and power efficiency.

5.3.1 Choice of Algorithm For Calculating Expectations

The calculation of the gradient, $A\,S_{\gamma^t}(Y^t)$ is, in general, an NP-hard problem w.r.t. N, and the use of some kind of approximation is mandatory. Moreover, we should exploit the *broadcast communication* characteristic of the WSN. Four kinds of broadcast message-passing approximate marginalization algorithms are known in literature (cf. Section 3.2.3, and also see [115] and references therein): Gibbs Sampling (GS), Mean Field Decoding (MFD), Iterated Conditional Models (ICM) and Broadcast Belief

Propagation (BBP). Unfortunately, while BBP, MFD and ICM have low complexity and give good performance for distributed filtering, they are not well suited to parameter estimation because they produce biased estimates of expectations of higher order (non-linear) basis functions, which leads to biases and instabilities in the recursive estimator (5.2). GS on the other hand always gives stable and *unbiased* estimates for the expectations of all the basis functions. Hence, similar to [174][136][137][138], we prefer its use for calculating the gradient needed by the recursion. In Section 5.3.2 we will describe the distributed implementation of the incremental estimator based on GS, and in Section 5.3.3 we will analyze its total power consumption in detail.

5.3.2 Distributed Scalable Gradient Computation Using GS

For clarity of exposition, we will describe the gradient computation for the specific example of the Boltzmann model from Section 2.3 where $\gamma^t \equiv (\theta_s^t, W_s^t)$. It is easy to generalize, by analogy, to other exponential models. It is known that GS is a stochastic method that yields an arbitrarily good approximation of the expectation of any function of a random variable X drawn according to $q(x|\gamma^t)$. For calculating this approximation, sufficiently many transitions are made of a specially constructed Markov Chain (MC) $\{\zeta^k\}$. We will say that one *iteration* has been completed when N transitions of the MC are made, and let n_{iters} denote the total number of such iterations. The state space of $\{\zeta^k\}$ is $\{+1, -1\}^N$. In the simplest construction, from the state ζ^k at the k-th iteration, outgoing transitions are allowed only by flipping at most one component of ζ^k. The probability of making a transition by flipping the i^{th} component is given in terms of the 'full conditionals' derived from $q(x|\gamma^t)$,

$$\begin{aligned} &\Pr\{[a_1, \ldots, a_i, \ldots, a_N]^T \rightarrow [a_1, \ldots, \tilde{a}_i, \ldots, a_N]^T\} \\ &= c \ \exp\left(\tilde{a}_i[\theta_{s\,i}^t + \textstyle\sum_{j \neq i} W_{s\,ij}^t a_j]\right), \end{aligned} \tag{5.10}$$

where c is a normalizing constant. The MCMC has the invariant distribution $q(x|\gamma^t)$, and the empirical average of $b(\zeta^k)$ converges a.s. to its expectation η_{γ^t} [127]. We can thereby calculate the conditional and unconditional means, $\eta_{\gamma^t+h(Y^t)}$ and η_{γ^t}, and the score $S_\gamma(Y^t)$, according to equations (2.19) and (2.30).

Consider first the calculation of η_{γ^t}. A sensor node with spatial index s merely needs to maintain the MCMC state component ζ_s, which corresponds to the field variable X_s^t. In one iteration, each site s receives the values of all the other local state variables from its statistical neighborhood. Each site s then computes a new realization for ζ_s, according to the distribution in equation (5.10), and broadcasts it back to all the sites in its statistical neighborhood. The order of the site updates is not critical, and may be chosen pseudo-randomly. After sufficiently many (n_{iter}) such iterations, the temporal averages are read-off as approximations of the true expectations of the respective basis functions. Similarly, $\eta_{\gamma^t+h(Y^t)}$ can be calculated by running a GS that re-samples sites based on the distribution $q(x|\gamma^t + h(Y^t))$.

This shows that we can make a distributed scalable calculation of the per-sample score $S_{\gamma^t}(Y^t) = \eta_{\gamma^t+h(Y^t)} - \eta_{\gamma^t}$. For the time being suppose $A = I$. Since we can maintain the components of the estimated parameter vector γ^t at respective local sites, we then already have a fully distributed (local, component-wise) implementation of the recursion in equation (5.2). Now suppose we use a non-trivial A, hence a "natural gradient" algorithm. We can still implement the recursion in equation (5.2) in an efficient distributed fashion provided we ensure that the non-zero entries in A have a structure such that communication is needed only between sensors in close physical proximity, say one or two hops apart. Thus note that there are two conflicting demands in choosing A: On one hand, to achieve covariance efficiency (in the Cramér-Rao sense), it needs to be chosen close $(F_{\gamma^*}^Y)^{-1}$. At the same time, to achieve power efficiency, it needs to be sparse in a particular way. Fortunately, $(F_{\gamma^*}^Y)^{-1}$ is *already almost sparse* in the required sense

provided W is sparse with localized interactions (cf. model assumptions in Section 2.2) and the SCR is not too small. A very instructive special case arises for a Gauss-Markov random field [134], where this statement is exactly true because $(F_{\gamma^*}^X)^{-1} = -W$, where W is now called the *sensitivity matrix*.

5.3.3 Power Efficiency

The total power consumption of the distributed implementation of GS is determined by three factors, namely, the frequency of the estimator updates, the number of GS iterations per update, and the communication load per iteration per update. We will now analyze the total power consumption and make comparisons with a centralized estimator.

Frequency of Estimator Updates

The statistical model of a natural field is typically quasi-static, i.e. it changes with a time constant of χ sampling intervals, where $\chi \gg 1$. For adequate tracking we only need to ensure that the algorithm time constant (cf. Lemma D.2) satisfies $\tau(\epsilon) \leq \chi$, hence $\epsilon \geq \frac{2}{\chi+1}$. However, the same real-time tracking performance (in seconds) can also be achieved if we choose any $n_{update} \geq 1$, set $\epsilon \geq \frac{2n_{update}}{\chi+1}$, and update the parameter only once every n_{update} samples. By using an $n_{update} > 1$ we can improve power efficiency by a factor of n_{update}, while sacrificing covariance efficiency by the same factor, a technique we call *interlacing*.

Number of GS Iterations Per Update

It is known that, depending on how strongly the MC 'mixes', the number of GS iterations, n_{iters}, may need to be in the thousands to produce very accurate expectations. However

in practice we can make an early termination of the GS iterations in each update, paying with an increased covariance of the calculated per-sample score, but still maintaining *unbiasedness*.

Communication Load Per Iteration Per Update

The statistical neighborhood of each mote mirrors a small physical neighborhood (cf. Section 2.2). Owing to the nature of wireless communication, the "communication graph" is also determined by physical proximity. Hence the statistical neighborhood can be covered by a single broadcast transmission or perhaps a small number of hops, n_{hops} [30, page 64]. Thus the communication load per mote per iteration per update remains invariant as N increases.

Analysis of Power Budget With Multi-Hop Message-Passing

Let the radius of interaction in the field be r meters. Let the sampling interval be T_{samp} seconds. Assume that the transmission of a message from a mote to another mote is done over an additive white Gaussian noise (AWGN) channel and allowed a time slot of T seconds and a bandwidth of W Hz. Thus one mote transmission consists of $n_S = WT$ quadrature symbols. In the scenario of multi-hop communication between motes with $n_{hops} \geq 1$ hops, we will pessimistically assume that an end-to-end transmission must be completed in T seconds, implying that each hop of r/n_{hops} meters is allowed only T/n_{hops} seconds. Let ϱ denote the path-loss exponent for all the wireless transmissions [93]. Denote the noise power spectral density (PSD) of the front-end receiver in a mote by N_0 Watts/Hz, and define $\sigma^2_{noise} \doteq TWN_0$. Lastly, let P_R be the power consumed by the *receiver* in each mote under the pessimistic assumption that it is always 'ON'. P_R is obviously independent of N.

In order to implement recursion (5.2) with a distributed Gibbs Sampler, each mote

Table 5.1: (A) Upper bound on increase in power consumption due to multi-hop message passing. (B) Lower bound on system gain of distributed Vs centralized model estimation.

k	Upper Bound on $\frac{\mathcal{P}_{distrib}[n_{hops}=k]}{\mathcal{P}_{distrib}[n_{hops}=1]}$		
	$n_S = 4$	$n_S = 8$	$n_S = 16$
2	1.09	1.04	1.02
3	1.20	1.09	1.04
4	1.31	1.14	1.07

(A)

n_{iters}	Lower bound on $\frac{\mathcal{P}_{central}}{\mathcal{P}_{distrib}}$, dB		
	$\frac{d}{r} = 100$	$\frac{d}{r} = 1000$	$\frac{d}{r} = 10000$
50	18.7	38.7	58.7
100	15.7	35.7	55.7
200	12.7	32.7	52.7

(B)

needs to communicate $2n_{iters}$ one-bit messages over a maximal distance of r meters. (One half each for the Conditioned-GS and the Unconditioned-GS). Note that the communication of the GS messages is *delay constrained*, hence we must assume that the messages are not compressed. Similarly, long channel codes cannot be used, due to which an energy penalty λ is suffered w.r.t. Shannon capacity. Using the same technique as in [115], which was based on Shannon's capacity theorem [5] and the Friis path-loss model [93], we can derive the following upper bound on power consumption of *each mote* while implementing the incremental estimator of equation (5.2) in a distributed fashion:

$$\mathcal{P}_{distrib} \leq P_R + 2\frac{\lambda\, n_{iters}}{n_{update} T_{samp}} \sigma^2_{noise} r^{\varrho}\, \frac{2^{\frac{n_{hops}}{n_S}} - 1}{n_{hops}^{\varrho-1}} \quad \text{[Watts]}. \tag{5.11}$$

Note that n_{hops} is a constant independent of N, as has been already discussed in 3) above. Similarly, we will see simulation evidence in Section 5.4.3 that even n_{iters} can be chosen to be a constant independent of N. Hence it follows that bound (5.11) is *independent* of the network size N, thus demonstrating that our algorithm is strictly

scalable. Secondly, the power consumption varies as $\frac{n_{iters}}{n_{update}}$, and this allows us to choose any desirable tradeoff w.r.t. Cramer-Rao efficiency. Thirdly, the bound also demonstrates that if the statistical neighborhood needs to be covered by $n_{hops} > 1$ hops, then in the worst case the power consumption is increased by a factor $\frac{2^{\frac{n_{hops}}{n_S}}-1}{(2^{\frac{1}{n_S}}-1)n_{hops}^{\varrho-1}}$, which is quite small when $n_{hops} = 1 \sim 4$. For example, Table 5.1(A) demonstrates the increase in power consumption due to multi-hop communication for the case of $\varrho = 2$ (free space propagation) and several values of n_S. We conclude that the communication of GS messages with a small number of hops is not a fundamental drawback.

Distributed vs Centralized Estimation

We will now analyze the system gain that the distributed identification algorithm enjoys over a centralized algorithm. Note that the details of the centralized algorithm are immaterial to the power efficiency comparison because the energy cost incurred in centralized processing is solely attributed to the aggregation of raw data at the FC; there is no 'algorithmic' energy cost.

Analogous to the assumptions made in 4) above, let $\hat{N}_0$ be the noise PSD of the front-end receiver in the FC, and let $\hat{\sigma}^2_{noise} \doteq TW\hat{N}_0$. Let the distance of the WSN to the FC be d meters. We will assume that the communication with the FC is done by direct long distance transmissions, without repeaters. This is a valid assumption when the FC is far removed from the WSN array ($d \gg r$). In a centralized scheme, after every n_{update} sampling intervals, we need to extract all the sensor data to the FC. This implies transmitting N real numbers, quantized to some sufficient precision $\frac{1}{2^{n_{acc}}}$, over a distance of d meters. Let us assume (optimistically, from the point of view of the centralized scheme) that the data is optimally compressed with Slepian-Wolf type non-cooperative encoders, and a capacity achieving channel code is used. We will also assume that a suitable time sharing schedule is used so that the burden of long distance

transmissions is distributed evenly among the motes. Let $\mathcal{P}_{central}$ denote the power spent by each mote on average in the centralized estimator implementation. Then we can loosely lower bound the "system gain" as

$$\frac{\mathcal{P}_{central}}{\mathcal{P}_{distrib}} \geq \frac{d^{\varrho}\,\hat{\sigma}^2_{noise}\,(2^{\frac{n_{acc}+\log_2\sigma\sqrt{2\pi e}}{n_S}}-1)}{n_{update}T_{samp}P_R + \frac{2n_{iters}\lambda}{n_{hops}^{\varrho-1}}\,r^{\varrho}\,\sigma^2_{noise}\,(2^{\frac{n_{hops}}{n_S}}-1)} \tag{5.12}$$

Since r and P_R are constants, as $d \to \infty$ the gain increases monotonically and unboundedly implying that there is always a break-even distance where it exceeds unity. Table 5.1(B) gives the power gain for exemplary values of n_{iters} and $\frac{d}{r}$, and a set of practical values for the other parameters as follows: $\varrho = 2.0, \sigma^2_{noise} = \hat{\sigma}^2_{noise}$, $\lambda = 10$, $n_{acc} = 4$, $n_S = 4$, $n_{hops} = 1$, $P_R = 0$ (negligible receiver consumption) and $\sigma^2 = -6$ dB. Clearly, on the basis of power efficiency, there is a strong incentive for in-network model estimation.

5.3.4 Identification of Partially or Fully Homogeneous Models

In Sections 5.3.2 and 5.3.3, we have assumed that the *every element* of θ_s and W_s was unique and needed to be estimated, which implies a fully non-homogeneous model. However, there can be special cases such that there are *replicas* among the elements of θ_s and W_s. An extreme case would arise when we have a fully homogeneous model, such as the one we used in simulations in Section 5.4, where all the elements of θ_s are copies of a single parameter, and the matrix W_s is symmetric Toeplitz. In physical terms, this implies that the behavior of the field in one neighborhood of the network is statistically (but not numerically) identical to the behavior of the field in some other far-away neighborhood.

Obviously, if have have such partial or full homogeneity, we can expect to attain

a further reduction in the CRLB on the estimator variance. Conversely, for achieving a fixed estimator variance we have the potential of using a smaller data window. A large part of this extra available efficiency can be extracted by the simple expedient of averaging over the individual estimates of the replicas. In a practical distributed implementation this would mean (a) imposing an additional clustering architecture, where all motes sharing a common parameter form a *cluster* (b) specifying cluster heads, and (c) after every update of the incremental identification algorithm, collecting the estimates of replicas at each cluster head, performing the averaging operation, and communicating this average back to the respective motes for use in the next estimator update. Notice that since the cluster head collects only the *estimates* of the replicas, the causes of power consumption discussed in 5.3.3- 5.3.3 are not involved, and hence the communication load of this additional clustering step is negligible.

5.4 Simulation Results and Discussion

In this section we present comprehensive simulations for an exemplary application where a uniform linear array of N sensors measures the presence or absence of an effluent released from a chemical plant into a river. We use the instance of the Boltzmann field model presented in Section 2.3.1 for the statistical distribution of the effluent. Recall that the spatial dependencies are given by an $N \times N$ Toeplitz matrix parameter. Let its first row be given by $\rho[0, \xi_1, \xi_2, \xi_3, 0, \ldots, 0]$. Note the additional factor ρ, which is used to control the strength of the spatial interactions, with a large ρ leading to large plumes on average. *Moderate dependencies* refers to the choice $\rho = 1.0$, while *strong dependencies* refers to $\rho = 2.0$. Note that we postulate a spatially homogeneous interaction (i.e. a Toeplitz structure) only for simplicity of exposition. The algorithm itself does not presume such a property; indeed, each entry of W_s can in principle be distinct. The

total truth parameter is $\gamma^* \equiv (\theta_s^*, W_s^*)$, of length $M = 4N - 6$. *Static model* will refer to the choice $[\xi_1, \xi_2, \xi_3] = [0.5, 0.3, -0.3]$, while *Time-varying model of period* χ will refer to the choice $[\xi_1, \xi_2, \xi_3] = [0.5, 0.3, -0.3] \times \left(1 + \frac{1}{2}\sin\frac{2\pi t}{\chi}\right)$. In the latter case, $\gamma^{nom} \equiv [0.5, 0.3, -0.3]$.

5.4.1 Covariance Efficiency

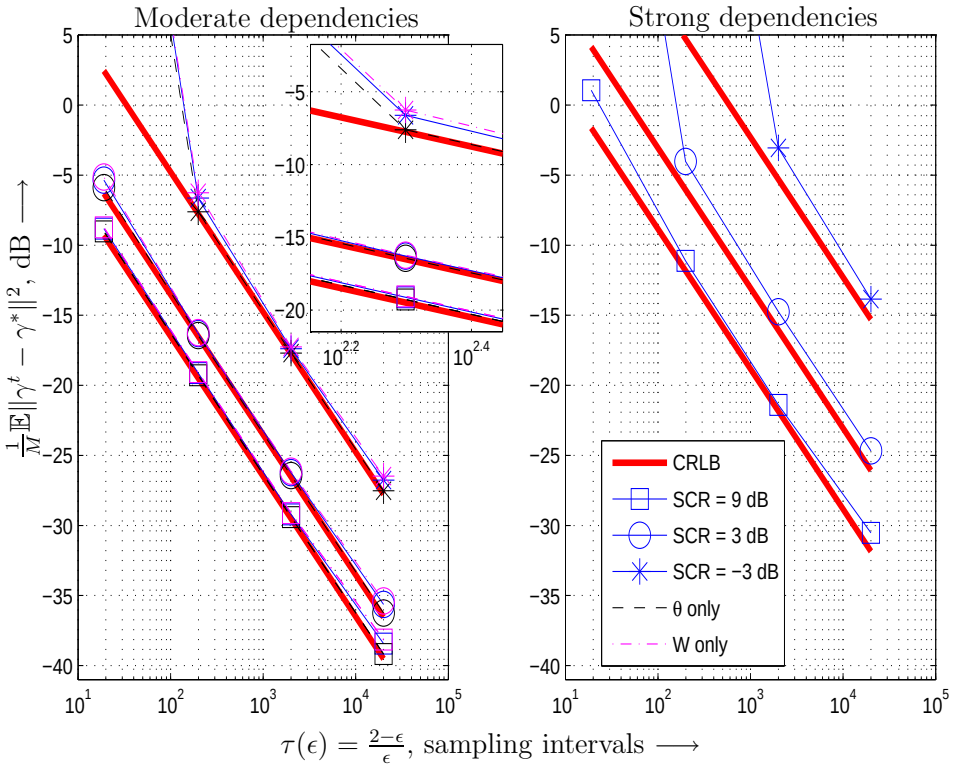

Figure 5.1: Dependence of the variance efficiency on SCR. Nominal parameters: $N = 8, n_{update} = 1, n_{iters} = 128, A = (F_{\gamma^*}^Y)^{-1}$.

First we will study the steady state variance-covariance efficiency vs $\tau(\epsilon)$ characteristic, as a function of the SCR, the update interval n_{update}, the number of GS iterations n_{iters}, and the type of constraints on the pre-scaling matrix A. As nominal values we choose SCR = 3 dB, $n_{iters} = 128$, $n_{update} = 1$ and $A = (F_{\gamma^*}^Y)^{-1}$ with no constraints. We vary each parameter individually while keeping the others fixed at their nominal values,

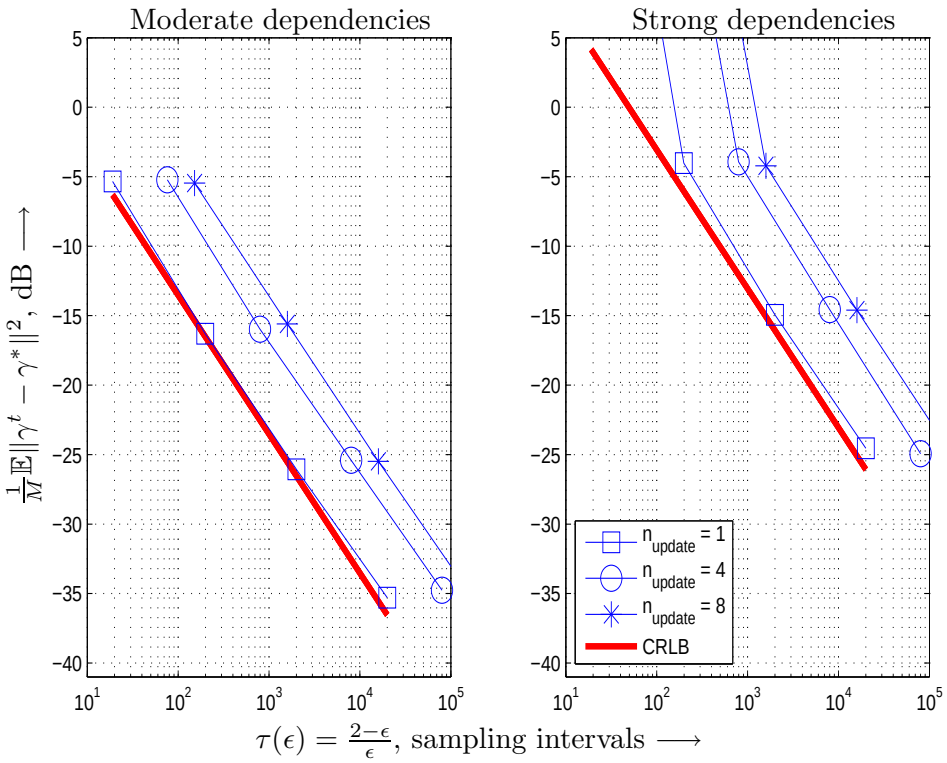

Figure 5.2: Dependence of the variance efficiency on update interval, n_{update}. Nominal parameters: $N = 8$, SCR$= 3.0, n_{iters} = 128, A = (F^Y_{\gamma^*})^{-1}$.

and the results are presented in Figure 5.1 through Figure 5.4. We also plot the CRLB as a baseline reference in each case. The first column in each sub-figure corresponds to moderate dependencies and the second column to strong dependencies.

We observe that (a) In Figure 5.1, for all practical SCRs and tracking speeds there is very good agreement between linearized analysis (i.e. the CRLB) and simulations. The simulations degrade from the CRLB for large ϵ because the estimator starts making large excursions, and sometimes loses 'lock'. It is interesting to note that the reduction in efficiency due to a lowering of the SCR is *not* dB-to-dB. Also, as an example, for the moderate dependencies case we have demonstrated that the MSE of the sub-components, i.e. $\frac{1}{N}\mathbb{E}\|\theta^t_s - \theta^*_s\|^2$ and $\frac{1}{M-N}\mathbb{E}\|W^t_s - W^*_s\|^2_F$ labeled as "θ only" and "W only" respectively, is qualitatively similar to the total MSE. This is a consequence of the optimal pre-scaling

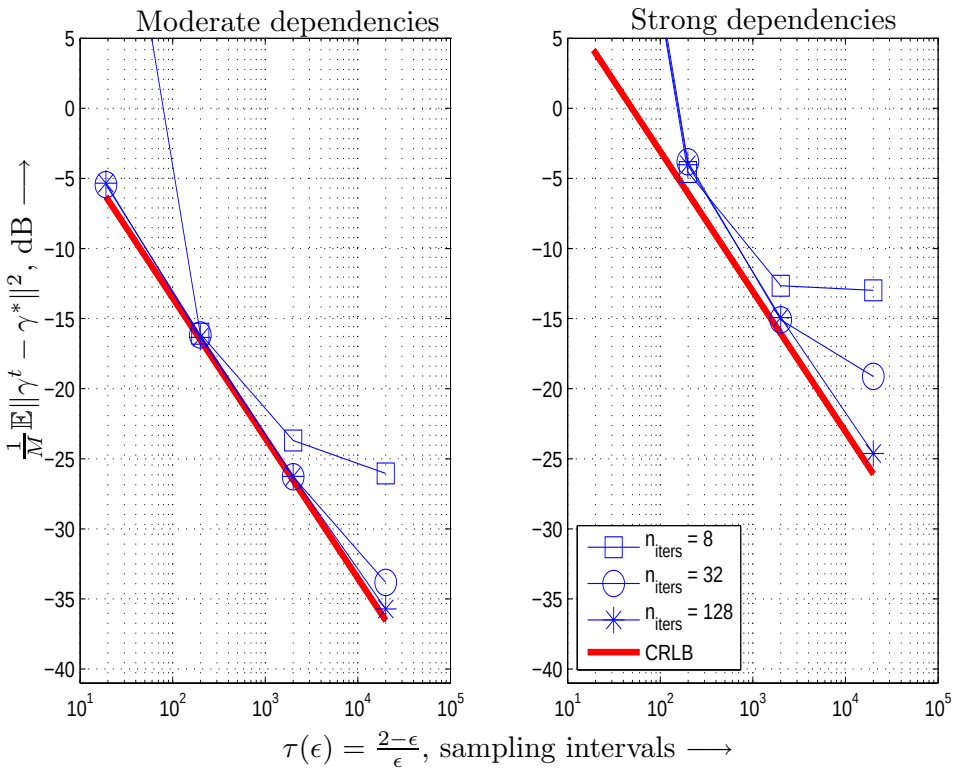

Figure 5.3: Dependence of the variance efficiency on the number of GS iterations, n_{iters}. Nominal parameters: $N = 8$, SCR$= 3.0, n_{update} = 1, A = (F^Y_{\gamma^*})^{-1}$.

operation. (b) As expected, in Figure 5.2, interlacing with a factor n_{update} causes the entire performance characteristic to shift right by an equal factor. (c) As predicted in Section 5.3.2, in Figure 5.3 the covariance efficiency is seen to approach the CRLB when n_{iters} is sufficiently large. The loss from the CRLB due to an insufficient n_{iters} depends on the tracking time-constant and the strength of the field dependencies. Interestingly, we find that even with $\tau(\epsilon) \approx 20000$, and a strong field dependencies, a practical choice like $n_{iters} = 128$ gives a small loss of 1.0 dB w.r.t. the CRLB. (d) Finally, in Figure 5.4 we postulate a user defined *locale* l (in units of distance), and obtain the pre-scaling matrix A by masking $(F^Y_{\gamma^*})^{-1}$ with a localization matrix that allows communication only between sensors that are within l units of each other. Thus, $l = 3.0$ implies that a mote is allowed to communicate with another mote no more than 3.0 units apart,

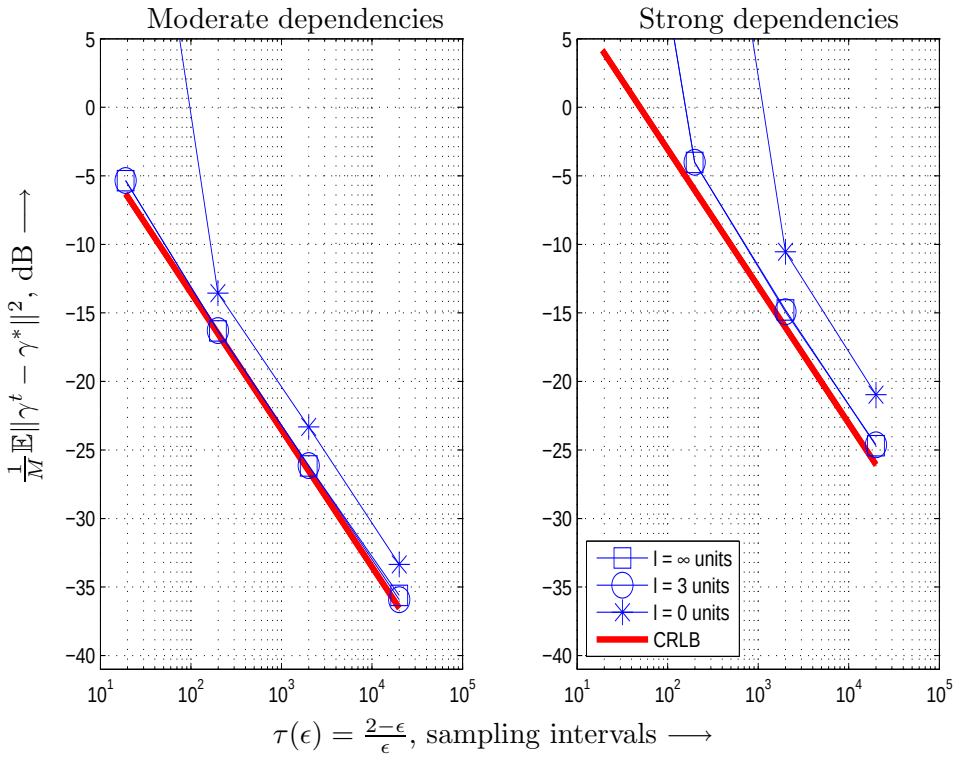

Figure 5.4: Dependence of the variance efficiency on constraints placed on the pre-scaling matrix A (l denotes the *locale* used to mask $(F^Y_{\gamma^*})^{-1}$). Nominal parameters: $N = 8$, SCR= 3.0, $n_{update} = 1, n_{iters} = 128$.

while $l = \infty \Rightarrow A = (F^Y_{\gamma^*})^{-1}$ exactly. We observe that the use of a locale equal to the radius of interaction causes very minor loss in performance. Using a locale of $l = 0$, hence a diagonal pre-scaling matrix, causes a more significant loss in efficiency, of the order of 4.0 dB. Thus we see that using an A that conforms with W_s gives most of the available efficiency, without putting any extra communication load on the motes (cf. Section 5.3.2).

5.4.2 Model Acquisition and Tracking

Now we investigate dynamic performance of the estimator, under a time varying model with a time constant $\chi = 16000$ and moderate dependencies. We let $A = (F^Y_{\gamma nom})^{-1}$, $\epsilon = 10^{-3}$, SCR = 9.0 dB. In Figure 5.5, in the first sub-plot we show the time varying

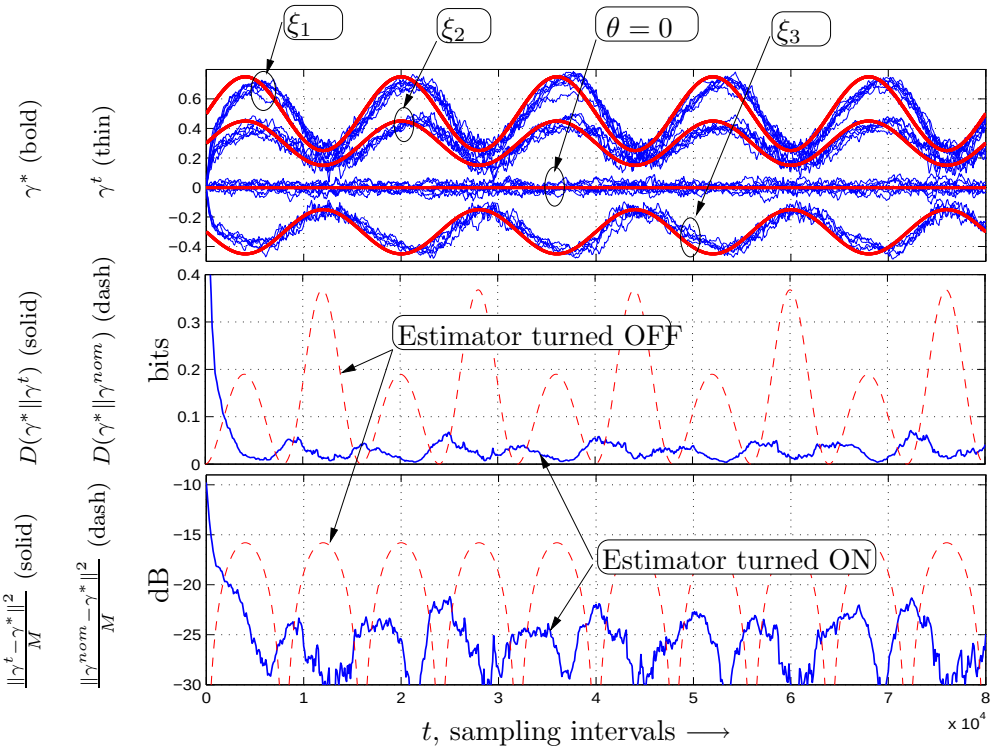

Figure 5.5: Acquisition and tracking performance under a time-varying model with period $\chi = 16000$ samples. $N = 8$, SCR $= 9.0$ dB, $n_{update} = 1, n_{iters} = 128, A = (F^Y_{\gamma^{nom}})^{-1}$.

truth parameter (bold lines), and the estimated parameter (thin lines). In the second sub-plot, we show the KL-divergence between the truth model and the estimated model, $D(\gamma^*\|\gamma^t)$, and also between the truth model and the nominal model $D(\gamma^*\|\gamma^{nom})$. The simulations are started form the maximally uninformative initial condition $\gamma^1 = 0$, so that we can also observe the acquisition dynamics (the *step response*). We observe that (i) The acquisition time agrees very well with the estimator time constant $\tau(\epsilon) = \frac{2-\epsilon}{\epsilon} = 2000$, and the estimator can consistently track the model because $\chi > \tau(\epsilon)$. (iii) An uncompensated (open-loop) system has large excursions in KL-divergence (up-to 0.35 bits), while the adaptive estimator gives comparatively small excursions (up-to 0.05 bits). Similarly, the MSE of the uncompensated case can rise up-to -15 dB per component,

while the MSE of the estimator never rises above −23 dB. (iv) There are no pathologies like ringing or under-damping.

5.4.3 Scalability

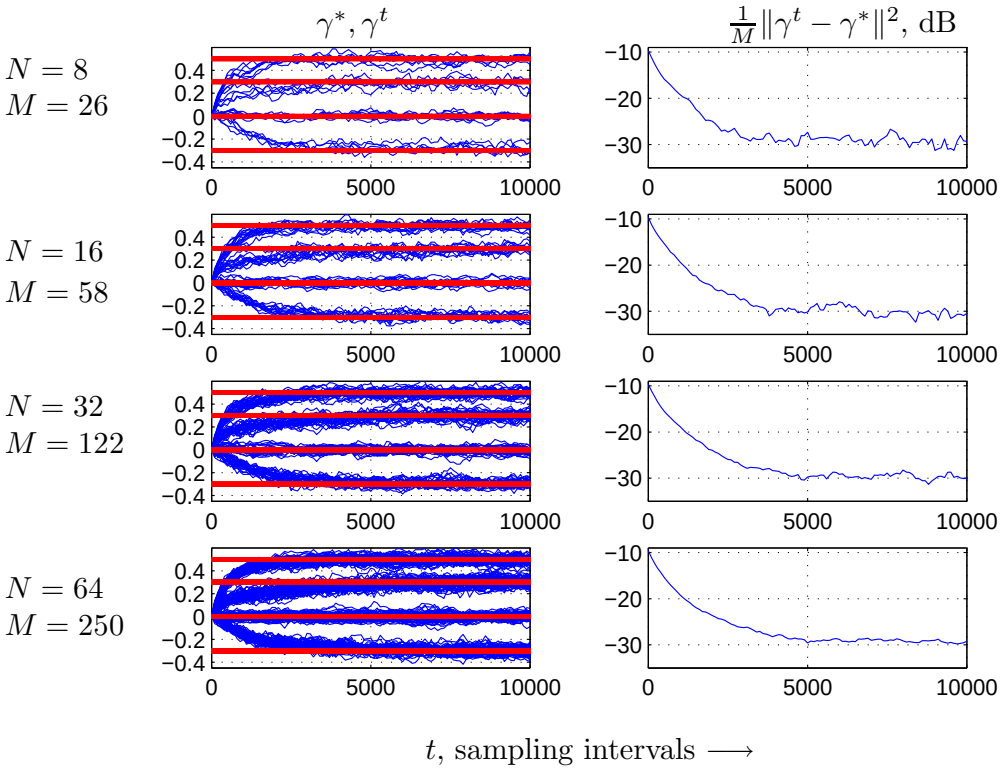

Figure 5.6: Scalability w.r.t. the size of the network N. $n_{iters} = 8$, SCR = 9.0 dB, $\epsilon = 10^{-3}$, $n_{update} = 1$ and a diagonal pre-scaling matrix A (cf. Section 5.4.3). Bold lines indicate γ^*, and thin lines indicate γ^t. $n_{iters} = 128$.

Finally we will consider the important issue of scalability. We have already seen that even if the network size N is increased, the computation and communication load on each mote remains invariant. Here we investigate whether the scalability also holds in terms of the performance. We choose a static truth model with moderate dependencies and essentially repeat the experiment in Section 5.4.2 with several values of N. Each experiment is started from the initial condition $\gamma^1 = 0$. Since it is difficult to exactly

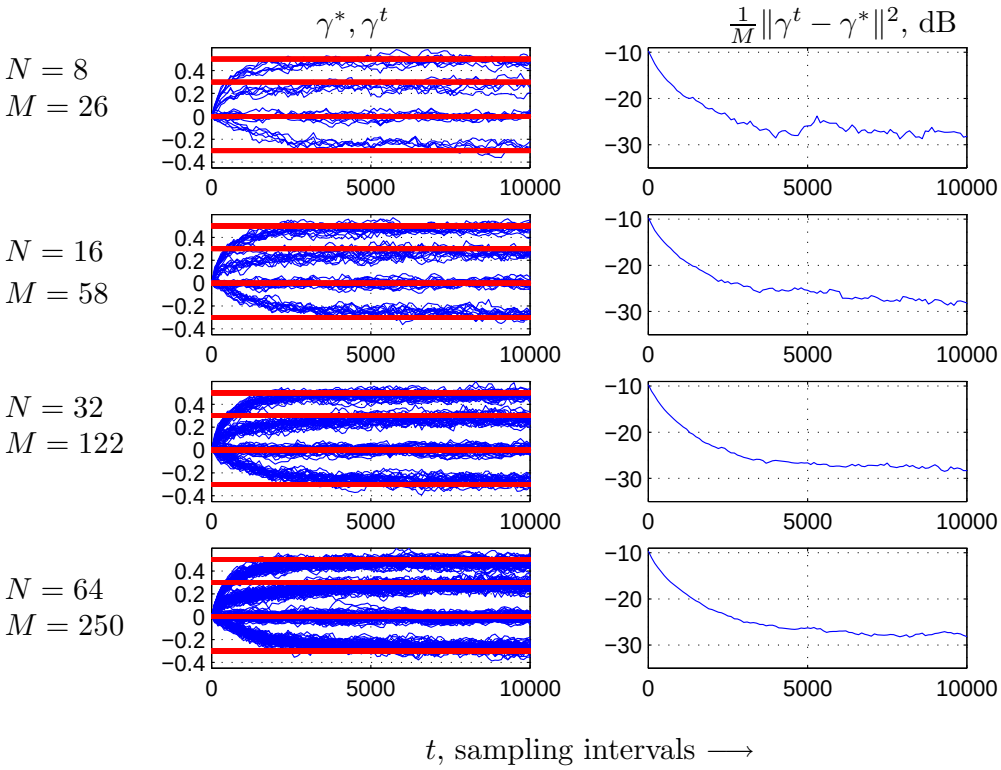

Figure 5.7: Scalability w.r.t. the size of the network N. $n_{iters} = 8$, SCR = 9.0 dB, $\epsilon = 10^{-3}$, $n_{update} = 1$ and a diagonal pre-scaling matrix A (cf. Section 5.4.3). Bold lines indicate γ^*, and thin lines indicate γ^t. $n_{iters} = 8$.

calculate $F^Y_{\gamma^*}$ for large networks (say $N \geq 32$) we will use a suboptimal *diagonal* pre-scaling matrix A. The values on the main diagonal of $(F^Y_{\gamma^*})^{-1}$ for $N = 8$ are interpreted as weights for the corresponding basis functions, and in constructing A for a larger N these weights are appropriately replicated for all the corresponding extra basis functions in the model. The spatial homogeneity of our model ensures that such a construction of A gives a fair comparison between networks of various sizes. We set $\epsilon = 10^{-3}$. In Figures 5.6 and 5.7, we show the normalized MSE of the estimator as a function of time, for various N. Figures 5.6 is for $n_{iters} = 128$ and Figures 5.7 is for $n_{iters} = 8$. We make the crucial observation that the acquisition dynamics and the steady state value of the MSE is essentially *invariant* w.r.t. N, even for a very small number of MCMC iterations

$n_{iters} = 8$. The increase in the MSE from $n_{iters} = 128$ to $n_{iters} = 8$ is small (about 2.0 dB) and is the *same* for all network sizes. Therefore, the estimator acquires and tracks in an identical manner, irrespective of the network size, provided n_{iters} is a little larger than the statistical neighborhood in the model.

5.5 Extension to Fields with Temporal Memory

In this section we will consider an extension of the incremental estimator proposed in Section 5.2 to fields with *temporal* memory. For simplicity we again consider only Boltzmann (and GMRF) fields and assume that the field MC is reversible, so that Lemma 1 holds. Let us revert back to the original definition of the observation density π from equation (2.16) which reads as

$$\pi(u^t|\gamma) \doteq \sum_{z^t} \mathcal{N}(u^t|z^t, \sigma^2 I_{2N}) Q(z^t|\gamma). \tag{5.13}$$

Note that $\pi(u^t|\gamma)$ so defined is the distribution induced on the observations U^t when $\{Z^t\}$ is distributed according to $Q(z^t|\gamma)$, and, on account of Lemma 1, the latter condition holds when the MC $\{Z^t\}$ is reversible and has passed into its stationary phase.

We also revert back to the symbols $Z, U, F_\gamma^Z, F_\gamma^U, \theta, W$ etc. In particular recall the definition of the log-likelihood ratio statistic $h(U)$ from equation (2.28).

5.5.1 Incremental Estimation of the Markov Chain Parameters

Consider the recursion

$$\gamma^{t+1} = \gamma^t + \epsilon\, A\, S_{\gamma^t}(U^t) = \gamma^t + \epsilon\, A \left(\eta_{\gamma^t + h(U^t)} - \eta_{\gamma^t}\right). \tag{5.14}$$

First note that this recursion can be implemented in a fully distributed scalable manner along the lines of Section 5.3.2, provided the properties of sparsity, localization and mild interactions are satisfied as before. Secondly, we claim that this recursion converges stochastically to the true parameter γ^*, and hence is able to acquire and track sufficiently slow changes in the transition probabilities of the MC. To substantiate this claim, analogous to the i.i.d. case, we can again construct an expected gradient system and assert a theorem that it respects the true parameter as an A.S. fixed point. The proof is virtually identical to that of Theorem 3, with minor notational changes. Furthermore, the implications of such averaged gradient stability for stochastic stability also continue to hold [177, 178, 180] provided the time constant of the estimator is sufficiently larger than the time constant of the hidden MC. This ensures that a sufficiently representative set of realizations are averaged over within the time constant of the estimator, and hence the coupling of the stochastic estimator trajectory with the averaged gradient trajectory occurs with high probability. The covariance efficiency analysis is a bit more involved. We can analyze the steady state efficiency via linearization as before, where the steady-state error evolution equation now reads

$$\alpha^{t+1} = (I - \epsilon A F^U_{\gamma^*})\alpha^t + \epsilon A S_{\gamma^*}(U^t). \tag{5.15}$$

However notice that due to the temporal memory of the chain, as well as the 'pairwise' definition of U^t, the stochastic increment $\epsilon A S_{\gamma^*}(U^t)$ is *no longer independent* of current estimator error α^t. Hence an extra cross-covariance term $2\epsilon(I - \epsilon A F^U_{\gamma^*})\mathbb{E}\left[\alpha^t\, S_{\gamma^*}(U^t)^T A\right]$ shows up in the RHS of equation (5.7). Unfortunately there is no simple closed form expression for this term, and neither is it sign definite. If we ignore this term and continue to use equation (5.7) as it stands, then the resulting solution Σ_α can only be treated as an approximation of the true error covariance. The quality of the approximation improves

as the chain becomes strongly mixing. Similarly, in principle we can also also calculate the CRLB using a dynamic programming approach, though the $O(n2^{2N})$ complexity can be prohibitive in the regime of interest to us, namely a large time constant n.

In general the efficiency of the estimator in (5.14) will be bounded away from the CRLB because the estimator does not utilize the temporal memory of the hidden MC. However the efficiency can be significantly improved if, instead of the raw log likelihood ratio statistic $h(U^t)$, we use a statistic h^t_{filter} provided by a causal filter, like the distributed algorithm proposed in Chapter 3 [115], which is based not only on the current observation U^t but also all the past history of observations $U^{t-1}, U^{t-2}, \ldots$. In fact, if we use an *a-causal* filter (*smoother*) which produces the log likelihood ratio statistic with a delay that is small relative to time constant of the estimator but large relative to the time constant of the MC, the recursion (5.14) can asymptotically achieve an efficiency close to the CRLB.

5.6 Conclusions

We have proposed a distributed incremental estimator for exponential models in Wireless Sensor Networks. We have shown that the algorithm is stable, asymptotically efficient, strictly scalable, and has a power consumption significantly smaller than a centralized estimator even in the scenario of multi-hop message passing. Although we temporally independent fields, we also indicated how it is possible to extend our model identification scheme in a straightforward manner to the case of fields with spatial as well as *temporal* dependencies [169].

6 Target Tracking With RFIDs

6.1 Introduction

The ability to remotely locate and track mobile targets is crucial in a wide variety of applications like security and access control, safety/emergency services, habitat monitoring and robotics, to name a few [110][112]. If a localization accuracy of the order of tens of meters is acceptable, and the application operates in an *outdoor environment* (for example, tracking vehicles on roads), commercial grade Global Positioning Systems (GPS) [181] like NAVSTAR, GLONASS, GNSS and even Cellular-Assisted-GPS (AGPS) [182] could be used. GPS based solutions are however relatively costly, especially for small-scale applications. Moreover, since GPS signals do not penetrate well into buildings, mines etc, a good precision is not possible in indoor environments unless very expensive (sensitive) equipment is used [183]. In fact, if accuracies of the order of a meter or less are required, the current generation of commercial grade GPS cannot give satisfactory performance.

In contrast, a more practical and inexpensive alternative solution is afforded by WSNs [1]. In the tracking problem presently considered, the *sensor* in each mote is in fact a radio receiver that detects radio signals from cheap unobtrusive radio frequency identification (RFID) tags mounted on the targets to be tracked. (Other types of sensing modalities like acoustic or magnetic sensors have a very limited scope of applicability.)

The motes communicate with each other and with a Fusion Center (FC) using multi-hop wireless transmissions. Hence, in addition to the sensing radio receiver, each mote is also be equipped with a communication radio-transceiver. WSNs are desirable from a practical point of view because they can be set up very quickly relative to wired networks, and allow the operator to avoid the cost of making massive renovation for laying new cables. On the other hand, since the motes are power constrained, their computation and communication resources need to be used frugally.

The sensor network can achieve localization and tracking of the targets based on one of several metrics of the radio signals from the tags as sensed by the motes, namely, the time of arrival (TOA), time difference of arrival (TDOA), angle of arrival (AOA), Doppler spread (DS), multi-path spread (MPS), and received signal strength indication (RSSI) (see [110],[112] and the references therein). While TOA and TDOA based techniques can in principle give fine grained localization, they need very stable clocks on each tag and each mote (with a dither of the order of a nanosecond), and the wireless transmissions from the tags need to be in the form of very narrow pulses like, for example, UWB-IR signals [113]. These requirements typically result in high hardware costs. Similarly, tracking based on AOA requires accurate multi-antenna beam-forming by the motes, which is usually impractical due to the constraints of small size and cost that the devices must satisfy. Tracking based on DS requires that there be relative motion of significant velocity between the tags and motes [184]. Finally, using the MPS signature we can, in principle, track a target even with a single mote [185]. However this technique too requires multi-antenna reception, and the tracking accuracy can be poor and extremely dependent on the radio environment.

In contrast to all the other alternatives, the option of RSSI based tracking is extremely cheap in terms of logistics since it uses a signal metric (the received signal strength) that can be sensed even with the most inexpensive sensor networks. In partic-

ular, it does not require multi-antenna reception or expensive signal processing by the motes. However, the performance of RSSI based tracking suffers, as do other techniques to a lesser extent, one serious drawback: If the operating environment has significant occlusions, shadowing or multi-path propagation, the tracking algorithm gives large biases in the position estimates *which cannot be eliminated even with an abundance of observation data*. Unfortunately, this is exactly the scenario that indoor office/factory type environments present. The biasing effect is also seen to occur if the transmit power from the tags loses calibration or if the antennae gains have a significant production variability around the nominal value (as can be expected in low cost equipment). This is the main reason why WSN tracking systems that rely exclusively on RSSI measurements have found a limited applicability to date, even though they are very cheap to implement. To quote [110],

"...its [RSSI's] usage in real world applications is still questionable. But considering its low cost, it is possible that a more sophisticated and precise use of RSSI could become the most used technology of distance estimation from the cost/precision point of view."

In this chapter we present such an advanced RSSI based tracking technique that gives a dramatically improved tracking performance in indoor environments relative to extant RSSI based methods, while still using inexpensive hardware as is found in typical sensor networks. This improvement is achieved due to two novel features of our proposed technique:

(1) The algorithm is significantly robust to the problems of radio occlusions, shadowing, multi-path reception, and de-calibration of transmit power and antenna gain, because it incorporates an incremental estimation algorithm that *learns* the radio environment of the tags and the motes using the state of the targets inferred by the tracking algorithm. The target state inference is in turn based on the most recent estimate of the radio environment, thus leading to a recursive structure. Note that recently [111] have also

attempted to solve the problem of tracking in indoor environments, in the specific limited case of UWB radio transmissions by tags. (Note that our method is applicable, but not limited, to UWB transmissions). Their approach is different than ours in that they treat the availability of line of sight (LOS) transmissions as a 0/1 random process, and track this process in tandem with the motion of the targets in a fully Bayesian setting. In contrast we treat the availability of LOS/non-LOS transmissions parametrically, and use a mixed Bayesian/Maximum-Likelihood approach. We will discuss the rationale behind this in more detail shortly.

(2) In literature the tracking/localization algorithm is typically chosen to be one of several popular options like basic trilateration [110], least squares estimation and its variants [186], extended and unscented (sigma-point) Kalman filters [41][187], and particle filters [188][128]. We prefer the choice of particle filters because our simulations indicate that they give the best approximation of the highly multi-modal a-posteriori propagated density when tracking with RSSI, an observation that has been corroborated by other researchers like [38][43][44]. However, in order to exploit the co-dependencies in the motion of the targets, [38][43][44] have formulated the filter on the joint multi-target probability density (JMPD). This solution requires a centralized implementation that is unsuited to the constraints of the WSN, since it can lead to network congestion and a rapid depletion of the batteries in the motes. In contrast, we take a novel approach where we use a bank of *distributed* particle filters on the *marginal* densities of the targets, and allow them to interact only through conditional marginal expectations. This enables our algorithm to exploit the dependencies in the motion of the targets to extract a significant *diversity gain*, while still maintaining a tractable, power efficient and scalable system.

Mixed ML-Bayesian approach vs pure Bayesian approach to target tracking: There are several reasons why we choose a mixed ML-Bayesian approach, i.e. target tracking with

a Bayesian particle filter, and radio environment estimation via ML principles. Firstly, the radio environment is typically *quasi-static* and hence it seems to be an over-kill to include hundreds of such quasi-static radio parameters into the state-vector along-side the target positions and velocities that change relatively very fast. Secondly, even if the radio parameters are viewed as random variables, typically almost nothing is known about their priors or their dynamics and one would be forced to make ad-hoc assumptions for the same, which is highly unsatisfactory. In contrast, with an ML approach such assumptions are not needed. Lastly, when the state vector is appended with many radio parameters, the distributed implementation of the Bayesian filter becomes much more complicated, if not impossible, *since a clean node-based partition of the total-state cannot be made*. Thus the pure Bayesian approach seems unsuited to the distributed filtering application. Finally, it is interesting to note that some researchers [45] have proposed the other extreme of a purely ML based approach to target tracking, wherein nuisance parameters as well as the position and velocities are all considered to be unknown but deterministic parameters and estimated by ML principles. This approach too does not seem appropriate in the present case.

Outline of the chapter: In Section 6.2 we will define the system model, which consists of a maneuver model (Section 6.2.1) and an observation model (Section 6.2.2). Section 6.2.3 defines the aims of the tracking technique and presents an overview. Section 6.3 describes the tracking sub-algorithm based on particle filtering, while Section 6.4 describes the radio-environment estimation sub-algorithm based on a variant of stochastic Expectation Maximization (EM). Section 6.5 discusses the practical aspects of a scalable implementation of the proposed tracking system with a WSN. In Section 6.6 we present simulations results for several realistic tracking scenarios, and study the effects of adaptation to the radio environment, sensor density, measurement noise and co-dependencies in the motion of the targets. We also study the effect of localized track-

ing and its implications for scalability. Section 6.7 concludes the chapter. We use the following conventions in this chapter: As usual, $\{\Upsilon_b^a : a = 1, 2, \ldots, A;\ b = 1, 2, \ldots, B\}$ will denote a *set* of objects formed by letting the indices run over the prescribed ranges. Where no confusion can arise we denote such a set by $\{\Upsilon_b^a\}$, with the ranges of the indices being defined implicitly, or even simply as $\{\Upsilon\}$. However note that *partially* enumerated indices always specify a *subset*. For example, $\{\Upsilon_b^a : b = 1, 2, \ldots, B\}$ will denote a subset of elements, where the superscript index is pinned to a and the subscript runs over the prescribed range.

6.2 System Model and Overview

Suppose we wish track in D dimensions ($D \in \{1, 2, 3\}$) in some bounded tracking area (or length or volume). We divide this tracking area into L *cells* which are chosen such that there is no appreciable spatial variability in the radio transmission characteristic from any points within a cell to a mote, apart form the standard distance dependent path-loss due to an expanding wavefront. (This notion will be clarified further shortly.) In many cases the cells may be chosen to be the rooms in the building. Within the tracking area N motes are placed at more or less uniformly spread out but arbitrary locations, with position vectors $r_n \in \mathbb{R}^D$, $n = 1, 2, \ldots, N$. All these position vectors are assumed to be known to each mote and the FC. Similarly, the knowledge of the boundaries of the cells is assumed to be available to all the motes and the FC.

For example, consider the system setup illustrated in Figure 6.1, which we also use in some of our simulations presented in Section 6.6. The tracking area in this case is one floor of a building with four rooms of dissimilar sizes. Within the tracking area M targets are allowed to move freely, each one carrying an active RFID tag. The position of target number $m \in \{1, 2, \ldots, M\}$ at discrete time index $t \in \{1, 2, \ldots\}$ is denoted by

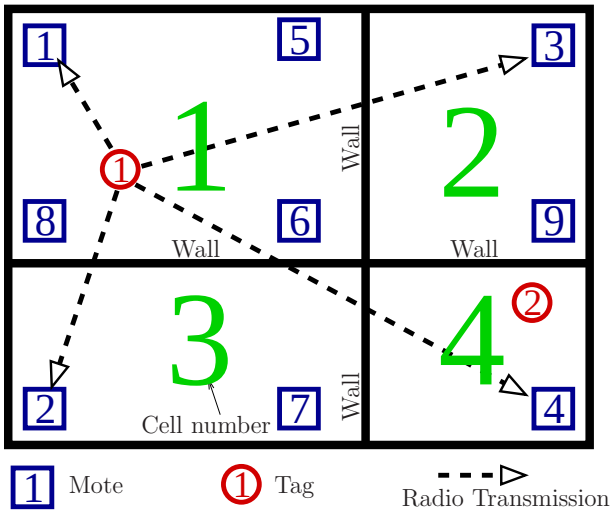

Figure 6.1: An exemplary setup for RSSI based target tracking in an indoor environment in a plane ($D = 2$ dimensions), using a WSN. There are nine installed motes (of which only $N = 4$ are shown to be active in the figure), $M = 2$ targets and $L = 4$ cells. The cells are defined to be the rooms of the building.

$X_m^t \in \mathbb{R}^D$. The set $\{X_m^t\}$ is viewed as a random process postulated to be governed by a known *maneuver model*, which will be described in Section 6.2.1. Our aim is to estimate this process optimally in a certain sense. Let $\hat{X}_m^t$ denote the estimate of X_m^t.

Let $Y_{m,n}^t \in \mathbb{R}$ denote the RSSI in decibel-milliwatt (dBmW) of the signal from tag m received at mote n at time t. Then $\{Y_{m,n}^t\}$ is also viewed as a random process whose statistics, conditioned on $\{X_m^t\}$, are postulated to be governed by a known *observation model*, which will be described in Section 6.2.2. Note that only $\{Y_{m,n}^t\}$ is observable, while all of $\{X_m^t\}$ is hidden.

Let the power of the signal transmitted by tag m at time t be Φ_m^t dBmW, and let the *gain* of the path from cell $l \in \{1, 2, \ldots, L\}$ to mote n at time t be $\Gamma_{l,n}^t$ dB. Note that $\Gamma_{l,n}^t$ *does not* include the propagation path loss due an expanding wavefront; such a path loss will be modeled separately in Section 6.2.2. It may be helpful to think of $\Gamma_{l,n}^t$ as the (negative of the) loss due to attenuation by intervening objects like walls. Even the choice of $L = 1$ is not trivial, since it can still model the de-calibration of antennae gains.

The quantities $\{\Phi_m^t\}$ and $\{\Gamma_{l,n}^t\}$ are viewed as unknown but deterministic *quasi-static* parameters. For the tracking algorithm to work accurately, the true values of these parameters need to be estimated and used. Let $\hat{\Phi}_m^t$ and $\hat{\Gamma}_{l,n}^t$ denote the estimates of Φ_m^t and $\Gamma_{l,n}^t$ respectively.

6.2.1 Maneuver Model

Although the velocity of a target is typically not of interest to the user, it nevertheless is an internal variable of the statistical model we will use, and hence needs to be estimated in tandem with the position. Let $V_m^t \in \mathbb{R}^D$ denote the velocity of target m at time t. Define the *sub-state* corresponding to tag m at time t as

$$Z_m^t \doteq [X_{m,1}^t V_{m,1}^t X_{m,2}^t V_{m,2}^t \dots X_{m,D}^t V_{m,D}^t]^T \tag{6.1}$$

and the *total-state* of all the targets as

$$Z^t \doteq [Z_1^{tT} Z_2^{tT} \dots Z_M^{t\ T}]^T. \tag{6.2}$$

In this chapter we will use a class of maneuver models that is popular in the literature of target tracking [42][41][38][189], where the total-state is assumed to be governed by a first-order Markov Chain (MC) whose state transition probabilities are defined via the transition equation

$$Z^{t+1} \doteq A\, Z^t + B\, U^t + \epsilon_{depen}\, g(B\, C\, Z^t). \tag{6.3}$$

In this equation the term $A\, Z^t$ captures the inertia of each individual target and the term $B\, U^t$ represents the effect of the individual independent accelerations ('maneuvers') of the targets. Hence $A \in \mathbb{R}^{2MD \times 2MD}$ is a block-diagonal matrix with the sub-matrix

$\begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}$ repeated along the diagonal, and $B \in \mathbb{R}^{2MD \times MD}$ is a block-diagonal matrix with the sub-matrix $\begin{bmatrix} \frac{T^2}{2} \\ T \end{bmatrix}$ repeated along the diagonal, with T being the time interval between successive maneuvers. The acceleration $U^t \in \mathbb{R}^{MD}$ is assumed to be independently and identically distributed for all t according to $\mathcal{N}(0, \sigma_U^2 \, I_{MD})$, and is assumed to be active throughout the real-time interval $[(t-1)T, tT]$ (i.e. the physical acceleration is assumed to be a piecewise constant process). The term $g(B\,C\,Z^t)$ captures the effect of causal interactions in-between the targets, with the scalar $\epsilon_{depen} \geq 0$ controlling the relative strength of these interactions. C is some suitably chosen *interaction feedback* matrix, and $g(\cdot)$ is some suitable *compressive* non-linearity applied component-wise to its vector argument.

Clearly, if we let $\epsilon_{depen} = 0$, we obtain a model where the tags move independently of each other, with tag m accelerating independently in each dimension according to component numbers $(m-1)D+1, \ldots, (m-1)D+D$ respectively of U^t. On the other hand if $\epsilon_{depen} > 0$, and $C, g(\cdot)$ are chosen to mimic certain heuristic rules known as Reynolds' *flocking rules* [189], we can model the motion of a variety of herds and swarms. These type of interactions fall in the category of *stochastic* constraints. Similarly, by using other appropriate $C, g(\cdot)$ (perhaps in a time-dependent fashion) we can model more deterministic constraints like 'convoy' motion or 'leapfrog' [38]. We would also like to mention that, although not stated explicitly in equation (6.3), we may have global constraints on the targets, like location restrictions within the tracking area or velocity restrictions [38]. These can be accommodated easily by our tracking algorithm (to be presented in Section 6.3) by simply applying the appropriate constraints to each *particle* in the particle representation of the propagated density. Irrespective of the nature of motion constraints, as a general rule if all other factors are held constant, we can obtain

an improvement in the tracking accuracy by exploiting the mutual information among the sub-states of co-dependently moving targets.

6.2.2 Observation Model

A single transmission from a tag consists of a packet of bits that uniquely identify it relative to other tags. An incorrect decoding of these bits can be easily and reliably detected using a Cyclic Redundancy Check (CRC) code, and such packets can be rejected by the algorithm. Hence, for a packet with a sufficiently high received signal strength (RSS), we can always correctly associate its RSSI with the source of its transmission, thus avoiding the problem of *data association* [41]. Recall that in Section 6.2.1 we had defined T to be the interval between successive maneuvers. For simplicity we will now assume that the sampling interval between the periodic transmissions from the tags is also T seconds, and that these transmissions are synchronous.[1] A commonly used model for the strength of the radio transmissions in the far-field of the transmitting tag is the *Friis model* [93]. In the tracking scenario [41] this model implies that the observations made by the motes at time tT are given by

$$Y^t_{m,n} = \Phi^t_m + \Gamma^t_{f(X^t_m),n} - \rho 10 \log_{10} \|X^t_m - r_n\| + W^t_{m,n} \quad (\text{dBmW}). \tag{6.4}$$

Here ρ is a known path loss exponent (equaling 2.0 for free-space propagation) and $W^t_{m,n}$ is a process of additive perturbations in the RSSI measurements (expressed in dB) with i.i.d. components for all t, m, n distributed according to $\mathcal{N}(0, \sigma^2_W)$. These perturbations can be attributed to several causes like the data noise in the signal, fast fading and other

[1] Neither of these assumptions is critical. The first can be relaxed by proper up-sampling of the slower of the maneuvering and observation processes. The second assumption can be easily accommodated in practice by simply using the most recent observation corresponding to each tag and pretending that they were all received synchronously. Provided T is small relative to the time constant of the dynamics of the tag, this practical solution to asynchronicity causes only a negligible degradation in performance.

imperfections of the RSSI measurement circuitry, and the thermal noise in the front-end receiver of the motes, c.f. e.g. [38][112][110]. The function $f(\cdot)$ is a *position quantizer* that takes a position vector in the tracking area and returns the index of the cell in which that position lies. Hence,

$$f : \mathbb{R}^D \to \{1, 2, \ldots, L\},\ x \mapsto f(x). \tag{6.5}$$

All *valid* observations (i.e. those which can be unambiguously associated with their transmitting targets) can in principle be used by the tracking algorithm. Due to the rapid fall of the RSS with distance, as evidenced from the form of equation (6.4), only RSSI values generated by sensors from a *local radio neighborhood* of the tag are typically valid, and hence we already have a coarse form of localization. However, as we shall see in Section 6.5, we can accentuate this effect by using a more stringent radio neighborhood for tracking a tag. Such a strategy has two useful consequences: (1) the perturbation of the accepted RSSI values is not dominated by the thermal noise of the mote receiver and hence the additive-log-perturbation model in equation (6.4) remains realistic, and (2) highly localized tracking greatly eases problems of network congestion in WSNs, as we shall see in Section 6.5.

6.2.3 Aims of the Algorithm and Overview

The aim of the tracking system is to find an estimate $\hat{X}_m^t$ for the position X_m^t of each tag $m = 1, 2, \ldots, M$ at time t, based on all the observations up to and including that epoch, namely

$$\{Y_{m,n}^\tau : \text{all } m, n;\ \tau = 1, 2, \ldots, t\}.$$

Ideally, the estimates $\{\hat{X}_m^t : \text{all } m\}$ should be optimal in the sense that they minimize the normalized total expected squared tracking error at time t,

$$(\Delta^2)^t = \frac{1}{M}\sum_{m=1}^{M} \mathbb{E}\left[\|\hat{X}_m^t - X_m^t\|^2\right]. \tag{6.6}$$

The tracking system consists of two sub-algorithms: (a) A target tracking sub-algorithm based on particle filtering, and (b) a stochastic incremental estimation sub-algorithm based on maximum likelihood principles, that learns the radio environment. These two modules use the observations provided by the motes as well as each other outputs, thus leading to a recursive or 'Turbo' structure.

Particle Filtering

Suppose, for the time being, that the parameter processes $\{\Phi_m^t\}$ and $\{\Gamma_{l,n}^t\}$ are *known* to the user. Then, the optimal position estimates are obtained via the well known Baum-Welch Hidden Markov Model (HMM) filter [117]. Unfortunately, due to the non-linearity of the observation model, the filter does not have a simple closed form solution, and an approximation must be used. One approach is to model the propagated state to have a Gaussian distribution and use an Extended or Unscented Kalman Filter [190][41][187]. However, our simulations as well as real-life experiments indicate that the propagated density is highly non-Gaussian and multi-modal, especially in scenarios of high noise and co-dependent motion, and hence a better approximation is obtained via a Particle Filter [188]. It is known that the quality of the particle approximation can in principle be improved arbitrarily close to the ideal filter by using a sufficiently large number of particles [188][128]. However, in view of tractability, we will be interested in achieving good performance with a *limited number of particles*, say 128.

State-space Reduction via Sub-state Filters

In an optimal implementation of the filter, at every sampling epoch we should calculate the joint distribution of the total-state, conditioned on all the observed RSSI values to-date. This joint distribution should then be marginalized exactly onto the sub-states, since these marginals form sufficient statistics for optimally inferring their positions. Unfortunately, the calculation and manipulation of the joint distribution and its subsequent exact marginalization becomes intractable as the number of targets M becomes large. This is true even if we use a particle filter, because the total system state increases in dimension as the number of targets tracked (M) is increased, and a particle approximation with a *fixed* number of particles becomes progressively poor. Moreover, a distributed implementation of the filter is clearly precluded. To circumvent this problem, analogous to the distributed filter proposed in [115], we always approximate the joint distribution of the total-state via a product of its marginals on the sub-states (the so-called *mean-field* approximation [154]). As a result we never calculate and store the joint distribution *per se*, but rather directly calculate the marginals of the current epoch using the marginals from the previous epoch, while taking into account the joint interactions implied by the maneuver model. This is done by implementing a bank of M particle filters, one each for the *sub-state* corresponding to every target. Each filter operates with a fixed number (Π) of particles of dimension $2D$, irrespective of the number of targets M. The dependencies in the maneuvers of the targets are exploited only by exchanging some summary information among the filters, like the a-posteriori expectations of the sub-states. Even these marginals statistics are gathered only from localized neighborhoods, to allow a consistently scalable solution. Note that an analogous approach called 'multiple particle filtering', based on state-partitioning and marginally interacting sub-state filters, has also been recently proposed by [130]. However they have not considered the question

of a physically distributed scalable implementation, and hence have assumed a global neighborhood for gathering sub-state marginals. Similarly, they do not compensate for unknown radio environments parameters, unlike our approach (as we shall see in the next section).

Note that if there are no dependencies in the motion of the targets (see Section 6.2.1) then the algorithm to be proposed in Section 6.3 will in fact give an exact realization of the filter, and not an approximation.

Stochastic Recursive Estimation

In reality the parameter processes $\{\Phi_m^t\}$ and $\{\Gamma_{l,n}^t\}$ are *not known* to the user. Hence we need to estimate them in tandem with the tracking procedure. Since typically nothing is known about their priors or their stochastic evolution, we cannot treat them as random processes and use a Bayesian approach. Instead we must use a maximum likelihood (ML) estimation approach. Even in the latter setup, since there are hidden variables in the overall statistical model (namely the sequence of positions and velocities of the tags), the calculation of the true ML estimate necessitates the use of the full-fledged EM algorithm of Dempster et al. [122]. This is impractical for two reasons. Firstly, just the storage and manipulation of very long sequences of observations can be difficult. Secondly, each recursion of the EM procedure requires an expectation step for which we must solve for the a-posteriori probability distribution of the *entire state sequence* $\{Z_m^t\}$. This is prohibitively expensive in terms of computational requirements even in a centralized implementation at the FC, and is decidedly incompatible with a distributed system.

In view of these difficulties, rather than using a full-fledged EM algorithm, we use a *recursive* estimator which is a form of stochastic EM [191][172][121]. In this method, based on the current estimate of the parameters, and using the RSSI observations re-

ceived in the current sampling interval, we first infer the a-posteriori distribution of the current total-state of the system from the distribution at the previous sampling time. Then we *re-estimate* the parameters by adding an innovation that depends only on the inferred distribution of the current state, the most recent parameter estimates, and only the current RSSI observations. This two step procedure is repeated for all times epochs. Such a recursive stochastic approximation is known to yield a consistent parameter estimation algorithm provided the parameters vary slowly as compared to the dynamics of the tags [178], and the Fisher information of the parameters at the desired fixed point has a full rank.

In the next section, we will consider the problem of multi-target tracking in a known radio environment, while in Section 6.4 we will tackle the problem of estimating the radio environment itself. Section 6.5 will then consider the issue of a distributed scalable implementation of the tandem operation of these two algorithms.

6.3 Tracking With an SIR Particle Filter

In this section we will use the current parameter estimates, namely $\{\hat{\Phi}_m^t : \text{ all } m\}$ and $\{\hat{\Gamma}_{l,n}^t : \text{ all } l, n\}$, as if they are the true parameters, and perform the filter update of the propagated probability density function (p.d.f.) of the total system state. We will the reserve the use of $q(z)$ to denote the a-posteriori p.d.f. of the total state, since it has special significance in the tracking algorithm.

The optimal estimate of the state in the sense of minimum mean squared error (MMSE) is always given by [144]

$$\hat{Z}^t = \int_{\mathbb{R}^{2MD}} z^t \, q(z^t) dz^t \tag{6.7}$$

where $q^t(z^t)$ is the a-posteriori density of the total-state at time t conditioned on all the observations up to and including sampling time index t. Since the maneuver and observation equations imply an HMM for the total state, this a-posteriori distribution is calculated exactly by an intractable stochastic filter [117]

$$q^t(z^t) = p(y^t|z^t) \int_{\mathbb{R}^{2MD}} p(z^t|z^{t-1})\, q^{t-1}(z^{t-1})\, dz^{t-1}. \tag{6.8}$$

As discussed in Sections 6.2.3 and 6.2.3, for achieving a tractable algorithm we must use some type of approximation. In particular, we will use a bank of particle filters, one each for the sub-state of the corresponding target. Let $\mathbb{R}^{2D} \ni \zeta^t_{m,\pi}, \pi = 1, 2, \ldots, \Pi$ be the Π particles representing $q^t_m(z^t_m)$, the marginal a-posteriori distribution at time t of the sub-state of target m. The particle representation is made via the identity

$$q^t_m(z^t_m) \;\approx\; \frac{1}{\Pi} \sum_{\pi=1}^{\Pi} \delta_{\zeta^t_{m,\pi}}(z^t_m), \tag{6.9}$$

where $\delta_\zeta(z)$ is the Dirac-delta function centered at ζ, that satisfies the sifting property $\int h(z)\delta_\zeta(z)\, dz = h(\zeta)$. Then, in our practical implementation, the estimate of the sub-state of target m is given by

$$\hat{Z}^t_m = \int_{\mathbb{R}^{2D}} z^t_m\, q_m(z^t_m) dz^t_m \approx \frac{1}{\Pi} \sum_{\pi=1}^{\Pi} \zeta^t_{m,\pi}. \tag{6.10}$$

The particles $\{\zeta^t_{m,\pi}, \pi = 1, 2, \ldots, \Pi\}$ are calculated using the previous particles $\{\zeta^{t-1}_{m,\pi}, \pi = 1, 2, \ldots, \Pi\}$ and the a-posteriori marginal *expectations* of all the other targets at time $t-1$, namely $\{\hat{Z}^{t-1}_{m'} : m' = 1, 2, \ldots, M,\ m' \neq m\}$. The calculation is done by using a standard Sampling-Importance-Resampling (SIR) particle filter [128] with certain modifications that allow the integration of the information available from the other cooperating sub-state filters. This involves the uplifting of the particles to

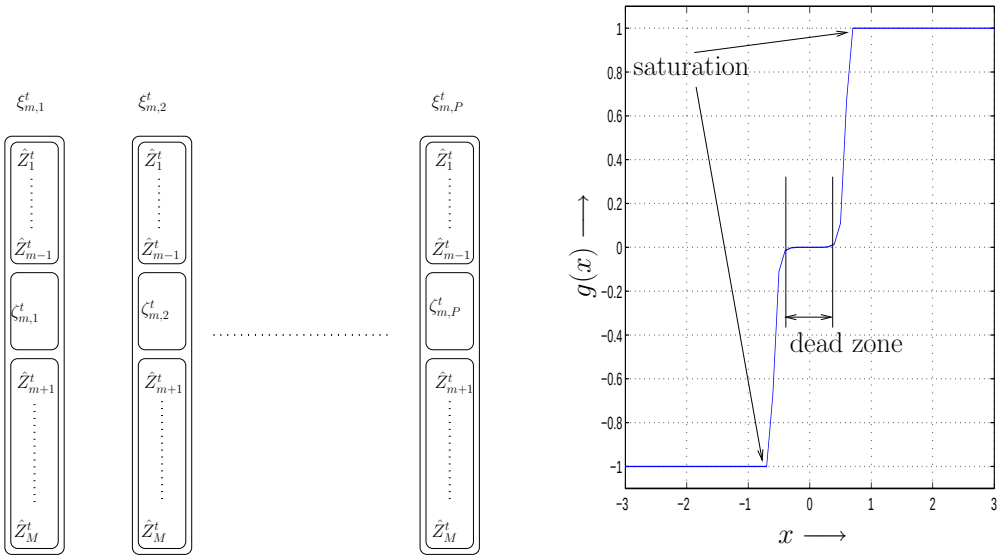

Figure 6.2: (a) Construction of full-state particles from sub-state particles of target m and marginal expectations of the sub-states of targets $m' \neq m$. (b) The non-linearity $g(\cdot)$ used in modeling the interaction in the motion of the targets.

the full state-space, merging the information about the other targets, performing the SIR update, and finally projecting the particles back to the marginal state-space. The following is a pseudo-code of our algorithm, executed for each target $m = 1, 2, \ldots, M$:

1. Initialize the Π sub-state particles $\{\zeta_{m,\pi}^{t-1}, \pi = 1, 2, \ldots, \Pi\}$ by drawing them from a *uniform distribution* over the entire tracking region (which is assumed to have a finite area). Then for each time epoch $t = 1, 2, 3, \ldots$, execute all the following steps.

2. From the Π sub-state particles $\{\zeta_{m,\pi}^{t-1}, \pi = 1, 2, \ldots, \Pi\}$ create Π full-state particles $\{\xi_{m,\pi}^{t-1}, \pi = 1, 2, \ldots, \Pi\}$ by adjoining each particle with the marginal expectations $\hat{Z}_{m'}^{t-1}$ of the sub-states for all the others targets $m' \neq m$. See Figure 6.2(a) for a graphical illustration.

3. Propagate all these full-state particles according to the maneuver model:

$$\xi^t_{m,\pi} \leftarrow A\,\xi^{t-1}_{m,\pi} + B\,\tilde{U}^t + \epsilon_{depen}\,g(B\,F\,\xi^{t-1}_{m,\pi}). \tag{6.11}$$

In doing so use generate the innovation process $\tilde{U}^t$ having the same statistical properties as U^t.

4. Importance-Resample the propagated full-state particles based on the observation likelihood. That is, calculate the *importance* of particle number π as

$$\begin{aligned} w^t_{m,\pi} &\doteq p(\{Y^t_{m,n} : \text{ all } n\}|\xi^t_{m,\pi}) && (6.12)\\ &= \prod_{n=1}^{N} \exp\left\{ \frac{-1}{2\sigma_W^2} \begin{pmatrix} Y^t_{m,n} - \hat{\Phi}^t_m - \hat{\Gamma}^t_{f(\xi^t_{m,\pi}),n} \\ +\rho 10 \log \|\xi^t_{m,\pi} - r_n\| \end{pmatrix}^2 \right\}. && (6.13) \end{aligned}$$

Normalize these importances so that they sum to one. Then draw Π numbers from $1,\ldots,\Pi$ (with replacement) using the distribution $\{w^t_{m,\pi} : \pi = 1,2,\ldots,\Pi\}$. Use these as indices to select Π particles from the pool $\{\xi^t_{m,\pi} : \pi = 1,2,\ldots,\Pi\}$. Call this importance-resampled set $\{\chi^t_{m,\pi} : \pi = 1,2,\ldots,\Pi\}$.

5. Project the full-state particles in $\{\chi^t_{m,\pi} : \pi = 1,2,\ldots,\Pi\}$ to the m^{th} sub-state, by discarding the components corresponding to $m' \neq m$. Call the resulting set $\{\zeta^t_{m,\pi} : \pi = 1,2,\ldots,\Pi\}$.

Due to the re-sampling operation, all the particles at the end of an update have equal weight, hence we do not need to explicitly maintain their weights. It is known [128] that such re-sampling eliminates the problem of *degeneracy*, but can in turn lead to *particle impoverishment*. While we preferred to use the proposal density $p(z^t|z^{t-1})$ because it is a conditional-normal, which is very easy to sample from, it is well known in literature that we can use more elaborate proposals to combat the problem of particle impoverishment.

(Then we also need to modify the weights appropriately.) In particular the 'optimal' proposal in this sense is $p(z^t|z^{t-1}, y^t)$, which depends on the measurements in addition to the previous state, and minimizes the variance of the weights [44]. Unfortunately, due to the serious non-linearity in the observation equation, the optimal proposal *cannot be written in a simple tractable closed form* and some kind of approximation needs to be calculated. Keep in mind that the computational cost of this approximation needs to be paid *in each update of the filter*. Comparing the complexity of drawing fewer particles from a complex prior vs the complexity of drawing somewhat more particles from a simple conditional normal, we concluded that the latter was a better option. Moreover we found that, at least in the presently considered RSSI-based tracking problem, impoverishment does not seem to be a very serious problem under the choice of $p(z^t|z^{t-1})$, and can be easily side-stepped by the simple stratagem of ensuring that the particle innovation process $\{\hat{U}^t\}$ has a significant power by design. That is, even though $\{\hat{U}^t\}$ is deemed to be statistically identical to the true innovation process $\{U^t\}$, we choose its power to be $\max(\sigma^2_{thresh}, \sigma^2_U)$. By using a σ^2_{thresh} as large as possible while still being compatible with the maneuver interval T and the end application, we can ensure that we do not suffer from particle impoverishment even if σ^2_U is itself small. In simulations just $\Pi = 128$ particles were found adequate and there was no significant degradation in the tracking accuracy in this approach relative to using more complicated proposals.

6.4 Stochastic Incremental Estimator of the Radio Environment

Now we will consider the problem of incrementally estimating the radio environment, utilizing the observations made by the motes as well as the results of the tracking

algorithm described in Section 6.3. Recall that $q_m^t(z_m^t), m = 1, 2, \ldots, M$ are the a-posteriori marginal distributions of all the tag sub-states at time t, and these are approximately calculated by the bank of particle filters and presented in the form of particles $\{\zeta_{m,\pi}^t : \pi = 1, 2, \ldots, \Pi\}, m = 1, 2, \ldots, M$. As far as the estimator is concerned the joint distribution of the tags is *assumed* to be the product of these marginals, hence the sub-states of the tags are assumed to be *independent* of each other and distributed according to $q_m^t(z_m^t), m = 1, 2, \ldots, M$ respectively. Similarly, we know the current estimate of the parameters, namely $\hat{\Phi}_m^t$ and $\hat{\Gamma}_{l,n}^t$, and we have access to the current observations $\{Y_{m,n}^t : \text{all } m, n\}$. Based on this data, and the knowledge of the observation model, we want to calculate the updated estimates of the parameters, namely $\hat{\Phi}_m^{t+1}$ and $\hat{\Gamma}_{l,n}^{t+1}$.

For conciseness, let us stack the parameters $\{\hat{\Phi}_m^t : \text{all } m\}$ and $\{\hat{\Gamma}_{l,n}^t : \text{all } l, n\}$ into a single vector parameter $\hat{\Upsilon}^t$ using a stacking function $f_{stack} : \mathbb{R}^M \times \mathbb{R}^{L \times N} \rightarrow \mathbb{R}^{M+LN}$. Now consider a stochastic incremental algorithm for updating $\hat{\Upsilon}^t$, which falls in the category of 'partial-M step' EM algorithms [121][191][172]:

$$\hat{\Upsilon}^{t+1} = \hat{\Upsilon}^t + \epsilon \, F^{-1} \, S(\hat{\Upsilon}^t; \{Y_{m,n}^t : \text{all } m, n\}). \tag{6.14}$$

Here $S(\hat{\Upsilon}^t; \{Y_{m,n}^t : \text{all } m, n\})$ denotes the *score* of the log-likelihood of the observations $\{Y_{l,n}^t : \text{all } m, n\}$ viewed as a function of the parameter $\hat{\Upsilon}^t$. That is,

$$S(\hat{\Upsilon}^t; \{Y_{l,n}^t : \text{all } m, n\}) \doteq \frac{\partial \lambda(\hat{\Upsilon}^t; \{Y_{m,n}^t : \text{all } m, n\})}{\partial \Upsilon} \tag{6.15}$$

where

$$\lambda(\hat{\Upsilon}^t; \{Y_{l,n}^t : \text{all } m, n\}) \doteq \log \Big(p(\{Y_{m,n}^t : \text{all } m, n\} | \hat{\Upsilon}^t) \Big). \tag{6.16}$$

The expression for the score has been evaluated in Appendix E.1, where we also indicate how it can be calculated in practice by using the particles provided by the tracking

algorithm. (Note that for brevity and clarity, we have dropped the time super-script on the various quantities in Appendix E.1.) The scalar ϵ is a step size that controls the tradeoff between the parameter tracking speed and steady-state error variance, and F^{-1} is a suitably chosen positive definite pre-scaling matrix for the score. Note that (6.14) is a stochastic recursion since the score $S(\hat{\Upsilon}^t; \{Y^t_{m,n} : \text{all } m, n\})$ is a random vector. It is known that the score, when evaluated at the true parameter value governing the observations, yields a random vector of *zero mean* [5]. Hence it follows that the *averaged gradient* [178] version of recursion (6.14) respects the true parameter as a fixed point. Its stability is determined by the covariance matrix of the score, i.e. the Fisher information F_Υ of the parameter Υ, which can be equivalently written as [146]

$$F_\Upsilon = -\mathbb{E}\left[\frac{\partial^2 \lambda(\Upsilon; \{Y_{m,n}\})}{\partial \Upsilon^2}\right]. \tag{6.17}$$

Provided that $F_\Upsilon > 0$, and we start from a suitable initial estimate $\hat{\Upsilon}^0$, recursion (6.14) converges to the true (quasi-static) parameter and stays in its vicinity with high probability [121][191][172].

An exact closed form derivation of the Fisher information in the present scenario seems infeasible, hence in Appendix E.1 we calculate an optimistic approximation

$$F_\Upsilon <\approx \hat{F}_\Upsilon = \frac{1}{\sigma_W^2} \begin{pmatrix} NI_M & \frac{1}{L}\mathbf{1}_{M\times LN} \\ \frac{1}{L}\mathbf{1}_{LN\times M} & \frac{M}{L} I_{LN} \end{pmatrix}, \tag{6.18}$$

where I_M denotes an $M \times M$ identity matrix and $\mathbf{1}_{M\times LN}$ denotes an $M \times LN$ matrix whose entries are all ones. Closer inspection of this matrix reveals that it is rank deficient by one. That is, it is a square matrix of size $M + LN$ but has rank $M + LN - 1$. Hence there is one degree of ambiguity regarding the parameter, given the observations. The physical reason for this is easily understood by noting that if we increase all the transmit

powers $\{\Phi\}$ by an arbitrary amount and reduce all the gains $\{\Gamma\}$ by an equal amount, we will get exactly the same set of observations from the model in equation (6.4). In other words, the observations cannot uniquely determine the model. This also implies that a gradient ascent algorithm (with or without pre-scaling), where the gradient is averaged w.r.t. all sources of stochasticity, is *not asymptotically stable* at the true parameter even though it respects it as a fixed point. As a result, the underlying stochastic algorithm will also be unstable.

Fortunately, this problem is easily solved by fixing at least one parameter from the set Υ and postulating that it does not need to be re-estimated. This reduces the size of the Fisher information matrix, which is now some principal sub-matrix of the above matrix and is therefore guaranteed to be full ranked, thus ensuring asymptotic stability. For example one could postulate that one tag has been very accurately calibrated so that its transmit power is perfectly known. Another good choice of the parameters that can be fixed is the following:

$$\hat{\Gamma}^t_{f(r_n),n} = \Gamma^t_{f(r_n),n} = 0, \quad n = 1, 2, \ldots, N, \, \forall t \tag{6.19}$$

That is, we may assume that there is no gain or attenuation (excluding distance dependent path loss) of the radio transmission from a tag to a mote whenever the tag is in the same cell as the mote. This is reasonable because if a tag and mote are in the same cell, then by construction they are bound to be close to each other and hence not have intervening objects that could lead to any extra attenuation.

Whatever the choice of the parameters to be fixed, let $\mu(\hat{F}_\Upsilon)$ denote the corresponding principal sub-matrix of $\hat{F}_\Upsilon$. Let

$$COV(\hat{\Upsilon}^t - \Upsilon) = \mathbb{E}\left[(\hat{\Upsilon}^t - \Upsilon)(\hat{\Upsilon}^t - \Upsilon)^T\right]$$

denote the expected covariance matrix of the error in estimating a constant truth parameter Υ. Then, in steady state and under the simplifying assumptions made in Appendix E.1, we have an optimistic approximation of the Cramer-Rao lower bound (CRLB) [123][124]

$$\mu\Big(COV(\hat{\Upsilon}^t - \Upsilon)\Big) \geq \frac{\epsilon}{2-\epsilon}\mu\Big(\hat{F}_\Upsilon\Big)^{-1}. \tag{6.20}$$

In the above expression, the factor $\frac{\epsilon}{2-\epsilon}$ accounts for the time-constant (data-window) of the incremental estimation algorithm. Note that the bound 6.20 is not necessarily achievable since we are using an optimistic approximation of F_Υ.

In recursion (6.14), if F is chosen to be an identity matrix, the algorithm performs a regular gradient ascent. On the other hand, if F is chosen to be F_Υ, the algorithm performs a natural gradient ascent [149], which has better stability and acquisition properties. Furthermore, it can be shown [191] that natural gradient ascent of the log-likelihood asymptotically achieves the minimal possible steady state estimation variance specified by the CRLB. Hence it is beneficial in practice to use the pre-scaling matrix $F^{-1} = \hat{F}_\Upsilon^{-1}$.

6.5 Distributed Implementation And Scalability

First consider the scenario of a centralized implementation of the filter bank of Section 6.3 at a *Fusion Center* (FC), while assuming that the radio environment parameter Υ is known by some means. In such an implementation, all valid observations of all motes need to be streamed to the FC via multi-hop communication. Hence, as the network scales, the communication load of each mote increases monotonically, and a regime inevitably occurs when the network is pushed into congestion and/or the lifetime of the motes becomes unacceptably small. This serious drawback of centralized tracking is inescapable whether or not we choose to exploit inter-target dependencies.

As a solution to this problem, we will now propose a distributed power efficient implementation of the filter bank which allows the system to scale seamlessly to larger tracking regions without encountering problems of network congestion and short lifetimes. Our proposal is based on two ideas: (A) *Localized ownership*: At any point in time, a single 'owner' mote lying in close proximity to a target is responsible for tracking that target. If the target starts moving out of its proximity, the owner transfers the ownership to a new mote that is closer to the tag. (B) *Localized data aggregation:* While tracking, an owner mote relies on measurements and sub-state information solely from a local neighborhood of the target.

In the following, a *transmission* by a mote will mean a single-hop broadcast transmission to one or more of its nearest neighbors. The *normalized communication load* will mean the average number of transmissions that are made by each mote in each observation interval T.

6.5.1 Localized Ownership

Definition 7 *At each time t, every target m always has a unique owner mote $\Omega^t(m) \in \{1, 2, , \ldots, N\}$, who is solely responsible for maintaining and updating the sub-state particles $\{\zeta^t_{m,p} : p = 1, 2, \ldots, \Pi\}$, according to the algorithm of Section 6.3.*

Initial ownership may be assigned arbitrarily. Later, the decision to change the ownership of tag m can be made only by its current owner $\Omega^t(m)$. It periodically examines the tag's current position, which it knows with high certainty from the particles it maintains. If tag m has moved out of its proximity, $\Omega^t(m)$ polls the mote that is now nearest to the target, and transfers ownership to that mote, provided it is willing to take on the responsibility. Otherwise it polls the next closest one, and so on. Assuming that the targets visit various parts of the tracking region equi-probably, this simple change-of-ownership protocol

ensures an equitable distribution of the computation and communication load across the sensor network.

We will require that the owner $\Omega^t(m)$ should make his ownership of m known to the motes that belong to a certain set which we call the *tracking neighborhood* of target m at time t.

Definition 8 *The tracking neighborhood is defined as*

$$\eta_\varrho^t(m) = \{n \in \{1, 2, \ldots, N\} : \|r_{\Omega^t(m)} - r_n\| < \varrho\}, \tag{6.21}$$

where ϱ is a user specified aggregation radius (in meters), that controls the size of the tracking neighborhood.

Using an efficient localized broadcast algorithm like [192], which assures guaranteed delivery throughout the neighborhood using $O(|\eta_\varrho^t(m)|)$ *fixed-length* messages, the normalized communication load of publishing the ownership information can be shown to be $O(\frac{M\varrho^D}{Nt_{ch}})$, where t_{ch} is the average time between the change of ownership of a target.

6.5.2 Localized Data Aggregation

The propagation of the sub-state particles for tag m via equation (6.11) requires that the previous sub-state estimates of all the other tags, namely $\hat{Z}_{m'}^{t-1}$, $m' \neq m$, be made available for the creation of the full-state particles, which entails a normalized communication load of $O(M)$. (Of course, if the tags are known, or presumed, to be moving *independently*, this contribution to the communication load will be zero.). Similarly, the importance sampling via equation (6.12) requires that all the valid observations $\{Y_{m,n}^t : \text{all } n\}$ be made available for calculation of the weights, which entails a normalized communication load of $O(N)$. For achieving a truly scalable tracking system, we

need to ensure that both these contributions to the normalized communication load are reduced to $O(1)$, by using some kind of approximation.

To this end, we postulate that data aggregation at $\Omega^t(m)$ be performed only from the tracking neighborhood $\eta^t_\varrho(m)$. That is, the propagation of the sub-state particles $\{\zeta^{t-1}_{m,\pi}, \pi = 1, 2, \ldots, \Pi\}$ is based only on the sub-states $\{\hat{X}^t_{m'} : \Omega^t(m') \in \eta^t_\varrho(m)\}$, with the unavailable sub-states being set to zero. (The interaction feedback matrix F also needs to be appropriately modified; for an example see Section 6.6.) Similarly, the importance re-sampling of the particles of target m at time t is performed only on the basis of the measurements $\{Y^t_{m,n} : n \in \eta^t_\varrho(m)\}$.

Our basic premise is that we can use a very localized neighborhood (i.e. a small ϱ), and still achieve a performance close to what can be achieved with non-localized tracking ($\varrho = \infty$). This is possible because the signal received at the motes in $\eta^t_\varrho(m)$ is considerably more informative about the position of tag m at time t, while the signal received by motes outside $\eta^t_\varrho(m)$ is dominated by the measurement noise and contains relatively little position information. (Errors in ranging and angle estimation are more deleterious when they occur in measurements made from far-away motes.)

We will give convincing simulation evidence of this hypothesis in Section 6.6 for the case of RSS based tracking, though, of course, this principle applies to other tracking metrics like TOA, DOA etc as well. Assuming for now that there is no loss in choosing a small constant value for ϱ, we now proceed to note that the data transfer from each mote in $\eta^t_\varrho(m)$ to $\Omega^t(m)$ can be accomplished with a small *constant* number of hops proportional to ϱ, possibly even a single hop. Hence it follows that the contribution to the normalized communication load from data aggregation is reduced to $O(\frac{M\varrho^{D+1}}{N})$. Thus it follows that our proposal for a distributed implementation of the particle filter bank has a total normalized communication load of $O(\frac{M\varrho^D}{Nt_{ch}} + \frac{M\varrho^{D+1}}{N})$, which implies that we have a truly scalable implementation. That is, provided the tracking area, the

number of motes and the number of targets are all increased in the same proportion, i.e. $M \propto N \propto \aleph, \aleph \uparrow \infty$, and the spatial density of the motes is kept roughly constant over the tracking region, the normalized communication load, and hence the power drain in each mote, remains *invariant*. Furthermore, since typically $t_{ch} \gg 1$ in sparsely deployed networks, this invariant communication load is dominated by the data aggregation than by the publishing of ownership information.

6.5.3 A Note on Parameter Estimation

Earlier we assumed that the estimated parameters $\{\hat{\Phi}_m\}$ and $\hat{\Gamma}_{l,n}\}$ are available at each mote. This can be achieved by using a fully distributed implementation of the incremental estimator, along the lines of the distributed filter. It requires a somewhat larger complexity of inter-mote coordination (for example we need two kinds of owners), but is essentially similar to the method outlined above. We will not discuss the details here, except noting two principal conclusions: (a) An $O(M)$ normalized communication load is again sufficient to achieve a fully distributed implementation in the most general case. (b) By restricting the gradient calculation of $\hat{\Gamma}_{l,n}$ to be based on particles only from tags that are (estimated to be) in cell l, the normalized communication load can be reduced to $O(1)$, and hence we can achieve strict scalability.

Finally, it should also be noted that the parameters are typically *quasi-static*. This implies that the incremental estimation in equation (6.14) need not be implemented in every sampling interval. Rather we can implement it with a very small duty cycle, like once every thousand sampling intervals. So in many environments we may simply choose to execute the parameter estimator in a centralized form at the FC. For this we need to gather, very infrequently, all the sensor observations and the sub-state particles at the FC, carry out the recursion in equation (6.14), and broadcast the new estimates back

to all the motes.

6.6 Simulations and Discussion

In Section 6.6.1 we will investigate the properties of the tracking system proposed in Sections 6.3 and 6.4 via simulations, while in Section 6.6.2 we will simulate the distributed implementation proposed in Section 6.5. All the simulations will be done in two dimensions ($D = 2$).

6.6.1 Simulations of Tracking and Parameter Estimation

The simulations in this section are done in two parts:

- *Target Tracking:* In the first part we will study the performance of the tracking sub-algorithm alone, by assuming a static radio environment and initializing the incremental estimator with the true environment parameters $\{\Phi\}, \{\Gamma\}$. Note that the *estimator is nevertheless kept operational* as it would be in a real-life scenario, in order to confirm our prediction regarding its asymptotic stability. In these experiments, we will investigate the root-mean-squared (RMS) tracking error Δ as a function of (a) σ_W, the measurement noise standard deviation, and (b) N, the number of active motes in the tracking area.
- *Parameter Estimation:* In the second part we will study the joint performance of the tracking and estimation algorithms from three points of view: (a) First we will consider a static model, presumed unknown to the system, and study the acquisition behavior of the estimator, and its effect on the RMS target tracking error Δ. (b) Secondly we will compare the steady state parameter estimation squared error to the estimate of the CRLB given in equation (6.20). (c) Lastly we will consider

a slowly varying radio-environment model, and demonstrate that the estimator is able to seamlessly track these parameter variations using an adequate step size, while ensuring that there is no noticeable effect on the performance of the target tracking algorithm.

In these simulation experiments we will use a realistic synthetic radio environment consisting of four rooms of unequal sizes, as shown in Figure 6.1. The total tracking area is 20.0 meters square, with nine motes placed in it on a uniform square grid of minimum distance 10 meters, and labeled as shown in Figure 6.1. The number of *active* motes is denoted by $1 \leq N \leq 9$, with a default value $N = 9$ unless otherwise specified. (Note that if we expand such a setup to a larger area, as we do in Section 6.6.2, while keeping the density of motes constant, the performance of the tracking and estimation algorithms remains essentially unchanged). We assume that each room is a cell (thus $L = 4$) and that the nominal gain of the tag's signal because of a wall is $\gamma_{wall} = -2.0$ dB. Let $\nu_{l,n}$ denote the number of walls crossed by a straight-line path from the cell l to the mote n. The numbers $\nu_{l,n}$ can be gleaned from the building plan in Figure 6.1, and are given by

$$\nu = \begin{pmatrix} 0 & 1 & 1 & 2 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 2 & 1 & 1 & 1 & 0 & 1 & 2 \\ 1 & 2 & 0 & 1 & 1 & 1 & 2 & 1 & 0 \\ 2 & 1 & 1 & 0 & 2 & 2 & 1 & 2 & 1 \end{pmatrix} \tag{6.22}$$

As discussed in Section 6.4, in order to ensure a full rank for the Fisher information, we will freeze at zero a few components of Γ and $\hat{\Gamma}$ according to equation (6.19).

In this subsection we will assume that $M = 8$ targets are moving around in the tracking area, with an innovation power of $\sigma_U^2 = 10^{-35/10}$ (i.e. -35 dB). (Recall that the number of targets is a constant, and presumed known to the tracking algorithm.)

In the tracking algorithm we use $\sigma^2_{thresh} = 0.1$ (i.e. -10 dB). For now we will consider only independent motion of targets by choosing $\epsilon_{depen} = 0$, while the scenario of co-dependent motion and its exploitation for improved tracking accuracy will be studied in Section 6.6.2.

As remarked earlier, we consider two types of radio environments: static and dynamic.

Static Radio Environment Model

The transmission from cell l to mote n is modeled to undergo a time-invariant gain given by

$$\Gamma_{l,n} = \nu_{l,n}\gamma_{wall} + \eta_{l,n}\,(1 - \mathbb{I}_l(r_n)),$$

where $\eta_{l,n}$ are independently drawn from a normal distribution of zero mean and standard deviation $\sigma_\Gamma = 3.0$ dB. (The function $\mathbb{I}_l(\cdot)$ is defined in Appendix E.1.) Similarly, the actual transmit power of tag m is modeled to be a time-invariant quantity given by

$$\Phi_m = \Phi_{nominal} + \phi_m \tag{6.23}$$

where ϕ_m are independently drawn, for all m, from a normal distribution with zero mean and a standard deviation $\sigma_\Phi = 3.0$ dB. Once the parameters $\{\Gamma_{l,n}\}$ and $\{\Phi_m\}$ are drawn, they are kept fixed for the duration of an experiment.

Dynamic Radio Environment Model

The cell-to-mote gains are modeled to be functions of time, given by

$$\Gamma^t_{ln} = \nu_{l,n}\gamma_{wall} + \sigma_\Gamma\,\sin(2\pi t/t_\Gamma)\,(1 - \mathbb{I}_l(r_n)).$$

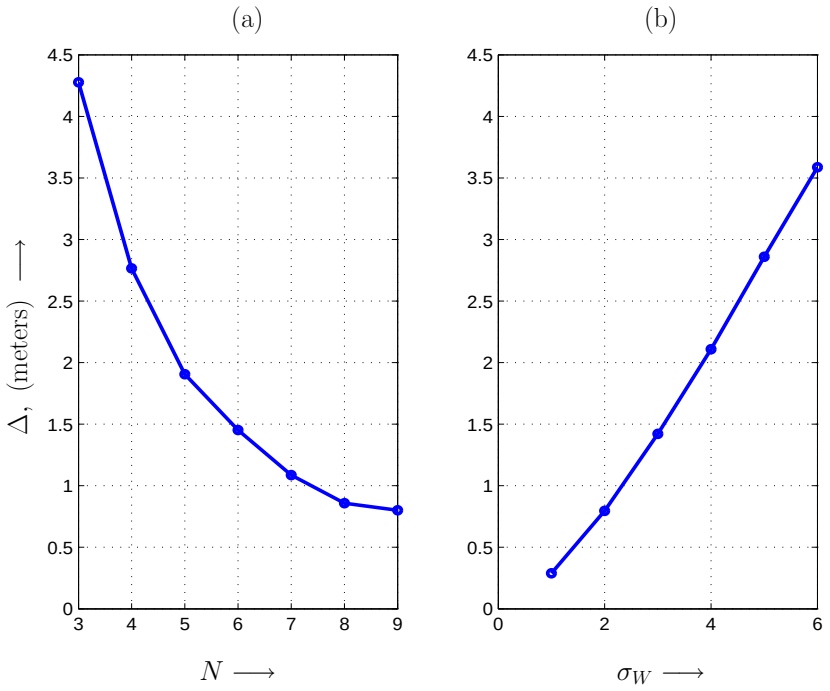

Figure 6.3: Effect on tracking accuracy of (a) the number of motes N, with σ_W fixed at 2.0, and (b) the measurement noise standard deviation σ_W, with N fixed at 9.

Similarly the transmit powers are modeled to be functions of time given by

$$\Phi_m^t = \Phi_{nominal} + \sigma_\Phi \sin(2\pi t/t_\Phi).$$

We again choose the hyper-parameter values $\sigma_\Phi = \sigma_\Gamma = 3.0$. Also, since our experiments on a real-life WSN test-bed suggest that the transmit power changes more slowly than the radio gains, we choose the following practically relevant time constants: $t_\Gamma = 5 \times 10^4$ and $t_\Phi = 2 \times 10^5$.

For the static as well as the dynamic model, the estimated parameters $\{\hat{\Gamma}_{l,n}\}$ and $\{\hat{\Phi}\}$ are initialized to their known nominal values, namely $\{\nu_{l,n}\gamma_{wall}\}$ and $\Phi_{nominal}$ respectively.

Simulation Results: Target Tracking

Figure 6.3 displays the effect on the RMS tracking accuracy Δ in meters, as defined in equation (6.6), of the number of motes N and the measurement noise standard deviation σ_W. The radio environment is kept static. The mean squared tracking accuracy Δ^2, in principle an ensemble expectation, is calculated by doing a single sufficiently long run and taking a time-average of the squared error (ignoring the small initial transient). An assumption of ergodicity, as is commonly made in literature, is implicit in such a calculation. It is seen that the dependence on both quantities N and σ_W is roughly linear, except when N is large and σ_W is small. The saturation effects can be attributed to our use of a finite number of particles and an artificially inflated power for the particle innovation process (cf. Section 6.3). Note that the domain of a large N and a small σ_W is in fact practically unimportant since we wish to operate with a sparse network and cheap receivers. The simulation results thus indicate that, within a broad range of parameters, we can exchange the mote density for an improved RSSI measurement accuracy, without sacrificing the tracking error.

Simulation results: Parameter Estimation

In Figure 6.4 we illustrate an example of acquisition of a randomly chosen static radio environment, as described in Section 6.6.1. We have

$$\Phi = [0.04, 4.78, 0.41, -7.85, -1.35, 5.20, -5.83, 3.63],$$

$$\Gamma = \begin{pmatrix} 0.00 & -3.86 & 2.40 & -4.89 & 0.00 & 0.00 & 5.91 & 0.00 & -11.04 \\ 2.05 & 0.00 & -5.52 & -4.91 & -7.69 & -0.59 & 0.00 & 0.33 & -5.37 \\ -4.61 & -4.27 & 0.00 & -6.62 & -1.94 & -4.24 & -10.72 & 4.65 & 0.00 \\ -4.82 & -5.78 & -5.91 & 0.00 & -2.11 & -11.73 & -1.75 & 2.35 & -3.23 \end{pmatrix}.$$

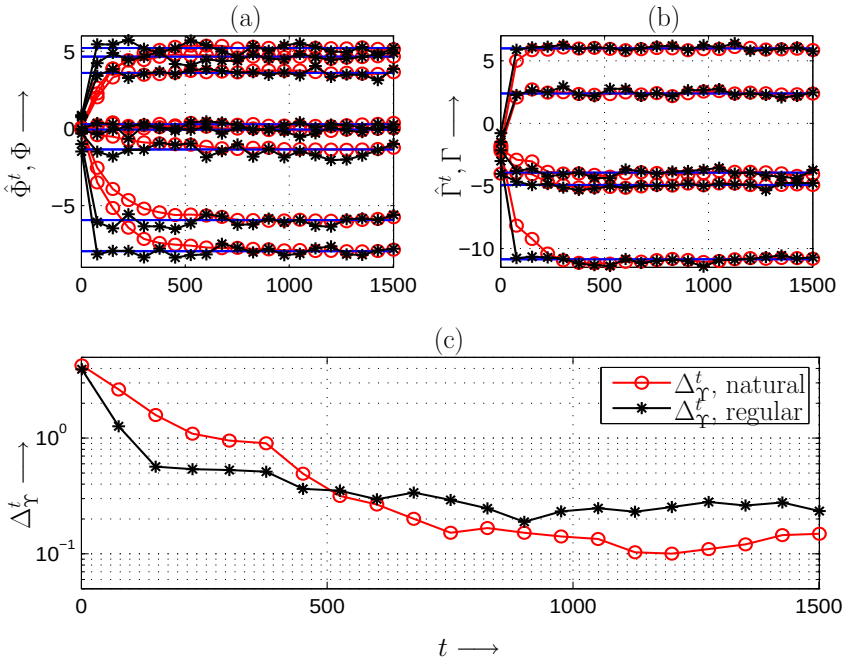

Figure 6.4: An example of acquisition of a static radio environment. (a) Estimated parameter $\hat{\Phi}^t$ with a natural gradient recursion (circle markers), estimated parameter $\hat{\Phi}^t$ with a regular gradient recursion (star markers), and true parameter Φ (no markers). (b) Estimated parameter $\hat{\Gamma}^t$ with natural gradient (circle), estimated parameter $\hat{\Gamma}^t$ with regular gradient (stars) and true parameter Γ (no markers). (c) Δ^t_Υ, the total parameter estimation error, along with the corresponding Cramer-Rao lower bound estimate given by equation (6.20)

We have chosen the system parameters $N = 9$ and $\sigma_W = 2.0$. The sub-plots (a) and (b) show the trajectories of the components of estimated parameters $\hat{\Phi}$ and $\hat{\Gamma}$ respectively. The lines with circle markers indicate a natural gradient algorithm ($F^{-1} = F_\Upsilon^{-1}$) while lines with star markers indicate a regular gradient algorithm ($F^{-1} = \|F_\Upsilon^{-1}\|\, I$). The true values are also shown (as solid lines) for reference. (In the case of $\Gamma, \hat{\Gamma}$, for clarity we have shown only the components in the first row; i.e. the gains from the first cell to all the motes.). We see that this 'step-response' has rapid acquisition (within 500 samples). With natural gradient all components tend to converge together, while with regular gradient the speeds are more varied. Similarly, sub-plot (c) shows the absolute

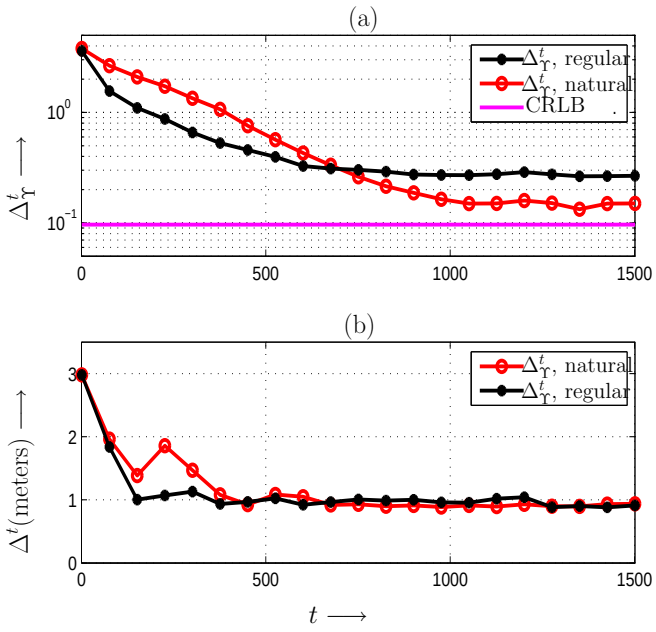

Figure 6.5: Acquisition and tracking performance averaged over ten independent experiments. (a) Δ^t_Υ, the RMS estimation error in Υ. (b) Δ^t, the normalized RMS tracking error in the positions of the targets, in meters.

error in the total estimated parameter, $\Delta^t_\Upsilon = \|\hat{\Upsilon}^t - \Upsilon^t\|$. We see that the MSE remains uniformly small after acquisition is complete (note that the ordinate scale is logarithmic).

In Figure 6.5, we provide the average acquisition performance where the averaging is done over ten experiments. In each experiment a new random static environment is chosen, and a new measurement noise process is simulated. Sub-plot (a) shows the RMS error in the parameter estimate, and the optimistic estimate of the CRLB resulting from equation (6.20), while sub-plot (b) shows the RMS tracking error (in meters) of the particle filter as it uses the parameter values provided by the estimator. We see that we practically achieve the estimate of the CRLB when a natural gradient is used, indicating that our estimator is close to being efficient. A small loss is suffered w.r.t. the estimated CRLB when a regular gradient is used. Similarly, we see that initially, when the radio

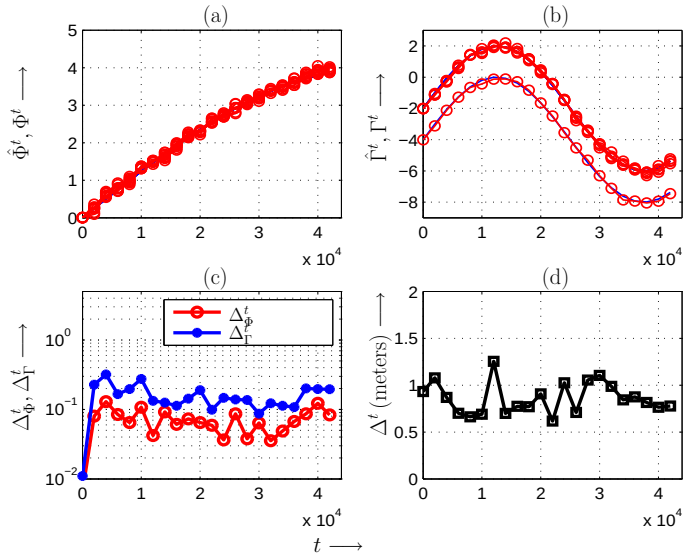

Figure 6.6: An example of tracking a time-varying radio environment, with a natural gradient recursion algorithm. (a) Estimated parameter $\hat{\Phi}^t$ (lines with circle markers), and the true parameter Φ^t (solid lines). (b) Estimated parameter $\hat{\Gamma}^t$ (lines with circle markers) and the true parameter Γ (solid lines). (c) The parameter estimation errors $\Delta^t_\Phi, \Delta^t_\Gamma$. (d) The normalized RMS tracking error in the positions of the targets Δ^t.

environment is not yet acquired, the average tracking error is quite large (of the order of four meters), but after acquisition is completed it drops to a fraction of a meter. These results thus clearly demonstrate the utility of using the radio environment estimator in tandem with the particle filter − without such an estimator the tracking error would be unacceptably large. It is interesting to note that while the steady state parameter estimation error is larger in the case of regular gradient as compared to natural gradient by a factor of two, there is only a minor loss in the tracking accuracy in meters.

Finally, in Figure 6.6, we illustrate an example of tracking a time-varying radio environment, as described in Section 6.6.1. We use the natural gradient, and again choose the system parameters $N = 9$ and $\sigma_W = 2.0$. The trajectories of the true parameters (solid lines) and the estimated parameter (lines with markers) are displayed in sub-plots (a) and (b) respectively for Φ and Γ. Sub-plot (c) shows the absolute

parameter estimation error and sub-plot (d) shows the target tracking error (in meters) during the same time interval. We see that the estimated parameter agrees well with the true parameter, yielding a target tracking error comparable to the experiments done with a static environment (Figure 6.5).

Remark: Note that in all our synthetic simulations, a *blind* acquisition of the radio parameter was successfully achieved in every case. Our *real life experiments based on Zig-Bee sensor networks* suggest that in very 'hard' practical radio environments (indoor areas with numerous occlusions and reflective surfaces), a short training sequence may sometimes be needed. As long as a sufficiently rich set of locations are covered in the tracking area, the estimator can come up with a good coarse estimate of the radio parameter. This estimate is now used as a starting point for the fully blind mode of operation, and the on-line estimator then accurately converges to the true parameter with high probability.

6.6.2 Simulations of Distributed Implementation

In this section we will present simulation results for RSS based tracking of *co-dependently* moving targets with the distributed implementation of the cooperative particle filter of Sections 6.3, as described in Section 6.5. For simplicity, we will assume that the radio parameters are perfectly known to the tracking algorithm. We simulate $M = 16$ targets moving co-dependently in a large square tracking region of dimensions 100×100 meters, within which $N = 121$ motes are deployed on a square grid. We choose $\epsilon_{depen} = 0.25$, $\sigma_U^2 = -20$ dB, and $\sigma_{thresh}^2 = -10$ dB. As before, we use a modest number of particles, $\Pi = 128$, an interaction feedback matrix[2] $C = \texttt{KRON}(\texttt{KRON}(\mathbf{1}_{M\times M} - MI_M, I_D), [1, 0])$, and the non-linearity $g(s) = \tanh(\sinh(x)/100)$, which is characterized by saturation to ± 1.0

[2] `KRON` is the standard ©MATLAB function for the Kronecker tensor product. We choose to use this notation because the direct enumeration of the matrix C is not feasible due to lack of space.

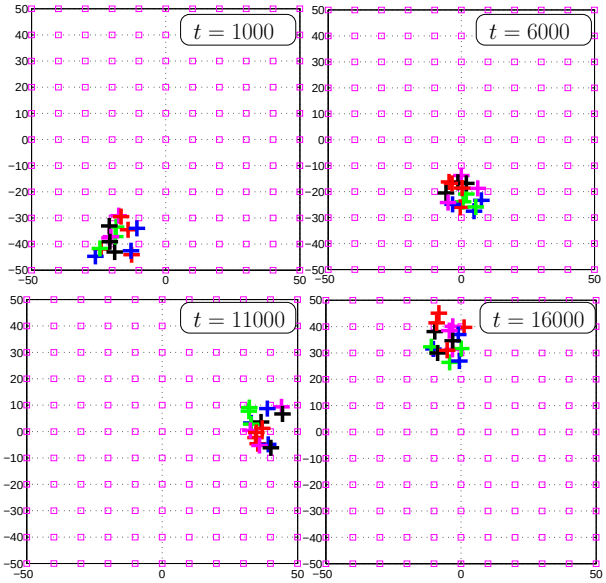

Figure 6.7: Herd motion of $M = 16$ targets governed by maneuver model of Section 6.2.1 with selection of parameters as in Section 6.6.2.

for large inputs, and a 'dead zone' for small inputs. These choices mimic the flocking rules postulated by [189], and result in a type of 'herd' motion, whose snapshots are displayed in Figure 6.7 at various time epochs. Note that all the members of the herd tend to be grouped in a region of 20×20 meters, while the herd as a whole moves over all parts of the tracking region.

In Figure 6.8, we first illustrate the notion of localized tracking. For this experiment, we choose the tracking neighborhood radius $\varrho = 20$ meters and noise deviation $\sigma_W = 3.0$. The filter update for target m at time t uses a modified interaction feedback matrix $F = \texttt{KRON}(\texttt{KRON}(\mathbf{1}_{M\times M} - \alpha^t M\, I_M, I_D), [1, 0])$, where α^t denotes the fraction of the substates that were available at time t from the tracking neighborhood $\eta_\rho^t(m)$. For clarity we only display the path of target number $m = 1$, though all $M = 16$ targets have been being simulated. Also shown are the estimated path and the error vectors at various points in time. The inset graph shows a zoomed out picture with the following elements

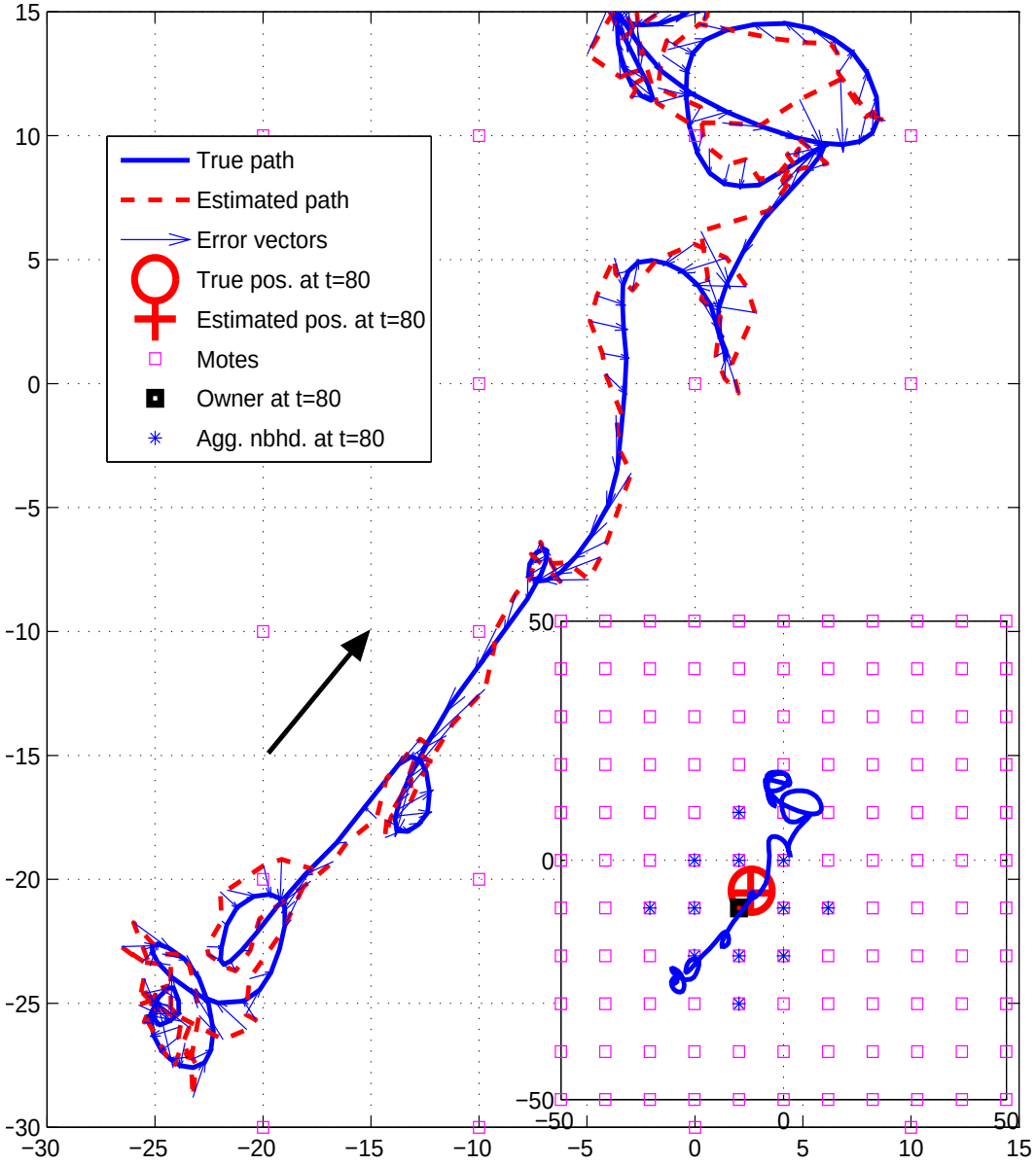

Figure 6.8: An example of target tracking with noise deviation $\sigma_W = 3.0$. Big figure shows the true path of target $m = 1$, and its estimate (note that all $M = 16$ targets were simulated). The arrow shows the direction of travel. The inset shows an illustration of target ownership and tracking neighborhood at epoch $t = 80$, with $\varrho = 20$ meters.

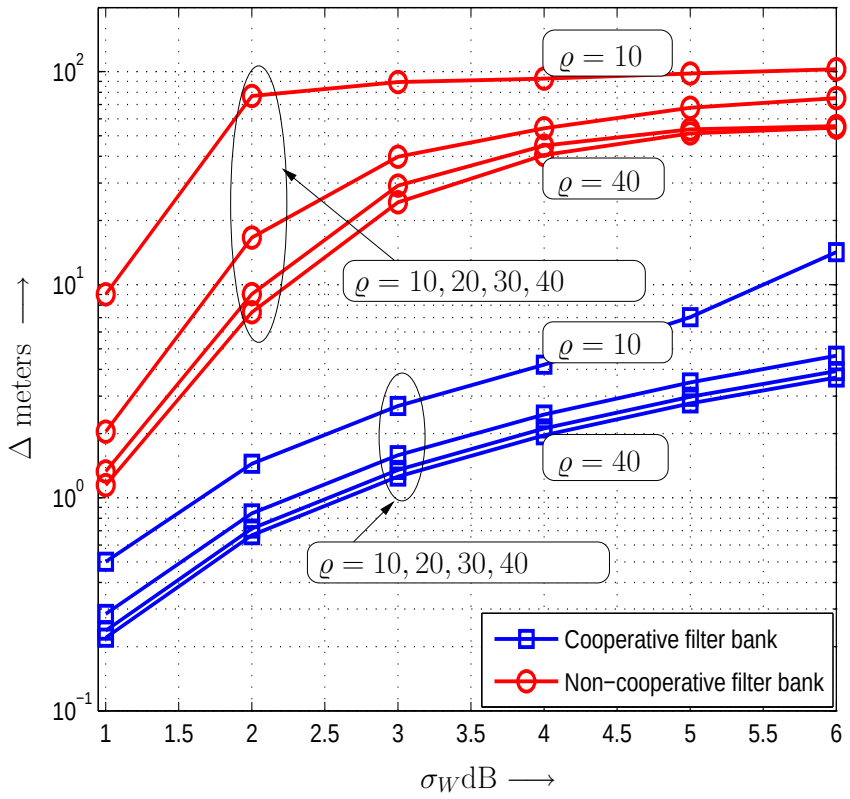

Figure 6.9: RMS tracking error Δ (meters) as a function of measurement noise deviation σ_W (dB), of a cooperative distributed filter bank and a non-cooperative distributed filter bank, for various values of the aggregation radius $\varrho = 10, 20, 30, 40$ meters.

at an arbitrary time $t = 80$: the true position of the target X_m^t (circle), the estimated position $\hat{X}_m^t$ (plus), the tracking neighborhood $\eta_\varrho^t(m)$ (stars) and the owner mote $\Omega^t(m)$ (square). Notice that the estimated position is close to the true position at all times, and hence the tracking neighborhood is clustered around the true position, as predicted in Section 6.5.2.

Finally, in Figure 6.9 we come to the crucial observation of these simulations. We consider the RMS tracking error Δ (mean is over all $M = 16$ targets) as a function of σ_W, and let ϱ be a parameter. We simulate the cooperative filter bank proposed in this paper, as well as a 'non-cooperative' filter bank as a benchmark. By non-cooperative,

we mean that although the targets move with dependencies, the filter bank presumes independent motion and hence uses $\epsilon_{depen} = 0$, thus implementing an independent filter for each target. Our main interest is in estimating (i) what aggregation radius ϱ is sufficient to extract almost all the tracking accuracy possible, and (ii) how much 'gain' our cooperative filter bank gives over the non-cooperative scheme. Both these questions are answered in Figure 6.9 (note that the ordinate axis is logarithmic). Firstly, we see that a very modest value of $\varrho = 20$ meters is sufficient to give most of the tracking accuracy. (Recall from Figure 6.8 that this corresponds to a tracking neighborhood of size $|\eta^t_\varrho(m)| \approx 13$ motes.) Secondly, we see that there is a huge gain in tracking accuracy due to the use of the cooperative filter bank, ranging from a factor of 8.0 to 20.0. Furthermore, this gain is enhanced for large values of measurement noise, which is exactly the regime of interest for low cost WSN tracking systems. (The saturation of the curves in the non-cooperative case at high noise levels is simply an artifact of the finite square tracking area used in the simulation.) The results thus clearly demonstrate the utility of the tracking technique proposed in this paper as compared to extant multi-target tracking methods.

6.7 Conclusions

We have demonstrated a new adaptive algorithm for localizing and tracking multiple co-dependently moving targets in indoor environments, using a wireless sensor network. The algorithm exploits the temporal and spatial dependencies in the motion of the targets via a distributed tractable particle filter, and exhibits good tracking accuracy and stability. Also, by using an incremental estimator for the radio environment, the algorithm achieves a high level of robustness to effects like shadowing and occlusions that are commonly found in indoor environments, as well as insensitivity to transmit power

de-calibration. This allows the proposed signal-strength based tracking technique to become a competitive alternative to more expensive techniques that rely on time difference or angle of arrival of the radio signal. It was further demonstrated that the proposed algorithm allows the system to scale seamlessly (increase the tracking area and/or the number of targets tracked) without causing network congestion or reduced battery lifetimes, which are especially important considerations for wireless sensor networks.

7 Ultra-Wide-Band Impulse Radio

7.1 Introduction

In Chapters 3-6 we have proposed distributed scalable broadcast message-passing algorithms to implement the tasks of filtering, compression, and model identification, with the ultimate goal being a reduction of the rate of out-of-network data transport. Now we will consider in some detail a practical power-efficient physical layer mechanism for implementing these message passing schemes, based on UWB-IR.

First recall that the messages transmitted in message passing algorithms are relatively small chunks of information (like conditional site expectations, conditional site decisions, local parity checks etc), typically ranging from 1 to 32 bits. This fact is not materially affected even if each message also needs to be appended with some header information for MAC housekeeping. Moreover, after a mote broadcasts a message, typically it needs to wait for the arrival of messages from it neighbors before it can compute a new message for broadcast. This immediately implies that a considerable amount of time passes between two successive broadcasts from any mote, i.e. the communication is *bursty with a low duty cycle*.

In the current generation of WSNs, the physical layer signaling scheme used for the communication of message packets is typically based on unlicensed narrow-band protocol stacks like Zig-Bee/IEEE 802.15.4-2003 [95]), which are quite inexpensive and

well-understood. However, as we discussed in Section 1.3.2, UWB radio [97][92][91] is widely regarded to be a more promising candidate for power-constrained applications like Wireless Sensor Networks (WSN) [1], on account of its ability to trade bandwidth for a reduced transmit power, its ability to coexist with extant licensed narrow-band systems, and its localized nature which is ideal for short-haul multi-hop transport. Impulse radio, in particular, is especially well suited to WSNs due to its low cost, immunity to severe multi-path fading even in indoor environments [91], and potential to provide accurate localization [113].

Notwithstanding these advantages, a UWB-IR physical layer has not been widely adopted due to the relative difficulty of implementing a coherent UWB-IR receiver for WSNs, where the transmitter in each mote periodically makes short bursts of transmissions as described earlier, and goes into a sleep mode in the relatively long inter-burst intervals to save power. In a burst, a small payload is modulated either in the amplitude or temporal position of very narrow ($\sim$ 1.0 nanosecond) IR pulses transmitted at a pulsing rate (baud rate) f_{baud}. In indoor settings this radio signal typically encounters a channel having tens or even hundreds of resolved multi-path components and a large temporal dispersion of $10 - 100$ nanoseconds [98]. Although a coherent all-digital receiver that implements maximum likelihood sequence estimation (MLSE) would be optimal in terms of the bit error rate (BER), it is impractically complex when there is heavy Inter-Symbol Interference (ISI), and furthermore requires an expensive and power-hungry high-speed analog-to-digital (ADC) converter on account of the large bandwidth [91]. On the other hand, analog equalization of the channel is also a formidable challenge, and results in a significant signal-to-noise ratio (SNR) penalty relative to MLSE. Consequently, a pragmatic solution often used [95] is to avoid ISI all-together by using a sufficiently low baud-rate $f_{baud} \ll 2\Omega = f_{nyquist}$, where Ω is the signal bandwidth. The MLSE then simplifies to a matched filter (MF), which can be implemented entirely in

the analog domain (no ADC) in the form of a maximum ratio combining (MRC) rake [91]. Of course, the choice of a low baud-rate translates to a low instantaneous data rate, longer channel occupancy, and a reduced number of supported transmitters, implying the promised trade-off between power and bandwidth (see equation (1.1)) cannot be fully exploited.

Even if one avoids high-speed ADC and MLSE complexity by using a small f_{baud} and a rake receiver, one still needs an accurate up-to-date estimate of the channel impulse response, and a timing synchronization that is correct to within a small fraction of the pulse width T_{pulse}. The problem of UWB channel estimation has been investigated in [104] under the assumption of Nyquist rate sampling, and in [103] based on a Compressed Sensing approach. Although the problem of timing synchronization is, in principle, subsumed in the problem of channel estimation, the variations in the time of arrival due to the drift of the transmitter's baud clock and the motion of the transmitter/receiver,[1] are *fast* relative to the changes in the physical environment. Such rapid changes in timing cannot be tracked by the channel estimator and hence there is essentially no timing information available from one burst to the next. Hence timing acquisition has to be done *afresh* for each burst, via techniques like correlation, serial search [91] or 'dirty template' [108], and to achieve this we need to modulate a *sufficiently long sequence of training bits* as a preamble to each burst before we modulate the comparatively small set of information carrying bits. This is highly wasteful of power and undermines the very rationale of using UWB-IR.

Alternative non-coherent approaches suggested in literature include energy detecting (ED) receivers [101], transmit-reference (TR) receivers [99] [193], and differential transmit-reference (DTR) receivers [100], all of which in principle need neither the chan-

[1]Small scale relative motion only alters the over-all time of arrival, while leaving the shape of channel response invariant.

nel response nor accurate timing synchronization [102]. However it is quite difficult to ensure robust operation of such non-coherent schemes in the regime of significant ISI [194], and hence they too usually remain restricted to low baud rates. Moreover, ED receivers suffer a very large SNR penalty relative to coherent systems, as do TR and DTR to a lesser extent. TR/DTR also involve the use of a very long analog delay line (of $1/f_{baud}$ seconds), which is difficult to implement with the requisite accuracy.

In this chapter we offer a solution that combines the advantages of MLSE coherent receivers (high system gain, high baud rate, ability to operate in ISI) and non-coherent receivers (low complexity, robustness to timing uncertainty and ignorance about the channel response), while avoiding their respective drawbacks. We propose a flexible and robust receiver architecture that performs a 'joint' decoding of the timing and amplitude information. This joint decoding is inspired by the principle of compressed sensing (CS) proposed by [50] [53]. The uncertainty in the arrival time of each burst is treated as 'sparsity' in the classical sense of [50] [53] and therefore tackled automatically in the reconstruction process which is a variation of the L_1-minimization used by [50] [53]. Furthermore, the fact that the amplitudes are antipodal $\{+1, -1\}$, and hence the overall transmit signal belongs to a relatively small discrete set rather than being a generic real-valued ultrawide-band signal, is also exploited by the reconstruction process. As a result, the receiver architecture completely bypasses the requirement of high-rate ADC conversion. Instead we use an analog front-end consisting of a bank of correlators with tractable test functions (like square waves), a low-rate ADC, and a DSP back-end that utilizes the knowledge of the channel response. The number of correlators can be significantly smaller than the requirement suggested by the Shannon-Nyquist sampling theorem, and nevertheless the performance degrades gracefully with such sub-Nyquist

sampling.[2] The work-horse of the DSP back-end is a computationally efficient quadratic program (QP). The proposed receiver works robustly even in significant ISI, and hence we are not restricted to a low baud-rate. At the same time, the complexity of the receiver is far smaller than a full-fledged MLSE. Moreover, we do not rely on long analog delay lines or any specific modulation format as in TR/DTR. The same architecture can operate with various levels of timing accuracy, ranging from a fraction of T_{pulse} to many multiples of T_{pulse}, and in each case a performance close to the MLSE receiver is attained. Furthermore, as the burst size becomes moderately large, the receiver implicitly acquires perfect timing 'on the fly' and hence the penalty associated with timing uncertainty becomes negligible. Therefore we can send bursts *without* training headers, and yet attain a power efficiency comparable to genie-aided timing. Finally, although the DSP back-end needs to know the channel response, it can blindly acquire and track it based on the same observations that are available for bit demodulation. Unlike [103], who use a matching-pursuit reconstruction and exploit the sparsity of the *received* signal, our channel estimator uses a maximum likelihood stochastic approximation that exploits the much more significant sparsity and cardinality properties of the *transmitted* signal.

CS has been used previously by [195] for mitigation of narrow band interference. Similarly it has been used for DOA estimation in MIMO radar by [196]. In an approach analogous to ours, [197][198] and [109] have used CS for direct detection of IR pulses without using a rake or a digital correlator. However note that [197][198] have formulated a generalized likelihood ratio test (GLRT) for the detection of a *single* bit in an ISI-free regime, and they presume accurate timing while doing so. Similarly, although [109] do explicitly address the timing problem, their proposal also assumes an ISI-free regime and involves the exact solution of a set of linear equations that are often ill-conditioned.

[2]The Shannon-Nyquist theorem only provides a *sufficient* condition on the sampling rate for the *perfect reconstruction* (in the sense of the L_2 norm) of bandlimited functions or random processes.

Outline of the chapter: In Section 7.2 we describe the system model and the architecture of the proposed receiver. In Section 7.3 we first formulate and analyze the maximum likelihood (ML) receiver (which is typically intractable), and then propose signal demodulation via a significantly simpler suboptimal QP optimization. Section 7.4 presents a stochastic recursive algorithm based on ML principles for identifying the channel response. Section 7.5 presents extensive simulations of the proposed receiver. In Section 7.6 we consider the issue of co-existence of UWB systems in the presence of primary licensed narrow-band interferers, and in particular how our receiver can be made very robust to such interference. Section 7.7 presents concluding remarks.

7.2 System Model and Receiver Architecture

In this section we will describe the overall UWB-IR system under consideration, and then present the architecture of our receiver. The reader is advised to refer to Figure 7.1 on page 177.

Transmitter

The UWB-IR transmitter consists of three main blocks, namely, a timing block that generates a clock signal at a nominal frequency f_{baud}, a payload block that supplies the information bits, and an IR pulse generator. The baud clock provides the timing for the IR pulses within each burst, as well as the timing for the start of each burst after requisite down-sampling. A total of K pulses are transmitted in each burst after which the transmitter hibernates till the start of the next burst.[3] At the k-th strobe of the

[3]Our receiver architecture continues to be applicable without modification even in the scenario where a repetition code is used, that is, one information bit is repeated N_f times in the payload $\{B_k\}$. The effect of the repetition code is simply to improve the BER vs signal to noise ratio (SNR) characteristic by a factor $10\log_{10}(N_f)$ dB, at the cost of a $\frac{1}{N_f}$ rate reduction. Unless otherwise stated we will assume that no repetition code is present ($N_f = 1$).

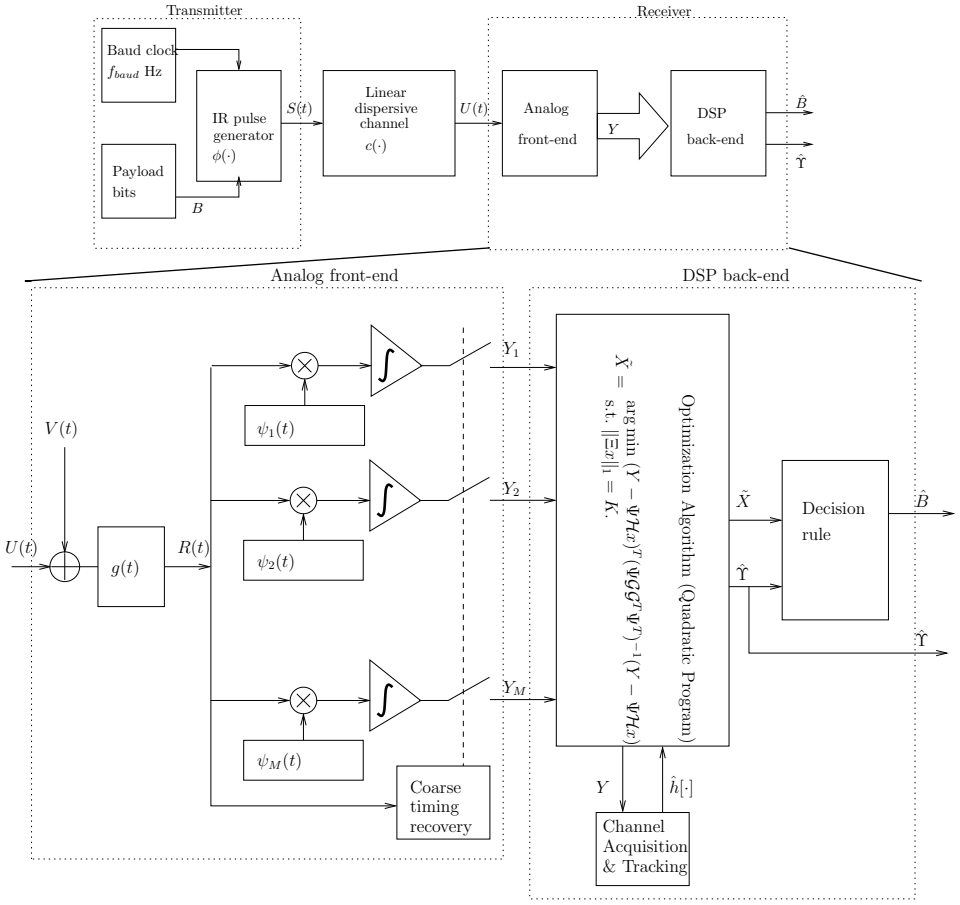

Figure 7.1: Block diagram of the UWB-IR system.

clock within a burst, the IR pulse generator sends on the air a pulse $\phi(t)$, amplitude modulated[4] by the bit B^k provided by the payload, drawn equiprobably from $\{+1, -1\}$. The pulse $\phi(t)$ is nominally centered at the frequency f_c with a bandwidth Ω. There is no other RF processing at the transmitter, like heterodyning or filtering, which makes this transmitter very simple, small and inexpensive to build.

For example, consider Figure 7.2 on page 178 which displays the Hanning modulated RF pulse of [92] which we used in our simulations, with a center frequency $f_c = 4.0$ GHz and a 6-dB bandwidth $\Omega = 2.0$ GHz. The pulse duration is small, $T_{pulse} = 1.0$

[4]A generalization to pulse position modulation (PPM) is straightforward and will not be discussed here.

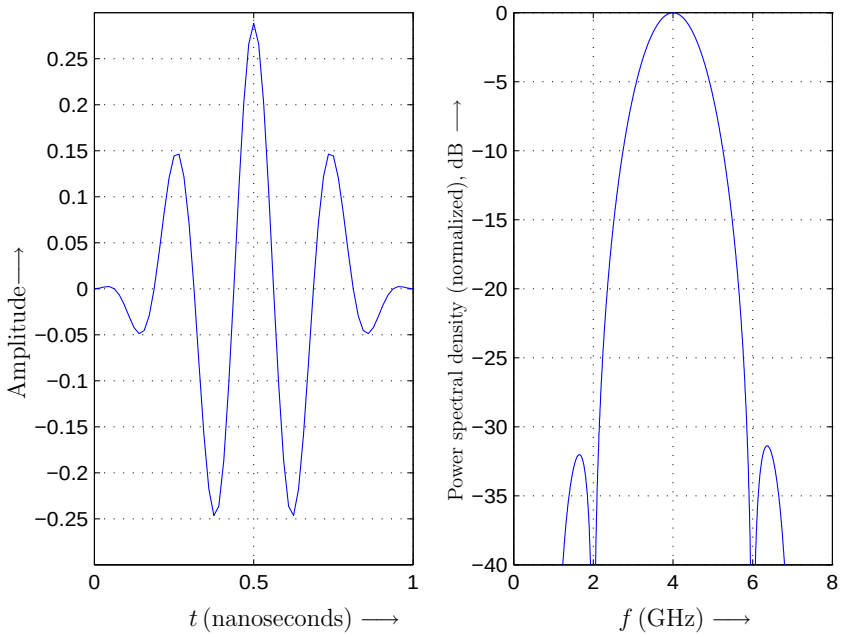

Figure 7.2: Impulse Radio pulse shape $\phi(t)$, and its power spectrum.

nanosecond. It is well-known that the maximum possible ISI-free baud-rate over an ideal channel of bandwidth $\Omega = 2.0$ GHz is $f_{nyquist} = 2\Omega = 4.0$ GBaud. However, since the temporal dispersion of the UWB channel in indoor environments is often as large as $\tau_{chan} = 100$ nanoseconds, a conventional UWB-IR system needs to choose a much smaller baud-rate, $f_{baud} \leq \frac{1}{\tau_{chan}} = 10$ Mbaud, to avoid ISI. Our receiver, on the other hand, can tolerate significant ISI and therefore we may choose a baud-rate close to the Nyquist frequency, say $f_{baud} = \frac{f_{nyquist}}{8} = 500$ Mbaud. Hence the interval between consecutive pulses is $T_{baud} = \frac{1}{f_{baud}} = 2.0$ nanoseconds, and a burst of $K = 64$ bits will therefore last for 127 nanoseconds. In contrast, the interval between consecutive bursts may be as large as $T_{burst} = 100$ microseconds. Since a practical inexpensive clock has a significant timing drift of $\rho \sim 40$ parts per million (p.p.m.) caused by random frequency modulation [95, 107], the total drift from the beginning to the end of a burst is limited to $K\rho f_{baud} = 5.1$ picoseconds, which is negligible considering the fact that a timing error

of up to 40 picoseconds causes an SNR penalty of no more than 1.0 dB for coherent demodulation. Thus if exact timing synchronization is available at the start of a burst, there is no further timing problem. On the other hand, the drift from one burst to the next is very large, $\sim$ 4.0 nanoseconds. Even with a coarse timing algorithm for predicting the start of the bursts, like a second-order tracking loop, a residual tracking error of the order of 1.0 nanosecond is unavoidable. A timing error of this magnitude results in a catastrophic loss of performance in rake receivers, and hence long headers of training bits are needed to bring the timing error down to tens of picoseconds. In contrast, we will demonstrate that our receiver is robust to such large timing uncertainty and does not require explicit headers.

Without loss of generality we can concentrate on the reception of a single burst, and treat the estimated epoch of arrival of that burst as the temporal origin, $t = 0$. The residual error of the coarse timing block is then perceived as a late arrival of the actual burst by an amount υ seconds. (By prefixing a sufficient guard interval in the coarse timing estimate, we can ensure that $\upsilon \geq 0$ with high probability, i.e. the true arrival can only be late but never early.) For simplicity suppose that the true arrival time υ is distributed over the interval $[0, \gamma]$ according to a uniform density. From the point of view of the receiver, the output of the transmitter during the burst is then written as

$$S(t) = \sum_{k=0}^{K-1} B^k \, \phi(t - k\, T_{baud} - \upsilon). \tag{7.1}$$

Notice that in writing this equation we ignore the timing drift within a burst, since we demonstrated earlier that it is negligible in the case of a high f_{baud}. It is noteworthy, however, that our setup also subsumes the case of a low baud rate $f_{baud} \leq \frac{1}{\tau_{chan}}$ (used to avoid ISI, as in [95]), if we choose $K = 1$ (one pulse in each burst), treat the pulse-to-pulse drift as the burst-to-burst drift υ, and demodulate each pulse independently. With

no ISI among the pulses, there is clearly no loss of optimality in such a formulation.

7.2.1 Channel

The UWB channel is known to be linear dispersive with tens or hundred of resolved multi-path components, depending on the radio environment. In [98], a set of standardized random models has been postulated covering several scenarios like indoor line-of-sight (LOS) in residential environments (CM1), indoor non-line-of-sight (NLOS) in residential environments (CM2), indoor LOS in office environments (CM3), indoor NLOS in office environments (CM4), outdoor LOS in farm environments (CM5), and so on. We will use realizations from these standardized models in our simulations. We will particularly be interested in the indoor models CM1-CM4, which are the dominant arena of application for WSNs.

The demodulation algorithm to be presented in Section 7.3 assumes that the total system response is known. Of course, apart from the random time of arrival υ, the shape of the channel impulse response (the set of multi-path amplitudes and relative delays) itself can vary with time because of the large-scale motion of the transmitter/receiver as well as random changes in the radio environment (log-normal shadowing). However, these variations are relatively slow (i.e. Doppler spread is small, channel is *under-spread*) and we can assume [91, 96] that in the duration of one burst the shape of the channel impulse response is a time-invariant function $c(t)$. (For the time being we will absorb any gain factor associated with the path-loss due to geometric spreading into $c(t)$. Later when investigating effects of narrow-band interference, we will decouple the path-loss from $c(t)$ and consider it explicitly.) In fact the channel coherence time is typically of the order of hundreds of milliseconds, which allows us to use an incremental estimator to *acquire and track* the shape of the channel response (cf. Section 7.4). We would once

again like to emphasize that the fast variations in the time of arrival due to the drift of the baud-clock and the relative small-scale motion between the transmitter and receiver will *not* be treated as channel variations, but instead be inferred explicitly from burst-to-burst and provided explicitly to the incremental channel estimator. The relatively slow dynamics of the estimator will integrate out the effects of the occasional error in the estimated time of arrival.

7.2.2 Receiver

The receiver consists of an analog front-end and a DSP back-end. The defining characteristic of our receiver is that we relieve the analog front-end of difficult tasks like fast ADC conversion and accurate delay lines, and instead compensate by using an elaborate DSP back-end. We keep the DSP back-end tractable by avoiding a full-fledged ML demodulator, and instead use a QP reconstruction. QP is considered an 'easy' problem in optimization theory, that can be solved in low-order polynomial time [199] by state-of-the-art interior point (IP) methods. A the same time, we will demonstrate that it gives negligible degradation relative to the ML decoder.

Analog Front-end

Let the received signal at the antenna be denoted by $U(t)$. The first block in the analog front-end is a noise-limiting bandpass-pass filter $g(\cdot)$ centered at f_c, having a noise equivalent bandwidth [46] $\approx \Omega$. The output of this filter is

$$R(t) = \sum_{k=0}^{K-1} B^k h(t - kT_{baud} - \upsilon) + W(t), \tag{7.2}$$

where $h(t)$ denotes the total impulse response, which is the convolution of the transmit pulse $\phi(t)$, the channel $c(t)$, and the filter response $g(t)$, and

$$W(t) = \int V(t-\tau)g(\tau)d\tau \tag{7.3}$$

is band-limited zero-mean additive Gaussian noise, modeled as the response of the filter to a white Gaussian thermal noise process $V(t)$ of single-sided power spectral density N_0.

The signal $R(t)$ is fed to a bank of M parallel analog correlators, followed by M integrators. This module replaces other conventional structures like a rake receiver, a fast ADC converter for subsequent MLSE or digital correlation, an ED receiver or a TR/DTR receiver. The test function used in correlator number m is denoted as $\psi_m(t)$, and the whole ensemble of test functions is denoted by $\{\psi_m(t)\}$. In Section 7.3.4, we will discuss the criteria for selecting the ensemble. At this point, it suffices to note that we do not need to tune the timing of these test functions (i.e. no analog delay lines), and hence they are relatively easy to implement. All we require is that the ensemble be known to the DSP back-end.

The integrators $m = 0, 1, \ldots, M-1$ are reset to zero at the epoch $t = 0$ and their output is sampled synchronously at the epoch $\lambda_h + \gamma + (K-1)T_{baud}$ when all of the energy of the burst is known to have arrived with high probability (recall that γ is the uncertainty in the time of arrival of the burst). Thus we have the *M measurements*

$$Y_m = \int_0^{\lambda_h+\gamma+(K-1)T_{baud}} R(t)\psi_m(t)\,dt, \quad m = 0, 1, \ldots, M-1. \tag{7.4}$$

The vector of measurements $Y = [Y_1, Y_2, \ldots, Y_M]^T$ is then fed to the DSP back-end, which recovers the payload bits $B^k,\ k = 0, 1 \ldots, K-1$ via a tractable QP algorithm.

Extension to the Case of Repetition Coding: Suppose an $N_f > 1$ repetition code is being used, hence payload bits $B^j,\ j = iN_f, \ldots, (i+1)N_f - 1$ are all copies of the i-th information bit C^i, and the total burst of K bits corresponds to K/N_f information bits. In this case we simply rewrite the received filtered signal as

$$R(t) = \sum_{i=0}^{K/N_f - 1} C^i h^{comp}(t - iT_{baud}^{comp} - \upsilon) + W(t), \tag{7.5}$$

where we define $h^{comp}(t)$ to be a *composite* impulse response

$$h^{comp}(t) = \sum_{j=0}^{N_f - 1} h(t - jT_{baud}). \tag{7.6}$$

Since equation (7.5) has the same mathematical form as equation (7.2), we can clearly use exactly the same DSP back-end to directly recover the information bits $C^i,\ i = 1, 2 \ldots, K/N_f$ from the measurement vector Y, by appropriately replacing $h(\cdot)$ with $h^{comp}(\cdot)$ in the reconstruction algorithm. An alternative method of exploiting the repetition code would be to continue to demodulate based on the representation in equation (7.2) and then do an algebraic decoding (hard decoding) of the repetition code via majority rule. Note that such hard decoding costs $\sim 1.0 - 2.0$ dB in SNR relative to optimal joint decoding [46].

Signal to Noise Ratio Let us use the convention that $H(f) \doteq \mathcal{F}\{h(t)\}$, $\Phi(f) \doteq \mathcal{F}\{\phi(t)\}$ etc. Let $h_U(t) \doteq \phi(t) \star c(t)$, and define

$$h_U^l(t) \doteq \sum_{k=0}^{K-1} b_l^k h_U(t - kT_{baud}), \tag{7.7}$$

$$\xi(f) \doteq \frac{1}{2^K} \sum_{l=0}^{2^K - 1} |H_U^l(f)|^2, \tag{7.8}$$

where $b_l^k \in \{+1, -1\}$ is the k-th bit of the number $l \in \{0, 1, \ldots, 2^K - 1\}$. An optimal (but intractable) receiver would replace the front-end filter $g(t)$ with a bank of 2^K matched filters (MFs), one each for the candidate matched signal $h_U^l(-t), l = 0, 1, \ldots, 2^K - 1$. Assuming that the *timing is perfectly known*, it would then declare as the estimate of the payload, the index l of the filter which has the maximum output at the correct sampling time. Such a hypothetical genie-timed MF receiver serves as a reference with which we can compare our suboptimal receiver. The average signal to noise ratio (SNR) per bit in the MF receiver is therefore given by

$$\text{SNR}_{bit} \doteq \frac{\int \xi(f) df}{K \frac{N_0}{2}}. \tag{7.9}$$

It is not difficult to show that since the K bits in the pay-load are i.i.d. Bernoulli($\frac{1}{2}$), (hence all the candidate signals $h_U^l(t)$ are a-priori equiprobable), we have the relation

$$\xi(f) = K \|H_U(f)\|^2. \tag{7.10}$$

Hence the SNR per bit in the MF receiver is given simply by

$$\text{SNR}_{bit} \doteq \frac{\int \|H_U(f)\|^2 df}{\frac{N_0}{2}}. \tag{7.11}$$

For consistency with literature, we will use this definition of SNR in all our analysis and simulations.

DSP Back-end

The demodulation of the payload by the DSP back-end relies on a consistent discrete time representation of the signal. Let f_s be a sufficiently large *virtual sampling frequency* [197] for the received UWB-IR signal. We would like to emphasize that this is only a

'thought-experiment' construction, and no ADC conversion is done at rate f_s in actuality. Choosing an f_s as large as possible reduces aliasing and timing quantization errors. On the other hand, it also increases the size of the optimization problem, hence a suitable tradeoff must be made. For example, for the IR pulse described in Section 7.2, the choice of $f_s = 2(f_c + \frac{\Omega}{2}) = 10$ GHz practically eliminates aliasing and limits the timing quantization penalty to 1.5 dB. Let $h[n]$ denote the sampled version of the total impulse response $h(t)$ at the rate f_s samples per second, and let h denote a vector representation of $h[n]$. That is, letting $T_s \doteq \dfrac{1}{f_s}$,

$$h[n] \doteq h(nT_s), \quad n = 0, 1, \ldots, \Lambda_h - 1, \tag{7.12}$$

$$h \doteq [h[0], h[1], \ldots, h[\Lambda_h - 1]^T, \tag{7.13}$$

where $\Lambda_h = \lceil \lambda_h f_s \rceil$ is the length of the discrete-time finite impulse response $h[n]$. A similar convention will apply to other signals like $g(t), \psi_m(t), W(t)$ etc.

Let γ and T_{baud} be multiples of T_s, which can be achieved by construction. Now, expressed in rate f_s samples, the arrival time uncertainty is $\Gamma \doteq \gamma f_s$ and the baud period (the interval between consecutive pulses) is $N_{baud} = T_{baud} f_s$. Define $\Lambda_X \doteq \Gamma + N_{baud}(K - 1)$. Then the length of the total burst response including the timing uncertainty is

$$N \doteq \Lambda_h + \Lambda_X - 1. \tag{7.14}$$

Let $\Upsilon = \text{round}(\upsilon f_s)$ be the burst arrival time υ quantized to a step size of T_s. As remarked earlier, this quantization introduces an extra measurement error which is negligible provided f_s is chosen large enough. Now, the sampled version of $R(t)$ can be

written as a vector $R \in \mathbb{R}^N$ given by

$$R = \mathcal{H}X + W. \tag{7.15}$$

Here the vector $X \in \mathbb{R}^{\Lambda_X}$ is a *virtual* discrete time information signal which has all samples equal to zero except for K non-zero samples. The k-th non-zero sample, for $k = 0, 1, \ldots, K-1$, has a random amplitude B^k drawn independently and equiprobably from $\{-1, +1\}$, and has a random location $\Lambda^k = \Upsilon + kN_{baud}$. On account of the modeling assumption made in Section 7.2, it follows that $\Upsilon \sim U([0, \Gamma])$. The vector $W \in \mathbb{R}^N$ is the sampled version of the additive Gaussian noise $W(t)$, and the matrix $\mathcal{H} \in \mathbb{R}^{N \times \Lambda_X}$ is the convolutional matrix (Toeplitz form) of $h[n]$,

$$\mathcal{H} = \begin{pmatrix} h[0] & 0 & 0 & \ldots & 0 & 0 & 0 \\ h[1] & h[0] & 0 & \ldots & 0 & 0 & 0 \\ h[\Lambda_{h-1}] & h[1] & h[0] & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & h[\Lambda_{h-1}] & h[1] & h[0] \\ 0 & 0 & 0 & \ldots & 0 & h[\Lambda_{h-1}] & h[1] \\ 0 & 0 & 0 & \ldots & 0 & 0 & h[\Lambda_{h-1}] \end{pmatrix}. \tag{7.16}$$

In a similar vein we can further relate the actually sampled measurements Y at the output of the integrators to the virtual information signal X. Define the $M \times N$ *measurement matrix* Ψ to be

$$\Psi \doteq \frac{1}{f_s}\Big[\psi_0, \psi_1, \ldots \psi_{M-1}\Big]^T, \tag{7.17}$$

where, for all $i = 0, 1, \ldots, M-1$,

$$\psi_i \doteq \Big[\psi_i[0], \psi_i[1], \ldots, \psi_i[N-1]\Big]^T. \tag{7.18}$$

The sampling lemma [46] tells us that for any signals $x(t), y(t)$ band-limited to $\frac{f_s}{2}$, sampling at rate f_s leaves the inner product invariant up to a scaling factor. That is, $\int x(\tau)y(\tau)d\tau = \frac{1}{f_s}\sum_n x[n]y[n]$. Hence we can write the *measurement equation*

$$Y = \Psi R = \Psi\mathcal{H}X + \Psi W. \tag{7.19}$$

Let $B \doteq [B^0, B^1, \ldots, B^{K-1}]^T$. Then the aim of the DSP back-end is to optimally estimate B, Υ from the measurement Y, based on the relation in equation (7.19) and the a-priori statistical knowledge about B, Υ. Note that B contains the payload which is of primary interest, while the quantity Υ is a 'nuisance' parameter.[5] As we shall see in the next section, for optimal performance we need to maximize the observation likelihood jointly over the informative parameter as well as the nuisance parameter.

7.3 Bit Demodulation Based on Incomplete Measurements

The maximum likelihood (ML) demodulation of B, based on the measurement Y given by equation (7.19), will be described in Section 7.3.1. It involves the maximization of the likelihood $P(Y|B, \Upsilon)$ over all the valid values of payload B and the nuisance timing parameter Υ. Since this can be complex to implement under a large timing

[5] In PPM, Υ carries the payload, while B is deterministic. In any case, Υ will always be informative in the context of localization.

uncertainty Γ and even moderately large burst length K, in Section 7.3.2 we propose an alternative computationally efficient reconstruction via a QP. We will see in simulation results discussed in Section 7.5, that the QP reconstruction gives only a small loss compared to ML demodulation.

7.3.1 ML Demodulation and BER Analysis

Let us define the set $\mathcal{X}$ as the set of all signals $x \in \mathbb{R}^{\Lambda_X}$ that satisfy the following properties:

1. $\|x\|_0 = K$ (burst size).
2. The first nonzero sample is located at $\ell^0 \in [0, \Gamma]$. The subsequent non-zero samples are located at positions $\ell^k = \ell^0 + kN_{baud},\ \forall k = 1, 2, \ldots, K-1$ (timing).
3. The amplitudes of all the nonzero samples are from $\{-1, +1\}$ (signaling alphabet).

Clearly, $\mathcal{X}$ is the finite equiprobable alphabet of the random information signal X (cf. Section 7.2.2), of cardinality $|\mathcal{X}| = 2^K(\Gamma + 1)$, and there is a one-to-one mapping

$$\{-1, +1\}^K \times \{0, 1, \ldots, \Gamma\} \ \rightarrow \ \mathcal{X} \tag{7.20}$$

$$(B, \Upsilon) \ \mapsto \ X(B, \Upsilon). \tag{7.21}$$

Hence we can write $P(Y|B, \Upsilon) = P(Y|X)$, which implies that, without losing optimality, we may first make the ML estimate $\hat{X}$ of the information signal X, and then map it to the optimal payload estimate $\hat{B}(\hat{X})$ and time of arrival estimate $\hat{\Upsilon}(\hat{X})$.

It is easy to see that the noise term ΨW in the measurement equation (7.19) is a zero mean multivariate Gaussian random variable with a covariance matrix $\sigma^2 \Psi \mathcal{G}\mathcal{G}^T \Psi^T$, where $\mathcal{G}$ is the Toeplitz form of the front-end filter $g[n]$, analogous to the definition in

equation (7.16), and

$$\sigma^2 = \frac{N_0}{2f_s}. \tag{7.22}$$

Hence, the likelihood of a candidate signal $x \in \mathcal{X}$ conditioned on the observation Y is given, up to a normalization factor, by

$$P(Y|x) \propto \exp\left\{\frac{-1}{2\sigma^2}(Y - \Psi\mathcal{H}x)^T(\Psi\mathcal{G}\mathcal{G}^T\Psi^T)^{-1}(Y - \Psi\mathcal{H}x)\right\}. \tag{7.23}$$

Therefore, the ML demodulator declares the estimated signal as

$$\hat{X} = \underset{x\in\mathcal{X}}{\text{argmax}}\ P(Y|x) = \underset{x\in\mathcal{X}}{\text{argmin}}\ (Y - \Psi\mathcal{H}x)^T(\Psi\mathcal{G}\mathcal{G}^T\Psi^T)^{-1}(Y - \Psi\mathcal{H}x). \tag{7.24}$$

Since B and Υ are drawn equiprobably from their alphabets, they do not have informative priors. Hence the ML estimate is also the Bayesian estimate, and is optimal in terms of the error rate. Suppose that $x^0 \in \mathcal{X}$ was the true information signal, hence

$$Y = \Psi\mathcal{H}x^0 + \Psi W. \tag{7.25}$$

Let $x^1 \neq x^0$, $x^1 \in \mathcal{X}$ be some other information signal. Then, under ML demodulation, the pair-wise error probability (PEP) is given by

$$\Pr(x^0 \to x^1) \doteq \Pr\{P(x^1|Y) > P(x^0|Y)\}. \tag{7.26}$$

With some straightforward manipulation it can be shown that

$$P(x^0 \to x^1) = \mathbf{Q}\left(\frac{\sqrt{(x^0 - x^1)^T\mathcal{H}^T\Psi^T(\Psi\mathcal{G}\mathcal{G}^T\Psi^T)^{-1}\Psi\mathcal{H}(x^0 - x^1)}}{2\sigma}\right), \tag{7.27}$$

where $\mathbf{Q}(a) = \int_a^\infty \frac{1}{\sqrt{2\pi}} \exp\left\{\frac{-x^2}{2}\right\} dx$ is the area under the tail of a standard normal distribution. Since we have $|\mathcal{X}| = 2^K(\Gamma+1)$ equiprobable candidates for the selecting the transmitted signal x^0, and the pair-wise error event $x^0 \to x^1$ leads to $\|\hat{B}(x^0) - \hat{B}(x^1)\|_0$ bit errors, we can write the following union bound on the BER,

$$P_e \leq \sum_{x^1, x^0 \in \mathcal{X}} \frac{\|\hat{B}(x^0)-\hat{B}(x^1)\|_0}{K2^K(\Gamma+1)} \mathbf{Q}\left(\frac{\sqrt{(x^0-x^1)^T\mathcal{H}^T\Psi^T(\Psi\mathcal{G}\mathcal{G}^T\Psi^T)^{-1}\Psi\mathcal{H}(x^0-x^1)}}{2\sigma}\right). \tag{7.28}$$

Note that $\mathcal{H} = T_s\mathcal{G}\mathcal{H}_U$, where $\mathcal{H}_U$ is the convolutional matrix of the response $h_U[n]$. Hence, as a sanity check, notice that if

1. there is no under-sampling (i.e. $M = N$),
2. Ψ is invertible (i.e. the ensemble $\{\Psi_m\}_{m=0}^{M-1}$ are linearly independent)
3. $\mathcal{G}$ is invertible (i.e. the translations of the pulse $g(t)$ in steps of T_s are linearly independent)
4. the timing is ideal (i.e. $\Gamma = 0$),
5. there is only one bit per burst (i.e. $K = 1$),

expression (7.28) reduces to the familiar expression for a perfectly timed MF,

$$P_e = \mathbf{Q}\left(\sqrt{\frac{\int |h_U(t)|^2 dt}{\frac{N_0}{2}}}\right) = \mathbf{Q}\left(\sqrt{\mathrm{SNR}_{bit}}\right), \tag{7.29}$$

where we used the definition of the SNR per bit from equation (7.11).

7.3.2 Suboptimal Computationally Efficient Demodulation Via Quadratic Programming

Motivation For a Sub-optimal Tractable Demodulator

The ML demodulation problem in equation (7.24) clearly becomes cumbersome when the timing uncertainty Γ or the burst length K is large. Even with a dynamic program like the Viterbi algorithm [46] for MLSE, the complexity is exponential in K or the channel memory, whichever is smaller. With our exemplary choice of $f_{baud} = 500$ MBaud, the channel memory will extend to at least 50 pulses and so the complexity will scale as 2^K for K up to 50. In light of this difficulty, now we will propose an alternative suboptimal demodulation technique whose complexity is $O(K^3)$. The technique is inspired by the philosophy of CS for sparse signal reconstruction under incomplete measurements [50, 53].

QP Demodulation

Let the vector $\xi(a, \ell_1, \ell_2)$ be a positive penalty vector for the candidate information signals $x \in \mathcal{X}$. It incorporates the available timing information by giving more penalty to those locations of x where the occurrence of the non-zero samples is unlikely. That is, for all $n = 0, 1, \ldots, \Lambda_X - 1$,

$$\xi(a, \ell_1, \ell_2)[n] \doteq \begin{cases} 1.0, \ n = \ell + kN_{baud}, \ \ell \in [a + \ell_1, a + \ell_2], \ k = 0, 1, \ldots, K - 1 \\ \mho, \ \text{otherwise}, \end{cases} \tag{7.30}$$

where $\mho$ is some suitable large number like 10^3. Also define a corresponding diagonal penalty matrix as $\Xi(a, \ell_1, \ell_2) = diag(\xi(a, \ell_1, \ell_2))$.

Now consider the following relaxation of the ML demodulation problem of equation (7.24):

$$\tilde{X} = \underset{x\in\mathbb{R}^N:\|\Xi(a,\ell_1,\ell_2)\,x\|_1=K}{\operatorname{argmin}} (Y-\Psi\mathcal{H}x)^T(\Psi\mathcal{G}\mathcal{G}^T\Psi^T)^{-1}(Y-\Psi\mathcal{H}x). \tag{7.31}$$

Notice that the new constraint set $\{x\in\mathbb{R}^N : \|\Xi(a,\ell_1,\ell_2)\,x\|_1 = K\}$ is not a discrete set, but rather a continuous set of signals of adequately small L_1 norm. Therefore notice that $\mathcal{X}\subset\{x\in\mathbb{R}^N : \|\Xi(0,0,\Gamma)\,x\|_1 = K\}$.

We can further re-write the problem (7.31) in a more amenable form [50] with some manipulation. Define

$$x^+ \doteq \max(x,0) \tag{7.32}$$

$$x^- \doteq \max(-x,0) \tag{7.33}$$

$$z \doteq [x^{+T}, x^{-T}]^T. \tag{7.34}$$

Then we have the identities $x = x^+ - x^-$ and $\|x\|_1 = x^+ + x^-$. We can now rewrite the problem (7.31) as

$$\begin{aligned} \tilde{X}_n &= \tilde{Z}_n - \tilde{Z}_{n+N},\ n=0,1,2,\ldots,N, \\ \tilde{Z} &= \begin{array}{l}\min f^Tz + \frac{1}{2}z^TQz \\ z\geq 0,\ [\xi(a,\ell_1,\ell_2)^T, \xi(a,\ell_1,\ell_2)^T]z = K,\end{array} \end{aligned} \tag{7.35}$$

where

$$Q = \begin{pmatrix} \mathcal{H}^T\Psi^T(\Psi\mathcal{G}\mathcal{G}^T\Psi^T)^{-1}\Psi\mathcal{H} & -\mathcal{H}^T\Psi^T(\Psi\mathcal{G}\mathcal{G}^T\Psi^T)^{-1}\Psi\mathcal{H} \\ -\mathcal{H}^T\Psi^T(\Psi\mathcal{G}\mathcal{G}^T\Psi^T)^{-1}\Psi\mathcal{H} & \mathcal{H}^T\Psi^T(\Psi\mathcal{G}\mathcal{G}^T\Psi^T)^{-1}\Psi\mathcal{H} \end{pmatrix}, \tag{7.36}$$

$$f = [-Y^T(\Psi\mathcal{G}\mathcal{G}^T\Psi^T)^{-1}\Psi\mathcal{H}, Y^T(\Psi\mathcal{G}\mathcal{G}^T\Psi^T)^{-1}\Psi\mathcal{H}]. \tag{7.37}$$

(7.35) is now a standard QP, which has several efficient large-scale techniques of solution like active set, conjugate gradient and interior point methods, of which the last is generally regarded as the fastest [199].

We perform the demodulation in two stages. In the first stage we solve the QP in (7.35) using $\xi(a = 0, \ell_1 = 0, \ell_2 = \Gamma)$. The result of this stage, $\tilde{X}^{(1)}$, is then used to extract an estimate $\hat{\Upsilon}$ of the arrival time via correlation with the template $\xi(0,0,0)[n]$ as follows:

$$\hat{\Upsilon} = \underset{n' \in \{0,1,\ldots,\Gamma\}}{\operatorname{argmax}} \sum_n |\tilde{X}^{(1)}[n - n']|\ \xi(0,0,0)[n]. \tag{7.38}$$

We then solve the QP in (7.35) again, using $\xi(a = \hat{\Upsilon}, \ell_1 = 0, \ell_2 = 0)$. The result of this stage, $\tilde{X}^{(2)}$, is not necessarily in the set $\mathcal{X}$. Hence, we cannot consistently map it back into an estimate $\hat{B}$ for the payload. To overcome this difficulty, we must implement a further simple *decision rule*: Once $\tilde{X}^{(2)}$ has been delivered, demodulate the payload as

$$\hat{B}^k = sign(\tilde{X}^{(2)}[\hat{\Upsilon} + k\, N_{baud}]),\ k = 0, 1, \ldots, K-1. \tag{7.39}$$

In summary, in lieu of the ML demodulation problem, which involves maximization over a large discrete set $\mathcal{X}$, we have formulated a relaxed continuous QP which jointly solves for the best payload and timing *without explicitly checking each timing epoch and bit pattern individually*. Since the optimization problem size is $\Lambda_X = \Gamma + (K-1)N_{baud}$, and interior point methods can solve a QP with polynomial complexity of degree-3 [199], the demodulation complexity is now only $O(K^3)$.

7.3.3 Relation Between QP Demodulation and L_1-minimization

While we have proposed the QP reconstruction as an inexpensive suboptimal substitute for ML demodulation, it is also worthwhile to briefly discuss its relationship with the

classical CS reconstruction method based on L_1-norm minimization.

Recall that the information signal X satisfies the property $\|X\|_0 \leq K$. Actually the constraint $\|X\|_0 \leq K$ by itself allows up to K nonzero samples to be placed at *arbitrary* locations within the signal and they can have *arbitrary* amplitudes, while in reality our information signal has considerably more structure. But let us ignore the extra structure for the time being. Let $\Psi_r \doteq \Psi\mathcal{H}$, and recall that since typically $M \ll N$, the system of equalities $Y = \Psi_r X$ is highly *under-determined*, and the classical least squares approach fails badly. In the CS literature [50, 54] the problem of sparse signal reconstruction from incomplete measurements is instead formulated as a *basis pursuit*:

$$\hat{X} = \begin{array}{l} \min ||x||_1 \\ \text{s.t. } Y = \Psi_r x \end{array}, \tag{7.40}$$

which is a relaxation of the intractable L_0-minimization problem. The central tenet of CS theory is that, if $M \geq \xi \log(\Gamma_X) K$, then perfect reconstruction of X with high probability is assured via (7.40), provided an appropriate 'decoherent' measurement ensemble is used. The factor ζ is a constant that depends on the choice of the ensemble, and is called the *over-sampling* factor.

The practical advantage of formulation (7.40) is that it can be re-cast as a linear program (LP) and hence can be solved very efficiently by interior point methods. Unfortunately, (7.40) is known to be very fragile to perturbations of the measurements, and we have verified that it performs poorly in even moderate amounts of noise. In light of this problem, it has been proposed that a regularized optimization in the form of a LASSO [55], a Dantzig selector [52], or a penalty function [200] would be a better choice.

For example, the LASSO optimization is written as

$$\hat{X} = \begin{array}{l} \min \|x\|_1 \\ \text{s.t. } \|Y - \Psi_r x\|_2^2 \leq \epsilon. \end{array} \tag{7.41}$$

Unfortunately, these regularized optimizations are considerably more complex than the linear program in (7.40), and can also suffer from problems of local optima and non-convergence.

However, as remarked earlier, classic CS reconstruction as well as regularized approaches like the LASSO exploit only generic sparsity $\|X\|_0 \leq K$. In contrast, in our application we know that the signal X has *exactly* K non-zero samples and they are spaced *exactly* N_{baud} samples apart. Hence the knowledge of the timing of the first sample fixes the locations of all the other samples. *In this sense the sparsity of* X *is not* K *but just* 1. Moreover, we have the following pieces of side-information: (i) the non-zero samples are always from a known fixed alphabet rather than being generic real-valued quantities, and (ii) the measurement noise is not white and its covariance matrix is known. All these extra pieces of information mean that we can improve upon a generic LASSO type reconstruction. Specifically, we can switch the cost and the constraints of the LASSO problem of (7.41) and recast it as a QP, which was precisely what was done in Section 7.3.2. This establishes the connection of the QP reconstruction to classical CS reconstruction.

Owing to the exploitation of the side-information, the QP receiver gives a better performance than techniques like LASSO, and allows a significantly higher level of under-sampling. Additionally, QP has the important advantage of being computationally much cheaper and more stable than the LASSO. Lastly, as we shall demonstrate in Section 7.5.2, the performance of the QP receiver is *invariant* w.r.t. the number of bits per packet, K, provided the number of front-end correlators, M, scales *linearly* with K.

7.3.4 Choice of Measurement Ensemble

The choice of the measurement ensemble needs to be made in such a way that M can be kept as small as possible while achieving an acceptable performance. Moreover, the ensemble should be easy to generate practically and the demodulation should be insensitive to imperfections in signal generation.

Canonical Nyquist and other Orthonormal Ensembles

If we set $M = N$, idealize the front-end filter be a low-pass Nyquist filter of bandwidth $f_s/2$ so that $g(t) = \frac{\sin \pi t f_s}{\pi t f_s}$, and let the test functions be the canonical functions $\psi_m(t) = \delta(t - mT_s)$, $m = 0, 1, \ldots, M$, the correlator bank simply provides uniform Nyquist samples of the incoming signal i.e. a uniform ADC. The resulting samples can then be used for a digital MLSE or MF as the case may be. In fact digital MLSE/MF can, in principle, be implemented in *any* orthonormal ensemble of full rank, because the invariance of the inner product holds for any orthonormal basis. Of course, if $M < N$, the choice of ensemble does become critical. This is because if the energy of the signal happens to fall in the null-space of the ensemble with high probability, there is no hope of reconstructing the signal. The classical approach to reduced rate sampling and compression is the so called *transform method*, where we *a-priori identify* the subspace in which the signal energy is concentrated and then take projections only on basis elements that span that subspace. In our application this sparsity subspace is spanned by the signals in $\mathcal{X}$, and is of very large dimension. Moreover, the basis for this subspace *depends on the channel, the payload size and the timing uncertainty*, and hence precludes a universal receiver.

Uniformly Decoherent Ensembles

This leads us to the central question: Is it possible to devise universal ensembles that reliably reconstruct any under-sampled signal provided, provided the under-sampling is not too severe? Moreover, can they allow a graceful SNR penalty in the presence of receiver noise? The surprising answer to the first question is known to be in the affirmative, as was shown in the ground breaking work of [50, 53]. In this chapter, we show through the ML demodulator analysis of Section 7.3.1 and extensive simulations in Section 7.5, that the answer to the second question also seems to be affirmative. These 'universal' ensembles are known to be sets of randomly generated noise-like signals. One example is that of binary pseudo-noise (PN) signals that transit independently and equiprobably between levels $\{\frac{+1}{\sqrt{N}}, \frac{-1}{\sqrt{N}}\}$ at intervals of T_s seconds. It is notable that the philosophy of choosing measurement signals having a *decoherence* property is the exact antithesis of the philosophy of transform coding.

The reason why such noise-like signals perform well is that [50, 53] (i) they are *uniformly decoherent* w.r.t *any* family of sparse signals (not just those that are temporally so), and (ii) any M such signals have a full rank M with high probability, for every $M \leq N$. In other words, $\Psi\Psi^T$ is invertible with high probability. In fact, though they are not necessarily exactly orthonormal, they are asymptotically so, i.e. $\Psi\Psi^T \rightarrow I_M$ as $N \uparrow \infty$.

In order to reject the out-of-band noise we must use a non-trivial front-end band-pass filter $g(t)$, and hence the *apparent* measurement ensemble matrix becomes $\Psi\mathcal{G}$. If we idealize $g(t)$ to be an ideal band-pass Nyquist filter of bandwidth Ω, we are assured that $\mathcal{G}\mathcal{G}^T = I$. Furthermore, if we pretend that $\Psi\Psi^T = I$ also holds, we have $\Psi\mathcal{G}\mathcal{G}^T\Psi^T = I$. Hence no matter which M measurement signals we choose from the underlying ensemble, we are very likely to capture roughly a fraction $\frac{Mf_s}{2\Omega N}$ of the received signal's energy. This

implies that reliable demodulation of the UWB-IR signal is possible after paying an under-sampling penalty of roughly $10\log_{10}\frac{2\Omega N}{Mf_s}$ dB, and this penalty will (on an average) decrease monotonically and vanish as $M \uparrow N\frac{2\Omega}{f_s}$. Although in practice $\Psi\mathcal{G}\mathcal{G}^T\Psi^T$ is not exactly an identity due to a non-deal filter $g(t)$ and a finite N, we will nevertheless see in extensive simulation results in Section 7.5 that the above described robustness to under-sampling does hold in all practical settings *irrespective of the timing uncertainty, the size of payload and the amount of ISI*.

Fourier and Deterministic Square Wave Ensembles

Actually we do not need a strictly universal measurement ensemble since we know that our signal sparsity due to timing uncertainty is always in the temporal domain. It is known [54, 201] that the *Fourier* ensemble, M sinusoids of random frequencies drawn uniformly from the band $[f_c-\frac{\Omega}{2}, f_c+\frac{\Omega}{2}]$, is *maximally decoherent* with respect to such signals (the renowned Heisenberg uncertainty principle), and would be the optimal ensemble for our signals in a noiseless setting. However, since we also need to deal with noise, the optimality in terms of BER performance is not guaranteed. In fact, our simulations indicate that the Fourier ensemble performs slightly worse than the PN-ensemble, presumably because of the point-like support of the test functions in the frequency domain. However note that the Fourier ensemble (with proper windowing) may still be desirable from the point of view of robustness to narrow-band interference, an issue which which we will discuss in Section 7.6 (see also [202]). Finally, the ensemble of *square waves* of amplitude $1/\sqrt{N}$ and frequencies selected *deterministically and uniformly* from the signal band $[f_c - \frac{\Omega}{2}, f_c + \frac{\Omega}{2}]$ is also seen to perform as well as the PN ensemble. From a practical perspective the square wave ensemble is perhaps more attractive than the PN ensemble because we do not need any pseudo-random generators.

Robustness to Non-Ideal Test Functions

Another important robustness property inherent to compressed sensing is that the generated test functions do not need to have an ideal waveform. For example the PN ensemble or the square wave ensemble need not have rectangular level transitions. Imperfections like ringing and non-ideal rise time are well-tolerated, provided we know these effects in advance so that we can compensate for them by choosing an appropriately modified Ψ in the reconstruction algorithm.

7.4 Channel Identification

In the discussion so far we have assumed that the total system impulse response $h(\cdot)$ is available to the receiver. We will now describe a technique to estimate the channel response via a stochastic recursive approximation [121, 203] of the Expectation Maximization (EM) algorithm [122]. It is noteworthy that our estimator uses only the observations Y and the demodulated virtual information signal $\hat{X}(\hat{B}, \hat{\Upsilon})$, and hence does not need any extra sensing hardware. Moreover, due to its simplicity, it can be easily accommodated in the DSP back-end without any significant increase in complexity.

Let $\hat{h} \in \mathbb{R}^{\Lambda_h}$ denote the current estimate of the total channel impulse response. Let $\hat{\mathcal{H}}$ be its Toeplitz matrix representation as in equation (7.16). We will rewrite $P(Y|x)$ from equation (7.23) as $P(Y|x, h)$, to make explicit its dependence of the total channel response h. Let $\epsilon[n]$ be a suitably chosen time dependent step size, $\hat{X} = \hat{X}(\hat{B}, \hat{\Upsilon})$ the information signal estimated by the QP algorithm by demodulating the burst, and $e = [e_0, e_1, \ldots, e_{M-1}]^T$ the estimated measurement error given by

$$e \doteq Y - \Psi\hat{\mathcal{H}}\hat{X}. \tag{7.42}$$

Let $\Psi_{m,b:b+\Lambda_h-1}$ denote the elements on the m-th row of Ψ, from column b through $b+\Lambda_h-1$. With these conventions in place, we implement the following update upon the arrival of each burst:

$$\begin{aligned} \hat{h} &\longleftarrow \hat{h} + \epsilon[n] \left. \frac{\partial \log P(Y|\hat{X},h)}{\partial h} \right|_{h=\hat{h}} \\ &= \hat{h} + \epsilon[n] \frac{1}{\sigma_W^2} \sum_{m=0}^{M-1} e_m \sum_{b=0}^{\Gamma_X-1} \hat{X}_b \Psi^T_{m,b:b+\Lambda_h-1}. \end{aligned} \tag{7.43}$$

The starting point of this algorithm can be simply chosen to be an all-zero response. Notice that we are updating the response based on bits $\hat{X}$, which are demodulated under the assumption that the current estimate $\hat{h}$ is the correct one. Therefore we have a totally blind algorithm. We will demonstrate with simulations in Section 7.5.4 that, in-spite of this blindness, it acquires and tracks the total channel response very robustly. Of course we can also accommodate the case of training bits by simply replacing $\hat{X}$ by the true bits X in the recursion (7.43), which typically improves the convergence speed and steady state characteristic of the estimator. However our simulations suggest that such training bits are not necessary in typical practical scenarios. Note that while the proposed estimator has similarities to other algorithms like LMS and decision feedback equalizers [46], its innovation is based not directly on the fully sampled received signal R but on under-sampled linear functionals Y thereof, and is made on a per-burst rather than per-symbol basis. The analysis of the almost sure convergence of such stochastic EM algorithms based on averaged gradient methods has been investigated in literature [177, 178, 179], as well as in Chapter 5 of this thesis in the context of model identification of HMRFs, and will not be revisited here.

7.5 Simulations

In this section we will describe the results of simulations that investigate the performance of the proposed receiver under practical conditions. Let 'CS-ML' denote a receiver having the CS analog front-end and ML demodulation in the DSP back-end, as described in Section 7.3.1. Similarly, let 'CS-QP' denote a receiver having the CS analog front-end and a QP demodulation in the DSP back-end, as described in Section 7.3.2. The performance of CS-ML is always an achievable lower bound with which we will compare the performance of the practical CS-QP receiver. Let 'Genie-MF' denote a receiver implementing a perfectly timed matched filter in an ISI-free environment. Clearly Genie-MF performance represents an ultimate lower bound, but it is not necessarily achievable when there is ISI, timing uncertainty or under-sampling.

Our discussion is divided into four parts. First, in Section 7.5.1, we will show an example of QP reconstruction of a transmitted burst. Then, in Section 7.5.2, we will investigate the effect of incomplete measurements, timing error and burst length on the BER of CS-ML and CS-QP receivers. In Section 7.5.3 we will demonstrate the robustness of CS-ML and CS-QP receivers to channels models and their random realizations. In Sections 7.5.1, 7.5.2 and 7.5.3 the total system impulse response $h[\cdot]$ is assumed to be perfectly known. In Section 7.5.4 we will demonstrate the performance of the blind incremental algorithm that identifies the total system response.

All simulations were performed with $f_s = 10$ GHz and the IR pulse described in Section 7.2 (Figure 7.2). The baud rate is $f_{baud} = 500$ MBaud. Hence note that there is significant ISI lasting up-to $\sim 25 - 100$ symbols. N ranged from 300 to 1000 samples, depending on the channel type and realization. In all cases, the measurement ensemble used was the square wave ensemble described in Section 7.3.4. The front-end filter $g(t)$ was chosen to be an ideal bandpass Nyquist-filter response truncated to $\pm\frac{5}{\Omega}$ seconds, and

delayed by $\frac{5}{\Omega}$ seconds for causality. No repetition code was used in any of the simulations ($N_f = 1$). In the following, the quantity $\frac{Mf_s}{2\alpha\Omega N}$ will be called the *under-sampling factor*. The case $\frac{Mf_s}{2\alpha\Omega N} = 1.0$ will be called *adequate sampling*, and the case $\frac{Mf_s}{2\alpha\Omega N} < 1.0$ will be called *under-sampling*. Ideally we would set $\alpha = 1.0$, but since the pulse is not strictly band-limited we have empirically found that a fixed value $\alpha = 1.5$ ensures that $\frac{Mf_s}{2\alpha\Omega N} = 1.0$ achieves a performance indistinguishable form MLSE under Nyquist rate sampling.

7.5.1 An Example of QP Reconstruction

Consider the illustration in the panel of plots in Figure 7.3 on page 203, which shows the various signals in the processing stream of the receiver. The simulation was done under the following conditions: $\text{SNR}_{bit} = 10$ dB, a CM1 channel, $N = 599$, $\Lambda_X = 151$, $M = 363$, $\Lambda_h = 449$, $\Gamma = 10$ samples ($\gamma = 1.0$ nanosecond), $K = 8$ bits per burst. The first (top) sub-plot shows the virtual information signal $X[n]$, that has only K non-zero sample of amplitudes B, with a random arrival time in the range $0 - 10$ samples. The second sub-plot shows the net impulse response of the channel and the pulse, $\psi[n] \star c[n]$. The third sub-plot shows the noiseless signal $U[n]$ impinging on the receiver antenna after passing through the linear channel $\psi[n] \star c[n]$, while the fourth sub-plot shows the noise contaminated signal $R[n]$ after the front-end filter. The final sub-plot displays the reconstruction $\tilde{X}$ made by the QP optimization. Notice that the CM1 channel realization has a very wide temporal dispersion ~ 40 nanosecond, yet the reconstruction $\hat{X}^k$ correctly estimates the location and sign of the impulses in X. As explained in Section 7.3.2, $\tilde{X}$ is not necessarily in the set $\mathcal{X}$, and hence we must use a further hard decision rule (cf. Section 7.3.2) to declare the bit estimates $\hat{B}$.

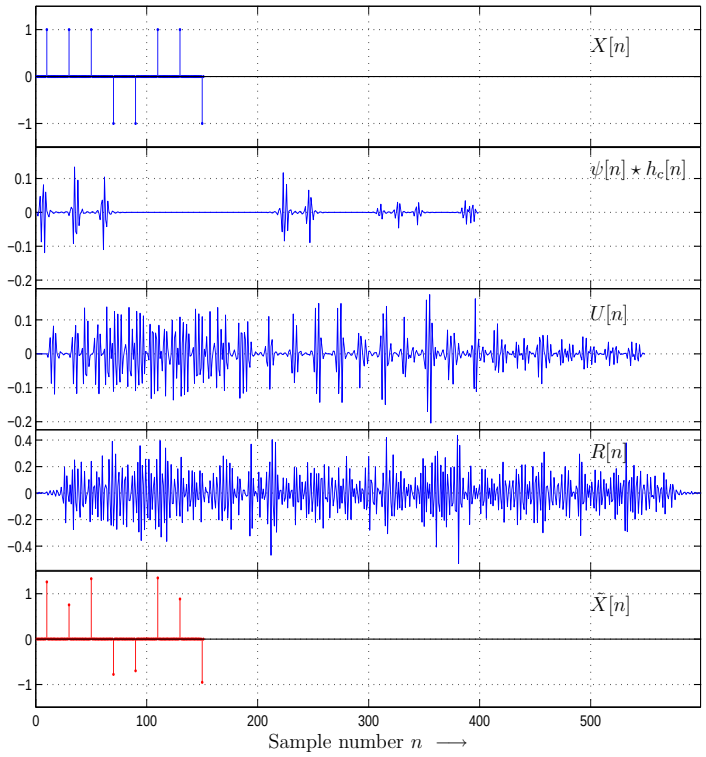

Figure 7.3: Various signals in the processing stream: the first (top) sub-plot is the virtual information signal $X[n]$, the second sub-plot is the response of the pulse and the channel, $\psi[n] \star c[n]$, the third sub-plot is the signal impinging on the antenna, $U[n]$, the fourth sub-plot is the signal after the front-end filter, $R[n]$, and the final sub-plot is the reconstructed information signal $\tilde{X}[n]$. $f_{baud} = 500$ Mbaud, $\text{SNR}_{bit} = 10$ dB, CM1 channel, $N = 599$, $\Lambda_X = 151$, $M = 363$, $\Lambda_h = 449$, $\Gamma = 10$ samples ($\gamma = 1.0$ nanoseconds), $K = 8$ bits per burst.

7.5.2 Under-Sampling, Timing Uncertainty and ISI

Now consider Figure 7.4 on page 204 which shows, again for a fixed CM1 channel, the effect of under-sampling (via M), timing uncertainty (via Γ) and the burst length K. Figures 7.4(a),(b) correspond to $\frac{Mf_s}{2\alpha\Omega N} = 1.0, 0.25$ under ideal timing $\Gamma = 0$, and Figures 7.4(c),(d) correspond to $\frac{Mf_s}{2\alpha\Omega N} = 1.0, 0.25$ under uncertain timing $\Gamma = 10$. In each sub-figure we simulate CS-QP with $K = 1, 2, 4, 8, 16$ bits per burst and plot it with dashed lines with circle markers. We plot with solid blue lines the analytical performance

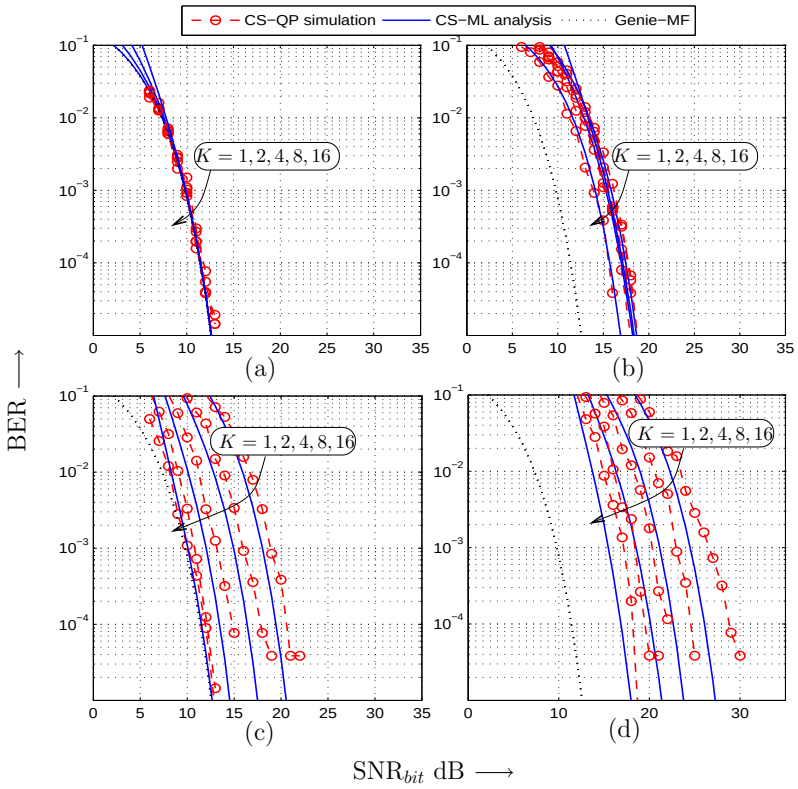

Figure 7.4: Effect of under-sampling, timing uncertainty and burst length on the receiver performance. Sub-plots (a),(b) correspond to $\frac{Mf_s}{2\alpha\Omega N} = 1.0, 0.25$ under $\Gamma = 0$, and sub-plots (c),(d) correspond to $\frac{Mf_s}{2\alpha\Omega N} = 1.0, 0.25$ under $\Gamma = 10$. In each sub-figure we simulate CS-QP with $K = 1, 2, 4, 8, 16$ bits per burst and plot it with dashed lines with circle markers. We plot with solid blue lines the analytical performance of CS-ML given by equation (7.28), for $K = 1, 2, 4, 8$. The dotted line is the Genie-MF performance in an ISI free regime.

of CS-ML given by equation (7.28), for $K = 1, 2, 4, 8$. Note that we do not give CS-ML performance for $K = 16$ because the calculation seems intractable. Finally we also plot the Genie-MF curve (dotted black line) for reference. The figure is very informative, and we can make several interesting observations:

(i) We see that with ideal timing $\Gamma = 0$ and various amounts of under-sampling in sub-plots (a),(b), the CS-QP receiver performance is very close to CS-ML, for all K. This demonstrates that we can indeed recover the performance of an ideal coherent receiver

with the proposed architecture. Furthermore with adequate sampling, all the CS-ML and CS-QP curves for various K coincide with the Genie-MF curve, implying that there is negligible loss due to the ISI. *Therefore there is no inherent justification for avoiding ISI by using a low baud rate, because it does not appreciably affect the distance spectrum of the modulation.* With under-sampling $\frac{Mf_s}{2\alpha\Omega N} = 0.25$, the curves of CS-QP and CS-ML for all K stay bunched together and have a consistent penalty of about 6.0 dB. w.r.t. the adequate sampling case, as predicted in Section 7.3.4.

(ii) Even with non-ideal timing $\Gamma = 10$, the CS-QP receiver performance is reasonably close to CS-ML, for each K respectively. The loss in performance with adequate sampling is less than one dB, while it is $1 - 2.5$ dB with under-sampling $\frac{Mf_s}{2\alpha\Omega N} = 0.25$. Note that now even in the adequate sampling case, the $K = 1$ curve of CS-ML suffers a penalty of ~ 7.0 dB w.r.t. the corresponding curve of ideal timing from sub-plot (a). While this penalty is big, it is not catastrophic like the rake receiver which suffers a loss of 20 dB or so in performance. (This can be inferred from the auto-correlation of $h(t)$). More interestingly, as the number of bits in a burst K increases, the CS-ML curves start paring the loss and approach the ideal timing curve. This makes sense heuristically, because as we have multiple bits in a burst we can acquire timing 'on the fly'. Asymptotically the timing acquisition will obviously become perfect. What is surprising is that with only $K = 8 - 16$ pulses we can practically eliminate the timing penalty.

(iii) Finally notice that in sub-plot (d), where we have both under-sampling as well as non-ideal timing, the penalty suffered by the CS-ML receiver is approximately the *additive composition* of the two individual penalties, and this is seen to consistently hold for all K. The CS-QP performance is also seen to mimic this behavior.

In summary, with CS-ML demodulation, the effects of under-sampling and timing uncertainty are approximately de-coupled. The loss due to under-sampling is consistently $10 \log_{10} \frac{Mf_s}{2\alpha\Omega N}$ dB, and unavoidable in principle. For lossless sampling we need

only $M = \frac{2\alpha\Omega}{f_s}N$ projections, rather than N samples as in direct ADC. Thus we are inherently exploiting the bandpass nature of the signal. Timing uncertainty can be combated by using sufficiently many bits per burst, and the associated penalty can thus be practically eliminated. All these observations hold, with minor caveats, for the tractable CS-QP receiver too.

7.5.3 Robustness to Stochasticity of Channel Realizations

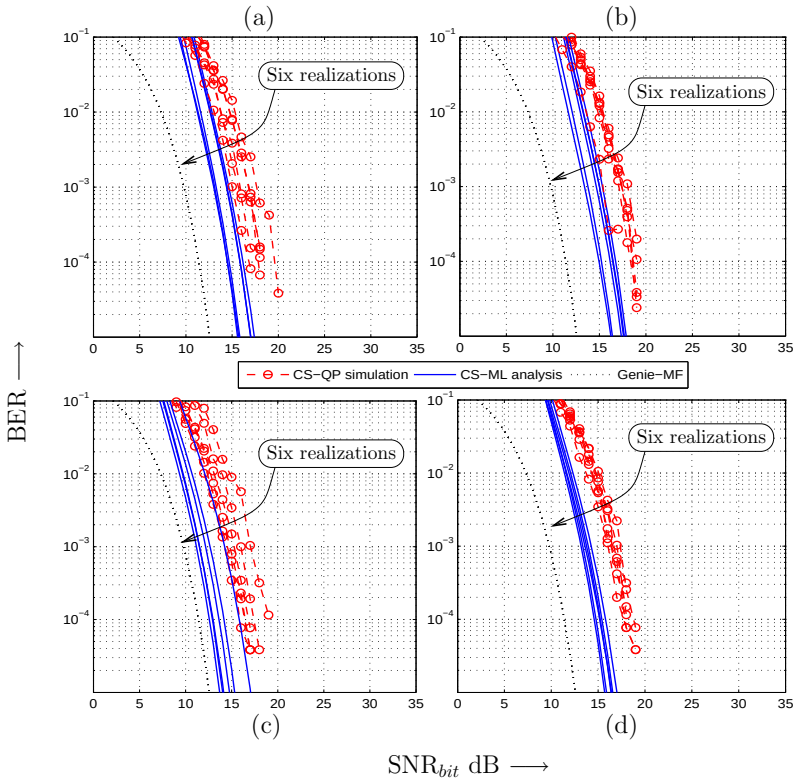

Figure 7.5: Robustness to stochastic channel realizations. Sub-plots (a) through (d) correspond to channel models CM1 through CM4 respectively. Six stochastic realizations are derived from each model. For each realization the BER vs SNR$_{bit}$ characteristic of CS-ML and CS-QP is provided. The Genie-MF curve is also shown in each sub-plot. In all cases $M = 128, \Gamma = 10$ and $K = 8$.

In the preceding discussion, we used one fixed realization from the CM1 channel

model. Now we will study the effect of the stochasticity of the channel realizations and variations in the channel models. In Figure 7.5(a)-(d) on page 206 we draw six random realizations from each model CM1 through CM4, and respectively plot the BER-vs-SNR_{bit} characteristic of the various receivers CS-QP, CS-ML and Genie-MF. In all cases we use a *constant number of projections* $M = 128$ which corresponds to a significant amount of under sampling ranging from $\frac{Mf_s}{2\alpha\Omega N} = 0.15$ to 0.25, depending of the channel model and realization. The timing uncertainty is $\Gamma = 10$ samples and the number of bits in a burst is $K = 8$.

First notice that, for every channel model, the CS-ML receiver performance does not vary by more than $2 - 3$ dB no matter what the realization of the channel is. This demonstrates the *universality* of the ensemble used in the analog front-end even in the under-sampled case. (The reader will recall that the ensemble is not tuned to any particular channel model or realization.). Obviously if we used adequate sampling (large enough M), all the curves would bunch together with no appreciable variation. Secondly, observe that similar remarks continue to apply to the CS-QP performance too. The stochastic spread in the curves is slightly more, say an additional dB, but otherwise it mimics the performance of CS-ML. The loss of CS-QP relative to CS-ML is the result of the sub-optimality of CS-QP and is in line with the loss observed in Figure 7.4(d). Finally, the Genie-MF performance obviously is invariant w.r.t. the channel model and realization, and is only an optimistic (unachievable) benchmark. We can thus conclude that the proposed receiver is indeed very robust to various channel models and the stochasticity of their realizations.

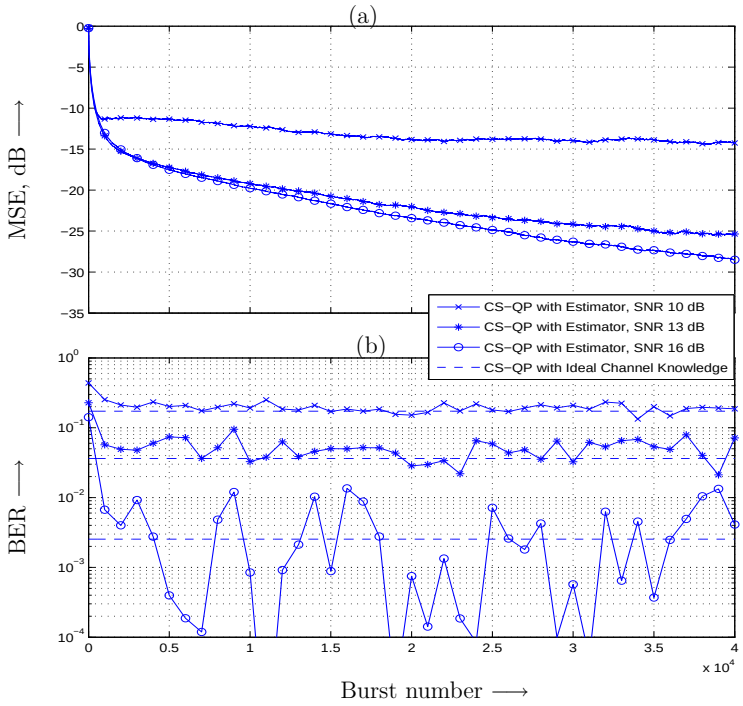

Figure 7.6: Performance of blind incremental channel acquisition starting from an all zero response. (a) Mean Squared Error (MSE), in dB, of the estimated response relative to the true response, $20\log_{10}\dfrac{\|h-\hat{h}\|_2}{\|h\|_2}$. (b) BER of the CS-QP receiver using the latest estimate of the channel, $\hat{h}$. Three values of SNR_{bit} have been simulated, namely $10, 13, 16$ dB. Horizontal red dashed lines are the corresponding BERs of the CS-QP receiver operating under ideal channel knowledge h. $M = 128, \Gamma = 10, K = 8$. The true channel realization, h, is from the CM1 model.

7.5.4 Channel Acquisition and Tracking

Finally, we will investigate the performance of the incremental channel estimator proposed in Section 7.5.4. Figure 7.6(a) on page 208 illustrates the Mean Squared Error (MSE) of the estimator under a realization from the CM1 model. Figure 7.6(b) shows the corresponding BER of the CS-QP receiver using the latest estimate of the channel. (We calculate the 'instantaneous' BER by performing a temporal averaging of the bit errors using an adequate IIR filter.) We also plot, with a dashed horizontal line, the BER

of the CS-QP receiver operating under ideal channel knowledge, which was calculated by a separate simulation.

The parameters of the simulation are exactly those used in Section 7.5.3, namely $M = 128, \Gamma = 10, K = 8$. We simulate three values of SNR_{bit} namely $10, 13, 16$ dB which are at the very low end of the operating range. (This is the most vulnerable region, where the error rates are significant even with ideal channel knowledge.) In each case we start the estimator from an all-zeros initial value $\hat{h}$. We use the step-size schedule $\epsilon[n] = \max\left(10^{-2}, \frac{10.0}{\sqrt{n}}\right),\ n = 1, 2, \ldots$, where n is the burst number. This schedule allows us to acquire the channel rapidly, and then settle into a steady state with a small MSE error floor. We would like to emphasize that we have simulated a fully blind algorithm where the bit decisions are supplied by the CS-QP demodulator of Section 7.3.2. From the MSE and BER results we conclude that the estimator acquires and tracks blindly and robustly even at low SNRs. We get qualitatively similar results with multiple realizations and with other models CM2-CM4, though we do not display them here for brevity. In every case we observe that the significant part of the acquisition is accomplished in less than a thousand bursts. The steady state MSE (which depends on the channel model and realization) is sufficiently low so that there is negligible degradation in the BER relative to the case of ideal channel knowledge. Since in steady state the step size is 10^{-2}, the estimator can track channel variations over intervals of around 100 bursts. For example if the burst frequency is 10^4 bursts/second, we can track channel variations of the order of a hundredth of a second or slower, which is adequate in most practical scenarios. Recall that we allow the time of arrival variations to be much faster, since they are not tracked by the estimator.

7.6 Co-existence With Narrow-band Systems Like WiMAX

7.6.1 Introduction

In this subsection we will study another critical robustness property of the receiver presented in Section 7.2.2 [131], namely its insensitivity to narrow-band interference (NBI) from primary licensed systems like WiMAX. We begin by noting that a generic UWB receiver needs to be kept 'wide open' in the frequency domain in order to gather all the signal energy, which makes it potentially susceptible to strong narrow-band interference from a variety of licensed and unlicensed sources. In a digital correlator or MLSE receiver the dynamic range of the front-end high-speed ADC converter can be easily saturated by the NBI. Similarly analog receivers like rake, ED and TR also suffer heavily because they have no inherent mechanism to reject the interference energy from their decision statistic. Hence some mechanism to 'notch out' the interferer needs to be implemented *in analog, before the signal enters the receiver*. Since one a-priori does not know the frequency location of the interferer, one must identify it and then tune the notch in real time, which adds to the cost and complexity.

In contrast, we will demonstrate in this section that the CS receiver of Section 7.2.2 [131] has an inherent robustness to narrow-band interference, thanks to its structural properties. Firstly, the correlator test functions used in the analog front end can be chosen to be *highly frequency selective signals* (rather than pseudo-random noise-like signals as in classical CS), without any significant loss in performance. As a consequence, an NBI can corrupt only a small fraction of all the CS measurements. Secondly, we can implement a 'digital notch' by identifying and dropping the affected measurements during reconstruction, thereby recovering essentially all the performance of the interference-free

case. Even multiple interferers are easily handled and require no hardware modifications.

Note that [195] have also considered a similar CS approach to NBI mitigation, although there also exist some significant differences. Firstly [195] require a low pulsing rate so that ISI is avoided, while our receiver can work at any pulsing rate up to the Nyquist frequency. Secondly they do not address the issue of imperfect timing, while our receiver is very robust to the same. Lastly, they use a random CS measurement matrix and hence need to explicitly identify the NBI sparsity sub-space by taking a Discrete Cosine transform. Our CS ensemble itself implements a Fourier analysis of sorts, due to which the NBI subspace is immediately apparent from the magnitude of the CS measurements.

7.6.2 System Model for Narrow-band Interference

7.6.3 Transmitter and Channel

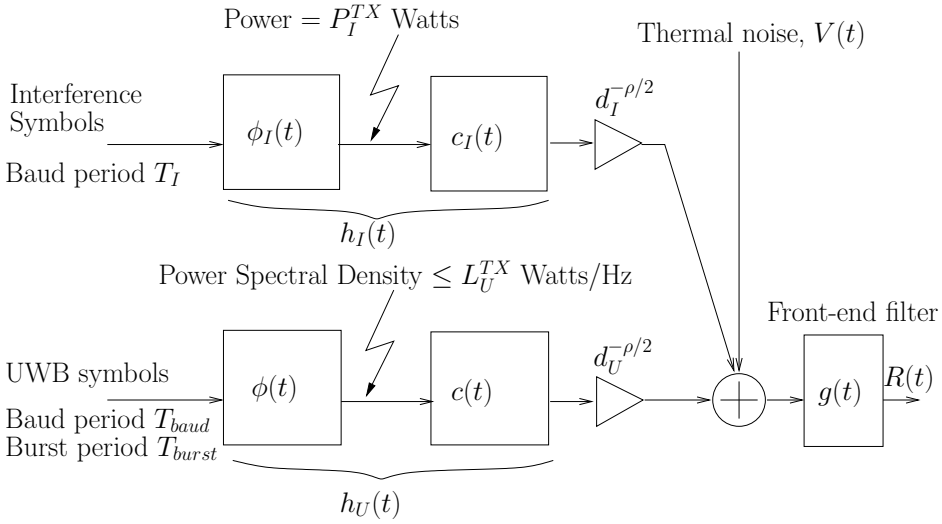

Figure 7.7: Signal paths taken by the UWB-IR and the NBI signals.

Please refer to the signal-path diagram shown in Figure 7.7 on page 211, which is an extension of Figure 7.1 with the NBI inserted into the picture, and the path loss

due to geometric spreading now explicitly shown (not clubbed into $c(t)$). Note that, for the UWB signal, the path loss of *power* is $d_U^{-\rho}$, where d_U be the distance of the UWB transmitter from the UWB receiver, and ρ is the path-loss exponent. Hence the path-loss *gain factor* is $d_U^{-\rho/2}$. Define

$$\phi^l(t) \doteq \sum_{k=0}^{K-1} b_l^k \phi(t - kT_{baud}), \tag{7.44}$$

$$\theta(f) \doteq \frac{1}{2^K} \sum_{l=0}^{2^K-1} |\Phi^l(f)|^2, \tag{7.45}$$

where $b_l^k \in \{+1, -1\}$ is the k-th bit of the number $l \in \{0, 1, \ldots, 2^K - 1\}$. Analogous to equation (7.10), we can further write

$$\theta(f) = K\|\Phi(f)\|^2. \tag{7.46}$$

Then the power spectral density (PSD) of the transmitted bursty UWB signal is given by [204]

$$PSD_U^{TX}(f) = \frac{\theta(f)}{T_{burst}} = \frac{K\|\Phi(f)\|^2}{T_{burst}}. \tag{7.47}$$

Let L_U^{TX} be the *maximum* equivalent isotropically radiated power spectral density (EIRP-SD) allowed under government regulations, and let us assume that

$$\max_f PSD_U^{TX}(f) = L_U^{TX}.$$

Interference Model

Suppose there is a narrow-band interferer (NBI) at a distance d_I from the UWB receiver. Let the pulse shape used by the NBI be $\phi_I(t)$, nominally centered at f_{cI} and having a bandwidth $\Omega_I \ll \Omega$, and let its signaling interval be T_I. Then, assuming that the NBI

uses a zero-mean unit-power signaling constellation, its EIRP is given by

$$P_I^{TX} = \frac{\int |\Phi_I(f)|^2 df}{T_I}. \tag{7.48}$$

The interferer sees a channel $c_I(t)$ to the UWB receiver, and a path-loss of $d_I^{-\rho}$.

SNR and SIR in an Optimal Matched Filter Receiver

The average signal to noise ratio (SNR) per bit of equations (7.9)(7.11) now gets re-defined as

$$\mathrm{SNR}_{bit} \doteq \frac{d_U^{-\rho} \int \xi(f) df}{K \frac{N_0}{2}} = \frac{d_U^{-\rho} \int |H_U(f)|^2 df}{\frac{N_0}{2}}. \tag{7.49}$$

Similarly define $h_I(t) \doteq \phi_I(t)$, and

$$h_{cross}^l(t) \doteq h_I(t) \star h_U^l(-t), \tag{7.50}$$

$$\chi(f) \doteq \frac{1}{2^K} \sum_{l=0}^{2^K-1} |H_{cross}^l(f)|^2 = K |H_{cross}(f)|^2. \tag{7.51}$$

Then the average signal to interference ratio (SIR) per bit is given by

$$\mathrm{SIR}_{bit} = \left(\frac{d_I}{d_U}\right)^{\rho} \frac{T_I \left(\int \xi(f) df\right)^2}{K \int \chi(f) df} = \left(\frac{d_I}{d_U}\right)^{\rho} \frac{T_I \left(\int |H_U(f)|^2 df\right)^2}{\int |H_{cross}(f)|^2 df}. \tag{7.52}$$

Note that the SIR so defined is additively compatible with the SNR in the MSE sense. That is, the net signal-to-perturbation ratio (SPR) per bit in the MF receiver is

$$\mathrm{SPR}_{bit} = \frac{1}{\frac{1}{\mathrm{SNR}_{bit}} + \frac{1}{\mathrm{SIR}_{bit}}}. \tag{7.53}$$

7.6.4 Robustness to Narrow-band Interference

Choice of Measurement Ensemble

We will now study the interference robustness of the CS-QP receiver proposed in Section 7.3.2. As we remarked earlier in Section 7.3.4, the Fourier ensemble of *sinusoids* of amplitude $1/\sqrt{N}$ and frequencies selected *deterministically and uniformly* from the signal band $[f_c - \frac{\Omega}{2}, f_c + \frac{\Omega}{2}]$ is optimally decoherent w.r.t. the UWB signal. In fact such an ensemble is highly desirable when we face strong NBI because the two are mutually *coherent*. As a consequence the NBI can affect only a few CS measurements.

Digital Notch

As we indicated above, if we choose an appropriate measurement ensemble that is coherent w.r.t. the NBI but decoherent w.r.t. the UWB signal, it is assured that the NBI can corrupt only a few CS measurements and we can therefore implement a *digital notch* to suppress those measurements. This is achieved as follows. Suppose for the time being that there can be at the most $n_I = 1$ NBIs. Let

$$\hat{m} = \underset{m \in \{0,1,\ldots,M-1\}}{\operatorname{argmax}} |Y_m|. \tag{7.54}$$

Now let D be an even number and

$$Y_{notched} = [Y_0, Y_1, \ldots, Y_{\hat{m}-\frac{D}{2}}, Y_{\hat{m}+\frac{D}{2}}, \ldots, Y_{M-2}, Y_{M-1}]^T \tag{7.55}$$

be a shortened vector obtained by notching out the $D+1$ measurements around the index $\hat{m}$. Now we simply execute the QP demodulation in equations (7.35),(7.36),(7.37) using $Y_{notched} \in \mathbb{R}^{N-D-1}$ in lieu of $Y \in \mathbb{R}^N$, along with the appropriate corresponding sub-matrix of Ψ. D can be chosen quite small, say $D \sim \beta\frac{\Omega_I}{\Delta}$, where $\Delta = \frac{\Omega}{M}$ is the

frequency spacing of the test functions and β is a safety factor to account for leakage into adjacent measurements. We found that $\beta \sim 4 - 8$ works well. Note that a smaller β can be used if we choose test functions with better frequency selectivity. Finally, if $n_I > 1$ interferers are expected, we simply apply the above notching procedure around the n_I largest absolute values in Y.

Our simulations presented in Section 7.6.4 indicate that whenever the NBI are of any significant strength (say $\text{SIR}_{bit} \leq 20$ dB), they can be very reliably detected and notched by the above method. If an interferer is very weak we may mis-detect and hence fail to suppress it. This, by itself, will have no noticeable impact (since it is weak). But will the unintended side-effect of notching out valid (uncorrupted) measurements be catastrophic? The decoherence property of the CS ensemble ensures that this is not the case. In fact, since each CS measurement on an average captures an equal fraction of the UWB signal energy, in the *interference-free regime* the performance penalty due to the notching of $(D+1)$ CS measurements, w.r.t. an un-notched matched filter receiver, is limited to a maximum of

$$-10\log_{10}\left(1 - \frac{n_I(D+1)}{M}\right) \sim -10\log_{10}\left(1 - \frac{n_I\beta\Omega_I}{\Omega}\right) \text{ dB}. \tag{7.56}$$

Of course, when strong NBIs are actually present, the digitally notched CS-QP receiver does *not* have any extra performance loss while the un-notched matched filter can be completely disabled. Thus the performance penalty in equation (7.56) is a cost we pay 'up-front' to achieve robustness against n_I interferers, whatever their actual strength (within reason). For example, if we plan for $n_I = 2$ WiMAX NBIs of bandwidth $\Omega_I = 20$ MHz, the performance penalty is only 0.75 dB, which is a very modest.

Finally, another variant is possible where, instead of deciding a-priori on the number of possible NBIs we wish to be immune against (n_I), we choose it on a burst-to-burst

basis. That is, in each burst we notch out the n_I CS measurements *whose absolute value crosses a certain pre-defined threshold*. Thus n_I is now a random variable. Therefore, we can in effect adapt the notching to the number of NBIs actually active. Note that we must set the threshold conservatively (i.e. not too high) so that we reliably detect and notch the NBIs when they are present. This means we will suffer some 'false alarms' and intermittently notch uncorrupted measurements. Hence, instead of a fixed deterministic penalty, we pay a stochastic penalty for achieving NBI robustness, and its average value will be given by the expectation of equation (7.56) w.r.t. the distribution of n_I conditioned on an interference-free regime.

Simulations of WiMAX Interference

As an exemplary case we simulate interference from a WiMAX transmitter (IEEE 802.16-2004 [205] and later). We consider the maximum allowed bandwidth under the draft standard, namely $\Omega_I = 20$ MHz, which constitutes the worst case from the point of view of the UWB system. We choose a center frequency of $f_{cI} = 4.0$ GHz (out of the possible range $2 - 66$ GHz), since it falls approximately in the middle of our UWB spectrum (cf. Section 7.6.3). We simulate the presence of WiMAX customer premise equipment (CPE) (uplink) with a standard transmit power $P_I^{TX} \sim 23$ dBm. From these results the performance in the presence of a base station (BS) (downlink) having $P_I^{TX} \sim 43$ dBm can be easily inferred, as we shall see shortly. For the UWB transmitter we choose the FCC specified PSD limit $L_U^{TX} = -41.3$ dBm/MHz [95].

We randomly generate two channel realizations from the CM1 model of [98], normalize them to unit energy (since we model the distance-based path loss separately) and specify them as the responses $c(t), c_I(t)$, which are held constant throughout. Note that the temporal dispersion of the CM1 channel can be as large as 50 to 100 nanoseconds. As before, we set $g(t)$ to be a Nyquist filter of bandwidth Ω around f_c, adequately delayed

and truncated in time for realizability. All simulations are performed with $f_s = 10$ GHz, $K = 8$ bits per burst, $\rho = 2.0$, $f_{baud} = 500$ MBaud (implying that ISI extends for $25-50$ pulses), $f_{burst} = 1$ Mbursts per second, and $T_I = \frac{1.5}{2\Omega_I} = 37.5$ nanoseconds (hence a 50% roll-off in the WiMAX modulation). We use the IR pulse described in Section 7.6.3 and the CS-QP demodulation described in Section 7.2.2. A Fourier ensemble is used, as described in Section 7.6.4, with the M test functions located uniformly from 2.5 to 5 GHz. (An asymmetric range is chosen around the center frequency $f_c = 4.0$ GHz to exploit the 'tilt' in the channel frequency response.) A Tuckey window is applied to each test function to ensure sufficiently rapid decay of its spectrum away from its center frequency, in order to minimize leakage. As before, the quantity $\frac{Mf_s}{2\alpha\Omega N}$, with $\alpha = 1.5$, will be called the *under-sampling factor*.

Figure 7.8(a) shows the PSD of the UWB and the NBI signals at the input of the filter $g(t)$ under the condition $\text{SIR}_{bit} = 25$ dB, as well as the spectra of a couple of exemplary test functions. Figures 7.8(b),(c) show the contribution of the UWB signal and the WiMAX NBI to the CS measurements when $\frac{Mf_s}{2\alpha\Omega N} = 1.0$. Sub-plot (b) shows the case when the NBI is co-located with a test function and (c) shows the case when it falls in in-between two adjacent test functions. It is clear that even for such relatively high SIR, the NBI stands out in magnitude in both cases and can be reliably detected. Note that if we operate with significant under-sampling $\frac{Mf_s}{2\alpha\Omega N} \ll 1.0$, the test functions are not densely packed and the NBI can fall 'in between the cracks'. This missed detection is not a problem however since in such a case the NBI will not affect the performance at all. In other words, whenever the NBI is in a position to degrade the receiver performance, we can also reliably detect and digitally notch it. Next we consider the performance of the digitally notched CS-QP receiver when there is a single NBI co-located with a test function, at various values of SIR_{bit}. We use a fixed value of $n_I = 1$ for the digital notch. Note that a given value of SIR_{bit} translates to a unique value of $\frac{d_I}{d_U}$ according to

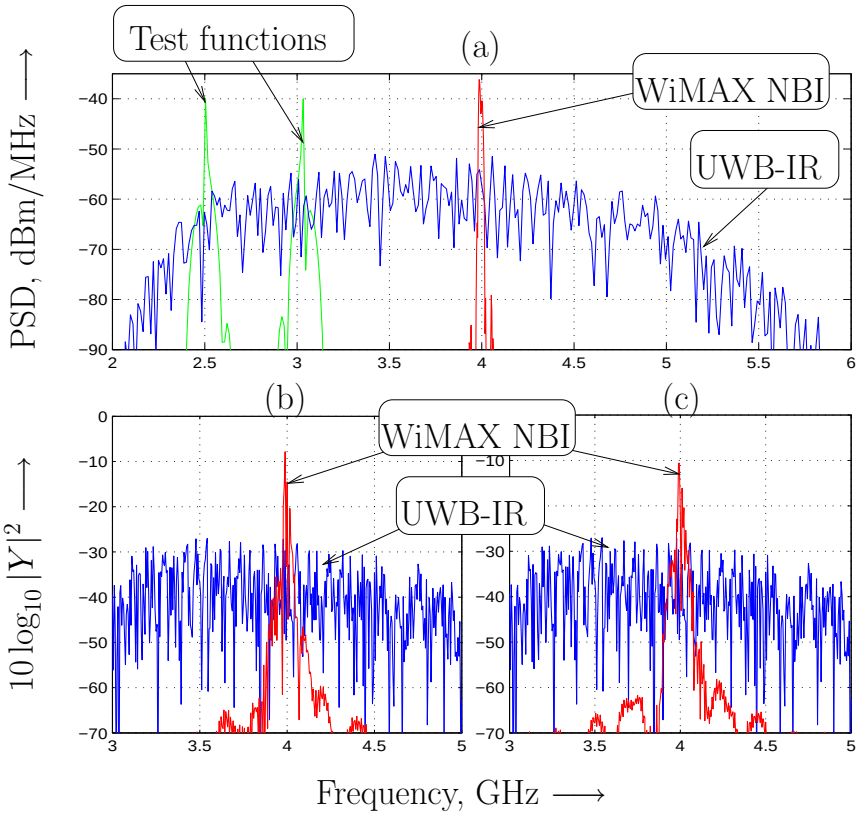

Figure 7.8: (a) Received power spectral density of NBI and UWB signals, and the spectra of test functions. (b) and (c) Contribution of UWB and NBI to CS measurements Y, respectively when the NBI falls in-between two adjacent test functions, and when it is co-located with a test function. $\text{SIR}_{bit} = 25$ dB.

equation (7.52). Some exemplary values of this relationship are provided in Table 7.6.4 for the case of a WiMAX CPE as well as BS. We simulate four cases, namely (i) Adequate sampling and perfect timing: $\frac{Mf_s}{2\alpha\Omega N} = 1.0$, $\gamma = 0$, (ii) Under-sampling and perfect timing: $\frac{Mf_s}{2\alpha\Omega N} = 0.25$, $\gamma = 0$, (iii) Adequate sampling and poor timing: $\frac{Mf_s}{2\alpha\Omega N} = 1.0$, $\gamma = 1.0$ nanosecond, and (iv) Under-sampling and poor timing: $\frac{Mf_s}{2\alpha\Omega N} = 0.25$, $\gamma = 1.0$ nanosecond. We present the results in the four respective sub-plots of Figure 7.6.4 on page 220. In sub-plot Figure 7.6.4(a) for reference we also show the performance of the matched filter receiver with perfect timing (cf. 7.6.3) but no interference rejection mechanism. The figure allows us to make several interesting observations.

SIR_{bit} dB	20	10	0	-10	-20
$\frac{d_I}{d_U}$ (WiMAX CPE)	9.9	3.1	0.9	0.3	0.09
$\frac{d_I}{d_U}$ (WiMAX BS)	99.5	31.4	9.9	3.1	0.99

Table 7.1: Relation between SIR_{bit} and d_I/d_U, for $K = 8, \rho = 2.0$.

Firstly, with perfect timing and adequate sampling (sub-plot (a)) we see that the digitally notched CS-QP receiver is 20 dB more robust to NBI than the un-notched matched filter. Specifically, an SIR_{bit} of 0 dB causes a loss of ~ 0.5 dB at BER 10^{-3} operating point, while for the matched filter an SIR_{bit} of 20 is needed for similar performance. From Table 7.6.4 we know that $\text{SIR}_{bit} = 0$ dB when $d_I/d_U \sim 0.9(9.9)$ for a WiMAX CPE (BS). This implies that the CPE can be as a distance comparable to that of the UWB transmitter, and the BS interferer need be only about ten times further away, to ensure that there is no noticeable impact on the digitally notched CS-QP UWB receiver. This is not an unreasonable scenario. In contrast, a generic un-notched UWB receiver will need the distances to be ten times larger, which essentially means that it cannot co-exist with the WiMAX system. Note that ultimately the CS-QP receiver degrades because of NBI leakage into adjacent CS measurements.

Secondly, we see that the robustness of the digitally notched CS-QP receiver to the NBI is also carried over to the cases of under sampling, imperfect timing or both. We observe, as was expected, that imperfect timing has little effect on performance, since with $K = 8$ the receiver can already acquire timing 'on the fly'. Similarly we verify that the degradation due to under-sampling is graceful and in proportion to the under-sampling factor. That is, $\frac{Mf_s}{2\alpha\Omega N} = 0.25$ leads to around 6 dB loss. In fact, we note that the NBI robustness *improves* when we have under-sampling. This is explained by the fact that with more frequency separation among the test-functions, the NBI leakage

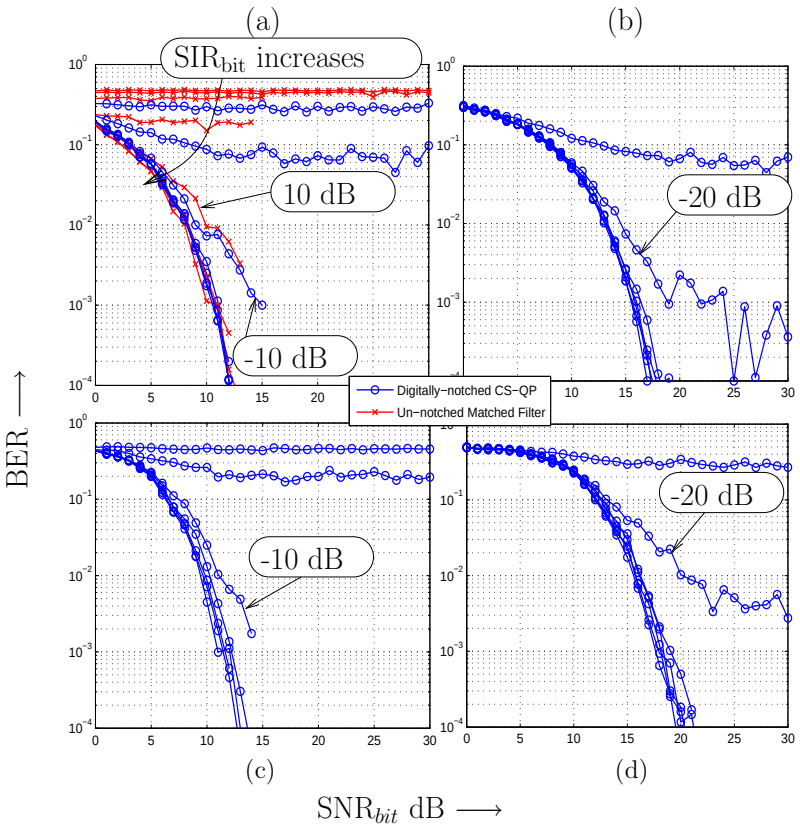

Figure 7.9: Effect of WiMAX interference on the BER vs SNR$_{bit}$ performance of a digitally notched CS-QP receiver, for various scenarios of under-sampling and timing uncertainty. (a) Adequate sampling and perfect timing (b) Under-sampling and perfect timing (c) Adequate sampling and poor timing, and (d) Under-sampling and poor timing. In sub-plot (a) we also show the performance of an un-notched genie-timed matched filter receiver. In each sub-plot the curves are parameterized by SIR$_{bit} = -30, -20, -10, 0, 10, 20, \infty$ dB.

into adjacent measurements is further reduced. Finally, we also see that the loss due to under-sampling and imperfect timing is decoupled, i.e. combines additively in dB. In summary, we have verified that all the observations made in Section 7.5.2 [131] for the interference-free case also hold for the scenario of strong NBI.

7.7 Concluding Remarks

We have proposed a novel receiver for UWB Impulse Radio transmission based on the principle of Compressed Sensing (CS). It is very robust to timing uncertainty, ISI and under-sampling, and gives a performance that is consistently close to that of an optimal (ML) receiver. It allows the use of baud-rates comparable to the Nyquist rate, and hence large network loading factors. The demodulation procedure is insensitive to the nature of the multi-path channel (CM1, CM2 etc). Finally, although the proposed receiver needs to know the channel response in performing the demodulation of the payload, it also has a built-in ability to blindly identify it based on the CS measurements. The receiver is thus ideally suited to low-power applications with bursty traffic, like wireless sensor networks.

While in this chapter we have only considered the problem of single-user UWB-IR demodulation, the proposed receiver architecture can also be used for demodulating multiple non-cooperating users. One approach can be based on the property that when the incoming signal is a mixture of 'signature' waveforms from several transmitters, out of which the QP is matched to one, the receiver reconstructs the data from the matched transmitter while treating the others as noise and hence suppressing them. While this approach is sub-optimal since it neglects the structure of the interference, it can be desirable due to its simplicity. Another approach, that can give better performance at the cost of increased receiver complexity, is the joint detection of the payloads (and the time of arrivals) of all the users. While a maximum likelihood detector would be optimal, for tractability we must consider sub-optimal approximations thereof. Our initial research indicates that a tractable joint detector can again be formulated as a quadratic program along the lines discussed in this chapter for the single user case, and we conjecture that it will give a similarly robust performance.

We have also shown that the CS-QP UWB-IR receiver can be made extremely robust to narrow-band interference by the simple expedient of (a) using frequency selective test functions in the correlators, and (b) implementing a simple digital notching mechanism wherein we identify and drop the few NBI corrupted measurements. For the exemplary case of WiMAX interference, we showed that the receiver remains practically unaffected even when the CPE is at a distance comparable to that of the UWB transmitter and the BS is only ten times farther off.

8 Summary and Conclusions

8.1 Recapitulation of the Thesis

WSNs have severely constrained lifetimes since they are powered by small batteries. The aim of this thesis was to develop communication optimization techniques to reduce the power spent on radio communication, and hence significantly extend the lifetime of the batteries. We identified two synergetic methods for achieving this goal: (i) Algorithmic data reduction (ADR), and (ii) Ultra-Wide-Band Impulse Radio (UWB-IR).

ADR aims to reduce out-of-network data transport based on intrinsic statistical properties of the field and the channel, and the particular requirements of the end-application. We considered two instances of this approach, namely filtering and compression, and proposed a distributed scalable algorithm in each case. The filtering algorithm achieves data reduction by inferring the underlying field of interest from the noisy sensor observations *before out-of-network transport*. Since typically the field alphabet is much more concise relative to the observation alphabet, this automatically yields a significant reduction in the data rate. The compression algorithm then further explicitly exploits the dependencies and a-priori biases in the field to reduce the data rate close to the ultimate limit given by the rate-distortion bound. We showed that each technique can independently reduce the power consumption by at least 10 dB in practice, and furthermore these strategies can be concatenated to get an addition of these gains (in dB). Since both

methods are statistical parametric in nature and require a knowledge of the field model, we also proposed a tandem distributed identification algorithm, that avoids the energy inefficiency of centralized identification.

Most researchers agree that UWB-IR is an ideal candidate as a power-efficient physical layer for WSNs since it can trade bandwidth for a reduced transmit power. Even by very conservative estimates, at least a 10 dB system gain can be achieved on realistic channels (with appropriate channel coding), relative to Zig-Bee type narrow-band systems that are currently being used. However there is a major problem in coping with poor synchronization and inter-symbol interference when implementing extant UWB-IR receivers in the *bursty traffic regime* that is commonly found in such networks. We solved these problems by proposing a new receiver architecture based on compressed sensing and quadratic programming demodulation, which *exploits* the timing uncertainty as a form of sparsity rather than succumbing to it. It similarly exploits the discrete alphabet of the transmit signal. As a result we need neither long headers nor low baud-rates, and our receiver can achieve a performance competitive to maximum likelihood decoding with Nyquist rate sampling, at a fraction of its cost and complexity.

By combining our ADR and UWB-IR techniques, it seems that we can reduce the power consumption of the WSN *related to radio transmissions* by at least 30 dB overall. Of course, since there are other causes of power drain like the electronics in the sensors and the microcontroller, this will not immediately translate into a 30 dB improvement in the lifetime. Nevertheless, considering the fact that radio transmissions typically dominate the power budget, it is reasonable to expect that a WSN that currently dies in a matter of days can be engineered to remain functional for several years, thus making it a viable alternative to wired sensing solutions in a variety of real world applications. As an example, we considered in detail the application of tracking targets carrying active RFID tags, based on the RSSI of their periodic transmissions. These RFID tags could

very well be inexpensive UWB-IR transmitters, in which case our UWB-IR receiver is a perfect match not only as a transceiver for inter-mote communication, but also for the *sensing* function. We showed that, even in indoor environments, a tracking accuracy of about one meter can be achieved for moderately fast-moving targets (like humans).

8.2 Future Work

There still are several open problems whose solution could be potentially of great benefit for a wider application of WSNs in real life. As we mentioned in Section 1.3.1, the general problem of universal distributed lossy source compression with variable amounts of inter-mote cooperation, tractable encoding and decoding, and an optimal use of feedback is still unsolved and has great significance for the power efficiency of WSNs. Similarly, while we have exploited the idea of rate-less codes and joint source channel decoding in the limited scenario of data aggregation from many motes to a central FC (sometimes called the *CEO problem*), their use and their ultimate bounds in a general multi-terminal scenario is very much an open problem and exemplifies the challenges of multi-user information theory. In UWB-IR, the application of our CS based reception technique to an asynchronous multi-user environment is similarly an open issue, though we have suggested some practical solutions based on PN signature sequences along the lines of code-division multiple access (CDMA), where the receiver essentially treats all multi-user interference as noise and neglects its structure (cf. Section 7.7). An optimal receiver, that jointly decodes all users, can extract a significant improvement in system gain, but unfortunately this is exponentially hard. The design of practical but efficient approximations is a major problem in multi-user detection theory [206]. With respect to tracking of RFIDs, we know that for very accurate results we would need use the TOA rather than the RSSI of the tag transmissions (cf. Section 6.1) [207]. While this is, in itself, a

straight-forward generalization, the practicality of using TOA in a hard radio environment with severe multi-path and occlusions is not trivial, and some form of cognizance of the radio environment (along the lines of our recursive estimator of Section 6.4) is probably mandatory. On the other hand, some of the other topics we have considered like distributed filtering and inference (Chapter 3) and model identification (Chapter 5) now appear to be mature topics, though one can still expect much forthcoming literature on specialized applications of these techniques to real-world problems.

8.3 Outlook for WSNs

In our opinion, the outlook for WSNs is very bright, and they are rapidly being transformed from a scientific curiosity to an indispensable *technology*. One need only look at the recent (April 2009) issue of the IEEE Spectrum magazine, to see how dramatic this change is - there are articles on applications of WSNs ranging from mobility of robotic swarms, to monitoring of aging bridges, to monitoring volcanic lightning for better prediction of eruptions (and this is not even a special issue dedicated to WSNs!). We expect that WSNs will also lead to a significant and important extension of the Internet (some have suggested the term *Sensor Net* [1]). Thus, the data viewed and manipulated by overlaid applications like the World Wide Web will no longer simply be human generated content like news, stock quotes, music, publications etc, but will also include real-world real-time data like weather, traffic, energy consumption, resource usage, pollution, localization and so on. In fact such 'sensed' content may eventually surpass human-generated content, a tipping point analogous to data traffic outstripping voice traffic on mobile wireless networks. How this will transform the Internet and electronic commerce remains to be seen, though it is quite likely that we are on the verge of a paradigm shift, from being an information society to being an ... *informed* society.

Bibliography

[1] D. Culler, D. Estrin, and M. Srivastava. Guest Editors' Introduction: Overview of Sensor Network. *Computer*, 37(8):41–49, August 2004.

[2] A. Chang. Wireless Sensors Extend Reach of Internet Into the Real World. *MSNBC (The Associated Press)*, 2007. http://www.msnbc.msn.com/id/17107484/.

[3] P. Rincon. Smart Dust to Explore Planets. *BBC News*, 2007. http://news.bbc.co.uk/go/pr/fr/-/2/hi/science/nature/6566317.stm.

[4] Brett Warneke, Matt Last, Brian Liebowitz, and Kristofer S. J. Pister. Smart Dust: Communicating with a Cubic-Millimeter Computer. *Computer*, 34(1):44–51, 2001.

[5] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc., 1991.

[6] H. Wang, Y. Yang, M. Ma, J. He, and X. Wang. Network Lifetime Maximization With Cross-Layer Design In Wireless Sensor Networks. *IEEE Trans. Wireless Commun.*, 7(10):3759–3768, October 2008.

[7] Y. Shi, Y. T. Hou, and H. D. Sherali. Cross-Layer Optimization for Data Rate Utility Problem in UWB-based Ad Hoc Networks. *IEEE Trans. Mobile Computing*, 7(6):764–777, June 2008.

[8] L. Song and D. Hatzinakos. A Cross-Layer Architecture of Wireless Sensor Networks for Target Tracking. *IEEE/ACM Trans. on Networking*, 15(1):145–158, February 2007.

[9] R. Madan, S. Cui, S. Lal, and A. Goldsmith. Cross-Layer Design for Lifetime Maximization in Interference-Limited Wireless Sensor Networks. *IEEE Trans. Wireless Commun.*, 5(11):3142–3152, November 2006.

[10] J. Misic, S. Shafi, and V. B. Misic. Cross-Layer Activity Management In An 802-15.4 Sensor Network. *IEEE Communications Magazine*, 44(1):131–136, January 2006.

[11] Yan Wu, S. Fahmy, and N. B. Shroff. Optimal QoS-aware Sleep/Wake Scheduling for Time-Synchronized Sensor Networks. *Conf. on Information Sciences and Systems*, pages 924–930, March 2006.

[12] A. Deshpande, C. Guestrin, S. R. Madden, J. M. Hellerstein, and W. Hong. Model-Based Approximate Querying in Sensor Networks. *The VLDB Journal*, 14(4):417–443, November 2005. Springer Berlin / Heidelberg.

[13] T. Wang and Q. Cheng. Collaborative Event-Region and Boundary-Region Detections in Wireless Sensor Networks. *IEEE Trans. Signal Processing*, 56(6):2547–2561, June 2008.

[14] E. Fox, J. W. Fisher, and A. S. Willsky. Detection and Localization of Material Releases With Sparse Sensor Configurations. *IEEE Trans. Signal Processing*, 55(5):1886–1898, May 2007.

[15] A. Dogandzic and B. Zhang. Distributed Estimation and Detection for Sensor Networks Using Hidden Markov Random Field Models. *IEEE Trans. Signal Processing*, 54(8):3200–3215, August 2006.

[16] V. Saligrama, M. Alanyali, and O. Savas. Distributed Detection in Sensor Networks With Packet Losses and Finite Capacity Links. *IEEE Trans. Signal Processing*, 54(11):4118–4132, November 2006.

[17] X. Nguyen, M. J. Wainwright, and M. I. Jordan. Non-parametric Decentralized Detection Using Kernel Methods. *IEEE Trans. Signal Processing*, 53(11):4053–4066, November 2005.

[18] J. Chamberland and V. V. Veeravalli. Asymptotic Results for Decentralized Detection in Power Constrained Wireless Sensor Networks. *IEEE J. Select. Areas Commun.*, 22(6):1007–1015, August 2004.

[19] R. Viswanathan and P. K. Varshney. Distributed detection with multiple sensors I. Fundamentals. *Proc. of the IEEE*, 85(1):54–63, January 1997.

[20] M. Cetin, Lei Chen, J. W. Fisher III, A. T. Ihler, R. L. Moses, M. J. Wainwright, and A. S. Willsky. Distributed Fusion in Sensor Networks. *IEEE Signal Processing Magazine*, 23(4):42–55, July 2006.

[21] S. Vijayakumaran, Y. Levinbook, and T. F. Wong. Maximum Likelihood Localization of a Diffusive Point Source Using Binary Observations. *IEEE Trans. Signal Processing*, 55(2):665–676, February 2007.

[22] T. Zhao and A. Nehorai. Distributed Sequential Bayesian Estimation of a Diffusive Source in Wireless Sensor Networks. *IEEE Trans. Signal Processing*, 55(4):1511–1524, April 2007.

[23] C. G. Lopes and A. H. Sayed. Diffusion Least-Mean Squares Over Adaptive Networks. *IEEE Int. Conf. Acoustics, Speech and Signal Processing*, 3:III–917–III–920, 2007.

[24] V. Delouille, R. Neelamani, and R. Baraniuk. Robust Distributed Estimation Using Embedded Subgraphs Algorithm. *IEEE Trans. Signal Processing*, 54(8):2998–3010, August 2006.

[25] J. Xiao, A. Ribeiro, Z. Luo, and G. B. Giannakis. Distributed Compression-Estimation Using Wireless Sensor Networks. *IEEE Signal Processing Magazine*, pages 27–41, July 2006.

[26] A. Ribeiro and G. B. Giannakis. Bandwidth-Constrained Distributed Estimation for Wireless Sensor Networks, Part I: Gaussian Case. *IEEE Trans. Signal Processing*, 54(3):1131–1143, March 2006.

[27] A. Ribeiro and G. B. Giannakis. Bandwidth-Constrained Distributed Estimation for Wireless Sensor Networks, Part II:Unknown Probability Density Function. *IEEE Trans. Signal Processing*, 54(7):2784–2796, July 2006.

[28] A. K. Das and M. Mesbahi. Distributed Linear Parameter Estimation in Sensor Networks based on Laplacian Dynamics Consensus Algorithm. *Proc. SECON*, 2:440–449, 2006.

[29] E. Sudderth, M. Wainwright, and A. Willsky. Embedded Trees: Estimation of Gaussian Processes on Graphs With Cycles. *IEEE Trans. Signal Processing*, 52(11):3136–3150, November 2004.

[30] J. B. Predd, S. B. Kulkarni, and H. V. Poor. Distributed Learning in Wireless Sensor Networks. *IEEE Signal Processing Magazine*, 23(4):56–69, July 2006.

[31] J. B. Predd, S. R. Kulkarni, and H. V. Poor. A Collaborative Training Algorithm for Distributed Learning. *IEEE Trans. Inform. Theory*, 55(4):1856–1871, April 2009.

[32] M. L. Stein. *Interpolation of Spatial Data: Some Theory for Kriging*. Springer (Springer Series in Statistics), 1999. ISBN: 978-0-387-98629-6.

[33] F. Cucker and S. Smale. On the Mathematical Foundations of Learning. *Bull. Am. Math. Soc.*, 39(1):1–49, October 2001.

[34] C. K. I. Williams. *Prediction With Gaussian Processes: From Linear Regression to Linear Prediction and Beyond*. MIT Press, 1998. Learning in graphical models: M. I. Jordan (ed), pp.599-612.

[35] S. N. Simic. A Learning Theory Approach to Sensor Networks. *Pervasive Computing*, pages 41–49, October 2003.

[36] C. Guestrin, P. Bodik, R. Thibaux, M. Paskin, and S. Madden. Distributed Regression: An Efficient Way for Modelling Sensor Network Data. *Proc. Int. Workshop on Information Processing in Sensor Networks*, pages 1–10, 2004.

[37] G. S. Kimeldorf and G. Wahba. A Correspondence Between Bayesian Estimation on Stochastic Processes and Smoothing By Splines. *The Annals of Mathematical Statistics*, 41(2):495–502, 1970.

[38] I. Kyriakides, D. Morrell, and A. Papandreou-Suppappola. Sequential Monte Carlo Methods for Tracking Multiple Targets With Deterministic and Stochastic Constraints. *IEEE Trans. Signal Processing*, 56(3):937–948, March 2008.

[39] S. Aeron, V. Saligrama, and D.A. Castaon. Efficient Sensor Management Policies for Distributed Target Tracking in Multihop Sensor Networks. *IEEE Trans. Signal Processing*, 56(6):2562–2574, June 2008.

[40] R. Rahman, M. Alanyali, and V. Saligrama. Distributed Tracking in Multihop Sensor Networks With Communication Delays. *IEEE Trans. Signal Processing*, 55(9):4656–4668, September 2007.

[41] T. Vercauteren and X. Wang. Decentralized Sigma-Point Information Filters for Target Tracking in Collaborative Sensor Networks. *IEEE Trans. Signal Processing*, 53(8):2997–3009, August 2005.

[42] X. R. Li and V. P. Jilkov. A Survey of Manuvering Target Tracking: Dynamic Models. *IEEE Trans. Aerospace and Electronic Systems*, 39(4):1333–1364, October 2003.

[43] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P. J. Nordlund. Particle Filters for Positioning, Navigation, and Tracking. *IEEE Trans. Signal Processing*, 50(2):425–437, February 2002.

[44] M. Orton and W. Fitzgerald. A Bayesian Approach to Tracking Multiple Targets Using Sensor Arrays and Particle Filters. *IEEE Trans. Signal Processing*, 50(2):216–223, February 2002.

[45] L. Frenkel and M. Feder. Recursive Expectation-Maximization (EM) Algorithms for Time-Varying Parameters With Applications to Multiple Target Tracking. *IEEE Trans. Signal Processing*, 47(2):306–320, February 1999.

[46] J. G. Proakis. *Digital Communications*. McGraw-Hill, New York, San Francisco, Toronto, London, 2001.

[47] A. Kumar, P. Ishwar, and K. Ramchandran. On distributed sampling of smooth non-bandlimited fields. *Proc. Int. Workshop on Information Processing in Sensor Networks*, pages 89–98, April 2004.

[48] P. Ishwar, A. Kumar, and K. Ramchandran. Distributed Sampling for Dense Sensor Networks: A Bit-Conservation Principle. *Proc. Int. Workshop on Information Processing in Sensor Networks*, April 2003.

[49] E. J. Candes and M. B. Wakin. An Introduction To Compressive Sampling. *IEEE Signal Processing Magazine*, 25(2):21–30, March 2008.

[50] D. L. Donoho. Compressed Sensing. *IEEE Trans. Inform. Theory*, 52(4):1289–1306, April 2006.

[51] E. J. Candes, M. J. Wakin, and S. P. Boyd. Enhancing Sparsity by Reweighted L1 Minimization. *ArXiv e-prints*, 711, November 2007.

[52] E. Candes and T. Tao. The Dantzig Selector: Statistical Estimation When p Is Much Larger Than n. *Ann. Statist.*, 35(6):2313–2351, 2007.

[53] E. J. Candes and T. Tao. Near-Optimal Signal Recovery From Random Projections: Universal Encoding Strategies? *IEEE Trans. Inform. Theory*, 52(12):5406–5425, December 2006.

[54] E. J. Candes, J. Romberg, and T. Tao. Robust Uncertainty Principles: Exact Signal Reconstruction From Highly Incomplete Frequency Information. *IEEE Trans. Inform. Theory*, 52(2):489–509, February 2006.

[55] R. Tibshirani. Regression Shrinkage and Selection via the Lasso. *J. of the Roy. Stat. Soc.*, 58 - Series B(1):267–288, 1996.

[56] C. E. Shannon. Coding Theorems for a Discrete Source with a Fidelity Criterion. *IRE Conv. Rec.*, 7:142–163, 1959.

[57] T. Berger and J. D. Gibson. Lossy Source Coding. *IEEE Trans. Inform. Theory*, 44(6):2693–2723, October 1998.

[58] R. Gray. Rate Distortion Functions for Finite-State Finite-Alphabet Markov Sources. *IEEE Trans. Inform. Theory*, 17(2):127–134, March 1971.

[59] D. A. Huffman. A Method for the Construction of Minimum-Redundancy Codes. *Proceedings of the IRE*, 40(9):1098–1101, September 1952.

[60] G. Caire, S. Shamai, A. Shokrollahi, and S. Verdu. Universal Variable-Length Data Compression of Binary Sources Using Fountain Codes. *Proc. IEEE Inf. Theory Workshop (ITW)*, pages 123–128, October 2004.

[61] D. Slepian and J. K. Wolf. Noiseless Coding of Correlated Information Sources. *IEEE Trans. Inform. Theory*, IT-19(4):471–480, July 1973.

[62] S. S. Pradhan and K. Ramchandran. Distributed Source Coding Using Syndromes (DISCUS): Design and Construction. *IEEE Trans. Inform. Theory*, IT-49(3):626–643, July 2003.

[63] R. Cristescu, B. Beferull-Lozano, and M. Vetterli. Networked Slepian-Wolf: Theory, Algorithms and Scaling Laws. *IEEE Trans. Inform. Theory*, 51(12):4057–4073, December 2005.

[64] A. D. Wyner and J. Ziv. The Rate Distortion Function for Source Coding with Side Information at the Decoder. *IEEE Trans. Inform. Theory*, IT-22(1):1–10, January 1976.

[65] R. Zamir and T. Berger. Multiterminal Source Coding with High Resolution. *IEEE Trans. Inform. Theory*, 45(1):106–117, April 1999.

[66] Z. Xiong, A. D. Liveris, and S. Cheng. Distributed Source Coding for Sensor Networks. *IEEE Signal Processing Magazine*, 21:80–94, September 2004.

[67] J. Barros, M. Tuc̈hler, and Seong Per Lee. Scalable Source/Channel Decoding for Large-Scale Sensor Networks. *IEEE Int. Conf. on Communications*, 2:881–885 Vol.2, 2004.

[68] A. G. Dimakis, P. B. Godfrey, M. J. Wainwright, and K. Ramchandran. Network Coding for Distributed Storage Systems. *Proc. IEEE Intl. Conf. on Computer Communications*, pages 2000–2008, May 2007.

[69] A. G. Dimakis, V. Prabhakaran, and K. Ramchandran. Distributed Fountain Codes for Networked Storage. *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 5:V–V, March 2006.

[70] A. G. Dimakis, V. Prabhakaran, and K. Ramchandran. Decentralized Erasure Codes for Distributed Networked Storage. *IEEE Trans. Inform. Theory*, 52(6):2809–2816, June 2006.

[71] Y. Lin, B. Liang, and B. Li. Data Persistence in Large-Scale Sensor Networks with Decentralized Fountain Codes. *Proc. IEEE Intl. Conf. on Computer Communications*, pages 1658–1666, May 2007.

[72] D. J. C. MacKay. Fountain codes. *IEE Proc. Communications*, 152(6):1062–1068, December 2005.

[73] T.J. Richardson, M. A. Shokrollahi, and R. L. Urbanke. Design of Capacity-Approaching Irregular Low-Density Parity-Check Codes. *IEEE Trans. Inform. Theory*, 47(2):619–637, Feb 2001.

[74] T. J. Richardson and R. L. Urbanke. Efficient Encoding of Low-Density Parity-Check Codes. *IEEE Trans. Inform. Theory*, 47(2):638–656, Feb 2001.

[75] T. J. Richardson and R. L. Urbanke. The Capacity of Low-Density Parity-Check Codes Under Message-Passing Decoding. *IEEE Trans. Inform. Theory*, 47(2):599–618, Feb 2001.

[76] Sae-Young Chung, G. D. Forney Jr., T. J. Richardson, and R. Urbanke. On the Design of Low-Density Parity-Check Codes Within 0.0045 dB of the Shannon Limit. *IEEE Commun. Letters*, 5(2):58–60, Feb 2001.

[77] C. Berrou. The Ten-Year-Old Turbo Codes Are Entering Into Service. *IEEE Communications Magazine*, 41(8):110–116, August 2003.

[78] C. Berrou and A. Glavieux. Near Optimum Error Correcting Coding and Decoding: Turbo-Codes. *IEEE Trans. Commun.*, 44(10):1261–1271, October 1996.

[79] C. Berrou, A. Glavieux, and P. Thitimajshima. Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-codes. *Proc. IEEE Int. Conf. Commun. (ICC)*, 2:1064–1070 vol.2, May 1993.

[80] O. Etesami and A. Shokrollahi. Raptor Codes on Binary Memoryless Symmetric Channels. *IEEE Trans. Inform. Theory*, 52(5):2033–2051, May 2006.

[81] A. Shokrollahi. Raptor Codes. *IEEE Trans. Inform. Theory*, 52(6):2551–2567, June 2006.

[82] R. G. Gallager. *Information Theory and Reliable Communication*. John Wiley & Sons, Inc., 1968.

[83] T. Cover, A. E. Gamal, and M. Salehi. Multiple Access Channels with Arbitrarily Correlated Sources. *IEEE Trans. Inform. Theory*, 26(6):648–657, November 1980.

[84] M. C. Vuran and I. F. Akyildiz. Spatial Correlation-Based Collaborative Medium Access Control in Wireless Sensor Networks. *IEEE/ACM Trans. on Networking*, 14(2):316–329, April 2006.

[85] Y. Sung, S. Misra, L. Tong, and A. Ephremides. Cooperative Routing for Distributed Detection in Large Sensor Networks. *IEEE Journal on Selected Areas in Communications*, 25(2):471–483, February 2007.

[86] G. Como, S. Yuksel, and S. Tatikonda. On the Burnashev Exponent for Markov channels. *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, pages 1871–1875, June 2007.

[87] G. Como, S. Yuksel, and S. Tatikonda. On the error exponent of Markov channels with ISI and feedback. *Proc. IEEE Inf. Theory Workshop (ITW)*, pages 184–189, Sept. 2007.

[88] A. Tchamkerten and I. E. Telatar. On the Universality of Burnashev's Error Exponent. *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, pages 1382–1385, September 2005.

[89] A. Sahai and T. Simsek. On the Variable-Delay Reliability Function of Discrete Memoryless Channels With Access to Noisy Feedback. *Proc. IEEE Inf. Theory Workshop (ITW)*, pages 336–341, October 2004.

[90] M. V. Burnashev. Data Transmission Over a Discrete Channel with Feedback: Random Transmission Time. *Probl. Inf. Transm.*, 12(4):250–265, 1976.

[91] H. Arslan, Z. N. Chen, and M Di Benedetto. *Ultra Wideband Wireless Communication*. John Wiley & Sons, Inc., 2006.

[92] S. Roy, J.R. Foerster, V.S. Somayazulu, and D.G. Leeper. Ultrawideband Radio Design: The Promise of High-Speed, Short-Range Wireless Connectivity. *Proc. of the IEEE*, 92(2):295–311, February 2004.

[93] T.S. Rappaport. *Wireless Communications*. Prentice-Hall, 2002.

[94] S. Verdu. Spectral efficiency in the wideband regime. *Information Theory, IEEE Transactions on*, 48(6):1319–1343, Jun 2002.

[95] IEEE 802.15.4a-2007. Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs); Amendment 1: Add Alternate PHYs, March 2007.

[96] M.Z. Win and R.A. Scholtz. Ultra-wide bandwidth time-hopping spread-spectrum impulse radio for wireless multiple-access communications. *IEEE Trans. Commun.*, 48(4):679–689, April 2000.

[97] M.Z. Win and R.A. Scholtz. Impulse Radio: How It Works. *IEEE Commun. Letters*, 2(2):36–38, February 1998.

[98] A. F. Molisch, D. Cassioli, C. C. Chong, S. Emami, A. Fort, B. Kannan, J. Karedal, J. Kunisch, H. G. Schantz, K. Siwiak, and M. Z. Win. A Comprehensive Standardized Model for Ultrawideband Propagation Channels. *IEEE Trans. on Antennas and Propagation*, 54(11):3151–3166, November 2006.

[99] Yi-Ling Chao and R. A. Scholtz. Ultra-wideband transmitted reference systems. *IEEE Trans. Veh. Technol.*, 54(5):1556–1569, September 2005.

[100] Yi-Ling Chao and R. A. Scholtz. Optimal and Suboptimal Receivers for Ultra-Wideband Transmitted Reference Systems. *Proc. IEEE Global Telecom. Conf. (GLOBECOM)*, 2:759–763 Vol.2, December 2003.

[101] A. A. D'Amico, U. Mengali, and E. Arias de Reyna. Energy-Detection UWB Receivers with Multiple Energy Measurements. *IEEE Trans. Wireless Commun.*, 6(7):2652–2659, July 2007.

[102] K. Witrisal, G. Leus, G.J.M. Janssen, M. Pausini, F. Troeschand Th. Zasowski, and J. Romme. Noncoherent Ultra-Wideband Systems. *IEEE Signal Processing Magazine*, 26(4), July 2009.

[103] J.L. Paredes, G. R. Arce, and Zhongmin Wang. Ultra-Wideband Compressed Sensing: Channel Estimation. *IEEE J. of Select. Topics in Signal Processing*, 1(3):383–395, October 2007.

[104] V. Lottici, A. D'Andrea, and U. Mengali. Channel Estimation for Ultra-Wideband Communications. *IEEE J. Select. Areas Commun.*, 20(9):1638–1645, December 2002.

[105] R. Miri, L. Zhou, and P. Heydari. Timing Synchronization in Impulse-Radio UWB: Trends and Challenges. *Proc. IEEE Northeast Workshop on Circuits and Systems and TAISA Conference*, pages 221–224, June 2008.

[106] H. Xu and L. Yang. Timing with Dirty Templates for Low-Resolution Digital UWB Receivers. *IEEE Trans. Wireless Commun.*, 7(1):54–59, January 2008.

[107] L. Huang, N. El Ghouti, O. Rousseaux, and B. Gyselinckx. Timing Tracking Algorithms for Impulse Radio (IR) Based Ultra Wideband (UWB) Systems. *Int. Conf. Wireless Communications, Networking and Mobile Computing*, pages 570–573, September 2007.

[108] L. Yang and G. B. Giannakis. Timing Ultra-Wideband Signals with Dirty Templates. *IEEE Trans. Commun.*, 53(11):1952–1963, November 2005.

[109] J. Kusuma, I. Maravic, and M. Vetterli. Sampling with Finite Rate of Innovation: Channel and Timing Estimation for UWB and GPS. *Proc. IEEE Int. Conf. Commun. (ICC)*, 5:3540–3544 vol.5, May 2003.

[110] A. Boukerche, H. A. B. F. Oliveira, E. F. Nakamura, and A. A. F. Loureiro. Localization Systems for Wireless Sensor Networks. *IEEE Wireless Communications Magazine*, 14(6):6–12, December 2007.

[111] C. Morelli, M. Nicoli, V. Rampa, and U. Spagnolini. Hidden Markov Models for Radio Localization in Mixed LOS/NLOS Conditions. *IEEE Trans. Signal Processing*, 55(4):1525–1542, April 2007.

[112] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero III, R. L. Moses, and N. S. Correal. Locating the Nodes: Cooperative Localization in Wireless Sensor Networks. *IEEE Signal Processing Magazine*, 22(4):54–69, July 2005.

[113] S. Gezici, Zhi Tian, G.B. Giannakis, H. Kobayashi, A.F. Molisch, H.V. Poor, and Z. Sahinoglu. Localization via ultra-wideband radios: A look at positioning aspects for future sensor networks. *IEEE Signal Processing Magazine*, 22(4):70–84, July 2005.

[114] O. E. Barndorff-Nielsen. *Information and Exponential Families in Statistical Theory*. Wiley, New York, 1978.

[115] A. Oka and L. Lampe. Energy Efficient Distributed Filtering With Wireless Sensor Networks. *IEEE Trans. Signal Processing*, 56(5):2062–2075, May 2008.

[116] A. Oka and L. Lampe. Distributed Filtering with Wireless Sensor Networks. *Proc. IEEE Global Telecom. Conf. (GLOBECOM)*, pages 843–848, December 2007.

[117] L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A Maximization Technique Occuring in the Statistical Analysis of Probabilistic Functions of Markov Chains. *Ann. Math. Statist.*, 41:164–171, August 1970.

[118] A. Oka and L. Lampe. Data Extraction from Wireless Sensor Networks Using Fountain Codes. *Proc. Intl. Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, pages 229–232, Dec. 2007.

[119] A. Oka and L. Lampe. Data Extraction From Wireless Sensor Networks Using Distributed Fountain Codes. Accepted for publication in the IEEE Trans. on Communications. Preprint available at http://www.ece.ubc.ca/˜anando/publications.htm.

[120] A. Oka and L. Lampe. Compressed Sensing of Gauss-Markov Random Fields With Wireless Sensor Networks. *Proc. IEEE Sensor Array and Multichannel Signal Processing Workshop*, pages 257–260, July 2008.

[121] R. M. Neal and G. E. Hinton. A View of the EM Algorithm That Justifies Incremental, Sparse, and Other Variants. pages 355–368, 1998.

[122] A. Dempster, N. Laird, and D. Rubin. Maximum Likelihood From Incomplete Data Via the EM Algorithm. *J. of the Roy. Stat. Soc.*, 39 (Series B):1–38, 1977.

[123] H. Cramer. *Mathematical Methods of Statistics*. Princeton Univ. Press, 1946.

[124] C. Rao. Information and the Accuracy Attainable in the Estimation of Statistical Parameters. *Bull. Calcutta Math. Soc.*, 37:81–89, 1945.

[125] A. Oka and L. Lampe. Distributed Target Tracking Using Signal Strength Measurements by a Wireless Sensor Network. Submitted to the IEEE JSAC. Preprint available at www.ece.ubc.ca/∼anando/.

[126] A. Oka and L. Lampe. Distributed Scalable Multi-Target Tracking with a Wireless Sensor Network. *Proc. IEEE Int. Conf. Commun. (ICC)*, 2009.

[127] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, 2004.

[128] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. *IEEE Trans. Signal Processing*, 50(2):174–188, February 2002.

[129] Andrieu C, A. Doucet, S. S. Singh, and V. B. Tadic. Particle methods for change detection, system identification, and control. *Proc. of the IEEE*, 92(3):423–438, Mar 2004.

[130] P.M. Djuric, Ting Lu, and M.F. Bugallo. Multiple Particle Filtering. *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 3:III–1181–III–1184, April 2007.

[131] A. Oka and L. Lampe. A Compressed Sensing Receiver for UWB Impulse Radio in Bursty Applications like Wireless Sensor Networks. In revision with Elsevier - Physical Communications (Special Issue on Advances in Ultra-Wideband Wireless Communication). Preprint available at www.ece.ubc.ca/∼anando/.

[132] A. Oka and L. Lampe. A Compressed Sensing Receiver for Bursty Communication with UWB Impulse Radio . Accepted at the IEEE Intl. Conf. on Ultra-Wideband, 2009.

[133] R. J. Perkins, N. A. Malik, and J. C. H. Fung. Cloud Dispersion Models. *Applied Scientific Research*, 51:539–545, 1993. Kluwer Academic Publishers.

[134] H. Rue and L. Held. *Gaussian Markov Random Fields: Theory and Applications*, volume 104 of *Monographs on Statistics and Applied Probability*. Chapman & Hall, London, 2005.

[135] S. Geman and D. Geman. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, November 1984.

[136] L. Younes. Estimation and Annealing for Gibbsian fields. *Annales de l'institut Henri Poincare (B) Probabilits et Statistiques*, 24(2):269–294, 1988.

[137] M. V. Ibanez and A. Simo. Parameter Estimation in Markov Random Field Image Modeling with Imperfect Observations. A Comparative Study. *Pattern Recognition Letters*, 24(14):2377–2389, 2003.

[138] H. Zhu, M. Gu, and B. Peterson. Maximum likelihood from spatial random effects models via the stochastic approximation expectation maximization algorithm. *Statistics and Computing*, 17(2):163–177, June 2007.

[139] F. T. Ramos, S. Kumar, B. Upcroft, and H. Durrant-Whyte. A Natural Feature Representation for Unstructured Environments. *IEEE Trans. on Robotics*, 24(6):1329–1340, December 2008.

[140] J. Zhang, H. Li, and C Chen. Distributed Image Coding Based On Integrated Markov Random Field Modeling and LDPC decoding. *IEEE Workshop on Multimedia Signal Processing*, pages 261–266, October 2008.

[141] Sung eok Jeon and Chuanyi Ji. Nearly Optimal Distributed Configuration Management Using Probabilistic Graphical Models. *IEEE Int. Conf. on Mobile Adhoc and Sensor Systems*, pages 8 pp.–226, November 2005.

[142] M. Wainwright. *Stochastic Processes On Graphs With Cycles: Geometric and Variational Approaches*. Massechusetts Institute of Technology, 2002. Doctoral Thesis, M.I.T.

[143] F. R. Kschischang, B. J. Frey, and H. Loeliger. Factor Graphs and the Sum Product Algorithm. *IEEE Trans. Inform. Theory*, 47(2):498–519, February 2001.

[144] G. R. Grimmett and D. R. Stirzaker. *Probability and Random Processes*. Oxford University Press, USA, 2001.

[145] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski. A Learning Algorithm for Boltzmann Machines. *Cognitive Science*, 9:147–169, 1985.

[146] S. Amari and H. Nagaoka. *Methods of Information Geometry*. AMS/Oxford University Press, 1993.

[147] S. Ikeda, T. Tanaka, and S. Amari. Stochastic Reasoning, Free Energy and Information Geometry. *Neural Computation*, 16:1779–1810, 2004.

[148] S. Amari. Information Geometry of the EM and em Algorithms for Neural Networks. *Neural Networks*, 8(9):1379–1408, 1995.

[149] S. Amari. Natural Gradient Works Efficiently in Learning. *Neural Computation*, 10(2):251–276, 1998.

[150] R. Nowak, U. Mitra, and R. Willett. Estimating Inhomogeneous Fields Using Wireless Sensor Networks. *IEEE Journal on Selected Areas in Communications*, 22(6):999–1006, August 2004.

[151] S. S. Pradhan, J. Kusuma, and K. Ramachandran. Distributed Compression in a Dense Microsensor Network. *IEEE Signal Processing Magazine*, 19(2):51–60, Mar 2002.

[152] J. Zhang and M. Fossorier. Mean Field and Mixed Mean Field Iterative Decoding for Low Density Parity Check Codes. *IEEE Trans. Inform. Theory*, 52(7):3168–3185, July 2006.

[153] A. T. Ihler, J. W. Fisher III, R. L. Moses, and A. S. Willsky. Nonparametric Belief Propagation for Self-Localization of Sensor Networks. *IEEE J. Select. Areas Commun.*, 23(4):809–819, April 2005.

[154] T. Tanaka. Information Geometry of Mean-Field Approximation. *Neural Computation*, 12:1951–1968, 2000.

[155] Xavier Boyen and Daphne Koller. Exploiting the architecture of dynamic systems. In *In Proc. of the National Conference on Artificial Intelligence (AAAI*, pages 313–320, 1999.

[156] Xavier Boyen. Tractable inference for complex stochastic processes. In *In Proc. UAI*, pages 33–42, 1998.

[157] L. K. Saul, T. S. Jaakkola, and M. I. Jordan. Mean Field Theory for Sigmoid Belief Networks. *Journal of Artificial Intelligence Research*, 4:61–76, 1996.

[158] S. Amari, S. Ikeda, and H. Shimokawa. Information Geometry and Mean Field Approximation: the Alpha-Projection Approach. *Advanced Mean Field Methods – Theory and Practice*, pages 241–257, April 2001. Chapter 16, MIT Press, Cambridge, MA, ISBN 0-262-15054-9.

[159] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference (2nd edition)*. Morgan Kaufmann (San Francisco), 1988.

[160] D. J. C. MacKay. Good Error-Correcting Codes Based on Very Sparse Matrices. *IEEE Trans. Inform. Theory*, 45(2):399–431, March 1999.

[161] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Constructing Free Energy Approximations and Generalized Belief Propogation Algorithms. *Mitsubishi Electric Research Laboratories, Cambridge, MA*, May 2004. TR-2004-040 (http://www.merl.com).

[162] S. Ikeda, T. Tanaka, and S. Amari. Information Geometry of Turbo and Low-Density Parity-Check Codes. *IEEE Trans. Inform. Theory*, 50(6):1097–1114, June 2004.

[163] B. J. Frey. *Graphical Models for Machine Learning and Digital Communication*. MIT Press, 1998.

[164] S. Cui, A. Goldsmith, and A. Bahai. Energy-Constrained Modulation Optimization. *IEEE Trans. Wireless Commun.*, 4(5):2349–2360, September 2005.

[165] R. S. Varga. *Matrix Iterative Analysis*. Springer, 2000.

[166] M. Luby. LT Codes. *Proc. IEEE Symp. on Foundations of Computer Science*, pages 271–280, June 2002.

[167] J. Haupt, R. Castro, , and R. Nowak. Adaptive discovery of sparse signals in noise. *Proc. 42nd Asilomar Conf. on Signals, Systems, and Computers*, October 2008.

[168] S. Sesia, G. Caire, and G. Vivier. Incremental Redundancy Hybrid ARQ Schemes Based on Low-Density Parity-Check Codes. *IEEE Trans. Commun.*, 52(8):1311–1321, August 2004.

[169] A. Oka and L. Lampe. Model Identification for Wireless Sensor Networks. *Proc. IEEE Global Telecom. Conf. (GLOBECOM)*, pages 3013–3018, December 2007.

[170] S. ten Brink. Convergence Behavior of Iteratively Decoded Parallel Concatenated Codes. *IEEE Trans. Commun.*, 49(10):1727–1737, October 2001.

[171] M. Tüchler. Design of Serially Concatenated Systems Depending on the Block Length. *IEEE Trans. Commun.*, 52(2):209–218, Feb. 2004.

[172] D. M. Titterington. Recursive Parameter Estimation Using Incomplete Data. *J. of the Roy. Stat. Soc.*, 46 (Series B):256 –267, 1984.

[173] V. Krishnamurthy and J. B. Moore. On-line Estimation of Hidden Markov Model Parameters Based on the Kullback-Leibler Information Measure. *IEEE Trans. Signal Processing*, 41(8):2557–2573, August 1993.

[174] T. Ryden. On recursive estimation for hidden Markov models. *Stochastic Processes and their Applications*, 66(1):79–96, February 1997.

[175] K.N. Plataniotis, S. K. Katsikas, D. G. Lainiotis, and A. N. Venetsanopoulos. Optimal Seismic Deconvolution: Distributed Algorithms. *IEEE Trans. on Geoscience and Remote Sensing*, 36(3):779–792, May 1998.

[176] R. D. Nowak. Distributed EM Algorithms for Density Estimation and Clustering in Sensor Networks. *IEEE Trans. Signal Processing*, 51(8):2245–2253, August 2003.

[177] L. Ljung. Analysis of Recursive Stochastic Algorithms. *IEEE Trans. Automatic Control*, AC-22(4):205–221, August 1977.

[178] H. J. Kushner and J. Yin. *Stochastic Approximation Algorithms and Applications*. Springer-Verlag, 1997.

[179] M. Metivier and P. Priouret. Applications of a Kushner and Clark Lemma to General Classes of Stochastic Algorithms. *IEEE Trans. Inform. Theory*, 30(2):140–151, March 1984.

[180] J. Huang, I. Kontoyiannis, and S. P. Meyn. The ODE Method and Spectral Theory of Markov Operators. *Proceedings of Stochastic Theory and Control Workshop*, pages 205–221, 2002. Springer, New York.

[181] P. Daly. Navstar GPS and GLONASS: Global Satellite Navigation Systems. *IEEE Electronics and Communication Engineering Journal*, 5(6):349–357, December 1993.

[182] G.M. Djuknic and R.E. Richton. Geolocation and Assisted GPS. *Computer*, 34(2):123–125, February 2001.

[183] S. Schon and O. Bielenberg. On the Capability of High Sensitivity GPS for Precise Indoor Positioning. *Proc. of the Fifth Workshop on Positioning, Navigation and Communication*, pages 121–127, March 2008.

[184] A. Amar and A. J. Weiss. Optimal Radio Emmiter Location Based on the Doppler Effect. *Proc. Fifth IEEE Sensor Array and Multichannel Signal Processing Workshop*, July 2008.

[185] M. Wax, Y. Meng, and O. Hilsenrath. Subspace Signature Matching for Location Ambiguity Resolution in Wireless Communication Systems. *United States Patent 6064339*, 2000.

[186] A. Beck, P. Stoica, and J. Li. Exact and Approximate Solutions of Source Localization Problems. *IEEE Trans. Signal Processing*, 56(5):1770–1778, May 2008.

[187] T. M. Nguyen, V. P. Jilkov, and X. R. Li. Comparison of sampling-based algorithms for multisensor distributed target tracking. *Proc. of the Sixth Int. Conf. on Information Fusion*, 1:114–121, 2003.

[188] D. Crisan and A. Doucet. A Survey of Convergence Results on Particle Filtering Methods for Practitioners. *IEEE Trans. Signal Processing*, 50(3):736–746, March 2002.

[189] C. W. Reynolds. Flocks, Herds and Schools: A Distributed Behavioral Model. *Comput. Graph. (Proc. ACM SIGRAPH'87)*, 21:25–34, July 1987.

[190] C. K. Chui and G. Chen. *Kalman Filtering*. Springer Verlag, 1991.

[191] A. Oka and L. Lampe. Incremental Distributed Identification of Markov Random Field Models in Wireless Sensor Networks. *IEEE Trans. Signal Processing*, 57(6):2396–2405, June 2009.

[192] M. Khabbazian and V. K. Bhargava. Localized Broadcasting with Guaranteed Delivery and Bounded Transmission Redundancy. *IEEE Trans. on Computers*, 57(8):1072–1086, Aug. 2008.

[193] T. Q. S. Quek and M.Z. Win. Analysis of UWB Transmitted-Reference Communication Systems in Dense Multipath Channels. *IEEE J. Select. Areas Commun.*, 23(9):1863–1874, September 2005.

[194] M. Pausini, G.J.M. Janssen, and K. Witrisal. Performance Enhancement of Differential UWB Autocorrelation Receivers Under ISI. *IEEE Journal on Selected Areas in Communications*, 24(4):815–821, April 2006.

[195] Z. Wang, G. R. Arce, B. M. Sadler, J. L. Paredes, S. Hoyos, and Z. Yu. Compressed UWB Signal Detection with Narrowband Interference Mitigation. *IEEE Int. Conf. on UWB*, 2:157–160, September 2008.

[196] Yao Yu, A. P. Petropulu, and H. V. Poor. Compressive sensing for mimo radar. *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pages 3017–3020, April 2009.

[197] Z. Wang, G. R. Arce, B. M. Sadler, J. L. Paredes, and X. Ma. Compressed Detection for Pilot Assisted Ultra-Wideband Impulse Radio. *IEEE Int. Conf. on UWB*, pages 393–398, September 2007.

[198] Z. Wang, G. R. Arce, J. L. Paredes, and B. M. Sadler. Compressed Detection for Ultra-Wideband Impulse Radio. *IEEE Workshop on Sig. Proc. Advances in Wireless Communications*, pages 1–5, June 2007.

[199] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[200] C. Zhu. Stable Recovery of Sparse Signals Via Regularized Minimization. *IEEE Trans. Inform. Theory*, 54(7):3364–3367, July 2008.

[201] T. Blu, P. L. Dragotti, M. Vetterli, P. Marziliano, and L. Coulot. Sparse Sampling of Signal Innovations. *IEEE Signal Processing Magazine*, 25(2):31–40, March 2008.

[202] A. Oka and L. Lampe. Compressed Sensing Reception of Bursty UWB Impulse Radio is Robust to Narrow-band Interference. Accepted for presentation at the IEEE Global Communications Conference (GLOBECOM) 2009.

[203] H. Robbins and S. Monro. A Stochastic Approximation Method. *Ann. Math. Stat.*, 22:400–407, 1951.

[204] J. Romme and L. Piazzo. On the Power Spectral Density of Time-Hopping Impulse Radio. *IEEE Conference on Ultra Wideband Systems and Technologies*, pages 241–244, 2002.

[205] IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems. *IEEE Std 802.16-2004 (Revision of IEEE Std 802.16-2001)*, 2004.

[206] E. Fishler, S. Gezici, and H. Poor. Iterative (Turbo) Multiuser Detectors for Impulse Radio Systems. *IEEE Trans. Wireless Commun.*, 7(8):2964–2974, August 2008.

[207] S. Gezici and H. V. Poor. Position Estimation via Ultra-Wide-Band Signals. *Proc. of the IEEE*, 97(2):386–403, February 2009.

[208] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 2005.

A Publications Related to This Thesis

The following publications were made on the basis of the research work conducted for this thesis.

Journal Papers

1. Anand Oka and Lutz Lampe, "Energy Efficient Distributed Filtering With Wireless Sensor Networks", IEEE Trans. Signal Processing, Vol. 56, No. 5, pp. 2062-2075, May 2008.
2. Anand Oka and Lutz Lampe, "Incremental Distributed Model Identification for Wireless Sensor Networks", IEEE Trans. Signal Processing, Vol. 57, No. 6, pp. 2396-2405, June 2009.
3. Anand Oka and Lutz Lampe, "Data Extraction From Wireless Sensor Networks Using Distributed Fountain Codes", Accepted for publication in the IEEE Trans. on Communications.
4. Anand Oka and Lutz Lampe, "A Compressed Sensing Receiver for UWB Impulse Radio Communication in Wireless Sensor Networks", Accepted for publication subject to revision in Elsevier - Physical Communications (Special Issue on Advances in Ultra-Wideband Wireless Communication).
5. Anand Oka and Lutz Lampe, "Distributed Target Tracking Using Signal Strength Measurements by a Wireless Sensor Network", Submitted to the IEEE Journal on Selected Areas in Communications.

Conference Papers

1. Anand Oka and Lutz Lampe, "Data Extraction from Wireless Sensor Networks Using Fountain Codes", Proc. Intl. Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), pp. 229-232, Dec. 2007.
2. Anand Oka and Lutz Lampe, "Model Identification for Wireless Sensor Networks", Proc. IEEE Global Telecom. Conf. (GLOBECOM), pp. 3013-3018, Dec. 2007.
3. Anand Oka and Lutz Lampe, "Distributed Filtering with Wireless Sensor Networks", Proc. IEEE Global Telecom. Conf. (GLOBECOM), pp. 843-848, Dec. 2007.

4. Anand Oka and Lutz Lampe, "Compressed Sensing of Gauss-Markov Random Fields With Wireless Sensor Networks", Proc. IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM), pp. 257-260, July 2008.
5. Anand Oka and Lutz Lampe, "Distributed Scalable Multi-Target Tracking with a Wireless Sensor Network", Proc. IEEE Intl. Conf. on Communications (ICC), June 2009.
6. Anand Oka and Lutz Lampe, "A Compressed Sensing Receiver for Bursty Communication with UWB Impulse Radio", Accepted for presentation at the IEEE Intl. Conf. on Ultra-Wideband (ICUWB), 2009.
7. Anand Oka and Lutz Lampe, "Compressed Sensing Reception of Bursty UWB Impulse Radio is Robust to Narrow-band Interference", Accepted for presentation at the IEEE Global Communications Conference (GLOBECOM), 2009.

B Proofs for Chapter 2

B.1 Proof of Lemma 1

When $G = G^T$, $Q([x^{tT}, x^{t-1T}]^T|\theta, W)$ is symmetric in the arguments x^t, x^{t-1}. Suppose X^{t-1} is distributed according to $q(x^{t-1})$. Then

$$\begin{aligned}
\Pr\{X^t = x^t\} &= \sum_{x^{t-1}} \Pr\{X^t = x^t | X^{t-1} = x^{t-1}\} q(x^{t-1}) \\
&= \sum_{x^{t-1}} \frac{Q([x^{tT}, x^{t-1T}]^T)}{q(x^{t-1})} q(x^{t-1}) \\
&= \sum_{x^{t-1}} Q([x^{tT}, x^{t-1T}]^T) = \sum_{x^{t-1}} Q([x^{t-1T}, x^{tT}]^T) \\
&= q(x^t).
\end{aligned}$$

Similarly, suppose Z^{t-1} is distributed according to $\Pr\{Z^{t-1} = [\alpha^T, x^{t-2T}]^T\} = Q([\alpha^T, x^{t-2T}]^T)$. Then, noting that the transition $[\alpha^T, x^{t-2T}]^T \to [x^{tT}, x^{t-1T}]^T$ is prohibited except when $\alpha = x^{t-1}$, we have

$$\begin{aligned}
\Pr\{Z^t = [x^{tT}, x^{t-1T}]^T\} &= \sum_{\alpha, x^{t-2}} \Pr\{Z^{t-1} = [\alpha^T, x^{t-2T}]^T\} \frac{Q([x^{tT}, x^{t-1T}]^T)}{q(x^{t-1}} \\
&= \sum_{x^{t-2}} Q([x^{t-1T}, x^{t-2T}]^T) \frac{Q([x^{tT}, x^{t-1T}]^T)}{q(x^{t-1})}
\end{aligned}$$

$$\begin{aligned}
&= \sum_{x^{t-2}} Q([x^{t-2T}, x^{t-1T}]^T) \frac{Q([x^{tT}, x^{t-1T}]^T)}{q(x^{t-1})} \\
&= Q([x^{tT}, x^{t-1T}]^T).
\end{aligned}$$

The respective stationary distributions $q(x^t)$ and $Q(x^t, x^{t-1})$ are unique since the MCs are finite homogeneous irreducible aperiodic. Time reversibility can be checked by direct verification of the equality $q(x^{t-1})P(x^t|x^{t-1}) = q(x^t)P(x^{t-1}|x^t)$.

C Proofs for Chapter 3

C.1 Proof of Lemma 2

Absence of an observation process is equivalent to $\sigma^2 = \infty$, hence $h^t = 0,\ \forall t$. The chain is reversible $\Leftrightarrow$ $Q(x^t, x^{t-1})$ is symmetric $\Leftrightarrow$ $\mathcal{M}_1(\theta, W) = \mathcal{M}_2(\theta, W) = \beta$. Hence the iteration in step 4 of the algorithm with $\alpha^{t-1} = \beta$ produces $\alpha^t = \beta$. To see that β is a locally asymptotically stable equilibrium we will first make a change of coordinates. Let

$$f(\alpha) \doteq \mathcal{M}_1\left(\theta + \begin{bmatrix} h^t \\ \alpha - \beta \end{bmatrix}, W\right). \tag{C.1}$$

Define

$$\begin{aligned} \delta &\doteq \alpha - \beta && \text{(C.2)} \\ g(\delta) &\doteq f(\alpha) - f(\beta). && \text{(C.3)} \end{aligned}$$

Then the fixed point equation $f(\alpha) = \alpha$ has been linearly mapped to the origin, $g(0) = 0$, and it will suffice to show that the origin is an asymptotically stable stationary point of $g(\delta)$. From basic definitions, for any $i \in \{1, 2, \ldots, N\}$,

$$g_i(\delta) = \frac{1}{2}\log\left(\frac{b_i(x_i^t = +1)}{b_i(x_i^t = -1)}\right) - \beta_i \tag{C.4}$$

where $b_i(x_s^t)$ is the marginal for X_i^t under the model $\left(W, \theta + \begin{bmatrix} 0 \\ \delta \end{bmatrix}\right)$. Then after some algebraic manipulation it can be shown that

$$\begin{aligned} \frac{\partial g_i}{\partial \delta_j} &= \frac{1}{2}\left(\frac{\frac{\partial}{\partial \delta_j} b_i(x_i^t = +1)}{b_i(x_i^t = +1)} - \frac{\frac{\partial}{\partial \delta_j} b_i(x_i^t = -1)}{b_i(x_i^t = -1)}\right) \\ &= \frac{1}{2}\left(\mathbb{E}\left[X_j^{t-1} | X_i^t = +1\right] - \mathbb{E}\left[X_j^{t-1} | X_i^t = -1\right]\right), \end{aligned} \tag{C.5}$$

where the conditional expectation is derived from the model $\left(W, \theta + \begin{bmatrix} 0 \\ \delta \end{bmatrix}\right)$. Noting that these conditional expectations are continuously differentiable functions of G, and

the fact that when $G = 0$, X_i^t and X_j^{t-1} are independent, it follows that as $||G|| \to 0$

$$\begin{aligned} \mathbb{E}\left[X_j^{t-1}|X_i^t = +1\right] \to \;\; m_j^{t-1} \;\; \leftarrow \mathbb{E}\left[X_j^{t-1}|X_i^t = -1\right], \\ \frac{\partial g_i}{\partial \delta_j} \;\; \to \;\; 0. \end{aligned} \tag{C.6}$$

Clearly then, for $||G||$ sufficiently small, all the entries in the matrix $\left[\frac{\partial g_i}{\partial \delta_j}\right]$ can be guaranteed to be smaller than $\frac{1}{N}$ in absolute value, which ensures that all its eigen values are inside the unit circle, and hence the system $\delta^{t+1} = g(\delta^t)$ is asymptotically stable.

C.2 Proof of Lemma 3

Case 1: $G = 0 \Leftrightarrow Q(x^t, x^{t-1}) = f(x^t)q(x^{t-1})$, where $f(x^t) \doteq \sum_{x^{t-1}} Q(x^t, x^{t-1})$. Then equation (3.4) simplifies to

$$\begin{aligned} p_s^t(x_s^t) &= \sum_{x_{\neq s}^t} P(y^t|x^t) f(x^t) \sum_{x^{t-1}} q(x^{t-1}) \frac{p^{t-1}(x^{t-1})}{q(x^{t-1})} \\ &= \sum_{x_{\neq s}^t} P(y^t|x^t) f(x^t). \end{aligned} \tag{C.7}$$

Thus the approximations for p^{t-1} and q have no effect on the calculation of $p_s^t(x_s^t)$, which is calculated exactly in step 4 of the algorithm.
Case 2: If $W_s = 0$ and G is diagonal, then there is no statistical dependence between X_i^t and $X_j^{t'}$ for any t, t', when $i \neq j$. Then the replacement of p^{t-1} and q by the product of their respective marginals is not an approximation and in fact holds exactly.
Case 3: From Lemma 2 we know that β is the fixed point of the algorithm. But, under time reversibility, β is also the exponential parameter of the marginals of the stationary distribution of the chain, $\pi(\cdot)$. This is known to be the optimal inference, since $h^t = 0$, $\forall t \geq 0$ implies the lack of an observation process and hence the a-posteriori distribution of the chain is π for all times $t \geq 0$.

D Proofs for Chapter 5

D.1 Proof of Theorem 3

Define $g(\gamma) \doteq \gamma + \epsilon\, \bar{f}(\gamma)$. Since the expectation of the score under the truth model is always zero [5], it follows that $g(\gamma^*) = \gamma^* + \epsilon A\, \mathbb{E}_{\pi(y^t|\gamma^*)}\left[S_{\gamma^*}(Y^t)\right] = \gamma^* + 0_M$. Hence γ^* is a fixed point of the recursion (5.3). To see that the recursion is asymptotically stable, note that

$$\nabla_\gamma g(\gamma)\Big|_{\gamma=\gamma^*} \;=\; I - \epsilon A F^Y_{\gamma^*}. \tag{D.1}$$

Due to the property $\lim_{\sigma^2\to 0} F^Y_\gamma = F^X_\gamma$ (cf. Section 2.5) and the continuous dependence of the eigen values of a matrix on its elements, there exists a $\sigma^2_{thresh} > 0$ such that for $\sigma^2 < \sigma^2_{thresh}$ we have $F^Y_{\gamma^*} > 0$, and hence $AF^Y_{\gamma^*} > 0$ (since $A > 0$ by hypothesis). Then, by Weyl's theorem [208], there exists $\bar{\epsilon} > 0$ such that for any $0 \le \epsilon \le \bar{\epsilon}$, all eigenvalues of $\nabla_\gamma g(\gamma)\Big|_{\gamma=\gamma^*}$ are inside the unit circle, and γ^* becomes an A.S. fixed point of the recursion (5.3).

D.2 Proof of Lemma 4

Note that $\frac{\epsilon}{2-\epsilon}$ is the normalized noise equivalent bandwidth, and $\tau(\epsilon) = \frac{2-\epsilon}{\epsilon}$ is the time constant, of a first order Infinite Impulse Response (IIR) filter $x^{t+1} = (1-\epsilon)\, x^t + \epsilon\, u^t$. Since the incremental estimator in equation (5.2) is unbiased, the CRLB is applicable. However, for a fair comparison, one must postulate a data window $n = \tau(\epsilon)$. Now by choosing $n \gg 1$ (hence $\epsilon \ll 1$), we can ensure that linearized analysis holds with arbitrary precision. Then a comparison of equation (5.8) and the CRLB (2.27) makes it clear that the incremental estimator is asymptotically ($n \to \infty, \epsilon \to 0$) efficient.

E Proofs for Chapter 6

E.1 Calculation of the Score and the Fisher Information

Denote the projection of a sub-state to its spatial position by $\psi : \mathbb{R}^{2D} \to \mathbb{R}^D$, $\psi(a) = [a_1, a_3, \ldots, a_{2D-1}]^T$. In the following development, for clarity we drop the temporal index for the various quantities. Thus for example, at any generic time, $Y_{m,n}$ will denote the observation of tag m made by mote n, $\Gamma_{l,n}$ will denote the gain from cell l to mote n, etc. Using the presumed mutual independence of the sub-states, the log-likelihood of the parameters can be written as

$$\lambda(\Upsilon; \{Y_{l,n}\}) = K + \sum_{m=1}^{M} \log \int p(\{Y_{m,n} : \text{ all } n\} | \Phi_m, \{\Gamma\}, z_m)\, q_m(z_m) dz_m \tag{E.1}$$

where K is some constant. Owing to the observation model in equation (6.4), the conditional likelihood is given by

$$\begin{array}{l} p(\{Y_{m,n} : \text{ all } n\} | \Phi_m, \{\Gamma\}, z_m) = \\ \dfrac{\exp\left\{ \frac{-1}{2\sigma_W^2} \sum_{n=1}^{N} \left(Y_{m,n} - \Phi_m - \Gamma_{f(\psi(z_m)),n} + \rho 10 \log \|\psi(z_m) - r_n\| \right)^2 \right\}}{(2\pi\sigma_W^2)^{N/2}}. \end{array} \tag{E.2}$$

Then the *score* of the parameters is given by

$$\begin{array}{l} S_{\Phi;m} \doteq \frac{\partial \lambda(\{Y_{l,n}\})}{\partial \Phi_m} \\ = \frac{1}{\sigma_W^2} \dfrac{\int \left(\begin{array}{l} q_m(z_m) p(\{Y_{m,n'} : \text{ all } n'\} | \Phi_m, \{\Gamma\}, z_m) \cdot \\ \sum_{n=1}^{N} \left(Y_{m,n} - \Phi_m - \Gamma_{f(\psi(z_m)),n} + \rho 10 \log \|\psi(z_m) - r_n\| \right) \end{array} \right) dz_m}{\int q_m(z_m) p(\{Y_{m,n'} : \text{ all } n'\} | \Phi_m, \{\Gamma\}, z_m) dz_m} \\ S_{\Gamma;l,n} \doteq \frac{\partial \lambda(\{Y_{l,n}\})}{\partial \Gamma_{l,n}} \\ = \frac{1}{\sigma_W^2} \sum_{m=1}^{M} \dfrac{\int \left(\begin{array}{l} q_m(z_m) p(\{Y_{m,n'} : \text{ all } n'\} | \Phi_m, \{\Gamma\}, z_m) \cdot \\ \left(Y_{m,n} - \Phi_m - \Gamma_{f(\psi(z_m)),n} + \rho 10 \log \|\psi(z_m) - r_n\| \right) \end{array} \right) \mathbb{I}_l(\psi(z_m)) dz_m}{\int q_m(z_m) p(\{Y_{m,n'} : \text{ all } n'\} | \Phi_m, \{\Gamma\}, z_m) dz_m}, \end{array} \tag{E.3}$$

where $\mathbb{I}_l(a) = 1$ if $l = f(a)$, zero otherwise (i.e. an indicator function for the condition that position a lies in cell number l), and the total score is given by

$$S(\Upsilon; \{Y_{l,n}\}) \doteq f_{stack}(\{S_{\Phi;m}\}, \{S_{\Gamma;l,n}\}). \tag{E.4}$$

The integrations involved in the RHS of equations (E.3) are approximated in practice by taking sample averages over the set of sub-state particles. For example,

$$S_{\Gamma;l,n} \approx \frac{1}{\sigma_W^2} \sum_{m=1}^{M} \frac{\sum_{\pi=1}^{\Pi} p(\{Y_{m,n'} : \text{ all } n'\}|\Phi_m, \{\Gamma\}, \zeta_{m,\pi}) \cdot \left(Y_{m,n} - \Phi_m - \Gamma_{f(\psi(\zeta_{m,\pi})),n} + \rho 10 \log \|\psi(\zeta_{m,\pi}) - r_n\|\right) \cdot \mathbb{I}_l(\psi(\zeta_{m,\pi}))}{\sum_{\pi=1}^{\Pi} p(\{Y_{m,n'} : \text{ all } n'\}|\Phi_m, \{\Gamma\}, \zeta_{m,\pi})}. \tag{E.5}$$

For calculating the Fisher information F_Υ we need to further differentiate the score and take an expectation w.r.t. the joint distribution of the state and the observations made available to the estimator. Since no closed form expression is available for the propagated density $q(\cdot)$ of the total-state (which is the reason why we use a particle representation in the first place), it appears infeasible to obtain a closed form expression for F_Υ. However we can calculate an optimistic approximation by differentiating the conditional likelihood assuming *perfect* state information at the estimator. That is,

$$F_\Upsilon \lessapprox \hat{F}_\Upsilon = \mathbb{E}_{p(Z_{true})} \mathbb{E}_{p(\{Y_{m,n}\}|\Upsilon, Z_{true})} \left[-\frac{\partial^2}{\partial \Upsilon^2} \log p(\{Y_{m,n}\}|\Upsilon, Z_{true}) \right]. \tag{E.6}$$

Since

$$\begin{aligned} &\log p(\{Y_{m,n}\}|\Upsilon, Z) = \\ &K - \textstyle\sum_{m=1}^{M} \sum_{n=1}^{N} \frac{\left(Y_{m,n} - \Phi_m - \Gamma_{f(\psi(Z_m)),n} + \rho 10 \log \|\psi(Z_m) - r_n\|\right)^2}{2\sigma_W^2} \end{aligned} \tag{E.7}$$

where K is a constant, the expectations of the various second derivatives of the log-likelihood w.r.t. $\{Y_{m,n}\}|\Upsilon, Z_{true}$ are given by

$$\mathbb{E}_{p(\{Y_{m,n}\}|\Upsilon, Z_{true})} \left[\frac{\partial^2 \log p(\{Y_{m,n}\}|\Upsilon, Z_{true}))}{\partial \Phi_m \partial \Phi_{m'}} \right] = \begin{cases} \frac{-N}{\sigma_W^2}, \text{ if } m = m' \\ 0, \text{ otherwise} \end{cases}, \tag{E.8}$$

$$\mathbb{E}_{p(\{Y_{m,n}\}|\Upsilon, Z_{true})} \left[\frac{\partial^2 \log p(\{Y_{m,n}\}|\Upsilon, Z_{true}))}{\partial \Gamma_{l,n} \partial \Gamma_{l',n'}} \right]$$

$$= \begin{cases} \frac{-\sum_{m=1}^{M} \mathbb{I}_l(\psi(Z_{true,m}))}{\sigma_W^2}, \text{ if } (l,n) = (l',n') \\ 0, \text{ otherwise} \end{cases}, \tag{E.9}$$

$$\mathbb{E}_{p(\{Y_{m,n}\}|\Upsilon, Z_{true})} \left[\frac{\partial^2 \log p(\{Y_{m,n}\}|\Upsilon, Z_{true}))}{\partial \Phi_m \partial \Gamma_{l,n}} \right] = \frac{-\mathbb{I}_l(\psi(Z_{true,m}))}{\sigma_W^2}. \tag{E.10}$$

Finally, let us make the simplifying assumption that all the cells are of equal area and that, after the MC of equation (6.3) becomes stationary, the true positions of the tags are a-priori uniformly distributed in the tracking region (the velocity distribution is imma-

terial). By taking the expectations of the (negative of the) RHS of equations (E.8),(E.9) and (E.10) under such a $p(Z_{true})$, and packing them into a matrix form compatible with the stacking operation f_{stack}, we get the final form of the optimistic approximation $\hat{F_{\Upsilon}}$ given by equation (6.18).