



*Técnicas Bayesianas aplicadas al
procesado de señal en
Comunicaciones y Genómica*

TESIS DOCTORAL
por
Isabel María Tienda Luna

Departamento de Física Aplicada, Universidad de Granada

Directores:

Dr. M^a del Carmen Carrión Pérez
Dr. Diego Pablo Ruiz Padillo
Dr. Yufei Huang

Julio, 2006

Editor: Editorial de la Universidad de Granada
Autor: Isabel maría Tienda Luna
D.L.: Gr. 1274 - 2006
ISBN: 978-84-338-4013-4

*A mis padres y mis abuelos
por ser siempre modelos a seguir.
A Pablo...*

Dña. M^a del Carmen Carrión Pérez, Doctora en Ciencias Físicas y Profesora Titular del Departamento de Física Aplicada de la Facultad de Ciencias de la Universidad de Granada y

D. Diego Pablo Ruiz Padillo, Doctor en Ciencias Físicas y Profesor Titular del Departamento de Física Aplicada de la Facultad de Ciencias de la Universidad de Granada y

D. Yufei Huang, Doctor por la State University of New York at Stony Brook y Assistant Professor del Departamento de Ingeniería Electrónica y Computadores de la Universidad de Texas en San Antonio,

MANIFIESTAN:

Que la presente Memoria titulada “Técnicas Bayesianas aplicadas al procesamiento de señal en Comunicaciones Digitales y Genómica”, presentada por Isabel María Tienda Luna para optar al Grado de Doctor en Ciencias Físicas, ha sido realizada bajo nuestra dirección en el Departamento de Física Aplicada de la Universidad de Granada y el Departamento de Ingeniería Electrónica y Computadores de la Universidad de Texas en San Antonio.

Con esta fecha autorizamos la presentación de la misma.

Granada, 25 de Mayo de 2006

Fdo:M^a del Carmen Carrión Pérez

Fdo: Diego Pablo Ruiz Padillo

Fdo: Yufei Huang

Memoria presentada por Isabel María Tienda Luna para optar al Grado de Doctor en Ciencias Físicas por la Universidad de Granada.

Fdo: Isabel María Tienda Luna

Agradecimientos

En primer lugar me gustaría agradecer a mis directores de tesis M^a Carmen, Diego y Yufei por su ayuda y apoyo, por creer en mi trabajo y motivarme cuando lo he necesitado. A ellos les debo la formación académica e investigadora que he recibido durante este periodo de cuatro años. Y en este aspecto puedo afirmar que me siento privilegiada de haber podido trabajar con ellos. Pero no sólo en lo académico se merecen mi reconocimiento sino también en lo personal ya que me han animado en los momentos buenos y en los no tan buenos. Hacia ellos reitero mi más sincero agradecimiento por todo lo que han hecho por mí.

El ambiente del 114 ha sido también fundamental en todo este tiempo. Es por eso que me gustaría agradecer primero a David y a Enrique por ayudarme y aconsejarme cuando lo he necesitado. A Pepe por haberme ayudado con cualquier cosa de informática que ha estado en su mano y por haberme hecho ver que compartir es bueno. A Diego Salas gracias por las discusiones y la ayuda (además del buen rollo). A Susana y Otilia gracias por el punto femenino de marujeo, sin duda las cosas no habrían sido las mismas sin ellas. A todos gracias por escucharme y aguantarme en los buenos ratos y los no tan buenos, todos han contribuido a que estos cuatro años estén llenos de buenos recuerdos. También me gustaría dejar patente aquí mi agradecimiento a mis compañeras de despacho en San Antonio: Sandy y Jiayin. Ellas hicieron que me sentiera como en casa y no dudaron en prestarme ayuda de cualquier tipo en cualquier momento, la mayoría de mis buenos momentos en Texas se los debo a ellas. Y como no, también me gustaría agradecer aquí a mis amigos por escucharme, entenderme y darme ánimos.

Por supuesto mi más especial agradecimiento a mi familia. A mis padres y a mis abuelos por darme todo lo que tengo y por enseñarme con el ejemplo que cualquier cosa puede conseguirse con voluntad y esfuerzo. Ellos han sido un modelo a seguir en todo momento. Muchas gracias por comprenderme, ayudarme, animarme cuando lo he necesitado y por el esfuerzo que han tenido que hacer para que yo esté hoy aquí. También tengo que agradecer a mis hermanos, Inma y Fernando por haber sabido tratarme y comprenderme, por aguantarme en los momentos de nervios.

Finalmente quiero mostrar mi más sincero agradecimiento a Pablo, sin su apoyo y sus ánimos constantes no habría podido ni querido terminar esta Tesis. Gracias por creer en mí, por escucharme, por hacerme reír, por la infinita e incalculable ayuda y la infinita paciencia que ha tenido conmigo, por las largas noches de messenger y teléfono durante mis estancias... La

verdad es que podría escribir otra Tesis y me faltaría espacio para agradecer cada una de las cosas que ha hecho por mí y cuánto han significado.

Título de Doctor con Mención Europea

Con el fin de obtener la Mención Europea en el Título de Doctor (aprobada en Junta de Gobierno de la Universidad de Granada el 5 de Febrero de 2001), se han cumplido, en lo que atañe a esta Tesis Doctoral y a su Defensa, los siguientes requisitos:

1. Se ha incluido un amplio resumen en inglés como parte de esta Memoria.
2. Uno de los miembros del tribunal, el Doctor, proviene de una universidad europea no española, en particular de
3. Una parte de la defensa se ha realizado en inglés.
4. Una parte de esta Tesis Doctoral se ha realizado en Francia, en el Departamento de I+D+I de la empresa Diginext.

Financiación

Las principales fuentes de financiación que han permitido llevar a cabo esta Tesis Doctoral han sido las siguientes:

- Beca de Formación de Profesorado Universitario (FPU) del Ministerio de Educación y Ciencia, en su orden del 19 de Abril de 2002. BOE 01-09-2001. El periodo de disfrute de la misma ha sido desde el 1 de Abril de 2002 al 28 de Febrero de 2006.
- Proyecto denominado “Desarrollo de algoritmos de tratamiento de señal en comunicaciones. Aplicación a la identificación ciega de canales y el análisis tiempo frecuencia de su respuesta,” del Ministerio de Ciencia y Tecnología con código TIC 2001-2902, con una duración de 2001-2003.
- Proyecto denominado “Tratamiento de señal en ambientes no estacionarios usando técnicas de procesado conjunto ICA, HOS y PF. Aplicaciones en comunicaciones e imágenes,” del Ministerio de Ciencia y Tecnología con código TEC 2004-06096-C03-02/TCM, con una duración de 2004-2007.
- Ayudas para estancias breves incluidas dentro de la beca FPU del Ministerio de Educación y Ciencia para realizar una estancia de 3 meses y 21 días en la Universidad de Texas en San Antonio en 2003, una estancia de 3 meses y 21 días en la empresa Diginext en Aix en Provence (Francia) en 2004 y una estancia de 2 meses en la Universidad de Texas en San Antonio en 2005.

Summary in English

Bayesian Theory offers a methodological framework and mathematical concepts for model selection and parameter learning. Bayesian inference is an approach to statistics in which all forms of uncertainty are expressed in terms of probability. A Bayesian approach to a problem starts with the formulation of a model that is hoped to be adequate to describe the situation of interest. Then a prior distribution is formulated over the unknown parameters of the model, which is meant to capture the beliefs about the situation before seeing the data. After observing some data, the Bayes' Rule is applied to obtain a posterior distribution for these unknowns, which takes account of both the prior and the data. Due to this fact, the posterior distribution is more informative than the likelihood function. Finally, from this posterior distribution we can compute predictive distributions for future observations.

This theoretically simple process can be justified as the proper approach to uncertain inference by various arguments. Despite this, many people are uncomfortable with the Bayesian approach, often because they view the selection of a prior as being arbitrary and subjective. It is indeed subjective, but for this reason it is not arbitrary. There is (in theory) just one correct prior, the one that captures your (subjective) prior beliefs. In contrast, other statistical methods are truly arbitrary, in that there are usually many methods that are equally good according to non-Bayesian criteria of goodness, with no principled way of choosing between them.

Despite all its advantages, Bayesian methods are more computationally demanding than other methods based on the likelihood. Exact Bayesian analysis is usually analytically intractable due to the integration processes related to it. Thus, the dominant computational task in Bayesian inference is numerical integration. Then, new methods for fast and accurate integration are very important and can have a great impact. This thesis proposes new approaches to approximate Bayesian inference in some important applications and addresses the complexity issue in Bayesian computation.

This thesis applies the Bayesian Theory to the signal processing field. A separation has been done depending on the techniques used to approximate the Bayesian inference process. Specifically, the first part of the thesis is devoted to Particle Filters which are Bayesian Monte Carlo approaches. This techniques are applied to two Communication problems. The first one is

blind Multiuser Detection for fast flat Rayleigh fading channels and synchronous systems. The second problem in the field of Communication is the joint decoding and phase estimation in channels with phase noise. In the second part of the thesis a Variational Bayesian algorithm is presented. In this part, our approach is applied to the emerging field of Genomic signal processing.

In Chapter 1 an overview of the problems treated in this thesis is presented. Chapter 2 is an introduction to Communications. In Chapter 3 some general ideas about Bayesian Filtering are introduced.

In Chapter 4 the use of the stochastic M-algorithm (SMA), a novel and efficient particle filtering algorithm designed for the discrete unknowns, is proposed for blind Multiuser Detection. Since multiuser detection (MUD) was introduced in the eighties, it has received a great deal of attention due to its ability to reduce multiple access interference (MAI) and potential for increasing the capacity of CDMA systems. Since then, numerous detectors have been proposed in the literature for both synchronous and asynchronous transmission and some popular ones include the decorrelating detector, the minimum mean square error (MMSE) detector, the multi-stage detector, and the decision feedback detector [84].

In practice, distortion in signal strength due to time varying fading channels must be attended while performing MUD. Even though, non-coherent detection methods as proposed in [1] are often appealing owing to its simplicity since no inference on fading channels is needed, coherent detection has been proved to deliver better performance [89]. With coherent detection, estimation of channels can be obtained with or without pilot signals. Between them, significant amount of research has been devoted to schemes without using pilot signals, or blind MUD methods. Blind MUD methods are bandwidth more efficient and the approaches proposed, to name a few, include the recursive least square (RLS) [43, 51], subspace based [86], expectation-maximization [32], genetic algorithm [93] and Kalman filtering [34, 53, 61, 62, 82, 90]. However, most of the approaches cited above assumes slow or quasi static fading channels.

This thesis is focused on blind MUD for fast flat Rayleigh fading channels and in synchronous systems. In particular, it is assumed to know *a priori* the second order statistics of the underlying channel, based on which a mathematical tractable approximation using autoregressive-moving-average (ARMA) model is adopted. The approximation enables a dynamic state space modeling (DSSM) of the problem, which lends itself naturally to a Kalman filtering related detection solution. The use of Kalman filtering for blind MUD on the similar modeling has been seen in [34, 82, 90], where the decision direct approach was used to estimate the channel variable necessary for the Kalman filtering. One inherent drawback with the decision directed

approach is the error propagation, which greatly limits the performance of such implementation.

Recently, the combined (mixture) Kalman filtering and sequential importance sampling (particle filtering) algorithms have been applied to blind detection of convolutional codes [19], space-time trellis codes [96], and blind MUD [72] over fading channels. The mixture Kalman filtering (MKF) approach is shown to greatly reduce the error propagation of the decision direct implementations and thus exhibits considerable performance improvement. However, in the proposed use of the MKF to blind MUD in [72], particle filtering (PF) was mainly intended for channel tracking and the embedded MUD at a symbol interval is achieved by an optimum detector, which has exponential complexity with the number of users. Consequently, the proposed MKF algorithm becomes prohibitively complex even for systems with moderate number of users.

In this thesis, unlike all existing Kalman filtering detectors, a completely different viewpoint to multiuser systems is taken and it is proposed a novel time-observation state space model (TOSSM). Even though the TOSSM is equivalent to the common DSSM, it allows the PF based multiuser detection to be naturally blended with the mixture Kalman filtering for channel estimation. The new mixture Kalman filtering algorithm samples one user at a time and therefore permits efficient implementation. It is further proposed to use a more efficient PF algorithm known as the stochastic M -algorithm (SMA), which has shown to attain additional complexity reduction over the generic PF implementation yet maintain comparable performance.

In Chapter 5 Factor Graph representation and the Sum-Product algorithm are introduced and in Chapter 6 a novel Particle Filtering technique is applied to perform iterative decoding and phase estimation for channels with phase noise using a Factor Graph representation. Several iterative decoding algorithms under Bayesian frameworks have been considered in the literature for channels with unknown constant or time-varying phase. While non-Bayesian schemes are efficient for constant phase estimation, they perform significantly worse in the presence of time-varying phase. That motivates the study of the iterative decoding over channels with time-varying phase noise and to use Bayesian sequential Monte Carlo methods, or particle filters, to perform the on-line phase estimation. The particle filters are powerful tools for numerically computing the optimal estimators when exact solutions cannot be analytically derived. They employ discrete measures with random supports, or particles, for sequentially estimating the posterior probability distributions of the unknowns. When applied to phase estimation, the particle filters allow a continuous exploration of the phase space with the evolution of particles in the time.

On the other hand, factor graph representation [57] provides a powerful framework to reveal important yet often obscure factorizations in different systems. Such factorization allow the use of the efficient Sum-Product algorithm for inference. Sum-Product algorithm has shown to be an important tool to reinterpret a number of well-known algorithm in digital communications [88] and to design more efficient and powerful algorithms.

In this thesis, the combination of the Sum-Product algorithm with the Particle Filtering Algorithm is proposed in order to perform iterative decoding over channels with phase noise. In [88], the authors propose to discretize the phase space using equally spaced values. However, in this case the complexity may be very high because all the possible branches in a trellis must be analyzed. Since the complexity of Particle Filtering is directly proportional to the number of particles and hence, it may be much smaller than that of a full-trellis exploration. Here, it thus proposed to reduce the complexity of the discrete approach by using particle filtering methods [2]. Particle Filtering techniques were applied to this problem in [23] for the first time. However, the number of particles used in this algorithm is high and depends on the length of the code. In this thesis, Particle Filtering is used to estimate the phase and, afterward, that estimation is used to perform decoding. The proposed algorithm is independent of the length of the code and thus the number of particles used can be reduced. One of the major drawbacks of the Particle Filtering techniques is the degeneracy phenomenon. In order to reduce as much as possible the negative effects of that phenomenon a resampling step has to be introduced. That resampling step is not considered in [23] and as a result, several iterations are needed to estimate the phase. However the proposed algorithm here includes the resampling step and thus the number of iterations can be reduced. Ultimately, the computational complexity is also reduced.

In Chapters 7 and 8 a novel Variational Bayesian Expectation Maximization algorithm is developed to uncover Gene Networks. Gene networks are views of molecular interactions at genomic levels [16]. The interactions are the manifestation of gene-gene regulation, protein-protein interaction, and protein-metabolites association. Understanding molecular interactions is of considerable importance as many complex cellular functions are controlled through such interactions. Essentially, gene networks inject function to the otherwise seemingly isolated genetic sequences. As a result, uncovering gene networks can suggest new interpretation or questions for experiments, provides insights for novel drug targets, and leads to principles for treating complex diseases and concepts for drug design.

Whereas experiments are indispensable, computations have played an increasingly important role in uncovering gene networks. Such shift in paradigm

is much due to the proliferation of high throughput data that provide direct or indirect information about molecular interactions from different perspectives. Through extracting and piecing together relevant evidence from data using computational tools, we strive to efficiently and effectively reconstruct gene networks.

This thesis is focused on a Bayesian approach for reconstructing gene networks from microarray data. Microarray data provide first-hand measurements on genome-wide gene expressions, molecular productions out of molecular interactions, and thus microarray data are a natural source for the gene network uncovering. Yet, the task of reconstructing gene networks is complicated due to the enormously large scale of the unknowns with rather small sample size, inherent experimental defects, noisy readings, and many other factors. These difficulties call for heavy involvement of statistical signal processing including Bayesian methods.

Bayesian solutions for uncovering gene networks have been reported in the literature [35, 36, 45, 49, 54, 70, 74] and they can be loosely categorized as point (hard) and probabilistic (soft) solutions. The aim of point solution is to find a single point estimate of the networks through mainly optimization. The approaches under this category include step-wise search, greedy search, and simulated annealing. On the other hand, soft solutions intend to estimate the *a posteriori probabilities* (APPs) of the networks and the approaches herein include Monte Carlo sampling and variational Bayes approach. Currently, most of the uses of Bayesian framework in this research rely on point solutions. Despite many advantages of hard solutions including especially their relative low complexity, salient features of soft solutions are much appealing. First, the soft solutions produce estimates on APPs, providing a direct measurement of confidence in the solution, which is unavailable in point solutions. Secondly, the obtained probabilistic information is essential for Bayesian integration of additional data sets from the same and/or different sources, an important task in the gene networks research that can not be undertaken with hard solutions.

Of particular interest to this thesis are soft Bayesian solutions. The existing soft solutions for the gene networks recovering include, most notably, Monte Carlo sampling (Gibbs sampling [49] and reversible jump MCMC sampling [45]), and variational Bayes EM (VBEM) [12]. Monte Carlo sampling algorithms require a large number of samples for guaranteed accuracy and robust operation, which make their use computationally prohibitive on large networks. The VBEM solution, on the other hand, has shown to be less burdensome while providing, in many cases, comparable performance to Monte Carlo sampling. However, the VBEM solution proposed in [12] does not consider explicitly the uncertainty of the network topology in its modeling. The

problem is formulated as a parameter estimation and the topology is determined by measuring the closeness of the estimated regulatory weights to zero. Although such treatment can reduce the complexity of learning, it does not faithfully reflect the nature of the problem. This motivates us to investigate the VBEM solutions for joint topology learning and model parameter estimation.

In specific, the topology learning is formulated as a problem of variable selection, in which the network topology is determined by selecting a set of predictors (time order microarray data). Based on the formulation, a novel VBEM scheme is developed for joint estimation of continuous parameter and binary indicators variable. Compared with the Gibbs sampling, the proposed VBEM solution is shown to have much reduced complexity and good inference performance. In addition, we develop a Bayesian integration scheme for combining two different microarray data sets, taking advantage of the APPs of topology obtained from the VBEM solution. The scheme provides a viable and efficient framework for Bayesian data integration.

Finally, in Chapter 9 the most concluding remarks are presented. Many opportunities for future work are available either in the field of communications or in genomic signal processing, all of them are also presented in Chapter 9.

Índice general

1. <i>Introducción</i>	1
1.1. Comunicaciones: Detección Multiusuario	3
1.1.1. Planteamiento del Problema	3
1.1.2. Planteamientos anteriores	7
1.1.3. Aportaciones del Trabajo	9
1.2. Comunicaciones: Detección, decodificación y estimación de fase conjunta	10
1.2.1. Planteamiento del problema	10
1.2.2. Aportaciones anteriores	11
1.2.3. Aportaciones del trabajo	12
1.3. Genómica	12
1.3.1. Planteamiento del Problema	12
1.3.2. Aportaciones anteriores	15
1.3.3. Aportaciones del trabajo	16
1.4. Organización	17

I Algoritmos Bayesianos Monte Carlo aplicados a Comunicaciones **19**

2. <i>Comunicaciones Digitales</i>	21
2.1. Codificación de la fuente	22
2.1.1. Codificación Diferencial	23
2.2. Codificación de la Señal	23
2.3. Codificación de la Señal en sistemas DS-CDMA	24
2.3.1. Ventajas del DS-CDMA	25
2.4. Interespaciado	27
2.5. Modulación Digital	27
2.5.1. Modulación digital en fase (PSK)	29
2.5.2. Modulación digital en frecuencia (FSK)	30
2.5.3. Modulación digital en amplitud (ASK)	30

2.5.4.	Modulación por amplitud de cuadratura (QAM)	31
2.5.5.	Objetivos de la modulación	31
2.6.	Modelado del canal	32
2.6.1.	Caracterización de la propagación	34
2.6.2.	Desvanecimiento a gran escala	35
2.6.3.	Desvanecimiento a pequeña escala	36
2.7.	Caracterización del error de transmisión: BER	38
2.8.	Caracterización de la calidad de la señal: SNR	38
3.	<i>Filtrado Bayesiano</i>	41
3.1.	Inferencia Bayesiana	43
3.2.	Sistemas Probabilísticos Dinámicos	44
3.2.1.	Definición y análisis	44
3.2.2.	Ejemplos	45
3.3.	Teoría de estimación de Bayes	46
3.3.1.	Estimaciones MAP	46
3.3.2.	Estimación Bayesiana con distribuciones Gaussianas	47
3.4.	Estimaciones recursivas	49
3.5.	Filtrado de Kalman	52
3.5.1.	Modelo de espacio de estados	52
3.5.2.	Ecuaciones del Filtro de Kalman	53
3.6.	Filtros de Kalman Extendidos	56
3.6.1.	Modelo de espacio de estados para sistemas no lineales	56
3.6.2.	Linealización en sistemas dinámicos	57
3.6.3.	Ecuaciones del Filtro de Kalman Extendido	59
3.7.	Introducción al Filtrado de Partículas	60
3.7.1.	Importance Sampling	61
3.7.2.	Importance Resampling	64
3.8.	Algoritmo M -Estocástico	67
3.9.	Filtros de Kalman de Mezcla	71
3.9.1.	Modelos para sistemas dinámicos lineales condicionados	71
3.9.2.	Método de Mezcla de filtrado de Kalman	71
4.	<i>Aplicación del Algoritmo M-Estocástico a Detección Multi-usuario</i>	75
4.1.	Formulación del Problema	77
4.2.	Modelado de espacio de estados de tiempo-observación	78
4.3.	Detección Ciega Bayesiana Óptima	81
4.4.	Una aproximación dirigida por decisión a la DMU ciega	83
4.5.	Filtrado de Partículas aplicado a la DMU ciega	85
4.6.	Detector M -Estocástico para DMU ciega	88

4.7. Detector Genie-Aided	90
4.8. Comparación entre los Detectores	90
4.9. Escenarios con Near-Far	91
4.10. Simulaciones	91
4.10.1. Usuarios de Igual potencia	91
4.10.2. Usuarios con distinta potencia	94
4.11. Conclusiones	97
5. LDPC Codes and Factor Graph representation	99
5.1. Shannon's Theorem	100
5.2. LDPC Codes, an introduction	101
5.2.1. Regular LDPC Codes	103
5.2.2. Irregular LDPC Codes	103
5.3. Factor Graphs: some historical remarks	104
5.4. Definition of Factor Graph	104
5.5. Sum-product Algorithm	107
6. Particle Filtering for joint iterative decoding and phase es-	111
 timation	
6.1. System Model	113
6.2. Phase Noise	113
6.3. Factor Graph representation of the system	115
6.4. Solutions of the Sum Product Algorithm	119
6.5. Equally Spaced Discretization	119
6.6. Tikhonov Parametrization	120
6.7. Gaussian Parametrization	121
6.8. First Particle Filtering approach	123
6.9. Second Particle Filtering approach	124
6.10. Third Particle Filtering approach	126
6.11. Simulation results	127
6.12. Conclusions	132
6.13. Future Work	133

II Algoritmos Bayesianos Variacionales aplicados a Genómica 135

7. Selección de variable mediante algoritmos Bayesianos varia-	137
 cionales	
7.1. Formulación del Problema	139
7.2. Aprendizaje Variacional Bayesiano	140

7.2.1.	Ideas Generales	140
7.2.2.	Solución Óptima	142
7.2.3.	Solución VBEM	144
7.2.4.	Primera Condición	145
7.2.5.	Segunda Condición	145
7.3.	Aprendizaje con VBEM	146
7.3.1.	Inicialización	146
7.3.2.	Condición I	147
7.3.3.	Condición II	147
7.3.4.	Paso VBE	147
7.3.5.	Paso VBM	148
7.3.6.	Cálculo del límite inferior \mathcal{F}	149
7.3.7.	Resumen del Algoritmo	150
7.4.	Algoritmo VBEM con Ligaduras	150
7.4.1.	Objetivo	151
7.4.2.	Ejemplo con dos variables	151
7.5.	Cálculos en una dimensión	154
7.5.1.	Cálculo de μ_z	154
7.5.2.	Cálculo de σ_z^2	155
7.6.	Cálculos en el espacio G -Dimensional	156
7.6.1.	Cálculo de μ_i	156
7.6.2.	Cálculo de \mathbf{C}	156
7.7.	Solución VBEM-PDA	156
7.7.1.	Algoritmos de tipo Asociación Probabilística de Datos (PDA)	157
7.7.2.	Algoritmo VBEM-PDA	158
7.8.	Resultados	161
7.8.1.	Comparación entre el algoritmo VBEM y el algoritmo de Gibbs Sampling	161
7.8.2.	Comparación entre el algoritmo VBEM y el VBEM con ligaduras	162
7.8.3.	Estimación de los parámetros del canal	164
7.9.	Conclusiones	165
7.10.	Trabajo futuro	165
8.	<i>Aplicación del algoritmo VBEM a Redes Regulatoras de Genes</i>	167
8.1.	Bases Biológicas	169
8.1.1.	Composición de los Ácidos Nucleicos	170
8.1.2.	ADN	170
8.1.3.	ARN	172

8.1.4. ARN heteronuclear (ARNhn)	172
8.1.5. ARN mensajero (ARNm)	172
8.1.6. ARN ribosómico (ARNr)	173
8.1.7. ARN de transferencia (ARNt)	173
8.2. Funcionamiento de los Ácidos Nucleicos	173
8.3. Técnicas de datos de Microarray	175
8.4. Formulación del Problema	178
8.5. Datos utilizados	180
8.6. Integración de datos	182
8.7. Resultados	184
8.8. Conclusiones	190
8.9. Trabajo Futuro	192
9. Conclusiones y Futuras líneas de trabajo	195
9.1. Detección Multiusuario	195
9.2. Detección y estimación de fase iterativa	196
9.3. Algoritmo VBEM aplicado Genómica	197
A. Lema de inversión de matrices	201
B. Inversa de una matriz de bloques	203
C. Cálculo de $f(x Y = y)$ cuando X e Y son vectores aleatorios Gaussianos.	205
D. Obtención de la probabilidad $p(\bar{y}_l \mathbf{b}_{n,1:k}, \mathbf{b}_{1:n-1}, \bar{y}_{1:l})$	207
E. Fórmulas para trabajar con el producto de gaussianas	209
Bibliografía	219

Índice de figuras

1.1. Sistema Inalámbrico de comunicaciones multiusuario de acceso múltiple	3
1.2. Esquemas FDMA, TDMA y CDMA	4
1.3. Ilustración de LOS y NLOS	6
2.1. Sistema de Comunicaciones Digitales	21
2.2. Código PN	24
2.3. Sistema CDMA	26
2.4. Tipos de modulación digital	28
2.5. Modulación PSK	29
2.6. Modulación FSK	30
2.7. Modulación ASK	31
2.8. Efecto conjunto de los desvanecimientos a pequeña y gran escala	35
2.9. Componente del desvanecimiento a pequeña escala	35
2.10. Resumen del fenómeno de desvanescencia	37
3.1. Esquema del proceso de Resampling	65
3.2. Esquema del Filtrado de partículas	67
4.1. Figura ilustrativa del MEETO.	81
4.2. Estructura en árbol de la solución óptima. Cada camino en el árbol representa un filtro de Kalman.	83
4.3. BERs vs. SNR para varios detectores. Igual Potencia. $\Omega = 0,03$, $f_d = 300Hz$ y 15 usuarios	92
4.4. BERs vs. SNR para varios detectores. Igual Potencia. $\Omega = 0,05$, $f_d = 500Hz$ y 15 usuarios	93
4.5. Comportamiento para varios detectores del BER vs. SNR. $\Omega = 0,03$ y 10 usuarios	94
4.6. Comportamiento para varios detectores del BER vs. SNR. $\Omega = 0,05$ y 10 usuarios	95

4.7. BERs vs. SNR para varios detectores. Usuarios con distinta potencia. $\Omega = 0,03$, $f_d = 300Hz$, 15 usuarios	96
4.8. BERs vs. SNR para varios detectores. Usuarios con distinta potencia. $\Omega = 0,05$, $f_d = 500Hz$, 15 usuarios	97
5.1. Graphical representation of a LDPC code	102
5.2. Graphical representation of g	105
5.3. Alternative graphical representation of g	105
5.4. Example SPA	108
6.1. Spectrum analyzer display of phase noise	114
6.2. Factor Graph representation of the system	117
6.3. QPSK modulation and $\sigma_v = 0,01$ model	128
6.4. BPSK modulation and $\sigma_v = 0,01$	128
6.5. BPSK modulation $\sigma_v = 0,005$ model	130
6.6. Mean Squared Error in the phase estimation for BPSK modulation and $\sigma_v = 0,01$	130
6.7. Performance of the Particle Filtering algorithm with different pilot distributions	131
6.8. Squared error in the phase estimation with the Particle Filtering algorithm using different pilot distributions	132
7.1. $\sigma_{12} = -0,9\sigma_1\sigma_2$	151
7.2. $p(z)$ correspondiente a un espacio bidimensional	152
7.3. $p(z)$ correspondiente a un espacio G -dimensional	154
7.4. Errores en los parámetros del canal para $G = 30$ cuando \mathbf{b}_3 y \mathbf{b}_2	164
8.1. Estructuras del ARN y ADN	171
8.2. Proceso de obtención de un microarray para medir concentraciones de ARNm.	176
8.3. Red de Genes Real alrededor de CDC28	181
8.4. Red de Genes Real en la zona de comprobación de daños del ADN	182
8.5. Red de Genes Real en la zona de control de giro	183
8.6. Red de Genes Estimada alrededor de CDC28 con el conjunto de datos sometidos al tratamiento con el factor de apareamiento α	186
8.7. Red de Genes Estimada en la zona de comprobación de daños del ADN con el conjunto de datos sometidos al tratamiento con el factor de apareamiento α	187

8.8. Red de Genes Estimada en la zona de control de giro con el conjunto de datos sometidos al tratamiento con el factor de apareamiento alpha	188
8.9. Red de Genes Estimada alrededor de CDC28 con el conjunto de datos sometidos al tratamiento con el mutante cdc15	189
8.10. Red de Genes Estimada en la zona de comprobación de daños del ADN con el conjunto de datos sometidos al tratamiento con el mutante cdc15	190
8.11. Red de Genes Estimada en la zona de control de giro con el conjunto de datos sometidos al tratamiento con el mutante cdc15	191
8.12. Red de Genes Estimada alrededor de CDC28	192
8.13. Red de Genes Estimada en la zona de comprobación de daños del ADN	193
8.14. Red de Genes Estimada en la zona de control de giro	194

Capítulo 1

Introducción

Aunque la teoría Bayesiana, que debe su nombre al reverendo Thomas Bayes (1701?-1761), surgió a principios del siglo XVIII para dar respuesta a problemas propios de la época, principalmente relacionados con las apuestas y los seguros, no ha sido hasta el final de la década de los años 80 y principios de los 90 cuando la posibilidad de contar con potentes medios de cálculo ha posibilitado su pleno desarrollo.

El reciente interés por la estadística Bayesiana ha dado lugar no sólo a una extensa investigación sobre su metodología Bayesiana sino también al uso de métodos Bayesianos para resolver importantes problemas con aplicaciones en diversas áreas como la astrofísica [64], el pronóstico del tiempo [39], la justicia criminal [6], las comunicaciones [25], las pólizas de asistencia sanitaria [58] o la biología [36].

Las técnicas de procesamiento estadístico de señal, que se usan para aproximar el modelado de datos, proporcionan un método de construir modelos comenzando con un modelo generador definido por una serie de parámetros desconocidos y buscando iterativamente el conjunto de parámetros del modelo que mejor se ajustan a los datos [11]. El principal objetivo del procesamiento

de señal estadístico es, por tanto, encontrar el conjunto de parámetros que definen un modelo que explica las observaciones y predice los eventos futuros de forma adecuada. Entre este tipo de técnicas se encuentran los métodos Bayesianos que proporcionan una base adecuada para seleccionar entre distintos modelos, presentando la ventaja de que permiten la incorporación del conocimiento a priori y evitan problemas de “over-fitting”. Desde un punto de vista Bayesiano, todo aquello que es desconocido se trata como variable aleatoria y se manipula usando las leyes de la teoría de probabilidad. Los métodos Bayesianos calculan la distribución a posteriori de un conjunto de modelos usando el conocimiento a priori y las observaciones. Bajo el marco Bayesiano, la información a priori sobre las variables que se desconocen se introduce en el proceso de inferencia a través de la distribución a priori. Y es precisamente gracias a la distribución a priori que la distribución a posteriori puede ser más informativa que la función de verosimilitud. Además el análisis con métodos Bayesianos es claro y directo y el aprendizaje conceptualmente simple y natural. Además, la distribución a posteriori puede incorporarse en procesos de inferencia futuros como distribución a priori. Esta metodología es la base de técnicas ampliamente utilizadas como la fusión de datos o la detección turbo. A pesar de todas las ventajas que se han expuesto, los métodos Bayesianos son computacionalmente más costosos que otros métodos basados en la función verosimilitud por eso es importante investigar y proponer nuevos métodos que sean capaces de reducir la complejidad computacional de los ya existentes.

Esta tesis aplica la Teoría Bayesiana al campo del procesado de señal. Como se ha comentado anteriormente el inconveniente de los métodos Bayesianos es su gran carga computacional. El análisis exacto de un problema desde un punto de vista Bayesiano es normalmente intratable analíticamente debido a las integrales que aparecen en este proceso. El origen de la carga computacional del proceso Bayesiano de inferencia está ligado al proceso de integración numérica. Por tanto, es necesario desarrollar nuevos métodos para realizar integraciones rápidas y lo más exactas posibles. En esta tesis se proponen nuevos métodos para aproximar el proceso de inferencia Bayesiana prestando una especial atención al problema de la complejidad computacional. En función del tipo de técnica utilizada se ha dividido este trabajo en dos partes, en la primera se utilizan técnicas Monte Carlo para resolver las integrales que aparecen en el proceso de inferencia bayesiano. En esta parte se aplican los algoritmos desarrollados al campo de las Comunicaciones. En concreto, en primer lugar se trata el problema de la detección multiusuario y en segundo lugar el problema relacionado con la detección, decodificación y estimación de fase conjunta en sistemas donde el canal introduce ruido de fase. En la segunda parte de la tesis se desarrollan métodos variacionales

Bayesianos y se adaptan y aplican al emergente campo del procesado de señal en genómica.

1.1. Comunicaciones: Detección Multiusuario

1.1.1. Planteamiento del Problema

En la actualidad, los sistemas de comunicaciones inalámbricas han de ser capaces de acomodar a varios usuarios, además de proporcionar altas velocidades de transmisión de datos. En la figura 1.1 aparece representado un típico sistema de comunicaciones multiusuario. Este tipo de sistemas representa la situación en la que varios usuarios intentan comunicarse con un único receptor a través de lo que se denomina canal. Este canal no es más que el medio a través del cual se transmite la información y sus efectos sobre la misma serán comentados más adelante.

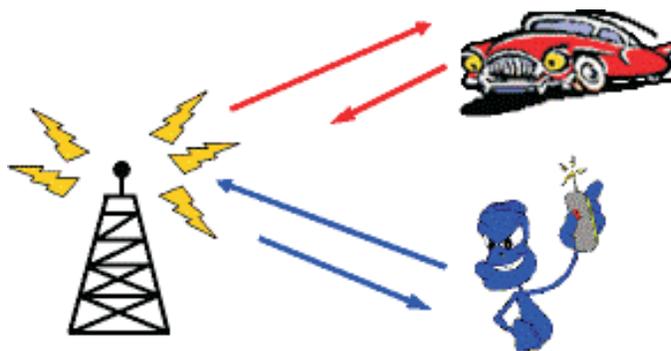


Figura 1.1: Sistema Inalámbrico de comunicaciones multiusuario de acceso múltiple

A pesar de su aparente simplicidad, el modelo aquí presentado posee la arquitectura básica de la mayoría de los sistemas de comunicaciones celulares empleados en todo el mundo. Los sistemas de comunicaciones modernos hacen posible el acceso múltiple a través de la combinación de tres métodos básicos: TDMA (*Time Division Multiple Access*), CDMA (*Code Division Multiple Access*) y FDMA (*Frequency Division Multiple Access*). Los esquemas de los métodos anteriores aparecen representados en la Figura 1.2 y en ella puede apreciarse que las técnicas TDMA y FDMA reparten los recursos de tiempo y frecuencia entre los usuarios. En CDMA, por el contrario, se

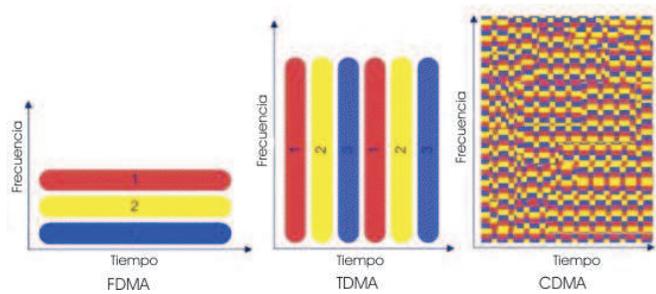


Figura 1.2: Esquemas FDMA, TDMA y CDMA

permite a todos los usuarios usar los recursos de tiempo y frecuencia de forma simultánea.

Los sistemas de comunicaciones de Segunda Generación (2G) o GSM (*Global System for Mobile Communications*) y los sistemas entre la Segunda y la Tercera Generación (2.5G) o GPRS (*General Packet Radio Service*) utilizan CDMA como esquema de acceso múltiple. Los sistemas GSM están muy limitados por la velocidad de transmisión que pueden alcanzar por lo que los principales servicios que pueden proporcionar son los de voz, fax y datos a baja velocidad (Muy por debajo de 64 kbps). GPRS es sólo una modificación de la forma de transmitir datos en una red GSM, pasando de la conmutación de circuitos en GSM (donde el circuito está permanentemente reservado mientras dure la comunicación aunque no se envíe información en un momento dado) a la conmutación de paquetes.

En los últimos años, se ha producido un desarrollo importante de la nueva generación de sistemas de comunicaciones móviles (3G) motivado por la demanda creciente de servicios avanzados de comunicaciones móviles. Los sistemas móviles 3G son la futura dirección para todas las comunicaciones móviles y permitirán una alta velocidad en la transmisión de datos y servicios multimedia en movimiento en cualquier lugar.

La tecnología 3G representa una gran oportunidad jamás vista en el mundo, incrementando el potencial de Internet Móvil. Los consumidores ahora exigen más servicios y el funcionamiento mejorado de los mismos, el mundo móvil debe desarrollarse para dirigir estas exigencias hacia el logro de equipos con capacidades multimedia, acceso al mundo entero para servicios personalizados de banda ancha y fácil acceso directo a Internet Móvil.

Los sistemas 3G tienen el nombre de UMTS (*Universal Mobile Telecommunication System*) en Europa, y en EE.UU. y Japón IMT-2000 (*International Mobile Telecommunication by the year 2000*). La tecnología UMTS permite transmitir datos a alta velocidad. La conexión a Internet alcanza hasta 384 Kbps, seis veces más rápido que con los terminales GPRS ac-

tuales. Esta velocidad de transmisión permite que el usuario acceda a servicios multimedia interactivos y a nuevas aplicaciones desde el móvil como la videotelefonía. Por tanto, es muy importante ser capaz de realizar un buen procesado en sistemas basados en CDMA.

CDMA permite que todos los usuarios utilicen todos los recursos de tiempo y frecuencia a las vez. Los usuarios se distinguen unos de los otros modulando su información a través de una secuencia que es específica para cada uno de ellos. Si estas secuencias se eligen de forma que sean ortogonales unas con las otras, cada usuario puede ser separado de los demás en el receptor que es común para todos. Sin embargo, esta ortogonalidad es imposible de mantener en las comunicaciones inalámbricas reales y, por tanto, la señal que cada usuario envía aparece afectada por lo que se denomina IAM (*Interferencia de Acceso Multiple*). La Detección Multiusuario que no es otra cosa más que la recuperación de la señal enviada por cada usuario en presencia de IAM, se convierte así en un punto clave para el funcionamiento eficiente de un sistema CDMA.

Existe otro factor que influye y modifica la señal enviada por cada uno de los usuarios y que hay que tener en cuenta en el proceso de Detección Multiusuario: el Canal de Comunicaciones. Dependiendo de las características del mismo la señal emitida por cada usuario se verá afectada de una u otra forma. Uno de los fenómenos más importantes a tener en cuenta es el desvanecimiento que afecta de forma drástica el rendimiento de los sistemas de comunicaciones digitales pudiendo dar lugar a pérdidas de datos o de llamadas a teléfonos móviles. Un canal se considera desvanescente si varía en el tiempo y en la frecuencia. También se puede hablar de canales desvanescentes cuando los parámetros que lo describen son variables estocásticas que pertenecen a ciertas distribuciones.

Un concepto importante relacionado con lo anterior es el del multitrayecto. Esta cuestión surge del hecho de que en las propagaciones inalámbricas, la señal transmitida llega al receptor desde diferentes direcciones que determinan una serie de caminos. Estos múltiples caminos tienen su origen en la existencia de diversos obstáculos y reflectores. Por tanto, el llamado multitrayecto no es más que un conjunto impredecible de reflexiones y/o ondas que llegan de forma directa cada una de ellas con su propio grado de atenuación y retardo. Por lo general, para describir el canal multitrayecto se suelen utilizar:

1. LOS (*Line-of-sight*) que representa la conexión directa entre el transmisor y el receptor.
2. NLOS (*Non-line-of-sight*) que representa el camino seguido tras una o varias reflexiones.

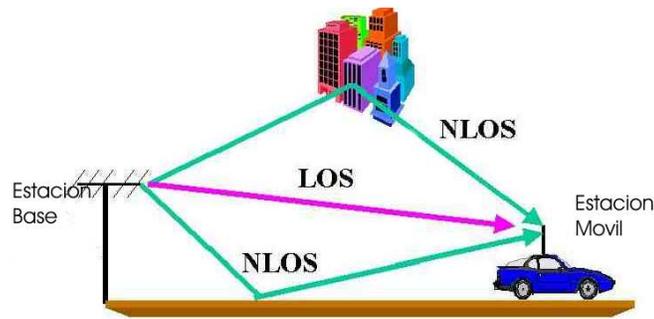


Figura 1.3: Ilustración de LOS y NLOS

En la figura 1.3 se muestra un esquema para clarificar los conceptos anteriores.

Los problemas asociados con el multitrayecto son básicamente fluctuaciones de amplitud y fase en las señales recibidas. Estas fluctuaciones, que pueden combatirse haciendo uso de ciertos esquemas de diversidad, dan lugar a dos importantes fenómenos:

- **Desvanecimiento Multitrayecto.** Se produce cuando las ondas de las señales multitrayecto no están en fase, de forma que la intensidad de la señal multitrayecto se reduce en el receptor. Este fenómeno se denomina *Desvanecimiento tipo Rayleigh* o *Desvanecimiento Rápido*.
- **Ensanchamiento en Retardos Multitrayecto.** Debido a las reflexiones que puedan producirse, las señales transmitidas llegan al receptor a diferentes tiempos y este fenómeno puede tener como resultado lo que se conoce con el nombre de IIS (*Interferencia Intersimbólica*). La IIS no es más que la interferencia entre los símbolos de señales enviadas por un usuario dado en distintos instantes de tiempo pero que llegan al receptor a la vez debido a los retardos producidos por las reflexiones. A veces, el receptor es incapaz de evitar la IIS. Esta dispersión temporal causada por el canal es un parámetro importante a tener en cuenta en lo relacionado con el rendimiento de los sistemas inalámbricos. Normalmente, la medida del Ensanchamiento en Retardos Multitrayecto se hace a través de la raíz cuadrada media del ensanchamiento en retardos (rms). Para comunicaciones fiables sin utilizar ecualización adaptativa u otras técnicas que eliminen el multitrayecto, la velocidad de transmisión de datos debe ser mucho más pequeña que la inversa del rms (a este valor se le denomina ancho de banda de coherencia). Cuando la velocidad de transmisión de datos es mucho menor que el ancho de banda

de coherencia, el canal inalámbrico se denomina Canal Plano o Canal de Ancho de Banda Estrecho. Cuando la velocidad de transmisión de datos es parecida o mayor que el ancho de banda de coherencia, el canal se denomina Canal Selectivo en Frecuencia o Canal de Banda Ancha.

En relación al tratamiento del canal dentro de la Detección Multiusuario existen dos grandes métodos. Por un lado están los llamados Algoritmos de Detección Multiusuario Semiciegos que son aquellos que utilizan lo que se denomina señales piloto o secuencias de entrenamiento. Estas señales son conocidas en el receptor y se utilizan para hacer una estimación previa del canal para poder realizar posteriormente una detección del resto de la señal enviada por cada uno de los usuarios. En segundo lugar se encuentran los Algoritmos de Detección Multiusuario Ciegos que son aquellos en los que no se dispone de ningún tipo de secuencia de la señal transmitida. Dentro de este último grupo se encuentran enmarcados los algoritmos que se presentan en esta memoria.

Otro problema importante en comunicaciones inalámbricas es el llamado “near-far” [28], [55], [83], [81], [27], [71]. Cuando un usuario está lejos de la estación base, es probable que su señal se vea sobreafectada por los usuarios que se encuentran más cerca. Generalmente, este problema se soluciona mediante lo que se denomina control de potencia. Este proceso requiere un complicado sistema de control tanto en la estación base como en los móviles. Los usuarios más lejanos consumen normalmente gran cantidad de potencia, lo que inevitablemente disminuye la vida de las baterías de los dispositivos móviles además de dañar los amplificadores.

La primera parte de esta tesis se centra en el análisis de detectores multiusuario ciegos óptimos y casi óptimos con complejidad computacional aceptable en los que el canal se considera plano desvanescente rápido de tipo Rayleigh y la transmisión síncrona. Además se consideran tanto situaciones donde los usuarios tienen la misma potencia como escenarios near-far en que los usuarios tienen distinta potencia.

1.1.2. Planteamientos anteriores

Desde que la detección multiusuario (DMU) se introdujo en los ochenta, ésta ha recibido mucha atención debido a su habilidad para reducir la Interferencia de Acceso Múltiple (IAM) y su potencial para aumentar la capacidad de los sistemas CDMA (*Code Division Multiple Access*). Desde entonces, se han propuesto gran cantidad de detectores en la literatura tanto para transmisión síncrona como asíncrona. Algunos de los más populares son

el detector decorrelador, el de mínimo error cuadrático medio (MMSE), el detector multi-etapa, y el detector decision feedback [84].

En la práctica, la distorsión en la intensidad de la señal debido a los canales dependientes del tiempo y desvanescentes debe tenerse en cuenta cuando se realiza detección multiusuario. A pesar de que los métodos de detección no coherentes como los propuestos en [1] son a veces apetecibles debido a su simplicidad ya que no se necesita ningún tipo de conocimiento sobre el canal desvanescente, la detección coherente ha demostrado mejor rendimiento [89]. En detección coherente, la estimación de los canales puede realizarse con o sin señales de entrenamiento. Se ha realizado gran cantidad de investigación en relación a esquemas que no usan señales de entrenamiento. Son los llamados métodos DMU ciegos. Este tipo de métodos son más eficientes y las aproximaciones que se han propuesto hasta ahora, por nombrar algunas, incluyen las recursivas de mínimos cuadrados (RMC) [43, 51], las basadas en subespacios [86], las de maximización del valor esperado [32], los de algoritmos genéticos [93] y los de filtrado de Kalman [34, 53, 61, 62, 82, 90]. Sin embargo, la mayoría de las aproximaciones citadas anteriormente asumen canales desvanescentes casi estáticos o que varían muy lentamente.

La primera parte de esta tesis se centra en detección multiusuario ciega en canales planos desvanescentes rápidos de tipo Rayleigh donde la transmisión se realiza de forma síncrona. En particular, se asume conocida *a priori* la estadística de segundo orden del canal. Basada en ésta, se adopta una aproximación matemática en la que se usa un modelo ARMA. Esta aproximación permite usar un modelo de espacio de estados dinámico (MEED) lo que permite usar un filtrado de Kalman como solución. El uso del filtrado de Kalman para DMU ciega se ha visto en [34, 82, 90], donde la aproximación dirigida por decisión se usaba para estimar las variables del canal necesarias para el filtrado de Kalman. Un inconveniente inherente a la aproximación dirigida por decisión es la propagación del error, que limita de gran forma el rendimiento de las implementaciones realizadas con esta aproximación.

Recientemente se han aplicado el filtrado de Kalman de mezcla y los algoritmos de muestreo de importancia secuencial (filtrado de partículas) a detección ciega de códigos convolucionales [19], códigos space-time trellis [96], e incluso detección ciega multiusuario [72] en canales desvanescentes.

El uso de filtrado de Kalman mezcla (FKM) ha demostrado poder reducir bastante el error de propagación de implementaciones dirigidas por decisión y, por tanto, permite la mejora del rendimiento. Sin embargo, en el uso de FKM para el caso de detección multiusuario ciega de [72], el filtrado de partículas se usaba principalmente para seguimiento del canal y el proceso de detección multiusuario se realiza con un detector óptimo que tiene complejidad exponencial con el número de usuarios. Por tanto, el algoritmo FKM propuesto

es muy complejo incluso para un número moderado de usuarios.

En esta tesis, a diferencia de todos los detectores basados en filtrado de Kalman que existen, se adopta un punto de vista totalmente diferente de los sistemas multiusuario y se utiliza un nuevo modelo llamado Modelo de Espacio de Estados de Observación Temporal (MEETO) que fue propuesto en [94].

1.1.3. Aportaciones del Trabajo

La primer parte de esta tesis se centra en el desarrollo y análisis de un algoritmo para la DMU ciega en canales planos de tipo Rayleigh con desvanecimiento rápido. Este algoritmo denominado *M*-Estocástico, está basado en el método de Filtrado de Partículas y en comparación con otros algoritmos que no son del tipo de Filtrado de Partículas, el algoritmo propuesto en esta memoria (Algoritmo *M*-Estocástico (AME)) tiene la ventaja de proporcionar información estadística relacionada con la probabilidad. Esta información permite mantener y estudiar varias trayectorias, varias posibilidades en lugar de centrarse en una sola que ha de ser elegida en cada paso. Además, tiene la ventaja de que cada trayectoria lleva asociado un peso que indica la importancia de la trayectoria en la decisión final. Por tanto, pueden mantenerse trayectorias que en un primer momento no tengan mucho peso, pero que lleguen a tener una importancia determinante en etapas posteriores. Con otro tipo de algoritmos en los que no se cuenta con esta información estadística, estas trayectorias serían eliminadas en un primer momento dando lugar a errores. Con el algoritmo propuesto, gracias a su naturaleza, estas trayectorias se mantienen. Por tanto, se evitan errores en decisiones futuras. Este es uno de los principales problemas de los algoritmos dirigidos por decisión por ejemplo.

Uno de los puntos importantes relacionados con los algoritmos de detección en general es la de la complejidad computacional. De modo que el trabajar con un algoritmo de excesiva complejidad computacional puede hacer que el mismo no sea aplicable en situaciones reales. En general la complejidad de los algoritmos de filtrado de partículas en general (y la del algoritmo propuesto en esta memoria en particular) es mucho menor que la de la solución MAP (*Maximum a Posteriori*). Así, el algoritmo que se propone en este trabajo presenta un rendimiento muy cercano al óptimo ya que logra una aproximación eficiente a la distribución posterior verdadera.

Y lo que es más, el AME ha demostrado una reducción adicional de la complejidad en comparación con el algoritmo clásico de filtrado de partículas manteniendo un rendimiento similar [95], [66], [67].

Una característica importante en el problema de detección multiusuario

al que se le aplican técnicas de Filtrado de Partículas es el de mantener la diversidad en las trayectorias. Esta diversidad es garantía de buen rendimiento. Los algoritmos clásicos de Filtrado de Partículas sufren degradación con el tiempo, además, cuando trabajan en espacios discretos no son capaces de mantener la diversidad pudiéndose llegar a situaciones en las que se tengan varias trayectorias que sean copia de una dada. Así, otra importante aportación del AME reside en su capacidad para mantener diversidad en las trayectorias. Este hecho está relacionado con el modo de hacer el denominado “resamplin”. Por todo ello, al AME se convierte en un algoritmo mucho más eficiente que los clásicos de tipo Filtrado de Partículas y, por tanto, mucho más adecuado para aplicaciones reales.

Otra cuestión importante en este trabajo está relacionada con el uso del modelo MEETO. Aunque el MEETO es equivalente al MEED, tiene la ventaja de que permite que el detector multiusuario basado en filtrado de partículas se combine de forma natural con una estimación del canal basada en filtros de Kalman de mezcla. Otra de las grandes ventajas de MEETO es que el muestreo se realiza para un usuario para cada instante de tiempo de forma que se consigue una implementación más eficiente. Además, se evita la propagación de errores relacionada con el empleo de un modelo MEED y que constituye una de sus principales limitaciones.

Como aportación final se tiene que el AME puede usarse en escenarios con “near-far” sin que esta circunstancia afecte en absoluto a su rendimiento. Así se evita tener una carga computacional adicional dedicada al control de potencia.

1.2. Comunicaciones: Detección, decodificación y estimación de fase conjunta

1.2.1. Planteamiento del problema

Los códigos LDPC (Low Density Parity Check) fueron introducidos por Gallager en 1963. Se demostró entonces que estos códigos poseían un rendimiento muy bueno cuando se utiliza un esquema de decodificación iterativo especialmente cuando se trata de códigos con una longitud larga. Pero la falta de capacidad de cálculo de la época hizo que estos códigos fueran rápidamente olvidados. Unos treinta años después los códigos LDPC fueron redescubiertos y actualmente se utilizan en diversos campos debido a que se ha demostrado que son mejores que los códigos turbo. A este hecho contribuyó el descubrimiento de que pueden llegar a conseguir un rendimiento en decodificación de sólo 0.0045 dB por encima del límite de Shannon.

Básicamente, un código LDPC es un código lineal cuya matriz de control de la paridad es una matriz poco densa y además los elementos que no son cero se distribuyen de forma aleatoria. Estos códigos pueden representarse haciendo uso de gráficos, en particular, las representaciones Factor Graph son especialmente útiles. Además, la decodificación puede realizarse mediante un esquema de transmisión de mensajes en la representación Factor Graph denominado Algoritmo de Suma Producto (SPA) [57].

Esta tesis se centra en el estudio de códigos LDPC en presencia de ruido de fase. En concreto, se pretende realizar una detección y estimación de fase conjunta cuando estos códigos se transmiten a través de un canal con ruido de fase. El ruido de fase está fundamentalmente relacionado con los osciladores y está constituido por fluctuaciones aleatorias y rápidas en la señal causadas por inestabilidades en el dominio del tiempo.

1.2.2. Aportaciones anteriores

Muchos algoritmos iterativos Bayesianos se han aplicado al proceso de decodificación en modelos con un canal que tiene fase desconocida bien constante o bien dependiente del tiempo. Aunque los algoritmos no Bayesianos son eficientes cuando la fase es constante, su rendimiento empeora en presencia de fases que varía con el tiempo. El algoritmo no Bayesiano más popular aplicado al problema en que la fase del canal es desconocida es el algoritmo EM (Expectation-Maximization) [65]. Los métodos no Bayesianos son apropiados para estimar parámetros, pero no para estimar señales aleatorias. Para adaptar el algoritmo a la dependencia con el tiempo de la fase del canal, se modifica el algoritmo EM mediante una ventana deslizante optimizada [22]. Otra posibilidad es encontrar una parametrización eficiente del proceso de ruido de fase de forma que una realización arbitraria de la fase puede expresarse a través de un pequeño número de variables deterministas desconocidas. Una solución de este tipo puede ser la basada en una expansión de Karhunen-Loeve combinada con el algoritmo EM que se propone en [22]. Un algoritmo Bayesiano se propuso también en [22] como resultado de la aplicación del algoritmo de Suma-Producto a una representación Factor Graph de la distribución a posteriori de los bits de información y los parámetros del canal dada la salida del canal. Como el SPA aplicado a variables continuas aleatorias involucra integración y cálculo de pdfs continuas, en [21] se propone el uso de distribuciones canónicas para solucionar el problema. Se usan entonces una discretización del espacio de parámetros, una parametrización de Tikhonov, una de Fourier y una parametrización gaussiana.

1.2.3. Aportaciones del trabajo

Motivados por el mal rendimiento de los detectores no Bayesianos cuando la fase del canal depende del tiempo, en esta tesis se estudia la decodificación iterativa en canales con ruido de fase que depende del tiempo mediante el uso de métodos de Filtrado de Partículas. Estos métodos Bayesianos se utilizan para realizar una estimación de la fase “on-line”. Los filtros de partículas son herramientas potentes para calcular numéricamente los estimadores óptimos cuando las soluciones exactas no pueden derivarse analíticamente. Aplicados a la estimación de fase, los filtros de partículas permiten una exploración continua del espacio de fases a lo largo del tiempo.

En concreto, se propone combinar el algoritmo SPA con un algoritmo de Filtrado de partículas. En [88], los autores proponen discretizar el espacio de las fases utilizando un conjunto equiespaciado de valores. El rendimiento de este método es prácticamente óptimo cuando el número de valores del conjunto aumenta, sin embargo su principal limitación es su carga computacional excesiva. El método de filtrado de partículas propuesto consigue reducir la carga computacional y mantener e incluso en algunos casos mejorar el comportamiento del algoritmo equiespaciado ya que evita una exploración de todos los posibles estados.

Los algoritmos de filtrado de partículas se han aplicado con anterioridad a este problema [23]. Sin embargo, el número de partículas a usar en ese primer algoritmo depende de la longitud del código. En esta tesis, se propone un algoritmo de filtrado de Partículas cuyo número de partículas no depende de la longitud del código y, por tanto, puede utilizarse para trabajar con códigos largos sin inducir una excesiva carga computacional.

Una de los principales problemas de los algoritmos de Filtrado de Partículas es su rápida degeneración. Para reducir en la medida de lo posible este fenómeno se introduce un paso de “resampling” en el algoritmo. Este paso no aparece en [23] y, como resultado, el número de iteraciones para estimar la fase es grande. En el algoritmo de Filtrado de Partículas propuesto en esta tesis se incluye un paso de resampling con lo que se reduce el número de iteraciones.

1.3. Genómica

1.3.1. Planteamiento del Problema

El objetivo fundamental de la biología es comprender cómo funcionan las células vivas. El campo de la biología molecular analiza el funcionamiento de las células y los procesos de herencia principalmente en términos de tres im-

portantes clases de macromoléculas: ADN, ARN y proteínas. Las proteínas son moléculas que hacen posible la ejecución de la mayoría de los procesos en la célula. El ADN es el portador de la información hereditaria en forma de genes y dirige el proceso de producción de proteínas. Al ARN es un intermediario clave entre el ADN y las proteínas. Tanto la biología molecular como la genética están experimentando cambios revolucionarios guiados por la visión de la célula como una colección de subsistemas interrelacionados cada uno de ellos influenciado por las interacciones entre genes y proteínas. Por tanto el interés ha pasado de centrarse en el estudio de genes o proteínas individuales a la exploración del genoma completo de un organismo y al estudio de las Redes de Genes y proteínas. Conforme aumenta el conocimiento a través de la producción de más y más cantidad de datos, este campo se convierte en más y más dependiente de tareas como el modelado matemático, análisis matemático y computación.

Por tanto, uno de los campos que está sufriendo un desarrollo más espectacular es el área de la genética concerniente al estudio del genoma: la genómica. Se denomina perfil de expresión de un gen a la información resultante del proceso mediante el cual la secuencia de ADN de un gen da lugar a ciertas estructuras o funciones dentro de la célula. De las señales procedentes del perfil de expresión de los genes puede extraerse información para explicar fenotipos específicos de las células (enfermedades) o clasificar enfermedades a nivel molecular, de ahí su importancia.

Actualmente es fácil acceder a la mayoría de la información relacionada con secuenciación de genes que se está realizando. Pero el problema es que sólo unos pocos de los conjuntos de genes identificados en esta secuenciación poseen elementos asociados con una funcionalidad conocida. Una forma de tener una visión más amplia sobre el papel de un determinado gen en la actividad celular es estudiar su perfil de expresión en diferentes circunstancias y contextos que básicamente es ver cómo responde a su ambiente y a la acción de otros genes. Recientemente, los nuevos métodos han proporcionado estudios a gran escala sobre las expresiones de los genes en los que los niveles de transcripción pueden determinarse para miles de genes a la vez. En particular, los microarrays de expresión son el resultados de un sistema complejo bioquímico-óptico que incorpora observación robótica y análisis de imágenes por ordenador. Ya que el proceso de control del proceso de transcripción se lleva a cabo con un método que interpreta una serie de entradas, es necesario usar herramientas analíticas para tratar los perfiles de expresión que pueden detectar los tipos de influencias en el proceso de decisión realizado por una red de genes. Las señales generadas por el genoma han de ser procesadas para caracterizar sus efectos reguladores así como su relación con los cambios tanto a nivel de fenotipo como de genotipo. El procesado

de señal aplicado a Genómica (PSG) es una disciplina emergente dentro de la ingeniería que procesa las señales relacionadas con los genes. El objetivo del procesado de señal aplicado a genómica es la integración de las teorías y los métodos del procesado de señal con el conocimiento sobre genómica funcional, poniendo especial énfasis en la regulación de genes. Por tanto, este tipo de procesado de señal abarca varias metodologías relacionadas con los perfiles de expresión: detección, predicción, clasificación, control, y modelado estadístico y dinámico de las redes de genes.

Los dos temas principales del PSG son la clasificación del fenotipo y las redes reguladoras de genes. Otros campos como el procesado de imagen, la normalización de datos, la cuantización, la compresión o los métodos de procesado de señal aplicados al análisis de secuencias juegan también papeles importantes y necesarios.

Un clasificador basado en perfiles de expresión proporciona un conjunto de genes cuya abundancia de producto es indicativa de importantes diferencias en el estado de la célula como enfermedad y salud, un tipo de cáncer u otro, etc. Especialmente, se busca clasificar esos genes en función del papel que juegan sus productos en la iniciación, progresión y mantenimiento de una enfermedad. Una vez conocidos estos conjuntos de genes y su función el objetivo es usarlos para diagnosticar enfermedades y producir tratamientos. Sin embargo existen tres problemas importantes a tratar desde un punto de vista estadístico en relación a este proceso de clasificación. En primer lugar están los problemas de diseño relacionados con la elección de un clasificador óptimo para un población basándose en un conjunto de muestras particular. El segundo problema está relacionado con el cálculo del error de ese estimador que se ha seleccionado cuando los datos con los que se cuenta son limitados. Finalmente, el último problema está relacionado con el proceso de selección del conjunto de variables que se han de usar para realizar el proceso de clasificación. Algunas técnicas de clasificación se han usado para analizar datos de perfiles de expresión como “K-means”, mapas de autoorganización, etc. Las principales dificultades relacionadas con la aplicación de estos métodos son la elección de parámetros, de condiciones iniciales o medida de distancias.

Las Redes de Genes son muestras de las relaciones moleculares a nivel genómico. Estas interacciones son las manifestaciones de la regulación entre genes, las interacciones entre proteínas y las asociaciones entre las proteínas y los metabolitos. La comprensión de las interacciones moleculares es de gran importancia ya que muchas funciones celulares complejas están controladas a través de estas relaciones. Esencialmente, las Redes de Genes infieren funciones a las aparentemente aisladas secuencias de genes. Como resultado, descubrir las Redes de Genes puede sugerir nuevas interpretaciones o cuestiones sobre los experimentos, puede proporcionar una comprensión más

profunda de nuevos blancos de medicamentos, y puede guiar hacia leyes para tratar enfermedades complejas así como conceptos importantes a la hora de diseñar nuevos medicamentos.

Aunque los experimentos son indispensables, los cálculos han jugado un papel cuya importancia está creciendo a la hora de descubrir Redes de Genes. Este cambio en el paradigma es debido a la proliferación de grandes cantidades de datos que proporcionan información bien directa o indirecta sobre las interacciones moleculares desde perspectivas diferentes. Mediante la extracción y la unión de evidencias relevantes de los datos utilizando herramientas computacionales, puede lucharse para reconstruir las Redes de Genes de forma eficiente y efectiva.

1.3.2. Aportaciones anteriores

Esta tesis se centra en una aproximación Bayesiana para reconstruir Redes de Genes basándose en datos de microarray. Los datos de microarray proporcionan medidas de primera mano sobre la expresión de los genes, producciones moleculares de las interacciones moleculares y, por tanto, los datos de microarray constituyen una fuente natural para el descubrimiento de las Redes de Genes. Sin embargo, la tarea de reconstruir redes de genes es extremadamente complicada debido a la gran cantidad de parámetros desconocidos, el pequeño tamaño del conjunto de muestras, los defectos experimentales inherentes y muchos otros factores. Estas dificultades hacen que sea necesaria una fuerte implicación del procesado estadístico de señal incluyendo, naturalmente, a los métodos Bayesianos.

En la literatura se han desarrollado distintas soluciones Bayesianas a la reconstrucción de Redes de Genes [35,36,46,49,54,70,74]. Estas aportaciones pueden clasificarse aproximadamente en dos grupos de soluciones puntuales (“hard”) y probabilísticas (“soft”). La finalidad de las soluciones puntuales es encontrar un único punto que estime la red a través de un proceso de optimización fundamentalmente. Las técnicas en esta categoría incluyen la búsqueda paso a paso (“step-wise search”), búsquedas de tipo “greedy” y las “simulated annealing”. Por otro lado las llamadas soluciones blandas tienen como objetivo la estimación de las *probabilidades a posteriori* (APPs) de las redes y los métodos dentro de esta categoría incluyen al muestreo Monte Carlo y las técnicas Variacionales Bayesianas. En la actualidad, la mayoría de los usos de marcos Bayesianos en este campo de investigación están constituidos por métodos puntuales. A pesar de las muchas ventajas de las soluciones “duras” incluyendo especialmente su relativamente baja complejidad, las notables propiedades de los métodos blandos son mucho más atractivas. En primer lugar, las soluciones blandas proporcionan estimadores de las APPs

lo que constituye a su vez una medida directa de la confianza o certidumbre en la solución. En segundo lugar, la información probabilística obtenida es esencial para la integración Bayesiana de diferentes conjuntos de datos procedentes de la misma y/o de diferentes fuentes, una tarea importantísima en la investigación sobre Redes de Genes que no puede llevarse a cabo con soluciones duras.

Las soluciones blandas Bayesianas constituyen el principal interés de esta tesis. Las soluciones blandas existentes para construir redes de genes incluyen, el muestreo Monte Carlo (“Gibbs sampling” [49] y el muestreo basado en técnicas “reversible jump MCMC” [45]) y “Variational Bayes EM” (VBEM) [12]. Los algoritmos basados en el muestreo Monte Carlos necesitan de un número grande de muestras para garantizar exactitud así como la realización de operaciones robustas. Es esto precisamente lo que convierte a estos algoritmos en computacionalmente prohibitivos cuando se trabaja con redes grandes. Por otro lado, la solución VBEM propuesta en [12] no tiene en cuenta explícitamente la indeterminación en la topología de la red en su modelado. El problema se formula como una estimación de parámetros y la topología se determina midiendo la cercanía de los pesos reguladores estimados a cero. Aunque este tratamiento puede reducir la complejidad del proceso de aprendizaje, no refleja de forma fiel la naturaleza del problema. Esto motiva la investigación de soluciones VBEM para realizar conjuntamente un aprendizaje de la topología y estimar los parámetros del modelo.

1.3.3. Aportaciones del trabajo

En esta tesis se desarrolla un nuevo algoritmo variacional en un marco Bayesiano basado en una solución tipo EM (Expectation Maximization). Se trata de un método Bayesiano que proporciona información estadística a través de las APPs por lo que posee la ventaja de proporcionar información a cerca de la certidumbre de los estimadores que calcula. Además esta información estadística puede utilizarse para integrar información procedente de distintos conjuntos de datos, aspecto este fundamental en el campo en el que se trabaja ya que se cuenta con muestras de pequeño tamaño y cualquier información adicional es relevante. La principal novedad del algoritmo propuesto está precisamente en que la formulación ya que el proceso de aprendizaje de la topología de una Red de Genes se trata como un problema de selección de variables, en el que la topología de la red se determina seleccionando un conjunto de predictores (datos temporales de microarray). Por tanto, se desarrolla un nuevo esquema VBEM para realizar conjuntamente la estimación de parámetros continuos y la variable indicadora binaria. Este nuevo tratamiento es más completo y apropiado que el propuesto en [12].

En comparación con otros algoritmos Bayesianos que también proporcionan información estadística, algoritmo de Gibbs Sampling, la solución VBEM propuesta en esta tesis ha mostrado una complejidad mucho menor y mejor rendimiento en el proceso de inferencia.

Finalmente añadir que debido precisamente a la naturaleza del algoritmo y al hecho de que proporciona información estadística ha sido posible desarrollar un esquema de integración Bayesiano para combinar dos conjuntos de datos de microarray diferentes aprovechando las probabilidades a posteriori (APPS) de la topología obtenidas en la solución VBEM. Este esquema proporciona un marco viable y eficiente para realizar integración de datos Bayesiana.

1.4. Organización

Como ya se ha comentado, esta tesis se divide en dos partes diferenciadas en función del tipo de algoritmos Bayesianos empleados. En la primera parte se utilizan algoritmos de tipo Filtrado de Partículas y se desarrollan aplicaciones a Comunicaciones. La segunda parte está dedicada a algoritmos Bayesianos variacionales y se desarrolla una aplicación del algoritmo al campo del procesamiento de señal en Genómica.

Los capítulos de la primera parte de esta tesis tienen la siguiente estructura: En el Capítulo 2 se realiza una introducción a los sistemas de comunicaciones digitales y se hace especial hincapié en el estudio del canal, de su modelado y de los distintos efectos del mismo sobre una señal dada. El Capítulo 3 está dedicado al Filtrado Bayesiano que constituye el fundamento del algoritmo propuesto y de las principales soluciones existentes al problema. Se comienza con un estudio de la Teoría de Bayes y los Métodos Secuenciales de Monte Carlo para seguir con una presentación del Filtrado de Kalman y del Filtrado de Partículas. En el final de este capítulo se introduce el Algoritmo M Estocástico y el Filtrado de Kalman de Mezcla. En el Capítulo 4 se presenta el problema al que se van a aplicar los distintos algoritmos presentados en el Capítulo 3. Se comienza con la introducción del modelo que se aplica al canal para seguir con un estudio de los detectores más importantes que se van a comparar la propuesta del algoritmo M -Estocástico y se finaliza con una muestra de los resultados obtenidos en simulaciones. En el Capítulo 5 se realiza una introducción a los códigos LDPC (Low Parity Check Codes) y a las técnicas “Machine Learning”, en concreto se presenta la representación “Factor Graph” y se desarrolla el algoritmo Suma Producto. En el Capítulo 6 se resuelve el problema de la detección conjunta y estimación del ruido de fase mediante la aplicación de técnicas de Filtrado de Partículas cuando se

representa la probabilidad conjunta a posteriori de símbolos y fase mediante la técnica “Factor Graph” y se aplica el algoritmo Suma Producto.

Los capítulos de la segunda parte de esta tesis se organizan de la siguiente forma: En el Capítulo 7 se desarrolla un algoritmo Variacional Bayesiano aplicado al problema de selección de variables. Este algoritmo es capaz de detectar los símbolos estimados así como los parámetros del modelo que representa al sistema que se estudia proporcionando además información estadística sobre estos estimadores. En el Capítulo 8 se aplica el algoritmo desarrollado en el Capítulo 7 al problema de estimación de Redes de Genes.

Finalmente, en el Capítulo 9 se presenta un resumen de las conclusiones más relevantes y las líneas de trabajo futuro.

Parte I

Algoritmos Bayesianos Monte Carlo aplicados a Comunicaciones

Capítulo 2

Comunicaciones Digitales

En el diseño de cualquier sistema de comunicaciones, no sólo hay que tener en cuenta las características de las fuentes que generan la señal sino que también hay que considerar los efectos de las características del canal así como la configuración del receptor.

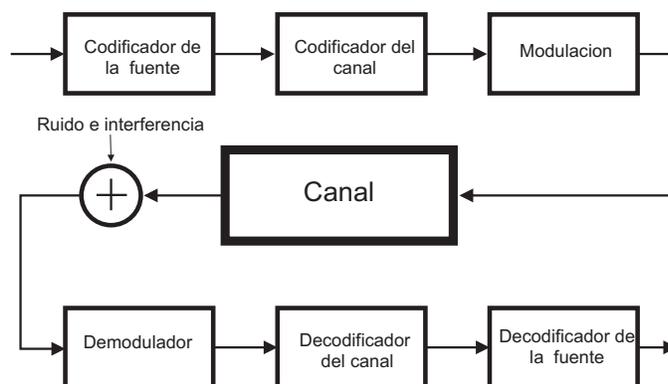


Figura 2.1: Sistema de Comunicaciones Digitales

En este capítulo se realiza un estudio de sistemas de Comunicaciones digitales. Estos sistemas de Comunicaciones se caracterizan porque la información (analógica o digital) que se genera mediante una fuente o un conjunto de ellas, se procesa de forma digital. En la Figura 2.1 se muestra un esquema de este tipo de sistemas cuyas partes se analizan a lo largo de todo el capítulo. En concreto, en la Sección 2.1 se tratan los distintos tipos de procesos de Codificación de la fuente. En la Sección 2.2 se habla de la Codificación de la Señal haciendo un especial hincapié en el proceso de codificación de la señal en sistemas DS-CDMA que son los que se tratan en la primera aplicación a Comunicaciones. En la Sección 2.3 se presenta brevemente el concepto de interespaciado. Uno de los conceptos importantes en Comunicaciones es el de la Modulación de la señal, en esta tesis se han utilizado distintos tipos de modulación y todos ellos junto con sus características aparecen presentados en la Sección 2.4. Un elemento clave en Comunicaciones es el canal a través del cual se transmite la señal. En la Sección 2.5 se presenta un estudio de distintos tipos de canales así como de los efectos que producen en la señal. Finalmente, en las Secciones 2.6 y 2.7 se definen dos parámetros importantes a través de los cuales se estudia el rendimiento de los algoritmos que se presentan en esta tesis: el BER (Bit Error Rate) y la SNR (Signal to Noise Ratio).

2.1. Codificación de la fuente

La codificación en la fuente se conoce también con el nombre de cuantización y es una forma de procesar los datos para reducir la redundancia o prepararlos para un procesamiento posterior. La conversión analógica-digital o la compresión de datos son dos tipos de codificación.

Una buena compresión en la fuente contribuye a un buen rendimiento del sistema, sin embargo tiene la desventaja de que altos niveles de compresión pueden dar lugar a pérdidas de información. Por tanto, hay que buscar un equilibrio entre un nivel de compresión aceptable y una pérdida de información baja.

La codificación en la fuente se divide en dos procedimientos básicos: encriptado y desencriptado. El encriptado convierte a la señal de la fuente en una señal digital haciendo uso de un método de cuantización. Los símbolos de la señal resultante son enteros no negativos en cierto rango finito. El desencriptado recupera la información original de la señal codificada de la fuente.

2.1.1. Codificación Diferencial

En parte de Comunicaciones de esta tesis, el tipo de codificación que se ha empleado ha sido Codificación Diferencial. En lugar de encriptar la señal directamente, la técnica de codificación diferencial codifica la diferencia entre la señal y la predicción de la misma. Es por esto que se la conoce también como codificación predictiva. Se trata, por tanto, de un método eficiente y además simple desde un punto de vista computacional.

2.2. Codificación de la Señal

Para asegurar un buen diseño del sistema digital, es necesario comenzar codificando la señal. Para ello, cada secuencia de información pasa en primer lugar por un codificador del canal que introduce de manera controlada cierta estructura cuidadosamente diseñada para proteger los datos de errores de transmisión. Esta redundancia añadida permite que el sistema receptor sea capaz de detectar y posiblemente corregir errores debidos a la corrupción introducida por el canal. Con todo ello se mejoran la precisión y fiabilidad del sistema en presencia de ruido e interferencias.

Existen distintos tipos de códigos:

- **Códigos sistemáticos.** Son aquellos códigos en los que la palabra de información aparece de forma explícita en la palabra codificada.
- **Códigos no sistemáticos.** Son aquellos códigos en los que la palabra de información no aparece de forma explícita en la palabra codificada.
- **Códigos de bloque.** Se trata de aquellos códigos en los que todas las palabras tienen la misma longitud y la codificación se hace de forma estática.
- **Códigos lineales.** Son aquellos en los que cualquier combinación lineal de palabras de código válida (por ejemplo la suma módulo 2) produce otra palabra válida.
- **Códigos cíclicos.** Se trata de aquellos en los que cualquier desplazamiento cíclico de una palabra de código da lugar a otra palabra de código.

2.3. Codificación de la Señal en sistemas DS-CDMA

CDMA (*Code Division Multiple Access*) junto con los esquemas clásicos de FDMA (*Frequency Division Multiple Access*) y TDMA (*Time Division Multiple Access*) forman el grupo de técnicas de acceso múltiple. A diferencia de las dos últimas, CDMA es una técnica de espectro ensanchado en la que los usuarios comparten la misma banda de frecuencias y pueden separarse en el receptor si se conoce su código que es único.

Existen diversas formas de utilizar un código en los transmisores de espectro ensanchado siendo las técnicas DS (*Direct Sequence*) las más populares dentro de la familia CDMA. Estas técnicas se utilizan para permitir la transmisión de varias señales distintas que llevan información a través del mismo canal. La señal de datos se multiplica por una secuencia de bits pseudo aleatoria a la que normalmente se la conoce con el nombre de código de ruido pseudo aleatorio (código pn). Un código pn es una secuencia de chips con valores -1 y 1 (polares) o 0 y 1 (no polares). Estas secuencias de bits tienen propiedades parecidas a las del ruido como baja correlación cruzada o autocorrelación. Pero aunque esta secuencia parezca aleatoria, esconde un mensaje de débil potencia escondido en el ruido de fondo. Este mensaje puede obtenerse con un determinado receptor, sin embargo, cualquiera que no conozca la secuencia del código pn, no es capaz de detectar su presencia en la señal. En la Figura 2.2 aparece un esquema de los códigos pn así como de la forma en la que afectan a la señal. En ella se puede apreciar de forma gráfica el efecto de este tipo de códigos sobre la señal.

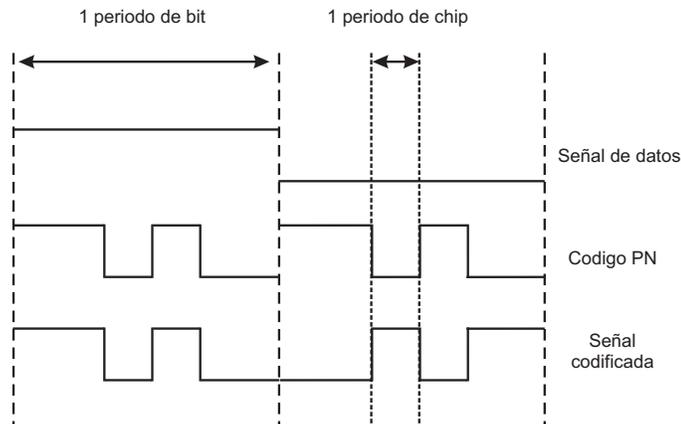


Figura 2.2: Código PN

Otro efecto que produce la utilización de estos códigos es un ensan-

chamamiento del espectro de la señal en un amplio rango de frecuencias. Para el caso de sistemas DS de ensanchamiento de espectro, este fenómeno se produce debido al uso de símbolos más pequeños llamados chips en la función de onda pn. De esta forma, el código $c_i(t)$ puede descomponerse en una secuencia de chips binarios que modulan a un pulso $\alpha(t)$ de duración T_c , i.e.,

$$c(t) = \sum_{j=0}^{L-1} c(j)\alpha(t - kT_c) \quad (2.1)$$

donde T_c es el periodo de chip. T_c está relacionado con el periodo de símbolo T mediante la ecuación $L = \frac{T}{T_c}$, donde L es la longitud del código o ganancia de procesamiento. El mensaje original de banda estrecha es multiplicado por $c(t)$ dando lugar a una señal con componentes en frecuencia mayores y un espectro más ancho. Sin embargo, la multiplicación por $c(t)$ se lleva a cabo en el receptor. Este proceso provoca a su vez un ensanchamiento del espectro de cualquier interferencia, por tanto, baja su densidad de potencia y permite que el diseño del canal sea más robusto a los perjuicios del canal. En la Figura 2.3 aparece representada una aplicación del esquema anterior para el caso en el que varios usuarios comparten simultáneamente el mismo canal en un sistema CDMA (*Code Division Multiple Access*). En este caso, cada usuario tiene su propia secuencia de chips de modo que el código para el i -ésimo usuario se expresa,

$$c_i(t) = \sum_{j=0}^{L-1} c_i(j)\alpha(t - kT_c) \quad (2.2)$$

Para asegurar un bajo nivel de interferencias entre los usuarios así como una separación eficiente de las señales, las secuencias han de poseer buenas propiedades de correlación cruzada.

2.3.1. Ventajas del DS-CDMA

Ventajas para los usuarios

- *Calidad de voz y comunicación.* CDMA posibilita una calidad de voz comparable a las calidad de una línea inalámbrica. Es capaz de filtrar los ruidos de fondo así como las interferencias con lo que se consigue una mejora notable en lo que a calidad de llamada se refiere.
- *Disminución del consumo de energía.* Las unidades móviles que transmiten con CDMA emplean normalmente fuentes de energía menores

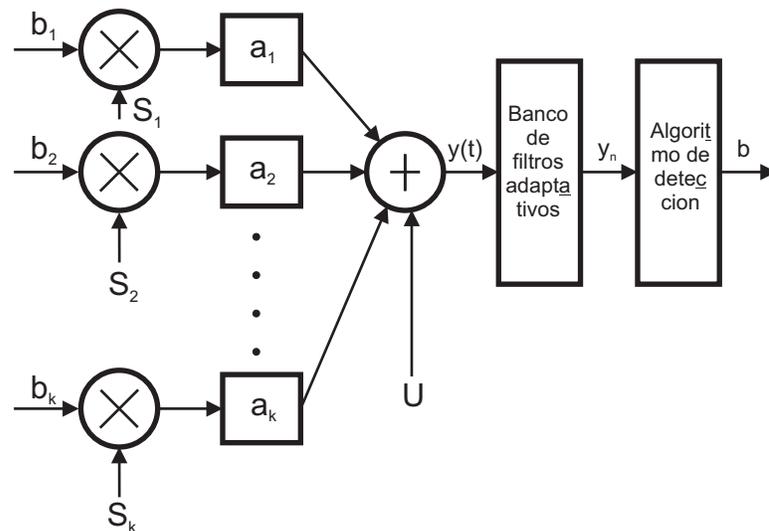


Figura 2.3: Sistema CDMA

que los móviles que utilizan otras tecnologías ya que su forma de transmisión es discontinua. Este hecho posibilita una mayor duración de las baterías así como la utilización de baterías más pequeñas con los que lo móviles pueden ser más pequeños y ligeros.

- *Solución de problemas de congestión del sistema.* Una de las principales ventajas de la utilización de CDMA es su capacidad para aumentar la capacidad del sistema evitándose así problemas de congestión del mismo.

Ventajas para los proveedores de servicios

- *Mayor capacidad.* Debido fundamentalmente al uso de códigos ortogonales, CDMA es capaz de proporcionar un aumento entre 10 y 20 veces de la capacidad de las tecnologías inalámbricas con lo que los proveedores pueden ofrecer mayor cantidad y calidad de servicios así como un volumen de tráfico inalámbrico mayor.
- *Cobertura más amplia.*
- *Frecuencia Universal.* Esta ventaja está relacionada con el hecho de que en CDMA todos los usuarios comparten el mismo espectro.

Por todo lo anterior, la calidad de servicios que puede ofrecerse con CDMA es mucho mayor. Además, al haberse reconocido en toda la comunidad

científica como la tecnología digital más avanzada, CDMA está sometido a una constante mejora que da respuesta a la constante demanda de nuevos servicios y mayor rapidez.

2.4. Interespaciado

El interespaciado se realiza, a similitud de la codificación, para evitar los errores debidos a interrupciones que se presentan en distintos tipos de canales.

2.5. Modulación Digital

Existen básicamente dos tipos de modulación: la modulación *analógica*, que se realiza a partir de señales analógicas de información, por ejemplo la voz humana, audio y vídeo en su forma eléctrica y la modulación *digital*, que se lleva a cabo a partir de señales generadas por fuentes digitales, por ejemplo un ordenador.

Los distintos tipos dentro de cada una de las dos grandes clases descritas anteriormente son:

Modulación Analógica: AM, FM, PM

Modulación Digital: ASK, FSK, PSK, QAM

La modulación analógica es el proceso mediante el cual una señal de información se convierte en una forma de onda senoidal. En la modulación digital la información senoidal de duración T segundos se denomina símbolo digital. Esta función senoidal tiene tres características que pueden ser usadas para distinguirla de otras funciones de la misma naturaleza: la amplitud, la frecuencia y la fase. Entonces la modulación puede definirse como el proceso en el que la amplitud, la frecuencia o la fase de una señal, o una combinación de ellas se cambia de acuerdo con la información que desee transmitirse. La forma general de una señal senoidal, $s(t)$, es:

$$s(t) = A(t) \cos(\theta(t))$$

donde $A(t)$ es la amplitud variante con el tiempo y $\theta(t)$ es el ángulo variante con el tiempo. Es conveniente escribir el ángulo de la siguiente forma:

$$\theta(t) = \omega_0 t + \phi(t)$$

y así

$$s(t) = A(t) \cos [\omega_0 t + \phi(t)]$$

Coherente	No coherente
Phase Shift Keying (PSK)	Diferencial Phase Shift Keying (DPSK)
Frecuency Shift Keying (FSK)	Frecuency Shift Keying (FSK)
Amplitude Shift Keying (ASK)	Amplitude Shift Keying (ASK)
Continuous Phase Modulation (CPM)	Continuous Phase Modulation (CPM)
Híbridos	Híbridos

Figura 2.4: Tipos de modulación digital

Donde $\theta(t)$ es la frecuencia en radianes de la señal y $\phi(t)$ es la fase.

Los tipos básicos de modulación se presentan en la tabla 2.4. Al proceso de detección en el que el receptor explota el conocimiento de la fase de la señal se le denomina *detección coherente*; cuando el receptor no utiliza tal información de referencia de fase, el proceso se llama *detección no coherente*. En comunicaciones digitales, los términos demodulación y detección se usan indistintamente, no obstante es importante notar que la detección involucra procesos de decisión. En la detección coherente ideal, el prototipo de cada señal que llega se encuentra disponible en el receptor. Estas formas de onda prototipo intentan duplicar el conjunto de señales transmitidas en cualquier aspecto. Entonces se dice que el receptor está amarrado (“phase locked”) a la señal de llegada. Durante la detección, el receptor multiplica e integra (correlaciona) la señal de llegada con cada una de sus réplicas prototipo.

En detección no coherente los sistemas que emplean demoduladores están diseñados para operar sin el conocimiento del valor absoluto de la fase de la función sinusoidal que llega, por lo tanto, no se requiere la estimación de fase. Por tanto, la ventaja de los sistemas no coherentes sobre los coherentes es la reducción de la complejidad y el precio que hay que pagar es un incremento en la probabilidad de error (P_E).

Existe una modulación denominada PSK que puede clasificarse como no coherente (o diferencialmente coherente) ya que esta no requiere información sobre la fase de referencia de la portadora recibida. Esta “pseudo PSK”, conocida como PSK diferencial (DPSK), utiliza la información de fase del símbolo anterior como una fase de referencia para detectar el símbolo actual.

Las Figuras 2.5, 2.6 y 2.7 muestran algunos ejemplos de los formatos de modulación más comunes: PSK, FSK, ASK. En el caso de codificación general M -aria, el procesador acepta grupos de k bits e instruye al modulador para producir una de las formas de onda disponibles del conjunto de $M = 2k$. La modulación binaria, en donde $k = 1$, es sólo un caso especial de modulación M -aria.

2.5.1. Modulación digital en fase (PSK)

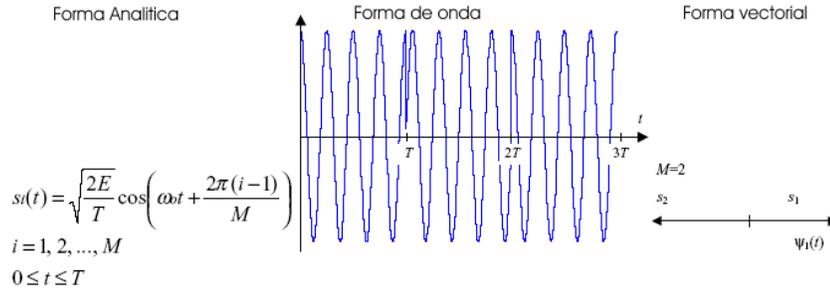


Figura 2.5: Modulación PSK

La modulación digital en fase (PSK) fue desarrollada durante los primeros días del programa espacial. PSK se utiliza actualmente en sistemas de comunicación militares y comerciales. La expresión analítica general para PSK es

$$s_i(t) = \sqrt{\frac{2E}{T}} \cos[\omega_0 t + \phi_i(t)] \quad i = 1, 2, \dots, M$$

donde el término de fase, $\phi_i(t)$, contiene M valores discretos, típicamente dados por:

$$\phi_i = \frac{2\pi(i-1)}{M} \quad i = 1, 2, \dots, M$$

Para el ejemplo de PSK binario de la Figura 2.5 (BPSK), $M = 2$ el parámetro E es la energía del símbolo, T es la duración temporal del símbolo y $0 \leq t \leq T$. En la modulación BPSK, la señal de datos moduladora desplaza la forma de onda $s_i(t)$, en uno de dos estados, ya sea 0 o π . La Figura 2.5 muestra una señal BPSK con cambios abruptos de fase en transiciones de símbolo; si el flujo de datos modulador estuviera formado por unos y ceros alternados, habría cambios abruptos en cada transición. Las formas de onda pueden representarse como vectores en una gráfica polar. La longitud del vector corresponde a la amplitud de la señal, y la dirección del vector, para el caso general M -ario, corresponde a la fase de la señal relativa a las otras $M-1$ señales del conjunto. Para el caso particular de BPSK, la representación gráfica muestra dos vectores opuestos 180° . Los conjuntos de señales que pueden ser representados con tales vectores opuestos son conocidos con el nombre de conjuntos de señales antipodales.

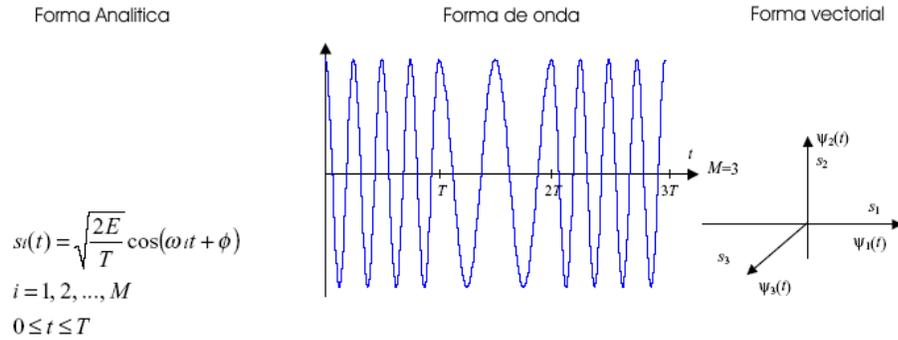


Figura 2.6: Modulación FSK

2.5.2. Modulación digital en frecuencia (FSK)

La expresión analítica general para la modulación digital FSK es

$$s_i(t) = \sqrt{\frac{2E}{T}} \cos [\omega_i t + \phi(t)] \quad i = 1, 2, \dots, M$$

donde el término de frecuencia, ω_i , contiene M valores discretos, y el término de fase, $\phi(t)$, es una constante arbitraria. La forma de onda FSK de la Figura 2.6 muestra cambios de frecuencia abruptos en cada transición entre símbolos. En este ejemplo, $M = 3$, correspondiendo al mismo número de tipos de forma de onda (3-ario). En la práctica, M es normalmente una potencia de 2 (2, 4, 8, 16, ...). El conjunto de señales está caracterizado por coordenadas cartesianas, de manera que los ejes perpendiculares representan una senoidal con frecuencia diferente. Los conjuntos de señales que pueden ser caracterizados mediante vectores perpendiculares se llaman señales ortogonales.

2.5.3. Modulación digital en amplitud (ASK)

La expresión general para la modulación digital en amplitud es

$$s_i(t) = \sqrt{\frac{2E_i}{T}} \cos [\omega_0 t + \phi(t)] \quad i = 1, 2, \dots, M$$

Donde $E_i(t)$ representa la amplitud variante en el tiempo, ω_0 la frecuencia constante de la señal y $\phi(t)$ es una constante arbitraria de fase. En el ejemplo de la Figura 2.7, se ha elegido $M = 1$, de manera que la amplitud se alterna entre dos niveles de voltaje.

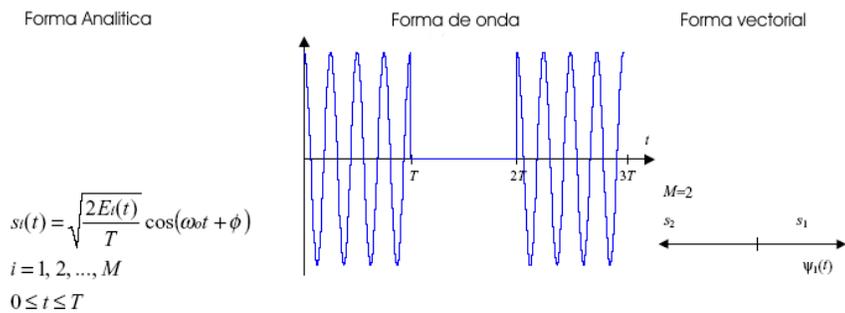


Figura 2.7: Modulación ASK

2.5.4. Modulación por amplitud de cuadratura (QAM)

La Modulación QAM es un método que combina dos señales con modulación doblando de esta forma el ancho de banda efectivo. La Modulación QAM se utiliza junto con la modulación PAM (Pulse Amplitude Modulation) en sistemas digitales, especialmente en aplicaciones inalámbricas.

En una señal QAM, hay dos portadoras, cada una de ellas tienen la misma frecuencia pero sus fases se diferencian en 90° . Una señal se denomina la señal I y otra la señal Q . Matemáticamente, una de las señales puede representarse a través de una función seno y la otra con una función coseno. Cuando llegan al receptor, las señales portadoras se separan y los datos se extraen de cada una. En ese momento los datos se combinan recuperándose la información original.

2.5.5. Objetivos de la modulación

Existen varias razones para modular, entre ellas pueden destacarse las siguientes:

- Facilita la propagación de la señal de información por cable o por el aire.
- Ordena el radioespectro, distribuyendo canales a cada información distinta.
- Disminuye dimensiones de antenas.
- Optimiza el ancho de banda de cada canal
- Evita interferencia entre canales.
- Protege a la Información de las degradaciones por ruido.

- Define la calidad de la información transmitida.

Modulación para reducir el ruido y la interferencia: Se ha dicho que es imposible eliminar totalmente el ruido del sistema. Y aunque es posible eliminar la interferencia, puede no ser práctico. Por fortuna, ciertos tipos de modulación tienen la útil propiedad de suprimir tanto el ruido como la interferencia. La supresión, sin embargo, ocurre a un cierto precio; generalmente requiere de un ancho de banda de transmisión mucho mayor que el de la señal original; de ahí la designación del ruido de banda ancha. Este convenio de ancho de banda para la reducción del ruido es uno de los intereses y a veces desventajosos aspectos del diseño de un sistema de comunicación.

Modulación por asignación de frecuencia: El propietario de un aparato de radio o televisión puede seleccionar una de varias estaciones, aún cuando todas las estaciones estén transmitiendo material de un programa similar en el mismo medio de transmisión. Es posible seleccionar y separar cualquiera de las estaciones, porque cada una tiene asignada una frecuencia portadora diferente. Si no fuera por la modulación, solo operaría una estación en un área dada. Dos o más estaciones que transmitan directamente en el mismo medio, sin modulación, producirán una mezcla inútil de señales interferentes.

Modulación para multicanalización: A menudo se desea transmitir muchas señales en forma simultánea entre dos puntos. Las técnicas de multicanalización son formas intrínsecas de modulación, permiten la transmisión de múltiples señales a través de un canal, de tal manera que cada señal puede ser captada en el extremo receptor. Las aplicaciones de la multicanalización comprenden telemetría de datos, emisión de FM estereofónica y telefonía de larga distancia. Es muy común, por ejemplo, tener hasta 1,800 conversaciones telefónicas de ciudad a ciudad, multicanalizadas y transmitidas sobre un cable coaxial de un diámetro menor de un centímetro.

Modulación para superar las limitaciones del equipo: El diseño de un sistema queda generalmente a la disponibilidad de equipo, el cual a menudo presenta inconvenientes en relación con las frecuencias involucradas. La modulación se puede usar para situar una señal en la parte del espectro de frecuencia donde las limitaciones del equipo sean mínimas o donde se encuentren más fácilmente los requisitos de diseño. Para este propósito, los dispositivos de modulación se encuentran también en los receptores, como ocurre en los transmisores.

2.6. Modelado del canal

Generalmente cuando se analiza el comportamiento del canal en un sistema de comunicaciones se suele partir de un modelo clásico para el canal,

esto es, considerar un canal con ruido Gaussiano aditivo, con muestras de ruido gaussianas estadísticamente independientes afectando a las muestras de datos que están libres de interferencia intersimbólica. La principal fuente de degradación del rendimiento es el ruido térmico generado en el receptor. Normalmente, la interferencia externa en la antena es más significativa que el ruido termal. Este ruido termal tiene normalmente una densidad espectral de potencia plana y una función densidad de probabilidad del voltaje gaussiana de media cero. Cuando se modelan sistemas prácticos, el siguiente paso consiste en la introducción de filtros limitadores de banda. El filtrado en el transmisor se utiliza normalmente para satisfacer algunas condiciones en el contenido espectral. El filtrado en el receptor, por su parte, es normalmente el resultado de la implementación de un filtro adaptativo. Debido a las propiedades de limitación de banda y las de distorsión de fase, técnicas especiales de señales y técnicas de ecualización pueden necesitarse para eliminar la interferencia intersimbólica.

Si las características de propagación no se especifican, se asume normalmente que la atenuación de la señal en la distancia se comporta como si la propagación ocurriese idealmente en el espacio libre. El modelo de espacio libre trata la región entre el transmisor y la antena receptora como libre de todo objeto que pueda absorber o reflejar energía en radiofrecuencia. También asume que, dentro de esta región, la atmósfera se comporta como un medio perfectamente uniforme y no absorbente. Además, se considera que la tierra está infinitamente lejos de la señal que se propaga (o equivalentemente se considera que tiene un coeficiente de reflexión despreciable). Básicamente, en este modelo idealizado de espacio libre, la atenuación de la energía en radiofrecuencias entre el transmisor y el receptor se comporta de acuerdo a una ley inversa de cuadrados. La potencia recibida puede expresarse como una atenuación de la potencia transmitida y el factor de atenuación es $L_s(d)$. Este factor se llama pérdida de trayecto o “*path loss*” y si la antena receptora es isótropa puede escribirse como:

$$L_s(d) = \left(\frac{4\pi d}{\lambda} \right)^2 \quad (2.3)$$

En (2.3), d es la distancia entre el emisor y el receptor y λ es la longitud de onda de la señal que se propaga. Para el caso de propagación ideal, la potencia de la señal es muy predecible. Para la mayoría de los canales, donde la propagación de la señal ocurre en la atmósfera y cerca del suelo, el modelo de propagación en el espacio libre no es adecuado para describir el comportamiento del canal y predecir el comportamiento del canal. En los sistemas de comunicaciones inalámbricas móviles, la señal puede viajar desde el transmisor al receptor a través de diferentes caminos. Este fenómeno se denomina

propagación multitrayecto y puede causar variaciones en la amplitud, fase y ángulo de llegada de la señal recibida dando lugar a lo que se denomina desvanecimiento multitrayecto.

2.6.1. Caracterización de la propagación

Existen varios tipos de mecanismos que pueden afectar a la propagación en un sistema de comunicaciones móviles, de entre ellas cabe destacar la reflexión, la difracción y el scattering.

- *Reflexión.* La reflexión ocurre cuando una onda electromagnética choca con una superficie suave de grandes dimensiones comparada con su longitud de onda.
- *Difracción.* La difracción ocurre cuando el trayecto entre el transmisor y el receptor se obstaculiza debido a un objeto con alta densidad y con dimensiones que son grandes en comparación a la longitud de onda dando lugar a ondas secundarias que se forman detrás del obstáculo.
- *Scattering.* El fenómeno de scattering ocurre cuando una onda choca bien con una superficie rugosa o con una superficie cuyas dimensiones son del orden de la longitud de onda o menores. El resultado es que la energía se refleja en todas las direcciones. En ambientes urbanos, los obstáculos que pueden dar lugar a scattering son vegetación, letreros en las calles o farolas.

Como ya se ha comentado, existen dos tipos de desvanecimiento básicos que afectan a las comunicaciones móviles: desvanecimiento a gran escala y desvanecimiento a pequeña escala. Veamos matemáticamente primero cómo afectan ambos fenómenos a la señal que se envía. Usando notación compleja, podemos escribir la señal transmitida como:

$$s(t) = \text{Re}\{g(t)e^{j2\pi f_c t}\} \quad (2.4)$$

donde $\text{Re}\{\bullet\}$ significa que tomamos la parte real de $\{\bullet\}$, f_c es la frecuencia de la portadora. La forma de onda en banda base $g(t)$ se llama envolvente compleja de $s(t)$ y puede expresarse de la siguiente forma:

$$g(t) = |g(t)|e^{j\phi(t)} = R(t)e^{j\phi(t)} \quad (2.5)$$

En un ambiente desvanescente, $g(t)$ se ve modificada por un factor complejo multiplicativo $\alpha(t)e^{-j\theta(t)}$. Entonces, la forma de onda en banda base modificada puede escribirse como $\alpha(t)e^{-j\theta(t)}g(t)$. A continuación vamos a analizar

la magnitud $\alpha(t)R(t)$ que puede expresarse como:

$$\alpha(t)R(t) = \alpha_p(t)\alpha_g(t)R(t) \quad (2.6)$$

donde $\alpha_g(t)$ es el término de desvanecimiento a gran escala y $\alpha_p(t)$ es el término de desvanecimiento a pequeña escala. Los efectos de cada uno de estos dos términos pueden verse en las Figuras 2.8 y 2.9.

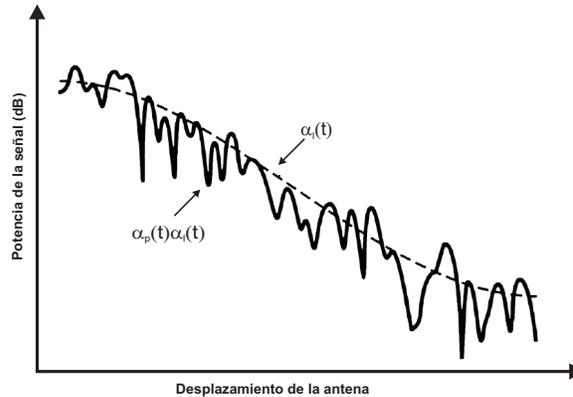


Figura 2.8: Efecto conjunto de los desvanecimientos a pequeña y gran escala

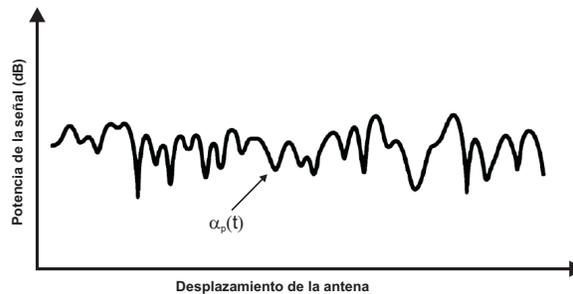


Figura 2.9: Componente del desvanecimiento a pequeña escala

2.6.2. Desvanecimiento a gran escala

El *desvanecimiento a gran escala* es la media de la atenuación de la potencia de la señal debida a la propagación a lo largo de grandes distancias. Este fenómeno puede producirse debido a perfiles del terreno abruptos como montañas, bosques, vallas de anuncios, conjuntos de edificios, entre el receptor y el transmisor. La estadística de este tipo de desvanecimiento proporciona una forma de calcular un estimador de las atenuaciones en función

de la distancia. Dicha estadística depende de las características del sistema receptor, la frecuencia a la que se opera y el tipo de medio en el que se transmite la señal. Generalmente la función densidad de probabilidad utilizada para describir este tipo de fenómeno es una log-normal:

$$p(x) = \begin{cases} \frac{1}{\sqrt{\pi}\sigma x} \exp \left\{ -\frac{(\log x - \mu)^2}{2\sigma^2} \right\}, & x > 0 \\ 0, & x < 0 \end{cases} \quad (2.7)$$

En la ecuación anterior, x es una variable aleatoria que representa las bajas fluctuaciones del nivel de la señal. μ y σ son la media y la desviación típica de x , expresadas en decibeles, respectivamente.

2.6.3. Desvanecimiento a pequeña escala

El *desvanecimiento a pequeña escala* hace referencia a los cambios en la amplitud de la señal y de la fase que pueden producirse como resultado de los pequeños cambios (tan pequeños como media longitud de onda) de la posición espacial entre el receptor y el transmisor. El desvanecimiento a pequeña escala se manifiesta en dos mecanismos: el ensanchamiento de temporal de la señal (también llamado dispersión) y la dependencia del comportamiento del canal con el tiempo. Para aplicaciones móviles de radio, el canal también cambia con el tiempo debido a que el movimiento entre el transmisor y el receptor se traduce en cambios en el camino de propagación. La tasa de cambio de estas condiciones de está relacionada con la rapidez de desvanecimiento. El desvanecimiento a pequeña escala se denomina *desvanecimiento tipo Rayleigh* si hay varias trayectorias o caminos debidos a múltiples y distintas reflexiones. Si no hay componente de la señal directa en la LOS (“line of sight”), la envolvente de la señal recibida puede describirse estadísticamente a través de una función densidad de probabilidad de tipo Rayleigh:

$$p(y) = \begin{cases} \frac{1}{\sigma^2} \exp \left\{ -\frac{y^2}{2\sigma^2} \right\}, & y > 0 \\ 0, & y < 0 \end{cases} \quad (2.8)$$

Si hay un camino directo, entonces lo que se tiene es una distribución tipo Rician. La correspondiente pdf viene dada por:

$$p(y) = \begin{cases} \frac{y}{\sigma^2} \exp \left\{ -\frac{y^2 + s^2}{2\sigma^2} \right\} J_0 \left(\frac{ys}{\sigma^2} \right), & y > 0 \\ 0, & y < 0 \end{cases} \quad (2.9)$$

O lo que es lo mismo, la estadística del desvanecimiento a pequeña escala es Rayleigh si el camino directo está bloqueado y Rician en cualquier otro caso. Entonces, una señal de radio móvil en un gran área puede sufrir tanto desvanecimiento a pequeña escala como desvanecimiento a gran escala.

Las manifestaciones del desvanecimiento a pequeña escala dos tipos: la dispersión temporal (o ensanchamiento en retardos) de la señal y la variación temporal del canal. Ambos fenómenos pueden examinarse en dos dominios: el del tiempo y el de la frecuencia como queda reflejado en el cuadro de la Figura 2.10.

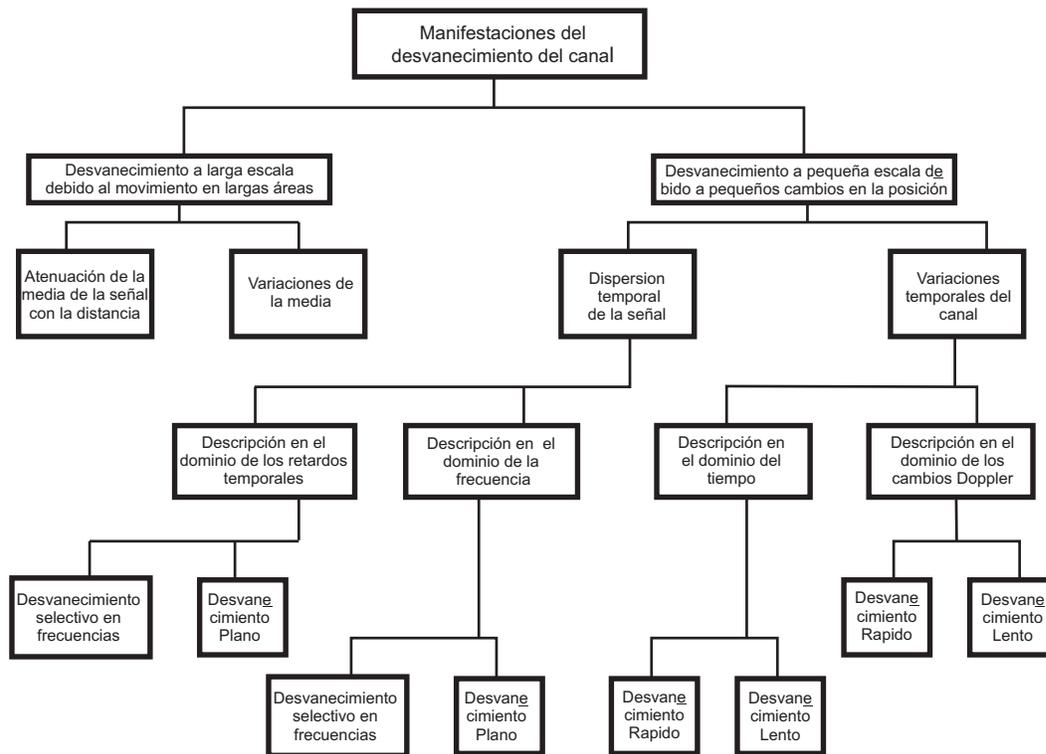


Figura 2.10: Resumen del fenómeno de desvanescencia

El ensanchamiento en retardos produce un desvanecimiento selectivo en frecuencias, lo que a su vez hace que el desvanecimiento sea también función de la frecuencia. Este fenómeno puede caracterizarse a través del ancho de banda de coherencia, que representa la máxima separación entre frecuencias para la que la respuesta en frecuencia del canal para dos cambios en frecuencia permanece fuertemente correlacionada. El ancho de banda de coherencia es inversamente proporcional al ensanchamiento de retardos y es una medida de la selectividad del canal en frecuencias. Un cociente pequeño entre el ancho de banda de coherencia y el ancho de banda de la señal indica un canal

selectivo en frecuencia. Por tanto, podemos hablar de dos tipos de canales desvanescentes dependiendo del grado en el que se manifieste este fenómeno: aquellos que son selectivos en frecuencia y los que no lo son (también llamados planos).

En cuanto a la variación temporal del canal, los tipos de degradación desvanescente dependerán de la velocidad con las que se produzcan los cambios en el canal. Es por estos que podemos dividir los tipos de canal en dos: desvanescentes rápidos y desvanescentes lentos.

2.7. Caracterización del error de transmisión: BER

En transmisiones relacionadas con telecomunicaciones, el BER (*Índice de Error de Bit*) es el porcentaje de bits que tienen error dividido por el número de bits recibidos en una transmisión. Normalmente se expresa como una potencia negativa de diez. La expresión que se usa para su cálculo es la siguiente:

$$BER = \frac{b_e}{b_t}$$

donde b_e es el número total de bits con error u b_t es el número total de bits transmitidos.

El BER es un indicador de cómo de frecuentemente un conjunto de datos tiene que ser retransmitido debido a un error. Un BER demasiado alto puede indicar que la transmisión de datos de forma más lenta mejoraría el tiempo de transmisión para una cantidad dada de datos enviados ya que el BER puede reducirse bajando el número de paquetes que estén presentes en cada unidad de tiempo.

2.8. Caracterización de la calidad de la señal: SNR

SNR (*Signal-to-Noise-Ratio*) es el cociente entre la amplitud de la señal de datos analógica o digital y la amplitud del ruido en el canal de transmisión en un momento determinado. Normalmente se expresa de forma logarítmica en decibeles (dB). La expresión que se utiliza para su cálculo es la siguiente:

$$SNR(dB) = 10 \log \left(\frac{P_s}{P_r} \right)$$

donde P_s es la potencia de la señal y P_r es la potencia del ruido.

La SNR mide la calidad de un canal de transmisión o de una señal de audio a través de un determinado canal. Cuanto más grande es el cociente, más fácil es identificar , y por lo tanto, aislar y eliminar la fuente de ruido. Un valor de SNR cero indica que la señal deseada tiene la misma potencia que el ruido.

Capítulo 3

Filtrado Bayesiano

La teoría del filtrado es la que se ocupa de estimar secuencialmente el estado de un sistema usando las medidas obtenidas a lo largo del tiempo. Esta teoría tiene numerosas aplicaciones en diversos campos. Un gran número de aplicaciones se encuentran en el campo de la ingeniería aunque también ha sido aplicada a campos como la medicina, el financiero o incluso el medioambiental.

A lo largo de esta tesis, se trabaja con sistemas que se pueden modelar a través de lo que se denomina *Modelo de Espacio de Estados*. En este modelo se tienen dos ecuaciones importantes denominadas Ecuación de Medida y Ecuación de Observación de las que se hablará detalladamente a lo largo de este capítulo y de los que siguen. En el caso de que estas ecuaciones sean lineales y los ruidos que aparecen en ellas blancos, gaussianos y aditivos, la solución óptima se obtiene usando usando Filtrado de Kalman.

Para no restringirse a un caso específico de modelo para obtener una solución exacta, un gran número de filtros aproximados han sido creados. Históricamente, el primero de estos filtros aproximados fue el Filtro de Kalman Extendido (EKF)(Jazwinzi, 1973). La idea es linealizar las ecuaciones de observación y de medida de modo que se pueda usar un Filtro de Kalman para

obtener estimadores de la variable de estado así como las covarianzas de estos estimadores.

Tan pronto como se empezó a contar con más y mejores posibilidades de computación fueron apareciendo filtros que requerían mayor complejidad computacional. Este tipo de filtros pueden dividirse en cuatro categorías. En primer lugar se encuentran las variantes de los EKF (Filtros de Kalman Extendidos) que obtienen estimadores puntuales de la variable de estado, y estimadores del error cuadrático medio de los estimadores. En segundo lugar, están los filtros que aproximan la densidad a posteriori de la variable de estado con una mezcla de distribuciones. Este tipo fue propuesto por primera vez por Sorenson y Alspach (1971), en este caso, su filtro usaba mezclas de Gaussianas. En tercer lugar, se encuentran los filtros basados en un mallado, que evalúan la densidad en una serie de nodos escogidos para cubrir por completo el espacio de estados. Este conjunto de nodos con su peso asociado se usa como aproximación discreta (Sorenson, 1988) de la densidad a posteriori o como base de algunas aproximaciones continuas a la misma, por ejemplo usando splines (Kitagawa, 1987). La última clase de filtros son aquellos que usan métodos de Monte Carlo. Su origen puede buscarse en Handschin (1970) y Mayne (1969). Este tipo de filtros usan métodos de Monte Carlo para dar lugar a estimadores de la media y covarianza de la distribución a posteriori. Desde los años 90 (Gordon et al., 1993), los métodos secuenciales de Monte Carlo han venido usándose como conjuntos de puntos que aproximan la función densidad de probabilidad a posteriori.

Este capítulo se encuentra estructurado de la siguiente forma: en la Sección 3.1 se presenta una breve introducción a la teoría Bayesiana en general y al proceso de inferencia en particular. Los Sistemas Probabilísticos Dinámicos se introducen en la Sección 3.2 así como algunos ejemplos de los mismos. En la Sección 3.3 se presentan las ideas básicas de la teoría de estimación Bayesiana. La Sección 3.4 está dedicada al proceso de estimación recursiva. En la Sección 3.5 se trata un método de estimación recursiva para sistemas representados por ecuaciones lineales con ruidos Gaussianos aditivos: el Filtrado Kalman. En la Sección 3.6 se presenta el Filtrado de Kalman Extendido que es un proceso de estimación recursiva utilizado cuando las ecuaciones del sistema no son lineales. Los métodos de Filtrado de Partículas se presentan en la Sección 3.7 así como todos los conceptos de “Importance Sampling” and “Importance Resampling”. El algoritmo M Estocástico se presenta en la Sección 3.8 como alternativa a los algoritmos genéricos de Filtrado de Partículas cuando la variable con la que se trata es discreta. Finalmente, en 3.9 se trata el Filtrado de Kalman de Mezcla utilizado cuando se trabaja con Modelos Lineales Condicionados.

3.1. Inferencia Bayesiana

La incertidumbre y la toma de decisiones teniendo en cuenta esta incertidumbre son elementos de la vida cotidiana de cualquier persona. La incertidumbre puede estar relacionada con situaciones pasadas donde el conocimiento directo o evidencia no está disponible. También puede estar ligada a desarrollos presentes o futuros que no están completos. En cualquier circunstancia estas decisiones se toman en base a inferencias que se obtienen a partir de modelos de lo que se espera observar [13]. La Inferencia Bayesiana es el proceso de ajustar un modelo de probabilidad a un conjunto de datos y resumir los resultados a través de una distribución de probabilidad en los parámetros del modelo y las magnitudes que no han sido observadas como predicciones para usar con nuevas observaciones.

Las hipótesis científicas típicamente se expresan a través de las distribuciones de probabilidad de los datos científicos observables. Estas distribuciones de probabilidad dependen de ciertas magnitudes desconocidas que se denominan parámetros. En el paradigma Bayesiano, el conocimiento sobre los parámetros del modelo se expresa mediante una distribución de probabilidad en los parámetros denominada “distribución a priori” y generalmente se escribe como $p(\theta)$. Cuando se dispone de nuevos datos \mathbf{y} , la información que éstos contienen en relación a los parámetros del modelo se expresa en a través de la “verosimilitud”, que es proporcional a la distribución de los datos observados dados los parámetros del modelo y puede escribirse como $p(\mathbf{y}|\theta)$. Esta información se puede combinar con la distribución a priori para dar lugar a una distribución de probabilidad actualizada denominada “distribución a posteriori”. Es precisamente en esta última distribución de probabilidad en la que se basa todo el proceso de inferencia Bayesiano. El Teorema de Bayes, que no es más que una identidad elemental en teoría de probabilidades, dice cómo se realiza el proceso de actualización de forma matemática: la probabilidad a posteriori es proporcional al producto de la probabilidad a priori por la verosimilitud o, de forma más explícita,

$$p(\theta|\mathbf{y}) = \frac{p(\theta)p(\mathbf{y}|\theta)}{\int p(\theta)p(\mathbf{y}|\theta)}$$

En teoría, siempre se cuenta con la distribución a posteriori, pero en los modelos complejos realistas, los cálculos analíticos necesarios son normalmente intratables. Durante algunos años, desde finales de 1980 y principios de 1990, la comunidad investigadora se dio cuenta de que la obtención de muestras de la distribución a posteriori podía constituir una opción adecuada y aplicable.

Hay muchas razones para usar métodos Bayesianos, y sus aplicaciones están presentes en diversos campos. Mucha gente recurre a técnicas Bayesianas debido a su consistencia filosófica. Algunos teoremas fundamentales muestran que si una persona quiere tomar decisiones consistentes cuando tiene que hacer frente a elementos desconocidos, la única forma de operar pasa por utilizar métodos Bayesianos. Otros prefieren hacer uso de los métodos frecuentistas. Por otro lado, las probabilidades a priori son intrínsecamente subjetivas y muchos estadísticos ven esto como el inconveniente fundamental de los métodos Bayesianos. Los seguidores de las técnicas Bayesianas argumentan que esto es ineludible, y que los métodos frecuentistas también conllevan elecciones subjetivas. Este tema ha sido la fuente de disputa entre los partidarios fundamentalistas de los dos paradigmas estadísticos durante al menos 50 años. Pero han sido precisamente las ventajas de los métodos Bayesianos las que han impulsado su fuerte desarrollo durante los 20 años pasados. Y además son la razón por la cual se ha producido un incesante aumento en su aplicación en una gran variedad de campos. Las herramientas computacionales potentes permiten a los métodos Bayesianos afrontar problemas estadísticos grandes y complejos con relativa facilidad. Problemas en los que la aplicación de métodos frecuentistas ha fallado o sólo ha constituido una aproximación. Los métodos de modelado Bayesiano proporcionan técnicas naturales para que los investigadores en distintas disciplinas puedan estructurar sus datos y sus conocimientos, y para que puedan proporcionar respuestas directas e intuitivas a diversos problemas.

3.2. Sistemas Probabilísticos Dinámicos

En la parte dedicada a comunicaciones de esta tesis, se van a usar métodos computacionales de Monte Carlo para realizar un análisis en tiempo real de sistemas dinámicos. Es por esto que antes de comenzar con conceptos más complejos, tiene sentido caracterizar este tipo de sistemas.

3.2.1. Definición y análisis

Un *Sistema probabilístico dinámico* es una secuencia de distribuciones de probabilidad $p(\mathbf{x}_t | \mathbf{y}_{0:t})$, donde \mathbf{x}_t es la variable de estado, \mathbf{y}_t el vector de observaciones y el índice t es de naturaleza temporal y discreta.

La variable de estado x_t puede evolucionar en el tiempo bien aumentando su dimensión, bien disminuyéndola o sin ningún tipo de cambio. En lo que sigue, la variable de estado va a evolucionar aumentando la dimensión siempre.

En la mayoría de las aplicaciones, la diferencia entre $p(\bullet|y_{t+1})$ y $p(\bullet|y_t)$ se debe a la incorporación de nueva información. Los siguientes procesos son de gran interés en estos sistemas:

Predicción: $p(\mathbf{x}_{t+l}|\mathbf{y}_{0:t}), l \geq 1$. Cuando $p(\bullet|y_t)$ puede extenderse a una nueva componente x_{t+1} , la mejor predicción de x_{t+1} antes de que la nueva información llegue es a través $p(\bullet|y_t)$.

Suavizado: $p(\mathbf{x}_t|\mathbf{y}_{0:T}), T > t$. En este caso, lo que se hace es una estimación de la variable de estado en un estado anterior (t) a la luz de la nueva información en el instante $t + 1$.

Filtrado: $p(\mathbf{x}_t|\mathbf{y}_{0:t})$. Ahora se trata de estimar la variable x_{t+1} usando la información recibida en el instante $t + 1$.

3.2.2. Ejemplos

Como ejemplos típicos sistemas dinámicos se tienen los siguientes:

Problema Bayesiano de datos perdidos

Si se supone que se tiene un conjunto de variables i.i.d. $z_1, z_2, z_3, \dots, z_n$ del modelo $p(z|\theta)$, pero que algunas de estas variables han sido sólo parcialmente observadas. Si descomponemos la variable z de la forma $z_i = (y_i, x_i)$, donde y_i es la parte observada y x_i la no observada y si notamos $\mathbf{y}_t = (y_1, y_2, \dots, y_t)$ y $\mathbf{x}_t = (x_1, x_2, \dots, x_t)$, donde $x_0 = \theta$, el sistema dinámico en este caso es $p(\mathbf{x}_t|\mathbf{y}_t)$.

Modelo de espacio de estados

El siguiente modelo será el que se emplee en todo lo que resta de este trabajo. Las ecuaciones que lo definen son dos. La ecuación de estado,

$$\mathbf{x}_t = \mathbf{f}_t(\mathbf{x}_{t-1}, \mathbf{u}_t) \quad (3.1)$$

y la ecuación de medida,

$$\mathbf{y}_t = \mathbf{g}_t(\mathbf{x}_t, \mathbf{v}_t) \quad (3.2)$$

donde \mathbf{y}_t es el vector de observaciones, \mathbf{x}_t es el vector de estado, \mathbf{g}_t es la función de medida, \mathbf{f}_t es la función de estado, \mathbf{u}_t y \mathbf{v}_t son vectores de ruido y el índice t es temporal. Las suposiciones que normalmente se hacen sobre estas ecuaciones son que las formas analíticas de las funciones y de las distribuciones de los dos ruidos son conocidas. El objetivo es estimar \mathbf{x}_t de forma recursiva basándose en las observaciones \mathbf{y}_t y las suposiciones anteriores.

El método que se ha investigado mayoritariamente y que ha sido el que se ha aplicado más frecuentemente es el filtro de Kalman [10]. La solución dada por el filtro de Kalman es óptima en el caso importante en el que las ecuaciones son lineales y los ruidos independientes, aditivos y Gaussianos. En este caso, las distribuciones de interés (de filtrado, de suavizado o predicción) son también Gaussianas y el filtro de Kalman puede calcularlas de forma exacta sin necesidad de aproximarlas.

Para escenarios donde los modelos no son lineales o los ruidos no son Gaussianos, se han propuesto varios métodos alternativos de entre los cuales, los Filtros de Kalman Extendidos son probablemente los más destacables [10].

Recientemente, los métodos de Filtrado de Partículas se han convertido en una alternativa importante a los Filtros de Kalman Extendidos. Con el Filtrado de Partículas, distribuciones continuas pueden aproximarse con medidas aleatorias discretas que están compuestas por partículas pesadas, donde las partículas son muestras de estados desconocidos del espacio de estados, y los pesos de las partículas son “probabilidades másicas” calculadas usando la teoría de Bayes. En la implementación de filtrado de partículas, el muestreo de importancia (Importance Sampling) juega un papel crucial y, ya que es un método creado para un uso secuencial, se le denomina muestreo de importancia secuencial. La ventaja de los métodos de filtrado de partículas sobre otros métodos es que la aproximación que se realiza no conlleva el uso de linealizaciones de los estimadores sino aproximaciones en la representación las distribuciones usando medidas aleatorias discretas.

3.3. Teoría de estimación de Bayes

3.3.1. Estimaciones MAP

Un estimador MAP (*Maximum a Posterior Probability*) es aquel que se obtiene de maximizar la función densidad de probabilidad a posteriori (esto es, es la moda de la función densidad de probabilidad a posteriori).

Si la función densidad de probabilidad a posteriori de la variable θ dado X es unimodal y simétrica, entonces es fácil ver que el estimador MAP y el estimador de mínimos cuadrados coinciden, ya que en estos casos la función densidad de probabilidad a posteriori alcanza su valor máximo con la moda y el valor esperado. Entonces, si se trabaja con distribuciones de este tipo, ambos estimadores tienden al mismo valor.

Consideremos que la variable aleatoria θ está distribuida de acuerdo con la función densidad de probabilidad a priori $f_{\theta}(\vartheta)$. La función de distribución a posteriori viene dada entonces según el Teorema de Bayes por:

$$f_{\theta|X}(\vartheta|x) = \frac{f_{X|\theta}(x|\vartheta)}{f_X(x)} f_{\theta}(\vartheta) \quad (3.3)$$

Si el logaritmo de la función densidad de probabilidad a posteriori es diferenciable respecto a la variable θ , el estimador MAP viene dado por la solución

$$\frac{\partial \log f_{\theta|X}(\vartheta|x)}{\partial \vartheta} \Big|_{\vartheta=\hat{\theta}_{MAP}} = 0 \quad (3.4)$$

La ecuación anterior se conoce como *Ecuación MAP*.

Si se calcula el logaritmo a la ecuación 3.3 queda

$$\log f_{\theta|X}(\vartheta|x) = \log f_{X|\theta}(x|\vartheta) + \log f_{\theta}(\vartheta) - \log f_X(x) \quad (3.5)$$

y como $\log f_X(x)$ no es función de θ , la ecuación MAP se convierte en

$$\frac{\partial \log f_{\theta|X}(\vartheta|x)}{\partial \vartheta} = \frac{\partial \log f_{X|\theta}(x|\vartheta)}{\partial \vartheta} + \frac{\partial \log f_{\theta}(\vartheta)}{\partial \vartheta} \quad (3.6)$$

Comparando 3.6 con la ecuación estándar de máxima verosimilitud

$$\frac{\partial L(\theta, x)}{\partial \theta} \Big|_{\theta=\hat{\theta}_{ML}} = 0 \quad (3.7)$$

la única diferencia entre las dos es el término $\frac{\partial \log f_{\theta}(\vartheta)}{\partial \vartheta}$. Si $f_{\theta}(\vartheta)$ $\partial \vartheta$ es suficientemente "plana" (esto es, si su varianza es muy grande) su logaritmo también será plano, de esta forma el gradiente del logaritmo será prácticamente cero, y la función densidad de probabilidad a posteriori se maximizará, en el caso límite, para el estimador de máxima verosimilitud.

3.3.2. Estimación Bayesiana con distribuciones Gaussianas

En esta sección se considera el problema de dos variables aleatorias y Gaussianas, (X, Y) , o vectores aleatorios, como (\mathbf{X}, \mathbf{Y}) . Como las distribuciones de variables aleatorias Gaussianas son unimodales y simétricas, y como la distribución condicional $f_{X|Y}$ es también Gaussianas, esta distribución condicional proporciona lo que se necesita para estimar la variable aleatoria X .

Por tanto, la determinación de la distribución condicional y la identificación de su media, proporciona los estimadores necesarios para la mayoría de las funciones Bayesianas de pérdida.

En el apéndice C se muestra como la distribución de una variable aleatoria $\mathbf{Z} = (\mathbf{X}, \mathbf{Y})$, donde $X \in e$ $X \sim \mathcal{N}(\mu_x, R_{xx})$, $Y \in y$ $Y \sim \mathcal{N}(\mu_y, R_{yy})$, es

$$f_{\mathbf{z}}(\mathbf{z}) = f_{\mathbf{z}}(\mathbf{x}, \mathbf{y}) = \frac{1}{(2\pi)^{p/2} \det(R_{zz})} \exp \left[-\frac{1}{2} (\mathbf{z} - \mu_z)^T R_{zz}^{-1} (\mathbf{z} - \mu_z) \right] \quad (3.8)$$

donde $p = m + n$, y

$$R_{zz} = \text{cov}(\mathbf{Z}) = \begin{bmatrix} R_{xx} & R_{xy} \\ R_{yx} & R_{yy} \end{bmatrix} \quad (3.9)$$

A continuación se aborda el problema de la estimación. Dada la medida de \mathbf{Y} , se pretende estimar \mathbf{X} . Esto requiere encontrar $f_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y})$. En el apéndice C se demuestra que $f_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y})$ es Gaussiano con media:

$$\mu_{x|y} = E[X|Y = y] = \mu_x + R_{xy}R_{yy}^{-1}(y - \mu_y) \quad (3.10)$$

y covarianza

$$\text{cov}(\mathbf{X}|\mathbf{Y} = \mathbf{y}) = R_{xx} - R_{xy}R_{yy}^{-1}R_{yx} = P \quad (3.11)$$

La cantidad $\hat{\mathbf{x}} = \mu_{x|y}$ es el estimador Bayesiano de \mathbf{x} , dada la medida \mathbf{Y} , en el sentido de ser la media, la moda, y mediana de la distribución. Puede interpretarse como sigue: Antes de cualquier medida, el mejor estimador de \mathbf{X} se obtiene de la función densidad de probabilidad a priori $f_{\mathbf{X}}(\mathbf{x})$ y es μ_x , la media de \mathbf{X} . A través de las medidas, la función densidad de probabilidad a priori $f_{\mathbf{X}}(\mathbf{x})$ se convierte en la función densidad de probabilidad a posteriori según la expresión

$$f_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y}) = \frac{f_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x})}{f_{\mathbf{Y}}(\mathbf{y})} f_{\mathbf{X}}(\mathbf{x}) \quad (3.12)$$

El estimador a priori se modifica una cantidad que es proporcional a la diferencia entre la medida \mathbf{y} y el valor esperado de la misma. La proporcionalidad depende de como de fuertemente correlacionados (a través de R_{xy}) estén \mathbf{X} e \mathbf{Y} e inversamente de la varianza de R_{yy}^{-1} : las medidas con gran varianza no tienen asociado tanto peso como las que tienen baja varianza.

El siguiente paso consiste en examinar más a fondo el estimador $\hat{x} = \mu_{x|y}$.

1. El estimador no es sesgado:

$$E\hat{\mathbf{x}} = E\mu_x + R_{xy}R_{yy}^{-1}E(\mathbf{Y} - \mu_y) = \mu_x \quad (3.13)$$

2. El error del estimador $\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}}$ no está correlacionado con $\hat{\mathbf{x}} - \mu_x$:

$$E\mathbf{e}(\hat{\mathbf{x}} - \mu_x)^T = 0 \quad (3.14)$$

3. El error no está correlacionado con $\hat{\mathbf{y}} - \mu_y$:

$$E\mathbf{e}(\hat{\mathbf{y}} - \mu_y)^T = 0 \quad (3.15)$$

4. La covarianza de $\hat{\mathbf{x}}$

$$\text{cov}(\hat{\mathbf{x}}) = E[\mathbf{e}\mathbf{e}^T] = R_{xx} - R_{xy}R_{yy}^{-1}R_{yx} \quad (3.16)$$

Por tanto, este estimador tiene menor covarianza que la covarianza a priori R_{xx} .

Para el caso de un modelo lineal

$$\mathbf{Y} = H\mathbf{X} + \mathbf{v} \quad (3.17)$$

donde \mathbf{v} es una variable aleatoria con media cero y con $\text{cov}(\mathbf{v}) = R$, entonces

$$R_{xy} = R_{xx}H^T \quad \text{y} \quad R_{yy} = R \quad \text{y} \quad \mu_y = H\mu_x \quad (3.18)$$

Entonces (3.10) puede escribirse como

$$\mu_{x|y} = \mu_x + R_x H^T R^{-1}(\mathbf{y} - \mu_y) \quad (3.19)$$

Es conveniente escribir la cantidad $K = R_x H^T R^{-1}$, donde K se llama *ganancia de Kalman*. Entonces se tiene,

$$\mu_{x|y} = \mu_x + K(\mathbf{y} - \mu_y) \quad (3.20)$$

3.4. Estimaciones recursivas

A continuación el problema a abordar es el de estimar el estado de un sistema usando las observaciones del mismo, donde el estado evoluciona en presencia de ruido y las observaciones se hacen secuencialmente en presencia de ruido. La notación a usar es la siguiente: con \mathbf{x}_t se hace referencia al parámetro a estimar (en lugar de θ utilizado anteriormente) y con \mathbf{y}_t se indica los datos observados. El problema es estimar el estado \mathbf{x}_t , $t = 0, 1, \dots$ basado en la secuencia de observaciones \mathbf{y}_t , $t = 0, 1, \dots$. En este desarrollo,

se asume que la secuencia de estados \mathbf{x}_t es un proceso aleatorio de Markov, esto es, para cualquier variable aleatoria z que sea función de \mathbf{x}_s , $s \geq t$,

$$f(z | \mathbf{x}_t, \mathbf{x}_{t-1}, \dots, \mathbf{x}_0) = f(z | \mathbf{x}_t) \quad (3.21)$$

En el caso en estudio

$$f(\mathbf{x}_{t+1} | \mathbf{x}_t, \dots, x_0) = f(\mathbf{x}_{t+1} | \mathbf{x}_t) \quad (3.22)$$

Además, se asume que la observación \mathbf{y}_{t+1} depende de \mathbf{x}_{t+1} y posiblemente de ruido aleatorio que es independiente de muestra a muestra, pero es condicionalmente independiente de observaciones anteriores, dado \mathbf{x}_{t+1} ; esto es,

$$f(\mathbf{y}_{t+1} | \mathbf{x}_{t+1}, \mathbf{y}_t, \dots, \mathbf{y}_0) = f(\mathbf{y}_{t+1} | \mathbf{x}_{t+1}) \quad (3.23)$$

En la ecuación anterior, los vectores \mathbf{x}_t e \mathbf{y}_t son aleatorios.

En lo que sigue se adopta la siguiente notación. El conjunto de medidas $\{y_0, \dots, y_t\}$ se representa por Y_t . La notación $\hat{\mathbf{x}}_{t|\tau}$ se utiliza para notar el estimador de Bayes de \mathbf{x}_t , dados los datos $\mathbf{y}_0, \dots, \mathbf{y}_\tau = Y_\tau$. La covarianza del estimador $\hat{\mathbf{x}}_{t|\tau}$ se representa a través de $\hat{P}_{t|\tau}$,

$$P_{t|\tau} = E(\hat{\mathbf{x}}_{t|\tau} - E\hat{\mathbf{x}}_{t|\tau})(\hat{\mathbf{x}}_{t|\tau} - E\hat{\mathbf{x}}_{t|\tau})^T \quad (3.24)$$

Por conveniencia notacional, también se elimina el subíndice que denota las funciones densidad de probabilidad, de forma que ahora son los argumentos los que indican las variables aleatorias,

$$f(\mathbf{x}_{t+1} | Y_{t+1}) = f_{\mathbf{x}_{t+1}|Y_{t+1}}(\mathbf{x}_{t+1} | Y_{t+1}) \quad (3.25)$$

Empezando con la densidad de probabilidad a priori, $f(\mathbf{x}_0)$, la primera observación \mathbf{y}_0 se usa para calcular la distribución a posteriori usando el teorema de Bayes del siguiente modo

$$f(\mathbf{x}_0 | \mathbf{y}_0) = \frac{f(\mathbf{y}_0 | \mathbf{x}_0)}{f(\mathbf{y}_0)} f(\mathbf{x}_0) \quad (3.26)$$

Basándose en $f(\mathbf{x}_0 | \mathbf{y}_0)$, se obtiene un estimador $\hat{x}_{0|0}$. Este es el paso de actualización. Esta densidad se propaga ahora en el tiempo para obtener $f(\mathbf{x}_1 | \mathbf{y}_0)$, de la cual se obtiene el estimador $\hat{x}_{1|0}$.

El siguiente paso es la generalización de este primer paso, esto es, actualizar un estimador condicionado por Y_t a otro condicionado por Y_{t+1} . Desde el punto de vista de la estimación Bayesiana, el problema ahora es determinar la densidad posterior $f(\mathbf{x}_{t+1} | Y_{t+1})$ de forma recursiva partiendo de la

densidad posterior $f(\mathbf{x}_t | Y_{t+1})$. Esto es, se busca encontrar una función F que cumpla

$$f(\mathbf{x}_{t+1} | Y_{t+1}) = F[f(\mathbf{x}_t | Y_t), \mathbf{y}_{t+1}] \quad (3.27)$$

La identificación de la función F dará la función densidad de probabilidad que se buscaba y con ella se obtiene el estimador buscado. El proceso es el siguiente

$$f(\mathbf{x}_{t+1} | Y_{t+1}) = f(\mathbf{x}_{t+1} | Y_t, \mathbf{y}_{t+1}) \quad (3.28)$$

$$= \frac{f(\mathbf{y}_{t+1} | \mathbf{x}_{t+1}, Y_t)}{f(\mathbf{y}_{t+1} | Y_t)} f(\mathbf{x}_{t+1} | Y_t) \quad (3.29)$$

Usando las características de las variables que se definieron al principio,

$$f(\mathbf{y}_{t+1} | \mathbf{x}_{t+1}, Y_t) = f(\mathbf{y}_{t+1} | \mathbf{x}_{t+1}) \quad (3.30)$$

El paso anterior se obtiene teniendo en cuenta

$$f(\mathbf{y}_{t+1} | \mathbf{x}_{t+1}, Y_t) = f(\mathbf{y}_{t+1} | \mathbf{x}_{t+1}, \dots, \mathbf{x}_0, \mathbf{y}_t, \dots, \mathbf{y}_0) \quad (3.31)$$

$$= f(\mathbf{y}_{t+1} | \mathbf{x}_{t+1}, \dots, \mathbf{x}_0) \quad (3.32)$$

$$= f(\mathbf{y}_{t+1} | \mathbf{x}_{t+1}) \quad (3.33)$$

Sustituyendo (3.30) en (3.28) se obtiene,

$$\underbrace{f(\mathbf{x}_{t+1} | Y_{t+1})}_{\text{Posterior}} = \frac{f(\mathbf{y}_{t+1} | \mathbf{x}_{t+1})}{f(\mathbf{y}_{t+1} | Y_t)} \underbrace{f(\mathbf{x}_{t+1} | Y_t)}_{\text{apriori}} \quad (3.34)$$

A la ecuación anterior podemos llamarla de actualización.

El cálculo de la actualización requiere encontrar previamente $f(\mathbf{x}_{t+1} | Y_t)$. Pero para encontrar esa función densidad de probabilidad, previamente hay que dar el paso de propagación. Este paso puede escribirse de la siguiente forma:

$$f(\mathbf{x}_{t+1} | Y_t) = \int f(\mathbf{x}_{t+1} | \mathbf{x}_t, Y_t) f(\mathbf{x}_t | Y_t) d\mathbf{x}_t \quad (3.35)$$

$$= \int f(\mathbf{x}_{t+1} | \mathbf{x}_t) f(\mathbf{x}_t | Y_t) d\mathbf{x}_t \quad (3.36)$$

la igualdad anterior se logra usando la propiedad de Markov par \mathbf{x}_t y el hecho de que en realidad, \mathbf{y}_t depende de \mathbf{x}_t .

3.5. Filtrado de Kalman

El filtro de Kalman es un estimador recursivo utilizado para estimar el estado de una ecuación de estado lineal y que cambia con el tiempo, en la cual los estados tienen ruido aditivo y las observaciones se hacen en presencia de ruido. Existe más de una forma de obtener las expresiones que definen al filtro de Kalman, cada una de las cuales con su propio criterio de optimización. En lo que sigue se va a desarrollar la basada en la teoría de Bayes recursiva

3.5.1. Modelo de espacio de estados

El filtro de Kalman es una aplicación de los resultados generales de la estimación secuencial que se han presentado en el apartado anterior para el caso de un sistema cuya variable de estado es discreta en el tiempo, que tiene ruido y con observaciones con ruido también,

$$\mathbf{x}_{t+1} = A_t \mathbf{x}_t + \mathbf{w}_t \quad (3.37)$$

$$\mathbf{y}_t = C_t \mathbf{x}_t + \mathbf{v}_t \quad (3.38)$$

Como ya se ha dicho, el sistema que viene representado por el espacio de estados puede evolucionar en el tiempo. A continuación se enumeran las distintas suposiciones que se hacen sobre el sistema.

- Por conveniencia todos los procesos se suponen reales.
- El ruido que aparece en la ecuación de estados \mathbf{w}_t tiene de media cero y su covarianza es

$$E w_t w_\tau^T = Q_t \delta_{t,\tau} \quad (3.39)$$

El ruido entre muestras no está correlacionado. Como se está haciendo un estudio Bayesiano del filtro de Kalman, se supondrá que \mathbf{w}_t es Gaussiano.

- El ruido de las observaciones \mathbf{v}_t tiene media cero y su covarianza es

$$E v_t v_\tau^T = R_t \delta_{t,\tau} \quad (3.40)$$

El ruido entre muestras no está correlacionado. Al igual que para el ruido en el caso anterior, se supondrá que \mathbf{v}_t es de naturaleza Gaussiana.

- Se asume que los dos tipos de ruidos anteriores están correlacionados

$$Ew_t v_\tau^T = M_t \delta_{t,\tau} \quad \forall t, \tau \quad (3.41)$$

Aunque se va a comenzar suponiendo que ambos ruidos no están correlacionados, esta restricción se eliminará después.

- Hay una condición inicial, o función densidad de probabilidad a priori, sobre la variable \mathbf{x}_0 con media $\mu_x [0]$ y covarianza Π_0 . De nuevo, para el caso Bayesiano, se asume que \mathbf{x}_0 es Gaussiano. Además, todo el desarrollo matemático que sigue se simplifica notablemente si se asume que $\mu_x [0] = 0$.

El problema de filtrado de Kalman puede presentarse como sigue: dada una secuencia de medidas $\mathbf{y}_0, \mathbf{y}_1, \dots$, se pretende determinar una secuencia de estimadores del estado del sistema \mathbf{x}_t de forma recursiva y con complejidad computacional aceptable.

3.5.2. Ecuaciones del Filtro de Kalman

Una vez sentadas las bases sobre las que se va a desarrollar el proceso de obtención de las ecuaciones del filtro de Kalman desde un punto de vista Bayesiano, se va a mostrar el modo de deducción de las mismas. Es importante comenzar recordando que un estimador de Bayes de \mathbf{x}_t es aquel que trata de encontrar la media de \mathbf{x}_t dado el conjunto de las observaciones.

Las ecuaciones clave de este proceso desde el punto de vista Bayesiano son las del paso de propagación (3.35)

$$f(x_{t+1} | Y_t) = \int f(x_{t+1} | x_t) f(x_t | Y_t) dx_t \quad (3.42)$$

desde la que el estimador es propagado usando la ecuación de actualización del estado en el futuro, y

$$f(\mathbf{x}_{t+1} | Y_{t+1}) = \frac{f(\mathbf{y}_{t+1} | \mathbf{x}_{t+1})}{f(\mathbf{y}_{t+1} | Y_t)} f(\mathbf{x}_{t+1} | Y_t) \quad (3.43)$$

que es el paso de actualización de la medida.

En primer lugar se va a tratar de encontrar fórmulas explícitas para el paso de propagación en (3.42).

1. La función densidad de probabilidad $f(\mathbf{x}_t | Y_t)$ corresponde al estimador de \mathbf{x}_t , dadas las medidas hasta el tiempo t . Bajo la suposición de ruido

Gaussiano y usando la notación que se ha introducido en esta sección, la variable aleatoria \mathbf{x}_t condicionada por \mathbf{Y}_t es también Gaussiana,

$$\mathbf{x}_t | Y_t \sim \mathcal{N}(\hat{\mathbf{x}}_{t|t}, P_{t|t}) \quad (3.44)$$

2. La función densidad de probabilidad $f(\mathbf{x}_{t+1} | \mathbf{x}_t)$ se obtiene de (3.37), teniendo en cuenta que \mathbf{x}_{t+1} condicionada por \mathbf{x}_t se distribuye de la siguiente forma

$$\mathbf{x}_{t+1} | \mathbf{x}_t \sim \mathcal{N}(A_t \mathbf{x}_t, Q_t) \quad (3.45)$$

Introduciendo (3.44) en (3.45) y (3.42) e integrando, se llega a que $\mathbf{x}_t | Y_t$ es Gaussiano con media

$$\hat{\mathbf{x}}_{t+1|t} = A_t \hat{\mathbf{x}}_{t|t} \quad (3.46)$$

y covarianza

$$P_{t+1|t} = A_t P_{t|t} A_t^T + Q_t \quad (3.47)$$

Con la ecuación (3.46) se tiene un medio de propagar el estimador en el tiempo, en ausencia de medidas, y la ecuación (3.47) muestra que, sin medidas, el estimador de la covarianza crece en el tiempo.

El siguiente paso es analizar las expresiones relacionadas con el proceso de actualización. En este caso se trata de una actualización Bayesiana de una variable Gaussiana aleatoria. La media de $\mathbf{x}_{t+1} | Y_{t+1}$ se obtiene como en C, en la que la media de la distribución a priori se actualiza de la siguiente forma:

$$\hat{\mathbf{x}}_{t+1|t+1} = E[\mathbf{x}_{t+1} | Y_t] = \hat{\mathbf{x}}_{t+1|t} + R_{xy, Y_t} R_{yy, Y_t}^T (\mathbf{y}_{t+1} - E[\mathbf{y}_{t+1} | Y_t]) \quad (3.48)$$

En la expresión aparecen una serie de valores cuyas expresiones se analizan y detallan a continuación:

1. La expresión R_{xy, Y_t} denota la correlación condicionada por Y_t :

$$R_{xy, Y_t} = E[(\mathbf{x}_{t+1} - E[\mathbf{x}_{t+1}]) (\mathbf{y}_{t+1} - E[\mathbf{y}_{t+1}])^T | Y_t] \quad (3.49)$$

Entonces se tiene

$$R_{xy, Y_t} = E \left[(\mathbf{x}_{t+1} - \hat{\mathbf{x}}_{t+1|t}) (C_{t+1} (\mathbf{x}_{t+1} - \hat{\mathbf{x}}_{t+1|t}) + \mathbf{v}_{t+1})^T | Y_t \right] \quad (3.50)$$

$$= P_{t+1|t} C^T \quad (3.51)$$

donde se ha usado la definición de $P_{t+1|t}$ dada en la ecuación (3.24).

2. La notación R_{yy, Y_t} se utiliza para la covarianza de \mathbf{y}_t condicionada por Y_t :

$$R_{yy, Y_t} = E \left[(\mathbf{y}_{t+1} - E[\mathbf{y}_{t+1}]) (\mathbf{y}_{t+1} - E[\mathbf{y}_{t+1}])^T | Y_t \right] \quad (3.52)$$

$$= E \left[(C_{t+1} (\mathbf{x}_{t+1} - \hat{\mathbf{x}}_{t+1|t}) + \mathbf{v}_{t+1}) (C_{t+1} (\mathbf{x}_{t+1} - \hat{\mathbf{x}}_{t+1|t}) + \mathbf{v}_{t+1})^T | Y_t \right] \quad (3.53)$$

$$= C_{t+1} P_{t+1|t} C_{t+1}^T + R_{t+1} \quad (3.54)$$

$$(3.55)$$

3. La media $E[\mathbf{y}_{t+1} | Y_t]$ es igual a $C_{t+1} \hat{\mathbf{x}}_{t+1|t}$.

Si ponemos todo lo anterior junto se llega a la siguiente ecuación de actualización:

$$\hat{\mathbf{x}}_{t+1|t+1} = \hat{\mathbf{x}}_{t+1|t} + P_{t+1|t} C_{t+1}^T (C_{t+1} P_{t+1|t} C_{t+1}^T + R_{t+1})^{-1} (\mathbf{y}_{t+1} - C_{t+1} \hat{\mathbf{x}}_{t+1|t}) \quad (3.56)$$

Es conveniente introducir la siguiente magnitud,

$$K_{t+1} = P_{t+1|t} C_{t+1}^T (C_{t+1} P_{t+1|t} C_{t+1}^T + R_{t+1})^{-1} \quad (3.57)$$

de forma que la actualización de la media puede escribirse de la siguiente forma,

$$\hat{\mathbf{x}}_{t+1|t+1} = \hat{\mathbf{x}}_{t+1|t} + K_{t+1} (\mathbf{y}_{t+1} - C_{t+1} \hat{\mathbf{x}}_{t+1|t}) \quad (3.58)$$

La cantidad K_t se denomina *ganancia de Kalman*.

Si ahora se considera la covarianza de $\mathbf{x}_{t+1} | Y_{t+1}$ que es la varianza del error del estimador $\tilde{\mathbf{x}}_{t+1|t+1} = \hat{\mathbf{x}}_{t+1} - \hat{\mathbf{x}}_{t+1|t+1}$. Esta covarianza puede hallarse por identificación sabiendo que un modelo condicional Gaussiano

$$\text{cov}(\mathbf{X} | \mathbf{Y}) = R_{xx} - R_{xy} R_{yy}^{-1} R_{yx} \quad (3.59)$$

Para aplicar este resultado hay que identificar la variable aleatoria \mathbf{X} en la expresión anterior con el estimador $\hat{\mathbf{x}}_{t+1|t}$; y la observación \mathbf{Y} con la observación \mathbf{y}_{t+1} . La covarianza R_{xx} de la expresión anterior es análoga a $P_{t+1|t}$. La matriz R_{xy} es análoga a la matriz $R_{xy,Y}$ y R_{yy} es análoga a $R_{yy,Y}$. Sustituyendo en (3.50) y (3.56), se tiene

$$P_{t+1|t+1} = P_{t+1|t} - P_{t+1|t} C_{t+1}^T (C_{t+1} P_{t+1|t} C_{t+1}^T + R_{t+1})^{-1} C_{t+1} P_{t+1|t} \quad (3.60)$$

$$= (I - K_{t+1} C_{t+1}^T) P_{t+1|t} \quad (3.61)$$

Lo anterior completa la obtención de las ecuaciones para el filtro de Kalman. El proceso se puede esquematizar de la siguiente forma: se parte de un estimador inicial $\hat{x}_{0|-1}$ con una covarianza inicial $P_{-1|-1}$, para cada observación \mathbf{y}_t , $t = 0, 1, \dots$, el estimador del estado se actualiza usando los siguientes pasos:

1. **Extrapolación del estimador del estado:** $A_t \hat{\mathbf{x}}_{t|t}$
2. **Extrapolación del error de la covarianza:** $P_{t+1|t} = A_t P_{t|t} A_t^T + Q_t$
3. **Ganancia de Kalman:** $K_{t+1} = P_{t+1|t} C_{t+1}^T (C_{t+1} P_{t+1|t} C_{t+1}^T + R_{t+1})^{-1}$
4. **Actualización del estimador del estado:**
 $\hat{\mathbf{x}}_{t+1|t+1} = \hat{\mathbf{x}}_{t+1|t} + K_{t+1} (\mathbf{y}_{t+1} - C_{t+1} \hat{\mathbf{x}}_{t+1|t})$
5. **Actualización del error de la covarianza:** $P_{t+1|t+1} = (I - K_{t+1} C_{t+1}^T) P_{t+1|t}$

3.6. Filtros de Kalman Extendidos

3.6.1. Modelo de espacio de estados para sistemas no lineales

Las ecuaciones que definen el modelo de espacio de estados para el caso de un sistema general no lineal y en tiempo discreto son las siguientes:

$$\mathbf{x}_{t+1} = A(\mathbf{x}_t, t) + \mathbf{w}_t \quad (3.62)$$

$$\mathbf{y}_t = C(\mathbf{x}_t, t) + \mathbf{v}_t \quad (3.63)$$

para $t = 0, 1, \dots$ con $\{\mathbf{w}_t, t = 0, 1, \dots\}$ y $\{\mathbf{v}_t, t = 0, 1, \dots\}$ representan los ruidos de media cero asociados a las ecuaciones de estado y medida respectivamente. El problema no lineal general es muy difícil y no existe ninguna solución general para él. Al no ser A lineal, no se garantiza que la distribución de

estados sea Gaussiana, y si A o C son no lineales, entonces no puede garantizarse que la distribución del valor esperado condicional $\hat{\mathbf{x}}_{t+1|\tau}$ sea Gaussiana. Por lo tanto, no se puede, en general, obtener el estimador como función de los dos primeros momentos de la distribución condicional solamente. Una solución general requeriría la propagación de la distribución condicional entera. Por tanto, no se puede obtener una solución exacta fácilmente y hay que recurrir a soluciones aproximadas que se obtienen usando linealizaciones.

Históricamente el *Filtro de Kalman Extendido* (EKF) ha sido el filtro aproximado más utilizado. El EKF linealiza las ecuaciones de estado y de medida en primer lugar para después usar un filtro de Kalman para resolver el problema linealizado.

3.6.2. Linealización en sistemas dinámicos

Los sistemas dinámicos no lineales pueden linealizarse usando series de Taylor. La siguiente Linealización es para sistemas discretos en el tiempo. Para linealizar las ecuaciones de un sistema dinámico

$$\mathbf{x}[t+1] = \mathbf{f}(\mathbf{x}[t], \mathbf{u}[t], t) \quad (3.64)$$

$$\mathbf{y}[t] = \mathbf{h}(\mathbf{x}[t], \mathbf{u}[t], t) \quad (3.65)$$

donde \mathbf{f} y \mathbf{h} . Se asume la existencia de una entrada $\mathbf{u}_0[t]$, un estado $\mathbf{x}_0[t]$, y una observación $\mathbf{y}_0[t]$ nominales. También se supone que la entrada $\mathbf{u}[t]$ y el estado $\mathbf{x}[t]$ no difieren mucho de la entrada y del estado nominales. Entonces la Linealización se hace truncando la expansión de \mathbf{f} . Si

$$\mathbf{u}[t] = \mathbf{u}_0[t] + \mathbf{u}_\delta[t] \quad (3.66)$$

$$\mathbf{x}[t] = \mathbf{x}_0[t] + \mathbf{x}_\delta[t] \quad (3.67)$$

$$\mathbf{y}[t] = \mathbf{y}_0[t] + \mathbf{y}_\delta[t] \quad (3.68)$$

donde $\mathbf{u}_\delta[t]$ es un vector con determinado número de componentes. Utilizando lo anterior, la ecuación del sistema no lineal dinámico puede escribirse como:

$$\mathbf{x}_0[t+1] + \mathbf{x}_\delta[t+1] = f(\mathbf{x}_0[t] + \mathbf{x}_\delta[t], \mathbf{u}_0[t] + \mathbf{u}_\delta[t], t) \quad (3.69)$$

Suponiendo que $f \in \mathcal{C}^1$, se expande f en serie de Taylor. Como $\mathbf{x}_\delta[t]$ y $\mathbf{u}_\delta[t]$ se suponen pequeñas, se mantiene sólo el término de primer orden de la serie. La i -ésima componente de \mathbf{f} se expande de la siguiente forma

$$f_i(\mathbf{x}_0 + \mathbf{x}_\delta, \mathbf{u}_0 + \mathbf{u}_\delta, t) \approx f_i(\mathbf{x}_0, \mathbf{u}_0, t) + \frac{\partial f_i}{\partial \mathbf{x}_1}(\mathbf{x}_0, \mathbf{u}_0, t)\mathbf{x}_{\delta,1} + \dots + \frac{\partial f_i}{\partial \mathbf{x}_p}(\mathbf{x}_0, \mathbf{u}_0, t)\mathbf{x}_{\delta,p} \quad (3.70)$$

$$+ \frac{\partial f_i}{\partial \mathbf{u}_1}(\mathbf{x}_0, \mathbf{u}_0, t)\mathbf{u}_{\delta,1} + \dots + \frac{\partial f_i}{\partial \mathbf{u}_p}(\mathbf{x}_0, \mathbf{u}_0, t)\mathbf{u}_{\delta,p} \quad (3.71)$$

Si se usa la siguiente notación:

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_2}{\partial x_1} & \cdots & \frac{\partial f_p}{\partial x_1} \\ \frac{\partial f_1}{\partial x_2} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_p}{\partial x_2} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_p}{\partial x_1} & \frac{\partial f_p}{\partial x_1} & \cdots & \frac{\partial f_p}{\partial x_1} \end{bmatrix} \quad (3.72)$$

La matriz $\frac{\partial \mathbf{f}}{\partial \mathbf{x}}$ de dimensión $(p \times p)$ es el *Jacobiano* de \mathbf{f} con respecto a \mathbf{x} . De igual forma, $\frac{\partial \mathbf{f}}{\partial \mathbf{u}}$ es el Jacobiano de \mathbf{f} con respecto a \mathbf{u} . Teniendo en cuenta las expresiones para los Jacobianos, la Linealización realizada de la expansión de Taylor puede escribirse como

$$\mathbf{x}_0[t+1] + \mathbf{x}_\delta[t+1] = \mathbf{f}(\mathbf{x}_0[t], \mathbf{u}_0[t], t) + \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x}_0[t], \mathbf{u}_0[t], t)\mathbf{x}_\delta[t] + \frac{\partial \mathbf{f}}{\partial \mathbf{u}}(\mathbf{x}_0[t], \mathbf{u}_0[t], t)\mathbf{u}_\delta[t] \quad (3.73)$$

La solución nominal cumple la ecuación original del sistema dinámico

$$\mathbf{x}_\delta[t+1] = \mathbf{f}(\mathbf{x}_0[t], \mathbf{u}_0[t], t) \quad (3.74)$$

Si se le resta la ecuación anterior a (3.73), se obtiene la ecuación lineal dependiente del tiempo para el sistema dinámico,

$$\mathbf{x}_\delta[t+1] = \frac{\partial \mathbf{f}(\mathbf{x}_0[t], \mathbf{u}_0[t], t)}{\partial \mathbf{x}}\mathbf{x}_\delta[t] + \frac{\partial \mathbf{f}(\mathbf{x}_0[t], \mathbf{u}_0[t], t)}{\partial \mathbf{u}}\mathbf{u}_\delta[t] \quad (3.75)$$

Si se usa la siguiente notación:

$$A(t) = \frac{\partial \mathbf{f}(\mathbf{x}_0[t], \mathbf{u}_0[t], t)}{\partial \mathbf{x}} \text{ y } B(t) = \frac{\partial \mathbf{f}(\mathbf{x}_0[t], \mathbf{u}_0[t], t)}{\partial \mathbf{u}}$$

se obtiene:

$$\mathbf{x}_\delta[t+1] = A(t)\mathbf{x}_\delta[t] + B(t)\mathbf{u}_\delta[t]$$

De igual forma se puede operar para linealizar la ecuación de observación. La expresión que se obtiene es:

$$\mathbf{y}_\delta[t] = C(t)\mathbf{x}_\delta[t] + D(t)\mathbf{u}_\delta[t]$$

donde $C(t)$ y $D(t)$ tienen la siguiente forma:

$$C(t) = \frac{\partial \mathbf{h}(\mathbf{x}_0[t], \mathbf{u}_0[t], t)}{\partial \mathbf{x}} \quad \text{y} \quad D(t) = \frac{\partial \mathbf{h}(\mathbf{x}_0[t], \mathbf{u}_0[t], t)}{\partial \mathbf{u}}$$

3.6.3. Ecuaciones del Filtro de Kalman Extendido

Una vez que linealizadas las ecuaciones para el sistema dinámico no lineal, las ecuaciones que definen el filtro de Kalman extendido se obtienen usando un filtro de Kalman para estas ecuaciones linealizadas

Actualización de la medida

$$\hat{\mathbf{x}}_{t+1|t+1} = \hat{\mathbf{x}}_{t+1|t} + K_{t+1} (\mathbf{y}_{t+1} - C(\hat{\mathbf{x}}_{t+1|t}, t)) \quad (3.76)$$

$$P_{t+1|t+1} = (\mathbf{I} - K_{t+1}C_{t+1}) P_{t+1|t} \quad (3.77)$$

donde

$$K_{t+1} = P_{t+1|t} C_{t+1}^T (C_{t+1} P_{t+1|t} C_{t+1}^T + R_{t+1})^{-1} \quad (3.78)$$

$$C_t = \left. \frac{\partial C(\mathbf{x}, t)}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{t+1|t}} \quad (3.79)$$

Actualización temporal

$$\hat{\mathbf{x}}_{t+1|t} = A(\hat{\mathbf{x}}_{t|t}, t) \quad (3.80)$$

$$P_{t+1|t} = A_t P_{t|t} A_t^T + Q_t \quad (3.81)$$

donde

$$A_t = \left. \frac{\partial A(\mathbf{x}, t)}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{t+1|t}} \quad (3.82)$$

Hay dos fuentes que pueden generar errores en los EKF. La primera es la Linealización del modelo en cada paso temporal. La segunda es que, debido a esta Linealización, la suposición de que la función de distribución a priori o a posteriori en un instante de tiempo dado es Gaussiana con la media y covarianza que dan las ecuaciones puede ser falsa. Las no linealidades en el modelo pueden dar lugar a que las distribuciones a priori y a posteriori no sean Gaussianas y ni siquiera unimodales, y por tanto, la media y varianza que se calculan serían sólo aproximaciones de las verdaderas. Para solventar de algún modo este tipo de problemas, se han propuesto soluciones como los Filtros de Kalman iterativos [87], [5], los cambios de coordenadas [3] o la inclusión de términos cuadráticos [50]. Para un estudio de cada uno de estos procesos, sus ventaja e inconvenientes puede consultarse [33].

3.7. Introducción al Filtrado de Partículas

Consideremos un sistema cuya representación en el espacio de estado viene dada por las ecuaciones (3.1) y (3.2). Como ya se ha comentado, el objetivo es estimar secuencialmente el estado \mathbf{x}_t a partir de las medidas \mathbf{y}_t . También se ha comentado ya que existen principalmente tres funciones de distribución de probabilidad de interés que son:

Función de distribución de Filtrado: $p(\mathbf{x}_t | \mathbf{y}_{0:t})$. En este caso, el objetivo es obtener recursivamente $p(\mathbf{x}_t | \mathbf{y}_{0:t})$ de $p(\mathbf{x}_{t-1} | \mathbf{y}_{0:t-1})$.

Función de distribución de Suavizado: $p(\mathbf{x}_t | \mathbf{y}_{0:T})$, $T > t$. En este caso, el objetivo es obtener recursivamente $p(\mathbf{x}_t | \mathbf{y}_{0:T})$ de $p(\mathbf{x}_{t-1} | \mathbf{y}_{0:T})$.

Función de distribución Predictiva: $p(\mathbf{x}_{t+l} | \mathbf{y}_{0:t})$, $l \geq 1$. En este caso, el objetivo es obtener recursivamente $p(\mathbf{x}_{t+l} | \mathbf{y}_{0:t+l})$ de $p(\mathbf{x}_{t+l-1} | \mathbf{y}_{0:t-1})$.

Los algoritmos que obtienen estas distribuciones exactamente se denominan algoritmos óptimos. Sin embargo, en la mayoría de los casos prácticos, estos algoritmos son imposibles de implementar, principalmente porque las actualizaciones de las funciones de distribuciones requieren calcular una serie de integrales que no pueden hacerse de forma analítica o bien una serie de sumas que no pueden hacerse debido al gran número de términos de las mismas.

Para la distribución conjunta a posteriori de $\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_t$, en el caso de que se tengan muestras independientes del ruido, se puede escribir

$$p(\mathbf{x}_{0:t} | \mathbf{y}_{0:t}) \propto p(\mathbf{x}_0 | \mathbf{y}_0) \prod_{k=1}^t p(\mathbf{x}_k | \mathbf{y}_k) p(\mathbf{x}_k | \mathbf{x}_{k-1}) \quad (3.83)$$

Partiendo de la expresión anterior, la fórmula que relaciona $p(\mathbf{x}_{0:t} | \mathbf{y}_{0:t})$ con $p(\mathbf{x}_{0:t-1} | \mathbf{y}_{0:t-1})$ es

$$p(\mathbf{x}_{0:t} | \mathbf{y}_{0:t}) = \frac{p(\mathbf{x}_t | \mathbf{y}_t) p(\mathbf{x}_t | \mathbf{x}_{t-1})}{p(\mathbf{y}_t | \mathbf{y}_{0:t-1})} p(\mathbf{x}_{0:t-1} | \mathbf{y}_{0:t-1}) \quad (3.84)$$

Son precisamente los términos del cociente los que llevan asociados cálculos muy complejos que generalmente son analíticamente intratables. Es por esto que hay que recurrir a métodos aproximados.

Como ya se ha comentado, en el Filtrado de Partículas, las distribuciones se aproximan usando medidas aleatorias discretas que se llaman *partículas* y una serie de pesos que se asignan a las mismas. Para comprender este proceso, se propone el siguiente ejemplo en el que se supone que la función de distribución de interés es $p(x)$. Entonces, esta función de distribución se aproxima a través del siguiente conjunto de medidas aleatorias:

$$X = \{x^{(j)}, w^{(j)}\}_{j=1}^J \quad (3.85)$$

donde $x^{(j)}$ son las partículas, $w^{(j)}$ los pesos asociados a las mismas y J es el número de partículas que se usan en la aproximación. De esta forma, X aproxima la distribución $p(x)$ de la siguiente forma:

$$p(x) \approx \sum_{j=1}^J w^{(j)} \delta(x - x^{(j)}) \quad (3.86)$$

donde $\delta(\bullet)$ es la función delta de Dirac. Con esta aproximación, el cálculo de los valores esperados que antes conllevaba el cálculo de complicadas integrales, ahora se reduce al cálculo de simples sumas.

$$E(g(X)) = \int g(x)p(x)dx \Rightarrow E(g(X)) \approx \sum_{j=1}^J w^{(j)} g(x^{(j)}) \quad (3.87)$$

3.7.1. Importance Sampling

Con las ideas expuestas hasta este punto, se pueden comprender fácilmente las bases del Filtrado de Partículas. Sin embargo hay un punto importante del que no se ha hablado hasta aquí y del que va a tratar esta sección: el Importance Sampling.

Se ha explicado que usando Filtrado de Partículas, una distribución se aproxima a través de una serie de muestras aleatorias a las que se les asocia cierto peso relacionado con su probabilidad. El verdadero problema aparece a la hora de plantearse el proceso de muestreo. Si se pudieran generar (supongamos que su número es J) partículas directamente de la función de distribución $p(x)$, cada una de ellas tendría el mismo peso igual a $1/J$. Pero esta situación no es la que se encuentra normalmente. Por lo general, no es posible muestrear directamente de la función de distribución $p(x)$ así que hay que recurrir a la introducción de una función de distribución llamada “Importance Function”. De modo que la obtención de las muestras $x^{(j)}$ se lleva a cabo haciendo uso de esta función de distribución que denotaremos $\pi(x)$. A estas muestras se les asocian unos pesos que se calculan según la expresión:

$$w^{*(j)} = \frac{p(x)}{\pi(x)} \quad (3.88)$$

los pesos anteriores no están normalizados, tras su normalización su expresión es:

$$w^{(j)} = \frac{w^{*(j)}}{\sum_{i=1}^J w^{*(i)}} \quad (3.89)$$

El modo de obtención de estas muestras de un instante de tiempo al siguiente es otro punto importante a tratar. Existen una serie de métodos llamados *Métodos Estáticos* (como por ejemplo la mayoría de los famosos esquemas Monte Carlo Cadenas de Markov [17], [18]). En estos métodos, se hace un tratamiento separado para la distribución en estudio para cada instante de tiempo de modo que se repite el mismo proceso iterativo para cada t . Esto significa que todas las muestras aleatorias que se obtienen para un instantes de tiempo t , no se usan de nuevo cuando el sistema evoluciona al siguiente instante de tiempo.

Sin embargo, cuando el sistema varía lentamente (esto es, cuando la distancia en L^2 entre $p(\mathbf{x}_{t-1}|\mathbf{y}_{t-1})$ y $p(\mathbf{x}_{t-1}|\mathbf{y}_t)$ es pequeña), las muestras aleatorias correspondientes al instante de tiempo $t-1$ pueden usarse para obtener las muestras aleatorias en el instante t de modo que se mejore la eficiencia del método.

Una vez expuestas las bases del proceso de Importance Sampling, se va a hacer un desarrollo para llegar a una expresión que permita obtener los pesos asociados a las distintas muestras de forma recursiva. Se comienza suponiendo que la distribución a posteriori $p(\mathbf{x}_{0:t-1}|\mathbf{y}_{0:t-1})$ se aproxima a través del conjunto de medidas aleatorias $\mathcal{X}_{t-1} = \{\mathbf{x}_{0:t-1}^{(j)}, w_{0:t-1}^{(j)}\}$. El objetivo entonces sería obtener \mathcal{X}_t a partir de \mathcal{X}_{t-1} teniendo en cuenta las observaciones \mathbf{y}_t . Como ya se ha comentado, los métodos secuenciales de Importance Sampling realizan lo anterior generando partículas $\mathbf{x}_t^{(j)}$ y añadiéndolas a $\mathbf{x}_{0:t-1}^{(j)}$ para formar $\mathbf{x}_t^{(j)}$. El siguiente paso es actualizar los pesos w_t^j de forma que el conjunto \mathcal{X}_t permita obtener buenos estimadores para las variables que no se conocen en el instante de tiempo t .

La única restricción que este tipo de algoritmos de Filtrado de Partículas impone sobre la función de importancia es que ésta pueda factorizarse de la siguiente forma:

$$\pi(\mathbf{x}_{0:t}|\mathbf{y}_{0:t}) = \pi(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{0:t}) \pi(\mathbf{x}_{0:t-1}|\mathbf{y}_{0:t-1}) \quad (3.90)$$

Para llegar a unas ecuaciones generales, supongamos que en el instante $t-1$ las muestras:

$$\mathbf{x}_{0:t-1}^{(j)} \sim \pi(\mathbf{x}_{0:t-1}|\mathbf{y}_{0:t-1}) \quad (3.91)$$

y

$$w_{t-1}^{(j)} \propto \frac{p\left(\mathbf{x}_{0:t-1}^{(j)} \mid \mathbf{y}_{0:t-1}\right)}{\pi\left(\mathbf{x}_{0:t-1}^{(j)} \mid \mathbf{y}_{0:t-1}\right)} \quad (3.92)$$

entonces puede agregarse la muestra $\mathbf{x}_t^{(j)}$ obtenidas aleatoriamente de

$$\mathbf{x}_t^{(j)} \sim \pi\left(\mathbf{x}_t \mid \mathbf{x}_{0:t-1}^{(j)}, \mathbf{y}_{0:t}\right) \quad (3.93)$$

al conjunto $\mathbf{x}_{0:t-1}^{(j)}$. Y si los pesos actualizados se calculan de acuerdo con la siguiente expresión:

$$w_t^{(j)} \propto \frac{p\left(\mathbf{y}_t \mid \mathbf{x}_t^{(j)}\right) p\left(\mathbf{x}_t^{(j)} \mid \mathbf{x}_{0:t-1}^{(j)}\right)}{\pi\left(\mathbf{x}_t^{(j)} \mid \mathbf{x}_{0:t-1}^{(j)}, \mathbf{y}_{0:t}\right)} w_{t-1}^{(j)} \quad (3.94)$$

En este punto se plantea un gran problema, el de la elección de la Función de Importancia. De esta elección dependerá en gran medida la bondad de los resultados obtenidos al aplicar este algoritmo. En general, cuanto más parecida sea esta Función de Importancia a la distribución en estudio mejor es la aproximación. Debido al gran abanico de posibilidades que surge ante la elección de una posible Función de Importancia, este proceso ha sido ampliamente estudiado. Gran cantidad de funciones de importancia con diferentes características aparecen en la literatura. De entre ellas cabe destacar las que se muestran a continuación.

Prior distribution

En este caso, la forma de la función de importancia es:

$$\pi(\mathbf{x}_t \mid \mathbf{x}_{0:t-1}, \mathbf{y}_t) = p(\mathbf{x}_t \mid \mathbf{x}_{0:t-1}) \quad (3.95)$$

Este es el tipo de función de importancia más usado debido precisamente a la simplicidad del cálculo que implica se elección. Esto se puede ver fácilmente en la expresión tan sencilla a la que queda reducida la expresión general para la obtención de los pesos actualizados.

$$w_t^{(j)} \propto w_{t-1}^{(j)} p(\mathbf{y}_t \mid \mathbf{x}_t^{(j)}) \quad (3.96)$$

El mayor inconveniente que encierra el uso de este tipo de Función de Importancia se debe a que no tiene en cuenta la información que aportan las observaciones en el instante de tiempo t . Como puede verse en la expresión (3.95), la Función de Importancia sólo depende de la variable de estado en el instante de tiempo t y en los instantes anteriores.

Distribución Óptima

La expresión para la Función de Importancia en el caso de que se escoja como tal la distribución óptima es:

$$\pi(\mathbf{x}_t | \mathbf{x}_{0:t-1}, \mathbf{y}_t) = p(\mathbf{x}_t | \mathbf{x}_{0:t-1}, \mathbf{y}_t) \quad (3.97)$$

Su nombre se debe a que precisamente minimiza la varianza de los pesos w_t^j condicionada por $\mathbf{x}_{0:t-1}^j$ y las observaciones \mathbf{y}_t .

La expresión a la que se llega en este caso para la actualización de los pesos es algo más compleja que la que se obtenía en el caso anterior:

$$\begin{aligned} w_t^{(j)} &\propto w_{t-1}^{(j)} \frac{p(\mathbf{y}_t | \mathbf{x}_t^{(j)}) p(\mathbf{x}_t^{(j)} | \mathbf{x}_{t-1}^{(j)})}{p(\mathbf{x}_t^{(j)} | \mathbf{x}_{t-1}^{(j)}, \mathbf{y}_t)} \\ w_t^{(j)} &\propto w_{t-1}^{(j)} p(\mathbf{y}_t | \mathbf{x}_{t-1}^{(j)}) \end{aligned} \quad (3.98)$$

En este caso se cuenta con la mejora que supone la introducción de la información aportan las observaciones más recientes con respecto al caso anterior. Sin embargo esta mejora en muchos modelos se convierte en un inconveniente ya que la expresión $w_{t-1}^{(j)} p(\mathbf{y}_t | \mathbf{x}_{t-1}^{(j)})$ lleva asociada el cálculo de integrales cuya evaluación es imposible

Otras distribuciones

Las anteriores no constituyen más que una muestra de las posibles elecciones para las funciones de distribución. A lo largo de la literatura relacionada con este tema pueden encontrarse gran cantidad de distribuciones subóptimas que muestran mejores o peores comportamientos dependiendo del problema al que se apliquen. En este momento aún no se ha podido encontrar un esquema de elección de la Función de Importancia que garantice buenos resultados. Una revisión bastante completa de algunos de tipos de distribuciones subóptimas puede encontrarse en [26].

3.7.2. Importance Resampling

Uno de los mayores problemas del uso de filtrado de partículas es que las medidas discretas aleatorias sufren de degeneración de forma rápida. Esto es, en muy pocas iteraciones se llega a una situación en la que se tiene un grupo muy pequeño de partículas con pesos asociados grandes y el resto (que constituye la mayoría) con pesos despreciables. Esta degeneración puede reducirse en primer lugar usando funciones de importancia adecuadas y en segundo lugar mediante la técnica de Resampling.

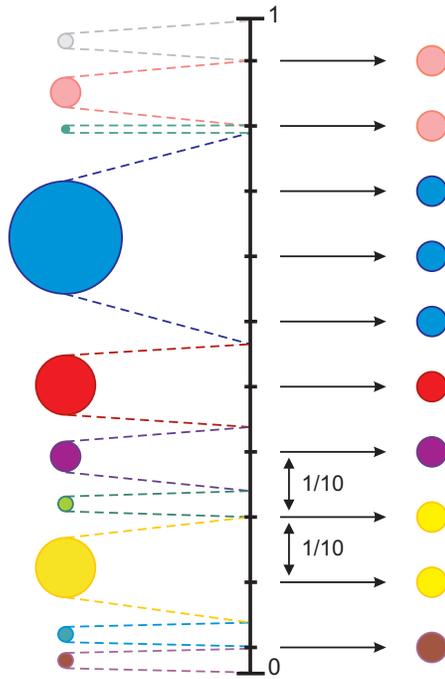


Figura 3.1: Esquema del proceso de Resampling

La idea del Resampling es bastante sencilla y en esencia consiste en obtener J partículas $\mathbf{x}_t^{(j)}$ de la distribución discreta \mathcal{X}_t y después asignar a cada una de estas partículas el mismo peso que a las demás ($1/J$).

En la Figura 3.1 aparece un esquema sencillo que muestra el proceso de Resampling. Cada uno de los círculos representa una partícula. El diámetro de cada círculo es proporcional a su peso. En general, lo que ocurre es que tras aplicar el algoritmo de Resampling las partículas con pesos grandes sobreviven y se multiplican mientras que las que tienen asociado un peso pequeño suelen desaparecer. En la Figura anterior puede observarse este fenómeno. La tercera partícula tiene un radio mucho mayor que el de las dos primeras, es por esto que las dos primeras no se mantienen y la tercera permanece y se multiplica. Otro punto importante a resaltar es que tras el proceso de Resampling, todas las partículas pasan a tener el mismo peso asociado, eso es, aparecen representadas como círculos de igual diámetro.

En la literatura existen distintas técnicas de Resampling. Dos de los métodos más usados son los aleatorios y los basados en procedimientos Bootstrap. Los métodos Bootstrap toman una serie de muestras combinadas como representación de la distribución a la que pertenecen los datos y crean las nuevas muestras (con sustitución) tras el Resampling usando el conjunto anterior como pseudo-distribución. Los métodos aleatorios también comienzan con

los datos originales pero en lugar de obtener muestras con sustitución, el procedimiento consiste en sistemática o aleatoriamente reordenar los datos y calcular el test estadístico apropiado en cada reordenamiento.

La principal diferencia entre los métodos anteriores es la sustitución. Sin embargo, además de la sustitución existen otras diferencias fundamentales. El método de Bootstrap se centra fundamentalmente en la estimación de parámetros relacionados con la distribución e intenta obtener información sobre ésta. Las técnicas aleatorias no se interesan por parámetros relacionados con la distribución. Su interés se centra principalmente en los mecanismos que hacen que los datos se distribuyan de una forma dada.

Otro punto importante a tratar dentro de este apartado es el de las condiciones que tienen que darse para que se realice el proceso de Resampling. No está claro porqué se necesita hacer Resampling en un determinado instante de tiempo t . Existen gran variedad de publicaciones que tratan este problema en la literatura, de entre ellas cabe destacar [52]. En lo que sigue sólo se pretenden dar algunas ideas sobre este tema. En primer lugar, si los pesos son constantes o prácticamente constantes para todos los instantes (esto ocurre cuando las muestras se obtienen de la distribución π_t directamente) el Resampling sólo reduce el número de muestras distintas que introduce variación Monte Carlo extra. Esto sugiere que debe realizarse Resampling cuando el coeficiente de variación cv_t^2 , para un determinado peso es pequeño. Este coeficiente de variación está relacionado con el tamaño relativo entre los pesos de las distintas muestras para un instante de tiempo dado. Como se argumenta en [56], el tamaño efectivo de las muestras es inversamente proporcional a $1 + cv_t^2$. Y es precisamente haciendo uso de este tamaño efectivo con lo que se determina cuando hacer Resampling. El procedimiento consiste básicamente en escoger un umbral de modo que si el tamaño efectivo de alguna de las partículas es menor que ese umbral, se hace Resampling y si no, el Resampling no se incluye como paso en el algoritmo. Aquí surge de nuevo otra disquisición y es como establecer un umbral apropiado. Sobre este tema también existen distintas publicaciones en la literatura donde se comparan distintos umbrales. Sin embargo, este sigue siendo un problema abierto dentro del Filtrado de Partículas.

Para finalizar, se presenta una figura en la que puede verse de forma gráfica todo el proceso que se ha expuesto hasta aquí para el filtrado de partículas (Figura 3.2). Así, puede verse que en el instante $t - 1$ se tiene una serie de partículas representadas por círculos todas con el mismo peso por eso tienen el mismo diámetro. Usando la función de distribución se les asocia a estas partículas un peso determinado y es por eso que en la siguiente representación los círculos aparecen con distintos diámetros. El siguiente paso es de Resampling de modo que tras el mismo se tendrá el mismo número de

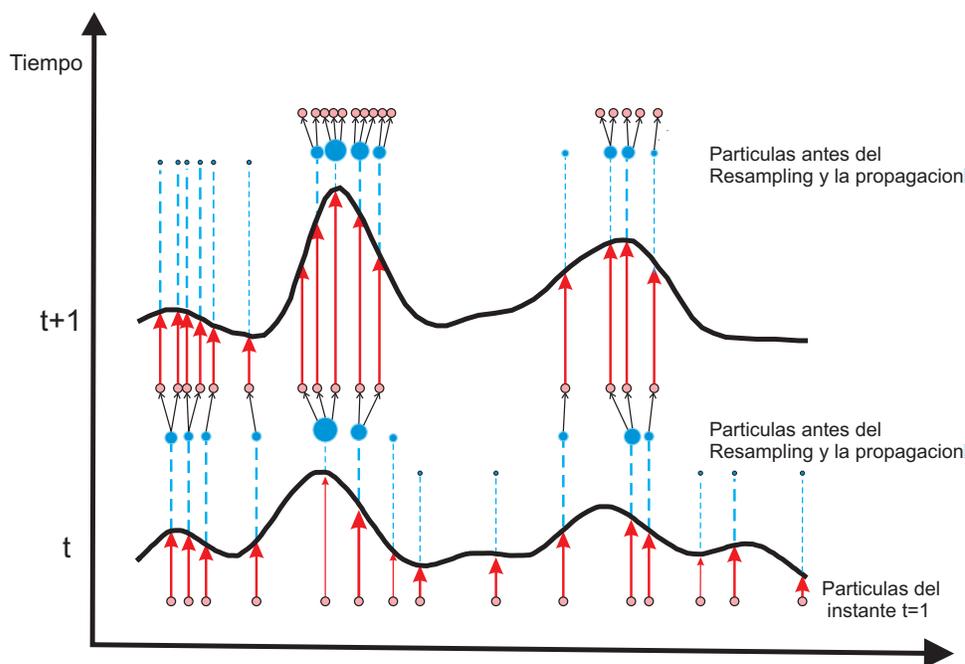


Figura 3.2: Esquema del Filtrado de partículas

partículas pero con igual peso. Se puede ver aquí de nuevo que, por lo general, las partículas con pesos grandes no sólo permanecen sino que también se replican mientras que las partículas con pesos pequeños desaparecen. En este punto se ha llegado a la situación de la que se partía y el procedimiento a seguir sería totalmente análogo al ya descrito.

3.8. Algoritmo M -Estocástico

Aunque los algoritmos genéricos de filtrado de partículas anteriores poseen gran potencia a la hora de dar información blanda (estadística) precisa y rendimiento casi óptimo, en realidad, son muy ineficientes y el proceso de Resampling puede dar lugar a soluciones subóptimas. En primer lugar, en la iteración m -ésima en filtrado de partículas, el espacio de variables de la distribución posterior deseada tiene dimensión K^m , donde K es el número de posibles trayectorias que pueden derivarse de una dada. En el caso de m pequeños, se podría calcular la distribución a posteriori exacta. Cuando esto es verdad, no se necesita muestreo para los m primeros pasos, y entonces el filtrado de partículas debería aplicarse en la iteración $m + 1$ -ésima. En segundo lugar, de entre las J trayectorias, podría haber múltiples copias de la misma trayectoria. La idea original de tener múltiples copias de la misma

trayectoria es que esas múltiples copias podrían producir partículas con diferentes valores en posteriores pasos y por tanto, se aumentaría la diversidad de las partículas generadas de esa trayectoria. A pesar de que esto puede ser cierto para espacios de muestreo continuos ya que el tamaño de los mismos es infinito, sin embargo, no es siempre el caso para espacios de muestreo discretos. En nuestro problema, hay sólo K posibles valores del espacio de muestreo que cada trayectoria puede tomar. Cuando la probabilidad másica sólo se concentra en unos pocos valores, las partículas generadas podrían ser réplicas de esos pocos valores muestreados. Como resultado, múltiples copias de la misma trayectoria en el estado presente podrían producir partículas con el mismo valor en las siguientes iteraciones y por tanto, seguir siendo copias de la misma trayectoria. Este fenómeno se convierte en más evidente después del Resampling porque en él se producen aún más copias de la misma trayectoria. Por tanto, producir diversas copias de la misma trayectoria es un gasto innecesario de recursos en este caso. Recursos que deberían emplearse en el procesado de otras trayectorias. Una de las posibles soluciones a este problema de mantener múltiples copias de una trayectoria dada podría ser combinarlas en una sola. Sin embargo, aún se necesitan medidas para obtener diversidad en las partículas generadas de una trayectoria. Otro problema es el debido al uso del Resampling, que empobrece la diversidad de la muestra y podría descartar trayectorias óptimas en las primeras iteraciones por sus pequeños pesos. Por último, supongamos que no hay dos trayectorias iguales entre las J . Para hacer evolucionar la j -ésima trayectoria de la iteración m a la $m + 1$, hay que calcular los dos candidatos en primer lugar y después obtener una muestra de entre ellos. De acuerdo con esto, no importa cómo sea el peso asociado a esta trayectoria, sólo una de ellas será seleccionada para el instante $m + 1$. Sin embargo, sería deseable poder retener más de un candidato de una trayectoria cuando su peso es grande. En total habría $2xJ$ trayectorias de las que se tienen que elegir las J finales. Por tanto, es preferible muestrear las J de todo el conjunto de candidatos de acuerdo con los pesos de cada uno de los candidatos que sería $w_m^{(j)} \lambda_m^{(j)}$ donde $\lambda_m^{(j)}$ es la probabilidad de transición de la trayectoria j -ésima a cada una de las dos que se crean a partir de ella. Este inconveniente de los algoritmos de filtrado de partículas ha sido propuesto en numerosas ocasiones a lo largo de la literatura.

En esta tesis se propone el uso de una implementación de Filtrado de Partículas alternativa que se usa bajo el nombre de Algoritmo *M* Estocástico que soluciona los inconvenientes mencionados anteriormente. Un esquema de dicho algoritmo se muestra a continuación:

Algoritmo M Estocástico

Inicialización: Establecer $J=1$;

En la m -ésima iteración,

- *Expansión de Trayectorias:*
Para j desde 1 hasta J Para k desde 1 hasta K
 - Añadir a_k a la trayectoria $s_{1:m-1}^{(j)}$ y obtener la trayectoria extendida $x_{1:m}^{((j-1)K+k)}$.
 - Evaluar la probabilidad de transición $\lambda_j^{(j)}$ y calcular el peso $v_m^{((j-1)K+k)} = w_{m-1}^j \lambda_k^j$.
- *Muestreo de Trayectorias:*
 - Si $JxK \leq M$,
Establecer $s_{1:m}^{(i)} = x_{1:m}^{(i)}$ y $v_{1:m}^{(i)} = w_{1:m}^{(i)}$ con $i = 1, \dots, JxK$. Establecer $J = JxK$
 - En cualquier otro caso
Muestrear M trayectorias de las JxK trayectorias extendidas $x_{1:m}^{(i)}$ con probabilidad proporcional a $v_{1:m}^{(i)}$ usando el algoritmo de Resampling óptimo para obtener $s_{1:m}^{(j)}$ y $w_{1:m}^{(j)}$ para $j = 1, \dots, JxK$. Establecer $J = M$;
- Normalizar los pesos $w_m^{(j)} = w_m^{(j)} / \sum_{j=1}^J w_m^{(j)}$

El algoritmo de Resampling óptimo se propuso en [33] y es óptimo en el sentido de que el error cuadrático medio entre los pesos originales y los pesos muestreados es mínimo. El algoritmo para muestrear M trayectorias de las JxK trayectorias extendidas $x_{1:m}^{(i)}$ pesadas con el conjunto $v_{1:m}^{(i)}$ se resume a continuación:

Algoritmo de Resampling óptimo

- *Ordenación:*
Reordenar los pesos $v_{1:m}^{(i)}$ de forma que el primero sea el mayor
- *Cálculo de c :*
Cálculo de c como solución de la ecuación $\sum_{i=1}^{i=JxK} \min(cv_m^{(i)}) = M$
- *Herencia*
Determinar el mayor índice l tal que $cv_m^{(l)} > 1$. Heredar la trayectoria y los pesos estableciendo $s_{1:m}^{(j)} = x_{1:m}^{(j)}$ y $v_{1:m}^{(j)} = w_{1:m}^{(j)}$ con $j = 1, \dots, l$.
- *Muestreo Estratificado*
 - Establecer $U = (r + l)/M$, $\mathbf{I} = [1, \dots, JxK]$, $j = l + 1$, y $Q = l/M$ donde $r \sim U(0, 1)$.
 - Mientras que $U < 1$
si $Q > U$,
Establecer $U = U + 1/M$, $l = l + 1$, $s_{1:m}^{(l+1)} = x_{1:m}^{(l)}$ y $w_{1:m}^{(l)} = 1/c$
en otro caso
Escoger un índice k para $\{j, j + 1, \dots, JxK\}$. Establecer $i = \mathbf{I}_k$ y $\mathbf{I}_k = \mathbf{I}_j$;
Establecer $Q = Q + cv_m^{(i)}/M$ y $j = j + 1$.

En [33] se muestra que el cálculo de c puede realizarse de forma recursiva y que la complejidad computacional del muestreo óptimo es $O(JxK)$ que es del mismo orden que el muestreo residual. Aunque algoritmos alternativos como el de muestreo residual pueden usarse en su lugar, una característica atractiva del algoritmo de Resampling óptimo es que produce M trayectorias distintas y por tanto, mantiene la mayor diversidad posible entre las partículas. Y se ha encontrado gracias a diversos experimentos que la diversidad de las partículas es esencial para obtener un buen rendimiento. Un algoritmo de partículas muy parecido fue propuesto en [33]. Sin embargo, el algoritmo M Estocástico es más eficiente.

Por otro lado, aunque este algoritmo se parece al Algoritmo M clásico, hay diferencias claves entre ambos. El Algoritmo M es determinista, en él sólo se mantienen las M trayectorias con los pesos mayores y el objetivo es encontrar la mejor trayectoria mientras que en el algoritmo M Estocástico la selección de trayectorias se basa en un muestreo aleatorio y el objetivo es encontrar las mejores muestras aleatorias que aproximen la distribución posterior. Y es precisamente a esto último a lo que este algoritmo debe su nombre.

3.9. Filtros de Kalman de Mezcla

3.9.1. Modelos para sistemas dinámicos lineales condicionados

Un *modelo dinámico lineal condicionado* (CDLM) queda definido a través de las siguientes ecuaciones:

$$\begin{aligned}x_t &= H_\lambda x_t + W_\lambda w_t \\y_t &= G_\lambda x_t + V_\lambda v_t\end{aligned}\quad (3.99)$$

donde $\Lambda_t = \lambda$, $w_t \sim \mathcal{N}(0, I)$, $v_t \sim \mathcal{N}(0, I)$ y todos los coeficientes de las matrices son conocidos dado λ . El conjunto Λ_t puede ser discreto o continuo y es un indicador latente del proceso con cierta estructura probabilística

3.9.2. Método de Mezcla de filtrado de Kalman

La notación que se emplea a continuación es la siguiente: $\mathbf{y}_t = (y_1, y_2, \dots, y_t)$, $\Lambda_t = (\Lambda_1, \Lambda_2, \dots, \Lambda_t)$ y λ_t y λ_s son realizaciones de Λ_t y Λ_s respectivamente. Entonces,

$$p(x_t | \mathbf{y}_t) = \int p(x_t | \lambda_t, \mathbf{y}_t) p(\lambda_t | \mathbf{y}_t) d\lambda_t \quad (3.100)$$

donde $p(x_t | \lambda_t, \mathbf{y}_t) \sim \mathcal{N}\{\mu_t(\lambda_t), \Sigma_t(\lambda_t)\}$, en la que $(\mu_t(\lambda_t), \Sigma_t(\lambda_t))$ puede obtenerse usando un filtro de Kalman dada la trayectoria λ . La idea de FKM es partir de un conjunto de muestras de indicadores a los que se les asocia ciertos pesos

$$\mathcal{S}_t = \left\{ \left(\lambda_t^{(1)}, \mathbf{w}_t^{(1)} \right), \dots, \left(\lambda_t^{(m)}, \mathbf{w}_t^{(m)} \right) \right\} \quad (3.101)$$

para representar la distribución $p(\Lambda_t | \mathbf{y}_t)$, y entonces se usa esa mezcla aleatoria de distribuciones Gaussianas

$$\frac{1}{W_t} \sum_{j=1}^m w_t^{(j)} N \left\{ \mu_t(\lambda_t^{(j)}), \Sigma_t(\lambda_t^{(j)}) \right\} \quad (3.102)$$

donde $W_t = \sum_{j=1}^m w_t^{(j)}$, para aproximar la distribución $p(x_t | \mathbf{y}_t)$. De este modo los valores esperados de cualquier magnitud representada a través de la función integrable $h(\bullet)$

$$\widehat{E} \{h(x_t) | \mathbf{y}_t\} = \frac{1}{W_t} \sum_{j=1}^m w_t^{(j)} \int h(x) \varphi \left(x; \mu_t(\lambda_t^{(j)}), \Sigma_t(\lambda_t^{(j)}) \right) dx \quad (3.103)$$

donde φ es la función densidad Gaussiana.

Mientras que los filtros de Monte Carlo normales usan muestras pesadas de la variable de estado $\left\{x_t^{(j)}, w_t^{(j)}\right\}_{j=1}^J$, para aproximar $p(x_t|\mathbf{y}_t)$, los FKM operan en el espacio de la variable indicadora: esto es equivalente a marginalizar x_t . Los filtros normales de Monte Carlo aproximan recursivamente la función de distribución posterior de x_t usando una serie de muestras discretas, mientras que los FKM aproximan esta función con una mezcla de distribuciones Gaussianas cuyo número aumenta con t .

Si se denota por $KF_t^{(j)} = \left(\mu_t(\lambda_t^{(j)}), \Sigma_t(\lambda_t^{(j)})\right)$ al conjunto de las medias a posteriori y las matrices de covarianza de x_t , condicionada por \mathbf{y}_t y dado $\lambda_t^{(j)}$. Los elementos del conjunto anterior se calculan con un filtro de Kalman. En los siguientes puntos se resume el proceso recursivo de actualización que se realiza en MKF

1. Generar $\lambda_{t+1}^{(j)}$ de la distribución $\pi_{t+1}\left(\lambda_{t+1} \mid \lambda_t^{(j)}, \mathbf{KF}_t^{(j)}\right)$;
2. Obtener $KF_{t+1}^{(j)}$ usando un filtro de Kalman, condicionado por $\left\{KF_t^{(j)}, y_{t+1}, \lambda_{t+1}^{(j)}\right\}$,

$$\begin{aligned} P_{t+1} &= H_{\lambda_{t+1}} \Sigma_t H'_{\lambda_{t+1}} + W_{\lambda_{t+1}} W'_{\lambda_{t+1}} \\ S_{t+1} &= G_{\lambda_{t+1}} P_t G'_{\lambda_{t+1}} + V_{\lambda_{t+1}} V'_{\lambda_{t+1}} \\ \mu_{t+1} &= H_{\lambda_{t+1}} \mu_t + P_{t+1} G'_{\lambda_{t+1}} S_{t+1}^{-1} (y_{t+1} - G_{\lambda_{t+1}} H_{\lambda_{t+1}} \mu_t) \\ \Sigma_{t+1} &= P_{t+1} - P_{t+1} G'_{\lambda_{t+1}} S_{t+1}^{-1} G_{\lambda_{t+1}} P_{t+1} \end{aligned} \quad (3.104)$$

3. Actualizar el nuevo peso según la expresión $w_{t+1}^{(j)} = w_t^{(j)} \times u_{t+1}^{(j)}$ donde

$$u_{t+1}^{(j)} = \frac{p\left(\lambda_t^{(j)}, \lambda_{t+1}^{(j)} \mid \mathbf{y}_{t+1}\right)}{p\left(\lambda_t^{(j)} \mid \mathbf{y}_t\right) \pi_{t+1}\left(\lambda_{t+1}^{(j)} \mid \lambda_t^{(j)}, \mathbf{KF}_t^{(j)}\right)} \quad (3.105)$$

4. Por último se procede a volver a muestrear según los criterios que se hayan escogido para llevar a cabo esta operación.

Cuando Λ_t toma valores dentro de un conjunto finito y discreto \mathcal{I} , se suele escoger $\pi_{t+1}\left(\lambda_{t+1} \mid \lambda_t^{(j)}, \mathbf{KF}_t^{(j)}\right) = p_{t+1}\left(\lambda_{t+1} \mid \lambda_t^{(j)}, \mathbf{KF}_t^{(j)}, \mathbf{y}_{t+1}\right)$. En este caso se llega a la siguiente simplificación de cálculo

$$w_{t+1}^{(j)} \propto w_t^{(j)} \sum_{i \in \mathcal{I}} p\left(y_{t+1} \mid \Lambda_{t+1} = i, KF_t^{(j)}\right) p\left(\Lambda_{t+1} = i \mid \lambda_t^{(j)}\right) \quad (3.106)$$

Si ahora se aplica este resultado a las ecuaciones generales para la actualización que se han obtenido anteriormente, se llega a una particularización de las mismas para el caso en que la variable indicadora pertenezca a un conjunto finito y discreto que es el caso que se va a estudiar en el siguiente capítulo de esta tesis.

Capítulo 4

Aplicación del Algoritmo M-Estocástico a Detección Multiusuario

Las técnicas de detección multiusuario han ido ganando en relevancia desde su introducción [84]. Este Capítulo se centra en la detección multiusuario ciega en canales Rayleigh con desvanecimiento plano y rápido para sistemas síncronos. En particular, se supone conocida *a priori* la estadística de segundo orden del canal. Basándose en esto se adopta una aproximación matemática en la que se usa un modelo ARMA. Esta aproximación permite usar un Modelo de Espacio de Estados Dinámico (MEED) que a su vez permite usar el filtrado de Kalman como solución. Recientemente se han aplicado el filtrado de Kalman de mezcla y los algoritmos de filtrado de partículas a la detección ciega de códigos convolucionales [19], códigos space-time trellis [96], e incluso detección ciega multiusuario [72] en canales desvanescentes. Se ha demostrado que el uso de filtrado de Kalman de mezcla (FKM) puede reducir bastante el error de propagación en implementaciones dirigidas por decisión

y, por tanto, permite mejorar el rendimiento. Sin embargo, el uso de FKM para el caso de detección óptima multiusuario ciega implica una complejidad exponencial con el número de usuarios [72]. Por tanto, este algoritmo es muy complejo incluso para un número moderado de usuarios. A continuación se presenta un punto de vista totalmente diferente de los sistemas multiusuario y se utiliza un nuevo Modelo de Espacio de Estados de Tiempo-Observación (MEETO) que fue propuesto en [94]. Aunque el MEETO es equivalente al MEED, posee importantes ventajas sobre este último gracias a las que se consigue una implementación más eficiente. Una vez presentado este modelo, se propone el uso de un algoritmo de Filtrado de Partículas más eficiente denominado Algoritmo M -Estocástico (AME), que ha demostrado una reducción adicional de la complejidad en comparación con el algoritmo clásico de Filtrado de Partículas manteniendo un comportamiento similar.

La organización de este capítulo es la siguiente: en la Sección 4.1 se plantea el problema de detección ciega multiusuario. En la Sección 4.2 se describe el nuevo Modelado de espacio de estados de tiempo-observación propuesto. Las ecuaciones básicas del proceso de detección Bayesiana óptima bajo esta formulación así como su análisis se presentan en la Sección 4.3. En la Sección 4.4 se introduce una solución dirigida por decisión para resolver el problema que se propone, además se analiza la complejidad computacional de este algoritmo así como su rendimiento. En la Sección de 4.5 se presenta la aplicación de un algoritmo genérico de Filtrado de Partículas al problema de la detección ciega multiusuario bajo la formulación nueva que se propone. El detector basado en el algoritmo M Estocástico que se propone en esta tesis aparece desarrollado en la Sección 4.6. Finalmente, en la Sección 4.7 se introduce brevemente el detector “Genie Aided”, que, al basarse en un algoritmo no ciego, se utiliza como límite inferior en la comparación entre los distintos detectores. En la Sección 4.8 se comparan los rendimientos de los distintos detectores presentados en este capítulo desde un punto de vista teórico. En la Sección 4.9 se introduce el problema de los escenarios con “near-far”, o situaciones en las que diferentes usuarios transmiten con distinta potencia ya que los algoritmos presentados en este capítulo van a probarse en este tipo de situaciones. Por último, en la Sección 4.10 se presentan y discuten los resultados de las simulaciones y en la Sección 4.11 se comentan las conclusiones más destacadas.

4.1. Formulación del Problema

Consideremos un sistema síncrono CDMA con ganancia de procesamiento C y K usuarios. Llamemos T a la duración del símbolo y sea $s_k(t)$ la forma de onda determinista y normalizada asignada al k -ésimo usuario. Entonces, en el n -ésimo intervalo de símbolo, la señal recibida $y(t)$ puede expresarse como la suma de K formas de onda moduladas antipodalmente más ruido, i.e.,

$$y(t) = \sum_{k=1}^K a_{n,k} b_{n,k} s_k(t) + u(t) \quad t \in [(n-1)T, nT] \quad (4.1)$$

donde $b_{n,k} \in \{-1, +1\}$ es el bit con modulación BPSK transmitido por el k -ésimo usuario, $a_{k,n}$ el coeficiente desvanescente del k -ésimo usuario que da información sobre el estado del canal (IEC), y $u(t)$ es el ruido complejo blanco Gaussiano y aditivo añadido cuya varianza es σ^2 . La correlación cruzada entre las formas de onda de los usuarios viene dada por la matriz de correlación cruzada \mathbf{R} , donde el elemento r_{k_1, k_2} representa la correlación cruzada entre la forma de onda del k_1 -ésimo y el k_2 -ésimo usuarios y se define de acuerdo con la siguiente expresión

$$r_{k_1 k_2} = \langle s_{k_1}, s_{k_2} \rangle = \int_{(n-1)T}^{nT} s_{k_1}(t) s_{k_2}(t) dt. \quad (4.2)$$

El canal para cada usuario se considera como Rayleigh desvanescente plano de forma que puede adoptarse un modelo ARMA para modelar su correlación temporal [19, 44, 90]. Dado un proceso ARMA(r_1, r_2), la IEC del k -ésimo usuario en el n -ésimo intervalo $a_{k,n}$ puede representarse de acuerdo con

$$\begin{aligned} a_{n,k} &+ \phi_{k,1} a_{n-1,k} \cdots \phi_{k,r_1} a_{n-r_1,k} \\ &= \rho_{k,0} v_{n,k} + \cdots + \rho_{k,r_2} v_{n-r_2,k} \end{aligned} \quad (4.3)$$

donde $v_{n,k}$ es un proceso i.i.d., aleatorio, complejo y Gaussiano que dirige el proceso ARMA, $\{\phi_{k,1}, \dots, \phi_{k,r_1}\}$ y $\{\rho_{k,1}, \dots, \rho_{k,r_2}\}$ son los coeficientes AR y MA del modelo. Se asume que se conoce *a priori* la estadística de segundo orden del canal desvanescente, y por tanto, los coeficientes del modelo ARMA pueden calcularse de antemano adaptando el espectro de (4.2) al del proceso desvanescente. Por conveniencia, asumimos $r_1 = r_2 = r$; en otro caso se podrían añadir ceros a los coeficientes para hacer iguales los órdenes. Una forma equivalente de la ecuación (4.1) viene dada por el conjunto de estadísticos suficientes representados por la salida de un filtro adaptativo,

$$y_{n,k} = \langle y(t), s_k(t) \rangle = \int_{(n-1)T}^{nT} y_n(t) s_k(t) dt. \quad (4.4)$$

El conjunto de salidas del filtro adaptativo, $\mathbf{y}_n = [y_{n,1}, \dots, y_{K,n}]^\top$, donde $(\cdot)^\top$ es la matriz traspuesta, puede representarse en forma vectorial o matricial de acuerdo con

$$\mathbf{y}_n = \mathbf{R}\mathbf{A}_n\mathbf{b}_n + \mathbf{u}_n \quad (4.5)$$

donde $\mathbf{A}_n = \text{diag}\{a_{n,1}, \dots, a_{n,K}\}$ es la matriz diagonal de información del estado del canal, $\mathbf{b}_n = [b_{n,1}, \dots, b_{n,K}]^\top$ es el vector de datos de los usuarios, y \mathbf{u}_n es el vector de ruido complejo Gaussiano con componentes reales e imaginarias independientes y con matriz de covarianza igual a $\sigma^2\mathbf{R}$. El objetivo es realizar una detección secuencial de los símbolos sin conocer la IEC $a_{n,k}$, esto es, realizar detección multiusuario ciega.

4.2. Modelado de espacio de estados de tiempo-observación

Una representación matemática sucinta de un sistema que cambia en el tiempo es el Modelo de Espacio de Estados Dinámico (MEED). La representación de sistemas CDMA en espacio de estados para el caso de canales planos desvanescentes puede encontrarse en la literatura existente [90] y puede expresarse como

$$\begin{cases} \mathbf{h}_{k,n} &= \mathbf{Q}_k\mathbf{h}_{k,n-1} + \mathbf{g}v_{k,n} & \forall k \\ a_{k,n} &= \boldsymbol{\rho}_k^\top\mathbf{h}_{k,n} & \forall k \\ \mathbf{y}_n &= \mathbf{R}\mathbf{A}_n\mathbf{b}_n + \mathbf{u}_n \end{cases} \quad (4.6)$$

donde $\mathbf{h}_{n,k}^\top = [h_{n,k} \dots h_{n-r,k}]$ es el vector de estado del canal de dimensiones $(r+1) \times 1$, $\boldsymbol{\rho}_k^\top = [\rho_{k,0} \dots \rho_{k,r}]$,

$$\mathbf{Q}_k = \begin{bmatrix} -\phi_{k,1} & \dots & -\phi_{k,r} & 0 \\ 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 1 & 0 \end{bmatrix}, \quad \text{y } \mathbf{g} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

En (4.6), $\mathbf{h}_{k,n} \forall k$ y \mathbf{b}_n son las variables desconocidas las que se pretende estimar. Notar que las observaciones \mathbf{y}_n no son lineales en $\mathbf{h}_{k,n} \forall k$ y \mathbf{b}_n , y por tanto usando filtrado de Kalman no puede llegarse a una solución óptima. En realidad, la solución óptima se puede obtener mediante el llamado filtro de Kalman de tipo splitting, donde, en el tiempo n , se necesitan 2^n filtros de Kalman. La complejidad de este filtro de Kalman es exponencial en el tiempo y en el número de usuarios y por tanto computacionalmente prohibitiva. Así, en su lugar pueden usarse algoritmos de Filtrado de Partículas para

obtener buenas aproximaciones de la solución óptima con una complejidad inferior. Los algoritmos de filtrado de partículas en (4.6) incorporados con filtrado de Kalman se propusieron en [72]. Sin embargo, como se mencionó al principio de este capítulo, debido a la estructura de (4.6), las partículas de \mathbf{b}_n deben de muestrearse conjuntamente, y la complejidad llega a ser exponencial con el número de usuarios. De modo que se llega de nuevo a complejidades prohibitivas desde un punto de vista computacional cuando hay un gran número de usuarios lo que significa que los algoritmos de filtrado de partículas no son apropiados en aplicaciones prácticas.

Para solventar esta dificultad, se utiliza el siguiente Modelo de Espacio de Estados de Tiempo-Observación para el sistema (MEETO) [94].

Para desarrollar el MEETO, se comienza con una factorización de Cholesky de la matriz de correlación cruzada \mathbf{R} de la siguiente forma:

$$\mathbf{R} = \mathbf{F}^T \mathbf{F}.$$

donde \mathbf{F} está definida de forma única como una matriz $K \times K$ triangular inferior. Multiplicando por la derecha $(\mathbf{F}^T)^{-1}$ la salida del filtro adaptativo se obtiene:

$$\begin{aligned} \bar{\mathbf{y}}_n &= (\mathbf{F}^T)^{-1} \mathbf{y}_n \\ &= \mathbf{F} \mathbf{A}_n \mathbf{b}_n + \bar{\mathbf{u}}_n \end{aligned}$$

o equivalentemente

$$\bar{\mathbf{y}}_n = \mathbf{F} \mathbf{B}_n \mathbf{a}_n + \bar{\mathbf{u}}_n \quad (4.7)$$

donde $\mathbf{B}_n = \text{diag}\{b_{n,1}, \dots, b_{n,K}\}$ es la matriz diagonal de datos de los usuarios, y $\mathbf{a}_n = [a_{n,1}, \dots, a_{n,K}]$ es el vector de la IEC de dimensiones $K \times 1$. Como la matriz de covarianza $\bar{\mathbf{u}}_n$ se convierte en $E[\bar{\mathbf{u}}_n \bar{\mathbf{u}}_n^T] = \sigma^2 \mathbf{F}^{-T} \mathbf{R} \mathbf{F}^{-1} = \sigma^2 \mathbf{I}$, donde \mathbf{I} es la matriz identidad, $\bar{\mathbf{y}}_n$ se denomina salida del filtro adaptativo de blanqueado (FAB). El siguiente paso es definir un vector del canal de dimensiones $K(r+1) \times 1$ como $\mathbf{h}_n = [\mathbf{h}_{1,n}^T \dots \mathbf{h}_{K,n}^T]^T$ y la transición para el canal es

$$\mathbf{h}_n = \mathbf{Q} \mathbf{h}_{n-1} + \mathbf{G} \mathbf{v}_n \quad (4.8)$$

donde $\mathbf{v}_n = [v_{1,n}, \dots, v_{K,n}]^T$, $\mathbf{Q} = \text{diag}(\mathbf{Q}_1, \dots, \mathbf{Q}_K)$ y $\mathbf{G} = \text{diag}(\underbrace{\mathbf{g}, \dots, \mathbf{g}}_K)$

son las matrices de dimensiones $K(r+1) \times K(r+1)$ y $K(r+1) \times K$ respectivamente. Usando el vector anteriormente definido, el vector \mathbf{a}_n puede expresarse en función de \mathbf{h}_n de forma compacta como

$$\mathbf{a}_n = \mathbf{P} \mathbf{h}_n \quad (4.9)$$

donde $\mathbf{P} = \text{diag}(\boldsymbol{\rho}_1^\top, \dots, \boldsymbol{\rho}_K^\top)$ tiene dimensiones $K \times K(r + 1)$. El siguiente paso es sustituir \mathbf{a}_n en (4.7) usando (4.9), se tiene

$$\bar{\mathbf{y}}_n = \mathbf{F}\mathbf{B}_n\mathbf{P}\mathbf{h}_n + \bar{\mathbf{u}}_n. \quad (4.10)$$

Si se denota la k -ésima fila de \mathbf{F} mediante \mathbf{f}_k^\top , la k -ésima salida del FAB $\bar{\mathbf{y}}_n$ puede escribirse del siguiente modo

$$\bar{y}_{n,k} = \mathbf{f}_k^\top \mathbf{B}_n \mathbf{P} \mathbf{h}_n + \bar{u}_{n,k} \quad (4.11)$$

donde $\bar{u}_{n,k}$ es el k -ésimo elemento de $\bar{\mathbf{u}}_n$. Ahora, en lugar de considerar el sistema evolucionando solamente en el tiempo, imaginemos un sistema que progresa alternativamente a lo largo del tiempo y de las observaciones del FAB $\bar{y}_{n,k}$. Esta nueva manera de evolucionar para el sistema aparece descrita en la Figura 4.1. Para describir este nuevo sistema se define un nuevo índice que surge de la unión del índice temporal t y del de observación k . Este será el índice de tiempo-observación l , y es fácil ver que $l = (n - 1)K + k$. Esta conversión es reversible, o en otras palabras, también se puede calcular k y n si se tiene l teniendo en cuenta que $k = \text{mod}(l, K)$ y $n = (l - k)/K + 1$, donde $\text{mod}(k, K)$ es la operación módulo k de K . En la siguiente descripción del MEETO usando índices l , todos los k y n se suponen conocidos ya que pueden obtenerse del l correspondiente.

El siguiente paso es introducir una matriz auxiliar de dimensiones $K \times K$ $\tilde{\mathbf{B}}_l = \text{diag}\{b_{n,1}, \dots, b_{n,k}, 0 \dots, 0\}$. La representación en espacio de estados para el nuevo índice de tiempo-observación para el sistema con índice l puede construirse como

$$\begin{cases} \mathbf{h}_l = \begin{cases} \mathbf{Q}\mathbf{h}_{l-1} + \mathbf{G}\mathbf{v}_l & \text{si } k = 1 \\ \mathbf{h}_{l-1} & \text{si } k \neq 1 \end{cases} \\ \bar{y}_l = \mathbf{f}_k^\top \tilde{\mathbf{B}}_l \mathbf{P} \mathbf{h}_l + \bar{u}_l \end{cases} \quad (4.12)$$

y a las ecuaciones (4.12) se las denomina MEETO. Notar que (4.12) y (4.6) describen el mismo sistema. Sin embargo, hay diferencias fundamentales entre los dos modelos. A diferencia de (4.6), las transiciones de estado de \mathbf{h}_l en el MEETO varían en el tiempo, esto es, para diferentes l , se aplican diferentes transiciones. Específicamente, cuando $k = 1$, o equivalentemente, n aumenta en una unidad, \mathbf{h}_l se actualiza de acuerdo al modelo ARMA del canal, y a la inversa, cuando $k \neq 1$ y n mantienen los valores que tenían para $l - 1$, \mathbf{h}_l se considera estático. Además, en el MEETO, el número de bits que no se conocen cambia con l , y en particular, sólo se incluye una nueva señal desconocida $b_{n,k}$ cada vez que l se incrementa en una unidad. Por tanto, si se asume una detección perfecta en $l - 1$, i.e., $b_{n,1}, \dots$, y $b_{n,k-1}$ se conocen

exactamente, entonces hay sólo un bit de un usuario para detectar. Notar que en el MEED convencional (4.6), tienen que detectarse K bits desconocidos todos juntos cuando el sistema evoluciona en el tiempo al instante n .

Esta es la idea principal del modelo que da lugar a soluciones eficientes de tipo Filtrado de Partículas. Sin embargo, es importante hacer hincapié en que la decisión sobre $b_{n,k}$ (excepto para $k = K$) no está completa en l . Ya que las observaciones desde y_{l+1} hasta y_{l+r} con $r = K - k$ contienen información sobre $b_{n,k}$, la decisión final debe hacerse en $l + r$, o en general, cuando $k = K$.

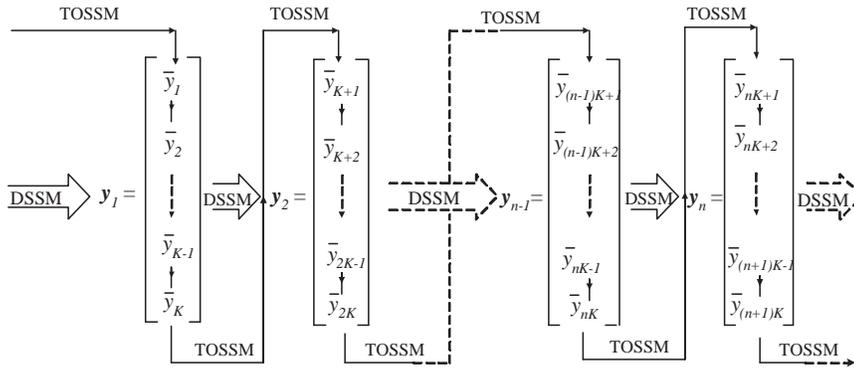


Figura 4.1: Figura ilustrativa del MEETO.

4.3. Detección Ciega Bayesiana Óptima

Desde el punto de vista Bayesiano, la decisión óptima sobre \mathbf{b}_N puede obtenerse usando el criterio MPM (Marginalized Posterior Mode) que se expresa del siguiente modo:

$$(\hat{b}_{N,k})_{MPM} = \text{sgn} \left(\sum_{\mathbf{b}_N \in \{-1,1\}^K} b_{n,k} p(\mathbf{b}_N | \bar{y}_{1:NK}) \right) \quad (4.13)$$

donde $p(\mathbf{b}_N | \bar{y}_{1:NK})$ es la distribución posterior que es esencial para calcular (4.13) y el subíndice $1:NK$ significa el conjunto variables que van desde 1 a NK , e.g., $\bar{y}_{1:NK} = \{\bar{y}_1, \dots, \bar{y}_{NK}\}$. La distribución posterior $p(\mathbf{b}_N | \bar{y}_{1:NK})$ es independiente de $\mathbf{b}_{1:(N-1)}$, i.e, los bits transmitidos antes del tiempo n . Además, la marginalización en (4.13) indica que $(\hat{b}_{N,k})_{MPM}$ es también independiente de los bits transmitidos por otros usuarios en el instante n . Por tanto, la solución MPM es inmune a errores de decisión sobre $\mathbf{b}_{1:(N-1)}$ y los bits de otros usuarios transmitidos en n .

Para obtener $p(\mathbf{b}_n|\bar{y}_{1:NK})$, es necesario llevar a cabo una marginalización en $p(\mathbf{b}_{1:N}|\bar{y}_{1:NK})$ en $\mathbf{b}_{1:(N-1)}$ del siguiente modo

$$\begin{aligned} p(\mathbf{b}_N|\bar{y}_{1:NK}) &= \sum_{\mathbf{b}_{1:N-1}} p(\mathbf{b}_{1:N}|\bar{y}_{1:NK}) \\ &= \frac{\sum_{\mathbf{b}_{1:N-1}} p(\mathbf{b}_{1:N}, \bar{y}_{1:NK})}{\sum_{\mathbf{b}_{1:N}} p(\mathbf{b}_{1:N}, \bar{y}_{1:NK})} \end{aligned} \quad (4.14)$$

Considerando el MEETO se tiene

$$\begin{aligned} p(\mathbf{b}_{1:N}, \bar{y}_{1:NK}) &= p(\bar{y}_{NK}|\mathbf{b}_{1:N}, \bar{y}_{1:NK-1})p(\mathbf{b}_{1:N}, \bar{y}_{1:NK-1}) \\ &= p(\bar{y}_{NK}|\mathbf{b}_{1:N}, \bar{y}_{1:NK-1})p(b_{N,K}|\mathbf{b}_{N,1:K-1}, \mathbf{b}_{1:N-1}, \bar{y}_{1:NK-1}) \\ &\quad p(\mathbf{b}_{N,1:K-1}, \mathbf{b}_{1:N-1}, \bar{y}_{1:NK-1}) \\ &= p(\bar{y}_{NK}|\mathbf{b}_{1:N}, \bar{y}_{1:NK-1})p(b_{N,K})p(\mathbf{b}_{N,1:K-1}, \mathbf{b}_{1:N-1}, \bar{y}_{1:NK-1}) \\ &= p(\bar{y}_{NK}|\mathbf{b}_{1:N}, \bar{y}_{1:NK-1})p(\mathbf{b}_{N,1:K-1}, \mathbf{b}_{1:N-1}, \bar{y}_{1:NK-1}) \end{aligned} \quad (4.15)$$

donde la última ecuación se obtiene suponiendo que $p(b_{N,K} = 1) = 0,5$. La ecuación (4.15) indica un cálculo recursivo de $p(\mathbf{b}_{1:N}, \bar{y}_{1:NK})$ desde $l = 1$ a NK multiplicando la probabilidad marginal $p(\bar{y}_l|\mathbf{b}_{n,1:k}, \mathbf{b}_{1:n-1}, \bar{y}_{1:l})$ en cada recursión. Estas probabilidades $p(\bar{y}_l|\mathbf{b}_{n,1:k}, \mathbf{b}_{1:n-1}, \bar{y}_{1:l})$ para $l = 1, \dots, NK$ se obtienen marginalizando el vector de estado del canal \mathbf{h}_l de $p(\bar{y}_l, \mathbf{h}_l|\mathbf{b}_{n,1:k}, \mathbf{b}_{1:n-1}, \bar{y}_{1:l})$, y en el apéndice D se muestra que

$$p(\bar{y}_l|\mathbf{b}_{n,1:k}, \mathbf{b}_{1:n-1}, \bar{y}_{1:l}) = \mathcal{N}(m_l, c_l) \quad (4.16)$$

y la media m_l y la varianza c_l pueden calcularse secuencialmente mediante el filtrado de Kalman. Esto es equivalente a afirmar que $p(\bar{y}_l|\mathbf{b}_{n,1:k}, \mathbf{b}_{1:n-1})$ puede calcularse usando un filtro de Kalman. Si ahora se vuelve sobre la ecuación (4.14), puede apreciarse que para calcular $p(\mathbf{b}_N|\bar{y}_{1:NK})$, ha de evaluarse $p(\bar{y}_l|\mathbf{b}_{n,1:k}, \mathbf{b}_{1:n-1})$ para 2^{NK} combinaciones de $\mathbf{b}_{1:N}$, o lo que es lo mismo, se necesitan 2^{NK} filtros de Kalman, cada uno correspondiente a una posible combinación. Como resultado, se necesitan un total de 2^{NK} filtros de Kalman para una solución MPM. La expansión del número de filtro de Kalman con l presenta una estructura en árbol que aparece ilustrada en la Figura 4.2. La solución MPM tiene por tanto una complejidad que crece exponencialmente con el tiempo n y con el número de usuarios K . Es por esto que la solución MPM es aparentemente demasiado compleja desde el punto de vista computacional como para usarla en aplicaciones reales. Por tanto ha de recurrirse a soluciones subóptimas con complejidad aceptable. Una posible opción es usar filtrado de partículas.

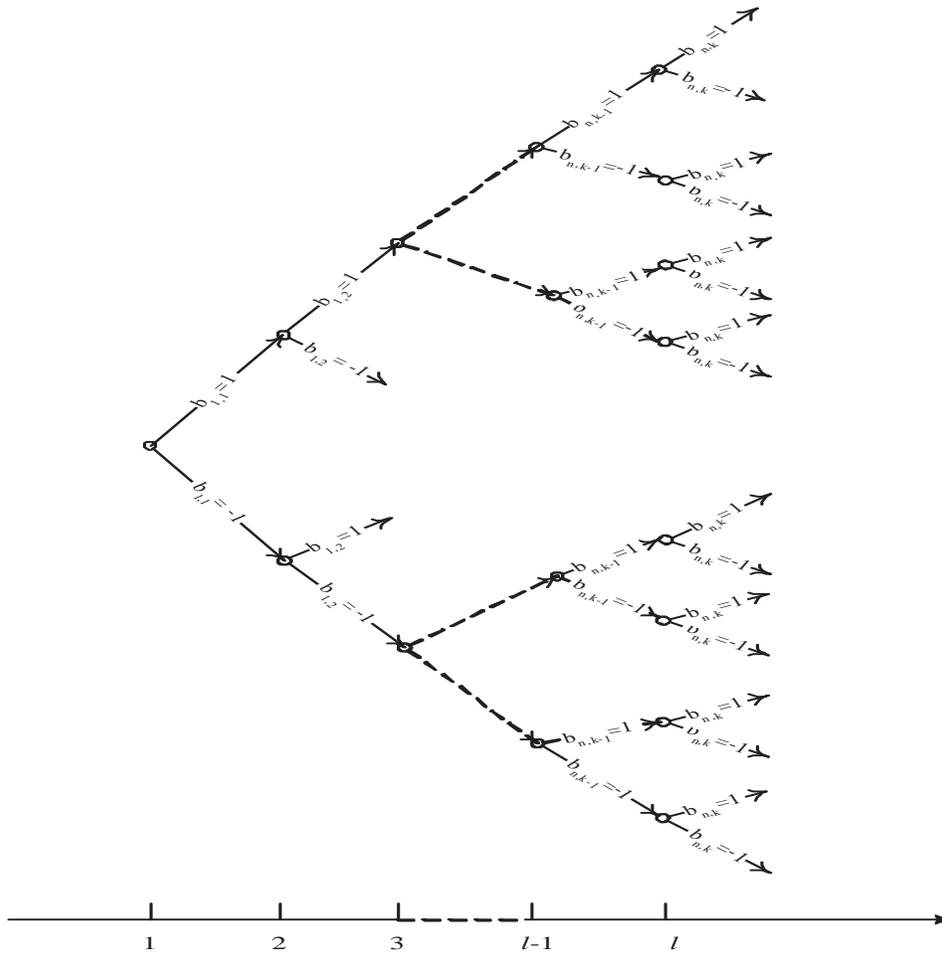


Figura 4.2: Estructura en árbol de la solución óptima. Cada camino en el árbol representa un filtro de Kalman.

4.4. Una aproximación dirigida por decisión a la DMU ciega

Una aproximación dirigida por decisión a la DMU ciega se propuso en [90] basada en MEED (4.6). A continuación se describe la correspondiente aproximación dirigida por decisión para el MEETO (4.12). Una característica distintiva de la aproximación dirigida por decisión para el MEETO es que la decisión se lleva a cabo para un único bit de un único usuario para cada l . Específicamente, si $\hat{b}_{n,k-1}$ y $\hat{\mathbf{h}}_{l-1}$ representan las decisiones sobre $b_{n,k-1}$ y \mathbf{h}_{l-1} en $l-1$, entonces la aproximación dirigida por decisión en l puede resumirse de la siguiente forma:

Algoritmo Dirigido por Decisión para MEETO

- *Predicción:*

$$\begin{aligned}\tilde{\mathbf{h}}_l &= \begin{cases} Q\hat{\mathbf{h}}_{l-1} + Gu_l & \text{si } k = 1 \\ \hat{\mathbf{h}}_{l-1} & \text{si } k \neq 1 \end{cases} \\ \hat{\Sigma}_l &= \begin{cases} Q\hat{\Xi}_{l-1}Q^T + \sigma^2GG^T & \text{si } k = 1 \\ \hat{\Xi}_{l-1} & \text{si } k \neq 1 \end{cases}\end{aligned}\quad (4.17)$$

- *Detección:*

$$\hat{b}_{n,k} = \text{sgn}(z_{n,k}) \text{ donde } z_{n,k} = (\bar{y}_l - \sum_{j=1}^{k-1} f_{k,j}a_{i,j}b_{i,j})a_{i,k}^* \text{ con } a_{i,k} = \boldsymbol{\rho}_k \tilde{\mathbf{h}}_l.$$

- *Actualización:*

$$\hat{\mathbf{K}}_l = \hat{\Sigma}_l \hat{\mathbf{C}}_l^H / \hat{c}_l \text{ con } \hat{c}_l = \hat{\mathbf{C}}_l \hat{\Sigma}_l \hat{\mathbf{C}}_l^H + \sigma^2$$

$$\hat{\mathbf{h}}_l = \tilde{\mathbf{h}}_l + \hat{\mathbf{K}}_l (\bar{y}_l - \hat{\mathbf{C}}_l \tilde{\mathbf{h}}_l)$$

$$\hat{\Xi}_l = (\mathbf{I} - \hat{\mathbf{K}}_l \hat{\mathbf{C}}_l) \hat{\Sigma}_l$$

$$\text{donde } \hat{\mathbf{C}}_l = \mathbf{f}_k^T \hat{\mathbf{B}}_l \mathbf{P} \text{ y } \hat{\mathbf{B}}_l = \text{diag}\{\hat{b}_{n,1}, \dots, \hat{b}_{n,k}, 0 \dots, 0\}$$

Claramente, el algoritmo dirigido por decisión anterior es equivalente a un filtro de Kalman, y por tanto, es mucho más simple que la solución MPM. Sin embargo, el bit de usuario se determina teniendo en cuenta la predicción de los estados del canal y las decisiones sobre los bits de los usuarios anteriores, y por tanto, no es óptima. Comparado con el algoritmo basado en el MEED (4.6), en el tiempo k con k desde 1 a K , el algoritmo anterior realiza decisión sobre un usuario en cada instante de tiempo y actualiza el vector de estado del canal \mathbf{h}_l cuando la decisión se alcanza. El vector \mathbf{h}_l actualizado influirá entonces en la decisión sobre $b_{n,k+1}$. Por tanto, tanto para lo bueno como para lo malo, las decisiones en las primeras etapas (pequeños k) tendrían mayor impacto en las decisiones en los estados posteriores (valores de k mayores) que las hechas usando el algoritmo basado en el MEED.

Si hay error en la detección en las primeras etapas de la misma, estos errores se propagarán a los estados posteriores. Es, por tanto, bueno clasificar los usuarios de acuerdo con el SNR estimado. El rendimiento del algoritmo dirigido por decisión está, sin embargo, limitado en último extremo por la propagación del error.

4.5. Filtrado de Partículas aplicado a la DMU ciega

El Filtrado de Partículas pertenece a la familia de los métodos de muestreo de Monte Carlo que tienen como objetivo generar muestras de la distribución posterior deseada. Sin embargo, el filtrado de partículas se distingue por emplear un esquema de Muestreo de Importancia (Importance Sampling), y en particular, está diseñado para sistemas no lineales y no Gaussianos descritos a través del modelo de espacio de estados.

Para el problema en estudio, cuando \mathbf{y}_N , o equivalentemente $\bar{\mathbf{y}}_N$, se observa para el tiempo N , el objetivo del Filtrado de Partículas es obtener, digamos, J muestras aleatorias a las que se les asocia un determinado peso $\{\mathbf{b}_{1:N}^{(j)}, w_{NK}^{(j)}\}$ todo ello usando $p(\mathbf{b}_{1:N}|\bar{\mathbf{y}}_{1:nK})$ donde $w_{NK}^{(j)}$ es el peso de la j -ésima muestra $\mathbf{b}_{1:N}^{(j)}$. Con las muestras, $p(\mathbf{b}_N|\bar{\mathbf{y}}_{1:NK})$ puede aproximarse de la siguiente forma:

$$p(\mathbf{b}_{1:N}|\bar{\mathbf{y}}_{1:nK}) \approx \sum_{j=1}^J w_{NK}^{(j)} \prod_{l=1}^{NK} \delta(b_{n,k} - b_{n,k}^{(j)}) \quad (4.18)$$

donde $\delta(\cdot)$ es la función delta de Dirac, y por tanto la solución MPM para \mathbf{b} usando una sumatoria y teniendo en cuenta los pesos es

$$(\hat{b}_{N,k})_{MPM} \approx \text{sgn}\left(\sum_{j=1}^J w_{NK}^{(j)} b_{N,k}^{(j)}\right) \quad (4.19)$$

para $k = 1, \dots, K$. Según la ley de los grandes números, la aproximación convergerá a la solución MPM verdadera cuando el número de muestras J aumente. Si las muestras se toman directamente de la distribución todas tendrán el mismo peso. Sin embargo, el muestreo directo de $p(\mathbf{b}_{1:N}|\bar{\mathbf{y}}_{1:NK})$ no es posible porque todas las posibles combinaciones de $\mathbf{b}_{1:N}$ tendrían que evaluarse en $p(\mathbf{b}_{1:N}|\bar{\mathbf{y}}_{1:NK})$, y esto requiere de nuevo 2^{NK} filtros de Kalman. Para solucionar este problema, se lleva a cabo un proceso de Importance Sampling donde las muestras se toman de una distribución que se llama Importance Function $\pi(\mathbf{b}_{1:KN}|\bar{\mathbf{y}}_{1:KN})$ y los pesos se calculan según

$$w_{KN}^{(j)} = \frac{p(\mathbf{b}_{1:KN}^{(j)}|\bar{\mathbf{y}}_{1:KN})}{\pi(\mathbf{b}_{1:KN}^{(j)}|\bar{\mathbf{y}}_{1:KN})}, \quad \forall j. \quad (4.20)$$

$\pi(\mathbf{b}_{1:KN}|\bar{\mathbf{y}}_{1:KN})$ es una distribución de gran dimensión y es complicado muestrear las variables y calcular los pesos a la vez. Afortunadamente, el MEETO permite una factorización Markoviana de la distribución posterior de la siguiente

forma

$$\begin{aligned}
p(\mathbf{b}_{1:N}, \bar{y}_{1:NK}) &\propto p(\bar{y}_{NK} | \mathbf{b}_{1:N}, \bar{y}_{1:NK-1}) p(b_{N,K}) \\
&\quad \times p(\mathbf{b}_{N,1:K-1}, \mathbf{b}_{1:N-1} | \bar{y}_{1:NK-1}) \\
&= p(\bar{y}_{NK} | \mathbf{b}_{1:N}, \bar{y}_{1:NK-1}) \\
&\quad \times p(\mathbf{b}_{N,1:K-1}, \mathbf{b}_{1:N-1} | \bar{y}_{1:NK-1}).
\end{aligned} \tag{4.21}$$

Entonces, si elegimos la función de importancia del siguiente modo

$$\begin{aligned}
\pi(\mathbf{b}_{1:N} | \bar{y}_{1:NK}) &= p(b_{N,k} | \mathbf{b}_{N,1:K-1}, \mathbf{b}_{1:N-1}, \bar{y}_{1:NK}) \\
&\quad \times \pi(\mathbf{b}_{N,1:K-1}, \mathbf{b}_{1:N-1} | \bar{y}_{1:NK-1})
\end{aligned} \tag{4.22}$$

el peso puede calcularse como

$$\begin{aligned}
w_{KN}^{(j)} &= \frac{p(\bar{y}_{NK} | \mathbf{b}_{1:N}^{(j)}, \bar{y}_{1:NK-1}) p(b_{N,K}^{(j)})}{p(b_{N,K}^{(j)} | \mathbf{b}_{N,1:K-1}^{(j)}, \mathbf{b}_{1:N-1}^{(j)}, \bar{y}_{1:NK})} \\
&\quad \times \frac{p(\mathbf{b}_{N,1:K-1}^{(j)}, \mathbf{b}_{1:N-1}^{(j)} | \bar{y}_{1:NK-1})}{\pi(\mathbf{b}_{N,1:K-1}^{(j)}, \mathbf{b}_{1:N-1}^{(j)} | \bar{y}_{1:NK-1})} \\
&= \frac{p(\bar{y}_{NK} | \mathbf{b}_{1:N}^{(j)}, \bar{y}_{1:NK-1}) p(b_{N,K}^{(j)})}{p(b_{N,K}^{(j)} | \mathbf{b}_{N,1:K-1}^{(j)}, \mathbf{b}_{1:N-1}^{(j)}, \bar{y}_{1:NK})} w_{KN-1}^{(j)} \\
&\propto p(\bar{y}_{NK} | \mathbf{b}_{N,1:K-1}^{(j)}, \mathbf{b}_{1:N-1}^{(j)}, \bar{y}_{1:NK-1}) w_{KN-1}^{(j)} \\
&= \sum_{b_{N,K}} p(\bar{y}_{NK} | \mathbf{b}_{1:N}^{(j)}, \bar{y}_{1:NK-1}) w_{KN-1}^{(j)} \\
&= \mu_{KN-1}^{(j)} w_{KN-1}^{(j)}
\end{aligned} \tag{4.23}$$

donde $\mu_{KN-1}^{(j)}$ es el peso incremental.

Examinando (4.22) y (4.23), se encuentra que dados $w_{KN-1}^{(j)}$ y $p(\mathbf{b}_{N,1:K-1}, \mathbf{b}_{1:N-1} | \bar{y}_{1:NK-1})$, la Función de Importancia (4.22) y los pesos (4.23) se conocen exactamente si $p(\bar{y}_{NK} | \mathbf{b}_{1:N}^{(j)}, \bar{y}_{1:NK-1})$ puede obtenerse. En realidad, $p(\bar{y}_{NK} | \mathbf{b}_{1:N}, \bar{y}_{1:NK-1})$ puede calcularse usando un filtro de Kalman [44] según:

$$\begin{aligned}
\lambda_{NK}(i) &= p(\bar{y}_{NK} | b_{N,K} = 2 * i - 3, \mathbf{b}_{N,1:K}^{(j)}, \mathbf{b}_{1:N}^{(j)}, \bar{y}_{1:NK-1}) \\
&= \mathcal{N}_c(m_{NK}^{(j)}(i), c_{NK}^{(j)}(i))
\end{aligned} \tag{4.24}$$

para $i = 1, 2$ donde $m_l^{(j)}(i)$ y $c_l^{(j)}(i)$ se calculan de la misma forma que se muestra en el Apéndice D pero para cierto grupo de $\mathbf{b}_{1:NK}$ dado en (4.24).

Por tanto, se pueden obtener tanto las muestras como los pesos usando un algoritmo recursivo. El procedimiento sería el siguiente: se asume que en $l-1$, se han obtenido usando una recursión previa las trayectorias (muestras) $\{\mathbf{b}_{0:l-1}^{(j)}\}_{j=1}^J$ adecuadamente pesadas con pesos $\{w_{l-1}^{(j)}\}_{j=1}^J$. Haciendo uso de las observaciones recientes \bar{y}_l , se actualizan las trayectorias y pesos en la recursión l -ésima de la siguiente forma:

Algoritmo: Detector de Filtrado de Partículas(DFP)

- Para j desde $j = 1$ hasta J ,

1. **Predicción:**

$$\text{Calcular } \boldsymbol{\mu}_l^{(j)} = \begin{cases} \mathbf{Q}\boldsymbol{\eta}_{l-1}^{(j)} & \text{si } k = 1 \\ \boldsymbol{\eta}_{l-1}^{(j)} & \text{si } k \neq 1 \end{cases} \quad \text{y}$$

$$\boldsymbol{\Sigma}_l^{(j)} = \begin{cases} \mathbf{Q}\boldsymbol{\Xi}_{l-1}^{(j)}\mathbf{Q}^\top + \sigma^2\mathbf{G}\mathbf{G}^\top & \text{si } k = 1 \\ \boldsymbol{\Xi}_{l-1}^{(j)} & \text{si } k \neq 1 \end{cases}$$

2. **Muestreo:**

a) Para $i = 1$ y -1 , calcular

- $m_l^{(j)}(i) = \mathbf{c}_l^{(j)}(i)\boldsymbol{\mu}_l^{(j)}$ y $c_l^{(j)}(i) = \mathbf{c}_l(i)\boldsymbol{\Sigma}_l^{(j)}\mathbf{c}_l^{(j)\text{H}}(i) + \sigma^2$, donde $\mathbf{c}_l^{(j)} = \mathbf{f}_k^\top \mathbf{B}_l^{(j)}(i)\mathbf{P}$,
 $\mathbf{B}_l^{(j)}(i) = \text{diag}\{b_{n,1}^{(j)}, \dots, b_{n,k-1}^{(j)}, i, 0, \dots, 0\}$
- $\lambda_l^{(j)}(i) = \mathcal{N}_c(m_l^{(j)}(i), c_l^{(j)}(i))$.

b) Muestrear $m \in \{-1, 1\}$ con probabilidad proporcional $\lambda_l^{(j)}(i) \forall i$;

c) Establecer $b_l^{(j)} = m$;

d) Calcular $\mu_l^{(j)} = \sum_{i \in \{-1, 1\}} \lambda_l^{(j)}(i)$ y el peso sin normalizar $\bar{w}_l^{(j)} = \mu_l^{(j)} w_{l-1}^{(j)}$.

3. **Actualización:** Calcular

- $\mathbf{K}_l^{(j)} = \boldsymbol{\Sigma}_l^{(j)}\mathbf{c}_l^{(j)}(m)\mathbf{c}_l^{(j)\text{H}}(m)$;
- $\boldsymbol{\eta}_l^{(j)} = \boldsymbol{\mu}_l^{(j)} + \mathbf{K}_l(\bar{y}_l - \mathbf{c}_l^{(j)}(m)\boldsymbol{\mu}_l^{(j)})$;

$$\bullet \hat{\Xi}_l^{(j)} = (\mathbf{I} - \mathbf{K}_l^{(j)} \mathbf{c}_l^{(j)}(m)) \Sigma_l^{(j)}.$$

- Crear las nuevas trayectorias $\{b_l^{(j)}, \mathbf{b}_{0:l-1}^{(j)}\}_{j=1}^J$;
- Normalizar los peso según $w_l^{(j)} = \bar{w}_l^{(j)} / \sum_{j=1}^J \bar{w}_l^{(j)}$.

Este proceso de obtención de partículas de forma recursiva se llama Filtrado de Partículas. Tras cada recursión, los vectores de media $\boldsymbol{\eta}_l^{(j)}$ y covarianza $\hat{\Xi}_l^{(j)}$ se pasan a la siguiente recursión. De (4.19), podemos ver también que se necesita calcular todos los elementos de $\{\mathbf{b}_N\}_{MPM}$, $w_{NK}^{(j)}$. Por tanto, la decisión sólo puede realizarse tras la recursión $l = KN$ y las partículas $b_{N,k}$ con $k = 1, 2, \dots, K - 1$ deben almacenarse.

En el proceso anterior de Filtrado de Partículas, la función de importancia que se ha usado se conoce como óptima ya que minimiza la varianza de los pesos. El Filtrado de Partículas anterior sufre de un empobrecimiento de las partículas, esto es, después de algunas iteraciones, algunos pesos son despreciables y dejan de contribuir. Para evitarlo, se inserta un paso de muestreo residual [19] tras cada recursión. En particular, durante el paso de Resampling en la recursión l -ésima, las partículas para $\mathbf{b}_{n,1:k}^{(j)}$, los vectores de media y las matrices de covarianza deben tratarse como un conjunto en el proceso de Resampling.

4.6. Detector M -Estocástico para DMU ciega

Recientemente, ha sido propuesto [47] un algoritmo de filtrado de partículas muy eficiente llamado algoritmo M -Estocástico (AME) para utilizar en problemas donde las magnitudes que no se conocen son discretas. El AME tiene un comportamiento parecido al de los algoritmos genéricos de Filtrado de Partículas con funciones Delta discretas como funciones de importancia. Además, cada trayectoria da lugar a dos muestras (-1 y 1) para el caso binario en lugar de la única muestra que se genera en el algoritmo genérico de FP.

Un punto importante del AME es que no hay dos trayectorias iguales, lo que ocurre raramente en los algoritmos generales de FP. Como resultado, el AME da lugar a una mayor diversidad en las muestras con menos trayectorias que el algoritmo de FP genérico. Sin embargo, al doblarse el número de trayectorias después de cada muestro, es necesario introducir un paso de selección para evitar el aumento exponencial del número de trayectorias. En

este caso, se propone usar el algoritmo de Resampling óptimo [33] ya que es un algoritmo de muestro sin sustitución que no produce réplicas de la misma trayectoria que es una característica a tener en cuenta en el AME. A continuación se detalla el AME para la recursión l -ésima:

Algoritmo: Detector M-Estocástico (DME)

■ **Expansión de las Trayectorias**

Desde $j = 1$ hasta J

- Realizar el *Paso Predictivo* del Algoritmo DFP;
 - Realizar el paso 2.a en el Algoritmo DFP;
 - Establecer $b_l^{(2j-1)} = 1$ y calcular el peso utilizando $\bar{w}_l^{(2j-1)} = \lambda_l^{(j)}(1)w_{l-1}^{(j)}$;
 - Establecer $b_l^{(2j)} = -1$ y calcular el peso utilizando $\bar{w}_l^{(2j-1)} = \lambda_l^{(j)}(-1)w_{l-1}^{(j)}$;
 - Crear $2J$ nuevas trayectorias estableciendo $\mathbf{b}_l^{(2j-1)} = \{b_l^{(2j-1)}, \mathbf{b}_{0:l}^{(j)}\}$ y $\mathbf{b}_l^{(2j)} = \{b_l^{(2j)}, \mathbf{b}_{0:l}^{(j)}\}$.
- Normalizar los pesos $\bar{w}_k^{(j)}$ para obtener $w_k^{(j)}$;
- **Selección de Trayectorias:** Seleccionar J trayectorias de las $2M$ trayectorias usando el algoritmo de muestreo óptimo.
- **Paso de Actualización:** Desde $j = 1$ a J ;
Realizar el *paso de Actualización* del Algoritmo DFP.
-

La estructura del AME se parece a la del algoritmo M . Sin embargo, ya que el AME es un algoritmo de tipo FP, éste puede proporcionar información estadística sobre las variables que no se conocen y por tanto, puede ser aplicado de forma iterativa a Detección Multiusuario de sistemas codificados.

4.7. Detector Genie-Aided

Este detector se presenta como límite inferior para la comparación de los rendimientos de los demás. Esto se debe a que el procedimiento a seguir no es completamente ciego. En primer lugar se genera un vector de observaciones (señal piloto) y con él se realiza una estimación del canal a través de un Filtrado de Kalman. A continuación se realiza el proceso de detección según un algoritmo de FP donde el vector de estado del canal se supone conocido y viene dado por la estimación que se ha hecho con la señal piloto.

4.8. Comparación entre los Detectores

Si comparamos las soluciones dadas por el DFP y el DME con la que se obtienen haciendo uso del algoritmo dirigido por decisión, vemos que los procesos a lo largo de cada trayectoria son prácticamente idénticos a los realizados para el algoritmo dirigido por decisión excepto por el paso de muestreo que se usó en lugar del de detección, y se parecen al uso de un filtro de Kalman que se corresponde con un camino en el árbol de la Figura 4.2.

Hay dos caminos saliendo de cada nodo del árbol, y en la selección del camino, el algoritmo dirigido por decisión usa una aproximación determinista mientras que los DFP y DME adaptan una medida blanda que se basa en la probabilidad. Y lo que es más, cada trayectoria está además asociada con un peso que indica la importancia de la trayectoria en la decisión final. A pesar de que las trayectorias con peso pequeño parecen no contribuir demasiado en las decisiones que se hacen en el estado actual, pueden florecer en recursiones posteriores y llevar asociados pesos significativos para la decisión. Las medidas blandas aparentemente pueden evitar que los errores en las decisiones actuales puedan afectar en gran manera las decisiones futuras, una ventaja clave sobre la aproximación dirigida por decisión.

Si se comparan los DFP y los DME con las soluciones óptimas MPM, puede verse que los DME tienen un punto importante a su favor y está relacionado con el hecho de que sólo mantiene J trayectorias o equivalentemente J filtros de Kalman todo el tiempo mientras que el número de filtros de Kalman requerido por la MPM crece exponencialmente en el tiempo. Además, el DFP y el DME alcanzan una aproximación efectiva y eficiente a la distribución posterior verdadera y por tanto, presentan un rendimiento en decisión próximo al óptimo. Como ambos detectores producen resultados blandos (probabilísticos), están preparados para aplicarse a turbo MUD. Esta es otra importante ventaja a favor de estos algoritmos.

4.9. Escenarios con Near-Far

Uno de los problemas a los que se le ha dedicado gran cantidad de investigación dentro de los sistemas CDMA es el de “near-far”. Se trata de una situación en la que los distintos usuarios tienen asociada distinta potencia. De entre las soluciones propuestas para este problema, la que se ha impuesto con más fuerza es el control de potencia. Sin embargo, esta técnica necesita de condiciones especiales para que su aplicación sea efectiva [28]. Además, su aplicación es difícil para el caso de canales tipo Rayleigh con desvanecimiento rápido. Para que el detector multiusuario convencional (banco de filtros adaptativos) alcance un rendimiento óptimo es necesario que las variaciones en el sistema de control de potencia sean pequeñas (menores de 1 dB) [55]. Aunque el problema de near-far pudiera resolverse con un estricto control de la potencia, aún habría que intentar controlar otro de los grandes problemas en CDMA: la IAM (Interferencia de Acceso Múltiple). Motivados por estos dos grandes problemas, se han propuesto en la literatura un gran número de algoritmos de detección multiusuario [83], [81], [27], [71].

Los algoritmos anteriormente descritos han sido aplicados también en este tipo de escenarios. Los resultados obtenidos de las simulaciones así como las conclusiones a las que se pueden llegar aparecen en los siguientes apartados.

4.10. Simulaciones

4.10.1. Usuarios de Igual potencia

En esta sección se muestra el comportamiento del BER en función del SNR para distintos detectores en el siguiente escenario:

- Las señales enviadas por la fuente se modulan BPSK.
- La frecuencia de la señal portadora es de 2GHz con lo que nos encontramos trabajando dentro del rango estipulado para los móviles de Tercera Generación.
- La velocidad de los móviles está en torno a los 100 Km/h y a los 180 Km/h.
- Se consideran distintos casos en los que 10 y 15 usuarios acceden al medio a la misma vez.
- La potencia de todos los usuarios es la misma por lo que no se contemplan efectos near-far.

- El fenómeno de desvanescencia introducido por el canal se modela haciendo uso de un modelo ARMA de segundo orden y se normaliza para que tenga potencia unidad de modo que la expresión utilizada para el SNR es $10 \log(1/\sigma^2)$.
- Se consideran dos tipos transmisiones, una en la que la frecuencia Doppler es 300 Hz y otra en la que su valor es 500 Hz.

Haciendo uso de un Modelo de Espacio de Estados de Tiempo Observación se han estudiado y comparado el rendimiento de los siguientes detectores:

- Detector de Filtrado de Partículas (DFP) para el cual se consideran 151 partículas.
- Detectores basados en el Algoritmo M -Estocástico (DME) en los cuales se consideran 32 y 4 trayectorias.
- Detector Dirigido por Decisión (DDD)
- Detector Genie Aided que se incluye como límite inferior al no tratarse de un detector ciego.

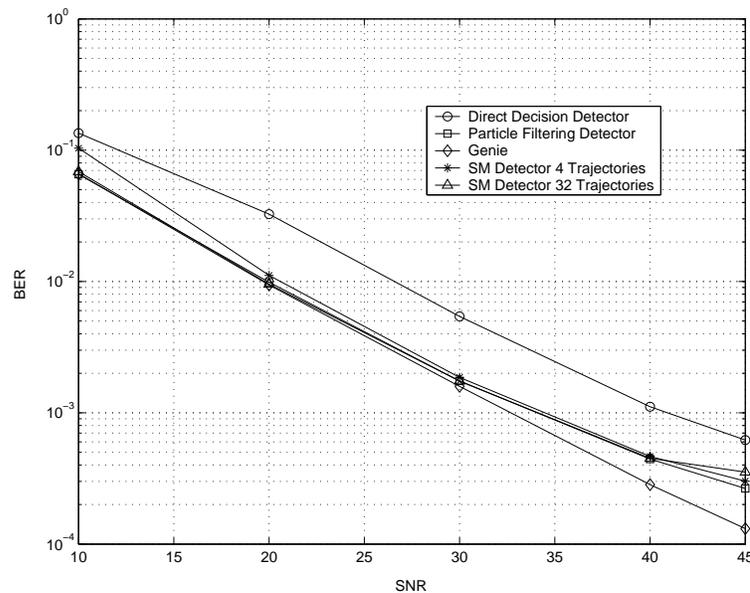


Figura 4.3: BERs vs. SNR para varios detectores. Igual Potencia. $\Omega = 0,03$, $f_d = 300Hz$ y 15 usuarios

En la figura 4.3 se representan los resultados de las simulaciones para el caso en que $\Omega_d = 0,03$. Los resultados más importantes son los siguientes:

- Los DFP y DME demuestran ser mejores que los DDD y un ejemplo está en el hecho de que logran una ganancia de 7 dB sobre los DDD para un valor de BER 10^{-3} .
- Se obtienen valores para el BER del mismo orden de magnitud para el DFP, DME con 32 trayectorias y el Genie Aided para SNR bajo (menos de 30 dB).
- Los valores de BER obtenidos para los DME con 4 y 32 trayectorias son muy similares entre sí especialmente para SNR mayor de 20 dB y comparables a los obtenidos con DFP.
- Por todo lo anterior, puede afirmarse que el DME es más favorable ya que requiere aproximadamente sólo un 1/35 de la complejidad del DFP.

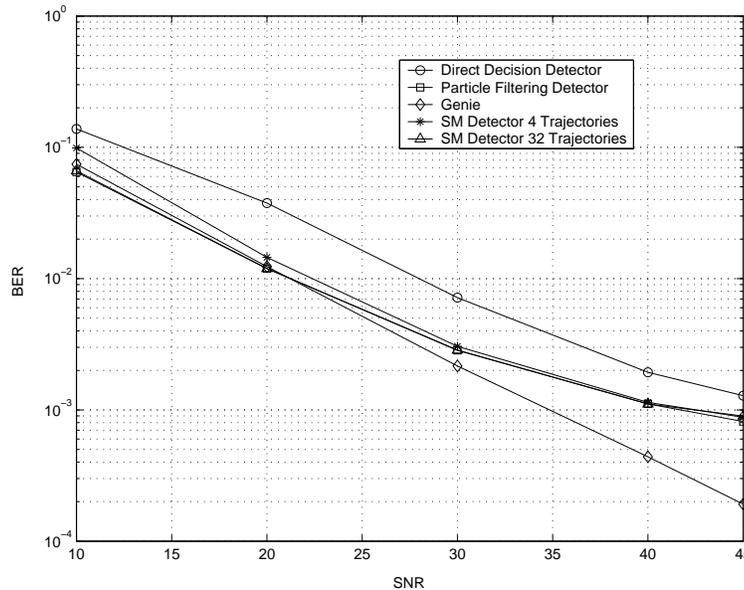


Figura 4.4: BERs vs. SNR para varios detectores. Igual Potencia. $\Omega = 0,05$, $f_d = 500Hz$ y 15 usuarios

En la Figura 4.4 se representan los resultados de las simulaciones para el caso en que $\Omega_d = 0,05$. Los resultados más importantes son los siguientes:

- En cuanto a la comparación entre los rendimientos de los distintos detectores, se pueden hacer las mismas observaciones que se han hecho para las simulaciones correspondientes a $\Omega_d = 0,03$. Siendo de nuevo el DME con 4 trayectorias el más favorable.

- El comportamiento global de todos los detectores en este caso es peor si lo comparamos con el rendimiento cuando $\Omega_d = 0,03$ lo cual era de esperar ya que los efectos negativos del canal toman más relevancia porque el desvanecimiento es más rápido.

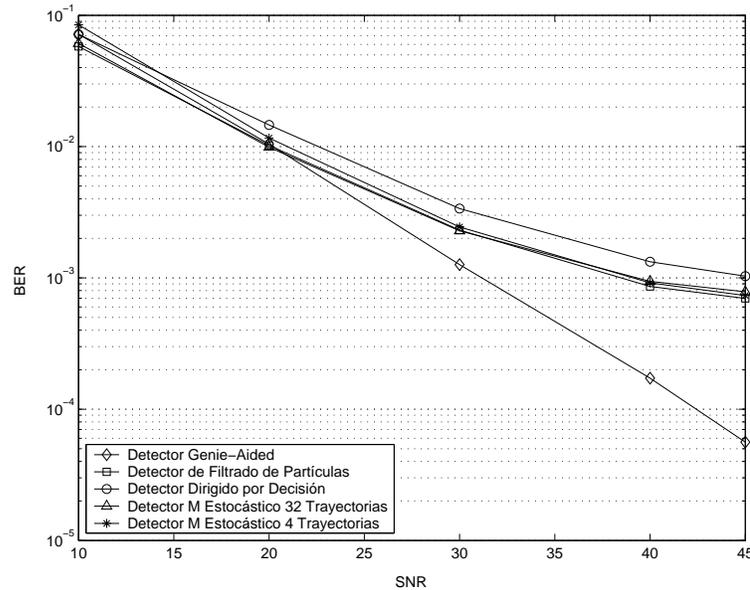


Figura 4.5: Comportamiento para varios detectores del BER vs. SNR. $\Omega = 0,03$ y 10 usuarios

En la figura 4.5, se representa el BER frente al SNR para los diferentes algoritmos en el caso en que $\Omega_d = 0,03$ y el número de usuarios es 10.

En la figura 4.6, mostramos el BER frente a la SNR para una frecuencia Doppler $\Omega_d = 0,05$. En este caso se pueden hacer las mismas observaciones que para el caso anterior aunque el comportamiento global de todos los detectores sea peor, lo cual es razonable teniendo en cuenta que el desvanecimiento del canal es más rápido.

4.10.2. Usuarios con distinta potencia

En esta sección se muestra el comportamiento del BER en función del SNR del primer usuario para distintos detectores en el siguiente escenario:

- Las señales enviadas por la fuente se modulan BPSK.
- La frecuencia de la señal portadora es de 2GHz con lo que nos encontramos trabajando dentro del rango estipulado para los móviles de Tercera Generación.

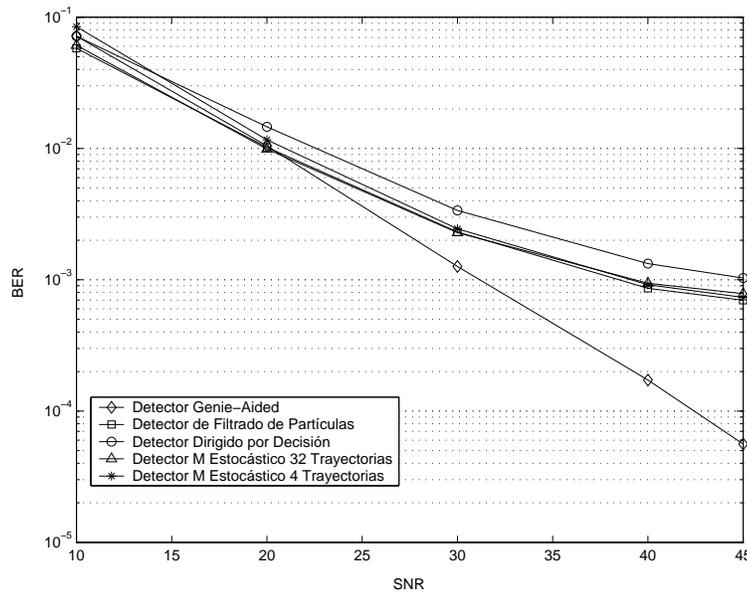


Figura 4.6: Comportamiento para varios detectores del BER vs. SNR. $\Omega = 0,05$ y 10 usuarios

- La velocidad de los móviles está en torno a los 100 Km/h.
- Se consideran 15 usuarios accediendo al medio a la misma vez.
- La potencia asociada a los distintos usuarios es distinta con lo cual se contemplan efectos near-far.
- El fenómeno de desvanescencia introducido por el canal se modela haciendo uso de un modelo ARMA de segundo orden y se normaliza de modo que el primer usuario potencia unidad de modo que la expresión utilizada para el SNR es $10 \log(1/\sigma^2)$.
- Se consideran dos tipos transmisiones, una en la que la frecuencia Doppler es 300 Hz y otra en la que su valor es 500 Hz.

Haciendo uso de un Modelo de Espacio de Estados de Tiempo Observación se han estudiado y comparado el rendimiento de los siguientes detectores:

- Detector de Filtrado de Partículas (DFP) para el cual se consideran 151 partículas.
- Detectores basados en el Algoritmo *M*-Estocástico (DME) en los cuales se consideran 32 y 4 trayectorias.

- Detector Dirigido por Decisión (DDD)
- Detector Genie Aided que se incluye como límite inferior al no tratarse de un detector ciego.

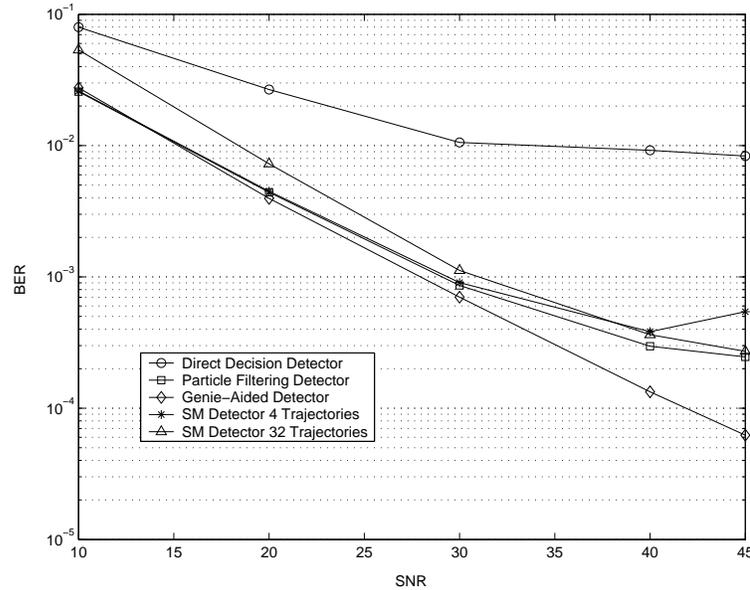


Figura 4.7: BERs vs. SNR para varios detectores. Usuarios con distinta potencia. $\Omega = 0,03$, $f_d = 300Hz$, 15 usuarios

En la figura 4.7 se representan los resultados de las simulaciones para el caso en que $\Omega_d = 0,03$. Los resultados más importantes son los siguientes:

- Los DFP y DME demuestran ser mejores que los DDD siendo las diferencias entre sus rendimientos mayores que las existentes para el escenario sin near-far. Un ejemplo está en el hecho de que logran una ganancia de 11 dB sobre los DDD para un valor de BER 10^{-2} .
- Los DPF, DME de 32 trayectorias y el Detector Genie Aided muestran resultados de aproximadamente el mismo orden de magnitud para valores de SNR del primer usuario menores de 30 dB.
- Los valores de BER obtenidos para los DME con 4 y 32 trayectorias son muy similares entre sí especialmente para SNR mayor de 30 dB y comparables a los obtenidos con DFP.
- Por todo lo anterior, puede afirmarse que el DME es también más favorable ya que requiere aproximadamente sólo un 1/35 de la complejidad del DFP.

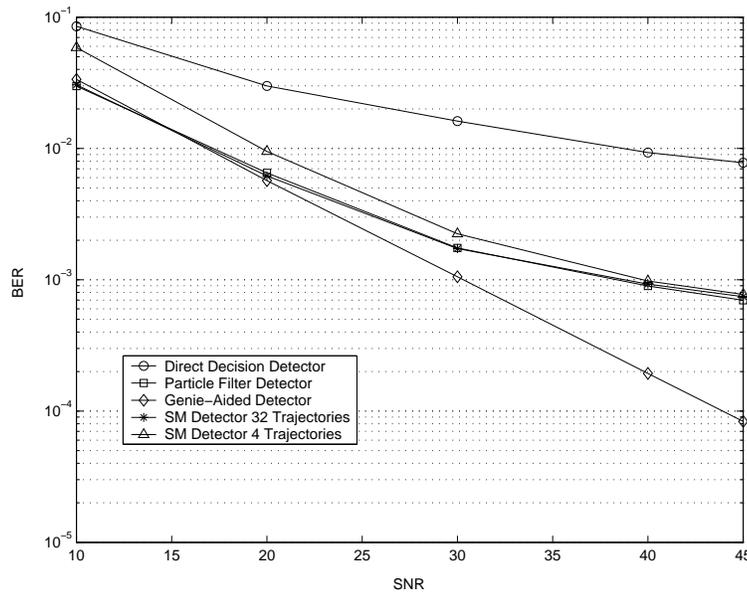


Figura 4.8: BERs vs. SNR para varios detectores. Usuarios con distinta potencia. $\Omega = 0,05$, $f_d = 500Hz$, 15 usuarios

En la Figura 4.8 se representan los resultados de las simulaciones para el caso en que $\Omega_d = 0,05$. Los resultados más importantes son los siguientes:

- En cuanto a la comparación entre los rendimientos de los distintos detectores, se pueden hacer las mismas observaciones que se han hecho para las simulaciones correspondientes a $\Omega_d = 0,03$. Siendo de nuevo el DME con 4 trayectorias el más favorable.
- El comportamiento global de todos los detectores en este caso es peor si lo comparamos con el rendimiento cuando $\Omega_d = 0,03$ lo cual era de esperar ya que los efectos negativos del canal toman más relevancia porque el desvanecimiento es más rápido. Es de destacar el hecho de que los DFP y DME poseen una ganancia de 20 dB sobre la de los DDD para un valor de BER 10^{-2} .

4.11. Conclusiones

En este capítulo se ha partido de un Modelo de Espacio de Estados de Tiempo y Observación para un sistema de comunicaciones digitales en el se produce acceso múltiple al medio modelado como un canal plano tipo

Rayleigh con desvanecimiento rápido. Para resolver el problema de la Detección Multiusuario en las condiciones explicadas se ha propuesto en este trabajo la aplicación de un algoritmo de tipo Filtrado de Partículas que hemos denominado Algoritmo *M*-Estocástico. Las mejoras de este tipo de algoritmo son:

- Proporciona información estadística relacionada con la probabilidad por lo cual se permite conservar y estudiar varias trayectorias en el algoritmo de filtrado a cada una de las cuales se le asocia un peso que da idea de su importancia. De este modo se evitan errores de decisión como los que pueden ocurrir en el Algoritmo Dirigido por Decisión, en el que estos errores en una etapa se propagan a las etapas posteriores. De aquí las diferencias en comportamiento que se aprecian en las simulaciones.
- Resuelve el problema de degeneración de trayectorias que se produce en los algoritmos generales de Filtrado de Partículas cuando tratan con magnitudes discretas. La existencia de degeneración implica ineficacia. Por tanto, el Algoritmo *M*-Estocástico es más eficiente que los algoritmos de filtrado de partículas generales y por tanto es más adecuado para aplicaciones reales.
- El uso del Algoritmo *M*-Estocástico no se ve afectado por situaciones "near-far" por lo que se evitan costes computacionales adicionales relacionados con la aplicación de algoritmos de control de potencia.
- El DME ha demostrado un rendimiento similar al DFP con una complejidad computacional mucho menor, del orden de $1/35$ para 4 trayectorias y $1/10$ para 32 con lo que resulta de nuevo ser un detector mucho más adecuado para aplicaciones reales.

Chapter 5

LDPC Codes and Factor Graph representation

The Factor Graph (FG) representation and the Sum Product algorithm (SPA) provide a general and powerful framework to derive low-complexity Bayesian detection and decoding algorithms. In particular, the algorithms to be derived in this part of the thesis are iterative. Due to this characteristic, they are particularly suited to the decoding of codes such as low-density parity-check (LDPC) and turbo codes, whose decoding algorithm are typically iterative (and suboptimal) even in the fully coherent setting (all channel parameters known). In this chapter, the theoretical background of the Factor Graph representation and the LDPC codes is briefly introduced. In Section 5.1 the Shannon Theorem is presented and its importance in the field of Communications is discussed. In Section 5.2 an introduction to LDPC codes is made. A brief historical review on Factor Graphs is presented in Section 5.3. Factor Graphs are introduced in Section 5.4 and, finally, the Sum Product Algorithm is presented in Section 5.5.

5.1. Shannon's Theorem

This theorem describes the maximum possible efficiency of error-correcting methods versus levels of noise interference and data corruption. The theory does not describe how to construct the error-correcting method, it only tells how good the best possible method can be. Shannon's theorem has a wide ranging applications in both communications and data storage applications. This theorem is of foundational importance to the modern field of information theory. If we had such a thing as a infinite-bandwidth and noise-free analog channel we could transmit unlimited amounts of error-free data over it per unit of time. However real life signals have both bandwidth and noise-inference limitations. Bandwidth limitations alone do not impose a cap on maximum information transfer. This is because it is still possible (at least in a thought-experiment model) for the signal to take on an infinite number of different voltage levels on each cycle, with each slightly different level being assigned a different meaning or bit sequence. If we combine both noise and bandwidth limitations, however, we do find there is a limit to the amount of information that can be transferred, even when clever multi-level encoding techniques are used. This is because the noise signal obliterates the fine differences that distinguish the various signal levels, limiting in practice the number of detection levels we can use in our scheme.

Reliable communications are in great demand today. People desire higher bandwidth communications in devices consuming minimal energy. It is, therefore, important to use the transmission systems available as effectively as possible. The science of finding efficient schemes by which information can be coded for reliable transmission through a noisy channel is called coding theory. To achieve reliable communication, it is imperative to send input elements that are correlated. This leads to the concept of a code defined as a finite set of vectors over the input alphabet. If we assume that all the vectors have the same length, and call this length the block length of the code. If the number of vectors is $K = 2^k$, then every vector can be described with K bits. If the length of the vectors is n , then in n times use of the channel k bits have been transmitted. We say then that the code has a rate of k/n bits per channel use, or k/n bpc.

Supposing we send a code word and receive a vector over the output alphabet. If the channel allows for errors, then there is no general way of telling which codeword was sent with absolute certainty. However, we can find the most likely codeword that was sent, in the sense that the probability that this codeword was sent given the observed vector is maximized. To see that we really can find such a codeword, simply list all the K codewords, and calculate the conditional probability for the individual codewords. Then find

the vector or vectors that yield the maximum probability and return one of them. This decoder is called the maximum likelihood decoder and it is not perfect: it takes a lot of time and it may make mistakes.

Given a communication channel, Shannon proved that there exists a number, called capacity of the channel, such that reliable transmission is possible for rates arbitrarily close to the capacity, and reliable transmission is not possible for rates above capacity.

Shannon also proved the existence of codes of rates arbitrarily close to capacity for which the probability of error of the maximum likelihood decoder goes to zero as the block length of the code goes to infinity. In fact, Shannon proved that the decoding error of the maximum likelihood decoder goes to zero exponentially fast with the block length.

5.2. LDPC Codes, an introduction

LDPC codes were developed in the PhD thesis of Gallager in 1963. Using an iterative decoding scheme the codes were shown to perform very well especially at long block lengths. However, these codes were largely forgotten, as their complexity was considered impractical for the computing capabilities of the time. Besides, the lack of convincing empirical evidence and problems with implementation at the time resulted in the code being overlooked as near capacity error correcting codes. However, the focus on Turbo codes ignited interest in related schemes, which resulted in the nearly simultaneous rediscovery of LDPC codes by several different research groups. So, over thirty years later the codes were rediscovered and they are now well known to outperform even the best known turbo codes. Breakthroughs were made in finding LDPC codes that could theoretically use almost all of a channel capacity. The discovery that such codes could achieve a decoding performance just 0.0045 dB from the Shannon limit means that they now possess the unrivaled status as the best known practical codes that exist. Amazingly, the remarkable error correcting performance is not well understood. Consequently, a great deal of research has gone into LDPC codes in the last ten years, resulting in hundreds of authors on the subject and thousands of conference and journal papers.

In essence, an LDPC code is a linear code in which the parity-check matrix is very sparse, and in which the nonzero entries are placed randomly, selected from a specified ensemble. As in the case with Turbo codes, it is possible to represent an LDPC code on a graph. In this context, LDPC are defined as linear codes obtained from sparse bipartite graphs. LDPC codes can be represented by a bipartite graph with n left nodes (called message nodes) and

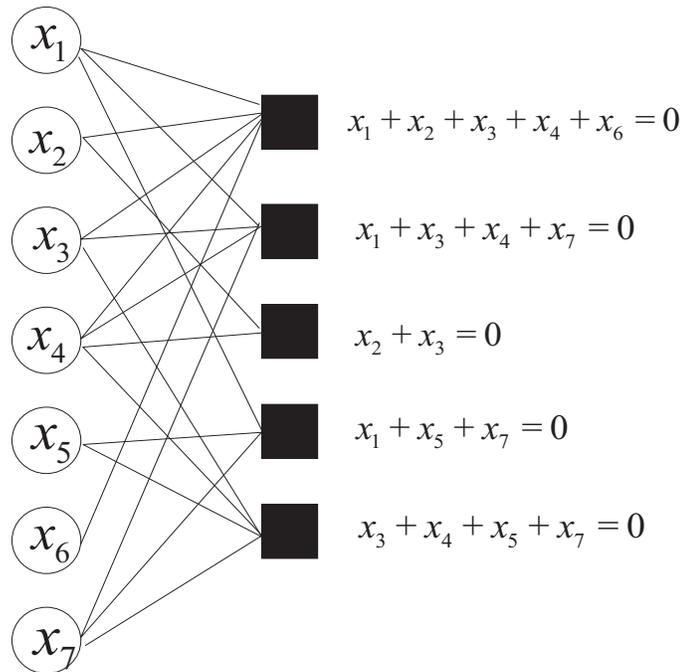


Figure 5.1: Graphical representation of a LDPC code

r right nodes (called check nodes). Bipartite means that the graph contains two classes of nodes, so called variable nodes and function nodes. The graph gives rise to a linear code of block length n and dimension at least $n - r$ in the following way: the n coordinates of the codewords are associated with the n message nodes. The codewords are those vectors such that for all check nodes the sum of the neighboring positions among the message nodes is zero. This graphical representation is shown in Figure 5.1

Useful representations for LDPC codes are Factor Graphs, which are related to both Tanner graphs and Bayesian networks. Furthermore, belief propagation decoding can be accomplished using a message-passing scheme over the factor graph called the Sum-Product Algorithm (SPA). As well, the sparseness and randomness of the parity check matrix and its associated factor graph can be shown to lead to the property of long cycles.

The graph representation is analogous to a matrix representation by looking at the adjacency matrix of the graph: let H be a binary rxn -matrix which the entry (i, j) is 1 if and only if the i th check node is connected to the j th message node in the graph. Then the LDPC defined by the graph is the set of vectors $c = (c_1, \dots, c_n)$ such that $H.c^T = 0$. The matrix H is called a parity check matrix of the code. Conversely, any binary rxn -matrix gives rise to a bipartite graph between n messages and r check nodes, and the

code defined as the null space of H is precisely the code associated to this graph. Therefore, any linear code has a representation as a code associated to a bipartite graph. However, not every bilinear code has a representation by a sparse bipartite graph. If it does, then the code is called a low-density parity-check (LDPC) code.

5.2.1. Regular LDPC Codes

Initial research on LDPC codes focused on regular codes, in which the every row of the parity check matrix had d_c nonzero entries, and every column had d_v nonzero entries, where $d_v < d_c$. Empirical results for these codes showed a performance that was comparable to Turbo codes, with a small gap to the Shannon bound. However, while Turbo codes seemed to be plagued with an error floor at low probability of error, the empirical results indicated that LDPC codes might avoid this phenomenon. Indeed, under (albeit infeasible) optimal decoding, LDPC codes have been shown to achieve the Shannon limit.

Floor of Turbo codes has been shown to be a fundamental property of the code's low free distance [21], which would not be mitigated under optimal decoding. In 2001, a landmark paper by Richardson and Urbanke established analytically that regular LDPC codes, decoded with the SPA, can achieve arbitrarily low probabilities of error in the limit of long block length. This achievement generalized earlier analytical work by Luby. To accomplish this analysis, the authors proposed density evolution (DE), which exploited the property of long cycles. By examining how messages were calculated, the long cycles were shown to ensure that all the messages passed under the SPA were independent. Thus, using a very complicated procedure, the probability density functions of the messages could be tracked from iteration to iteration, and from these functions the probability of error at each iteration could be calculated. Density evolution is currently the highest standard of analysis that can be applied to any LDPC decoding application, and has been applied in many areas, including: LDPC decoding in Rayleigh fading [24]; multiuser detection-decoding [25]; decoding of LDPC coded modulation [26].

5.2.2. Irregular LDPC Codes

Meanwhile, it was noticed that the remaining gap to the Shannon bound could be closed by using irregular LDPC codes [23]. In an irregular LDPC code, the number of ones in each row and column of the parity-check matrix is no longer constant. Instead, for a row (or column) selected at random uniformly from the set of all rows (or columns, respectively), the number of

nonzero entries is a random variable governed by a distribution known as a degree sequence. Obtaining a good degree sequence is a non-trivial design problem. DE can be adapted to search for good degree sequences [27]; however, the high complexity of DE and the large number of candidate degree sequences motivated the search for alternatives. Several research groups have proposed methods for approximating DE analysis. Approximate DE normally distills the exact message densities down to a single parameter, where this single parameter frequently indexes a family of densities, and the parameter value is chosen as the member of the family that is the best approximation for the true density of the messages.

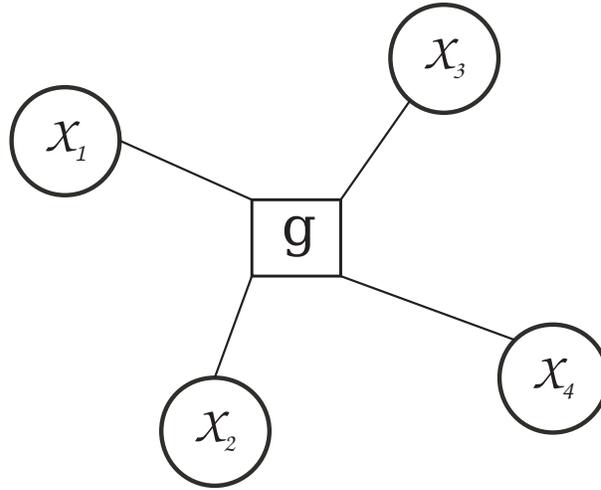
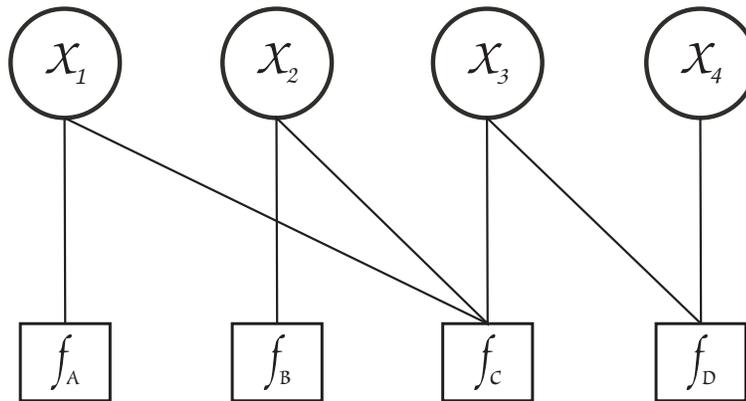
Recently, an irregular LDPC code was found that performed to within 0.0045 dB of the Shannon limit in Gaussian noise [32]. With the problem of communicating over the Gaussian channel seemingly solved for arbitrarily long codes, many researchers have begun adapting LDPC codes to more complicated channels. In particular, attention has turned to channels with memory, which are present in a number of practically interesting scenarios, such as wireless fading channels.

5.3. Factor Graphs: some historical remarks

In 1981 Tanner introduced a new type of graph that can be used to represent codes. It contains two types of nodes: the first class of nodes represents the symbols of a code (i.e., one node for each x_i of the codeword x) and the second class represents the parity-check functions (i.e., one node for each row of the parity-check matrix). There is a connection between a node which belongs to the symbol class and a node representing a parity-check function if and only if the corresponding symbol is involved in this particular parity-check. As a code is uniquely defined if the number of codeword symbols is known (i.e., if the length of a codeword is known) and if all parity-check functions are given (where it must be clearly specified which symbols are involved in which parity-check), such a Tanner graph uniquely defines the code. Tanner formulated two algorithms that realize a procedure for decoding a codeword when applied to the graph.

5.4. Definition of Factor Graph

A factor graph is a bipartite graph that expresses how a global function of several variables factors into a product of local functions. Bipartite means that the graph contains two classes of nodes, so called variable nodes and

Figure 5.2: Graphical representation of g Figure 5.3: Alternative graphical representation of g

function nodes. The variable nodes represent variables and the factor nodes represent each local function. A variable node is connected to a function node if and only if this variable is an argument of the corresponding function. There are no edges between nodes that belong to the same class, and the edges are not directed.

The following example will help us to understand the concept of Factor Graph better. Consider a function g of four different variables and suppose that this function can be expressed as a product in the next way

$$g(x_1, x_2, x_3, x_4) := f_A(x_1) f_B(x_2) f_C(x_1, x_2, x_3) f_D(x_3, x_4) \quad (5.1)$$

In Figures 5.2 and 5.3 we can see two different factor graph representation. Obviously the first one does not give any new information. However, the

second one is very interesting since we can see graphically the relationships between the local functions and the variables.

Now suppose that we are interested in evaluating the marginal function $g_1(x_1)$, where

$$g_1(x_1) = \sum_{x_2, x_3, x_4} g(x_1, x_2, x_3, x_4) \quad (5.2)$$

In the following the notation introduced in [57] is going to be used. This notation is called the non-sum or summary. So, instead of indicating the variables being summed over, the variables not being summed over are indicated. So, the expression 5.2 can be written as:

$$g_1(x_1) = \sum_{\sim\{x_1\}} g(x_1, x_2, x_3, x_4) \quad (5.3)$$

Taking into account the factorization of g given by equation 5.1 the last equation can be written as:

$$g_1(x_1) = f_A(x_1) \left(\sum_{x_2} f_B(x_2) \left(\sum_{x_3} f_C(x_1, x_2, x_3) \left(\sum_{x_4} f_D(x_3, x_4) \right) \right) \right) \quad (5.4)$$

And using the summary notation equation 5.4 can be written as:

$$g_1(x_1) = f_A(x_1) \sum_{\sim\{x_1\}} \left(f_B(x_2) f_C(x_1, x_2, x_3) \sum_{\sim\{x_3\}} f_D(x_3, x_4) \right) \quad (5.5)$$

If we denote:

$$f_I(x_3) = \sum_{\sim\{x_3\}} f_D(x_3, x_4) f_{II}(x_1) \quad (5.6)$$

$$= \sum_{\sim\{x_1\}} (f_B(x_2) f_C(x_1, x_2, x_3) f_I(x_3)) g_1(x_1) \quad (5.7)$$

$$= f_A(x_1) f_{II}(x_1) \quad (5.8)$$

Thus, it can be seen that the task of evaluating the complicated sum in 5.4 can be split into several smaller tasks which can be done one after another. The summation process actually happens locally and only the local results are used in a more global context. This idea can easily be translated to the factor graph and it is realized in the sum-product algorithm.

5.5. Sum-product Algorithm

The aim of this section is to introduce the Sum-product algorithm in an intuitive way. This algorithm is used to obtain all the marginal function at the same time using the Factor Graph representation. One of the most used similitude to understand this process is to imagine that there is a processor associated with each vertex of the factor graph and that the Factor-graph edges represent channels by which these processors may communicate. Thus, messages sent between processors are always simply some appropriate description of some marginal function. This process of sending messages is initiated at the leaves. Each vertex remains idle until messages have arrived on all but one of the edges incident on it. Once these messages have arrived, the vertex is able to compute a message to be sent on the one remaining edge to its neighbor.

The sum-product algorithm operates according to a simple rule called **The Sum-Product Update Rule**. This rule sets that *the message sent from a node v on an edge c is the product of the local function at v (or the unit function if v is a variable node) with all messages received at v on edges other than c , summarized for the variable associated with c .*

So if one goes from a variable node to a local function the corresponding message is calculated as follows:

$$\mu_{x \rightarrow f}(x) = \prod_{h \in n(x) \setminus \{f\}} \mu_{h \rightarrow x}(x) \quad (5.9)$$

On the other hand, if one goes from a local function node to a variable node, the message is calculated as:

$$\mu_{f \rightarrow x}(x) = \sum_{\sim \{x\}} \left(f(X) \prod_{y \in n(f) \setminus \{x\}} \mu_{y \rightarrow f}(y) \right) \quad (5.10)$$

In order to clear up the way of using the sum-product update Rule we are going to apply it the the example shown in the Figure 5.4.

The calculation of the messages is divided into four different steps. The first step begins in the leaves of the graph, this is, the variable nodes corresponding to the variables x_1 , x_3 and x_4 . The messages calculated in this step are:

$$\mu_{x_1 \rightarrow f_A} = 1 \quad (5.11)$$

$$\mu_{x_3 \rightarrow f_B} = 1 \quad (5.12)$$

$$\mu_{x_4 \rightarrow f_C} = 1 \quad (5.13)$$

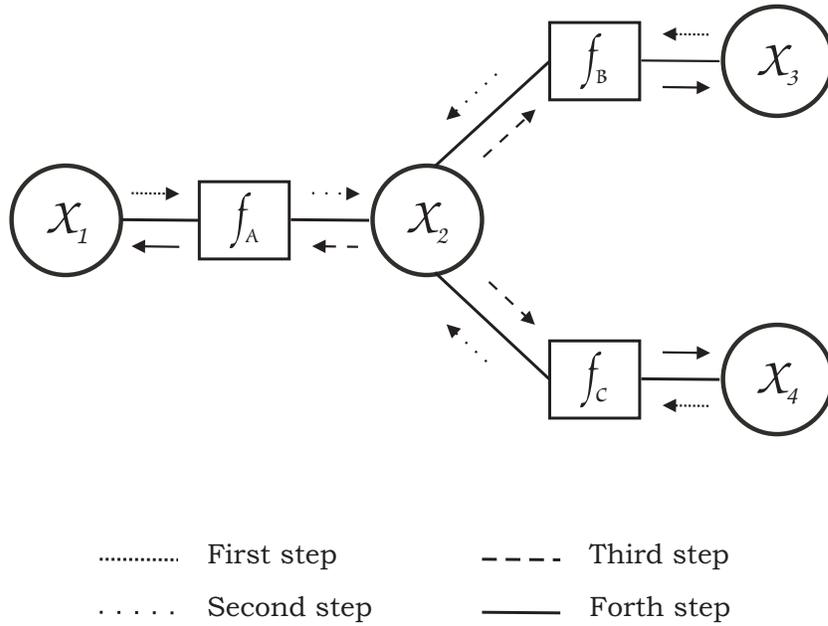


Figure 5.4: Example SPA

Following the Figure 5.4 the Sum-Product Algorithm calculates the messages sent from the local functions f_A , f_B and f_C to the variable node x_2 in the second step as follows:

$$\mu_{f_A \rightarrow x_2} = \sum_{\sim\{x_2\}} (f_A(x_1, x_2) \mu_{x_1 \rightarrow f_A}) \quad (5.14)$$

$$= \sum_{\sim\{x_2\}} (f_A(x_1, x_2))$$

$$= \sum_{\{x_1\}} (f_A(x_1, x_2))$$

$$\mu_{f_B \rightarrow x_2} = \sum_{\{x_3\}} (f_B(x_2, x_3)) \quad (5.15)$$

$$\mu_{f_C \rightarrow x_2} = \sum_{\{x_4\}} (f_C(x_2, x_4)) \quad (5.16)$$

In the third step the Sum-Product Algorithm computes the messages from the variable node x_2 to the nodes associated to the local functions f_A , f_B and f_C .

$$\begin{aligned}
\mu_{x_2 \rightarrow f_A} &= \mu_{f_B \rightarrow x_2} \mu_{f_C \rightarrow x_2} \\
\mu_{x_2 \rightarrow f_B} &= \mu_{f_A \rightarrow x_2} \mu_{f_C \rightarrow x_2} \\
\mu_{x_2 \rightarrow f_C} &= \mu_{f_A \rightarrow x_2} \mu_{f_B \rightarrow x_2}
\end{aligned} \tag{5.17}$$

The last step of the Sum-Product Algorithm calculates the message sent from the local function node f_A to the variable node x_1 , the one sent from local function node f_B to the variable node x_3 and the message sent from f_C to variable node x_4 .

$$\begin{aligned}
\mu_{f_A \rightarrow x_1} &= \sum_{\sim\{x_1\}} (f_A(x_1, x_2) \mu_{x_2 \rightarrow f_A}) \\
\mu_{f_B \rightarrow x_3} &= \sum_{\sim\{x_3\}} (f_B(x_2, x_3) \mu_{x_2 \rightarrow f_B}) \\
\mu_{f_C \rightarrow x_4} &= \sum_{\sim\{x_4\}} (f_C(x_2, x_4) \mu_{x_2 \rightarrow f_C})
\end{aligned} \tag{5.18}$$

Once all the messages are computed, all the marginalized function can be calculated. The way to express them is very simple. There are two different ways of calculating all the $g_i(x_i)$:

- as the product of all the messages directed toward the variable x_i
- as the product of two messages that are passed (in opposite directions) over any single edge incident on x_i .

Applying the previous rules, the marginalized functions are:

$$g_1(x_1) = \mu_{f_A \rightarrow x_1}(x_1) \tag{5.19}$$

$$\begin{aligned}
g_2(x_2) &= \mu_{f_A \rightarrow x_2}(x_2) \mu_{f_B \rightarrow x_2}(x_2) \mu_{f_C \rightarrow x_2}(x_2) \\
&= \mu_{f_A \rightarrow x_2}(x_2) \mu_{x_2 \rightarrow f_A}(x_2) \\
&= \mu_{f_B \rightarrow x_2}(x_2) \mu_{x_2 \rightarrow f_B}(x_2) \\
&= \mu_{f_C \rightarrow x_2}(x_2) \mu_{x_2 \rightarrow f_C}(x_2)
\end{aligned} \tag{5.20}$$

$$g_3(x_3) = \mu_{f_B \rightarrow x_3}(x_3) \tag{5.21}$$

$$g_4(x_4) = \mu_{f_C \rightarrow x_4}(x_4) \tag{5.22}$$

Chapter 6

Particle Filtering for joint iterative decoding and phase estimation

Several iterative decoding algorithms under Bayesian frameworks have been considered in the literature for channels with unknown constant or time-varying phase. While non-Bayesian schemes are efficient for constant phase estimation, they perform significantly worse in the presence of time-varying phase. That motivates us to study the iterative decoding over channels with time-varying phase noise and to use Bayesian sequential Monte Carlo methods, or Particle Filters, to perform the on-line phase estimation. The Particle Filters are powerful tools for numerically computing the optimal estimators when exact solutions cannot be analytically derived. They employ discrete measures with random supports, or particles, for sequentially estimating the posterior probability distributions of the unknowns. When applied to phase estimation, the Particle Filters allow a continuous exploration of the phase space with the evolution of particles in the time.

On the other hand, Factor Graph representation [57] provides a powerful framework to reveal important yet often obscure factorizations in different systems. Such factorization allows the use of the efficient Sum-Product algorithm for inference. Sum-Product algorithm has shown to be an important tool to reinterpret a number of well-known algorithm in digital communications [88] and to design more efficient and powerful algorithms.

In this chapter, we propose to combine the Sum-Product algorithm with the Particle Filtering Algorithm in order to perform iterative decoding over channels with phase noise. In [88], the authors propose to discretize the phase space using equally spaced values. However, in this case the complexity may be very high because all the possible branches in a trellis must be analyzed. Since the complexity of Particle Filtering is directly proportional to the number of particles and hence, it may be much smaller than that of a full-trellis exploration. Here, we thus propose to reduce the complexity of the discrete approach by using Particle Filtering methods [2]. Particle Filtering techniques were applied to this problem in [23] for the first time. However, the number of particles used in this algorithm is high and depends on the length of the code. In this chapter, we use Particle Filtering to estimate the phase and, afterwards, we use that estimation to perform decoding. The proposed algorithm is independent of the length of the code and thus the number of particles used can be reduced. One of the major drawbacks of the Particle Filtering techniques is the degeneracy phenomenon. In order to reduce as much as possible the negative effects of that phenomenon a resampling step has to be introduced. That resampling step is not considered in [23] and as a result, several iterations are needed to estimate the phase. However the proposed algorithm here includes the resampling step and thus the number of iterations can be reduced. Ultimately, the computational complexity is also reduced.

This chapter is organized in the following way: in Section 6.1 the problem is formulated. In Section 6.2 the Phase Noise is briefly introduced. The application of the Sum Product algorithm to the Factor Graph representation of the system is done in Section 6.3. In Section 6.5 the equally spaced discretization of the phase space proposed by [88] is presented. Sections 6.6 and 6.7 are devoted to solutions based on canonical distributions: the Tikhonov and the Gaussian parametrizations. Three different Particle Filtering approaches are introduced in Sections 6.8, 6.9 and 6.10. The simulations results are presented in Section 6.11. Section 6.12 contains some concluding remarks.

6.1. System Model

The system under consideration in the present chapter is represented by the equation:

$$r_k = c_k e^{j\theta_k} + n_k, \quad k = 0, 1, \dots, K - 1 \quad (6.1)$$

Where:

- c_k are the components of the sequence of a complex linearly modulated symbols $\mathbf{c} = (c_0, c_1, \dots, c_{K-1})$. It is assumed that the sequence \mathbf{c} is a codeword of the channel code \mathcal{C} . They are constructed over an M-ary modulation constellation $\chi \subset \mathbb{C}$. Furthermore, pilot symbols and differential encoding can be included as a part of the code \mathcal{C} .
- θ_k are the components of the vector of the channel phase $\theta = (\theta_0, \theta_1, \dots, \theta_{K-1})$. This phase is modeled according to the following random walk model:

$$\theta_k = \theta_{k-1} + v_k \quad (6.2)$$

where v_k is a white real gaussian process with zero mean and variance σ_p .

- n_k is the k -th component of the vector of noise samples. This vector has complex circularly symmetric components. Each one of these components are $n_k \sim \mathcal{N}_{\mathcal{C}}(0, 2\sigma^2)$.

6.2. Phase Noise

Phase noise is a topic of theoretical and practical interest in fields like communications, electronic circuits, optics, biology, etc Phase noise is related to oscillators, this kind of systems are present everywhere in physical systems, especially electronic and optical ones. For example, in communication systems, they are used for frequency translation of information signals and for channel selection. Oscillators are also present in digital electronic systems which require a time reference, i.e., a clock signal, in order to synchronize operations.

Phase noise can be caused by a number of conditions, but it is mostly affected by an oscillators frequency stability. An oscillators frequency stability is defined as the measure of the degree to which an oscillator maintains the same value of frequency over a given time. Noise is of major concern in oscillators, because introducing even small noise into an oscillator leads to dramatic changes in its frequency spectrum and timing properties. This phenomenon,

peculiar to oscillators, is known as phase noise. A perfect oscillator would have localized tones at discrete frequencies but any corrupting noise spreads these perfect tones, resulting in high power levels at neighboring frequencies. This effect is the major contributor to undesired phenomena such as inter-channel interference, leading to increased bit-error-rates in communication systems. Another manifestation of the same phenomenon, is important in clocked and sampled-data systems; uncertainties in switching instant caused by “thermal” and “flicker” or $1/f$ noise, lead to synchronization problems. Characterizing how noise affects oscillators is therefore crucial for practical applications. The problem is challenging, since oscillators constitute a special class among noise physical systems: their autonomous nature makes them unique in their response to perturbations. As it has been explained, the

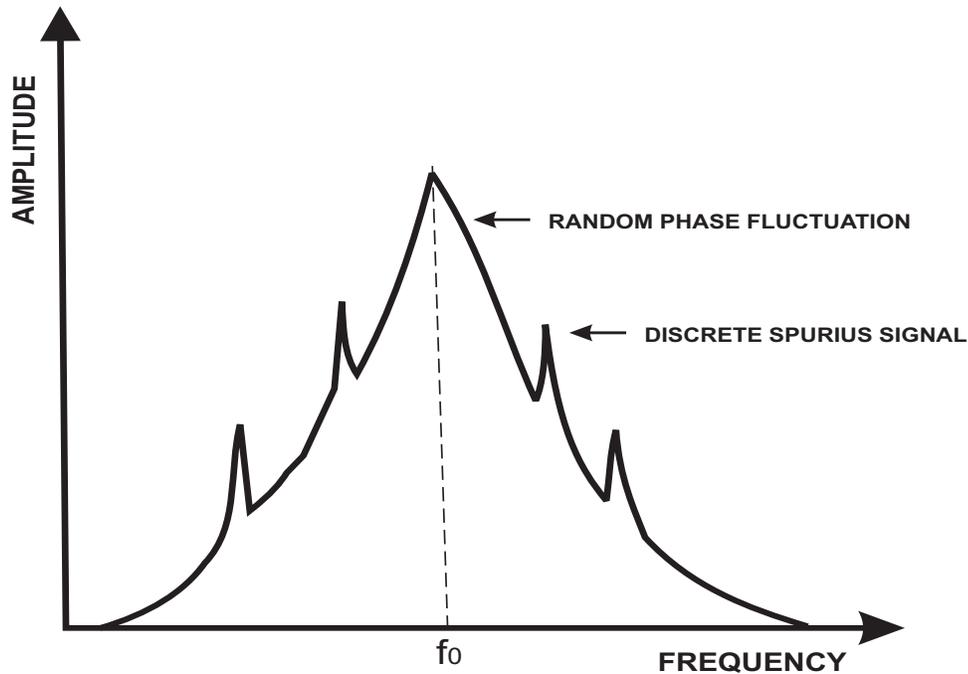


Figure 6.1: Spectrum analyzer display of phase noise

term phase noise is used to describe phase fluctuations due to the random frequency fluctuations of a signal. In order to visualize these fluctuations, simple case where we can see them graphically is going to be presented. Lets consider an ideal sinewave oscillator that may be described by:

$$V(t) = V_0 \sin 2\pi ft$$

In this equation, V_0 is the nominal amplitude of the signal, and f is the nominal frequency of oscillation. The instantaneous output of an oscillator

may be represented by:

$$V(t) = V_0|1 + A(t)| \sin(2\pi ft + \theta(t))$$

where $A(t)$ and $\theta(t)$ represent the amplitude and phase fluctuations of the signal respectively. The phase term may be random or discrete and can be displayed on a spectrum analyzer as it can be seen in 6.1.

As shown in this figure, there are two types of fluctuating phase terms. The first term, described by the discrete signals, is called “spurious”. They appear as distinct components in the spectral density plot. The second term, random in nature, appears as random phase fluctuations and is commonly called “phase noise”.

6.3. Factor Graph representation of the system

In this section the application of the Sum-Product Algorithm to the Factor Graph representation is presented. This representation is used to perform joint iterative detection and decoding and phase estimation. The first step is to select the function whose Factor Graph representation is analyzed. This function is the joint posterior probability function $P(\mathbf{b}, \theta | \mathbf{r})$. If one is interested in making the optimal decision, one possibility is to use the rule that minimizes the average bit-error probability. This is given by:

$$\hat{b}_i = \arg \max_{b \in \mathbb{F}_2} P_{bit,i}(b | \mathbf{r}) \tag{6.3}$$

where $P_{bit,i}(b | r)$ is the posteriori *pmf* for the *i*-th information bit given the received signal vector $\mathbf{r} = (\mathbf{r}_0, \mathbf{r}_1, \dots, \mathbf{r}_{K-1})$. This function is precisely a marginalization of the joint posterior probability function with respect to the variable θ and to all b_j for $j \neq i$. This marginalization can be calculated approximately in a low complexity way using the Sum-Product Algorithm to the Factor Graph representation of the latter.

If the Bayes theorem is used, the function $P(\mathbf{b}, \theta | \mathbf{r})$ can be factorized as follows:

$$P(\mathbf{b}, \theta | \mathbf{r}) \propto P(\mathbf{b}) p(\theta) p(\mathbf{r} | \theta, \mathbf{b}) \tag{6.4}$$

The factorization goes on if one takes into account that the information bits are i.i.d. and that the AWGN channel for given θ is memoryless. Furthermore, the code indicator function, $\chi[\mathbf{c} = \mu_C(\mathbf{b})]$, can be introduced. This function is equal to 1 if \mathbf{c} is a codeword corresponding to \mathbf{b} and zero otherwise. After

these considerations, the expression 6.4 can be rewritten as:

$$\begin{aligned} P(\mathbf{b}, \theta | \mathbf{r}) &\propto \chi[\mathbf{c} = \mu_C(\mathbf{b})] p(\theta) p(\mathbf{r} | \theta, \mathbf{c} = \mu_C(\mathbf{b})) \\ &\propto \chi[\mathbf{c} = \mu_C(\mathbf{b})] p(\theta) \prod_{k=0}^{K-1} p(r_k | \theta_k, c_k) \end{aligned} \quad (6.5)$$

The next step is to factorize the function $p(\theta)$. This function only depends on the phase noise vector. Taking into account that the components of this vectors follow a 1st order Markov model, it can be written:

$$p(\theta) = p(\theta_0) \prod_{k=1}^K p(\theta_k | \theta_{k-1}) \quad (6.6)$$

Substituting this expression for $p(\theta)$ in 6.5

$$P(\mathbf{b}, \theta | \mathbf{r}) \propto \chi[\mathbf{c} = \mu_C(\mathbf{b})] p(\theta_0) \prod_{k=1}^{K-1} p(\theta_k | \theta_{k-1}) \prod_{k=0}^{K-1} p(r_k | \theta_k, c_k) \quad (6.7)$$

Finally, for shortness sake the notation introduced in the paper [21] related to the probability $p(r_k | \theta_k, c_k)$ is going to be used. This function can be written as:

$$f_k(\theta_k, c_k) \triangleq \exp \left\{ \frac{1}{\sigma^2} \operatorname{Re} [r_k c_k^* e^{-j\theta_k}] - \frac{|c_k|^2}{2\sigma^2} \right\} \propto \exp \left\{ -\frac{1}{2\sigma^2} |r_k - c_k e^{j\theta_k}|^2 \right\} \quad (6.8)$$

Thus, the final factorization of the function under study is:

$$P(\mathbf{b}, \theta | \mathbf{r}) \propto \chi[\mathbf{c} = \mu_C(\mathbf{b})] p(\theta_0) \prod_{k=1}^{K-1} p(\theta_k | \theta_{k-1}) \prod_{k=0}^{K-1} f_k(\theta_k, c_k) \quad (6.9)$$

In Figure 6.2 the Factor Graph for the previous factorization is shown. In this Figure, the arrows show the direction of the messages. Once the graphic representation has been presented, the next step is to analyze and calculate the different messages involved with it. These messages are:

- $P_d(c_k)$ is the message sent from variable node c_k to local function node f_k . In the first iteration, all the probabilities associated to different values of c_k are the same. At each iteration, these messages (probabilities) are updated.

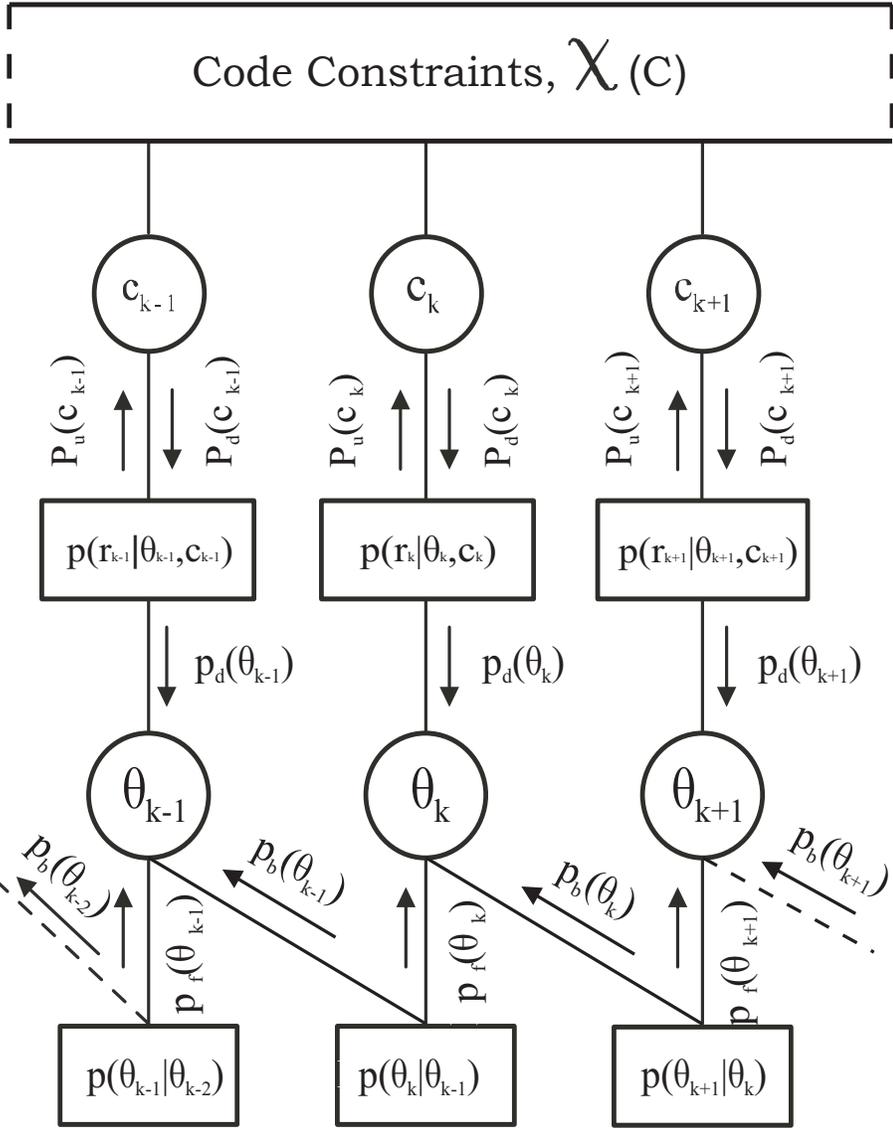


Figure 6.2: Factor Graph representation of the system

- $p_d(\theta)$ is the message from local variable node f_k to variable node θ . If the Updating Rule of the Sum-Product Algorithm is used, these function can be calculated as:

$$p_d(\theta_k) \propto \sum_{x \in \mathcal{X}} P_d(c_k = x) f_k(c_k = x, \theta_k) \quad (6.10)$$

- $p_f\theta_k$ is the message sent from local function node $p(\theta_k|\theta_{k-1})$ to variable node θ_k . In order to calculate the expression of this message the Update Rule has to be used taking into account that the movement is being done from a local function node to a variable node (see expression 5.10):

$$p_f(\theta_k) \propto \int_0^{2\pi} p_d(\theta_{k-1}) p_f(\theta_{k-1}) p(\theta_k|\theta_{k-1}) d\theta_{k-1} \quad (6.11)$$

- $p_b(\theta_k)$ is the message sent from local function node $p(\theta_{k+1}|\theta_k)$ to variable node θ_k . Again, the Updating Rule has to be applied to calculate this message:

$$p_b(\theta_k) \propto \int_0^{2\pi} p_d(\theta_{k+1}) p_b(\theta_{k+1}) p(\theta_{k+1}|\theta_k) d\theta_{k+1} \quad (6.12)$$

There is an initial condition for the iterative decoding method. This initial condition consist of setting the following values:

$$p_b(\theta_K) = p_f(\theta_0) = \frac{1}{2\pi} \quad (6.13)$$

- $P_u(c_k)$ is the message sent from local function node f_k to variable node c_k . Its expression is:

$$P_u(c_k) \propto \int_0^{2\pi} p_f(\theta_k) p_b(\theta_k) f_k(c_k, \theta_k) d\theta_k \quad (6.14)$$

The vector of messages $P_u(c_k) : k = 0, \dots, K - 1$ is a sequence of a posteriori pmfs $p(r_k|c_k)$. These messages are processed in the upper part of the graph according to a standard Sum Product Algorithm. After each iteration, there are two important results, on the one hand, the vector of updated messages $P_d(c_k)$ and on the other hand, the updated estimates of the a posteriori probabilities $P_{bit,i}$.

6.4. Solutions of the Sum Product Algorithm

The main problem of the application of the Sum Product Algorithm to this problem is the one derived from the fact that the phase space is continuous. The integrals in the expressions (6.12), (6.11) and (6.14) cannot be solved analytically in most of the cases. Some solutions have been already proposed like Fourier parametrization, Tikhonov parametrization, etc. In the following, different algorithms are going to be proposed. These algorithms try to solve the integral discretizing the continuous phase space.

6.5. Equally Spaced Discretization

The first method is the most simple because the discretization is done dividing the phase space into L different levels. In this way, the possible values that the phase can take are:

$$\Theta = \left\{0, \frac{2\pi}{L}, \dots, \frac{2\pi(L-1)}{L}\right\}$$

Using the previous discretization, the message $p_d(\theta_k)$ can be written as:

$$p_d\left(\theta_k^{(i)}\right) \propto \sum_{x \in \mathcal{X}} P_d(c_k = x) f_k(c_k = x, \theta_k^{(i)}) \quad (6.15)$$

where θ_i represents each of the possible discrete values that the phase can take. In the rest of the message expressions obtained with the Sum-Product Algorithm, the integrals are replaced by summations in the following way:

$$p_f\left(\theta_k^{(i)}\right) \propto \sum_j p_d\left(\theta_{k-1}^{(j)}\right) p_f\left(\theta_{k-1}^{(j)}\right) p\left(\theta_k^{(i)}|\theta_{k-1}^{(j)}\right) \quad (6.16)$$

$$p_b\left(\theta_k^{(i)}\right) \propto \sum_j p_d\left(\theta_{k+1}^{(j)}\right) p_b\left(\theta_{k+1}^{(j)}\right) p\left(\theta_{k+1}^{(j)}|\theta_k^{(i)}\right) \quad (6.17)$$

$$P_u(c_k) \propto \sum_j p_d\left(\theta_k^{(j)}\right) p_f\left(\theta_k^{(j)}\right) f_k\left(c_k, \theta_k^{(j)}\right) \quad (6.18)$$

The greater L is, the closer the approximation to the optimal solution is. However, obviously, if the number of levels is increased, also the computational complexity is also being augmented. According to [88] a number of levels equal to $8M$ is enough to have no performance loss when working with $M - PSK$ signals.

6.6. Tikhonov Parametrization

In [21], the Tikhonov parametrization was proposed in order to solve the problem of iterative decoding and detection we are dealing with. The Tikhonov parametrization solution shows a near optimum performance (very close to the known phase situation). This good performance is almost insensitive to the pilot symbols distribution. Besides, its computational cost is not that high.

The starting point of this algorithm is the approximation of the conditional probability $p(r_k | \theta_k)$ by the Gaussian pdf at minimum divergence: $g_{\mathbb{C}}(\alpha_k e^{j\theta_k}, 2\sigma^2 + \beta_k - |\alpha_k|^2, r_k)$, where

$$\alpha_k \triangleq \sum_{x \in \mathbb{N}} x P_d(c_k = x) \quad (6.19)$$

$$\beta_k \triangleq \sum_{x \in \mathbb{N}} |x|^2 P_d(c_k = x) \quad (6.20)$$

If the messages $P_d(c_k)$ were the exact a posteriori probabilities of the code symbols, we can use the above approach in the following way:

$$\begin{aligned} p_d(\theta_k) &\propto p(r_k | \theta_k) \\ &\simeq g_{\mathbb{C}}(\alpha_k e^{j\theta_k}, 2\sigma^2 + \beta_k - |\alpha_k|^2, r_k) \\ &\propto \exp\left(2 \frac{\text{Re}[r_k \alpha_k^* e^{j\theta_k}]}{2\sigma^2 + \beta_k - |\alpha_k|^2}\right) \end{aligned} \quad (6.21)$$

Using this approach, the rest of the messages can be calculated just substituting (6.21) in (6.12) and (6.11):

$$p_f(\theta_k) \simeq \int_0^{2\pi} \exp\left(2 \frac{\text{Re}[r_{k-1} \alpha_{k-1}^* e^{j\theta_{k-1}}]}{2\sigma^2 + \beta_{k-1} - |\alpha_{k-1}|^2}\right) p_f(\theta_{k-1}) p(\theta_k | \theta_{k-1}) d\theta_{k-1} \quad (6.22)$$

$$p_b(\theta_k) \simeq \int_0^{2\pi} \exp\left(2 \frac{\text{Re}[r_{k+1} \alpha_{k+1}^* e^{j\theta_{k+1}}]}{2\sigma^2 + \beta_{k+1} - |\alpha_{k+1}|^2}\right) p_b(\theta_{k+1}) p(\theta_{k+1} | \theta_k) d\theta_{k+1} \quad (6.23)$$

The above expressions can be calculated in an approximate way as the following sequence of Tikhonov pdfs,

$$p_f(\theta_k) \propto \exp(\text{Re}[a_{f,k} e^{-j\theta_k}]) \quad (6.24)$$

$$p_b(\theta_k) \propto \exp(\text{Re}[a_{b,k} e^{-j\theta_k}]) \quad (6.25)$$

Finally, substituting (6.24) and (6.25) into (6.14), the message sent from f_k to c_k is calculated using the following expression:

$$P_u(c_k) \propto \exp\left(-\frac{|c_k|^2}{2\sigma^2}\right) I_0\left(|a_{f,k} + a_{b,k} + \frac{r_k c_k^*}{\sigma^2}|\right) \quad (6.26)$$

where $a_{f,k}$ and $a_{b,k}$ are coefficients that can be calculated recursively. Their mathematical expressions depend on whether the phase is slowly-varying or rapidly-varying. If the phase is considered to be slowly-varying, then $p(\theta_k|\theta_{k-1}) = \delta(\theta_k - \theta_{k-1})$. In this case, $a_{f,k}$ and $a_{b,k}$ are calculated according to:

$$a_{f,k} = a_{f,k-1} + 2 \frac{r_{k-1} \alpha_{k-1}^*}{2\sigma^2 + \beta_{k-1} - |\alpha_{k-1}|^2} \quad (6.27)$$

$$a_{b,k} = a_{f,k+1} + 2 \frac{r_{k+1} \alpha_{k+1}^*}{2\sigma^2 + \beta_{k+1} - |\alpha_{k+1}|^2} \quad (6.28)$$

where the initial conditions for the above recursions are $a_{f,0} = 0$ in (6.27) and $a_{b,K-1} = 0$ in (6.28).

On the other hand, if the changes of the phase cannot be considered to be slowly-varying the recursion formulas need to be changed while the expressions for $p_f(\theta_k)$ and $p_b(\theta_k)$ still hold. These new recursion formulas are:

$$a_{f,k} = \left[a_{f,k-1} + 2 \frac{r_{k-1} \alpha_{k-1}^*}{2\sigma^2 + \beta_{k-1} - |\alpha_{k-1}|^2} \right] \times \gamma(\sigma_p^2, |a_{f,k-1} + 2 \frac{r_{k-1} \alpha_{k-1}^*}{2\sigma^2 + \beta_{k-1} - |\alpha_{k-1}|^2}|) \quad (6.29)$$

$$a_{b,k} = \left[a_{f,k+1} + 2 \frac{r_{k+1} \alpha_{k+1}^*}{2\sigma^2 + \beta_{k+1} - |\alpha_{k+1}|^2} \right] \times \gamma(\sigma_p^2, |a_{f,k+1} + 2 \frac{r_{k+1} \alpha_{k+1}^*}{2\sigma^2 + \beta_{k+1} - |\alpha_{k+1}|^2}|) \quad (6.30)$$

where

$$\gamma(x_1, x_2) = \frac{1}{1 + x_1 x_2} \quad (6.31)$$

In the above equation, x_1 and x_2 are real arguments. A detailed deduction of the previous expressions can be found in [21].

6.7. Gaussian Parametrization

The Gaussian Parametrization is also presented in [21] as a method that shows very good performance in certain conditions. Its low computational cost makes it an interesting option to perform detection and decoding. However, the performance of this solution is affected by the distribution of the pilots and it degrades when long sequences of pilots are used.

There are two basic ideas under this parametrization. The first one is to model the phasor process $h_k \triangleq e^{j\theta_k}$ as a complex circularly symmetric Gauss-Markov process. The second one consists of treating $\mathbf{h} = (h_0, \dots, h_{K-1})$ and \mathbf{r} as jointly Gaussian. This last assumption means that the minimum divergence parametrization can be used as it was done in the Tikhonov parametrization:

$$p(r_k|h_k) \simeq g_{\mathbb{C}}(\alpha_k h_k, 2\sigma^2 + \beta_k - |\alpha_k|^2, r_k)$$

where β_k and α_k are defined as in (6.19) and (6.20).

Take into account the two main assumptions described above, the following state and observation equation can be written:

■ *Forward process*

$$h_{k+1} = \rho h_k + v_k \quad v_k \sim \mathcal{N}_{\mathbb{C}}(0, 1 - \rho^2) \quad (6.32)$$

$$r_k = \alpha_k h_k + w_k \quad w_k \sim \mathcal{N}_{\mathbb{C}}(0, 2\sigma^2 + \beta_k + |\alpha_k|^2) \quad (6.33)$$

■ *Backward process*

$$h_{k-1} = \rho h_k + v'_k \quad v'_k \sim \mathcal{N}_{\mathbb{C}}(0, 1 - \rho^2) \quad (6.34)$$

$$r_k = \alpha_k h_k + w'_k \quad w'_k \sim \mathcal{N}_{\mathbb{C}}(0, 2\sigma^2 + \beta_k + |\alpha_k|^2) \quad (6.35)$$

where $\{w_k\}$, $\{w'_k\}$, $\{v_k\}$ and $\{v'_k\}$ are independent Gaussian processes with independent components.

Using the above model, the probability $p(h_k|\{r_j : j \neq k\}) = g_{\mathbb{C}}(m_k, \Sigma_k, h_k)$ where the conditional mean and mode can be calculated iteratively using a Kalman Smoother in the backward and forward processes [57], [15], [57]:

$$m_k = \frac{\Xi_{k|k+1}}{\Sigma_{k|k-1} + \Xi_{k|k+1}} m_{k|k-1} + \frac{\Sigma_{k|k-1}}{\Sigma_{k|k-1} + \Xi_{k|k+1}} \mu_{k|k+1} \quad (6.36)$$

$$\Sigma_k = \frac{\Sigma_{k|k-1} \Xi_{k|k+1}}{\Sigma_{k|k-1} + \Xi_{k|k+1}} \quad (6.37)$$

where $m_{k|k-1}$ and $\Sigma_{k|k-1}$ are the conditional mean and variance of h_k given $\{r_j : j = 0, \dots, k-1\}$, i.e. forward prediction. On the other hand, $\mu_{k|k+1}$ and $\Xi_{k|k+1}$ are the condition mean and variance of h_k given $\{r_j : j = k+1, \dots, K-1\}$, i.e. backward prediction.

In this point, the calculation of the probability of the code symbol given the observation and the phasor estimate $h_k \sim \mathcal{N}_{\mathbb{C}}(m_k, \Sigma_k)$ is done as:

$$\begin{aligned} P_u(c_k) &\propto \int \exp\left(-\frac{1}{2\sigma^2}|r_k - c_k e^{j\theta_k}|^2\right) g_{\mathbb{C}}(m_k, \Sigma_k, R_k e^{j\theta_k}) R_k dR_k d\theta_k \\ &\propto e^{-\frac{|c_k|^2}{2\sigma^2}} \int_0^\infty e^{-z} I_0\left(\sqrt{\frac{|r_k c_k^*|^2}{\sigma^4} + 4\frac{z|m_k|^2}{\Sigma_k} + 4\frac{\sqrt{z}}{\sqrt{\Sigma_k}\sigma^2} \text{Re}\{r_k c_k^* m_k^*\}}\right) dz \end{aligned} \quad (6.38)$$

where $h_k = R_k e^{j\theta_k}$. The latter integral can be solved using any numerical integration methods.

6.8. First Particle Filtering approach

In this section the Particle Filtering method proposed in [23] is introduced. The objective of the authors in the paper is to use Particle Filtering Methods in order to locate the argmax unimodal . The distribution to deal with is $p_d(\theta_k)$. Similarly to the other Particle Filtering Methods presented in this chapter, the set used is represented by:

$$\{\theta_k^{(i)}, \omega_k^{(i)}\}_{i=1}^M$$

where $k = 1, \dots, K$ and K is the length of the code. M is the number of particles used. According to this algorithm, the process to follow is:

- **Initialization** The first step is to initialize the set of samples. There is an initial list of samples that is the same for each time instant. This initial list is generated taking M/L samples from each $p_d(\theta_k)$.

In this algorithm, the authors introduce two types of weights, the ones calculated in a forward process $\omega_{k,FW}^{(i)}$ and the ones calculated in the backward process $\omega_{k,BW}^{(i)}$. In the previous expressions, the subindex k is the time instant, i indicates the sample, FW means forward and BW backward. In the initialization step, the weights $\omega_{1,FW}^{(i)}$ and $\omega_{L,BW}^{(i)}$ are set equal to one.

- **Update of the Weights** The updating process is done according to:

$$\tilde{\omega}_{k,FW}^{(i)} = \tilde{\omega}_{k-1,FW}^{(i)} p_d(\theta_{k-1}^{(i)}) \quad (6.39)$$

$$\tilde{\omega}_{k,BW}^{(i)} = \tilde{\omega}_{k+1,BW}^{(i)} p_d(\theta_{k+1}^{(i)}) \quad (6.40)$$

- **Normalization of the Weights** The next step is to normalize the weights.

$$\omega_{k,FW}^{(i)} = \frac{\tilde{\omega}_{k,FW}^{(i)}}{\sum_i \tilde{\omega}_{k,FW}^{(i)}} \quad (6.41)$$

$$\omega_{k,BW}^{(i)} = \frac{\tilde{\omega}_{k,BW}^{(i)}}{\sum_i \tilde{\omega}_{k,BW}^{(i)}} \quad (6.42)$$

- **Calculation and normalization of the final weights** The final weights are calculated using the forward and backward weights as:

$$\tilde{\omega}_k^{(i)} = \tilde{\omega}_{k,BW}^{(i)} \tilde{\omega}_{k,FW}^{(i)} \quad (6.43)$$

$$\omega_k^{(i)} = \frac{\tilde{\omega}_k^{(i)}}{\sum_i \tilde{\omega}_k^{(i)}} \quad (6.44)$$

- **Updating the list of samples** The list of samples is updated according to the following expression:

$$\theta_k^{(i)new} = \arg \left[(1 - \varepsilon) e^{j\theta_k^{(i)old}} + \varepsilon e^{j\hat{\theta}_k} \right] \quad (6.45)$$

where ε is a parameter with an arbitrary value between zero and one. $\hat{\theta}_k$ is the phase estimation and it is calculated as:

$$\hat{\theta}_k = \arg \sum_i \omega_k^{(i)} e^{j\theta_k^{(i)old}} \quad (6.46)$$

The previous process is an iterative one and it can be repeated as many times as one wishes.

The main problem of this algorithm is related to the number of particles that has to be used. According to the formulation of the process, the number of particles used has to be at least equal to the length of the code or a multiple of it. So, when we are working with a high number of particles, the complexity of this algorithm becomes prohibitive. In the following sections we propose a different particle filtering algorithm that tries to solve this problem maintaining the performance of this method.

6.9. Second Particle Filtering approach

In this section it is shown how Particle Filtering Techniques can be used to discretize the phase space. This discretization is expected to be better than the equally spaced proposed in the previous section since the particles are supposed to be distributed around the real value of the phase noise.

In Particle Filtering Theory, the distributions are approximated by discrete random measures defined by particles and weights assigned to the particles. Suppose the distribution of interest were $p(\theta_k)$ and its approximating random measure:

$$\{\theta_k^{(i)}, \omega_k^{(i)}\}_{i=1}^M$$

where $\theta_k^{(i)}$ are the particles, $\omega_k^{(i)}$ are their weights and M is the number of particles. Then, this distribution could be calculated as the following approximation:

$$p(\theta_k) \approx \sum \omega_k^{(i)} \delta(\theta_k - \theta_k^{(i)})$$

The next step in Particle Filtering is the Importance Sampling. If we want to approximate the previous distribution with a discrete random measure, we need to generate the particles from it. When direct sampling from the distribution is not possible, one can generate the particles $\theta_k^{(i)}$ from a distribution $\pi(\theta_k)$, known also as Importance Function and assign weights according to:

$$\omega_k^{*(i)} = \frac{p(\theta_k)}{\pi(\theta_k)}$$

These weights are not normalized, in order to normalize them, they have to be divided by the summation $\sum \omega_k^{*(i)}$.

Suppose that the set $\{\theta_{k-1}^{(i)}, \omega_{k-1}^{(i)}\}_{i=1}^M$ approximates the posterior distribution $\pi(\theta_{0:k-1}^{(i)} | r_{0:k-1})$. Sequential Sampling methods are used to obtain the set $\{\theta_k^{(i)}, \omega_k^{(i)}\}_{i=1}^M$ generating particles $\theta_k^{(i)}$ taking into account the observation at time k and adding them to the set of particles for time $k - 1$. If the Importance Function can be factored as,

$$\pi(\theta_{0:k} | r_{0:k}) = \pi(\theta_k | \theta_{0:k-1}, r_{0:k}) \pi(\theta_{0:k-1} | r_{0:k-1}) \quad (6.47)$$

and if:

$$\theta_{0:k-1}^{(i)} \sim \pi(\theta_{0:k-1} | r_{0:k-1}) \quad (6.48)$$

the weights at time $k - 1$ are calculated as

$$\omega_{k-1}^{(i)} \propto \frac{p(\theta_{0:k-1}^{(i)} | r_{0:k-1})}{\pi(\theta_{0:k-1}^{(i)} | r_{0:k-1})} \quad (6.49)$$

Then, the new particles at time k are obtained using:

$$\theta_k^{(i)} \sim \pi(\theta_k | \theta_{0:k-1}^{(i)}, r_{0:k}) \quad (6.50)$$

and finally, the weights are updated in the following way:

$$\omega_k^{(i)} \propto \frac{p(y_k | \theta_k^{(i)}) p(\theta_k^{(i)} | \theta_{k-1}^{(i)})}{\pi(\theta_k^{(i)} | \theta_{0:k-1}^{(i)}, r_{0:k})} \omega_{k-1}^{(i)} \quad (6.51)$$

If we apply all the previous Particle Filtering formalism to the proposed problem the integrals corresponding to the messages sent given by the expressions (6.12), (6.11) and (6.14) can be calculated numerically. This can be done using, for instance, the parallelogram rule.

However, there are some problems related to the use of this approach. The behavior of the integrals and the functions involved with them is not the expected one and the algorithm degrades if the noise added to the carrier phase increases. Another drawback is the computational complexity associated to the integrals calculations. This makes the final algorithm very slow, and thus, less useful.

6.10. Third Particle Filtering approach

Due to the problems found trying to solve numerically the integrals, we propose another approach using Particle Filtering in order to avoid the integral calculations. Instead of using all the particles and try to calculate the integrals, we propose to use the set to estimate the phase and introduce afterward this estimation in the Sum-Product Algorithm.

Thus, the first step is to use Particle Filtering to estimate the phase. For simplicity the Importance Function is selected as the prior distribution. Then, the set of particles are generated according to:

$$\pi(\theta_{0:k}|r_{0:k}) = p(\theta_{0:k}|r_{0:k}) \quad (6.52)$$

And according to (6.51), taking into account (6.52) the update of the particles is done

$$\omega_k^{(i)} \propto p(r_k|\theta_k^{(i)})\omega_{k-1}^{(i)} \quad (6.53)$$

The previous weight update process can be rewritten using the Factor Graph messages as:

$$\omega_k^{(i)} \propto p_d(\theta_k^{(i)})\omega_{k-1}^{(i)} \quad (6.54)$$

Once the weights and the samples have been obtained, the phase estimation for each time instant can be calculated as:

$$\hat{\theta}_k = \arg \sum_i \omega_k^{(i)} e^{j\theta_k^{(i)}} \quad (6.55)$$

Finally the messages needed to perform decoding according to the Sum-Product Algorithm are ready to be calculated using the phase estimations for each time instant,

$$p_b(\hat{\theta}_k) \propto p_b(\hat{\theta}_{k+1})p_d(\hat{\theta}_{k+1})p(\hat{\theta}_{k+1}|\hat{\theta}_k) \quad (6.56)$$

$$p_f(\hat{\theta}_k) \propto p_f(\hat{\theta}_{k-1})p_d(\hat{\theta}_{k-1})p(\hat{\theta}_k|\hat{\theta}_{k-1}) \quad (6.57)$$

$$p_u(c_k) \propto p_f(\hat{\theta}_k)p_d(\hat{\theta}_k)f_k(\hat{\theta}_k, c_k) \quad (6.58)$$

The above Particle Filtering procedure suffers from particle impoverishment, that is, after several recursions, the weights associated to some samples become negligible and stop contributing to the overall evaluation. To prevent it a residue resampling step is inserted.

As it can be seen, no restriction related to the number of particles used is imposed in this algorithm. Thus, it can be used for any code, no matter the length of it yet maintaining the same performance.

6.11. Simulation results

In this section the bit error rate (*BER*) performance of the proposed algorithm is studied through experiments versus E_b/N_0 . In all the experiments, the considered code is a (3,6)-regular LDPC code with codewords of length 4000 found in [68]. The chosen modulations are *QPSK* and *BPSK*.

Pilot symbols are inserted in the transmitted codeword in order to make the iterative decoding bootstrap. Different symbol distributions have been employed in the simulations, all of them having the same transmission rate.

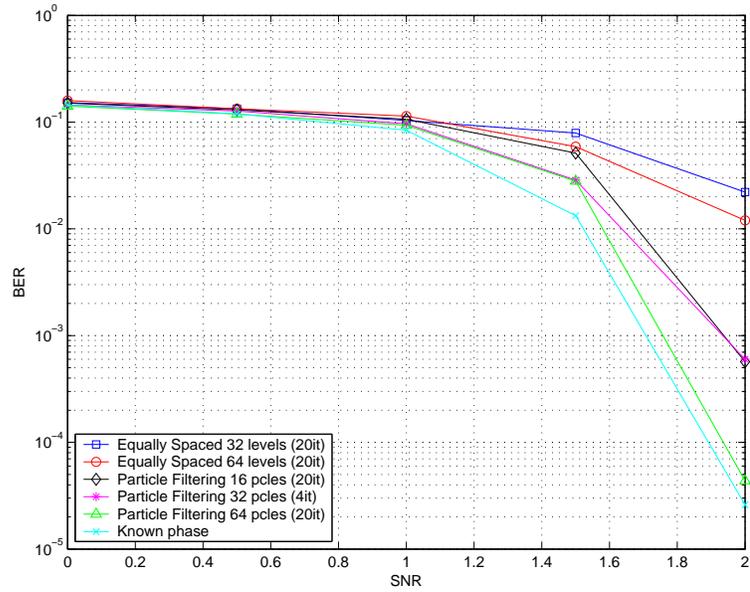
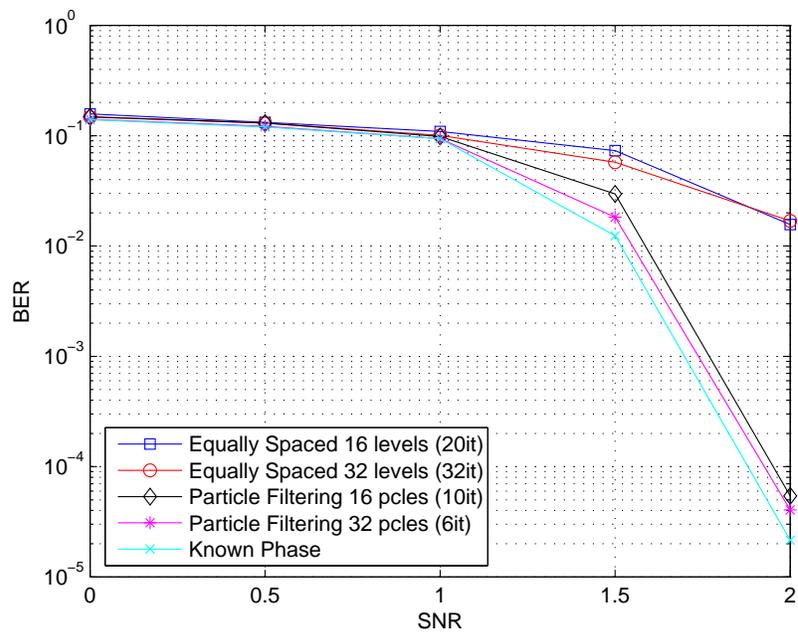
As for the phase model, we have used a Wiener model with $\sigma_v = 0,01$ and $\sigma_w = 0,005$.

In the particle filtering algorithm sets of 16, 32, 64 and 100 particles and the mode as estimator have been used. The mean has not been used since the results obtained with this estimator are worse than the results with the mode [79]. Furthermore, a resampling step is introduced in these algorithms in order to prevent from particle degeneration.

In the equally spaced discretization the number of levels used is $L = 16$, $L = 32$ and $L = 64$, the chosen number of levels is justified in [69] where the authors state that for $M - PSK$ signals, $L = 8M$ values are sufficient to have no performance loss. This previous algorithm can be regarded as a “practically optimal” benchmark when an appropriate number of levels is used.

In both methods, the total number of iterations in the decoder is 20. From now on, when we talk about number of iterations, we will mean number of global iterations in the graph. Thus, the computational complexity of the Particle Filtering algorithm is proportional to the product of the number of particles and the number of global iterations. On the other hand, the computational complexity of the Equally Spaced Discretization is proportional to the product of the number of global iterations and the number of discrete levels to the second.

In Figure 6.3 the performance of the proposed Particle Filtering algorithm is compared with the performance of the equally space discretization method

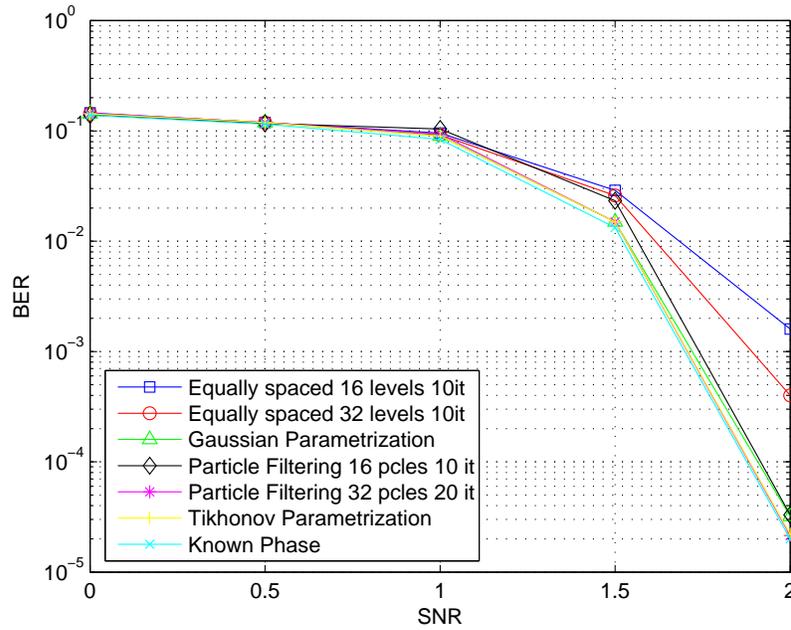
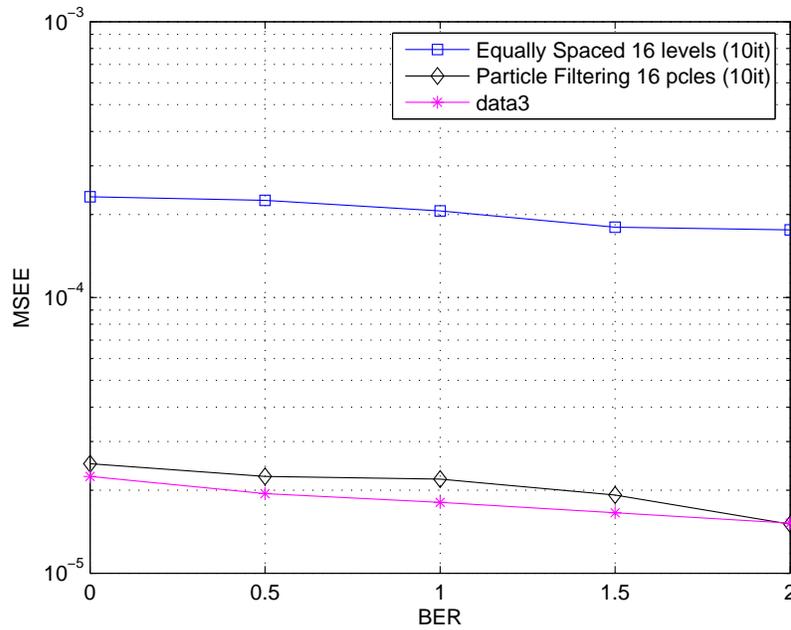
Figure 6.3: QPSK modulation and $\sigma_v = 0,01$ modelFigure 6.4: BPSK modulation and $\sigma_v = 0,01$

assuming 20 pilot symbols in every block of 400 transmitted symbols. The modulation used is *QPSK* and the phase noise model with $\sigma_v = 0,01$ has been considered. In the Equally Spaced method 32 and 64 discretization levels have been used. The number of global iterations in both equally spaced discretization methods is 20. The number of global iterations is 20 when the number of particles is 16, 4 when we use 32 particles and 20 when we use 64 particles. As it can be seen, the Particle Filtering algorithm performs better than the Equally spaced and its computational complexity is lower.

In Figure 6.4 the performance of the algorithms presented in Sections 6.10 and 6.5 are compared when a modulation format characterized by a less dense constellation: BPSK is used. In this case, it is expected to obtain better or similar performance with less computational complexity. This is the reason why the number of levels in the Equally Spaced algorithm are 16 and 32. And the number of particles also is 16 and 32. It is, again, assumed a pilot scheme characterized by 20 pilot symbols sent in every block of 400 transmitted symbols. The phase noise model is supposed to be characterized by $\sigma_v = 0,01$. As it can be seen, the performance of the proposed Particle Filtering algorithm is much better than the performance of the algorithm based on a Equally Spaced discretization. Besides, the computational complexity of the Particle Filtering algorithm is less than the one of the Equally Spaced discretization.

In Figure (6.5) the proposed Particle Filtering algorithm is again compared with the Equally Spaced discretization algorithm. In this simulation, the modulation format is still BPSK and the pilot scheme is composed again by 20 known symbols followed by 400 unknown symbols. In this case, the $\sigma_v = 0,005$ model has been considered for the phase noise process. In this Figure, the performance of two other algorithms described in the literature is also shown for the shake of comparison. Both of them were proposed in [21] and are based on the canonical distribution approach proposed in [88]. The first one is based on a Tikhonov parametrization. The second one is based on a Gaussian parametrization. As it can be seen, the Particle Filtering algorithm, the Tikhonov parametrization and the Gaussian parametrization exhibit a practically optimal performance, very close to the known phase curve. On the other hand, the performance of the Equally Spaced algorithm is worse than the performance of any other employed algorithm and its computational complexity is higher.

In Figure 6.6 we represent the mean squared error of the estimation of the phase in the phase noise process with $\sigma_v = 0,01$ and when BPSK modulation is considered. As it was expected, the Particle Filtering algorithm performs better than the one based on an Equally Spaced discretization of the phase space.

Figure 6.5: BPSK modulation $\sigma_v = 0,005$ modelFigure 6.6: Mean Squared Error in the phase estimation for BPSK modulation and $\sigma_v = 0,01$

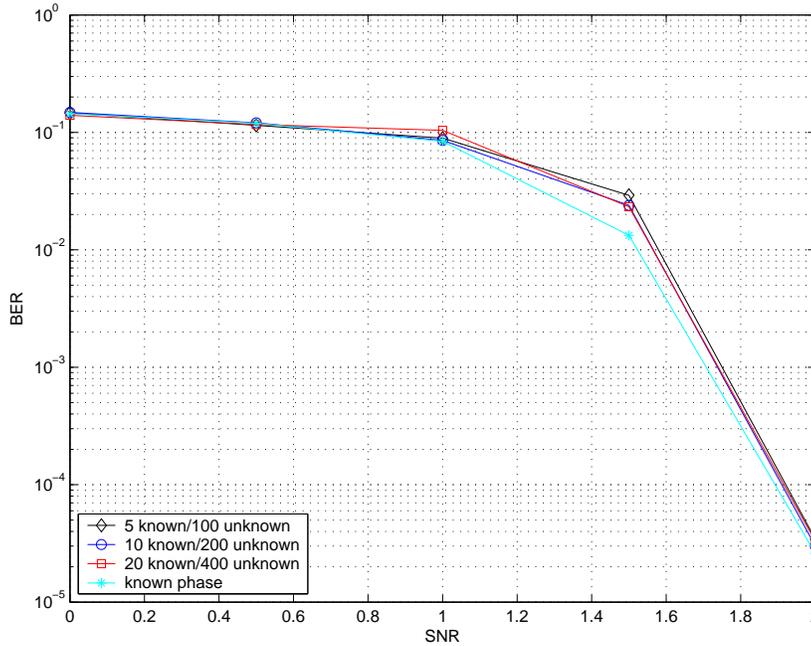


Figure 6.7: Performance of the Particle Filtering algorithm with different pilot distributions

Finally, we present a study on the sensitivity to distributions of pilot symbols of the proposed Particle Filtering Algorithm. Different distributions have been considered, a first one in which 20 known symbols are followed by 400 unknown symbols, a second one in which 10 known symbols are followed by 200 unknown symbols and, finally, a third distribution in which 5 known symbols are followed by 100 unknown symbols. Thus, the transmission rate has been maintained constant. The number of particles used has been 16. In Figure 6.7 we present the results for the case in which BPSK modulation has been used and the $\sigma_v = 0,005$ model has been considered for the phase noise. We can observe that the proposed Particle Filtering algorithm is almost insensitive to the pilot distribution scheme. However, we have noticed through simulations that the Particle Filtering algorithm degrades when it is used with pilot distributions in which only one pilot is followed by a number of known symbols. In Figure 6.8 we show the squared error in the estimation of the phase for the Particle Filtering Algorithm when different pilot distributions are used. It can be observed that no significant performance loss appears in the phase estimation depending on the pilot distribution.

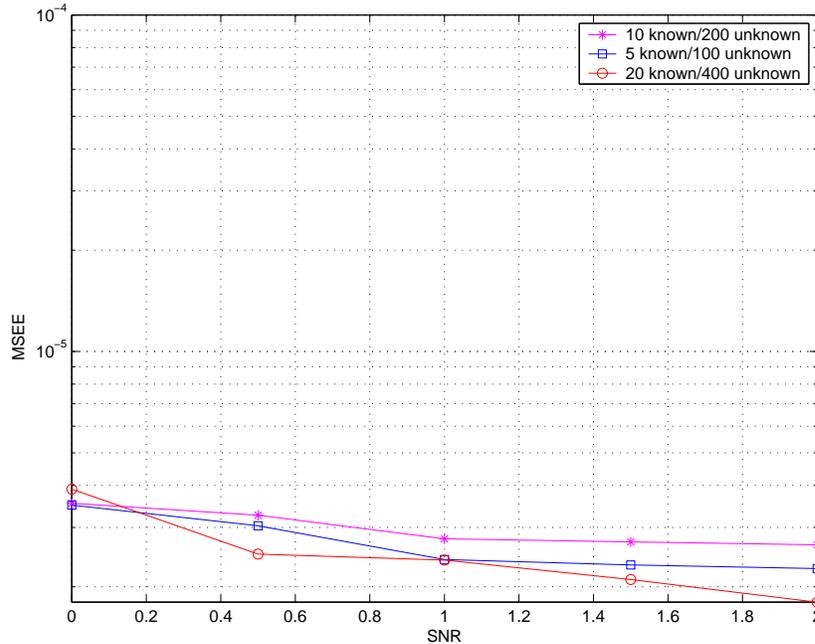


Figure 6.8: Squared error in the phase estimation with the Particle Filtering algorithm using different pilot distributions

6.12. Conclusions

In this chapter, the problem of joint iterative decoding of LDPC codes and phase estimation when the symbols are transmitted over a channel affected by phase noise, has been considered. A Particle Filtering algorithm have been proposed to solve the integrals that appear when the Sum-Product Algorithm is applied to the Factor Graph representation of the system. The use of the proposed Particle Filtering algorithm has the next advantages:

- In contrast to other Particle Filtering solutions previously proposed, the proposed algorithm is not limited by the length of the code employed. For the First Particle Filtering method, the number of particles used has to be a multiplex of the length of the code. This is a very important limitation since if we were interested in using this algorithm with the codes employed in the simulations, we would need to use at least 4000 particles. With the proposed Particle Filtering it can be seen that a number of particles equal to 100 is enough to obtain a good performance.
- Another advantage is the fact that the computational complexity of

the proposed algorithm is lower since it avoids the exploration of the whole trellis in contrast to what the Equally spaced algorithm does. The Equally spaced solution is equivalent to a Viterbi algorithm so its complexity is $L \times L$. Thus, the total complexity of the Equally Spaced method is $4 \times 32 \times 32$ when we use 32 levels and $20 \times 64 \times 64$ when we use 64 levels. The numbers multiplying the quantity $L \times L$ are the number of iterations in the Sum-Product algorithm. On the other hand, the computational complexity of the proposed Particle Filtering method is proportional to the number of particles times the number of iterations. As it can be noticed, the complexity of the proposed Particle Filtering method is less than the one of the Equally Spaced discretization methods in all the simulations that have been done.

- The proposed Particle Filtering method has also been compared to other algorithms proposed in the literature [21]. These algorithms have shown a very good performance and a small computational complexity, however, they do not provide with a way to estimate the phase. The proposed Particle Filtering algorithm has shown similar performance and it is also able to estimate the phase.
- As for the estimation of the estimation of the phase, the proposed algorithm performs much better than the Equally spaced approach.
- Finally, it has been shown through the presented simulations that the proposed Particle Filtering algorithm is almost insensitive to the pilot distribution scheme in contrast to the behavior of the Gaussian parametrization [21] or the Equally spaced discretization.

6.13. Future Work

The following objective to pursue in this research topic is the application of the proposed algorithm in situations where the phase noise changes faster. The model used for the phase changes is suitable only for small changes is we want to consider bigger changes we need to take more elements from the series expansion of the exponential. We think the resulting algorithm will perform at least as well as the analyzed in this chapter.

Parte II

Algoritmos Bayesianos Variacionales aplicados a Genómica

Capítulo 7

Selección de variable mediante algoritmos Bayesianos variacionales

En el proceso de selección de variables se aborda el problema de selección del conjunto de variables de entrada que mejor predicen un conjunto dado de observaciones. Su aplicación se encuentra en distintos campos tales como “machine learning”, clasificación supervisada o no supervisada, regresión, predicción de series temporales, etc. Recientemente, debido a la gran proliferación de bases de datos con gran cantidad de información, la selección de variables se ha convertido en un área de investigación muy relevante dada además, la cantidad de campos de aplicación: todos aquellos en los que se cuenta con decenas o centenares de variables. De entre estos campos cabe destacar dos: el *procesamiento de textos* (especialmente su aplicación a documentos de Internet) y el de la *Genómica*, relacionado con los datos de expresión de genes. En concreto en este último campo se trata de realizar un proceso de selección de genes dado un conjunto de datos de microarray.

Las variables en este caso son las concentraciones de ARN mensajero (expresión de los genes). El número de muestras con el que se cuenta (número de pacientes) es mucho menor que el número de datos, siendo esta relación de 1 a 3, en el mejor de los casos. Una aplicación a este campo puede ser por ejemplo la clasificación de pacientes de cáncer en función del perfil de expresión de sus genes.

Los objetivos de la selección de variables son:

- Mejorar el rendimiento del proceso de predicción.
- Proporcionar métodos de predicción más rápidos.
- Proporcionar una mejor comprensión del proceso a través del cual se generan los datos.

Los problemas de selección de variables están relacionados con problemas de reducción de la dimensión del conjunto de entrada así como la disminución del conjunto de parámetros del modelo. Todos estos problemas giran en torno al problema más general de selección de modelo. Sin embargo, la selección de variables tiene sus propios retos tanto desde el punto de vista teórico como práctico.

Como ya se ha comentado, la selección de variables es un método que se aplica a situaciones en las que se pretende modelar relaciones entre una determinada salida y un conjunto de variables que caracterizan el modelo cuando existe indeterminación sobre qué conjunto debe usarse. Los métodos de selección de variables pueden ayudar a estimar la importancia de distintos predictores, mejorar la exactitud de la predicción así como reducir el coste computacional en la obtención de datos. La aplicación de métodos Bayesianos a la selección de variables fue propuesta por George y McCulloch (JASA,1993). Brown, Vannucci and Fearn (1998, JRSSB) generalizaron la aproximación anterior al caso en que se trabaja con distintas variables. La idea clave del modelo es usar un vector de variables latentes para caracterizar los diferentes subconjuntos de variables (modelos). Las distribuciones a priori se introducen en los parámetros de regresión así como en el conjunto de posibles modelos. Finalmente la selección se basa en las probabilidades a posteriori del modelo que se obtienen en principio utilizando el teorema de Bayes. Cuando el número de posibles modelos es muy grande (con p predictores tendríamos 2^p posibles subconjuntos) se utilizan técnicas de Markov Chain Monte Carlo (MCMC) para realizar una búsqueda estocástica de modelos con probabilidad a posteriori alta.

Este capítulo se estructura de la siguiente forma: en la Sección 7.1 se plantea el problema a tratar. La Sección 7.2 está dedicada al aprendizaje variacional Bayesiano. En la Sección 7.3 se particulariza este proceso de

aprendizaje introduciendo el algoritmo VBEM (Variational Bayesian Expectation-Maximization) propuesto en esta tesis. En la Sección 7.5 se introduce una modificación al algoritmo VBEM para permitir la inclusión de información a priori a través de ligaduras. En la Sección 7.6 se ejemplifica esta introducción de ligaduras en un espacio monodimensional para pasar después al espacio multidimensional en la Sección 7.7. En la Sección 7.8 se desarrolla un algoritmo PDA (Probabilistic Data Association) que aún se encuentra bajo desarrollo y que pretende aplicarse a la solución de tipo VBEM presentada para mejorar el proceso de detección. Finalmente, en la Sección 7.9 se presentan los resultados de las simulaciones y las conclusiones más importantes se encuentran en la Sección 7.10.

7.1. Formulación del Problema

En este capítulo se estudian sistemas caracterizados por la siguiente ecuación:

$$\mathbf{y}_i = \mathbf{R}\mathbf{W}_i\mathbf{b}_i + \mathbf{e}_i \quad (7.1)$$

donde $\mathbf{R} = \mathbf{T}\mathbf{Y}^\top$ con $\mathbf{T} = \begin{pmatrix} 1 & & 0 \\ & \ddots & \vdots \\ & & 1 & 0 \end{pmatrix}$, $\mathbf{y}_i = [y_i(1), \dots, y_i(N)]^\top$,

$\mathbf{W}_i = \text{diag}\{\mathbf{w}_i\}$, $\mathbf{b}_i = [b_i(1), \dots, b_i(G)]^\top$ y $\mathbf{e}_i = [e_i(1), e_i(2), \dots, e_i(N)]^\top$.

El vector \mathbf{y}_i contiene N muestras que constituyen las observaciones de la variable \mathbf{b}_i en cada uno de los instantes de tiempo en los que se muestrea. El conjunto de todas las observaciones se agrupa en la matriz $\mathbf{Y} \in \mathcal{R}^{G \times (N+1)}$. El vector \mathbf{b}_i constituye el conjunto de las llamadas variables ocultas ya que no se observan directamente aunque interactúan con los parámetros del sistema para proporcionar el conjunto de observaciones. Finalmente, se consideran además dos tipos de parámetros: los elementos de la matriz diagonal \mathbf{W}_i y la varianza del ruido.

Dado el conjunto de observaciones \mathbf{Y} , el objetivo es determinar \mathbf{b}_i para todo i . Además es muy importante hacer notar que tanto \mathbf{W}_i como σ_i^2 son desconocidos.

Al fijarse detenidamente en la expresión (7.1), puede verse que el modelo de este problema tiene exactamente la misma estructura que las que se encuentran en detección multiusuario y es esto precisamente lo que permite aplicar todo el conocimiento anterior relacionado con detección multiusuario a este nuevo problema. Por tanto, el objetivo ahora es desarrollar un algoritmo de tipo Bayesiano de forma que se logre una solución general y eficiente al problema que pretende tratarse.

En este proceso de búsqueda de la solución hay que tener en cuenta que tanto el canal, o lo que es lo mismo en este caso, los elementos de la matriz diagonal \mathbf{W}_i , y la varianza del ruido son desconocidas. Por lo tanto, hay que introducir un paso de estimación de los mismos dentro de la solución Bayesiana general. Otro problema importante al que hay que hacer frente es el relacionado con el tamaño de las observaciones, ya que se pretende desarrollar un algoritmo con buen rendimiento al aplicarlo tanto en casos sobredeterminados como indeterminados, esto es, tanto cuando el tamaño del conjunto de muestras N es mucho mayor que G como a la inversa. En este capítulo se pretende desarrollar un algoritmo bajo un marco Bayesiano que pueda hacer frente a estos dos problemas mediante una solución variacional basada en el algoritmo EM (Expectation Maximization). Las soluciones Variacionales Bayesianas basadas en el algoritmo EM, han demostrado poseer menor carga computacional que otras técnicas Bayesianas como por ejemplo las basadas en muestreo Monte Carlo y similar comportamiento. Una solución de este tipo fue propuesta en [12], esta solución no considera explícitamente la incertidumbre en el modelado de las variables \mathbf{b}_i . El problema es formulado como una estimación de parámetros y la los vectores \mathbf{b}_i se determina midiendo la cercanía de los elementos de \mathbf{W}_i a cero. Aunque este tratamiento puede reducir la complejidad del proceso de aprendizaje, no refleja fielmente la naturaleza del problema. Esto motiva la investigación de una solución en la que se realicen de forma conjunta el aprendizaje de las variables \mathbf{b}_i y la estimación de los parámetros del modelo.

7.2. Aprendizaje Variacional Bayesiano

7.2.1. Ideas Generales

En este capítulo se va a desarrollar un algoritmo que se aplicará en el siguiente capítulo al aprendizaje de la topología y a la estimación de parámetros conjuntas en una red de genes. En la mayoría de los casos, los fenómenos que tratan de estudiarse no se comprenden con suficiente profundidad aún por lo que el proceso de creación de un modelo que pueda reproducir el fenómeno de forma precisa es complicado. Por tanto, puede considerarse que los modelos que aquí se presentan para la interpretación de los datos, son simplemente aproximaciones más o menos simplificadas del fenómeno a estudiar.

En las técnicas de tipo “machine learning”, el modelado de datos se realiza mediante la introducción de un grupo de parámetros del modelo que hay que estimar. Las observaciones o datos se representan a través de la variable \mathbf{y} . Los parámetros pueden ser continuos o discretos y se representarán

a través la variable θ . Dado un modelo, el objetivo es encontrar el grupo de parámetros óptimo en el sentido de que se trate del grupo que mejor explique las observaciones o que mejor se ajuste a éstas. A este proceso se le llama **aprendizaje del modelo**. El proceso de aprendizaje puede realizarse haciendo uso de diversas técnicas como por ejemplo las de Máxima Verosimilitud (MV) o de Máxima Probabilidad a Posteriori (MAP). Otra posibilidad es utilizar métodos Bayesianos donde la idea es tratar todo aquello que se desconoce como una variable aleatoria y utilizar las reglas de la probabilidad para manipular estas variables desconocidas.

Además de las observaciones y los parámetros, el modelo puede incluir variables escondidas o latentes que no se observan aunque influyen junto con los parámetros a la hora de generar las observaciones. Estas variables se representan por \mathbf{b} en nuestro problema.

La selección de modelos es un paso esencial en el proceso de entendimiento y representación de las observaciones. Este proceso incluye tareas como:

- Aprendizaje de estructuras: qué variables están relacionadas entre sí.
- Selección de variables: qué variables de entrada son necesarias para predecir las variables de salida.
- Selección de la cardinalidad de las variables latentes cuando éstas son discretas.
- Inferencia de la dimensión del vector de variables latentes.

Los objetivos en el proceso de aprendizaje Bayesiano son los siguientes:

1. aproximar la verosimilitud marginal $p(\mathbf{y}|m)$ para poder realizar comparaciones entre los distintos modelos,
2. aproximar la distribución a posteriori $p(\theta|\mathbf{y}, \mathbf{m})$, que luego se puede usar par realizar predicción.

El proceso de aprendizaje Bayesiano comienza con el establecimiento del conocimiento inicial que se plasma a través de la probabilidad a priori del modelo, $p(m)$. Cada modelo tiene un conjunto de parámetros a los que se le asocia también una probabilidad a priori $p(\theta|\mathbf{m})$

El objetivo del proceso de selección Bayesiana de modelos es calcular la siguiente distribución de probabilidad a posteriori:

$$p(m|\mathbf{y}) = \frac{p(m)p(\mathbf{y}|m)}{p(\mathbf{y})} \quad (7.2)$$

donde la verosimilitud marginal $p(\mathbf{y}|m)$ se calcula teniendo en cuenta los parámetros del modelo de la siguiente forma:

$$p(\mathbf{y}|m) = \int d\boldsymbol{\theta} \mathbf{p}(\boldsymbol{\theta}|\mathbf{m}) \mathbf{p}(\mathbf{y}|\boldsymbol{\theta}, \mathbf{m}) \quad (7.3)$$

La verosimilitud anterior puede usarse para calcular a su vez la probabilidad a posteriori de los parámetros dada la observación y el modelo de la siguiente manera:

$$p(\boldsymbol{\theta}|\mathbf{y}, \mathbf{m}) = \frac{\mathbf{p}(\boldsymbol{\theta}|\mathbf{m}) \mathbf{p}(\mathbf{y}|\boldsymbol{\theta}, \mathbf{m})}{\mathbf{p}(\mathbf{y}|\mathbf{m})} \quad (7.4)$$

Los pasos anteriores necesitan del cálculo de la verosimilitud marginal. El problema que se encuentra tanto en la resolución de esta integral como en el de otras como por ejemplo las relacionadas con la predicción, es que no pueden calcularse analíticamente por lo que hay que utilizar una aproximación. La forma más sencilla de aproximar las integrales es evaluar el integrando utilizando un estimador para los parámetros como el Máxima Verosimilitud o el de Máxima Probabilidad a Posteriori. Otra posibilidad es estimar la integral numéricamente evaluando el integrando en varios valores de $\boldsymbol{\theta}$ utilizando métodos Monte Carlo. Si el número de muestras usadas tiende a infinito, este método proporciona un resultado exacto. Sin embargo, la mayor dificultad de este tipo de métodos es su gran complejidad computacional que puede llegar a ser prohibitiva. La tercera posibilidad para aproximar la integral es el uso de *Métodos Variacionales*. La idea clave de los Métodos Variacionales es la aproximación de la integral con una forma más simple que sí es tratable matemáticamente mediante el cálculo de un límite inferior o superior. Entonces, la integración se traduce en un problema más simple a la hora de implementarlo: la optimización de un límite haciendo este límite lo más cercano posible al valor real.

7.2.2. Solución Óptima

En este capítulo, el principal objetivo que se persigue es la detección de los valores de la variable oculta \mathbf{b} . En esta sección se hace uso de las ideas sobre los algoritmos variacionales presentadas en el apartado anterior. Para ello, comenzaremos con la derivación de la solución óptima para este equivalente del problema de detección multiusuario. La solución óptima usando un criterio de tipo MAP puede escribirse de la siguiente forma:

$$\hat{b}_i(l) = \operatorname{argmax}_{b_i(l)} p(b_i(l)|\mathbf{y}_i) \quad l = 1, \dots, G \quad (7.5)$$

donde $p(b_i(l)|\mathbf{y}_i) \propto p(\mathbf{y}_i|b_i(l))p(b_i(l))$ y $p(\mathbf{y}_i|b_i(l))$ es la función verosimilitud obtenida usando la regla Bayesiana:

$$p(\mathbf{y}_i|b_i(l)) = \sum_{(\mathbf{b}_i)_{-l}} p(\mathbf{y}_i|\mathbf{b}_i)p((\mathbf{b}_i)_{-l}) \quad (7.6)$$

y $\sum_{(\mathbf{b}_i)_{-l}}$ significa suma sobre todas las posibles configuraciones de \mathbf{b}_i excepto $\mathbf{b}_i(l)$ y $(\mathbf{b}_i)_{-l}$ representa un vector de tamaño $(G-1) \times 1$ que contiene todas las variable latentes excepto $b_i(l)$.

Además, la verosimilitud marginal $p(\mathbf{y}_i|\mathbf{b}_i)$ se obtiene integrando en los parámetros desconocidos la función de verosimilitud completa:

$$p(\mathbf{y}_i|\mathbf{b}_i) = \iint p(\mathbf{y}_i|\mathbf{w}_i, \sigma_i^2, \mathbf{b}_i)p(\mathbf{w}_i, \sigma_i^2|\mathbf{b}_i)d\mathbf{w}_i d\sigma_i^2 \quad (7.7)$$

donde $p(\mathbf{w}_i, \sigma_i^2|\mathbf{b}_i)$ es la probabilidad a priori de los parámetros para \mathbf{w}_i y σ_i^2 . En esta tesis, se escoge una distribución Gaussian-Inverse-Gamma estándar como probabilidad a priori conjugada [13]

$$p(\mathbf{w}_i, \sigma_i^2|\mathbf{b}_i) = \mathcal{N}_{\mathbf{w}_i}(\mathbf{0}, \sigma_i^2 \tilde{\mathbf{R}}) \mathcal{IG}_{\sigma_i^2}(\nu_0, \gamma_0) \quad (7.8)$$

donde $\tilde{\mathbf{R}}^{-1} = (\mathbf{H})^\top \mathbf{H}$ con $\mathbf{H} = \mathbf{R}\mathbf{B}_i$, $\mathbf{B}_i = \text{diag}\{\mathbf{b}_i\}$ y, para no ser informativa, γ_0 y ν_0 toman valores pequeños, reales y positivos. Si se tiene en cuenta todo lo anterior y se usa las probabilidades a priori definidas, en [46] se demostró que la verosimilitud marginal tiene la siguiente forma

$$p(\mathbf{y}_i|\mathbf{b}_i) \propto |\mathbf{P}^\perp|^{-\frac{1}{2}} (\nu_0 + \mathbf{y}_i^\top \mathbf{P}^\perp \mathbf{y}_i)^{-\frac{N+\gamma_0}{2}} \quad (7.9)$$

donde $\mathbf{P}^\perp = \mathbf{I}_N - \mathbf{H}^\top (\mathbf{H}^\top \mathbf{H} + \tilde{\mathbf{R}}^{-1})^{-1} \mathbf{H}^\top$ y \mathbf{I}_N es la matriz identidad $N \times N$.

Aparentemente la solución dada en (7.6) tiene complejidad que crece exponencialmente G de forma que si el número variables ocultas con el que se trabaja es grande, la solución es computacionalmente prohibitiva, esto se debe a que (7.6) es una mezcla de distribuciones de Student.

Un método de muestreo para aproximar este proceso de inferencia Bayesiana fue propuesto en [30]. Se trata de un algoritmo de tipo “Reversible jump Markov chain Monte Carlo” (RJMCMC). Aunque los resultados obtenidos con este algoritmo son buenos, su aplicación está limitada a casos en los que G no es demasiado grande ya que su complejidad computacional es muy elevada .

Por tanto, si se quiere hacer frente a estas dificultades, se ha de ser capaz de desarrollar algoritmos eficientes y que sean capaces de disminuir esta enorme carga computacional cuando el tamaño del conjunto de variables ocultas sea grande.

7.2.3. Solución VBEM

En este capítulo se propone y desarrolla un algoritmo Bayesiano Variacional EM (VBEM) para realizar selección de variable.

Bajo este marco Bayesiano de aprendizaje, las variables desconocidas son tratadas como aleatorias. La idea básica tras el algoritmo VBEM es la de aproximar la distribución en los parámetros y variables latentes de la forma más simple posible construyendo un límite inferior para la verosimilitud marginal intentando optimizar este límite usando un esquema iterativo.

Regla de aprendizaje Variacional Bayesiano

Antes de revisar las reglas de aprendizaje Variacional Bayesiano, se va a examinar qué distribuciones simples son adecuadas para representar las variables desconocidas en el modelo. Para reducir la complejidad computacional de (7.6), se considera que las variables discretas \mathbf{b}_i se distribuyen según una Gaussiana. El punto central de esta aproximación es que la marginalización de una distribución Gaussiana conjunta no requiere una suma en un espacio de alta dimensión en las variables, en su lugar, lo único que hay que hacer es escoger la media y varianza correspondientes del vector de media y matriz de covarianza correspondientemente. Utilizando esta aproximación, escogemos una Gaussiana como distribución a priori con la misma media y varianza que la verdadera distribución a priori de \mathbf{b}_i . Por otro lado, seleccionamos una distribución de tipo Gaussian Inverse-Gamma como distribución a priori para \mathbf{w}_i y σ_i^2 [46]. De esta forma, las distribuciones a priori de los parámetros y las variables latentes son también conjugadas para aproximar la distribución posterior.

Dada la información contenida en la distribución a priori y las observaciones, la verosimilitud marginal $p(\mathbf{y})$ puede acotarse inferiormente utilizando la desigualdad de Jensen de la siguiente forma

$$\begin{aligned} \ln p(\mathbf{y}_i) &= \ln \int d\boldsymbol{\theta}_i d\mathbf{b}_i p(\mathbf{b}_i, \mathbf{y}_i, \boldsymbol{\theta}_i) \\ &\geq \ln \int d\boldsymbol{\theta}_i d\mathbf{b}_i q(\boldsymbol{\theta}_i, \mathbf{b}_i) \ln \frac{p(\mathbf{b}_i, \mathbf{y}_i | \boldsymbol{\theta}_i)}{q(\boldsymbol{\theta}_i, \mathbf{b}_i)} \end{aligned} \quad (7.10)$$

donde $\boldsymbol{\theta}_i = [\mathbf{w}_i^\top \sigma_i^2]^\top$. Además, limitamos la distribución a posteriori $q(\boldsymbol{\theta}_i, \mathbf{b}_i)$ a que pueda factorizarse de la siguiente forma $q(\boldsymbol{\theta}_i, \mathbf{b}_i) \approx q(\boldsymbol{\theta}_i)q(\mathbf{b}_i)$. Como consecuencia, podemos reescribir $\ln p(\mathbf{y}_i)$ como sigue:

$$\ln p(\mathbf{y}_i) \geq \ln \int d\boldsymbol{\theta}_i \left[\int d\mathbf{b}_i q(\mathbf{b}_i) \ln \frac{p(\mathbf{b}_i, \mathbf{y}_i | \boldsymbol{\theta}_i)}{q(\mathbf{b}_i)} + \ln \frac{p(\boldsymbol{\theta}_i)}{q(\boldsymbol{\theta}_i)} \right] \quad (7.11)$$

El algoritmo VBEM maximiza iterativamente la expresión (7.11) con respecto a $q(\boldsymbol{\theta}_i)q(\mathbf{b}_i)$ de acuerdo con la regla de aprendizaje Bayesiana [11]:

Paso VBE:

$$q^{(t+1)}(\mathbf{b}_i) = \frac{1}{Z_{\mathbf{b}_i}} \exp \left[\int d\boldsymbol{\theta}_i q^{(t)}(\boldsymbol{\theta}_i) \ln p(\mathbf{b}_i, \mathbf{y}_i | \boldsymbol{\theta}_i) \right] \quad (7.12)$$

Paso VBM:

$$q^{(t+1)}(\boldsymbol{\theta}_i) = \frac{1}{Z_{\boldsymbol{\theta}_i}} p(\boldsymbol{\theta}_i) \exp \left[\int d\mathbf{b}_i q^{(t+1)}(\mathbf{b}_i) \ln p(\mathbf{b}_i, \mathbf{y}_i | \boldsymbol{\theta}_i) \right] \quad (7.13)$$

donde el superíndice $(t + 1)$ significa número de iteración.

VBEM para Modelos Conjugados-Exponenciales

La distribución a posteriori $p(\mathbf{b}_i, \boldsymbol{\theta}_i | \mathbf{y}_i)$ contiene términos de interacción hasta segundo orden. Como resultado, el hecho de integrar en las variables latentes y en los parámetros no es analíticamente posible. Sin embargo, como el modelo es Conjugado-Exponencial, se cuenta con una herramienta para superar esta dificultad, pidiéndose derivar un algoritmo general de tipo VBEM basado en las propiedades de los modelos Conjugados-Exponenciales (CE) [11]. Antes de comenzar con la derivación de este algoritmo se van a repasar algunos conceptos básicos de los modelos CE.

Los modelos CE satisfacen dos condiciones [11]:

7.2.4. Primera Condición

La verosimilitud del conjunto completo de datos pertenece a la familia exponencial:

$$p(\mathbf{b}_i, \mathbf{y}_i | \boldsymbol{\theta}_i) = g(\boldsymbol{\theta}_i) f(\mathbf{b}_i, \mathbf{y}_i) \exp [\boldsymbol{\phi}(\boldsymbol{\theta}_i)^\top \mathbf{u}(\mathbf{b}_i, \mathbf{y}_i)] \quad (7.14)$$

donde $\boldsymbol{\phi}(\boldsymbol{\theta}_i)$ es el vector de parámetros naturales, \mathbf{u} y f son funciones que definen la familia exponencial, y g es una constante de normalización.

7.2.5. Segunda Condición

La distribución a priori de parámetros es conjugada de la distribución de verosimilitud del conjunto completo de datos:

$$p(\boldsymbol{\theta}_i | \eta, \boldsymbol{\nu}) = h(\eta, \boldsymbol{\nu}) g(\boldsymbol{\theta}_i)^\eta \exp [\boldsymbol{\phi}(\boldsymbol{\theta}_i)^\top \boldsymbol{\nu}] \quad (7.15)$$

donde η y $\boldsymbol{\nu}$ son hiperparámetros de la distribución a priori y h es una constante de normalización.

Si se usan las propiedades anteriores del modelo CE, las reglas de actualización del algoritmo VBEM pueden escribirse como sigue:

Paso VBE:

$$q^{(t+1)}(\mathbf{b}_i) \propto f(\mathbf{b}_i, \mathbf{y}_i) \exp \left[\bar{\boldsymbol{\phi}}^\top \mathbf{u}(\mathbf{b}_i, \mathbf{y}_i) \right] \quad (7.16)$$

donde el vector de parámetros naturales se escribe de la siguiente forma $\bar{\boldsymbol{\phi}} = \int d\boldsymbol{\theta}_i q(\boldsymbol{\theta}_i) \boldsymbol{\phi}(\boldsymbol{\theta}_i) \equiv \langle \boldsymbol{\phi}(\boldsymbol{\theta}_i) \rangle_{q(\boldsymbol{\theta}_i)}$.

Paso VBM:

$$q(\boldsymbol{\theta}_i) = h(\tilde{\eta}, \tilde{\boldsymbol{\nu}}) g(\boldsymbol{\theta}_i)^{\tilde{\eta}} \exp \left[\boldsymbol{\phi}(\boldsymbol{\theta}_i)^\top \tilde{\boldsymbol{\nu}} \right] \quad (7.17)$$

donde

$$\begin{aligned} \tilde{\eta} &= \eta + 1 \\ \tilde{\boldsymbol{\nu}} &= \boldsymbol{\nu} + \mathbf{u}(\bar{\mathbf{y}}_i) \end{aligned} \quad (7.18)$$

y

$$\mathbf{u}(\bar{\mathbf{y}}_i) = \langle \mathbf{u}(\mathbf{b}_i, \mathbf{y}_i) \rangle_{q(\mathbf{b}_i)} \quad (7.19)$$

7.3. Aprendizaje con VBEM

7.3.1. Inicialización

Como punto de partida, es conveniente especificar la forma de las distribuciones a priori de los parámetros y las variables latentes. Como se ha comentado con anterioridad, en primer lugar se identifican las distribuciones a priori conjugadas y se reescriben según el modelo Conjugado Exponencial.

Distribuciones a priori de los parámetros y las variables latentes

$$p(\mathbf{b}_i) = \mathcal{N}(\mathbf{b}_i | \boldsymbol{\mu}_0, \sigma_0^2 \mathbf{I}_G) \quad (7.20)$$

$$\begin{aligned} p(\boldsymbol{\theta}_i) &= p(\mathbf{w}_i, \sigma_i^2) = p(\mathbf{w}_i | \sigma_i^2) p(\sigma_i^2) \\ &= \mathcal{N}(\mathbf{w}_i | \boldsymbol{\mu}_{\mathbf{w}_i}, \sigma_i^2 \mathbf{I}_G) \mathcal{IG} \left(\frac{\nu_0}{2}, \frac{\gamma_0}{2} \right) \end{aligned} \quad (7.21)$$

donde $\boldsymbol{\mu}_0$ y $\sigma_0^2 \mathbf{I}_G$ son la media y la covarianza de la densidad de probabilidad a priori $p(\mathbf{b}_i)$.

Estas distribuciones a priori satisfacen las dos condiciones de los modelos CE que han sido expuestas antes:

7.3.2. Condición I

La verosimilitud del conjunto completo de datos pertenece a la familia de funciones exponenciales

$$p(\mathbf{b}_i, \mathbf{y}_i | \boldsymbol{\theta}_i) = p(\mathbf{y}_i | \mathbf{b}_i, \boldsymbol{\theta}_i) p(\mathbf{b}_i) \quad (7.22)$$

$$= \mathcal{N}(\mathbf{y}_i | \mathbf{R}\mathbf{W}_i \mathbf{b}_i, \sigma_i^2 \mathbf{I}_N) \mathcal{N}(\mathbf{b}_i | \boldsymbol{\mu}_0, \sigma_0^2 \mathbf{I}_G) \quad (7.23)$$

$$= g(\boldsymbol{\theta}_i) f(\mathbf{b}_i, \mathbf{y}_i) \exp [\boldsymbol{\phi}^\top(\boldsymbol{\theta}_i) \mathbf{u}(\mathbf{b}_i, \mathbf{y}_i)] \quad (7.24)$$

donde

$$\boldsymbol{\phi}^\top(\boldsymbol{\theta}_i) = \left[-\frac{1}{2} \sigma_i^{-2} \quad \sigma_i^{-2} \mathbf{R}\mathbf{W}_i \quad -\frac{1}{2} \sigma_i^{-2} (\mathbf{R}\mathbf{W}_i)^\top (\mathbf{R}\mathbf{W}) \right] \quad (7.25)$$

$$\boldsymbol{\mu}(\mathbf{b}_i, \mathbf{y}_i) = [\mathbf{y}_i^\top \mathbf{y}_i \quad \mathbf{y}_i \mathbf{b}_i^\top \quad \mathbf{b}_i \mathbf{b}_i^\top]^\top$$

$$f(\mathbf{b}_i, \mathbf{y}_i) = \mathcal{N}(\mathbf{b}_i | \boldsymbol{\mu}_0, \sigma_0^2 \mathbf{I}_G)$$

$$g(\boldsymbol{\theta}_i) = (2\pi\sigma_i^2)^{-\frac{N}{2}}$$

En las expresiones anteriores $[A]$ representa un vector de tamaño $QN \times 1$ que se obtiene reordenando todas las filas de $A_{Q \times N}$ en una única fila y $[A \ B] = \begin{bmatrix} A & B \end{bmatrix}$.

7.3.3. Condición II

La probabilidad a priori de los parámetros es conjugada de la verosimilitud del conjunto completo de datos. El objetivo es reescribir la probabilidad $p(\boldsymbol{\theta}_i)$ en función de los hiperparámetros $\boldsymbol{\nu}$ y η para demostrar que esta probabilidad a priori conjugada en los parámetros satisface la segunda condición del modelo Conjugado Exponencial.

$$p(\boldsymbol{\theta}_i | \eta, \boldsymbol{\nu}) = h(\eta, \boldsymbol{\nu}) g(\boldsymbol{\theta}_i)^\eta \exp [\boldsymbol{\phi}(\boldsymbol{\theta}_i)^\top \boldsymbol{\nu}] \quad (7.26)$$

donde

$$\eta = \frac{\gamma_0 + 2 + G}{N} \quad (7.27)$$

$$\boldsymbol{\nu} = [\nu_0 \ 0 \ \mathbf{V}_2]^\top$$

7.3.4. Paso VBE

En este paso el objetivo es encontrar las distribuciones variacionales a posteriori de las variables latentes. Para ello se aplica (7.16) tomando como

punto de partida una función de distribución arbitraria en los parámetros:

$$q(\mathbf{b}_i) = \mathcal{N}(\mathbf{b}_i | \boldsymbol{\mu}_{\mathbf{b}_i}, \Sigma_{\mathbf{b}_i}) \quad (7.28)$$

donde

$$\begin{aligned} \boldsymbol{\mu}_{\mathbf{b}_i} &= \Sigma_{\mathbf{b}_i}(\sigma_0^{-2}\boldsymbol{\mu}_0 + \mathbf{f}) \\ \Sigma_{\mathbf{b}_i} &= (\sigma_0^{-2}\mathbf{I}_G + D)^{-1} \end{aligned} \quad (7.29)$$

con

$$\begin{aligned} \mathbf{D} &= \mathbf{B} \otimes [(\mathbf{m}_{\mathbf{w}_i} \mathbf{m}_{\mathbf{w}_i})^\top \langle \sigma_i^{-2} \rangle_{q(\theta_i)} + \mathbf{A}^{-1}] \\ \mathbf{f}^\top &= \mathbf{y}_i^\top \mathbf{R} \text{diag}(\mathbf{m}_{\mathbf{w}_i}) \langle \sigma_i^{-2} \rangle_{q(\theta_i)} \\ \mathbf{B} &= \mathbf{R}^\top \mathbf{R} \\ \mathbf{A} &= \mathbf{I}_G + \mathbf{K} \\ \mathbf{K} &= \mathbf{B} \otimes (\Sigma_{\mathbf{b}_i} + \boldsymbol{\mu}_{\mathbf{b}_i} \boldsymbol{\mu}_{\mathbf{b}_i}^\top) \end{aligned} \quad (7.30)$$

La distribución marginal a posteriori $p(b_i(l) | \mathbf{y}_i)$ se actualiza de la siguiente forma hasta que el algoritmo converja:

$$p(b_i(l) | \mathbf{y}_i) = \mathcal{N}(\mathbf{m}_{\mathbf{b}_i}(l), \Sigma_{\mathbf{b}_i}(l, l)) \quad (7.31)$$

y entonces, la decisión en $\mathbf{b}_i(l)$ se realiza de la siguiente forma

$$\hat{b}_i(l) = \text{argmax}_{b_i(l)} p(b_i(l) | \mathbf{y}_i) \quad (7.32)$$

7.3.5. Paso VBM

En cuanto al paso VBM, en primer lugar se calcula $q(\boldsymbol{\theta}_i)$ según lo explicado en(7.17).

$$q(\boldsymbol{\theta}_i) = \mathcal{N}(\mathbf{w}_i | \mathbf{m}_{\mathbf{w}_i}, \Sigma_{\mathbf{w}_i}) \mathcal{IG} \left(\frac{\alpha}{2}, \frac{\beta}{2} \right) \quad (7.33)$$

donde

$$\begin{aligned} \mathbf{m}_{\mathbf{w}_i} &= (\mathbf{I}_G + \mathbf{K})^{-1} (\mathbf{y}_i^\top \mathbf{R} \mathbf{M}_x)^\top \\ \Sigma_{\mathbf{w}_i} &= \sigma_i^2 (\mathbf{I}_G + \mathbf{K})^{-1} \\ \frac{\alpha}{2} &= \frac{N(\eta + 1) - 2}{2} \\ \frac{\beta}{2} &= -\frac{c}{2} \end{aligned} \quad (7.34)$$

con

$$\begin{aligned} \mathbf{M}_x &= \text{diag}(\mathbf{m}_{\mathbf{b}_i}) \\ c &= \mathbf{y}_i^\top \mathbf{R} \mathbf{M}_x \mathbf{m}_{\mathbf{w}_i} - \mathbf{y}_i^\top \mathbf{y}_i - \nu_0 \end{aligned} \quad (7.35)$$

7.3.6. Cálculo del límite inferior \mathcal{F}

En el algoritmo VBEM, se actualiza iterativamente el límite inferior basándose en las últimas funciones de distribución $q(\mathbf{b}_i)$ y $q(\boldsymbol{\theta}_i)$ calculadas. Este proceso se repite hasta que el algoritmo converja. De ahora en adelante, se usará \mathcal{F} para representar el límite inferior de $\ln p(\mathbf{y}_i)$. En este proceso de cálculo del límite inferior, se hace uso de la distancia de Kulback-Leiber.

La distancia Kulback-Leiber mide la diferencia entre dos distribuciones de probabilidad. Esta distancia también se conoce con el nombre de entropía relativa. Matemáticamente puede expresarse de la siguiente forma:

$$\begin{aligned} \text{KL}[q(\mathbf{b}_i)||p(\mathbf{b}_i, \mathbf{y}_i|\boldsymbol{\theta}_i)] &= - \int d\mathbf{b}_i q(\mathbf{b}_i) \ln \frac{p(\mathbf{b}_i, \mathbf{y}_i|\boldsymbol{\theta}_i)}{q(\mathbf{b}_i)} \quad (7.36) \\ \text{KL}[q(\boldsymbol{\theta}_i)||p(\boldsymbol{\theta}_i)] &= - \int d\boldsymbol{\theta}_i \ln \frac{p(\boldsymbol{\theta}_i)}{q(\boldsymbol{\theta}_i)} \end{aligned}$$

Una vez definida matemáticamente la distancia Kulback-Leiber, se está en disposición de comenzar con el cálculo del límite inferior \mathcal{F} .

$$\begin{aligned} \mathcal{F} &= \ln \int d\boldsymbol{\theta}_i q(\boldsymbol{\theta}_i) \left[\int d\mathbf{b}_i q(\mathbf{b}_i) \ln \frac{p(\mathbf{b}_i, \mathbf{y}_i|\boldsymbol{\theta}_i)}{q(\mathbf{b}_i)} + \ln \frac{p(\boldsymbol{\theta}_i)}{q(\boldsymbol{\theta}_i)} \right] \quad (7.37) \\ &= - \int d\boldsymbol{\theta}_i q(\boldsymbol{\theta}_i) \text{KL} [q(\mathbf{b}_i)||p(\mathbf{b}_i, \mathbf{y}_i|\boldsymbol{\theta}_i)] - \text{KL} [q(\boldsymbol{\theta}_i)||p(\boldsymbol{\theta}_i)] \\ &= -\frac{N}{2} \ln(2\pi) - \frac{N}{2} \langle \ln \sigma_i^2 \rangle_{q(\sigma_i^2)} - \frac{G}{2} \ln \sigma_0^2 + \frac{1}{2} \ln |\Sigma_{\mathbf{b}_i}| \\ &\quad - \frac{1}{2} \langle [\mathbf{E}] \rangle_{q(\sigma_i^2)}^\top [\Sigma_{\mathbf{b}_i}] - \frac{1}{2} \boldsymbol{\mu}_{\mathbf{b}_i}^\top \langle \mathbf{E} \rangle_{q(\sigma_i^2)} \boldsymbol{\mu}_{\mathbf{b}_i} + \frac{G}{2} \\ &\quad - \left(\frac{1}{2} \langle \sigma_i^{-2} \rangle_{q(\sigma_i^2)} \mathbf{m}_{\mathbf{w}_i}^\top \mathbf{m}_{\mathbf{w}_i} + \frac{1}{2} \text{tr}(\mathbf{A}^{-1}) - \frac{1}{2} G + \frac{1}{2} \ln |\mathbf{A}| \right. \\ &\quad \left. - \frac{1}{2} (\alpha - \gamma_0) \langle \ln \sigma_i^2 \rangle_{q(\sigma_i^2)} - \frac{1}{2} (\beta - \nu_0) \langle \sigma_i^{-2} \rangle_{q(\sigma_i^2)} \right) \end{aligned}$$

donde

$$\begin{aligned} \mathbf{E} &= \sigma_0^{-2} \mathbf{I}_G + \sigma_i^{-2} \mathbf{B} \otimes \mathbf{w}_i \mathbf{w}_i^\top \quad (7.38) \\ \langle \mathbf{E} \rangle_{q(\sigma_i^2)} &= \sigma_0^{-2} \mathbf{I}_G + \mathbf{D} \end{aligned}$$

con el conjunto de estadísticos suficientes de los parámetros obtenidos en el paso VBM,

$$\begin{aligned} \langle \sigma_i^{-2} \rangle_{q(\sigma_i^2)} &= \frac{\alpha}{\beta} \quad (7.39) \\ \langle \ln \sigma_i^2 \rangle_{q(\sigma_i^2)} &= -\Psi \left(\frac{\alpha}{2} \right) + \ln \frac{\beta}{2} \end{aligned}$$

donde $\Psi(x) \equiv \partial/\partial \ln \Gamma(x)$ es la función digamma.

7.3.7. Resumen del Algoritmo

En el siguiente cuadro se resume el algoritmo VBEM:

Algoritmo VBEM aplicado a la selección de variables

1. Inicialización

μ_0, σ_0^2 : inicialización de las distribuciones a priori de las variables latentes \mathbf{b}_i

$\mu_{\mathbf{w}}, \sigma_i^2, \nu_0, \gamma_0$: inicializar las distribuciones a priori de los parámetros θ_i

$\mu_{\mathbf{b}_i}, \Sigma_{\mathbf{b}_i}$: inicializar las distribuciones a posteriori de las variables latentes \mathbf{b}_i

$\mathbf{m}_{\mathbf{w}_i}, \Sigma_{\mathbf{w}_i}, \alpha, \beta$: inicializar las distribuciones a posteriori de los parámetros θ_i

2. Paso VBE

Inferir la distribución de las variables latentes $q(\mathbf{b}_i)$ haciendo uso de las ecuaciones (7.28)

3. Paso VBM

Inferir las distribuciones a posteriori de los parámetros $q(\theta_i)$ haciendo uso de las ecuaciones (7.33)

4. Calcular \mathcal{F}

Si \mathcal{F} crece, volver al punto 2). Si no, pasar al paso 5).

5. Detección

Detectar las $b_i(l)$ utilizando la expresión (7.32).

7.4. Algoritmo VBEM con Ligaduras

En algunos problemas de procesamiento de señal (especialmente aquellos relacionados con la genómica), el tamaño del conjunto de las observaciones es mucho menor que el número de variables latentes. El modelo que se estudia se convierte por tanto en indeterminado y el proceso de inferencia es más difícil. Para aliviar este efecto, se propone incorporar ligaduras en las

distribuciones a priori de las variables latentes. En particular, se propone introducir limitaciones en el número de variables latentes que afectan a una dada.

7.4.1. Objetivo

El objetivo, como ya se ha dicho, es la introducción de información en la distribución de probabilidad a priori de las variables latentes $p(\mathbf{b})$. En el problema que se estudia, esta información está relacionada con la limitación del número máximo de variables latentes que afectan a una dada. En términos de probabilidad, esto significa que tenemos que diseñar una función de probabilidad a priori $p(\mathbf{b})$ de forma que las probabilidades más altas se asocien a aquellas situaciones en las que el número de variables latentes que afectan a una dada sea igual a un número medio establecido y que esta probabilidad decrezca rápidamente a partir de un número de variables máximo permitido.

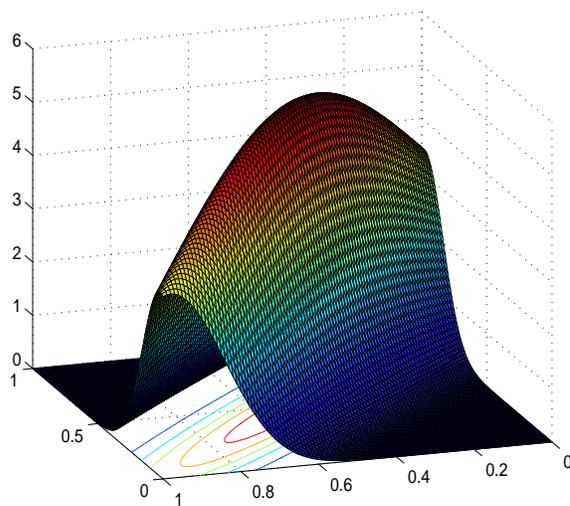


Figura 7.1: $\sigma_{12} = -0,9\sigma_1\sigma_2$

7.4.2. Ejemplo con dos variables

En el siguiente ejemplo, se presenta la forma de introducir la información deseada en la distribución a priori. Para ello se va a empezar con el caso más sencillo en el que se trabaja con dos variables latentes, o sea, $G = 2$ y el

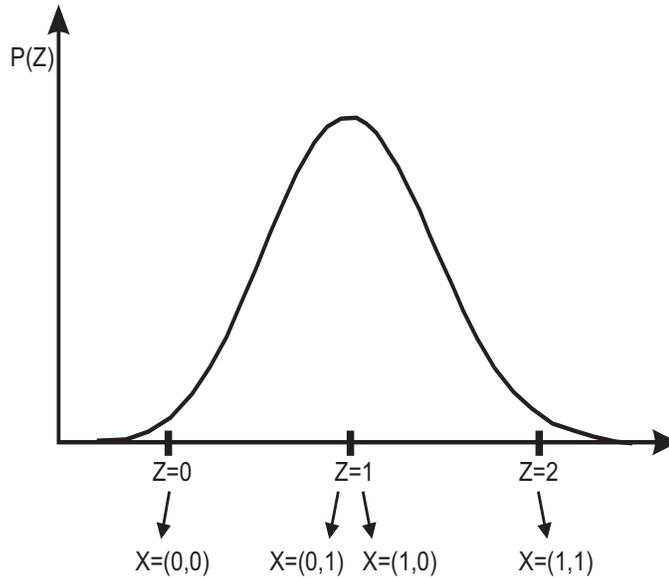


Figura 7.2: $p(z)$ correspondiente a un espacio bidimensional

máximo número de variables que pueden afectar a una dad es 1. Entonces, el objetivo será evitar las siguientes dos situaciones:

- Cada variable latente depende del valor de ella misma y del valor de la otra en el instante anterior. Esta situación se representa por el vector $\mathbf{b} = (1, 1)$.
- El valor de cada variable latente no depende ni de su propio valor en el instante anterior ni del valor de la otra variable en el instante anterior. Esta situación se representa por el vector $\mathbf{b} = (0, 0)$.

Ya que se está trabajando con distribuciones a priori que tienen distribución Gaussiana, nuestra distribución a priori es una Gaussiana bidimensional. Y además se está interesado en favorecer situaciones representadas por $\mathbf{b} = (1, 0)$ and $\mathbf{b} = (0, 1)$. Gráficamente, podemos representar esta preferencia en términos de probabilidad como puede verse en la Figura (7.1). En esta figura puede observarse que la zona de alta probabilidad está lejos de las configuraciones $\mathbf{b} = (0, 0)$ and $\mathbf{b} = (1, 1)$. Sin embargo, es difícil estudiar cómo el control del máximo número de padres afecta a la probabilidad si se trabaja con una Gaussiana bidimensional. Para solucionar este problema, se propone relacionar el vector \mathbf{b} en función del máximo número de padres. Esta relación se muestra en la siguiente transformación lineal:

$$z = \mathbf{d}^T \mathbf{b} \quad (7.40)$$

donde, $\mathbf{d} = [\mathbf{1}, \dots, \mathbf{1}]^T$, su tamaño es $(G \times 1)$ y en el caso bidimensional su tamaño es (2×1) . z es el número de variables de las que depende una dada, esto es, el número de unos del vector \mathbf{b} . Desde ahora se va a trabajar con la variable z . Recordar que el objetivo es limitar el máximo número de padres, así que se quiere distribuir la variable z de forma que las probabilidades más altas estén asociadas con valores de z cercanos al número medio de variables que se quiere que afecten a una dada y que la probabilidad decrezca rápidamente a partir de valores de z mayores al número máximo de padres permitido. En la Figura (7.2) se representa la distribución deseada, $p(z)$, con la equivalencia entre la variable \mathbf{b} y z en el caso que se estudia. Si se supone que la variable z se distribuye como una Gaussiana, su expresión matemática es:

$$z \sim \mathcal{N}(\mathbf{d}^T \boldsymbol{\mu}, \mathbf{d}^T \mathbf{C} \mathbf{d}) \quad (7.41)$$

donde \mathbf{C} es la matriz de covarianza y $\boldsymbol{\mu}$ es el vector de media de la variable \mathbf{b} . Así que la media y la varianza de la variable z pueden calcularse de la siguiente forma:

$$\mu_z = \mathbf{d}^T \boldsymbol{\mu} \quad (7.42)$$

$$\sigma_z^2 = \mathbf{d}^T \mathbf{C} \mathbf{d} \quad (7.43)$$

Como se trabaja con una distribución Gaussiana, sólo se necesita contar con dos valores para fijar los estadísticos suficientes de la distribución. El primer valor que se fija es el de la media de la distribución que se escoge atendiendo al número de variables medio que queremos que afecten a una dada. El segundo valor a escoger es el número máximo de variables que afectan a una dada.

Con el valor de la media fijado, se puede calcular el valor de la media de la distribución a priori $p(\mathbf{b})$ utilizando la ecuación 7.42.

Por otro lado, el valor de la varianza σ_z^2 se fijará utilizando el valor máximo de variables latentes que afecten a una dada. En este proceso se introducen condiciones que conduzcan a hacer que la probabilidad asociada a valores de variables latentes que afectan a una dada mayores que el máximo sean lo más cercano a cero que sea posible (en el caso bidimensional estas situaciones se corresponderían con ninguna variable afecta o todas afectan).

Ahora, puede volverse al caso bidimensional para aclarar lo anterior. La matriz de covarianza es:

$$\mathbf{C} = \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{21} & \sigma_2^2 \end{bmatrix}$$

Por tanto, $\sigma_z^2 = \sigma_1^2 + \sigma_2^2 + \sigma_{12} + \sigma_{21}$. En nuestro caso, $\sigma_1^2 = \sigma_2^2$ y $\sigma_{12} = \sigma_{21}$, así que $\sigma_z^2 = 2\sigma_1^2 + 2\sigma_{21}$. Fijar 1 como el máximo número de variables que

pueden afectar a una dada significa que se rechazan situaciones como aquellas en las que ninguna variable afecta o las dos afectan como se representa en la Figura (7.2) deben de ser muy picudas, o en otras palabras, σ_z^2 debe ser tan cercana a cero como sea posible. Si se fija $\sigma_z^2 = 0$ se tiene la relación $\sigma_{21} = -\sqrt{\sigma_1}$. Y esta es precisamente la relación que es necesario usar para calcular **C** la distribución a priori.

7.5. Cálculos en una dimensión

Una vez que se han introducido las ideas principales relacionadas con el proceso de incluir ligaduras en el algoritmo VBEM original en dos dimensiones, puede abordarse el caso en que G es mayor que 2. Teniendo presentes las ideas presentadas en 7.4.2 el primer paso es calcular los valores de μ_z y σ_z .

7.5.1. Cálculo de μ_z

Si se supone fijado el número medio de variables que pueden afectar a una dada igual a Pa_{ave} , el cálculo de μ_z que se propone es muy directo: concentrar los valores más altos de probabilidad alrededor de ese número medio de variables que afectan a una dada. Por tanto, esto significa que:

$$\mu_z = Pa_{ave} \quad (7.44)$$

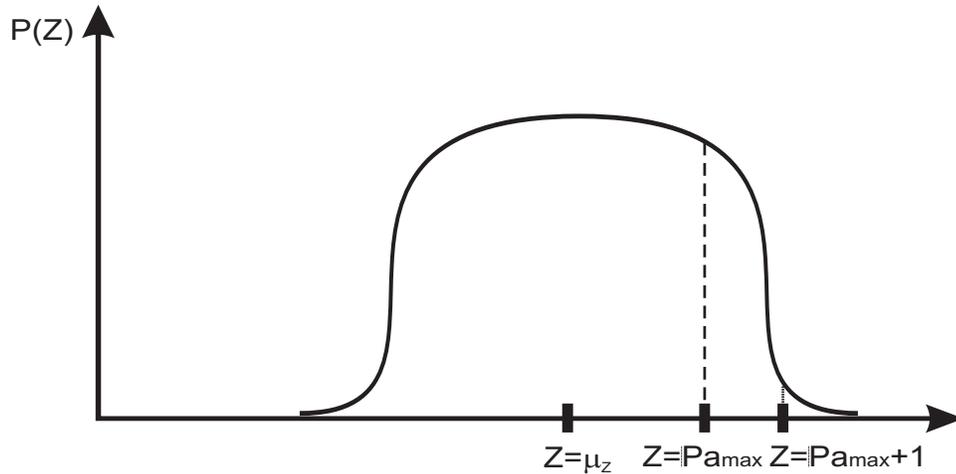


Figura 7.3: $p(z)$ correspondiente a un espacio G -dimensional

7.5.2. Cálculo de σ_z^2

Como se ha dicho anteriormente, la idea principal del procedimiento en una dimensión es fijar el conjunto de condiciones que nos permitirán calcular σ_z . En este apartado se va a proponer un método para calcular σ_z^2 basado en dos restricciones:

1. La Gaussiana monodimensional debe ser tan plana como sea posible en la zona entre la media y el valor máximo de variables que afecten a una dada, Pa_{max} . Esta condición se traduce en que el cociente entre la probabilidad de $z = Pa_{max}$ y la probabilidad de la media debe ser tan grande como sea posible.
2. La probabilidad debe decrecer lo más rápido posible para valores de la variable superiores a Pa_{max} .

Se ha presentado la forma de $p(z)$ para un caso ideal de acuerdo con las restricciones de la Figura (7.3). Obviamente, se está haciendo referencia a un caso ideal porque se está trabajando con una distribución Gaussiana y es difícil lograr esa forma si trabajamos con una Gaussiana.

El primer objetivo puede escribirse de forma matemática con la maximización de la función:

$$\frac{p(z = Pa_{max})}{p(z = Pa_{ave})} \quad (7.45)$$

En cuanto al segundo objetivo, éste puede escribirse también matemáticamente a través de la maximización de la siguiente función:

$$\frac{p(z = Pa_{max})}{p(z = Pa_{max} + 1)} \quad (7.46)$$

Así que, finalmente, la función global que tenemos que maximizar tiene la expresión:

$$J(\sigma_z^2) = \frac{p(z = Pa_{max})}{p(z = Pa_{ave})} + \frac{p(z = Pa_{max})}{p(z = Pa_{max} + 1)} \quad (7.47)$$

Si maximizamos (7.47) con respecto a σ_z^2 se obtiene:

$$\sigma_z^2 = \frac{1}{2} \frac{(2Pa_{max} + 1 - 2\mu + (Pa_{max} - \mu)^2)}{\ln\left(\frac{(Pa_{max} - \mu)^2}{2Pa_{max} + 1 - 2\mu}\right)} \quad (7.48)$$

7.6. Cálculos en el espacio G -Dimensional

Como se ha descrito antes, el objetivo que se persigue es la obtención de un conjunto de estadísticos suficientes para la distribución a priori $p(\mathbf{b})$. Se ha trabajado en una dimensión para incluir las ligaduras de forma más directa. Por tanto, una vez que este conjunto de estadísticos suficientes se ha determinado, puede pasarse del espacio monodimensional al espacio multidimensional. En otras palabras, puede calcularse μ_i y la matriz \mathbf{C} .

7.6.1. Cálculo de μ_i

Una vez que se ha fijado μ_z puede usarse su valor y la expresión (7.42) para calcular μ_i . Ya que a priori no se tiene ninguna información acerca de ningún elemento de \mathbf{b} en particular, se establece la siguiente condición:

$$\mu_i = \frac{1}{G} (Pa_{ave}) \quad (7.49)$$

7.6.2. Cálculo de \mathbf{C}

En esta sección analizamos el problema de la elección de la matriz \mathbf{C} . Esta matriz tiene la siguiente forma:

$$\mathbf{C} = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \dots & \sigma_{1G} \\ \sigma_{21} & \sigma_2^2 & \dots & \sigma_{2G} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{G1} & \sigma_{G2} & \dots & \sigma_{GG} \end{bmatrix}$$

Si usamos ahora la transformación lineal descrita en 7.43 puede escribirse la siguiente expresión:

$$\sigma_z^2 = \sum_{i=1}^G \sigma_i^2 + \sum_{i<j} \sigma_{ij} \quad (7.50)$$

Si se fijan $\sigma_i^2 = \sigma^2$ y $\sigma_{ij} = \tilde{C}$ se tiene la siguiente ecuación:

$$\tilde{C} = \frac{\sigma_z^2 - G\sigma^2}{G^2 - G} \quad (7.51)$$

7.7. Solución VBEM-PDA

En esta sección, se presenta una nueva solución al problema de selección de variables mediante la modificación del algoritmo de tipo VBEM propuesto

en las secciones anteriores. La idea es aprovechar la aproximación a la distribución a posteriori que se obtiene de la aplicación del algoritmo de tipo VBEM e introducirla como probabilidad a priori en un algoritmo de tipo PDA (Probabilistic Data Association).

7.7.1. Algoritmos de tipo Asociación Probabilística de Datos (PDA)

Recientemente se ha propuesto un algoritmo iterativo de detección muy prometedor llamado “Detector PDA” pues está basado en los algoritmos de tipo PDA que constituyen una aproximación bastante popular en el campo de seguimiento de varios blancos en ambientes con scattering acentuado. Este tipo de detectores posee mejor comportamiento así como menor carga computacional en comparación con otros detectores subóptimos populares. Además, otra ventaja llamativa de este tipo de algoritmos es el hecho de que proporcionan información probabilística sobre las variables desconocidas, en este caso los bits enviados. Es por esto, que este tipo de algoritmos puede aplicarse de forma natural a sistemas codificados de detección multiusuario de tipo turbo. Precisamente es esta última característica la que nos lleva a introducir este tipo de algoritmos en la solución propuesta al problema de selección de variables.

Entonces, si se parte del modelo del sistema dado por 7.1, el objetivo del algoritmo detector de tipo PDA es obtener la probabilidad a posteriori $p(b_i|\mathbf{y})\forall i$. Esta probabilidad a posteriori puede calcularse utilizando la fórmula de Bayes:

$$p(b_i(l)|\mathbf{y}_i) = \frac{p(\mathbf{y}_i|b_i(l))p(b_i(l))}{\int db_i(l)p(\mathbf{y}_i|b_i(l))p(b_i(l))} \quad (7.52)$$

de donde puede verse que $p(b_i(l)|\mathbf{y}_i)$ puede calcularse de forma sencilla si se conoce $p(\mathbf{y}_i|\mathbf{b}_i(l))$. Sin embargo, para calcular esta última probabilidad es necesario llevar a cabo un proceso de marginalización:

$$p(b_i(l))p(\mathbf{y}_i|b_i(l)) = \int d(\mathbf{b}_i)_{-l}p(\mathbf{y}_i|\mathbf{b}_i)p((\mathbf{b}_i)_{-l})p(b_i(l)) \quad (7.53)$$

En este caso, la integral representada en la ecuación anterior es una mezcla de Gaussianas con lo que la complejidad computacional asociada a esta operación es muy grande e incrementa con el número de variables latentes. Una posible simplificación consiste en la realización de un proceso de asociación de datos, o lo que es lo mismo, aproximar la mezcla de Gaussiana por una única Gaussiana con media y varianza adaptadas a la mezcla. Esta

aproximación puede refinarse un poco más mediante la introducción de un proceso iterativo.

7.7.2. Algoritmo VBEM-PDA

En el proceso de búsqueda de una solución al problema de selección de variables, se va a intentar introducir un paso de detección PDA una vez que el algoritmo VBEM ha convergido. Es por esto que la parte de detección PDA se considera aquí como un proceso de refinamiento o mejora de las Probabilidades a Posteriori (APPs) suministradas por el algoritmo VBEM.

Cuando el algoritmo VBEM que se ha desarrollado en las secciones anteriores ha convergido, identificamos lo que se ha llamado $q(\mathbf{b}_i)$ con la probabilidad a posteriori $p(\mathbf{b}_i|\mathbf{y}_i)$. Esta probabilidad se utiliza en el primer paso del algoritmo PDA para calcular la verosimilitud $p(\mathbf{y}_i|\mathbf{b}_i)$ con la fórmula de Bayes:

$$p(\mathbf{y}_i|\mathbf{b}_i) \propto \frac{p(\mathbf{b}_i|\mathbf{y}_i)}{p(\mathbf{b}_i)} \quad (7.54)$$

Como las dos distribuciones anteriores son Gaussianas, se puede usar el teorema dado en el Apéndice E para el caso en que las dos Gaussianas se estén dividiendo. El resultado es otra Gaussiana con media y matriz de covarianza dadas por las siguientes expresiones:

$$\begin{aligned} \boldsymbol{\mu}_{\mathbf{y}_i} &= (\boldsymbol{\Sigma}_{\mathbf{b}_i}^{-1} - \boldsymbol{\Sigma}_0^{-1})^{-1} (\boldsymbol{\Sigma}_{\mathbf{b}_i}^{-1} \boldsymbol{\mu}_{\mathbf{b}_i} - \boldsymbol{\Sigma}_0^{-1} \boldsymbol{\mu}_0) \\ \boldsymbol{\Sigma}_{\mathbf{y}_i} &= (\boldsymbol{\Sigma}_{\mathbf{b}_i}^{-1} - \boldsymbol{\Sigma}_0^{-1}) \end{aligned} \quad (7.55)$$

donde $\boldsymbol{\Sigma}_{\mathbf{b}_i}$ y $\boldsymbol{\mu}_{\mathbf{b}_i}$ son respectivamente la matriz de covarianza y el vector de media de $p(\mathbf{b}_i|\mathbf{y}_i)$ y $\boldsymbol{\Sigma}_0$ y $\boldsymbol{\mu}_0$ son respectivamente es la matriz de covarianza y el vector de media de $p(\mathbf{b}_i)$.

Una vez calculada la verosimilitud $p(\mathbf{y}_i|\mathbf{b}_i)$, ésta puede usarse para determinar la nueva probabilidad a posteriori de la siguiente manera: en primer lugar multiplicamos en ambos lados de la expresión 7.53 por $p(b_i(l))$:

$$\begin{aligned} p(b_i(l))p(\mathbf{y}_i|b_i(l)) &= \int d(\mathbf{b}_i)_{-l} p(\mathbf{y}_i|\mathbf{b}_i) p((\mathbf{b}_i)_{-l}) p(b_i(l)) \\ p(b_i(l)|\mathbf{y}_i) &= \int d(\mathbf{b}_i)_{-l} p(\mathbf{y}_i|\mathbf{b}_i) p((\mathbf{b}_i)_{-l}) p(b_i(l)) \\ p(b_i(l)|\mathbf{y}_i) &= \int d(\mathbf{b}_i)_{-l} p(\mathbf{y}_i|\mathbf{b}_i) p(\mathbf{b}_i) \end{aligned} \quad (7.56)$$

El integrando de la expresión anterior es el producto de dos Gaussianas que es a su vez otra Gaussiana cuya media y varianza pueden calcularse usando el teorema que aparece en el Apéndice E. Estas expresiones son:

$$\begin{aligned}\widetilde{\boldsymbol{\mu}}_{\mathbf{b}_i} &= (\boldsymbol{\Sigma}_{\mathbf{y}_i}^{-1} + \underline{\boldsymbol{\Sigma}}_0^{-1})^{-1} \left(\boldsymbol{\Sigma}_{\mathbf{y}_i}^{-1} \boldsymbol{\mu}_{\mathbf{y}_i} + \underline{\boldsymbol{\Sigma}}_0^{-1} \boldsymbol{\mu}_0 \right) \\ \widetilde{\boldsymbol{\Sigma}}_{\mathbf{b}_i}^{-1} &= (\boldsymbol{\Sigma}_{\mathbf{y}_i}^{-1} + \underline{\boldsymbol{\Sigma}}_0^{-1})\end{aligned}\quad (7.57)$$

Entonces,

$$p(\mathbf{y}_i | \mathbf{b}_i) p(\mathbf{b}_i) \sim \mathcal{N} \left(\widetilde{\boldsymbol{\mu}}_{\mathbf{b}_i}, \widetilde{\boldsymbol{\Sigma}}_{\mathbf{b}_i}^{-1} \right) \quad (7.58)$$

Finalmente, el resultado de la integral de 7.56 será de nuevo una Gaussiana de la siguiente forma:

$$p(b_i(l)) p(\mathbf{y}_i | b_i(l)) \sim \mathcal{N} \left(\widetilde{\boldsymbol{\mu}}_{\mathbf{b}_i}(l), \widetilde{\boldsymbol{\Sigma}}_{\mathbf{b}_i}(l, l) \right) \quad (7.59)$$

Al final de una iteración completa el primer paso es calcular la probabilidad de que $b_i(l) = 1$, o sea, calcular $P_l \forall l$. Con esos valores se calcula el vector de media y las componentes diagonales de la matriz de covarianza actualizada para usarlas en la siguiente iteración. Como se considera que se trabaja con un proceso de Bernoulli, esta media y varianza se escriben como:

$$\begin{aligned}\underline{\boldsymbol{\mu}}_0(l) &= P_l \\ \underline{\boldsymbol{\Sigma}}_0^{-1}(l, l) &= P_l(1 - P_l)\end{aligned}\quad (7.60)$$

Además, para cada índice l se calcula la distancia Kulback-Leibler entre la distribución de Bernoulli en la iteración anterior y la actual. El máximo de esta distancia en cada iteración se usa para compararlo con un umbral que indica la convergencia del algoritmo. Finalmente apuntar que cuando se alcanza la convergencia se lleva a cabo el proceso de detección para el que se usan los P_l calculados en la última iteración de la siguiente forma:

$$\mathbf{b}_i(l) = \begin{cases} 1, & \text{si } P_l \geq 0,5 \\ -1, & \text{en otro caso} \end{cases}$$

En la siguiente tabla se presenta un resumen del algoritmo:



Algoritmo VBEM-PDA aplicado a la selección de variables

1. **Inicialización**

μ_0, σ_0^2 : inicialización de las distribuciones a priori de las variables latentes \mathbf{b}_i

$\mu_{\mathbf{w}}, \sigma_i^2, \nu_0, \gamma_0$: inicializar las distribuciones a priori de los parámetros θ_i

$\mu_{\mathbf{b}_i}, \Sigma_{\mathbf{b}_i}$: inicializar las distribuciones a posteriori de las variables latentes \mathbf{b}_i

$\mathbf{m}_{\mathbf{w}_i}, \Sigma_{\mathbf{w}_i}, \alpha, \beta$: inicializar las distribuciones a posteriori de los parámetros θ_i

2. **Paso VBE**

Inferir la distribución de las variables latentes $q(\mathbf{b}_i)$ haciendo uso de las ecuaciones (7.28)

3. **Paso VBM**

Inferir las distribuciones a posteriori de los parámetros $q(\theta_i)$ haciendo uso de las ecuaciones (7.33)

4. **Calcular \mathcal{F}**

Si \mathcal{F} crece, volver al punto 2). Si no, pasar al paso 5).

5. **Cálculo de $p(\mathbf{y}_i|\mathbf{b}_i)$**

Calcular el vector de media, $\mu_{\mathbf{y}_i}$, y la matriz de covarianza, $\Sigma_{\mathbf{y}_i}$, usando 7.55.

6. **Cálculo de $p(\mathbf{b}_i(l)|\mathbf{y}_i)$**

Para cada l desde 1 hasta G , calcular el vector de media, $\underline{\mu}_0(l)$, y la matriz de covarianza, $\underline{\Sigma}_0^{-1}(l, l)$, usando 7.57.

7. **Cálculo de P_l**

Para cada l desde 1 hasta G , calcular P_l usando la media y la varianza obtenidas en el apartado anterior.

8. **Cálculo de la distancia Kulback-Leibler**

Calcular la distancia de Kulback-Leibler y compararla con el umbral, si es menor que el umbral pasar a y si es mayor pasar al siguiente paso.

9. **Actualización del vector de media y la matriz de covarianza**

Actualizar las componentes del vector de media y los elementos diagonales de la matriz de covarianza de acuerdo con 7.60.

10. **Detección**

Para cada l desde 1 hasta G , realizar detección usando 7.61

7.8. Resultados

7.8.1. Comparación entre el algoritmo VBEM y el algoritmo de Gibbs Sampling

En esta sección se compara el comportamiento del algoritmo VBEM con el de Gibbs sampling. Para ello, se han generado distintos conjuntos de observaciones utilizando distintas configuraciones del vector \mathbf{b} . En todos los casos, se ha seleccionado $G = 8$ y los valores $N = 8$, $N = 10$ y $N = 15$. Se han utilizado siempre valores de N mayores o iguales que G porque el algoritmo de Gibbs Sampling no es aplicable a casos en los que G es mayor que N . En cuanto a los valores del vector \mathbf{b} , se han elegido los siguientes:

- $\mathbf{b}_1 = [00100010]$
- $\mathbf{b}_2 = [10101010]$
- $\mathbf{b}_3 = [10111011]$

En el algoritmo de Gibbs Sampling el número de muestras con las que se ha trabajado ha sido 500. Finalmente apuntar que la desviación típica del ruido añadido ha sido de 0.01.

A continuación se muestran en una tabla el número de errores obtenidos de 100 realizaciones de los dos algoritmos anteriores para las distintas configuraciones comentadas:

Condiciones	Variable \mathbf{b}	Errores Gibbs Sampling	Errores VBEM
G=8,N=8	\mathbf{b}_1	168	0
G=8,N=8	\mathbf{b}_2	306	0
G=8,N=8	\mathbf{b}_3	522	12
G=8,N=10	\mathbf{b}_1	125	0
G=8,N=10	\mathbf{b}_2	266	0
G=8,N=10	\mathbf{b}_3	473	0
G=8,N=8	\mathbf{b}_1	100	0
G=8,N=8	\mathbf{b}_2	207	0
G=8,N=8	\mathbf{b}_3	398	0

Como puede verse, en todos los casos el algoritmo VBEM muestra un comportamiento mucho mejor que el del algoritmo de “Gibbs sampling”. Además, es importante resaltar que la complejidad computacional del algoritmo propuesto es mucho menor que la del algoritmo de Gibbs Sampling por lo que su uso es mucho más apropiado para este tipo de sistemas sobre determinados.

7.8.2. Comparación entre el algoritmo VBEM y el VBEM con ligaduras

En la subsección anterior se ha mostrado cómo el algoritmo propuesto en este capítulo de la tesis mejora al algoritmo de “Gibbs sampling” tanto en rendimiento como en complejidad computacional cuando el sistema es sobre determinado. En esta parte se muestra cómo el algoritmo VBEM con ligaduras es más apropiado que el algoritmo VBEM genérico para aplicación de a sistemas indeterminados ($G > N$). Para comparar el comportamiento de ambos algoritmos, se ha procedido de igual forma que se hizo en el apartado anterior: se han generado distintos conjuntos de observaciones correspondientes a distintas configuraciones del vector \mathbf{b} . En concreto, estas configuraciones son:

- $\mathbf{b}_1 = [001000100000000000000000000000]$

- $\mathbf{b}_2 = [001000100000100001001000000100]$

- $\mathbf{b}_3 = [001000100000101001001000000101]$

- $\mathbf{b}_4 = [101110100100101001011000000101]$

Con estos conjuntos de observaciones se ha calculado el número de errores obtenido con ambos algoritmos. Además, al igual que en el apartado anterior, esta comparación se ha hecho para distintos valores de N . En concreto, se ha escogido $G = 30$ y $N = 7$, $N = 10$, $N = 15$, $N = 30$.

En la siguiente tabla se muestra el número de errores cometidos con el algoritmo VBEM y con el VBEM con ligaduras cuando la desviación típica del ruido del modelo es 0.1 y para un total de 100 realizaciones.

Condiciones	Variable \mathbf{b}	Errores VBEM	Errores VBEM ligaduras
G=30,N=7	\mathbf{b}_1	2	1
G=30,N=7	\mathbf{b}_2	743	620
G=30,N=7	\mathbf{b}_3	904	763
G=30,N=7	\mathbf{b}_4	1183	1141
G=30,N=10	\mathbf{b}_1	0	0
G=30,N=10	\mathbf{b}_2	328	283
G=30,N=10	\mathbf{b}_3	563	545
G=30,N=10	\mathbf{b}_4	1126	1106
G=30,N=15	\mathbf{b}_1	0	0
G=30,N=15	\mathbf{b}_2	0	0
G=30,N=15	\mathbf{b}_3	97	87
G=30,N=15	\mathbf{b}_4	861	818
G=30,N=30	\mathbf{b}_1	0	0
G=30,N=30	\mathbf{b}_2	0	0
G=30,N=30	\mathbf{b}_3	0	0
G=30,N=30	\mathbf{b}_4	0	0

Como puede verse, el número de errores cometidos por el algoritmo VBEM con ligaduras es siempre menor al que comete el algoritmo VBEM. Además el número de iteraciones del algoritmo con ligaduras es siempre menor que el número de iteraciones del algoritmo VBEM con lo que se reduce la carga computacional además de mejorar el rendimiento.

A la vista de los resultados anteriores puede afirmarse que cuanto menor sea el número de muestras con el que se cuente, más se recomienda al uso de ligaduras para obtener mejores resultados.

Para estudiar cómo influye el conocimiento a priori en los resultados obtenidos, a continuación se muestra el número de errores para distintas configuraciones de la distribución de probabilidad a priori. Estos resultados corresponden a 100 realizaciones, la desviación típica del ruido es 0.1, $N = 7$, $G = 30$ y el número de unos en el vector \mathbf{b} es igual a 12.

Algoritmo	Número Errores	Número de iteraciones
VBEM sin ligaduras	1250	2700
VBEM $P_{ave} = 11P_{max} = 16$	1220	2414
VBEM $P_{ave} = 8P_{max} = 12$	1234	2555
VBEM $P_{ave} = 10P_{max} = 14$	1222	2414
VBEM $P_{ave} = 11P_{max} = 15$	1220	2414
VBEM $P_{ave} = 12P_{max} = 15$	1215	2345
VBEM $P_{ave} = 12P_{max} = 16$	1218	2414

Como puede verse, los resultados del algoritmo VBEM con ligaduras son siempre mejores que los del algoritmo VBEM sin ligaduras. Naturalmente, cuanto mejor sea el conocimiento a priori que se posea, mejores serán los resultados pero en cualquier caso, aunque los resultados no sean los óptimos porque el conocimiento a priori no sea tan bueno, la reducción del número total de iteraciones es importante.

7.8.3. Estimación de los parámetros del canal

En esta subsección se presentan los resultados obtenidos para los parámetros de canal estimados. El proceso de estimación de estos parámetros es muy sencillo, una vez que se ha realizado la detección se usan esos valores detectados de la variable latente para estimar los parámetros utilizando un detector MMSE (Minimum Mean Squared Error). Los resultados que se presentan se ha obtenido con 100 realizaciones del algoritmo VBEM para el caso en que $G = 30$ y distintos valores de N . Se ha estimado el vector de parámetros y se ha calculado el error de este vector comparándolo con el vector real de parámetros. Este proceso se ha realizado a su vez para dos configuraciones diferentes del vector \mathbf{b} . Los resultados de estas simulaciones pueden verse en la Figura 7.4. Como puede verse este error está muy cercano a cero incluso para casos en los que N es menor que G .

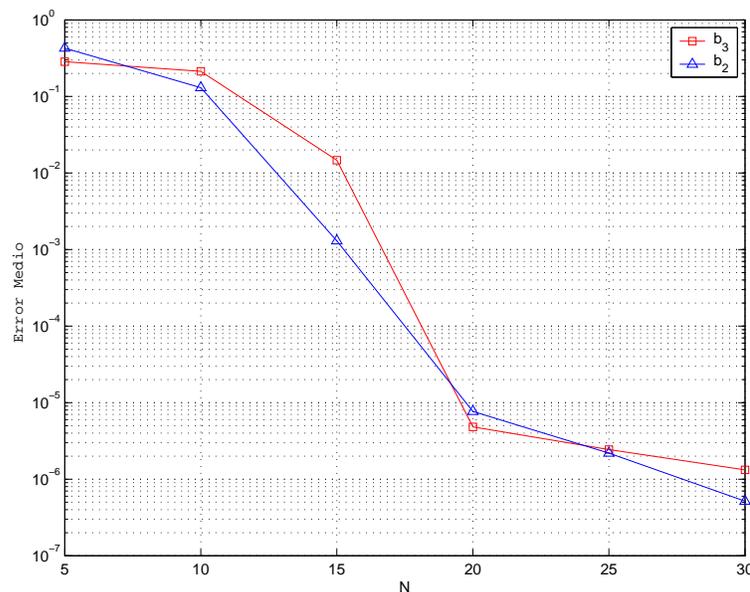


Figura 7.4: Errores en los parámetros del canal para $G = 30$ cuando \mathbf{b}_3 y \mathbf{b}_2

7.9. Conclusiones

En este capítulo se pretendía desarrollar un algoritmo para resolver un problema de selección de variables dentro de un marco Bayesiano. Como ya se ha comentado, la selección de variables es un método que se aplica a situaciones en las que se pretende modelar relaciones entre una determinada salida y un conjunto de variables explicatorias cuando existe indeterminación sobre qué conjunto debe usarse. En particular, se estaba interesado en desarrollar un algoritmo en marco Bayesiano que proporcionara información estadística que pudiera refinarse en una segunda etapa aplicando un algoritmo de tipo PDA. Además existían dos tipos de limitaciones impuestas por el modelo que usamos para el sistema a tratar. La primera limitación está relacionada con el hecho de que los parámetros del sistema que estudiamos son desconocidos. Este primer problema ha sido solucionado por el algoritmo VBEM como puede verse en las simulaciones ya que estima iterativamente estos parámetros. Además como se ha mostrado la complejidad computacional de este algoritmo es mucho menor que la del algoritmo de Gibbs Sampling. Por otro lado, se ha propuesto una solución en la que se realizan de forma conjunta el aprendizaje de las variables \mathbf{b}_i y la estimación de los parámetros del modelo. El segundo problema al que había que hacer frente al principio de este capítulo venía del hecho de que trabajamos con sistemas en los que el tamaño de las observaciones es mucho menor que el número de variables latentes que se pretenden detectar. Para hacer frente a este problema, se ha propuesto la introducción en el algoritmo VBEM de ciertas ligaduras que permitan enfrentarse a este problema reduciendo el espacio de búsqueda. Los resultados de las simulaciones muestran que este algoritmo VBEM con ligaduras presenta un mejor rendimiento que el algoritmo VBEM tanto para sistemas indeterminados como sobredeterminados y que su complejidad computacional es menor.

7.10. Trabajo futuro

Como continuación de este trabajo se está terminando el desarrollo en este momento un algoritmo de tipo PDA para aplicar al final del algoritmo VBEM utilizando la información estadística que este proporciona. El principal propósito es la mejora de la APP que el VBEM proporciona. Este algoritmo está prácticamente desarrollado y en este momento estamos solamente a la espera de optimizar los programas que se han utilizado para obtener simulaciones y poder comparar su rendimiento con el del algoritmo VBEM.

En una segunda fase se pretende desarrollar un algoritmo de turbo detección dentro del cual se integrará el algoritmo presentado en este capítulo de la tesis. Los algoritmos turbo han sido utilizados con gran éxito dentro del campo de las comunicaciones y son capaces de aprovechar la información estadística que proporciona el algoritmo propuesto. El modelo de sistema estudiado en este capítulo es muy general y podría corresponder por ejemplo a un sistema de comunicaciones multiusuario CDMA. Es por esto que pensamos que la utilización de técnicas turbo aplicadas a este problema podría mejorar los resultados obtenidos. El resto de las motivaciones para utilizar este tipo algoritmos serán expuestas con más detalle en el capítulo siguiente.

Capítulo 8

Aplicación del algoritmo VBEM a Redes Reguladoras de Genes

Las Redes Reguladoras de Genes son colecciones de relaciones reguladoras gen-gen en el genoma y son modelos que muestran las relaciones causales entre las actividades de los genes. La visión a nivel de sistema de las funciones de los genes que proporcionan las Redes de Genes es de tremenda importancia en el descubrimiento de los procesos biológicos que tienen lugar en todos los seres vivos. Este conocimiento puede proporcionar nuevas ideas relacionadas con el tratamiento de distintas enfermedades o el diseño de nuevas medicinas.

Actualmente la comunidad investigadora en el campo del procesado de la señal está dedicando una gran cantidad de esfuerzos a la aplicación de técnicas que tradicionalmente han sido desarrolladas con vistas a su aplicación en comunicaciones al campo de la bioinformática [40, 45, 73, 77].

En este capítulo, se presenta una aproximación Bayesiana para construir Redes de Genes basadas en datos de microarray. Los datos de microar-

ray proporcionan abundante información sobre los niveles de expresión de ARN mensajero de cientos de genes. En particular, se está interesado en una formulación del problema como un proceso de selección de variables, esto es, la determinación de qué genes regulan a cada gen utilizando una regla Bayesiana de aprendizaje. El principal problema en este proceso es el relacionado con el tipo de datos con los que se cuenta ya que contienen niveles de transcripción de miles de genes mientras que los conjuntos de datos, lo que serán nuestras observaciones, no son más que como mucho un par de docenas de muestras. Así, habrá que hacer frente tanto a problemas relacionados con la complejidad computacional así como con el significado de la red resultante [76], [46]. Recientemente se ha abordado el problema relacionado con el pequeño tamaño del conjunto de muestras utilizando “Reversible jump Markov Chain Monte Carlo” (RJMCMC) [46]. La idea es descubrir Redes Reguladoras de genes restringiendo el número de padres y diseñando cuidadosamente la topología de las probabilidades a priori. Este algoritmo se ha probado en datos de microarray de hongos en el proceso de división de la célula y proporcionó resultados más precisos que otros algoritmos deterministas. RJMCMC fue propuesto por Green en [38] y es un algoritmo de tipo “Markov Chain Monte Carlo” (MCMC) que aproxima una solución de tipo “Maximum a Posteriori (MAP)”. Sin embargo, el principal problema que presenta es el hecho de que necesita utilizar un gran número de muestras para producir resultados robustos y precisos. Esto hace que este tipo de algoritmos sea normalmente computacionalmente costosos. Para solucionar este problema, se propone en este capítulo la utilización del algoritmo de tipo VBEM (Variational Bayesian Expectation Maximization) desarrollado en el capítulo anterior para realizar selección de variables en las Redes de Genes ya que ha demostrado ser especialmente adecuado para este tipo de situaciones. En concreto se pretende trabajar con muestras de datos pertenecientes al ciclo celular del hongo *Saccharomyces cerevisiae* obtenidas en dos condiciones diferentes. El estudio de este tipo de organismos eucariotas es atractivo por dos razones principales, la primera es la facilidad con la que pueden cultivarse y la segunda es el hecho de que su mapa genético comparte la compleja estructura celular interna de las plantas y los animales.

Este capítulo se estructura de la siguiente forma: en la Sección 8.1 aparecen explicadas las principales bases biológicas relacionadas con los procesos que se estudian. En la Sección 8.2 se comentan las principales funciones de los Ácidos Nucleicos. Las técnicas de Microarray utilizadas para obtener los datos que se utilizan en esta parte de la tesis se explican en la Sección 8.3. En la Sección 8.4 se presenta la formulación del problema desde un punto de vista matemático. En la Sección 8.5 se describen los datos utilizados en las simulaciones. En la Sección 8.6 se propone un esquema de integración de

los datos de microarray procedentes de dos fuentes distintas con los que se cuenta. Finalmente, en la Sección 8.7 se presentan los resultados obtenidos al aplicar el algoritmo VBEM propuesto a los datos de microarray y en la Sección 8.8 se comentan las principales conclusiones. El trabajo futuro a desarrollar en este campo se propone en la Sección 8.9.

8.1. Bases Biológicas

El conocimiento de los ácidos nucleicos ha sido el motor que más ha impulsado la Biología de los últimos cincuenta años. Para entender este desarrollo es necesario fundamentar los conocimientos que han conducido hasta la conformación de la Biología Molecular actual, para lo cual es imprescindible conocer el soporte estructural en que se basa.

Los ácidos nucleicos son sustancias de carácter ácido que se detectaron en primer lugar en el núcleo celular, de ahí su nombre.

Desde finales del siglo diecinueve, se sabía que la información genética que es transferida de una generación a la siguiente debía estar en el núcleo, de hecho, éste es la parte más voluminosa del espermatozoide y la única que se transfiere íntegramente al óvulo para la formación de un nuevo organismo. Sin embargo, hasta los años cincuenta del siglo pasado se suponía que los ácidos nucleicos tenían una estructura muy repetitiva, por lo que se pensaba que información genética era transportada por las proteínas nucleares, en las cuales las distintas posibilidades de ordenación de los aminoácidos a lo largo de la cadena polipeptídica podrían dar la variabilidad necesaria para la constitución de diferentes estructuras en un organismo. Las experiencias de Avery y sus colaboradores, determinaron, sin ningún género de dudas, que los ácidos nucleicos eran los responsables del mensaje genético, aunque no se podía suponer de qué modo estaba contenida esta información.

La hipótesis de Watson y Crick sobre la estructura del ADN proporcionó un modelo estructural con una variabilidad basada en las distintas ordenaciones de cuatro grupos moleculares denominados bases nitrogenadas, emparejadas de dos en dos según cuatro combinaciones posibles (A-T, T-A, C-G, G-C). Si la cadena tuviese diez posiciones de dichos pares de bases, el número de posibilidades de ordenaciones distintas sería de 4^{10} .

Obviamente, el conocimiento de este modelo cambió las expectativas de estudio de la Genética Molecular y abrió un abanico de posibilidades para la comprensión de los fenómenos biológicos.

Desde los años cincuenta a los ochenta, el principal reto de esta rama

de la ciencia ha sido el de establecer los principios básicos que rigen esta información (replicación, transcripción y traducción). En la actualidad, las investigaciones van dirigidas a determinar cómo la secuencia de nucleótidos que se traduce en una estructura tridimensional concreta del organismo, lo que se conoce como morfogénesis.

8.1.1. Composición de los Ácidos Nucleicos

Existen dos tipos de ácidos nucleicos: el ADN (ácido desoxirribonucleico) y el ARN (ácido ribonucleico), cuya composición se representa en la Figura 8.1.

Ambos se forman por la polimerización de unidades moleculares sencillas llamadas nucleótidos. Éstos, a su vez, están constituidos por la unión de un nucleósido al ácido fosfórico. Los nucleósidos resultan de la unión de un azúcar (pentosa) a una base nitrogenada. Ese azúcar puede ser la D-ribosa, que forma ARN, o la 2-desoxi-D-ribosa, que forma ADN. La base nitrogenada puede derivar de la purina (bases púricas) o la pirimidina (bases pirimidínicas). Las bases púricas son la adenina y la guanina; las bases pirimidínicas son la citosina, la timina y el uracilo. De las dos últimas la timina sólo puede formar ADN, mientras que el uracilo es específico del ARN. Los nucleótidos se forman por la unión de un carbono de la pentosa con el grupo fosfato. De este modo, se forman los nucleótidos monofosfato, pero el grupo que lleva el fósforo puede enlazar un segundo o un tercer fosfato dando lugar al nucleótidos difosfato y el trifosfato, respectivamente. Los nucleótidos pueden enlazarse de un modo encadenado para formar los polinucleótidos o ácidos nucleicos. La cadena se forma al enlazar un carbono de la pentosa con el grupo fosfato de otro nucleótido; de este modo, se obtendrá un dinucleótido que, a través de su pentosa o de su grupo fosfato, según el radical no utilizado en el enlace, podrá incorporar otro nucleótido para constituir un trinucleótido, el cual, a su vez, puede formar un tetranucleótido y así sucesivamente. Salvo casos especiales, los polinucleótidos contienen un sólo tipo de pentosa: D-ribosa en el ARN y 2-desoxiribosa en el ADN.

8.1.2. ADN

Los componentes del ADN son los nucleótidos; cada nucleótido está formado por un grupo fosfato, una desoxirribosa y una base nitrogenada. Existen cuatro bases: dos purínicas (o púricas) denominadas adenina (A) y guanina (G) y dos pirimidínicas (o pirimidínicas) denominadas citosina (C) y timina (T). La estructura del ADN es una pareja de largas cadenas de nucleótidos. La estructura de doble hélice (ver figura) del ADN no fue descubierta hasta

1953 por James Watson y Francis Crick y establece que una larga hebra de ácido nucleico está enrollada alrededor de otra hebra formando un par entrelazado. El rasgo fundamental es que cada base nitrogenada de una hebra “casa” con la base de la otra, en el sentido de que la adenina siempre se enfrenta a la timina (lo que se denomina A-T) y la guanina siempre a la citosina (G-C). Estas uniones se producen a través de enlaces de puentes de hidrógeno. Los puentes de hidrógeno son uniones más débiles que los típicos enlaces químicos, tales como interacciones hidrófobas, enlaces de Van der Waals, etc... Esto significa que las dos hebras de la hélice pueden separarse con relativa facilidad, quedando intactas.

El modelo de doble hélice permite explicar las propiedades que se esperan del ADN:

- Capacidad para contener información: lenguaje codificado en la secuencia de pares de nucleótidos
- Capacidad de replicación: dar origen a dos copias iguales
- Capacidad de mutación: justificando los cambios evolutivos

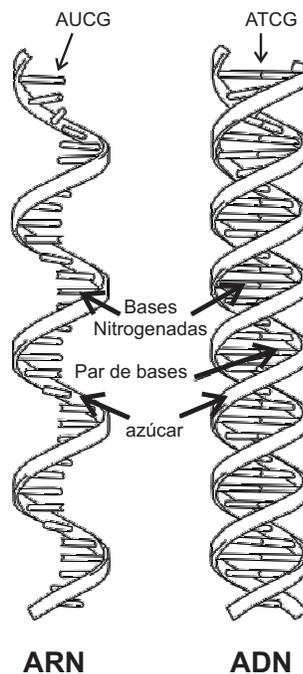


Figura 8.1: Estructuras del ARN y ADN

8.1.3. ARN

La estructura del ARN es muy similar a la del ADN. Como ya se ha comentado anteriormente, las diferencias estriban en que el ADN posee timina en vez de uracilo, que es la base propia del ARN. Mucha mayor trascendencia presenta el hecho de que el ADN está formado por 2-desoxi-D-ribosa, mientras que el ARN está formado por D-ribosa. Esta circunstancia implica que en la cadena base del ARN exista un OH en posición 2' que dificulta la formación de dobles hélices. Sin embargo, en determinadas condiciones, éstas pueden llegar a formarse. En ciertas ocasiones se forman pares híbridos constituidos por una cadena de ADN y otra de ARN. En estos casos el apareamiento de bases es el mismo que en las dobles hélices de ADN, pero la base que se enfrenta a la adenina en el ARN es el uracilo.

Existen distintos tipos de ARNs que se clasifican en función de la actividad biológica que desarrollan. Entre ellos cabe destacar: los ARN mensajeros (ARNm), los ribosómicos (ARNr), los de transferencia (ARNt) y los distintos ARN nucleares (ARNhn).

8.1.4. ARN heteronuclear (ARNhn)

Al margen de los precursores de los ARNm, ARNr y ARNt, en el núcleo se pueden aislar algunos otros ARN, que aparecen transitoriamente. Entre ellos cabría destacar algunas estructuras mixtas de ADN y ARN que se forman durante el proceso de duplicación del ADN. También pertenecen a este grupo ciertos ARN implicados en la maduración de ARNm, algunos de los cuales tiene, sorprendentemente, función catalítica, pueden cortar y empalman diversos segmentos del transcrito primario (precursor del ARNm).

8.1.5. ARN mensajero (ARNm)

El ARNm se forma utilizando como molde biológico algunas secuencias de ADN mediante el principio de complementariedad de bases. El ARN así formado se llama transcrito primario y, posteriormente, sufre diversas modificaciones en un proceso denominado maduración, después del cual, y en las células eucariotas, pasa al citoplasma donde se traduce a proteína. Los transcritos primarios constituyen una parte importante de los ARNhn y entre las principales modificaciones que sufren para transformarse en ARNm se encuentran operaciones de corte y empalme, a nivel molecular, según las cuales determinadas secuencias denominadas intrones son eliminadas. El reconocimiento de estos tramos y la operación de corte y empalme es realizada por algunos ARNhn.

8.1.6. ARN ribosómico (ARNr)

El ARN ribosómico forma parte de unas estructuras citoplasmáticas denominadas ribosomas, cuya función es la de leer los ARNm y formar, en consecuencia, proteína correspondiente. Los ARNr se construyen en una estructura nuclear denominada nucleolo. Algunos de estos ARNr poseen estructuras muy complejas con diversos tramos en forma de doble hélice, pero intracatenario, debido a las uniones interiores entre ribonucleótidos con secuencias complementarias.

8.1.7. ARN de transferencia (ARNt)

Los ARN de transferencia son cadenas cortas y con una estructura básico común. Del mismo modo que los ARNr, los ARNt forman parte del complejo de traducción y poseen la peculiaridad de unirse específicamente a determinados aminoácidos, de tal manera que cada uno de los 80 ARNt sólo se pueden enlazar a un aminoácido. Los ARNt poseen entre 70 y 90 bases, algunas de las cuales están modificadas. Si se despliegan parcialmente poseen forma de hoja de trébol con cuatro tramos en doble hélice y tres asas, de las cuales una contiene un triplete denominado triplete del anticodón a través del cual el ARNt lee el ARNm. Esta estructura en forma de hoja de trébol se pliega adquiriendo aspecto de una L invertida, en uno de cuyos extremos se encuentra el anticodón y, en el otro, la unión al aminoácido.

8.2. Funcionamiento de los Ácidos Nucleicos

Una vez establecida la estructura básica de los ácidos nucleicos es necesario señalar el papel general que desempeñan en el funcionamiento de la célula y, por extensión, del organismo. El mensaje genético se halla contenido en la o las cadenas de ADN. Para que la célula se divida es necesario que dichas cadenas se dupliquen (replicación); a continuación existe un reparto equitativo de cada una de las copias (división celular o mitosis).

Durante el periodo de tiempo que transcurre entre una mitosis y otra, el funcionamiento normal de la célula está dirigido por las proteínas. Para sintetizarlas, se forma una ARN transcrito primario (ARN_{hn}) utilizando como molde una de las dos cadenas de ADN y siguiendo el principio de complementariedad de las bases, proceso conocido como transcripción. A partir del transcrito primario se forma el ARNm, que se utilizará como molde para su traducción en proteínas. Durante esta fase los ribosomas con sus ARNr asociados señalan los segmentos de ARNm que se deben leer, y los ARNt aportan los aminoácidos que corresponden a dichos segmentos.

En un gen, la secuencia de los nucleótidos a lo largo de una hebra de ADN se transcribe a un ARN mensajero (ARNm) y esta secuencia a su vez se traduce a una proteína que un organismo es capaz de sintetizar o “expresar” en uno o varios momentos de su vida, usando la información de dicha secuencia. El promotor es una secuencia de ADN que permite que un gen sea transcrito, sirve para dar la señal de comienzo a la ARN polimerasa. El promotor ADN determina cuál de las dos cadenas de ADN será copiada.

La relación entre la secuencia de nucleótidos y la secuencia de aminoácidos de la proteína viene determinada por el código genético, que se utiliza durante el proceso de traducción o síntesis de proteínas. La unidad codificadora del código genético es un grupo de tres nucleótidos (triplete), representado por las tres letras iniciales de las bases nitrogenadas (por ej., ACT, CAG, TTT). Cuando estos tripletes están en el ARN mensajero se les llama codones. En el ribosoma cada codón del ARN mensajero interacciona con una molécula de ARN de transferencia (ARNt) que contenga el triplete complementario (denominado anticodón). Cada ARNt porta el aminoácido correspondiente al codón de acuerdo con el código genético, de modo que el ribosoma va uniendo los aminoácidos para formar una nueva proteína de acuerdo con las “instrucciones” de la secuencia del ARNm. Existen 64 codones posibles, por lo cual corresponde más de uno para cada aminoácido; algunos codones indican la terminación de la síntesis, el fin de la secuencia codificante; estos codones de terminación o codones de parada son UAA, UGA y UAG (en inglés, nonsense codons o stop codons).

En muchas especies de organismos, sólo una pequeña fracción del total de la secuencia del genoma codifica proteínas; por ejemplo, sólo un 3% del genoma humano consiste en exones que codifican proteínas. La función del resto por ahora sólo es especulación, es conocido que algunas secuencias tienen afinidad hacia proteínas especiales que tienen la capacidad de unirse al ADN (como los homeodominios, los complejos receptores de hormonas esteroides, etc.) que tienen un papel importante en el control de los mecanismos de transcripción y replicación. Estas secuencias se llaman frecuentemente secuencias reguladoras, y los investigadores asumen que sólo se ha identificado una pequeña fracción de las que realmente existen. El llamado ADN basura representa secuencias que no parecen contener genes o tener alguna función; la presencia de tanto ADN no codificante en genomas eucarióticos y las diferencias en tamaño del genoma representan un misterio que es conocido como el enigma del valor de C.

Algunas secuencias de ADN desempeñan un papel estructural en los cromosomas: los telómeros y centrómeros contienen pocos o ningún gen codificante de proteínas, pero son importantes para estabilizar la estructura de los cromosomas. Algunos genes codifican ARN: ARN ribosómico, ARN de

transferencia), ARN interferentes (ARNi, que son ARN que bloquean la expresión de genes específicos). La estructura de intrones y exones de algunos genes (como los de inmunoglobulinas y protocadherinas) son importantes por permitir cortes y armados alternativos del pre-ARN mensajero que hacen posible la síntesis de diferentes proteínas a partir de un mismo gen (sin esta capacidad no existiría el sistema inmunológico). Algunas secuencias de ADN no codificante representan pseudogenes que tienen valor evolutivo ya que permiten la creación de nuevos genes con nuevas funciones. Otros ADN no codificantes proceden de la duplicación de pequeñas regiones del ADN; esto tiene mucha utilidad ya que el rastreo de estas secuencias repetitivas permite estudios sobre el linaje humano

La secuencia también determina la susceptibilidad del ADN para ser cortado por determinadas enzimas de restricción, lo que se aplica en la realización de la técnica de RFLP, popularmente conocida como la Huella genética, que se usa para determinar la identidad y la paternidad de personas, aunque esta poderosa técnica también tiene aplicaciones en agricultura, ganadería y microbiología.

8.3. Técnicas de datos de Microarray

En este capítulo se aplican los algoritmos desarrollados en el capítulo anterior para estimar y caracterizar las Redes Reguladoras de genes así que se utilizará cierta nomenclatura característica de la biología que es conveniente definir. Además, los datos con los que se trabaja normalmente provienen de un tipo concreto de experimentos que es conveniente conocer, al menos de manera básica para poder entender bien tanto la relevancia como las posibles limitaciones de nuestros resultados.

En primer lugar, se hablará de *expresión de los genes* o de expresión de las proteínas para hacer referencia al proceso mediante el cual la secuencia de ADN de un gen da lugar a ciertas estructuras o funciones dentro de la célula. La expresión de un gen es un proceso multi etapa que empieza con la transcripción del ADN en ARNm y acaba con el transporte del producto (proteína) del gen que se ha creado. La cantidad de proteína que un célula expresa depende del tejido, del estado de desarrollo del organismo y de estados metabólicos o fisiológicos de la célula.

De forma indirecta, la expresión de determinados genes puede medirse con una tecnología denominada *Microarray de ADN* que puede proporcionar de forma aproximada medidas de la concentración celular de distintos ARN mensajeros [24], [63]. A este proceso suele denominársele obtención del perfil de expresión.

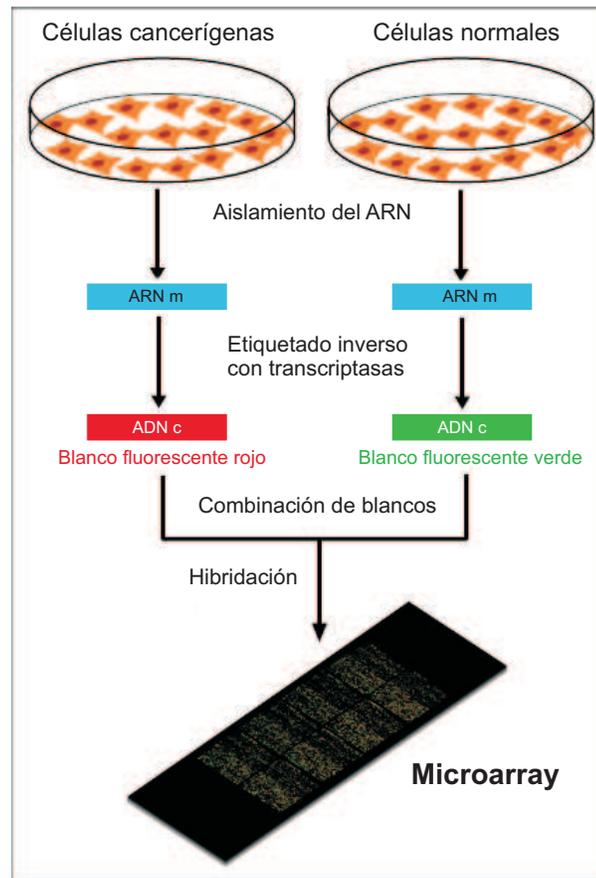


Figura 8.2: Proceso de obtención de un microarray para medir concentraciones de ARNm.

Recientemente, los importantes avances tecnológicos en hibridación de muestras (tendencia de unirse que presentan las moléculas complementarias o casi complementarias) así como los incesantes avances en la secuenciación del genoma, han dado lugar al desarrollo y mejora de técnicas de Microarray de ADN.

La tecnología de Microarray se basa en la *hibridación de ADN*: un proceso en el que la hebra de ADN se une a su única hebra complementaria. En este proceso, un conjunto de muestras de secuencias conocidas se fijan a una superficie y se ponen en contacto con un conjunto de blancos de distintas secuencias etiquetados con fluorescencia. Después de la hibridación, la fluorescencia indica la identidad de los blancos y la intensidad de la señal fluorescente está en correlación con la cantidad de cada blanco. Debido a las diferentes afinidades de hibridación entre los clones y el hecho de que una

cantidad desconocida de ADN complementario (ADNc) es constante en cada muestra, no podemos asociar directamente el nivel de hibridación con una cantidad de transcripción. En lugar de eso, los experimentos de microarray de ADNc comparan un conjunto de referencia con un conjunto de blancos de referencia. Típicamente, el verde se usa para etiquetar el conjunto de referencia, representando el nivel base de expresión y el rojo se usa para etiquetar la muestra blanco de interés (esto es, las muestras blanco pueden ser células de diferentes clases o células que se han tratado en ciertas condiciones de interés). Se híbrida la mezcla de los conjuntos de referencia y blanco y se obtiene una señal verde en los casos donde el nivel de expresión se reduce comparado con el conjunto de referencia y una señal roja en los casos donde el nivel de expresión aumenta en comparación con el nivel de referencia. Cuando se aplica la tecnología de microarray de ADN para medir niveles de ARNm, se realiza un proceso transcripción inversa en el que, a partir de los ARNm se obtiene ADNc. Este ADNc se híbrida entonces para obtener la secuencia de muestras de ADN del microarray. La razón por la que los niveles de ARNm no se miden directamente es que el medio está lleno de enzimas digestivas ARN de modo que el ARN que esté libre en el medio se degrada rápidamente mientras que el ADN es más estable y, por tanto, se usa para la hibridación. En la Figura 8.2 se muestra el proceso de obtención de un microarray para medir niveles de ARNm.

Existen diferentes tecnologías de microarray, las diferencias entre unas y otras pueden deberse a:

- lo que se imprime en los chips
- los protocolos que se emplean

Como ya se ha comentado, la medida de la transcripción de todo el genoma se llama perfil de expresión y está formado de medidas de niveles de ARNm de los genes en la célula. Biológicamente hablando, lo que medimos es cómo la expresión de cada gen cambia para desempeñar tareas complejas coordinadas para adaptarse a un medio cambiante. Mientras que los microarrays de ADN pueden medir cambios en los niveles de ARNm que son consecuencia directa de la regulación transcripcional, hay muchos aspectos celulares que no se pueden observar en los microarrays como por ejemplo la expresión de las proteínas o la actividad de las proteínas. Además, los datos medidos en los microarrays son ruidosos con los que pueden fluctuar de forma significativa incluso cuando el mismo experimento se repite.

Para obtener un rango amplio de perfiles de expresión, se usan distintos tipos de perturbaciones (como mutaciones [48] o shocks de calor [37]) así como diferentes tipos de células. Los resultados de estos experimentos son unas

matrices cuyas filas representan niveles de expresión diferentes genes y las columnas representan diferentes microarrays.

Finalmente apuntar que se están desarrollando otras tecnologías para muestrear distintos atributos de la célula a nivel del genoma. Estas incluyen nuevas tecnologías para medir interacciones proteína-proteína, complejos de proteína, niveles de abundancia de proteínas e interacciones entre el ADN y las proteínas. Todas las medidas anteriores son también de gran importancia para el estudio de la regulación entre genes.

8.4. Formulación del Problema

Consideremos un microarray que mide el nivel de expresión de G genes muestreados en $N + 1$ instantes de tiempo consecutivos. Se define una matriz de variable aleatoria $\mathbf{Y} \in \mathcal{R}^{G \times (N+1)}$ cuyo i, n ésimo elemento, $y_i(n-1)$, denota el nivel de expresión del gen i medido en el instante $n-1$. Además, se asume que el proceso de regulación es lineal y sigue un proceso de Markov de primer orden homogéneo en el tiempo. Por tanto, sólo es necesario considerar la relación reguladora entre dos instantes de tiempo consecutivos ya que esta relación permanece inalterada durante todo el proceso de microarray. Esta consideración puede ser insuficiente pero facilitará el modelado y la inferencia y constituye un primer paso.

En todo lo que queda, se llamará a los genes que regulan a uno dado padres de éste. Finalmente, se define para el gen i el vector binario columna $\mathbf{b}_i(n)$ de tamaño $G \times 1$ de la siguiente forma: el elemento j -ésimo de este vector es 1 si el gen j es padre del gen i y 0 en otro caso. El conjunto $\{\mathbf{b}_i(n), \forall i, n\}$ se trata como variables latentes que identifican la estructura de la Red Reguladora de Genes.

Teniendo en cuenta todo lo anterior, el proceso de regulación del gen i puede escribirse de la siguiente manera:

$$y_i(n) = \mathbf{w}_i \otimes \mathbf{y}(n-1) \otimes \mathbf{b}_i(n-1) + e_i(n), \quad n = 1, 2, \dots, N \quad (8.1)$$

donde \otimes representa el producto de Kronecker, $\mathbf{y}(n-1) = [y_1(n-1), \dots, y_G(n-1)]^\top$, $\mathbf{w}_i \in \mathcal{R}^{G \times 1}$ es el vector de pesos independiente del tiempo n y $e_i(n)$ es ruido blanco Gaussiano de varianza σ_i^2 . El vector de pesos es muy informativo y proporciona el grado y tipo de regulación, esto es, inhibición o activación. Para ser consistente con \mathbf{w}_i , $\mathbf{b}_i(n)$ es también invariante en el tiempo y se representa como \mathbf{b}_i .

Uno de los puntos a los que no se le ha dedicado mucha atención en este campo es el del significado del ruido en los modelos que se usan para

representar relaciones reguladoras entre genes. En nuestro caso, el ruido de la variable se introduce para incluir dos tipos de errores:

- errores de modelado
- errores experimentales

La ecuación (8.1) puede expresarse de forma equivalente y más compacta usando la siguiente representación matricial:

$$\mathbf{y}_i = \mathbf{R}\mathbf{W}_i\mathbf{b}_i + \mathbf{e}_i \quad (8.2)$$

donde $\mathbf{R} = \mathbf{T}\mathbf{Y}^\top$ con $\mathbf{T} = \begin{pmatrix} 1 & & 0 \\ & \ddots & \vdots \\ & & 1 & 0 \end{pmatrix}$, $\mathbf{y}_i = [y_i(1), \dots, y_i(N)]^\top$,

$\mathbf{W}_i = \text{diag}\{\mathbf{w}_i\}$, $\mathbf{b}_i = [b_i(1), \dots, b_i(G)]^\top$ y $\mathbf{e}_i = [e_i(1), e_i(2), \dots, e_i(N)]^\top$.

Dado el conjunto de datos del microarray \mathbf{Y} , el objetivo a perseguir es encontrar las Redes Reguladoras de genes, esto es, determinar \mathbf{b}_i para todo i . Además es muy importante hacer notar que tanto \mathbf{W}_i como σ_i^2 son desconocidos.

Si nos fijamos detenidamente en la expresión (8.2), se ve que el modelo del problema que se está tratando tiene exactamente la misma estructura que los que se encuentran en detección multiusuario y es esto precisamente lo que permite aplicar todo el conocimiento anterior relacionado con detección multiusuario a este nuevo problema. Por tanto, el objetivo ahora es desarrollar un algoritmo de tipo Bayesiano de forma que se logre una solución general y eficiente al problema propuesto.

En este proceso de búsqueda de la solución hay que tener en cuenta que tanto el canal, o lo que es lo mismo en este caso, los pesos, y la varianza del ruido son desconocidas. Por lo tanto, hay que introducir un paso de estimación de los mismos dentro de la solución Bayesiana general. Otro problema importante al que hay que hacer frente es el relacionado con el tamaño de las muestras, ya se ha comentado anteriormente que, de forma general, el tamaño del conjunto de muestras N es mucho menor que el número total de genes G . Un resultado directo de este hecho es que el problema se convierte en mal condicionado.

En este capítulo se pretende aplicar el algoritmo desarrollado en el capítulo anterior a los datos de microarray ya que se ha demostrado que es especialmente apropiado en los casos en que no se conocen ni los pesos ni la varianza del ruido. Además, es de esperar que concretamente el algoritmo VBEM con ligaduras proporcione buenos resultados al aplicarse a este tipo de observaciones debido al pequeño tamaño de las muestras en comparación

con el número de genes. El conocimiento a priori necesario en este algoritmo puede introducirse gracias a evidencias experimentales proporcionadas por diversos estudios de biólogos.

La necesidad de calcular la probabilidad a posteriori (APP) para resolver este problema hace que no puedan utilizarse la mayoría de los algoritmos que existen en la literatura y es por eso que se ha desarrollado un algoritmo nuevo. La información que suministra la APP es esencial ya que permite caracterizar la bondad de las estimaciones. Por esto, como segundo objetivo se propone investigar sobre la posibilidad de una solución basada en algoritmos de tipo PDA (Probabilistic Data Association). Esto está básicamente motivado por las similitudes que existen entre el modelo regulador lineal-Gaussiano y el modelo de comunicaciones multiusuario CDMA. Así, puede considerarse a cada gen como un usuario móvil en un sistema de comunicaciones multiusuario que transmite bits que pueden ser 0 o 1. Que el bit transmitido sea 1 significa que el gen regula al gen blanco y, por tanto, ambos genes pueden unirse en la red reguladora. Entonces, los datos de microarray se convierten en observaciones que son combinaciones ruidosas de lo que transmite cada gen escalado por cierto canal que nosotros vamos a llamar peso. El problema de descubrir las Redes Reguladoras de Genes es equivalente a identificar los bits transmitidos por cada gen haciendo uso de las observaciones. Recientemente se ha demostrado que los algoritmos de tipo PDA muestran un rendimiento óptimo con relativamente poca complejidad computacional. Por todo esto, puede afirmarse que este tipo de solución puede ser adecuada para aplicar al problema planteado. Sin embargo, una dificultad es que en este problema ni los pesos ni la varianza del ruido se conocen y, por tanto, deben estimarse dentro del PDA. Por es apropiado el desarrollo de una solución combinando PDA con un algoritmo variacional Bayesiano. Otro problema al que hay que hacer frente es el del pequeño tamaño de las muestras y el hecho de que en esta situación las matrices el el algoritmo PDA son singulares.

8.5. Datos utilizados

En este capítulo se aplica el algoritmo desarrollado en el capítulo anterior a datos de microarray, en concreto, se cuenta con dos tipos de datos procedentes de datos de microarray para procedentes del estudio del ciclo celular del hongo *Saccharomyces cerevisiae* bajo dos condiciones distintas.

Saccharomyces cerevisiae pertenece a la familia de los hongos. Es quizás la especie de hongo más importante ya que se ha utilizado desde la antigüedad para fermentar ya que es el responsable de la mayoría de los procesos de fermentación. Este tipo de hongo constituye el modelo de organismo celu-

lar eucariota más estudiado en biología molecular y celular, el nivel de su estudio es comparable al de la *Escherichia coli* como modelo de célula procarionota. El estudio del ciclo celular de este organismo es útil no sólo porque es fácil de cultivar sino también porque, como eucariota, comparte la compleja estructura celular interna de las plantas y los animales. Su genoma fue el primero que se secuenció de forma completa para organismos eucariotas. Este genoma se compone de unos 6275 genes pero se cree que sólo unos 5800 son verdaderos genes funcionales. Otra característica que hace interesante su estudio es que se ha estimado que estos hongos comparten un 23% de su genoma con el de los humanos.

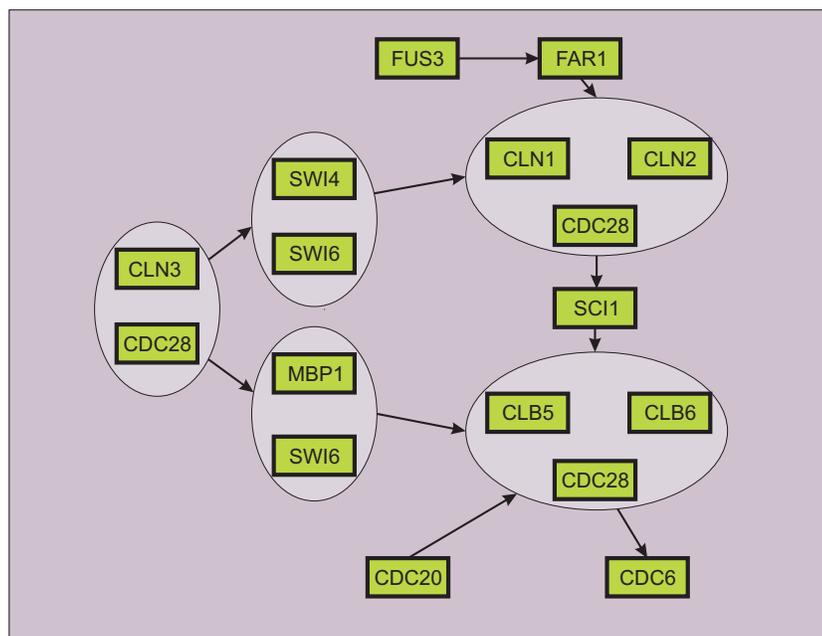


Figura 8.3: Red de Genes Real alrededor de CDC28

La relación entre los distintos genes que intervienen en el proceso ha sido estudiada con anterioridad y puede verse en las figuras (8.3), (8.4) y (8.5). Estas figuras se han construido con ayuda de la KEGG: Kyoto Encyclopedia of Genes and Genomes. Esta enciclopedia surge como respuesta al gran reto de lograr una representación completa por ordenador de las células, los organismos y la biosfera que permita la predicción computacional de los comportamientos complejos en células y organismos de la información genómica y molecular. Para lograr este objetivo, se ha desarrollado una base bioinformática llamada KEGG como parte de los proyectos de investigación de los laboratorios Kanehisa en el Bioinformatics Center de la Universidad de Kyoto y el Human Genome Center de la Universidad de Tokyo.

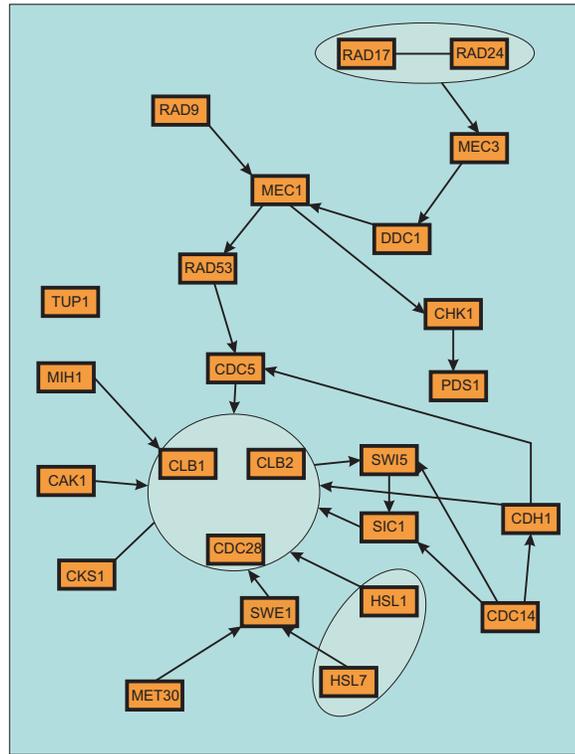


Figura 8.4: Red de Genes Real en la zona de comprobación de daños del ADN

Los mapas representados en esta enciclopedia son una colección que muestra el conocimiento de las interacciones moleculares y de las redes de reacciones correspondientes a: Procesos metabólicos, Procesado de la información genética, Procesado de la información del medio, Procesos celulares, Enfermedades humanas y Desarrollo de medicamentos.

8.6. Integración de datos

La elección de la probabilidad a priori de las variables latentes cuando los conjuntos de datos con los que se trabajan poseen sólo un número de muestras pequeño es un asunto un tanto peliagudo ya que a veces puede afectar bastante a los resultados del proceso de inferencia. Esta es precisamente la situación a la que los investigadores se enfrentan en el campo del procesamiento de señal aplicado a genómica. Una solución interesante es la que se ha propuesto en la Sección (7.4) del capítulo anterior. En este método se introducen una serie de ligaduras o restricciones en el diseño de la probabilidad

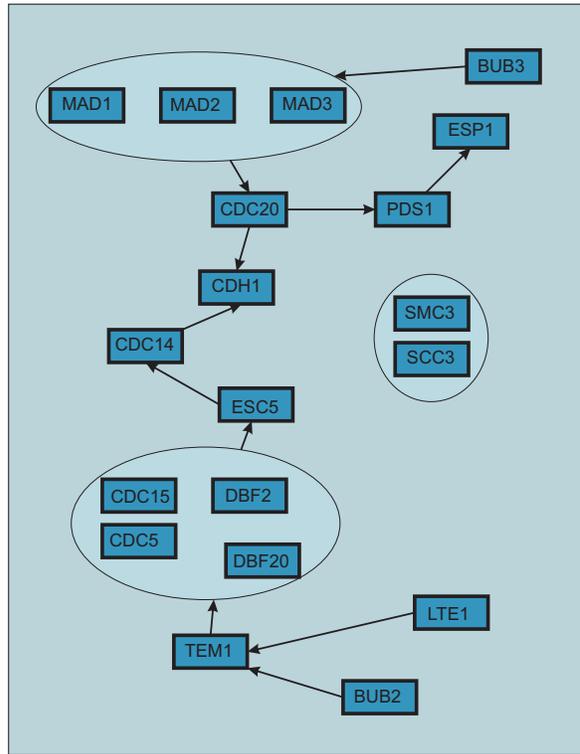


Figura 8.5: Red de Genes Real en la zona de control de giro

a priori de las variables latentes. Como resultado, el algoritmo con ligaduras proporciona mejores resultados que el algoritmo sin ligaduras como se ha mostrado en los resultados de simulaciones en 7.8.2 disminuyendo además la complejidad computacional pues reduce el espacio de búsqueda.

Es ampliamente aceptado dentro de la comunidad científica que los datos de microarray no pueden proporcionar por sí solos suficiente evidencia como para realizar una reconstrucción exacta de las Redes de Genes. Es necesario integrar otros tipos de datos junto a los de microarray como por ejemplo los procedentes de las interacciones entre proteínas. En cualquier caso el algoritmo desarrollado constituye un punto de partida acertado ya que la información estadística que proporciona puede usarse en este proceso de fusión de datos. Todo lo anterior justifica la elección que introducimos a continuación para aplicar el algoritmo propuesto en el capítulo anterior a aquellas situaciones especiales en las que se cuente con dos conjuntos de datos \mathbf{Y}^1 y \mathbf{Y}^2 de la misma Red de Genes tomados en diferentes condiciones experimentales. En esta nueva situación, el objetivo es encontrar las probabilidades a posteriori (APPs) $p(\mathbf{b}_i | \mathbf{y}_i^1, \mathbf{y}_i^2)$ del conjunto combinado de datos. Las probabilidades a posteriori anteriores pueden utilizarse entonces para realizar inferencia de la

estructura de la Red de Genes. La probabilidad $p(\mathbf{b}_i|\mathbf{y}_i^1, \mathbf{y}_i^2)$, puede calcularse usando el teorema de Bayes de la siguiente forma:

$$\begin{aligned} p(\mathbf{b}_i|\mathbf{y}_i^1, \mathbf{y}_i^2) &= \frac{p(\mathbf{y}_i^2|\mathbf{b}_i)p(\mathbf{y}_i^1|\mathbf{b}_i)p(\mathbf{b}_i)}{p(\mathbf{y}_i^1)p(\mathbf{y}_i^2)} \\ &= \frac{p(\mathbf{y}_i^2|\mathbf{b}_i)p(\mathbf{b}_i|\mathbf{y}_i^1)}{p(\mathbf{y}_i^2)} \end{aligned}$$

donde $p(\mathbf{y}_i^j|\mathbf{b}_i), \forall j$ son las funciones de probabilidad marginales, $p(\mathbf{b}_i|\mathbf{y}_i^j)$ son las APPs del conjunto de datos j . La ecuación anterior sugiere un esquema simple para integrar los dos conjuntos de datos con los que se cuenta. El procedimiento a seguir que se propone es el siguiente: en primer lugar se comienza con uno de los conjuntos de datos (es aconsejable comenzar con el que mayor conjunto de muestras posea), por ejemplo el \mathbf{Y}_1 . Entonces, se calculan las APPs $p(\mathbf{b}_i|\mathbf{y}_i^1)$ utilizando el algoritmo VBEM propuesto en el capítulo anterior. Después se usan esas $p(\mathbf{b}_i|\mathbf{y}_i^1)$ como información a priori identificándolas directamente con las probabilidades a priori correspondientes al conjunto de datos \mathbf{Y}_1 . Con este procedimiento se habrá integrado el conjunto de datos \mathbf{y}_i^1 con el conjunto \mathbf{y}_i^2 . De esta forma, se obtienen las APPs $p(\mathbf{b}_i|\mathbf{y}_i^1, \mathbf{y}_i^2)$ del conjunto de datos combinados aprovechando las APPs de la topología obtenidas con la solución VBEM. Este nuevo esquema proporciona un marco viable y eficiente para realizar una integración de datos Bayesiana que se propone como futura continuación de este trabajo.

8.7. Resultados

En esta sección se presentan los resultados obtenidos al aplicar el algoritmo propuesto en el capítulo anterior a datos de microarray de ADNc de *Saccharomyces cerevisiae*. Se cuenta con dos conjuntos de datos de 62 genes descritos en [78] and [20]. El conjunto descrito en [78] consta de 18 muestras tomadas cada 7 minutos durante un total de 119 minutos. En este conjunto se ha empleado un tratamiento de sincronización basado en el factor de apareamiento alpha. La duración total del ciclo celular en este primer caso es de 64 minutos. Por otro lado, el conjunto de datos de [78] posee 17 muestras tomadas cada 10 minutos durante un total de 160 minutos. Para sincronizar en este conjunto se ha usado un mutante del *CDC15* sensible a la temperatura. Finalmente apuntar que la duración total del ciclo celular en este proceso es de 85 minutos. Para cada gen, los datos se representan usando el $\log_2\{(\text{expresión en } t)/(\text{expresión en el conjunto de mezcla de células de control})\}$. En ambos conjuntos de datos hay valores perdidos lo que

indica que la señal utilizada no era suficientemente fuerte. En estos casos, se utilizó un método de interpolación con splines para rellenar los lugares con valores perdidos.

Desde	Hasta	APPs [78] + [20]	APPs [78]	APPs [20]
MET30	RAD9	0.9788	0.7602	0.8878
SWI4	SWI4	0.9691	0.5294	0.9523
CLN1	CLN3	0.9398	0.7909	0.6876
RAD9	CDC28	0.9382	0.7188	0.7177
HSL7	RAD53	0.9065	0.8553	0.5205
MET30	CDH1	0.8875	0.6617	0.6355
CDC6	CLB6	0.8797	–	–
MIH1	RAD53	0.8534	–	0.8424
CDC6	CDC28	0.8148	–	0.7999
RAD17	MEC3	0.8083	0.7584	–
MAD3	DBF20	0.8034	0.6358	–
MAD3	MAD1	0.7946	0.5234	–

Para mejorar los resultados del proceso de inferencia, se propone aplicar el algoritmo VBEM propuesto en el capítulo anterior a una combinación de los dos conjuntos de datos con los que se cuenta. Este procedimiento permite aprovechar la información contenida en ambos conjuntos de datos y, además, permite contrarrestar en la medida de lo posible la ausencia de algunos valores así como la pequeña cantidad de muestras de las que se dispone. El método que se ha usado es el descrito en la Sección 8.6. En concreto, en primer lugar se usa el conjunto de datos con mayor número de muestras (el descrito en [78]) construyendo la probabilidad a priori de las variables latentes como $P_{ave} = 2$ and $P_{max} = 5$. De la aplicación del algoritmo VBEM se obtiene el conjunto de APPs. Entonces esa información es introducida como probabilidad a priori para utilizar el algoritmo VBEM en el conjunto de datos descrito en ([20]).

En la Tabla (8.7) se muestran las APPs de algunas de las relaciones que se han encontrado con el algoritmo VBEM en tres situaciones diferentes: cuando se integran los dos conjuntos de datos y cuando se usan por separado cada uno de los conjuntos. Como puede verse, la integración de los dos conjuntos de datos hace que aumenten las APPs con lo que podemos concluir que el empleo de ambos conjuntos mejora los resultados. Además, con el uso de la integración de datos propuesta se han descubierto relaciones entre los genes que no se obtienen si se usa cada conjunto de datos por separado. Así, por ejemplo la conexión entre SWI6 y SWI4 aparece cuando se usa el conjunto de datos integrados o el conjunto de datos de ([78]) pero no aparece si se usa el conjunto de datos de ([20]). Otro ejemplo es relación entre CDC14 y CDH1.

Esta conexión sólo aparece cuando usamos el conjunto de datos integrados y no aparece cuando usamos los conjuntos de datos por separado.

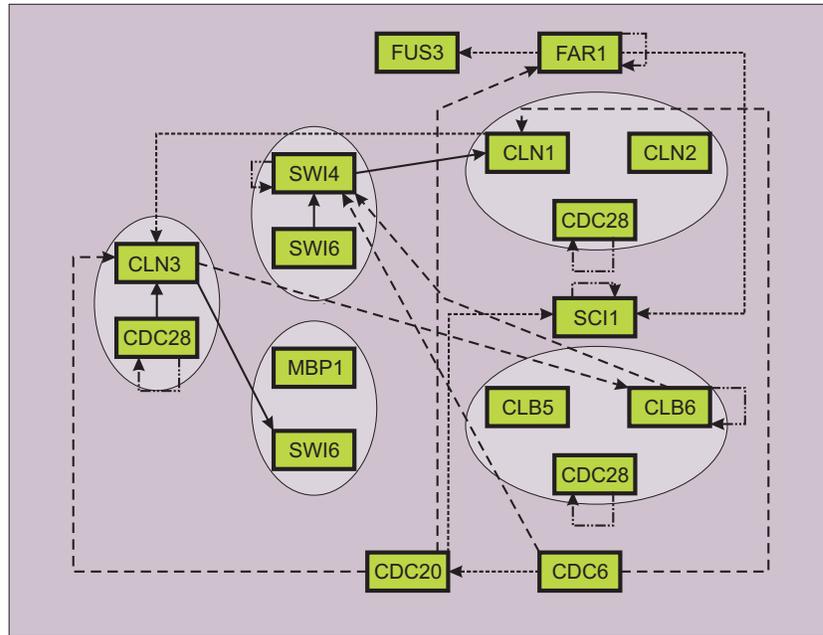


Figura 8.6: Red de Genes Estimada alrededor de CDC28 con el conjunto de datos sometidos al tratamiento con el factor de apareamiento alpha

A continuación se presentan las Redes de Genes estimadas con el algoritmo propuesto cuando se usan los conjuntos por separado y cuando se combinan. Estas Redes pueden compararse con las que proporciona el KEGG que aparecen representadas en las Figuras (8.3),(8.4) y (8.5). El mapa total se ha dividido en tres zonas sólo por claridad, para que sea más fácil apreciar las relaciones. Los elementos que aparecen en las representaciones rodeados por círculos se expresan conjuntamente con algún retardo. En cuanto a las distintas relaciones encontradas por el algoritmo VBEM, utilizamos una línea continua para aquellas relaciones confirmadas por el mapa de KEGG. Las líneas punteadas se usan para aquellas relaciones en las que bien se ha saltado un elemento intermedio como mucho y/o muestran un sentido de la relación erróneo. Puesto que este algoritmo pretende ser un primer paso de un algoritmo más completo, el principal interés es estudiar su habilidad para identificar relaciones. Es por eso que se considera importante distinguir aquellas relaciones en las que falta más de un elemento intermedio o/y poseen un sentido erróneo. Para ello se han utilizado líneas discontinuas. Finalmente, se han usado líneas formadas por sucesiones de puntos y rayas para representar las relaciones (generalmente de autorregulación, no confirmadas por el mapa

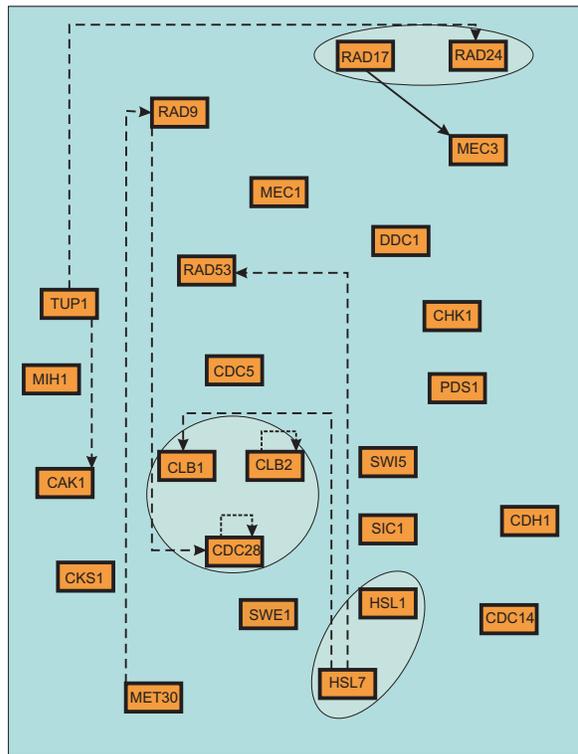


Figura 8.7: Red de Genes Estimada en la zona de comprobación de daños del ADN con el conjunto de datos sometidos al tratamiento con el factor de apareamiento alpha

KEGG).

En las figuras (8.6), (8.8) y (8.7) se representan las redes de genes estimadas cuando los datos usados han sido sometidos a un tratamiento de sincronización basado en el factor de apareamiento alpha. En la Figura (8.8) sólo se han representado aquellas relaciones con APP superior a 0.7 para lograr de nuevo mayor claridad en el gráfico.

En las figuras (8.9), (8.11) y (8.10) se representan las redes de genes estimadas cuando los datos usados han sido sometidos a un tratamiento de sincronización basado en el mutante cdc15. De nuevo, el mapa total se ha dividido en tres zonas sólo por claridad, para que sea más fácil apreciar las relaciones. En la Figura (8.10) sólo se han representado aquellas relaciones con APP superior a 0.7 al igual que en el resto de representaciones para poder hacer una comparación en igualdad de condiciones.

Las Redes de Genes que se han estimado al aplicar el VBEM al conjunto de datos integrados aparecen representadas en las figuras (8.12),(8.13) y (8.14). De nuevo, se representan los resultados obtenidos en tres gráficos

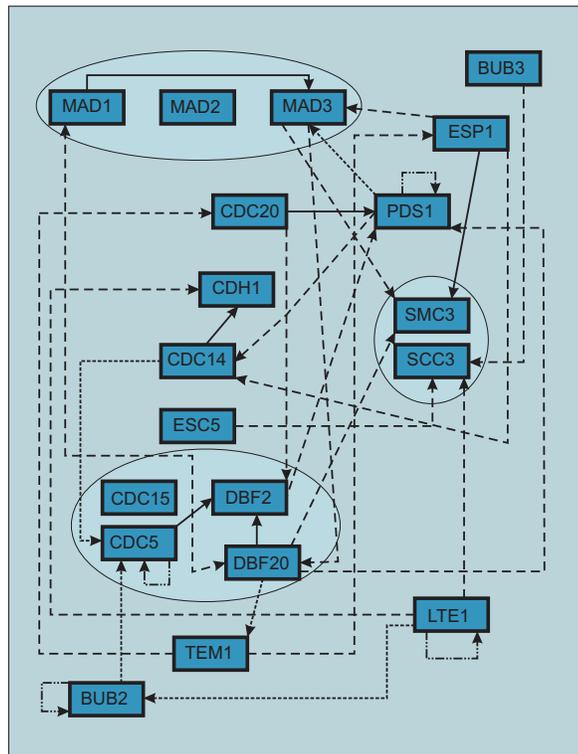


Figura 8.8: Red de Genes Estimada en la zona de control de giro con el conjunto de datos sometidos al tratamiento con el factor de apareamiento α

diferentes simplemente por claridad. Cada representación corresponde a una parte diferente del mapa.

El algoritmo VBEM que ha sido propuesto y desarrollado en esta tesis surgió con un doble propósito, en primer lugar se pretendía que fuera capaz de proporcionar información estadística sobre la variable latente. Con esta información estadística no sólo se es capaz de estimar el valor de esta variable latente sino que también es posible cuantificar la bondad de esta estimación. Pero además se perseguía la estimación de los parámetros que caracterizan el canal: los pesos. En el problema que se trata en este capítulo, estos pesos nos dan información a cerca del tipo de relación que existe entre los dos elementos de una conexión. Así pues, el algoritmo propuesto no es sólo capaz de predecir si existe relación o no y de cuantificarla, sino que también proporciona información sobre la naturaleza de la misma. A continuación se analizan algunas de estas relaciones en las diferentes figuras en función de tres aspectos: el primero es el sentido de la relación (representado por el sentido de la flecha), el segundo es la APP calculada por el algoritmo y el

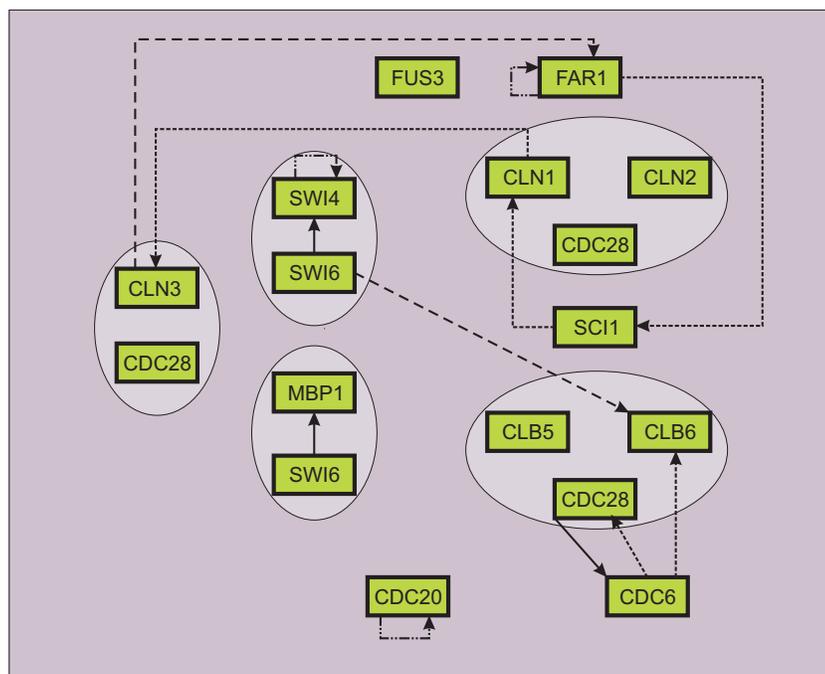


Figura 8.9: Red de Genes Estimada alrededor de CDC28 con el conjunto de datos sometidos al tratamiento con el mutante *cdc15*

tercero es la naturaleza de la relación caracterizada por el peso estimado por el algoritmo.

La Figura (8.12) se centra en el estudio de las relaciones en la zona del mapa alrededor de CDC28 (YBR160w; cyclin-dependent kinase). En esta representación, la conexión entre SWI4 y CLB1 se ha predicho correctamente y la APP calculada por el algoritmo VBEM es de 0.6757. El peso asociado a esta relación es 0.0633, su valor positivo indica un proceso de activación. Esta conclusión es correcta y puede verificarse en el mapa de KEGG.

En la Figura 8.13 puede verse la representación correspondiente a la parte de la red donde se encuentra la zona de comprobación de daños en el ADN. Por claridad, sólo aparecen representadas aquellas conexiones cuya APP es superior a 0.7. De nuevo, la Red de Genes estimada muestra algunas conexiones de naturaleza correcta, por ejemplo la unión entre el RAD17 y el MEC3 con APP igual a 0.8083 aparece en el sentido correcto y además tiene asociada un peso de 0.1503. De nuevo, un peso positivo nos indica activación y es precisamente este fenómeno el que aparece en el mapa de KEGG. Otro ejemplo es la conexión del CLB2 al SWI5, conexión con una APP de 0.7205, que se produce en el sentido correcto y con un peso de -0.1103. Un peso negativo es sinónimo de inhibición con lo que podemos afirmar que el algoritmo

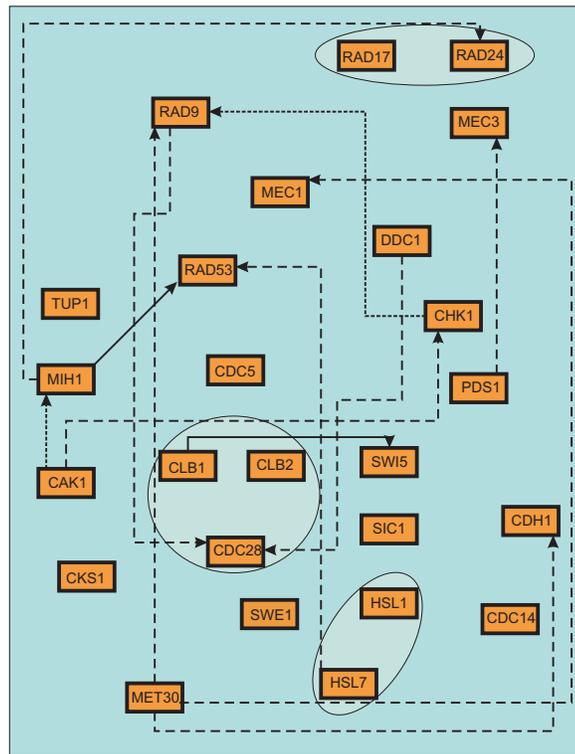


Figura 8.10: Red de Genes Estimada en la zona de comprobación de daños del ADN con el conjunto de datos sometidos al tratamiento con el mutante *cdc15*

propuesto es capaz en este caso de predecir correctamente todos los aspectos de la relación entre estos dos elementos de la red.

Finalmente, la Figura (8.14) muestra la Red de Genes estimada correspondiente a la parte del mapa del punto comprobación de giro. Un ejemplo de relación correcta predicha por el algoritmo en esta representación es la conexión desde el *ESP1* al *SMC3*, su *APP* es 0.7481 y el peso estimado por el algoritmo es -0.2183. De nuevo, el algoritmo propuesto predice un proceso de inhibición que se confirma por el mapa de KEGG.

8.8. Conclusiones

En este capítulo se ha aplicado el algoritmo VBEM desarrollado en el capítulo anterior a datos de microarray procedentes de dos experimentos en condiciones distintas del ciclo celular del hongo *Saccharomyces cerevisiae*. Ambos conjuntos de datos se han integrado a través de un esquema

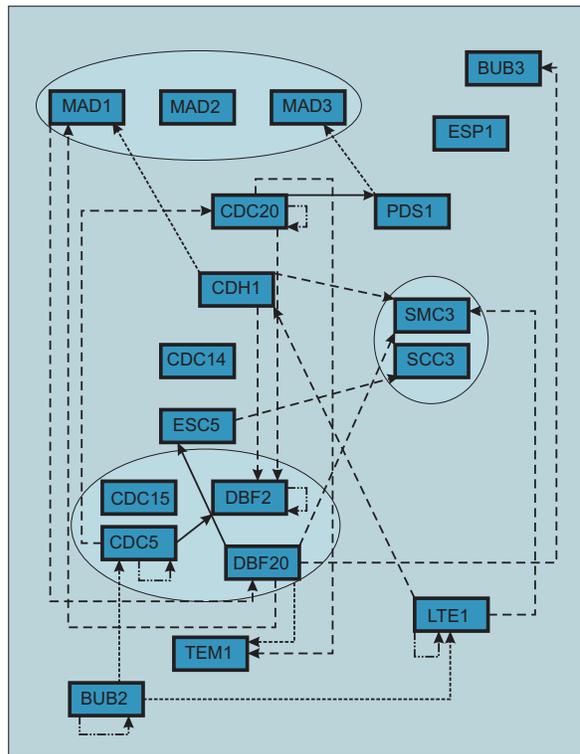


Figura 8.11: Red de Genes Estimada en la zona de control de giro con el conjunto de datos sometidos al tratamiento con el mutante *cdc15*

Bayesiano que se ha propuesto. Este proceso de integración se ha analizado a través de la comparación de las APPs que proporciona el algoritmo VBEM. Estas APPs se han comparado con las proporcionadas cuando se usan los conjuntos de datos por separado observándose que en todos los casos la probabilidad aumenta por tanto puede concluirse que el uso de este esquema de integración es más apropiado. Además, las APPs constituyen un indicador de la certidumbre de las estimaciones.

Además, la habilidad del algoritmo propuesto para introducir ligaduras ha hecho posible la reducción del espacio de búsqueda con lo que se ha reducido la complejidad.

Finalmente apuntar que otra de las importantes conclusiones a las que se llega en vista de los resultados de las simulaciones es que el algoritmo propuesto no es sólo capaz de predecir algunas de las relaciones entre los genes sino que es también capaz de predecir la naturaleza de las mismas.

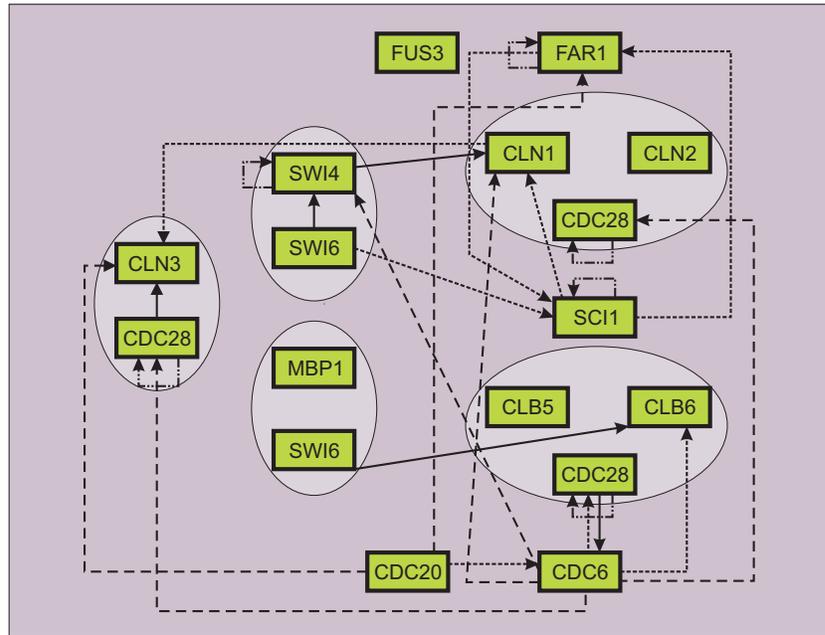


Figura 8.12: Red de Genes Estimada alrededor de CDC28

8.9. Trabajo Futuro

Uno de los principales problemas que se encuentran a la hora de trabajar con los datos de microarray está relacionado con el pequeño tamaño de las muestras temporales con las que se cuenta. Además, muchas veces estas muestras proceden de procesos de muestreo inadecuados de forma que se pierden detalles de algunos de los fenómenos de regulación que ocurren. Otros factores negativos que afectan a los datos de microarray son los relacionados con diversos defectos experimentales o incluso el ruido asociado a los resultados. Una posible solución a este problema consistiría en combinar de algún modo conjuntos de datos correspondientes al mismo proceso aunque tomados en condiciones diferentes. Pero no sólo podemos encontrar problemas con el muestreo. Otros problemas pueden ser el desfase temporal entre los conjuntos o incluso la diferencia de factores de escala entre unas muestras y otras. Por ello, se ha dedicado un gran esfuerzo investigador [7], [8] a combinar conjuntos de muestras procedentes de experimentos distintos. Existen diversas formas de atacar este problema en la literatura. Quizás una de las formas más extendidas para combinar distintos conjuntos de datos consiste en el modelado de los mismos como un proceso continuo haciendo uso de splines. En este capítulo se ha propuesto una primera aproximación al tratamiento de este problema dentro del marco Bayesiano y se ha demostrado con los

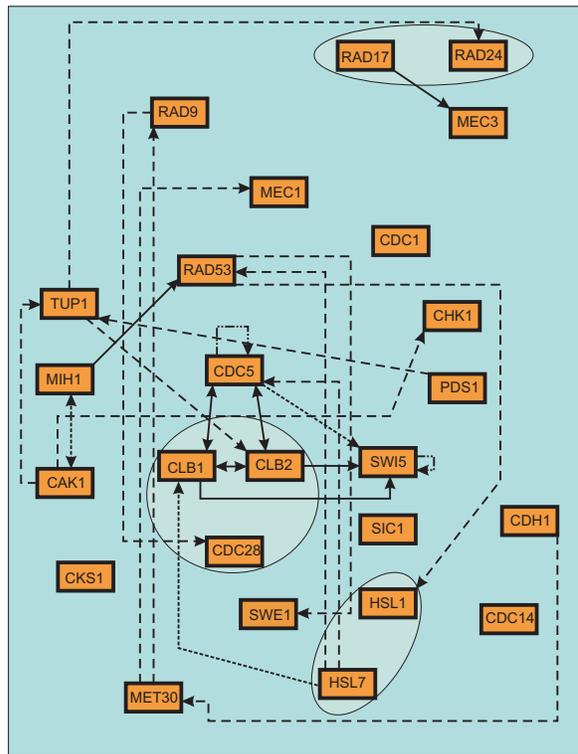


Figura 8.13: Red de Genes Estimada en la zona de comprobación de daños del ADN

resultados de las simulaciones la bondad del mismo.

Pero si queremos lograr buenos resultados al llevar a cabo un proceso de inferencia, no sólo debemos contar con datos de microarray sino que es conveniente poder combinar con estos datos otros como por ejemplo los procedentes de proteínas. Estos datos completarían la información relacionada con el proceso de regulación y proporcionarían una herramienta más potente para descubrir las distintas interacciones en las Redes de Genes. Este campo, al que se le está dedicando bastante atención en el mundo científico, constituye una interesante continuación de este trabajo ya que para obtener buenos resultados en el proceso de inferencia que nos planteamos es necesario contar con tantas fuentes de conocimiento como sea posible y ser capaces de integrar la información que proporcionan de forma adecuada. El algoritmo que se ha propuesto en esta tesis sólo constituye un buen primer paso hacia el desarrollo de un algoritmo Bayesiano que permita integrar datos procedentes de distintas fuentes. Como trabajo futuro se propone el desarrollo un algoritmo global mediante el cual pueda integrarse la información estadística de distintos conjuntos de datos. Así, se propone la aplicación en una primera

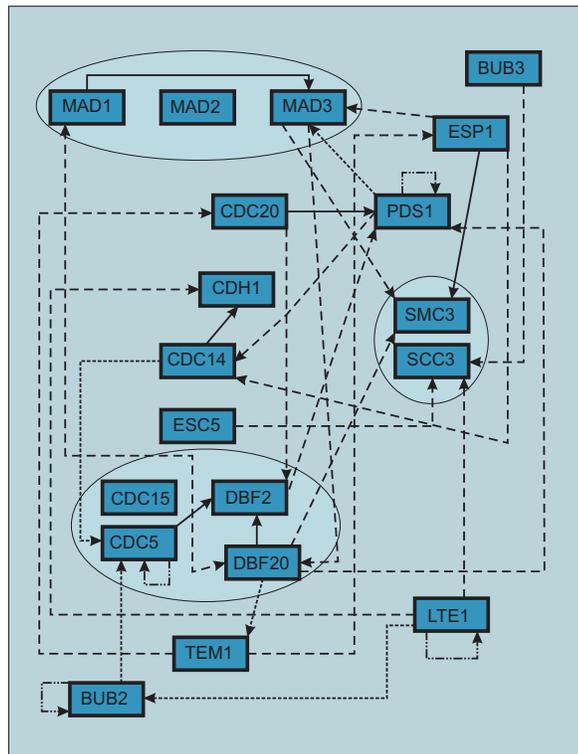


Figura 8.14: Red de Genes Estimada en la zona de control de giro

etapa el algoritmo de tipo PDA que se ha desarrollado para mejorar la información estadística que proporciona el algoritmo VBEM. En una segunda fase se propone investigar la aplicación de técnicas turbo de cara al proceso de integración.

Capítulo 9

Conclusiones y Futuras líneas de trabajo

Cada uno de los capítulos de esta tesis se ha terminado con una exposición de las conclusiones más relevantes del mismo. Además se ha incluido un apartado donde se proponían las más destacadas líneas de trabajo que se proponen como continuación de la investigación desarrollada en esta tesis. A continuación se exponen de nuevo tanto las conclusiones como las futuras líneas de trabajo de forma resumida.

9.1. Detección Multiusuario

En la primera parte de esta tesis se ha aplicado un algoritmo tipo Filtrado de Partículas a la Detección Multiusuario. Gracias a su estructura, este algoritmo posee una mejor complejidad computacional que los algoritmos de detección de tipo Filtrado de Partículas convencionales que existen hasta ahora, manteniendo el mismo rendimiento. Además de reducir la complejidad, este algoritmo es más eficiente debido a que el procedimiento que usa para

muestrear, es capaz de mantener la diversidad entre las distintas trayectorias que se escogen a diferencia de los algoritmos clásicos de Filtrado de Partículas. Las simulaciones muestran un hecho importante: el detector basado en el AME no se ve afectado en escenarios “near-far” ya que su rendimiento es parecido al que se obtiene en la situación en la que todos los usuarios tienen igual potencia.

En cuanto a las futuras líneas de investigación, debido a que este algoritmo aplicado a la detección ha demostrado su buena aplicabilidad, y sobre todo, debido a que proporciona información estadística, sería conveniente aplicarlo a turbo BLAST. El procesado turbo es una técnica iterativa que ya ha sido aplicada de forma satisfactoria a diversos problemas de detección y decodificación entre las que se encuentra la detección multiusuario [4], [60], [85], [92], [9], [59], [75], [80], [97]. Ya que el algoritmo propuesto puede suministrar la información estadística que podría ser aprovechada en una posterior etapa de detección turbo BLAST.

Otra posible línea de trabajo interesante sería la aplicación de este tipo de algoritmo a la detección ciega a técnicas OFDM (*Orthogonal Frequency Division Multiplexing*) [14], [91]. Esta línea de trabajo es muy interesante ya que el OFDM es una técnica bastante apropiada para la transmisión de datos a muy alta velocidad. Además, esta técnica ha sido adoptada como el estándar europeo para la difusión de audio de forma digital (DAB) [29] y la difusión de vídeo digital en sistemas terrestres (DVB-T) [31].

9.2. Detección y estimación de fase iterativa

En el Capítulo 6 de esta tesis se ha tratado el problema de la detección, decodificación y estimación de códigos LDPC, transmitidos a través de un canal afectado por ruido de fase. Se ha propuesto un algoritmo de tipo Filtrado de Partículas para resolver las integrales que aparecen cuando se aplica el algoritmo Suma-Producto a la representación Factor Graph del sistema. A continuación se enumeran las conclusiones que se derivan de la aplicación del algoritmo propuesto de Filtrado de partículas. En primer lugar, apuntar que al contrario que otras soluciones de tipo Filtrado de Partículas propuestas con anterioridad, este algoritmo no está limitado por la longitud del código utilizado. En el primer método de Filtrado de Partículas del que se habla en el Capítulo 6 [23], el número de partículas usado tiene que ser forzosamente un múltiplo de la longitud del código. Esta limitación es muy importante ya que si por ejemplo hubiésemos utilizado este algoritmo con el código que se ha usado en las simulaciones habríamos necesitado al menos 4000 partículas.

Otra importante ventaja es que la complejidad computacional del algo-

ritmo propuesto es mucho menor que la del algoritmo basado en una discretización equiespaciada del espacio de fases ya que el algoritmo propuesto evita la exploración de todos los estados. La solución equiespaciada es equivalente al algoritmo de Viterbi y su complejidad computacional es proporcional a $L \times L$, donde L es el número de estados discretos que consideran. La complejidad computacional del algoritmo propuesto es proporcional al producto del número de iteraciones y el número de partículas. Esta cantidad es en general mucho menor que la complejidad del algoritmo equiespaciado como se ha mostrado en las distintas simulaciones. Además, el algoritmo propuesto muestra un rendimiento similar a otros algoritmos propuestos en la literatura [21]. La ventaja del algoritmo propuesto sobre éstos es que proporciona una manera de estimar al fase de forma eficiente. Finalmente, apuntar como conclusión que el algoritmo de Filtrado de Partículas propuesto es prácticamente insensible al esquema de distribución de los símbolos piloto a diferencia del que propone un equiespaciado del espacio de las fases y de algunos de los propuestos en [21].

El siguiente objetivo a seguir en este campo de investigación es la aplicación del algoritmo propuesto a situaciones en las que el ruido cambie más rápido. El modelo que se ha utilizado para los cambios de de la fase es sólo apropiado en situaciones donde los cambios considerados son pequeños y corresponde a un desarrollo en primer orden del término exponencial. El siguiente paso sería la inclusión de más términos en este desarrollo de forma que el algoritmo resultante muestre un comportamiento al menos tan bueno como el del propuesto.

9.3. Algoritmo VBEM aplicado Genómica

En los Capítulos 7 y 8 de esta tesis se ha desarrollado un algoritmo variacional Bayesiano basado en una solución EM (Expectation Maximization). Este algoritmo es especialmente interesante porque es capaz de proporcionar información estadística a través de las APPs. En primer lugar se ha comparado el rendimiento de este algoritmo con otro algoritmo Bayesiano que también proporciona información estadística y que ha sido aplicado con anterioridad al mismo tipo de problemas: Gibbs sampling. A través de las simulaciones se ha demostrado que el algoritmo propuesto en esta tesis posee un mejor rendimiento que el algoritmo Gibbs sampling tanto para sistemas sobredeterminados como indeterminados. Otra ventaja del algoritmo propuesto en esta tesis es la reducción de la carga computacional que supone frente al algoritmo de Gibbs sampling. La tercera ventaja del algoritmo VBEM propuesto está relacionada con la información estadística que proporciona ya que

permite desarrollar un esquema de integración Bayesiano para combinar dos conjuntos de datos de microarray diferentes. Este esquema proporciona un marco viable y eficiente para realizar integración de datos Bayesiana que es de vital importancia en los casos con los que se trata debido al pequeño tamaño de las muestras con las que se cuenta.

En vista de los resultados obtenidos al aplicar el algoritmo propuesto a datos reales de microarray, otra importante conclusión es que el VBEM no es sólo capaz de estimar algunas de las relaciones entre los genes sino que también es capaz de estimar la naturaleza de las mismas a través de los pesos. Además, la comparación de las APPs calculadas al aplicar el algoritmo VBEM tras el proceso de integración y las calculadas sin integración muestra la relevancia de este proceso.

Uno de los principales problemas que se encuentran a la hora de trabajar con los datos de microarray está relacionado con el pequeño tamaño de las muestras temporales con las que se cuenta. Además, muchas veces estas muestras proceden de procesos de muestreo inadecuados de forma que se pierden detalles de algunos de los fenómenos de regulación que ocurren. Otros factores negativos que afectan a los datos de microarray son los relacionados con diversos defectos experimentales o incluso el ruido asociado a los resultados. Una posible solución a este problema consistiría en combinar de algún modo conjuntos de datos correspondientes al mismo proceso aunque tomados en condiciones diferentes. Pero no sólo podemos encontrar problemas con el muestreo. Otros problemas pueden ser el desfase temporal entre los conjuntos o incluso la diferencia de factores de escala entre unas muestras y otras. Por ello, se ha dedicado un gran esfuerzo investigador [7], [8] a combinar conjuntos de muestras procedentes de experimentos distintos. Existen diversas formas de atacar este problema en la literatura. Quizás una de las formas más extendidas para combinar distintos conjuntos de datos consiste en el modelado de los mismos como un proceso continuo haciendo uso de splines. En este capítulo se ha propuesto una primera aproximación al tratamiento de este problema dentro del marco Bayesiano y se ha demostrado con los resultados de las simulaciones la bondad del mismo.

Pero si queremos lograr buenos resultados al llevar a cabo un proceso de inferencia, no sólo debemos contar con datos de microarray sino que es conveniente poder combinar con estos datos otros como por ejemplo los procedentes de proteínas. Estos datos completarían la información relacionada con el proceso de regulación y proporcionarían una herramienta más potente para descubrir las distintas interacciones en las Redes de Genes. Este campo, al que se le está dedicando bastante atención en el mundo científico, constituye una interesante continuación de este trabajo ya que para obtener buenos resultados en el proceso de inferencia que nos planteamos es necesario contar

con tantas fuentes de conocimiento como sea posible y ser capaces de integrar la información que proporcionan de forma adecuada. El algoritmo que se ha propuesto en esta tesis sólo constituye un buen primer paso hacia el desarrollo de un algoritmo Bayesiano que permita integrar datos procedentes de distintas fuentes. Como trabajo futuro se propone el desarrollo un algoritmo global mediante el cual pueda integrarse la información estadística de distintos conjuntos de datos. Así, se propone la aplicación en una primera etapa el algoritmo de tipo PDA que se ha desarrollado para mejorar la información estadística que proporciona el algoritmo VBEM. En una segunda fase se propone investigar la aplicación de técnicas turbo de cara al proceso de integración.

Apéndice A

Lema de inversión de matrices

Supongamos que se tiene una matriz A de la que se conoce su expresión así como la de su inversa A^{-1} . Consideremos ahora una transformación sobre esta matriz A , de modo que se le suma cierta matriz de rango 1 para producir la matriz

$$B = A + \mathbf{xy}^H \tag{A.1}$$

Si se quiere calcular la expresión de la inversa de la matriz B , sería conveniente poder usar la información que se tiene sobre la inversa de A en lugar de hacer un cálculo complejo de la inversa de B directamente. Lo anterior se consigue usando la siguiente fórmula:

$$B^{-1} = (A + \mathbf{xy}^H)^{-1} = A^{-1} \frac{A^{-1}\mathbf{xy}^H A^{-1}}{1 + \mathbf{y}^H A^{-1}\mathbf{x}} \tag{A.2}$$

A la ecuación anterior se le conoce con el nombre de *lema de inversión de matrices*, *fórmula de Sherman-Morrison* o **identidad de Woodbury**.

La expresión anterior puede generalizarse. Supongamos que $A, B \in M_n$ y que

$$B = A + XRY \quad (\text{A.3})$$

donde X tiene dimensiones $(n \times r)$, Y tiene dimensiones $(r \times n)$ y R tiene dimensiones $(r \times r)$. En este caso, la inversa (si existe) puede calcularse haciendo uso de la siguiente fórmula

$$B^{-1} = (A + XRY)^{-1} = A^{-1} - A^{-1}X(R^{-1} + YA^{-1}X)^{-1}YA^{-1} \quad (\text{A.4})$$

Si r es lo suficientemente pequeño comparado con n , el cálculo de las inversas en la expresión anterior es mucho más fácil que un cálculo directo de la inversa de B .

Una identidad útil que puede derivarse de la expresión (A.4) es

$$(A + XRY)^{-1}XR = A^{-1}X(R^{-1} + YA^{-1}X)^{-1} \quad (\text{A.5})$$

Apéndice B

Inversa de una matriz de bloques

En este apartado se desarrolla el cálculo de la inversa de una matriz de bloque.

Si la matriz A puede escribirse de la forma:

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \quad (\text{B.1})$$

entonces su inversa puede escribirse de la siguiente forma

$$A^{-1} = \begin{bmatrix} A_{11}^{-1} + A_{11}^{-1}A_{12}S^{-1}A_{21}A_{11}^{-1} & -A_{11}^{-1}A_{12}S^{-1} \\ -S^{-1}A_{21}A_{11}^{-1} & S^{-1} \end{bmatrix} \quad (\text{B.2})$$

donde $S = A_{22} - A_{21}A_{11}^{-1}A_{12}$. A S se le conoce como complemento de Schur de A . La ecuación anterior puede validarse por multiplicación. Cuando A_{12} es invertible, el bloque $(1, 1)$ de A^{-1} puede escribirse también como $(A_{11} - A_{12}A_{22}^{-1}A_{21})^{-1}$ usando el lema de inversión de matrices.

En particular, para el caso en el que el bloque A_{21} es cero,

$$A^{-1} = \begin{bmatrix} A_{11} & A_{12} \\ \mathbf{0} & A_{22} \end{bmatrix}^{-1} = \begin{bmatrix} A_{11}^{-1} & -A_{11}^{-1}A_{12}A_{22}^{-1} \\ \mathbf{0} & A_{22}^{-1} \end{bmatrix} \quad (\text{B.3})$$

Otra forma útil de expresar la inversa de una matriz de bloques es la que se viene dada por el siguiente lema.

Lema 1. La inversa de una matriz A que puede escribirse según (B.1) tiene la siguiente expresión

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}^{-1} = \begin{bmatrix} A_{11}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} -A_{11}^{-1}A_{12} \\ I \end{bmatrix} S^{-1} \begin{bmatrix} -A_{21}A_{11}^{-1} & I \end{bmatrix} \quad (\text{B.4})$$

Prueba. La prueba es fácil, sólo hay que llevar a cabo las multiplicaciones y sumas que aparecen en la expresión anterior y se llega a (B.2).

Las fórmulas para calcular inversas de matrices por bloques se han usado en la obtención de las ecuaciones para el Filtro de Kalman. En lo que sigue se van a desarrollar los cálculos que se han omitido en la sección dedicada a los filtros de Kalman.

Sean \mathbf{X} e \mathbf{Y} vectores aleatorios con matrices de covarianza R_{xx} y R_{yy} respectivamente, y sea $\mathbf{Z} = [\mathbf{X}^T, \mathbf{Y}^T]$ otro vector aleatorio. Entonces,

$$\begin{aligned} R_{zz} &= \text{cov}(Z) = E \left[(\mathbf{Z} - \mu_z) (\mathbf{Z} - \mu_z)^T \right] \\ &= \begin{bmatrix} R_{xx} & R_{xy} \\ R_{yx} & R_{yy} \end{bmatrix} \end{aligned} \quad (\text{B.5})$$

Y si además se usa la expresión (B.2) queda

$$R_{zz}^{-1} = \begin{bmatrix} R_{xx}^{-1} + R_{xx}^{-1}R_{xy}S^{-1}R_{yx}R_{xx}^{-1} & -R_{xx}^{-1}R_{xy}S^{-1} \\ -S^{-1}R_{yx}R_{xx}^{-1} & S^{-1} \end{bmatrix} \quad (\text{B.6})$$

donde $S = R_{yy} - R_{yx}R_{xx}^{-1}R_{xy}$ es el complemento de Schur. Otra forma de expresar la ecuación (B.6) es usando el resultado del Lema 1. En este caso se tiene

$$R_{zz}^{-1} = \begin{bmatrix} R_{xx}^{-1} & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} -R_{xx}^{-1}R_{xy} \\ I \end{bmatrix} S^{-1} \begin{bmatrix} -R_{yx}R_{xx}^{-1} & I \end{bmatrix} \quad (\text{B.7})$$

Además, si se usa el teorema de inversión de matrices la inversa de la componente de Schur puede escribirse como

$$S^{-1} = R_{yy}^{-1} + R_{yy}^{-1}R_{xy} (R_{xx} + R_{xy}R_{yy}^{-1}R_{yx})^{-1} R_{xy}R_{yy}^{-1} \quad (\text{B.8})$$

Apéndice C

Cálculo de $f(x|Y = y)$ cuando X e Y son vectores aleatorios Gaussianos.

Supongamos que \mathbf{X} y \mathbf{Y} son vectores aleatorios Gaussianos de dimensiones m y n respectivamente. Entonces el vector $\mathbf{Z} = [\mathbf{X}, \mathbf{Y}]^T$ es un vector aleatorio Gaussiano de media:

$$E[\mathbf{Z}] = \mu_{\mathbf{z}} = \begin{bmatrix} \mu_{\mathbf{x}} \\ \mu_{\mathbf{y}} \end{bmatrix} \quad (\text{C.1})$$

y de covarianza dad por la expresión (B.5).

La función densidad de probabilidad para la variable \mathbf{Z} tiene la siguiente expresión

$$f(\mathbf{z}) = f(\mathbf{x}, \mathbf{y}) = \frac{1}{(2\pi)^{p/2} |R_{zz}|^{1/2}} \exp \left[-\frac{1}{2} (\mathbf{z} - \mu_{\mathbf{z}})^T R_{zz}^{-1} (\mathbf{z} - \mu_{\mathbf{z}}) \right] \quad (\text{C.2})$$

donde $p = m + n$. Si $\mathbf{Y} = \mathbf{y}$ ha sido observado y quiere actualizarse la distribución en \mathbf{X} , puede usarse el Teorema de Bayes de modo que se tenga:

$$f(\mathbf{x}|\mathbf{y}) = \frac{f(\mathbf{x},\mathbf{y})}{f(\mathbf{y})} \\ = C_1 \frac{\exp\left[-\frac{1}{2}[(\mathbf{x}-\mu_{\mathbf{x}})^T, (\mathbf{y}-\mu_{\mathbf{y}})^T] \begin{bmatrix} R_{xx} & R_{xy} \\ R_{yx} & R_{yy} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{x} - \mu_{\mathbf{x}} \\ \mathbf{y} - \mu_{\mathbf{y}} \end{bmatrix}\right]}{\exp\left[-\frac{1}{2}(\mathbf{y}-\mu_{\mathbf{y}})^T R_{yy}^{-1}(\mathbf{y}-\mu_{\mathbf{y}})\right]} \quad (\text{C.3})$$

Si llamamos E al exponente de la expresión (C.3) y usamos (B.6), se obtiene

$$E = (\mathbf{x} - \mu_{\mathbf{x}} - \mathbf{z})^T A (\mathbf{x} - \mu_{\mathbf{x}} - \mathbf{z}) + \text{términos que no dependen de } \mathbf{x} \quad (\text{C.4})$$

Usando la anterior expresión se tiene que

$$\mathbf{z} = A^{-1}\mathbf{b} = (R_{xx} + R_{xy}R_{yy}^{-1}R_{yx}) (R_{xx}^{-1}R_{xy}S^{-1}) (\mathbf{y} - \mu_{\mathbf{y}}) \\ = -R_{xy}R_{yy}^{-1} (\mathbf{y} - \mu_{\mathbf{y}}) \quad (\text{C.5})$$

Los resultados anteriores permiten concluir que $f(\mathbf{x}|\mathbf{y})$ corresponde a una distribución Gaussiana de media

$$E[\mathbf{X}|\mathbf{Y}] = \mu_{\mathbf{x}} - \mathbf{R}_{\mathbf{xy}}\mathbf{R}_{\mathbf{yy}}^{-1}(\mathbf{y} - \mu_{\mathbf{y}}) \quad (\text{C.6})$$

y covarianza

$$\text{cov}(\mathbf{X}|\mathbf{Y}) = A^{-1} = R_{xx} + R_{xy}R_{yy}^{-1}R_{yx} \quad (\text{C.7})$$

Apéndice D

Obtención de la probabilidad

$$p(\bar{y}_l | \mathbf{b}_{n,1:k}, \mathbf{b}_{1:n-1}, \bar{y}_{1:l})$$

La probabilidad $p(\bar{y}_l | \mathbf{b}_{n,1:k}, \mathbf{b}_{1:n-1}, \bar{y}_{1:l})$ puede obtenerse como

$$\begin{aligned} & p(\bar{y}_l | \mathbf{b}_{n,1:k}, \mathbf{b}_{1:n-1}, \bar{y}_{1:l-1}) \\ &= \int p(\bar{y}_l, \mathbf{h}_l | \mathbf{b}_{n,1:k}, \mathbf{b}_{1:n-1}, \bar{y}_{1:l}) d\mathbf{h}_l \\ &= \int p(\bar{y}_l | \mathbf{h}_l, \mathbf{b}_{n,1:k}) p(\mathbf{h}_l | \mathbf{b}_{n,1:k-1}, \mathbf{b}_{1:n-1}, \bar{y}_{1:l-1}) d\mathbf{h}_l \end{aligned} \quad (\text{D.1})$$

donde la última igualdad se deriva del hecho de que, dado \mathbf{h}_l , y $\mathbf{b}_{n,1:k}$, \bar{y}_l , es independiente de las otras variables, y \mathbf{h}_l es independiente de $b_{n,k}$. En (D.1), aparecen dos distribuciones en la integral. La primera distribución es la probabilidad definida por ecuación de observación que es

$$p(\bar{y}_l | \mathbf{h}_l, \mathbf{b}_{n,1:k}) = \mathcal{N}(\mathbf{C}_l \mathbf{h}_l, \sigma^2). \quad (\text{D.2})$$

donde $\mathbf{C}_l = \mathbf{f}_k^T \tilde{\mathbf{B}}_l \mathbf{P}$. La segunda distribución $p(\mathbf{h}_l | \mathbf{b}_{n,1:k-1}, \mathbf{b}_{1:n-1}, \bar{y}_{1:l-1})$ es la densidad predictiva que puede obtenerse del paso de predicción de un filtro

de Kalman [41, 42], i.e.,

$$p(\mathbf{h}_l | \mathbf{b}_{n,1:k-1}, \mathbf{b}_{1:n-1}, \bar{y}_{1:l-1}) = \mathcal{N}(\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l) \quad (\text{D.3})$$

donde

$$\boldsymbol{\mu}_l = \begin{cases} \mathbf{Q}\boldsymbol{\eta}_{l-1} & \text{si } k = 1 \\ \boldsymbol{\eta}_{l-1} & \text{si } k \neq 1 \end{cases} \quad (\text{D.4})$$

y

$$\boldsymbol{\Sigma}_l = \begin{cases} \mathbf{Q}\boldsymbol{\Xi}_{l-1}\mathbf{Q}^\top + \sigma^2\mathbf{G}\mathbf{G}^\top & \text{si } k = 1 \\ \boldsymbol{\Xi}_{l-1} & \text{si } k \neq 1 \end{cases}$$

En (D.4) y (D.5), $\boldsymbol{\eta}_{l-1}$ y $\boldsymbol{\Xi}_{l-1}$ se calculan de los pasos de actualización del filtro de Kalman expresados en términos de l del siguiente modo

$$\boldsymbol{\eta}_l = \boldsymbol{\mu}_l + \mathbf{K}_l(\bar{y}_l - m_l) \quad (\text{D.5})$$

y

$$\boldsymbol{\Xi}_l = (\mathbf{I} - \mathbf{K}_l\mathbf{C}_l)\boldsymbol{\Sigma}_l \quad (\text{D.6})$$

donde $m_l = \mathbf{C}_l\boldsymbol{\mu}_l$ y $\mathbf{K}_l = \boldsymbol{\Sigma}_l\mathbf{C}_l^\text{H}/c_l$ con $c_l = \mathbf{C}_l\boldsymbol{\Sigma}_l\mathbf{C}_l^\text{H} + \sigma^2$. Ahora la integración en (D.1) queda de la siguiente forma

$$p(\bar{y}_l | \mathbf{b}_{n,1:k}, \mathbf{b}_{1:n-1}, \bar{y}_{1:l}) = \mathcal{N}(m_l, c_l). \quad (\text{D.7})$$

Apéndice E

Fórmulas para trabajar con el producto de gaussianas

En este apéndice se presenta y demuestra un teorema que facilita el cálculo cuando se trabaja con productos o divisiones de funciones de distribución Gaussianas. Supongamos que estamos trabajando con el producto de dos Gaussianas. La primera de ellas en la variable \mathbf{y}_i , de media $\mathbf{H}\boldsymbol{\theta}$ que como vemos depende de la variable $\boldsymbol{\theta}$ y una matriz de covarianza cuya inversa es \mathbf{U} . La segunda de las Gaussianas es en la variable $\boldsymbol{\theta}$, tiene como media el vector $\boldsymbol{\theta}_a$ y como inversa de la matriz de covarianza \mathbf{A} . Ese producto es proporcional a otra gaussiana en la variable $\boldsymbol{\theta}$ de media $\boldsymbol{\theta}_b$ y con matriz de covarianza cuya inversa es \mathbf{B} . Como estamos trabajando con el producto de dos funciones que son exponenciales, podemos operar equivalentemente con la suma de sus exponentes. De forma que se llega al siguiente **Teorema**:

$$\begin{aligned} (\mathbf{y}_i - \mathbf{H}\boldsymbol{\theta})^T \mathbf{U} (\mathbf{y}_i - \mathbf{H}\boldsymbol{\theta}) + (\boldsymbol{\theta} - \boldsymbol{\theta}_a)^T \mathbf{A} (\boldsymbol{\theta} - \boldsymbol{\theta}_a) &= (\boldsymbol{\theta} - \boldsymbol{\theta}_b)^T \mathbf{B} (\boldsymbol{\theta} - \boldsymbol{\theta}_b) \quad (\text{E.1}) \\ + \mathbf{y}_i^T \mathbf{U} \mathbf{y}_i + \boldsymbol{\theta}_a^T \mathbf{A} \boldsymbol{\theta}_a - (\mathbf{H}^T \mathbf{U} \mathbf{y}_i + \mathbf{A} \boldsymbol{\theta}_a)^T \mathbf{B}^{-1} (\mathbf{H}^T \mathbf{U} \mathbf{y}_i + \mathbf{A} \boldsymbol{\theta}_a) \end{aligned}$$

donde

$$\boldsymbol{\theta}_{b_i} = (\mathbf{H}^T \mathbf{U} \mathbf{H} + \mathbf{A})^{-1} (\mathbf{H}^T \mathbf{U} \mathbf{y}_i + \mathbf{A} \boldsymbol{\theta}_a) \quad (\text{E.2})$$

$$\mathbf{B} = \mathbf{H}^T \mathbf{U} \mathbf{H} + \mathbf{A} \quad (\text{E.3})$$

Prueba. Para probar la igualdad anterior simplemente hay que desarrollar la suma de exponentes y sustituir las expresiones de $\boldsymbol{\theta}_{b_i}$ y \mathbf{B} . Si desarrollamos primero:

$$\begin{aligned} (\boldsymbol{\theta} - \boldsymbol{\theta}_b)^T \mathbf{B} (\boldsymbol{\theta} - \boldsymbol{\theta}_b) &= (\boldsymbol{\theta} - (\mathbf{H}^T \mathbf{U} \mathbf{H} + \mathbf{A})^{-1} (\mathbf{H}^T \mathbf{U} \mathbf{y}_i + \mathbf{A} \boldsymbol{\theta}_a))^T (\mathbf{H}^T \mathbf{U} \mathbf{H} + \mathbf{A}) \\ &\quad (\boldsymbol{\theta} - (\mathbf{H}^T \mathbf{U} \mathbf{H} + \mathbf{A})^{-1} (\mathbf{H}^T \mathbf{U} \mathbf{y}_i + \mathbf{A} \boldsymbol{\theta}_a)) \\ &= \boldsymbol{\theta}^T (\mathbf{H}^T \mathbf{U} \mathbf{H} + \mathbf{A}) \boldsymbol{\theta} + (\mathbf{H}^T \mathbf{U} \mathbf{y}_i + \mathbf{A} \boldsymbol{\theta}_a)^T (\mathbf{H}^T \mathbf{U} \mathbf{H} + \mathbf{A}) \\ &\quad (\mathbf{H}^T \mathbf{U} \mathbf{y}_i + \mathbf{A} \boldsymbol{\theta}_a) - (\mathbf{H}^T \mathbf{U} \mathbf{y}_i + \mathbf{A} \boldsymbol{\theta}_a)^T \boldsymbol{\theta} \\ &\quad - \boldsymbol{\theta}^T (\mathbf{H}^T \mathbf{U} \mathbf{y}_i + \mathbf{A} \boldsymbol{\theta}_a) \end{aligned}$$

y luego :

$$\begin{aligned} (\mathbf{H}^T \mathbf{U} \mathbf{y}_i + \mathbf{A} \boldsymbol{\theta}_a)^T \mathbf{B}^{-1} (\mathbf{H}^T \mathbf{U} \mathbf{y}_i + \mathbf{A} \boldsymbol{\theta}_a) &= (\mathbf{H}^T \mathbf{U} \mathbf{y}_i + \mathbf{A} \boldsymbol{\theta}_a)^T (\mathbf{H}^T \mathbf{U} \mathbf{H} + \mathbf{A}) \\ &\quad (\mathbf{H}^T \mathbf{U} \mathbf{y}_i + \mathbf{A} \boldsymbol{\theta}_a) \end{aligned}$$

Y finalmente sustituimos las expresiones anteriores en el miembro derecho de la ecuación E.1 tendremos:

$$\begin{aligned} &\mathbf{y}_i^T \mathbf{U} \mathbf{y}_i + (\mathbf{H} \boldsymbol{\theta})^T \mathbf{U} (\mathbf{H} \boldsymbol{\theta}) - \mathbf{y}_i^T \mathbf{U} (\mathbf{H} \boldsymbol{\theta}) - (\mathbf{H} \boldsymbol{\theta})^T \mathbf{U} \mathbf{y}_i \\ &+ \boldsymbol{\theta}^T \mathbf{A} \boldsymbol{\theta} + \boldsymbol{\theta}_a^T \mathbf{A} \boldsymbol{\theta}_a - \boldsymbol{\theta}^T \mathbf{A} \boldsymbol{\theta}_a - \boldsymbol{\theta}_a^T \mathbf{A} \boldsymbol{\theta} \end{aligned} \quad (\text{E.4})$$

que es precisamente el desarrollo del miembro derecho de la ecuación E.1.

Bibliografía

- [1] A A. Russ and M. K. Varanasi. Noncoherent multiuser detection for nonlinear modulation over the Rayleigh-fading channel. *IEEE Transactions on Information Theory*, 47:295–307, Jan. 2001.
- [2] A. Doucet, J.F.G. de Freitas, and N.J. Gordon Eds. Sequential Monte Carlo methods in practice. New York: Springer-Verlag, 2nd edition edition, 2001.
- [3] V. Aidala and S. Hammel. Utilization of modified polar coordinates for bearings-only tracking. *IEEE Transactions on Automatic Control*, 28:283–294, Mar 1983.
- [4] P. D. Alexander, M. C. Reed, J. A. Asenstorfer, and C. B. Schlegel. Iterative multiuser interference reduction: Turbo CDMA. *IEEE Trans. Commun.*, 49:1008–1014, 1999.
- [5] M. Athans, R. P. Wishner, and A. Bertolini. Suboptimal state estimation for continuous-time nonlinear systems from discrete noisy measurements. *IEEE Transactions on Automatic Control*, TAC-13(6):504–518, October 1968.
- [6] J.R. Baker, P.K. Lattimore, and L.A. Matheson. Constructing optimal drug-testing plans using a bayesian acceptance sampling model. *Mathematical and Computer Modeling*, 17(2):77–88, 1993.
- [7] Z. Bar-Joseph. Analyzing time series gene expression data. *Bioinformatics*, 20(16):2493–2503, 2004.
- [8] Z. Bar-Joseph, G. Gerber, I. Simon, D. K. Gifford, and T. S. Jaakkola. Comparing the continuous representation of time-series expression profiles to identify differentially expressed genes. *PNAS*, 100(18):10146–10151, Sept 2003.

-
- [9] S. Båro and G. Bauch. Improving BLAST performance using space-time block codes and turbo decoding. *Proceedings of IEEE Global Telecommunications Conference*, 2:1067–1071, 2000.
- [10] B.D. Anderson and J. B. Moore. *Optimal Filtering*. NJ:Prentice-Hall, 1979.
- [11] M. J. Beal. *Variational algorithm for approximate bayesian inference*. PhD thesis, University of London, 2003.
- [12] M. J. Beal, F. Falciani, Z. Ghahramani, C. Rangel, and D. L. Wild. A Bayesian approach to reconstructing genetic regulatory networks with hidden factors. *Bioinformatics*, 20:1361–1372, Sept. 2004.
- [13] J. M. Bernardo and A. F. Smith, editors. *Bayesian Theory*. John Wiley and Son Ltd, 1994.
- [14] J.A.C. Bingham. Multicarrier modulation for data transmission: an idea whose time has come, 1990.
- [15] B.K. Moon and W.C. Stirling. *Mathematical Methods and Algorithms for Signal Processing*. Prentice-Hall, 2000.
- [16] P. Brazhnik, A. de la Fuente, and P. Mendes. Gene networks: how to put the function in genomics. *Trends in Biotechnology*, 20(11):467–472, Nov. 2002.
- [17] Bradley P. Carlin, Nicholas G. Polson, and David S. Stoffer. A Monte Carlo approach to nonnormal and nonlinear state-space modeling. *Journal of the American Statistical Association*, 87(418):493–??, 1992.
- [18] C. K. Carter and R. Kohn. On gibbs sampling for state space models. *Biometrika*, 81:541–553, 1994.
- [19] R. Chen, X. Wang, and J. S. Liu. Adaptive joint detection and decoding in flat-fading channels via mixture Kalman filtering. *IEEE Transactions on Information Theory*, 46(6):2079–2094, Sept. 2000.
- [20] R. Cho, M. Campbell, E. Winzeler, L. Steinmetz, A. Conway, L. Wodicka, T. Wolfsberg, A. Gabrielian, D. Landsman, D. Lockhart, and R. Davis. A genome-wide transcriptional analysis of the mitotic cell cycle. *Mol. Cell.*, (2):65–73, 1998.
- [21] G. C. Colavolpe, A. Barbieri, and G. Caire. Iterative decoding in the presence of strong phase noise. submitted to IEEE JSAC.

-
- [22] G. C. Colavolpe, A. Barbieri, and G. Caire. Bayesian and non-bayesian methods for iterative joint decoding and detection in the presence of phase noise. *Proc. ISIT 2004*, June 2004.
- [23] J. Dauwels and H. A. Loeliger. Phase estimation by message passing. In *Proceedings of the 2004 IEEE International Conf. on Communications*, 2004.
- [24] J. DeRisi, V. Iyer, and P. Brown. Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, pages 680–686, 1997.
- [25] P. M. Djurić, J. M. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Míguez. Applications of particle filtering to selected problems in communications. *Signal Processing Magazine*, 20:19–38, Sept. 2003.
- [26] A. Doucet, J. de Freitas, and N. Gordon, editors. *Sequential Monte-Carlo Methods in Practice*. Springer-Verlag, 2001.
- [27] A. Duel-Hallen. Decorrelation decision-feedback multiuser detector for synchronous code-division multiple-access channel. *IEEE Transactions on Communications*, COM-41:285–290, Feb. 1993.
- [28] K.S. Gilhousen et al. On the capacity of a cellular cdma system. *IEEE Transactions on Vehicular Technology*, 40:303–312, May 1991.
- [29] European Telecommunications Standards Institute 2nd ed. ETS 300 401. Radio broadcasting systems; digital audio broadcasting (dab) to mobile, portable and fixed receivers, May, 1997.
- [30] European Telecommunications Standards Institute ETS 300 744. Digital terrestrial broadcasting systems, Feb, 1997.
- [31] European Telecommunications Standards Institute ETS 300 744. Digital terrestrial broadcasting systems, Feb, 1997.
- [32] U. Fawer and B. Aazhang. A multiuser receiver for code division multiple access communications over multipath channels. *IEEE Trans. on Communications*, 43:1556–1565, Feb./Mar./April 1995.
- [33] P. Fearnhead. *Sequential Monte Carlo Method in filter theory*. PhD thesis, Oxford University, 1998. Available from <http://www.maths.lancs.ac.uk/fearnhea/>.

-
- [34] B. Flanagan, C. Suprin, S. Kumaresan, and J. Duniak. Performance of a joint Kalman demodulator for multiuser detection. In *Proceedings of IEEE 56th Vehicular Technology Conference*, pages 1525–1529, 2002.
- [35] N. Friedman. Inferring cellular networks using probabilistic graphical models. *Science*, 303:799–805, Feb 2004.
- [36] N. Friedman, M. Linial, I. Nachman, and D. Pe’er. Using bayesian networks to analyze expression data. *Journal of Computational Biology*, 7(3-4):601–620, 2000.
- [37] A. P. Gasch, P. T. Spellman, C. M. Kao, O. Carmel-Harel, M. B. Eisen, G. Storz, D. Botstein, and P. O. Brown. Genomic expression program in the response of yeast cells to environmental changes. *Mol. Bio.*, 11:4241–4257, 2000.
- [38] P. Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *IEEE Transactions on Communications*, 82:711–732, 1995.
- [39] J.M. Gutierrez, R. Cano, A.S. Cofiño, and C.M. Sordo. *Redes Probabilísticas y Neuronales en las Ciencias Atmosféricas*. Ministerio de Medio Ambiente, 2004.
- [40] A.J. Hartemink, D.K. Giord, T.S. Jaakkola, and R.A. Young. Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks. *PacLc Symposium on Biocomputing*, 6:23–32, 2001.
- [41] M. H. Hayes. *Statistical digital signal processing and modeling*. John Wiley & Sons, Inc, New York, 1996.
- [42] S. Haykin. *Adaptive Filter Theory*. Prentice Hall, Upper Saddle River, NJ, 4th edition, 2002.
- [43] M. Honig, U. Madhow, and S. Verdu. Blind adaptive multiuser detection. *IEEE Transactions on Information Theory*, 41:944–960, July 1995.
- [44] Y. Huang and P. M. Djurić. A blind particle filtering detector of signals transmitted over flat fading channels. *IEEE Transactions on Signal Processing*, 52(7):1891–1900, 2004.
- [45] Y. Huang, J. Wang, Y. Wang, J. Zhang, and M. Sanchez. Inference of cell cycle regulatory networks using dynamic Bayesian networks. *Submitted*

- to IEEE Transactions on Signal Processing (Special Issue on Genomic Signal Processing)*, 2005.
- [46] Y. Huang, J. Wang, J. Zhang, M. Sanchez, and Y. Wang. Bayesian inference of genetic regulatory networks from time series microarray data using dynamic bayesian networks, *Bioinformatics*. Under review.
- [47] Y. Huang, J. Zhang, and P. M. Djurić. Bayesian detection for BLAST. *IEEE Transactions on Signal Processing*, 53(3):1086–1096, 2005.
- [48] T. R. Hughes, M. J. Marton, A. R. Jones, C. J. Roberts, R. Stoughton, C. D. Armour, H. A. Bennett, E. Cokey, H. Dai, Y. D. He, M. J. Kidd, A. M. King, M. R. Meyer, D. Slade, P. Y. Lum, S. B. Stepaniants, D. D. Shoemaker, D. Gachotte, K. Chakraburttty, J. Simon, M. Bard, and S. H. Friend. Functional discovery via a compendium of expression profiles ,. *Cell*, pages 109–26, 2000.
- [49] D. Husmeier. Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks. *Bioinformatics*, 19(17):2271–2282, 2003.
- [50] H.W.Sorenson and A.R. Stubberud. Nonlinear filtering by approximation of the a posterior density. *Int.J.Contr.*, 18:15–33, 1968.
- [51] R. A. Iltis. Exact and approximate maximum-likelihood parameter estimation for QS-CDMA signals. *IEEE Trans. Communications*, 48:1208–1216, July 2000.
- [52] J.S.Liu and R.Chen. Blind deconvolution via sequential imputations. *Journal of the American Statistical Association*, 90:567–576, 1995.
- [53] K. J. Kim and R.A. Iltis. Joint detection and channel estimation algorithms for QS-CDMA signals over time-varying channels. *IEEE Transactions on Communications*, 50(5):845 –855, May 2002.
- [54] S. Y. Kim, S. Imoto, and S. Miyano. Inferring gene networks from time series microarray data using dynamic Bayesian networks. *Briefings in Bioinformatics*, 4(3):228–235, 2003.
- [55] R. Kohno, R. Meidian, and L.B. Milstein. Spread spectrum access methods for wireless communications. *IEEE Communication magazine*, 33:58–67, Jan. 1995.

-
- [56] A. Kong, J.S.Liu, and W.H. Wong. Sequential imputations and bayesian missing data problems. *Journal of the American Statistical Association*, 89:278–288, 1994.
- [57] F. R. Kschischang, B.J. Frey, and H. A. Loeliger. Factor graphs and the sum product algorithm. *IEEE Trans. on Info. Theory*, 47:498–519, February 2001.
- [58] S. Lang, P. Kragler, G. Haybach, and L. Fahrmeir. *Bayesian space-time analysis of health insurance data*. 2002.
- [59] C. Laot, A. Glavieux, and J. Labat. Turbo equalization: Adaptive equalization and channel decoding jointly optimized. *IEEE Journal on Selected Areas in Communications*, 19(9):1744–1752, Sept. 2001.
- [60] Q. Li, X. Wang, and C.Ñ. Georgiades. Turbo multiuser detection for turbo-coded CDMA in multipath fading channels. *IEEE Transactions on Vehicular Technology*, 51(5):1096–1108, Sept. 2002.
- [61] T. J. Lim, L. K. Rasmussen, and H. Sugimoto. An asynchronous multiuser CDMA detector based on the Kalman filter. *IEEE Journal on Selected Areas in Communications*, 16(9):1711–1722, Dec 1998.
- [62] Teng Joon Lim and Yao Ma. The Kalman filter as the optimum MMSE multiuser CDMA detector. *IEEE Transactions on Information Theory*, 46(7):2561–2566, Nov. 2000.
- [63] D. J. Lockhart, H. Dong, M. C. Byrne, M. T. Follettie, M. V. Gallo, M. S. Chee, M. Mittmann, C. Wang, M. Kobayashi, H. Horton, and E. L. Brown. Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nat Biotechnol*, 14:1675–80, 1996.
- [64] T.J. Loredo. *From Laplace to Supernova SN 1987A: A Bayesian Inference in Astrophysics*. PhD thesis, University of Chicago, 1995.
- [65] V. Lottici and M. Luise. Carrier phase recovery for turbo-coded linear modulations. *Proc. IEEE International Conf. Commun.*, pages 1541–1545, Apr. 2002.
- [66] Isabel M. Tienda Luna, Y. Huang, Diego Salas González, and Diego P. Ruiz Padillo. Un algoritmo para la detección ciega multiusuario basado en un modelado de espacio de estados, Setp.2004. URSI 2004.

-
- [67] Isabel M. Tienda Luna, Y. Huang, José A. Garzón Guerrero, and M. Carmen Carrión Pérez. Estudio del rendimiento de distintos detectores multiusuario en canales rayleigh planos desvanescentes rápidos en escenarios near-far, Setp.2004. URSI 2004.
- [68] D. J. C. MacKay. Regular ldpc online database. available at the url <http://www.inference.phy.cam.ac.uk/mackay/>.
- [69] M. Peleg, S. Shamai, and S. Galan. Iterative decoding for coded noncoherent mpsk communications over phase-noisy awgn channel. *IEE Proc. Commun.*, 147(3):87–95, April 2000.
- [70] Iosifina Pournara and Lorenz Wernisch. Reconstruction of gene networks using Bayesian learning and manipulation experiments. *Bioinformatics*, 20(17):2934–2942, 2004.
- [71] P. Patel and J. Holtzman. Analysis of a simple successive interference cancellation scheme in ds/cdma system. *IEEE JSA-Special issue on CDMA*, 12:796–807, June 1994.
- [72] E. Punskeya, C. Andrieu, A. Doucet, and W.J. Fitzgerald. Particle filtering for multiuser detection in fading CDMA channels. In *Proc. 11th IEEE Signal Processing Workshop*, pages 38–41, 2001.
- [73] P. Sebastiani, E. Gussoni, I. S. Kohane, and M. Ramoni. Statistical challenges in functional genomics (with discussion). *Statistical Science*, 18(1):33–60., 2003.
- [74] E. Segal, M. Shapira, A. Regev, D. Pe’er, D. Botstein, D. Koller, and N. Friedman. Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nature Genetics*, 2004.
- [75] M. Sellathurai and S. Haykin. Turbo-BLAST for wireless communications: theory and experiments. *IEEE Transactions on Signal Processing*, 50(10):2538–2546, 2002.
- [76] R. Shamir. Lecture notes: Bayesian networks. <http://www.cs.tau.ac.il/rshamir/ge/05>.
- [77] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang. Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, 18(1), 2002.

-
- [78] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher. Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Molecular Biology of the Cell*, 9:3273–3297, 1998.
- [79] I. M. Tienda-Luna, Y. Huang, D.P. Ruiz, and M. C. Carrion. Iterative decoding in factor graphs representation using particle filtering. In *Proceedings of the 2005 IEEE SPAWC*, 2005.
- [80] E. Tungsrisaguan and R. M. A. P. Rajatheva. Turbo equalization with sequential sequence estimation over multipath fading channels. *IEEE Communication Letters*, 6(3):93–95, March 2002.
- [81] M.K. Varanasi and B. Aazhang. Near-optimum detector in synchronous code-division multiple access channel. *IEEE Transactions on Information Theory*, COM-39:725–736, May 1991.
- [82] S. Vasudevan and M. K. Varanasi. Achieving near optimal asymptotic efficiency and fading resistance over the time-varying Rayleigh-faded CDMA channel,. *IEEE Trans. Communications*, 44:1130–1143, Sept. 1996.
- [83] S. Verdú. Minimum probability of error for asynchronous gaussian multiple-access channel. *IEEE Transactions on Information Theory*, IT-32:85–96, Jan. 1986.
- [84] S. Verdú. *Multiuser Detection*. Cambridge University Press, New York, 1998.
- [85] X. Wang and H. V. Poor. Iterative (turbo) soft interference cancellation and decoding for coded CDMA. *IEEE Trans. Commun.*, 47:1046–1061, July 1999.
- [86] X. Wang and V. Poor. Blind multiuser detection: A subspace approach. *IEEE Transactions on Information Theory*, 44(2):677–690, Mar 1998.
- [87] W.F.Denham and S.Pines. Sequential estimation when measurement function nonlinearity is comparable to measurement error. *AIAA*, 4:1071–1076, 1966.
- [88] A. P. Worthen and W.E. Stark. Unified design of iterative receivers using factor graphs. *IEEE Trans. on Info. Theory*, 47:843–849, February 2001.

-
- [89] Hsin-Yu Wu and A. Duel-Hallen. On the performance of coherent and noncoherent multiuser detectors for mobile radio CDMA channels. In *Proceedings of 5th IEEE International Conference on Universal Personal Communications*, volume 1, pages 76–80, 1996.
- [90] P. H. Wu and A. Duel-Hallen. Multiuser detectors with disjoint Kalman channel estimators for synchronous CDMA mobile radio channels. *IEEE Transaction on Communications*, 48(5):752–756, May 2000.
- [91] Y. Wu and W. Y. Zou. Orthogonal frequency division multiplexing: a multicarrier modulation squeme, 1995.
- [92] Z. Yang and X. Wang. Blind turbo multiuser detection for long-code multipath CDMA. *IEEE Transactions on Communications*, 50(1):112–125, Jan. 2002.
- [93] Kai Yen and L. Hanzo. Genetic algorithm assisted joint multiuser symbol detection and fading channel estimation for synchronous CDMA systems. *IEEE Journal on Selected Areas in Communications*, 19:985–998, June 2001.
- [94] Y.Huang and J. Zhang. Adaptative blind multiuser detection over flat fast fading channels using particle filtering. Submitted to *IEEE Transactions on Signal Processing*.
- [95] Y.Huang, J. Zhang, Isabel Tienda Luna, and P.M. Djurić. Adaptative blind multiuser detection over flat fast fading channels using particle filtering, Setp.2004. *Globecom 2004*.
- [96] J. Zhang and P. M. Djurić. Joint estimation and decoding of space-time trellis codes. *EURASIP Journal on Applied Signal Processing*, pages 305–315, March 2002.
- [97] Y. Zhang and R. S. Blum. Iterative multiuser detection for turbo-coded synchronous CDMA in Gaussian and non-Gaussian impulsive noise. *IEEE Transactions on Communications*, 49(3):397–400, March 2001.