



**HAL**  
open science

## Bayesian estimation of discrete signals with local dependencies.

Mohammad Hassan Majidi

► **To cite this version:**

Mohammad Hassan Majidi. Bayesian estimation of discrete signals with local dependencies.. Other. Supélec, 2014. English. NNT : 2014SUPL0014 . tel-01080355

**HAL Id: tel-01080355**

**<https://theses.hal.science/tel-01080355>**

Submitted on 5 Nov 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



N° d'ordre : 2014-14-TH

# SUPELEC

ECOLE DOCTORALE STITS

« Sciences et Technologies de l'Information des Télécommunications et des Systèmes »

## THÈSE DE DOCTORAT

DOMAINE : STIC

Spécialité : Télécommunications

Soutenue le 24 Juin 2014

par :

**Mohammad-Hassan MAJIDI**

**Estimation bayésienne de signaux discrets à dépendances locales**

Bayesian estimation of discrete signals with local dependencies

<b>Directeur de thèse :</b>	Antoine O. BERTHET	Supélec
<b>Encadrant de thèse :</b>	Mithridad POURMIR	Supélec

**Composition du jury :**

<i>Président du jury :</i>	Ali MOHAMMAD-DJAFARI	Université PARIS XI
<i>Rapporteurs :</i>	Salah BOURENNANE	Ecole Centrale Marseille
	Jean-Pierre CANCES	ENSIL
<i>Examineurs :</i>	Jean-Pierre BARBOT	ENS Cachan

---

---

---

*“To My Parents and my family”*



# Acknowledgments

First of all, I would like to thank my respected supervisors Prof. BERTHET and Dr. POURMIR for their invaluable suggestions, guidance, patience and continuous encouragement throughout my research work, without which it would not have been possible to execute my research work.

I would also like to thank Prof. Salah BOURENNANE and Prof. Jean-Pierre CANCES for reviewing this work, providing insightful comments and attending my defense. I would like to thank as well Prof. Jean-Pierre BARBOT and Prof. Ali MOHAMMAD-DJAFARI for attending my defense as examiners and for their helpful comments.

I am very grateful to the ministry of science and technology of IRAN for supporting me financially to complete my Ph.D.

I offer my heartfelt thanks to my wife, for her intense support, endless patience and great devotion. Indeed, this journey would not have been possible without her encouragement and understanding and I am eternally indebted to her for what she has sacrificed during this work. The influence of my sons, is present in this thesis; they have filled my moments at home with so much joy.

I can not complete these acknowledgements without expressing how appreciative I am of all the love and affection that my parents, especially my mother, have provided through my life. I could have never made it this far without them.

---



# Abstract

The aim of this thesis is to study the problem of data detection in wireless communication system, for both case of perfect and imperfect channel state information at the receiver. As well known, the complexity of MLSE being exponential in the channel memory and in the symbol alphabet cardinality is quickly unmanageable and forces to resort to sub-optimal approaches. Therefore, **first** we propose a new iterative equalizer when the channel is unknown at the transmitter and perfectly known at the receiver. This receiver is based on continuation approach, and exploits the idea of approaching an original optimization cost function by a sequence of more tractable functions and thus reduce the receiver's computational complexity.

**Second**, in order to data detection under linear dynamic channel, when the channel is unknown at the receiver, the receiver must be able to perform joint equalization and channel estimation. In this way, we formulate a combined state-space model representation of the communication system. By this representation, we can use the Kalman filter as the best estimator for the channel parameters. The aim in this section is to motivate rigorously the introduction of the Kalman filter in the estimation of Markov sequences through Gaussian dynamical channels. By this we interpret and make clearer the underlying approximations in the heuristic approaches.

**Finally**, if we consider more general approach for non linear dynamic channel, we can not use the Kalman filter as the best estimator. Here, we use switching state-space model (SSSM) as non linear state-space model. This model combines the hidden Markov model (HMM) and linear state-space model (LSSM). In order to channel estimation and data detection, the expectation and maximization (EM) procedure is used as the natural approach. In this way extended Kalman filter (EKF) and particle filters are avoided.

---



---

---

# Résumé

L'objectif de cette thèse est d'étudier le problème de la détection de données dans le système de communication sans fil, à la fois pour le cas de l'information d'état de canal parfaite et imparfaite au niveau du récepteur. Comme on le sait, la complexité de MLSE est exponentielle en la mémoire de canal et la cardinalité de l'alphabet symbole est rapidement ingérable, ce qui force à recourir à des approches sous-optimales. Par conséquent, en premier lieu, nous proposons une nouvelle égalisation itérative lorsque le canal est inconnu à l'émetteur et parfaitement connu au niveau du récepteur. Ce récepteur est basé sur une approche de continuation, et exploite l'idée d'approcher une fonction originale de coût d'optimisation par une suite de fonctions plus dociles et donc de réduire la complexité de calcul au récepteur.

En second lieu, en vue de la détection de données sous un canal dynamique linéaire, lorsque le canal est inconnu au niveau du récepteur, le récepteur doit être en mesure d'effectuer conjointement l'égalisation et l'estimation de canal. De cette manière, on formule une représentation de modèle état-espace combiné du système de communication. Par cette représentation, nous pouvons utiliser le filtre de Kalman comme le meilleur estimateur des paramètres du canal. Le but de cette section est de motiver de façon rigoureuse la mise en place du filtre de Kalman dans l'estimation des séquences de Markov par des canaux dynamiques Gaussien. Par la présente, nous interprétons et explicitons les approximations sous-jacentes dans les approches heuristiques.

Enfin, si nous considérons une approche plus générale pour le canal dynamique non linéaire, nous ne pouvons pas utiliser le filtre de Kalman comme le meilleur estimateur. Ici, nous utilisons des modèles commutation d'espace-état (SSSM) comme modèles espace-état non linéaires. Ce modèle combine le modèle de Markov caché (HMM) et le modèle espace-état linéaire (LSSM). Pour l'estimation de canal et la

---

---

détection de données, l'approche espérance et maximisation (EM) est utilisée comme approche naturelle. De cette façon, le filtre de Kalman étendu (EKF) et les filtres à particules sont évités.

# Contents

<b>Acknowledgements</b>	<b>iii</b>
<b>Abstract</b>	<b>v</b>
<b>Résumé</b>	<b>vii</b>
<b>Contents</b>	<b>ix</b>
<b>List of Figures</b>	<b>xiii</b>
<b>Notations</b>	<b>xv</b>
<b>Résumé étendu en Français</b>	<b>xix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Research Context . . . . .	1
1.2 State-of-The-Art in Equalization and Estimation . . . . .	3
1.2.1 Basic Baseband Model of a Communication System . . . . .	3
1.2.2 Equalization Techniques . . . . .	5
1.2.2.1 ML per Sequence . . . . .	6
1.2.2.2 MAP per Symbol . . . . .	6
1.2.2.3 Linear Equalization . . . . .	8
1.2.2.4 Decision-Feedback Equalization . . . . .	11
1.2.2.5 Reduced-State Sequence Estimation . . . . .	13
1.2.3 Channel Estimation Techniques . . . . .	14
1.2.3.1 The Kalman Filter . . . . .	17
1.2.3.2 The RLS Algorithm . . . . .	20

---

1.2.3.3	The LMS Algorithm . . . . .	21
1.2.4	Joint Data Detection and Channel Estimation . . . . .	22
1.3	Contribution . . . . .	24
1.4	Thesis Outline . . . . .	25
<b>2</b>	<b>A Continuation Approach to Iterative Sequence Estimation</b>	<b>27</b>
2.1	Introduction . . . . .	27
2.2	System Model and Problem Formulation . . . . .	28
2.3	A Continuation Approach . . . . .	29
2.3.1	Principle . . . . .	29
2.3.2	Iterative Maximization of $P^{(\alpha)}(\mathbf{y}, \mathbf{x})$ Using EM . . . . .	30
2.3.3	Interpretation from Statistical Mechanics . . . . .	31
2.3.4	Exact Implementation of The Algorithm. . . . .	31
2.4	Approximations . . . . .	32
2.4.1	Choice of An Appropriate Basis. . . . .	34
2.5	Convergence and Stability Issues . . . . .	34
2.5.1	Convergence of the Annealing Process. . . . .	34
2.5.2	Convergence of the Iterative Algorithm. . . . .	36
2.6	Numerical Results . . . . .	39
2.6.1	Performance on a Stanford University Interim (SUI) Channel . . . . .	40
2.6.2	Performance on a Rayleigh Fading Channel (Exponential) . . . . .	41
2.7	Conclusion . . . . .	42
<b>3</b>	<b>Estimation of Linear Dynamic Channels based on Kalman Filtering</b>	<b>43</b>
3.1	Introduction . . . . .	43
3.2	System Model . . . . .	45
3.3	Proposed Method based on Joint Kalman and Viterbi Algorithm . . . . .	48
3.3.1	Normal Density Assumption . . . . .	49
3.3.2	Approximate Recursive Maximization . . . . .	50
3.3.3	Evaluation of Dependency . . . . .	50
3.3.4	Final Form of Approximate Recursive Maximization . . . . .	51
3.4	Numerical Results . . . . .	52
3.4.1	Fixed Step Size PSP . . . . .	52

---

## Contents

---

3.4.2	PSP via RLS Algorithm . . . . .	54
3.4.3	Proposed Method . . . . .	54
3.5	Conclusion . . . . .	56
<b>4</b>	<b>EM-based Estimation of Discrete-Data Transmitted over Non-Linear Dynamic Wireless Channels</b>	<b>59</b>
4.1	Introduction . . . . .	59
4.2	System Model . . . . .	60
4.3	Proposed EM-based Approach to Data Detection . . . . .	63
4.3.1	Expectation Step . . . . .	66
4.3.1.1	Computing Densities $p_n(\mathbf{t}', \mathbf{t}, \mathbf{x}', \mathbf{x})$ . . . . .	67
4.3.1.2	Computing Expected Quantities $J(n, \mathbf{d}_n, \alpha_n)$ , $H_1(n)$ , $H_2(n)$ , $E_0[\ln p(\mathbf{c}_0^{(1)})]$ and $E_0[\ln p(\mathbf{c}_0^{(2)})]$ . . . . .	69
4.3.1.3	Determining The Final Conditional Expectation . . . . .	71
4.3.2	Maximization Step . . . . .	71
4.4	Numerical Results . . . . .	71
4.5	Conclusion . . . . .	72
<b>5</b>	<b>Conclusion and Future Research Perspectives</b>	<b>75</b>
5.1	Summary . . . . .	75
5.2	Research Perspectives . . . . .	76
<b>A</b>	<b>A Review of The EM Algorithm</b>	<b>79</b>
A.1	Introduction . . . . .	79
A.2	General Statement of The EM Algorithm . . . . .	81
A.2.1	Mathematical Formulation . . . . .	81
A.2.2	Monotonicity of The EM Algorithm . . . . .	82
A.2.3	Convergence to a Stationary Value . . . . .	83
A.3	Extension of the EM Algorithm to MAP Parameter Estimation . . . . .	83
<b>B</b>	<b>Detailed Algorithm. See Chapter 3.</b>	<b>85</b>
<b>C</b>	<b>A Review of EKF</b>	<b>87</b>
C.1	The Nonlinear State Space Model (NSSM) . . . . .	87

C.2	Linearization of NSSM . . . . .	88
<b>D</b>	<b>A Review of Particle Filtering</b>	<b>91</b>
<b>E</b>	<b>Proofs of The Main Equations. See Chapter 4.</b>	<b>99</b>
E.1	Recall: . . . . .	100
E.2	Recursion on $\Lambda_n^{(a)}(\mathbf{t}', \mathbf{t}, \mathbf{x}', \mathbf{x})$ . . . . .	100
E.3	Recursion on $\Lambda_n^{(b)}(\mathbf{x})$ . . . . .	104
E.4	Computing densities $p_n(\mathbf{t}', \mathbf{t}, \mathbf{x}', \mathbf{x})$ . . . . .	105
	<b>Bibliography</b>	<b>107</b>

# List of Figures

1	Représentation du modèle graphique pour la commutation des modèles état-espace (SSSM). $\alpha_n$ est la variable de commutation discrète et $\mathbf{c}_n(1)$ , $\mathbf{c}_n(2)$ sont les vecteurs d'état de valeurs complexes. . . . .	xlv
1.1	Time-varying channel occurs by mobility of transmitter, reflector and/or receiver. . . . .	2
1.2	Baseband model of a digital communication system, consisting of the transmitter, channel and equalizer (receiver). . . . .	3
1.3	Linear Transversal Equalizer. . . . .	8
1.4	A basic structure of a DFE with forward and feedback filters. . . . .	11
1.5	The adaptive receiver model for joint data detection and channel estimation. . . . .	23
2.1	BER versus SNR performance. . . . .	40
2.2	BER versus SNR performance (Binary i.i.d symbols, $N = 64$ , Rayleigh fading multipath, EXP $M = 9$ , $a = 1$ ). . . . .	41
2.3	BER versus SNR performance (Binary i.i.d symbols, $N = 64$ , Rayleigh fading multipath, EXP $M = 9$ , $a = 0.5$ ). . . . .	42
3.1	BER performance of fixed step size PSP at fading rate $f_D T = 0.0001$ . . . . .	52
3.2	BER performance of fixed step size PSP at fading rate $f_D T = 0.001$ . . . . .	53
3.3	BER performance of fixed step size PSP at fading rate $f_D T = 0.01$ . . . . .	53
3.4	BER performance of PSP via RLS algorithm at fading rate $f_D T = 0.0001$ . . . . .	54
3.5	BER performance of PSP via RLS algorithm at fading rate $f_D T = 0.001$ . . . . .	55

---



---

**List of Figures**

3.6	BER performance of PSP via RLS algorithm at fading rate $f_D T = 0.01$ .	55
3.7	BER performance at fading rate $f_D T = 0.0001$ .	56
3.8	BER performance at fading rate $f_D T = 0.01$ .	57
4.1	Graphical model representation for switching state-space model (SSSM). $\alpha_n$ is the discrete switch variable and $\mathbf{c}_n(1)$ , $\mathbf{c}_n(2)$ are the complex-valued state vectors.	64
4.2	HMM of switch variable $\alpha_n$ .	64
4.3	BER performance at switching fading rate $f_D T(1) = 0.0001$ and $f_D T(2) = 0.0003$	73
4.4	BER performance at switching fading rate $f_D T(1) = 0.0001$ and $f_D T(2) = 0.001$	73
4.5	BER performance at switching fading rate $f_D T(1) = 0.0001$ and $f_D T(2) = 0.003$	74
A.1	An illustration of the complete- and incomplete-data sets of the EM algorithm.	80
D.1	A schematic description of resampling	92
D.2	A pictorial description of particle filtering	94
D.3	A block diagram of particle filtering	96

# List of Notations

## Acronyms

APP	A Posteriori Probability
AWGN	Additive White Gaussian Noise
BCJR	Bahl-Cocke-Jelinek-Raviv (algorithm)
BER	Bit Error Rate
CIR	Channel Impulse Response
CSIR	Channel State Information at the Receiver
DFT	Discrete Fourier Transform
EM	Expectation-Maximization (algorithm)
i.i.d.	independent and identically distributed
ISI	Inter-Symbol Interference
LS	Least Squares
MAP	Maximum A Posteriori
MIMO	Multiple-Input Multiple-Output
ML	Maximum Likelihood
MMSE	Minimum Mean Square Error
MSE	Mean-Square Error
pdf	probability density function
SISO	Soft-Input Soft-Output
SNR	Signal-to-Noise Ratio
ZF	Zero-Forcing
DFE	Decision Feedback Equalization
MLSD	Maximum Likelihood Sequence Detection

---

PSP	Per Survivor Processing
RLS	Recursive Least Squares
RSSE	Reduced-State Sequence Estimation
DFSE	Decision-Feedback Sequence Estimation
EKF	Extended Kalman Filter
LMS	Least Mean Square
ACS	Add-Compare-Select
BM	Branch Metric
DPSK	Differential Phase Shift Keying
MG	Multipath Gain
LTE	Linear Transversal Equalizer
PSK	Phase Shift Keying
IC	Interference Canceller
CCM	Channel Convolution Matrix
CDMA	Code Division Multiple Access
CP	Cyclic Prefix
iff	if and only if
SSM	State Space Model
SSSM	Switching State Space Model
LSSM	Linear State Space Model
NSSM	Nonlinear State Space Model
HMM	Hidden Markov Model
SISO	Single-Input Single-Output
DSC	Doubly Selective Channel
SDSC	Switching Doubly Selective Channel

## Mathematical Notations

$\mathcal{N}_c(\mathbf{m}, \Sigma)$	complex circularly Gaussian distribution with parameters $\mathbf{m}$ and $\Sigma$
$diag(\mathbf{x})$	diagonal matrix having the vector $\mathbf{x}$ on its diagonal
$E_{\mathbf{x}}[\cdot]$	expectation with respect to the vector $\mathbf{x}$
$\mathbf{I}_n$	$(n \times n)$ identity matrix
$\mathbf{0}_{m \times n}$	$(m \times n)$ all-zero matrix
$\approx$	is approximated by
$\propto$	is proportional to
$(\cdot)^*$	matrix or vector conjugation
$(\cdot)^T$	matrix or vector transpose
$(\cdot)^\dagger$	matrix or vector Hermitian transpose
$det\{\cdot\}$	matrix determinant
$exp\{\cdot\}$	exponential
$tr(\cdot)$	matrix trace
$Re(\cdot)$	real part
$Im(\cdot)$	imaginary part
$\mathbb{F}$	field
$\mathbb{C}$	field of complex numbers
$A \propto B$	equality between $A$ and $B$ up to a multiplicative factor
$\mathcal{M}_{m \times n}(\mathbb{F})$	the set of $m \times n$ matrices over $\mathbb{F}$
$\mathcal{M}_n(\mathbb{F})$	the set of $n$ -square matrices over $\mathbb{F}$
$\mathbf{a}_l$	the $l^{th}$ column of $\mathbf{A}$ and $\mathbf{A} \in \mathcal{M}_{m \times n}(\mathbb{F})$
$a_{kl}$ or $[\mathbf{A}]_{kl}$	the $(k, l)$ entry of $\mathbf{A}$
$\mathcal{M}_{m \times 1}(\mathbb{F})$	the set of $m$ -dimensional column vectors over $\mathbb{F}$
$\mathbf{0}_m$	all-zero $m$ -dimensional column vectors
$\mathbf{1}_m$	all-one $m$ -dimensional column vectors
$\mathbf{U}_m^n$	$\mathbf{U}_m^n \in \mathcal{M}_{m-1, m}(\mathbb{F})$ is the matrix defined as:

$$\mathbf{U}_m^n = \begin{pmatrix} \mathbf{I}_{n-1} & \mathbf{0}_{n-1} & \mathbf{0}_{(n-1) \times (m-n)} \\ \mathbf{0}_{(m-n) \times (n-1)} & \mathbf{0}_{m-n} & \mathbf{I}_{m-n} \end{pmatrix}$$

- $\mathbf{x}_{|n}$  the vector  $\mathbf{x}$  with  $n^{th}$  component removed and  $\mathbf{x} \in \mathcal{M}_{m \times 1}(\mathbb{F})$ .  
Reindexed components of  $\mathbf{x}_{|n}$  are denoted  $\{x_{|n;k} : k = 0, \dots, m - 2\}$   
and  $\mathbf{x}_{|n} = \mathbf{U}_m^n \mathbf{x}$ .
- $\mathbf{A}_{|n}$  the matrix  $\mathbf{A}$  with  $n^{th}$  row and  $n^{th}$  column removed and  
 $\mathbf{A} \in \mathcal{M}_m(\mathbb{F})$ . Reindexed entries of  $\mathbf{A}_{|n}$  are denoted by  
 $\{a_{|n;kl} : k, l = 0, \dots, m - 2\}$  and  $\mathbf{A}_{|n} = \mathbf{U}_m^n \mathbf{A} \mathbf{U}_m^{n \dagger}$ .
- $\mathbf{F}_N \in \mathcal{M}_N(\mathbb{C})$  the discrete Fourier transform (DFT) matrix for N-point complex  
signals with entries  $[\mathbf{F}_N]_{kl} = \frac{1}{\sqrt{N}} w^{kl}$ ,  $w = e^{-2i\pi/N}$ ,  $k, l = 0, \dots, N - 1$ .
- $\tilde{\mathbf{x}}$  if  $\mathbf{x} \in \mathcal{M}_{N \times 1}(\mathbb{C})$  represents an N-point complex signal, then  
 $\tilde{\mathbf{x}} = \mathbf{F}_N \mathbf{x}$  denotes the Fourier transform of  $\mathbf{x}$ .

# Résumé Étendu en Français

## Chapitre 1 : Introduction

### Contexte de La Recherche

Au cours des dernières années, les télécommunications sans fil ont changé de manière significative de nombreux aspects de notre monde. Il y'a dix ans, les services de télécommunications mobiles étaient, pour la plupart, dédiés à la communication vocale. Aujourd'hui, cependant, nous comptons sur eux, non seulement pour la communication vocale, mais aussi pour la communication des données. Considérons, par exemple, les e-mails et les services de recherche d'information, ainsi que les nombreux services à usage spécial pour le divertissement, le commerce électronique, l'éducation et la santé. La révolution du sans fil engendre une demande sans cesse croissante de débit de données plus élevé et d'une plus grande mobilité. Les utilisateurs sans fil veulent avoir accès à tout, partout et n'importe quand. Ainsi, les appareils sans fil doivent envoyer et recevoir des flux d'information à haut débit par le biais de canaux variables rapidement dans le temps. Grosso modo, les réflexions de signaux sur des structures telles que des bâtiments, des montagnes, et les véhicules provoquent la sélectivité dans le signal de réception. Les réflexions provoquées par des objets physiques entre l'émetteur et le récepteur créent de la dispersion dans le temps et ces signaux réfléchis sont parfois additionnés de manière destructive, provoquant ce que l'on appelle "multipath fading". Dans ce cas, l'énergie de chaque symbole se disperse et se répand sur les symboles adjacents, que nous appelons inter-symboles-interférence (ISI). Ce canal se comporte comme un filtre, dont la réponse fréquentielle présente une sélectivité de fréquence. Ainsi, il est aussi appelé canal sélectif en fréquence. Quand l'émetteur, le réflecteur et / ou le récepteur sont en

---

mouvement, les signaux réfléchis se déplacent à travers différents canaux à chaque instant. Ainsi, le signal reçu subit l'effet de canal variant dans le temps, c'est-à-dire sélectif en temps. Dans le domaine de fréquence, le spectre du signal reçu subit un phénomène appelé étalement Doppler. Des canaux sélectifs à la fois en temps et en fréquence sont dits doublement sélectifs.

Afin de pallier le comportement du canal et de maintenir une communication fiable avec un taux d'erreurs binaire (BER) acceptable, il faut utiliser des techniques d'égalisation. L'égalisation en général consiste à estimer la réponse, ou l'état, du canal et utiliser l'estimation pour compenser les effets de canal afin d'améliorer les performances du système de transmission. L'égalisation pour le canal doublement sélectif est un problème difficile. Au lieu d'estimer un seul processus aléatoire, comme dans le cas du canal à évanouissement plat variant dans le temps, il y'a de nombreux paramètres à estimer. Ce peut être sous la forme de l'estimation de la réponse impulsionnelle de canal (CIR) vecteur, ou l'estimation des poids adaptatifs pour robinets dans le modèle de ligne à retard. Dans les canaux sélectifs en fréquence, l'égalisation consiste à estimer le CIR, puis à utiliser ces informations pour régler les paramètres d'une forme de filtre linéaire ou non linéaire pour compenser les effets sélectifs en fréquence. Le filtre linéaire peut être sous forme d'un égaliseur transversal, et le filtre non linéaire peut être un égaliseur à décision rétroactive (DFE), un détecteur de séquence de vraisemblance maximale (MLSD), ou un détecteur de type maximum a posteriori (MAP).

La détection fiable de données cohérentes n'est possible que si une estimation de canal précise est disponible à la réception. L'utilisation d'un estimateur, le CIR du canal mobile (variant dans le temps) doit être reconnue. Le filtre de Kalman est l'algorithme d'estimation optimal et peut être utilisé pour le suivi des canaux sans fil à évanouissement. Les techniques d'estimation sous-optimales telles que least mean square (LMS) ou les moindres carrés récursifs (RLS) dans de nombreux systèmes pratiques sont choisies à la place du filtre de Kalman pour réduire les coûts de mise en œuvre. Tous les algorithmes d'estimation nécessitent la séquence de données transmise en tant que données d'entrée de canal. Toutefois, les données transmises ne sont pas disponibles au niveau du récepteur dans des situations pratiques. Ce problème est parfois appelé estimation d'état avec l'incertitude du modèle, où l'es-

timateur de canal doit estimer les états du système linéaire alors que et la séquence de données transmises est inconnue.

Le filtre de Kalman est une approche optimale lorsque les équations sont linéaires et les bruits sont indépendants, additifs et Gaussiens. Pour les scénarios où les modèles sont non linéaires ou le bruit est non Gaussien, diverses méthodes approchées ont été proposées dont le filtre de Kalman étendu est peut-être le plus important. La méthode de filtrage particulière est devenue une alternative importante au filtre de Kalman étendu. En plus de ces deux approches, la procédure EM en tant qu'approche naturelle, est une procédure d'itération des étapes d'espérance et de maximisation. Dans la première étape, l'espérance conditionnelle des données complètes est calculée compte tenu des données incomplètes et des estimations courantes des paramètres. La seconde étape donne de nouvelles estimations des paramètres en maximisant l'espérance conditionnelle sur les paramètres inconnus.

## Contributions

L'objectif de cette thèse est de proposer de nouveaux schémas itératifs de détection pour les deux cas de l'information parfaite et imparfaite d'état de canal au niveau du récepteur. Tout d'abord, afin de réduire la complexité de calcul de MLSE, nous proposons un nouveau détecteur itératif basé sur une approche en continuation qui a utilisé l'espérance-maximisation (EM). Dans ce schéma, nous prenons la forme générale d'une continuation ou processus de recuit déterministe. A partir d'une formulation exacte de l'algorithme, nous obtenons une série d'approximations, conduisant à une structure finale qui a quelques similitudes avec l'égaliseur classique minimum mean square error (MMSE) itératif par blocs.

Deuxièmement, en vue de la détection de données sous un canal dynamique linéaire, lorsque le canal est inconnu au niveau du récepteur, le récepteur doit être en mesure d'effectuer conjointement l'égalisation et l'estimation de canal. De cette manière, on formule une représentation de modèle état-espace combiné du système de communication. Par cette représentation, nous pouvons utiliser le filtre de Kalman comme le meilleur estimateur des paramètres du canal. Le but de ce chapitre est de motiver de façon rigoureuse la mise en place du filtre de Kalman dans l'estimation des séquences de Markov par des canaux dynamiques Gaussien. Par la présente,



nous interprétons et explicitons les approximations sous-jacentes dans les approches heuristique.

Finalement, si l'on considère un canal dynamique non linéaire nous ne pouvons pas utiliser le filtre de Kalman comme un meilleur estimateur. Ici, nous utilisons des modèles à commutation d'espace à état (SSSM) comme les modèles espace-état non linéaires. Ce modèle combine le modèle de Markov caché (HMM) et le modèle espace-état linéaire (LSSM). Afin d'estimer le canal et de détecter les données, la procédure EM est utilisée comme méthode naturelle. De cette façon EKF et les filtres à particules sont évités.

## Chapitre 2 : Une Approche de Continuation l'Estimation Itérative de Séquence

### Introduction

L'estimation "maximum de vraisemblance" de séquences (MLSE) de symboles discrets issus de la sortie bruyante d'un canal convolutif a toujours motivé une recherche relativement intensive. Comme on le sait, la complexité de MLSE étant exponentielle en la mémoire des canaux et en la cardinalité de l'alphabet des symboles, elle est rapidement ingérable et force à recourir à des approches sous-optimales. La structure que nous obtenons dans ce chapitre repose principalement sur la conversion du problème de recherche discrète, d'origine, en un problème d'estimation de paramètres continu. En général, l'idée d'approcher le coût d'une fonction d'optimisation d'origine par une séquence de fonctions plus traitables est connue sous le nom de continuation ou de recuit déterministe. Dans notre contexte, un tel processus peut être formulé comme suit : les mesures communes de Dirac centrées sur les symboles de constellation sont remplacées par une séquence de mélanges de Gaussiennes convergeant vers la première. Également, la probabilité a posteriori (APP) originale de symboles (discrète par sa nature) est définie comme la limite d'une suite d'autres continues. Au début du processus de recuit déterministe, la variance (ou température) de la Gaussienne est choisie suffisamment élevée de manière à assurer un maximum unique pour l'APP transformé. Cet optimum est atteint par quelques itérations de l'espérance-maximisation (EM) avec un choix approprié des variables cachées. La variance est ensuite diminuée assez lentement pour faire en sorte que la dernière estimation du bloc de données reste dans le bassin d'attraction du nouveau maximum absolu. Une nouvelle estimation est calculée et l'ensemble du processus est itéré jusqu'à une valeur suffisamment petite de la variance. De cette façon, une séquence d'estimations a posteriori est construite convergente vers la distribution de probabilité a posteriori discrète maximale. Ce chapitre expose la construction et la mise en œuvre de ce processus. D'abord, nous formulons le problème d'estimation de séquence dans un cadre précis et la formule d'itération principale est calculée en la séquence. L'utilisation d'un préambule cyclique permet de réaliser la sommation

contenue dans la dite formule d'itération principale dans le domaine des fréquences, alors que le recours à une approximation Gaussienne permet de simplifier son évaluation. Enfin, les principales caractéristiques de notre approche sont résumées dans la conclusion.

## Aperçu de La Méthode

Soit  $\mathbf{x}$  un vecteur de dimension  $N$  dont les composantes appartiennent à un alphabet fini  $\mathcal{X}$ . Les composantes (parfois appelés symboles de constellation) de  $\mathbf{x}$  sont indépendantes et uniformément distribuées (iid). On considère la transmission en bloc de  $\mathbf{x}$  sur un canal convolutif single-input single-output (SISO) avec une mémoire  $M$ . Le vecteur reçu équivalent en bande de base, à temps discret et de dimension  $N$  est donné par

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{w} \quad (1)$$

où  $\mathbf{H}$  est la matrice de Toeplitz du canal de dimension  $N \times N$  et  $\mathbf{w}$  est un vecteur de bruit additif modélisé comme une Gaussienne circulaire symétrique centrée de dimension  $N$ , avec matrice de covariance  $\mathbb{E}\{\mathbf{w}\mathbf{w}^\dagger\} = \beta\mathbf{I}_N$ . Nous supposons que le canal est inconnu à l'émetteur et parfaitement connu au récepteur. La fonction objectif à maximiser est la probabilité conjointe conditionnelle  $p(\mathbf{x}|\mathbf{y}, \mathbf{H})$ , c'est à dire, nous voulons savoir :

$$\mathbf{x}^* = \arg \max_{\mathbf{x} \in \mathcal{X}^N} p(\mathbf{x}|\mathbf{y}, \mathbf{H}) \quad (2)$$

Il s'agit d'un problème d'optimisation discrète, dont la résolution exhaustive est intraitable pour des  $\mathcal{X}$  et  $N$  grands. Dans la suite, le conditionnement par  $\mathbf{H}$  est implicite et omis dans les expressions de probabilités pour des raisons de simplicité de notation.

### L'Approche Poursuite :

Nous proposons de reformuler le problème d'optimisation discrète d'origine comme suit

$$\mathbf{x}^{(\alpha)*} = \arg \max_{\mathbf{x} \in \mathcal{C}^N} p^{(\alpha)}(\mathbf{y}, \mathbf{x}) \quad (3)$$

où

$$p^{(\alpha)}(\mathbf{y}, \mathbf{x}) = p(\mathbf{y}|\mathbf{x})p^{(\alpha)}(\mathbf{x}) \quad (4)$$

## Résumé étendu en Français

---

est une famille de pdfs dans laquelle la probabilité  $p(\mathbf{y}|\mathbf{x})$  est donnée par :

$$p(\mathbf{y}|\mathbf{x}) = (\pi\beta)^{-N} e^{-\frac{1}{\beta}\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2} \quad (5)$$

et dans laquelle le support de la pdf  $p^{(\alpha)}(\mathbf{x})$  converge vers le support de la fonction de masse de probabilité (pmf)  $p(\mathbf{x})$  comme  $\alpha \rightarrow 0^+$ . Pour plus de commodité de calcul, la pdf  $p^{(\alpha)}(\mathbf{x})$  est définie comme

$$p^{(\alpha)}(\mathbf{x}) = \sum_{\mathbf{s}} p^{(\alpha)}(\mathbf{x}, \mathbf{s}) = \sum_{\mathbf{s}} p(\mathbf{s}) p^{(\alpha)}(\mathbf{x}|\mathbf{s}) \propto \sum_{\mathbf{s}} p(\mathbf{s}) e^{-\frac{1}{\alpha}\|\mathbf{H}(\mathbf{x}-\mathbf{s})\|^2} \quad (6)$$

où la somme s'étend sur  $\mathcal{X}^N$ . De toute évidence, cette famille de pdfs obéit au critère limitant quand  $\alpha \rightarrow 0^+$ .

### La Maximisation Itérative en Utilisant L'Algorithme EM

Nous déclarons une séquence  $\mathbf{s} \in \mathcal{X}^N$  de symboles de constellation comme une variable aléatoire cachée. La *log* pdf  $\log p^{(\alpha)}(\mathbf{y}, \mathbf{x})$  peut être réécrite comme

$$\begin{aligned} \log p^{(\alpha)}(\mathbf{y}, \mathbf{x}) &= \log p(\mathbf{y}|\mathbf{x}) + \log p^{(\alpha)}(\mathbf{x}) \\ &= \log p(\mathbf{y}|\mathbf{x}) + \log \sum_{\mathbf{s}} p^{(\alpha)}(\mathbf{x}, \mathbf{s}) \\ &= \log p(\mathbf{y}|\mathbf{x}) + \log \sum_{\mathbf{s}} p^{(\alpha)}(\mathbf{s}|\mathbf{x}') \frac{p^{(\alpha)}(\mathbf{x}, \mathbf{s})}{p^{(\alpha)}(\mathbf{s}|\mathbf{x}')} \end{aligned} \quad (7)$$

avec  $\mathbf{x}'$  une estimation initiale de la séquence  $\mathbf{x}$ . Puis, en utilisant l'inégalité de Jensen [51], elle peut être minorée en

$$\log p^{(\alpha)}(\mathbf{y}, \mathbf{x}) \geq \chi - \frac{1}{\beta}\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 + \sum_{\mathbf{s}} p^{(\alpha)}(\mathbf{s}|\mathbf{x}') \log p^{(\alpha)}(\mathbf{x}, \mathbf{s})$$

où  $\chi$  est un groupe de mots indépendants de  $\mathbf{x}$ . Enfin, la séquence (continue)  $\mathbf{x}$  est sélectionnée comme celle correspondante à la maximisation de la limite inférieure au dessus, c'est-à dire au zéro de son gradient

$$\frac{1}{\beta}\{\mathbf{H}^\dagger \mathbf{y} - \mathbf{H}^\dagger \mathbf{H}\mathbf{x}\} + \mathbb{E}_{\mathbf{s}|\mathbf{x}'}[\nabla_{\mathbf{x}} \log p^{(\alpha)}(\mathbf{x}|\mathbf{s})] = 0 \quad (8)$$

où l'opérateur  $\mathbb{E}_{\mathbf{s}|\mathbf{x}'}$  désigne l'espérance sur toutes les séquences  $\mathbf{s}$  par rapport à la mesure de probabilité  $p^{(\alpha)}(\mathbf{s}|\mathbf{x}')$ . En appliquant la règle de Bayes, cette dernière peut être exprimée comme

$$p^{(\alpha)}(\mathbf{s}|\mathbf{x}') = \frac{p(\mathbf{s})p^{(\alpha)}(\mathbf{x}'|\mathbf{s})}{\sum_{\mathbf{s}'} p(\mathbf{s}')p^{(\alpha)}(\mathbf{x}'|\mathbf{s}')} \quad (9)$$

De l'équation (8), nous obtenons la formule de récurrence

$$\mathbf{x} = \gamma(\mathbf{H}^\dagger \mathbf{H})^{-1} \mathbf{H}^\dagger \mathbf{y} + (1 - \gamma) \mathbb{E}_{\mathbf{s}|\mathbf{x}'}[\mathbf{s}] \quad (10)$$

où  $\gamma$  est le paramètre défini comme

$$\gamma = \left(1 + \frac{\beta}{\alpha}\right)^{-1} \in ]0, 1[ \quad (11)$$

qui diminue progressivement à mesure que  $\alpha \rightarrow 0^+$  à chaque itération (et en  $\mathbf{x}$   $\beta$  fixes) et où

$$\mathbb{E}_{\mathbf{s}|\mathbf{x}'}[\mathbf{s}] = \frac{\sum_{\mathbf{s}} \mathbf{s} p(\mathbf{s}) e^{-\frac{1}{\alpha} \|\mathbf{H}(\mathbf{x}' - \mathbf{s})\|^2}}{\sum_{\mathbf{s}} p(\mathbf{s}) e^{-\frac{1}{\alpha} \|\mathbf{H}(\mathbf{x}' - \mathbf{s})\|^2}} \quad (12)$$

Notez que  $p(\mathbf{s}) = |\mathcal{X}|^{-N}$  pour des symboles de constellation indépendants et uniformément distribués.

## Approximations

En vue d'atteindre une complexité polynomiale en la mémoire des canaux (et en la cardinalité de l'alphabet), l'évaluation de la somme-sur-états (12) doit être approximée. Les procédures comme le champ moyen standard [75] ont été essayées et abandonnées car les résultats sont plutôt médiocres. Dans la suite, nous modifions légèrement le problème initial en supposant que le préambule cyclique a été successivement inséré dans la séquence émise  $\mathbf{x}$ . Cela rend le carré de la matrice de Toeplitz  $\mathbf{H}$  circulaire. Introduisons la matrice de Gram  $\mathbf{G} = \mathbf{H}^\dagger \mathbf{H}$  et le vecteur  $\xi = \mathbf{G}\mathbf{x}'$ . Nous avons

$$\begin{aligned} \mathbb{E}_{\mathbf{s}|\mathbf{x}'}[s_n] &= \sum_a a p(s_n = a | \mathbf{x}') = \sum_a a \sum_{\mathbf{s}_{|n}} p(s_n = a, \mathbf{s}_{|n} | \mathbf{x}') \\ &= \sum_a a p(s_n = a) \sum_{\mathbf{s}_{|n}} p^{(\alpha)}(\mathbf{x}' | s_n = a, \mathbf{s}_{|n}) p(\mathbf{s}_{|n}) \end{aligned} \quad (13)$$

La matrice  $\mathbf{H}$  étant circulaire,  $\mathbf{G}$  est circulaire et peut être ramenée à la forme diagonale par biais de la matrice DFT :  $\mathbf{F}_N$ . Ce résultat n'est plus valable pour la matrice tronquée

$$\mathbf{G}_{|n} = \mathbf{U}_N^n \mathbf{H}^\dagger \mathbf{H} \mathbf{U}_N^{n\dagger} \quad (14)$$

Néanmoins,  $\mathbf{G}_{|n}$  étant normale, il existe une matrice unitaire  $\mathbf{Q}_{N-1}^n$  telle que

$$\mathbf{\Lambda}^n = \text{diag}\{\lambda_1^n, \dots, \lambda_{N-1}^n\} = \mathbf{Q}_{N-1}^n \mathbf{G}_{|n} \mathbf{Q}_{N-1}^{n\dagger} \quad (15)$$

## Résumé étendu en Français

---

Soient  $\bar{\xi}_{|n}$ ,  $\bar{\mathbf{s}}_{|n}$  et  $\bar{\mathbf{g}}_{n|n}$  des vecteurs définis comme  $\mathbf{Q}_{N-1}^n \xi_{|n}$ ,  $\mathbf{Q}_{N-1}^n \mathbf{s}_{|n}$  et  $\mathbf{Q}_{N-1}^n \mathbf{g}_{n|n}$ , respectivement. à partir de

$$\begin{aligned} & \sum_{\mathbf{s}_{|n}} p^{(\alpha)}(\mathbf{x}' | s_n = a, \mathbf{s}_{|n}) p(\mathbf{s}_{|n}) \propto \\ & \sum_{\mathbf{s}_{|n}} p(\mathbf{s}_{|n}) e^{\frac{1}{\alpha} [2\text{Re}(\xi_n^* a + \xi_{|n}^\dagger \mathbf{s}_{|n} - a^* \mathbf{g}_{n|n}^\dagger \mathbf{s}_{|n}) - g_{nn} |a|^2 - \mathbf{s}_{|n}^\dagger \mathbf{G}_{|n} \mathbf{s}_{|n}]} \end{aligned} \quad (16)$$

nous allons d'abord reformuler la somme

$$\begin{aligned} & \sum_{\mathbf{s}_{|n}} p^{(\alpha)}(\mathbf{x}' | s_n = a, \mathbf{s}_{|n}) p(\mathbf{s}_{|n}) \propto \\ & e^{\frac{1}{\alpha} [2\text{Re}(\xi_n^* a) - g_{nn} |a|^2]} \sum_{\mathbf{s}_{|n}} p(\mathbf{s}_{|n}) e^{\frac{1}{\alpha} [2\text{Re}(\xi_{|n}^\dagger \mathbf{s}_{|n} - a^* \mathbf{g}_{n|n}^\dagger \mathbf{s}_{|n}) - \mathbf{s}_{|n}^\dagger \mathbf{G}_{|n} \mathbf{s}_{|n}]} \end{aligned} \quad (17)$$

ou de manière équivalente, après le changement de variable  $\bar{\mathbf{s}}_{|n} = \mathbf{Q}_{N-1}^n \mathbf{s}_{|n}$ , comme

$$\begin{aligned} & \sum_{\mathbf{s}_{|n}} p^{(\alpha)}(\mathbf{x}' | s_n = a, \mathbf{s}_{|n}) p(\mathbf{s}_{|n}) \propto \\ & e^{\frac{1}{\alpha} [2\text{Re}(\xi_n^* a) - g_{nn} |a|^2]} \sum_{\bar{\mathbf{s}}_{|n}} p(\bar{\mathbf{s}}_{|n}) e^{\frac{1}{\alpha} [2\text{Re}(\bar{\xi}_{|n}^\dagger \bar{\mathbf{s}}_{|n} - a^* \bar{\mathbf{g}}_{n|n}^\dagger \bar{\mathbf{s}}_{|n}) - \bar{\mathbf{s}}_{|n}^\dagger \mathbf{\Lambda}^n \bar{\mathbf{s}}_{|n}]} \end{aligned} \quad (18)$$

Concentrons-nous maintenant sur la somme discrète

$$\sum_{\bar{\mathbf{s}}_{|n}} p(\bar{\mathbf{s}}_{|n}) e^{\frac{1}{\alpha} [2\text{Re}(\bar{\xi}_{|n}^\dagger \bar{\mathbf{s}}_{|n} - a^* \bar{\mathbf{g}}_{n|n}^\dagger \bar{\mathbf{s}}_{|n}) - \bar{\mathbf{s}}_{|n}^\dagger \mathbf{\Lambda}^n \bar{\mathbf{s}}_{|n}]} \quad (19)$$

La théorie (théorème de la limite central) et les simulations numériques montrent que, pour  $N$  grand (typiquement  $N \geq 64$ ), les composantes de  $\bar{\mathbf{s}}_{|n}$  sont des Gaussiennes de moyenne nulle et de covariance unité. Nous allons donc appliquer une approximation Gaussienne pour approximer davantage (19) :

$$\begin{aligned} & \sum_{\bar{\mathbf{s}}_{|n}} p(\bar{\mathbf{s}}_{|n}) e^{\frac{1}{\alpha} [2\text{Re}(\bar{\xi}_{|n}^\dagger \bar{\mathbf{s}}_{|n} - a^* \bar{\mathbf{g}}_{n|n}^\dagger \bar{\mathbf{s}}_{|n}) - \bar{\mathbf{s}}_{|n}^\dagger \mathbf{\Lambda}^n \bar{\mathbf{s}}_{|n}]} \\ & = \pi^{-(N-1)} \int_{\mathbb{C}^{N-1}} e^{-\|\bar{\mathbf{s}}_{|n}\|^2} e^{\frac{1}{\alpha} [2\text{Re}(\bar{\xi}_{|n}^\dagger \bar{\mathbf{s}}_{|n} - a^* \bar{\mathbf{g}}_{n|n}^\dagger \bar{\mathbf{s}}_{|n}) - \bar{\mathbf{s}}_{|n}^\dagger \mathbf{\Lambda}^n \bar{\mathbf{s}}_{|n}]} \mathbf{dV}(\bar{\mathbf{s}}_{|n}) \end{aligned} \quad (20)$$

Après un peu d'algèbre, le terme de droite peut être réécrit comme

$$\begin{aligned} & \pi^{-(N-1)} e^{\frac{1}{\alpha} [(\bar{\xi}_{|n} - a \bar{\mathbf{g}}_{n|n})^\dagger (\mathbf{\Lambda}^n + \alpha \mathbf{I}_{N-1})^{-1} (\bar{\xi}_{|n} - a \bar{\mathbf{g}}_{n|n})]} \times \\ & \int_{\mathbb{C}^{N-1}} e^{-\frac{1}{\alpha} [(\bar{\mathbf{s}}_{|n} - (\mathbf{\Lambda}^n + \alpha \mathbf{I}_{N-1})^{-1} (\bar{\xi}_{|n} - a \bar{\mathbf{g}}_{n|n}))^\dagger (\mathbf{\Lambda}^n + \alpha \mathbf{I}_{N-1}) (\bar{\mathbf{s}}_{|n} - (\mathbf{\Lambda}^n + \alpha \mathbf{I}_{N-1})^{-1} (\bar{\xi}_{|n} - a \bar{\mathbf{g}}_{n|n}))]} \mathbf{dV}(\bar{\mathbf{s}}_{|n}) \end{aligned} \quad (21)$$

Et enfin, la somme initiale s'évalue en :

$$\sum_{\mathbf{s}_{|n}} p^{(\alpha)}(\mathbf{x}' | s_n = a, \mathbf{s}_{|n}) p(\mathbf{s}_{|n}) \propto e^{\frac{1}{\alpha} [2\text{Re}(\xi_n^* a) - g_{nn}|a|^2 + (\bar{\xi}_{|n} - a \bar{\mathbf{g}}_{n|n})^\dagger (\Lambda^n + \alpha \mathbf{I}_{N-1})^{-1} (\bar{\xi}_{|n} - a \bar{\mathbf{g}}_{n|n})]} \quad (22)$$

et

$$\propto e^{\frac{1}{\alpha} [|a|^2 (\bar{\mathbf{g}}_{n|n}^\dagger (\Lambda^n + \alpha \mathbf{I}_{N-1})^{-1} \bar{\mathbf{g}}_{n|n} - g_{nn}) + 2\text{Re}(a(\xi_n^* - \bar{\xi}_{|n}^\dagger (\Lambda^n + \alpha \mathbf{I}_{N-1})^{-1} \bar{\mathbf{g}}_{n|n}))]} \quad (23)$$

ou, de façon équivalente,

$$\propto e^{\frac{1}{\alpha} [|a|^2 (\mathbf{g}_{n|n}^\dagger \mathbf{Q}_{N-1}^{n\dagger} (\Lambda^n + \alpha \mathbf{I}_{N-1})^{-1} \mathbf{Q}_{N-1}^n \mathbf{g}_{n|n} - g_{nn}) + 2\text{Re}(a(\xi_n^* - \xi_{|n}^\dagger \mathbf{Q}_{N-1}^{n\dagger} (\Lambda^n + \alpha \mathbf{I}_{N-1})^{-1} \mathbf{Q}_{N-1}^n \mathbf{g}_{n|n}))]} \quad (24)$$

Notez que, pour  $\mathbf{x}$ , l'expression se simplifie comme

$$\mathbb{E}_{\mathbf{s}|\mathbf{x}'}[s_n] = \tanh \left\{ \frac{2}{\alpha} \text{Re}(\xi_n^* - \bar{\xi}_{|n}^\dagger (\Lambda^n + \alpha \mathbf{I}_{N-1})^{-1} \bar{\mathbf{g}}_{n|n}) \right\} \quad (25)$$

ou

$$\mathbb{E}_{\mathbf{s}|\mathbf{x}'}[s_n] = \tanh \left\{ \frac{2}{\alpha} \text{Re}(\xi_n^* - \xi_{|n}^\dagger \mathbf{Q}_{N-1}^{n\dagger} (\Lambda^n + \alpha \mathbf{I}_{N-1})^{-1} \mathbf{Q}_{N-1}^n \mathbf{g}_{n|n}) \right\} \quad (26)$$

Comme on le voit (26) la structure d'un retour de décision douce convergeant vers un dur Limiteur dur  $\alpha \rightarrow 0^+$ . Ce n'est rien d'autre que l'interpolateur Wiener pour l'estimation MMSE de l'échantillon  $n^{\text{th}}$  de la réponse du canal à partir des autres  $N-1$  dans le bruit fictif de variance  $\alpha$ . D'où le terme entre parenthèses est constitué essentiellement d'un annulateur d'interférences MMSE raffiné. Pour  $\alpha$  fixe, les  $N$  réponses impulsionnelles  $\xi_{|n}^\dagger$  sont déduites par permutation circulaire des éléments constitutifs de l'une d'entre elles.

### Choix d'Une Base Appropriée

Des simulations numériques montrent que pour  $n$  grand et pour  $M \ll N$ , les matrices tronquées  $\mathbf{G}_{|n}$  peuvent être quasi-diagonalisées dans la base de Fourier  $\mathbf{F}_{N-1}$ . Ceci justifie la seconde approximation

$$\mathbf{G}_{|n} \approx \mathbf{F}_{N-1}^\dagger \Lambda^n \mathbf{F}_{N-1} \quad (27)$$

c'est à dire,  $\mathbf{Q}_{N-1}^n \approx \mathbf{F}_{N-1}$ ,  $\forall n$

## Questions de Convergence et de Stabilité

Cette section vise à discuter de diverses questions de convergence et de stabilité de l'algorithme proposé. Pour cela, il est nécessaire de développer une formule complète de la probabilité conjointe.

### Convergence du Processus de Recuit

Rappelons que  $p^{(\alpha)}(\mathbf{x})$  est une famille de distributions normales convergeant vers

$$\sum_{\mathbf{s}} p(\mathbf{s})\delta(\mathbf{x} - \mathbf{s}) \quad (28)$$

lorsque  $\alpha \rightarrow 0^+$  où  $\mathbf{s}$  passe sur  $\mathcal{X}^N$ . Plus explicitement, pour  $\mathbf{x} \in \mathbb{C}^N$ ,

$$\begin{aligned} p^{(\alpha)}(\mathbf{x}, \mathbf{y}) &= p(\mathbf{y}|\mathbf{x})p^{(\alpha)}(\mathbf{x}) \\ &= (\pi\beta)^N e^{-\frac{1}{\beta}\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2} \sum_{\mathbf{s}} p(\mathbf{s})(\pi\alpha)^{-N} e^{-\frac{1}{\alpha}\|\mathbf{H}(\mathbf{x}-\mathbf{s})\|^2} \end{aligned} \quad (29)$$

Après un peu d'algèbre, nous arrivons à l'expression équivalente

$$p^{(\alpha)}(\mathbf{x}, \mathbf{y}) = \sum_{\mathbf{s}} Q^\alpha(\mathbf{s})(\pi\beta\gamma)^{-N} e^{-\frac{1}{\beta\gamma}\|\mathbf{H}\mathbf{x}-J^{(\alpha)}(\mathbf{s})\|^2} \quad (30)$$

dans laquelle

$$Q^{(\alpha)}(\mathbf{s}) = p(\mathbf{s})(\pi(\alpha + \beta))^{-N} e^{-\frac{1}{(\alpha+\beta)}\|\mathbf{y}-\mathbf{H}\mathbf{s}\|^2} \quad (31)$$

et

$$J^{(\alpha)}(\mathbf{s}) = \gamma\mathbf{y} + (1 - \gamma)\mathbf{H}\mathbf{s} \quad (32)$$

Afin d'analyser la convergence du processus de recuit vers l'estimation ML  $\mathbf{x}^*$ , nous supposons une diminution continue de la température  $\alpha$ . De ces dernières expressions, il est évident que lorsque  $\alpha \rightarrow \infty, \gamma \rightarrow 1^-$ ,  $p^{(\alpha)}(\cdot, \mathbf{y})$  possède un unique maximum à  $\mathbf{x} = \mathbf{x}_\infty = \mathbf{y}$ . Inversement, lorsque  $\alpha \rightarrow 0^+, \gamma \rightarrow 0^+$ ,  $p^{(\alpha)}(\cdot, \mathbf{y})$  possède  $|\mathcal{X}|^N$  maxima situés aux sommets de la maille  $\mathcal{X}^N$ , avec un maximum global arbitrairement proche de l'estimation ML  $\mathbf{x} = \mathbf{x}^* = \arg \max_{\mathbf{s}} p(\mathbf{s})(\pi\beta)^{-N} e^{-\frac{1}{\beta}\|\mathbf{y}-\mathbf{H}\mathbf{s}\|^2}$ . Ainsi, la densité  $p^{(\alpha)}(\mathbf{x}, \mathbf{y})$  se comporte correctement aux limites. Cela implique également l'existence d'une suite décroissante de températures singulières  $\infty > \bar{\alpha}_0 > \bar{\alpha}_1 > \dots$  de telle sorte que le nombre de maxima est constant dans chaque intervalle ouvert  $\mathbb{I}_i = (\bar{\alpha}_{i+1}, \bar{\alpha}_i)$  et subit une variation (de transition de phase ou bifurcation) entre



deux intervalles successifs. La proposition suivante concerne le comportement du processus de recuit dans le domaine régulier des températures  $\mathbb{A} = \bigcup_i \mathbb{I}_i$ .

**Proposition 1.** Soit  $\mathbf{x}^{(\alpha_0)*} = \arg \max_{\mathbf{x}} p^{(\alpha_0)}(\mathbf{x}, \mathbf{y})$  pour tout  $\alpha_0 \in \mathbb{I}_{i \geq 0}$ . Soit  $\alpha \mapsto \mathbf{x}^*(\alpha)$  une courbe continue de telle sorte que  $\mathbf{x}^*(\alpha = \alpha_0) = \mathbf{x}^{(\alpha_0)*}$  et de telle sorte que  $\mathbf{x}^*(\alpha)$  est la solution de l'équation  $\nabla_{\mathbf{x}} p^{(\alpha)}(\mathbf{x}, \mathbf{y}) = 0$  pour tout  $\bar{\alpha}_{i+1} < \alpha < \alpha_0$ . Ensuite, pour la quasi-totalité réalisation de  $\mathbf{y}$ ,  $\mathbf{x}^*(\alpha) = \arg \max_{\mathbf{x}} p^{(\alpha)}(\mathbf{x}, \mathbf{y})$  sur l'intervalle  $(\bar{\alpha}_{i+1}, \alpha_0]$ .

En gros, cette proposition signifie que si  $\mathbf{x}^*(\alpha)$  est initialisé sur le lieu du maximum global pour une certaine valeur de la température et verrouillé sur le déplacement de ce maximum quand  $\alpha$  diminue, aucun accident ne peut se produire entre au moins deux températures singulières successives, à savoir, la possibilité que le maximum suivi devienne sous-optimal pour certains  $\alpha$  ultérieurs est exclue. Notez de nouveau que cette proposition est valable pour un processus idéalisé où la température est diminuée de façon continue. Comme (pour les problèmes de complexité évidentes) l'algorithme itératif proposé ne considère un nombre fini de chutes de température, il existe une possibilité de perdre la trace de maximum global si les diminutions ne sont pas assez petit.

### Convergence de L'Algorithme Itératif.

Nous nous concentrons maintenant sur  $\nabla_{\mathbf{x}} p^{(\alpha)}(\mathbf{x}, \mathbf{y}) = 0$ . En prenant la dérivée partielle de (30) nous obtenons l'équation de point fixe

$$\mathbf{x} = \varphi(\mathbf{x}) = \gamma(\mathbf{H}^\dagger \mathbf{H})^{-1} \mathbf{H}^\dagger \mathbf{y} + (1 - \gamma) \bar{\mathbf{s}}(\mathbf{x}) \quad (33)$$

dans lequel

$$\bar{\mathbf{s}}(\mathbf{x}) = \sum_{\mathbf{s}} \mathbf{s} \varpi(\mathbf{s}) \quad (34)$$

et

$$\varpi(\mathbf{s}) = \frac{p(\mathbf{s}) e^{-\frac{1}{\alpha} \|\mathbf{H}(\mathbf{x} - \mathbf{s})\|^2}}{\sum_{\mathbf{s}'} p(\mathbf{s}') e^{-\frac{1}{\alpha} \|\mathbf{H}(\mathbf{x} - \mathbf{s}')\|^2}} \quad (35)$$

**Proposition 2.** L'équation de point fixe (33) a une solution pour tous  $\alpha > 0$ .

Rappelez-vous que l'algorithme itératif procède par alternance de deux phases, à savoir la recherche gradient du maximum de  $\ln p^{(\alpha)}(\cdot, \mathbf{y})$  et le refroidissement. Le taux de la convergence est conditionné par le Hessien de  $\bar{\mathbf{s}}(\mathbf{x})$ . En effet, avec  $\alpha$  fixe,

nous obtenons de (33) et le Théorème des accroissements finis

$$\|\mathbf{x}_{p+1} - \mathbf{x}_p\| = (1 - \gamma)\|\bar{\mathbf{s}}(\mathbf{x}_p) - \bar{\mathbf{s}}(\mathbf{x}_{p-1})\| = (1 - \gamma)\|\Phi(\bar{\mathbf{x}})(\mathbf{x}_p - \mathbf{x}_{p-1})\| \quad (36)$$

Ici,  $\bar{\mathbf{x}}$  est un point sur le segment joignant  $\mathbf{x}_{p-1}$  et  $\mathbf{x}_p$  (à itérations  $p - 1$  et  $p$ ) et  $\Phi$  est la Hessian de  $\bar{\mathbf{s}}(\mathbf{x})$  exprimé en

$$\Phi(\mathbf{x}) = \frac{1}{\alpha}\Lambda(\mathbf{x})\mathbf{H}^\dagger\mathbf{H} \quad (37)$$

où

$$\Lambda(\mathbf{x}) = \sum_{\mathbf{s}} \varpi(\mathbf{s})\mathbf{s}\mathbf{s}^\dagger - \left\{ \sum_{\mathbf{s}} \varpi(\mathbf{s})\mathbf{s} \right\} \left\{ \sum_{\mathbf{s}} \varpi(\mathbf{s})\mathbf{s} \right\}^\dagger \quad (38)$$

a la structure d'une matrice de covariance. Pour que les itérations soient convergents, il faut veiller à ce que le spectre de  $\Phi$  soit strictement délimité par l'unité.

**Proposition 3.** Le spectre de  $\Phi$  est strictement délimité par l'unité pour la gamme intermédiaire des températures.

Ce résultat ne tient pas pour  $\alpha$  très haut et  $\alpha$  très bas en raison de la non validité du théorème de limite centrale. Néanmoins, la proposition suivante permet d'obtenir une image globale de la vitesse de convergence dans toutes les gammes. Notons  $\lambda_{\max}$  la plus grande valeur propre de  $\Lambda(\mathbf{x})$ . En raison du caractère dissipatif du canal, le spectre de  $\Phi$  est délimité par  $\lambda_{\max}/\alpha$ .

**Proposition 4.** Pour presque tous les  $\mathbf{x}$ , les limites suivantes  $\lim_{\alpha \rightarrow 0} \frac{\lambda_{\max}}{\alpha} = \lim_{\alpha \rightarrow \infty} \frac{\lambda_{\max}}{\alpha} = 0$  sont atteintes avec un taux exponentiel.

## Conclusion

Dans ce chapitre, une nouvelle approche a été présentée pour résoudre le problème de MLSE de symboles discrets de la sortie bruyante d'un canal convolutif. Une caractéristique de notre méthode est son attrait théorique, à savoir le fait que sa structure itérative émerge naturellement des approximations successives continues de l'APP ciblée discrète exacte sur les symboles et donc ne nécessite aucune hypothèse ad hoc sur sa forme définitive par opposition à [73] [77]. Nous avons montré que, dans certaines circonstances, une version approchée de l'algorithme donne la forme d'un annulateur d'interférence itératif par blocs MMSE avec retour de décision souple. Le recours au calcul dans le domaine de Fourier s'avère crucial pour réduire

la complexité. Compte tenu de sa nature intrinsèquement probabiliste et son rapprochement de la performance optimale à faible SNR, nous pensons que l'égaliseur proposé est parfaitement adapté pour les turbo-égalisations des symboles codés.

## Chapitre 3 : Estimation des Canaux Dynamiques Linéaires basée sur Le Filtrage de Kalman

### Introduction

Le MLSE en tant que récepteur optimal et qui a été proposé par Forney pour les canaux ISI peut être mis en pratique par l'algorithme de Viterbi basé sur une réponse impulsionnelle du canal connue (CIR). L'algorithme de Viterbi initialement proposé par Viterbi pour le décodage à maximum de vraisemblance des codes convolutifs est un cas particulier de la programmation dynamique vers l'avant. L'algorithme de Viterbi trouve les trajectoires optimales (chemins survivants) à chaque étape pour tous les états dans le diagramme en treillis. Cet algorithme a été utilisé par Forney dans le récepteur MLSE pour détecter un signal numérique transmis par le canal ISI corrompu par un bruit additif Gaussien. L'algorithme MLSE calcule le coût (probabilité d'erreur) à travers toutes les trajectoires de chaque état à l'étape suivante pour tous les états possibles (calcul des métriques de branche), puis trouve la trajectoire au coût minimum (chemin survivant) pour chaque état. Donc à l'étape finale le chemin de meilleure survie correspond à la séquence de données avec la probabilité minimale d'erreur de séquence. Afin de mettre en œuvre le MLSE en utilisant l'algorithme de Viterbi, nous avons besoin de la connaissance du canal qui est pratiquement inconnu au niveau du récepteur et doit être estimée. Pour résoudre ce problème, des méthodes communes de l'estimation de canaux et de détection des données ont été proposées. Certains algorithmes d'estimation tels que moindres carrés moyens (LMS), moindres carrés récursif (RLS) et le filtrage de Kalman ont été utilisés dans l'estimation de canal. Lorsque l'algorithme de Viterbi est utilisé en raison du retard inhérent à la décision, nous avons une pauvre estimation du canal dans un environnement variant dans le temps. L'idée de per survivor processing (PSP) a été proposée pour lutter contre le problème des retards de décision, où chaque chemin survivant dans le diagramme en treillis a son propre estimateur. La précision et les propriétés de convergence de LMS déterminent la performance globale de l'algorithme PSP. Dans le décodeur PSP, le least mean square (LMS), les moindres carrés récursifs (RLS) et des algorithmes de filtre de Kalman peuvent

être utilisés pour estimer les paramètres du canal. Il sera montré que les paramètres d'état de l'espace objet peuvent être facilement obtenus au niveau du récepteur par l'estimation du décalage de fréquence Doppler maximum ou de manière équivalente en trouvant estimation spectrale AR du CIR. Cela nous permet d'utiliser le filtre de Kalman optimal pour l'estimation de canal.

## Modèle de Système

Nous considérons la transmission de bloc sur une single-input single-output (SISO) doubly selective channel (DSC) (longueur de bloc :  $N$ , la mémoire de canal :  $M$ , décalage de fréquence Doppler maximale :  $f_D$ , durée de symbole :  $T$ ) et binary phase shift keying (BPSK) modulation, de sorte que le bit est transmis à l'instant  $n$ ,  $b_n \in \{-1, +1\}$ .

Nous supposons un canal discret Rayleigh de mémoire  $M$ , simulé avec une méthode introduite dans [92], où les éléments de la réponse impulsionnelle  $\{c_n^i\}_{i=0}^M$  sont modélisés comme des variables aléatoires complexes de moyenne nulle indépendants Gaussiennes avec variance  $a_i$  :

$$c_n^i = \sqrt{\frac{a_i}{N_0}} \sum_{q=0}^{N_0-1} \cos(2\pi n f_D T \cos \gamma_{qi} + \phi_{qi}) + j \sin(2\pi n f_D T \sin \gamma_{qi} + \phi'_{qi}) \quad (39)$$

où

$$\gamma_{qi} = \frac{2\pi q}{4N_0} + \frac{2\pi i}{4N_0(M+1)} + \frac{\pi}{8N_0(M+1)}$$

$\phi_{qi}, \phi'_{qi}$ , pour  $q = 0, 1, 2, \dots, N_0 - 1$  et  $i = 0, 1, 2, \dots, M$ , sont  $2(M+1)N_0$  phases aléatoires indépendantes, chacun d'entre elles est répartie uniformément dans  $[0, 2\pi)$ , aussi nous considérons  $N_0 > 16$ .

Afin de saisir avec précision la dynamique du canal sans fil, nous formulons un modèle de canal approprié pour une utilisation dans le système de suivi de canal. Ce modèle doit être encore mathématiquement docile pour la mise en œuvre dans un contexte d'espace d'état en temps discret. Selon le processus de décoloration qui est modélisée comme un processus gaussien complexe, un modèle approprié est donc un modèle autorégressif (AR) . Des résultats de théorie d'informations ont

## Résumé étendu en Français

---

montré qu'un modèle AR premier ordre est suffisant pour représenter avec précision le comportement local du canal sans fil variant dans le temps. Un modèle d'ordre supérieur tout en fournissant des estimations plus précises de canal à long terme, exige nécessairement un ordre de AR de 100 – 200 coefficients, et est donc très intraitable pour le modèle de l'état. En prenant l'hypothèse de premier ordre, nous réalisons enfin l'évolution de l'état à l'instant  $n$  comme :

$$c_n^i = \xi c_{n-1}^i + v_n^i \quad i = 0, \dots, M \quad (40)$$

où  $\xi$  est le coefficient d'AR statique et  $v_n^i \sim \mathcal{N}_{\mathbb{C}}(0, \sigma_v^2)$  est le bruit de conduite complexe du modèle. Donc sous la forme de modèle d'état, nous avons :

$$\mathbf{c}_n = \mathbf{F} \mathbf{c}_{n-1} + \mathbf{v}_n \quad (41)$$

où  $\mathbf{c}_n$  est un vecteur de longueur  $M + 1$  dont chaque élément est le gain de canal au temps  $n$ .

$$\mathbf{c}_n = [c_n^0, c_n^1, \dots, c_n^M]^T \quad (42)$$

la matrice de transition d'état est donnée par :

$$\mathbf{F} = \xi \mathbf{I}_{M+1} \quad (43)$$

et le vecteur de bruit de processus proposé par :

$$\mathbf{v}_n = [v_n^0, v_n^1, \dots, v_n^M]^T \quad (44)$$

avec la matrice de covariance égal à :

$$\mathbf{Q} = (\sigma_v^2) \mathbf{I}_{M+1} \quad (45)$$

Afin de paramétrer (40), nous notons de [3] que l'auto-corrélation du processus de décoloration de canal est :

$$E[c_n^i c_{n-k}^{i*}] = a_i J_0(2\pi k f_D T) \quad (46)$$

où  $J_0()$  est la fonction de Bessel d'ordre zéro,  $T$  est la durée de symbole, et  $f_D$  désigne la fréquence Doppler résultant du déplacement relatif entre l'émetteur et le récepteur. Le décalage Doppler est donné par

$$f_D = \frac{v}{c} f_c \quad (47)$$

où  $v$  est la vitesse de mobile,  $c$  est la vitesse de la lumière, et  $fc$  est la fréquence de la porteuse. assimilant (40) à l'auto-corrélation de (46) pour l'instant  $n = \{0,1\}$ , Nous avons respectivement

$$\xi^2 a_i + \sigma_v^2 = a_i \quad (48)$$

$$\xi = J_0(2\pi f_D T) \quad (49)$$

Par exemple, si le taux d'évanouissement normalisé est  $f_D T = 0.01$  (un taux d'évanouissement rapide typique), ensuite  $\xi = 0.9990$ .

Si l'on considère le modèle autorégressif approximatif d'ordre un (AR(1)) introduit en [3](pp.74-75), nous avons :

$$\xi = 2 - \cos(2\pi f_D T) - \sqrt{(2 - \cos(2\pi f_D T))^2 - 1} \quad (50)$$

et

$$v_n^{(i)} \simeq \mathcal{N}_c(0, a_i(1 - \xi^2)) \quad (51)$$

L'observation bruitée complexe reçue à l'instant  $n$  est de la forme suivante :

$$y_n = \sum_{i=0}^M c_n^i b_{n-i} + w_n \quad (52)$$

où  $w_n$  est bruit additif modélisé comme une Gaussienne centrée avec variance  $\beta$ .  
Donc

$$y_n = \mathbf{d}_n^T \mathbf{c}_n + w_n \quad (53)$$

où

$$\mathbf{d}_n = [b_n, b_{n-1}, \dots, b_{n-M}]^T$$

Enfin, notre système de communication peut être décrit comme un modèle d'espace d'état linéaire, dont la dynamique est donnée par :

$$\begin{aligned} \mathbf{c}_n &= \mathbf{F} \mathbf{c}_{n-1} + \mathbf{v}_n \\ y_n &= \mathbf{d}_n^T \mathbf{c}_n + w_n \end{aligned} \quad (54)$$

où  $(\mathbf{d}_n)$  est indépendant de  $\mathbf{c}_n, w_n, \mathbf{v}_n$ , et  $\mathbf{v}_n, w_n$  sont des bruits blancs indépendants normaux de moyenne nulle, avec la matrice de covariance  $\mathbf{Q}$  et variance  $\beta$ . L'ensemble des valeurs de  $\mathbf{d}_n$  est  $\Omega$ .

## Méthode Proposée sur La Base des Algorithmes de Kalman et Viterbi Conjoints

Le but de cette section est de motiver rigoureusement l'introduction du filtre de Kalman dans l'estimation des séquences de Markov, à travers les canaux Gaussiens dynamiques. Par cela, nous interprétons et clarifions les approximations sous-jacentes dans les approches heuristiques de [45, 55, 95].

### Obtention de La Formule de Récurrence

Commençons par le problème de l'estimation de la probabilité a posteriori maximale d'un bloc de données fini  $\mathbf{d}_{1:n+1}$ . Notez que contrairement à de nombreuses contributions, nous n'avons pas tenté d'estimer la réalisation en cours du processus de canal. la fonction de vraisemblance d'une séquence de données  $\mathbf{d}_{1:n+1}$  se décompose comme suit :

$$\begin{aligned}
 L_{n+1}(\mathbf{d}_{1:n+1}) &= p(\mathbf{d}_{1:n+1}; y_{1:n+1}) \\
 &= p(y_{n+1} | \mathbf{d}_{1:n+1}, y_{1:n}) p(\mathbf{d}_{n+1} | \mathbf{d}_n) p(\mathbf{d}_{1:n}; y_{1:n}) \\
 &= f_{n+1}(\mathbf{d}_{1:n+1}) p(\mathbf{d}_{n+1} | \mathbf{d}_n) L_n(\mathbf{d}_{1:n})
 \end{aligned} \tag{55}$$

par conséquent, la séquence estimée est égale à :

$$\hat{\mathbf{d}}_{1:n+1} = \arg \max_{\mathbf{d}_{1:n+1}} L_{n+1}(\mathbf{d}_{1:n+1})$$

Introduisons maintenant le canal comme suit :

$$\begin{aligned}
 f_{n+1}(\mathbf{d}_{1:n+1}) &= \int p(y_{n+1}, \mathbf{c}_{n+1} = \mathbf{x} | \mathbf{d}_{1:n+1}; y_{1:n}) d\mathbf{x} \\
 &= \int p(y_{n+1} | \mathbf{c}_{n+1} = \mathbf{x}, \mathbf{d}_{n+1}) p(\mathbf{c}_{n+1} = \mathbf{x} | \mathbf{d}_{1:n+1}, y_{1:n}) d\mathbf{x}
 \end{aligned} \tag{56}$$

Les relations (55),(56) sont tout ce qui est nécessaire pour construire des maximiseurs approximatifs récursifs pour  $L_n$ .



## Normale de L'Assomption de La Densité

D'abord, par hypothèse, les termes de l'intégrale sont des normales, elle devrait donc être calculée explicitement tel que :

$$f_{n+1}(\mathbf{d}_{1:n+1}) = \int \mathcal{N}_c(y_{n+1}, \mathbf{d}_{n+1}\mathbf{x}, \beta) \mathcal{N}_c(\mathbf{x}, \hat{\mathbf{c}}_{n+1|n}(\mathbf{d}_{1:n}), \mathbf{P}_{n+1|n}(\mathbf{d}_{1:n})) d\mathbf{x} \quad (57)$$

où  $\mathcal{N}_c(u, \mu, R)$  est la densité normale complexe en  $u$  de moyenne  $\mu$  et de covariance  $R$ . et

$$\begin{aligned} \hat{\mathbf{c}}_{n+1|n}(\mathbf{d}_{1:n}) &= E[\mathbf{c}_{n+1} | \mathbf{d}_{1:n}, y_{1:n}] \\ \mathbf{P}_{n+1|n}(\mathbf{d}_{1:n}) &= cov[\mathbf{c}_{n+1} | \mathbf{d}_{1:n}, y_{1:n}] \end{aligned} \quad (58)$$

Après avoir groupé les exposants et les avoir factorisés, un calcul standard donne :

$$\begin{aligned} f_{n+1}(\mathbf{d}_{1:n+1}) &= [\pi\beta |\Gamma_{n+1} \mathbf{P}_{n+1|n}|]^{-1} exp[\mathbf{q}_{n+1}^\dagger \Gamma_{n+1}^{-1} \mathbf{q}_{n+1} - y_{n+1}^* \beta^{-1} y_{n+1} \\ &\quad - \hat{\mathbf{c}}_{n+1|n}^\dagger \mathbf{P}_{n+1|n}^{-1} \hat{\mathbf{c}}_{n+1|n}] \end{aligned} \quad (59)$$

où

$$\begin{aligned} \mathbf{q}_{n+1} &= \mathbf{d}_{n+1}^* \beta^{-1} y_{n+1} + \mathbf{P}_{n+1|n}^{-1}(\mathbf{d}_{1:n}) \hat{\mathbf{c}}_{n+1|n}(\mathbf{d}_{1:n}) \\ \Gamma_{n+1} &= \mathbf{P}_{n+1|n}^{-1}(\mathbf{d}_{1:n}) + \mathbf{d}_{n+1}^* \beta^{-1} \mathbf{d}_{n+1}^T \end{aligned} \quad (60)$$

## Maximisation Réursive Approximative

Il est évident à partir de (55) que l'obstruction d'exiger la maximisation réursive de  $L_{n+1}(\mathbf{d}_{1:n+1})$  est par le couplage apparemment total entre  $f_{n+1}(\mathbf{d}_{1:n+1})$  et  $L_n(\mathbf{d}_{1:n})$  c'est à dire le fait que ces termes part l'ensemble de la gamme de leurs variables, à partir de  $n$  à 1. En dernière analyse, ce couplage reflète les caractères non Gaussiens du modèle commun  $(\mathbf{c}_n, \mathbf{d}_n)$ . Pour construire la maximisation réursive approximative, il est souhaitable de lier ce couplage. Notez que les algorithmes heuristique dérivés en [96] peuvent être interprétés comme suit. Soient

$$\begin{aligned} \mathbf{d}_{1:n-1}^*(\mathbf{w}) &= \arg \max_{\mathbf{d}_{1:n-1}} L_n(\mathbf{d}_{1:n-1}, \mathbf{d}_n = \mathbf{w}) \\ \mathbf{c}_{n+1|n}^*(\mathbf{w}, \mathbf{d}_{n+1}) &= E[\mathbf{c}_{n+1} | \mathbf{d}_{1:n-1}^*(\mathbf{w}), \mathbf{d}_n = \mathbf{w}, \mathbf{d}_{n+1}, y_{1:n}] \quad (\mathbf{w} \in \Omega) \end{aligned}$$

par conséquent, nous pouvons écrire :

$$\mathbf{P}[\mathbf{c}_{n+1} | \mathbf{d}_{1:n-1}, \mathbf{d}_n = \mathbf{w}, \mathbf{d}_{n+1}, y_{1:n}] = \delta(\mathbf{c}_{n+1} - \mathbf{c}_{n+1|n}^*(\mathbf{w}, \mathbf{d}_{n+1}))$$

## L'Évaluation de La Dépendance

Pour générer des approximations d'une manière systématique, il est nécessaire d'évaluer la dépendance de  $f_{n+1}$  vs  $\mathbf{d}_{1:n+1}$ . Par (54), pour tout  $r \geq 1$  on a

$$\mathbf{c}_{n+1} = \mathbf{F}^r \mathbf{c}_{n-r+1} + \mathbf{v}_{n+1} \quad (61)$$

où  $\mathbf{v}_{n+1}$  est (statistiquement) indépendant de  $(\mathbf{d}_{1:n-r}, y_{1:n-r})$ .

Maintenant,  $f_{n+1}(\mathbf{d}_{1:n+1}) = p(y_{n+1} | \mathbf{d}_{1:n+1}, y_{1:n})$  en tant que densité normale est entièrement caractérisé par les premier et second moments. Par (54) et (61) nous obtenons ainsi :

$$\begin{aligned} E[y_{n+1} | \mathbf{d}_{1:n+1}, y_{1:n}] &= \mathbf{d}_{n+1}^T \mathbf{F}^r E[\mathbf{c}_{n-r+1} | \mathbf{d}_{1:n+1}, y_{1:n}] \\ &+ \mathbf{d}_{n+1}^T E[\mathbf{v}_{n+1} | \mathbf{d}_{n-r+1:n+1}, y_{n-r+1:n}] \end{aligned} \quad (62)$$

Par conséquent, la dépendance du côté gauche en  $\mathbf{d}_{1:n-r}$  n'est que par le premier terme du côté droit. Pour une large classe de modèles dynamiques stables les matrices  $\mathbf{F}$  ont des spectres strictement délimités par  $0 \leq \lambda_0 < 1$ . En supposant cela dans la suite, on voit que la dépendance du côté gauche en  $\mathbf{d}_{1:n-r}$  se désintègre au moins de façon exponentielle en  $r$  (à condition que la séquence  $E[\mathbf{c}_{n-r+1} | \mathbf{d}_{1:n+1}, y_{1:n}]$  reste limitée par une constante en moyenne quadratique, ce qui est le cas par les mêmes hypothèses sur  $\mathbf{F}$ ). Un calcul simple montre que la même cohésion est valable pour le second moment  $E[y_{n+1} y_{n+1}^* | \mathbf{d}_{1:n+1}, y_{1:n}]$ . En conclusion, on peut considérer que dans un sens, le couplage entre  $f_{n+1}(\mathbf{d}_{1:n+1})$  et  $L_n(\mathbf{d}_{1:n})$  dans la gamme  $\mathbf{d}_{1:n-r}$  diminue au moins de façon exponentielle en  $r$ . C'est ce qui rend possible l'introduction de l'estimateur de Kalman dans la maximisation récursive approximative de  $L_n(\mathbf{d}_{1:n})$ .

## Forme Définitive de La Maximisation Récursive Approximative

Considérant que la dépendance de  $f_{n+1}$  à  $\mathbf{d}_{1:n-r}$  est faible, pour  $\mathbf{d}_{n-r+1:n+1}$  fixé, le lieu géométrique des composants  $\mathbf{d}_{1:n-r}$  du maximum de  $L_{n+1}(\mathbf{d}_{1:n+1})$  n'est pas très affecté par  $f_{n+1}$ . Plus précisément, pour  $\mathbf{w}_{1:r} \in \Omega^r$  et tout  $n$  définir :

$$\bar{L}_n(\mathbf{w}_{1:r}) = \max_{\mathbf{d}_{1:n-r}} L_n(\mathbf{d}_{1:n-r}, \mathbf{d}_{n-r+1:n} = \mathbf{w}_{1:r}) \quad (63)$$

et

$$\mathbf{d}_{1:n-r}^*(\mathbf{w}_{1:r}) = \arg \max_{\mathbf{d}_{1:n-r}} L_n(\mathbf{d}_{1:n-r}, \mathbf{d}_{n-r+1} = \mathbf{w}_1, \dots, \mathbf{d}_n = \mathbf{w}_r) \quad (64)$$

puis

$$\begin{aligned}
 \max_{\mathbf{d}_{1:n+1}} L_{n+1}(d_{1:n+1}) &= \max_{\mathbf{w}_{2:r+1}} \bar{L}_{n+1}(\mathbf{w}_{2:r+1}) & (65) \\
 &= \max_{\mathbf{w}_{2:r+1}} \max_{\mathbf{w}_1} \max_{\mathbf{d}_{1:n-r}} [f_{n+1}(\mathbf{d}_{1:n-r}, \mathbf{d}_{n-r+1} = \mathbf{w}_1, \mathbf{d}_{n-r+2:n} = \mathbf{w}_{2:r+1}) p(\mathbf{w}_{r+1} | \mathbf{w}_r) \\
 &\quad L_n(\mathbf{d}_{1:n-r}, \mathbf{d}_{n-r+1:n} = \mathbf{w}_{1:r})]
 \end{aligned}$$

Par notre approximation :

$$f_{n+1}(\mathbf{d}_{1:n-r}, \mathbf{w}_{1:r+1}) \simeq f_{n+1}(\mathbf{d}_{1:n-r}^*(\mathbf{w}_{1:r}), \mathbf{w}_{1:r+1})$$

de sorte que nous avons mis la récurrence suivante :

$$\bar{L}_{n+1}(\mathbf{w}_{2:r+1}) \simeq \max_{\mathbf{w}_1} [f_{n+1}(\mathbf{d}_{1:n-r}^*(\mathbf{w}_{1:r}), \mathbf{w}_{1:r+1}) \bar{L}_n(\mathbf{w}_{1:r})] \quad (66)$$

Les définitions (63),(64) et les relations (65),(66) sont la solution de programmation dynamique à notre problème de maximisation récursive approximative. Avec (66) les mises à jour de récurrence sont effectuées par (58),(59) et (60) où  $\mathbf{d}_{1:n-r}$  sont forcées à  $\mathbf{d}_{1:n-r}^*(\mathbf{w}_{1:r})$  et les quantités  $\hat{\mathbf{c}}_{n+1|n}(\mathbf{d}_{1:n-r}^*(\mathbf{w}_{1:r}), \mathbf{w}_{1:r})$  et  $\mathbf{P}_{n+1|n}(\mathbf{d}_{1:n-r}^*(\mathbf{w}_{1:r}), \mathbf{w}_{1:r})$ ,  $\mathbf{w}_{1:r} \in \Omega^r$  quantités sont générées à chaque étape par  $|\Omega|^r$  l'équation de récurrence de Kalman [55]. L'algorithme détaillé pour  $r = 1$  est donné en annexe B.

## Conclusion

Ce chapitre a été consacré à notre méthode proposée de détection de données à travers un canal dynamique linéaire. Nous avons introduit un modèle espace d'état du canal linéaire dynamique et basés sur cette représentation, nous avons appliqué le filtre de Kalman comme le meilleur estimateur du canal. Contrairement à de nombreuses contributions, nous n'avons pas tenté d'estimer la réalisation en cours du processus de canal pour l'estimation du canal conjointement à la détection des données. Des simulations numériques ont montré que la méthode proposée était meilleure que PSP à pas fini et une taille de pas optimal et PSP via l'algorithme RLS avec un facteur d'oubli optimal.

## Chapitre 4 : Estimation des Données Discrètes Transmises sur des Canaux Non Linéaires Dynamiques Sans Fil basée sur La Technique d'EM

### Introduction

Dans le chapitre précédent, nous avons considéré un modèle espace d'état linéaire et l'avons analysé. Dans ce chapitre, nous examinons une approche plus générale c'est à dire le modèle espace d'état non linéaire. Le filtre de Kalman est une approche optimale lorsque les équations sont linéaires et les bruits sont indépendants, additifs et Gaussiens. Dans cette situation, les distributions d'intérêt sont également Gaussiennes et le filtre de Kalman peut les calculer exactement, sans approximations. Pour les scénarios où les modèles sont non linéaires ou le bruit est non Gaussien, deux approches principales dans la littérature sont prises en considération. La première approche consiste en le filtre de Kalman étendu et l'autre est le filtrage des particules.

En plus de ces deux approches, la procédure EM en tant que l'approche naturelle, est une procédure itérative efficace pour calculer l'estimation à maximum de vraisemblance (ML) en présence de données manquantes ou cachées. Sous paramètres de canal inconnus, il n'est pas possible de maximiser la fonction de vraisemblance pour obtenir directement le critère ML. Dans ce cas, l'algorithme EM qui permette d'atteindre le critère ML de manière itérative est idéalement adapté à ce problème. L'algorithme EM comporte deux étapes, l'espérance et la maximisation. La première étape prend l'espérance de la fonction de log-vraisemblance des données complètes compte tenu des paramètres actuels estimés et les données incomplètes. La seconde étape donne une nouvelle estimation des paramètres inconnus en maximisant l'espérance de la fonction de log-vraisemblance sur les paramètres inconnus. Ces étapes sont répétées de façon itérative pour augmenter la probabilité jusqu'à ce que les nouveaux paramètres estimés deviennent égaux à (ou arbitrairement proches de) la même valeur de paramètre estimée à l'itération précédente. L'algorithme EM couple l'estimation et la détection et utilise intrinsèquement le retour de décision ; ainsi EM peut être utilisé dans le problème d'estimation du canal conjointement à la détection

de données.

## Modèle de Système

Nous considérons la transmission de bloc sur une single-input single-output (SISO) switching doubly selective channel (SDSC) (longueur de bloc :  $N$ , la mémoire de canal :  $M$ , décalage de fréquence Doppler maximale à l'état de commutation  $m$  :  $f_D(m)$ , durée de symbole :  $T$ ) et binary phase shift keying (BPSK) modulation, de sorte que le bit est transmis à l'instant  $n$ ,  $b_n \in \{-1, +1\}$ .

Nous supposons un canal discret Rayleigh de la mémoire  $M$ , simulé avec méthode introduite dans [92], où les éléments de la réponse impulsionnelle  $\{c_n^i\}_{i=0}^M$  sont modélisés comme des variables aléatoires complexes de moyenne nulle indépendantes Gaussiennes avec variance  $a_i(m)$  :

$$c_n^i(m) = \sqrt{\frac{a_i(m)}{N_0}} \sum_{q=0}^{N_0-1} \cos(2\pi n f_D(m) T \cos \gamma_{qi} + \phi_{qi}) + j \sin(2\pi n f_D(m) T \sin \gamma_{qi} + \phi'_{qi}) \quad m \in \{1, 2\} \quad (67)$$

où

$$\gamma_{qi} = \frac{2\pi q}{4N_0} + \frac{2\pi i}{4N_0(M+1)} + \frac{\pi}{8N_0(M+1)}$$

$\phi_{qi}, \phi'_{qi}$ , pour  $q = 0, 1, 2, \dots, N_0 - 1$  et  $i = 0, 1, 2, \dots, M$ , sont  $2(M+1)N_0$  phases aléatoires indépendantes, chacune d'entre elle est répartie uniformément dans  $[0, 2\pi)$ , aussi nous considérons  $N_0 > 16$ .

Afin de saisir avec précision la dynamique du canal sans fil, nous formulons un modèle de canal approprié pour une utilisation dans le système de suivi de canal. Ce modèle doit être encore mathématiquement docile pour la mise en œuvre dans un contexte d'espace d'état en temps discret. Selon le processus de décoloration qui est modélisé comme un processus gaussien complexe, un modèle approprié est donc un modèle autorégressif (AR). Des résultats de théorie de l'informations ont montré qu'un modèle AR du premier ordre est suffisant pour représenter avec précision le comportement local du canal sans fil variant dans le temps. Un modèle d'ordre supérieur tout en fournissant des estimations plus précises de canal à long terme, exige nécessairement un ordre de AR de 100 – 200 coefficients, et est donc très

## Résumé étendu en Français

---

intraitable pour le modèle de l'état. En prenant l'hypothèse de premier ordre, nous réalisons enfin l'évolution de l'état à l'instant  $n$  comme :

$$c_n^i(m) = \xi(m)c_{n-1}^i(m) + v_n^i(m) \quad i = 0, \dots, M \quad m \in \{1, 2\} \quad (68)$$

où  $\xi(m)$  st le coefficient d'AR statique à l'état de commutation  $m$  et  $v_n^i(m) \sim \mathcal{N}_{\mathbb{C}}(0, \sigma_v^2(m))$  est le bruit de conduite complexe du modèle. Donc sous la forme de modèle d'état, nous avons :

$$\mathbf{c}_n^{(m)} = \mathbf{F}(m)\mathbf{c}_{n-1}^{(m)} + \mathbf{v}_n(m) \quad (69)$$

où  $\mathbf{c}_n^{(m)}$  est un vecteur de longueur  $M + 1$  dont chaque élément est le gain de canal au temps  $n$ .

$$\mathbf{c}_n^{(m)} = [c_n^0(m), c_n^1(m), \dots, c_n^M(m)]^T \quad (70)$$

la matrice de transition d'état est donnée par :

$$\mathbf{F}(m) = \xi(m)\mathbf{I}_{M+1} \quad (71)$$

et le vecteur de bruit de processus proposé par :

$$\mathbf{v}_n(m) = [v_n^0(m), v_n^1(m), \dots, v_n^M(m)]^T \quad (72)$$

avec la matrice de covariance égale à :

$$\mathbf{Q}(m) = (\sigma_v^2(m))\mathbf{I}_{M+1} \quad (73)$$

Afin de paramétrer (68), nous notons de [3] que l'auto-corrélation du processus de décoloration de canal est :

$$E[c_n^i(m)c_{n-k}^i(m)^*] = a_i(m)J_0(2\pi k f_D(m)T) \quad m \in \{1, 2\} \quad (74)$$

où  $J_0(\cdot)$  est la fonction de Bessel d'ordre zéro,  $T$  est la durée de symbole, et  $f_D(m)$  désigne la fréquence Doppler à l'état de commutation  $m$  résultant du déplacement relatif entre l'émetteur et le récepteur. Le décalage Doppler est donné par

$$f_D(m) = \frac{v(m)}{c} f_c \quad (75)$$

où  $v(m)$  est la vitesse de mobile, à l'état de commutation  $m$ ,  $c$  est la vitesse de la lumière, et  $fc$  est la fréquence de la porteuse. Assimilant (68) à l'auto-corrélation de (74) pour l'instant  $n = \{0,1\}$ , nous avons respectivement

$$\xi(m)^2 a_i(m) + \sigma_v(m)^2 = a_i(m) \quad (76)$$

$$\xi(m) = J_0(2\pi f_D(m)T) \quad (77)$$

Par exemple, si le taux d'évanouissement normalisé est  $f_D T = 0.01$  (un taux d'évanouissement rapide typique), alors  $\xi = 0.9990$ .

L'observation bruitée complexe reçue à l'instant  $n$  est de la forme suivante :

$$y_n = \sum_{i=0}^M c_n^i(\alpha_n = m) b_{n-i} + w_n \quad (78)$$

où  $w_n$  est bruit additif modélisé comme une Gaussienne centrée avec variance  $\beta$ , aussi  $\alpha_n$  est la variable de commutation et  $m$  est l'état de commutation  $m \in \{1, 2\}$ .

Donc

$$y_n = \mathbf{d}_n^T \mathbf{c}_n^{(\alpha_n=m)} + w_n \quad (79)$$

où

$$\mathbf{d}_n = [b_n, b_{n-1}, \dots, b_{n-M}]^T$$

Enfin, notre système de communication peut être décrit comme un modèle d'espace d'état à commutation, dont la dynamique est donnée par :

$$\begin{aligned} \mathbf{c}_n^{(1)} &= \mathbf{F}(1) \mathbf{c}_{n-1}^{(1)} + \mathbf{v}_n(1) \\ \mathbf{c}_n^{(2)} &= \mathbf{F}(2) \mathbf{c}_{n-1}^{(2)} + \mathbf{v}_n(2) \\ y_n &= \mathbf{d}_n^T \mathbf{c}_n^{(\alpha_n=m)} + w_n \quad m \in \{1, 2\} \end{aligned} \quad (80)$$

où l'état de commutation  $m$  a été choisi à l'aide des aprioris  $\pi^1 = \pi^2 = 1/2$  et des probabilités de transition  $\Phi_{11} = \Phi_{22} = 0.95$ ; et  $\Phi_{12} = \Phi_{21} = 0.05$ . La représentation de modèle graphique pour la commutation modèles espace-état est indiquée en Figure 1.

## Approche Proposée pour L'Estimation des Données basée sur EM

On considère le problème de l'estimation du maximum de vraisemblance de la séquence  $\theta_N = (\mathbf{d}_{0:N}, \alpha_{0:N})$  donnée  $y_{0:N}$ . Afin d'estimer  $\theta_N$ , il est typique d'introduire

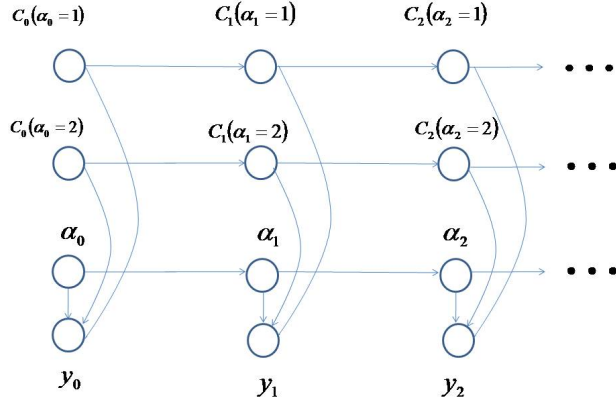


FIGURE 1 – Représentation du modèle graphique pour la commutation des modèles état-espace (SSSM).  $\alpha_n$  est la variable de commutation discrète et  $\mathbf{c}_n(1)$ ,  $\mathbf{c}_n(2)$  sont les vecteurs d'état de valeurs complexes.

la fonction de log-vraisemblance définie comme :

$$L(\theta_N) = \ln p(y_{0:N}|\theta_N) \quad (81)$$

comme  $\ln(x)$  est une fonction strictement croissante, la valeur de  $\theta_N$  qui maximise  $p(y_{0:N}|\theta_N)$  maximise également  $L(\theta_N)$ . La normalité conditionnelle du modèle suggère la procédure d'EM comme l'approche naturelle, au moins de façon algorithmique. De cette manière, EKF et les filtres à particules sont évités. L'algorithme EM est une procédure itérative pour maximiser  $L(\theta_N)$ . On suppose que l'estimation actuelle de  $\theta_N$  est donnée par  $\theta'_N = (\mathbf{d}'_{0:N}, \alpha'_{0:N})$ . Puisque l'objectif est de maximiser  $L(\theta_N)$ , nous souhaitons calculer une estimation actualisée  $\theta_N$  telle que,

$$L(\theta_N) > L(\theta'_N) \quad (82)$$

De manière équivalente, nous voulons maximiser la différence

$$L(\theta_N) - L(\theta'_N) = \ln p(y_{0:N}|\theta_N) - \ln p(y_{0:N}|\theta'_N) \quad (83)$$

Jusqu'à présent, nous n'avons pas tenu compte des variables non observées ou manquantes. Dans les problèmes où de telles données existent, l'algorithme EM fournit un cadre naturel pour leur inclusion. Alternativement, les variables cachées peuvent



être introduites uniquement comme un artifice pour effectuer l'estimation du maximum de vraisemblance  $\theta_N$  docile. Dans ce cas, on suppose que la connaissance des variables cachées rendra la maximisation de la fonction de vraisemblance plus facile. Les paramètres  $(\mathbf{d}_{0:N}, \alpha_{0:N}, \mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)})$  forment un ensemble complet à condition que la famille de densité  $p(y_{0:N}, \mathbf{d}_{0:N}, \alpha_{0:N}, \mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)})$  soit exponentielle. La partie continue  $\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)}$  sera traitée comme une variable cachée. La probabilité totale  $p(y_{0:N}|\theta_N)$  peut être rédigée en terme des variables cachées  $\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)}$

$$p(y_{0:N}|\theta_N) = \int p(y_{0:N}|\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)}, \theta_N) p(\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)}|\theta_N) d\mathbf{c}_{0:N}^{(1)} d\mathbf{c}_{0:N}^{(2)}$$

Nous pouvons alors réécrire l'équation (83) comme,

$$\begin{aligned} L(\theta_N) - L(\theta'_N) &= \ln \int p(y_{0:N}|\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)}, \theta_N) \\ &\quad \times p(\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)}|\theta_N) d\mathbf{c}_{0:N}^{(1)} d\mathbf{c}_{0:N}^{(2)} - \ln p(y_{0:N}|\theta'_N) \end{aligned}$$

Si nous introduisons les constantes  $p(\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)}|y_{0:N}, \theta'_N)$  dans l'équation, nous avons :

$$\begin{aligned} L(\theta_N) - L(\theta'_N) &= \ln \int p(y_{0:N}|\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)}, \theta_N) \\ &\quad \times p(\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)}|\theta_N) \frac{p(\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)}|y_{0:N}, \theta'_N)}{p(\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)}|y_{0:N}, \theta'_N)} d\mathbf{c}_{0:N}^{(1)} d\mathbf{c}_{0:N}^{(2)} - \ln p(y_{0:N}|\theta'_N) \end{aligned}$$

En utilisant l'inégalité de Jensen comme d'habitude la limite inférieure, la log-vraisemblance de ces paramètres est comme suit :

$$\begin{aligned} L(\theta_N) - L(\theta'_N) &\geq \int p(\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)}|y_{0:N}, \theta'_N) \\ &\quad \ln \frac{p(y_{0:N}|\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)}, \theta_N) p(\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)}|\theta_N)}{p(\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)}|y_{0:N}, \theta'_N) p(y_{0:N}|\theta'_N)} d\mathbf{c}_{0:N}^{(1)} d\mathbf{c}_{0:N}^{(2)} = \Delta(\theta_N|\theta'_N) \end{aligned} \quad (84)$$

Nous écrivons ensuite :

$$L(\theta_N) \geq L(\theta'_N) + \Delta(\theta_N|\theta'_N) \quad (85)$$

et pour plus de commodité définissons

$$l(\theta_N|\theta'_N) = L(\theta'_N) + \Delta(\theta_N|\theta'_N) \quad (86)$$

de sorte que

$$L(\theta_N) \geq l(\theta_N|\theta'_N) \quad (87)$$

Notre objectif est de choisir une valeur de  $\theta_N$  afin que  $L(\theta_N)$  soit maximisée. Afin d'atteindre la plus grande augmentation possible de la valeur de  $L(\theta_N)$ , l'algorithme EM est appliqué pour sélectionner  $\theta_N$  telle que  $l(\theta_N|\theta'_N)$  soit maximisée. On note cette valeur mise à jour comme  $\theta''_N$  formellement, nous avons :

$$\theta''_N = \arg \max_{\theta_N} \{l(\theta_N|\theta'_N)\} \quad (88)$$

Si nous laissons tomber les termes qui sont constants par rapport à  $\theta_N$  nous avons

$$\begin{aligned} \theta''_N &= \arg \max_{\theta_N} \left\{ \int p(\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)} | y_{0:N}, \theta'_N) \ln p(y_{0:N}, \mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)} | \theta_N) d\mathbf{c}_{0:N}^{(1)} d\mathbf{c}_{0:N}^{(2)} \right\} \\ &= \arg \max_{\theta_N} \{E_{c^{(1)}, c^{(2)}|y, \theta'} \{\ln p(y_{0:N}, \mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)} | \theta_N)\}\} \end{aligned} \quad (89)$$

Dans l'équation (89) les étapes d'espérance et maximisation sont évidentes. L'algorithme EM consiste donc à itérer les :

1. Étape-E : Déterminer l'espérance conditionnelle  $E_{c^{(1)}, c^{(2)}|y, \theta'} \{\ln p(y_{0:N}, \mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)} | \theta_N)\}$
2. Étape-M : Maximiser cette expression par rapport à  $\theta_N$

### Étape d'Espérance

Afin de déterminer l'espérance conditionnelle  $E_{c^{(1)}, c^{(2)}|y, \theta'} \{\ln p(y_{0:N}, \mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)} | \theta_N)\}$ , nous décomposons d'abord l'intégrale (à l'aide d'indépendance et caractères de Markov) comme :

$$\begin{aligned} E_{c^{(1)}, c^{(2)}|y, \theta'} \{\ln p(y_{0:N}, \mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)} | \theta_N)\} &= \int p(\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)} | y_{0:N}, \theta'_N) \ln p(y_{0:N}, \mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)} | \theta_N) \\ &\quad d\mathbf{c}_{0:N}^{(1)} d\mathbf{c}_{0:N}^{(2)} = \int p(\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)} | y_{0:N}, \theta'_N) \ln p(y_{0:N} | \mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)}, \theta_N) d\mathbf{c}_{0:N}^{(1)} d\mathbf{c}_{0:N}^{(2)} \\ &\quad + \int p(\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)} | y_{0:N}, \theta'_N) \ln p(\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)} | \theta_N) d\mathbf{c}_{0:N}^{(1)} d\mathbf{c}_{0:N}^{(2)} \\ &= E_{c^{(1)}, c^{(2)}|y, \theta'} [\ln p(y_{0:N} | \mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)}, \theta_N)] + E_{c^{(1)}, c^{(2)}|y, \theta'} [\ln p(\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)} | \theta_N)] \end{aligned} \quad (90)$$

Si l'on considère

$$\begin{aligned} J(n, \mathbf{d}_n, \alpha_n) &= E_n[\ln p(y_n | \mathbf{c}_n^{(m)}, \mathbf{d}_n, \alpha_n = m)] \\ H_1(n) &= E_n[\ln p(\mathbf{c}_n^{(1)} | \mathbf{c}_{n-1}^{(1)})] \\ H_2(n) &= E_n[\ln p(\mathbf{c}_n^{(2)} | \mathbf{c}_{n-1}^{(2)})] \end{aligned} \quad (91)$$

où  $E_n[\cdot]$ , dénote l'espérance par rapport à la densité

$$p_n(\mathbf{t}', \mathbf{t}, \mathbf{x}', \mathbf{x}) = p(\mathbf{c}_{n-1}^{(1)} = \mathbf{t}', \mathbf{c}_n^{(1)} = \mathbf{t}, \mathbf{c}_{n-1}^{(2)} = \mathbf{x}', \mathbf{c}_n^{(2)} = \mathbf{x} | y_{0:N}, \theta'_N) \quad n = 1 : N \quad (92)$$

et

$$p_n(\mathbf{t}, \mathbf{x}) = p(\mathbf{c}_n^{(1)} = \mathbf{t}, \mathbf{c}_n^{(2)} = \mathbf{x} | y_{0:N}, \theta'_N) \quad n = 0$$

alors nous avons :

$$\begin{aligned} E_{c^{(1)}, c^{(2)} | y, \theta'} \{ \ln p(y_{0:N}, \mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)} | \theta_N) \} &= \sum_{n=0}^N J(n, \mathbf{d}_n, \alpha_n) + \sum_{n=1}^N H_1(n) + \sum_{n=1}^N H_2(n) \\ &+ E_0[\ln p(\mathbf{c}_0^{(1)})] + E_0[\ln p(\mathbf{c}_0^{(2)})] \quad (93) \end{aligned}$$

La partie la plus laborieuse est bien sûr la génération de conditionnelle marginale. Ainsi, grâce au caractère Markovien et au caractère normal conditionnel, d'abord, nous calculons les densités  $p_n(\mathbf{t}', \mathbf{t}, \mathbf{x}', \mathbf{x})$  en utilisant une variante de la procédure générale avant-arrière qui est très efficace. Ensuite, nous calculons les quantités attendues  $J(n, \mathbf{d}_n, \alpha_n)$ ,  $H_1(n)$ ,  $H_2(n)$ ,  $E_0[\ln p(\mathbf{c}_0^{(1)})]$  et  $E_0[\ln p(\mathbf{c}_0^{(2)})]$ .

### Calcul des Densités $p_n(\mathbf{t}', \mathbf{t}, \mathbf{x}', \mathbf{x})$

$$\begin{aligned} p_n(\mathbf{t}', \mathbf{t}, \mathbf{x}', \mathbf{x}) &= p(\mathbf{c}_{n-1}^{(1)} = \mathbf{t}', \mathbf{c}_n^{(1)} = \mathbf{t}, \mathbf{c}_{n-1}^{(2)} = \mathbf{x}', \mathbf{c}_n^{(2)} = \mathbf{x} | y_{0:N}, \theta'_N) \\ &= \int \frac{p(\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)}, y_{0:N} | \theta'_N)}{p(y_{0:N} | \theta'_N)} d\mathbf{c}_{0:n-2}^{(1)} d\mathbf{c}_{0:n-2}^{(2)} d\mathbf{c}_{n+1:N}^{(1)} d\mathbf{c}_{n+1:N}^{(2)} \quad (94) \end{aligned}$$

par division

$$\begin{aligned} p(\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)}, y_{0:N} | \theta'_N) &= p(y_{0:N} | \mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)}, \theta'_N) p(\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)} | \theta'_N) \\ &= p(y_{0:n} | \mathbf{c}_{0:n}^{(1)}, \mathbf{c}_{0:n}^{(2)}, \theta'_N) p(\mathbf{c}_{0:n}^{(1)}, \mathbf{c}_{0:n}^{(2)} | \theta'_N) p(y_{n+1:N} | \mathbf{c}_{n+1:N}^{(1)}, \mathbf{c}_{n+1:N}^{(2)}, \theta'_N) \\ &\quad \times p(\mathbf{c}_{n+1:N}^{(1)}, \mathbf{c}_{n+1:N}^{(2)} | \mathbf{c}_n^{(1)}, \mathbf{c}_n^{(2)}, \theta'_N) \quad (95) \end{aligned}$$

où l'on a utilisé l'indépendance du bruit et de la propriété de Markov pour  $\mathbf{c}_{0:N}$ , on constate que :

$$p_n(\mathbf{t}', \mathbf{t}, \mathbf{x}', \mathbf{x}) = \frac{\Lambda_n^{(a)}(\mathbf{t}', \mathbf{t}, \mathbf{x}', \mathbf{x}) \Lambda_n^{(b)}(\mathbf{t}, \mathbf{x})}{p(y_{0:N} | \theta'_N)} \quad (96)$$

## Résumé étendu en Français

---

avec :

$$\Lambda_n^{(a)}(\mathbf{t}', \mathbf{t}, \mathbf{x}', \mathbf{x}) = \int_{C^{n-2}} p(y_{0:n} | \mathbf{c}_{0:n-2}^{(1)}, \mathbf{c}_{0:n-2}^{(2)}, \mathbf{c}_{n-1}^{(1)} = \mathbf{t}', \mathbf{c}_n^{(1)} = \mathbf{t}, \mathbf{c}_{n-1}^{(2)} = \mathbf{x}', \mathbf{c}_n^{(2)} = \mathbf{x}, \theta'_N) \\ p(\mathbf{c}_{0:n-2}^{(1)}, \mathbf{c}_{0:n-2}^{(2)}, \mathbf{c}_{n-1}^{(1)} = \mathbf{t}', \mathbf{c}_n^{(1)} = \mathbf{t}, \mathbf{c}_{n-1}^{(2)} = \mathbf{x}', \mathbf{c}_n^{(2)} = \mathbf{x} | \theta'_N) d\mathbf{c}_{0:n-2}^{(1)} d\mathbf{c}_{0:n-2}^{(2)}$$

$$\Lambda_n^{(b)}(\mathbf{t}, \mathbf{x}) = \int_{C^{N-n}} p(y_{n+1:N} | \mathbf{c}_{n+1:N}^{(1)}, \mathbf{c}_{n+1:N}^{(2)}, \theta'_N) p(\mathbf{c}_{n+1:N}^{(1)} | \mathbf{c}_n^{(1)} = \mathbf{t}, \theta'_N) \\ p(\mathbf{c}_{n+1:N}^{(2)} | \mathbf{c}_n^{(2)} = \mathbf{x}, \theta'_N) d\mathbf{c}_{n+1:N}^{(1)} d\mathbf{c}_{n+1:N}^{(2)}$$

puis avec le calcul de la récursivité des  $\Lambda_n^{(a)}(\mathbf{t}', \mathbf{t}, \mathbf{x}', \mathbf{x})$  et  $\Lambda_n^{(b)}(\mathbf{t}, \mathbf{x})$  et considérant le fait que toutes les densités figurant dans les relations précédentes sont normales, afin qu'ils réduisent facilement à avant-arrière récurrence sur premier et deuxième moments conditionnels, qui sont calculées en Annexe E. En utilisant les résultats qui sont présentés en Annexe E, nous avons :

pour  $n = 1 : N$

$$p_n(\mathbf{t}', \mathbf{t}, \mathbf{x}', \mathbf{x}) = \frac{\gamma_n^{(a)} \gamma_n^{(b)}}{p(y_{0:N} | \theta'_N)} \exp[-\mathbf{z}^\dagger \Gamma_n^{(zb)} \mathbf{z} + 2\text{Re}(\mathbf{z}^\dagger \varphi_n^{(zb)}) \\ - \mathbf{1}^\dagger \Gamma_n^{(lb)} \mathbf{1} + 2\text{Re}(\mathbf{1}^\dagger \varphi_n^{(lb)})] \quad (97)$$

où

$$\mathbf{z}^\dagger = [\mathbf{t}'^\dagger, \mathbf{t}^\dagger] \\ \mathbf{1}^\dagger = [\mathbf{x}'^\dagger, \mathbf{x}^\dagger] \\ \varphi_n^{(zb)\dagger} = [\varphi_{n,1}^{(z)\dagger}, \varphi_{n,2}^{(z)\dagger} + \varphi_{n,1}^{(b)\dagger}] \\ \varphi_n^{(lb)\dagger} = [\varphi_{n,1}^{(l)\dagger}, \varphi_{n,2}^{(l)\dagger} + \varphi_{n,2}^{(b)\dagger}] \\ \Gamma_n^{(zb)} = \begin{pmatrix} \Gamma_n^{(z)}(11) & \Gamma_n^{(z)}(12) \\ \Gamma_n^{(z)\dagger}(12) & \Gamma_n^{(z)}(22) + \Gamma_{n,1}^{(b)} \end{pmatrix} \\ \Gamma_n^{(lb)} = \begin{pmatrix} \Gamma_n^{(l)}(11) & \Gamma_n^{(l)}(12) \\ \Gamma_n^{(l)\dagger}(12) & \Gamma_n^{(l)}(22) + \Gamma_{n,2}^{(b)} \end{pmatrix}$$

pour  $n = 0$

$$p_n(\mathbf{t}, \mathbf{x}) = \frac{\gamma_n^{(a)} \gamma_n^{(b)}}{p(y_{1:N} | \theta'_N)} \exp[-\mathbf{t}^\dagger (\Gamma_n^{(t)} + \Gamma_{n,1}^{(b)}) \mathbf{t} + 2\text{Re}(\mathbf{t}^\dagger (\varphi_n^{(t)} + \varphi_{n,1}^{(b)})) - \mathbf{x}^\dagger (\Gamma_n^{(x)} + \Gamma_{n,2}^{(b)}) \mathbf{x} \\ + 2\text{Re}(\mathbf{x}^\dagger (\varphi_n^{(x)} + \varphi_{n,2}^{(b)}))] ]$$

Les valeurs d'éléments de vecteurs  $\varphi_n^{(zb)}$ ,  $\varphi_n^{(lb)}$ ,  $\varphi_n^{(x)}$  et les éléments des matrices  $\mathbf{\Gamma}_n^{(zb)}$ ,  $\mathbf{\Gamma}_n^{(lb)}$ ,  $\mathbf{\Gamma}_n^{(x)}$  figurent en Annexe E.

**Calculer Les Quantités Attendues**  $J(n, \mathbf{d}_n, \alpha_n)$ ,  $H_1(n)$ ,  $H_2(n)$ ,  $E_0[\ln p(\mathbf{c}_0^{(1)})]$  et  $E_0[\ln p(\mathbf{c}_0^{(2)})]$

Nous pouvons maintenant exprimer les quantités figurant dans  $E_{c^{(1)}, c^{(2)}|y, \theta'} \{ \ln p(y_{0:N}, \mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)} | \theta_N) \}$  en fonction de ces moments.

Nous avons pris :

$$\begin{aligned} \mathbf{z}_n^\dagger &= [\mathbf{c}_{n-1}^{(1)\dagger}, \mathbf{c}_n^{(1)\dagger}] \\ \mathbf{l}_n^\dagger &= [\mathbf{c}_{n-1}^{(2)\dagger}, \mathbf{c}_n^{(2)\dagger}] \\ \mathbf{V} &= [\mathbf{0}_{(M+1) \times (M+1)}, \mathbf{I}_{M+1}] \\ \psi^{(1)} &= [-\mathbf{F}(1), \mathbf{I}_{M+1}] \\ \psi^{(2)} &= [-\mathbf{F}(2), \mathbf{I}_{M+1}] \end{aligned}$$

à partir des équations de rappel qui ont été montrées en Annexe E, pour  $1 \leq n \leq N$ , nous avons :

$$\begin{aligned} E_n[\mathbf{z}_n] &= \mathbf{\Gamma}_n^{(zb)-1} \varphi_n^{(zb)} \\ E_n[\mathbf{z}_n \mathbf{z}_n^\dagger] &= \mathbf{\Gamma}_n^{(zb)-1} + \mathbf{\Gamma}_n^{(zb)-1} \varphi_n^{(zb)} \varphi_n^{(zb)\dagger} \mathbf{\Gamma}_n^{(zb)-1} \\ E_n[\mathbf{l}_n] &= \mathbf{\Gamma}_n^{(lb)-1} \varphi_n^{(lb)} \\ E_n[\mathbf{l}_n \mathbf{l}_n^\dagger] &= \mathbf{\Gamma}_n^{(lb)-1} + \mathbf{\Gamma}_n^{(lb)-1} \varphi_n^{(lb)} \varphi_n^{(lb)\dagger} \mathbf{\Gamma}_n^{(lb)-1} \end{aligned}$$

et

$$\begin{aligned} E_n[\mathbf{c}_n^{(1)}] &= \mathbf{V} E_n[\mathbf{z}_n] \\ E_n[\mathbf{c}_n^{(1)} \mathbf{c}_n^{(1)\dagger}] &= \mathbf{V} E_n[\mathbf{z}_n \mathbf{z}_n^\dagger] \mathbf{V}^T \\ E_n[\mathbf{c}_n^{(2)}] &= \mathbf{V} E_n[\mathbf{l}_n] \\ E_n[\mathbf{c}_n^{(2)} \mathbf{c}_n^{(2)\dagger}] &= \mathbf{V} E_n[\mathbf{l}_n \mathbf{l}_n^\dagger] \mathbf{V}^T \end{aligned}$$

## Résumé étendu en Français

---

quand  $n = 0$ , nous avons :

$$\begin{aligned}
 E_n[\mathbf{c}_0^{(1)}] &= (\Gamma_n^{(t)} + \Gamma_{n,1}^{(b)})^{-1}(\varphi_0^{(t)} + \varphi_{0,1}^{(b)}) \\
 E_n[\mathbf{c}_0^{(1)} \mathbf{c}_0^{(1)\dagger}] &= (\Gamma_n^{(t)} + \Gamma_{n,1}^{(b)})^{-1} + (\Gamma_n^{(t)} + \Gamma_{n,1}^{(b)})^{-1}(\varphi_0^{(t)} + \varphi_{0,1}^{(b)})(\varphi_0^{(t)} + \varphi_{0,1}^{(b)})^\dagger(\Gamma_n^{(t)} + \Gamma_{n,1}^{(b)})^{-1} \\
 E_n[\mathbf{c}_0^{(2)}] &= (\Gamma_n^{(x)} + \Gamma_{n,2}^{(b)})^{-1}(\varphi_0^{(x)} + \varphi_{0,2}^{(b)}) \\
 E_n[\mathbf{c}_0^{(2)} \mathbf{c}_0^{(2)\dagger}] &= (\Gamma_n^{(x)} + \Gamma_{n,2}^{(b)})^{-1} + (\Gamma_n^{(x)} + \Gamma_{n,2}^{(b)})^{-1}(\varphi_0^{(x)} + \varphi_{0,2}^{(b)})(\varphi_0^{(x)} + \varphi_{0,2}^{(b)})^\dagger(\Gamma_n^{(x)} + \Gamma_{n,2}^{(b)})^{-1}
 \end{aligned}$$

Par définition :

$$\begin{aligned}
 J(n, \mathbf{d}_n, \alpha_n) &= E_n[\ln p(y_n | \mathbf{c}_n^{(\alpha_n)}, \mathbf{d}_n)] \\
 &= -\ln[\pi\beta] - E_n[\|y_n - \mathbf{d}_n^T \mathbf{c}_n^{(\alpha_n)}\|_{\beta^{-1}}^2] \\
 &= -\ln[\pi\beta] - \|y_n\|^2/\beta + 2Re[y_n^* \beta^{-1} \mathbf{d}_n^T E_n[\mathbf{c}_n^{(\alpha_n)}]] - tr \mathbf{d}_n^* \beta^{-1} \mathbf{d}_n^T E_n[\mathbf{c}_n^{(\alpha_n)} \mathbf{c}_n^{(\alpha_n)\dagger}]
 \end{aligned}$$

de manière similaire :

$$\begin{aligned}
 H_1(n) &= E_n[\ln p(\mathbf{c}_n^{(1)} | \mathbf{c}_{n-1}^{(1)})] = -\ln[\pi^{M+1} |\mathbf{Q}(1)|] - E_n[\|\mathbf{c}_n^{(1)} - \mathbf{F}(1) \mathbf{c}_{n-1}^{(1)}\|_{\mathbf{Q}^{-1}(1)}^2] \\
 &= -\ln[\pi^{M+1} |\mathbf{Q}(1)|] - tr \psi^\dagger(1) \mathbf{Q}^{-1}(1) \psi(1) E_n[\mathbf{z}_n \mathbf{z}_n^\dagger]
 \end{aligned}$$

$$\begin{aligned}
 H_2(n) &= E_n[\log p(\mathbf{c}_n^{(2)} | \mathbf{c}_{n-1}^{(2)})] = -\ln[\pi^{M+1} |\mathbf{Q}(2)|] - E_n[\|\mathbf{c}_n^{(2)} - \mathbf{F}(2) \mathbf{c}_{n-1}^{(2)}\|_{\mathbf{Q}^{-1}(2)}^2] \\
 &= -\ln[\pi^{M+1} |\mathbf{Q}(2)|] - tr \psi^\dagger(2) \mathbf{Q}^{-1}(2) \psi(2) E_n[\mathbf{l}_n \mathbf{l}_n^\dagger]
 \end{aligned}$$

$$\begin{aligned}
 E_0[\ln p(\mathbf{c}_0^{(1)})] &= -\ln[\pi^{M+1} |\mathbf{P}_0^{(1)}|] - E_0[\|\mathbf{c}_0^{(1)} - \hat{\mathbf{c}}_0^{(1)}\|_{\mathbf{P}_0^{(1)-1}}^2] \\
 &= -\ln[\pi^{M+1} |\mathbf{P}_0^{(1)}|] + 2Re[\hat{\mathbf{c}}_0^{(1)\dagger} \mathbf{P}_0^{(1)-1} E_0[\mathbf{c}_0^{(1)}]] - tr \mathbf{P}_0^{(1)-1} E_0[\mathbf{c}_0^{(1)} \mathbf{c}_0^{(1)\dagger}]
 \end{aligned}$$

$$\begin{aligned}
 E_0[\ln p(\mathbf{c}_0^{(2)})] &= -\ln[\pi^{M+1} |\mathbf{P}_0^{(2)}|] - E_0[\|\mathbf{c}_0^{(2)} - \hat{\mathbf{c}}_0^{(2)}\|_{\mathbf{P}_0^{(2)-1}}^2] \\
 &= -\ln[\pi^{M+1} |\mathbf{P}_0^{(2)}|] + 2Re[\hat{\mathbf{c}}_0^{(2)\dagger} \mathbf{P}_0^{(2)-1} E_0[\mathbf{c}_0^{(2)}]] - tr \mathbf{P}_0^{(2)-1} E_0[\mathbf{c}_0^{(2)} \mathbf{c}_0^{(2)\dagger}]
 \end{aligned}$$

### Détermination de L'Espérance Conditionnelle Finale

En utilisant les résultats de calcul de la section précédente, nous sommes en mesure de déterminer l'espérance conditionnelle  $E_{c^{(1)}, c^{(2)} | y, \theta'} \{\ln p(y_{0:N}, \mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)} | \theta_N)\}$

comme suit :

$$\begin{aligned}
E_{c^{(1)},c^{(2)}|y,\theta'}\{\ln p(y_{0:N}, \mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)}|\theta_N)\} &= \sum_{n=0}^N J(n, \mathbf{d}_n, \alpha_n) + \sum_{n=1}^N H_1(n) + \sum_{n=1}^N H_2(n) \\
+ E_n[\ln p(\mathbf{c}_0^{(1)})] + E_n[\ln p(\mathbf{c}_0^{(2)})] &= \sum_{n=0}^N (-\ln[\pi\beta] - \|y_n\|^2/\beta + 2Re[y_n^*\beta^{-1}\mathbf{d}_n^T E_n[\mathbf{c}_n^{(\alpha_n)}]]) \\
- tr\mathbf{d}_n^*\beta^{-1}\mathbf{d}_n^T E_n[\mathbf{c}_n^{(\alpha_n)}\mathbf{c}_n^{(\alpha_n)\dagger}] &+ \sum_{n=1}^N -\ln[\pi^{M+1}|\mathbf{Q}(1)|] - tr\psi^\dagger(1)\mathbf{Q}^{-1}(1)\psi(1)E_n[\mathbf{z}_n\mathbf{z}_n^\dagger] \\
+ \sum_{n=1}^N -\ln[\pi^{M+1}|\mathbf{Q}(2)|] - tr\psi^\dagger(2)\mathbf{Q}^{-1}(2)\psi(2)E_n[\mathbf{1}_n\mathbf{1}_n^\dagger] &- \ln[\pi^{M+1}|\mathbf{P}_0^{(1)}|] + 2Re[\hat{\mathbf{c}}_0^{(1)\dagger}\mathbf{P}_0^{(1)-1}E_0[\mathbf{c}_0^{(1)}]] \\
- tr\mathbf{P}_0^{(1)-1}E_0[\mathbf{c}_0^{(1)}\mathbf{c}_0^{(1)\dagger}] - \ln[\pi^{M+1}|\mathbf{P}_0^{(2)}|] &+ 2Re[\hat{\mathbf{c}}_0^{(2)\dagger}\mathbf{P}_0^{(2)-1}E_0[\mathbf{c}_0^{(2)}]] - tr\mathbf{P}_0^{(2)-1}E_0[\mathbf{c}_0^{(2)}\mathbf{c}_0^{(2)\dagger}]
\end{aligned} \tag{98}$$

### Étape de Maximisation

Afin de maximiser l'espérance en (98) par rapport à  $\theta_N$ , nous laissons tomber les termes qui sont constants à l'égard de  $\theta_N$  et maintenons les autres termes.

$$E_{c^{(1)},c^{(2)}|y,\theta'}\{\ln p(y_{0:N}, \mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)}|\theta_N)\} = I_N(\theta_N|\theta'_N) + const. \tag{99}$$

où

$$I_N(\theta_N|\theta'_N) = \sum_{n=0}^N (2Re[y_n^*\beta^{-1}\mathbf{d}_n^T E_n[\mathbf{c}_n^{(\alpha_n)}]] - tr\mathbf{d}_n^*\beta^{-1}\mathbf{d}_n^T E_n[\mathbf{c}_n^{(\alpha_n)}\mathbf{c}_n^{(\alpha_n)\dagger}]) \tag{100}$$

et const. contient les termes qui sont indépendants de  $\theta_N$ . Par conséquent, afin de maximiser l'équation (99) par rapport à  $\theta_N$ , il suffit de maximiser  $I_N(\theta_N|\theta'_N)$ . La maximisation est effectuée par une procédure de programmation dynamique (Viterbi).

### Conclusion

Le problème de la détection de données via un canal dynamique non linéaire a été étudié. Nous avons proposé une nouvelle approche pour la détection des données via un canal dynamique non linéaire. Lorsque nous travaillons sur le modèle de l'espace d'états linéaire, le filtre de Kalman comme le meilleur estimateur peut être utilisé. Mais pour le modèle à commutation d'espace-état (SSSM) qui est l'hybride

## Résumé étendu en Français

---

du modèle de Markov caché, et du modèle de l'espace d'états linéaire, nous devons utiliser l'approche EM comme l'approche naturelle. Des simulations numériques ont montré que la méthode proposée était meilleure que l'approche PSP.





# Chapter 1

## Introduction

### 1.1 Research Context

The dispersive and time varying nature of the propagation environment is a fundamental limiting factor in the performance of mobile wireless systems. In order to represent the signal distortion under practical situations, suitable models of the channel are required [1, 2]. In the multipath propagation each path delay may be conceptually divided into two parts: the cluster delay, which is on the order of a symbol interval and can be preserved in the channel model, and the fine delay, which is on the order of the carrier period and can be represented together with the path attenuation as a time-varying complex gain. In addition, each path undergoes a Doppler shift, due to the relative movement of the transmitter and receiver. Thus, the received signal is the sum of many Doppler shifted, scaled and delayed versions of transmitted signal. The complex envelope of this signal usually obeys a Rayleigh distribution, which is widely used in modeling the channel [2–4]. For low signaling rates, the channel shows more time selectivity and it is more flat in the frequency domain over the signal bandwidth. This channel is called flat fading, where the received signal is scaled by a complex gain. In the flat fading case, the multipath delay spread is small and the equalization consists of estimating the complex gain of the channel due to the fine delay and compensating for its effect. At higher signaling rates, the channel is typically-selective, but usually the channel characteristics changes very slowly compared to the symbol rate. Equalizers have been historically

---

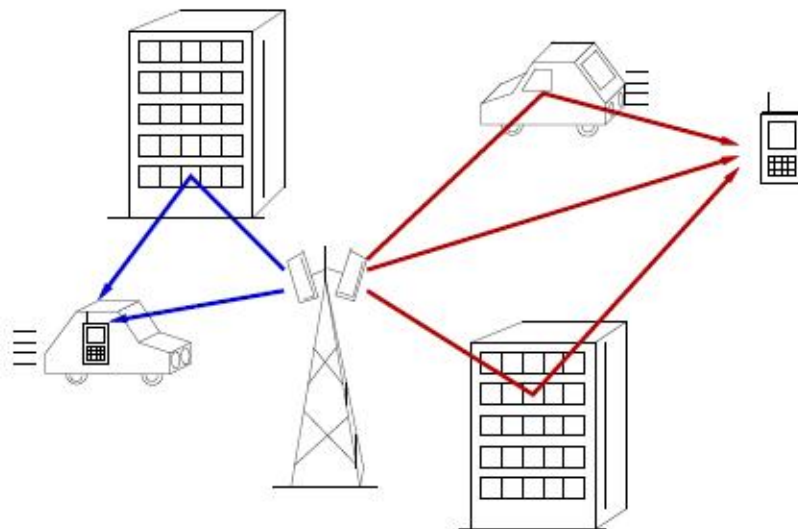


Figure 1.1: Time-varying channel occurs by mobility of transmitter, reflector and/or receiver.

developed for such channels [5]. In general, channels can be classified into four categories based on their time-selectivity and frequency-selectivity behavior. The most general case for equalization is the time-selective and frequency-selective channel, also known as doubly selective channel (see Fig 1.1).

The inter-symbol interference (ISI), and the time-variable behavior of the multipath channel that results in fading, are the major reasons for concern in mobile wireless channels. ISI causes each transmitted symbol to be extended over a much longer interval than its original duration [1] [6]. When the incoming signals from different paths interfere destructively with each other, fading happens. In order to repair the channel behavior and maintain a reliable communication with acceptable bit error rate (BER) [7–9], we must use equalization techniques. Equalization in general consists of estimating the response or states of the channel and using the estimate to compensate for the channel effects so as to improve transmission system performance. Equalization for the doubly selective channel is a challenging problem. Instead of estimating only one random process, as in time-varying flat fading channel, there are many parameters to estimate. This can be in the form of estimating the channel impulse response (CIR) vector, or estimating the adaptive weights for taps in tapped delay line model [1,6]. In the frequency-selective chan-

## 1.2. State-of-The-Art in Equalization and Estimation

---

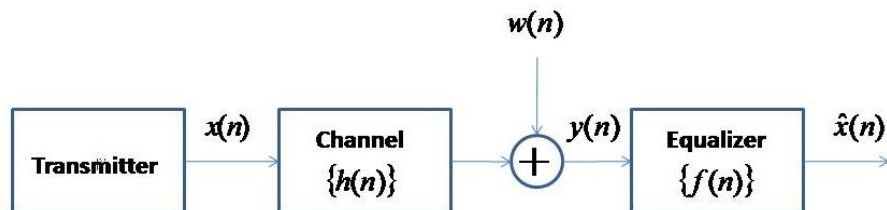


Figure 1.2: Baseband model of a digital communication system, consisting of the transmitter, channel and equalizer (receiver).

nels, equalization consists of estimating the CIR and then using this information to regulate the parameters of some form of linear or nonlinear filter to compensate for the frequency-selective effects. The linear filter can be in form of a transversal equalizer [5], and the nonlinear filter can be a decision feedback equalizer (DFE) [5, 6], a maximum likelihood sequence detector (MLSD) [10, 11], or a maximum a posteriori (MAP) type of detector [6, 12].

## 1.2 State-of-The-Art in Equalization and Estimation

### 1.2.1 Basic Baseband Model of a Communication System

Almost all baseband digital communication systems consist of three basic building blocks, the transmitter, the channel and the equalizer (receiver) as shown in Figure 1.2. In the figure,  $x(n)$  is the transmitted symbol,  $\{h(n)\}$  is the multipath gains (MGs) sequence,  $w(n)$  is the additive noise sample,  $y(n)$  is the received sample,  $\{f(n)\}$  are the equalizer coefficients,  $\hat{x}(n)$  is the estimated signal after equalization and  $n$  is the discrete time index.

The transmitter is one of the most important parts of a digital communication system. The main function of the transmitter is to convert the raw data into an appropriate form suitable for transmission, e.g., the voice signal is sampled and encoded into binary signals to transmit. The original band of frequencies occupied by the encoded binary signals is called a baseband signal. The baseband signal has wide frequency spectrum centered at zero frequency, which is bandlimited before

transmission with a filter called a transmit filter. Usually the binary signals contain low frequencies, which are difficult to propagate. Hence, signals centered around higher frequencies are preferred. The second function of the transmitter is therefore to shift the frequency spectrum of the bandlimited signal to some higher frequency centered at  $f_c$  called the carrier frequency. To shift the frequency spectrum to a higher frequency, the bandlimited signal is multiplied by the high frequency sinusoidal signal of frequency  $f_c$  [1]. The output signal is termed as the passband and the mapping of the baseband signal into the passband signal is called modulation.

The transmitted signal passes through the channel that can be considered as a finite impulse response (FIR) filter and then arrives at the receiver. The received signal is again passed through a filter called the receive filter matched to the frequency band of the transmitter. In general, the effects of the transmit filter, the transmission medium and the receive filter are included in the channel model  $h(n)$  with finite time support. Therefore, if the support of the modelled channel is  $L$  and the sampling rate at the receiver is equal to the symbol transmission rate then the received signal can be written as

$$y(n) = \sum_{l=0}^{L-1} h(l)x(n-l) + w(n). \quad (1.1)$$

Before proceeding, we consider the following assumptions:

- The transmitted symbols  $\{x(n)\}$  are independently and identically distributed (i.i.d).
- The additive noise samples  $\{w(n)\}$  are zero mean white circularly Gaussian distributed with variance  $\beta$ .
- The channel is an FIR filter of support  $L$ .

Let the multipath component  $h(m)$  possess the highest relative amplitude in the sequence  $\{h(n)\}$ , this multipath is termed as main multipath, multipaths before and after the main multipath are respectively called pre- and post-cursors. The energy of the wanted signal is conveyed mainly by the contribution of the main path. In addition to that the received signal is also contributed to by the convolution of pre and post-cursors. Therefore, the received signal in (1.1) can be written as

$$y(n) = h(m)x(n-m) + \sum_{l=0, l \neq m}^{L-1} h(l)x(n-l) + w(n) \quad (1.2)$$

## 1.2. State-of-The-Art in Equalization and Estimation

---

where the term  $\sum_{l=0, l \neq m}^{L-1} h(l)x(n-l)$  is the interference from the other symbols due to pre- and post-cursors and is called ISI. In the noiseless case, if  $h(m)$  is known then the decision device at the receiver may reconstruct the transmitted signal  $x(n)$  iff

$$|h(m)| > \sum_{l=0, l \neq m}^{L-1} |h(l)| \quad (1.3)$$

However, if this condition is not satisfied an error may occur. The ISI effects can be cancelled by employing an equalizer that accumulates the energy transmitted for  $x(n)$ , reduces the energy from other transmitted symbols and produces a decision variable,  $\hat{x}(n)$ . Ideally,

$$\hat{x}(n) = x(n) + \vartheta(n) \quad (1.4)$$

where  $\vartheta(n)$  is additive colored noise with the same variance as  $w(n)$ . If equalization is effective, a decision device can determine  $x(n)$  with the same reliability as if the channel did not introduce any ISI. If  $\{f(n)\}$  is the impulse response sequence of the equalizer, then ideally, in the absence of additive noise the following identity will hold

$$\begin{aligned} h(n) * f(n) &= \delta(n) \\ &= \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases} \end{aligned} \quad (1.5)$$

Although in practice a non zero delay and complex amplitude scaling can be tolerated.

### 1.2.2 Equalization Techniques

Equalization techniques have been developed since the 1960s/70s, [13–15], and the research in this area is continuously evolving to provide better performance. One of the reasons for this on going research is due to the ever increasing demands for higher capacity and efficient bandwidth utilization of the channel. In the sense of minimizing the probability of sequence error, channel equalization techniques to mitigate the effects of bandlimited time dispersive channel may be subdivided into two general types optimal and suboptimal equalization. In this section, the most commonly used equalizers in practice are briefly reviewed. Here we assume that channel state information is known at the receiver.

### 1.2.2.1 ML per Sequence

Maximum likelihood sequence estimator (MLSE) was first proposed by Forney [11] in 1973. It is an optimal equalizer in the sense that it minimizes the probability of sequence error. In MLSE a dynamic programming algorithm known as the Viterbi algorithm is used to determine in a computationally efficient manner the most likely transmitted sequence from the received noisy and ISI-corrupted sequence [1, 16]. Because the Viterbi decoding algorithm is the way in which the MLSE equalizer is implemented, the equalizer is often referred to as the Viterbi equalizer. The MLSE equalizer tests all possible data sequences, rather than decoding each received symbol by itself, and chooses the data sequence that is the most probable in all combinations. Therefore, for a memoryless channel,  $p(\mathbf{y}|\mathbf{x})$  denotes the conditional probability of receiving  $\mathbf{y}$ , when code vector  $\mathbf{x}$  corresponding to sequence  $\{x(n)\}$  is transmitted. Then, the likelihood function,  $p(\mathbf{y}|\mathbf{x})$ , can be written as

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{(\pi\beta)^N} \prod_{n=1}^N e^{-\frac{\|y(n)-x(n)\|^2}{\beta}} \quad (1.6)$$

For channels with memory the likelihood function to maximize can be written as

$$p(\mathbf{y}|\mathbf{c}; \mathbf{d}_n) = \frac{1}{(\pi\beta)^N} \prod_{n=1}^N e^{-\frac{\|y(n)-\mathbf{d}_n^T \mathbf{c}\|^2}{\beta}} \quad (1.7)$$

where  $\mathbf{d}_n = [x(n), x(n-1), \dots, x(n-L+1)]^T$  and  $\mathbf{c} = [h(0), h(1), \dots, h(L-1)]^T$ . The main drawback of the MLSE is its search complexity, measured in number of states, which increases exponentially with the channel support and large constellation points in the modulation, such as 8PSK or 16PSK schemes. Let  $M$  be the order of modulation and  $L$  the support of the channel then the number of equalizer states will be  $M^L$ .

### 1.2.2.2 MAP per Symbol

The advent of "turbo processing", has revitalized interest in MAP equalization in preference to MLSE. The MAP algorithm is a symbol-by-symbol estimator which accepts observations (in the form of matched filter outputs) together with *a priori* symbol probabilities (soft inputs) and produces *a posteriori* symbol probabilities.

---

## 1.2. State-of-The-Art in Equalization and Estimation

---

The decoded symbol is declared to be that with the maximum *a posteriori* probability which minimizes the bit error rate (BER). When the probabilities are retained as soft outputs, the MAP equalizer is suited to receiver structures in which subsequent stages (e.g., outer decoding) utilize soft decisions. The BCJR algorithm [17] is an efficient algorithm for MAP estimation. This algorithm calculates the *a posteriori* probability of each transmitted symbol, i.e.,  $p(x(n) = q | \mathbf{y}_0^{N-1})$ , for each of the  $Q$ -ary symbols,  $q$ , where  $\mathbf{y}_{k_1}^{k_2}$  is the set of observations  $\{y(k_1), \dots, y(k_2)\}$ . When soft outputs are required (e.g., in turbo processing), the *a posteriori* probabilities (APP) are retained, and the algorithm may be referred to as the APP algorithm. The  $j$ th state of the trellis at time  $n$  is labeled  $S_{n,j}$ , which represents one of the  $Q^{L-1}$  possible values for  $\{x(n-L+2), \dots, x(n)\}$ . We denote the particular value by  $\{\tilde{x}(n-L+2, j), \dots, \tilde{x}(n, j)\}$ , where tilde indicates a hypothesized value. For ISI channels more than one  $S_{n,j}$  will correspond to a particular  $x(n) = q$ , therefore:

$$p(x(n) = q | \mathbf{y}_0^{N-1}) = \sum_{j:x(n)=q} p(S_{n,j} | \mathbf{y}_0^{N-1}) \quad (1.8)$$

The *a posteriori* state probabilities,  $\gamma_{n,j} = p(S_{n,j} | \mathbf{y}_0^{N-1})$ , can be calculated using the forward backward recursion procedure. The forward variable is  $\alpha_{n,j} = p(S_{n,j}, \mathbf{y}_0^n)$  and the backward variable is  $\beta_{n,j} = p(\mathbf{y}_{n+1}^{N-1} | S_{n,j})$ . Denoting the *a priori* transition probability from state  $S_{n-1,i}$  to state  $S_{n,j}$  by  $a_{n,ij}$ , and the probability of observations  $y_n$  on that transition by  $b_{n,ij}$ , the recursions for the forward and backward variables are

$$\alpha_{n,j} = \sum_{i=1}^{Q^*} b_{n,ij} a_{n,ij} \alpha_{n-1,i} \quad (1.9)$$

$$\beta_{n,j} = \sum_{i=1}^{Q^*} b_{n+1,ij} a_{n+1,ij} \beta_{n+1,i} \quad (1.10)$$

where  $Q^*$  is the number of states in the trellis and the *a priori* state transition probabilities are derived from the *a priori* information provided to the equalizer (soft inputs)

$$a_{n,ij} = p(S_{n,j} | S_{n-1,i}) = p(\tilde{x}(n-L+2, j), \dots, \tilde{x}(n, j) | \tilde{x}(n-L+1, i), \dots, \tilde{x}(n-1, i)). \quad (1.11)$$

also the observation probabilities  $b_{n,ij}$  are then given by:

$$b_{n,ij} = p(y(n) | S_{n-1,i}, S_{n,j}) = \frac{1}{\pi\beta} e^{-\frac{1}{\beta} \|y(n) - \tilde{\mathbf{d}}_n^T \mathbf{c}\|^2} \quad (1.12)$$



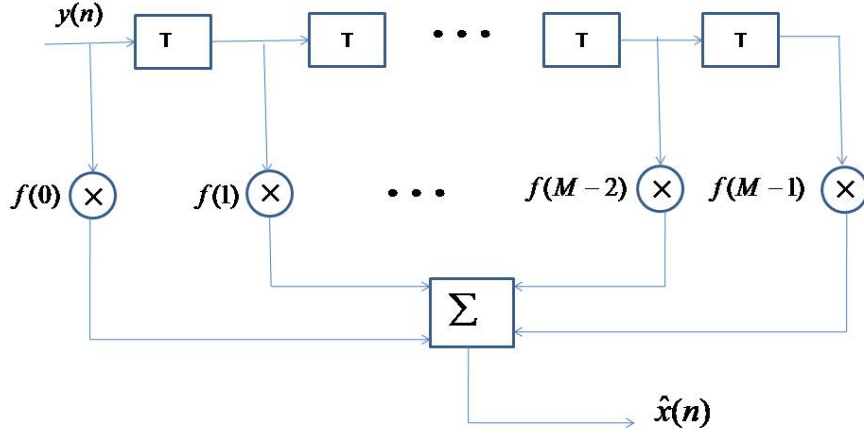


Figure 1.3: Linear Transversal Equalizer.

The initial values may be chosen to be equal (i.e.,  $1/Q^*$ ), or alternatively (without affecting any result)  $\alpha_{-1,1} = 1$  and  $\alpha_{-1,j} = 0$  for  $j = 2, \dots, Q^*$ . At the end of the block,  $\beta_{N,j} = 1/Q^*$  for  $j = 1, \dots, Q^*$  should be chosen, unless zero-tailing is used to ensure that  $\beta_{N,1} = 1$  and  $\beta_{N,j} = 0$  for  $j = 2, \dots, Q^*$ . It has been shown that the impact of zero-tailing on performance is not significant, especially for a large block size. The *a posteriori* state probabilities are then given by:

$$\gamma_{n,j} = p(S_{n,j} | \mathbf{y}_0^{N-1}) = \frac{\alpha_{n,j} \beta_{n,j}}{\sum_{j=1}^{Q^*} \alpha_{n,j} \beta_{n,j}} \quad (1.13)$$

### 1.2.2.3 Linear Equalization

A basic structure of a linear transversal equalizer (LTE) is shown in Figure 1.3. The computational complexity of this filter structure is a linear function of the channel dispersion length  $L$ . In such equalizers the current and past values of the received signal are linearly weighted by equalizer coefficients,  $f(l)$ , and assumed to produce the estimate of the transmitted signal as an output that can be written

## 1.2. State-of-The-Art in Equalization and Estimation

---

as [1]

$$\hat{x}(n) = \sum_{l=0}^{M-1} f^*(l)y(n-l) = \mathbf{f}^\dagger \mathbf{y}(n) \quad (1.14)$$

where  $(\cdot)^\dagger$  denotes the conjugate transpose operation,  $M$  is the length of equalizer taps,  $\mathbf{f} = [f(0), f(1), \dots, f(M-1)]^T$  is the tap weight vector and  $\mathbf{y}_n = [y(n), y(n-1), \dots, y(n-M+1)]^T$  is the received signal vector to estimate  $\hat{x}(n)$ .

Considerable research has been performed on the criterion for optimizing the equalizer coefficients,  $f(l)$ . Since the most meaningful measure of performance for a digital communications system is the average probability of error, it is desirable to choose the coefficients to minimize this performance index. However, the probability of error is a highly nonlinear function of  $\{f_l\}$ . Consequently, the probability of error as a performance index for optimizing the tap weight coefficients of the equalizer is impractical. Two criteria have found widespread use in optimizing the equalizer coefficients  $\{f_l\}$ . One is the peak distortion criterion which was introduced by Lucky [13, 18] and the other is the mean square error criterion which was proposed by Widrow [19] and developed by Godard [20].

**Peak Distortion Criterion:** The peak distortion is simply defined as the worst-case intersymbol interference at the output of the equalizer. The minimization of this performance index is called the peak distortion criterion. If we consider

$$d(k) = \sum_{j=0}^{M-1} f^*(j)h(k-j), \quad (1.15)$$

then

$$\hat{x}(n) = d(0)x(n) + \sum_{k=0, k \neq n}^{M+L-2} x(k)d(n-k) + \sum_{j=0}^{M-1} f^*(j)w(n-j). \quad (1.16)$$

The peak value of this interference, which is called the peak distortion is

$$\mathcal{D}(f) = \sum_{k=0, k \neq n}^{M+L-2} |d(k)| = \sum_{k=0, k \neq n}^{M+L-2} \left| \sum_{j=0}^{M-1} f^*(j)h(k-j) \right|. \quad (1.17)$$

From above equation we can see that there are  $M + L - 1$  non zero values in the response while the equalizer has  $M$  coefficient, therefore there is always some residual interference when the optimum coefficient are used.

For one special case, the solution for the minimization of  $\mathcal{D}(f)$  is known. This is the case in which the distortion at the input to the equalizer, defined as

$$D_0 = \frac{1}{|f(0)|} \sum_{n=1}^L |f(n)| \quad (1.18)$$

is less than unity. Under this condition, the peak distortion  $\mathcal{D}(f)$  is minimized by selecting the equalizer coefficients, to force  $d(n) = 0$  for  $1 \leq n \leq M-1$  and  $d(0) = 1$ . In other words, the general solution to the minimization of  $\mathcal{D}(f)$ , when  $D_0 < 1$ , is the zero-forcing solution for  $\{d(n)\}$  in the rang  $1 \leq n \leq M-1$ . However, the values of  $\{d(n)\}$  for  $M \leq n \leq M+L-2$  are nonzero which constitute the residual ISI at the output of equalizer.

**Mean Square Error (MSE) Criterion:** A more robust criterion called the minimum mean square error (MMSE) is very commonly used. Here, the equalizer tap weights are chosen to minimize the mean squared error between the transmitted symbol and the output, the sum of all squares of all terms plus the power of the noise [13, 15]. The cost function for this criterion can be written as

$$\begin{aligned} J(\mathbf{f}) &= E\{\|\hat{x}(n) - x(n-d)\|^2\} \\ &= E\{\|\mathbf{f}^\dagger \mathbf{y}(n) - x(n-d)\|^2\}. \end{aligned} \quad (1.19)$$

To find the filter tap weights, minimization of this cost function with respect to  $\mathbf{f}$  yields the equalizer tap weight vector

$$\mathbf{f} = (\mathbf{H}\mathbf{H}^\dagger + \beta\mathbf{I}_M)^{-1}\mathbf{H}\mathbf{i}_d \quad (1.20)$$

where

$$\mathbf{H} = \begin{bmatrix} h_0 & 0 & \cdots & 0 & 0 \\ h_1 & h_0 & 0 & 0 & 0 \\ \vdots & \ddots & \ddots & 0 & 0 \\ h_{L-1} & \cdots & h_1 & h_0 & 0 \\ 0 & \cdots & h_{L-1} & \cdots & h_0 \end{bmatrix}, \quad M \times (M+L-2)$$

and  $\mathbf{i}_d$  is the  $d$ th column vector of an identity matrix of size  $(M+L-2) \times (M+L-2)$  and defines the delay in estimating the transmitted symbol. If the values of the channel impulse response (CIR) at the sampling instances are known, the  $M$

## 1.2. State-of-The-Art in Equalization and Estimation

---

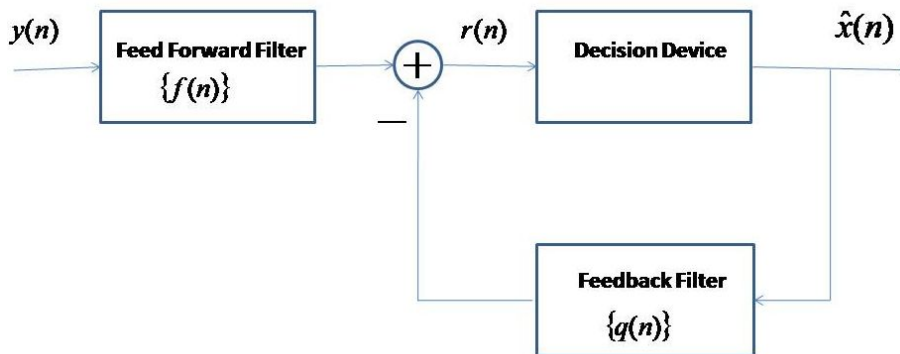


Figure 1.4: A basic structure of a DFE with forward and feedback filters.

coefficients of the zero forcing and MMSE equalizer can be obtained from (1.20). An LTE does not perform well in channels with deep spectral nulls in their frequency response characteristics [14]. In an attempt to compensate for channel distortion the LTE places a large gain in that null region, and as a consequence, significantly increases the noise in the received signal. Decision feedback equalization (DFE) is, however, superior to linear equalizers in applications where the channel has deep nulls or distortion is too severe for an LTE.

### 1.2.2.4 Decision-Feedback Equalization

Decision-feedback equalization (DFE) was proposed and analyzed by Austin [21]. Analyses of the performance of the DFE can be found in [22–26]. A basic structure of a DFE is shown in Figure 1.3. It is a nonlinear equalizer, which is widely used in situations where the ISI is very high. It exploits the already detected symbols to cancel the ISI by feeding them back. As shown in the figure, the equalized signal is the sum of the outputs of the forward and feedback filters.

The forward filter is just like the LTE and decisions made on the equalized signals are feedback via a second LTE. The idea behind the decision feedback equalization approach is that if the previous or past symbols are known then in current decision the ISI contribution of these symbols can be removed by subtracting past symbols with appropriate weighting from the equalizer output. The combined output of a

forward and feedback filter can be written as

$$r(n) = \sum_{k=0}^{N_f-1} f^*(k)y(n-k) - \sum_{l=1}^{N_b} q^*(l)\hat{x}(n-l) = \mathbf{f}^\dagger \mathbf{y}(n) - \mathbf{q}^\dagger \hat{\mathbf{x}}(n) \quad (1.21)$$

which is quantized into a hard decision by a nonlinear decision device

$$\hat{x}(n) = \text{sign}[r(n)] \quad (1.22)$$

where  $\mathbf{f} = [f(0), f(1), \dots, f(N_f - 1)]^T$  is the forward filter tap weight vector and  $\mathbf{q} = [q(1), q(2), \dots, q(N_b)]^T$  is the feedback tap weight vector. The vectors  $\mathbf{f}$  and  $\mathbf{q}$  are chosen to minimize jointly the minimum mean square error

$$\begin{aligned} J(\mathbf{f}, \mathbf{q}) &= E\{\|r(n) - x(n)\|^2\} \\ &= E\{\|\mathbf{f}^\dagger \mathbf{y}(n) - \mathbf{q}^\dagger \hat{\mathbf{x}}(n) - x(n)\|^2\} \end{aligned} \quad (1.23)$$

If the channel convolution matrix (CCM) is defined by

$$\mathbf{H} = \mathbf{H}_u + \mathbf{H}_c \quad (1.24)$$

where  $\mathbf{H}_u = [\mathbf{h}_1 \mathbf{h}_2 \dots \mathbf{h}_k | \mathbf{0} \dots \mathbf{0}]$  and  $\mathbf{H}_c = [\mathbf{0} \dots \mathbf{0} | \mathbf{h}_{k+1} \dots \mathbf{h}_{N-1}]$  are respectively referred to uncanceled and canceled symbols and  $\mathbf{h}_k$  is the  $k$ th column of CCM  $\mathbf{H}$ . Then the expression for forward and feedback tap weights can be written as [27] [28]

$$\mathbf{f} = \mathbf{R}_u^{-1} \mathbf{h}_k \quad (1.25)$$

$$\mathbf{q} = \mathbf{H}_c^\dagger \mathbf{f} \quad (1.26)$$

where  $\mathbf{R}_u = (\mathbf{H}_u \mathbf{H}_u^\dagger + \beta \mathbf{I})$ . The drawback of the DFE is that, at low SNR ratios, the already detected symbols may have higher probability of errors and when a particular incorrect decision is fed back, the DFE output reflects this error during the next few symbols due to incorrect decision on the feedback delay line. This phenomenon is called error propagation. It has been shown that the DFE always outperforms an LTE of equivalent complexity and offers ISI cancellation with reduced noise enhancement [15], hence it provides better BER performance as compared to an LTE [29].

---

## 1.2. State-of-The-Art in Equalization and Estimation

---

### 1.2.2.5 Reduced-State Sequence Estimation

Computational complexity of the MLSE receiver increases exponentially when channel memory is increased. Some suggestions were proposed in the literature [30–34] in order to reduction of computational complexity. One group by using an equalizer as a pre-filter, reduces the time-duration of the channel impulse response [31, 35–38]. Another group decreases the number of states by using Ungerboeck set partitioning to fuse some states permanently [39, 40]. Important methods from this latter group are reduced-state sequence estimation (RSSE) [32] and decision-feedback sequence estimation (DFSE) [33].

In MLSE, the Viterbi algorithm searches a trellis with  $M^K$  states ( $K$  is the length of the overall channel impulse response and  $M$  is the size of the signal set), and therefore has to keep track of  $M^K$  paths. For large value of  $M$ , the complexity can be large, even for very small  $K$ . RSSE can achieve nearly the performance of MLSE at significantly reduced complexity. The primary idea is the construction of trellises with a reduced number of states. These states are formed by combining the state of the ML trellis using Ungerboeck-like set partitioning principles [39, 40]. The RSSE is then implemented using the Viterbi algorithm to search this reduced state trellis. Suppose  $\mathbf{p}_n = [x_{n-1}, x_{n-2}, \dots, x_{n-K}]$  are trellis states which each element in the state vector can take one of  $M$  values, therefore the ML trellis has  $M^K$  states and there are  $M$  transitions to and from each state. In order to reduce the number of states, for each element  $x_{n-k}$  in the vector  $\mathbf{p}_n$ , a two dimensional set partitioning denoted as  $\Omega(k)$  are defined. Specifically, for the  $k$ th element  $x_{n-k}$ , the signal set is partitioned into  $J_k$  subsets where  $J_k$  can range anywhere from 1 to  $M$ . The index of the subset of a symbol  $x_i$  in the partitioning  $\Omega(k)$  is, in general denoted as  $a_i(k)$ , which can be taken as an integer between 0 and  $J_{k-1}$ . The set partitioning is constrained such that

- the numbers  $J_k$  are nonincreasing.
- the partitioning  $\Omega(k)$  is a further partition of the subsets of  $\Omega(k+1)$  for each  $k$  between 1 and  $K-1$ .

Therefore the subset state of a sequence at time  $n$  is defined as :

$$\mathbf{t}_n = [a_{n-1}(1), a_{n-2}(2), \dots, a_{n-K}(K)] \quad (1.27)$$

Since in (1.27),  $a_{n-k}(k)$  can take one of  $J_k$  values, the total number of states in the subset trellis is given by product of the  $J_k$ . If  $J_k = 1$  for all  $k$ , the RSSE degenerates into a zero-forcing DFE. If  $J_k = M$  for all  $k$ , the RSSE becomes an MLSE. Then by choice of the  $J_k$ , a tradeoff of performance versus complexity between a DFE and MLSE can be obtained. In order to optimize the performance/complexity tradeoff for the RSSE,  $J_k$  must be a power of two [40].

In Practice, there are still many applications where the principal source of MLSE complexity is the length of channel memory. In this case, a reduced-state trellis can be formed by simply truncating the ML state vector to some suitable length  $K' < K$  and defining the reduced state vector as  $\mathbf{p}'_n = [x_{n-1}, x_{n-2}, \dots, x_{n-K'}]$ . In RSSE subset formulation, this is equivalent to choosing  $J_k = M$  for  $k$  between 1 and  $K'$  and  $J_k = 1$  for  $k$  between  $K'+1$  and  $K$ . This is a special case of the RSSE which called a DFSE.

### 1.2.3 Channel Estimation Techniques

It is well known that in any wireless communication system, reliable coherent data detection is not possible unless an accurate channel estimate is available at the receiver. Using an estimator, the CIR of the mobile channel (time-variant) needs to be recognized. Typically, channel estimation techniques can be classified into two different families:

1. Techniques based on the transmission of training sequences [41–43].
2. Blind channel estimation methods (or joint data detection and channel estimation [44, 45]).

Blind methods are based on the statistics of the unknown data symbols and the statistical properties of the channel and do not require any pilot symbol. Between these two extremes, there exists semi-blind methods which require a small number of pilots, usually used for the algorithm initialization.

First families consists of pilot-only based channel estimation techniques and decision-directed techniques. Pilot-only based channel estimation techniques which are also called pilot symbol assisted modulation (PSAM), were introduced for single carrier systems by Moher and Lodge [46] and later analyzed by Cavers [47]. The main drawback of pilot-only based channel estimation techniques is the loss of spectral

## 1.2. State-of-The-Art in Equalization and Estimation

---

efficiency due to the pilot overhead. The simple idea of decision-directed method [6] is that in the absence of transmission errors, one can use the detected symbols as *a posteriori* reference signals for channel estimation instead of pilot symbols. A pioneering work in this area is that initiated by Frenger and Svensson [48] where a decision directed coherent detector for single carrier and multicarrier systems based on an MMSE channel estimation is proposed. However, the latter technique assumes that at each instant, all the previous decisions are correct. Obviously, this is not a realistic assumption in practical situations.

The increasing need for high data rates motivated the search for "blind" channel identification and equalization methods as they save bandwidth by avoiding the use of training sequences. Numerous blind algorithms have been developed in the literature [49], where several works have focused specifically on multicarrier systems.

Blind methods can also be used in cooperation with training data in order to better track channel variations and to enable faster convergence. In that case, they are referred to as "semi-blind" methods [50]. Usually, one or two pilot symbols are transmitted at the beginning of each frame for synchronization and initial channel estimation purposes. Most of blind algorithms can be extended to a semi-blind method. An efficient and extensively used method for semi-blind channel estimation is that based on the EM algorithm. The EM algorithm is an iterative algorithm that can be used to approximate an ML or MAP solution of the unknown channel when the transmitted symbols are unknown at the channel estimator (blind situation). For a more detailed and general exposition of the EM algorithm, the reader is urged to read [46,51]. Several papers have addressed ML channel estimation without using any *a priori* information for the unknown channel. In [52] (see also [53]), the authors proposed several improvements to the EM algorithm for MIMO channel estimation. In particular, they proposed an unbiased EM channel estimator that outperforms the classical EM estimator.

In all of the above methods the overall bit error rate of the receiver depends on the quality of the channel estimation method. Therefore, the estimation of the fading channel with high accuracy plays a key role in the receiver design [1].

The process of channel estimation can be performed using of a linear filter [54] [55]. Channel information at time  $n$  is obtained by using received data up to and



including time  $n$ . The channel is modelled as a system with unknown parameters and the received data is considered as a noisy measurement of these parameters. Certain statistics, such as mean and covariance matrix, of the channel random parameters can be available to the estimator. The linear filter process the received data as a noisy measurement of the channel states and according to some statistical criterion minimize the effect of noise at the filter output. Considerable research has been performed on the criterion for optimizing the filter coefficient. Two criteria have found common use in optimizing the filter coefficients. One is the peak distortion criterion and the other is the mean square error criterion. The peak distortion is simply defined as the worst-case at the output of the filter. The minimization of this performance index is called the peak distortion criterion. In the MSE criterion, the filter coefficient are adjusted to minimize the mean square value of the estimation error. The estimation error is the difference between the true parameter value and the output of the estimator (the estimate).

The Kalman filter is a solution to minimize the mean square of the estimation error and it has been successfully used in many applications [1, 54–57]. The output of the Kalman filter is computed recursively, and each state update is computed from the new input data and the previous estimate. The mathematical formulation of the Kalman filtering problem can be described based on a state space model. The model itself is assumed to be known to the Kalman filter. The information is in the form of state space model parameters and the statistical knowledge of the system variables. An input random process can be used to determine the inner states of the model. The Kalman filter is an optimum estimator which receives a noisy measurement of the inner states and provides the minimum mean-squared estimation of the state values based on its knowledge of the system model and the received signal.

For implementation of the Kalman filter, the state space model of the channel must be known. When the system model information is not available sub-optimal methods such as the RLS family of adaptive filters can be employed. In [56] Sayed and Kailath showed that several different variants of the RLS algorithm can be directly related to the Kalman filtering problem. The optimum Kalman filter requires the exact parameters of the state space model and the second order statistics of the random model-parameters. The RLS algorithm is a special case of the Kalman filter

## 1.2. State-of-The-Art in Equalization and Estimation

---

where the required information about the state space model are simply replaced by constant values [56].

Another sub-optimal solution when the channel state space model is not available is the LMS algorithm [58]. The LMS algorithm has been widely used in practice due to its simplicity. The LMS algorithm uses a special estimate of the error surface gradient to update its state estimate. The performance of the receiver strongly depends on how well the estimator can track the rapid changes of the CIR in the fast fading conditions. In this section we present the Kalman filtering algorithm for the estimation of channel impulse response, and also the RLS and LMS algorithms as channel estimation methods.

### 1.2.3.1 The Kalman Filter

In 1960, R. E. Kalman introduce the Kalman filter [59] as an optimal linear minimum variance estimator. Since then, it has been widely applied in many areas in particular signal processing and communication. The Kalman filter provides linear, unbiased and minimum variance estimates for unknown state vectors of a linear state space model therefore it is considered as an optimal estimator.

In order to employ the Kalman filter to estimate the impulse response of the mobile fading channel, a linear state space model is introduced:

Let  $\mathbf{c}_n$  be a complex vector valued Gauss-Markov process, channel process obeying the dynamics

$$\mathbf{c}_n = \mathbf{F}\mathbf{c}_{n-1} + \mathbf{v}_n, \quad (1.28)$$

where  $\mathbf{F}$  is a deterministic matrix sequence, this channel is probed through the observation law

$$y_n = \mathbf{d}_n^T \mathbf{c}_n + w_n, \quad (1.29)$$

where  $\mathbf{d}_n$  is a vector-valued finite one-step Markov sequence data process independent of  $\mathbf{c}_n, w_n, \mathbf{v}_n$ .  $\mathbf{v}_n, w_n$  are independent normal zero-mean white noises of covariance matrix  $\mathbf{Q}$  and variance  $\beta$ . The set of values of  $\mathbf{d}_n$  is  $\Omega$ .

The Kalman filter is composed of two parts: the measurement update and the time update. In the measurement update stage the optimal Kalman filter uses its latest measurement of the channel output and minimizes mean squared error conditioned on the received data up to time  $n$ . The measurement update estimate

of the channel state at time  $n$ , is computed given observation  $\{y_0, y_1, \dots, y_n\}$  and will be denoted as  $\hat{\mathbf{c}}_{n|n}$ . The estimation error is defined as the difference between the true value of the channel state  $\mathbf{c}_n$  and the estimate  $\hat{\mathbf{c}}_{n|n}$ . The task of the Kalman filter is to minimize

$$E[(\mathbf{c}_n - \hat{\mathbf{c}}_{n|n})^\dagger (\mathbf{c}_n - \hat{\mathbf{c}}_{n|n}) | y_{0:n}] \quad (1.30)$$

The measurement update estimation is called a filtering process since it is performed by using data measured up to and including time  $n$ . The next step is a prediction process and is called time update estimation, in which the Kalman filter predicts the channel estimates at time  $n + 1$  based on the measurement up to and including time  $n$ . In this part the Kalman filter takes advantage of its information about the state space model and employs the state transition matrix to predict the channel at time  $n + 1$ . This estimate can be presented as  $\hat{\mathbf{c}}_{n+1|n}$ .

From the above discussion, the basic computation to perform the Kalman filtering algorithm involves an estimation of the states based on the current observation and a prediction for the next time instant. The prediction is independent of the observation sample and can be computed without waiting for the future observation. Therefore, the computations involved in the estimation and prediction can be done recursively and separated in two different groups called the measurement update equations and the time update equations as follows [54]:

Measurement update equations:

$$\hat{\mathbf{c}}_{n|n} = \hat{\mathbf{c}}_{n|n-1} + \mathbf{k}_n (y_n - \mathbf{d}_n^T \hat{\mathbf{c}}_{n|n-1}) \quad (1.31)$$

$$\mathbf{k}_n = \mathbf{P}_{n|n-1} \mathbf{d}_n^* (\mathbf{d}_n^T \mathbf{P}_{n|n-1} \mathbf{d}_n^* + \beta)^{-1} \quad (1.32)$$

$$\mathbf{P}_{n|n} = \mathbf{P}_{n|n-1} - \mathbf{k}_n \mathbf{d}_n^T \mathbf{P}_{n|n-1} \quad (1.33)$$

Time update equations:

$$\hat{\mathbf{c}}_{n+1|n} = \mathbf{F} \hat{\mathbf{c}}_{n|n} \quad (1.34)$$

$$\mathbf{P}_{n+1|n} = \mathbf{F} \mathbf{P}_{n|n} \mathbf{F}^\dagger + \mathbf{Q} \quad (1.35)$$

The covariance matrix for the measurement update estimation error,  $\mathbf{P}_{n|n}$  is defined as

$$\mathbf{P}_{n|n} = E[(\mathbf{c}_n - \hat{\mathbf{c}}_{n|n})(\mathbf{c}_n - \hat{\mathbf{c}}_{n|n})^\dagger] \quad (1.36)$$

## 1.2. State-of-The-Art in Equalization and Estimation

---

and the covariance matrix for the time update estimation error,  $\mathbf{P}_{n|n-1}$  is

$$\mathbf{P}_{n|n-1} = E[(\mathbf{c}_n - \hat{\mathbf{c}}_{n|n-1})(\mathbf{c}_n - \hat{\mathbf{c}}_{n|n-1})^\dagger] \quad (1.37)$$

The error covariance matrix is positive definite with Hermitian symmetry and provides a statistical description of the error in the estimates. It can be noticed that the error covariance matrix is computed in a recursive form and is independent of the measurements from the channels,  $y_n$ . This means that any set of measurements have the same effect in eliminating the uncertainty about  $\mathbf{c}_n$ . The Kalman gain,  $\mathbf{k}_n$ , shows the influence of the new measurement,  $y_n$ , in modifying the estimate  $\hat{\mathbf{c}}_{n|n-1}$ . The Kalman gain is also independent of the input measurement. In general case,  $\mathbf{P}_{n|n-1}$  and  $\mathbf{k}_n$  can be precomputed before the filter is actually run. The initial conditions for the state estimate,  $\hat{\mathbf{c}}_{0|-1}$ , and for the error covariance matrix,  $\mathbf{P}_{0|-1}$ , are required to start the recursive loop. In the absence of observed data at time  $n = 0$ , we may choose the initial estimate as [55]

$$\hat{\mathbf{c}}_{0|-1} = E[\mathbf{c}_0] \quad (1.38)$$

and the initial error covariance matrix as

$$\mathbf{P}_{0|-1} = E[(\mathbf{c}_0 - E[\mathbf{c}_0])(\mathbf{c}_0 - E[\mathbf{c}_0])^\dagger] \quad (1.39)$$

If the state vector size is assumed to be  $m$ , then the approximate number of multiplications/additions needs for the kalman filter is  $\mathcal{O}(m^3)$ . Using the measurement update equations, the Kalman filter estimates the next state vector of the linear system or the CIR based on a noisy measurement which is the input signal at the receiver. In this part of the estimation only the received signal and the information about the measurement matrix,  $\mathbf{d}_n$ , and the noise variance,  $\beta$ , are used. In the next stage using the time update equations, the Kalman filter yields its estimate of the next state vector according to its knowledge of the linear system parameters such as  $\mathbf{F}$  and  $\mathbf{Q}$ . To obtain the matrices  $\mathbf{F}$  and  $\mathbf{Q}$  the receiver has to estimate the maximum Doppler frequency shift, and calculate the fading filter parameters based on this estimation. Sometimes it is not practical to obtain the parameters of the channel state space model at the receiver. Therefore, with the lack of this information we may just perform the measurement update estimation and eliminate the state prediction stage of the Kalman filter. This results in the sub-optimal RLS estimator which is described in sequence.

### 1.2.3.2 The RLS Algorithm

The RLS algorithm [1,55] is a least square method to estimate the states of a system based on the noisy observation inputs. When the parameters of the state space model are unknown to the estimator the RLS algorithm can replace the Kalman filter. As mentioned before the Kalman filter minimized the estimation mean square error, which is a statistical average. In the RLS algorithm we deal directly with received data to minimize a weighted time average of  $y_n - \mathbf{d}_n^T \hat{\mathbf{c}}_{n|n-1}$ . We receive the signal  $y_n$  and we wish to minimize the cost function

$$\xi(i) = \sum_{n=1}^i \lambda^{i-n} \|y_n - \mathbf{d}_n^T \hat{\mathbf{c}}_{n|n-1}\|^2 \quad (1.40)$$

which is a time average squared error with exponential weighting. The parameter  $\lambda$  is a forgetting factor and we have  $0 < \lambda < 1$ .  $y_n - \mathbf{d}_n^T \hat{\mathbf{c}}_{n|n-1}$  is the noise component at the receiver according to the estimates of the channel impulse response. Using this cost function the estimator tries to estimate  $\hat{\mathbf{c}}_{n|n-1}$  so that  $\mathbf{d}_n^T \hat{\mathbf{c}}_{n|n-1}$  is as close as possible to the received signal noise.

The RLS algorithm to minimize the above cost function is given as

$$\hat{\mathbf{c}}_{n|n} = \hat{\mathbf{c}}_{n|n-1} + \mathbf{k}_n (y_n - \mathbf{d}_n^T \hat{\mathbf{c}}_{n|n-1}) \quad (1.41)$$

$$\mathbf{k}_n = \mathbf{P}_{n|n-1} \mathbf{d}_n^* (\mathbf{d}_n^T \mathbf{P}_{n|n-1} \mathbf{d}_n^* + \lambda)^{-1} \quad (1.42)$$

$$\mathbf{P}_{n|n} = \lambda^{-1} (\mathbf{P}_{n|n-1} - \mathbf{k}_n \mathbf{d}_n^T \mathbf{P}_{n|n-1}) \quad (1.43)$$

By comparing the Kalman filter and the RLS algorithm, we observe that the RLS algorithm is basically the same as the measurement update equations of the Kalman filter. The RLS estimator uses the information of the received signal to update its state estimates and the estimation is performed in one stage similar to the measurement update equations of the Kalman filter. The Kalman filter performs extra computations for predicting the states at the next time step using the time update equation. To do so, the Kalman filter uses its knowledge about the linear system, obtained from the matrix  $\mathbf{F}$ , and updates the estimated values once more. If the state vector size is assumed to be  $m$ , then the computational complexity of the RLS algorithm is  $\mathcal{O}(m^2)$ , while the computational complexity of the Kalman filter is  $\mathcal{O}(m^3)$ .

---

## 1.2. State-of-The-Art in Equalization and Estimation

---

### 1.2.3.3 The LMS Algorithm

The LMS algorithm is a classical tracking method and is important because of its simplicity and ease of computation [58]. If the state vector size is assumed to be  $m$ , then the computational complexity of the LMS algorithm is  $\mathcal{O}(m)$ . The state update formula is given as:

$$\hat{\mathbf{c}}_{n+1} = \hat{\mathbf{c}}_n + \mu(y_n - \mathbf{d}_n^T \hat{\mathbf{c}}_n) \mathbf{d}_n \quad (1.44)$$

where  $\mu$  is a constant step-size and regulates the speed and stability of the tracking method. Evans [60], proposed a variable step-size method for the LMS adaptive algorithm. The step-size factor used in the LMS algorithm for all the survivor paths is the same and fixed. The variable step-size LMS (VS-LMS) algorithm can be expressed as:

$$\hat{\mathbf{c}}_{n+1} = \hat{\mathbf{c}}_n + \mu_n(y_n - \mathbf{d}_n^T \hat{\mathbf{c}}_n) \mathbf{d}_n \quad (1.45)$$

and error is:

$$e_n = (y_n - \mathbf{d}_n^T \hat{\mathbf{c}}_n) \quad (1.46)$$

Notice the step-size parameter  $\mu_n$  becomes a vector indexed by each survivor path and is adjustable individually with time. This parameter can be updated using the VS algorithms when the survivor paths progress. The variable step-size LMS algorithm [60] updates the step size by multiplying (adding) or dividing (subtracting) the previous step size by a factor in the adaptation process

$$\begin{aligned} \text{if } \mu_{n+1} < \mu_{\max} \quad \text{then } \mu_{n+1} &= \mu_n \times \alpha \\ \text{if } \mu_{n+1} > \mu_{\min} \quad \text{then } \mu_{n+1} &= \mu_n / \alpha \end{aligned} \quad (1.47)$$

where  $\alpha$  is the step-size updating factor and  $\alpha > 1$  or

$$\begin{aligned} \text{if } \mu_{n+1} < \mu_{\max} \quad \text{then } \mu_{n+1} &= \mu_n + \alpha' \\ \text{if } \mu_{n+1} > \mu_{\min} \quad \text{then } \mu_{n+1} &= \mu_n - \alpha' \end{aligned} \quad (1.48)$$

where  $\alpha'$  is the step-size updating factor and  $\alpha' > 0$ . The above step-size parameter updating occurs based on the sign changes of the error vector. If its sign changes consecutively for a specified number  $m_0$  of times, the step size is decreased. On the other hand, if its sign stays the same for another specified number  $m_1$  of times,

the step size is increased. This VS algorithm proposed in [60], showed significant improvement over the conventional fixed step-size LMS with faster convergence and higher accuracy. Unfortunately, the above VS-LMS suffers from another drawback, i.e., the performance is very sensitive to the selection of another parameter  $\alpha$ . The algorithm transfers the performance dependency on step size  $\mu$  into the dependency on step-size updating factor  $\alpha$ . Zhu [61] proposed a new step-size update scheme to eliminate the dependency on any selection of parameters. The new variable step-size algorithm is based on the absolute estimation error. In this approach, a variable step-size factor was chosen for each path based on the estimated data sequence related to the survivor path. This approach essentially breaks and separate all possible dependencies between different paths to estimate the CIR, and for a fast time-varying system, this can improve the performance considerably. The step-size updating scheme is given by

$$\begin{aligned}\mu_{n+1} &= \min(\mu_n + |e_{n+1}|, \mu_{\max}) \\ \mu_{n+1} &= \max(\mu_n - |e_{n+1}|, \mu_{\min})\end{aligned}\tag{1.49}$$

where  $e_{n+1}$  is the estimation error on the last survivor path. The  $\mu_{\max}$  and  $\mu_{\min}$  are chosen to constrain the step size so that the mean square errors remain bounded while minimal tracking ability is obtained.

The performance of the RLS algorithm has been compared to that of LMS algorithm in the literature extensively [59, 62–64]. The RLS algorithm has a faster rate of convergence than the LMS algorithm, while the LMS algorithm exhibits better tracking behavior than the RLS algorithm.

### 1.2.4 Joint Data Detection and Channel Estimation

MLSE is the optimum detection technique for data transmitted over selective fading channels. The channel impulse response (CIR) must be known, to do the task of MLSE. Using an estimator, the CIR of the mobile channel (time-variant) needs to be recognized. On the other hand in order to perform of channel estimation the channel input data must be known, while, the channel input data are not exactly known.

## 1.2. State-of-The-Art in Equalization and Estimation

---

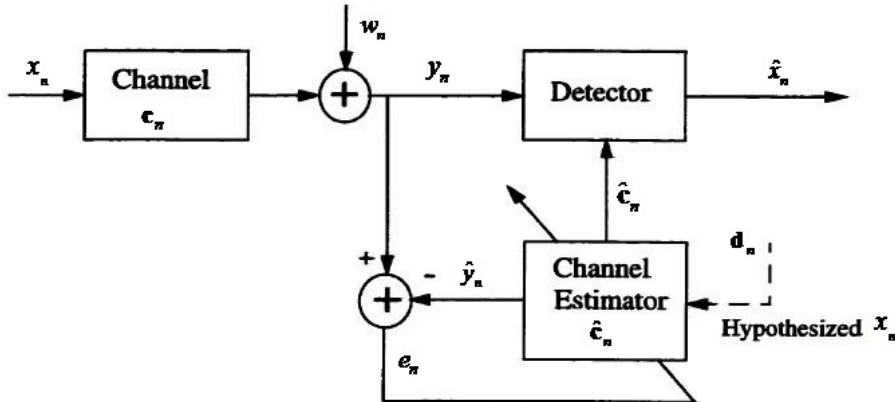


Figure 1.5: The adaptive receiver model for joint data detection and channel estimation.

Figure 1.5. illustrate the model of the adaptive receiver in which a channel estimator is employed to provide an estimate of the channel parameters for the detector. The estimator assesses the validity of its current estimate of CIR,  $\hat{\mathbf{c}}_n$ , by constructing  $\hat{y}_n = \mathbf{d}_n^T \hat{\mathbf{c}}_n$ , and comparing it against  $y_n$ . This is the convolution sum of the estimated CIR and the transmitted signal. The error signal defined as,  $e_n = y_n - \mathbf{d}_n^T \hat{\mathbf{c}}_n$ , is used for adaption of the channel estimator and to update the current estimate of CIR, used by the detector in branch metric generation. All of the estimation algorithms require the vector  $\mathbf{d}_n$ , which depends on the transmitted data sequence. However, the transmitted data is not available at the receiver in practical situations. This problem is sometimes called state estimation with model uncertainty, where the channel estimator has to estimate the states of the linear system and the  $\mathbf{d}_n$  is unknown.

One solution in this situation is to use the tentative decisions instead of the actual transmitted data to construct  $\mathbf{d}_n$ . This can be viewed as incorporating the decision feedback mechanism within the Viterbi decoder. It is clear that the quality of the channel estimation will depend on the quality of the tentative decisions fed to the estimator. In low SNR there will be a degenerative loop of poor estimates and poor decisions in the receiver. However, if the detector's decisions are highly reliable, then the decision-directed adaption of the channel estimates can be a successful method. There is normally a decision delay involved in this process which leaves the estimator with outdated information about the channel for adaption of its estimates.



This can be tolerable in slowly time varying channels and for fast fading channels it can results in a poor performance [6].

The other solution proposed in the literature [65–67] is to implement the channel estimation and the Viterbi algorithm in a per-survivor processing (PSP) fashion. In this case, to overcome the problem of uncertainty in  $\mathbf{d}_n$ , on each branch of the trellis a hypothesized data vector,  $\mathbf{d}_n$ , will be chosen according to the state transition corresponding to that branch. Then a separate estimator is required for any of the hypothesized  $\mathbf{d}_n$ , on each branch.

### 1.3 Contribution

The objective of this thesis is to propose new iterative detection schemes for both case of perfect and imperfect channel state information at the receiver. First, in order to reduce computational complexity of MLSE, we propose a new iterative detector based on continuation approach which used the EM algorithm [51]. In this scheme we take the general form of a continuation [68] or deterministic annealing process [69]. Starting from an exact formulation of the algorithm, we derive a series of approximation, leading to a final structure which has some similarities with the conventional block-iterated minimum mean square error (MMSE) equalizer [70, 71].

Second, under linear dynamic channel, in order to deal with channel uncertainty, we propose a combined state-space model which represent communication system. Rely on this representation we can use the Kalman filter as the best estimator.

Finally, if we consider non linear dynamic channel we can not use the Kalman filter as the best estimator. Here we use switching state-space model (SSSM) as non linear state-space model. This model combines the hidden Markov model (HMM) and linear state-space model (LSSM). In order to channel estimation and data detection, the EM procedure is used as the natural approach. In this way EKF and particle filters are avoided.

The main questions motivating our research can be summarized as follows:

1. How can the continuation approach be exploited to reduce the complexity of MLSE (treated in Chapter 2).
2. How to design an improved receiver in order to use Kalman filter as the best

## 1.4. Thesis Outline

---

estimator when the channel is unknown at the receiver (treated in Chapter 3).

3. How to perform the estimation of non-linear time-varying channels jointly with data detection (treated in Chapter 4).

In the following, we give an overview on how these questions have been addressed in this thesis.

## 1.4 Thesis Outline

This dissertation is organized in three main chapters:

In Chapter 2, under the assumption of unknown channel at the transmitter and perfectly known channel at the receiver, we propose a new iterative equalizer based on continuation approach. We formulate the sequence estimation problem in a precise setting. We obtain the main iteration formula using EM. Then we discuss about complexity and convergence issues. The use of a cyclic preamble allows carrying out the summation contained in the so-called main iteration formula into the frequency domain, while resort to a Gaussian approximation enables to simplify its evaluation. Finally, the numerical results proving the interest of the method and the validity of our successive approximation are presented, as well as the main feature of our approach are summed up in the conclusion.

In Chapter 3, we consider block transmission over a SISO doubly selective channel which is a linear dynamic channel. In order to accurately capture the dynamics of the channels, we formulate a first-order auto-regressive (AR) model and based on this representation of channel we use the Kalman filter as the best estimator of the channel parameters. Then, by joint use of Kalman filter and Viterbi algorithm, we propose our method. Finally we perform some simulation and give some conclusions.

In chapter 4, data detection under switching doubly selective channel (SDSC) as a nonlinear dynamic channel is considered. A first-order AR model is used to accurately represent the local behavior of the channel. Then we propose new iterative method based on EM approach. After determining the conditional expectation, the maximization is performed by dynamic programming (Viterbi) procedure. We

finally perform some simulation results and conclude remarks on our assumptions.

# Chapter 2

## A Continuation Approach to Iterative Sequence Estimation

### 2.1 Introduction

Maximum likelihood sequence estimation (MLSE) of discrete symbols from the noisy output of a convolutional channel has always motivated quite an intensive research. As well known, the complexity of MLSE being exponential in the channel memory and in the symbol alphabet cardinality is quickly unmanageable and forces to resort to sub-optimal approaches. This old issue begins in the seventies [29, 72]. Per-block minimum mean square estimation (MMSE) interference cancelation idea explicitly appears in [70, 71]. Noteworthy advances in block iterative MMSE equalization are later found in [73]. The structure we derive in this chapter is primarily based on the conversion of the original discrete search problem into a continuous parameter estimation problem. In general, the idea of approaching an original optimization cost function by a sequence of more tractable functions is known as continuation [68] or deterministic annealing [69]. In our context, such a process can be formulated as follows: the joint Dirac measures centered on the constellation symbols are replaced by a sequence of mixtures of Gaussian probability density functions (pdfs) converging towards the former. Also, the original a posteriori probability (APP) of symbols (discrete in nature) is set as the limit of a sequence of continuous ones. At the beginning of the deterministic annealing process, the pa-

---

---

## 2. A Continuation Approach to Iterative Sequence Estimation

---

parameter variance (or 'temperature') of Gaussian is chosen high enough so as to ensure a unique maximum for the transformed APP. This optimum is reached via a few iterations of an expectation-maximization (EM) procedure [51] with an appropriate choice of hidden variables. The variance is then decreased slowly enough to ensure that the last estimate of the data block remains within the attraction basin of the new absolute maximum. A new estimate is computed and the whole process is iterated until a sufficiently small value of the variance is reached. In this way, a sequence of posterior estimates is constructed converging towards the maximizer of the discrete posterior probability distribution. This chapter expounds the construction and implementation of this process. It is organized as follows. First the system model and problem formulation are introduced in section 2.2. Then, in section 2.3, we formulate the sequence estimation problem in a precise setting and the main iteration formula is derived in subsection 2.3.2. The use of a cyclic preamble allows carrying out the summation contained in the so-called main iteration formula into the frequency domain, while the resort to a Gaussian approximation enables to simplify its evaluation. The suboptimal resulting algorithm has an interesting structure detailed in section 2.4. Also complexity and convergence issues are discussed in section 2.5. Numerical results proving the interest of the method and the validity of our successive approximations are presented in section 2.6. Finally, the main features of our approach are summed up in the conclusion.

### 2.2 System Model and Problem Formulation

Let  $\mathbf{x}$  be an  $N$ -dimensional vector whose components belong to a finite alphabet  $\mathcal{X}$ . Components (sometimes referred to as constellation symbols) of  $\mathbf{x}$  are independent and uniformly distributed (i.i.d.). We consider the block transmission of  $\mathbf{x}$  over a single-input single-output convolutional channel with memory  $M$ . The discrete-time baseband-equivalent  $N$ -dimensional received vector  $\mathbf{y}$  is given by

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{w} \tag{2.1}$$

where  $\mathbf{H}$  is the  $N \times N$  Toeplitz channel matrix and  $\mathbf{w}$  is a vector of additive noise modeled as an  $N$ -dimensional zero-mean circularly symmetric Gaussian random vector with covariance matrix  $\mathbb{E}\{\mathbf{w}\mathbf{w}^\dagger\} = \beta\mathbf{I}_N$ . We assume that the channel is unknown

## 2.3. A Continuation Approach

---

at the transmitter and perfectly known at the receiver. The objective function to maximize is the joint conditional probability  $p(\mathbf{x}|\mathbf{y}, \mathbf{H})$ , i.e., we want to find out

$$\mathbf{x}^* = \arg \max_{\mathbf{x} \in \mathcal{X}^N} p(\mathbf{x}|\mathbf{y}, \mathbf{H}) \quad (2.2)$$

This is a discrete optimization problem, whose brute force resolution is intractable for large  $\mathcal{X}$  and  $N$ . In the sequel, conditioning by  $\mathbf{H}$  is implicit and omitted in probability expressions for the sake of notation simplicity.

## 2.3 A Continuation Approach

### 2.3.1 Principle

We propose to reformulate the original discrete optimization problem as follows

$$\mathbf{x}^{(\alpha)*} = \arg \max_{\mathbf{x} \in \mathbb{C}^N} p^{(\alpha)}(\mathbf{y}, \mathbf{x}) \quad (2.3)$$

where

$$p^{(\alpha)}(\mathbf{y}, \mathbf{x}) = p(\mathbf{y}|\mathbf{x})p^{(\alpha)}(\mathbf{x}) \quad (2.4)$$

is a family of pdfs in which the likelihood  $p(\mathbf{y}|\mathbf{x})$  is given by

$$p(\mathbf{y}|\mathbf{x}) = (\pi\beta)^{-N} e^{-\frac{1}{\beta}\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2} \quad (2.5)$$

and in which the support of the pdf  $p^{(\alpha)}(\mathbf{x})$  converges towards the support of the probability mass function (pmf)  $p(\mathbf{x})$  as  $\alpha \rightarrow 0^+$ . For computation convenience, the pdf  $p^{(\alpha)}(\mathbf{x})$  is defined as

$$p^{(\alpha)}(\mathbf{x}) = \sum_{\mathbf{s}} p^{(\alpha)}(\mathbf{x}, \mathbf{s}) = \sum_{\mathbf{s}} p(\mathbf{s})p^{(\alpha)}(\mathbf{x}|\mathbf{s}) \propto \sum_{\mathbf{s}} p(\mathbf{s})e^{-\frac{1}{\alpha}\|\mathbf{H}(\mathbf{x}-\mathbf{s})\|^2} \quad (2.6)$$

where the sum extends on  $\mathcal{X}^N$ . Clearly, this family of pdfs obeys the limiting criterion when  $\alpha \rightarrow 0^+$ .

---

## 2. A Continuation Approach to Iterative Sequence Estimation

---

### 2.3.2 Iterative Maximization of $P^{(\alpha)}(\mathbf{y}, \mathbf{x})$ Using EM

We declare a sequence  $\mathbf{s} \in \mathcal{X}^N$  of constellation symbols as a hidden random variable. The *log* pdf  $\log p^{(\alpha)}(\mathbf{y}, \mathbf{x})$  can be rewritten as

$$\begin{aligned}
 \log p^{(\alpha)}(\mathbf{y}, \mathbf{x}) &= \log p(\mathbf{y}|\mathbf{x}) + \log p^{(\alpha)}(\mathbf{x}) \\
 &= \log p(\mathbf{y}|\mathbf{x}) + \log \sum_{\mathbf{s}} p^{(\alpha)}(\mathbf{x}, \mathbf{s}) \\
 &= \log p(\mathbf{y}|\mathbf{x}) + \log \sum_{\mathbf{s}} p^{(\alpha)}(\mathbf{s}|\mathbf{x}') \frac{p^{(\alpha)}(\mathbf{x}, \mathbf{s})}{p^{(\alpha)}(\mathbf{s}|\mathbf{x}')} \quad (2.7)
 \end{aligned}$$

with  $\mathbf{x}'$  an initial estimate of the (continuous) sequence  $\mathbf{x}$ . Then, using Jensen's inequality [51], it can be lower bounded as

$$\log p^{(\alpha)}(\mathbf{y}, \mathbf{x}) \geq \chi - \frac{1}{\beta} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 + \sum_{\mathbf{s}} p^{(\alpha)}(\mathbf{s}|\mathbf{x}') \log p^{(\alpha)}(\mathbf{x}, \mathbf{s})$$

where  $\chi$  is a group of terms independent of  $\mathbf{x}$ . Finally, the (continuous) sequence  $\mathbf{x}$  is selected as the one maximizing the above lower bound, i.e., corresponding to the zero of its gradient

$$\frac{2}{\beta} \{\mathbf{H}^\dagger \mathbf{y} - \mathbf{H}^\dagger \mathbf{H}\mathbf{x}\} + \mathbb{E}_{\mathbf{s}|\mathbf{x}'}[\nabla_{\mathbf{x}} \log p^{(\alpha)}(\mathbf{x}|\mathbf{s})] = 0 \quad (2.8)$$

where the operator  $\mathbb{E}_{\mathbf{s}|\mathbf{x}'}[\cdot]$  refers to the expectation over all sequences  $\mathbf{s}$  with respect to the probability measure  $p^{(\alpha)}(\mathbf{s}|\mathbf{x}')$ . Applying Bayes rule, the latter can be expressed as

$$p^{(\alpha)}(\mathbf{s}|\mathbf{x}') = \frac{p(\mathbf{s})p^{(\alpha)}(\mathbf{x}'|\mathbf{s})}{\sum_{\mathbf{s}'} p(\mathbf{s}')p^{(\alpha)}(\mathbf{x}'|\mathbf{s}')} \quad (2.9)$$

From equation (2.8), we obtain the recursive formula

$$\mathbf{x} = \gamma(\mathbf{H}^\dagger \mathbf{H})^{-1} \mathbf{H}^\dagger \mathbf{y} + (1 - \gamma) \mathbb{E}_{\mathbf{s}|\mathbf{x}'}[\mathbf{s}] \quad (2.10)$$

where  $\gamma$  is the parameter defined as

$$\gamma = (1 + \frac{\beta}{\alpha})^{-1} \in ]0, 1[ \quad (2.11)$$

which gradually decreases as  $\alpha \rightarrow 0^+$  at each iteration (for fixed  $\beta$ ) and where

$$\mathbb{E}_{\mathbf{s}|\mathbf{x}'}[\mathbf{s}] = \frac{\sum_{\mathbf{s}} \mathbf{s} p(\mathbf{s}) e^{-\frac{1}{\alpha} \|\mathbf{H}(\mathbf{x}' - \mathbf{s})\|^2}}{\sum_{\mathbf{s}} p(\mathbf{s}) e^{-\frac{1}{\alpha} \|\mathbf{H}(\mathbf{x}' - \mathbf{s})\|^2}} \quad (2.12)$$

### 2.3. A Continuation Approach

---

Note that  $p(\mathbf{s}) = |\mathcal{X}|^{-N}$  for independent uniformly distributed constellation symbols.

**Remark 1.** The sum-over-states term can be easily viewed as some kind of soft decision in the classical terminology of block-iterative MMSE equalization. To make it clearer, let us consider the case of i.i.d. discrete symbols belonging to  $\mathcal{X} = \{-1, +1\}$ . We obtain

$$\begin{aligned} \mathbb{E}_{\mathbf{s}|\mathbf{x}'}[s_n] &= \frac{\sum_{\mathbf{s}:s_n=+1} e^{-\frac{1}{\alpha}\|\mathbf{H}(\mathbf{x}'-\mathbf{s})\|^2} - \sum_{\mathbf{s}:s_n=-1} e^{-\frac{1}{\alpha}\|\mathbf{H}(\mathbf{x}'-\mathbf{s})\|^2}}{\sum_{\mathbf{s}} e^{-\frac{1}{\alpha}\|\mathbf{H}(\mathbf{x}'-\mathbf{s})\|^2}} \\ &= \tanh \left\{ \frac{1}{2} \log \left( \frac{\sum_{\mathbf{s}:s_n=+1} e^{-\frac{1}{\alpha}\|\mathbf{H}(\mathbf{x}'-\mathbf{s})\|^2}}{\sum_{\mathbf{s}:s_n=-1} e^{-\frac{1}{\alpha}\|\mathbf{H}(\mathbf{x}'-\mathbf{s})\|^2}} \right) \right\} \end{aligned} \quad (2.13)$$

#### 2.3.3 Interpretation from Statistical Mechanics

In statistical mechanics, the quantity defined by (2.12) is called a sum-over-states and the parameter  $\gamma$  plays the role of a (normalized) temperature. The iteration formula (2.10) tells us that, at each iteration, the estimate  $\mathbf{x}$  results from a dynamical balance between the unconstrained ML estimate  $(\mathbf{H}^\dagger \mathbf{H})^{-1} \mathbf{H}^\dagger \mathbf{y}$  and the term  $\mathbb{E}_{\mathbf{s}|\mathbf{x}'}[\mathbf{s}]$  referred to as the mean-state (sequence) in the canonical ensemble with free energy  $\frac{1}{\alpha} \|\mathbf{H}(\mathbf{x}' - \mathbf{s})\|^2 - \ln p(\mathbf{s})$ .

#### 2.3.4 Exact Implementation of The Algorithm.

In order to implement the algorithm (2.10)-(2.12) it is necessary to evaluate the discrete sums contained in (2.12). This can be done exactly via the sum-product algorithm [74] or approximately as shown in section 2.4. Equation (2.10)-(2.12) were implemented with the sum-product algorithm and the convergence of the algorithm towards ML performance checked with several channel models and appropriate choices of the sequence of temperatures, thus confirming the correctness of the theory. In this case, the complexity of our algorithm turns out to be the same as the conventional Viterbi algorithm (i.e., exponential in the channel memory). However, an advantage of our algorithm is its direct applicability to multidimensional signals (e.g., Markov random fields).



## 2.4 Approximations

In order to attain a polynomial complexity in the channel memory (and in the alphabet cardinality), the evaluation of the sum-over-states (2.12) has to be approximated. Procedures like standard Mean-Field [75] were attempted and abandoned for results are rather poor. In the sequel, we slightly modify the original problem by assuming that a cyclic preamble has been sequentially inserted into the emitted sequence  $\mathbf{x}$ . This makes the square Toeplitz matrix  $\mathbf{H}$  circular. Let us introduce the Gram matrix  $G = \mathbf{H}^\dagger \mathbf{H}$  and the vector  $\xi = \mathbf{G}\mathbf{x}'$ . We have

$$\begin{aligned} \mathbb{E}_{\mathbf{s}|\mathbf{x}'}[s_n] &= \sum_{a \in \mathcal{X}} ap(s_n = a|\mathbf{x}') = \sum_{a \in \mathcal{X}} a \sum_{\mathbf{s}_{|n}} p(s_n = a, \mathbf{s}_{|n}|\mathbf{x}') \\ &= \sum_{a \in \mathcal{X}} ap(s_n = a) \sum_{\mathbf{s}_{|n}} p^{(\alpha)}(\mathbf{x}'|s_n = a, \mathbf{s}_{|n})p(\mathbf{s}_{|n}) \end{aligned} \quad (2.14)$$

Matrix  $\mathbf{H}$  being circular,  $\mathbf{G}$  is circular and can be reduced to diagonal form via the N-dimensional DFT matrix  $\mathbf{F}_N$ . This result is no more valid for the truncated matrix

$$\mathbf{G}_{|n} = \mathbf{U}_N^n \mathbf{H}^\dagger \mathbf{H} \mathbf{U}_N^{n\dagger} \quad (2.15)$$

Nevertheless,  $\mathbf{G}_{|n}$  being normal, there exists a unitary matrix  $\mathbf{Q}_{N-1}^n$  such that

$$\mathbf{\Lambda}^n = \text{diag}\{\lambda_1^n, \dots, \lambda_{N-1}^n\} = \mathbf{Q}_{N-1}^n \mathbf{G}_{|n} \mathbf{Q}_{N-1}^{n\dagger} \quad (2.16)$$

Let  $\bar{\xi}_{|n}$ ,  $\bar{\mathbf{s}}_{|n}$  and  $\bar{\mathbf{g}}_{n|n}$  the vectors defined as  $\mathbf{Q}_{N-1}^n \xi_{|n}$ ,  $\mathbf{Q}_{N-1}^n \mathbf{s}_{|n}$  and  $\mathbf{Q}_{N-1}^n \mathbf{g}_{n|n}$ , respectively. Starting from

$$\begin{aligned} \sum_{\mathbf{s}_{|n}} p^{(\alpha)}(\mathbf{x}'|s_n = a, \mathbf{s}_{|n})p(\mathbf{s}_{|n}) &\propto \\ \sum_{\mathbf{s}_{|n}} p(\mathbf{s}_{|n}) e^{\frac{1}{\alpha} [2\text{Re}(\xi_n^* a + \xi_{|n}^\dagger \mathbf{s}_{|n} - a^* \mathbf{g}_{n|n}^\dagger \mathbf{s}_{|n}) - g_{nn}|a|^2 - \mathbf{s}_{|n}^\dagger \mathbf{G}_{|n} \mathbf{s}_{|n}]} &\quad (2.17) \end{aligned}$$

we first reformulate the sum as

$$\begin{aligned} \sum_{\mathbf{s}_{|n}} p^{(\alpha)}(\mathbf{x}'|s_n = a, \mathbf{s}_{|n})p(\mathbf{s}_{|n}) &\propto \\ e^{\frac{1}{\alpha} [2\text{Re}(\xi_n^* a) - g_{nn}|a|^2]} \sum_{\mathbf{s}_{|n}} p(\mathbf{s}_{|n}) e^{\frac{1}{\alpha} [2\text{Re}(\xi_{|n}^\dagger \mathbf{s}_{|n} - a^* \mathbf{g}_{n|n}^\dagger \mathbf{s}_{|n}) - \mathbf{s}_{|n}^\dagger \mathbf{G}_{|n} \mathbf{s}_{|n}]} &\quad (2.18) \end{aligned}$$

## 2.4. Approximations

---

or equivalently, after the change of variable  $\bar{\mathbf{s}}_{|n} = \mathbf{Q}_{N-1}^n \mathbf{s}_{|n}$ , as

$$\begin{aligned} & \sum_{\mathbf{s}_{|n}} p^{(\alpha)}(\mathbf{x}' | s_n = a, \mathbf{s}_{|n}) p(\mathbf{s}_{|n}) \propto \\ & e^{\frac{1}{\alpha} [2\text{Re}(\xi_n^* a) - g_{nn} |a|^2]} \sum_{\bar{\mathbf{s}}_{|n}} p(\bar{\mathbf{s}}_{|n}) e^{\frac{1}{\alpha} [2\text{Re}(\bar{\xi}_{|n}^\dagger \bar{\mathbf{s}}_{|n} - a^* \bar{\mathbf{g}}_{n|n}^\dagger \bar{\mathbf{s}}_{|n}) - \bar{\mathbf{s}}_{|n}^\dagger \mathbf{\Lambda}^n \bar{\mathbf{s}}_{|n}]} \end{aligned} \quad (2.19)$$

Let us now focus on the discrete sum

$$\sum_{\bar{\mathbf{s}}_{|n}} p(\bar{\mathbf{s}}_{|n}) e^{\frac{1}{\alpha} [2\text{Re}(\bar{\xi}_{|n}^\dagger \bar{\mathbf{s}}_{|n} - a^* \bar{\mathbf{g}}_{n|n}^\dagger \bar{\mathbf{s}}_{|n}) - \bar{\mathbf{s}}_{|n}^\dagger \mathbf{\Lambda}^n \bar{\mathbf{s}}_{|n}]} \quad (2.20)$$

Both theory (central limit theorem) and numerical simulations show that, for large  $N$  (typically  $N \geq 64$ ), the components of  $\bar{\mathbf{s}}_{|n}$  are zero-mean Gaussian distributed with unit covariance. We thus apply a Gaussian approximation to further approximate (2.20) by

$$\begin{aligned} & \sum_{\bar{\mathbf{s}}_{|n}} p(\bar{\mathbf{s}}_{|n}) e^{\frac{1}{\alpha} [2\text{Re}(\bar{\xi}_{|n}^\dagger \bar{\mathbf{s}}_{|n} - a^* \bar{\mathbf{g}}_{n|n}^\dagger \bar{\mathbf{s}}_{|n}) - \bar{\mathbf{s}}_{|n}^\dagger \mathbf{\Lambda}^n \bar{\mathbf{s}}_{|n}]} \\ & = \pi^{-(N-1)} \int_{\mathbb{C}^{N-1}} e^{-\|\bar{\mathbf{s}}_{|n}\|^2} e^{\frac{1}{\alpha} [2\text{Re}(\bar{\xi}_{|n}^\dagger \bar{\mathbf{s}}_{|n} - a^* \bar{\mathbf{g}}_{n|n}^\dagger \bar{\mathbf{s}}_{|n}) - \bar{\mathbf{s}}_{|n}^\dagger \mathbf{\Lambda}^n \bar{\mathbf{s}}_{|n}]} \mathbf{dV}(\bar{\mathbf{s}}_{|n}) \end{aligned} \quad (2.21)$$

After some algebra, the RHS can be rewritten as

$$\begin{aligned} & \pi^{-(N-1)} e^{\frac{1}{\alpha} [(\bar{\xi}_{|n} - a \bar{\mathbf{g}}_{n|n})^\dagger (\mathbf{\Lambda}^n + \alpha \mathbf{I}_{N-1})^{-1} (\bar{\xi}_{|n} - a \bar{\mathbf{g}}_{n|n})]} \times \\ & \int_{\mathbb{C}^{N-1}} e^{-\frac{1}{\alpha} [(\bar{\mathbf{s}}_{|n} - (\mathbf{\Lambda}^n + \alpha \mathbf{I}_{N-1})^{-1} (\bar{\xi}_{|n} - a \bar{\mathbf{g}}_{n|n}))^\dagger (\mathbf{\Lambda}^n + \alpha \mathbf{I}_{N-1}) (\bar{\mathbf{s}}_{|n} - (\mathbf{\Lambda}^n + \alpha \mathbf{I}_{N-1})^{-1} (\bar{\xi}_{|n} - a \bar{\mathbf{g}}_{n|n}))]} \mathbf{dV}(\bar{\mathbf{s}}_{|n}) \end{aligned} \quad (2.22)$$

and finally, the initial sum amounts to

$$\begin{aligned} & \sum_{\mathbf{s}_{|n}} p^{(\alpha)}(\mathbf{x}' | s_n = a, \mathbf{s}_{|n}) p(\mathbf{s}_{|n}) \propto \\ & e^{\frac{1}{\alpha} [2\text{Re}(\xi_n^* a) - g_{nn} |a|^2 + (\bar{\xi}_{|n} - a \bar{\mathbf{g}}_{n|n})^\dagger (\mathbf{\Lambda}^n + \alpha \mathbf{I}_{N-1})^{-1} (\bar{\xi}_{|n} - a \bar{\mathbf{g}}_{n|n})]} \end{aligned} \quad (2.23)$$

and, by only saving terms involving a

$$\propto e^{\frac{1}{\alpha} [|a|^2 (\bar{\mathbf{g}}_{n|n}^\dagger (\mathbf{\Lambda}^n + \alpha \mathbf{I}_{N-1})^{-1} \bar{\mathbf{g}}_{n|n} - g_{nn}) + 2\text{Re}(a (\xi_n^* - \bar{\xi}_{|n}^\dagger (\mathbf{\Lambda}^n + \alpha \mathbf{I}_{N-1})^{-1} \bar{\mathbf{g}}_{n|n}))]} \quad (2.24)$$

or, equivalently,

$$\begin{aligned} & \propto e^{\frac{1}{\alpha} [|a|^2 (\mathbf{g}_{n|n}^\dagger \mathbf{Q}_{N-1}^{n\dagger} (\mathbf{\Lambda}^n + \alpha \mathbf{I}_{N-1})^{-1} \mathbf{Q}_{N-1}^n \mathbf{g}_{n|n} - g_{nn}) + 2\text{Re}(a (\xi_n^* - \xi_{|n}^\dagger \mathbf{Q}_{N-1}^{n\dagger} (\mathbf{\Lambda}^n + \alpha \mathbf{I}_{N-1})^{-1} \mathbf{Q}_{N-1}^n \mathbf{g}_{n|n}))]} \end{aligned} \quad (2.25)$$

---

## 2. A Continuation Approach to Iterative Sequence Estimation

---

Note that, for  $\mathcal{X} = \{-1, +1\}$ , the expression simplifies as

$$\mathbb{E}_{\mathbf{s}|\mathbf{x}'}[s_n] = \tanh \left\{ \frac{2}{\alpha} \text{Re}(\xi_n^* - \bar{\xi}_{|n}^\dagger (\Lambda^n + \alpha \mathbf{I}_{N-1})^{-1} \bar{\mathbf{g}}_{n|n}) \right\} \quad (2.26)$$

or

$$\mathbb{E}_{\mathbf{s}|\mathbf{x}'}[s_n] = \tanh \left\{ \frac{2}{\alpha} \text{Re}(\xi_n^* - \xi_{|n}^\dagger \mathbf{Q}_{N-1}^{n\dagger} (\Lambda^n + \alpha \mathbf{I}_{N-1})^{-1} \mathbf{Q}_{N-1}^n \mathbf{g}_{n|n}) \right\} \quad (2.27)$$

As we see it has the structure of a soft decision feedback converging to a hard limiter as  $\alpha \rightarrow 0^+$ . This is equivalent to the Wiener interpolator for the MMSE estimation of the  $n^{\text{th}}$  sample of the channel response from the  $N - 1$  others in the fictitious noise of variance  $\alpha$ . Hence, the term in between parentheses essentially consists of a refined MMSE interference canceler. For fixed  $\alpha$ , the  $N$  impulse responses  $\xi_{|n}^\dagger$  are deduced by cyclic permutation of the components of one (e.g., the first) of them.

### 2.4.1 Choice of An Appropriate Basis.

Numerical simulations show that for large  $N$  and for  $M \ll N$ , the truncated matrices  $\mathbf{G}_{|n}$  can be quasi-diagonalized in the  $(N - 1)$ -dimensional Fourier basis  $\mathbf{F}_{N-1}$ . This statement justifies the second approximation

$$\mathbf{G}_{|n} \approx \mathbf{F}_{N-1}^\dagger \Lambda^n \mathbf{F}_{N-1} \quad (2.28)$$

i.e.,  $\mathbf{Q}_{N-1}^n \approx \mathbf{F}_{N-1}$ ,  $\forall n$  (stationarity assumption).

## 2.5 Convergence and Stability Issues

This section aims at discussing various convergence and stability issues of the proposed algorithm. For this, it is necessary to develop a complete formula of the joint probability.

### 2.5.1 Convergence of the Annealing Process.

Recall that  $p^{(\alpha)}(\mathbf{x})$  is a family of normal distribution converging to

$$\sum_{\mathbf{s}} p(\mathbf{s}) \delta(\mathbf{x} - \mathbf{s}) \quad (2.29)$$

## 2.5. Convergence and Stability Issues

---

As  $\alpha \rightarrow 0^+$  where  $\mathbf{s}$  runs over  $\mathcal{X}^N$ . More explicitly, for  $\mathbf{x} \in \mathbb{C}^N$ ,

$$\begin{aligned} p^{(\alpha)}(\mathbf{x}, \mathbf{y}) &= p(\mathbf{y}|\mathbf{x})p^{(\alpha)}(\mathbf{x}) \\ &= (\pi\beta)^{-N} e^{-\frac{1}{\beta}\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2} \sum_{\mathbf{s}} p(\mathbf{s})(\pi\alpha)^{-N} e^{-\frac{1}{\alpha}\|\mathbf{H}(\mathbf{x}-\mathbf{s})\|^2} \end{aligned} \quad (2.30)$$

After some algebra, we come up to the equivalent expression

$$p^{(\alpha)}(\mathbf{x}, \mathbf{y}) = \sum_{\mathbf{s}} Q^\alpha(\mathbf{s})(\pi\beta\gamma)^{-N} e^{-\frac{1}{\beta\gamma}\|\mathbf{H}\mathbf{x}-J^{(\alpha)}(\mathbf{s})\|^2} \quad (2.31)$$

in which

$$Q^{(\alpha)}(\mathbf{s}) = p(\mathbf{s})(\pi(\alpha + \beta))^{-N} e^{-\frac{1}{(\alpha+\beta)}\|\mathbf{y}-\mathbf{H}\mathbf{s}\|^2} \quad (2.32)$$

and

$$J^{(\alpha)}(\mathbf{s}) = \gamma\mathbf{y} + (1 - \gamma)\mathbf{H}\mathbf{s} \quad (2.33)$$

In order to analyze the convergence of the annealing process towards the ML estimate  $\mathbf{x}^*$ , we assume a continuous decreasing of the temperature  $\alpha$ . From the last expressions, it is obvious that as  $\alpha \rightarrow \infty, \gamma \rightarrow 1^-$ ,  $p^{(\alpha)}(\cdot, \mathbf{y})$  has a unique maximum at  $\mathbf{x} = \mathbf{x}_\infty = \mathbf{y}$ . Conversely, as  $\alpha \rightarrow 0^+, \gamma \rightarrow 0^+$ ,  $p^{(\alpha)}(\cdot, \mathbf{y})$  has  $|\mathcal{X}|^N$  maxima located at the vertices of the lattice  $\mathcal{X}^N$ , with a global maximum arbitrarily close to the ML estimate  $\mathbf{x} = \mathbf{x}^* = \arg \max_{\mathbf{s}} p(\mathbf{s})(\pi\beta)^{-N} e^{-\frac{1}{\beta}\|\mathbf{y}-\mathbf{H}\mathbf{s}\|^2}$ . So the density  $p^{(\alpha)}(\mathbf{x}, \mathbf{y})$  behaves correctly at the limits. This also implies the existence of a decreasing sequence of singular temperatures  $\infty > \bar{\alpha}_0 > \bar{\alpha}_1 > \dots$  such that the number of maxima is constant in each open interval  $\mathbb{I}_i = (\bar{\alpha}_{i+1}, \bar{\alpha}_i)$  and undergoes variation (phase transition or bifurcation) between two successive intervals. The following claim concerns the behavior of the annealing process in the regular domain of temperatures  $\mathbb{A} = \bigcup_i \mathbb{I}_i$ .

**Proposition 1.** Let  $\mathbf{x}^{(\alpha_0)*} = \arg \max_{\mathbf{x}} p^{(\alpha_0)}(\mathbf{x}, \mathbf{y})$  for any  $\alpha_0 \in \mathbb{I}_{i \geq 0}$ . Let  $\alpha \mapsto \mathbf{x}^*(\alpha)$  be a continuous curve such that  $\mathbf{x}^*(\alpha = \alpha_0) = \mathbf{x}^{(\alpha_0)*}$  and such that  $\mathbf{x}^*(\alpha)$  is the solution of the equation  $\nabla_{\mathbf{x}} p^{(\alpha)}(\mathbf{x}, \mathbf{y}) = 0$  for all  $\bar{\alpha}_{i+1} < \alpha < \alpha_0$ . Then, for almost all realization of  $\mathbf{y}$ ,  $\mathbf{x}^*(\alpha) = \arg \max_{\mathbf{x}} p^{(\alpha)}(\mathbf{x}, \mathbf{y})$  on the interval  $(\bar{\alpha}_{i+1}, \alpha_0]$ .

**Proof.** We prove that the set of values of  $\mathbf{y}$  for which the proposition is not true is exceptional in some sense. In fact, the left member of the equation  $\nabla_{\mathbf{x}} p^{(\alpha)}(\mathbf{x}, \mathbf{y}) = 0$  is continuously differentiable w.r.t.  $\alpha, \mathbf{x}$  and  $\mathbf{y}$ , so that, by the classical implicit

---

## 2. A Continuation Approach to Iterative Sequence Estimation

---

function theorem, the curves  $\alpha \mapsto \mathbf{x}^*(\alpha, \mathbf{y})$  and  $\alpha \mapsto p^{(\alpha)}(\mathbf{x}^*(\alpha, \mathbf{y}), \mathbf{y})$ . Now, assume that the proposition fails for some  $\mathbf{y}$ , *i.e.*, there exists  $\alpha_1 < \alpha_0$  such that  $\mathbf{x}^*(\alpha_1, \mathbf{y})$  is no longer the global maximum. By a continuity argument, this implies the existence of a temperature  $\alpha$  in the considered range for which there are two distinct local maxima  $\mathbf{x}_1^*(\alpha, \mathbf{y})$  and  $\mathbf{x}_2^*(\alpha, \mathbf{y})$  with equal amplitudes. Observation vector  $\mathbf{y}$  would then belong to the (possibly empty) intersection  $\mathfrak{S}$  of  $2N + 1$  differentiable hyper surfaces of  $\mathbb{C}^N$  determined by the independent implicit equations:

$$\begin{aligned} \nabla_{\mathbf{x}} p^{(\alpha)}(\mathbf{x}_1^*(\alpha, \mathbf{y}), \mathbf{y}) &= 0 \\ \nabla_{\mathbf{x}} p^{(\alpha)}(\mathbf{x}_2^*(\alpha, \mathbf{y}), \mathbf{y}) &= 0 \\ p^{(\alpha)}(\mathbf{x}_1^*(\alpha, \mathbf{y}), \mathbf{y}) - p^{(\alpha)}(\mathbf{x}_2^*(\alpha, \mathbf{y}), \mathbf{y}) &= 0 \end{aligned} \tag{2.34}$$

where  $\mathfrak{S}$  is a nowhere dense subset of  $\mathbb{C}^N$  and the claim is proved. ■

Roughly, this proposition means that if  $\mathbf{x}^*(\alpha)$  is initialized on the locus of the global maximum for some value of the temperature and locked on the displacement of this maximum as  $\alpha$  decreases, no accident can occur at least between two successive singular temperatures, *i.e.*, the possibility that the tracked maximum becomes sub-optimum for some subsequent  $\alpha$  is ruled out. Note again that this proposition is valid for an idealized process where the temperature is continuously decreased. Since (for obvious complexity issues) the proposed iterative algorithm only considers a finite number of drops in temperature, there exists a possibility to loose track of the global maximum if the decrements are not small enough.

### 2.5.2 Convergence of the Iterative Algorithm.

We now focus on  $\nabla_{\mathbf{x}} p^{(\alpha)}(\mathbf{x}, \mathbf{y}) = 0$ . Taking partial derivative of (2.31) yields the fixed-point equation

$$\mathbf{x} = \varphi(\mathbf{x}) = \gamma(\mathbf{H}^\dagger \mathbf{H})^{-1} \mathbf{H}^\dagger \mathbf{y} + (1 - \gamma) \bar{\mathbf{s}}(\mathbf{x}) \tag{2.35}$$

in which

$$\bar{\mathbf{s}}(\mathbf{x}) = \sum_{\mathbf{s}} \mathbf{s} \varpi(\mathbf{s}) \tag{2.36}$$

and

$$\varpi(\mathbf{s}) = \frac{p(\mathbf{s}) e^{-\frac{1}{\alpha} \|\mathbf{H}(\mathbf{x} - \mathbf{s})\|^2}}{\sum_{\mathbf{s}'} p(\mathbf{s}') e^{-\frac{1}{\alpha} \|\mathbf{H}(\mathbf{x} - \mathbf{s}')\|^2}} \tag{2.37}$$

## 2.5. Convergence and Stability Issues

---

**Proposition 2.** The fixed-point equation (2.35) has a solution for all  $\alpha > 0$ .

**Proof.** For every  $\mathbf{x}$ ,  $\bar{\mathbf{s}}(\mathbf{x})$  is in the lattice  $\mathcal{X}^N$  and  $\varphi(\mathbf{x}) \in \mathbb{K}$ , where  $\mathbb{K}$  is the closed cone with vertex  $(\mathbf{H}^\dagger \mathbf{H})^{-1} \mathbf{H}^\dagger \mathbf{y}$  and basis  $\mathcal{X}^N$ . Restricted to  $\mathbb{K}$ ,  $\varphi$  is a continuous mapping of a closed convex of  $\mathbb{R}^N$  into itself. The conclusion directly follows from the Bohl-Brouwer theorem in topology which asserts that such a mapping has a fixed point inside  $\mathbb{K}$ . ■

Remember that the iterative algorithm proceeds by alternation of two phases, namely gradient search of the max of  $\ln p^{(\alpha)}(\cdot, \mathbf{y})$  and cooling. The rate of the convergence is conditioned by the Hessian of  $\bar{\mathbf{s}}(\mathbf{x})$ . Indeed, with fixed  $\alpha$ , we obtain from (2.35) and the mean value theorem

$$\|\mathbf{x}_{p+1} - \mathbf{x}_p\| = (1 - \gamma) \|\bar{\mathbf{s}}(\mathbf{x}_p) - \bar{\mathbf{s}}(\mathbf{x}_{p-1})\| = (1 - \gamma) \|\Phi(\bar{\mathbf{x}})(\mathbf{x}_p - \mathbf{x}_{p-1})\| \quad (2.38)$$

Here,  $\bar{\mathbf{x}}$  is a point on the segment joining  $\mathbf{x}_{p-1}$  and  $\mathbf{x}_p$  (at iterations  $p - 1$  and  $p$ ) and  $\Phi$  is the Hessian of  $\bar{\mathbf{s}}(\mathbf{x})$  expressed as

$$\Phi(\mathbf{x}) = \frac{1}{\alpha} \Lambda(\mathbf{x}) \mathbf{H}^\dagger \mathbf{H} \quad (2.39)$$

where

$$\Lambda(\mathbf{x}) = \sum_{\mathbf{s}} \varpi(\mathbf{s}) \mathbf{s} \mathbf{s}^\dagger - \left\{ \sum_{\mathbf{s}} \varpi(\mathbf{s}) \mathbf{s} \right\} \left\{ \sum_{\mathbf{s}} \varpi(\mathbf{s}) \mathbf{s} \right\}^\dagger \quad (2.40)$$

has the structure of a covariance matrix. For the iterations to be convergent, one must ensure that the spectrum of  $\Phi$  is strictly bounded by unity.

**Proposition 3.** The spectrum of  $\Phi$  is strictly bounded by unity for the intermediate range of temperatures.

**Proof.** Define the moment-generating (partition) function of the vector variable  $\lambda$  as

$$Z(\lambda) = \sum_{\mathbf{s}} e^{-\frac{1}{\alpha} \|\mathbf{H}(\mathbf{x} - \mathbf{s})\|^2 + \lambda^\dagger \mathbf{s}} \quad (2.41)$$

and observe that  $\Lambda(\mathbf{x}) = \nabla_\lambda^2 \ln Z(\lambda)|_{\lambda=0}$ . Since  $\mathbf{s}$  is zero-mean with unit covariance (i.i.d. assumption) and since  $\mathbf{H}^\dagger \mathbf{H}$  obeys the spectral decomposition  $\mathbf{Q}^\dagger \mathbf{D} \mathbf{Q}$  where

---

## 2. A Continuation Approach to Iterative Sequence Estimation

---

$\mathbf{D}$  is diagonal and  $\mathbf{Q}$  is unitary,  $\zeta = \mathbf{Q}\mathbf{s}$  is asymptotically ( $N \rightarrow \infty$ ) normal, with zero mean and unit covariance and the discrete sum  $Z(\lambda)$  can be approximated by the integral

$$\mathbf{Z}(\lambda) \propto \int_{\mathbb{C}^N} e^{-\|\zeta\|^2} e^{-\frac{1}{\alpha}\|\zeta - \mathbf{Q}\mathbf{x}\|_D^2 + 2Re(\lambda^\dagger \mathbf{Q}^\dagger \zeta)} \mathbf{dV}(\zeta) \quad (2.42)$$

where  $\propto$  means equality up to a multiplicative constant in  $\lambda$ . Let us introduce  $\Delta = \mathbf{I}_N + \alpha^{-1}\mathbf{D}$  and  $\eta = \mathbf{Q}(\lambda + \alpha^{-1}\mathbf{D}\mathbf{x})$ . The total exponent in the integrand can be cast as

$$-\|\zeta - \Delta^{-1}\eta\|^2 + \|\eta\|^2 \quad (2.43)$$

After integrating and developing the last term in  $\eta$ , it is found that the second order term in  $\lambda$  is  $\|\mathbf{Q}\lambda\|_{\Delta^{-1}}^2$ . Hence,

$$\nabla_\lambda^2 \ln Z(\lambda)|_{\lambda=0} = \alpha(\mathbf{H}^\dagger \mathbf{H} + \alpha \mathbf{I}_N)^{-1} \quad (2.44)$$

and, from (2.39), we get

$$\Phi(\mathbf{x}) = (\mathbf{H}^\dagger \mathbf{H} + \alpha \mathbf{I}_N)^{-1} \mathbf{H}^\dagger \mathbf{H} \quad (2.45)$$

which is clearly strictly inferior to unity. This concludes the proof. ■

This result does not hold for very high and very low  $\alpha$  because of the non validity of the central limit theorem. Nevertheless, the next proposition allows to get a global picture of the convergence rate in all ranges. Denote  $\lambda_{\max}$  the largest eigenvalue of  $\Lambda(\mathbf{x})$ . Due to the dissipative character of the channel, the spectrum of  $\Phi$  is bounded by  $\lambda_{\max}/\alpha$ .

**Proposition 4.** For almost all  $\mathbf{x}$ , the following limits  $\lim_{\alpha \rightarrow 0} \frac{\lambda_{\max}}{\alpha} = \lim_{\alpha \rightarrow \infty} \frac{\lambda_{\max}}{\alpha} = 0$  hold with exponential rate.

**Proof.** From (2.39), we have

$$\lambda_{\max} < tr(\Lambda(\mathbf{x})) = \sum_{\mathbf{s}} \varpi(\mathbf{s}) \|\mathbf{s}\|^2 - \left\| \sum_{\mathbf{s}} \varpi(\mathbf{s}) \mathbf{s} \right\|^2 = N - \|\bar{\mathbf{s}}(\mathbf{x})\|^2 \quad (2.46)$$

Organizing the sequences  $\mathbf{s}$  in order of decreasing  $\varpi$  as

$$e^{-\frac{1}{\alpha}\|\mathbf{H}(\mathbf{x}-\mathbf{s}_l)\|^2} \geq e^{-\frac{1}{\alpha}\|\mathbf{H}(\mathbf{x}-\mathbf{s}_{l+1})\|^2} \quad : l = 0, \dots, |\mathcal{X}|^N - 1 \quad (2.47)$$


---

## 2.6. Numerical Results

---

yields the following inequality

$$\|\bar{\mathbf{s}}(\mathbf{x})\|^2 = \left\| \sum_{l \geq 0} \varpi_l \mathbf{s}_l \right\|^2 \geq N\varpi_0^2 + 2\varpi_0 \mathbf{s}_0^\dagger \sum_{l \geq 1} \varpi_l \mathbf{s}_l \quad (2.48)$$

Using the fact that

$$\mathbf{s}_0^\dagger \mathbf{s}_l \geq -\|\mathbf{s}_0\|^2 = -N \quad (2.49)$$

the trace is upper bounded as

$$\text{tr}(\Lambda(\mathbf{x})) \leq N(1 + 2\varpi_0 - 3\varpi_0^2) \quad (2.50)$$

The RHS of this inequality is convex upward and bounded by its tangent at the abscissa  $\varpi_0 = 1$ . This yields

$$\lambda_{\max} < \text{tr}(\Lambda(\mathbf{x})) \leq 4N(1 - \varpi_0) \quad (2.51)$$

We now introduce the increasing sequence of  $\mathbf{r}_l(\mathbf{x})$  defined as

$$\mathbf{r}_l(\mathbf{x}) = \|\mathbf{H}(\mathbf{x} - \mathbf{s}_l)\|^2 - \|\mathbf{H}(\mathbf{x} - \mathbf{s}_0)\|^2 : l = 0, \dots, |\mathcal{X}|^N - 1 \quad (2.52)$$

Bound above and below by linear sequences such that

$$lr_b(\mathbf{x}) \leq \mathbf{r}_l(\mathbf{x}) \leq lr_a(\mathbf{x}) : l = 0, \dots, |\mathcal{X}|^N - 1 \quad (2.53)$$

Up to an exceptional (zero-measure) set of values of  $\mathbf{x}$ ,  $r_b(\mathbf{x}) > 0$ , so that

$$\begin{aligned} \frac{\lambda_{\max}}{\alpha} &< \frac{4N \sum_{l \geq 1} e^{-\frac{r_l(\mathbf{x})}{\alpha}}}{\alpha \sum_{l \geq 0} e^{-\frac{r_l(\mathbf{x})}{\alpha}}} \\ &\leq \frac{4N \sum_{l \geq 1} e^{-\frac{lr_b(\mathbf{x})}{\alpha}}}{\alpha \sum_{l \geq 0} e^{-\frac{lr_a(\mathbf{x})}{\alpha}}} \\ &\leq \frac{4N}{\alpha} \frac{e^{-\frac{r_b(\mathbf{x})}{\alpha}}}{1 - e^{-\frac{r_b(\mathbf{x})}{\alpha}}} \frac{1 - e^{-\frac{r_a(\mathbf{x})}{\alpha}}}{1 - e^{-\frac{2^N r_a(\mathbf{x})}{\alpha}}} \end{aligned} \quad (2.54)$$

This last bound concludes the proof. ■

## 2.6 Numerical Results

We implemented the algorithm (2.10)-(2.12) using an exact sum-product procedure or the Gaussian approximation (2.27) for the evaluation of the sum-over-states.



---

## 2. A Continuation Approach to Iterative Sequence Estimation

---

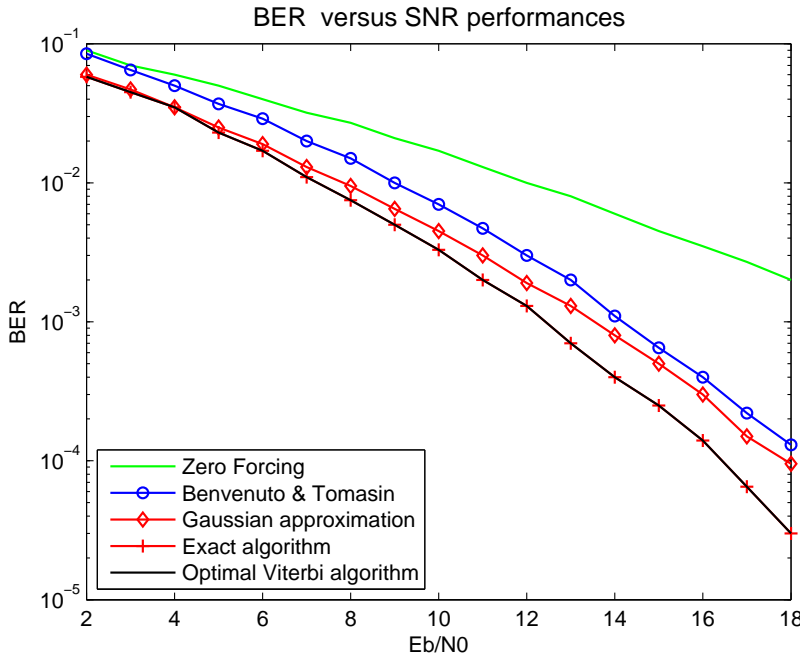


Figure 2.1: BER versus SNR performance.

### 2.6.1 Performance on a Stanford University Interim (SUI) Channel

First, numerical simulations were carried out with i.i.d. BPSK data symbols and a SUI channel with 3 taps with normalized powers 0 dB, -12 dB, -15 dB and delays  $0\mu s, 0.4\mu s, 1.1\mu s$  [76]. For each value of SNR, convergence was attained after 8 iterations with the following values of the normalized temperature  $\gamma = \{0.9999; 0.6990; 0.3979; 0.3010; 0.2218; 0.1549; 0.0969; 0.00458\}$ . A unique pass of EM (2.10)-(2.12) was needed for each iteration (or  $\gamma$  value). We also implemented the turbo equalizer proposed in [77] over the same channel for comparison purpose. The BER versus SNR performance of all those algorithms is depicted in Fig 2.1. We observe that the proposed algorithm (even in its simplified form, i.e., GA) performs very close to the optimal Viterbi algorithm (especially in the low SNR region) and outperforms the algorithm in [77] by approximately 1 dB.

## 2.6. Numerical Results

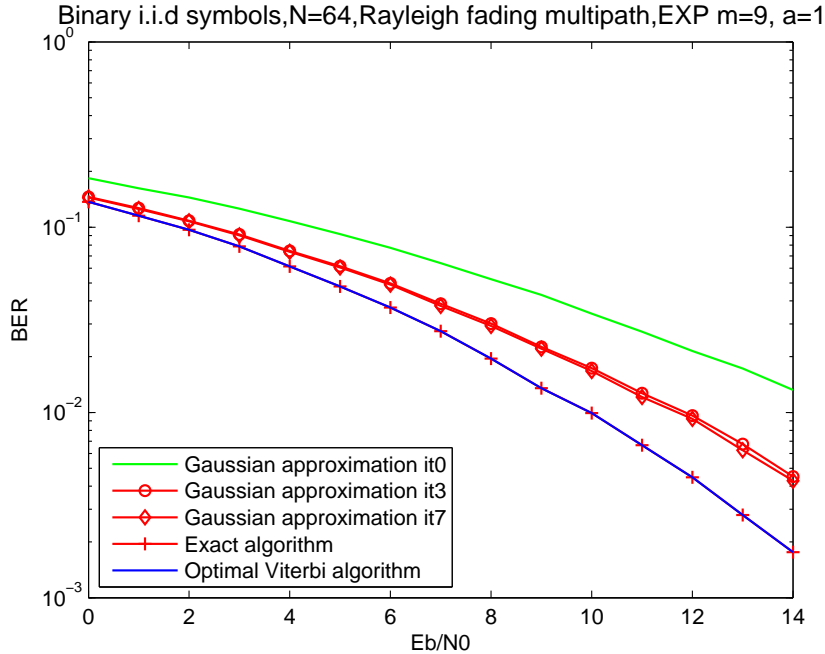


Figure 2.2: BER versus SNR performance (Binary i.i.d symbols,  $N = 64$ , Rayleigh fading multipath, EXP  $M = 9$ ,  $a = 1$ ).

### 2.6.2 Performance on a Rayleigh Fading Channel (Exponential)

Second, numerical simulations were carried out with i.i.d. BPSK data symbols and a Rayleigh fading channel, EXP, with channel memory equal  $M = 9$ . The standard deviation of each tap  $h[i]$  is equal to  $\sqrt{\frac{e^{-2ai}}{\sum_{l=0}^M e^{-2al}}}$ . For each value of SNR, convergence was attained after 8 iterations with the following values of the normalized temperature  $\gamma = \{0.9999; 0.6990; 0.3979; 0.3010; 0.2218; 0.1549; 0.0969; 0.00458\}$ . A unique pass of EM (2.10)-(2.12) was needed for each iteration (or  $\gamma$  value). The BER versus SNR performance of all those algorithms is depicted in Fig 2.2 and Fig 2.3. We observe that the proposed algorithm (even in its simplified form, i.e., GA) performs very close to the optimal Viterbi algorithm (especially in the low SNR region).

---

## 2. A Continuation Approach to Iterative Sequence Estimation

---

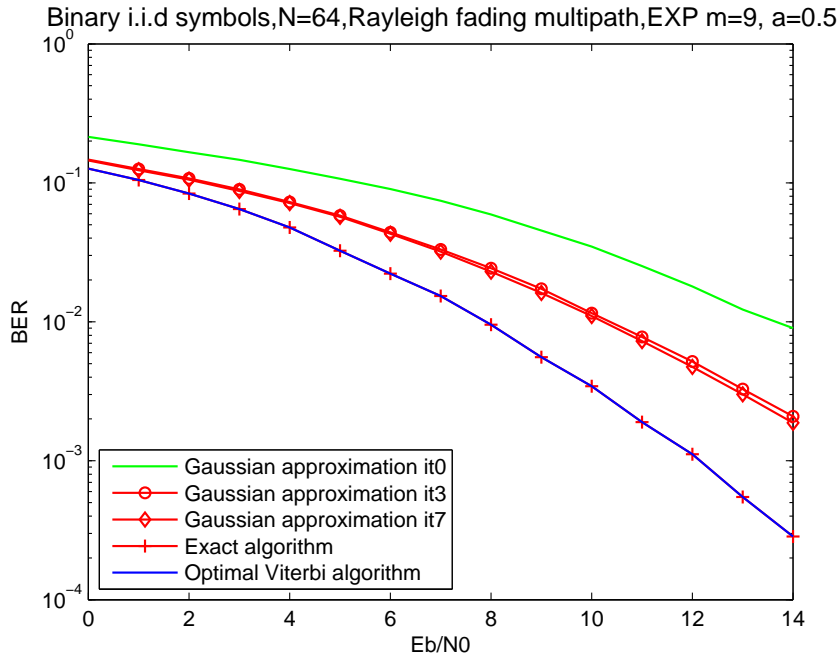


Figure 2.3: BER versus SNR performance (Binary i.i.d symbols,  $N = 64$ , Rayleigh fading multipath, EXP  $M = 9$ ,  $a = 0.5$ ).

## 2.7 Conclusion

In this chapter, a novel approach was presented to solve the problem of MLSE of discrete symbols from the noisy output of a convolutional channel. One feature of our method is its theoretical attractiveness, namely the fact that its iterative structure naturally emerges from the successive continuous approximations of the targeted exact discrete APP on symbols and thus comes without any ad hoc assumption on its final form by opposition to [73] [77]. We have shown that, under certain circumstances, an approximated version of the algorithm yields the form of a block-iterative MMSE interference canceler with soft decision feedback. Resorting to computation in the Fourier domain turns out to be crucial for lowering the complexity. Given its intrinsically probabilistic nature and its close to optimal performance at low SNR, we believe that the proposed equalizer is perfectly suitable for block turbo-equalization of coded symbols.

# Chapter 3

## Estimation of Linear Dynamic Channels based on Kalman Filtering

### 3.1 Introduction

Modern portable and mobile digital communication systems require reliable signaling methods over multipath fading channels in the presence of ISI and additive white Gaussian noise (AWGN). To defeat the signal distortion caused by the time varying characteristics of the channel, different classes of equalization techniques have been suggested in the literature for different practical situation. The mobile digital wireless channel needs high performance equalizers, because of different challenging facts such as the mobility of the transmitter and receiver with respect to each other, and the multipath nature of propagation environment. Usually we divide equalization in two classes; linear and nonlinear. In linear equalization using structures such as transversal and lattice equalizers the received signal is delayed and weighted by equalizer coefficients and summed to produce the output. Nonlinear equalizers are able to combat sever channel distortion which cannot be handled by linear equalizers.

For channel estimation a training sequence is often transmitted and the detected symbols are compared to the known training sequence. The result is used to estimate the unknown channel coefficients and obtain the tap weights, then the channel is continually tracked in a decision-directed mode [5]. The decisions should be highly

---

### **3. Estimation of Linear Dynamic Channels based on Kalman Filtering**

reliable, otherwise a string of decision errors might happen during a deep fade and cause the detector to fail. For the adaption of channel parameters, different estimation algorithms such as the Kalman filter, the RLS algorithm, and the LMS algorithm can be used. The Kalman filter is the optimum estimation algorithm and can be used for the tracking of wireless fading channels [54, 55, 57]. Sub optimal estimation techniques such as least mean square (LMS) algorithm [58], or recursive least square (RLS) algorithm [56] in many practical systems are chosen instead of the Kalman filter to reduce the implementation costs.

MLSE as an optimal receiver which was first proposed by Forney [10] for ISI channels can be practically implemented by Viterbi algorithm based on a known channel impulse response (CIR) [11]. The Viterbi algorithm originally proposed by Viterbi [78] for maximum likelihood decoding of convolutional codes is a special case of forward dynamic programming [79]. The Viterbi algorithm finds optimal trajectories (survivor paths) at each stage for all states in the trellis diagram. This algorithm was used by Forney in the MLSE receiver to detect a digital signal transmitted through the ISI channel corrupted by an additive Gaussian noise [11]. The MLSE algorithm computes the cost (probability of error) through all trajectories from each state to the next stage for all possible states (computing the branch metrics) and then finds the minimum cost trajectory (survivor path) for each state. Therefore at the final stage we have the best survivor path corresponds to the data sequence with minimum probability of sequence error. The MLSE receiver for fading channels whose statistical parameters are known was considered in [80–83]. In order to implement MLSE using the Viterbi algorithm, we need the knowledge of the channel which is practically unknown at the receiver and should be estimated. To solve this problem, joint channel estimation and data detection methods were proposed in [36, 80, 84–86]. Some estimation algorithms such as least mean square (LMS), recursive least square (RLS) and Kalman filtering were used in channel estimation. When the Viterbi algorithm is used because of the inherent decision delay, we have poor channel tracking in a time variant environment.

The idea of per survivor processing (PSP) was proposed to overcome the decision delay problem [65], where each survivor path in the trellis diagram has its own estimator. The accuracy and the convergence properties of LMS determine the overall

## 3.2. System Model

---

performance of the PSP algorithm. The tracking performance of this approach was showed in [87,88]. In the PSP decoder, the LMS, RLS and Kalman Filter algorithms can be used to estimate the channel parameters [89–91]. It will be shown that the state space model parameters can be easily obtained at the receiver by estimating the maximum Doppler frequency shift or equivalently finding the AR spectral estimation of CIR. This enables us to use the optimal Kalman filter for channel estimation. The aim in this chapter is to motivate rigorously the introduction of the Kalman filter in the estimation of Markov sequences through Gaussian dynamical channels. By this we interpret and make clearer the underlying approximations in the heuristic approaches.

This chapter is organized as follows: First the system model is introduced in section 3.2. Then proposed method based on joint Kalman and Viterbi algorithms is presented in section 3.3. The simulation results of the proposed algorithm are presented in section 3.4. Finally, the main features of our approach are summed up in the conclusion.

## 3.2 System Model

We consider block transmission over a single-input single-output (SISO) doubly selective channel (DSC) (block length:  $N$ , channel memory:  $M$ , maximum Doppler frequency shift:  $f_D$ , symbol duration:  $T$ ) and binary phase shift keying (BPSK) modulation, so that the bit is transmitted at instant  $n$ ,  $b_n \in \{-1, +1\}$ .

We assume a discrete Rayleigh fading channel of memory  $M$ , simulated with method introduced in [92], which the elements of the impulse response  $\{c_n^i\}_{i=0}^M$  are modeled as independent zero-mean complex Gaussian random variables with variance  $a_i$ :

$$c_n^i = \sqrt{\frac{a_i}{N_0}} \sum_{q=0}^{N_0-1} \cos(2\pi n f_D T \cos \gamma_{qi} + \phi_{qi}) + j \sin(2\pi n f_D T \sin \gamma_{qi} + \phi'_{qi}) \quad (3.1)$$

where

$$\gamma_{qi} = \frac{2\pi q}{4N_0} + \frac{2\pi i}{4N_0(M+1)} + \frac{\pi}{8N_0(M+1)}$$

### 3. Estimation of Linear Dynamic Channels based on Kalman Filtering

$\phi_{qi}, \phi'_{qi}$ , for  $q = 0, 1, 2, \dots, N_0 - 1$  and  $i = 0, 1, 2, \dots, M$ , are  $2(M + 1)N_0$  independent random phases, each of which is uniformly distributed in  $[0, 2\pi)$ , also we consider  $N_0 > 16$ .

In order to accurately capture the dynamics of the wireless channel, we formulate a channel model suitable for use in the channel tracker. This model must be remaining mathematically tractable for implementation in a discrete-time state-space context. According to fading process which is modelled as a complex Gaussian process, a suitable model is thus an auto-regressive (AR) model. Information theoretic results have shown that a first-order AR model is sufficient to accurately represent the local behavior of the time-varying wireless channel [93]. A higher order model while providing more accurate long-term channel estimates, necessarily requires an AR order of 100 – 200 coefficients [94], and is thus highly intractable for the state model. Using the first-order assumption, we finally realize the state evolution at time  $n$  as:

$$c_n^i = \xi c_{n-1}^i + v_n^i \quad i = 0, \dots, M \quad (3.2)$$

where  $\xi$  is the static AR coefficient and  $v_n^i \sim \mathcal{N}_{\mathbb{C}}(0, \sigma_v^2)$  is the complex driving noise of the model. Therefore in the form of state model we have:

$$\mathbf{c}_n = \mathbf{F}\mathbf{c}_{n-1} + \mathbf{v}_n \quad (3.3)$$

where  $\mathbf{c}_n$  is a vector of length  $M + 1$  which each element is the channel gain at time  $n$ .

$$\mathbf{c}_n = [c_n^0, c_n^1, \dots, c_n^M]^T \quad (3.4)$$

the state transition matrix is given by:

$$\mathbf{F} = \xi \mathbf{I}_{M+1} \quad (3.5)$$

and the process noise vector given by:

$$\mathbf{v}_n = [v_n^0, v_n^1, \dots, v_n^M]^T \quad (3.6)$$

with covariance matrix equal to:

$$\mathbf{Q} = (\sigma_v^2) \mathbf{I}_{M+1} \quad (3.7)$$

Additional advantages of using the AR model for describing the evolution of the channel state include:

### 3.2. System Model

---

1. The model is simple and mathematically tractable.
2. The true channel impulse response tends to revert to zero; the behavior of (3.2) also tends to revert to zero.
3. Like the wireless channel, the AR model is a Markov process. This implies that the pdf for the current estimate is not dependent upon all previous estimates but only on the most recent estimate. Owing to the Markovian property, the AR model greatly simplifies the complexity of the recurrence relations used in our algorithm.

In order to parameterize (3.2), we note from [3] that the autocorrelation of the channel fading process is:

$$E[c_n^i c_{n-k}^{i*}] = a_i J_0(2\pi k f_D T) \quad (3.8)$$

where  $J_0()$  is the zeroth-order Bessel function,  $T$  is symbol duration, and  $f_D$  denotes the Doppler frequency resulting from relative motion between the transmitter and receiver. The Doppler shift itself is given by

$$f_D = \frac{v}{c} f_c \quad (3.9)$$

where  $v$  is the mobile speed,  $c$  is the speed of light, and  $f_c$  is the carrier frequency. Equating (3.2) to the autocorrelation of (3.8) for time  $n = \{0,1\}$ , we respectively have

$$\xi^2 a_i + \sigma_v^2 = a_i \quad (3.10)$$

$$\xi = J_0(2\pi f_D T) \quad (3.11)$$

For example, if the normalized desired fading rate is  $f_D T = 0.01$  (a typical fast fading rate), then  $\xi = 0.9990$ .

If we consider approximate autoregressive model of order one (AR(1)) introduced in [3](pp.74-75), we have:

$$\xi = 2 - \cos(2\pi f_D T) - \sqrt{(2 - \cos(2\pi f_D T))^2 - 1} \quad (3.12)$$

and

$$v_n^{(i)} \simeq \mathcal{N}_c(0, a_i(1 - \xi^2)) \quad (3.13)$$



### 3. Estimation of Linear Dynamic Channels based on Kalman Filtering

---

The received complex noisy observation at instant  $n$  has the following form:

$$y_n = \sum_{i=0}^M c_n^i b_{n-i} + w_n \quad (3.14)$$

where  $w_n$  is the additive noise as zero mean complex random with variance  $\beta$ . Therefore

$$y_n = \mathbf{d}_n^T \mathbf{c}_n + w_n \quad (3.15)$$

where

$$\mathbf{d}_n = [b_n, b_{n-1}, \dots, b_{n-M}]^T$$

Finally our communication system can be described as Linear state space model, whose dynamics are given by:

$$\begin{aligned} \mathbf{c}_n &= \mathbf{F} \mathbf{c}_{n-1} + \mathbf{v}_n \\ y_n &= \mathbf{d}_n^T \mathbf{c}_n + w_n \end{aligned} \quad (3.16)$$

where  $\mathbf{d}_n$  is independent of  $\mathbf{c}_n, w_n, \mathbf{v}_n$ , and  $\mathbf{v}_n, w_n$  are independent normal zero-mean white noises with covariance matrix  $\mathbf{Q}$  and variance  $\beta$ . The set of values of  $\mathbf{d}_n$  is  $\Omega$ .

### 3.3 Proposed Method based on Joint Kalman and Viterbi Algorithm

The aim in this section is to motivate rigorously the introduction of the Kalman filter in the estimation of Markov sequences through Gaussian dynamical channels. By this we interpret and make clearer the underlying approximations in the heuristic approaches of [45, 55, 95].

Start with the problem of the maximum posterior probability estimation of a finite data block  $\mathbf{d}_{1:n+1}$  (in this and the sequel we use Matlab notation for concatenating indexed objects). Note that unlike many contributors, we do not attempt to estimate the current realization of the channel process.

The likelihood function of a sequence  $\mathbf{d}_{1:n+1}$  of data decomposes as follows:

### 3.3. Proposed Method based on Joint Kalman and Viterbi Algorithm

---

$$\begin{aligned}
L_{n+1}(\mathbf{d}_{1:n+1}) &= p(\mathbf{d}_{1:n+1}; y_{1:n+1}) \\
&= p(y_{n+1}|\mathbf{d}_{1:n+1}, y_{1:n})p(\mathbf{d}_{n+1}|\mathbf{d}_n)p(\mathbf{d}_{1:n}; y_{1:n}) \\
&= f_{n+1}(\mathbf{d}_{1:n+1})p(\mathbf{d}_{n+1}|\mathbf{d}_n)L_n(\mathbf{d}_{1:n})
\end{aligned} \tag{3.17}$$

therefore the estimated sequence equals to:

$$\hat{\mathbf{d}}_{1:n+1} = \arg \max_{\mathbf{d}_{1:n+1}} L_{n+1}(\mathbf{d}_{1:n+1})$$

Let us now introduce the channel as follows:

$$\begin{aligned}
f_{n+1}(\mathbf{d}_{1:n+1}) &= \int p(y_{n+1}, \mathbf{c}_{n+1} = \mathbf{x}|\mathbf{d}_{1:n+1}; y_{1:n})d\mathbf{x} \\
&= \int p(y_{n+1}|\mathbf{c}_{n+1} = \mathbf{x}, \mathbf{d}_{n+1})p(\mathbf{c}_{n+1} = \mathbf{x}|\mathbf{d}_{1:n+1}, y_{1:n})d\mathbf{x}
\end{aligned} \tag{3.18}$$

Relations (3.17),(3.18) are all that is needed for building approximate recursive maximization for  $L_n$ .

#### 3.3.1 Normal Density Assumption

First under the assumption that the terms in the integral are normals, we can write:

$$f_{n+1}(\mathbf{d}_{1:n+1}) = \int \mathcal{N}_c(y_{n+1}, \mathbf{d}_{n+1}\mathbf{x}, \beta)\mathcal{N}_c(\mathbf{x}, \hat{\mathbf{c}}_{n+1|n}(\mathbf{d}_{1:n}), \mathbf{P}_{n+1|n}(\mathbf{d}_{1:n}))d\mathbf{x} \tag{3.19}$$

where  $\mathcal{N}_c(u, \mu, R)$  is the complex normal density in  $u$  with mean  $\mu$  and covariance  $R$ , and

$$\begin{aligned}
\hat{\mathbf{c}}_{n+1|n}(\mathbf{d}_{1:n}) &= E[\mathbf{c}_{n+1}|\mathbf{d}_{1:n}, y_{1:n}] \\
\mathbf{P}_{n+1|n}(\mathbf{d}_{1:n}) &= cov[\mathbf{c}_{n+1}|\mathbf{d}_{1:n}, y_{1:n}]
\end{aligned} \tag{3.20}$$

After grouping exponents and factors, standard calculation leads to:

$$\begin{aligned}
f_{n+1}(\mathbf{d}_{1:n+1}) &= [\pi\beta|\Gamma_{n+1}\mathbf{P}_{n+1|n}|]^{-1}exp[\mathbf{q}_{n+1}^\dagger\Gamma_{n+1}^{-1}\mathbf{q}_{n+1} - y_{n+1}^*\beta^{-1}y_{n+1} \\
&\quad - \hat{\mathbf{c}}_{n+1|n}^\dagger\mathbf{P}_{n+1|n}^{-1}\hat{\mathbf{c}}_{n+1|n}]
\end{aligned} \tag{3.21}$$

### 3. Estimation of Linear Dynamic Channels based on Kalman Filtering

---

where

$$\begin{aligned}\mathbf{q}_{n+1} &= \mathbf{d}_{n+1}^* \beta^{-1} y_{n+1} + \mathbf{P}_{n+1|n}^{-1}(\mathbf{d}_{1:n}) \hat{\mathbf{c}}_{n+1|n}(\mathbf{d}_{1:n}) \\ \mathbf{\Gamma}_{n+1} &= \mathbf{P}_{n+1|n}^{-1}(\mathbf{d}_{1:n}) + \mathbf{d}_{n+1}^* \beta^{-1} \mathbf{d}_{n+1}^T\end{aligned}\quad (3.22)$$

#### 3.3.2 Approximate Recursive Maximization

It is obvious from (3.17) that the exact recursive maximization of  $L_{n+1}(\mathbf{d}_{1:n+1})$  is very difficult due to coupling between  $f_{n+1}(\mathbf{d}_{1:n+1})$  and  $L_n(\mathbf{d}_{1:n})$  i.e. the fact that these terms share the entire range of their variables from  $n$  down to 1. In last analysis this coupling reflects the non-Gaussian characters of the joint  $(\mathbf{c}_n, \mathbf{d}_n)$  model. To build Approximate recursive maximization, it is desirable to bound this coupling. Notice that the heuristically derived algorithms in [96] can be interpreted as follows. Let

$$\begin{aligned}\mathbf{d}_{1:n-1}^*(\mathbf{w}) &= \arg \max_{\mathbf{d}_{1:n-1}} L_n(\mathbf{d}_{1:n-1}, \mathbf{d}_n = \mathbf{w}) \\ \mathbf{c}_{n+1|n}^*(\mathbf{w}, \mathbf{d}_{n+1}) &= E[\mathbf{c}_{n+1} | \mathbf{d}_{1:n-1}^*(\mathbf{w}), \mathbf{d}_n = \mathbf{w}, \mathbf{d}_{n+1}, y_{1:n}] \quad (\mathbf{w} \in \Omega)\end{aligned}$$

therefore we can write :

$$p[\mathbf{c}_{n+1} | \mathbf{d}_{1:n-1}, \mathbf{d}_n = \mathbf{w}, \mathbf{d}_{n+1}, y_{1:n}] = \delta(\mathbf{c}_{n+1} - \mathbf{c}_{n+1|n}^*(\mathbf{w}, \mathbf{d}_{n+1}))$$

#### 3.3.3 Evaluation of Dependency

To obtain approximations in a systematic way, it is necessary to evaluate the dependency of  $f_{n+1}$  vs  $\mathbf{d}_{1:n+1}$ . By (3.16), for any  $r \geq 1$  one has

$$\mathbf{c}_{n+1} = \mathbf{F}^r \mathbf{c}_{n-r+1} + \mathbf{v}_{n+1} \quad (3.23)$$

where  $\mathbf{v}_{n+1}$  is (statistically) independent of  $(\mathbf{d}_{1:n-r}, y_{1:n-r})$ .

Now,  $f_{n+1}(\mathbf{d}_{1:n+1}) = p(y_{n+1} | \mathbf{d}_{1:n+1}, y_{1:n})$  as a normal density is entirely characterized by its first and second moments. By (3.16) and (3.23) we thus get:

$$\begin{aligned}E[y_{n+1} | \mathbf{d}_{1:n+1}, y_{1:n}] &= \mathbf{d}_{n+1}^T \mathbf{F}^r E[\mathbf{c}_{n-r+1} | \mathbf{d}_{1:n+1}, y_{1:n}] \\ &\quad + \mathbf{d}_{n+1}^T E[\mathbf{v}_{n+1} | \mathbf{d}_{n-r+1:n+1}, y_{n-r+1:n}]\end{aligned}\quad (3.24)$$

Hence, the dependency of LHS in  $\mathbf{d}_{1:n-r}$  is only through the first term of RHS. For a large class of stable dynamical models the matrices  $\mathbf{F}$  have spectra strictly bounded

### 3.3. Proposed Method based on Joint Kalman and Viterbi Algorithm

by  $0 \leq \lambda_0 < 1$ . Assuming this in the sequel, it is seen that the dependency of LHS on  $\mathbf{d}_{1:n-r}$  decays at least exponentially in  $r$  (provided the sequence  $E[\mathbf{c}_{n-r+1} | \mathbf{d}_{1:n+1}, y_{1:n}]$  remains bounded by a constant in quadratic mean, which is the case by the same assumptions on  $\mathbf{F}$ ). A straightforward calculation shows that the same cohesion holds for the second moment  $E[y_{n+1}y_{n+1}^* | \mathbf{d}_{1:n+1}, y_{1:n}]$ . In conclusion, one can consider that in a sense the coupling between  $f_{n+1}(\mathbf{d}_{1:n+1})$  and  $L_n(\mathbf{d}_{1:n})$  in the range  $\mathbf{d}_{1:n-r}$  decreases at least exponentially in  $r$ . This is what renders possible the introduction of the Kalman estimator in the approximate recursive maximization of  $L_n(\mathbf{d}_{1:n})$ .

#### 3.3.4 Final Form of Approximate Recursive Maximization

Considering that the dependency of  $f_{n+1}$  on  $\mathbf{d}_{1:n-r}$  is weak, for  $\mathbf{d}_{n-r+1:n+1}$  fixed, the locus of the components  $\mathbf{d}_{1:n-r}$  of the maximum of  $L_{n+1}(\mathbf{d}_{1:n+1})$  is not much affected by  $f_{n+1}$ . More precisely, for  $\mathbf{w}_{1:r} \in \Omega^r$  and any  $n$  define:

$$\bar{L}_n(\mathbf{w}_{1:r}) = \max_{\mathbf{d}_{1:n-r}} L_n(\mathbf{d}_{1:n-r}, \mathbf{d}_{n-r+1:n} = \mathbf{w}_{1:r}) \quad (3.25)$$

and

$$\mathbf{d}_{1:n-r}^*(\mathbf{w}_{1:r}) = \arg \max_{\mathbf{d}_{1:n-r}} L_n(\mathbf{d}_{1:n-r}, \mathbf{d}_{n-r+1} = \mathbf{w}_1, \dots, \mathbf{d}_n = \mathbf{w}_r) \quad (3.26)$$

then

$$\begin{aligned} \max_{\mathbf{d}_{1:n+1}} L_{n+1}(\mathbf{d}_{1:n+1}) &= \max_{\mathbf{w}_{2:r+1}} \bar{L}_{n+1}(\mathbf{w}_{2:r+1}) \\ &= \max_{\mathbf{w}_{2:r+1}} \max_{\mathbf{w}_1} \max_{\mathbf{d}_{1:n-r}} [f_{n+1}(\mathbf{d}_{1:n-r}, \mathbf{d}_{n-r+1} = \mathbf{w}_1, \mathbf{d}_{n-r+2:n} = \mathbf{w}_{2:r+1}) p(\mathbf{w}_{r+1} | \mathbf{w}_r) \\ &\quad L_n(\mathbf{d}_{1:n-r}, \mathbf{d}_{n-r+1:n} = \mathbf{w}_{1:r})] \end{aligned} \quad (3.27)$$

By our approximation:

$$f_{n+1}(\mathbf{d}_{1:n-r}, \mathbf{w}_{1:r+1}) \simeq f_{n+1}(\mathbf{d}_{1:n-r}^*(\mathbf{w}_{1:r}), \mathbf{w}_{1:r+1})$$

so that we set the following recursion:

$$\bar{L}_{n+1}(\mathbf{w}_{2:r+1}) \simeq \max_{\mathbf{w}_1} [f_{n+1}(\mathbf{d}_{1:n-r}^*(\mathbf{w}_{1:r}), \mathbf{w}_{1:r+1}) \bar{L}_n(\mathbf{w}_{1:r})] \quad (3.28)$$

The definitions (3.25),(3.26) and relations (3.27),(3.28) are the dynamic programming solution to our approximate recursive likelihood maximization problem.

### 3. Estimation of Linear Dynamic Channels based on Kalman Filtering

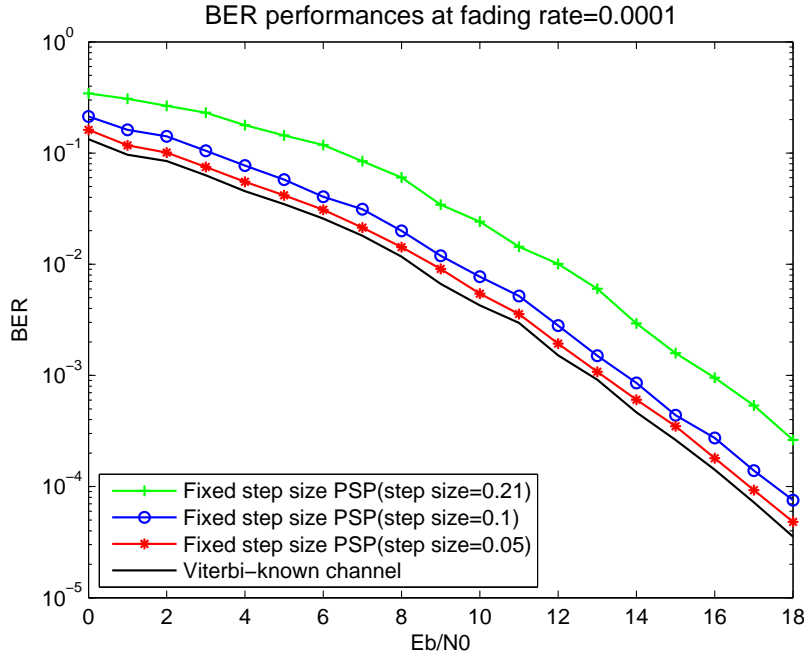


Figure 3.1: BER performance of fixed step size PSP at fading rate  $f_D T = 0.0001$ .

Together with (3.28) the recursion updates are performed by (3.20), (3.21) and (3.22) where  $\mathbf{d}_{1:n-r}$  are forced to  $\mathbf{d}_{1:n-r}^*(\mathbf{w}_{1:r})$  and quantities  $\hat{\mathbf{c}}_{n+1|n}(\mathbf{d}_{1:n-r}^*(\mathbf{w}_{1:r}), \mathbf{w}_{1:r})$  and  $\mathbf{P}_{n+1|n}(\mathbf{d}_{1:n-r}^*(\mathbf{w}_{1:r}), \mathbf{w}_{1:r})$ ,  $\mathbf{w}_{1:r} \in \Omega^r$  are generated at each step by  $|\Omega|^r$  Kalman recursion equation [55]. Detailed algorithm for  $r = 1$  are given at Appendix B.

## 3.4 Numerical Results

Regarding the wireless channel model, we consider a memory-2 Rayleigh fading channel simulated with the method introduced in [92]. The standard deviations of the resulting three complex processes  $[c_n^0, c_n^1, c_n^2]$  are set at (0.407, 0.815, 0.407). we assume that the initial channel  $\mathbf{c}_0 = [c_0^0, c_0^1, c_0^2]^T$  is known (with using pilot symbols) at the receiver and the block length is  $N = 100$ .

### 3.4.1 Fixed Step Size PSP

The BER performance for different fading rate is depicted in Fig 3.1, 3.2, 3.3. At fading rate  $f_D T = 0.0001$ , the optimal step equal to  $\mu = 0.05$ , at the other hand the optimal step size is  $\mu = 0.1$  and  $\mu = 0.21$  for fading rate  $f_D T = 0.001$  and

### 3.4. Numerical Results

---

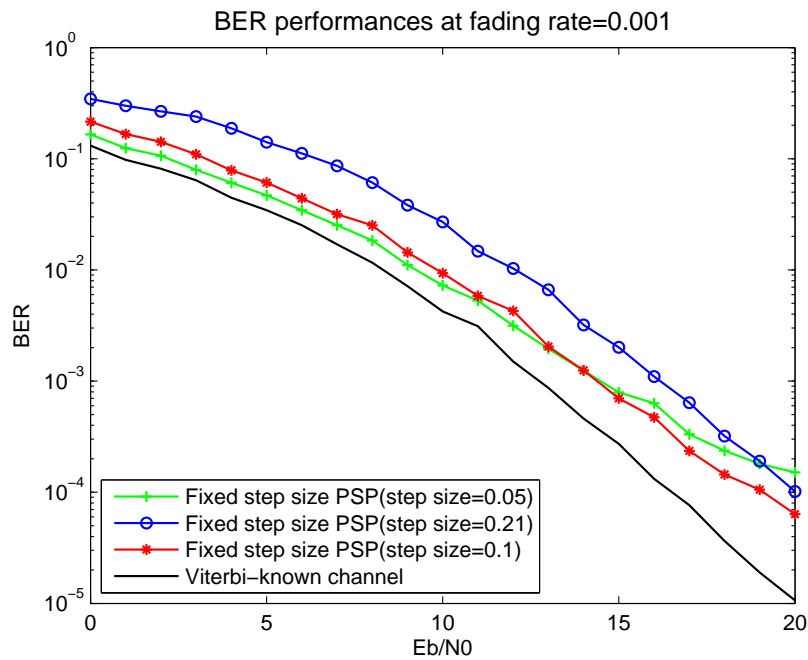


Figure 3.2: BER performance of fixed step size PSP at fading rate  $f_D T = 0.001$ .

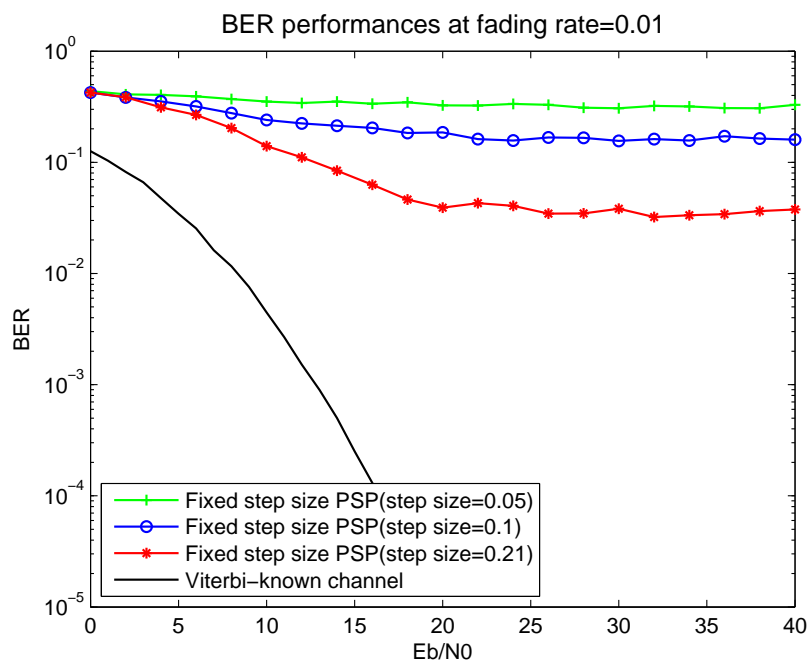


Figure 3.3: BER performance of fixed step size PSP at fading rate  $f_D T = 0.01$ .

### 3. Estimation of Linear Dynamic Channels based on Kalman Filtering

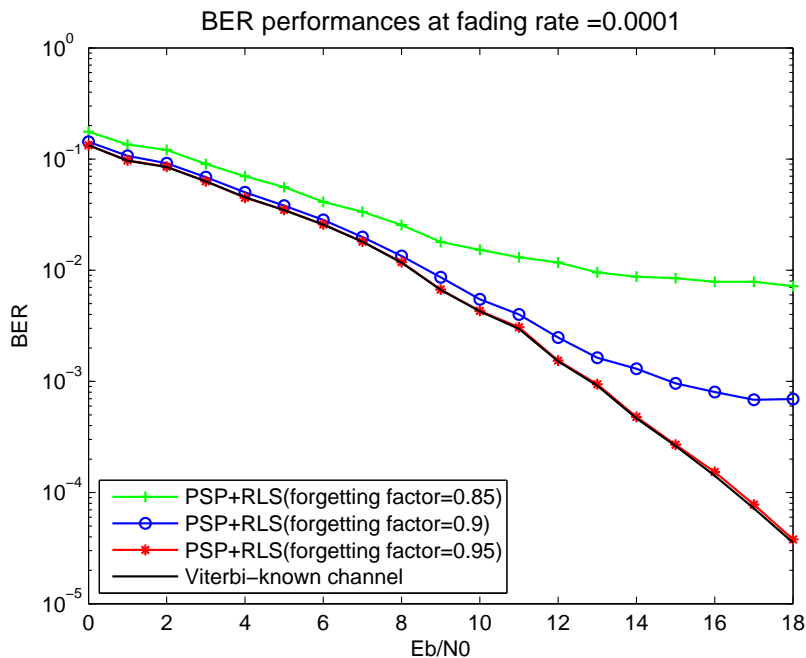


Figure 3.4: BER performance of PSP via RLS algorithm at fading rate  $f_D T = 0.0001$ .

$f_D T = 0.01$  respectively. Therefore, fixed step-size PSP is depend on step size ( $\mu$ ) and not equal for different fading rate. This factor regulated the speed and stability of the tracking method.

#### 3.4.2 PSP via RLS Algorithm

As mentioned before, the RLS algorithm is a least square method, and the forgetting factor very important in this method. Fig 3.4, shows the bit error rate performance for fading rate  $f_D T = 0.0001$ , while Fig 3.5 and Fig 3.6, shows the bit error rate performance for fading rate  $f_D T = 0.001$  and  $f_D T = 0.01$  respectively. As has been shown, the optimal forgetting factor is depend on fading rate.

#### 3.4.3 Proposed Method

The BER performance of our proposed method scheme for  $r = 1$  and different fading rate is shown in Fig 3.7,3.8. We observe that the proposed algorithm (for slow fading rate) performs near to the optimal Viterbi algorithm . We compare our

### 3.4. Numerical Results

---

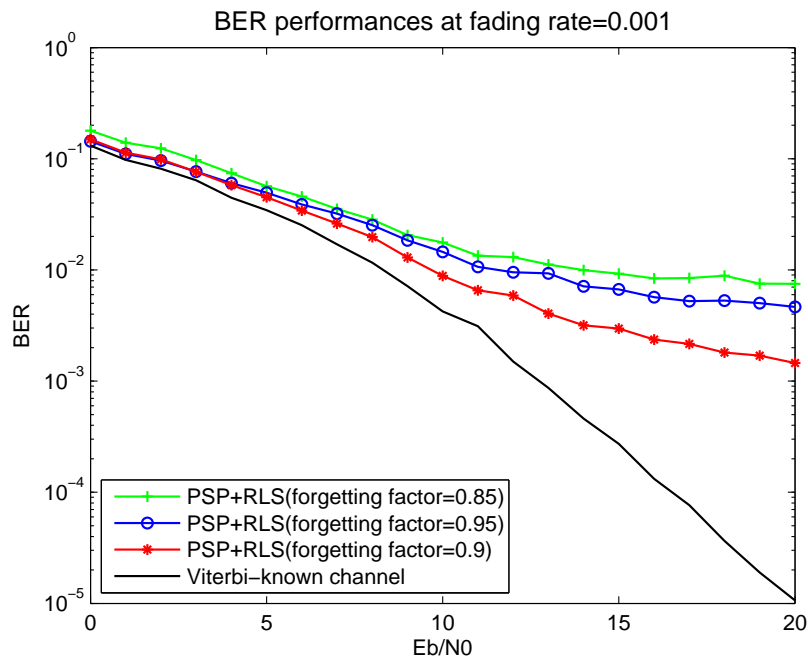


Figure 3.5: BER performance of PSP via RLS algorithm at fading rate  $f_D T = 0.001$ .

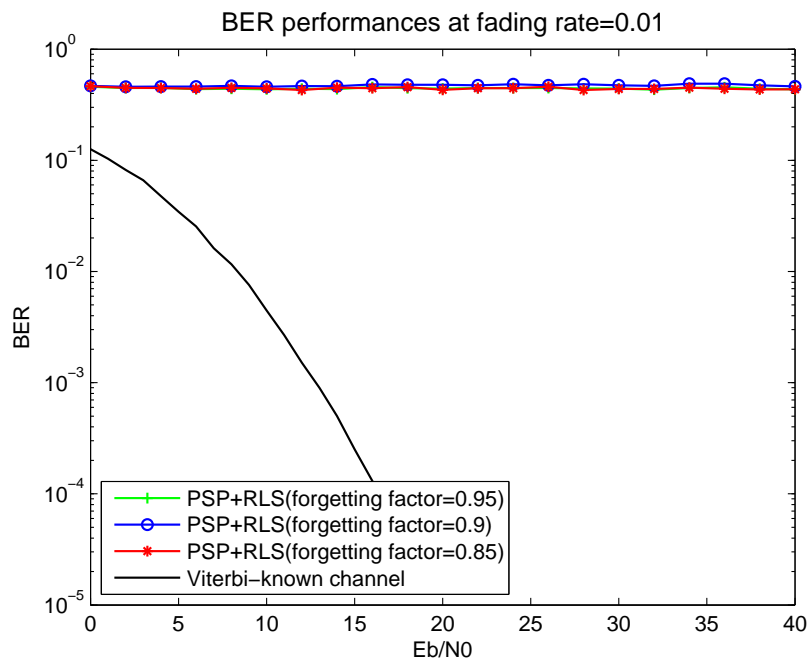


Figure 3.6: BER performance of PSP via RLS algorithm at fading rate  $f_D T = 0.01$ .



### 3. Estimation of Linear Dynamic Channels based on Kalman Filtering

---

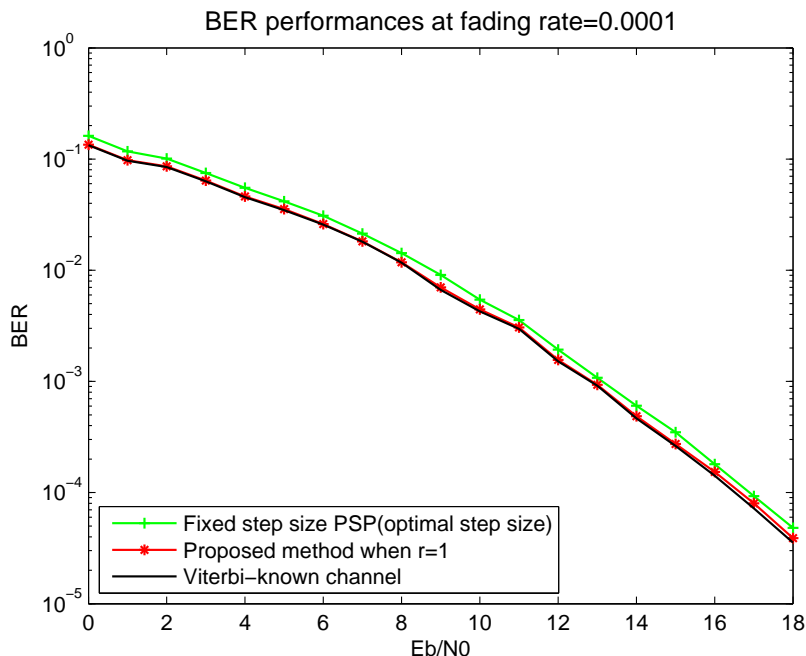


Figure 3.7: BER performance at fading rate  $f_D T = 0.0001$ .

algorithm with fixed step size PSP with optimal step size. Our algorithm outperforms the fixed step size PSP with optimal step size and this is we expected from the Kalman filter as the optimum estimator.

### 3.5 Conclusion

This chapter was devoted to our proposed data detection method under linear dynamic channel. We started by a review of different data detection technique under linear dynamic channel. First, we presented fixed step-size method which is the classical tracking method and is important because of its simplicity and ease of computation. The step-size factor depends on fading rate and regulates the speed and stability of the tracking method. Then, we presented PSP via RLS algorithm which is a least square method. As discussed, the RLS algorithm has a faster rate of convergence than the LMS algorithm, while the LMS algorithm exhibits better tracking behavior than the RLS algorithm. The RLS and LMS algorithms offer similar performance which highly depends on the value of their parameters (i.e.,  $\lambda$  and  $\mu$ ). We introduced a state space model for linear dynamic channel and based on

### 3.5. Conclusion

---

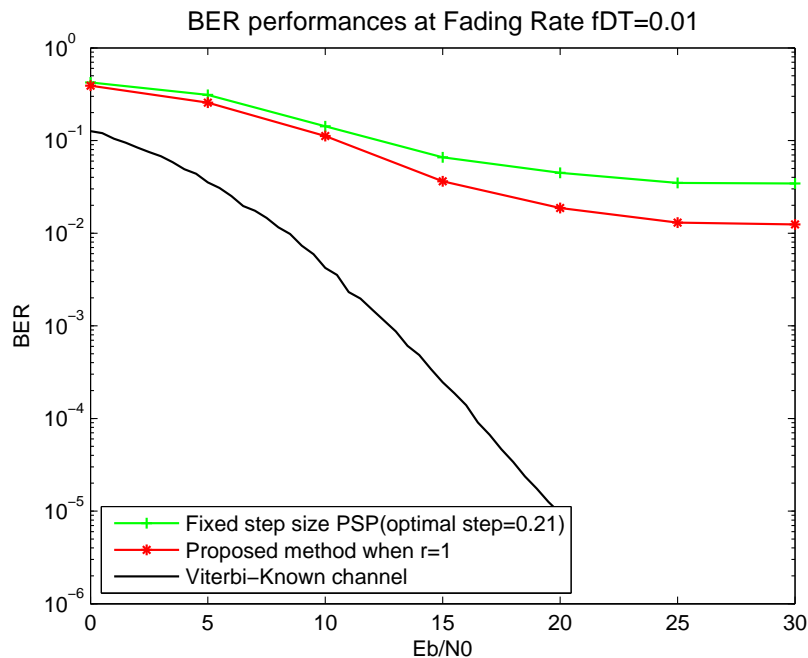


Figure 3.8: BER performance at fading rate  $f_D T = 0.01$ .

this representation, we applied the Kalman filter as the best estimator in the channel estimation. Unlike many contributors, we do not attempt to estimate the current realization of the channel process for joint channel and data detection. Numerical simulations showed that the proposed method outperforms the fixed step size PSP with optimal step size and PSP via RLS algorithm with optimal forgetting factor.

### 3. Estimation of Linear Dynamic Channels based on Kalman Filtering

# Chapter 4

## EM-based Estimation of Discrete-Data Transmitted over Non-Linear Dynamic Wireless Channels

### 4.1 Introduction

The Kalman filter is an optimal approach when the equations are linear and the noises are independent, additive, and Gaussian. In this situation, the distributions of interest are also Gaussian and the Kalman filter can compute them exactly without approximations. For scenarios where the models are nonlinear or the noise is non-Gaussian, two major approaches in the literature are considered. First approach is the extended Kalman filter (EKF) [97] and the other is called particle filtering [98]. The EKF is the nonlinear version of the Kalman filter which linearizes about an estimate of the current mean and covariance. The particle filtering as an important alternative method to the EKF is a sequential Monte Carlo methodology where the basic idea is the recursive computation of relevant probability distributions using the concepts of importance sampling and approximation of probability distributions with discrete random measures. The underlying principle of the methodology is the approximation of relevant distributions with random measures composed of particles

---

## 4. EM-based Estimation of Discrete-Data Transmitted over Non-Linear Dynamic Wireless Channels

---

(samples from the space of the unknowns) and their associated weights [99].

In addition to these two approaches, the EM procedure as the natural approach, is an efficient iterative procedure to compute the maximum likelihood (ML) estimate in the presence of missing or hidden data. Under unknown channel parameters, it is not possible to maximize the likelihood function directly to obtain the ML criterion. In this situation, the EM algorithm which achieves the ML criterion in an iterative manner is ideally suited to this problem. The EM algorithm has two steps, expectation and maximization [51]. The first step takes the expectation of the log-likelihood function of the complete data given the current estimated parameters and the incomplete data (e.g. received signal). The second step provides a new estimate of the unknown parameters by maximizing the expectation of the log-likelihood function over the unknown parameters. These steps are repeated iteratively to increase the likelihood till the new estimated parameters become (or get arbitrarily close to) the same parameter value estimated at the previous iteration [100]. The EM algorithm couples estimation and detection and uses decision feedback inherently; thus it motivates to use EM in the joint channel estimation and data detection problem.

This chapter is organized as follows: First the system model is introduced in section 4.2. Here, we use switching state-space model (SSSM) as non linear state-space model. This model combines the hidden Markov model (HMM) and linear state-space model (LSSM). In order to channel estimation and data detection, we propose a new approach based on EM algorithm in section 4.3. After determining the conditional expectation, the maximization is performed by dynamic programming (Viterbi) procedure in section 4.3.2. The simulation results of the proposed algorithm are presented in section 4.4. Finally, the main features of our approach are summed up in the conclusion.

### 4.2 System Model

We consider block transmission over a single-input single-output (SISO) switching doubly selective channel (SDSC) (block length:  $N$ , channel memory:  $M$ , maximum Doppler frequency shift at switch state  $m$ :  $f_D(m)$ , symbol duration:  $T$ ) and

## 4.2. System Model

---

binary phase shift keying (BPSK) modulation, so that the bit is transmitted at instant  $n$ ,  $b_n \in \{-1, +1\}$ .

We assume a discrete Rayleigh fading channel of memory  $M$ , simulated with method introduced in [92], which the elements of the impulse response  $\{c_n^i(m)\}_{i=0}^M$  are modeled as independent zero-mean complex Gaussian random variables with variance  $a_i(m)$ :

$$c_n^i(m) = \sqrt{\frac{a_i(m)}{N_0}} \sum_{q=0}^{N_0-1} \cos(2\pi n f_D(m) T \cos \gamma_{qi} + \phi_{qi}) + j \sin(2\pi n f_D(m) T \sin \gamma_{qi} + \phi'_{qi}) \quad m \in \{1, 2\} \quad (4.1)$$

where

$$\gamma_{qi} = \frac{2\pi q}{4N_0} + \frac{2\pi i}{4N_0(M+1)} + \frac{\pi}{8N_0(M+1)}$$

$\phi_{qi}, \phi'_{qi}$ , for  $q = 0, 1, 2, \dots, N_0 - 1$  and  $i = 0, 1, 2, \dots, M$ , are  $2(M+1)N_0$  independent random phases, each of which is uniformly distributed in  $[0, 2\pi)$ , also we consider  $N_0 > 16$ .

In order to accurately capture the dynamics of the wireless channel, we formulate a channel model suitable for use in the channel tracker. This model must be remaining mathematically tractable for implementation in a discrete-time state-space context. According to fading process which is modelled as a complex Gaussian process, a suitable model is thus an auto-regressive (AR) model. Information theoretic results have shown that a first-order AR model is sufficient to accurately represent the local behavior of the time-varying wireless channel [93]. A higher order model while providing more accurate long-term channel estimates, necessarily requires an AR order of 100 – 200 coefficients [94], and is thus highly intractable for the state model. Using the first-order assumption, we finally realize the state evolution at time  $n$  as:

$$c_n^i(m) = \xi(m)c_{n-1}^i(m) + v_n^i(m) \quad i = 0, \dots, M \quad m \in \{1, 2\} \quad (4.2)$$

where  $\xi(m)$  is the static AR coefficient at switch state  $m$  and  $v_n^i(m) \sim \mathcal{N}_{\mathbb{C}}(0, \sigma_v^2(m))$  is the complex driving noise of the model. Therefore in the form of state model we have:

$$\mathbf{c}_n^{(m)} = \mathbf{F}(m)\mathbf{c}_{n-1}^{(m)} + \mathbf{v}_n(m) \quad (4.3)$$

#### 4. EM-based Estimation of Discrete-Data Transmitted over Non-Linear Dynamic Wireless Channels

---

where  $\mathbf{c}_n^{(m)}$  is a vector of length  $M + 1$  which each element is the channel gain at time  $n$ .

$$\mathbf{c}_n^{(m)} = [c_n^0(m), c_n^1(m), \dots, c_n^M(m)]^T \quad (4.4)$$

the state transition matrix is given by:

$$\mathbf{F}(m) = \xi(m)\mathbf{I}_{M+1} \quad (4.5)$$

and the process noise vector given by:

$$\mathbf{v}_n(m) = [v_n^0(m), v_n^1(m), \dots, v_n^M(m)]^T \quad (4.6)$$

with covariance matrix equal to:

$$\mathbf{Q}(m) = (\sigma_v^2(m))\mathbf{I}_{M+1} \quad (4.7)$$

Additional advantages of using the AR model for describing the evolution of the channel state include:

1. The model is simple and mathematically tractable.
2. The true channel impulse response tends to revert to zero; the behavior of (4.2) also tends to revert to zero.
3. Like the wireless channel, the AR model is a Markov process. This implies that the pdf for the current estimate is not dependent upon all previous estimates but only on the most recent estimate. Owing to the Markovian property, the AR model greatly simplifies the complexity of the recurrence relations used in the EM algorithm.

In order to parameterize (4.2), we note from [3] that the autocorrelation of the channel fading process is:

$$E[c_n^i(m)c_{n-k}^i(m)^*] = a_i(m)J_0(2\pi k f_D(m)T) \quad m \in \{1, 2\} \quad (4.8)$$

where  $J_0()$  is the zeroth-order Bessel function,  $T$  is symbol duration, and  $f_D(m)$  denotes the Doppler frequency at switch state  $m$  resulting from relative motion between the transmitter and receiver. The Doppler shift itself is given by

$$f_D(m) = \frac{v(m)}{c} f_c \quad (4.9)$$

### 4.3. Proposed EM-based Approach to Data Detection

---

where  $v(m)$  is the mobile speed at switch state  $m$ ,  $c$  is the speed of light, and  $fc$  is the carrier frequency. Equating (4.2) to the autocorrelation of (4.8) for time  $n = \{0,1\}$ , we respectively have

$$\xi(m)^2 a_i(m) + \sigma_v(m)^2 = a_i(m) \quad (4.10)$$

$$\xi(m) = J_0(2\pi f_D(m)T) \quad (4.11)$$

For example, if the normalized desired fading rate is  $f_D T = 0.01$  (a typical fast fading rate), then  $\xi = 0.999$ .

The received complex noisy observation at instant  $n$  has the following form:

$$y_n = \sum_{i=0}^M c_n^i(\alpha_n = m) b_{n-i} + w_n \quad (4.12)$$

where  $w_n$  is the additive noise as zero mean complex random with variance  $\beta$ , also  $\alpha_n$  is the switch variable and  $m$  is switch state  $m \in \{1, 2\}$ . Therefore

$$y_n = \mathbf{d}_n^T \mathbf{c}_n^{(\alpha_n=m)} + w_n \quad (4.13)$$

where

$$\mathbf{d}_n = [b_n, b_{n-1}, \dots, b_{n-M}]^T$$

Finally our communication system can be described as switching state space model, whose dynamics are given by:

$$\begin{aligned} \mathbf{c}_n^{(1)} &= \mathbf{F}(1)\mathbf{c}_{n-1}^{(1)} + \mathbf{v}_n(1) \\ \mathbf{c}_n^{(2)} &= \mathbf{F}(2)\mathbf{c}_{n-1}^{(2)} + \mathbf{v}_n(2) \\ y_n &= \mathbf{d}_n^T \mathbf{c}_n^{(\alpha_n=m)} + w_n \quad m \in \{1, 2\} \end{aligned} \quad (4.14)$$

The graphical models representation for switching state-space models and HMM of switch variable  $\alpha_n$  are shown in Figure 4.1 and Figure 4.2 respectively.

### 4.3 Proposed EM-based Approach to Data Detection

We consider the problem of the maximum likelihood (ML) estimation of the sequence  $\theta_N = (\mathbf{d}_{0:N}, \alpha_{0:N})$  given  $y_{0:N}$ . In order to estimate  $\theta_N$ , it is typical to



#### 4. EM-based Estimation of Discrete-Data Transmitted over Non-Linear Dynamic Wireless Channels

---

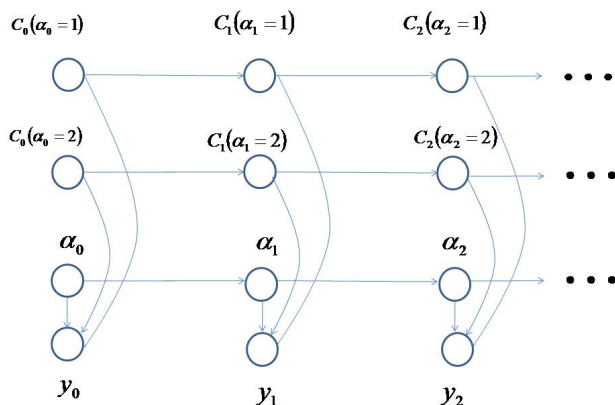


Figure 4.1: Graphical model representation for switching state-space model (SSSM).  $\alpha_n$  is the discrete switch variable and  $\mathbf{c}_n(1)$ ,  $\mathbf{c}_n(2)$  are the complex-valued state vectors.

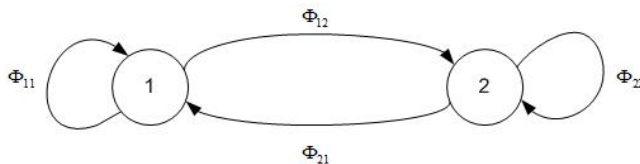


Figure 4.2: HMM of switch variable  $\alpha_n$ .

introduce the log likelihood function defined as:

$$L(\theta_N) = \ln p(y_{0:N}|\theta_N) \quad (4.15)$$

since  $\ln(x)$  is a strictly increasing function, the value of  $\theta_N$  which maximize  $p(y_{0:N}|\theta_N)$  also maximize  $L(\theta_N)$ . The conditional normality of the model suggests the EM procedure [51] as the natural approach, at least algorithmically. In this way EKF and particle filters are avoided. The EM algorithm is an iterative procedure for maximizing  $L(\theta_N)$ . Assume that the current estimate for  $\theta_N$  is given by  $\theta'_N = (\mathbf{d}'_{0:N}, \alpha'_{0:N})$ . Since the objective is to maximize  $L(\theta_N)$ , we wish to compute an updated estimate  $\theta_N$  such that,

$$L(\theta_N) > L(\theta'_N) \quad (4.16)$$

Equivalently we want to maximize the difference

$$L(\theta_N) - L(\theta'_N) = \ln p(y_{0:N}|\theta_N) - \ln p(y_{0:N}|\theta'_N) \quad (4.17)$$

### 4.3. Proposed EM-based Approach to Data Detection

---

So far, we have not considered any unobserved or missing variables. In problems where such data exist, the EM algorithm provides a natural framework for their inclusion. Alternately, hidden variables may be introduced purely as an artifice for making the maximum likelihood estimation of  $\theta_N$  tractable. In this case, it is assumed that knowledge of the hidden variables will make the maximization of the likelihood function easier. Parameters  $(\mathbf{d}_{0:N}, \alpha_{0:N}, \mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)})$  form a complete set respecting that the density family  $p(y_{0:N}, \mathbf{d}_{0:N}, \alpha_{0:N}, \mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)})$  is exponential. The continuous part  $\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)}$  will be treated as hidden variables. The total probability  $p(y_{0:N}|\theta_N)$  may be written in term of the hidden variables  $\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)}$

$$p(y_{0:N}|\theta_N) = \int p(y_{0:N}|\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)}, \theta_N) p(\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)}|\theta_N) d\mathbf{c}_{0:N}^{(1)} d\mathbf{c}_{0:N}^{(2)}$$

We may then rewrite Equation (4.17), as:

$$\begin{aligned} L(\theta_N) - L(\theta'_N) &= \ln \int p(y_{0:N}|\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)}, \theta_N) \\ &\quad \times p(\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)}|\theta_N) d\mathbf{c}_{0:N}^{(1)} d\mathbf{c}_{0:N}^{(2)} - \ln p(y_{0:N}|\theta'_N) \end{aligned}$$

If we introduce the constants  $p(\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)}|y_{0:N}, \theta'_N)$  in the equation, we have:

$$\begin{aligned} L(\theta_N) - L(\theta'_N) &= \ln \int p(y_{0:N}|\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)}, \theta_N) \\ &\quad \times p(\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)}|\theta_N) \frac{p(\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)}|y_{0:N}, \theta'_N)}{p(\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)}|y_{0:N}, \theta'_N)} d\mathbf{c}_{0:N}^{(1)} d\mathbf{c}_{0:N}^{(2)} - \ln p(y_{0:N}|\theta'_N) \end{aligned}$$

Using Jensen's inequality as usual lower-bound, the log-likelihood with these parameters are as:

$$\begin{aligned} L(\theta_N) - L(\theta'_N) &\geq \int p(\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)}|y_{0:N}, \theta'_N) \\ &\quad \ln \frac{p(y_{0:N}|\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)}, \theta_N) p(\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)}|\theta_N)}{p(\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)}|y_{0:N}, \theta'_N) p(y_{0:N}|\theta'_N)} d\mathbf{c}_{0:N}^{(1)} d\mathbf{c}_{0:N}^{(2)} = \Delta(\theta_N|\theta'_N) \end{aligned} \quad (4.18)$$

We continue by writing

$$L(\theta_N) \geq L(\theta'_N) + \Delta(\theta_N|\theta'_N) \quad (4.19)$$

and for convenience define

$$l(\theta_N|\theta'_N) = L(\theta'_N) + \Delta(\theta_N|\theta'_N) \quad (4.20)$$

## 4. EM-based Estimation of Discrete-Data Transmitted over Non-Linear Dynamic Wireless Channels

---

so that

$$L(\theta_N) \geq l(\theta_N|\theta'_N) \quad (4.21)$$

Our objective is to choose a value of  $\theta_N$  so that  $L(\theta_N)$  is maximized. In order to achieve the greatest possible increase in the value of  $L(\theta_N)$ , the EM algorithm calls for selecting  $\theta_N$  such that  $l(\theta_N|\theta'_N)$  is maximized. We denote this updated value as  $\theta''_N$ , formally, we have:

$$\theta''_N = \arg \max_{\theta_N} \{l(\theta_N|\theta'_N)\} \quad (4.22)$$

If we drop terms which are constant with respect to  $\theta_N$ , we have

$$\begin{aligned} \theta''_N &= \arg \max_{\theta_N} \left\{ \int p(\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)} | y_{0:N}, \theta'_N) \ln p(y_{0:N}, \mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)} | \theta_N) d\mathbf{c}_{0:N}^{(1)} d\mathbf{c}_{0:N}^{(2)} \right\} \\ &= \arg \max_{\theta_N} \{ E_{c^{(1)}, c^{(2)}|y, \theta'} \{ \ln p(y_{0:N}, \mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)} | \theta_N) \} \} \end{aligned} \quad (4.23)$$

In Equation (4.23) the expectation and maximization steps are apparent. The EM algorithm thus consists of iterating the:

1. E-step: Determine the conditional expectation  $E_{c^{(1)}, c^{(2)}|y, \theta'} \{ \ln p(y_{0:N}, \mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)} | \theta_N) \}$
2. M-step: Maximize this expression with respect to  $\theta_N$

### 4.3.1 Expectation Step

In order to determine the conditional expectation  $E_{c^{(1)}, c^{(2)}|y, \theta'} \{ \ln p(y_{0:N}, \mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)} | \theta_N) \}$ , first we decompose the integral (using independence and Markovian characters) as:

$$\begin{aligned} E_{c^{(1)}, c^{(2)}|y, \theta'} \{ \ln p(y_{0:N}, \mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)} | \theta_N) \} &= \int p(\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)} | y_{0:N}, \theta'_N) \ln p(y_{0:N}, \mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)} | \theta_N) \\ &\quad d\mathbf{c}_{0:N}^{(1)} d\mathbf{c}_{0:N}^{(2)} = \int p(\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)} | y_{0:N}, \theta'_N) \ln p(y_{0:N} | \mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)}, \theta_N) d\mathbf{c}_{0:N}^{(1)} d\mathbf{c}_{0:N}^{(2)} \\ &\quad + \int p(\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)} | y_{0:N}, \theta'_N) \ln p(\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)} | \theta_N) d\mathbf{c}_{0:N}^{(1)} d\mathbf{c}_{0:N}^{(2)} \\ &= E_{c^{(1)}, c^{(2)}|y, \theta'} [\ln p(y_{0:N} | \mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)}, \theta_N)] + E_{c^{(1)}, c^{(2)}|y, \theta'} [\ln p(\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)} | \theta_N)] \end{aligned} \quad (4.24)$$

If we consider

$$\begin{aligned} J(n, \mathbf{d}_n, \alpha_n) &= E_n [\ln p(y_n | \mathbf{c}_n^{(m)}, \mathbf{d}_n, \alpha_n = m)] \\ H_1(n) &= E_n [\ln p(\mathbf{c}_n^{(1)} | \mathbf{c}_{n-1}^{(1)})] \\ H_2(n) &= E_n [\ln p(\mathbf{c}_n^{(2)} | \mathbf{c}_{n-1}^{(2)})] \end{aligned} \quad (4.25)$$

### 4.3. Proposed EM-based Approach to Data Detection

---

where  $E_n[\cdot]$  denotes expectation with respect to the density

$$p_n(\mathbf{t}', \mathbf{t}, \mathbf{x}', \mathbf{x}) = p(\mathbf{c}_{n-1}^{(1)} = \mathbf{t}', \mathbf{c}_n^{(1)} = \mathbf{t}, \mathbf{c}_{n-1}^{(2)} = \mathbf{x}', \mathbf{c}_n^{(2)} = \mathbf{x} | y_{0:N}, \theta'_N) \quad n = 1 : N \quad (4.26)$$

and

$$p_n(\mathbf{t}, \mathbf{x}) = p(\mathbf{c}_n^{(1)} = \mathbf{t}, \mathbf{c}_n^{(2)} = \mathbf{x} | y_{0:N}, \theta'_N) \quad n = 0$$

then we have:

$$E_{c^{(1)}, c^{(2)} | y, \theta'} \{ \ln p(y_{0:N}, \mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)} | \theta'_N) \} = \sum_{n=0}^N J(n, \mathbf{d}_n, \alpha_n) + \sum_{n=1}^N H_1(n) + \sum_{n=1}^N H_2(n) + E_0[\ln p(\mathbf{c}_0^{(1)})] + E_0[\ln p(\mathbf{c}_0^{(2)})] \quad (4.27)$$

The most consuming part is of course the generation of conditional marginal. Therefore, thanks to the Markov and conditionally normal character, first we compute densities  $p_n(\mathbf{t}', \mathbf{t}, \mathbf{x}', \mathbf{x})$  by using a variant of the general forward-backward procedure which is very efficient. Then we calculate expected quantities  $J(n, \mathbf{d}_n, \alpha_n)$ ,  $H_1(n)$ ,  $H_2(n)$ ,  $E_0[\ln p(\mathbf{c}_0^{(1)})]$  and  $E_0[\ln p(\mathbf{c}_0^{(2)})]$ .

#### 4.3.1.1 Computing Densities $p_n(\mathbf{t}', \mathbf{t}, \mathbf{x}', \mathbf{x})$

$$p_n(\mathbf{t}', \mathbf{t}, \mathbf{x}', \mathbf{x}) = p(\mathbf{c}_{n-1}^{(1)} = \mathbf{t}', \mathbf{c}_n^{(1)} = \mathbf{t}, \mathbf{c}_{n-1}^{(2)} = \mathbf{x}', \mathbf{c}_n^{(2)} = \mathbf{x} | y_{0:N}, \theta'_N) = \int \frac{p(\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)}, y_{0:N} | \theta'_N)}{p(y_{0:N} | \theta'_N)} d\mathbf{c}_{0:n-2}^{(1)} d\mathbf{c}_{0:n-2}^{(2)} d\mathbf{c}_{n+1:N}^{(1)} d\mathbf{c}_{n+1:N}^{(2)} \quad (4.28)$$

by splitting

$$\begin{aligned} p(\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)}, y_{0:N} | \theta'_N) &= p(y_{0:N} | \mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)}, \theta'_N) p(\mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)} | \theta'_N) \\ &= p(y_{0:n} | \mathbf{c}_{0:n}^{(1)}, \mathbf{c}_{0:n}^{(2)}, \theta'_N) p(\mathbf{c}_{0:n}^{(1)}, \mathbf{c}_{0:n}^{(2)} | \theta'_N) p(y_{n+1:N} | \mathbf{c}_{n+1:N}^{(1)}, \mathbf{c}_{n+1:N}^{(2)}, \theta'_N) \\ &\quad \times p(\mathbf{c}_{n+1:N}^{(1)}, \mathbf{c}_{n+1:N}^{(2)} | \mathbf{c}_n^{(1)}, \mathbf{c}_n^{(2)}, \theta'_N) \end{aligned} \quad (4.29)$$

where one has used independence for the noise and Markov property for  $\mathbf{c}_{0:N}$ , one finds that:

$$p_n(\mathbf{t}', \mathbf{t}, \mathbf{x}', \mathbf{x}) = \frac{\Lambda_n^{(a)}(\mathbf{t}', \mathbf{t}, \mathbf{x}', \mathbf{x}) \Lambda_n^{(b)}(\mathbf{t}, \mathbf{x})}{p(y_{0:N} | \theta'_N)} \quad (4.30)$$

#### 4. EM-based Estimation of Discrete-Data Transmitted over Non-Linear Dynamic Wireless Channels

---

with:

$$\Lambda_n^{(a)}(\mathbf{t}', \mathbf{t}, \mathbf{x}', \mathbf{x}) = \int_{C^{n-2}} p(y_{0:n} | \mathbf{c}_{0:n-2}^{(1)}, \mathbf{c}_{0:n-2}^{(2)}, \mathbf{c}_{n-1}^{(1)} = \mathbf{t}', \mathbf{c}_n^{(1)} = \mathbf{t}, \mathbf{c}_{n-1}^{(2)} = \mathbf{x}', \mathbf{c}_n^{(2)} = \mathbf{x}, \theta'_N) \\ p(\mathbf{c}_{0:n-2}^{(1)}, \mathbf{c}_{0:n-2}^{(2)}, \mathbf{c}_{n-1}^{(1)} = \mathbf{t}', \mathbf{c}_n^{(1)} = \mathbf{t}, \mathbf{c}_{n-1}^{(2)} = \mathbf{x}', \mathbf{c}_n^{(2)} = \mathbf{x} | \theta'_N) d\mathbf{c}_{0:n-2}^{(1)} d\mathbf{c}_{0:n-2}^{(2)}$$

$$\Lambda_n^{(b)}(\mathbf{t}, \mathbf{x}) = \int_{C^{N-n}} p(y_{n+1:N} | \mathbf{c}_{n+1:N}^{(1)}, \mathbf{c}_{n+1:N}^{(2)}, \theta'_N) p(\mathbf{c}_{n+1:N}^{(1)} | \mathbf{c}_n^{(1)} = \mathbf{t}, \theta'_N) \\ p(\mathbf{c}_{n+1:N}^{(2)} | \mathbf{c}_n^{(2)} = \mathbf{x}, \theta'_N) d\mathbf{c}_{n+1:N}^{(1)} d\mathbf{c}_{n+1:N}^{(2)}$$

then with calculation of recursion of  $\Lambda_n^{(a)}(\mathbf{t}', \mathbf{t}, \mathbf{x}', \mathbf{x})$  and  $\Lambda_n^{(b)}(\mathbf{t}, \mathbf{x})$  and regarding to point that all density figuring in the preceding relations are normal so that they reduce conveniently to forward-Backward recursions on first and second conditional moments, which are calculated in Appendix E. By using results that are presented in Appendix E, we have:

for  $n = 1 : N$

$$p_n(\mathbf{t}', \mathbf{t}, \mathbf{x}', \mathbf{x}) = \frac{\gamma_n^{(a)} \gamma_n^{(b)}}{p(y_{0:N} | \theta'_N)} \exp[-\mathbf{z}^\dagger \Gamma_n^{(zb)} \mathbf{z} + 2\text{Re}(\mathbf{z}^\dagger \varphi_n^{(zb)}) \\ - \mathbf{1}^\dagger \Gamma_n^{(lb)} \mathbf{1} + 2\text{Re}(\mathbf{1}^\dagger \varphi_n^{(lb)})] \quad (4.31)$$

where

$$\mathbf{z}^\dagger = [\mathbf{t}'^\dagger, \mathbf{t}^\dagger] \\ \mathbf{1}^\dagger = [\mathbf{x}'^\dagger, \mathbf{x}^\dagger] \\ \varphi_n^{(zb)\dagger} = [\varphi_{n,1}^{(z)\dagger}, \varphi_{n,2}^{(z)\dagger} + \varphi_{n,1}^{(b)\dagger}] \\ \varphi_n^{(lb)\dagger} = [\varphi_{n,1}^{(l)\dagger}, \varphi_{n,2}^{(l)\dagger} + \varphi_{n,2}^{(b)\dagger}] \\ \Gamma_n^{(zb)} = \begin{pmatrix} \Gamma_n^{(z)}(11) & \Gamma_n^{(z)}(12) \\ \Gamma_n^{(z)\dagger}(12) & \Gamma_n^{(z)}(22) + \Gamma_{n,1}^{(b)} \end{pmatrix} \\ \Gamma_n^{(lb)} = \begin{pmatrix} \Gamma_n^{(l)}(11) & \Gamma_n^{(l)}(12) \\ \Gamma_n^{(l)\dagger}(12) & \Gamma_n^{(l)}(22) + \Gamma_{n,2}^{(b)} \end{pmatrix}$$

for  $n = 0$

$$p_n(\mathbf{t}, \mathbf{x}) = \frac{\gamma_n^{(a)} \gamma_n^{(b)}}{p(y_{1:N} | \theta'_N)} \exp[-\mathbf{t}^\dagger (\Gamma_n^{(t)} + \Gamma_{n,1}^{(b)}) \mathbf{t} + 2\text{Re}(\mathbf{t}^\dagger (\varphi_n^{(t)} + \varphi_{n,1}^{(b)})) - \mathbf{x}^\dagger (\Gamma_n^{(x)} + \Gamma_{n,2}^{(b)}) \mathbf{x} \\ + 2\text{Re}(\mathbf{x}^\dagger (\varphi_n^{(x)} + \varphi_{n,2}^{(b)}))]$$

### 4.3. Proposed EM-based Approach to Data Detection

---

The quantities of elements of vectors  $\varphi_n^{(zb)}$ ,  $\varphi_n^{(lb)}$ ,  $\varphi_n^{(x)}$  and the elements of matrices  $\mathbf{\Gamma}_n^{(zb)}$ ,  $\mathbf{\Gamma}_n^{(lb)}$ ,  $\mathbf{\Gamma}_n^{(x)}$  are shown in Appendix E.

#### 4.3.1.2 Computing Expected Quantities $J(n, \mathbf{d}_n, \alpha_n)$ , $H_1(n)$ , $H_2(n)$ , $E_0[\ln p(\mathbf{c}_0^{(1)})]$ and $E_0[\ln p(\mathbf{c}_0^{(2)})]$

We can now express the quantities figuring in  $E_{c^{(1)}, c^{(2)}|y, \theta'} \{\ln p(y_{0:N}, \mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)} | \theta_N)\}$  in terms of these moments.

We set:

$$\begin{aligned} \mathbf{z}_n^\dagger &= [\mathbf{c}_{n-1}^{(1)\dagger}, \mathbf{c}_n^{(1)\dagger}] \\ \mathbf{l}_n^\dagger &= [\mathbf{c}_{n-1}^{(2)\dagger}, \mathbf{c}_n^{(2)\dagger}] \\ \mathbf{V} &= [\mathbf{0}_{(M+1) \times (M+1)}, \mathbf{I}_{M+1}] \\ \psi^{(1)} &= [-\mathbf{F}(1), \mathbf{I}_{M+1}] \\ \psi^{(2)} &= [-\mathbf{F}(2), \mathbf{I}_{M+1}] \end{aligned}$$

from the recall equations that has been shown in Appendix E, for  $1 \leq n \leq N$ , we have:

$$\begin{aligned} E_n[\mathbf{z}_n] &= \mathbf{\Gamma}_n^{(zb)-1} \varphi_n^{(zb)} \\ E_n[\mathbf{z}_n \mathbf{z}_n^\dagger] &= \mathbf{\Gamma}_n^{(zb)-1} + \mathbf{\Gamma}_n^{(zb)-1} \varphi_n^{(zb)} \varphi_n^{(zb)\dagger} \mathbf{\Gamma}_n^{(zb)-1} \\ E_n[\mathbf{l}_n] &= \mathbf{\Gamma}_n^{(lb)-1} \varphi_n^{(lb)} \\ E_n[\mathbf{l}_n \mathbf{l}_n^\dagger] &= \mathbf{\Gamma}_n^{(lb)-1} + \mathbf{\Gamma}_n^{(lb)-1} \varphi_n^{(lb)} \varphi_n^{(lb)\dagger} \mathbf{\Gamma}_n^{(lb)-1} \end{aligned}$$

and

$$\begin{aligned} E_n[\mathbf{c}_n^{(1)}] &= \mathbf{V} E_n[\mathbf{z}_n] \\ E_n[\mathbf{c}_n^{(1)} \mathbf{c}_n^{(1)\dagger}] &= \mathbf{V} E_n[\mathbf{z}_n \mathbf{z}_n^\dagger] \mathbf{V}^T \\ E_n[\mathbf{c}_n^{(2)}] &= \mathbf{V} E_n[\mathbf{l}_n] \\ E_n[\mathbf{c}_n^{(2)} \mathbf{c}_n^{(2)\dagger}] &= \mathbf{V} E_n[\mathbf{l}_n \mathbf{l}_n^\dagger] \mathbf{V}^T \end{aligned}$$

#### 4. EM-based Estimation of Discrete-Data Transmitted over Non-Linear Dynamic Wireless Channels

---

when  $n = 0$ , we have:

$$\begin{aligned}
E_n[\mathbf{c}_0^{(1)}] &= (\Gamma_n^{(t)} + \Gamma_{n,1}^{(b)})^{-1}(\varphi_0^{(t)} + \varphi_{0,1}^{(b)}) \\
E_n[\mathbf{c}_0^{(1)} \mathbf{c}_0^{(1)\dagger}] &= (\Gamma_n^{(t)} + \Gamma_{n,1}^{(b)})^{-1} + (\Gamma_n^{(t)} + \Gamma_{n,1}^{(b)})^{-1}(\varphi_0^{(t)} + \varphi_{0,1}^{(b)})(\varphi_0^{(t)} + \varphi_{0,1}^{(b)})^\dagger(\Gamma_n^{(t)} + \Gamma_{n,1}^{(b)})^{-1} \\
E_n[\mathbf{c}_0^{(2)}] &= (\Gamma_n^{(x)} + \Gamma_{n,2}^{(b)})^{-1}(\varphi_0^{(x)} + \varphi_{0,2}^{(b)}) \\
E_n[\mathbf{c}_0^{(2)} \mathbf{c}_0^{(2)\dagger}] &= (\Gamma_n^{(x)} + \Gamma_{n,2}^{(b)})^{-1} + (\Gamma_n^{(x)} + \Gamma_{n,2}^{(b)})^{-1}(\varphi_0^{(x)} + \varphi_{0,2}^{(b)})(\varphi_0^{(x)} + \varphi_{0,2}^{(b)})^\dagger(\Gamma_n^{(x)} + \Gamma_{n,2}^{(b)})^{-1}
\end{aligned}$$

By definition:

$$\begin{aligned}
J(n, \mathbf{d}_n, \alpha_n) &= E_n[\ln p(y_n | \mathbf{c}_n^{(\alpha_n)}, \mathbf{d}_n)] \\
&= -\ln[\pi\beta] - E_n[\|y_n - \mathbf{d}_n^T \mathbf{c}_n^{(\alpha_n)}\|_{\beta^{-1}}^2] \\
&= -\ln[\pi\beta] - \|y_n\|^2/\beta + 2\text{Re}[y_n^* \beta^{-1} \mathbf{d}_n^T E_n[\mathbf{c}_n^{(\alpha_n)}]] - \text{tr} \mathbf{d}_n^* \beta^{-1} \mathbf{d}_n^T E_n[\mathbf{c}_n^{(\alpha_n)} \mathbf{c}_n^{(\alpha_n)\dagger}]
\end{aligned}$$

similarly:

$$\begin{aligned}
H_1(n) &= E_n[\ln p(\mathbf{c}_n^{(1)} | \mathbf{c}_{n-1}^{(1)})] = -\ln[\pi^{M+1} |\mathbf{Q}(1)|] - E_n[\|\mathbf{c}_n^{(1)} - \mathbf{F}(1) \mathbf{c}_{n-1}^{(1)}\|_{\mathbf{Q}^{-1}(1)}^2] \\
&= -\ln[\pi^{M+1} |\mathbf{Q}(1)|] - \text{tr} \psi^\dagger(1) \mathbf{Q}^{-1}(1) \psi(1) E_n[\mathbf{z}_n \mathbf{z}_n^\dagger]
\end{aligned}$$

$$\begin{aligned}
H_2(n) &= E_n[\log p(\mathbf{c}_n^{(2)} | \mathbf{c}_{n-1}^{(2)})] = -\ln[\pi^{M+1} |\mathbf{Q}(2)|] - E_n[\|\mathbf{c}_n^{(2)} - \mathbf{F}(2) \mathbf{c}_{n-1}^{(2)}\|_{\mathbf{Q}^{-1}(2)}^2] \\
&= -\ln[\pi^{M+1} |\mathbf{Q}(2)|] - \text{tr} \psi^\dagger(2) \mathbf{Q}^{-1}(2) \psi(2) E_n[\mathbf{I}_n \mathbf{I}_n^\dagger]
\end{aligned}$$

$$\begin{aligned}
E_0[\ln p(\mathbf{c}_0^{(1)})] &= -\ln[\pi^{M+1} |\mathbf{P}_0^{(1)}|] - E_0[\|\mathbf{c}_0^{(1)} - \hat{\mathbf{c}}_0^{(1)}\|_{\mathbf{P}_0^{(1)-1}}^2] \\
&= -\ln[\pi^{M+1} |\mathbf{P}_0^{(1)}|] + 2\text{Re}[\hat{\mathbf{c}}_0^{(1)\dagger} \mathbf{P}_0^{(1)-1} E_0[\mathbf{c}_0^{(1)}]] - \text{tr} \mathbf{P}_0^{(1)-1} E_0[\mathbf{c}_0^{(1)} \mathbf{c}_0^{(1)\dagger}]
\end{aligned}$$

$$\begin{aligned}
E_0[\ln p(\mathbf{c}_0^{(2)})] &= -\ln[\pi^{M+1} |\mathbf{P}_0^{(2)}|] - E_0[\|\mathbf{c}_0^{(2)} - \hat{\mathbf{c}}_0^{(2)}\|_{\mathbf{P}_0^{(2)-1}}^2] \\
&= -\ln[\pi^{M+1} |\mathbf{P}_0^{(2)}|] + 2\text{Re}[\hat{\mathbf{c}}_0^{(2)\dagger} \mathbf{P}_0^{(2)-1} E_0[\mathbf{c}_0^{(2)}]] - \text{tr} \mathbf{P}_0^{(2)-1} E_0[\mathbf{c}_0^{(2)} \mathbf{c}_0^{(2)\dagger}]
\end{aligned}$$

## 4.4. Numerical Results

---

### 4.3.1.3 Determining The Final Conditional Expectation

By using calculating results from previous section, we are able to determine the conditional expectation  $E_{c^{(1)}, c^{(2)}|y, \theta'} \{\ln p(y_{0:N}, \mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)} | \theta_N)\}$  as follow:

$$\begin{aligned}
E_{c^{(1)}, c^{(2)}|y, \theta'} \{\ln p(y_{0:N}, \mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)} | \theta_N)\} &= \sum_{n=0}^N J(n, \mathbf{d}_n, \alpha_n) + \sum_{n=1}^N H_1(n) + \sum_{n=1}^N H_2(n) \\
&+ E_n[\ln p(\mathbf{c}_0^{(1)})] + E_n[\ln p(\mathbf{c}_0^{(2)})] = \sum_{n=0}^N (-\ln[\pi\beta] - \|y_n\|^2/\beta + 2Re[y_n^* \beta^{-1} \mathbf{d}_n^T E_n[\mathbf{c}_n^{(\alpha_n)}]]) \\
&- tr \mathbf{d}_n^* \beta^{-1} \mathbf{d}_n^T E_n[\mathbf{c}_n^{(\alpha_n)} \mathbf{c}_n^{(\alpha_n)\dagger}] + \sum_{n=1}^N -\ln[\pi^{M+1} |\mathbf{Q}(1)|] - tr \psi^\dagger(1) \mathbf{Q}^{-1}(1) \psi(1) E_n[\mathbf{z}_n \mathbf{z}_n^\dagger] \\
&+ \sum_{n=1}^N -\ln[\pi^{M+1} |\mathbf{Q}(2)|] - tr \psi^\dagger(2) \mathbf{Q}^{-1}(2) \psi(2) E_n[\mathbf{1}_n \mathbf{1}_n^\dagger] - \ln[\pi^{M+1} |\mathbf{P}_0^{(1)}|] + 2Re[\hat{\mathbf{c}}_0^{(1)\dagger} \mathbf{P}_0^{(1)-1} E_0[\mathbf{c}_0^{(1)}]] \\
&- tr \mathbf{P}_0^{(1)-1} E_0[\mathbf{c}_0^{(1)} \mathbf{c}_0^{(1)\dagger}] - \ln[\pi^{M+1} |\mathbf{P}_0^{(2)}|] + 2Re[\hat{\mathbf{c}}_0^{(2)\dagger} \mathbf{P}_0^{(2)-1} E_0[\mathbf{c}_0^{(2)}]] - tr \mathbf{P}_0^{(2)-1} E_0[\mathbf{c}_0^{(2)} \mathbf{c}_0^{(2)\dagger}]
\end{aligned} \tag{4.32}$$

### 4.3.2 Maximization Step

In order to maximize expectation (4.32) with respect to  $\theta_N$ , we drop terms which are constant with respect to  $\theta_N$  and maintain other terms.

$$E_{c^{(1)}, c^{(2)}|y, \theta'} \{\ln p(y_{0:N}, \mathbf{c}_{0:N}^{(1)}, \mathbf{c}_{0:N}^{(2)} | \theta_N)\} = I_N(\theta_N | \theta'_N) + const. \tag{4.33}$$

where

$$I_N(\theta_N | \theta'_N) = \sum_{n=0}^N (2Re[y_n^* \beta^{-1} \mathbf{d}_n^T E_n[\mathbf{c}_n^{(\alpha_n)}]]) - tr \{\mathbf{d}_n^* \beta^{-1} \mathbf{d}_n^T E_n[\mathbf{c}_n^{(\alpha_n)} \mathbf{c}_n^{(\alpha_n)\dagger}]\} \tag{4.34}$$

and const. is the terms are independent of  $\theta_N$ . Therefore in order to maximize Equation (4.33) with respect to  $\theta_N$ , it is sufficient to maximize  $I_N(\theta_N | \theta'_N)$ . The maximization is performed by dynamic programming (Viterbi) procedure.

## 4.4 Numerical Results

In this section, we present some simulation results to compare the performance of our proposed method in comparison to the conventional fixed step size PSP. Regarding the wireless channel model, we consider a memory-2 Rayleigh fading channel simulated with the method introduced in [92]. The standard deviations of the resulting



#### 4. EM-based Estimation of Discrete-Data Transmitted over Non-Linear Dynamic Wireless Channels

---

three complex processes  $[c_n^0(m), c_n^1(m), c_n^2(m)]$  are set at  $(0.407, 0.815, 0.407)$ . We assume that the initial channel  $\mathbf{c}_0^{(1)} = [c_0^0(1), c_0^1(1), c_0^2(1)]^T$  and  $\mathbf{c}_0^{(2)} = [c_0^0(2), c_0^1(2), c_0^2(2)]^T$  is known (with using pilot symbols) at the receiver and the block length is  $N = 100$ . The switch state  $m$  was chosen using priors  $\pi^1 = \pi^2 = 1/2$  and transition probabilities  $\Phi_{11} = \Phi_{22} = 0.95$ ; and  $\Phi_{12} = \Phi_{21} = 0.05$ . For initializing of EM procedure we consider  $\alpha'_{0:N} = \alpha_0 * \mathbf{1}_N$  (where  $\alpha_0 = m$  is initial state which is known for the receiver) and  $\mathbf{d}'_{0:N}$  obtained from fixed step size PSP. For each value of SNR, convergence was attained after 3 iterations. The BER performance for different fading rate is depicted in Fig 4.3, Fig 4.4 and Fig 4.5. Results for switching state space model (SSSM) with fading rate 0.0001 for first LSSM and fading rate 0.0003 for second LSSM showed in Fig 4.3. We observe that for different transition probabilities  $\Phi_{11} = \Phi_{22}$ , we have the same result and the proposed algorithm performs near the optimal Viterbi algorithm, and outperforms the performance of PSP. Fig 4.4 shows result for SSSM with fading rate 0.0001 for first LSSM and fading rate 0.001 for second LSSM. We observe that the proposed algorithm has better result when transition probabilities  $\Phi_{11} = \Phi_{22}$  decreased and performs near the optimal Viterbi algorithm, and outperforms the performance of PSP. Results for SSSM with fading rate 0.0001 for first LSSM and fading rate 0.003 for second LSSM showed in Fig 4.5.

## 4.5 Conclusion

In this chapter, the problem of data detection under non linear dynamic channel was investigated. In order to perform joint channel estimation and data detection, a new iterative approach based on the EM algorithm, was proposed. After determining the conditional expectation, the maximization was performed by dynamic programming (Viterbi) procedure. Numerical simulations showed that the proposed method outperforms of PSP approach.

## 4.5. Conclusion

---

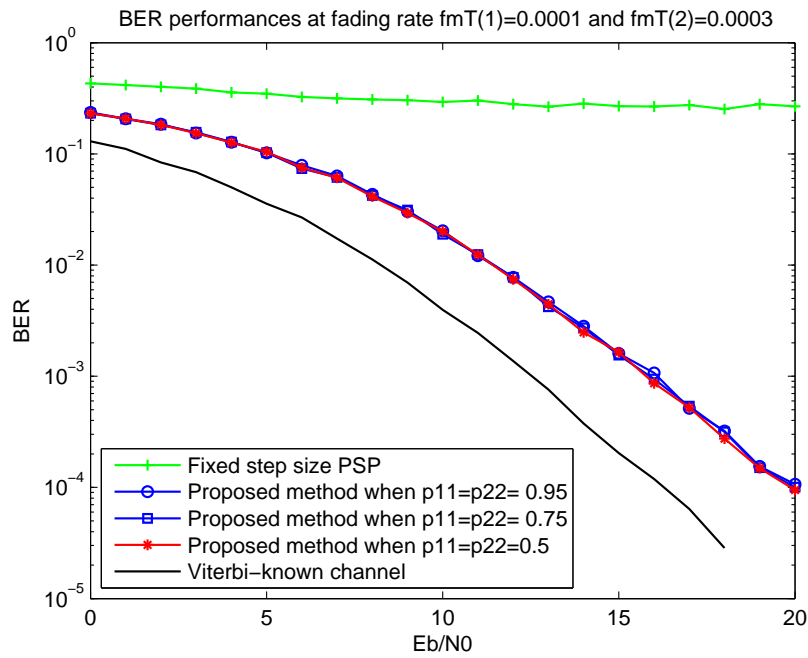


Figure 4.3: BER performance at switching fading rate  $f_D T(1) = 0.0001$  and  $f_D T(2) = 0.0003$

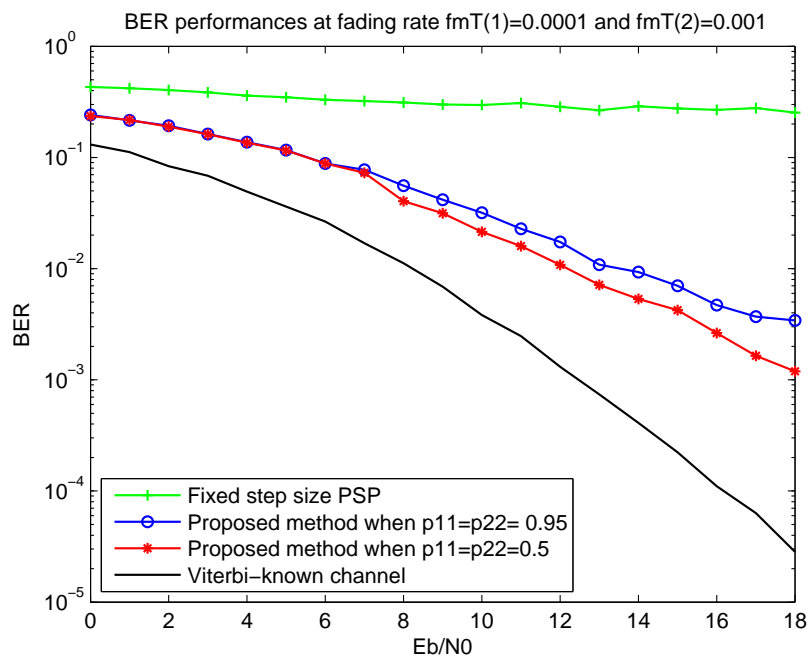


Figure 4.4: BER performance at switching fading rate  $f_D T(1) = 0.0001$  and  $f_D T(2) = 0.001$

#### 4. EM-based Estimation of Discrete-Data Transmitted over Non-Linear Dynamic Wireless Channels

---

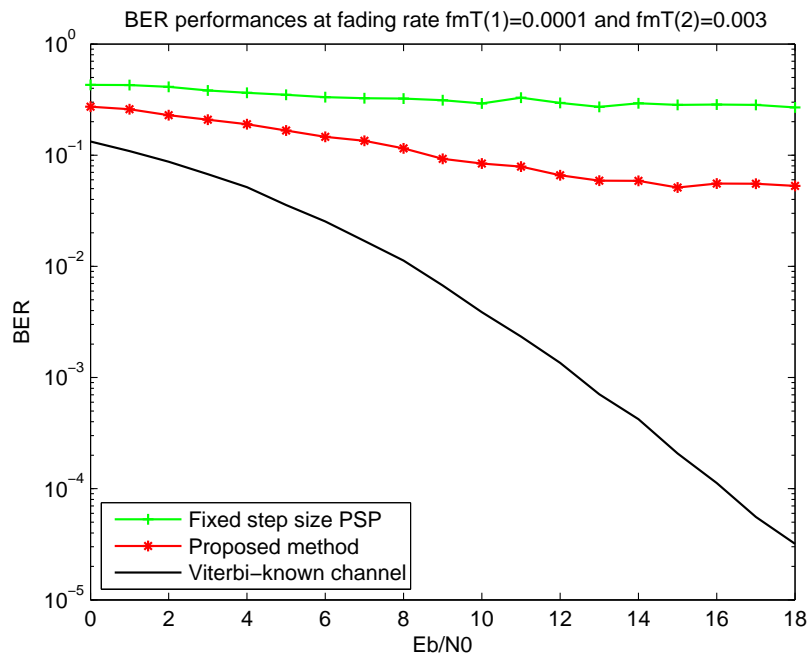


Figure 4.5: BER performance at switching fading rate  $f_D T(1) = 0.0001$  and  $f_D T(2) = 0.003$

# Chapter 5

## Conclusion and Future Research Perspectives

### 5.1 Summary

With the advent of multimedia services in mobile communication the demand for high data rate is continuously increasing. High data rate transmission on a bandlimited channel gives rise to ISI, the effects of which can be mitigated by employing an equalizer at the receiver. Optimal equalization methods for minimizing the bit error rate (BER) and the sequence error rate are nonlinear and are based on maximum likelihood sequence estimation (MLSE). As well known, the complexity of MLSE being exponential in the channel memory and in the symbol alphabet cardinality is quickly unmanageable and forces to resort to sub-optimal approaches. So reducing computational complexity is one of our favorite topic. In time-varying channels the estimation of channel parameters becomes a challenging problem. Therefore, in this thesis the parameter estimation and equalization techniques for doubly selective channels have been considered.

In Chapter 2, in order to reduce computational complexity of maximum likelihood sequence estimation (MLSE), we proposed a novel equalization structure with manageable complexity. By adopting a continuation approach, we proposed a new iterative equalizer. We formulated the sequence estimation problem in a precise setting to obtain the main iteration formula using EM. Then, we discussed about

---

---

## 5. Conclusion and Future Research Perspectives

---

complexity and convergence issues. The use of a cyclic preamble allows carrying out the summation contained in the so-called main iteration formula into the frequency domain, while resort to a Gaussian approximation enables to simplify its evaluation. Finally, the numerical results proving the interest of the method and the validity of our successive approximation were presented.

Data detection under linear dynamic channel was considered in Chapter 3. We introduced linear state space model in order to use the Kalman filter as the best estimator of the channel parameters. Then proposed method based on joint Kalman and Viterbi algorithms was presented. The aim in this chapter was to motivate rigorously the introduction of the Kalman filter in the estimation of Markov sequences through Gaussian dynamical channels. In this way we interpreted and quantified the underlying approximations in the heuristic approaches. Finally, we performed numerical results and conclusion.

Chapter 4 focused on the issue of data detection under non linear dynamic channel. Switching state-space model (SSSM) was considered as non linear state-space model in this chapter. This model combines the hidden Markov model (HMM) and linear state-space model (LSSM). In order to perform joint channel estimation and data detection, a new approach based on EM algorithm was presented. After determining the conditional expectation, the maximization was performed by dynamic programming (Viterbi) procedure. Finally, some simulation results and concluded remarks on assumptions were presented.

### 5.2 Research Perspectives

There are several interesting areas for future works on proposed algorithms and their application. First by considering the advantages of MIMO system such as significant increasing in data throughput and link rang without additional bandwidth or increased transmitting power, extending proposed algorithms for this system would be interesting to investigate in future research.

Second, application of proposed algorithms to higher-order modulation can be considered for future work.

Third, because of intrinsically probabilistic nature of proposed algorithm in

## 5.2. Research Perspectives

---

Chapter 2 and its close to optimal performance at low SNR, we believe that the proposed equalizer is perfectly suitable for block turbo-equalization of coded symbols.

Finally, in Chapter 4, if we consider an exponential basis expansion with  $L$  bases (EBE( $L$ )) [101] as an alternative model for mobile Rayleigh fading channels, we can derive a new method which has an important feature that is independent from fading rate. This feature is useful since the velocity of the receiver is usually unknown.

## 5. Conclusion and Future Research Perspectives

---

# Appendix A

## A Review of The EM Algorithm

### A.1 Introduction

The EM algorithm is a broadly applicable approach to the iterative computation of ML estimates, useful in a variety of incomplete-data problems, where other iterative algorithms may turn out to be more complicated. At each iteration of the EM algorithm, there are two processing steps called the expectation step (or the E-step) and the maximization step or (the M-step). That is why the algorithm is called EM. This name was given by Dempster, Laird, and Rubin (1977), referred usually as DLR, in their fundamental paper [51]. However, the EM algorithm was discovered and employed independently by several different researchers until DLR brought their ideas together, proved its convergence and coined the term "EM algorithm". The idea behind the EM algorithm being intuitive and natural, algorithms like EM had already been formulated and applied to a variety of problems.

In signal processing applications, the largest area of interest for the EM algorithm is in ML estimation/detection problems with incomplete-data, where there are missing data, truncated distributions, censored and grouped observations which result in complicated likelihood functions. However, the EM principle can be applied to a variety of situations where the incompleteness of data is not so natural or evident. These include statistical models such as random effects, mixtures, convolutions, log linear models, etc. A large list of references is found in [100].

The basic idea behind the EM algorithm is to associate with the given incomplete-

---



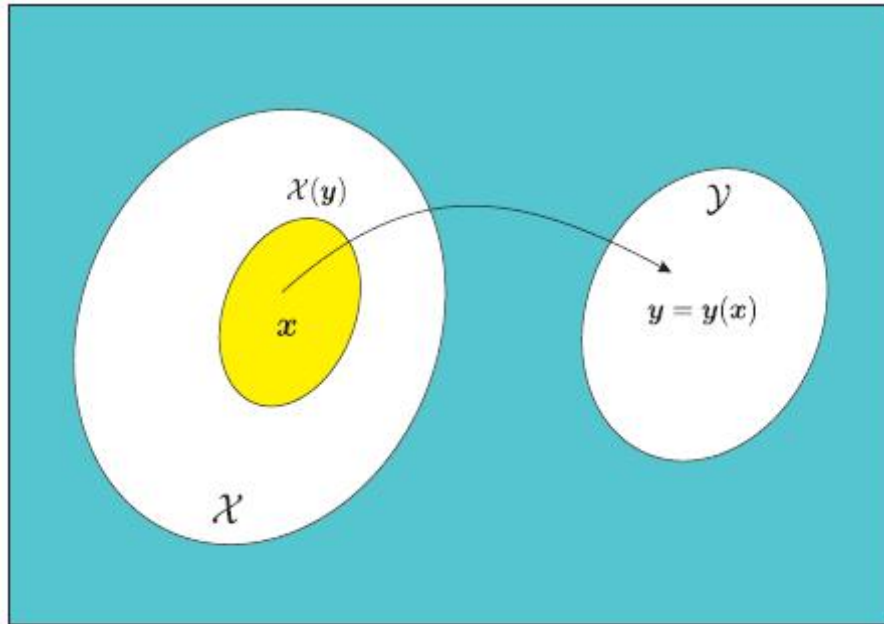


Figure A.1: An illustration of the complete- and incomplete-data sets of the EM algorithm.

data problem, a complete-data problem for which ML estimation is computationally more tractable. The methodology of the EM algorithm then consists in reformulating the problem in terms of this more easily solved complete-data problem. The E-step consists in manufacturing data for the complete-data problem using the incomplete observed data set and the current value of the unknown parameters, so that a simpler M-step computation can be applied to this "completed" data set. More precisely, it is the log-likelihood of the complete-data problem that is computed in the E-step. As it is partly based on unobservable (or hidden) data, it is replaced by its conditional expectation given the observed data, where this E-step is affected using the current estimate of the unknown parameters. Starting from suitable initial parameter values, the E and M-steps are repeated until convergence.

# A.2 General Statement of The EM Algorithm

## A.2.1 Mathematical Formulation

Let  $\mathcal{Y}$  denote the sample space of the observations, and let  $\mathbf{y} \in \mathbb{R}^m$  denote an observation from  $\mathcal{Y}$  of size  $m$ . Let  $\mathcal{X}$  denote an underlying space and let  $\mathbf{x} \in \mathbb{R}^n$  be an outcome from  $\mathcal{X}$  with  $m < n$ .

The data vector  $\mathbf{x}$  is referred to as the complete-data. The complete data  $\mathbf{x}$  is not observed directly but only by means of  $\mathbf{y}$  where  $\mathbf{y} = \mathbf{y}(\mathbf{x})$ , and  $\mathbf{y}(\mathbf{x})$  is a many-to-one mapping from  $\mathcal{X}$  to  $\mathcal{Y}$ . As shown in Fig A.1, an observation  $\mathbf{y}$  determines a subset of  $\mathcal{X}$ , which is denoted as  $\mathcal{X}(\mathbf{y})$ . The pdf of the complete-data vector is  $f(\mathbf{x}|\theta)$ , where  $\theta \in \Theta \subset \mathbb{R}^r$  is the set of unknown parameters that we have to estimate (we will refer to the density of the random variables for convenience, even for discrete random variables for which probability mass functions (pmf) would be appropriate). Moreover, the pdf  $f$  is assumed to be a continuous function of  $\theta$  and appropriately differentiable. The ML estimate of  $\theta$  is assumed to lie within the region  $\Theta$ . The pdf of the incomplete-data is

$$g(\mathbf{y}|\theta) = \int_{\mathcal{X}(\mathbf{y})} f(\mathbf{x}|\theta) d\mathbf{x} \quad (\text{A.1})$$

and denotes the incomplete-data likelihood function. Let  $L_i(\theta) = \log g(\mathbf{y}|\theta)$  and  $L_c(\theta) = \log f(\mathbf{x}|\theta)$  denote respectively the incomplete- and complete-data log-likelihood. The integral operation in (A.1) may render very difficult the estimation of the parameter  $\theta$  which maximizes the likelihood function  $g(\mathbf{y}|\theta)$ , even if the function  $\log f(\mathbf{x}|\theta)$  is easy to maximize. This remark justifies the idea of the EM algorithm.

As stated before, the basic idea behind the EM algorithm is that we would like to find  $\theta$  to maximize  $L_c(\theta) = \log f(\mathbf{x}|\theta)$ , but we do not have the data  $\mathbf{x}$  to compute the log-likelihood. So instead, we maximize the expectation of  $\log f(\mathbf{x}|\theta)$  given the data  $\mathbf{y}$  and our current estimate of  $\theta$ . This can be expressed in two steps.

More specifically, let  $\theta^{(0)}$  be some initial value for  $\theta$ . Then at the first iteration, the E-step requires the calculation of

$$Q(\theta, \theta^{(0)}) = E[\log f(\mathbf{x}|\theta) | \mathbf{y}, \theta^{(0)}] \quad (\text{A.2})$$

where  $Q(\cdot, \cdot)$  is called the auxiliary function. It is important to distinguish between the first and the second arguments of the auxiliary function. The second argument

---

## A. A Review of The EM Algorithm

is a conditioning argument to the expectation and is regarded as fixed and known at every E-step. The first argument conditions the likelihood of the complete-data. The M-step requires the maximization of  $Q(\theta, \theta^{(0)})$  with respect to  $\theta$  over the parameter space  $\Theta$ . That is, we choose  $\theta^{(1)}$  such that

$$Q(\theta^{(1)}, \theta^{(0)}) \geq Q(\theta, \theta^{(0)}). \quad (\text{A.3})$$

For all  $\theta \in \Theta$ , the E- and M-steps are then carried out again, but this time with  $\theta^{(0)}$  replaced by the current estimate  $\theta^{(1)}$ . On the  $(t + 1)$ -th iteration, the E- and M-steps are defined as follows.

- **E-step:** Calculate  $Q(\theta, \theta^{(t)})$  where

$$Q(\theta, \theta^{(t)}) = E[\log f(\mathbf{x}|\theta)|\mathbf{y}, \theta^{(t)}]. \quad (\text{A.4})$$

- **M-step:** Choose  $\theta^{(t+1)}$  to be any value of  $\theta \in \Theta$  that maximizes  $Q(\theta, \theta^{(t)})$  as

$$\theta^{(t+1)} = \arg \max_{\theta} Q(\theta, \theta^{(t)}). \quad (\text{A.5})$$

We mention that the expectation in the E-step is with respect to all unobserved (or hidden) variables in the complete-data set  $\mathcal{X}$ . We also note that the maximization in the M-step is with respect to the first argument of the  $Q$  function, i.e., the conditioner of the complete-data likelihood.

After initialization, the E- and M-steps are alternated repeatedly until convergence. Convergence may be determined by examining when the parameters remain almost unchange, i.e., stop when  $\|\theta^{(t)} - \theta^{(t-1)}\| < \epsilon$  or  $L_i(\theta^{(t)}) - L_i(\theta^{(t-1)}) < \epsilon$ , for some small value of  $\epsilon$  and some appropriate distance measure  $\|\cdot\|$ .

### A.2.2 Monotonicity of The EM Algorithm

DLR showed that the incomplete-data likelihood function  $g(\mathbf{y}|\theta)$  is not decreased after an EM iteration. This is formulated in the following theorem proved in [51].

**Theorem 1.**

$$Q(\theta^{(t+1)}, \theta^{(t)}) \geq Q(\theta^{(t)}, \theta^{(t)}) \Rightarrow g(\mathbf{y}|\theta^{(t+1)}) \geq g(\mathbf{y}|\theta^{(t)}) \quad (\text{A.6})$$

### A.3. Extension of the EM Algorithm to MAP Parameter Estimation

---

#### A.2.3 Convergence to a Stationary Value

As shown in the last section, for a sequence of likelihood values  $\{g(\mathbf{y}|\theta^{(t)})\}$ ,  $g(\mathbf{y}|\theta^{(t)})$  converges monotonically to some stationary value  $g^*$ . The stationary point may be a local maximum or a saddle point of the likelihood function. In general, if  $g(\mathbf{y}|\theta)$  has several stationary points, convergence of the EM sequence to either type (global or local maximum, saddle points) depends on the choice of the starting point  $\theta^0$ . Obviously, when the likelihood function is unimodal in  $\theta$  (and a certain differentiability is satisfied), any EM sequence converges to the unique global maximum irrespective of its starting. In what follows, we state without proof the main convergence theorem given by Wu in [102].

**Theorem2.** Let  $\{\theta^{(t)}\}$  be a sequence of parameters obtained from successive maximization of the auxiliary function  $Q(\theta, \theta^{(t)})$  at the M-step. Then all the limit points of  $\{\theta^{(t)}\}$  are stationary points of  $g(\mathbf{y}|\theta^{(t)})$  and  $g(\mathbf{y}|\theta^{(t)})$  converges monotonically to  $g^* = g(\mathbf{y}|\theta^*)$  for some stationary point  $\theta^*$ .

### A.3 Extension of the EM Algorithm to MAP Parameter Estimation

Up to now, we addressed the EM algorithm for ML estimation. Let us now consider a MAP criterion for the estimation of the unknown parameter  $\theta$  of which ML estimation is a particular case. Considering some prior distribution  $\pi(\theta)$  for the unknown parameter, the MAP estimate is given by

$$\hat{\theta} = \arg \max_{\theta} \{\log g(\mathbf{y}|\theta) + \log \pi(\theta)\} \quad (\text{A.7})$$

When the likelihood function  $g(\mathbf{y}|\theta)$  is hard to maximize, the EM algorithm is a mean for obtaining MAP estimates of a parameter  $\theta$ . The EM algorithm for MAP estimation can be summarized as follows.

- **E-step:** Calculate  $Q_{map}(\theta, \theta^{(t)})$  where

$$Q_{map}(\theta, \theta^{(t)}) = E[\log f(\mathbf{x}|\theta) + \log \pi(\theta) | \mathbf{y}, \theta^{(t)}] = Q(\theta, \theta^{(t)}) + \log \pi(\theta) \quad (\text{A.8})$$

- **M-step:** Choose  $\theta^{(t+1)}$  to be any value of  $\theta \in \Theta$  that maximizes  $Q_{map}(\theta, \theta^{(t)})$

## A. A Review of The EM Algorithm

---

as

$$\theta^{(t+1)} = \arg \max_{\theta} Q_{map}(\theta, \theta^{(t)}). \quad (\text{A.9})$$

We note that the E-step of MAP estimation differs from the E-step of ML estimation by the additive term  $\log \pi(\theta)$ . The presence of the term  $\log \pi(\theta)$  can also be exploited to render the auxiliary function concave. The M-step is also different since the maximization is performed over a modified auxiliary function.

The aforementioned convergence properties of the ML based EM are also valid for MAP estimation [51] [46]. Thus, each iteration of the EM algorithm is guaranteed to increase the logarithm of the incomplete-data a posteriori probability, that is

$$\log g(\mathbf{y}|\theta^{(t+1)}) + \log \pi(\theta^{(t+1)}) \geq \log g(\mathbf{y}|\theta^{(t)}) + \log \pi(\theta^{(t)}) \quad (\text{A.10})$$

# Appendix B

## Detailed Algorithm. See Chapter 3.

Here we give detailed algorithm for single input single output (SISO) system with  $r = 1$ . By using standard Kalman filter we have:

$$\begin{aligned}\hat{\mathbf{c}}_{n+1|n} &= \mathbf{F}_{n+1}\hat{\mathbf{c}}_{n|n-1} + \mathbf{F}_{n+1}\mathbf{k}_n(y_n - \mathbf{d}_n^T\hat{\mathbf{c}}_{n|n-1}) \\ \mathbf{P}_{n+1|n} &= \mathbf{F}_{n+1}(\mathbf{P}_{n|n-1} - \mathbf{k}_n\mathbf{d}_n^T\mathbf{P}_{n|n-1})\mathbf{F}_{n+1}^\dagger + \mathbf{Q}_{n+1} \\ \mathbf{k}_n &= \mathbf{P}_{n|n-1}\mathbf{d}_n^*(\mathbf{d}_n^T\mathbf{P}_{n|n-1}\mathbf{d}_n^* + \beta_n)^{-1}\end{aligned}$$

Outline of the algorithm:

1. For  $\mathbf{s} \in \Omega$  (priors)

$$\text{set } \hat{\mathbf{c}}_{1|0}(\mathbf{s}) = E(\mathbf{c}_1); \quad \mathbf{P}_{1|0}(\mathbf{s}) = \text{cov}(\mathbf{c}_1)$$

end( $\mathbf{s}$ )

2. For any  $\mathbf{s} \in \Omega$  compute:

$$\begin{aligned}\mathbf{q}(\mathbf{s}) &= \mathbf{P}_{1|0}^{-1}\hat{\mathbf{c}}_{1|0} + y_1\mathbf{s}^*\beta_0^{-1} \\ \Gamma(\mathbf{s}) &= \mathbf{P}_{1|0}^{-1} + \mathbf{s}^*\mathbf{s}^T\beta_0^{-1} \\ L_1(\mathbf{s}) &= |\Gamma(\mathbf{s})|^{-1}e^{\mathbf{q}^\dagger(\mathbf{s})\Gamma^{-1}(\mathbf{s})\mathbf{q}(\mathbf{s})} \\ \mathbf{d}_1^*(\mathbf{s}) &= \mathbf{d}_0 \quad \text{fixed}\end{aligned}$$

end ( $\mathbf{s}$ )

3. For  $n = 1 : T - 1$

For  $\mathbf{s} \in \Omega$

---

---

## B. Detailed Algorithm. See Chapter 3.

---

For  $\mathbf{s}' \in \Omega$

$$\mathbf{s}'' = d_n^*(\mathbf{s}')$$

$$\mathbf{k}_n(\mathbf{s}') = \mathbf{P}_{n|n-1}(\mathbf{s}'') \mathbf{s}'^* [\mathbf{s}'^T \mathbf{P}_{n|n-1}(\mathbf{s}'') \mathbf{s}'^* + \beta_n]^{-1}$$

$$\mathbf{P}_{n+1|n}(\mathbf{s}') = \mathbf{F}_{n+1}^\dagger [\mathbf{P}_{n|n-1}(\mathbf{s}'') - \mathbf{k}_n(\mathbf{s}') \mathbf{s}'^T \mathbf{P}_{n|n-1}(\mathbf{s}'')] \mathbf{F}_{n+1}^T + \mathbf{Q}_{n+1}$$

$$\hat{\mathbf{c}}_{n+1|n}(\mathbf{s}') = \mathbf{F}_{n+1} \hat{\mathbf{c}}_{n|n-1}(\mathbf{s}'') + \mathbf{F}_{n+1} \mathbf{k}_n(\mathbf{s}') (y_n - \mathbf{s}'^T \hat{\mathbf{c}}_{n|n-1}(\mathbf{s}''))$$

$$\mathbf{q}(\mathbf{s}', \mathbf{s}) = \mathbf{P}_{n+1|n}^{-1}(\mathbf{s}') \hat{\mathbf{c}}_{n+1|n}(\mathbf{s}') + \frac{y_{n+1} \mathbf{s}}{\beta_{n+1}}$$

$$\mu(\mathbf{s}') = \hat{\mathbf{c}}_{n+1|n}^\dagger(\mathbf{s}') \mathbf{P}_{n+1|n}^{-1}(\mathbf{s}') \hat{\mathbf{c}}_{n+1|n}(\mathbf{s}') + \frac{\|y_{n+1}\|^2}{\beta_{n+1}}$$

$$\Gamma(\mathbf{s}', \mathbf{s}) = \mathbf{P}_{n+1|n}^{-1}(\mathbf{s}') + \frac{\mathbf{s}^* \mathbf{s}^T}{\beta_{n+1}}$$

$$f_{n+1}^*(\mathbf{s}', \mathbf{s}) = |\mathbf{P}_{n+1|n}(\mathbf{s}') \Gamma(\mathbf{s}', \mathbf{s})|^{-1} \exp[(\mathbf{q}^\dagger(\mathbf{s}) \Gamma^{-1}(\mathbf{s}', \mathbf{s}) \mathbf{q}(\mathbf{s}) - \mu(\mathbf{s}'))]$$

$$L_{n+1}(\mathbf{s}', \mathbf{s}) = f_{n+1}^*(\mathbf{s}', \mathbf{s}) L_n(\mathbf{s}') p(\mathbf{s}|\mathbf{s}')$$

*End*( $\mathbf{s}'$ )

$$L_{n+1}(\mathbf{s}) = \max_{\mathbf{s}'} L_{n+1}(\mathbf{s}', \mathbf{s})$$

$$\mathbf{d}_{n+1}(\mathbf{s}) = \arg \max_{\mathbf{s}'} L_{n+1}(\mathbf{s}', \mathbf{s})$$

*End*( $\mathbf{s}$ )

*End*( $n$ )

4.  $\hat{\mathbf{d}}_T = \arg \max_{\mathbf{s}} L_T(\mathbf{s})$

For  $n = T - 1 : -1 : 1$

$$\hat{\mathbf{d}}_n = \mathbf{d}_{n+1}^*(\hat{\mathbf{d}}_{n+1})$$

*End*( $n$ )

5.  $\hat{\mathbf{d}}_1, \dots, \hat{\mathbf{d}}_T$  is the approximate ML estimate.

# Appendix C

## A Review of EKF

The Kalman filter is based on the assumption of a linear state space model(LSSM). In this section, in order to address nonlinear systems we introduce the extended Kalman filter [99].

### C.1 The Nonlinear State Space Model (NSSM)

Let us now consider a general nonlinear state space model of the form

$$\mathbf{c}_{n+1} = \phi(\mathbf{c}_n) + \mathbf{v}_n \quad (\text{C.1})$$

$$\mathbf{y}_n = \gamma(\mathbf{c}_n) + \mathbf{w}_n \quad (\text{C.2})$$

where  $\mathbf{v}_n$  and  $\mathbf{w}_n$  are uncorrelated, zero-mean, white noise processes with covariance matrices  $\mathbf{Q}_n$  and  $\mathbf{R}_n$ , respectively. The operators  $\phi(x)$  and  $\gamma(x)$  represent nonlinear vector-valued functions of  $x$  and  $n$ . Let  $x$  consist of  $m$  states; then  $\phi$  has the form

$$\phi(x) = \begin{bmatrix} \phi_1(x) \\ \phi_2(x) \\ \vdots \\ \phi_m(x) \end{bmatrix}$$

Since the measurement  $\mathbf{y}_n$  are  $p$ -vectors,  $\gamma$  consists of  $p$  nonlinear scalar-valued

---



function  $\gamma_i(x)$ , i.e.,

$$\gamma(x) = \begin{bmatrix} \gamma_1(x) \\ \gamma_2(x) \\ \vdots \\ \gamma_p(x) \end{bmatrix}$$

## C.2 Linearization of NSSM

The Kalman filter assumes a linear state space model (LSSM), so the next step involves linearization of the original SSM. We assume  $\phi$  and  $\gamma$  are sufficiently smooth in  $x$  so that each has a valid Taylor series expansion. Then, we expand  $\phi$  into a Taylor series about  $\hat{\mathbf{c}}_{n|n}$ :

$$\phi(\mathbf{c}_n) = \phi(\hat{\mathbf{c}}_{n|n}) + J_\phi(\hat{\mathbf{c}}_{n|n})[\mathbf{c}_n - \hat{\mathbf{c}}_{n|n}] + \cdots, \quad (\text{C.3})$$

where  $J_\phi(x)$  is the Jacobian of  $\phi$  evaluated at  $x$ . Recall that if  $\beta(x)$  is a vector valued function consisting of  $k$  scalar-valued functions  $\beta_i(x)$  and  $x$  is an  $N$ -vector, i.e.,

$$\beta(x) = \begin{bmatrix} \beta_1(x) \\ \beta_2(x) \\ \vdots \\ \beta_k(x) \end{bmatrix} \quad \text{and} \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}$$

then  $J_\beta(x) = \partial\beta/\partial x$  denoted the  $k \times N$  Jacobian matrix of  $\beta(x)$  with respect to  $x$ :

$$J_\beta(x) = \frac{\partial\beta}{\partial x} = \begin{bmatrix} \frac{\partial\beta_1}{\partial x_1} & \frac{\partial\beta_1}{\partial x_2} & \cdots & \frac{\partial\beta_1}{\partial x_N} \\ \frac{\partial\beta_2}{\partial x_1} & \frac{\partial\beta_2}{\partial x_2} & \cdots & \frac{\partial\beta_2}{\partial x_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial\beta_k}{\partial x_1} & \frac{\partial\beta_k}{\partial x_2} & \cdots & \frac{\partial\beta_k}{\partial x_N} \end{bmatrix} \quad (\text{C.4})$$

Likewise, we expand  $\gamma$  about the realization  $\hat{\mathbf{c}}_{n|n-1}$

$$\gamma(\mathbf{c}_n) = \gamma(\hat{\mathbf{c}}_{n|n-1}) + J_\gamma(\hat{\mathbf{c}}_{n|n-1})[\mathbf{c}_n - \hat{\mathbf{c}}_{n|n-1}] + \cdots, \quad (\text{C.5})$$

We keep only the first two terms in the expansions. The resulting expressions create first-order approximations of  $\phi$  and  $\gamma$  and provide linear function of  $\mathbf{c}_n$ .

We now have a linearized SSM that is given by:

## C.2. Linearization of NSSM

---

$$\mathbf{c}_{n+1} = \phi(\hat{\mathbf{c}}_{n|n}) + J_\phi(\hat{\mathbf{c}}_{n|n})[\mathbf{c}_n - \hat{\mathbf{c}}_{n|n}] + \mathbf{v}_n, \quad (\text{C.6})$$

$$\mathbf{y}_n = \gamma(\hat{\mathbf{c}}_{n|n-1}) + J_\gamma(\hat{\mathbf{c}}_{n|n-1})[\mathbf{c}_n - \hat{\mathbf{c}}_{n|n-1}] + \mathbf{w}_n, \quad (\text{C.7})$$

and we use this model for implementing the EKF.

Given initial conditions  $\hat{\mathbf{c}}_{0|-1}$  and  $\mathbf{P}_{0|-1}$ , the recursion proceeds as

– Measurement update:

$$\mathbf{K}_n = \mathbf{P}_{n|n-1} J_\gamma^T(\hat{\mathbf{c}}_{n|n-1}) [J_\gamma(\hat{\mathbf{c}}_{n|n-1}) \mathbf{P}_{n|n-1} J_\gamma^T(\hat{\mathbf{c}}_{n|n-1}) + \mathbf{R}_n]^{-1} \quad (\text{C.8})$$

$$\hat{\mathbf{c}}_{n|n} = \hat{\mathbf{c}}_{n|n-1} + \mathbf{K}_n [\mathbf{y}_n - \gamma(\hat{\mathbf{c}}_{n|n-1})] \quad (\text{C.9})$$

$$\mathbf{P}_{n|n} = \mathbf{P}_{n|n-1} + \mathbf{K}_n J_\gamma(\hat{\mathbf{c}}_{n|n-1}) \mathbf{P}_{n|n-1} \quad (\text{C.10})$$

– Time update:

$$\mathbf{P}_{n+1|n} = J_\phi(\hat{\mathbf{c}}_{n|n}) \mathbf{P}_{n|n-1} J_\phi^T(\hat{\mathbf{c}}_{n|n}) + \mathbf{Q}_n \quad (\text{C.11})$$

$$\hat{\mathbf{c}}_{n+1|n} = \phi(\hat{\mathbf{c}}_{n|n}) \quad (\text{C.12})$$

– Time increment. Increment  $n$  and repeat.

In 1996 Bellaire [103] developed an alternative measurement update that can yield much better estimates in general than those provided by the EKF formulation.

If we consider

$$\Psi(\mathbf{c}_n) = \begin{bmatrix} \mathbf{c}_n - \hat{\mathbf{c}}_{n|n-1} \\ \mathbf{y}_n - \gamma(\mathbf{c}_n) \end{bmatrix} \quad \text{and} \quad [\tilde{\mathbf{P}}_{n|n-1}]^{-1} = (\mathbf{P}_{n|n-1})^{-1} + \mu \mathbf{I} \quad (\text{C.13})$$

then, the measurement update equation are:

$$\mathbf{K}_n = \tilde{\mathbf{P}}_n J_\Psi^T(\hat{\mathbf{c}}_{n|n-1}) [J_\Psi^T(\hat{\mathbf{c}}_{n|n-1}) \tilde{\mathbf{P}}_n J_\Psi(\hat{\mathbf{c}}_{n|n-1}) + \mathbf{R}_n]^{-1} \quad (\text{C.14})$$

$$\hat{\mathbf{c}}_{n|n} = \hat{\mathbf{c}}_{n|n-1} + \mathbf{K}_n [\mathbf{y}_n - \gamma(\hat{\mathbf{c}}_{n|n-1})] \quad (\text{C.15})$$

$$\mathbf{P}_{n|n} = [\mathbf{I} - \mathbf{K}_n J_\gamma(\hat{\mathbf{c}}_{n|n-1})] \mathbf{P}_{n|n-1} [\mathbf{I} - \mathbf{K}_n J_\gamma(\hat{\mathbf{c}}_{n|n-1})]^T + \mathbf{K}_n \mathbf{R}_n \mathbf{K}_n^T \quad (\text{C.16})$$

The equations for the time update are the same as in the EKF formulation. We note that the expressions reduce to the EKF measurement update in the case when  $\mu = 0$  ( $\mu$  is Levenberg-Marquardt parameter).



# Appendix D

## A Review of Particle Filtering

Particle filtering is a sequential Monte Carlo methodology where the basic idea is the recursive computation of relevant probability distributions using the concepts of importance sampling and approximation of probability distributions with discrete random measures. A large portion of the theory on sequential signal processing is about signals and systems that are represented by state-space and observation equations, that is, equations of the form

$$\begin{aligned}\mathbf{c}_n &= \mathbf{f}_n(\mathbf{c}_{n-1}, \mathbf{v}_n) \\ \mathbf{y}_n &= \mathbf{g}_n(\mathbf{c}_n, \mathbf{w}_n)\end{aligned}\tag{D.1}$$

where  $\mathbf{y}_n$  is a vector of observations,  $\mathbf{c}_n$  is a state vector,  $\mathbf{g}_n(\cdot)$  is a measurement function,  $\mathbf{f}_n(\cdot)$  is a system transition function,  $\mathbf{v}_n$  and  $\mathbf{w}_n$  are noise vectors, and the subscript  $n$  denotes time index. The first equation is known as state equation, and the second, as measurement equation. The standard assumptions are that the analytical forms of the functions and the distributions of the two noises are known. Based on the observations  $\mathbf{y}_n$  and the assumptions, the objective is to estimate  $\mathbf{c}_n$  recursively.

For scenarios where the models are nonlinear or the noise is non-Gaussian, the particle filtering method has become an important alternative to the extended Kalman filter. With particle filtering, continuous distributions are approximated by discrete random measures, which are composed of weighted particles, where the particles are samples of the unknown states from the state space, and the particle weights are probability masses computed by using Bayes theory. In the implemen-

---

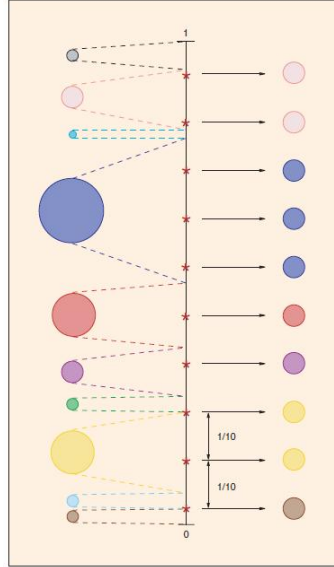


Figure D.1: A schematic description of resampling

tation of particle filtering, importance sampling plays a crucial role and, since the procedure is designed for sequential use, the method is also called sequential importance sampling. The advantage of particle filtering over other methods is in that the exploited approximation does not involve linearizations around current estimates but rather approximations in the representation of the desired distributions by discrete random measures.

The main task of sequential signal processing is the estimation of the state  $\mathbf{c}_n$  recursively from the observations  $\mathbf{y}_n$ . In general, there are three probability distribution functions of interest, and they are the filtering distribution  $p(\mathbf{c}_n|\mathbf{y}_{0:n})$ ; the predictive distribution  $p(\mathbf{c}_{n+l}|\mathbf{y}_{0:n}), l \geq 1$ ; and the smoothing distribution  $p(\mathbf{c}_n|\mathbf{y}_{0:N})$ , where  $N > n$ . All the information about  $\mathbf{c}_n$  regarding filtering, prediction, or smoothing is captured by these distributions, respectively, and so the main goal is their tracking, which is obtaining  $p(\mathbf{c}_n|\mathbf{y}_{0:n})$  from  $p(\mathbf{c}_{n-1}|\mathbf{y}_{0:n-1})$ ,  $p(\mathbf{c}_{n+l}|\mathbf{y}_{0:n})$  from  $p(\mathbf{c}_{n+l-1}|\mathbf{y}_{0:n})$ , or  $p(\mathbf{c}_n|\mathbf{y}_{0:N})$  from  $p(\mathbf{c}_{n+1}|\mathbf{y}_{0:N})$ . The algorithms that exactly track these distributions are known as optimal algorithms. In many practical situations, however, the optimal algorithms are impossible to implement, primarily because the distribution updates require integrations that can not be performed analytically or summations that are impossible to carry out due to the number of terms in the summations.

## D. A Review of Particle Filtering

---

For the joint *a posteriori* distribution of  $\mathbf{c}_0, \mathbf{c}_1, \dots, \mathbf{c}_n$ , in case of independent noise samples which are assumed throughout the article, we can write

$$p(\mathbf{c}_{0:n}|\mathbf{y}_{0:n}) \propto p(\mathbf{c}_0|\mathbf{y}_0) \prod_{k=1}^n p(\mathbf{y}_k|\mathbf{c}_k)p(\mathbf{c}_k|\mathbf{c}_{k-1}). \quad (\text{D.2})$$

It is straightforward to show that a recursive formula for obtaining  $p(\mathbf{c}_{0:n}|\mathbf{y}_{0:n})$  from  $p(\mathbf{c}_{0:n-1}|\mathbf{y}_{0:n-1})$  is given by

$$p(\mathbf{c}_{0:n}|\mathbf{y}_{0:n}) = \frac{p(\mathbf{y}_n|\mathbf{c}_n)p(\mathbf{c}_n|\mathbf{c}_{n-1})}{p(\mathbf{y}_n|\mathbf{y}_{0:n-1})}p(\mathbf{c}_{0:n-1}|\mathbf{y}_{0:n-1}) \quad (\text{D.3})$$

Since the transition from  $p(\mathbf{c}_{0:n-1}|\mathbf{y}_{0:n-1})$  to  $p(\mathbf{c}_{0:n}|\mathbf{y}_{0:n})$  is often analytically intractable, we resort to methods that are based on approximations.

In particle filtering, the distributions are approximated by discrete random measures defined by particles and weights assigned to the particles. If the distribution of interest is  $p(x)$  and its approximating random measure is

$$\chi = \{x^{(m)}, w^{(m)}\}_{m=1}^M \quad (\text{D.4})$$

where  $x^{(m)}$  are the particles,  $w^{(m)}$  are their weights, and  $M$  is the number of particles used in the approximation,  $\chi$  approximates the distribution  $p(x)$  by

$$p(x) = \sum_{m=1}^M w^{(m)}\delta(x - x^{(m)}) \quad (\text{D.5})$$

where  $\delta(\cdot)$  is the Dirac delta function. With this approximation, computations of expectations (which involve complicated integrations) are simplified to summations, that is, for example,

$$E(g(x)) = \int g(x)p(x)dx \quad (\text{D.6})$$

is approximated by

$$E(g(x)) = \sum_{m=1}^M w^{(m)}g(x^{(m)}) \quad (\text{D.7})$$

The next important concept used in particle filtering is the principle of importance sampling. Suppose we want to approximate a distribution  $p(x)$  with a discrete random measure. If we can generate the particles from  $p(x)$ , each of them will be assigned a weight equal to  $1/M$ . When direct sampling from  $p(x)$  is intractable,

## D. A Review of Particle Filtering

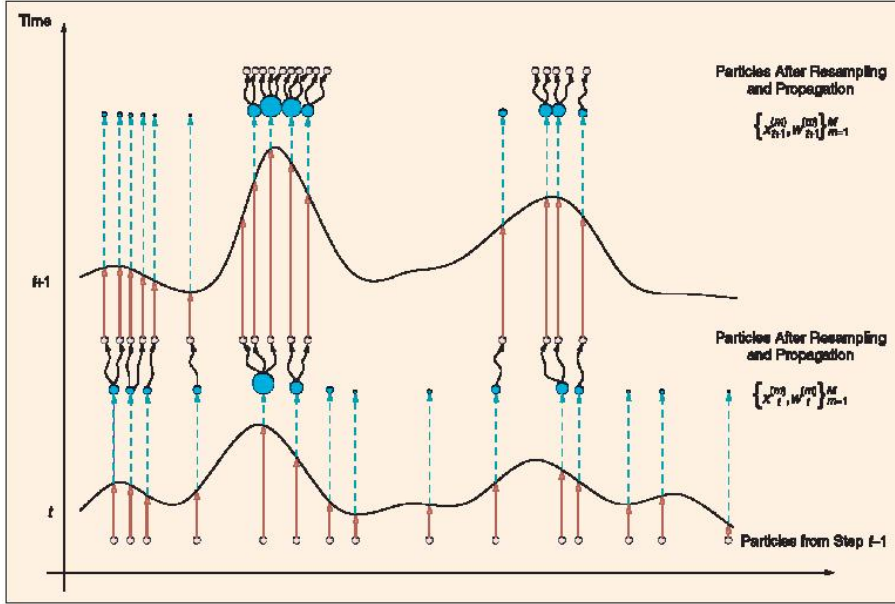


Figure D.2: A pictorial description of particle filtering

one can generate particles  $x^{(m)}$  from a distribution  $\pi(x)$ , known also as importance function, and assign (nonnormalized) weights according to

$$w^{*(m)} = \frac{p(x)}{\pi(x)} \quad (\text{D.8})$$

which upon normalization become

$$w^{(m)} = \frac{w^{*(m)}}{\sum_{i=1}^M w^{*(i)}} \quad (\text{D.9})$$

Suppose now that the posterior distribution  $p(\mathbf{c}_{0:n-1} | \mathbf{y}_{0:n-1})$  is approximated by the discrete random measure  $\mathcal{C}_{n-1} = \{\mathbf{c}_{0:n-1}^{(m)}, w_{n-1}^{(m)}\}_{m=1}^M$ . Note that  $\mathbf{c}_{0:n-1}^{(m)}$  can be considered particles of  $p(\mathbf{c}_{0:n-1} | \mathbf{y}_{0:n-1})$ . Given the discrete random measure  $\mathcal{C}_{n-1}$  and the observation  $\mathbf{y}_n$ , the objective is to exploit  $\mathcal{C}_{n-1}$  in obtaining  $\mathcal{C}_n$ . Sequential importance sampling methods achieve this by generating particles  $\mathbf{c}_n^{(m)}$  and appending them to  $\mathbf{c}_{0:n-1}^{(m)}$  to form  $\mathbf{c}_{0:n}^{(m)}$ , and updating the weights  $w_n^{(m)}$  so that  $\mathcal{C}_n$  allows for accurate estimates of the unknowns of interest at time  $n$ .

If we use an importance function that can be factored as

$$\pi(\mathbf{c}_{0:n} | \mathbf{y}_{0:n}) = \pi(\mathbf{c}_n | \mathbf{c}_{0:n-1}, \mathbf{y}_{0:n}) \pi(\mathbf{c}_{0:n-1} | \mathbf{y}_{0:n-1}) \quad (\text{D.10})$$

and if

$$\mathbf{c}_{0:n-1}^{(m)} \sim \pi(\mathbf{c}_{0:n-1} | \mathbf{y}_{0:n-1}) \quad (\text{D.11})$$

## D. A Review of Particle Filtering

---

and

$$w_{n-1}^{(m)} \propto \frac{p(\mathbf{c}_{0:n-1}^{(m)} | \mathbf{y}_{0:n-1})}{\pi(\mathbf{c}_{0:n-1}^{(m)} | \mathbf{y}_{0:n-1})} \quad (\text{D.12})$$

then we can augment the trajectory  $\mathbf{c}_{0:n-1}^{(m)}$  with  $\mathbf{c}_n^{(m)}$  where

$$\mathbf{c}_n^{(m)} \sim \pi(\mathbf{c}_n | \mathbf{c}_{0:n-1}^{(m)}, \mathbf{y}_{0:n}) \quad (\text{D.13})$$

and easily associate with it an updated weight  $w_n^{(m)}$  obtained according to

$$w_n^{(m)} \propto \frac{p(\mathbf{y}_n | \mathbf{c}_n^{(m)}) p(\mathbf{c}_n^{(m)} | \mathbf{c}_{n-1}^{(m)})}{\pi(\mathbf{c}_n^{(m)} | \mathbf{c}_{0:n-1}^{(m)}, \mathbf{y}_{0:n})} w_{n-1}^{(m)} \quad (\text{D.14})$$

The sequential importance sampling algorithm can thus be implemented by performing the following two steps for every  $n$ :

1. Draw particles  $\mathbf{c}_n^{(m)} \sim \pi(\mathbf{c}_n | \mathbf{c}_{0:n-1}^{(m)}, \mathbf{y}_{0:n})$ , where  $m = 1, 2, \dots, M$ .
2. Compute the weights of  $w_n^{(m)}$  according to (D.14).

The importance function plays a very important role in the performance of the particle filter. This function must have the same support as the probability distribution that is being approximated. In general, the closer the importance function to that distribution, the better the approximation is. In the literature, the two most frequently used importance functions are the prior and the optimal importance function. The prior importance function is given by  $p(\mathbf{c}_n | \mathbf{c}_{n-1}^{(m)})$ , and it implies particle weight updates by

$$w_n^{(m)} \propto w_{n-1}^{(m)} p(\mathbf{y}_n | \mathbf{c}_n^{(m)}). \quad (\text{D.15})$$

The optimal importance function minimizes the variance of the importance weights conditional on the trajectory  $\mathbf{c}_{0:n-1}^{(m)}$  and the observations  $\mathbf{y}_{0:n}$  and is given by  $p(\mathbf{c}_n | \mathbf{c}_{0:n-1}^{(m)}, \mathbf{y}_{0:n})$  [104]. When the optimal function is used, the update of the weights is carried out according to

$$w_n^{(m)} \propto w_{n-1}^{(m)} p(\mathbf{y}_n | \mathbf{c}_{n-1}^{(m)}). \quad (\text{D.16})$$

Note that implementations of particle filters with prior importance functions are much easier than those with optimal importance functions. The reason is that the computation of  $p(\mathbf{y}_n | \mathbf{c}_{n-1}^{(m)})$  requires integration.

A major problem with particle filtering is that the discrete random measure degenerates quickly. In other words, all the particles except for a very few are



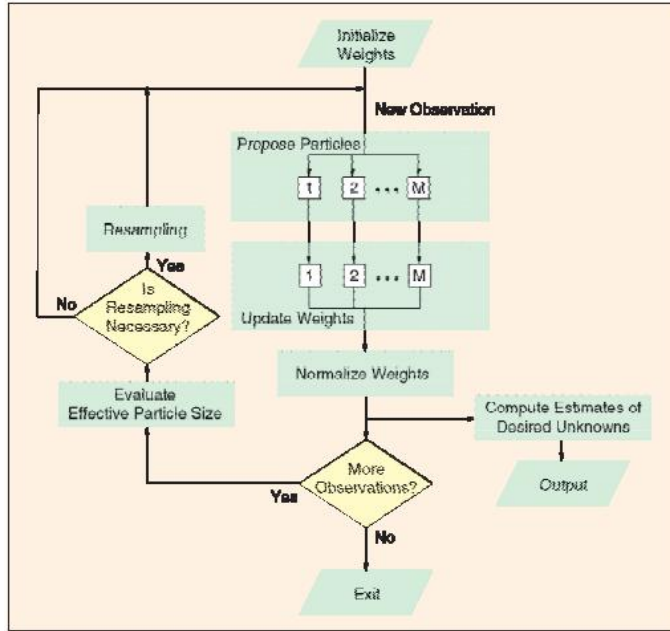


Figure D.3: A block diagram of particle filtering

assigned negligible weights. The degeneracy implies that the performance of the particle filter will deteriorate. Degeneracy, however, can be reduced by using good importance sampling functions and resampling.

Resampling is a scheme that eliminates particles with small weights and replicates particles with large weights. In principle, it is implemented as follows:

1. Draw  $M$  particles,  $\mathbf{c}_n^{*(m)}$  from the discrete distribution  $\mathcal{C}_n$
2. Let  $\mathbf{c}_n^{(m)} = \mathbf{c}_n^{*(m)}$  and assign equal weights ( $1/M$ ) to the particles.

The idea of resampling is depicted in Figure D.1 with  $M = 10$  particles. There, the left column of circles represents particles before resampling, where the diameters of the circles are proportional to the weights of the particles. The right column of circles are the particles after resampling. In general the large particles are replicated and the small particles are removed. For example, the "blue" particle with the largest weight is replicated three times and the "yellow" particle, two times, whereas the green particles, which have small weights, are removed. Also, after resampling all the circles have equal diameters, that is, all the weights are set to  $1/M$ . In Figure D.2, we represent pictorially the random measures and the actual probability distributions of interest as well as the three steps of particle filtering: particle

## D. A Review of Particle Filtering

---

generation, weight update, and resampling. In the figure, the solid curves represent the distributions of interest, which are approximated by the discrete measures. The sizes of the particles reflect the weights that are assigned to them.

Finally in Figure D.3, we display a flowchart that summarizes the particle filtering algorithm. At time  $n$ , a new set of particles is generated, and their weights are computed. Thereby we obtain the random measure  $\mathcal{C}_n$ , which can be used for estimation of the desired unknowns. Before we proceed with the generation of the set of particles for time instant  $n+1$ , we estimate the effective particle size (a metric that measures the degeneracy of the particles [105] [106]). If the effective particle size is below a predefined threshold, resampling takes place; otherwise we proceed with the regular steps of new particle generation and weight computation [98].



# Appendix E

## Proofs of The Main Equations. See Chapter 4.

$$p_n(\mathbf{t}', \mathbf{t}, \mathbf{x}', \mathbf{x}) = \frac{\Lambda_n^{(a)}(\mathbf{t}', \mathbf{t}, \mathbf{x}', \mathbf{x}) \Lambda_n^{(b)}(\mathbf{t}, \mathbf{x})}{p(y_{1:N} | \theta'_N)}$$

where:

$$\Lambda_n^{(a)}(\mathbf{t}', \mathbf{t}, \mathbf{x}', \mathbf{x}) = \int_{C^{n-2}} p(y_{0:n} | \mathbf{c}_{0:n-2}^{(1)}, \mathbf{c}_{0:n-2}^{(2)}, \mathbf{c}_{n-1}^{(1)} = \mathbf{t}', \mathbf{c}_n^{(1)} = \mathbf{t}, \mathbf{c}_{n-1}^{(2)} = \mathbf{x}', \mathbf{c}_n^{(2)} = \mathbf{x}, \theta'_N) \\ p(\mathbf{c}_{0:n-2}^{(1)}, \mathbf{c}_{0:n-2}^{(2)}, \mathbf{c}_{n-1}^{(1)} = \mathbf{t}', \mathbf{c}_n^{(1)} = \mathbf{t}, \mathbf{c}_{n-1}^{(2)} = \mathbf{x}', \mathbf{c}_n^{(2)} = \mathbf{x} | \theta'_N) d\mathbf{c}_{0:n-2}^{(1)} d\mathbf{c}_{0:n-2}^{(2)}$$

$$\Lambda_n^{(b)}(\mathbf{t}, \mathbf{x}) = \int_{C^{N-n}} p(y_{n+1:N} | \mathbf{c}_{n+1:N}^{(1)}, \mathbf{c}_{n+1:N}^{(2)}, \theta'_N) p(\mathbf{c}_{n+1:N}^{(1)} | \mathbf{c}_n^{(1)} = \mathbf{t}, \theta'_N) \\ p(\mathbf{c}_{n+1:N}^{(2)} | \mathbf{c}_n^{(2)} = \mathbf{x}, \theta'_N) d\mathbf{c}_{n+1:N}^{(1)} d\mathbf{c}_{n+1:N}^{(2)}$$

with calculation of recursion of  $\Lambda_n^{(a)}(\mathbf{t}', \mathbf{t}, \mathbf{x}', \mathbf{x})$  and  $\Lambda_n^{(b)}(\mathbf{t}, \mathbf{x})$  and regarding to point that all density figuring in the preceding relations are normal so that they reduce conveniently to forward-Backward recursions on first and second conditional moments.

---

## E.1 Recall:

$$\begin{aligned}
\int_{\mathbb{C}^{M+1}} \exp[-\mathbf{x}^\dagger \mathbf{M} \mathbf{x} + 2\text{Re}(\mathbf{x}^\dagger \varphi)] d\mathbf{x} &= \left[ \frac{(\pi)^{M+1}}{|\mathbf{M}|} \right] e^{\varphi^\dagger \mathbf{M}^{-1} \varphi} = c \\
\frac{1}{c} \int_{\mathbb{C}^{M+1}} \mathbf{x} \cdot \exp[-\mathbf{x}^\dagger \mathbf{M} \mathbf{x} + 2\text{Re}(\mathbf{x}^\dagger \varphi)] d\mathbf{x} &= \mathbf{M}^{-1} \varphi \\
\int_{\mathbb{C}^{M+1}} \mathbf{x} \mathbf{x}^\dagger \cdot \exp[-\mathbf{x}^\dagger \mathbf{M} \mathbf{x} + 2\text{Re}(\mathbf{x}^\dagger \varphi)] d\mathbf{x} &= \mathbf{M}^{-1} + \mathbf{M}^{-1} \varphi \varphi^\dagger \mathbf{M}^{-1} \quad (\text{E.1})
\end{aligned}$$

## E.2 Recursion on $\Lambda_n^{(a)}(\mathbf{t}', \mathbf{t}, \mathbf{x}', \mathbf{x})$

For  $n = 0$  we have:

$$\begin{aligned}
\Lambda_0^{(a)}(\mathbf{t}, \mathbf{x}) &= p(y_0 | \mathbf{c}_0^{(\alpha'_0)}, \mathbf{d}'_0) p(\mathbf{c}_0^{(1)} = \mathbf{t} | \theta'_N) p(\mathbf{c}_0^{(2)} = \mathbf{x} | \theta'_N) \\
&= \gamma_0^{(a)} \exp[-\mathbf{t}^\dagger \Gamma_0^{(t)} \mathbf{t} + 2\text{Re}(\mathbf{t}^\dagger \varphi_0^{(t)}) - \mathbf{x}^\dagger \Gamma_0^{(x)} \mathbf{x} + 2\text{Re}(\mathbf{x}^\dagger \varphi_0^{(x)})] \quad (\text{E.2})
\end{aligned}$$

if  $\alpha'_0 = 1$ :

$$\begin{aligned}
p(\mathbf{c}_0^{(1)} | \theta'_N) &= \mathcal{N}_c(\mathbf{c}_0^{(1)}, \hat{\mathbf{c}}_0^{(1)}, \mathbf{P}_0^{(1)}) \\
\varphi_0^{(t)} &= \mathbf{d}'_0{}^* \beta^{-1} y_0 + \mathbf{P}_0^{(1)-1} \hat{\mathbf{c}}_0^{(1)} \\
\varphi_0^{(x)} &= \mathbf{P}_0^{(2)-1} \hat{\mathbf{c}}_0^{(2)} \\
\Gamma_0^{(t)} &= \mathbf{d}'_0{}^* \beta^{-1} \mathbf{d}'_0{}^T + \mathbf{P}_0^{(1)-1} \\
\Gamma_0^{(x)} &= \mathbf{P}_0^{(2)-1}
\end{aligned}$$

if  $\alpha'_0 = 2$ :

$$\begin{aligned}
p(\mathbf{c}_0^{(2)} | \theta'_N) &= \mathcal{N}_c(\mathbf{c}_0^{(2)}, \hat{\mathbf{c}}_0^{(2)}, \mathbf{P}_0^{(2)}) \\
\varphi_0^{(t)} &= \mathbf{P}_0^{(1)-1} \hat{\mathbf{c}}_0^{(1)} \\
\varphi_0^{(x)} &= \mathbf{d}'_0{}^* \beta^{-1} y_0 + \mathbf{P}_0^{(2)-1} \hat{\mathbf{c}}_0^{(2)} \\
\Gamma_0^{(t)} &= \mathbf{P}_0^{(1)-1} \\
\Gamma_0^{(x)} &= \mathbf{d}'_0{}^* \beta^{-1} \mathbf{d}'_0{}^T + \mathbf{P}_0^{(2)-1}
\end{aligned}$$

## E.2. Recursion on $\Lambda_n^{(a)}(\mathbf{t}', \mathbf{t}, \mathbf{x}', \mathbf{x})$

---

For  $n = 1$  we have:

$$\begin{aligned} \Lambda_1^{(a)}(\mathbf{t}', \mathbf{t}, \mathbf{x}', \mathbf{x}) &= \Lambda_0^{(a)}(\mathbf{t}', \mathbf{x}') p(y_1 | \mathbf{c}_1^{(\alpha'_1)}, \mathbf{d}'_1) \\ &= p(\mathbf{c}_1^{(1)} = \mathbf{t} | \mathbf{c}_0^{(1)} = \mathbf{t}', \theta'_N) p(\mathbf{c}_1^{(2)} = \mathbf{x} | \mathbf{c}_0^{(2)} = \mathbf{x}', \theta'_N) = \\ &= \gamma_1^{(a)} \exp[-\mathbf{z}^\dagger \Gamma_1^{(z)} \mathbf{z} + 2Re(\mathbf{z}^\dagger \varphi_1^{(z)}) - \mathbf{l}^\dagger \Gamma_1^{(l)} \mathbf{l} + 2Re(\mathbf{l}^\dagger \varphi_1^{(l)})] \quad (\text{E.3}) \end{aligned}$$

where

$$\begin{aligned} \mathbf{z}^\dagger &= [\mathbf{t}'^\dagger, \mathbf{t}^\dagger] \\ \mathbf{l}^\dagger &= [\mathbf{x}'^\dagger, \mathbf{x}^\dagger] \\ \varphi_1^{(z)\dagger} &= [\varphi_{1,1}^{(z)\dagger}, \varphi_{1,2}^{(z)\dagger}] \\ \varphi_1^{(l)\dagger} &= [\varphi_{1,1}^{(l)\dagger}, \varphi_{1,2}^{(l)\dagger}] \\ \Gamma_1^{(z)} &= \begin{pmatrix} \Gamma_1^{(z)}(11) & \Gamma_1^{(z)}(12) \\ \Gamma_1^{(z)\dagger}(12) & \Gamma_1^{(z)}(22) \end{pmatrix} \\ \Gamma_1^{(l)} &= \begin{pmatrix} \Gamma_1^{(l)}(11) & \Gamma_1^{(l)}(12) \\ \Gamma_1^{(l)\dagger}(12) & \Gamma_1^{(l)}(22) \end{pmatrix} \end{aligned}$$

if  $\alpha'_1 = 1$ :

$$\begin{aligned} \varphi_{1,1}^{(z)} &= \varphi_0^{(t)} \\ \varphi_{1,2}^{(z)} &= \mathbf{d}'_1^* \beta^{-1} y_1 \\ \Gamma_1^{(z)}(11) &= \Gamma_0^{(t)} + \mathbf{F}^\dagger(1) \mathbf{Q}^{-1}(1) \mathbf{F}(1) \\ \Gamma_1^{(z)}(12) &= -\mathbf{F}^\dagger(1) \mathbf{Q}^{-1}(1) \\ \Gamma_1^{(z)}(22) &= \mathbf{d}'_1^* \beta^{-1} \mathbf{d}'_1^T + \mathbf{Q}^{-1}(1) \end{aligned}$$

$$\begin{aligned} \varphi_{1,1}^{(l)} &= \varphi_0^{(x)} \\ \varphi_{1,2}^{(l)} &= 0 \\ \Gamma_1^{(l)}(11) &= \Gamma_0^{(x)} + \mathbf{F}^\dagger(2) \mathbf{Q}^{-1}(2) \mathbf{F}(2) \\ \Gamma_1^{(l)}(12) &= -\mathbf{F}^\dagger(2) \mathbf{Q}^{-1}(2) \\ \Gamma_1^{(l)}(22) &= \mathbf{Q}^{-1}(2) \end{aligned}$$

---

**E. Proofs of The Main Equations. See Chapter 4.**

---

if  $\alpha'_1 = 2$ :

$$\begin{aligned}\varphi_{1,1}^{(z)} &= \varphi_0^{(t)} \\ \varphi_{1,2}^{(z)} &= 0 \\ \Gamma_1^{(z)}(11) &= \Gamma_0^{(t)} + \mathbf{F}^\dagger(1)\mathbf{Q}^{-1}(1)\mathbf{F}(1) \\ \Gamma_1^{(z)}(12) &= -\mathbf{F}^\dagger(1)\mathbf{Q}^{-1}(1) \\ \Gamma_1^{(z)}(22) &= \mathbf{Q}^{-1}(1)\end{aligned}$$

$$\begin{aligned}\varphi_{1,1}^{(l)} &= \varphi_0^{(x)} \\ \varphi_{1,2}^{(l)} &= \mathbf{d}'_1^* \beta^{-1} y_1 \\ \Gamma_1^{(l)}(11) &= \Gamma_0^{(x)} + \mathbf{F}^\dagger(2)\mathbf{Q}^{-1}(2)\mathbf{F}(2) \\ \Gamma_1^{(l)}(12) &= -\mathbf{F}^\dagger(2)\mathbf{Q}^{-1}(2) \\ \Gamma_1^{(l)}(22) &= \mathbf{d}'_1^* \beta^{-1} \mathbf{d}'_1{}^T + \mathbf{Q}^{-1}(2)\end{aligned}$$

For  $2 \leq n \leq N$

$$\begin{aligned}\Lambda_n^{(a)}(\mathbf{t}', \mathbf{t}, \mathbf{x}', \mathbf{x}) &= p(y_n | \mathbf{c}_n^{(\alpha'_n)}, \mathbf{d}'_n) p(\mathbf{c}_n^{(1)} = \mathbf{t} | \mathbf{c}_{n-1}^{(1)} = \mathbf{t}', \theta'_N) p(\mathbf{c}_n^{(2)} = \mathbf{x} | \mathbf{c}_{n-1}^{(2)} = \mathbf{x}', \theta'_N) \\ &\int_C dt'' d\mathbf{x}'' \int_{C^{n-3}} p(y_{0:n-1} | \mathbf{c}_{0:n-3}^{(1)}, \mathbf{c}_{0:n-3}^{(2)}, \mathbf{c}_{n-2}^{(1)} = \mathbf{t}'', \mathbf{c}_{n-1}^{(1)} = \mathbf{t}', \mathbf{c}_{n-2}^{(2)} = \mathbf{x}'', \mathbf{c}_{n-1}^{(2)} = \mathbf{x}', \theta'_N) \\ &p(\mathbf{c}_{0:n-3}^{(1)}, \mathbf{c}_{0:n-3}^{(2)}, \mathbf{c}_{n-2}^{(1)} = \mathbf{t}'', \mathbf{c}_{n-1}^{(1)} = \mathbf{t}', \mathbf{c}_{n-2}^{(2)} = \mathbf{x}'', \mathbf{c}_{n-1}^{(2)} = \mathbf{x}' | \theta'_N) d\mathbf{c}_{0:n-3}^{(1)} d\mathbf{c}_{0:n-3}^{(2)}\end{aligned}$$

so that:

$$\begin{aligned}\Lambda_n^{(a)}(\mathbf{t}', \mathbf{t}, \mathbf{x}', \mathbf{x}) &= p(y_n | \mathbf{c}_n^{(\alpha'_n)}, \mathbf{d}'_n) p(\mathbf{c}_n^{(1)} = \mathbf{t} | \mathbf{c}_{n-1}^{(1)} = \mathbf{t}', \theta'_N) p(\mathbf{c}_n^{(2)} = \mathbf{x} | \mathbf{c}_{n-1}^{(2)} = \mathbf{x}', \theta'_N) \\ &\times \int_C \Lambda_{n-1}^{(a)}(\mathbf{t}'', \mathbf{t}', \mathbf{x}'', \mathbf{x}') dt'' d\mathbf{x}'' = \\ &\gamma_n^{(a)} \exp[-\mathbf{z}^\dagger \Gamma_n^{(z)} \mathbf{z} + 2\text{Re}(\mathbf{z}^\dagger \varphi_n^{(z)}) - \mathbf{l}^\dagger \Gamma_n^{(l)} \mathbf{l} + 2\text{Re}(\mathbf{l}^\dagger \varphi_n^{(l)})] \quad (\text{E.4})\end{aligned}$$

where

$$\begin{aligned}\mathbf{z}^\dagger &= [\mathbf{t}'^\dagger, \mathbf{t}^\dagger] \\ \mathbf{l}^\dagger &= [\mathbf{x}'^\dagger, \mathbf{x}^\dagger] \\ \varphi_n^{(z)\dagger} &= [\varphi_{n,1}^{(z)\dagger}, \varphi_{n,2}^{(z)\dagger}] \\ \varphi_n^{(l)\dagger} &= [\varphi_{n,1}^{(l)\dagger}, \varphi_{n,2}^{(l)\dagger}] f\end{aligned}$$

## E.2. Recursion on $\Lambda_n^{(a)}(\mathbf{t}', \mathbf{t}, \mathbf{x}', \mathbf{x})$

---

$$\mathbf{\Gamma}_n^{(z)} = \begin{pmatrix} \mathbf{\Gamma}_n^{(z)}(11) & \mathbf{\Gamma}_n^{(z)}(12) \\ \mathbf{\Gamma}_n^{(z)\dagger}(12) & \mathbf{\Gamma}_n^{(z)}(22) \end{pmatrix}$$

$$\mathbf{\Gamma}_n^{(l)} = \begin{pmatrix} \mathbf{\Gamma}_n^{(l)}(11) & \mathbf{\Gamma}_n^{(l)}(12) \\ \mathbf{\Gamma}_n^{(l)\dagger}(12) & \mathbf{\Gamma}_n^{(l)}(22) \end{pmatrix}$$

if  $\alpha'_n = 1$ :

$$\begin{aligned} \varphi_{n,1}^{(z)} &= \varphi_{n-1,2}^{(z)} - \mathbf{\Gamma}_{n-1}^{(z)\dagger}(12)\mathbf{\Gamma}_{n-1}^{(z)-1}(11)\varphi_{n-1,1}^{(z)} \\ \varphi_{n,2}^{(z)} &= \mathbf{d}'_n \beta^{-1} y_n \\ \mathbf{\Gamma}_n^{(z)}(11) &= \mathbf{\Gamma}_{n-1}^{(z)}(22) - \mathbf{\Gamma}_{n-1}^{(z)\dagger}(12)\mathbf{\Gamma}_{n-1}^{(z)-1}(11)\mathbf{\Gamma}_{n-1}^{(z)}(12) + \mathbf{F}^\dagger(1)\mathbf{Q}^{-1}(1)\mathbf{F}(1) \\ \mathbf{\Gamma}_n^{(z)}(12) &= -\mathbf{F}^\dagger(1)\mathbf{Q}^{-1}(1) \\ \mathbf{\Gamma}_n^{(z)}(22) &= \mathbf{d}'_n \beta^{-1} \mathbf{d}'_n{}^T + \mathbf{Q}^{-1}(1) \end{aligned}$$

$$\begin{aligned} \varphi_{n,1}^{(l)} &= \varphi_{n-1,2}^{(l)} - \mathbf{\Gamma}_{n-1}^{(l)\dagger}(12)\mathbf{\Gamma}_{n-1}^{(l)-1}(11)\varphi_{n-1,1}^{(l)} \\ \varphi_{n,2}^{(l)} &= 0 \\ \mathbf{\Gamma}_n^{(l)}(11) &= \mathbf{\Gamma}_{n-1}^{(l)}(22) - \mathbf{\Gamma}_{n-1}^{(l)\dagger}(12)\mathbf{\Gamma}_{n-1}^{(l)-1}(11)\mathbf{\Gamma}_{n-1}^{(l)}(12) + \mathbf{F}^\dagger(2)\mathbf{Q}^{-1}(2)\mathbf{F}(2) \\ \mathbf{\Gamma}_n^{(l)}(12) &= -\mathbf{F}^\dagger(2)\mathbf{Q}^{-1}(2) \\ \mathbf{\Gamma}_n^{(l)}(22) &= \mathbf{Q}^{-1}(2) \end{aligned}$$

if  $\alpha'_n = 2$ :

$$\begin{aligned} \varphi_{n,1}^{(z)} &= \varphi_{n-1,2}^{(z)} - \mathbf{\Gamma}_{n-1}^{(z)\dagger}(12)\mathbf{\Gamma}_{n-1}^{(z)-1}(11)\varphi_{n-1,1}^{(z)} \\ \varphi_{n,2}^{(z)} &= 0 \\ \mathbf{\Gamma}_n^{(z)}(11) &= \mathbf{\Gamma}_{n-1}^{(z)}(22) - \mathbf{\Gamma}_{n-1}^{(z)\dagger}(12)\mathbf{\Gamma}_{n-1}^{(z)-1}(11)\mathbf{\Gamma}_{n-1}^{(z)}(12) + \mathbf{F}^\dagger(1)\mathbf{Q}^{-1}(1)\mathbf{F}(1) \\ \mathbf{\Gamma}_n^{(z)}(12) &= -\mathbf{F}^\dagger(1)\mathbf{Q}^{-1}(1) \\ \mathbf{\Gamma}_n^{(z)}(22) &= \mathbf{Q}^{-1}(1) \end{aligned}$$

$$\begin{aligned} \varphi_{n,1}^{(l)} &= \varphi_{n-1,2}^{(l)} - \mathbf{\Gamma}_{n-1}^{(l)\dagger}(12)\mathbf{\Gamma}_{n-1}^{(l)-1}(11)\varphi_{n-1,1}^{(l)} \\ \varphi_{n,2}^{(l)} &= \mathbf{d}'_n \beta^{-1} y_n \\ \mathbf{\Gamma}_n^{(l)}(11) &= \mathbf{\Gamma}_{n-1}^{(l)}(22) - \mathbf{\Gamma}_{n-1}^{(l)\dagger}(12)\mathbf{\Gamma}_{n-1}^{(l)-1}(11)\mathbf{\Gamma}_{n-1}^{(l)}(12) + \mathbf{F}^\dagger(2)\mathbf{Q}^{-1}(2)\mathbf{F}(2) \\ \mathbf{\Gamma}_n^{(l)}(12) &= -\mathbf{F}^\dagger(2)\mathbf{Q}^{-1}(2) \\ \mathbf{\Gamma}_n^{(l)}(22) &= \mathbf{d}'_n \beta^{-1} \mathbf{d}'_n{}^T + \mathbf{Q}^{-1}(2) \end{aligned}$$



### E.3 Recursion on $\Lambda_n^{(b)}(\mathbf{x})$

$$\begin{aligned}\Lambda_N^{(b)}(\mathbf{t}, \mathbf{x}) &\equiv 1 \\ p(\mathbf{c}_{n+1:N}^{(1)} | \mathbf{c}_n^{(1)} = \mathbf{t}, \theta'_N) &= p(\mathbf{c}_{n+2:N}^{(1)} | \mathbf{c}_{n+1}^{(1)} = \mathbf{t}', \theta'_N) p(\mathbf{c}_{n+1}^{(1)} = \mathbf{t}' | \mathbf{c}_n^{(1)} = \mathbf{t}, \theta'_N) \\ p(\mathbf{c}_{n+1:N}^{(2)} | \mathbf{c}_n^{(2)} = \mathbf{x}, \theta'_N) &= p(\mathbf{c}_{n+2:N}^{(2)} | \mathbf{c}_{n+1}^{(2)} = \mathbf{x}', \theta'_N) p(\mathbf{c}_{n+1}^{(2)} = \mathbf{x}' | \mathbf{c}_n^{(2)} = \mathbf{x}, \theta'_N)\end{aligned}$$

and so for  $0 \leq n \leq N - 1$ :

$$\begin{aligned}\Lambda_n^{(b)}(\mathbf{t}, \mathbf{x}) &= \int_C p(y_{n+1} | \mathbf{c}_{n+1}^{(\alpha'_{n+1})}, d'_{n+1}) p(\mathbf{c}_{n+1}^{(1)} = \mathbf{t}' | \mathbf{c}_n^{(1)} = \mathbf{t}, \theta'_N) \\ &\quad p(\mathbf{c}_{n+1}^{(2)} = \mathbf{x}' | \mathbf{c}_n^{(2)} = \mathbf{x}, \theta'_N) \Lambda_{n+1}^{(b)}(\mathbf{t}', \mathbf{x}') dt' dx' \\ &= \gamma_n^{(b)} \exp[-\mathbf{t}^\dagger \Gamma_{n,1}^{(b)} \mathbf{t} + 2Re(\mathbf{t}^\dagger \varphi_{n,1}^{(b)}) - \mathbf{x}^\dagger \Gamma_{n,2}^{(b)} \mathbf{x} + 2Re(\mathbf{x}^\dagger \varphi_{n,2}^{(b)})] \quad (\text{E.5})\end{aligned}$$

if  $\alpha'_{n+1} = 1$

$$\begin{aligned}\varphi_{n,1}^{(b)} &= \mathbf{F}^\dagger(1) \mathbf{Q}^{-1}(1) \mathbf{R}_{n+1}^{-1} (\varphi_{n+1,1}^{(b)} + \mathbf{d}'_{n+1}^* \beta^{-1} y_{n+1}) \\ \varphi_{n,2}^{(b)} &= \mathbf{F}^\dagger(2) \mathbf{Q}^{-1}(2) \mathbf{H}_{n+1}^{-1} \varphi_{n+1,2}^{(b)} \\ \Gamma_{n,1}^{(b)} &= \mathbf{F}^\dagger(1) \mathbf{Q}^{-1}(1) \mathbf{F}(1) - \mathbf{F}^\dagger(1) \mathbf{Q}^{-1}(1) \mathbf{R}_{n+1}^{-1} \mathbf{Q}^{-1}(1) \mathbf{F}(1) \\ \Gamma_{n,2}^{(b)} &= \mathbf{F}^\dagger(2) \mathbf{Q}^{-1}(2) \mathbf{F}(2) - \mathbf{F}^\dagger(2) \mathbf{Q}^{-1}(2) \mathbf{H}_{n+1}^{-1} \mathbf{Q}^{-1}(2) \mathbf{F}(2)\end{aligned}$$

where

$$\begin{aligned}\mathbf{R}_{n+1} &= \mathbf{d}'_{n+1}^* \beta^{-1} \mathbf{d}'_{n+1}^T + \mathbf{Q}^{-1}(1) + \Gamma_{n+1,1}^{(b)} \\ \mathbf{H}_{n+1} &= \mathbf{Q}^{-1}(2) + \Gamma_{n+1,2}^{(b)}\end{aligned}$$

if  $\alpha'_{n+1} = 2$

$$\begin{aligned}\varphi_{n,1}^{(b)} &= \mathbf{F}^\dagger(1) \mathbf{Q}^{-1}(1) \mathbf{R}_{n+1}^{-1} \varphi_{n+1,1}^{(b)} \\ \varphi_{n,2}^{(b)} &= \mathbf{F}^\dagger(2) \mathbf{Q}^{-1}(2) \mathbf{H}_{n+1}^{-1} (\varphi_{n+1,2}^{(b)} + \mathbf{d}'_{n+1}^* \beta^{-1} y_{n+1}) \\ \Gamma_{n,1}^{(b)} &= \mathbf{F}^\dagger(1) \mathbf{Q}^{-1}(1) \mathbf{F}(1) - \mathbf{F}^\dagger(1) \mathbf{Q}^{-1}(1) \mathbf{R}_{n+1}^{-1} \mathbf{Q}^{-1}(1) \mathbf{F}(1) \\ \Gamma_{n,2}^{(b)} &= \mathbf{F}^\dagger(2) \mathbf{Q}^{-1}(2) \mathbf{F}(2) - \mathbf{F}^\dagger(2) \mathbf{Q}^{-1}(2) \mathbf{H}_{n+1}^{-1} \mathbf{Q}^{-1}(2) \mathbf{F}(2)\end{aligned}$$

where

$$\begin{aligned}\mathbf{R}_{n+1} &= \mathbf{Q}^{-1}(1) + \Gamma_{n+1,1}^{(b)} \\ \mathbf{H}_{n+1} &= \mathbf{d}'_{n+1}^* \beta^{-1} \mathbf{d}'_{n+1}^T + \mathbf{Q}^{-1}(2) + \Gamma_{n+1,2}^{(b)}\end{aligned}$$

#### E.4. Computing densities $p_n(\mathbf{t}', \mathbf{t}, \mathbf{x}', \mathbf{x})$

---

for ( $n = N$ )

$$\begin{aligned}\varphi_{N,1}^{(b)} &= \varphi_{N,2}^{(b)} = 0 \\ \Gamma_{N,1}^{(b)} &= \Gamma_{N,2}^{(b)} = 0\end{aligned}$$

#### E.4 Computing densities $p_n(\mathbf{t}', \mathbf{t}, \mathbf{x}', \mathbf{x})$

By using results that are presented in previous section, we have:

for  $n = 1 : N$

$$\begin{aligned}p_n(\mathbf{t}', \mathbf{t}, \mathbf{x}', \mathbf{x}) &= \frac{\gamma_n^{(a)} \gamma_n^{(b)}}{p(y_{1:N} | \theta'_N)} \exp[-\mathbf{z}^\dagger \Gamma_n^{(zb)} \mathbf{z} + 2\text{Re}(\mathbf{z}^\dagger \varphi_n^{(zb)}) \\ &\quad - \mathbf{1}^\dagger \Gamma_n^{(lb)} \mathbf{1} + 2\text{Re}(\mathbf{1}^\dagger \varphi_n^{(lb)})] \quad (\text{E.6})\end{aligned}$$

where

$$\begin{aligned}\mathbf{z}^\dagger &= [\mathbf{t}'^\dagger, \mathbf{t}^\dagger] \\ \mathbf{1}^\dagger &= [\mathbf{x}'^\dagger, \mathbf{x}^\dagger] \\ \varphi_n^{(zb)\dagger} &= [\varphi_{n,1}^{(z)\dagger}, \varphi_{n,2}^{(z)\dagger} + \varphi_{n,1}^{(b)\dagger}] \\ \varphi_n^{(lb)\dagger} &= [\varphi_{n,1}^{(l)\dagger}, \varphi_{n,2}^{(l)\dagger} + \varphi_{n,2}^{(b)\dagger}] \\ \Gamma_n^{(zb)} &= \begin{pmatrix} \Gamma_n^{(z)}(11) & \Gamma_n^{(z)}(12) \\ \Gamma_n^{(z)\dagger}(12) & \Gamma_n^{(z)}(22) + \Gamma_{n,1}^{(b)} \end{pmatrix} \\ \Gamma_n^{(lb)} &= \begin{pmatrix} \Gamma_n^{(l)}(11) & \Gamma_n^{(l)}(12) \\ \Gamma_n^{(l)\dagger}(12) & \Gamma_n^{(l)}(22) + \Gamma_{n,2}^{(b)} \end{pmatrix}\end{aligned}$$

for  $n = 0$

$$\begin{aligned}p_n(\mathbf{t}, \mathbf{x}) &= \frac{\gamma_n^{(a)} \gamma_n^{(b)}}{p(y_{1:N} | \theta'_N)} \exp[-\mathbf{t}^\dagger (\Gamma_n^{(t)} + \Gamma_{n,1}^{(b)}) \mathbf{t} + 2\text{Re}(\mathbf{t}^\dagger (\varphi_n^{(t)} + \varphi_{n,1}^{(b)})) - \mathbf{x}^\dagger (\Gamma_n^{(x)} + \Gamma_{n,2}^{(b)}) \mathbf{x} \\ &\quad + 2\text{Re}(\mathbf{x}^\dagger (\varphi_n^{(x)} + \varphi_{n,2}^{(b)}))] \quad (\text{E.7})\end{aligned}$$

Note that the normalizing quantities  $\gamma_n^{(a)}, \gamma_n^{(b)}$  and  $p(y_{1:N} | \theta'_N)$ , do not need to be iterated, as they do not enter the recursion on quantities of interest via first and second moments.

**E. Proofs of The Main Equations. See Chapter 4.**

---

# Bibliography

- [1] J. G. Proakis, *Digital Communications*. McGraw-Hill, 5th edition, 2007.
  - [2] J. D. Parsons, *The Mobile Radio Propagation Channel*. Pentech press, 1992.
  - [3] G. L. Stuber, *Principles of Mobile Communications*. Norwell, Mass, USA: Kluwer Academic Publishers, 1999.
  - [4] W. C. Jakes, *Microwave Mobile Communications*. Wiley, 1974.
  - [5] S. U. H. Qureshi, *Adaptive Equalization*. chapt.12 in *Advanced Digital Communications: Systems and Signal Processing Techniques*, editor K. Feher, Prentice-Hall, 1987.
  - [6] E. A. Lee and D. G. Messerschmitt, *Digital Communications*. Kluwer, second edition, 1994.
  - [7] H. Meyr, M. Moeneclaey, and S. A. Fechtel, *Digital Communication Receivers: Synchronization, Channel Estimation, and Signal Processing*. Wiley, 1998.
  - [8] K. Pahlavan and A. H. Levesque, *Wireless Information Networks*. Wiley, 1995.
  - [9] D. P. Taylor, G. M. Vitetta, B. D. Hart, and A. Mammela, "Wireless channel equalization," *European Transactions on Telecommunications*, vol. 9, no. 2, pp. 117–143, 1998.
  - [10] G. D. Forney, "Maximum-likelihood sequence estimation of digital sequence in the presence of intersymbol interference," *IEEE Transactions on Information Theory*, vol. 18, pp. 363–378, May 1972.
  - [11] G. D. Forney, "The Viterbi algorithm," in *Proceedings of the IEEE*, vol. 61, pp. 268–278, 1973.
  - [12] R. Haeb and H. Myer, "A systematic approach to carrier recovery and detection of digitally phase modulated signals on fading channels," *IEEE Trans. on Communications*, vol. 37, pp. 748–754, July 1989.
-

- [13] R. W. Lucky, "Automatic equalization for digital communication," *Bell Syst. Tech. J.*, vol. 44, pp. 547–588, 1966.
- [14] J. G. Proakis and J. Miller, "An adaptive receiver for digital signaling through channels with inter-symbol-interference," *IEEE Transactions on Information Theory*, vol. 15, pp. 484–497, Jul. 1969.
- [15] S. U. H. Qureshi, "Adaptive equalization," in *IEEE Proceedings*, vol. 73, pp. 1349–1387, sept. 1985.
- [16] H. Chen, R. Perry, and K. Buckley, "On MLSE algorithms for unknown fast time-varying channels," *IEEE Transactions on Communications*, vol. 51, pp. 730–734, May 2003.
- [17] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Transactions on Information Theory*, vol. 20, pp. 284–287, March 1974.
- [18] R. W. Lucky, "Techniques for adaptive equalization of digital communication," *Bell Syst. Tech. J.*, vol. 45, pp. 255–286, 1966.
- [19] B. Widrow, "Adaptive filters, i: Fundamentals," Tech. Rep. 6764-6, Stanford Electroics Laboratory, Stanford University, December 1966.
- [20] D. N. Godard, "Channel equalization using a Kalman filter for fast data transmission," *IBM J. Res. Dev.*, vol. 18, pp. 267–273, 1974.
- [21] M. E. Austin, "Decision-feedback equalization for digital communication over dispersive channels," Tech. Rep. 437, MIT Lincoln Laboratory, Lexington, Mass, August 1967.
- [22] P. Monsen, "Feedback equalization for fading dispersive channels," *IEEE Transactions on Information Theory*, vol. 17, pp. 56–64, Jan. 1971.
- [23] D. A. George, R. R. Bowen, and J. R. Storey, "An adaptive decision-feedback equalizer," *IEEE Transactions on Communications*, vol. 19, pp. 281–293, June 1971.
- [24] J. Salz, "Optimum mean-square decision feedback equalization," *Bell Syst. Tech. J.*, vol. 52, pp. 1341–1373, October 1973.

## Bibliography

---

- [25] D. L. Duttweiler, J. E. Mazo, and D. G. Messerschmitt, "Error propagation in decision-feedback equalization," *IEEE Transactions on Information Theory*, vol. 20, pp. 490–497, July 1974.
- [26] S. Altekari and N. Beaulieu, "Upper bounds to the error probability of decision feedback equalization," *IEEE Trans. on Information Theory*, vol. 39, pp. 145–156, 1993.
- [27] V. D. Trajkovic, P. B. Rapajic, and R. A. Kennedy, "Turbo DFE algorithm with imperfect decision feedback," *IEEE Signal Processing Letters*, vol. 12, pp. 820–823, Dec. 2005.
- [28] W. Zi-Ning and J. M. Cioffi, "Low-complexity iterative decoding with decision-aided equalization for magnetic recording channels," *IEEE Journal on Selected Areas in Communication*, vol. 19, pp. 699–708, Apr. 2001.
- [29] C. A. Belfiore and J. H. J. Park, "Decision feedback equalization," in *Proc. IEEE.*, vol. 67, pp. 1143–1156, Aug. 1979.
- [30] J. P. Seymour and M. P. Fitz, "Near-optimal symbol-by-symbol detection schemes for flat Rayleigh fading," *IEEE Transactions on Communications*, vol. 43, pp. 1525–1533, Feb/Mar/Apr 1995.
- [31] D. D. Falconer and F. R. Magee, "Adaptive channel memory truncation for maximum likelihood sequence estimation," *Bell Syst. Tech. J.*, vol. 9, pp. 1541–1562, Nov. 1973.
- [32] M. V. Eyuboglu and S. U. H. Qureshi, "Reduced state sequence estimation with set partitioning and decision feedback," *IEEE Transactions on Communication*, vol. 36, pp. 13–20, Jan. 1988.
- [33] A. Duel-Hallen and C. Heegard, "Delayed decision-feedback sequence estimation," *IEEE Transactions on Communication*, vol. 37, pp. 428–436, May 1989.
- [34] S. J. Simmons, "Breadth first trellis decoding with adaptive effort," *IEEE Transactions on Communication*, vol. 38, pp. 3–12, Jan. 1990.
- [35] W. U. Lee and F. S. Hill, "A maximum-likelihood sequence estimator with decision feedback equalization," *IEEE Transactions on Communication*, vol. 25, pp. 971–979, Sep. 1977.

- [36] S. U. H. Qureshi and E. E. Newhall, "An adaptive receiver for data transmission over time-dispersive channels," *IEEE Transactions on Information Theory*, vol. 19, pp. 448–457, July 1973.
- [37] C. T. Beare, "The choice of the desired impulse response in combined linear Viterbi algorithm equalizer," *IEEE Transactions on Communication*, vol. 26, pp. 1301–1307, Aug. 1978.
- [38] K. Wesolowski, "An efficient DFE and ML suboptimum receiver for data transmission over dispersive channels using two-dimensional signal constellations," *IEEE Transactions on Communication*, vol. 35, pp. 336–339, March 1987.
- [39] G. Forney, R. Gallager, G. Lang, F. Longstaff, and S. Qureshi, "Efficient modulation for band-limited channels," *Selected Areas in Communications, IEEE Journal on*, vol. 2, pp. 632–647, Sep 1984.
- [40] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Transactions on Information Theory*, vol. 28, pp. 55–67, Jan 1982.
- [41] A. N. D'Andrea, A. Diglio, and U. Mengali, "Symbol-aided channel estimation with nonselective Rayleigh fading channels," *IEEE Transactions on Vehicular Technology*, vol. 44, pp. 41–49, Feb. 1995.
- [42] P. Ho and J. H. Kim, "Pilot symbol assisted detection of CPM schemes operating in fast fading channels," *IEEE Transactions on Communications*, vol. 44, pp. 337–347, March 1996.
- [43] J. Cavers and M. Liao, "A comparison of pilot tone and pilot symbol techniques for digital mobile communication," in *IEEE Globecom*, pp. 915–921, 1992.
- [44] J. Lin, F. Ling, and J. K. Proakis, "Joint data and channel estimation for TDMA mobile channels," *International Journal of Wireless Information Networks*, vol. 1, no. 4, pp. 229–238, 1994.
- [45] M. J. Omid, S. Pasupathy, and P. G. Gulak, "Joint data and Kalman estimation for Rayleigh fading channels," *Wireless Personal Communications*, vol. 10, pp. 319–339, 1999.
- [46] G. J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*. Wiley Series in probability and Statistics, Wiley, 1997.

## Bibliography

---

- [47] J. K. Cavers and P. Ho, "Analysis of the error performance of trellis-Coded modulations in Rayleigh-fading channels," *IEEE Trans. Commun.*, vol. 40, pp. 74–83, Jan. 1992.
- [48] P. K. Frenger, N. Arne, and B. Svensson, "Decision-directed coherent detection in multicarrier systems on Rayleigh fading channels," *IEEE Trans. Veh. Technol.*, vol. 48, pp. 490–498, Mar. 1999.
- [49] H. Liu, G. Xu, L. Tong, and T. Kailath, "Recent developments in blind channel equalization: From cyclostationarity to subspaces," *IEEE Trans. Signal Process.*, vol. 50, pp. 83–99, Apr. 1996.
- [50] E. D. Carvalho and D. Slock, "Cramer-Rao bounds for semi-blind, blind and training sequence based channel estimation," in *Proc. Signal Process. Advances Wireless Commun. (SPAWC)*, 1997.
- [51] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J.R. Statist. Soc.*, vol. 76, pp. 341–353, 1977.
- [52] X. Wautelet, C. Herzet, A. Dejonghe, J. Louveaux, and L. Vandendorpe, "Comparison of EM-Based algorithms for MIMO channel estimation," *IEEE Trans. Commun.*, vol. 55, pp. 216–226, Jan. 2007.
- [53] X. Wautelet, C. Herzet, A. Dejonghe, and L. Vandendorpe, "MMSE-based and EM iterative channel estimation methods," in *Proc IEEE 10th Symposium on Communications and Vehicular Technology in the Benelux (SCVT)*, Nov. 2003.
- [54] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Prentice-Hall, 1979.
- [55] S. Haykin, *Adaptive Filter Theory*. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.
- [56] A. H. Sayed and T. Kailath, "A state-space approach to adaptive RLS filtering," *IEEE Signal Processing Magazine*, July 1994.
- [57] M. S. Grewal and A. P. Andrews, *Kalman Filtering: Theory and Practice*. Prentice Hall, 1993.
- [58] B. Widrow, *Adaptive Signal Processing*. Prentice Hall, 1985.



- [59] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME, Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.
- [60] J. B. Evans and B. Liu, "Variable step size methods for the LMS adaptive algorithm," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 2, pp. 422–425, May 1987.
- [61] Z. Zhu and H. Sadjadpour, "An adaptive per-survivor processing algorithm," *IEEE Transactions on Communication*, vol. 50, pp. 1716–1718, November 2002.
- [62] E. Eleftheriou and D. D. Falconer, "Tracking properties and steady-state performance of RLS adaptive filter algorithms," *IEEE Trans. on Acoustic, Speech, Signal Processing*, vol. 34, pp. 1097–1110, Jan. 1986.
- [63] E. Eweda, "Convergence of the RLS and LMS adaptive filters," *IEEE Trans. on Circuits and Systems*, vol. 34, pp. 799–803, July 1987.
- [64] E. Eweda, "Comparison of RLS,LMS and sign algorithms for tracking randomly time-varying channels," *IEEE Trans. on Signal Processing*, vol. 42, pp. 2937–2944, 1994.
- [65] R. Raheli, A. Polydoros, and C. Tzou, "Per-survivor processing: A general approach to MLSE in uncertain environments," *IEEE Transaction on Communications*, vol. 43, pp. 354–364, 1995.
- [66] A. Polydoros and K. M. Chugg, "Per survivor processing (PSP)," *chap.2 in Wireless Communication:TDMA versus CDMA, editors S. G. Glisic, Kluwer, 1997.*
- [67] R. Raheli, G. Marino, and P. Castoldi, "PSP and tentative decisions: What is in between?," *IEEE Transactions on Communication*, vol. 44, pp. 127–129, Feb. 1996.
- [68] J. M. Ortega and W. C. Rheinboldt, "Iterative solution of nonlinear equations in several variables," *SIAM*, 2000.
- [69] K. Rose, "Deterministic annealing," in *Proc. IEEE.*, vol. 86, Nov. 1998.
- [70] G. K. Kaleh, "Channel equalization for block transmission systems," *IEEE J. Sel. Areas Commun.*, vol. 13, pp. 110–121, Jan. 1995.

## Bibliography

---

- [71] F. Tarköy, “MMSE-Optimal feedback and its applications,” in *IEEE ISIT 95, Whistler, BC, Canada*, p. 334, Sept. 1995.
- [72] D. P. Taylor, “The estimate feedback equalizer: A suboptimum nonlinear solution,” *IEEE Trans. Commun.*, vol. 21, pp. 979–990, Sept. 1973.
- [73] A. Chan and G. W. Wornell, “A class of block-iterative equalizers for inter-symbol interference channels: Fixed channel results,” *IEEE Trans. Commun.*, vol. 49, pp. 1966–1976, Nov. 2001.
- [74] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Trans. Inform. Theory*, vol. 47, pp. 498–519, Feb. 2001.
- [75] J. J. Binney and et al., *The Theory of Critical Phenomena*. Oxford, 1992.
- [76] D. Falconer, S. L. Ariyavisitakul, A. Benyamin-Seeyar, and B. Eidson, “Frequency domain equalization for single-carrier broadband wireless access systems),” *IEEE Commun. Mag.*, vol. 40, pp. 58–66, Apr. 2002.
- [77] N. Benvenuto and S. Tomasin, “Block iterative DFE for single carrier modulation,” *IEEE Electron. Letters.*, vol. 38, pp. 1144–1145, Sept. 2002.
- [78] A. J. Viterbi, “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm,” *IEEE Transactions on Information Theory*, vol. 13, pp. 260–269, April 1967.
- [79] J. K. Omura, “On the Viterbi decoding algorithm,” *IEEE Transactions on Information Theory*, vol. 15, pp. 177–179, Jan. 1969.
- [80] Q. Dai and E. Shwedyk, “Detection of bandlimited signals over frequency selective Rayleigh fading channels,” *IEEE Transactions on Communication*, vol. 42, pp. 941–950, Feb/Mar/Apr 1994.
- [81] X. Yu and S. Pasupathy, “Innovation-based MLSE for Rayleigh fading channels,” *IEEE Transactions on Communication*, vol. 43, pp. 1534–1544, Feb/Mar/Apr 1995.
- [82] J. H. Lodge and M. L. Moher, “Maximum likelihood sequence estimation of CPM signals transmitted over Rayleigh flat-fading channels,” *IEEE Transactions on Communication*, vol. 38, pp. 787–794, June 1990.

- [83] R. E. Morely and D. L. Snyder, "Maximum likelihood sequence estimation for randomly dispersive channels," *IEEE Transactions on Communication*, vol. 27, pp. 833–839, June 1979.
- [84] N. Seshadri, "Joint data and channel estimation using blind trellis search techniques," *IEEE Transactions on Communication*, vol. 42, pp. 1000–1011, Feb/Mar/Apr 1994.
- [85] F. R. Magee and J. G. Proakis, "Adaptive maximum likelihood sequence estimation for digital signaling in presence of intersymbol interference," *IEEE Transactions on Information Theory*, vol. 19, pp. 120–124, Jan. 1973.
- [86] M. Ghosh and C. L. Weber, "Maximum likelihood blind equalization," *Optical Engineering*, vol. 31, pp. 12224–12228, July 1992.
- [87] K. M. Chugg and A. Polydoros, "MLSE for an unknown channel- part 2: Tracking performance," *IEEE Transactions on Communication*, vol. 44, pp. 949–958, August 1996.
- [88] K. M. Chugg and A. Polydoros, "MLSE for an unknown channel-Part 1:optimality considerations," *IEEE Transactions on Communication*, vol. 44, pp. 836–846, July 1996.
- [89] M. E. Rollins and S. J. Simmons, "Simplified per-survivor Kalman processing in fast frequency-selective fading channels," *IEEE Transactions on Communication*, vol. 45, pp. 544–553, May 1997.
- [90] M. Chiu and C. Chao, "Analysis of LMS-adaptive MLSE equalization on multipath fading channels," *IEEE Transactions on Communication*, vol. 44, pp. 1684–1692, Dec. 1996.
- [91] D. Borah and B. Hart, "Receiver structures for timing-varying frequency-selective fading channels," *IEEE J. Select. Areas Commun.*, vol. 17, pp. 1863–1875, Nov. 1999.
- [92] Y. Li and X. Huang, "The simulation of independent Rayleigh faders," *IEEE Transactions on Communications*, vol. 50, no. 9, pp. 1503–1514, 2002.
- [93] H. Wang and P. Chang, " On verifying the first order Markovian assumption for a Rayleigh fading channel model," *IEEE Transactions on Vehicular Technology*, vol. 45, no. 2, pp. 353–357, 1996.

## Bibliography

---

- [94] K. E. Baddour and N. C. Beaulieu, "Autoregressive models for fading channel simulation," in *Proceedings of the IEEE Global Telecommunications Conference*, pp. 1187–1192, Nov. 2001.
- [95] M. J. Omid, P. G. Gulak, and S. Pasupathy, "Parallel structures for joint channel estimation and data detection over fading channels," *IEEE Journal of Selected Areas in Communications*, vol. 16, pp. 1616–1629, Dec. 1998.
- [96] A. J. Viterbi and J. K. Omura, *Digital communications; Coding theory*. New York : McGraw-Hill, 1979.
- [97] S. Julier and J. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, pp. 401–422, Mar. 2004.
- [98] P. M. Djuric, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Miguez, "Particle filtering," *IEEE Signal Processing Mag.*, vol. 20, pp. 19–38, Sept. 2003.
- [99] D. Simon, *Optimal State Estimation*. John Wiley, 2006.
- [100] T. K. Moon, "The expectation maximization algorithm," *IEEE Signal Processing Magazine*, pp. 47–60, Nov. 1996.
- [101] G. Giannakis and C. Tepedelenlioglu, "Basis expansion models and diversity techniques for blind identification and equalization of timevarying channels," in *Proc. IEEE*, vol. 86, pp. 1969–1986, Oct. 1998.
- [102] C. Wu, "On the convergence properties of the EM algorithm," *Ann. Statist.*, vol. 11, pp. 95–103, 1983.
- [103] W. Tinsson, *Nonlinear estimation with applications to target tracking*. PhD thesis, Georgia Institute of Technology, Atlanta, GA, 1996.
- [104] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Stat. Comput.*, vol. 3, pp. 197–208, 2000.
- [105] A. Kong, J. Liu, and W. Wong, "Sequential imputations and Bayesian missing data problems," *J. Amer. Stat. Assoc.*, vol. 89, pp. 278–288, 1994.
- [106] J. Liu, "Metropolized independent sampling with comparison to rejection sampling and importance sampling," *Stat. Comput.*, vol. 6, pp. 113–119, 1996.