Thèse

présentée pour obtenir le grade de docteur

de Télécom ParisTech

Spécialité : Signal et Images

Steffen Barembruch

Méthodes approchées de maximum de vraisemblances pour la classification et identification aveugles en communications numériques

Approximate Maximum Likelihood Methods for Blind Classification and Identification in Digital Communications

Contents

Acknowledgements

First of all I would like to thank my advisor Prof. Eric Moulines. During the three years of the thesis he not only gave me an expert guidance on any topic I was working on. He also supported me in any possible way like little things, conferences I wanted to go to, my stay in California; but he also let me choose in which direction I wanted to direct my research as much as possible in the strict project context the thesis was written in. He gave me a lot of space to work independently and was always open to my ideas. I am more than grateful to everyone in the group STA at TSI at Télécom ParisTech for their support, ideas, discussions and distractions. Of the whole group I would like to thank Aurélien Garivier in particular because he guided me very closely during the first year of the thesis.

The thesis was part of a big project set up by the DGA. Hence, without the DGA my thesis would not have been possible and would not have been financed either. I thank the coordinator Philippe Loubaton who was also part of the jury of the thesis defence and reviewed my thesis. He managed the project very professionally and made sure that the results were coherent and convincing.

Furthermore I am very grateful to the other members of the jury of my thesis defence: François Desbouvries who kindly accepted to review my thesis, Inbar Fijalkow who presided over the jury and Walid Hachem and Philippe Ciblat.

I am very grateful to Prof. Anna Scaglione who invited me to UC Davis for three months during the thesis which made it possible for me to spend an important part of my thesis abroad. I could deeply profit from here profound knowledge about Digital Communications in general and Compressive Sensing. During this period, we were able to develop together the Expectation Sparse Maximization Algorithm that is an important part of this thesis. In addition I would like to thank Matt Sharp from UC Davis for many discussions and interesting new ideas.

In addition, I would like to thank everyone who supported me in writing this thesis.

Chapter 1

Résumé (en français)

1.1 Introduction

Notre travail porte sur la démodulation autodidacte de canaux sélectifs en fréquence ou en temps, et sur la reconnaissance de modulation. L'objectif de notre travail était de développer des méthodes adaptées à des transmissions discontinues (trames ou "bursts") telles qu'on les rencontre dans les systèmes à évasion de fréquence ou à accès multiple par multiplexage temporel et fréquentiel. Ce contexte est très défavorable aux méthodes de déconvolution autodidacte classiques, comme la méthode du CMA par blocs ou de façon plus générale les estimateurs basés sur la minimisation de contraste pour lesquels la taille des blocs doit être suffisante pour que l'algorithme converge. Nous avons aussi considéré des modulations à grands nombres d'états.

Nous nous sommes intéressés à l'estimation des paramètres des canaux et de la modulation (coefficients et ordre) et à celle des symboles dans le cas d'une modulation linéaire, pour une transmission sur un canal soumis à un évanouissement sélectif en fréquence (lié à la présence de trajets multiples) en présence d'un bruit additif. Les paramètres des canaux sont considérés constants pendant toute la durée d'un bloc de symboles (bursts). Plus précisément, l'estimation de l'ordre du modèle désigne l'estimation du nombre d'états de la modulation. Autrement dit, étant donné une séquence d'observations reçues, une décision doit être prise sur l'alphabet de la modulation ayant généré cette séquence d'observations (Le signal vient-il d'une modulation BPSK, MAQ-16, MAQ-64, ...?). Dans la littérature, ce problème est connu sous le nom "Classification de Modulation"et le problème est résolu dans la plupart des travaux sur ce sujet dans la situation où le canal possède un seul trajet (évanouissement plat, sans sélectivité en fréquence) ou dans le modèle encore plus simple de canal gaussien (bruit additif, blanc et Gaussien, atténuation constante). En général, il existe deux principes différents de classification de la modulation : des méthodes de maximum de vraisemblance (Maximum Likelihood Estimation) et des méthodes basées sur les caractéristiques du signal (Feature Based Estimation). Un algorithme de maximum de vraisemblance a été proposé par Hong [2006] prenant en compte des canaux de trajets multiples. Par contre, cet algorithme est applicable uniquement pour des modulations linéaires (BPSK et QPSK) ayant un petit nombre d'états à cause de la complexité de calculs. Par conséquent, à notre connaissance il n'existe aucun algorithme de classification de modulation dans la littérature qui prenne en compte à la fois des canaux de trajets multiples (évanouissements sélectifs en fréquence) et des grandes modulations linéaires (ce qui empêche donc de comparer les nouveaux algorithmes à l'état de l'art).

Dans notre travail, nous proposons un algorithme traitant ce problème de type maximum de vraisemblance. Nous n'avons pas considéré les méthodes basées sur l'extraction de caractéristiques des modulations à partir des signaux reçus, car tous les algorithmes connus dans ce cadre s'appliquent uniquement aux canaux d'un seul trajet et il ne semble pas a priori possible de généraliser ces méthodes au cas des canaux de trajets multiples. Dans la suite, nous nous concentrons sur les modulations MAQ possédant des nombres d'états différents. Les algorithmes d'estimation des symboles néanmoins s'appliquent également à d'autres modulations linéaires, notamment aux modulations PSK, et les algorithmes d'estimation de l'ordre sont également capable de prendre en compte à la fois les modulations MAQ et d'autres modulations linéaires.

L'idée de l'estimation du nombre d'états de la modulation par maximum de vraisemblance consiste à évaluer la vraisemblance de la suite des observations reçues dans chacun des modèles potentiels et de prendre la décision en faveur de celui qui maximise la vraisemblance en prenant en compte des corrections additives afin d'obtenir une estimation non biaisée. Les modèles dans lesquels nous estimons consistent à exprimer le signal reçu comme une convolution de la réponse impulsionnelle du canal et du signal émis plus un bruit additif, étant donné l'ordre de la réponse impulsionelle.

L'ordre de la réponse impulsionelle peut être traité comme une autre incertitude de l'ordre du modèle. L'estimation de l'ordre est basée sur le principe de "*Minimum Description Length*"(MDL) proposé par Rissanen [1978]. Ce critère suggère de choisir le modèle qui est associé à la description la plus parcimonieuse des données. En d'autres termes, si deux modèles décrivent les données de façon statistiquement équivalente, on va choisir le modèle le plus parcimonieux en termes de nombres de paramètres. Une définition commune du MDL est le *critère d'information Bayésien* ou "*Bayesian Information Criterion*"(BIC) qui empêche la surestimation de l'ordre. En général, l'estimation d'ordre d'un modèle directement basé sur la vraisemblance (sans correction) dans chacun des modèles aurait une tendance à privilégier les modèles les plus complexes, car souvent les modèles d'ordre faible sont une sous-classe des modèles d'un ordre plus élevé. Par exemple, si on considère les réponses impulsionelles du canal, le modèle d'ordre k est une sous-classe du modèle l pour $l > k$, car si on met $l-k$ coefficients égaux à 0, on obtient effectivement un exemple du modèle d'ordre k. Pour cette raison, les vraisemblances dans chaque modèle sont pénalisées dans le BIC par rapport aux nombres des paramètres inconnus que le modèle correspondant inclus. Le nombre de paramètres est donc égal à l'ordre du canal (l'ordre du canal est le nombre des coefficients de la réponse impulsionelle) plus un paramètre pour l'écart-type. Étant donné l'ordre du canal, une pénalisation par rapport au nombre d'états de la modulation n'est donc pas nécessaire, parce que le nombre des paramètres du modèle est indépendant de la modulation.

Pour les modulations à grands nombres d'états ou lorsque l'ordre du canal est élevé, il est difficile d'évaluer de façon exacte la vraisemblance des observations dans chaque modèle de façon explicite. Pour un nombre d'états de modulation et un ordre du canal fixés, la vraisemblance dépend des paramètres du canal et de l'écart-type du bruit; il est donc nécessaire d'estimer ces différentes quantités pour déterminer la vraisemblance des observations. Ce problème est connu sous le nom de l'identification aveugle ("*Blind Identification*"). Si le nombre d'états de la modulation et l'ordre du canal sont assez faibles, l'algorithme itératif "Expectation-Maximization"(EM) couplé avec l'algorithme de Baum-Welch est la méthode de référence pour calculer l'estimateur de maximum vraisemblance des paramètres du canal et de l'écart-type. Puisque la complexité de l'algorithme Baum-Welch est de l'ordre $\mathcal{O}(d^L)$, où d est le nombre d'états de la modulation et L est l'ordre du canal, l'algorithme devient trop coûteux en temps de calcul en considérant des modulations à partir de MAQ-16 et des ordres de canal à partir de 3 ou 4.

Pour pouvoir appliquer les mêmes principes, nous avons tout d'abord étudié des algorithmes qui approchent les mêmes quantités que l'algorithme de Baum-Welch tout en ayant un coût de calcul réduit. La quantité centrale dans l'algorithme de Baum-Wech est la distribution de lissage des symboles. Cette quantité est essentielle pour mettre à jour les paramètres dans la partie de la maximisation de l'algorithme EM. L'état de l'art est l'algorithme EMVA de Nguyen and Levy [2003, 2005]. Il emploie l'algorithme Viterbi ou des variantes approximatives comme l'algorithme M ou T (M- and T-algorithm). Nous nous sommes concentrés sur les méthodes d'approximations par systèmes de particules en interactions. Cette méthode est calquée sur l'algorithme de Baum-Welch, qui consiste à estimer en une passe avant la loi de filtrage des symboles conditionnellement aux observations jusqu'à l'instant courant, puis d'évaluer dans une passe arrière la vraisemblance des observations futures étant donné l'état courant; ces deux quantités sont ensuite combinées pour déterminer la distribution de lissage marginale. Dans l'approche particulaire, on approche tout d'abord dans la passe avant la loi de filtrage des symboles en maintenant un ensemble d'états supports (dont le nombre est significativement plus faible que la taille de l'espace d'états) et en mettant à zéros les probabilités des autres valeurs possibles de l'état. En procédant de la sorte, on explore uniquement une partie des états possibles à chaque instant; on assigne de plus des poids d'importance aux particules retenues. Cette approximation de la loi de filtrage est construite de façon récursive en temps. Comme l'espace des états est fini, on procède à l'instant suivant en considérant tous les descendants possibles de l'ensemble des particules actuelles et en calculant des poids d'importance pour ces nouvelles particules. Un sous-ensemble des descendants est ensuite sélectionné et repondéré pour maintenir à l'instant suivant une approximation particulaire. Après avoir mis en œuvre l'algorithme particulaire de filtrage, les distributions de lissage (=distribution des symboles étant donné toute la suite des observations) peuvent être approchées en mettant en œuvre différentes méthodes.

Les résultats que nous avons obtenu au cours des nombreuses expériences de Monte-Carlo que nous avons effectué montrent que la meilleure méthode est un algorithme de lissage à horizon fixe basé sur un algorithme particulaire de filtrage marginal et une correction des poids dans une passe arrière et couplé avec une méthode de sélection aléatoire des particules qui minimise soit la norme $L2$ ou soit la distance du chi-2.

1.1.1 Algorithmes de lissage particulaire

Afin d'obtenir des algorithmes statistiquement efficaces, nous avons choisi de nous intéresser aux méthodes de maximum de vraisemblance. Le calcul et l'optimisation directe de la vraisemblance étant très difficile dans ce cadre, nous nous sommes intéressés à des méthodes basées sur l'algorithme d'"Expectation-Maximisation"(EM). Dans ce contexte, les symboles transmis sont considérés comme des données manquantes. La méthode EM est itérative et nécessite de calculer à chaque itération la distribution de lissage des symboles d'information conditionnellement aux observations pour la valeur courante du paramètre. Cette étape de lissage permet de calculer la corrélation conditionnelle des symboles transmis et l'intercorrélation des symboles et des signaux. Ces deux quantités sont utilisées pour mettre à jour les paramètres.

Dans les situations d'intérêt, la dimension des états pouvant être très élevée, les approches de lissage classiques basées sur l'algorithme de Baum-Welch ne peuvent être implémentées directement. La dimension des états est égale au nombre d'états de la modulation élevé à une puissance égale à la longueur du filtre)

Nous avons développé de nouvelles méthodes de lissage basées sur des approximations particulaires; que nous avons ensuite comparées à l'algorithme EMVA par Nguyen and Levy [2005] qui est considéré comme l'état de l'art du domaine dans le contexte des communications numériques. Nous avons en particulier développé des approximations particulaires de différentes méthodes de lissage développées dans le cadre des modèles linéaires Gaussiens et des modèles de Markov cachés à états discrets. Plus particulièrement, nous avons étudié:

1. l'algorithme de lissage à horizon fixe basé sur l'adaptation des méthodes de Rauch Tung Striebel, correspondant à l'algorithme de Baum-Welch pour les chaînes de Markov cachées

à nombre d'états finis. Nous avons étendu l'algorithme proposé par Doucet et al. [2000] en introduisant une nouvelle façon de calculer les poids permettant de préserver la diversité des particules.

2. l'algorithme de filtrage à deux filtres introduits par Kitagawa [1994] et améliorés dans les travaux récents de Fearnhead et al. [2008].

Notons que dans le cadre considéré, les algorithmes de lissage que nous avons étudiés possèdent tous une complexité croissant de façon linéaire en fonction du nombre de particules. Alors que, dans le cas général, les algorithmes de lissage ont une complexité qui croît de façon quadratique.

Nous avons formulé un nouvel algorithme de lissage, qui est une amélioration du lissage à deux filtres. Cet algorithme permet de s'affranchir des difficultés présentées par les algorithmes précédents. A l'instar de l'algorithme à deux filtres, cette méthode est basée sur deux approximations particulaires. La première, la plus classique, permet d'approcher de façon séquentielle les lois de filtrage, c'est-à-dire la loi du symbole à l'instant courant, conditionnellement aux observations jusqu'à l'instant courant. La deuxième permet d'approcher une quantité proportionnelle à la variable rétrograde, qui est renormalisée de sorte à être une loi de probabilité. Cette loi de probabilité peut être interprétée comme la loi du symbole courant, conditionnellement aux observations faites aux instants postérieurs à l'instant courant.

L'idée de la méthode à deux filtres joints est, dans une étape intermédiaire, d'enrichir les supports particulaires obtenus lors des passes avant et arrière, en effectuant une fusion pondérée des deux approximations particulaires. En ce sens, il est plus intéressant que l'algorithme classique à deux filtres, qui n'hybride pas les deux populations de particules. L'avantage de cet algorithme par rapport au filtrage à deux filtres est spectaculaire. La structure de la matrice de transition des états (qui est très creuse), fait que l'algorithme de filtrage à deux filtres aboutit souvent à des distributions dégénérées, au sens où tous les poids particulaires deviennent nuls du fait de l'impossibilité d'associer les particules entre les passes avant et arrières. Cette dégénérescence disparaît dans la formulation de l'algorithme à deux filtres joints.

Dans la méthode de filtrage particulaire classique, les particules sont simplement sélectionnées suivant une probabilité proportionnelle à leur poids d'importance, qui est évalué de façon séquentielle. Cette méthode n'est pas satisfaisante lorsqu'elle est appliquée à des situations où les variables d'états sont discrètes, car certaines particules peuvent alors être exactement répliquées (ce qui est clairement inefficace, puisque l'information est, dans ce cas, exactement répliquée). Il est possible d'éviter ce phénomène en construisant à chaque itération une approximation au moyen d'un système de particules pondérées. Dans ce cadre, à chaque itération, l'approximation particulaire sera exactement constituée de N points de supports distincts, associés à des poids d'importance. Ce principe a été introduit par Fearnhead and Clifford [2003], qui ont proposé une méthode consistant à calculer les poids des particules de façon à minimiser l'écart quadratique entre l'approximation particulaire et la loi à approcher. Nous avons montré comment ce principe de pondération pouvait être généralisé à d'autres mesures de divergence entre lois de probabilité (entropie relative, distance du chi2).

Nous avons obtenu, pour différentes mesures de divergence, un algorithme explicite permettant de calculer les poids optimaux (ou plutôt, nous avons mis en évidence un algorithme d'optimisation permettant de calculer les poids optimaux et la règle de sélection des particules) . Ces méthodes sont optimales au sens où les poids des particules sélectionnées sont calculés de façon à minimiser la divergence entre la loi cible et l'approximation particulaire à chaque itération de l'algorithme. Par rapport à l'approche “classique”de sélection des particules, cette méthode a pour intérêt d'accroître de façon significative la diversité des systèmes de particules.

Nous avons montré que la méthode de pondération / sélection associée à la divergence de Kullback-Leibler ou entropie mutuelle est en fait équivalente à la méthode associée à l'erreur quadratique introduite dans l'article de Fearnhead and Clifford [2003]. Cette méthode consiste à garder les particules dont les poids dépassent un certain seuil en conservant leur pondération, puis à sélectionner les particules dont les poids sont inférieurs à ce seuil avec une probabilité proportionnelle à leur poids d'importance et à associer un poids constant à ces différentes particules. La solution associée à la distance du chi-2 aboutit à des résultats différents: si le principe de préserver les particules associées aux poids les plus élevés est préservé, la sélection s'effectue sur la racine carrée des poids d'importance, ce qui a pour effet d'augmenter la diversité des particules rééchantillonnées (on sélectionne ainsi des particules qui ont des poids particulaires relativement faibles, ce qui permet "localement" de propager des hypothèses pouvant apparaître improbable).

1.1.2 Algorithmes de déconvolution autodidacte

Nous avons mis au point de nouveaux algorithmes de déconvolution autodidactes, basés sur des approximations particulaires de l'estimateur du maximum de vraisemblance des paramètres des canaux. Ces méthodes sont itératives et basées sur une version stochastique de l'algorithme EM (Expectation-Maximisation), qui revient à remplacer le calcul exact des espérances sous les lois de lissage des symboles par des approximations de type échantillonnage d'importance. Nous retenons un ensemble de valeurs possibles des symboles que nous pondérons à l'aide de poids d'importance. Les méthodes de calcul des symboles et des poids s'inspirent des méthodes de traitement “per-survivor processing”, dans le sens où un certain nombre de chemins (selon la complexité de la modulation et du canal) sont conservés dans une passe avant. Elles s'en distinguent par l'ajout d'une passe arrière, dont l'objectif est d'estimer des poids d'importance pour ensuite estimer les probabilités marginales de lissage . La complexité de ces algorithmes croît de façon linéaire avec le nombre d'états du treillis et le nombre de chemins conservés lors de la passe avant : elle dépend donc fortement de la complexité du problème posé. Nous avons testé plusieurs de ces algorithmes, dont le meilleur est l'algorithme de lissage à horizon fixe couplé à la méthode de sélection des poids minimisant la norme $L2$. Nous avons proposé différentes stratégies d'initialisation de cet algorithme, en démontrant l'intérêt de changer la direction de l'estimation d'une initialisation à l'autre, ce qui revient, dans la méthode de lissage à deux filtres joints, à commencer par la passe arrière de filtrage, avant de continuer avec la passe avant.

Finalement, une méthode a été proposée pour améliorer l'initialisation des paramètres par l'algorithme EM pour les modulations à grand nombre d'états. Il est bien connu que Le choix des valeurs initiales des paramètres inconnus est assez important pour la convergence de l'algorithme EM. Si l'initialisation est très éloignée du canal “réel”, la convergence de l'algorithme est en général plus lente car plus d'itérations sont nécessaires afin d'atteindre la précision requise. De plus, l'algorithme en présence d'une mauvaise initialisation risque de converger vers un maximum local de la vraisemblance éloigné du maximum global qu'on cherche. Cette faiblesse est d'autant plus marquée que le nombre d'états de la modulation est élevé: si le nombre d'états de la modulation est faible, le modèle est plus robuste et on peut explorer une plus grande partie de l'ensemble des états dans chaque instant; cet ensemble de faits concordent pour favoriser la convergence des itérations de l'EM vers un maximum global de la vraisemblance.

Nous avons montré que l'estimation pour des modulations à grand nombre d'états (MAQ-64, MAQ-128 et MAQ-256) peut être améliorée en estimant d'abord les coefficients de canal à l'aide d'une méthode de quasi-maximum de vraisemblance, dans laquelle nous remplaçons la vraisemblance exacte par la vraisemblance associée à une modulation ayant un plus petit nombre d'états (par exemple MAQ-16). Nous avons démontré, à la fois théoriquement et pratiquement, que cette façon de procéder conduisait à des estimateurs robustes, qui permettent d'obtenir de bonnes valeurs d'initialisation pour l'estimation dans les modèles complets. Chaque état dans le modèle biaisé peut être vu comme un macro-état regroupant un certain nombre d'états du vrai modèle. Nous avons conçu et mis en œuvre un algorithme innovant d'estimation de l'ordre de la modulation, qui combine des critères issus de la théorie de l'information (critère MDL), les nouveaux algorithmes d'estimation des coefficients du canal par maximum de vraisemblance approché et l'estimation par macro-états.

1.1.3 Reconnaissance de modulation

Nous avons introduit une nouvelle famille de tests qui exploitent la structure spécifique des alphabets de modulation MAQ. Ces tests sont basés sur la méthode du maximum de vraisemblance généralisé, qui a été adapté ici à la problématique de la classification de modulation.

Nous avons approfondi l'étude statistique de ces tests pour des canaux à évanouissements plats. Il est toutefois clair que les méthodes que nous avons proposées s'étendent aux canaux à évanouissements sélectifs en fréquence et en temps. Nous n'avons pu, faute de temps, poursuivre dans cette direction, qui comporte des modèles ouverts importants d'un point de vue pratique et théorique (par exemple la comparaison de modèles de chaînes de Markov cachées ayant des topologies différentes).

Supposons que nous cherchions à discriminer deux modulations MAQ-4 et MAQ-16. Nous introduisons un *méta-modèle paramétrique*, construit de telle sorte que les modulations MAQ-4 et MAQ-16 correspondent à des restrictions particulières des valeurs possibles des paramètres pour ce méta-modèle. Dans ce cas particulier, le méta-modèle est basé sur un alphabet à 16 éléments, les différents éléments de cet alphabet étant associés à des poids. Nous représentons la modulation MAQ-4 en affectant un poids égal à zéro aux 12 symboles "extérieurs". Pour la modulation MAQ-16, tous les poids sont pris égaux. Pour reconnaître une modulation, nous devons tester l'appartenance du paramètre de poids à des sous-ensembles particuliers de l'espace des paramètres. A cette fin, nous avons construit deux tests de maximum de vraisemblance généralisés.

Dans le premier test d'hypothèses, sous l'hypothèse nulle, nous supposons que les poids sont tous égaux. Sous l'hypothèse alternative, nous supposons qu'ils sont différents. Nous avons montré la convergence de la loi de ce test vers une distribution du Chi-2 à un degré de liberté sous l'hypothèse nulle, ce qui nous permet de déterminer aisément le seuil du test pour un niveau de signification donné mais aussi de calculer les p-valeurs.

Dans la seconde procédure, nous supposons sous l'hypothèse nulle que les poids des symboles extérieurs sont tous égaux à zéro. Comme sous l'hypothèse nulle les paramètres de la distribution appartiennent à la frontière de l'espace des paramètres, la détermination de la distribution asymptotique est plus délicate (il n'est plus possible d'appliquer la théorie classique des tests de rapport de vraisemblance généralisés, car cette théorie stipule que les paramètres appartiennent à l'intérieur de l'espace des paramètres). Nous sommes toutefois parvenus à déterminer la distribution asymptotique du test en nous appuyant sur les travaux de Andrews [2001]. Nous avons aussi démontré que les tests étaient asymptotiquement consistants au sens de Bahadur. D'un point de vue pratique, il s'avère que les deux tests sont équivalents, les courbes COR des deux tests étant similaires. Nous avons entrepris l'extension de ces tests aux modèles incorporant des évanouissements sélectifs en temps et en fréquence. Cette méthode est très prometteuse pour reconnaître des simulations linéaires sur des canaux longs, voire variables au cours du temps.

Nous avons en parallèle effectué une campagne importante de simulations sur des canaux sélectifs en temps et en fréquence, démontrant que les approches que nous avions proposées permettaient d'obtenir des performances en termes de classification, qui supplantent de façon significative l'état de l'art dans le domaine.

1.1.4 Identification autodidacte sous contrainte de parcimonie

Nous avons mis au point une nouvelle variante de l'algorithme EM, que nous avons appelée ESpaM (pour *Expectation Sparse Maximization*), qui exploite la parcimonie du vecteur de paramètres à estimer. Cette méthode étend les techniques proposées pour le "compressive sensing"à la problématique de l'estimation et de la déconvolution autodidacte. Cette problématique diffère des hypothèses classiques du compressive sensing dans le sens où la matrice d'observations est "partiellement"inconnue tout en étant fortement structurée (c'est le produit d'une matrice de convolution et d'une matrice de codes).

Nous avons proposé une nouvelle famille d'algorithmes itératifs, basés sur les itérations de l'EM et prenant en compte la parcimonie du vecteur de paramètres à reconstruire. Nous avons proposé un algorithme innovant, dont la complexité est à peu près similaire à celle de l'EM et qui s'avère être statistiquement beaucoup plus robuste que ce dernier quand le paramètre à estimer est parcimonieux. Lorsque le rang de la matrice d'observation est inférieur au nombre de paramètres, la fonctionnelle de l'algorithme EM, c'est-à-dire l'espérance de la vraisemblance complète conditionnellement aux observations et à la valeur courante des paramètres, ne possède pas un extrémum unique. Dans ce cas, l'espace des solutions constitue au contraire un sous-espace affine de dimension strictement positive. L'algorithme ESpaM choisit dans le sous espace des solutions la solution la plus parcimonieuse, en utilisant un algorithme comme l'"Orthogonal Matching Pursuit".

D'un point de vue théorique, nous avons établi que, sous certaines hypothèses techniques sur la parcimonie du modèle et sa régularité, la vraie valeur du paramètre est un point fixe de l'algorithme ESpaM. Nous avons aussi montré que l'algorithme ESpaM est monotone, au sens où chaque itération conduit à un accroissement de la vraisemblance de l'observation.

Nous avons tout d'abord appliqué cet algorithme à un canal sélectif en fréquence. Dans la situation considérée, l'équivalent à temps discret du canal de propagation est de grande dimension, mais peu de coefficients du canal sont significativement non nuls. Nous avons ensuite étendu cet algorithme à l'estimation de canaux doublement sélectifs en temps et en fréquence en nous basant sur une technique consistant à développer l'évolution des paramètres du filtre sur une base de fonctions. Nous avons appliqué cette technique à la fois à des modulations linéaires et à des modulations à porteuses multiples. Dans le cadre de porteuses multiples, nous avons étendu l'algorithme proposé à un contexte semi-autodidacte, dans lequel nous utilisons de façon conjointe les symboles transmis et une (partie) des pilotes.

Nous avons, dans une étude comparative assez complète, établi la robustesse des algorithmes proposés par rapport aux méthodes classiques et en particulier par rapport à l'algorithme EM. Nous avons établi que notre algorithme était supérieur à celui proposé par Ben Salem and Salut

[2004] pour des modulations doublement sélectives en fréquence et en temps.

1.2 Classification et identification aveugle

Nous nous sommes intéressés à l'estimation des paramètres des canaux et de la modulation (coefficients et ordre) et à celle des symboles dans le cas d'une modulation linéaire, pour une transmission sur un canal soumis à un évanouissement sélectif en fréquence (lié à la présence de trajets multiples) en présence d'un bruit additif . Les paramètres des canaux sont considérés constants pendant toute la durée d'un bloc de symboles (burts) . Plus précisément, l'estimation de l'ordre du modèle désigne l'estimation du nombre d'états de la modulation. Autrement dit, étant donné une séquence d'observations reçues, une décision doit être prise sur l'alphabet de la modulation ayant généré cette séquence d'observations (Le signal vient-il d'une modulation BPSK, MAQ-16, MAQ-64, ...?). Dans la littérature, ce problème est connu sous le nom 'Classification de Modulation' et le problème est résolu dans la plupart des travaux sur ce sujet dans la situation où le canal possède un seul trajet (évanouissement plat, sans sélectivité en fréquence) ou dans le modèle encore plus simple de canal gaussien (bruit additif, blanc et Gaussien, atténuation constante). En général, il existe deux principes différents de classification de la modulation : des méthodes de maximum de vraisemblance (Maximum Likelihood Estimation) et des méthodes basées sur les caractéristiques du signal (Feature Based Estimation). Un algorithme de maximum de vraisemblance a été proposé en 2006 prenant en compte des canaux de trajets multiples. Par contre, cet algorithme n'est applicable que pour des modulations linéaires (BPSK et QPSK) ayant un petit nombre d'états à cause de la complexité de calculs. Par conséquent, à notre connaissance il n'existe aucun algorithme de classification de modulation dans la littérature qui prenne en compte à la fois des canaux de trajets multiples (évanouissements sélectifs en fréquence) et des grandes modulations linéaires (ce qui empêche donc de comparer les nouveaux algorithmes à l'état de l'art).

Dans notre travail, nous proposons un algorithme traitant ce problème de type maximum de vraisemblance. Nous n'avons pas considéré les méthodes basées sur l'extraction de caractéristiques des modulations à partir des signaux reçus, car tous les algorithmes connus dans ce cadre s'appliquent uniquement aux canaux d'un seul trajet et il ne semble pas a priori possible de généraliser ces méthodes au cas des canaux de trajets multiples. Dans la suite et dans les simulations, les algorithmes sont expliqués et testés pour des modulations MAQ possédant des nombres d'états différents. Les algorithmes d'estimation des symboles néanmoins s'appliquent également à d'autres modulations linéaires, notamment aux modulations PSK, et les algorithmes d'estimation de l'ordre sont également capable de prendre en compte à la fois les modulations MAQ ayant des nombres d'états différents et d'autres modulations linéaires.

L'idée de l'estimation du nombre d'états de la modulation par maximum de vraisemblance consiste à évaluer et à comparer la vraisemblance de la suite des observations reçues dans chacun des modèles possibles (par exemple : MAQ-4, MAQ-16, MAQ-32, MAQ-64, MAQ-128 et MAQ-256) et de prendre la décision en faveur de celui qui maximise la vraisemblance (en prenant en compte des corrections additives afin d'obtenir une estimation non biaisée).

L'évaluation de la vraisemblances nécessite de disposer d'un modèle mathématique précis de la suite des symboles émis et des observations reçues. De façon synthétique, ce modèle consiste à exprimer le signal reçu comme une convolution de la réponse impulsionnelle du canal et du signal émis plus un bruit additif, étant donné l'ordre de la réponse impulsionelle. Il est utile pour mettre en oeuvre les algorithmes de décrire ce modèle comme une chaîne de Markov cachée, ce qui simplifie les notations et les calculs. Les paramètres inconnus sont donc le nombre d'états de la modulation, l'ordre de la réponse impulsionnelle, les coefficients de la réponse impulsionelle et l'écart-type du bruit.

L'ordre de la réponse impulsionelle peut être traité comme une autre incertitude de l'ordre du modèle. Puisque l'estimateur du maximum de vraisemblance compare les vraisemblances de chaque modèle potentiel - notamment chaque modulation potentielle couplée à chaque ordre de canal -, il est nécessaire de considérer uniquement un petit nombre d'ordres différents de la réponse impulsionelle pour que le nombre total de modèles potentiels reste raisonnable.

L'estimation de l'ordre est basée sur le principe de '*Minimum Description Length*' (MDL) proposé par Rissanen [1978]. Ce critère suggère de choisir le modèle qui est associé à la description la plus parcimonieuse des données. En d'autres termes, si deux modèles décrivent les données de façon statistiquement équivalente, on va choisir le modèle le plus parcimonieux en termes de nombres de paramètres. Une définition commune du MDL est le *critère d'information Bayésien* ou *Bayesian Information Criterion* (BIC) qui empêche la surestimation de l'ordre. En général, l'estimation d'ordre d'un modèle directement basé sur la vraisemblance (sans correction) dans chacun des modèles aurait une tendance à privilégier les modèles les plus complexes, car souvent les modèles d'ordre faible sont une sous-classe des modèles d'un ordre plus élevé. Par exemple, si on considère les réponses impulsionelles du canal, le modèle d'ordre k est une sous-classe du modèle l pour $l > k$, car si on met $l - k$ coefficients égaux à 0, on obtient effectivement un exemple du modèle d'ordre k. Pour cette raison, les vraisemblances dans chaque modèle sont pénalisées dans le BIC par rapport aux nombres des paramètres inconnus que le modèle correspondant inclus. Le nombre de paramètres est donc égal à l'ordre du canal (l'ordre du canal est le nombre des coefficients de la réponse impulsionelle) plus un paramètre pour l'écart-type. Étant donné l'ordre du canal, une pénalisation par rapport au nombre d'états de la modulation n'est donc pas nécessaire, parce que le nombre des paramètres du modèle est indépendant de la modulation.

Pour les modulations à grands nombres d'états ou lorsque l'ordre du canal est élevé, il est difficile d'évaluer de façon exacte la vraisemblance des observations dans chaque modèle de façon explicite. Pour un nombre d'états de modulation et un ordre du canal fixés, la vraisemblance dépend des paramètres du canal et de l'écart-type du bruit; il est donc nécessaire d'estimer ces différentes quantités pour déterminer la vraisemblance des observations. Ce problème est connu sous le nom de l'identification aveugle (*Blind Identification*). Si le nombre d'états de la modulation et l'ordre du canal sont assez faibles, l'algorithme EM couplé à l'algorithme de Baum-Welch est la méthode de référence pour calculer l'estimateur de maximum vraisemblance des paramètres du canal et de l'écart-type. Puisque la complexité de l'algorithme Baum-Welch est de l'ordre $\mathcal{O}(d^L)$, où d est le nombre d'états de la modulation et L est l'ordre du canal, l'algorithme devient trop coûteux en temps de calcul en considérant des modulations à partir de MAQ-16 et des ordres de canal à partir de 3 ou 4.

Pour pouvoir appliquer les mêmes principes, nous avons tout d'abord étudié des algorithmes qui approchent les mêmes quantités que l'algorithme de Baum-Welch tout en ayant un coût de calcul réduit. La quantité centrale dans l'algorithme de Baum-Wech est la distribution de lissage des symboles. Cette quantité est essentielle pour mettre à jour les paramètres dans la partie de la maximisation (M-step) de l'algorithme EM. Nous avons donc consacré la plupart de l'année dernière à définir des méthodes numériquement efficaces pour approcher la distribution de lissage marginal des symboles. L'état de l'art est l'algorithme EMVA (2003) Nguyen and Levy [2003, 2005]. Il emploie l'algorithme Viterbi ou des variantes approximatives comme l'algorithme M ou T (M- and T-algorithm). Nous nous sommes concentrés sur les méthodes d'approximations par systèmes de particules en interactions. Cette méthode est calquée sur l'algorithme de Baum-Welch, qui consiste à estimer en une passe avant la loi de filtrage des symboles conditionnellement aux observations jusqu'à l'instant courant, puis d'évaluer dans une passe arrière la vraisemblance des observations futures étant donné l'état courant; ces deux quantités sont ensuite combinées pour déterminer la distribution de lissage marginale. Dans l'approche particulaire, on approche tout d'abord dans la passe avant la loi de filtrage. des symboles. L'algorithme Baum-Welch pour la mise à jour de la loi de filtrage consiste à explore à chaque instant tous les états possible de la variable cachée, qui est ici une variable discrète dont le nombre d'états possibles est égale à d^L. L'algorithme particulaire approche cette probabilité de filtrage en maintenant un ensemble d'états supports (dont le nombre est significativement plus faible que d^L) et en mettant à zéros les probabilités des autres valeurs possibles de l'état. En procédant de la sorte, on explore uniquement qu'une partie (disons un échantillon de N particules) des états possibles à chaque instant; on assigne de plus des poids d'importance aux particules retenues. Cette approximation de la loi de filtrage est construite de façon récursive en temps. Comme l'espace des états est fini, on procède à l'instant suivant en considérant tous les descendants possibles de l'ensemble des particules actuelles et en calculant des poids d'importance pour ces nouvelles particules. Un sous-ensemble des descendants est ensuite sélectionné et repondéré pour maintenir à l'instant suivant une approximation particulaire possédant N de support. Après avoir mis en œuvre l'algorithme particulaire de filtrage, les distributions de lissage (=distribution des symboles étant donné toute la suite des observations) peuvent être approchées en mettant en oeuvre différentes méthodes.

Nous avons étudié trois algorithmes de lissage déjà existants, mais qui n'avaient jamais été encore appliqués dans ce cadre. Nous avons également proposé un nouvel algorithme de lissage qui utilise deux algorithmes particulaires de filtrage, un dans le sens avant et l'autre dans le sens arrière, renversé en temps. Les particules de ces deux algorithmes sont réunies afin d'approcher les lois de lissage. Ces travaux ont donné lieu à un article de conférence déjà publié et un article de journal en révision favorable. Un article comparant les performances théoriques des algorithmes de lissage est en cours de préparation. En marge de la comparaison des algorithmes de lissage, nous avons été amené à étudier différentes questions. Tout d'abord, nous avons étudié l'influence des méthodes de choix de la population initiale de particules. Nous avons proposé différentes méthodes, mais nous avons montré par simulation qu'il n'est pas possible d'améliorer significativement les résultats des algorithmes par rapport à la méthode la plus élémentaire qui consiste à sélectionner une population initiale en tirant dans l'ensemble de tous les états possibles suivant une loi uniforme. Nous avons aussi étudié soigneusement les méthodes de sélection des particules. A chaque étape, nous considérons l'ensemble des descendants de la génération courante: chaque particule de l'état courant engendre ainsi d descendants et il est donc nécessaire de sélectionner parmi cet ensemble des descendants un sous-ensemble afin de respecter une borne supérieure du nombre des particules. Nous avons généralisé une méthode initialement proposée par Fearnhead Fearnhead and Clifford [2003] pour l'analyse de certains modèles d'états conditionnellement linéaires et Gaussiens. Notre méthode permet de sélectionner les particules et d'assigner des poids d'importance de telle sorte à minimiser une distance (ou divergence) entre la distribution des descendants et la distribution des particules sélectionnées, sous des contraintes d'absences de biais (nous approchons une distribution déterministe par une distribution aléatoire, qui est égale en moyenne à la distribution déterministe et qui minimise, en moyenne, un critère de distance).

Les résultats que nous avons obtenu au cours des nombreuses expériences de Monte-Carlo que nous avons effectué montrent que la meilleure méthode est un algorithme de lissage à horizon fixe basé sur un algorithme particulaire de filtrage marginal et une correction des poids dans une passe arrière et couplé avec une méthode de sélection aléatoire des particules qui minimise soit la norme $L2$ ou soit la distance du chi-2. Nous avons décidé de prendre la méthode minimisant la norme $L2$, parce qu'elle ne nécessite pas le calcul des racines pour tous les poids, ce qui prendrait plus du temps pour la méthode minimisant la distance du chi-2. Nous avons également amélioré l'algorithme de lissage en changeant le sens du filtrage entre chaque initialisation, c'est-à-dire pour toutes les deux initialisations de l'algorithme EM, nous utilisons un algorithme de filtrage en sens avant et la correction des poids dans le sens arrière. Pour les autres initialisations nous utilisons l'algorithme de filtrage dans le sens renversé et ensuite les poids sont corrigés dans une passe avant. Cette démarche empêche des dégénérescences dans les cas de canaux dont le premier coefficient est plus faible que le dernier. Pour plus de détails sur les algorithmes, voir Chapitre 1.2.1. Nous avons avancé dans notre compréhension des algorithmes de lissage et nous proposerons à l'avenir des solutions encore plus performantes de lissage itératif.

Finalement, une méthode a été proposée pour améliorer l'initialisation des paramètres par l'algorithme EM pour les modulations à grand nombre d'états. Il est bien connu que Le choix des valeurs initiales des paramètres inconnus est assez important pour la convergence de l'algorithme

EM. Si l'initialisation est très éloignée du canal "réel", la convergence de l'algorithme est en général plus lente car plus d'itérations sont nécessaires afin d'atteindre la précision requise. De plus, l'algorithme en présence d'une mauvaise initialisation risque de converger vers un maximum local de la vraisemblance éloigné du maximum global qu'on cherche. Cette faiblesse est d'autant plus marquée que le nombre d'états de la modulation est élevé: si le nombre d'états de la modulation est faible, le modèle est plus robuste et on peut explorer une plus grande partie de l'ensemble des états dans chaque instant; cet ensemble de faits concordent pour favoriser la convergence des itérations de l'EM vers un maximum global de la vraisemblance. Nous avons montré qu'il est dans ce contexte intéressant d'utiliser l'estimateur du canal dans un modèle biaisé petit (par exemple 16-QAM) comme valeur initiale pour le ńvraiż plus grand modèle (par exemple 64-QAM ou 256-QAM). Si on ajuste l'échelle des symboles d'une modulation MAQ-16, chaque symbole est associé à quatre symboles d'une MAQ-64 et il se trouve au milieu de ces quatre symboles. Cette idée est nouvelle et elle n'avait jamais été abordée jusqu'à présent.

1.2.1 Déscription du modèle

Soit $\mathcal{X}$ l'alphabet des symboles (d'une modulation linéaire). Nous notons $d = |\mathcal{X}|$ le cardinal de cet alphabet. Nous supposons dans la suite que nous disposons d'une suite i.i.d. de symboles $a_k \in \mathcal{X}$. Nous ne prenons pas en compte la dépendance liée au codage; connaître le codage est une force, qui peut être exploitée dans une approche de turbo-égalisation, mais cette information n'est pas disponible dans un contexte autodidacte. Il serait intéressant, mais très ambitieux d'essayer d'identifier la procédure de codage en utilisant de multiples blocs, mais cette recherche (très risquée) n'a pas été entreprise.

Nous considérons une modulation linéaire en présence d'un bruit additif et soumise à un évanouissement sélectif en fréquence. Nous supposons par souci de simplicité que la période d'échantillonnage est égale à la période symbole. Le signal équivalent en temps discret peut être écrit comme la convolution de la suite des symboles par un filtre de réponse impulsionnelle finie. Soit L l'ordre du canal. Nous notons par $h = (h_0, \ldots, h_{L-1})^T$ les coefficients du canal. La suite des observations est donnée par:

$$y_k = \sum_{l=0}^{L-1} a_{k-l} h_l + \varepsilon_k, \quad \varepsilon_k \sim \mathcal{N}_{\mathbb{C}}\left(0, \sigma^2\right),$$

où le bruit ε_k est une suite i.i.d. gaussienne complexe de variance σ^2. Nous pouvons réécrire ce modèle de façon matricielle pour obtenir un modèle HMM:

$$\begin{aligned} s_k &= Q s_{k-1} + \mathbf{w}_k \\ y_k &= h^T s_k + \varepsilon_k, \quad \varepsilon_k \sim \mathcal{N}_{\mathbb{C}}\left(0, \sigma^2\right), \end{aligned}$$

où $s_k = [a_k, a_{k-1}, ..., a_{k-L+1}]^T$ et $\mathbf{w}_k = [a_k, 0, ..., 0]^T$. Q est une matrice de décalage, dont tous les éléments sont nuls à l'exception des éléments situés sous la diagonale principale. L'espace d'états de cette chaîne de Markov $(s_k)_k$ est $\mathcal{S} = \mathcal{X}^L$ et sa dimension est égale à d^L.

Le vecteur des paramètres inconnus est noté $\theta = (h, \sigma)$. Les probabilités de transitions de s_k à s_{k+1} sont données par $q\left(s_{k+1} \left| s_k\right.\right)$. Nous notons $g_k\left(y_k | s; \theta\right) = \frac{1}{\pi\sigma^2} \exp\left(-\frac{1}{\sigma^2}\left|h^T s - y_k\right|^2\right)$ la vraisemblance de l'observation. Pour $k \leq j$, nous notons $s_{k:j} = (s_k, \ldots, s_j)$.

Nous supposons qu'un vecteur d'observations $Y = y_{1:K} = y_{1:K}$ soit donné.

1.2.2 Estimation de l'ordre

Le problème d'estimation de l'ordre du modèle consiste à déterminer le nombre d'états d de la modulation MAQ et l'ordre L du canal. Soit θ_L l'ensemble de vecteurs des paramètres qui décrivent les paramètres du canal lorsque l'ordre du modèle est L. Ce vecteur $\theta \in \theta_L$ se compose des coefficients de la réponse impulsionnelle du canal et l'écart-type du bruit additif. La dimension de ce vecteur de paramètre est égale à $(L+1)$. Chaque type de modulation p-MAQ et chaque ordre du canal L définit une fonction de vraisemblance. Nous notons $\mathcal{M}_{p,L} = \{l_{p,L}(\cdot, \theta_L) | \theta_L \in \Theta_L\}$ l'ensemble de toutes les fonctions de vraisemblance, si bien que pour chaque $\theta_L \in \Theta_L$ la vraisemblance d'une séquence d'observations Y est $l_{p,L}(\cdot, \theta_L)$. Bien que l'ensemble Θ_L indexant le modèle ne dépende pas du nombre d'états de la modulation d, le modèle $\mathcal{M}_{p,L}$ et donc la fonction de vraisemblance dépend bien évidemment de l'alphabet de modulation. Nous notons $\hat{\theta}_{p,L}$ l'estimateur du maximum de vraisemblance pour le modèle $\mathcal{M}_{p,L}$ pour une séquence donnée d'observations Y.

Comme critère de l'estimation de modèle, nous utilisons le critère d'information de Bayes (BIC):

$$\mathrm{BIC}(\mathcal{M}_{p,L}) = -\log l\left(Y, \hat{\theta}_{p,L}\right) + \tfrac{\mathrm{Dim}(\Theta_L)}{2} \log K$$

L'estimateur du modèle basé sur le BIC est donc donné par

$$\hat{\mathcal{M}}_{\mathrm{BIC}} = \underset{\mathcal{M}_{p,L}}{\arg\min} \left\{ -\log l\left(Y, \hat{\theta}_{p,L}\right) + \tfrac{L+1}{2} \log K \right\}.$$

Pour mettre en oeuvre cette méthode, il est par conséquent nécessaire de déterminer l'estimateur du maximum de vraisemblance des paramètres du modèle. Comme nous l'avons mentionné précédemment, il est en général trop complexe en termes de calculs de déterminer l'estimateur du maximum de vraisemblance exact, mais il est toutefois possible de mettre en oeuvre des algorithmes efficaces pour calculer des estimateurs de maximum de vraisemblance approchés. Nous utilisons dans la suite l'algorithme de lissage à horizon fixe, dont nous avons démontré qu'il réalisait le meilleur compromis coût/complexité. Bien que cet algorithme soit plus efficace que les algorithmes de maximum de vraisemblance approchés proposés à ce jour, cet algorithme est toutefois coûteux si nous devons l'appliquer pour chaque modèle. Ceci est d'autant plus compliqué, qu'en l'absence d'une initialisation fiable (ce qui sera le cas lorsque les rapports signaux à bruit sont défavorables et / ou que la taille des bursts est limité), il est nécessaire pour chaque modèle d'utiliser des initialisations multiples pour l'estimateur du maximum de vraisemblance pour réduire la probabilité de converger vers une solution sous-optimale correspondant à un maximum local du critère.

1.2.3 Méthode du maximum de vraisemblance

Pour l'estimation des paramètres inconnus θ, on emploie le principe du maximum de vraisemblance, c'est-à-dire on considère la valeur des paramètres qui maximise la fonction de vraisemblance des observations comme la meilleure description du modèle, étant donnée la séquence d'observations. L'estimateur du maximum de vraisemblance du paramètre θ est donc le maximum de la fonction $\theta \to l(Y;\theta)$, où $l(Y;\theta)$ est la vraisemblance de la suite d'observations Y.

Puisque le maximum de vraisemblance ne peut pas être calculé de façon explicite dans ce modèle, on est obligé de mettre en œuvre un algorithme itératif. L'algorithme EM est la méthode de référence pour calculer cet estimateur. Cet algorithme prend en paramètre une valeur initiale, qui est ensuite mise a jour de manière itérative jusqu'à la convergence. L'algorithme est globalement convergent, mais il ne converge pas nécessairement vers un maximum absolu de la fonction de vraisemblance (il peut converger dans certaines conditions vers un attracteur local). Pour éviter la convergence vers un maximum local, l'algorithme est initialisé plusieurs fois avec des valeurs différentes. Chaque itération de l'algorithme EM consiste en deux étapes : l'étape E(xpectation) et l'étape M(aximisation). La quantité centrale, qui est calculée au cours de l'étape E, est la probabilité de lissage $p_k(\cdot|Y;\theta) = \mathbb{P}\left(s_k = \cdot \,|\, Y;\, \theta\right)$ de l'état s_k, étant données les observations Y et la valeur courante des paramètres θ. Dans des modèles de dimension faible, elle peut être calculée de façon exacte en utilisant l'algorithme de Baum-Welch. Par contre, dans des modulations plus grandes (à partir de MAQ-16 ou MAQ-32), la complexité des calculs de cet algorithme devient rapidement trop grande. Suivant les résultats numériques des rapports précédents, on a décidé d'approcher les lois de lissage à l'aide d'un système de particules en interractions pour réduire la complexité de calcul. Par souci de concision, nous notons $p_k(s|y_{1:k};\theta) = \mathbb{P}\left(s_k = s|y_{1:k};\theta\right)$.

1.2.4 Lissage à horizon fixe

L'algorithme particulaire de lissage a horizon fixe est utilisé pour implémenter l'étape E de l'algorithme EM. On peut donc supposer une estimation courante du vecteur de paramètres θ. Étant donné cette estimation, l'algorithme estime les lois marginales de lissage $p_k(\cdot|Y;\theta)$ en utilisant un filtre particulaire qui estime la loi de filtrage $p_k(s|y_{1:k};\theta)$ pour $k \leq K$ dans une première étape et une adaptation aux lois de lissage dans la deuxième étape. Le filtre particulaire établit un système de particules auxquels sont affectés des poids d'importance. Ces particules et ces poids sont calculés par un algorithme récursif en temps. En utilisant cette estimation de la loi de filtrage, l'estimateur de la loi de lissage à horizon fixe $p_k(\cdot|Y;\theta)$ consiste à corriger les poids d'importance calculés au cours de la passe ńavantż dans une passe ńarrièreż.

Supposons qu'à l'instant k la loi de filtrage est approchée par

$$\hat{p}_k(s|y_{1:k};\theta) = \sum_{i=0}^{N-1} w_k^i \delta_{\xi_k^i}(s) \tag{1.1}$$

où $\xi_k^i \in \mathcal{S}$ sont les particules, w_k^i sont les poids d'importance, et $\delta_{\xi_k^i}(s) = 1$, si $\xi_k^i = s$ et 0 autrement. Etant donné $\hat{p}_k(s|y_{1:k};\theta)$, la loi de filtrage à l'instant $(k+1)$ est approchée par

$$\hat{p}_{k+1}(s|y_{1:k+1};\theta) \propto \sum_{i=0}^{M-1} \tilde{w}_{k+1}^i \, \delta_{\tilde{\xi}_{k+1}^i}(s),$$

où $\tilde{\xi}_{k+1}^i$ sont les descendants des particules courantes et $\tilde{w}_{k+1}^i$ sont les poids mis à jour, donnés par:

$$\begin{aligned}
\tilde{w}_{k+1}^i &= \sum_{j=0}^{N-1} q\left(\tilde{\xi}_{k+1}^i \middle| \xi_k^j\right) w_k^j g_{k+1}\left(y_{k+1}|\tilde{\xi}_{k+1}^i;\theta\right) \\
&= \frac{1}{d} \sum_{j \in \mathcal{P}(\tilde{\xi}_{k+1}^i)} w_k^j g_{k+1}\left(y_{k+1}|\tilde{\xi}_{k+1}^i;\theta\right), \qquad (1.2)
\end{aligned}$$

Dans cette dernière expression, $\mathcal{P}(\tilde{\xi}_{k+1}^i)$ sont les particules parents de $\tilde{\xi}_{k+1}^i$. Cette expression est basée sur la décomposition suivante de la loi de filtrage $p_{k+1}(s|y_{1:k+1};\theta)$ en réutilisant $p_k(\cdot|y_{1:k};\theta)$:

$$\begin{aligned}
p_{k+1}(s|y_{1:k+1};\theta) &= \sum_{s' \in \mathcal{S}} \mathbb{P}\left(s_k = s', s_{k+1} = s|y_{1:k+1};\theta\right) \\
&\propto g_{k+1}\left(y_{k+1}|s;\theta\right) \sum_{s' \in \mathcal{S}} q\left(s\,|s'\right) p_k(s'|y_{1:k};\theta).
\end{aligned}$$

Le nombre de calculs nécessaires pour évaluer les poids proposés croît de façon linéaire en fonction du nombre de particules (voir le rapport précédent). Les nouvelles particules $(\xi_{k+1}^i)_{i<N}$ sont obtenues en retenant N particules de l'ensemble $\tilde{\xi}_{k+1}^i$, en utilisant un algorithme de sélection, puis en renormalisant les poids.

Dans la première approche, nous estimons la loi de lissage en conservant les points de support de la loi de filtrage en mettant à jour les poids d'importance. L'algorithme que nous utilisons est basé sur la décomposition suivante de la loi de lissage $p_k(s|Y;\theta)$:

$$p_k(s|Y;\theta) = \sum_{x\in\mathcal{S}} \mathbb{P}\left(s_k = s, s_{k+1} = x \,|\, Y;\, \theta\right) = p_k(s|y_{1:k};\theta) \sum_{x\in\mathcal{S}} \frac{p_{k+1}(x|Y;\theta)q\left(x\,|s\,\right)}{\sum_{z\in\mathcal{S}} q\left(x\,|z\,\right) p_k(z|y_{1:k};\theta)} \tag{1.3}$$

Algorithm 1: Algorithme particulaire de lissage à horizon fixe

```
Input  : $Y$        la suite des observations
         $L$        ordre du canal
         $N$        nombre de particules
         $\mathcal{X}$        alphabet de la modulation
         $\theta$        estimation actuelle de paramètres inconnu ($\theta = (h, \sigma)$)
Output: $\hat{p}_k(\cdot|Y;\theta)$   Estimateur des lois de lissage pour $k \in \{1, \cdots, K\}$
           $\hat{l}(Y)$   Estimateur de la vraisemblance de la suite des observations

• Instant initial ;
    ∗ Échantillonner $\xi_1^i \sim \mathcal{U}(\mathcal{X})$ et calculer $\tilde{w}_1^i = g_1\left(y_1|\xi_1^i;\theta\right)$ pour $i \in \{0, \cdots, N-1\}$.
    ∗ Mettre $l_1 = \sum_{i=0}^{N-1} \tilde{w}_1^i$ et $w_1^i = (l_1)^{-1}\tilde{w}_1^i$ pour $i \in \{0, \cdots, N-1\}$.
• Itérations en sens avant ;
  for $k$ de 1 à $(K-1)$ do
      ∗ Proposition des particules ;
        Soient $\tilde{\xi}_{k+1}^i$ pour $i \in \{0, \cdots, M_{k+1}-1\}$ les $M_{k+1}$ descendants de $\left(\xi_k^j\right)_{j<N}$. ;
        Calculer
            $\tilde{w}_{k+1}^i = \sum_{j=0}^{N-1} q\left(\tilde{\xi}_{k+1}^i \,\middle|\, \xi_k^j\right) w_k^j g_{k+1}\left(y_{k+1}|\tilde{\xi}_{k+1}^i;\theta\right)$
        Mettre $l_{k+1} = \sum_{i=0}^{M-1} \tilde{w}_{k+1}^i$ et $\breve{w}_{k+1}^i = (l_{k+1})^{-1}\tilde{w}_{k+1}^i$.
      ∗ Sélection des particules, voir la procédure (1.5) ;
        Input  : $\left(\tilde{\xi}_{k+1}^i, \breve{w}_{k+1}^j\right)_{i\in\{0,\cdots,M-1\}}$
        Output: $N$ positions $\xi_{k+1}^i$ avec des poids $w_{k+1}^i$
  end
• Itérations en sens arrière ;
  for $k$ de $(K-1)$ à 1 do
      $w_{k|K}^i = w_k^i \sum_{j=0}^{N-1} \frac{w_{k+1|K}^j q\left(\xi_{k+1}^j \,|\xi_k^i\right)}{\sum_{l=0}^{N-1} w_k^l q\left(\xi_{k+1}^j \,|\xi_k^l\right)}$
  end
• Estimation de la vraisemblance : $\hat{l}(Y) = \prod_{k=0}^{K} l_k$
```

Cette décomposition permet de calculer $p_k(\cdot|Y;\theta)$ récursivement en utilisant la probabilité de filtrage $p_k(\cdot|y_{1:k};\theta)$ et de la loi de lissage courante $p_{k+1}(\cdot|Y;\theta)$. Dans cette approche, l'approximation de la loi de lissage est donnée à l'instant k par:

$$\hat{p}_k(s|Y;\theta) = \sum_{i=0}^{N-1} w^i_{k|K} \delta_{\xi^i_k}(s)\,.$$

où $w^i_{k|K}$ sont les poids de lissage. L'équation (1.3) suggère l'algorithme de mise à jour récursif des poids de lissage suivant

$$w^i_{k|K} = w^i_k \sum_{j \in \mathcal{Q}(\xi^i_k)} \frac{w^j_{k+1|K}}{\sum_{l \in \mathcal{P}(\xi^j_{k+1})} w^l_k}, \tag{1.4}$$

Nous allons maintenant présenter une méthode de sélection de particules minimisant la distance de Kullback-Leibler. Étant donné un vecteur de probabilités $\mathbf{w} = (w_i)_{i<M}$ de taille M tel que $M > N$, une distribution $\mathbf{W} = (W_i)_{i<M}$, minimisant la distance de Kullback-Leibler vers la distribution $\mathbf{w}$ est donnée par :

Sélection aléatoire: divergence de Kullback-Leibler

Soit λ une solution de l'équation

$$\sum_{i=0}^{M-1} \min\left\{\frac{w_i}{\lambda}, 1\right\} = N.$$

Pour tout indice i tel que $w_i \geq \lambda$, posons

$$W_i = w_i,$$

et pour tout i tel que $w_i < \lambda$, posons

$$W_i = \begin{cases} \lambda & \text{with probability } \frac{w_i}{\lambda} \\ 0 & \text{with probability } 1 - \frac{w_i}{\lambda}. \end{cases} \tag{1.5}$$

Finalement, l'algorithme complet de lissage est donné dans l'Algorithme 1.

1.2.5 Un nouvel algorithme de lissage de deux filtres

Dans un modèle HMM, le processus conjoint (s_k, y_k) pour $k = 1, \cdots, K$ des états cachés et des symboles est une chaîne de Markov; cette propriété est préservée par retournement du sens du temps: (s_k, y_k) pour $k = K, K-1, \cdots, 1$ est aussi une chaîne de Markov. L'algorithme de filtrage (marginal) peut être appliqué dans en sens rétrograde pour estimer un équivalent en temps retourné de la distribution de filtrage, à savoir: $p_k(\cdot|y_{k:K};\theta) = \mathbb{P}(s_k = \cdot|y_{k:K};\theta)$ for $k = K, K-1 \cdots, 1$ pour $k = K, K-1 \cdots, 0$, basée sur la decomposition suivante:

$$\begin{aligned} p_k(s|y_{k:K};\theta) &\propto g_k(y_k|s;\theta)\,\mathbb{P}(s_k = s|y_{k+1:K};\theta) \\ &\propto g_k(y_k|s;\theta) \sum_{s' \in \mathcal{S}} p_{k+1}(s'|y_{k+1:K};\theta) q(s'|s)\,. \end{aligned} \tag{1.6}$$

Pour simplifier, supposons que nous disposions d'une approximation d'un filtre particulaire $\left(\xi^i_k,\, w^i_k\right)_{i<N}$ avec des poids non-normalisés, telle que

$$\hat{\pi}_k(s|y_{1:k};\theta) = \sum_{i=0}^{N-1} w^i_k \delta_{\xi^i_k}(s)$$

approxime

$$\pi_k(s|y_{1:k};\theta) = \sum_{s'\in\mathcal{S}} g_k\left(y_k|s;\theta\right) q\left(s\,\middle|s'\right) p_{k-1}(s'|y_{1:k-1};\theta).$$

Soit $\Omega_k = \sum_{i=0}^{N-1} w_k^i$. Évidemment , $\hat{\pi}_k(s|y_{1:k};\theta)/\Omega_k$ est une approximation de la loi de filtrage $p_k(s|y_{1:k};\theta)$. De façon équivalente, soit $\left(\xi^i_{k|k:K}, w^i_{k|k:K}\right)_{i<N}$ une approximation de la loi non-normalisée de filtrage renversé

$$\pi_k(s|y_{k:K};\theta) = g_k\left(y_k|s;\theta\right) \sum_{s'\in\mathcal{S}} p_{k+1}(s'|y_{k+1:K};\theta) q\left(s'\,|s\right).$$

Nous proposons donc d'améliorer ces approximations des lois de filtrage avant et arrière (non-normalisées) en enrichissant leurs supports, c'est-à-dire en ajoutant les positions des particules arrière et avant respectivement. Nous mettons en évidence maintenant de façon plus formelle cette procédure. Supposons que nous disposions d'une approximation particulaire "améliorée" de la loi de filtrage (non-normalisée) $\tilde{\pi}_{k-1}(\cdot|y_{1:k-1};\theta)$ à l'instant $k-1$, consistant en les particules $\left(\tilde{\xi}^i_{k-1}, \tilde{w}^i_{k-1}\right)_{i<N_{k-1}}$.

Soit $(\tilde{\xi}^i_k)_{i<N_k}$ l'ensemble des positions particulaires uniques de $\{\xi^i_k\}_{i<N}\cup\{\xi^i_{k|k:K}\}_{i<N}$, ordonné pour que $\tilde{\xi}^i_k = \xi^i_k$ for $i<N$.

Une approximation "améliorée" de $\pi_k(s|y_{1:k};\theta)$ est donc donnée par

$$\begin{aligned} \tilde{\pi}_k(s|y_{1:k};\theta) &= \sum_{i=0}^{N_k-1} \tilde{w}^i_k \delta_{\tilde{\xi}^i_k}(s), \\ &= p_k(s|y_{1:k};\theta) + \sum_{i=N}^{N_k-1} \tilde{w}^i_k \delta_{\tilde{\xi}^i_k}(s), \end{aligned} \tag{1.7}$$

où nous définissons $\tilde{w}^i_k = w^i_k$ pour $i<N$ et nous calculons les poids de filtrage $\tilde{w}^i_k$ des nouvelles positions des particules $\tilde{\xi}^i_k$, for $i \geq N$, pour que les probabilités de lissage de ces symboles soient approchées de façon optimale, étant donné $y_{1:k}$ et $\tilde{\pi}_{k-1}(\cdot|y_{1:k-1};\theta)$:

$$\tilde{w}^i_k = \sum_{j=0}^{N_{k-1}} g_k\left(y_k|\tilde{\xi}^i_k;\theta\right) q\left(\tilde{\xi}^i_k\,\middle|\tilde{\xi}^j_{k-1}\right) \frac{\tilde{w}^j_{k-1}}{\tilde{\Omega}_{k-1}}, \tag{1.8}$$

Définissons $\tilde{\Omega}_{k-1} = \sum_{i=0}^{N_{k-1}-1} \tilde{w}^i_{k-1}$. Nous utilisons les poids non-normalisés plutôt que des poids normalisés pour préserver l'information cachée dans le facteur de normalisation et pour éviter la sous- ou la surestimation de l'importance des nouvelles positions particulaires.

De façon semblable, des approximations améliorées de la passe arrière$\tilde{\pi}_k(\cdot|y_{k:K};\theta)$ de $\pi_k(\cdot|y_{k:K};\theta)$, consistant en les $N_{k|k:K}$ particules $\left(\tilde{\xi}^i_{k|k:K}, \tilde{w}^i_{k|k:K}\right)_{i<N_{k|k:K}}$ peuvent être calculées de façon itérative, en se basant sur la décomposition (1.6). Nous définissons encore $\tilde{\Omega}_{k|k:K} = \sum_{i=0}^{N_{k|k:K}-1} \tilde{w}^i_{k|k:K}$.

Maintenant, nous disposons des approximations mises à jour $\tilde{\pi}_k(\cdot|y_{1:k};\theta)/\tilde{\Omega}_k$ de la loi de filtrage $p_k(\cdot|y_{1:k};\theta)$ de la passe avant et $\tilde{\pi}_k(\cdot|y_{k:K};\theta)/\tilde{\Omega}_{k|k:K}$ de la loi de filtrage $p_k(\cdot|y_{k:K};\theta)$ dans la passe arrière. Nous pouvons en déduire une approximation de la loi de lissage:

1. **Réitérer la passe avant**, donnant $\left(\tilde{\xi}^i_k, \tilde{w}^i_k\right)_{i<N_k}$.

2. **Réitérer la passe arrière**, donnant $\left(\tilde{\xi}^i_{k|k:K}, \tilde{w}^i_{k|k:K}\right)_{i<N_{k|k:K}}$.

3. **Approximer la loi de lissage**. Calculer les poids

$$w^i_{k|K} = \tilde{w}^i_k \sum_{j=0}^{N_{k|k:K}-1} q\left(\tilde{\xi}^j_{k|k:K}\,\middle|\tilde{\xi}^i_k\right) \tilde{w}^i_{k|k:K},$$

et la constante de normalisation $\Omega_{k|K} = \sum_{i=0}^{N_k-1} w^i_{k|K}$. Alors,

$$\hat{p}_k(s|Y;\theta) = \frac{1}{\Omega_{k|K}} \sum_{i=0}^{N_k-1} \tilde{w}^i_{k|K} \delta_{\tilde{\xi}^i_k}(s).$$

La complexité des pas 1) et 2) est $\mathcal{O}(dN)$. Les pas 3) et 4) incluent chacun $\mathcal{O}(N)$ calculs par l'instant k, étant équivalents au calcul des poids de filtrage marginal dans (1.2). De façon similaire aux autres algorithmes, la réorganisation des données pour créer des ensembles des prédécesseurs et successeurs est de la complexité $\mathcal{O}(N \log N)$. Le dernier pas 5) est de la complexité $\mathcal{O}(2N \log(2N))$. En pratique, les deux premiers pas sont donc les plus complexes.

1.2.6 Simulations numériques

Nous considérons dans ces simulations les taux d'erreurs pour des données engendrées par le simulateur développé dans le cadre du projet. Les alphabets des modèles MAQ (MAQ-4, MAQ-16, MAQ-32, MAQ-64, MAQ-128, MAQ-256) sont normalisés pour que la puissance moyenne des symboles soit égale à 1. Les canaux considéré sont RAx, TUx, BUx et HTx. Nous fixons le rapport signal sur bruit (RSB) de chaque constellation pour que le taux d'erreur théorique soit égal à $1\text{e}-3$. L'excès de bande est fixé à $\lambda = 0.5$.

Nous commençons par présenter les résultats d'estimation des taux d'erreurs des symboles pour des canaux à temps fixe. Nous comparons nos algorithmes à l'algorithme CMA par blocs, qui est considéré comme l'état de l'art dans le contexte de la déconvolution autodidacte. Afin de pouvoir employer le CMA, il est nécessaire que le nombre d'états de la modulation soit connu. Par contre, l'ordre du canal n'a pas besoin d'être connu a priori et il est estimé parmi les ordres potentiels $L = 3, 5, 8$. Afin de réduire la complexité des simulations, nous réduisons la liste des ordres potentiels. Notons que l'algorithme CMA ajuste directement un filtre de déconvolution et ne procède donc pas à proprement parler à une estimation du canal.

Les résultats dépendent de façon très significative des choix de paramètres comme le nombre de particules, le nombre d'itérations et le nombre maximal d'itérations de l'algorithme EM approché. Les taux d'erreurs s'améliorent si on augmente le nombre de particules et le nombre d'itérations de l'algorithme EM. Mais l'augmentation de ces quantités augmente la complexité, de sorte que le choix des paramètres est un compromis entre le temps de calcul et la performance des méthodes d'estimation, mesurée au moyen du taux d'erreurs. Par exemple, il est préférable d'utiliser des ordres faibles du canal (par exemple $L = 3$), car l'estimation dans un tel modèle est plus simple et on peut donc diminuer à la fois le nombre de particules et le nombre d'itérations de l'algorithme EM. Les taux d'erreurs doivent donc être comparés en prenant en considération le choix correspondant des paramètres utilisés pour la simulation.

Taux d'erreurs pour le modèle MAQ-16

Paramètre	**Ordre du canal**		
	$L = 3$	$L = 5$	$L = 8$
N	100	500	500
I	2	4	4
η	50		

Table 1.1: Paramètres de simulation pour la modulation MAQ-16: N est le nombre de particules, I est le nombre d'initialisations et η est le nombre maximal des itérations

Le choix des paramètres pour la modulation MAQ-16 est donné dans la table 1.1. Dans les tables suivantes les colonnes correspondent à des ordres du canal avec lesquelles l'estimation a été effectuée. $L = \hat{L}$ veut dire que, pour chaque simulation, l'ordre du canal est estimé (parmi une liste d'ordre de modèles possibles) et que le taux d'erreur est ensuite estimé en utilisant cet estimateur du canal. Elles montrent la probabilité d'avoir un taux d'erreurs plus faible que 0.01 pour différents modèles de canaux et pour une longueur de la trame égal à $K = 500$.

Algorithme	**Ordre du canal**			
	$L = 3$	$L = 5$	$L = 8$	$L = \hat{L}$
CMA	88.59 %			
Algorithme Particulaire	74.77 %	79.88%	48.35 %	94.89%
EMVA	75.14 %	40.68 %	51.98%	

Table 1.2: RAx, MAQ-16, $n = 500$, probabilité d'un taux d'erreur <0.01

Algorithme	**Ordre du canal**			
	$L = 3$	$L = 5$	$L = 8$	$L = \hat{L}$
CMA	79.74 %			
Algorithme Particulaire	93.81 %	98.50%	62.10 %	100 %
EMVA	44.84 %	42.21 %	40.52%	%

Table 1.3: BUx, MAQ-16, $n = 500$, probabilité d'un taux d'erreur <0.01

Nous présentons uniquement les résultats obtenus par l'algorithme EM approché basé sur une estimation automatique de l'ordre du canal. Pour le canal Rax (voir 1.4), nous avons uniquement étudié les performances pour un ordre du canal $L = 3$ lorsque la taille du bloc excède $K = 1000$, ce qui explique la baisse de la performance de l'algorithme EM approché pour $K = 1000$, $K = 1500$ et $K = 2000$. Pour des valeurs de longueur de trames aussi longues (on peut s'interroger sur leur pertinence "pratique"),les algorithmes EM approchés sont très complexes et gourmands en mémoire.

Les résultats montrent que l'algorithme EM approché est supérieur à l'algorithme CMA, particulièrement pour les canaux BUx et HTx. Les résultats suivants mettent en évidence la dépendance de la performance des algorithmes par rapport à la longueur de la trame. L'algorithme CMA par bloc est particulièrement pénalisé lorsque la taille des blocs est courte, mais sa performance sur des blocs de taille plus importante est satisfaisante. Les résultats pour le canal TUx sont donnés dans la table 1.4.

Algorithme	**Nombre d'observations**					
	$K = 200$	$K = 300$	$K = 500$	$K = 1000$	$K = 1500$	$K = 2000$
CMA	12.00%	51.48%	97.18 %	100.00%	100.00%	100.00%
Algorithme Particulaire	98.00%	99.41%	98.31 %	98.01 %	98.01%	94.43 %

Table 1.4: TUx, MAQ-16, probabilité d'un taux d'erreur <0.01

Pour tous les canaux, on peut remarquer que la performance de l'algorithme particulaire est satisfaisante pour des trames courtes. Pour des trames longues, l'algorithme CMA redevient compétitif, car les algorithmes particulaires sont alors pénalisés par leur complexité qui croît linéairement en fonction du nombre d'observations.

Choix de modèle

Nous étudions maintenant la performance de la procédure de choix de modèle. Il n'existe aucun autre algorithme dans la littérature capable d'estimer l'ordre de la modulation et du canal pour des modulations autres que la QPSK et BPSK. Nous rapportons dans ce document donc uniquement les résultats pour l'algorithme que nous avons proposé. Nous avons étudié le canal BUx pour une longueur de la trame égale à $K = 300$.

Algorithme	**Vrai modèle**					
	MAQ-4	MAQ-16	MAQ-32	MAQ-64	MAQ-128	MAQ-256
Choix de modèle	98.29 %	98.33 %	65.84 %	99.19%	90.60%	73.80 %

Table 1.5: BUx, K, probabilité de l'estimation du vrai modèle

Dans la table 1.5 nous donnons le pourcentage de simulations pour lesquels le vrai modèle est estimé pour le canal BUx. Nous avons choisi un nombre de particules, un nombre d'itérations et un nombre d'initialisations assez limités et les résultats sont déjà encourageants. Les probabilités de détection pourraient être améliorées en augmentant la complexité de calcul.

1.3 Estimation au sens du maximum de vraisemblance d'une modulation linéaire basée sur un modèle de proportion

Lehmann and Romano [2005] Nous nous sommes intéressés à la question de trouver un estimateur de l'ordre de la modulation ayant une distribution asymptotique plus explicite que la distribution de l'estimateur d'ordre que l'on avait décrit dans les rapports précédents. Nous avons aussi travailler sur les estimateurs de canaux non-stationnaires structurés, permettant de prendre en compte la présence de Doppler ou de résidu de porteuse. Nous sommes en train de finaliser cet aspect du travail et nous allons donc consacrer ce rapport au premier point, qui est aujourd'hui quasiment achevé [il reste à écrire précisément les résultats].

Si on restreint d'estimation de l'ordre du modèle (autrement dit la classification de modèle) à deux modèles potentiels, le problème peut être considéré comme un test d'hypothèses multiples avec des paramètres de nuisance (ici, la variance du bruit et les coefficients des canaux de propagation). L'approche naturelle pour résoudre ce type de problèmes est de mettre en oeuvre un test de rapport de vraisemblance généralisé (GLRT = 'generalized likelihood ratio test'). Nous allons présenter différents tests basés sur cette idée dont on peut (à l'exception du premier) déterminer la distribution asymptotique sous l'hypothèse nulle. L'intérêt de pouvoir calculer cette distribution est de calculer, au moins de façon théorique, une probabilité asymptotique de fausse alarme, et dans certains cas, de construire la courbe COR.

Nous considérons une méthode de maximum de vraisemblance pour la classification aveugle des modulations MAQ transmises sur un canal sélectif en fréquence et perturbées par du bruit additif Gaussien. Des tests de rapport de vraisemblance pour la classification aveugle ont déjà été considérés dans Panagiotou et al. [2000], Dobre and Hameed [2006], Hong and Ho [2000] et d'autres. Les test ont toujours été présenté en tant que problème d'estimation du nombre de paramètres d'un modèle: chaque alphabet de modulation potentiel est associé dans ce contexte à un modèle. En pratique, ces algorithmes reviennent à tester le nombre de composantes dans un modèle de mélanges Gaussiens. En général, les estimateurs de ce type ne sont pas consistants, parce que l'estimation de l'ordre d'un mélange est un problème singulier (les méthodes proposées ne sont d'ailleurs pas consistantes). Nous avons développé et mis en oeuvre une approche différente. Plutôt que de décrire chaque alphabet de modulation par un modèle spécifique, nous considérons ici un seul modèle statistique englobant, qui décrit l'ensemble des modulations. Par rapport aux modèles définis pour chaque modulation individuelle, ce modèle possède un paramètre auxiliaire qui permet de discriminer les différentes modulations. Il est défini de telle sorte que le test de rapport de vraisemblance généralisé testera plutôt les valeurs de paramètre (plutôt que d'évaluer la vraisemblance pour des modèles différents). De plus, nous allons utiliser la structure particulière du modèle afin d'établir l'identifiabilité de ce modèle et développer les propriétés asymptotiques des test proposés.

Pour simplifier la présentation, nous traitons ici le problème de discrimination entre deux modulations potentielles. Nous supposons que l'alphabet $\mathcal{X}_0$ de la modulation du modèle 1 est un sous-ensemble de l'alphabet de modulation $\mathcal{X} = \mathcal{X}_0 \cup \mathcal{X}_1$. Cette approche s'étend naturellement à des situations où le nombre de simulations à discriminer est plus important [en faisant des tests d'hypothèses multiples par paires: il faudra simplement faire attention à la façon de combiner les résultats de tests pour éviter les erreurs]. Dans notre méthode, nous considérons deux modulations MAQ correspondant à deux alphabets de modulation différents et nous introduisons comme paramètre permettant de discriminer les différentes modulations la proportion des symboles (éventuellement, comme nous le discutons dans la suite, il est possible de rajouter une information a priori sur la structure de ces proportions). Si le vrai modèle correspond à l'alphabet global $\mathcal{X}$, les paramètres de tous les symboles sont égaux. Par contre, dans le cas contraire, une partie de ces probabilités doit être égale à 0. Basées sur ces proportions, nous discutons trois statistiques possibles de test.

La première statistique est seulement la reformulation du test que nous avions développé précédemment, mais en utilisant ce nouveau cadre statistique. Pour la deuxième statistiques, l'hypothèse nulle est que le vrai modèle corresponde à la MAQ d'alphabet $\mathcal{X}_0 \cup \mathcal{X}_1$, ce qui implique notamment que toutes les proportions soient égale. L'hypothèse est testée contre l'hypothèse alternative que les proportions ne soient pas égales. Cette statistique a l'avantage que la distribution asymptotique sous l'hypothèse nulle est connue et est libre (une distribution chi-2 centrée dont la distribution ne dépend d'aucun paramètre). Au contraire du premier test, la fausse alarme de la deuxième statistique de test est donc facile à calibrer.

La troisième statistique considère l'hypothèse nulle que le vrai modèle correspond à la MAQ d'alphabet $\mathcal{X}_0$, ce qui implique notamment qu'une partie des proportions est égale à 0. L'hypothèse alternative est donc que toutes les proportions sont différentes de 0. La distribution asymptotique sous l'hypothèse nulle n'est pas aussi facile à établir que dans les deux cas précédents, car les paramètres sous l'hypothèse nulle appartiennent à la frontière de l'espace de paramètres. Nous avons pu montrer qu'elle est une mélange de deux distribution chi-2 en utilisant les résultats de Andrews [2001] sur la distribution des tests de rapport de maximum de vraisemblance généralisés quand les paramètres appartiennent à la frontière de l'espace des paramètres.

Afin de simplifier les notations nous présentons les méthodes pour tester une modulation MAQ-4 contre une MAQ-16. Il est néanmoins facile de généraliser ces méthodes pour d'autres modèles MAQ. De plus, comme nous l'avons déjà mentionné précédemment, les méthodes que nous avons introduites peuvent aussi être généralisées pour le cas de plus que deux modèles potentiels. Nous avons pour l'instant établi les distributions de ces statistiques de tests pour un canal à évanouissement plat; il n'y a guère de doute que les mêmes distributions restent valables pour un canal sélectif en fréquences, mais la technique de preuve sera, dans ce cas, beaucoup plus complexe (et dépasse le cadre de ce projet).

1.3.1 Description de modèle

Nous considérons deux constellations MAQ potentielles, dont la plus petite consiste en l'alphabet $\mathcal{X}_s$ de d_s symboles et la constellation la plus grande ayant l'alphabet $\mathcal{X}_l$ de d_l états. Nous supposons que d_l est un multiple de d_s. Afin de faciliter la présentation, les alphabets ne sont pas normalisés tel que $\mathcal{X}_s$ soit un sous-ensemble de l'alphabet plus grand $\mathcal{X}_l$, i.e. $\mathcal{X}_s \subset \mathcal{X}_l$.

L'espace d'états $\mathcal{X}$ du modèle est égal à $\mathcal{X}_l = \mathcal{X}$. Chaque symbole est transmis avec la même probabilité et indépendamment (ce qui est bien entendu en toute rigueur erroné, mais valide en première approximation). Nous géneralisons ce concept en définissant deux sous-ensembles $\mathcal{X}_0$ et $\mathcal{X}_1$ of $\mathcal{X}_l$ ayant des probabilités ou proportions différentes de transmission de leur symboles correspondants. Soit $\mathcal{X}_0 = \mathcal{X}_s$, i.e. l'alphabet plus petit, et $\mathcal{X}_1 = \mathcal{X}_l \setminus \mathcal{X}_s$. Les deux sous-ensembles sont donc de l'ordre $d_0 = d_s$ and $d_1 = d_l - d_s$.

A l'instant k, le symbole s_k appartient à $\mathcal{X}$ avec probabilités $\frac{p_0}{d_0}$ pour tous les états dans $\mathcal{X}_0$ et $\frac{p_1}{d_1}$ pour les états dans $\mathcal{X}_1$. Plus précisement, $s_k \in \mathcal{X}_0$ avec probablité p_0 et dans $\mathcal{X}_1$ avec probabilité p_1. Tous les symboles dans $\mathcal{X}_0$ sont equiprobable, et également pour $\mathcal{X}_1$. For this to be meaningful, we assume

(A1) $p_0, p_1 \geq 0$

(A2) $p_0 + p_1 = 1$.

L'observation à l'instant k est donc donnée par :

$$y_k = hs_k + \varepsilon_k,$$

où h est le coefficient d'évanouissement du canal et $\varepsilon_k \sim \mathcal{N}_{\mathbb{C}}(0, \sigma^2)$ suit une distribution Gaussienne complexe de moyenne 0 et variance σ^2.

Le vecteur des paramètres inconnus θ consiste donc en la proportion p_1, le coefficient du canal h et l'écart type du bruit σ. Nous notons Θ l'espace des paramètres.

La densité de la distribution de l'observation y étant donnée l'état actuel a pour un certain choix des paramètres θ peut donc être écrit dans la forme

$$g\left(y|s;\theta\right) = \frac{1}{\pi\sigma^2}\exp\left(-\frac{1}{\sigma^2}\left|hs-y\right|^2\right).$$

La distribution inconditionnelle de l'observation y_k est donc un mélange de lois Gaussiennes de densité

$$f(y,\theta) = \frac{p_0}{d_0}\sum_{s\in\mathcal{X}_0} g\left(y|s;\theta\right) + \frac{p_1}{d_1}\sum_{s\in\mathcal{X}_1} g\left(y|s;\theta\right) \tag{1.9}$$

pour $y \in \mathbb{C}$.

Nous supposons qu'une suite d'observations $y_{1:K} = (y_1, \cdots, y_K)$ de longueur K est donnée et fixée.

Exemple: MAQ-4 et MAQ-16

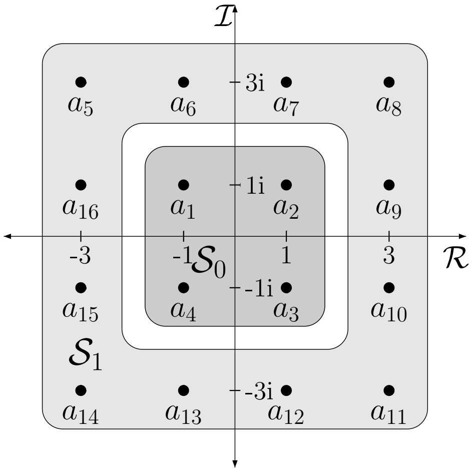

Figure 1.1: constellation de MAQ-16 séparée en $\mathcal{X}_0$ et $\mathcal{X}_1$

Pour les constellations MAQ-4 et MAQ-16, les alphabets de modulations sont donnés par:

$$\mathcal{X}_0 = \{1+i,\ 1-i,\ -1-i,\ -1+i\}$$

avec $d_0 = 4$ et

$$\mathcal{X}_1 = \{\pm a \pm bi :\ a,b \in \{1,3\}, a = 3 \vee b = 3\}.$$

avec $d_1 = 12$, voir aussi Figure 1.1.

1.3.2 Tests de rapport de vraisemblance

Soit $\mathcal{X}_s$ l'alphabet de la petite constellation (MAQ-d_s) de taille $d_s = |\mathcal{X}_s|$ et $\mathcal{X}_l$ l'alphabet de la grande constellation (MAQ-d_l) de la taille d_l.

Si le vrai modèle correspond à la MAQ-d_l, dalors $p_0 = p_1$. Par contre, s'il correspond à la constellation $\mathcal{X}_0$, nous aurons $p_1 = 0$. Nous allons donc définir les statistiques de test basées sur ces proportions. La vraisemblance comme fonction de $\theta \in \Theta$ étant donnée une séquence d'observations $y_{0:K}$ s'écrit donc

$$L_K(\theta) = \prod_{k=1}^{K} f(y_k, \theta) \tag{1.10}$$

Nous notons la log-vraisemblance

$$l_K(\theta) = \log L_K(\theta). \tag{1.11}$$

qui va nous servir afin de définir les tests de rapport de vraisemblance. Nous établissons maintenant la consistance de l'estimateur de maximum de vraisemblance (MV) pour les deux constellations potentielles.

Consistance de l'estimateur MV

Definition 1 *Nous notons l'estimateur de maximum de vraisemblance (MV) $\hat{\theta}_K$ dans le modèle étant donnée la séquence $y_{1:K}$ par*

$$\hat{\theta}_K = \arg\max_{\theta\in\Theta} l_K(\theta)$$

Theorem 1 *Supposons que $\theta_T = (0, h_T, \sigma_T)$ [la "vraie" modulation est une MAQ-4]. L'estimateur $\hat{\theta}_K$ est consistant, i.e. $\hat{\theta}_K$ converge en probabilité vers θ_T.*

Le même résultat est vrai si le vrai modèle correspond à la constellation MAQ-16.

Theorem 2 *Soit le vrai modèle égal à la constellation MAQ-16 de paramètre $\theta_T = (p_{1,T}, h_T, \sigma_T)$ où $p_{1,T} = \frac{d_l}{d}$. Alors l'estimateur $\hat{\theta}_K$ est consistent, notamment $\hat{\theta}_K$ converge en probabilité vers θ_T.*

Test d'hypothèses simples

La première statistique du test est un test d'hypothèses simples. L'hypothèse nulle est définie par $\mathrm{H}_0 : p_1 = 0$ correspondant à la constellation MAQ-d_s. Elle est testée contre l'hypothèse alternative $\mathrm{H}_1 : p_1 = \frac{d_l}{d}$ correspondant à la MAQ-d_l.

Les estimateurs MV correspondants aux deux hypothèses sont donnée par

$$\theta_0^* = \arg\max_{\theta\in\Theta : p_1 = 0} l_K(\theta) \text{ et } \theta_1^* = \arg\max_{\theta\in\Theta : p_1 = \frac{d_l}{d}} l_K(\theta).$$

La statistique du test de rapport de vraisemblance simple généralisé est donc défini par:

$$T_K^1 = l_K(\theta_0^*) - l_K(\theta_1^*). \tag{1.12}$$

Malheureusement, la distribution asymptotique de T_K^1 n'est pas connue et il n'est donc pas facile de calibrer ce test.

Test d'hypothèses composées avec $H_0 : p_0 = p_1$

La deuxième statistique est une test de rapport de vraisemblance composé généralisé d'hypothèse nulle égale à

$$\mathrm{H}_0 : p_0 = \frac{1}{d_0}.$$

L'hypothèse nulle correspond donc à la constellation MAQ-d_l. Elle est testée contre l'hypothèse que le vrai modèle ne coïncide pas avec la constellation MAQ-d_l.

Le test de rapport de vraisemblance généralisé compare la vraisemblance sous l'hypothèse nulle à la vraisemblance dans le modèle général, notamment pour $\theta \in \Theta$, voir Lehmann and Romano [2005] pour une introduction détaillée.

Les estimateurs MV correspondants sont donc donnés par

$$\theta_0^* = \underset{\theta \in \Theta : p_0 = \frac{1}{d_0}}{\arg\max} \; l_K(\theta)$$

et

$$\theta^* = \underset{\theta \in \Theta}{\arg\max}\, l_K(\theta).$$

La statistique s'écrit

$$T_K^2 = l_K(\theta^*) - l_K(\theta_0^*). \tag{1.13}$$

Au contraire de T_K^1, la distribution asymptotique de T_K^2 sous l'hypothèse nulle est connue Lehmann and Romano [2005].

Theorem 3 *Sous l'hypothèse nulle,*

$$2\,T_K^2 \xrightarrow{d} \chi_1^2,$$

où χ_1^2 représente une variable aléatoire de distribution chi-2 centrée à un degré de liberté.

Elle permet de ramener le problème du choix de l'ordre de la modulation à un problème simple de test d'une restriction paramétrique dans un modèle statistique régulier. Il est intéressant de remarquer que, sous les deux hypothèses, nulles et alternatives, les paramètres θ_0^* et θ^* convergent vers la vraie valeur du paramètre, car on apprend ici systématiquement à partir du modèle "le plus grand".

Test d'hypothèses composées avec $H_0 : p_1 = 0$

Pour la dernière statistique de test, on prend pour l'hypothèse nulle

$$\mathrm{H}_0 : p_1 = 0$$

ce qui revient à dire que le modèle est associé à la constellation MAQ-d_s : Les estimateurs MV correspondants sont donc donnés par

$$\theta_0^* = \underset{\theta \in \Theta : p_1 = 0}{\arg\max}\, l_K(\theta)$$

et

$$\theta^* = \underset{\theta \in \Theta}{\arg\max}\, l_K(\theta).$$

La statistique du rapport de vraisemblance généralisé est, dans ce contexte, donné par

$$T_K^3 = l_K(\theta^*) - l_K(\theta_0^*). \tag{1.14}$$

Au contraire de T_K^2, le théorème 12.4.2 de Lehmann and Romano [2005] ne s'applique pas à T_K^3, car le paramètre p_1 sous l'hypothèse nulle se trouve sur la frontière de l'espace de paramètres (la théorie classique des rapports de vraisemblance généralisés stipulent que les paramètres θ_0^* et θ^* convergent sous l'hypothèse nulle vers la "vraie" valeur du paramètre dans un point intérieur de l'espace des paramètres, la statistique de test étant liée à la distribution du score de Fisher au point limite, ce qui requiert de savoir différencier en ce point). Dans ce cas, la distribution asymptotique est un mélange d'une distribution du chi-2 centré et d'une masse de Dirac au point 0. Nous avons utilisé pour obtenir ce résultat des travaux étudiant la distribution du rapport de vraisemblance généralisé dans ce cas singulier où la valeur limite du paramètre appartient à la frontière de l'espace (voir par exemple les travaux de Andrews [1999, 2001], Shapiro [1985]).

Theorem 4 *Sous l'hypothèse nulle,*

$$2\,T_K^3 \xrightarrow{d} \bar{\chi}^2.$$

La variable aléatoire $\bar{\chi}^2(\theta_T)$ *s'écrit*

$$\bar{\chi}^2 = \frac{1}{2}\left(\chi_1^2 + \chi_0^2\right),$$

où χ_1^2 *signifie une distribution chi-2 centrée à 1 degré de liberté et* $\chi_0^2 \equiv$ *et constante égal à 0 presque sûrement.*

1.3.3 Simulations numériques pour la performance de classification

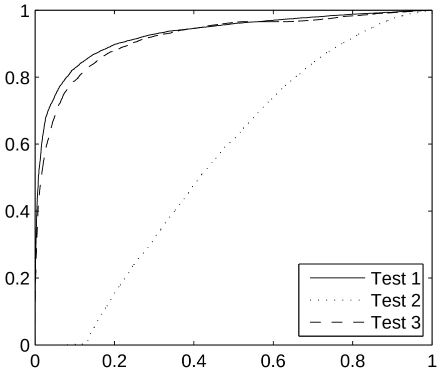

Figure 1.2: ROC curve of test for $K = 50$ and SNR $= 0dB$

Dans cette partie nous évaluons la capacité de classification par des courbes ROC qui ont l'avantage de permettre de comparer les performances sans avoir à utiliser les distributions asymptotiques des tests. Nous avons choisi de tester la constellation MAQ-4 contre la constellation MAQ-16. Pour les test composés T_K^2 et T_K^3, cela signifie que la performance est évaluée uniquement pour une hypothèse alternative simple.

Les alphabets sont définis comme dans la Figure 1.1.

La Figure 1.2 visualise la courbe ROC pour $K = 50$ et SNR $= 0dB$, ($K = 20$. Elle montre que particulièrement test 1 et test 3 nécessitent très peu d'observations. Le test 2 est clairement moins robuste. La raison est en partie que l'estimation dans le modèle avec $p_1 = 0$ (notamment la MAQ-4) est plus robuste. Alors que test 2 estime dans un modèle de 16 états pour les deux hypothèses, les deux autres tests utilisent le petit modèle pour une des deux hypothèses.

1.4 Expectation Sparse Maximization Algorithm

Pendant la dernière année nous nous sommes intéressés à l'étude d'identification du canal dans le cas où la réponse impulsionnelle du canal est longue (quelques dizaines de coefficients) mais où il est raisonnable de penser que seul un petit nombre de coefficients coefficients sont significativement différents de zéro.

Ce problème rejoint les problèmes d'estimation sous contrainte de parcimonie, qui a suscité un intérêt très important dans la communauté du traitement du signal au cours des dernières années. Le domaine de «Compressed Sensing», introduit dans les travaux pionniers de Donoho et de Candes (mais s'appuyant sur une tradition encore plus longue en statistique) a pour objectif de résoudre des problèmes linéaires sous-déterminés sous contrainte de parcimonie. Un vecteur x de dimension m est appelé parcimonieux si le nombre de composants *actifs* $\|x\|_0 = r \ll m$, i.e. les nombre de composants différents de 0 est "petit" par rapport à m. Un problème de «Compressed Sensing» peut être écrit comme

$$Y = Ax + \varepsilon$$

où Y es l'observation (connue) de dimension K, A est la matrice «Sensing» et ε est un vecteur de bruit Gaussien Candes and Tao [2005, 2006]. Plusieurs méthodes pour résoudre ce problème ont été proposées; les premières approches, sous-optimales, sont basées sur des approches gourmandes (greedy) dont l'exemple le plus notable est l'algorithme de «Matching Pursuit» (MP) Mallat and Zhang [1993] (amélioré dans «Orthogonal Matching Pursuit» (OMP) Pati et al. [1993]). Les approches plus récentes substituent à la minimisation sous contrainte ℓ_0 une une minimisation de l'erreur avec une pénalité $\ell 1$, dont il a été prouvé qu'elle permettait (sous des hypothèses précises et pas toujours satisfaites) de retrouver des solutions parcimonieuses.

Récemment, „Compressed Sensing“ a été appliqué à l'identification du canal en communications numériques. Dans de nombreuses situations, même le canal n'est pas parcimonieux, mais il existe un espace de représentation (plus précisément, on peut choisir une base) dans lequel la représentation du canal parcimonieuse. Une des premières applications à été Fuchs [1997] qui a étudié par cette méthode l'estimation des retards dans un canal spéculaire d'ordre inconnu. Bajwa et al. ont proposé plusieurs méthodes permettant de traiter des canaux plus généraux. Les travaux de Bajwa et al. [2008a,b,c] permettent de prendre en considération des canaux doublement-sélectifs (sélectifs en fréquence et en temps). Des canaux parcimonieux et rapidement variants en temps ont été considéré dans Lui and Borah [2003], Li and Preisig [2007]. Même si le canal n'est pas parcimonieux, les méthodes parcimonieuses peuvent améliorer les performances des estimateurs en choisissant un espace de représentation plus adapté Sharp and Scaglione [2008].

Dans les applications classiques du «Compressed Sensing» on suppose que la matrice A est connue. Nous allons généraliser cette contrainte à un modèle où la matrice peut être décomposée en un produit de deux matrices dont seulement une est connue. Plus précisément, le vecteur d'observation admet la représentation suivante

$$Y = S\Psi x + \varepsilon, \tag{1.15}$$

où seulement la matrice Ψ est connue. Dans le domaine de communications numériques, la matrice inconnue S correspond à la séquence transmise de symboles d'un alphabet fini. Dans les approches autodidactes que nous considérons, ces symboles ne sont pas connus de l'algorithme de réception. Afin de résoudre ce problème, nous avons introduit un algorithme EM sous contrainte de parcimonie („Expectation and Sparse Maximization (ESpaM) algorithm “).

Malheureusement, l'étape de maximisation standard de l'algorithme EM pour le modèle générale (1.15) n'est plus unique, i.e. l'ensemble de solutions est un sous-espace de dimension positive. Dans l'algorithme ESpaM, nous proposons donc d'utiliser une méthode parcimonieuse dans l'étape de maximisation comme le Matching Pursuit (MP), le Orthogonal Matching Pursuit (OMP) où le $\ell 1$ régularisation afin de choisir l'élément le plus parcimonieux dans ce sous-espace pour le nouvel estimateur de x dans chaque itération de l'algorithme itérative.

Même si l'étape Espérance de l'algorithme EM reste inchangée, son implémentation est assez délicate dans le cas général. Pour cette raison, nous allons présenter le modèle dans la forme la plus générale, mais ensuite nous allons nous concentrer sur un exemple pour lequel nous avons déjà étudié l'implementation de l'étape Espérance.

1.4.1 Description du modèle

Nous considérons le modèle linéaire parcimonieux

$$Y = S\Psi(\theta)x + \varepsilon, \tag{1.16}$$

où le vecteur x de taille $Q \times 1$ est parcimonieux, i.e. $r = \|x\|_0 \ll Q$. L' observation de taille K est donnée par $Y = (y_1, \dots, y_K)^T$ et le bruit par $\varepsilon = (\varepsilon_1, \cdots, \varepsilon_K)^T$. La matrice de symboles S consiste en K lignes et n colonnes. Chacun des $K \times n$ coefficients est élément d'un état d'espace $\mathcal{X}$, tel que l'état d'espace de S soit $\mathscr{S} = \mathcal{X}^{K \times n}$. La matrice $\Psi(\theta)$ est connue et dépend d'un paramètre $\theta \in \Theta$. Elle est de dimension $n \times Q$.

Nous allons présenter maintenant quelques exemples pour lesquels l'implémentation de l'étape Espérance est faisable. Si la matrice de symboles S correspond à une séquence de symboles en temps s_k en relation avec les observations y_k, il est raisonnable de considérer la structure de blocs suivante de S :

$$S = \begin{bmatrix} s_1 & 0 & \cdots & & 0 \\ 0 & s_2 & 0 & \cdots & 0 \\ & & \vdots & & \\ 0 & \cdots & & 0 & s_K \end{bmatrix}. \tag{1.17}$$

Nous supposons que s_k est une chaîne de Markov sur l'espace d'état $\mathcal{S} = \mathcal{X}^L$ de dimension $1 \times L$. Chacune des composantes $s_k = [s_k(1), \dots, s_k(L)]$ appartient à l'alphabet $\mathcal{X}$ de modulation. La matrice S possède, sous ces hypothèses, $n = KL$ colonnes. Dans ce cas, la matrice

$$\Psi(\theta) = \begin{bmatrix} \Psi_1(\theta) \\ \vdots \\ \Psi_K(\theta) \end{bmatrix} \tag{1.18}$$

peut aussi être décomposée sous forme de blocs, ou chaque bloc $\Psi_k(\theta)$ correspond à un symbole s_k. De plus, nous supposons que le vecteur de paramètres peut être écrit comme $\theta = (\theta_1, \cdots, \theta_Q)$ de telle sorte que la q-ème colonne $\psi_k(\theta_q)$ de $\Psi_k(\theta)$ est une fonction de θ_q. La matrice «Sensing» à l'instant k peut donc être décomposée de la façon suivante

$$\Psi_k(\theta) = [\psi_k(\theta_1), \dots, \psi_k(\theta_Q)] .$$

L'algoritme EM est bien applicable aux familles exponentielles , i.e. au bruit appartenant à une famille exponentielle. Nous supposons que chacune des composantes de ε_k est distribuée suivant une loi Gaussienne complexe $\mathcal{N}_{\mathbb{C}}(0, \sigma^2)$ de variance σ^2.

Nous supposons que la densité de la loi à posteriori de S étant donnés Y et x' existe et nous l'appelons $p(S|Y; x')$. L'observation y_k à l'instant k dépend uniquement de s_k, tel que (s_k, y_k) soit un modèle Markov caché (HMM). L'équation d'observation est donnée par

$$y_k = \sum_{q=1}^{Q} s_k \psi_k(\theta_q)\alpha_q + \varepsilon_k = s_k \Psi_k(\theta)x + \varepsilon_k. \tag{1.19}$$

Dorénavant, nous supposons que l'écart type du bruit σ est connu afin de faciliter la présentation. Soit $g_k\left(y_k|s;\theta\right)$ la fonction de vraisemblance de l'observation à l'instant k étant donné $s_k = s$, i.e.

$$g_k\left(y_k|s;\theta\right) = \frac{1}{\pi\sigma^2}\exp\left(-\frac{1}{\sigma^2}\left|s\Psi_k x - y_k\right|^2\right). \tag{1.20}$$

Nous supposons que les probabilités de transition de la chaîne de Markov s_k sont connues. Pour $s_k = s$ et $s_{k+1} = s'$, elle est définie par $q\left(s'\left|s\right.\right)$.

Nous supposons également que la loi de lissage marginale de s_k conditionnellement à l'observation Y est de densité $p_k(s_k|Y,x)$.

1.4.2 Estimation de maximum de vraisemblance (ML)

L'estimation des paramètres inconnus x est basée sur la maximisation de la fonction de vraisemblance $L\left(x\right)$ de la suite d'observations Y en fonction de x. Elle est définie par

$$L(x) = \int_{\mathscr{S}} p(S,Y;x)dS. \tag{1.21}$$

L'estimateur de maximum de vraisemblance (ML) de x est donc donné par

$$\hat{\theta} = \arg\max_x L(x). \tag{1.22}$$

Puisque il n'est pas possible de calculer $\hat{\theta}$ de façon analytique, nous utilisons une méthode itérative : L'algorithme Expectation-Maximization (EM) Dempster et al. [1977]. Au lieu de maximiser la vraisemblance directement, l'algorithme EM maximise la quantité intermédiaire dans chaque itération. Pour deux valeurs du paramètre x et x', elle est définie par

$$\begin{aligned} Q(\theta,\theta') &= \mathbb{E}_S\left[\log\left(f(S,Y;\theta)\right)|Y;\theta'\right] \\ &= \int_{\mathscr{S}} \log\left(f(S,Y;\theta)\right)p(S|Y;\theta')d\mathbb{P}(S). \end{aligned} \tag{1.23}$$

Si S se décompose de façon séquentielle comme dans (1.19), l'équation (1.23) se simplifie :

$$Q(\theta,\theta') = -\frac{1}{\sigma^2}\sum_{k=1}^{K}\int_{\mathcal{S}}\left\|s_k\Psi_k x - y_k\right\|^2 p_k(s_k|Y;\theta')d\mathbb{P}(s_k) - K\log(\sigma^2) + \text{const}, \tag{1.24}$$

où le terme constant ne dépend pas du canal inconnu x.

Après avoir défini un estimateur initial $\hat{\theta}_0$, l'algorithme EM se décompose en deux étapes qui sont itérées jusqu'à la convergence globale de l'algorithmique
1) Expectation: Calculer $Q(x,\hat{\theta}_i)$.
2) Maximization: Calculer $\hat{\theta}_{i+1} = \arg\max_x Q(x,\hat{\theta}_i)$.

L'étape Expectation consiste donc à calculer les lois marginales de lissage $p(S|Y;x)$ étant donné un estimateur du canal x. Cette étape peut être implémentée en utilisant l'algorithme Baum-Welch Baum et al. [1970] ou un algorithme particulaire de lissage tel que celui que nous avons présenté dans les rapports précédents.

Maximisation parcimonieuse des paramètres

Contrairement à la fonction de vraisemblance, $Q\left(x,x'\right)$ peut être maximisée plus facilement. Le meilleur estimateur du canal est donné par la solution du système suivant d'équations :

$$\mathrm{E}_{\mathrm{sy}}\left(x'\right) = \mathrm{E}_{\mathrm{ss}}\left(x'\right)x, \tag{1.25}$$

où

$$\begin{aligned}\mathrm{E}_{\mathrm{sy}}\left(x'\right) &= \mathbb{E}_S\left[(S\Psi)^H Y \Big| Y; x'\right] \\ \mathrm{E}_{\mathrm{ss}}\left(x'\right) &= \mathbb{E}_S\left[(S\Psi)^H S\Psi \Big| Y; x'\right].\end{aligned}$$

où nous avons noté par ^{H} le conjugué hermitien d'une matrice complexe.

Si $\mathrm{rk}\left(\mathrm{E}_{\mathrm{ss}}(x')\right) = Q$, la solution de (1.25) est unique et elle est donnée par $\left(\mathrm{E}_{\mathrm{ss}}(x')\right)^{-1}\mathrm{E}_{\mathrm{sy}}(x')$. Malheureusement dans notre modèle, $\mathrm{rk}\left(\mathrm{E}_{\mathrm{ss}}(x')\right) = \min(Q, L)$ et la solution est donc unique seulement si $L = Q$. En général, $L \ll Q$ et la solution de (1.25) est un sous-espace de dimension $Q-L$. Nous proposons donc d'utiliser un algorithme parcimonieux afin de sélectionner le vecteur le plus parcimonieux dans ce sous-espace. Il est donc nécessaire de résoudre le problème suivant :

$$\min_x \|x\|_0 \quad \text{s.t.} \quad \mathrm{E}_{\mathrm{sy}}\left(x'\right) = \mathrm{E}_{\mathrm{ss}}\left(x'\right) x, \tag{1.26}$$

le problème de Lasso associé Tibshirani [1996], Donoho [2006], Candes and Tao [2006]:

$$\min_x \|x\|_1 + \lambda \left\|\mathrm{E}_{\mathrm{sy}}\left(x'\right) - \mathrm{E}_{\mathrm{ss}}\left(x'\right) x\right\|^2, \tag{1.27}$$

pour $\lambda > 0$ ou le problème de «Basis Pursuit» Chen et al. [1998]:

$$\min_x \|x\|_1 \quad \text{s.t.} \quad \mathrm{E}_{\mathrm{sy}}\left(x'\right) = \mathrm{E}_{\mathrm{ss}}\left(x'\right) x. \tag{1.28}$$

Même si $L = Q$, les algorithmes parcimonieux peuvent être utilisés afin d'améliorer la performance et la robustesse de l'algorithme EM.

Nous proposons donc de résoudre le problème de maximisation de la quantité intermédiaire par un algorithme parcimonieux comme le Matching Pursuit (MP) Mallat and Zhang [1993], l'Orthogonal Matching Pursuit (OMP) Pati et al. [1993], Tropp [2004] ou une $\ell 1$-régularisation à l'équation (1.25). L'algorithme «Expectation and Sparse Maximization» (ESpaM) pour l'estimateur initial du canal $\hat{\theta}_0$ est donc donné en itérant les étapes suivantes :

1. Calculer $p\left(S|Y;\hat{\theta}_i\right)$ ou $p_k\left(s_k|Y;\hat{\theta}_i\right)$ pour $k = 1,\dots,K$ dans le modèle séquentiel.
2. Calculer $\mathrm{E}_{\mathrm{sy}}(\hat{\theta}_i)$ et $\mathrm{E}_{\mathrm{ss}}(\hat{\theta}_i)$.
3. $\hat{\theta}_{i+1} = \mathrm{Sparse}\left(\mathrm{E}_{\mathrm{sy}}(\hat{\theta}_i), \mathrm{E}_{\mathrm{ss}}(\hat{\theta}_i)\right)$.

La fonction $\mathrm{Sparse}(\cdot,\cdot)$ décrit l'algorithme parcimonieux qui prend comme entrées la matrice $\mathrm{E}_{\mathrm{ss}}(\hat{\theta}_i)$ et le vecteur $\mathrm{E}_{\mathrm{sy}}(\hat{\theta}_i)$ afin de résoudre le Problème (1.25).

Propriétés de convergence de l'algorithme ESpaM

Nous discutons maintenant de façon plus théorique les propriétés de convergence de l'algorithme ESpaM.

Lemma 1 *Soit $\left(\hat{\theta}_i\right)_i$ une séquence d'estimateurs de x obtenues par l'algorithme ESpaM. Alors pour chaque i*

$$L\left(\hat{\theta}_{i+1}\right) \geq L\left(\hat{\theta}_i\right).$$

Cette lemme suit directement du fait que l'algorithme ESpaM ne change pas le principe de l'algorithme EM. En effet, la quantité intermédiaire est encore maximisée. L'algorithme ESpam nous donne seulement un moyen pour choisir une solution dans le sous-espace de valeurs maximales.

Nous avons donc établie que l'algorithme ESpaM converge. En général comme pour l'algorithme EM, l'algorithme ne converge pas forcement vers le maximum global de la fonction de vraisemblance. Si la matrice Ψ est quadratique et de rang plein, la vraisemblance a des maximums locaux isolés, i.e. la maximisation de la quantité intermédiaire est unique. Une conséquence immédiate est donc que la vraie valeur est un point fixe de l'algorithme. Par contre, cela n'est pas évident si Ψ n'est pas de plein rang, car les valeurs maximales forment un sous-espace de l'espace de paramètres.

Nous donnons maintenant une condition suffisante pour laquelle le vrai paramètre x reste un point fixe de l'algorithme ESpaM.

Assumption 1 *Supposons que S ne soit pas dégénérée tel que $\mathbb{E}_x[S^H S]$ soit de plein rang. Puisque cette matrice est la matrice de covariance de S cela implique uniquement qu'il n'y ait pas de relation affine entre les colonnes de S.*

Assumption 2 *Supposons que chaque combination de $2p$ colonnes de Ψ est linéairement indépendante. Soit p le nombre de coefficients différents de zéro dans le vrai vecteur de paramètres x. Cela implique évidemment que $L \geq 2p$.*

Lemma 2 *Supposons que les hypothèses 1 et 2 soient vérifiées. Alors, le vrai vecteur de paramètres x est un point fixe de l'algorithme ESpaM, i.e. il n'existe aucune solution plus parcimonieuse.*

Chapter 2

Introduction

Before devoting ourselves to the dry topic of Blind Classification in digital communications let us step back and relax for a minute. Imagine being in the French Alps, surrounded by vast rock faces. You are walking through a silent valley with a person you know well, let us call her Alice. Alice is talking to you. It is easy to understand what she is saying, is it not? Now imagine you are half way up a steep canyon, at the bottom a roaring mountain stream. It is getting a little more difficult to grasp what Alice is saying, is it not? What if Alice is not walking right next to you but quite far away, maybe on the other side of the canyon? Additionally to the noise of the stream, the echoes of the faces of the canyon are now making it quite intricate to grasp what Alice wants to tell you. Maybe she is not even in the line of sight. We will play this even further. Imagine you do not know Alice and consequently you do not know the language Alice is speaking. Thus, before trying to understand what she is saying, for a start you need to make a guess about the language she is using. You might want to say, this has no longer any bearing with the idyllic scene we started off with and that we should stop dreaming about Alice and start working seriously.

Blind Classification in digital wireless communications fits pretty exactly in that image of a mountain scene. Only that in digital communications Alice is transmitting a sequence of digital symbols instead of natural speech. Instead of a roaring mountain stream, we have the noise in the atmosphere adding to the signal. The echoes of the rock faces translate equally well to reflections of the digital signal from any surface in the environment. Eliminating or estimating this echo is what is called channel Identification in digital communications. Finally, instead of not knowing Alice's language, we do not now the digital modulation alphabet that Alice is using. The estimation of the language is known as Blind Classification. After this short trip into the French Alps and with this image in mind we may now come back to the topic of this thesis and for the time reading put on our blinkers and ignore anything else except for engineering, communications, mathematics and statistics.

The red thread of this work is Blind Classification of a linear modulation scheme in wireless digital communications. Classification refers to the estimation of the modulation scheme that has been used by the transmitter. In general, the receiver has some knowledge about the used modulation, for example he has a list of potential modulations and will choose one from that list. Blindness refers on the other hand to estimation at the receiver side without cooperation between the transmitter and the receiver. The estimation is based on a block of observations at the receiver side. This received sequence is as in the example above the transmitted sequence (what Alice said) perturbed by the echoes and the additive atmospheric noise, or in other words the transmitted sequence is sent over a linear frequency-selective time-invariant multi-path channel. We will also consider frequency- and time-selective (doubly selective) channels, but to a lesser extent. The linear modulations we consider are Quadrature Amplitude Modulation (QAM) in various different sizes, but we note that all presented methods are easily applicable to other linear modulation schemes like Phase Shift Keying (PSK) and others.

The second thread that can be found throughout this work is likelihood based estimation of each unknown parameter. Likelihood is a probabilistic measure of how well a certain choice of the parameters fits the observations. The better they fit the less noise we need to explain the observations. Thus, the maximum likelihood estimate of a parameter is the value that needs the least noise to fit the observations compared to all other possible values of the parameter. Let us go back to Alice. We consider the parameter of how many reflecting walls there are. As we saw before, in the canyon the speech is severely perturbed by echoes. If we now want to test whether the number of reflecting walls is zero, then we cannot explain all the echoes that arrive. Hence, we would consider the echoes as noise and thus, the noise level is much stronger compared to if we say the number of reflecting walls is two. Consequently, in a likelihood based approach the parameter choice 'number of reflecting walls equals to two' is clearly preferable.

As we mentioned, we base the Blind Classification on the likelihood of the observations. We will explain later in detail why. Besides the unknown modulation, we have no channel state information (CSI), i.e. knowledge on the transmission channel. Therefore, likelihood based Blind Classification is no easy task. Two possible concepts exist. The first one applies if a priori the distribution of the unknown parameters is known. Then they may be integrated out. This is however not easily feasible for multi-path channels because of the large dimension of the unknown parameter space. We concentrate on the second approach, which is to derive the maximum likelihood estimate of the unknown channel parameters for each potential modulation before estimating the modulation scheme.

A likelihood based Blind Classification approach coupled with maximum likelihood estimation of the unknown channel parameters naturally splits up into several subsequent levels. At the top level, a model estimator has to be found to estimate the modulation scheme. This will thus be the main Classification algorithm. The next subsequent level is the estimation of the unknown channel parameters for each of the potential modulations. This problem is known as Blind Identification or blind deconvolution in digital communications. The numerous existing methods for Blind Identification do not only include maximum likelihood approaches but also various other concepts. Since we need however the maximum likelihood for the top level step, we concentrate solely on this concept. However, the maximum likelihood estimate is not available in an analytical form. We will use the well known concept of the iterative Expectation-Maximization (EM) algorithm and develop a sparse variant of this algorithm. The subsequent level is the reconstruction of the unknown signals given the modulation and an estimate of the unknown channel parameters. This can be either in the form of finding the most likely sequence, or by finding the posterior distribution of the signals. The posterior distribution is also called smoothing distribution and therefore this problem is known as smoothing in the signal processing community. Smoothing splits again up into several subsequent steps but we will discuss them later and concentrate on the big picture for the moment.

On each of these levels we consider different methods and algorithms, some of which are state of the art, some of which we develop in this work. It turns out that each of these different methods encounters the same problem on the subsequent level. Therefore, each of these three described big levels is crucial no matter which of the proposed methods is chosen. No matter which of the proposed model estimator we choose on the top level, we are still in the need of the subsequent maximum likelihood Blind Identification. As a consequence, the levels are mostly independent of each other and we will therefore present them in a very modular structure. Each of the levels is described in one chapter and each of these chapters is self-containing. The reader may thus choose to read only a selection of the chapters. This is even more important since the lower two levels constitute interesting and relevant problems even without considering the top level. For example, on the second level, the estimation of the unknown channel parameters is known as Blind Identification in digital communications and has seen numerous contributions by the communications community. The associated chapter may thus as well be seen as a contribution to the Blind Identification problem. However, the big picture throughout the thesis should be kept in mind, since the lower levels are essential for implementing the upper levels.

2.1 An Overview of the ML Classification Concept

To keep the work as modular as possible, we will give a more detailed introduction to each of these levels in the corresponding chapter, each followed by a summary of the state of the art in that area of research. We will now only give a short overview of the approaches on each level. Fig. 2.1 gives a schematic illustration of the big picture as well as of the different levels. This illustration is meant to give an overview of the methods and concepts that are used. For more details, we refer to the corresponding chapters. At this point, we will furthermore not go into detail what our contributions to each box are. We refer to Section 2.2. The rounded boxes, dyed boxes are representative of each level and describe the problem or question on this level. The angled, white boxes give possible solutions to the question in the box above. Then again, each solution raises new questions that are found in the boxes below. Thus, the top level question is that of the Maximum Likelihood (ML) Blind Classification. What is the best model estimator to use for the decision on the modulation scheme? We present two possible solutions. Both are Likelihood Ratio Tests (LRTs). The model LRT refers to a test where each potential modulation is considered as a potential model. Consequently, the likelihoods of the received signal in the potential models are calculated and their ratio is compared to a certain threshold. This can be considered as the state of the art. On the other hand, the Parametric LRT treats the unknown modulation as an additional parameter in an enclosing, larger model.

However, each of these possible solutions to the ML Blind Classification question raises the new question of ML Blind Identification, i.e. of the estimation of the unknown channel parameters, the subsequent level. As we mentioned before, a well known algorithm for this problem is the Expectation-Maximization (EM) algorithm by Dempster et al. [1977] that iterates a Maximization (M) and an Expectation (E) step. While the M step is simply given by a small well-determined linear system of equations, such that its solution is trivial, the E step is more demanding. It consists of calculating the posterior or smoothing probabilities of the unknown transmitted signals. Hence, the E step or Smoothing defines the next large level that we have introduced before.

We propose a second solution to the ML Blind Classification problem that combines elements from standard ML estimation and the concept of Compressive Sensing which is one of the currently most intensively discussed topics in the signal processing society. If the parameter vector is sparse, which means that most coefficients are equal to zero and only very few are active, then Compressive Sensing uses this knowledge to reduce the number of necessary received signals to recover the parameter vector. The Expectation Sparse Maximization (ESpaM) algorithm combines this concept with the EM algorithm such that it may be applied to Blind Identification. The maximization step is now implemented with a sparse algorithm like Matching Pursuit (MP) by Mallat and Zhang [1993] or Orthogonal Matching Pursuit (OMP) by Pati et al. [1993].

The standard solution to the Smoothing problem is the Baum-Welch algorithm by Baum et al. [1970] or to a lesser extent the very similar Viterbi algorithm by Viterbi [1967]. These algorithms are however only applicable to relatively small models, otherwise their computational complexity becomes too large.

Therefore, we consider Particle Smoothing as an alternative solution. Particle Smoothing is a Sequential Monte-Carlo concept to reduce the complexity of the exact counterpart, the Baum-Welch algorithm. We are aware that this is a very limited view of Particle Smoothing restricted to finite models, but it suffices as a one-line introduction at this point. We will explain later why this interpretation is to the point in digital communications. The solution of using Particle Smoothing to solve the Smoothing problem brings up the question of how the smoothing probabilities should be decomposed such that it is possible to approximate them by a Sequential Monte-Carlo algorithm. Several concepts are possible, namely Fixed-Interval Smoothing, Fixed-Lag Smoothing or the Joint-Two-Filter Smoothing among many other algorithms.

Each Particle Smoothing algorithm requires a Monte-Carlo approximation of the filtering distribution, i.e. the distribution of the signals knowing only the observations up to the present but not the future ones. This is the Filtering level. Luckily in digital communications, it is known that a deterministic Particle Filtering algorithm is the superior method that yields that approximation. Discrete Particle Filtering estimates the filtering probabilities sequentially from one time step to the next by exploring all possible symbol choices at the next time step given the approximation of the previous time step. To reduce the complexity, after each of these explorations, only a small number of possible symbols is selected, while the rest is neglected. This raises immediately the question of how these symbols should be selected. This is most bottom level of the big picture. The standard approaches to this problem are deterministic schemes. Random schemes that minimize statistical distances like the Chi-Squared divergence or the L2-norm are however superior. They are therefore the two methods we would recommend.

We have given a rough overview to introduce the various concepts we used to solve the problems on each of the discussed levels. This is meant as frame to explain how each step fits in the red thread of Blind Classification. For a more thorough introduction to each of the topics we refer again to the corresponding chapters. Once again, we stress the modularity of the following chapters. Each chapter is mostly self-comprehensive. We will now proceed with a survey of the main contributions to this work in Section 2.2. This is also meant as a rough overview. A more detailed description may be found in each chapter. In Section 2.3, we will then discuss the current knowledge about the theory and convergence results of these concepts, or rather on the lack of theoretical results. In Chapter 3, we will introduce the general model in several forms, such that in each of the following chapters we may refer to the most general form for which the methods apply. We will equally derive why this general model applies to a digital communications system. We will consider three examples. The first example is a linear time-invariant frequency-selective multi-path channel, for which a large part of the Monte-Carlo experiments in this work were carried out. The second example is a frequency- and time-selective (doubly-selective) multi-path channel model. For each path, we consider a different Doppler frequency, a delay and an attenuation. We introduce a fine grid on the parameter space to render the problem linear such that it fits in the general model. The third model stems from the Orthogonal Frequency Division Multiplexing (OFDM). The considered channel is as well doubly-selective, such that with some additional assumptions this model becomes the analog of the second example with Doppler frequencies and delays exchanged.

In Chapter 4, we discuss the problems and solutions from the bottom level up to the Smoothing level. After an introduction and a description of the state of the art, we explain in detail the Baum-Welch algorithm and the Viterbi algorithm. We then introduce the concept of Particle Filtering in finite state space models and show the similarity to approximate Viterbi algorithms like the M-algorithm by Anderson and Mohan [1984] and the T-algorithm by Simmons [1990]. Several schemes for the selection are presented, starting by the deterministic schemes and followed by the superior random selection schemes. We then turn to Particle Smoothing and show how the Particle Filtering is integrated into that concept. We present the Fixed-Interval Smoothing by Doucet et al. [2000], the Fixed-Lag Smoothing by Olsson et al. [2008], the Two-Filter Smoothing by Kitagawa [1994] and the Joint Two-Filter Smoothing.

In Chapter 5, the concept of Maximum Likelihood (ML) estimation is presented. We explain the EM algorithm by Dempster et al. [1977] as well as the novel Expectation Sparse Maximization (ESpaM) algorithm.

Chapter 6 introduces the model estimation in general and in particular for the presented application. The standard Generalized Likelihood Ratio Test (GLRT) for Blind Classification is presented. Two new model estimators are given that are based on an additional proportions parameter. Furthermore, we point out several ideas to speed up the model estimation and render it more robust. These ideas are very specific to QAM models and exploit their regular structure. Each chapter is concluded by a Monte-Carlo simulation study.

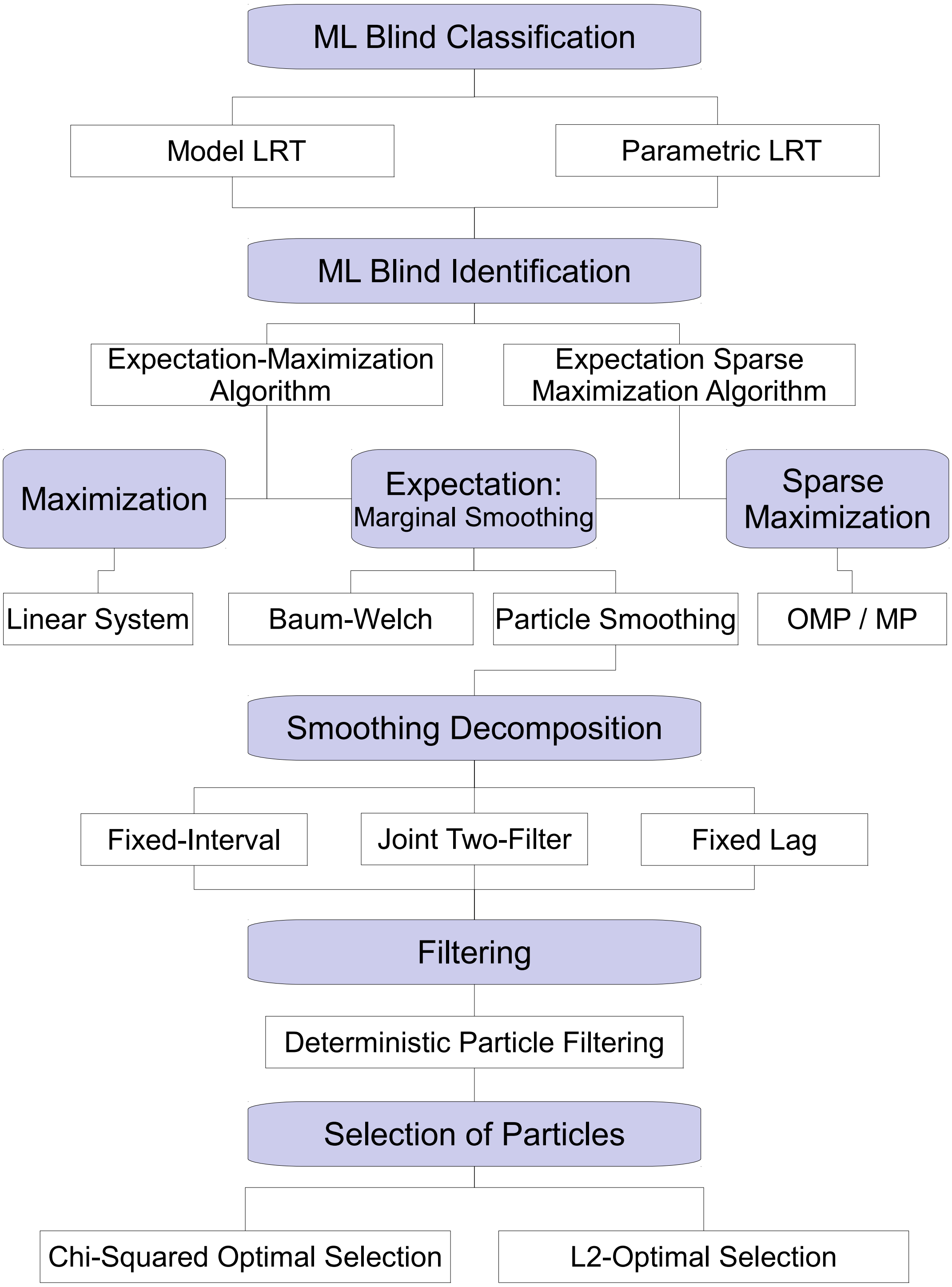

Figure 2.1: GLRT Blind Classification on One Page

2.2 A Survey of the Main Contributions

On each of the levels of the ML Blind Classification procedure, we have contributed several results that we will now summarize in more detail. Giving a thorough summary of the new concepts requires a certain depth of details that causes this section to be less self-comprehensive. For more explanations we refer to the corresponding chapters.

In Chapter 6 we derive a new interpretation of likelihood ratio tests of QAM modulations that exploits the specific structure of the QAM alphabets. The model in which we derive the test statistics is a Gaussian mixture and corresponds to a flat-fading channel. Assume that the two potential modulation alphabets correspond to a 4-QAM and a 16-QAM and that the true modulation is the 4-QAM. We place the estimation in an embracing model with 16 states like the 16-QAM alphabet but with flexible weights on each of these states. In a 4-QAM the 12 outer points have zero weights, while in a 16-QAM all states have equal weights. In this model, we propose two parametric generalized likelihood ratio tests. The null hypothesis of the first test is that all weights are equal. The asymptotic distribution under the null hypothesis is a Chi-Squared distribution with one degree of freedom. The second test statistic tests the null hypothesis that the outer states have zero weights. Since the parameters are on the boundary, the asymptotic distribution is more complex and we prove that the theorem by Andrews [2001] is applicable. Furthermore, we prove that the model estimation is consistent, i.e. that the likelihood in the wrong model is asymptotically smaller. In simulations we show that the second new test statistic is equivalent to the standard model GLRT in terms of the ROC curve. We have furthermore conducted a large study to test the performance of the model estimation in a variety of frequency-selective channels, i.e. in hidden Markov models.

In Chapter 5 we derive a novel variant of the EM algorithm that we call the Expectation Sparse Maximization (ESpaM) Algorithm that exploits the sparsity of the parameter vector. We have published this algorithm in Barembruch et al. [2010a,b,c]. In other words, we propose a method to extend Compressive Sensing to blind estimation, namely the case where the sensing matrix is (partially) unknown. On the one hand, we discuss why it is computationally not feasible to include the sparsity constraint directly inside the intermediate quantity. On the other hand, we derive an EM type algorithm that is computationally almost as fast as the original EM algorithm and is much more robust if the parameter vector is sparse. If the rank of the unknown sensing matrix is smaller than the length of the parameter vector, then the update step of the EM algorithm is not longer unique. The solution rather constitute a subspace of the parameter space. The ESpaM algorithm proposes to choose the sparsest vector amongst these solutions by using a sparse method like the Matching Pursuit, the Orthogonal Matching Pursuit or others. We show that under some constraints on the sparsity and regularity of the model the true parameter value is a fixpoint of the ESpaM algorithm and that any iteration of it will increase the likelihood. We have applied this method to blind estimation in frequency-selective channels if the impulse response is sparse. Furthermore, it has been applied to doubly-selective channel estimation for linear modulations and OFDM by using basis expansion methods. In OFDM we discuss as well a semi-blind variant that makes use of the pilots in the model. In a comprehensive simulation study we show the robustness and performance increase over the standard EM algorithm in a sparse model. In doubly-selective methods, we show that the performance is superior to the algorithm by Ben Salem and Salut [2004].

On the smoothing level, we have compared and applied several Particle Smoothing algorithms as well as the EMVA by Nguyen and Levy [2005] in this digital communications scenario. The considered particle algorithms were the Fixed-Interval Smoothing by Doucet et al. [2000], the Fixed-Lag Smoothing by Olsson et al. [2008], the Two-Filter Smoothing by Kitagawa [1994] and a novel Two-Filter Smoothing by Fearnhead et al. [2008]. We showed that the complexity of most of these smoothing algorithms is linear in the number of particles in this specific model. We developed a new smoothing algorithm, the Joint Two-Filter Smoothing, that diminishes the weak points of the other algorithms. Like the Two-Filter Smoothing, it is based on two particle filters, the first one estimating the forward filtering distribution and the second one estimating the backward filtering distribution. Before deriving the smoothing distributions, the Joint-Two-Filter Smoothing improves these two filtering approximations by boosting the support of the respectively other particle filter. After this step, the two particle filters are joined as in the Two-Filter Smoothing to approximate the smoothing distribution. Hence, the novel algorithm bases the support of the smoothing approximation not solely on either the forward or the backward particle filter like the Two-Filter Smoothing, but on both of them. The second advantage is that, due to the sparsity of the transition matrix, the smoothing approximations of the Two-Filter Smoothing are often degenerated in the sense that all probability weights are equal to zero because all possible transitions between forward and backward particles are impossible. This problem is not immanent for the Joint Two-Filter Smoothing because the supports have been joined before. The results have been published in Barembruch et al. [2009] and Barembruch et al. [2008a].

We have extended the concept of random optimal selection schemes by Fearnhead and Clifford [2003] to more general statistical distances. The constructive proof shows how to derive random selection schemes with respect to some statistical distance. By optimality we mean that the scheme minimizes the expected distance between proposed and selected particles. The Kullback-Leibler Optimal Selection scheme is equivalent to the L2-Norm Optimal Selection by Fearnhead and Clifford [2003]. In contrast to these schemes, the Chi-Squared Optimal Selection bases the selection on the square roots of the particle weights, diminishing the differences between these weights. Hence, this formalizes the idea why particles may be resampled with probabilities that are not proportional to their weights. The results have been published in Barembruch et al. [2009] and Barembruch et al. [2008b]. We have applied the Chi-Squared Optimal Selection to the approximate EMVA algorithm and showed that it is clearly superior to existing methods in Barembruch [2010].

We have run extensive simulation studies to compare the smoothing algorithms as well as the selection schemes. The results of these studies enable us to give clear advises for the use in practical applications.

2.3 A Survey of the Lack of Theoretical Results for Approximate ML Classification

The theoretical analysis of most of the presented methods at each level is a very difficult question, not even to speak of the whole as one big concept. That is to say, even by itself each of the presented methods - be it particle selection, Particle Smoothing, approximate EM algorithm, and the likelihood ratio test - are very hard to analyze and then it is not even clear how the approximation of each level influences the theoretical analysis of the next, overlying level.

It is surprising how little results are known today, although the presented model is fairly easy. One may think that this survey of these open questions sheds a negative light on this work because it reveals plenty of open questions. Indeed, the focus of this work is rather algorithmical and practical. Nevertheless, we were able to prove several results. Furthermore, in most works on Blind Classification the question of theoretical properties is not even asked at all. The situation is not as hopeless as it may seem now. For most problems, we have a more or less precise idea about how one would have to proceed to prove the desired theoretical results. That implies that we are confident that the methods work well in theory, although a proof has not been found yet and might not be found for a long time. Hence, we prefer to see this survey of the lack of theoretical knowledge not as a weak point of this work, but rather as a perspective for future work and as an intuition why the methods work theoretically.

Let us have a look in more detail from the bottom level to the top level. We derive and discuss several random schemes for particle selection that minimize a statistical distance between the proposed and the selected particles. The statistical distances that are covered include the $\ell 2$-norm, the Kullback-Leibler and the Chi-Squared distance. However, it is not known if a particle filter using these schemes will converge to the true distribution over several time steps for the number of particles converging to infinity. The analysis is quite involved because the particles are not at all independent after the selection. On the other hand, if multinomial resampling is used then the particles are independent of each other and the convergence is known; see for example the work by Douc and Moulines [2008]. However, in a discrete model multinomial resampling is inefficient. Hence, if even with this inefficient method the particle filter converges to the true distribution, then intuitively a particle selection that clearly reduces the distance to the unselected particles and thus to the true distribution has to converge as well. However, this remains to be proven.

On the next level, the situation is not much promising for the theoretical analysis of the Particle Smoothing algorithms. Olsson et al. [2008] proved a central limit theorem for the Fixed-Lag smoothing. In the presented model, this is however not the most efficient smoothing algorithm in the Monte-Carlo experiments. To our knowledge, the Fixed-Interval Smoothing has not been successfully analyzed yet. The difficulty lies in analyzing the integral of a random function with respect to a random measure. A promising approach for the analysis of the Two-Filter Smoothing is the ongoing work by Randal Douc et al.. The Joint Two-Filter Smoothing we propose in this work has not yet been analyzed and the work by Douc et al. will not be directly applicable. We show however why the Joint Two-Filter Smoothing improves the forward and backward filtering approximations in contrast to the Two-Filter Smoothing. In a nutshell, the algorithms that work best in the simulations in this setting - the Fixed-Interval Smoothing and the Joint Two-Filter Smoothing - have not been analyzed yet.

The convergence of the EM algorithm has been well studied. It is known that it converges to a local maximum and if the initial value is close enough to the global maximum, it will converge to the global maximum. What happens if an approximate algorithm like Particle Smoothing is used to estimate the smoothing distributions? The only answer to this is also due to Olsson et al. [2008] who proved that the EM algorithm retains its nice properties if the Fixed-Lag Smoothing is used. Their results may however not easily be extended to the other smoothing algorithms.

We show several results for the Expectation Sparse Maximization (ESpaM) algorithm we propose. Since in each iteration the algorithm still maximizes the intermediate quantity, a sequence of estimators of the ESpaM algorithm will also yield a sequence of increasing likelihoods equivalently to the EM algorithm. We showed furthermore that under some additional assumptions mainly on the sparsity the true parameter value is a fixpoint of the ESpaM algorithm. However, it is not shown if a parameter value is close to the true value that it will also converge to the true value. Furthermore, the analysis assumes that the exact sparsest solution is found. This is however not true if computationally feasible algorithms like Matching Pursuit or Orthogonal Matching Pursuit are used. For these algorithms, not much is known on their influence on the convergence. Additionally for the expectation step, the same holds true as for the EM algorithm. If the smoothing distributions are approximated, it has not yet been proved that the ESpaM algorithm still converges.

We arrive at the top level, namely the generalized likelihood ratio tests (GLRT). All we are able to say about these tests holds true only if the exact ML estimate in each model may be derived. Since the EM algorithm converges after an infinite number of steps and we do not want to wait that long, the output of the EM algorithm is imperatively only an approximation of the ML estimate. This is even more true if we use an approximate EM algorithm coupled with Particle Smoothing.

The standard model GLRT as for example described by Hong [2006] is consistent, i.e. when the number of observations tends to infinity the test identifies the correct model. This is due to the fact that the different models in this specific context are distinct. For example, if the true model is a 4-QAM, then the likelihood in a 16-QAM model is asymptotically smaller than in the true 4-QAM model. However, formally this has not been shown yet. We have shown it in a simpler model for a mixture of Gaussians. On the other hand, very little is known on the error exponents for simple GLRTs in hidden Markov models (HMM). Gassiat and Boucheron [2003] derived error exponents for finite emission alphabets. Their work already shows that a proof in more general models will be complicated and involve certainly large deviation theory for which not that much is known in HMMs.

We have placed the Blind Classification problem in a parameter testing problem. That means, instead of testing for different models, we derive GLRT statistics for testing a single parameter. That approach allows for the derivation of the asymptotic distribution of the test statistic under the null hypothesis. We show that in the simpler case of mixture of Gaussians. The difficulty is that under the null hypothesis the parameter is on the boundary of the parameter space, such that the derivation gets more involved. The same has not been done yet for HMMs.

To summarize the previous discussion, we are not able to say much about the theoretical properties of the estimators. However, we are very confident that the estimators converge in fact since for several proofs a sketch of the approach exists already and the simulations show in fact that the methods work really well and that the approximate methods like the Particle Smoothing are very close to their exact equivalents.

Chapter 3

Model

In accordance with the structure of this thesis, we will also describe the model including three examples in a modular form. We will start on the top level by the most general form of the model that will be used in Chapter 5, the blind identification part of the thesis. Since for this general form it is easy to formulate an EM algorithm, but impossible to implement its expectation step, we will then subsequently go into more detail and restrict the model to more specific cases for which we are then able to establish an implementation. Especially in Chapter 6, the classification chapter, we will often restrict the model even more, i.e. to a QAM modulation, for which we may then establish new and more efficient algorithms than for the standard, general model.

The main goal of this work is blind classification for large linear modulation schemes in frequency-selective channels. Each of the levels of the model presentation will coincide which the corresponding model. However, we like to maintain the flexibility of the model presentation, since most of the algorithms we present work for much more general models than in the blind classification case. We will therefore present each algorithm as generally as possible - without having to blow up the notation. At each step, we will note to what extent the model has to be known to understand the algorithm.

After having presented the model in its different forms, we will give three concrete examples for applications in digital communications. The first model is a linearly modulated symbol sequence transmitted over a frequency-selective channel model with a finite impulse response, where the observations are sampled at the same rate as the symbols are transmitted. The model is then easily shown to be linear. The second example is a time-selective and frequency-selective model, also called doubly-selective model. Since the channel impulse response is now changing in time, the model is now longer linear. Using Basis Expansion techniques as introduced by Giannakis and Tepedelenlioglu [1998] allows to linearize the doubly-selective such that it fits as well in each form of the general model we describe. The last example is an OFDM transmission over a highly doubly-selective channel. With some additional assumptions and the same Basis Expansion techniques it may also be linearized. In comparison to the doubly-selective linear modulation model, the role of Doppler frequencies and delays is now inversed.

After having presented the general model as well as the three examples, we will detail the probability distributions of interest, then we will describe what parameters are unknown in each of the following chapters and what the corresponding stakes are. Finally we will discuss and define several performance criteria, that we will use to compare the algorithms in numerical experiments throughout the thesis.

3.1 The General Model

In its most general form we consider the linear model

$$Y = S\Psi x + \varepsilon. \tag{3.1}$$

The vector $Y = (y_1, \dots, y_K)^T$ of length K denotes the observations. The vector $\varepsilon = (\varepsilon_1, \cdots, \varepsilon_K)^T$ denotes some additive noise. The unknown parameters of the model are given by the vector $x = (x_1, \dots, x_Q)^T$ of size $Q \times 1$.

The symbol matrix S has K rows and n columns. Each of the $K{\times}n$ entries lies in some state space $\mathcal{X}$, such that the state space of S is $\mathscr{S} = \mathcal{X}^{K \times n}$. We note that in the blind classification $\mathcal{X}$ is unknown.

The measurement matrix Ψ is assumed known. It may depend on some parameter α, in this case we may also write $\Psi(\alpha)$. The matrix is of size $n \times Q$.

This model in its universality is valid for various different applications. In digital wireless communications, Y obviously denotes the received signal at the receiver, while S corresponds to the data sequence at the transmitter and thus $\mathcal{X}$ is the alphabet of the (linear) modulation scheme and is consequently finite. The matrix Ψx describes then the finite impulse response of the channel, maybe varying over time and including the multi-path propagation as well as the pulse shaping filter at the transmitter.

We will show that the EM algorithm independently splits up in the expectation and the maximization step. Even this general model formulation is sufficient to implement the latter step. However, the expectation step requires one to compute or at least estimate the posterior conditional probability distribution of S given the observations and some value of the sparse vector x'. Thus, for the conditional posterior probabilities to exist more assumptions on the distribution of S and on the distribution of the noise have to be made and be known. Even then, in general calculating the posterior distribution is not feasible. We will therefore now introduce further exemplary assumptions on the model for which we may provide an efficient implementation of the expectation step. We stress that as an efficient implementation of the expectation step for the plain EM algorithm is available, the application of the sparse EM algorithm is straightforward.

If the symbol matrix S corresponds in fact to a sequence of symbols in time s_k relating to the observations y_k, then it is reasonable to consider the following block structure for S:

$$S = \begin{bmatrix} s_1 & 0 & \cdots & & 0 \\ 0 & s_2 & 0 & \cdots & 0 \\ & & \vdots & & \\ 0 & \cdots & & 0 & s_K \end{bmatrix}. \tag{3.2}$$

We assume that $(s_k)_k$ is a time-homogeneous (or possibly inhomogeneous) Markov chain on the state space $\mathcal{S} = \mathcal{X}^L$ of dimension $1 \times L$. Each of the entries $s_k = [s_k(1), \dots, s_k(L)]$ lies in the alphabet $\mathcal{X}$. Since we consider only digital communications scenarios, we assume that $\mathcal{X}$ is finite, relating to some (linear) modulation scheme at the transmitter. We assume as well that the transition kernel or more precisely the transition matrix is known. One could also allow s_k to be time-inhomogeneous, since the algorithms we present cater also for these cases as long as the transition kernels are known.

The matrix S thus has $n = KL$ columns. In this case, the measurement matrix

$$\Psi = \begin{bmatrix} \Psi_1 \\ \vdots \\ \Psi_K \end{bmatrix} \qquad \text{or depending on } \theta: \qquad \Psi(\theta) = \begin{bmatrix} \Psi_1(\theta) \\ \vdots \\ \Psi_K(\theta) \end{bmatrix} \tag{3.3}$$

is given in block form, where each block Ψ_k corresponds to one of the symbols s_k. Furthermore, we assume that the parameter decomposes as $\theta = (\theta_1, \cdots, \theta_Q)$ such that the q-th column $\psi_k(\theta_q)$ of $\Psi_k(\theta)$ is a function of θ_q. Hence, the sensing matrix at time k decomposes into

$$\Psi_k(\theta) = [\psi_k(\theta_1), \dots, \psi_k(\theta_Q)].$$

If S is given sequentially as in (3.2), then the observation y_k at time k only depends on s_k, such that (s_k, y_k) is a hidden Markov model (HMM). The observation equation is given by

$$y_k = \sum_{q=1}^{Q} s_k \psi_k(\theta_q) x_q + \varepsilon_k = s_k \Psi_k(\theta) x + \varepsilon_k. \tag{3.4}$$

In digital communications, we may finally denote the data symbol at the transmitter at the time step k by a_k, which is drawn from $\mathcal{X}$. Since we consider blind identification and classification, we consider the data symbols to be uncoded, or the coding to be unknown. For more information on coding schemes, see for example Bordin and Bruno [2008], Lehmann [2008]. Therefore, the sequence a_k may be considered to be independent in time and each symbol is drawn uniformly from $\mathcal{X}$. Then the symbol s_k at time k is given by a concatenation of the L most recent symbols:

$$s_k = [a_k, \ldots, a_{k-L+1}]. \tag{3.5}$$

Observe that in this case the transition matrix of s_k is obviously known.

The EM algorithm is known to work well for exponential families, i.e. for the noise stemming from that family. For the ease of notation we will assume that each of the components ε_k is independently drawn from the complex normal distribution $\mathcal{N}_{\mathbb{C}}(0, \sigma^2)$ with variance σ^2.

3.2 The Sparse General Model

We define the L0-norm of a vector $x = (x_1, \ldots, x_Q)^T$ by

$$\|x\|_0 = \sum_{q=1}^{Q} \mathbb{1}_{\{x_q \neq 0\}}, \tag{3.6}$$

i.e. the number of components that are different from zero.

A vector x of dimension Q is then said to be *sparse* if $r = \|x\|_0 \ll Q$, i.e. if only a few components are non-zero.

In many applications the parameter vector x is indeed sparse or may be considered sparse. For example, in digital communications the finite impulse response of a frequency-selective channels is often sparse, as in underwater communications, residential ultrawideband channels and digital television channels amongst others that have been considered by Bajwa et al. [2008a] and Cotter and Rao [2002] and others. We describe to corresponding frequency-selective channel model in Section 3.4.

Furthermore, x may be sparse, if an actual, non-linear model is linearized by introducing a basis on the parameter space. Then, an using an overcomplete basis may yield better estimation results. In digital communications, this has been for example considered by Sharp and Scaglione [2008]. Let the actual model be given by

$$Y = S\Psi(\lambda)\beta + \varepsilon,$$

where now the parameter λ is also unknown. It lies in some parameter space Θ. If the measurement function ψ is complicated, direct estimation of the unknown parameters λ_m and β_m becomes infeasible. In many cases it can be accurately enough estimated by approximating the continuous parameter space Θ by a finite, discrete grid $\theta = (\theta_1, \cdots, \theta_Q)$, see for example Section 3.5. The grid might be chosen such that the columns of the sensing matrix $\Psi(\theta)$ are an over-complete basis for the image space of S. This is obviously the case if $\operatorname{rank}(\Psi(\theta)) \geq KL$, implying $Q \geq KL$. It is exact if the true parameters λ_m lie on the grid and approximate if not. Then Model (3.4) clearly is sparse because only those grid points close to the λ_m will have corresponding non-zero coefficients.

3.3 Probability Distributions

For the ease of notation, let us assume that we are not in a blind classification setting such that the model, i.e. the modulation scheme, is known. We assume furthermore, that the model is given sequentially as in (3.4) and we neglect any dependence of the measurement matrices Ψ_k on any (known) parameters α.

Since in each example the probability distributions are either discrete or Gaussian, it is needless to discuss in detail the underlying probability space and existence of any kernels, distributions or integrals. We will simply assume that all quantities are defined on some probability space $(\Omega, \mathcal{B}, \mathbb{P})$, where Ω denotes the space, $\mathcal{B}$ its Borel sigma algebra, and $\mathbb{P}$ a probability measure on this space.

Let $\theta = (x, \sigma)$. The likelihood of the observation y_k at time step k conditional to the current state $s_k = s$ for some $s \in \mathcal{S}$ depends thus only on θ since Ψ_k is considered to be known. It may be expressed as

$$g_k\left(y_k|s;\theta\right) = \frac{1}{\pi\sigma^2}\exp\left(-\frac{1}{\sigma^2}\left|s\Psi_k x - y_k\right|^2\right). \tag{3.7}$$

As mentioned before, we assume that the transition kernel of the Markov chain s_k from time step k to $k+1$ is known. For $s_k = s$, it has the density $q\left(\cdot\,|s\right)$, such that

$$\mathbb{P}(s_{k+1} \in B|s_k = s) = \int_B q\left(s'\,|s\right)d\mathbb{P}(s')$$

for $B \in \mathcal{B}$. The Markov chain is considered to be time homogeneous, such that q does not depend on the time k. This assumption is mainly chosen to simplify notations and because none of the examples requires a non-homogeneous Markov chain. All the presented algorithms extend easily to the non-homogeneous case, as long as the transition probabilities remain known. If the state space $\mathcal{S}$ of s_k is finite, then obviously q denotes a transition matrix and $q\left(s'\,|s\right)$ is the transition probability from $s_k = s$ to $s_{k+1} = s'$.

If the symbols s_k of the Markov chain depend on the L most recent data symbols as in (3.5) or in any of the presented examples, then the transition probabilities from state $s = [a_0, \dots, a_{L-1}]^T$ at time step k to $s' = [a'_0, \dots, a'_{L-1}]^T$ at time step $k+1$ are given by

$$q\left(s'\,|s\right) = \left\{\begin{array}{ll} \frac{1}{d} & \text{if } [a_0, \dots, a_{L-2}] = [a'_1, \dots, a'_{L-1}] \\ 0 & \text{otherwise} \end{array}\right., \tag{3.8}$$

where d is again the size of the modulation alphabet. If the symbols coincide in each of the components except for the last one, then there is a possible transition, otherwise the probability is zero. This is due to the fact, that the propagation from s_k to s_{k+1} is a simple shift in the coefficients.

For $1 \leq k \leq j \leq K$, we introduce the vector short-hand notations $s_{k:j} = (s_k, \dots, s_j)$ for a sequence of states and similarly, $y_{k:j} = (y_k, \dots, y_j)$ for a sequence of observations.

We de note the density of the joint distribution of the unknown transmitted symbol matrix S and the received observation vector Y by

$$f(S, Y; \theta) \tag{3.9}$$

which depends on the parameter θ.

The filtering distribution of the symbol s_k is defined as the conditional distribution of s_k given all the past observations $y_{1:k}$ up to time k. Assume that it allows the density $p_k(\cdot|y_{1:k};\theta)$ such that for $B \in \mathcal{B}$

$$\mathbb{P}\left(s_k \in B|y_{1:k};\theta\right) = \int_B p_k(s|y_{1:k};\theta)d\mathbb{P}(s). \tag{3.10}$$

If the state space $\mathcal{S}$ is finite like in each of the examples, then

$$p_k(s|y_{1:k};\theta) = \mathbb{P}\left(s_k = s|y_{1:k};\theta\right) \tag{3.11}$$

denotes the filtering probability for each $s \in \mathcal{S}$.

The smoothing or posterior distribution of the symbol s_k is the conditional distribution of s_k given the past as well as the future observations $Y = y_{1:K}$ up to time K. We denote its density by $p_k(\cdot|Y;\theta)$ such that for $B \in \mathcal{B}$

$$\mathbb{P}\left(s_k \in B|Y;\theta\right) = \int_B p_k(s|Y;\theta)d\mathbb{P}(s). \tag{3.12}$$

Like for the filtering, if the state space $\mathcal{S}$ is finite, then

$$p_k(s|Y;\theta) = \mathbb{P}\left(s_k = s|Y;\theta\right) \tag{3.13}$$

denotes the smoothing probability for each $s \in \mathcal{S}$.

In the same way we define the (joint) smoothing distribution S as the conditional distribution of S given the observation vector Y. We denote its density by $p(\cdot|Y;\theta)$. In the finite state space we have

$$p(\cdot|Y;\theta) = \mathbb{P}(S = \cdot|Y;\theta). \tag{3.14}$$

Let the backward filtering distribution of s_k given the future observations $y_{k:K}$ from k to K exist. Its density is denoted $p_k(\cdot|y_{k:K};\theta)$ such that we have again for $B \in \mathcal{B}$

$$\mathbb{P}\left(s_k \in B|y_{k:K};\theta\right) = \int_B p_k(s|y_{k:K};\theta)d\mathbb{P}(s). \tag{3.15}$$

If the state space $\mathcal{S}$ is again finite, then

$$p_k(s|y_{k:K};\theta) = \mathbb{P}\left(s_k = s|y_{k:K};\theta\right) \tag{3.16}$$

denotes the backward filtering probability for each $s \in \mathcal{S}$.

The expressions $p_k(\cdot|y_{1:k};\theta)$, $p_k(\cdot|Y;\theta)$ and $p_k(\cdot|y_{k:K};\theta)$ should be read as functions from $\mathcal{S}$ to $\mathbb{R}^+$. They should not be read as functions of the observations as well, since obviously the dimension of the observation space is different for each of these quantities and for each k.

We equivalently define the joint distributions, i.e. of state trajectories $s_{1:k} \in \mathcal{S}^k$, and denote the densities $p_{1:k}(\cdot|y_{1:k};\theta)$ for the joint filtering.

3.4 The Standard Model: Frequency-Selective Multi-Path Channel

Let $\mathcal{X}$ be the alphabet of symbols of size $d = |\mathcal{X}|$ in a linear modulation scheme. Suppose, that we have a sequence $\{a_k\}_{k\geq 0}$ of symbols that are uniformly and independently drawn from $\mathcal{X}$ and transmitted at symbol rate T (the algorithm being truly blind, we do not take into account the coding schemes; see Bordin and Bruno [2008] and Lehmann [2008]).

Each symbol is modulated onto the carrier function $p : \mathbb{R}^+ \to \mathbb{R}$ to get the continuous, transmitted signal

$$x(t) = \sum_{j=-\infty}^{\infty} a_j g(jT - t).$$

The function $p(t)$ has a peak at $t = 0$ and is decaying for $t =\to \pm\infty$. The function g might for example be a Root Nyquist function.

The signal is passed through a constant multipath fading channel with additive complex Gaussian noise of unknown variance σ^2. If the mth of M paths has a delay τ_m and a fading coefficient β_m, the received signal is given by

$$y(t) = \sum_{m=1}^{M} \beta_m \sum_{j=-\infty}^{\infty} a_j p(jT - t - \tau_m) + \varepsilon(t),$$

where $\varepsilon(t)$ is complex Gaussian noise with constant variance σ^2. $y(t)$ is sampled at the symbol period T to receive K observations $y_k = y(kT)$, $k = 1, \dots, K$. If we rewrite the model as

$$y_k = \sum_{j=-\infty}^{\infty} a_j \sum_{m=1}^{M} \beta_m p((j-k)T - \tau_m) + \varepsilon_k,$$

we obtain a convolution model with constant channel coefficients. Normally in practice, we assume that the maximal delay has an upper bound and the function p decays quickly, such that it is sufficient to only consider a finite linear convolution of small order L, i.e

$$y_k = \sum_{l=0}^{L-1} a_{k-l} h_l + \varepsilon_k, \quad \varepsilon_k \sim \mathcal{N}_{\mathbb{C}}\left(0, \sigma^2\right),$$

where $h_l = \sum_{m=1}^{M} \beta_m p\big((l - \lfloor L/2 \rfloor)T - \tau_m\big)$ are the coefficients of the channel impulse response and $\{\varepsilon_k\}_k$ is a sequence of i.i.d. complex circular Gaussian variables with variance σ^2.

Denote by $h = (h_0, \dots, h_{L-1})^T$ the channel impulse response. We rewrite the model in matrix notation to obtain a Hidden Markov Model (HMM):

$$\begin{aligned} s_k &= Q s_{k-1} + \mathbf{w}_k \\ y_k &= h^T s_k + \varepsilon_k, \quad \varepsilon_k \sim \mathcal{N}_{\mathbb{C}}\left(0, \sigma^2\right), \end{aligned}$$

where $s_k = [a_k, a_{k-1}, \dots, a_{k-L+1}]^T$ and $\mathbf{w}_k = [a_k, 0, \dots, 0]^T$. Q is a shift matrix, which is zero except for the elements underneath the main diagonal. A state consists thus of the L most recent symbols, where the current symbol is stored at the first position of the L-dimensional vector. For a more detailed description of the model, see for example Tse and Viswanath [2005]. The state space of the Markov chain $\{s_k\}_k$ is $\mathcal{S} = \mathcal{X}^L$ and is of size d^L.

This model is a special case of Model (3.4) with $L = Q$. The vector x is equal to h and the measurement matrices are equal to identity matrices :

$$\Psi_k = \mathcal{I}_L,$$

where $\mathcal{I}_L$ denotes the identity matrix of dimension $L \times L$. Then we have equally

$$\Psi = [\underbrace{\mathcal{I}_L, \dots, \mathcal{I}_L}_{K \text{ times}}]^T.$$

In this case the matrix form of the model may be simplified to

$$Y = S'h + \varepsilon, \tag{3.17}$$

where S' denotes the matrix with rows equal to s_1 to s_K.

Alternatively, if the pulse shaping filter is known, one can let $\phi_l(\theta_q) = p(lT - \tau_q)$ so that each entry of x directly corresponds to an attenuation and the unknown parameters are given by τ_q and x.

3.5 The Doubly-Selective Channel Model

We consider a linear modulation scheme in presence of a doubly-selective multipath channel. Let $\mathcal{X}$ be the alphabet of the modulation scheme and $a_{2-L:K} = (a_{2-L}, \dots, a_K)$ a symbol sequence generated independently and uniformly from $\mathcal{X}$ at symbol rate T. Again we do not consider coding or assume the coding to be unknown. The analog transmitted signal is then given by

$$a(t) = \sum_{\kappa=2-L}^{K} a_\kappa p(t - \kappa T)$$

where p is the modulation pulse function which is assumed to be known. The constant L denotes the length of the support of p.

The impulse response of the channel with M paths is given by

$$h(t,\tau) = \sum_{m=1}^{M} \beta_m \mathrm{e}^{j\omega_m t} \delta(\tau - \tau_m),$$

where $\delta(0) = 1$ and 0 otherwise. The attenuations on each path are given by β_m, the Doppler frequencies by ω_m and the delays by τ_m. We assume that we have lower and upper bounds on the delay, $(\tau_{\min}, \tau_{\max})$, and on the Doppler frequencies, $(\omega_{\min}, \omega_{\max})$.

Then the observation at time t is given by

$$\begin{aligned} y(t) &= \int_{\tau_{\min}}^{\tau_{\max}} h(t,\tau) a(t) \mathrm{d}\tau + \varepsilon(t) && (3.18) \\ &= \sum_{\kappa=0}^{K} a_\kappa \sum_{m=1}^{M} \beta_m \mathrm{e}^{j\omega_m t} p(t - \kappa T - \tau_m) + \varepsilon(t). && (3.19) \end{aligned}$$

If the modulation function p decays quickly such that its support lies within L taps, then after resampling at the symbol rate the model is equivalent to

$$y_k = y(kT) = \sum_{l=0}^{L-1} a_{k-l} \sum_{m=1}^{M} \phi_l(\lambda_m, k) \beta_m + \varepsilon_k, \tag{3.20}$$

where

$$\phi_l(\lambda_m, k) = \mathrm{e}^{j\omega_m kT} \tilde{p}(lT - \tau_m) \tag{3.21}$$

with $\lambda_m = (\tau_m, \omega_m)$ and $\tilde{p}$ is the L-truncated version of the modulation function p.

Direct estimation of the unknown parameters λ_m and β_m is not feasible, especially if M is unknown. However, using methods like Basis Expansion, the model may linearize by using some basis on the parameter space. We propose here to use some grid $\alpha = (\alpha_1, \cdots, \alpha_Q)$ of Q points on the two-dimensional space of delays and frequencies as in Sharp and Scaglione [2008]. We stress, that it is not in the scope of this work to explain how to choose the grid. We will rather concentrate on the algorithms to solve the problem, once an appropriate grid is found. See the work by Sharp and Scaglione [2008] for more information on the grid. If the actual parameters lie on the grid, the new model is equivalent. Otherwise Model (3.20) is now approximated by

$$y_k = \sum_{l=0}^{L-1} a_{k-l} \sum_{q=1}^{Q} \phi_l(\alpha_q, k) x_q + \varepsilon_k. \tag{3.22}$$

We define again $x = (x_1, \cdots, x_Q)^T$ and $s_k = (a_k, \cdots, a_{k-L+1})$ and introduce the notation

$$\psi_k(\alpha_q) = \left(\phi_0(\alpha_q, k), \cdots, \phi_{L-1}(\alpha_q, k)\right)^T, \tag{3.23}$$

as well as the sensing matrix $\Psi_k(\alpha)$ with q-th column $\Psi_k(\alpha)[q] = \psi_k(\alpha_q)$. Then Model (3.22) rewrites

$$y_k = s_k \Psi_k(\alpha) x + \varepsilon_k \tag{3.24}$$

for $k = 1, \cdots, K$ or in matrix form

$$Y = S\Psi(\alpha)x + \varepsilon, \tag{3.25}$$

with the same notations as in (3.1), (3.3) and (3.2).

3.6 The OFDM Transmission Model

We consider an OFDM transmission over a doubly-selective channel consisting of M multipaths. Each path m has a delay τ_m, a Doppler frequency ω_m and the attenuation h_m. We denote the pulse-shaping filter at time t by $p(t)$.

The time-discrete baseband representation of the impulse response of the channel at time κ may be decomposed into

$$h[\kappa, l] = \sum_{m=1}^{M} e^{j\omega_m \kappa} p(lT_s - \tau_m) h_m$$

where T_s is the sampling period which is considered equal to the symbol period. The support of the impulse response is furthermore considered to be finite of order L and hence, $h[\kappa, l] = 0$ for $l < 0$ and $l \geq L$. This is common practice in the digital communications domain and reasonable if the impulse function decays quickly and the delays have an upper bound.

Let $\mathcal{X}$ be the alphabet of the modulation scheme (BPSK, QPSK, ...) chosen for the OFDM transmission. We assume that all sub-carriers are modulated with the same constellation. The block of K data symbols a_0 to a_{K-1} is sampled uniformly from $\mathcal{X}$. Then the transmitted OFDM signal with the cyclic prefix of length L is given by the inverse FFT of a_n, namely

$$s[\kappa] = \sum_{n=0}^{K-1} a_n e^{j2\pi \frac{(\kappa - L)n}{K}} \quad \text{for} \quad \kappa = 0, \ldots, K+L-1.$$

The FFT of the received sampled baseband signal writes

$$\begin{aligned} y[k] &= \sum_{\kappa=0}^{K-1} e^{-j2\pi \frac{k\kappa}{K}} \sum_{l=0}^{L-1} h[\kappa+L, l] s[\kappa+L-l] \\ &= \sum_{n=0}^{K-1} a_n \sum_{\kappa=0}^{K-1} \left(\sum_{l=0}^{L-1} h[\kappa+L, l] e^{-j2\pi \frac{ln}{K}} \right) e^{-j2\pi \frac{(m-n)\kappa}{K}} \end{aligned} \tag{3.26}$$

We assume that the delays are multiples of the symbol rate, i.e. $\tau_m = l_m T_s$ for $l_m \in \mathbb{N}$ and furthermore $p(lT_s - \tau_m) = \delta(l - l_m)$ where $\delta(0) = 1$ and 0 otherwise. We also assume that $\omega_m = \frac{\mu_m}{K}$ where $\mu_m \in \mathbb{N}$. Then the observations simplify to

$$\begin{aligned} y[k] &= \sum_{m=0}^{M-1} h_m \sum_{\kappa=0}^{K-1} e^{-j2\pi \frac{nl_q}{K}} \delta\Big((k-n-\mu_m) \bmod K\Big) a_n \\ &= \sum_{m=0}^{M-1} \tilde{h}_m e^{-j2\pi \frac{kl_q}{K}} \sum_{\kappa=0}^{K-1} \delta\Big((k-n-\mu_m) \bmod K\Big) a_n, \end{aligned}$$

where $\tilde{h}_m = h_m e^{j2\pi \frac{\mu_m l_q}{K}}$.

For $B \in \mathbb{N}$, let B/K denote an upper bound on the Doppler spread. We define the vector index function $\Delta(\mu_m) \in \mathbb{R}^{B+1}$ where the μ_m-th entry is equal to 1 and all others are 0. We regroup the $B+1$ current symbols in the vector

$$s_k = \left[a_{(k)_{\mathrm{mod}\,K}}, \ldots, a_{(k-B)_{\mathrm{mod}\,K}}\right] \in \mathcal{X}^{B+1}$$

where $a_{(k)_{\mathrm{mod}\,K}} = a_n$ for $n = (k)_{\mathrm{mod}\,K}$. Define as well

$$\psi_k(\lambda_q) = \Delta(\mu_m) e^{-j2\pi \frac{kl_q}{K}},$$

where the parameter vector is given by $\lambda_q = (\mu_m, l_q)$.

The observations are then finally given by

$$y[k] = s_k \sum_{m=0}^{M-1} \psi_k(\lambda_q) \tilde{h}_m. \tag{3.27}$$

Let $\alpha = (\alpha_1, \ldots, \alpha_G)$ be a grid of G points on the two-dimensional parameter space of Doppler frequencies and delays. If the true values lie on this grid, then Model (3.27) is equivalent to

$$y[k] = s_k \sum_{g=1}^{G} \psi_m(\alpha_g) x_g = s_k \Psi_k(\alpha) x, \tag{3.28}$$

where $\Psi_k(\alpha) = [\psi_k(\alpha_1), \ldots, \psi_k(\alpha_G)]$ is known and $x \in \mathbb{C}^G$ denotes the unknown sparse parameter vector whose non-zero elements are equal to $\tilde{h}_m$. Thus, even though the channel identification problem is not naturally sparse, it becomes sparse by introducing a fine grid on the parameter space.

Neglecting the cyclic prefix for $k < L$, the random process (s_k, y_k) from $k = L$ to $\tilde{K} = K+L-1$ is a Hidden Markov Model (HMM). The observation y_k is independent of $y_{k'}$ and $s_{k'}$ given s_k for $k \neq k'$. It depends only on an additive Gaussian complex noise $\varepsilon[k]$ with variance σ^2:

$$y_k = y[k] = s_k \Psi_k(\alpha) x + \varepsilon[k]. \tag{3.29}$$

We rewrite Model (3.29) in matrix form

$$Y = S\Psi(\alpha)x + \varepsilon, \tag{3.30}$$

where $Y = (y[L], \ldots, y[\tilde{K}])^T$,

$$S = \begin{bmatrix} s_L & 0 & \cdots & 0 \\ & \vdots & & \\ 0 & \cdots & 0 & s_{\tilde{K}} \end{bmatrix} \text{ and } \Psi(\alpha) = \begin{bmatrix} \Psi_L(\alpha) \\ \vdots \\ \Psi_{\tilde{K}}(\alpha) \end{bmatrix}. \tag{3.31}$$

Since the grid is considered to be fixed, the sensing matrix $\Psi_k(\alpha)$ is known. The only unknown parameter is thus x.

3.7 Linear Modulation Schemes

To complete the model description, we will now describe the alphabets of the most current linear modulation schemes.

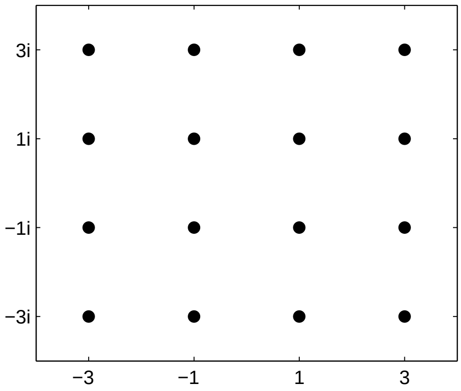

Figure 3.1: Alphabet of a 16-QAM

The alphabet $\mathcal{X}$ of any Quadrature-Amplitude Modulation (QAM) is a grid in the complex plane. If the alphabet consists of d points, then the modulation is called d-QAM. Its alphabet is then given by

$$\mathcal{X} = \left\{ a + ib \,|\, a,\, b \in \pm\{1, 3, ..., (\sqrt{d} - 1)\} \right\}.$$

Typical schemes are 4-QAM, 16-QAM, 64-QAM and 256-QAM. The alphabets of the 32-QAM and the 128-QAM are not exactly quadratic. They could be seen as 36-QAM and 144-QAM, where the most energetic symbols in the corners are removed. Very often, the alphabets are normalized, such that mean energy of the transmitted symbols is equal to one. The alphabet of a 16-QAM is given in Fig. 3.1 as an example.

The alphabet of a Binary Phase-Shift Keying (BPSK) modulation is equal to $\mathcal{X} = \{-1, 1\}$ and the alphabet of Quadrature Phase-Shift Keying is equal to the (normalized) alphabet of a 4-QAM. More generally, the alphabet of larger Phase Shift Keying modulations are equally space points on the unit circle in the complex plane. The alphabet of a d-PSK modulation is thus equal to

$$\mathcal{X} = \left\{ e^{i2\pi k/d} | k \in \{0, \ldots, d-1\} \right\}.$$

3.8 Unknown Quantities for Each Level of the Blind Classification Problem

In the 'easiest' problem, the filtering and smoothing, that we present in Chapter 4, we assume that only the symbols s_k are unknown. The parameter vector x, the variance σ^2, the channel order L, as well as the alphabet $\mathcal{X}$ of the modulation are known or at least fixed to some value - not necessarily the true values.

On the next level, the blind identification presented in Chapter 5, the parameters x and σ^2 are additionally unknown. We denote the vector of unknown parameters by $\theta = (x, \sigma)$. The modulation and the channel order L are still fixed.

Finally, in the blind classification everything is unknown, the symbols s_k, the channel order L, the parameter vector θ as well as the modulation alphabet $\mathcal{X}$. We presume however that the measurement matrix Ψ remains known, which is trivial for the time-invariant channel model since it is equal to a block of identity matrices.

In this work, we do not look at the estimation of other parameters like the symbol period T or at how to synchronize the observations with the transmitted symbols. We presume that their estimation has been carried out beforehand to the algorithms presented here. We do however show numerical results to simulate the influence of estimation errors on these parameters.

3.9 Comparison Criteria for the Monte-Carlo Experiments

For the numerical comparison of the algorithms we use mainly two criteria, the Symbol Error Rate (SER) and the Mean-Squared Error (MSE) of the unknown coefficients.

The SER is the percentage of symbols that have not been correctly estimated by the algorithm. To estimate the transmitted symbols, we use the soft-bit information. If $\hat{\theta}$ denotes the point estimate of the parameter vector x, then estimated symbol $\tilde{s_k}(\hat{\theta})$ is given by the expectation of s_k over the smoothing distribution $p_k(\cdot|Y;\hat{\theta})$ (or its estimation $\hat{p}_k(\cdot|Y;\hat{\theta})$):

$$\tilde{s_k}(\hat{\theta}) = \mathbb{E}_{\hat{\theta}}[s_k] = \int_{\mathcal{S}} s\, p_k(s|Y;\hat{\theta})ds.$$

Since in digital communications the modulation alphabet is finite, we may compare $\tilde{s_k}$ to each symbol in the state space and define the estimate

$$\hat{s_k}(\hat{\theta}) = \arg\min_{s\in\mathcal{S}} \|\tilde{s_k}(\hat{\theta}) - s\|^2.$$

The symbol error rate is then given by

$$\mathrm{SER}(\hat{\theta}) = \frac{1}{K}\sum_{k=1}^{K} \mathbb{1}_{\hat{s_k}(\hat{\theta})\neq s_k}. \tag{3.32}$$

As a second criterion we consider the error in the estimation of the finite impulse response of the channel averaged over time. Let $\hat{x}$ be the estimate of x. Then the MSE of the channel is defined as

$$\mathrm{MSE}(\hat{x}) = \frac{1}{K}\sum_{k=1}^{K} \|\Psi_k(\hat{x} - x)\|^2\,. \tag{3.33}$$

We note that the MSE describes the estimation error in the (time-varying) finite impulse response of the channel and not in the parameters themselves.

We also only consider the MSE of the impulse response and do not include the standard deviation. The main issue in the numerous Monte-Carlo experiments we carried out was the convergence of the estimator of the finite impulse response of the the channel, since the convergence of the estimator of the standard deviation did not pose any serious problems. The subsequent symbol detection is much more sensitive to the quality of the channel estimator than to the standard deviation.

Chapter 4

Filtering and Smoothing

This chapter is dedicated to the lowest estimation level of the blind classification techniques. At this level, we place ourselves in the situation that all parameters of the model are known (or fixed). We are thus interested in the estimation of the symbols s_k. The channel, the noise level and the modulation are fixed. More specifically, the estimation of s_k may have two meanings. On the one hand, we may be interested in recovering the symbol, or in other words estimating the most likely sequence of symbols given the parameters and the observations. This is known as the maximum a posterior (MAP) estimate of the data. On the other hand, we might rather be interested in the distribution of s_k, be it the filtering distribution or the posterior distribution.

In this chapter, the model we use will be in its most restrictive form. The symbol matrix S is then given are given sequentially by $(s_k)_k$ as in (3.2) such that they form a hidden Markov chain. The observations y_k are hence given by (3.4) such that (s_k, y_k) is a hidden Markov model. The state space of s_k is finite and furthermore $s_k = [a_k, \dots, a_{k-L+1}]$ as in (3.5) such that the transition of s_k from k to $k+1$ is a simple shift of the components of s_k plus a uniform random component on the first position. The reason for the restriction is that even in this most restrictive form the model includes all presented digital communications examples. Even more important is that this level is the most critical level with respect to computational efficiency and computing time. In a typical blind classification setting, the smoothing level will take up to at least 90% of the computation time. It is therefore essential that the algorithms on this level are computationally as efficient as possible. It turns out that the restrictive model allows for much more efficient algorithms than in more general models. Furthermore, smoothing in general has already been considered in depth and we refer therefore to other sources like Doucet et al. [2001], Cappé et al. [2007], Baum et al. [1970], Viterbi [1967] or Fearnhead et al. [2008].

We will thus describe and develop algorithms that derive or approximate the filtering and smoothing probabilities of the the unknown data sequence, given the channel or at least some estimate of the unknown channel. The filtering probability

$$p_k(s|y_{1:k};\theta) = \mathbb{P}(s_k = s|y_{1:k};\theta)$$

of an unknown symbol s_k is the conditional probability given only the past observations up to time k, whereas the smoothing or a posteriori probabilities

$$p_k(s|Y;\theta) = \mathbb{P}(s_k = s|Y;\theta)$$

are conditional on the past as well as the future observations $Y = y_{1:K}$. Since deriving the the maximum a posteriori estimate of the data is at least algorithmically almost equivalent, we will include this problem in this chapter. Maximum a posteriori estimation means finding the data sequence with maximal probability given the knowledge of the complete observation vector Y.

In the big picture, these algorithms are essential to implement the expectation step of the expectation-maximization algorithm that itself is essential for the model estimator. On the other hand, the smoothing should not only be seen as a small piece of the big picture, but it is in itself an interesting problem. For example, if the finite impulse response of the transmission channel is known, than the algorithms we will describe are the state of the art to estimate the unknown data sequence.

If the state space of the discrete hidden Markov model as in Chapter 3 is not too large, it is feasible to derive the exact filtering and smoothing probabilities as well as the exact maximum a posteriori estimate. The smoothing distribution are most efficiently derived with the Baum-Welch algorithm by Baum et al. [1970] which we describe in Section 4.2.1. On the other hand, the Viterbi algorithm by Viterbi [1967] is the well-known and most efficient method to derive the maximum a posterior estimate. It will be described in Section 4.2.2.

If the state space becomes too large such that the complexity of the Baum-Welch algorithm and of the Viterbi algorithm become infeasible, we have to use approximate algorithms. The most well-known approximate variants of the Viterbi algorithm are the M Algorithm by Anderson and Mohan [1984] and the T Algorithm by Simmons [1990]. Both algorithms neglect a large part of the state space after each time step using a different deterministic scheme, see Section 4.3.3.

Particle Filtering is a very well-studied and powerful Monte-Carlo concept to estimate the filtering probabilities in very general hidden Markov models, especially for non-discrete state spaces. Since we only consider finite state spaces, the plain Particle Filtering may be significantly improved by considering all potential offsprings. We have thus no need of an importance sampling distribution. The resulting algorithm will be described in detail in Section 4.3.

Since we are mainly interested in the smoothing distributions as part of the EM algorithm, we describe several Particle Smoothing algorithms which all rely however on the particle filters described in Section 4.3. Most of these algorithms use additional backward iterations to adapt the approximation of the filtering distribution to get an approximation of the smoothing distribution.

Our contributions in this area are manifold. On the Particle Filtering level, we derived a very efficient implementation for this model such that the complexity of the marginal filtering is reduced from $\mathcal{O}(N^2)$ to $\mathcal{O}(N \log N)$ and showed that the same is true for the Fixed-Interval Smoothing. We applied several Particle Smoothing algorithms to this setting and developed a new algorithm, the Joint Two-Filter Smoothing, that removes some of the disadvantages of the other smoothing algorithms. It keeps the complexity low, but on the same time it is the only algorithm that does not base the support of the smoothing distribution exclusively on the support of the forward filtering distribution. We derived as well a general framework that includes approximate Viterbi algorithms like the M- and T-Algorithm as well as Particle Filtering algorithms. The insight is, that these algorithms that are meant to estimate quite different types of quantities, are essentially equivalent in terms of implementation of the algorithm. Furthermore, this framework easily allows to derive new algorithms by combining the knowledge about Particle Filtering and approximate Viterbi algorithms. On the lowest level, the selection step of the Particle Filtering, we showed that random selection schemes like the L2 optimal selection by Fearnhead and Clifford [2003] are clearly superior to deterministic selection schemes. Furthermore, we extended the L2 optimal selection to more general statistical distances like the Kullback-Leibler or the Chi-Square distance. We could therefore for example derive a Chi-Square optimal selection, where the selection of the particle positions is proportional to the square roots of the weights.

We will start by giving a short overview of the vast number of publications in this area. The results that are relevant for this work like the Baum-Welch algorithm, the Viterbi algorithm or several Particle Smoothing algorithms are described in more detail in the following sections. We will then describe exact smoothing and exact maximum a posterior estimation, followed by their approximate counterparts, the approximate Viterbi algorithms and the Particle Smoothing algorithms. We will in detail describe Particle Filtering in discrete state space models and then how to pass to a smoothing estimation. Finally, we will discuss the possible particle selection schemes.

We will conclude this chapter with an extensive Monte-Carlo study to compare the numerous considered algorithms in simulations. Since we are mainly interested in the smoothing algorithms as part of the blind identification problem, i.e. as part of the EM algorithm, most of the computational results are as well on this problem. This means, that we implemented the expectation step of the EM algorithm with the described smoothing algorithms and have thus a means to compare the smoothing algorithms to each other. In order to understand the results, it is however not necessary to have a deep knowledge of the EM algorithm. It is sufficient to know, that the EM algorithm is an iterative algorithm that starts with some random estimate of the unknown parameters $\theta = (x, \sigma)$ and then iterates the smoothing algorithms with an update step where a better estimate of θ is derived. The performance and robustness depends essentially on the smoothing step. For these reasons, we have decided to show the simulation results in this chapter.

4.1 State of the Art

Exact smoothing in discrete states space models is carried out via the Baum-Welch algorithm, first introduced by Baum et al. [1970].

On the other hand, the Viterbi algorithm by Viterbi [1967] is a very well known and the most efficient method to derive the maximum a posteriori (MAP) estimate of the unknown data sequence. Other references on the Viterbi algorithm are Forney [1973] and Bahl et al. [1974]. Robertson et al. [1995] give a more recent comparison on MAP algorithms, that operate in the log-domain. This is often more efficient, because it avoids the expensive calculation of the exponential function. A rigorous discussion of the complexity of several coding algorithms like the Viterbi is given by Anderson and Mohan [1984].

The Viterbi algorithm has been applied in numerous areas, like network survivability by Nikolopoulos et al. [1997], bioinformatics by Ehret et al. [2001], speech recognition (see for example the book by Jelinek [1997]), digital communications by Forney [1972] or underwater communications by Feder and Catipovic [1991]. A parallel version of the Viterbi has been presented by Reeve and Amarasinghe [2006] that allows an efficient implementation on FPGAs.

A List Viterbi Algorithm has been proposed by Seshadri and Sundberg [1994] and Nikolopoulos et al. [1997] to estimate not only the most likely sequence of symbols, but also the M best sequences, for some $M \geq 1$.

Many approximate methods have been presented to reduce the complexity of the Viterbi algorithm, as for example the M-algorithm by Anderson and Mohan [1984] and the T-algorithm by Simmons [1990] that both neglect a large part of the state space after each step. They will be discussed in Section 4.3.3. Other possibilities are a Beam Search as in Odell et al. [1994], Roark [2001] and Collins [2002] and Best-First Search as in Charniak et al. [1998].

One of the first works on Particle Filtering or Sequential Monte-Carlo methods was by Gordon et al. [1993] which was shortly followed by Kong et al. [1994] and Liu and Chen [1995, 1998]. In the last two decades an abundance of work followed. A good starting point is for example the work by Doucet et al. [2001]. Since the year 1993, several extensions have been developed as for example the Auxiliary Particle Filtering by Pitt and Shephard [1999]. A very recent work is by Cornebise [2009] concentrating on automatic and optimal estimation of the Particle Filtering parameters that are normally tuned by hand.

A lot of convergence results are known on Particle Filtering, many of them are due to Del Moral [2004]. A thorough description is also given by Cappé et al. [2007]. Some other results include the work by Crisan and Doucet [2000] and Douc and Moulines [2008]. A more practical summary may be found in Crisan and Doucet [2002].

Particle Filtering has been applied to various research areas, like target tracking (see e.g. Ristic et al. [2004]), sensor networks (see e.g. Djuric et al. [2004], Wang and Zhu [2009], biological engineering, biomedical engineering, finances and robotics to name but a few. In digital communications the concept has as well been applied excessively. We will therefore only give a few exemplary references. An overview over applications to communications up to 2002 is found in Djuric et al. [2002]. Combined data and symbol timing estimation has been considered by Ghirmai et al. [2005]. Fast Fading channels Bertozzi et al. [2003b] A recommendable reference is as well the PhD thesis by Punskaya [2003], which gives a thorough introduction to Particle Filtering in digital communications and shows as well that using a deterministic proposal for the particle positions is preferable in discrete state space models. Yee et al. [2007] applied Particle Filtering to OFDM models where in addition the channel order is unknown. Since the channel order is however considered fixed, i.e. the transition kernel is a Dirac measure, the derivation is somewhat artificial and rather complex.

Based on the vast research results on Particle Filtering, a related research area has emerged, the Particle Smoothing. All Particle Smoothing algorithms use a Particle Filter and additional steps to approximate the smoothing or a posteriori distributions.

The algorithm which is conceptually closest to the exact smoothing equivalent, the Baum-Welch algorithm, is the Fixed-Interval Smoothing introduced by Doucet et al. [2000] and Godsill et al. [2004].

Another idea to Particle Smoothing is the Fixed-Lag Smoothing. It is based on the forgetting properties of the hidden Markov chain which implies that the distribution of s_k given all observations Y will be close to the distribution of s_k given the observations $y_{1:k+\Delta}$ up to time $k+\Delta$. It was first introduced by Kitagawa [1996] followed by an application to computer vision by Isard and Blake [1998] while Olsson et al. [2008] made the presentation rigorous and proved the convergence to the distribution of interest under technical integrability conditions.

Based on a two-filter decomposition of the smoothing distribution by Mayne [1966], the Two-filter Smoothing by Kitagawa [1994, 1996] and Briers et al. [2004] has been developed. Recently, Fearnhead et al. [2008] published a novel Two-Filter Particle Smoothing algorithm, which is a step toward a particle algorithm that actually samples the particle positions from the smoothing distribution. Their algorithm involves a forward and a backward particle filter and a third filter that samples new smoothing positions based on couples of positions from the forward and backward particle filter. In general state space models this algorithm is very novel and yields promising insights as well as computational results especially if the transition kernel is not sparse. However, in the discrete Model 3 where the transition matrix is in addition very sparse, the smoothing had to be based on filtering distributions that are far apart, which would however make the computational complexity infeasible. For more details, see Section 4.6.3.

In the last years, there have been attempts by Godsill et al. [2001] and Bertozzi et al. [2003a] to combine the Viterbi Algorithm with Particle Filtering methods in continuous state spaces to estimate the maximum a posteriori (MAP) estimate of the complete data sequence. The idea consists of running a particle filter on the continuous state space and considering the particles as a discretization of the state space on which the Viterbi search is then used to derive the MAP estimate. The M- and T-algorithm can again be employed to reduce the complexity. The Viterbi search does however not take into account the way the particle positions have been proposed and resampled.

Another issue of Particle Filtering is the selection or resampling of particles. In general, continuous state space models, it is necessary to resample particles in order to avoid degeneracy of the particle weights. Many algorithms have been proposed, like multinomial sampling, stratified sampling by Carpenter et al. [1999] and many others. However, in the discrete state space models, these algorithms are inefficient. Standard approaches are here deterministic schemes like the Best-Weights Selection by Tugnait [1981] and the Threshold-Comparison Selection by Simmons [1990]. More recently, Fearnhead and Clifford [2003] introduced a random selection scheme that minimizes the expected distance measured in the L2 norm between the proposed and the selected particles.

4.2 Exact Smoothing and Maximum A Posteriori Estimate

The most efficient algorithm to derive the exact smoothing distribution $p_k(\cdot|Y;\theta)$ is the Baum-Welch algorithm by Baum et al. [1970]. It splits up into three parts: a forward filtering part, a backward filtering part, and a third part where these filtering probabilities are combined. Since the filtering part is a very close analog of Particle Filtering in discrete state space models and the complete Baum-Welch algorithm is as well a very close analog of the Fixed-Interval Smoothing, we will now give a detailed description of it. In general models the complexity of the Baum-Welch algorithm is $\mathcal{O}(KD^2)$, where K is the number of observations and D is the size of the state space of the hidden Markov chain. However, because of the specific sparse structure of the model the complexity reduces to $\mathcal{O}(KD)$. Therefore, the complexity of the Baum-Welch algorithm is feasible on up-to-date computers even in moderately sized models like a 16-QAM modulation with $L = 4$. Hence, whenever it is possible to use the Baum-Welch algorithm, it is clearly preferable, since it derives the exact smoothing probabilities in contrast to Particle Smoothing algorithms.

On the other hand, the Viterbi algorithm by Viterbi [1967] is a well-known, efficient algorithm to derive the maximum a posteriori estimate of the transmitted data sequence. We show that algorithmically it is very close to the filtering part of the Baum-Welch algorithm. The essential difference is that it takes the maximum of the quantities which the Baum-Welch algorithm is summing over.

4.2.1 Baum-Welch Algorithm

The Baum-Welch algorithm introduced by Baum et al. [1970] is a well-known and well-studied method to derive the marginal smoothing probabilities $p_k(\cdot|Y;\theta)$ in discrete state space models and it is optimal in the sense of computational efficiency. The algorithm splits up into two parts. The first part is an iterative procedure to calculate the filtering probabilities $p_k(\cdot|y_{1:k};\theta)$, while the second part is as well iterative but rather calculating the time-reversed look-ahead probabilities

$$p_k(\cdot|y_{k+1:K};\theta) = \mathbb{P}(s_k = \cdot|y_{k+1:K};\theta).$$

Let us for the moment assume that we have already run these two parts and thus know $p_k(\cdot|y_{1:k};\theta)$ and $p_k(\cdot|y_{k+1:K};\theta)$ for all time steps $k \in \{1,\ldots,K\}$. Then the smoothing probabilities $p_k(\cdot|Y;\theta)$ are derived by calculating for each $s \in \mathcal{S}$:

$$\begin{aligned} p_k(s|Y;\theta) &= \mathbb{P}(s_k = s|y_{1:k}, y_{k+1:K};\theta) \\ &= \frac{f_{1:k}(y_{1:k}|s_k;\theta)\mathbb{P}(s_k = s|y_{k+1:K};\theta)}{f_{1:k}(y_{1:k}|y_{k+1:K};\theta)} \\ &= \frac{\mathbb{P}(s_k = s|y_{1:k};\theta)\mathbb{P}(s_k = s)\mathbb{P}(s_k = s|y_{k+1:K};\theta)}{f_{1:k}(y_{1:k};\theta)f_{1:k}(y_{1:k}|y_{k+1:K};\theta)} \\ &\propto \mathbb{P}(s_k = s|y_{1:k};\theta)\mathbb{P}(s_k = s|y_{k+1:K};\theta), \end{aligned}$$

where $f_{1:k}(\cdot;\theta)$ denotes the density function $y_{1:k}$ conditional on s_k . $f_{1:k}(\cdot|s_k;\theta)$ and $f_{1:k}(\cdot|y_{k+1:K};\theta)$ denote the respective conditional densities. Since similar decompositions will occur frequently in this chapter, we provide a lot of details here to familiarize with the ideas. Note, that the second and third line are just applications of Bayes' theorem. The first application of Bayes' theorem holds since $y_{1:k}$ is independent of $y_{k+1:K}$ given s_k. In the last line we neglected all constant terms that do not depend on s. The proportionality factor is not important since the weights are easily normalized since we know they have to sum to one. The complexity is thus $\mathcal{O}(KD)$.

We now turn to the derivation of $p_k(\cdot|y_{1:k};\theta)$. It is based on a similar decomposition:

$$\begin{aligned} p_{k+1}(s|y_{1:k+1};\theta) &= \sum_{s'\in\mathcal{S}} \mathbb{P}\left(s_k = s', s_{k+1} = s|y_{1:k+1};\theta\right) \\ &\propto g_{k+1}\left(y_{k+1}|s;\theta\right) \sum_{s'\in\mathcal{S}} q\left(s\left|s'\right.\right) p_k(s'|y_{1:k};\theta) \qquad (4.1) \end{aligned}$$

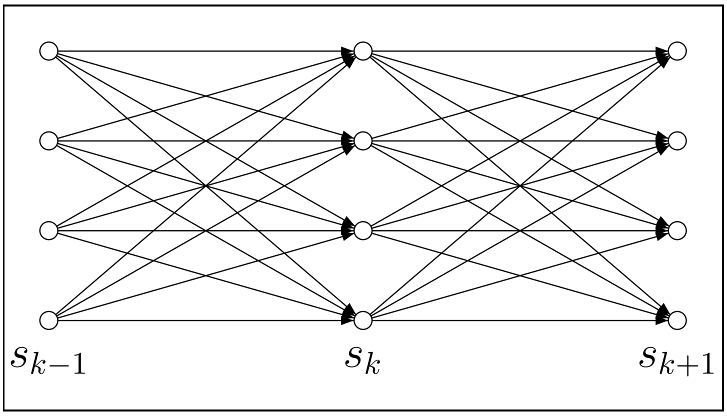

Figure 4.1: Schematics of Filtering Part of the Baum-Welch algorithm

The schematics of the resulting algorithm are illustrated in Fig. 4.1 for a small example of a state space $\mathcal{S}$ with 4 states. To derive the probability $p_{k+1}(s|y_{1:k+1};\theta)$ for each $s \in \mathcal{S}$, we have to know the filtering probability $p_k(s'|y_{1:k};\theta)$ of each state $s' \in \mathcal{S}$ at time k. This is multiplied by the transition probabilities $q\left(s\left|s'\right.\right)$ to yield the lookahead probabilities

$$\mathbb{P}(s_{k+1} = s, s_k = s'|y_{1:k};\theta) = q\left(s\left|s'\right.\right) p_k(s'|y_{1:k};\theta). \qquad (4.2)$$

The filtering probability $p_{k+1}(s|y_{1:k+1};\theta)$ is then obtained by summing all the lookahead probabilities and reweighting that by the data likelihood $g_{k+1}\left(y_{k+1}|s;\theta\right)$. To prevent a possible computational underflow, the probabilities have to be normalized appropriately.

The last remaining part is the derivation of the backward lookahead probabilities $p_{k+1}(\cdot|y_{k+1:K};\theta)$. These may be decomposed into

$$p_{k+1}(s|y_{k+1:K};\theta) \propto \sum_{s'\in\mathcal{S}} g_{k+2}\left(y_{k+2}|s';\theta\right) p_{k+2}(s'|y_{k+2:K};\theta) q\left(s'\left|s\right.\right).$$

The derivation of $p_{k+1}(s|y_{1:k+1};\theta)$ is thus very similar to the filtering, now however the iterations are backward in time.

The complexity of the forward and the backward filter are each of $\mathcal{O}(KD^2)$ in general discrete state space models. Because of the specific structure of the model in Chapter 3 the matrix of transition probabilities is however very sparse. With a clever arrangement of the state space and clever coding, the complexity may be significantly reduced to $\mathcal{O}(KD)$, i.e. the quadratic factor may be removed.

4.2.2 Viterbi Algorithm

In contrast to the Baum-Welch algorithm the Viterbi algorithm by Viterbi [1967] is the most efficient algorithm to derive the maximum a posteriori estimate of the transmitted data sequence in a discrete hidden Markov model, which is given by

$$\arg\max_{x_{1:K}\in\mathcal{S}^K} p_{1:K}(x_{1:K}|Y;\theta), \qquad (4.3)$$

where the joint smoothing distribution is given by $p_{1:K}(\cdot|Y;\theta) = \mathbb{P}(s_{1:K} = \cdot|Y;\theta)$.

The Viterbi algorithm also exploits the sequential structure of the trellis as given in Fig.4.1. It is not necessary to calculate $p_{1:K}(x_{1:K}|Y;\theta)$ for all possible data sequences $x_{1:K} \in \mathcal{S}^K$. It is instead possible to update the shortest path to each symbol at $k-1$ to the shortest path of each symbol at k in a sequential way similar to the Baum-Welch filtering. Observe, that the shortest path to a symbol s at time step k includes the shortest path to one of the symbols at time step $k-1$. Let

$$V_k(x_k) = \underset{x_{1:k-1}\in\mathcal{S}^{k-1}}{\arg\max}\; p_{1:k}(x_{1:k}|y_{1:k};\theta) \tag{4.4}$$

denote the shortest path to symbol x_k at time step k. If $V_k(x_k) = x_{1:k-1}$ for some $x_{1:k-1} \in \mathcal{S}^{k-1}$ then $V_{k-2}(x_{k-2}) = x_{1:k-2}$. In other words, the shortest path to x_{k-1} is part of the shortest path to x_k. To simplify the update to the next time step, we define the extended path $W(x_k)$ by adding x_k to the vector $V(x_k)$:

$$W_k(x_k^i) = \big(V_k(x_k^i), x_k^i\big) \in \mathcal{S}^k.$$

Similar to (4.1), we have

$$p_{1:k}(x_{1:k}|y_{1:k};\theta) \propto g_k\left(y_k|x_k;\theta\right) q\left(x_k\,|x_{k-1}\right) p_{1:k-1}(x_{1:k-1}|y_{1:k-1};\theta).$$

and therefore

$$\begin{aligned} V_k(x_k) &= \underset{x_{1:k-1}\in\mathcal{S}^{k-1}}{\arg\max}\; q\left(x_k\,|x_{k-1}\right) p_{1:k-1}(x_{1:k-1}|y_{1:k-1};\theta) && (4.5)\\ &= \underset{W_{k-1}(x_{k-1})}{\arg\max}\; q\left(x_k\,|x_{k-1}\right) p_{1:k-1}(W_{k-1}(x_{k-1})|y_{1:k-1};\theta), && (4.6) \end{aligned}$$

since $p_{1:k-1}(V_{k-1}(x_{k-1})|y_{1:k-1};\theta) \geq p_{1:k-1}(x_{1:k-1}|y_{1:k-1};\theta)$ for any $x_{1:k-1} \in \mathcal{S}^{k-1}$.

Observe that the maximization in (4.6) is much easier than in (4.5), because to derive the maximum in (4.6) it is only necessary to regard each state $x_{k-1} \in \mathcal{S}$ since $W_{k-1}(x_{k-1})$ is known. (4.6) is thus a one dimensional search, while (4.5) is a $k-1$-dimensional search.

Thus, to update to k we have to calculate the lookahead probabilities for each x_k and x_{k-1} like in the Baum-Welch filtering:

$$\mathbb{P}\big(s_k = x_k, s_{1:k-1} = W_{k-1}(x_{k-1})\big|y_{1:k-1};\theta\big) = q\left(x_k\,|x_{k-1}\right) p_{1:k-1}(W_{k-1}(x_{k-1})|y_{1:k-1};\theta).$$

Instead of summing over the lookahead probabilities as in the Baum-Welch filtering we will now take the maximum:

$$p_{1:k}(W_k(x_k)|y_{1:k};\theta) = g_k\left(y_k|x_k;\theta\right)\, \underset{W_{k-1}(x_{k-1})}{\arg\max}\; \mathbb{P}(s_k = x_k, s_{1:k-1}{=}W_{k-1}(x_{k-1})|y_{1:k-1};\theta).$$

Hence, the Viterbi algorithm is algorithmically very close to the filtering part of the Baum-Welch algorithm. The complexity is thus the same as well. For that reason we have chosen to present the two algorithms in the same chapter.

4.3 Particle Filtering

Particle Filtering is a powerful concept that is used to approximate the filtering distribution $p_k(\cdot|y_{1:k};\theta)$ of s_k in general state space models. A Particle Filtering approximation $\hat{p}_k(\cdot|y_{1:k};\theta)$ is a discretization of the measure $p_k(\cdot|y_{1:k};\theta)$ on the state space $\mathcal{S}$. A 'representative' sample of points ξ_k^i - called particles - is chosen from $\mathcal{S}$ and each of these particles is assigned a probability weight. With the help of the resulting discrete distribution functionals of s_k like e.g. the

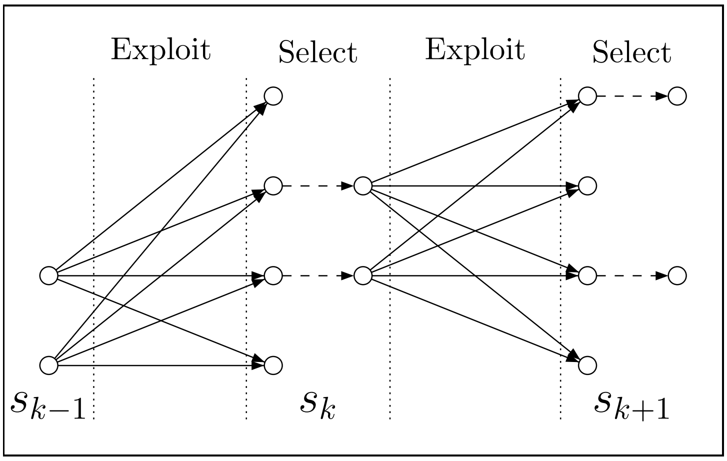

Figure 4.2: Schematics of Update Step in Marginal Particle Filtering

expectation may be easily approximated. A further aspect of Particle Filtering is the sequential nature. Given an approximation at time k, the particle filter may be easily and sequentially updated based on the knowledge of the transition of the hidden Markov chain $(s_k)_k$ and of the observation y_{k+1}. In general models, a proposal or importance distribution has to be chosen from which new particles will be drawn before being reweighted to account for the discrepancy between the proposal and the target distribution. Since the state space in digital communications is finite, it is possible to improve significantly the *plain* particle estimation algorithms by considering all the potential offsprings of a given particle. As a consequence, there is no need to select an importance distribution. Punskaya [2003] showed that this is superior to the common importance distributions. We will therefore only consider Particle Filtering in discrete state space models and explain in detail how to iterate the particle filter. For Particle Filtering in general state space models we refer the reader to plenty of other well written sources like Doucet et al. [2000], Godsill et al. [2004], Cappé et al. [2007], Punskaya [2003].

In discrete state space models there is a small but essential difference between marginal and joint filtering, i.e. between particle filters that approximate the marginal filtering distribution $p_k(\cdot|y_{1:k};\theta) = \mathbb{P}(s_k = \cdot|y_{1:k};\theta)$ and filters that approximate the joint filtering distribution $p_{1:k}(\cdot|y_{1:k};\theta) = \mathbb{P}(s_{1:k} = \cdot|y_{1:k};\theta)$. In the latter case, a particle ξ_k^i is a whole trajectory of symbols from time 1 to k, whereas in the former filter a particle denotes a single state in $\mathcal{S}$. Since in the marginal filtering many particles would often have the same offsprings, it is more efficient and intelligent to combine these particles. That is why we describe marginal and joint particle filters separately in Sections 4.3.1 and 4.3.2.

In general state space models, a resampling step is necessary to prevent the particle filter from degenerating. A particle filter is called degenerated if only very few particles have considerable weights and all other particles have weights equal or very close to zero. In discrete models with exhaustive offspring exploration, the equivalent is a selection step, where a small number of particles has to be selected from the set of all possible offsprings. It would be inefficient to use standard resampling algorithms that often and intendedly duplicate particles, since these particles would then forever coincide due to the deterministic propagation. In Section 4.4, we will therefore discuss more efficient existing selection schemes and introduce random selection schemes that are optimal with respect to general statistical distances.

The schematics of Particle Filtering in discrete state space models are illustrated in Fig. 4.2. At each time step, the Particle Filtering consists of two steps. The first step is an exploration step in which all the possible offsprings of the particles of the previous time step are considered. It is followed by a selection step to maintain a small number of particles. These two steps are iterated at each time step. Note that if a deterministic selection scheme is used, the whole Particle Filtering algorithm is deterministic, since the exploration is as well deterministic. Hence, running a particle filter a second time would produce the exact same particle positions and weights. This differs from Particle Filtering in continuous state space models, since in the latter case the resulting Particle Filtering distribution are random variables. If however a random selection scheme is used, then of course the Particle Filtering in discrete models is as well random.

4.3.1 Marginal Filtering

A Particle Filtering approximation at time k consists of a set of N particles $\xi_k^i \in \mathcal{S}$ (elements of $\mathcal{S}$) and assigned probability weights w_k^i for $i \in \{0, \cdots, N-1\}$. The weights w_k^i are normalized such that their sum is equal to one. We assume that the discrete distribution given by

$$\hat{p}_k(s|y_{1:k};\theta) = \sum_{i=0}^{N-1} w_k^i \delta_{\xi_k^i}(s) \tag{4.7}$$

for $s \in \mathcal{S}$ is an approximation of the target distribution, namely the marginal filtering distribution $p_k(s|y_{1:k};\theta)$. The indicator function δ is given by $\delta_{\xi_k^i}(s) = 1$, if $s = \xi_k^i$ and 0 otherwise.

We may then update the particle filter to time step $k+1$ based on the standard decomposition (4.1) of the filtering probabilities, where $p_k(\cdot|y_{1:k};\theta)$ is now replaced by its filtering approximation $\hat{p}_k(\cdot|y_{1:k};\theta)$:

$$\begin{aligned}
\hat{p}_{k+1}(s|y_{1:k+1};\theta) &\propto \sum_{s' \in \mathcal{S}} g_{k+1}(y_{k+1}|s;\theta)\, q\left(s\,\middle|\,s'\right) \hat{p}_k(s'|y_{1:k};\theta) && (4.8)\\
&= \sum_{j=0}^{N-1} g_{k+1}(y_{k+1}|s;\theta)\, q\left(s\,\middle|\,\xi_k^j\right) w_k^j \\
&\propto \sum_{i=0}^{M-1}\sum_{j=0}^{N-1} w_k^j g_{k+1}\left(y_{k+1}|\tilde{\xi}_{k+1}^i;\theta\right) q\left(\tilde{\xi}_{k+1}^i\,\middle|\,\xi_k^j\right) w_k^j\, \delta_{\tilde{\xi}_{k+1}^i}(s) \\
&\propto \sum_{i=0}^{M-1} \tilde{w}_{k+1}^i\, \delta_{\tilde{\xi}_{k+1}^i}(s), && (4.9)
\end{aligned}$$

where $\{\tilde{\xi}_{k+1}^i\}_{i<M}$ is the set of the $M \leq Nd$ possible offsprings of the current particles. Observe, that each of the N offsprings has exactly d offsprings. However, the offsprings of several particles may coincide and in that case we consider only one instance of each offspring. Thus, M may be smaller than Nd and it depends obviously on the particle positions ξ_k^i at time step k. For the remaining states in $\mathcal{S}$, the transition probabilities are zero, such that it suffices to consider the approximation only on the set of possible offsprings. The updated (unnormalized) weights $\tilde{w}_{k+1}^i$ are given by

$$\begin{aligned}
\tilde{w}_{k+1}^i &= \sum_{j=0}^{N-1} q\left(\tilde{\xi}_{k+1}^i\,\middle|\,\xi_k^j\right) w_k^j g_{k+1}\left(y_{k+1}|\tilde{\xi}_{k+1}^i;\theta\right) \\
&= \frac{1}{d} \sum_{j \in \mathcal{P}(\tilde{\xi}_{k+1}^i)} w_k^j g_{k+1}\left(y_{k+1}|\tilde{\xi}_{k+1}^i;\theta\right), && (4.10)
\end{aligned}$$

where $\mathcal{P}(\tilde{\xi}_{k+1}^i)$ is the set of indices of the particles ξ_k^i that are possible predecessors of $\tilde{\xi}_{k+1}^i$ and the transition probabilities are derived as in (3.8).

The number of necessary calculations to derive the proposed weights is linear in dN, due to the following reasoning. If $\tilde{\xi}_{k+1}^i = [a_{k+1}, \ldots, a_{k-L+2}]^T$, then the set of predecessors is equal to

$$\mathcal{P}(\tilde{\xi}_{k+1}^i) = \{j \,|\, \xi_k^j = [a_k, \cdots, a_{k-L+2}, x]^T \text{ for some } x \in \mathcal{X}\}.$$

Hence, $\mathcal{P}(\tilde{\xi}_{k+1}^i)$ and $\mathcal{P}(\tilde{\xi}_{k+1}^j)$ are equal, if $\tilde{\xi}_{k+1}^i$ and $\tilde{\xi}_{k+1}^j$ differ only in the first component, and disjoint otherwise. We introduce an order on the alphabet $\mathcal{X} = \{a^0, \cdots, a^{d-1}\}$ and to ease notation we identify symbols with their indices corresponding to this order. We denote the components (indices or symbols) of the particles by $\xi_k^i = (a_{k,0}^i, \cdots, a_{k,L-1}^i) \in \mathcal{X}^L$, such that $a_{k,0}^i$ corresponds to the current data symbol and $a_{k,L-1}^i$ to the data symbol at time $k-L+1$.

The weights and the offsprings may then be calculated in a very efficient way which we detail now. Although the exact notations are tedious, the big lines are quite intuitive. The particles are first sorted according to some order on the state space, then it becomes easy to establish equivalence classes of particles having the same offsprings as well as calculating the lookahead probabilities for these equivalence classes.

1. Eliminate the last component in ξ_k^i for $i < N$, obtaining:

   $p_k^i = (a_{k,0}^i, \cdots, a_{k,L-2}^i) \in \mathcal{X}^{L-1}$.

   Note that if $p_k^i = p_k^j$ then ξ_k^i and ξ_k^j have the same offsprings. Hence, the offsprings need only be computed once, i.e. the p_k^i split up into equivalence classes.

2. We reduce the cost of comparing the p_k^i for equality - to establish the equivalence classes - by ordering them first in lexicographical order which has in average the complexity $\mathcal{O}(N \log N)$. For $i < N$, this results in the sorted $p_k^{(i)}$, such that for all $l \leq L-2$, if $a_{k,l}^{(i)} < a_{k,l}^{(j)}$ (and $a_{k,h}^{(i)} = a_{k,h}^{(j)}$ for $h < l$), then $(i) < (j)$.

3. Now, removing the duplicates is quick (N comparisons). Let $\{u_k^i\}_{i<d_k}$ be the d_k unique elements in $\{p_k^i\}_{i<N}$ and let j_k^i denote the smallest index such that $u_k^i = p_k^{j_k^i}$ (define $j_k^{d_k+1} = N-1$).

4. Calculate sum of weights for $i < d_k$ (N operations): $\quad v_k^i = \dfrac{1}{d} \displaystyle\sum_{j=j_k^i}^{j_k^{i+1}-1} w_k^{(j)}$

5. Create the $M = d_k d$ offsprings and their weights:

$$\begin{aligned}
\tilde{\xi}_{k+1}^0 &= [a_0, u_k^0], && \text{with weight } \tilde{w}_{k+1}^0 = v_k^0 g_{k+1}\left(y_{k+1} | \tilde{\xi}_{k+1}^0; \theta\right) \\
&&& \vdots \\
\tilde{\xi}_{k+1}^{d-1} &= (a_{d-1}, u_k^0), && \text{with weight } \tilde{w}_{k+1}^{d-1} = v_k^0 g_{k+1}\left(y_{k+1} | \tilde{\xi}_{k+1}^{d-1}; \theta\right), \\
\tilde{\xi}_{k+1}^{d} &= (a_0, u_k^1), && \text{with weight } \tilde{w}_{k+1}^{d} = v_k^1 g_{k+1}\left(y_{k+1} | \tilde{\xi}_{k+1}^{d}; \theta\right), \\
&&& \vdots \\
&&& \vdots \\
\tilde{\xi}_{k+1}^{M-1} &= (a_{d-1}, u_k^{d_k}), && \text{with weight } \tilde{w}_{k+1}^{M-1} = v_k^{d_k} g_{k+1}\left(y_{k+1} | \tilde{\xi}_{k+1}^{M-1}; \theta\right).
\end{aligned}$$

Apart from the sorting, each step is linear in N. The last step is of order $\mathcal{O}(Nd)$ and in practice $\log N \ll d$, such that $\mathcal{O}(Nd)$ is dominating.

Example 1 *We will now give an illustration for $\mathcal{X} = \{x_1, x_2\}$ and $L = 3$. The state space $\mathcal{S} = \{s^0, \cdots, s^7\}$ consists of the (ordered) states $s^0 = (x_1, x_1, x_1)$, $s^1 = (x_1, x_1, x_2)$, $s^2 = (x_1, x_2, x_1)$, $\cdots$, $s^7 = (x_2, x_2, x_2)$. We consider the particle approximation $\left(\xi_k^i, w_k^i\right)$ at time k consisting of $N = 3$ particles, where e.g. $\xi_k^0 = s^0$, $\xi_k^1 = s^6$ and $\xi_k^2 = s^1$. Then, the particles ξ_k^0 and ξ_k^2 have exactly the same offsprings, namely*

$$\tilde{\xi}_{k+1}^0 = (x_1, x_1, x_1) = s^0 \text{ and } \tilde{\xi}_{k+1}^1 = (x_2, x_1, x_1) = s^4,$$

with (unnormalized) weights

$$\tilde{w}_{k+1}^0 = v_k^0 g_{k+1}\left(y_{k+1}|\tilde{\xi}_{k+1}^0;\theta\right) \text{ and } \tilde{w}_{k+1}^1 = v_k^0 g_{k+1}\left(y_{k+1}|\tilde{\xi}_{k+1}^1;\theta\right),$$

where $v_k^0 = w_k^0 + w_k^2$.

The remaining particle ξ_k^1 has different offsprings, namely $\tilde{\xi}_{k+1}^2 = s^3$ and $\tilde{\xi}_{k+1}^3 = s^7$, with weights $\tilde{w}_{k+1}^2 = v_k^1 g_{k+1}\left(y_{k+1}|\tilde{\xi}_{k+1}^2;\theta\right)$ and $\tilde{w}_{k+1}^3 = v_k^1 g_{k+1}\left(y_{k+1}|\tilde{\xi}_{k+1}^3;\theta\right)$, where $v_k^1 = w_k^1$.

To maintain a feasible computational complexity and to prevent the number of particles from growing exponentially, it is necessary to reduce the number of particles after each proposal step. That is, the new particles $(\xi_{k+1}^i)_{i<N}$ at time step $k+1$ are obtained by selecting N particles from the set of proposed particles $\tilde{\xi}_{k+1}^i$, following a certain selection scheme which will be discussed in detail in Section 4.4. If the selection algorithm yields unnormalized weights, they have now to be normalized in a final step.

We might for example use a multinomial selection scheme, where N samples are drawn from a multinomial distribution with a probability vector proportional to $\left(\tilde{w}_{k+1}^i\right)_{i<M}$. This makes the analysis of the algorithm easier; nevertheless, this procedure is known to be suboptimal.

After having selected a particle set $(\xi_{k+1}^i)_{i<N}$, we may now reiterate the described steps to obtain a Particle Filtering approximation for each time step. Note, that the particle filter may be easily initialized for $k = 1$, since the initial distribution of the symbols is known and uniform on the whole state space. The particles ξ_0^i for $i < N$ may thus be sampled randomly from a uniform distribution on the state space. They are assigned weights proportional to the data likelihood $g_0\left(y_0|\xi_0^i;\theta\right)$.

4.3.2 Joint Filtering

A joint Particle Filtering algorithm approximates the joint filtering distribution

$$p_{1:k}(\cdot|y_{1:k};\theta) = \mathbb{P}\left(s_{1:k} = \cdot|y_{1:k};\theta\right)$$

in contrast to the marginal algorithm which approximates $p_k(\cdot|y_{1:k};\theta)$ on the state space $\mathcal{S}$. The particles ξ_k^i will thus consist of trajectories of states from time step 1 to k, i.e. they lie in $\mathcal{S}^k$.

Hence, a joint particle filter $(\xi_k^i, w_k^i)_{i<N}$ constitutes the Particle Filtering approximation

$$\hat{p}_{1:k}(x_{1:k}|y_{1:k};\theta) = \sum_{i=0}^{N-1} w_k^i \delta_{\xi_k^i}\left(x_{1:k}\right)$$

of the joint filtering distribution $p_{1:k}(x_{1:k}|y_{1:k};\theta)$ for $x_{1:k} \in \mathcal{S}^k$.

The update procedure from k to $k+1$ is similar to the marginal filtering but now using the following decomposition of the joint filtering distribution:

$$p_{1:k+1}(x_{1:k+1}|y_{1:k+1};\theta) \propto g_{k+1}\left(y_{k+1}|x_{k+1};\theta\right) q\left(x_{k+1}\,|x_k\right) p_{1:k}(x_{1:k}|y_{1:k};\theta). \tag{4.11}$$

Similarly to the marginal filtering, all possible offsprings of ξ_k^i are considered yielding the proposed particles $\tilde{\xi}_{k+1}^i$. The number of proposed particles is exactly equal to dN, since none of the offsprings of the current particles ξ_k^i coincides because the history of a particle is not neglected as in the marginal filtering, but kept in the trajectory. The weight calculation is derived from (4.11) equivalently to the weights derivation in the marginal filtering. This exploration step is equally followed by a selection step using the same selection schemes as for the marginal filtering.

4.3.3 Approximate Viterbi Algorithms

The approximate Viterbi algorithms (AVAs) reduce the complexity of the original Viterbi algorithm by neglecting a certain part of the states after each time step. They are described in the Particle Filtering section because - as we will show later - they are really close and may also be seen as some kind of approximate filtering algorithm. We will describe the M-Algorithm by Anderson and Mohan [1984]. However, the T-algorithm by Simmons [1990] differs only in the way the kept states are selected.

Let x_k^i for $i \in \{1, \dots, N\}$ be N states of the state space $\mathcal{S}$ and let, as in (4.4),

$$\begin{aligned} V_k(x_k^i) &= \operatorname*{arg\,max}_{x_{1:k-1} \in \mathcal{S}^{k-1}} p_{1:k}((x_{1:k-1}, x_k^i) | y_{1:k}; \theta) && (4.12) \\ &= \operatorname*{arg\,max}_{V_{k-1}(x_{k-1})} q\left(x_k^i \,|x_{k-1}\right) p_{1:k-1}(V_{k-1}(x_{k-1}) | y_{1:k-1}; \theta) && (4.13) \end{aligned}$$

be the shortest path to x_k^i. Define again the extended path by the vector

$$W_k(x_k^i) = \left(V_k(x_k^i), x_k^i\right) \in \mathcal{S}^k.$$

Let us assume, that the shortest path to each of the states s_{k+1} at time $k+1$ passes through one of the states x_k^i. Then the shortest path to each state $x_{k+1} \in \mathcal{S}$ is given by the index

$$I_{k+1}(x_{k+1}) = \max_{i \in \{1,\dots,N\}} q\left(x_{k+1} \,|x_k^i\right) p_{1:k}(W_k(x_k^i) | y_{1:k}; \theta). \tag{4.14}$$

Then obviously, the best path to x_{k+1} is given by

$$V_{k+1}(x_{k+1}) = W_k(x_k^{I_{k+1}(x_{k+1})})$$

To reduce the complexity, a large part of the state space is now dropped and only N states are kept, which we call x_{k+1}^i for $i \in \{1, \dots, N\}$. The states are chosen in a way such that the maximal path metrics $V_{k+1}(x_{k+1})$ are kept. More formally, for any $i \in \{1, \dots, N\}$ we require that

$$V_{k+1}(x_{k+1}^i) > V_{k+1}(x_{k+1})$$

for any state x_{k+1} that has not been kept.

This procedure of calculating path metrics for each state and then dropping a large part of the state space is iterated for each time step.

Note that if the transition matrix q is sparse as in Model 3.1 then it is not necessary to calculate the path metric for each state in the state space, since most states will have a transition probability of zero. In that case it is sufficient to consider only the offsprings of the positions of the previous time step. This is similar to the described procedure of the Particle Filtering.

The T-Algorithm is similar to the M-Algorithm. However, in the selection step it is using the Threshold-Comparison Selection as given in Section 4.4.

4.4 Particle Selection

At each update step of the marginal or joint particle filter, we consider the $M \leq Nd$ offsprings of the current particles. It is therefore vital to discard a large part of these offsprings to maintain the maximal number N of particles. This problem may be more generally formulated as follows: given a discrete probability vector $\mathbf{w} = (w_i)_{i<M}$, on some finite set Ξ with $|\Xi| = M$ and $M > N$, the problem is to find a discrete distribution $\mathbf{W} = (W_i)_{i<M}$, approximating $\mathbf{w}$ in some sense, but with a number of support points in average less than N.

A natural idea consists of using one of the two following purely deterministic selection schemes.

1. **Best-Weights Selection**
   This scheme by Tugnait [1981] is a purely deterministic scheme keeping the N highest weights:
   $$W_i = \begin{cases} w_{(i)} & \text{for } i \leq N \\ 0 & \text{otherwise,} \end{cases} \tag{4.15}$$
   where $w_{(i)}$ are the largest order rank statistics, $w_{(0)} \geq w_{(1)} \geq \cdots \geq w_{(M-1)}$.

2. **Threshold-Comparison Selection This scheme by Simmons [1990]** keeps the weights which are larger than a certain fraction T of the maximal weight $w_{max} = \max_{i<M} w_i$:
   $$W_i = \begin{cases} w_i & \text{for } w_i > T w_{max} \\ 0 & \text{otherwise.} \end{cases} \tag{4.16}$$

A similar idea is the Stratified Sampling by Carpenter et al. [1999], which consists of drawing one single random variable and basing the new weights on the outcome of this variable. The new weights calculated by these approaches are biased, in the sense that $\mathbb{E}[W_i] \neq w_i$ for the weights w_i that have not been selected.

In the sequel, we will now focus on random selection schemes where the weights W_i are now considered random variables and develop new random selection schemes that optimize general statistical distances. We require that the new weights fulfill

(A1) $\mathbb{E}\left[W_i\right] = w_i$,

(A2) $\mathbb{E}\left[\sum_{i=0}^{M-1} \mathbb{1}_{\{W_i > 0\}}\right] = N$.

The first requirement guarantees the unbiasedness, whereas the second ensures that the support of the sampled distribution is in average equal to N. In general $\sum_{i=0}^{M-1} W_i \neq 1$ and it is necessary to normalize the weights after selection in order to obtain a probability distribution.

There are many ways to find a distribution satisfying these requirements. We impose additional constraints by seeking a solution $\mathbf{W}$, which minimizes $\mathbb{E}[d(\mathbf{W}, \mathbf{w})]$ for a distance (or divergence) function $d(\cdot, \cdot)$. Assume that $d(\mathbf{W}, \mathbf{w}) = \sum_{i=0}^{M-1} \phi(W_i, w_i)$, where $\phi : \mathbb{R}_+^2 \to \mathbb{R}$ fulfills the following conditions:

(H1) $\phi(w, w) = 0$ for any $w \in \mathbb{R}_+$,

(H2) for any $w' \in \mathbb{R}_+$, the function $w \mapsto \phi(w, w')$ is twice differentiable and convex,

(H3) for any $w' \in \mathbb{R}_+$, the function $w \mapsto \phi(w, w')$ is sublinear, and

$$\lim_{|w| \to \infty} \frac{\phi(w, w')}{w} = \infty.$$

We say, that a selection scheme is 'optimal' with respect to d, if the distribution of the random variable $\mathbf{W}$ minimizes $\mathbb{E}[d(\mathbf{W}, \mathbf{w})]$ among all distributions satisfying (A1) and (A2).

Theorem 5 *Any optimal selection scheme* $\boldsymbol{W}$ *with respect to* d *satisfies*

$$W_i = \begin{cases} \frac{w_i}{p_i} & \textit{with probability } p_i \\ 0 & \textit{with probability } 1{-}p_i, \end{cases}$$

and $p_i = \min\{\psi(w_i, \lambda), 1\}$, *where* $\psi(w_i, \lambda)$ *is the unique solution for* x *of the equation*

$$\phi(0, w_i) - \phi\left(\frac{w_i}{x}, w_i\right) + \frac{w_i}{x} d_1\phi\left(\frac{w_i}{x}, w_i\right) = \lambda, \tag{4.17}$$

$d_1\phi$ *denotes the first partial derivative of* ϕ *with respect to the first argument, and the constant* λ *is given by*

$$\sum_{i=0}^{M-1} \min\{\psi(w_i, \lambda), 1\} = N.$$

The proof may be found in the next Section 4.4.1. It can be remarked that the solutions for $\phi(w, w') = |w - w'|^q$ for $q \to 1$, are converging (in probability) to the deterministic selection scheme of selecting the highest weights. More generally, taking smaller values of q leads to more sparse solutions, whereas on the contrary the choice of the Chi-Squared norm. In the following, we will focus on the Kullback-Leibler divergence and the Chi-Square distance.

For the Kullback-Leibler divergence, defined by $d(\mathbf{w}, \mathbf{w}') = \sum_{i=0}^{M-1} \phi(w_i, w_i')$ with $\phi(w_i, w_i') = w_i \log \frac{w_i}{w_i'}$, it is possible to propose a selection scheme, which ensures that both

$$\sum_{i=0}^{M-1} \mathbb{1}_{\{W_i > 0\}} = N \quad \text{and} \quad \sum_{i=0}^{M-1} W_i = 1$$

at the same time (but of course, the W_i are not independent in this case). It is not obvious that the same can be obtained for other selection schemes minimizing arbitrary distances. For the Kullback-Leibler divergence we have $\phi(0, w_i) = 0$ and Equation (4.17) thus becomes

$$0 - \frac{w_i}{x} \log \frac{1}{x} + \frac{w_i}{x}\left(\log \frac{1}{x} + 1\right) = \frac{w_i}{x} = \lambda.$$

Hence, $p_i = \min\left(\frac{w_i}{\lambda}, 1\right)$ and λ is given by

$$\sum_{i=0}^{M-1} \min\left\{\frac{w_i}{\lambda}, 1\right\} = N.$$

Random Selection Scheme 1: Kullback-Leibler/L2-norm Optimal Selection
Let λ be the solution of

$$\sum_{i=0}^{M-1} \min\left\{\frac{w_i}{\lambda}, 1\right\} = N.$$

For all i such that $w_i \geq \lambda$, let

$$W_i = w_i,$$

and for all i such that $w_i < \lambda$, let

$$W_i = \begin{cases} \lambda & \text{with probability } \frac{w_i}{\lambda} \\ 0 & \text{with probability } 1 - \frac{w_i}{\lambda}. \end{cases} \tag{4.18}$$

In words: sufficiently heavy particles are kept with the same weight, while the others survive with a probability proportional to their weight, and in that case they are assigned a fixed weight λ. Besides, the Kullback-Leibler divergence-optimal sampling schemes also minimize the expected L^2-norm with $\phi(w, w') = (w - w')^2$. The problem of finding a selection scheme for this norm had already been solved by Fearnhead and Clifford [2003], who coined the name 'optimal sampling' for this scheme. The proof by Barembruch et al. [2009] is thus a generalization of the proof by Fearnhead and Clifford [2003] and it follows essentially the same lines.

We consider now the Chi-Square divergence $d(\mathbf{W}, \mathbf{w}) = \sum_{i=0}^{M-1} \frac{(W_i - w_i)^2}{w_i}$. The solutions are characterized by:

Random Selection Scheme 2: Chi-Squared Optimal Selection
Let λ be the solution of

$$\sum_{i=0}^{M-1} \min\left\{\sqrt{\frac{w_i}{\lambda}}, 1\right\} = N.$$

For all i such that $w_i \geq \lambda$, let

$$W_i = w_i,$$

and for all i such that $w_i < \lambda$, let

$$W_i \quad = \quad \begin{cases} \sqrt{w_i \lambda} & \text{with probability } \sqrt{\frac{w_i}{\lambda}} \\ 0 & \text{with probability } 1 - \sqrt{\frac{w_i}{\lambda}}. \end{cases} \tag{4.19}$$

As for scheme 2, sufficiently heavy particles are kept with the same weight. But now, the others survive with probability proportional to the square root of their weight, promoting the smallest.

Hence, the selection schemes are presented in increasing order of variability: the deterministic scheme has no variability and too small weights never survive with random selection scheme 1, unlikely particles have a chance to survive proportional to their weights, while with random selection scheme 2 the differences in the weights are scaled down by considering their square roots.

4.4.1 Proof of Theorem 5

The following lemma is a technical ingredient used in the derivation of the general solution.

Lemma 3 *Assume that W is a random variable satisfying $E[W] = w$ and $\mathbb{P}[W > 0] = p$. Let $\phi : \mathbb{R}_+^2 \to \mathbb{R}$ fulfill conditions (H1)-(H3).*

Then

$$\mathbb{E}\left[\phi(W, w)\right] \geq (1-p)\phi(0, w) + p\phi\left(\frac{w}{p}, w\right).$$

Moreover, equality is reached if and only if $W \sim (1-p)\delta_0 + p\delta_{\frac{w}{p}}$.

Proof: First note that

$$\begin{aligned} w = E[W] \quad &= \quad (1-p)\mathbb{E}[W|W{=}0] + p\mathbb{E}[W|W{>}0] \\ &= \quad p\mathbb{E}[W|W > 0], \end{aligned}$$

and so $\mathbb{E}[W|W > 0] = \frac{w}{p}$. Now

$$\begin{aligned} \mathbb{E}\left[\phi(W, w)\right] \quad &= \quad \mathbb{P}(W = 0)\mathbb{E}\left[\phi(W, w)|W = 0\right] \\ &\quad\ + \mathbb{P}(W > 0)\mathbb{E}\left[\phi(W, w)|W > 0\right] \\ &= \quad (1-p)\phi(0, w) + p\mathbb{E}\left[\phi(W, w)|W > 0\right]. \end{aligned}$$

Hence, using the convexity of $\phi(\cdot, w)$,

$$\mathbb{E}\left[\phi(W, w) | W > 0\right] \geq \phi\left(\mathbb{E}[W|W > 0], w\right) = \phi\left(\frac{w}{p}, w\right)$$

with equality if and only if W is constant equal to $\frac{w}{p}$ on the event $\{W > 0\}$. ■

We now present the proof of Theorem 5.

Proof: Let $\mathbf{W}$ be some random vector satisfying the constraints, and let $p_i = \mathbb{P}\left[W_i > 0\right]$.

- By Lemma 3,

  $$\mathbb{E}\left[d(\mathbf{W}, \mathbf{w})\right) \geq \sum_{i=1}^{R} (1 - p_i)\phi(0, w_i) + p_i \phi\left(\frac{w_i}{p_i}, w_i\right),$$

  and this bound is reached if and only if for each i, W_i is constantly equal to $\frac{w_i}{p_i}$ on the event $\{W > 0\}$.

- Thus, the problem reduces to

  Problem 1 *Find $p = (p_i)_{1 \leq i \leq R} \in \mathbb{R}^R$ minimizing*

  $$\sum_{i=1}^{R} (1 - p_i)\phi(0, w_i) + p_i \phi\left(\frac{w_i}{p_i}, w_i\right)$$

  under the constraints $0 \leq p_i \leq 1$ and $\sum_{i=1}^{R} p_i = n$.

- Denote λ and λ_i the Lagrange multipliers associated respectively with constraint $\sum_{i=1}^{n} p_i = n$ and $p_i \leq 1$ for $1 \leq i \leq R$. The Karush-Kuhn-Tucker conditions for the primal p and dual λ, $\lambda_1, \dots, \lambda_R$ optimal points are given by:

  $$-\phi(0,w_i)+\phi\left(\frac{w_i}{p_i},w_i\right)-\frac{w_i}{p_i} d^1\phi\left(\frac{w_i}{p_i}, w_i\right)+\lambda+\lambda_i = 0 \qquad (4.20)$$

  $$\lambda, \lambda_i \geq 0 \qquad (4.21)$$

  $$\lambda_i \left(p_i - 1\right) = 0 \qquad (4.22)$$

  where $d_1\phi$ denotes the partial derivative of ϕ relative to its first argument.

- If $p_i < 1$, then $\lambda_i = 0$ and using (4.20) we obtain:

  $$p_i = \psi(w_i, \lambda) \wedge 1,$$

  where $\psi(w_i, \lambda)$ is the solution of

  $$g_{w_i,\lambda}(x) = \phi(0, w_i) - \phi\left(\frac{w_i}{x}, w_i\right) + \frac{w_i}{x} d_1\phi\left(\frac{w_i}{x}, w_i\right) = \lambda. \qquad (4.23)$$

- Equation (4.23) has at most one solution in $[0, 1]$. In fact, the convexity of ϕ implies:

  $$g'_{w_i,\lambda}(x) = -\frac{w_i^2}{x^3} d^2\phi\left(\frac{w_i}{x}, w_i\right) < 0. \qquad (4.24)$$

- Hypothesis (H3) ensures that Equation (4.23) has a solution if λ is large enough. In fact,

  $$\begin{aligned} \phi(y, w_i) - y d_1\phi(y, w_i) &\leq \phi(y, w_i) - y\frac{\phi(y, w_i) - \phi(1, w_i)}{y - 1} \\ &= \frac{y\phi(1, w_i) - \phi(y, w_i)}{y - 1} \to -\infty \end{aligned}$$

  when y grows to infinity and hence

  $$\begin{aligned} g_{w_i,\lambda}(x) &= \phi(0, w_i) - \left(\phi\left(\frac{w_i}{x}, w_i\right) - \frac{w_i}{x} d^1\phi\left(\frac{w_i}{x}, w_i\right)\right) \\ &\to +\infty \quad \text{when } x \to 0. \end{aligned}$$

■

4.5 A General Framework for Approximate Viterbi Algorithms and Particle Filtering

We will now provide a general framework for both the approximate Viterbi algorithms like the M and T algorithm and Particle Filtering algorithms in discrete state space models. The framework reveals that despite their significantly different interpretation the methods are methodically very similar. We will concentrate on the marginal versions of the algorithms. The same general framework may be extended to the joint Particle Filtering and its counterpart in the Viterbi domain, the List-Viterbi algorithm by Seshadri and Sundberg [1994] which is inclined to find the N best paths through a trellis. The elements of the state space are denoted $\mathcal{S} = \left\{s^0, \cdots, s^{D-1}\right\}$. A lower index always indicates thus the time step(s), while an upper index indicates an order on the state space or the index of a selected position.

In Section 4.3 we made extensive use of the specific structure of the model in Section 3 to reduce the complexity of the update step. In this discussion we will however assume a more general model, i.e. any discrete hidden Markov model. Therefore, all the states in $\mathcal{S}$ are possible offsprings of the current particles. The link to Section 4.3 is very easy, since states that are not offsprings of any particle will just have a transition probability equal to zero.

At each time step k, we store a set of N_k positions $\xi_k^i \in \mathcal{S}$ considered to be a representative sample of $\mathcal{S}$ at time k. They are assigned weights w_k^i, whose interpretation depends on the specific algorithm. If necessary, the information a_k^i on the most probable predecessor is also stored, leading to the system $\left(\xi_k^i,\, w_k^i,\, a_k^i\right)_{i<N_k}$.

```
Algorithm 2: Marginal framework, update step
  Input  : $\left(\xi_k^i,\, w_k^i,\, a_k^i\right)_{i<N_k}$
  Output: $\left(\xi_{k+1}^i,\, w_{k+1}^i,\, a_{k+1}^i\right)_{i<N_{k+1}}$

    • Exploration step ;
      for $j \in \{0, \cdots, D-1\}$ (loop over $\mathcal{S}$) do
          $\tilde{w}_{k+1}^i = \left\| \left( q\left(s^j \,|\xi_k^i\right) w_k^i \right)_{i<N_k} \right\|_p g_{k+1}\left(y_{k+1}|s^j;\theta\right)$    (4.25)
          Optional: $\tilde{a}_{k+1}^j = \arg\max_{i<N_k} \left( q\left(s^j \,|\xi_k^i\right) w_k^i \right)$
      end

    • Selection step ;
      Input  : $(s^j, \tilde{w}_{k+1}^j)_{j\in\{0,\cdots,D-1\}}$
      Output: $N_{k+1}$ positions $\xi_{k+1}^i$ with weights $w_{k+1}^i$ (and $a_{k+1}^i$)
```

Updating to time step $k+1$ is done in two steps. The first one is the exploration step, where a weight for each state in $\mathcal{S}$ is calculated, followed by a selection step, in which a part of the states is neglected. The schematics of this iterative procedure may be seen in Fig. 4.2.

The weight calculation of the Viterbi algorithm and of the Baum-Welch algorithm are very similar. While the latter is averaging over the look-ahead probabilities $p_k(\cdot|y_{k-1};\theta)$, the Viterbi Algorithm is maximizing over them. We may interpret the averaging as taking the ℓ_1-norm of $p_k(\cdot|y_{k-1};\theta)$ and the maximizing as ∞-norm. This may be generalized to any p-norm for $p \geq 1$. The algorithm for a fixed $p \geq 1$ is given in Alg. 2.

Observe that the weights calculation in (4.25) is an approximate analog of the VA update if we choose $p = \infty$ and of (4.10) if we choose $p = 1$. The displayed calculations allow for transformations (e.g. operating in the log-domain) reducing the computational cost. The selection scheme in the selection step may be any of the schemes presented in Section 4.4.

The approximate Viterbi Algorithms fit exactly into the marginal general framework if we choose $p = \infty$. After having reached the final time step K, the best path is found by backtracking through the auxiliary information variables $\tilde{a}_k^j$. The difference between the M-Algorithm by Anderson and Mohan [1984] and the T-Algorithm by Simmons [1990] is the selection procedure. The first one employs the Best-Weights Selection, while the latter one the Threshold-Comparison Selection. Furthermore, with the general framework it becomes obvious that it is as well possible to use one of the random selection schemes, which has never been considered so far. In this case, it is however no longer possible to operate in the log-domain, because the random selection schemes work with the (normalized) probability weights.

In marginal Particle Filtering we only consider the current state and the weight, the past of each particle is irrelevant. Therefore, a_k^i is not needed. Since we want to use Decomposition 4.1 to approximate the filtering probabilities, we choose $p = 1$ in Alg. 2, such that the norm averages the predictive probabilities, instead of taking the maximal weight of the predecessors as in the approximate Viterbi Algorithms. This turns out to be the only algorithmic difference between the two concepts.

4.6 Approximate Smoothing - Particle Smoothing

4.6.1 Fixed-Interval Smoothing

The Fixed-Interval Smoothing is a Particle Smoothing technique that has been introduced by Doucet et al. [2000] and Godsill et al. [2004]. It relies on a an approximation $\hat{p}_k(\cdot|y_{1:k};\theta)$ of the filtering probabilities $p_k(\cdot|y_{1:k};\theta)$ that has been derived by a forward particle filter $(\xi_k^i, w_k^i)_{i<N}$ for $1 \le k \le K$ as in Section 4.3. The smoothing distributions $p_k(\cdot|Y;\theta)$ are then approximated by additional backward iterations that leave the particle positions unchanged but adapt the particle weights to fit the smoothing distributions. The weight adaptation relies on a decomposition of $p_k(\cdot|y_{1:k};\theta)$ that is equivalent to the Baum-Welch algorithm, such that the Fixed-Interval Smoothing is a direct approximate analog of the Baum-Welch algorithm.

We assume thus that a particle filter $(\xi_k^i, w_k^i)_{i<N}$ for $1 \le k \le K$ as in Section 4.3 is given that yields the approximation $\hat{p}_k(\cdot|y_{1:k};\theta)$ of the filtering distributions $p_k(\cdot|y_{1:k};\theta)$. The Fixed-Interval Smoothing is then based on the following decomposition of the marginal smoothing distribution $p_k(s|Y;\theta)$ for $s \in \mathcal{S}$:

$$\begin{aligned}
p_k(s|Y;\theta) &= \sum_{x\in\mathcal{S}} \mathbb{P}\left(s_k = s, s_{k+1} = x \,|\, Y;\, \theta\right) \\
&= \sum_{x\in\mathcal{S}} p_{k+1}(x|Y;\theta)\mathbb{P}\left(s_k = s|\, s_{k+1} = x,\, y_{0:k};\, \theta\right) \\
&= \sum_{x\in\mathcal{S}} p_{k+1}(x|Y;\theta)\frac{q\,(x\,|s\,)\,p_k(s|y_{1:k};\theta)}{\sum_{z\in\mathcal{S}} q\,(x\,|z\,)\,p_k(z|y_{1:k};\theta)} \\
&= p_k(s|y_{1:k};\theta)\sum_{x\in\mathcal{S}} \frac{p_{k+1}(x|Y;\theta)q\,(x\,|s\,)}{\sum_{z\in\mathcal{S}} q\,(x\,|z\,)\,p_k(z|y_{1:k};\theta)} \qquad (4.26)
\end{aligned}$$

This decomposition allows to calculate $p_k(\cdot|Y;\theta)$ recursively based on the filtering probabilities

$p_k(\cdot|y_{1:k};\theta)$ and the knowledge of $p_{k+1}(\cdot|Y;\theta)$. It serves for the particle smoothing algorithm by replacing $p_k(\cdot|y_{1:k};\theta)$ by its Particle Filtering approximation $\hat{p}_k(\cdot|y_{1:k};\theta)$. An approximation is thus obtained by iterating (4.26) from time step $K-1$ to time step 1. Note that an approximation of the smoothing distribution at time K is already at hand by using the filtering distribution $\hat{p}_K(\cdot|y_{1:K};\theta)$ at the final time step K.

Let $\hat{p}_{k+1}(\cdot|Y;\theta)$ be a particle approximation of $p_{k+1}(\cdot|Y;\theta)$ based on the particle system $(\xi_{k+1}^i, w_{k+1|K}^i)_{i<N}$ where $w_{k+1|K}^i$ now denote the smoothing weights. The particle position ξ_{k+1}^i are left unchanged. An approximation of $p_k(\cdot|Y;\theta)$ is then given by:

$$p_k(s|Y;\theta) = \sum_{i=0}^{N-1} w_{k|K}^i \delta_{\xi_k^i}(s)\,.$$

The updated weights $w_{k|K}^i$ are derived iteratively following (4.26):

$$\begin{aligned} w_{k|K}^i &= \hat{p}_k(\xi_k^i|y_{1:k};\theta) \sum_{x\in\mathcal{S}} \frac{\hat{p}_{k+1}(x|Y;\theta) q\left(x\left|\xi_k^i\right.\right)}{\sum_{z\in\mathcal{S}} q\left(x\left|z\right.\right) \hat{p}_k(z|y_{1:k};\theta)} \\ &= w_k^i \sum_{j=0}^{N-1} \frac{w_{k+1|K}^j q\left(\xi_{k+1}^j\left|\xi_k^i\right.\right)}{\sum_{l=0}^{N-1} w_k^l q\left(\xi_{k+1}^j\left|\xi_k^l\right.\right)} = w_k^i \sum_{j\in\mathcal{Q}(\xi_k^i)} \frac{\frac{1}{d} w_{k+1|K}^j}{\sum_{l=0}^{N-1} w_k^l q\left(\xi_k^l, \xi_{k+1}^j\right)} \\ &= w_k^i \sum_{j\in\mathcal{Q}(\xi_k^i)} \frac{w_{k+1|K}^j}{\sum_{l\in\mathcal{P}(\xi_{k+1}^j)} w_k^l}, \end{aligned} \tag{4.27}$$

where $\mathcal{Q}\left(\xi_k^i\right)$ denotes the indices of those particles at time $k+1$ that are offsprings of the particle ξ_k^i. The weights are derived similar to (4.9). The sum in the denominator has already been derived in the filtering algorithm, i.e. when calculating the new weights in (4.10). The remaining calculations are of order $\mathcal{O}(N)$ for the same reason as in the marginal filtering, i.e. the sets $\mathcal{Q}\left(\xi_k^i\right)$ and $\mathcal{Q}\left(\xi_k^j\right)$ for $i \neq j$ are either disjoint or equal, such that each weight $w_{k+1|K}^j$ needs only be considered once in the calculation of these sums, implying $\mathcal{O}(N)$ operations. As in the filtering iterations, arranging the sets of successors is of complexity $\mathcal{O}(N\log N)$. This is clearly superior to general state space models, where the complexity of the backward weights correction is $\mathcal{O}(N^2)$ as stated by Godsill et al. [2004]. In practice, d is generally larger than $\log N$ and the complexity $\mathcal{O}(dN)$ of the particle filter is thus larger than that of the backward iterations.

Since we will use this algorithm extensively in the following chapters, we describe it again in a more algorithmic form in Algorithm 3.

4.6.2 Fixed-Lag Smoothing

Olsson et al. [2008] suggest a Particle Smoothing algorithm which makes use of the so-called forgetting properties of the hidden Markov model, i.e. that $p_k(\cdot|Y;\theta)$ is close to the fixed-lag smoothing distribution $p_{k+\Delta}(\cdot|Y;\theta)$ as soon as the lag Δ is large enough. Note that the marginal smoothing distribution $p_k(\cdot|Y;\theta)$ can be derived from the joint smoothing distribution $p_{1:K}(\cdot|Y;\theta) = \mathbb{P}(s_{1:K} = \cdot|Y;\theta)$ by marginalizing over all time steps except for k. The forgetting properties then say that an approximation of $p_k(\cdot|Y;\theta)$ may be derived by marginalizing $p_{1:k+\Delta}(\cdot|y_{1:k+\Delta};\theta) = \mathbb{P}(s_{1:k+\Delta} = \cdot|y_{1:k+\Delta};\theta)$ instead of $\mathbb{P}(s_{1:K} = \cdot|Y;\theta)$. Then again $p_{1:k+\Delta}(\cdot|y_{1:k+\Delta};\theta)$ is approximated by a joint particle filter.

Let the joint particle filter $(\xi_k^i, w_k^i)_{i<N}$ be given as described in Section 4.3.2. We denote the component of the particle trajectory ξ_k^i at time $t \leq k$ by $\xi_k^i(t)$ such that $\xi_k^i = \left(\xi_k^i(0), \cdots, \xi_k^i(k)\right)$.

```
Algorithm 3: Fixed-interval smoothing algorithm
  Input  : $Y$, $L$, $N$, $\mathcal{X}$, $\theta = (h, \sigma)$, $d$
  Output: $\hat{p}_k(\cdot|Y;\theta)$  Smoothing distribution for $k \in \{0, \cdots, K\}$
          $\hat{l}(Y)$  Likelihood of $Y$

    • Initial step
        * For $i \in \{0, \cdots, N-1\}$, sample $\xi_1^i \sim \mathcal{U}(\mathcal{X})$ and calculate $\tilde{w}_1^i = g_1\left(y_1|\xi_1^i;\theta\right)$ .
        * Put $l_1 = \sum_{i=0}^{N-1} \tilde{w}_1^i$ and $w_1^i = (l_1)^{-1}\tilde{w}_1^i$.
    • for $k$ from 1 to $(K-1)$ (Forward Iterations) do
        1. Proposal of particles ;
            * Establish the $M_{k+1}$ descendants $\tilde{\xi}_{k+1}^i$ pour $i \in \{0, \cdots, M_{k+1}-1\}$ of $\left(\xi_k^j\right)_{j<N\cdot}$ ;
            * Calculate $\tilde{w}_{k+1}^i$ according to Equation (4.10):
                  $\tilde{w}_{k+1}^i = \sum_{j=0}^{N-1} q\left(\tilde{\xi}_{k+1}^i \middle| \xi_k^j\right) w_k^j g_{k+1}\left(y_{k+1}|\tilde{\xi}_{k+1}^i;\theta\right)$
            * Put $l_{k+1} = \sum_{i=0}^{M-1} \tilde{w}_{k+1}^i$ and $\breve{w}_{k+1}^i = (l_{k+1})^{-1}\tilde{w}_{k+1}^i$.
        2. Selection of particles, see Procedure (4.18) ;
           Input  : $\left(\tilde{\xi}_{k+1}^i, \breve{w}_{k+1}^j\right)_{i\in\{0,\cdots,M-1\}}$
           Output: $N$ positions $\xi_{k+1}^i$ with weights $w_{k+1}^i$
      end
    • for $k$ from $K-1$ to 0 (Backward Iterations) do
        Calculate $w_{k|K}^i$ according to Equation (4.27) for $i \in \{0, \cdots, N-1\}$:
            $w_{k|K}^i = w_k^i \sum_{j=0}^{N-1} \frac{w_{k+1|K}^j q\left(\xi_{k+1}^j \middle| \xi_k^i\right)}{\sum_{l=0}^{N-1} w_k^l q\left(\xi_{k+1}^j \middle| \xi_k^l\right)}$
      end
    • Likelihood: $\hat{l}(Y) = \prod_{k=0}^{K} l_k$
```

An approximation of $p_k(\cdot|Y;\theta)$ at time step k may then be obtained by marginalization of $\hat{p}_{1:k+\Delta}(\cdot|y_{1:k+\Delta};\theta)$:

$$
\begin{aligned}
\hat{p}_k(x_k|Y;\theta) &= \sum_{x_0\in\mathcal{S}} \cdots \sum_{x_{k-1}\in\mathcal{S}} \sum_{x_{k+1}\in\mathcal{S}} \cdots \sum_{x_{k+\Delta}\in\mathcal{S}} \hat{p}_{1:k+\Delta}(x_{1:k+\Delta}|y_{1:k+\Delta};\theta) \\
&= \sum_{i=0}^{N-1} w_{k+\Delta}^i \delta_{\xi_{k+\Delta}^i(k)}(x_k) \, .
\end{aligned}
$$

In words, the approximation of the marginal smoothing probabilities is given by the filtering weights at time $k+\Delta$ for the particle positions at time k of the particle trajectories. The computational complexity is $\mathcal{O}(dN)$, namely linear in N. In contrast to the Fixed-Interval

Smoothing this remains true even in more general models. For further discussion and theoretical results we refer to Olsson et al. [2008].

4.6.3 Two-Filter Smoothing

All Particle Smoothing algorithms relying on a forward particle filter - like the fixed-interval smoothing or the fixed-lag smoothing by Olsson et al. [2008] - have a common drawback. Since the points of the support of the smoothing distributions are simulated from a Particle Filtering algorithm, the approximations will fail, if the correct smoothing distribution contains points with high smoothing probabilities but small filtering probabilities. Contrarily, the two-filter smoothing algorithm by Briers et al. [2004] includes as well an independent backward information filter to identify likely states with respect to the following observations.

In this context, not only is (s_k, y_k), for $k = 1, \cdots, K$, a hidden Markov chain, but also the time reversed chain for $k = K, K-1, \cdots, 1$. Indeed, the transition probabilities form as well a probability distribution on s_k, keeping s_{k+1} fixed, i.e. $\sum_{s \in \mathcal{S}} q\left(s' \,|s\right) = 1$ for each $s' \in \mathcal{S}$. Hence, the (marginal) Particle Filtering algorithm can as well be applied backwards in time to estimate the backward filtering distribution $p_k(\cdot|y_{k:K};\theta) = \mathbb{P}\left(s_k = \cdot|y_{k:K};\theta\right)$ for $k = K, K-1 \cdots, 1$, based on the decomposition:

$$\begin{aligned} p_k(s|y_{k:K};\theta) &\propto g_k\left(y_k|s;\theta\right) \mathbb{P}\left(s_k = s|y_{k+1:K};\theta\right) \\ &\propto g_k\left(y_k|s;\theta\right) \sum_{s' \in \mathcal{S}} p_{k+1}(s'|y_{k+1:K};\theta) q\left(s' \,|s\right). \end{aligned} \tag{4.28}$$

The Two-Filter Smoothing thus presumes a forward particle filter approximating $p_k(\cdot|y_{1:k};\theta)$ as well as a backward particle filter approximating $p_k(s|y_{k:K};\theta)$. The smoothing probabilities are then approximated by combining these two filters in a specific manner.

Hence, we assume having already simulated the forward particle system $\left(\xi_k^i,\, w_k^i\right)_{i<N_F}$ with N_F particles such that

$$\hat{p}_k(\cdot|y_{1:k};\theta) = \sum_{i=0}^{N_F-1} w_{k|k}^i \delta_{\xi_{k|k}^i}(\cdot)$$

approximates the forward filtering distributions $p_k(\cdot|y_{1:k};\theta)$ for each $k \leq K$. We assume as well the backward particle filter $\left(\xi_{k|k:K}^i,\, w_{k|k:K}^i\right)_{i<N_B}$ with N_B particles such that

$$\hat{p}_k(s|y_{k:K};\theta) = \sum_{i=0}^{N_B-1} w_{k|k:K}^i \delta_{\xi_{k|k:K}^i}(s)$$

approximates the backward filtering probabilities $p_k(s|y_{k:K};\theta)$ for each $k \leq K$. Note, that the backward particle filter is conducted completely independently of the forward particle filter.

In Mayne [1966], Kitagawa [1994, 1996] and Briers et al. [2004], a two-filter smoother is used based on the following decomposition of the smoothing probabilities:

$$\begin{aligned} p_k(s|Y;\theta) &\propto p_k(s|y_{1:k};\theta) \mathbb{P}\left(s_k = s|y_{k+1:K};\theta\right) \\ &= p_k(s|y_{1:k};\theta) \sum_{s' \in \mathcal{S}} \mathbb{P}\left(s_k = s, s_{k+1} = s'|y_{k+1:K};\theta\right) \\ &= p_k(s|y_{1:k};\theta) \sum_{s' \in \mathcal{S}} q\left(s' \,|s\right) p_{k+1}(s'|y_{k+1:K};\theta) \end{aligned} \tag{4.29}$$

With the help of this decomposition, an approximation of the smoothing distribution is obtained by replacing $p_k(\cdot|y_{1:k};\theta)$ by $\hat{p}_k(\cdot|y_{1:k};\theta)$ and $p_k(\cdot|y_{k:K};\theta)$ by $\hat{p}_k(\cdot|y_{k:K};\theta)$:

$$\begin{aligned}\hat{p}_k(s|Y;\theta) &= \sum_{i=0}^{N_F-1} w_k^i \delta_{\xi_k^i}(s) \sum_{j=0}^{N_B-1} w_{k+1|k+1:K}^j q\left(\xi_{k+1|k+1:K}^j \,|s\right) \\ &= \sum_{i=0}^{N_F-1} w_{k|K}^i \delta_{\xi_k^i}(s),\end{aligned}$$

with (unnormalized) weights equal to

$$w_{k|K}^i = w_k^i \sum_{j=0}^{N_B-1} q\left(\xi_{k+1|k+1:K}^j \,|\xi_k^i\right) w_{k+1|k+1:K}^j.$$

For general state space models, it is necessary to multiply the backward variable $p_k(\cdot|y_{k:K};\theta)$ by an artificial normalizing prior as in Briers et al. [2004] to handle cases for which it is not integrable. This is not necessary in this discrete state space model as it only involves finite sums and furthermore because the backward transition density is already a probability density.

The computational complexity of calculating the smoothing weights is of order $\mathcal{O}(K\max(N_F,N_B))$, with equivalent reasoning as for the backward weights correction in the Fixed-Interval Smoothing, see Section 4.6.1. The complexity stems therefore from conducting the forward and backward filters, which are of order $\mathcal{O}(KN_Fd)$ and $\mathcal{O}(KN_Bd)$ respectively.

Unfortunately, the specific sparse structure of the transition matrix will often result in all smoothing weights being zero for a certain time k, i.e. $q\left(\xi_{k+1|k+1:K}^j \,|\xi_k^i\right) = 0$ for all $i < N_F$ and $j < N_B$. Furthermore, the smoothing approximation is still exclusively based on the support of the forward particles. It is straightforward to adjust the decomposition such that the support of the smoothing approximation is exclusively based on the backward information filter, which would however result in a similar degeneracy.

The degeneracy may be avoided by adapting the two-filter smoothing idea, such that the smoothing distribution $p_k(\cdot|Y;\theta)$ is decomposed into a forward and a backward filtering distribution that are further away than one single time step. Any state s_k at time k consists of the L most recent symbols, i.e. $s_k = (a_k,\cdots,a_{k-L+1})$. If we use the decomposition into distributions at k and $k+1$ then the corresponding particles $\xi_k^i = (a_{k,0}^i,\ldots,a_{k,L-1}^i)$ and $\xi_{k+1|k+1:K}^j = (b_{k+1,0}^j,\ldots,b_{k+1,L-1}^j)$ have to coincide in $L-1$ of the symbols in order to give rise to a common offspring at time k, i.e. $(a_{k,0}^i,\cdots,a_{k,L-2}^i) = (b_{k+1,1}^j,\cdots,b_{k+1,L-1}^j)$, which is the reason for the degeneracy.

If we use instead a forward filtering distribution at $\underline{k} = (k-l)$ and a backward filtering distribution at $\overline{k} = (k+u)$ that are more than L time steps away, i.e. for $l+u \geq L$ (where l denotes the lower and u the upper gap), then none of the symbols of the corresponding particles

$$\xi_{\underline{k}}^i = (a_{\underline{k},0}^i,\cdots,a_{\underline{k},L-1}^i) \text{ and } \xi_{\overline{k}|\overline{k}:K}^j = (b_{\overline{k},0}^j,\cdots,b_{\overline{k},L-1}^j)$$

has to coincide. The bottom $L-u$ symbols of the potential offsprings/descendants at time k are determined by $(a_{\underline{k},0}^i,\cdots,a_{\underline{k},L-u-1}^i)$, and equivalently the top $(L-u)$ symbols by $(b_{\overline{k},u}^j,\cdots,b_{\overline{k},L-1}^j)$. Hence, if we use $l+u=L$ all symbols are already determined, such that each combination of particles $\xi_{\underline{k}}^i$ and $\xi_{\overline{k}|\overline{k}:K}^j$ has exactly one offspring, yielding a complexity of $\mathcal{O}(N^2)$. If $l+u > L$, each combination has exactly d^{l+u-L} offsprings and thus the complexity raises to $\mathcal{O}\left(N^2 d^{l+u-L}\right)$, which is in general much too costly. Furthermore, the calculation of the weights has an additional factor of $\mathcal{O}(L^2)$, since the calculation of the data likelihood involves a sum of L terms and it has to be calculated for each time step between $\underline{k}$ and $\overline{k}$. we may thus use $l+u=L$ in the simulations, since the complexity $\mathcal{O}(N^2)$ is manageable, but the complexity for $l+u > L$ is in practice too large.

The decomposition of $p_k(s|Y;\theta)$ for $s \in \mathcal{S}$ into $p_{[}(\underline{k}|y_{1:[};\theta)]\cdot$ and $p_{\overline{k}}(\cdot|y_{\overline{k}:K};\theta)$ is a generalization of the decomposition in (4.29):

$$\begin{aligned} p_k(s|Y;\theta) \quad &\propto \quad \sum_{x\in\mathcal{S}}\sum_{z\in\mathcal{S}} p_{\underline{k}}(x|y_{1:\underline{k}};\theta)\mathbb{P}\big(s_k = s|s_{\underline{k}} = x, s_{\overline{k}} = z, Y;\theta\big)\, p_{\overline{k}}(z|y_{\overline{k}:K};\theta) \\ &\propto \quad \sum_{x_{\underline{k}:k-1}}\sum_{z_{k+1:\overline{k}}} p_{\underline{k}}(x_{\underline{k}}|y_{1:\underline{k}};\theta) q\left(x_{\underline{k}+1}\,|x_{\underline{k}}\right) g_{\underline{k}}\left(y_{\underline{k}}|x_{\underline{k}+1};\theta\right) \\ &\qquad \cdots q\left(s\,|x_{k-1}\right) g_k\left(y_k|s;\theta\right) q\left(z_{k+1}\,|s\right) \cdots g_{\overline{k}-1}\left(y_{\overline{k}-1}|z_{\overline{k}-1};\theta\right) q\left(z_{\overline{k}}\,|z_{\overline{k}-1}\right) p_{\overline{k}}(z_{\overline{k}}|y_{\overline{k}:K};\theta). \end{aligned}$$

As before, we replace $p_{\underline{k}}(\cdot|y_{1:\underline{k}};\theta)$ and $p_{\overline{k}}(\cdot|y_{\overline{k}:K};\theta)$ by their particle approximations $\hat{p}_{\underline{k}}(\cdot|y_{1:\underline{k}};\theta)$ and $\hat{p}_{\overline{k}}(\cdot|y_{\overline{k}:K};\theta)$ to obtain the estimate $\hat{p}_k(\cdot|Y;\theta)$ of $p_k(\cdot|Y;\theta)$.

A version of this algorithm applicable to general state space models has been proposed by using $l = u = 1$ in Fearnhead et al. [2008]. They also consider a complexity reduction by sampling N combinations of ξ^i_{k-1} and $\xi^j_{k+1|k+1:K}$ instead of taking all N^2 combinations.

4.6.4 Joined Two-Filter Smoothing

The existing smoothing algorithms suffer from the specific structure of the model. Most of them base the support of the approximated smoothing distribution exclusively on the support of the approximated filtering distribution. This works well if the forward particle filter is able to identify the states with high smoothing probabilities. This is the case, if the channel is minimum phase. When this is not the case, the forward particle filter is less likely to identify these states. On the contrary, a backward particle filter would be easily able to identify them if the channel is maximum phase. This problem would of course disappear if we use an algorithm to sample from the smoothing distribution or some form of "lookahead" or "lookbackward" strategies such as the delayed sample method by Chen et al. [2000] which generates samples of the current state by marginalizing out the future states. The computational complexity of these methods becomes however quickly too large. We propose a novel two-filter smoothing algorithm, which shows that it is however at least possible to simulate half of the particles of the smoothing distribution from the forward filtering distribution and the other half from the backward filtering distribution. If the channel is maximum phase, this algorithm will still be able to identify states with high smoothing probabilities, since half of the particles are generated in the backward path. Hence, if we knew beforehand whether the channel is better approximated by a minimum or a maximum phase channel, we could either use the forward filtering pass or the backward filtering pass, but since we normally do not have this knowledge, we generate half of the particles in each pass.

The novel two-filter algorithm exploits furthermore the fact that simulating particle positions is much more expensive than calculating the weights for existing particle positions, since simulating the positions involves testing all possible offsprings of each particle. The key idea of the algorithm is to simulate forward and backward particle positions, joining them, and then computing properly assigned weights, such that the approximated smoothing distributions have a support equal to the union of the forward particles and backward particles.

For ease of the presentation, we will explain the algorithm with the help of unnormalized distributions. Let us thus assume that a unnormalized Particle Filtering approximation $\left(\xi^i_k,\, w^i_k\right)_{i<N}$ with unnormalized weights is given such that

$$\hat{\pi}_k(s|y_{1:k};\theta) = \sum_{i=0}^{N-1} w^i_k \delta_{\xi^i_k}(s)$$

approximates the unnormalized filtering probabilities

$$\pi_k(s|y_{1:k};\theta) = \sum_{s'\in\mathcal{S}} g_k\left(y_k|s;\theta\right) q\left(s\,|s'\right) p_{k-1}(s'|y_{1:k-1};\theta).$$

The filtering distribution is then obviously given by

$$p_k(s|y_{1:k};\theta) = \frac{1}{\sum_{s'\in\mathcal{S}} \pi_k(s'|y_{1:k};\theta)} \pi_k(s|y_{1:k};\theta).$$

Similarly, we define the normalizing constant $\Omega_k = \sum_{i=0}^{N-1} w_k^i$ for the particle approximation, such that $\tilde{\pi}_k(s|y_{1:k};\theta)/\Omega_k$ is an approximation of $p_k(s|y_{1:k};\theta)$.

Equivalently, let $\left(\xi^i_{k|k:K},\, w^i_{k|k:K}\right)_{i<N}$ be an approximation of the unnormalized backward filtering distribution

$$\pi_k(s|y_{k:K};\theta) = g_k\left(y_k|s;\theta\right) \sum_{s'\in\mathcal{S}} p_{k+1}(s'|y_{k+1:K};\theta) q\left(s'\,|s\right).$$

To ease notation we have assumed that both particle filters use the same number of particles N, which is straightforward to generalize to $N_F \neq N_B$.

We propose to improve those (unnormalized) forward and backward Particle Filtering approximations by extending their support, i.e. by adding the backward particle positions to the forward particle filter and the forward particle positions to the backward particle filter respectively. We will now demonstrate this more formally for the example of the forward filtering, i.e. how to update iteratively the forward particle approximations $\left(\xi^i_k,\, w^i_k\right)$. Suppose that we already have an joined (unnormalized) forward Particle Filtering distribution $\tilde{\pi}_{k-1}(\cdot|y_{1:k-1};\theta)$ at time step $k-1$, consisting of the particles $\left(\tilde{\xi}^i_{k-1}, \tilde{w}^i_{k-1}\right)_{i<N_{k-1}}$. The number of particles is denoted by N_{k-1} to mark that it is random at each time step.

Since we want to keep the information of how the forward particles are generated (i.e. the selection scheme), we do not alter the weights of the forward particles when updating the forward filter. We only calculate new weights for the particles which have been generated in the backward filter and which do not coincide with particles of the forward filter. Let $(\tilde{\xi}^i_k)_{i<N_k}$ be the set of unique particle positions in the union $\{\xi^i_k\}_{i<N} \cup \{\xi^i_{k|k:K}\}_{i<N}$, ordered such that the first N particles coincide with the forward particles, i.e. $\tilde{\xi}^i_k = \xi^i_k$ for $i < N$, and the remaining ones are those of the backward filter, that are disjoint from the forward filter.

An improved approximation of $\pi_k(s|y_{1:k};\theta)$ is then given by

$$\begin{aligned} \tilde{\pi}_k(s|y_{1:k};\theta) &= \sum_{i=0}^{N_k-1} \tilde{w}^i_k \delta_{\tilde{\xi}^i_k}(s), \\ &= p_k(s|y_{1:k};\theta) + \sum_{i=N}^{N_k-1} \tilde{w}^i_k \delta_{\tilde{\xi}^i_k}(s), \end{aligned} \tag{4.30}$$

where we set $\tilde{w}^i_k = w^i_k$ for $i < N$ and for $i \leq N$ we derive the filtering weights $\tilde{w}^i_k$ of the new particle positions ξ^i_k, such that the exact filtering probabilities of these symbols are approximated, conditionally on $y_{1:k}$ and $\tilde{\pi}_{k-1}(\cdot|y_{1:k-1};\theta)$:

$$\tilde{w}^i_k = \sum_{j=0}^{N_{k-1}} g_k\left(y_k|\tilde{\xi}^i_k;\theta\right) q\left(\tilde{\xi}^i_k\,\middle|\,\tilde{\xi}^j_{k-1}\right) \frac{\tilde{w}^j_{k-1}}{\tilde{\Omega}_{k-1}}, \tag{4.31}$$

where $\tilde{\Omega}_{k-1} = \sum_{i=0}^{N_{k-1}-1} \tilde{w}^i_{k-1}$. We use the unnormalized weights $\tilde{w}^i_k$ rather than the normalized ones $\tilde{w}^i_k/\tilde{\Omega}_k$, to keep the information hidden in the normalizing factor and to avoid over- or underemphasizing the importance of the added particle positions.

Along the same lines, improved backward approximations $\tilde{\pi}_k(\cdot|y_{k:K};\theta)$ of $\pi_k(\cdot|y_{k:K};\theta)$ consisting of the $N_{k|k:K}$ particles $\left(\tilde{\xi}^i_{k|k:K}, \tilde{w}^i_{k|k:K}\right)_{i<N_{k|k:K}}$ may be iteratively derived based on the decomposition in (4.28). We define again $\tilde{\Omega}_{k|k:K} = \sum_{i=0}^{N_{k|k:K}-1} \tilde{w}^i_{k|k:K}$.

Now we have new approximations $\tilde{\pi}_k(\cdot|y_{1:k};\theta)/\tilde{\Omega}_k$ of the forward filtering distributions $p_k(\cdot|y_{1:k};\theta)$ and $\tilde{\pi}_k(\cdot|y_{k:K};\theta)/\tilde{\Omega}_{k|k:K}$ of the backward filtering distributions $p_k(\cdot|y_{k:K};\theta)$. The approximation of the smoothing distributions based on these filtering approximations follows the decomposition in (4.29). Hence, the Joined-Two-Filter Algorithm consists of the following steps:

1. **Run a forward particle filter** with N particles resulting in $\left(\xi_k^i, w_k^i\right)_{i<N}$ for each $k \leq K$.

2. **Run a backward information filter** with N particles resulting in $\left(\xi_{k|k:K}^i, w_{k|k:K}^i\right)_{i<N}$ for each $k \leq K$.

3. **Update the forward particle filter** resulting in $\left(\tilde{\xi}_k^i, \tilde{w}_k^i\right)_{i<N_k}$ based on (4.30) and (4.31).

4. **Update the backward particle filter** resulting in $\left(\tilde{\xi}_{k|k:K}^i, \tilde{w}_{k|k:K}^i\right)_{i<N_{k|k:K}}$.

5. **Approximate the smoothing distribution:**. Calculate the weights

$$w_{k|K}^i = \tilde{w}_k^i \sum_{j=0}^{N_{k|k:K}-1} q\left(\tilde{\xi}_{k|k:K}^j \,\middle|\, \tilde{\xi}_k^i\right) \tilde{w}_{k|k:K}^i,$$

and the normalizing constant $\Omega_{k|K} = \sum_{i=0}^{N_k-1} w_{k|K}^i$. Then

$$\hat{p}_k(s|Y;\theta) = \frac{1}{\Omega_{k|K}} \sum_{i=0}^{N_k-1} \tilde{w}_{k|K}^i \delta_{\tilde{\xi}_k^i}(s).$$

The complexity of Steps 1) and 2) is $\mathcal{O}(dN)$. Steps 3) and 4) involve each $\mathcal{O}(N)$ calculations per time step k equivalently to the marginal filtering weights calculation in (4.10). Similar to the previous algorithms, reorganizing the data to create the sets of predecessors and offsprings is of complexity $\mathcal{O}(N \log N)$. The final step 5) is of complexity $\mathcal{O}(2N\log(2N))$, see Section 4.6.3. Hence, the first two steps of deriving the particle filter are in practice the most complex steps.

4.7 Computational Results

4.7.1 Comparison of Smoothing Algorithms and Selection Schemes

The performance of the previously described algorithms is measured by the accuracy of the approximated smoothing distribution and by the convergence results of the EM algorithm incorporating the different algorithms. The accuracy of the smoothing approximation is evaluated by looking at the expected distance of the approximated distributions $\hat{p}_k(\cdot|Y;\theta)$ to the exact smoothing distributions $p_k(\cdot|Y;\theta)$, averaged over the time steps:

$$D_d = \mathbb{E}\left(\frac{1}{K}\sum_{k=1}^{K} d\Big(p_k(\cdot|Y;\theta), \hat{p}_k(\cdot|Y;\theta)\Big)\right),$$

where $d(\cdot,\cdot)$ is a measure of the distance between two (discrete) probability distributions. We used the L1-norm, i.e. $d(x,y) = \|x-y\|_1$, but the Kullback-Leibler divergence and the Chi-Square distance yielded equivalent results. The exact smoothing probabilities $p_k(\cdot|Y;\theta)$ are derived with the Baum-Welch algorithm. We are therefore restricted to a small constellation

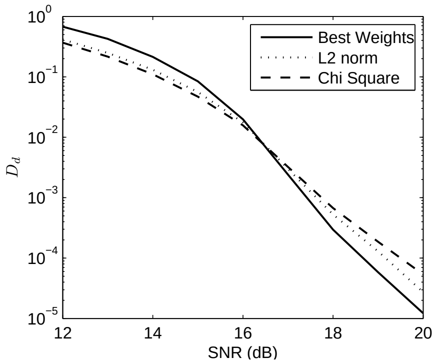

Figure 4.3: Accuracy of approximated smoothing distributions for the different selection schemes (D_d: $L1$-distance to exact smoothing distribution), 16-QAM, $h = (0.63, 0.05, -0.3)^T$, 100 particles.

size (16-QAM) and a small channel order ($L = 3$) to maintain a reasonable complexity. The convergence of the EM is measured in terms of the mean squared error

$$\mathrm{MSE}(\hat{h}) = \mathbb{E}\left(\sum_{l=0}^{L-1} \left|\hat{h}_l - h_l\right|^2\right)$$

of the current estimate of the channel coefficients $\hat{h}$ compared to the true coefficients h. We only consider the MSE of the channel coefficients and do not include the standard deviation. The main issue was the convergence of the channel, since the convergence of the standard deviation did not pose any serious problems. Furthermore, a good channel estimation is more vital for the subsequent symbol detection than the standard deviation.

If not specified otherwise the initial values for the approximate EM algorithms are chosen randomly due to the lack of additional information. We will also test initialization using the Blind Channel Acquisition (BCA) method introduced in Nguyen and Levy [2005], see Section 5.2.1. The modulation alphabets are normalized and the number of observations fixed to $K = 300$, for which the assumption of a constant channel is reasonable. Based on simulations, an appropriate choice of the fixed lag Δ for the fixed-lag smoothing is $\Delta = 15$.

We compared the Best-Weights Selection, the $L2$-Optimal Selection (equivalent to Kullback-Leibler) and the Chi-Squared Selection combined with the Fixed-Lag Smoothing of Olsson et al. [2008] on the task of estimating the smoothing distribution in a 16-QAM with channel coefficients equal to $h = (0.63, 0.05, -0.3)^T$ corresponding to a channel with two paths, where the second path has a certain delay, see Fig. 4.3. When the SNR is large, the accuracy of the best weights selection is slightly better, but if the SNR gets smaller, the results turn favorably for the random selection schemes. This is due to the fact that, if the SNR is large, the results are less corrupted by the noise, and thus the additional variability introduced by the random methods is counterproductive.

The same experiments with the fixed-interval smoothing yielded different results: The selection schemes approximated the smoothing distribution equally well regardless of the SNR. On the other hand, the selection schemes influence significantly the convergence of the EM algorithm - where the E-step is replaced by the fixed-interval smoothing coupled with each selection schemes. In a 16-QAM model, the MSE of the channel coefficients after 50 iterations is considerably smaller using the random selection schemes, see Fig. 4.4. The channel coefficients were randomly drawn for each Monte Carlo run and normalized such that $\|h\| = 1$. The difference between the selection methods increases with the size of the state space. Although in a 64-QAM, see Fig. 4.5, the best weights selection quickly approaches the true parameters after the first

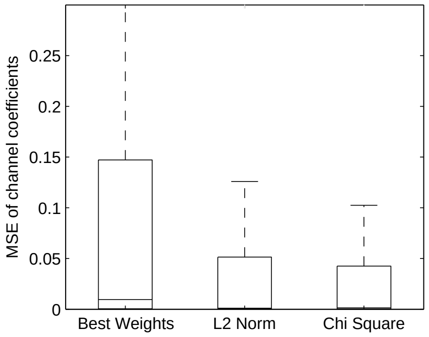

Figure 4.4: MSE of parameters after 50 iterations of EM for the different selection schemes, 16-QAM, random channel coefficients, $L = 3$.

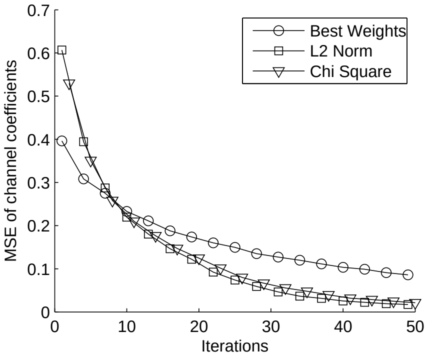

Figure 4.5: Convergence of channel coefficients for the different selection schemes, 64-QAM, random channel coefficients, $L = 3$, 200 particles.

couple of iterations, the MSE after 50 iterations is significantly larger, see Fig. 4.6. The results are similar for other parameter choices, e.g. a 256-QAM. The Chi-Square optimal selection and the Kullback-Leibler optimal selection are almost equivalent, in some cases with a slight advantage to the first one, in other cases to the second one.

In the remainder, we focus on the different smoothing algorithms, each of which is equipped with the best weights selection for a comparison, if not mentioned otherwise. We test the fixed-interval smoothing, the fixed-lag smoothing, the two-filter smoothing, the (novel) joined-two-filter smoothing and the two-filter smoothing with a larger gap of L. The EMVA algorithm by Nguyen and Levy [2005] is used as a reference. We describe it in 5.2.1. It employs the survivor paths of the Viterbi algorithm to approximate the smoothing distribution. The complexity of the Viterbi search is reduced by applying an approximate Beam Search by Jelinek [1997]. Its computational complexity of $\mathcal{O}(N \log N)$ is equivalent to the fixed-interval smoothing in this model. The accuracy of the EMVA approximation is also measured in terms of the distance D_d.

For an evaluation of the accuracy of the approximated smoothing distributions, we use again a 16-QAM modulation with random channel coefficients. Fig. 4.7 shows that the best approximation of the smoothing distribution is obtained by the fixed-interval smoothing and the joined-two-filter smoothing. The advantage of the joined-two-filter algorithm is however that it is still capable of dealing with channels, where the first coefficient is small compared to the remaining ones, see Fig. 4.8 for $h = [-0.3, 0.05, 0.8]$. The states with high smoothing probabilities are easily identified in a backward information filter, but not in a forward path, on which the support of the smoothing distributions is completely based for the other algorithms,

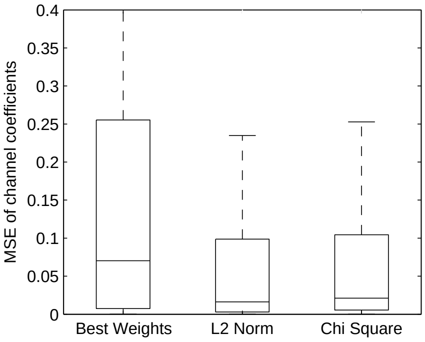

Figure 4.6: MSE of parameters after 50 iterations of EM for the different selection schemes, 64-QAM, random channel coefficients, $L = 3$, 200 particles.

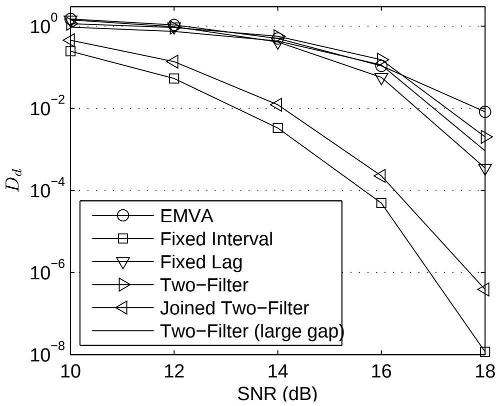

Figure 4.7: Accuracy of approximated smoothing distributions (D_d: $L1$-distance to exact smoothing distribution), uniformly random channels, 16-QAM, $N = 50$, $L = 3$

except for the two-filter smoothing algorithm with a large gap.

For the case of random channel coefficients ($L = 3$) in a 16-QAM and one random initialization, Fig. 4.9 shows that the convergence of the joined-two-filter smoothing algorithm is equivalent to the fixed-interval smoothing, while the remaining algorithms cannot keep up. By using several initializations with the BCA method, the results may be significantly improved, see Fig. 4.10, especially for the EMVA, which is now superior to the fixed-lag and the two-filter smoothing. When using three random initializations, the MSE is slightly inferior compared to the BCA method for each algorithm, although the EMVA is now only slightly superior to the fixed-lag smoothing. This indicates that the most superior algorithms are the fixed-interval smoothing and the joined-two-filter smoothing, followed by the EMVA, which is however much more sensible to the choice of the initial value.

Fig. 4.11 illustrates again the motivation for the joined-two-filter smoothing. The channel $h = [0.4, 0.05, -0.2, -0.8]$ is problematic for the fixed-interval smoothing since the first channel component is small compared to the other ones. Contrarily, the probable states are easily identified by a backward information filter, which benefits from this channel structure. As could be expected, the joined-two-filter smoothing is in between these two and significantly superior to the fixed-interval smoothing, since half of the particle positions are identified in a backward information filter.

The algorithms were as well tested in the 64-QAM model with random real channels of

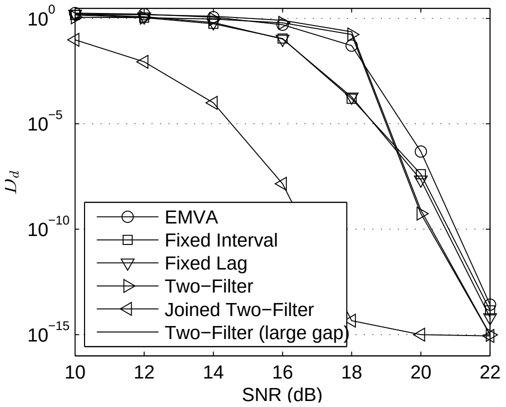

Figure 4.8: Accuracy of approximated smoothing distributions (D_d: $L1$-distance to exact smoothing distribution), $h = [-0.3, 0.05, 0.8]$, 16-QAM, $N = 50$, $L = 3$

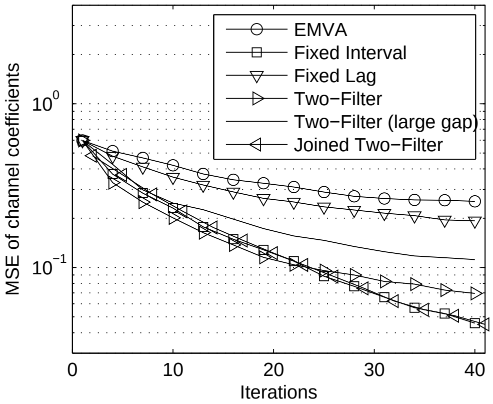

Figure 4.9: MSE of the channel coefficients, 16-QAM, at SNR 15dB and $N = 50$, random channel coefficients, 1 random initialization.

order $L = 4$ by using the BCA method. Fig. 4.12 shows that the joined two-filter smoothing remains superior to the other algorithms. The same experiment carried out with four random initializations yielded slightly inferior results for each algorithm, indicating that in this model the gain of the BCA method is less significant. We present as well the convergence of the Particle Smoothing algorithms employing the $L2$-optimal selection in Fig. 4.13. It shows that the algorithms strongly benefit from this random selection method except for the two-filter smoothing, which suffers massively from the degeneracy due to the large state space. The two-filter smoothing with a large gap is thus more powerful, although its complexity $\mathcal{O}(N^2L^2)$ is already significantly higher in this model. We note that the joined-two filter suffers least from employing the best-weights selection.

Finally, we evaluate the symbol error rate (SER) of the smoothing algorithms in comparison to the Block CMA by Godard [1980], Johnson et al. [1998]. We use a realistic multi-path channel model. To do so, the emitted symbols are generated at a certain symbol period and modulated onto a certain function (e.g. a Square-Root-Nyquist function) and then transmitted over a channel with multiple paths, where each path delays the symbols for a certain time and fades them by a certain amount. The signal is perturbed by white noise and sampled at the symbol period T at the receiver. The channel impulse response is then estimated by the approximate EM algorithms. We consider a 16-QAM model with a channel of three paths with random

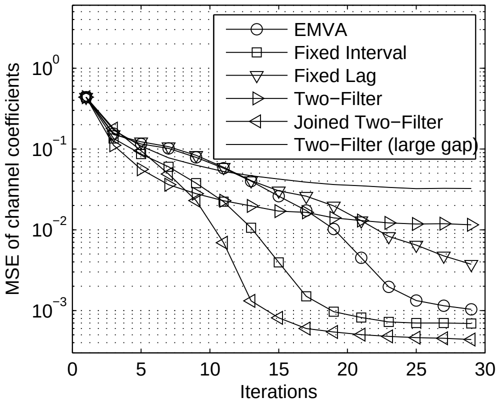

Figure 4.10: MSE of the channel coefficients, 16-QAM, at SNR 15dB and $N = 50$, random channel coefficients, BCA initialization method.

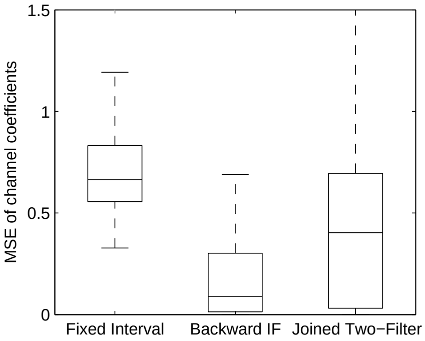

Figure 4.11: MSE after 60 iterations, 64-QAM, $h = [0.4, 0.05, -0.2, -0.8]$.

attenuations and delays $(0, 0.9T, 2.1T)$.

We use the acquisition probabilities as defined in Nguyen and Levy [2005] to measure the convergence of the channel estimation. The MSE is compared to a certain threshold, which has been set equal to σ by Nguyen and Levy [2005]. The acquisition probabilities for a single random initialization are shown in Table 4.1. For the cases, where the channel estimation has converged, we estimated as well the SER given in Fig. 4.14. It shows that all smoothing algorithms are well above the Block CMA, aside from the two-filter smoothing which suffers again from its degeneracy. With one initialization, the fixed-interval smoothing and the last two two-filter smoothing algorithms are superior in terms of the acquisition probability. Similar to Fig. 4.10, the convergence improves by using the BCA method, especially for the EMVA. With the BCA method, the smoothing algorithms are all well above 90%. We show this for the EMVA and the fixed-interval smoothing in Table 4.2. Additional simulations in less complex models like the QPSK modulation, not included here, indicate that even a single random initialization of the approximate EM algorithms is sufficient to estimate the channel (and achieve low SER) for all presented smoothing algorithms, including the EMVA.

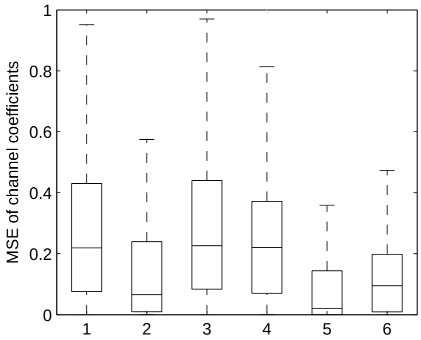

Figure 4.12: MSE after 50 iterations, 64-QAM, $L = 4$, $K = 500$, 4 initializations with BCA method, Best-Weights selection, (1) EMVA, (2) Fixed-Interval, (3) Fixed-Lag, (4) Two-Filter, (5) Joined-Two-Filter, (6) Two-Filter (Large Gap).

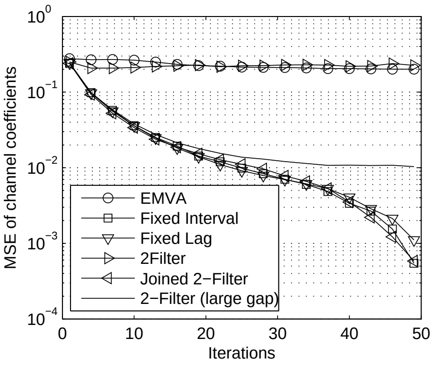

Figure 4.13: MSE of channel coefficients, 64-QAM, $L = 4$, $K = 500$, 4 initializations with BCA method, $L2$-optimal selection.

4.7.2 General Framework and Improving the EMVA

For data recovery with the channel assumed known, we compare the approximate Viterbi algorithms (AVA) - the M-algorithm when using the Best-Weights Selection and the T-algorithm for the Threshold-Comparison Selection - the marginal Particle-Filtering Algorithm (mPFA) and a joined Particle-Filtering Algorithm (jPFA) as well as the Fixed-Interval Smoothing (FIS) and a Block CMA by Godard [1980] in terms of the symbol error rate (SER), see Section 3.9. The symbols are sent over a time-constant, frequency-selective channel consisting of three paths with delays [$0T$, $0.9T$, $3.2T$], where T is the symbol period. The attenuations of each path were randomly chosen. A channel order $L = 5$ was sufficient to cover the relevant peaks in the corresponding channel impulse response. We fixed the length of the observations to $K = 2000$.

Fig 4.15 shows the performance of the different algorithms coupled with the Best-Weights Selection for the 16-QAM. The marginal PFA is not able to recover the symbols, since the marginal filtering distributions only involve information on the past observations but not on the future. However, the marginal smoothing algorithm, which is based on these marginal filtering distributions plus a backward weight correction, yields - together with the AVA - the lowest SER. The joined PFA is slightly inferior.

Table 4.3 presents the percentages of Monte Carlo runs for which the final MSE was smaller than 0.01. The SNR is 20dB, while the number of particles varies ($N = 100, 250$). The Threshold-Comparison Selection appears to be slightly inferior to the other selection schemes.

Algorithm	**SNR**			
	$12dB$	$15dB$	$18dB$	$20dB$
EMVA	30.0%	31.2%	30.2%	22.8%
Fixed-Interval Smoothing	78.4%	77.2%	72.4%	77.0%
Fixed-Lag Smoothing	52.8%	52.2%	41.6%	38.2%
Two-Filter Smoothing	73.8%	68.0%	63.8%	55.8%
Joined Two-Filter Smoothing	66.6%	77.6%	83.2%	73.2%
Two-Filter Smoothing (large gap)	86.2%	80.0%	73.2%	72.4%

Table 4.1: Acquisition probabilities in a 16-QAM, 1 random initialization.

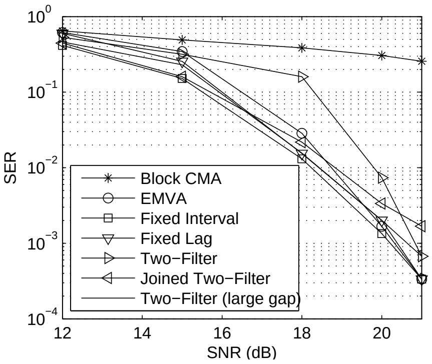

Figure 4.14: Symbol error rate of smoothing algorithms, 16-QAM.

Apart from the marginal PFA (mPFA) the algorithms are almost equivalent.

We now turn to blind identification of the unknown channel and evaluate the mean-squared error (MSE) performance of the EMVA (with Beam Search) compared to the EMVA where the Beam Search has been replaced by an AVA with a random selection scheme. The EMVA algorithm will be explained in Section 5.2.1, but since this comparison on the different selection schemes, the computational results will be given here.

The initial channel estimates are chosen according to the BCA initialization method by Nguyen and Levy [2005]. We simulate the data directly from the model in Section 3 with $L = 3$ and $K = 300$. For each Monte-Carlo run, the channel coefficients are drawn randomly from a uniform distribution. Fig. 4.16 shows the median over 500 Monte-Carlo runs of the MSE of the channel coefficients from the first iteration up to 50 iterations for a 16-QAM model with channel order $L = 3$ and an SNR of 16dB.

The difference between the $L2$- and the Chi-Square Optimal Selection is due to the fact that in the $L2$ Optimal Selection all the weights $\leq \lambda$ are resampled and assigned the same new weight λ. Especially during the first iterations, the filtering distributions are flat, i.e. often very few weights are larger than λ. Hence, the remaining particles often have the same weight. In the update to the next time step in (4.25), the maximization is then over a set of equal weights. The Chi-Square Optimal Selection is better adapted, since it assigns a weight proportional to their square root. These are hence never equal.

The Chi-Square Selection is superior to the original EMVA (with Best-Weights Selection) since during the EM, the states with high smoothing probabilities often have small filtering probabilities. In contrast to the random selection schemes the Best-Weights Selection will never select these states. Fig. 4.17 shows that this effect gets even more important in a 64-QAM model. Both random schemes become clearly superior now.

Algorithm	**SNR**				
	$12dB$	$14dB$	$16dB$	$18dB$	$20dB$
EMVA	91.6%	94.2%	93.2%	93.8%	94.4%
Fixed-Interval Smoothing	96.0%	97.6%	97.6%	98.4%	97.8%

Table 4.2: Acquisition probabilities in a 16-QAM, BCA initialization

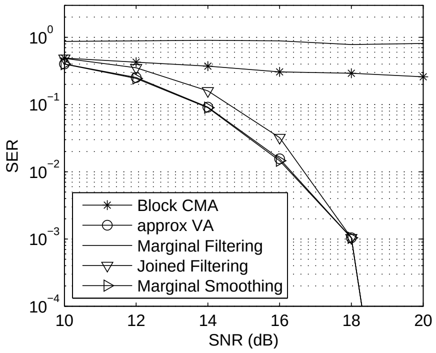

Figure 4.15: Symbol error rate, 16-QAM with $N = 250$ particles and Best-Weights Selection.

Alg.	**Selection**			
	Best W.	Threshold	L2	Chi Sq.
	$N = 100$			
AVA	87.47%	85.76%	88.21%	86.90%
mPFA	0.08%	0.08%	0.08%	0.08%
jPFA	85.76%	68.98%	85.10%	85.18%
FIS	87.47%	85.84%	86.25%	85.51%
	$N = 250$			
AVA	96.8%	87.8%	96.0%	95.8%
mPFA	0.6%	0.6%	0.6%	0.8%
jPFA	94.8%	75.0%	96.8%	95.8%
FIS	94.6%	88.6%	95.0%	95.2%

Table 4.3: Probability of $SER < 0.01$, SNR 20dB, 16-QAM, for different particle sizes N.

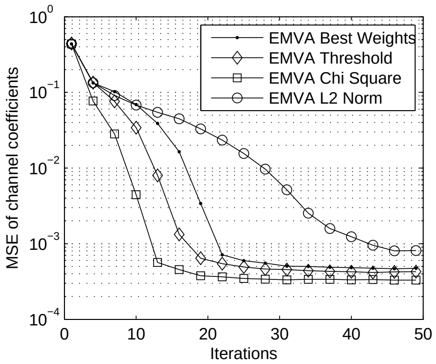

Figure 4.16: Convergence of EMVA with different selection schemes, 16-QAM, $L = 3$, 50 particles, SNR = 16dB

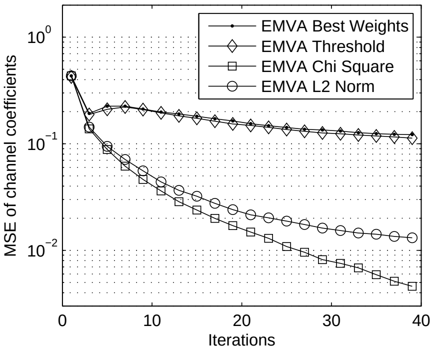

Figure 4.17: Convergence of EMVA with different selection schemes, 64-QAM, $L = 3$, 100 particles, SNR = 22dB

Chapter 5

Blind Identification

Coming back to Alice, we are now in the situation that Alice's language is known, but the echoes as well as the noise are unknown. They have to be estimated before recovering what she said.

In digital communications, this corresponds to the estimation of the unknown channel parameters as well as the unknown noise level. It is referred to as (channel) Identification or Deconvolution. Since the transmitted signals are not known at the receiver, the problem is called blind. Blind Identification is a vast research area in the communications society and we will give a very brief overview of some of the various approaches in Section 5.1.

For several reasons, we concentrate on the concept of Maximum Likelihood (ML) channel estimation. ML estimation is a search over the complete parameter space for the parameter value that maximizes the likelihood of the received signal sequence, i.e. the parameter value for which the least noise is necessary to explain the observations. A formal definition of the likelihood is given in Section 5.2. The most important reason is the fact that the overlying Blind Classification in Chapter 6 is already based on the ML estimation. It requires the ML estimate of the unknown parameters in each potential model as well as the likelihood of the received signal given this ML estimate. Furthermore, the considered signal sequences are relatively short and ML estimation also works very well on short sequences even if it is computationally heavy.

Due to the blindness, the ML estimate of the unknown channel parameters is unfortunately not easily derived. An analytic solution does not exist in contrast to the non-blind case. Hence, an algorithm is required that approximates the ML estimate. There are thus two sources of error in the ML estimation process. The first error type is the difference between the true ML estimate and the true parameter value. The second type is the difference between the estimate of the ML estimate and the true ML estimate.

The algorithm of our choice is the Expectation-Maximization (EM) algorithm first introduced by Dempster et al. [1977]. A formal description of the EM algorithm is given in Section 5.2. Since it is not possible to directly maximize the likelihood function, the main intuitive idea of this algorithm is to smooth the likelihood function such that it may be easily maximized. We recall that the unknown parameter vector is given by $\theta = (x, \sigma)$. Let some parameter estimate θ' of θ be given. The likelihood function is then smoothed with respect to the posterior distribution $p(s_k|Y;\theta')$ of the signals given the parameter estimate θ'. The smoothed version is called the intermediate quantity (with respect to θ'), which may be easily maximized, at least for the model as given in Chapter 3. Let us denote the value maximizing the intermediate quantity by θ''. The reason to smooth the likelihood function in this manner is that it can be easily shown, that the likelihood of the received signal is higher with respect to the new estimate θ'' compared to the 'old' estimate θ'. An iterative sequence of smoothing the likelihood, maximizing the smoothed version and then smoothing with respect to the new estimate and so on will thus yield a sequence of parameter estimates with increasing likelihood. It may furthermore be shown that this sequence will converge to a critical point of the likelihood function. If we are lucky it converges to the global maximum. It is therefore crucial to choose an initial parameter estimate that is not too far from the global maximum such that the sequence of estimates does not converge to a local but not to a global maximum. Since in Blind Identification, we consider cases where no knowledge at all of the channel is at hand, we may thus not ensure convergence to the global maximum. To increase the chances to attain the global maximum, the EM algorithm has therefore to be started with a couple of different initial parameter estimates, such that at least one of the corresponding sequences of parameter estimates will possibly converge to the global maximum.

In practice, the EM algorithm has obviously to be stopped after a finite number of iterations, such that the exact limit of the parameter sequence is not attained. However in general, the EM algorithm converges relatively quickly, such that after at most 50 to 100 iterations (for this application) the parameter estimate is not changing much anymore. The error of estimating the ML estimate via the EM algorithm thus consists of two sources. On the one hand, there is the risk of not converging to the global maximum, on the other hand there is the error of stopping the EM algorithm after a finite number of iterations. In practice, that means there is a small risk to be way off the correct ML estimate and a large risk of a very small error. Therefore, the estimation errors that occurred in practice are always a cumulation of these different types of errors.

In general, the expectation step is not easily implemented. Therefore, we will present the model in the most general form, but we will then rather concentrate on a number of examples for which an implementation of the expectation step is known or will be derived in this contribution (c.f Chapter 4).

For example, we may assume that the unknown input sequence is a Markov Chain with a finite state space, such that (3.4) becomes a Hidden Markov Model (HMM). The noise is assumed to be independently normally distributed. This setting corresponds to a blind deconvolution problem in digital communications. The expectation step is then efficiently implemented by the Baum-Welch algorithm introduced by Baum et al. [1970]. If the state space of the Markov chain is too large, the Baum-Welch algorithm is no longer applicable. In this case, we propose to use Particle Smoothing to reduce the complexity. Particle Smoothing algorithms approximate the smoothing distribution, i.e. the distribution of a symbol given all the observations up to time K. Most of them are based on a Particle Filter approximating the filtering distribution of s_k, i.e. given the observations up to time k. For more details we refer to Chapter 4 that describes the subsequent level in the estimation chain.

We propose as well a novel approach that combines the EM algorithm with ideas from Compressed Sensing. The field of Compressed Sensing has evolved in the recent years to tackle under-determined, but sparse linear systems. A vector x of dimension m is called sparse if the number of active components $\|x\|_0 = r \ll m$, i.e. the number of components different from 0 is small compared to m. A Compressed Sensing (CS) problem can then be written as

$$\min \|x\|_0 \quad \text{subject to} \quad Y = Ax + \varepsilon \tag{5.1}$$

where Y is the (known) observation of dimension K, also called the measurements of x. A is the (known) sensing or measurement matrix and ε is some Gaussian noise vector, see for example Candes and Tao [2005, 2006].

However, in Compressed Sensing the measurement matrix A is assumed to be known. We will generalize this to a blind setting where it can be decomposed into a product of two matrices where only one matrix is known. The Blind Compressed Sensing problem (BCS) is then given by

$$\min \|x\|_0 \quad \text{subject to} \quad Y = S\Psi x + \varepsilon, \tag{5.2}$$

where only the matrix Ψ is assumed known. We note that this form is quite general. For example, the matrix S may be considered as the measurement matrix and Ψ as a basis transformation. In that case, the true parameter vector $\mathbf{b} = \Psi x$ would not be sparse, and the original problem would be written as $Y = S\mathbf{b} + \varepsilon$. On the other hand, if Ψ is the identity matrix, then the problem is completely blind. The problem formulation differs however significantly from the

CS Problem as in (5.1). We consider S to be an unknown random variable with a known distribution. Therefore, it is obvious that in the BCS problem as in (5.2) more measurements are necessary to recover x as compared to (5.1). The focus of the CS problem is rather on how one may use the sparsity to reduce the necessary number of measurements while still being able to recover x. Whereas in the BCS problem, the focus is on how one may use the sparsity to improve the estimate of x given the measurements Y. To solve this problem we propose what we call the Expectation and Sparse Maximization (ESpaM) algorithm.

We will use again the Maximum-Likelihood (ML) estimation of the unknown vector x. Without the sparsity constraint, the Expectation-Maximization (EM) algorithm is as explained before an efficient method to derive the ML estimate.

However, it turns out that in general in Model (5.2) the solution of the maximization step of the EM algorithm is no longer unique, i.e. the set of solutions forms a subspace of positive dimension. In the ESpaM algorithm, we propose to perform a sparse signal reconstruction step using methods such as Matching Pursuit (MP), Orthogonal Matching Pursuit (OMP) or L1-regularization to choose the sparsest element of this subspace as the new parameter estimate of x for the EM iteration.

The ESpaM algorithm is applied to three concrete scenarios. We apply it to Blind Identification of a linear modulation scheme in a frequency-selective channel as in Section 3.4. We assume the finite impulse response of the channel to be sparse. Sparse frequency-selective channels occurring in underwater communications, residential ultrawideband channels and digital television channels amongst others have been considered for example by Bajwa et al. [2008a], Cotter and Rao [2002]. In that case, the known matrix Ψ is the identity matrix such that the standard EM algorithm is as well applicable. We show that if the parameter vector is sufficiently sparse, the ESpaM algorithm is clearly superior. The second application is a transmission of a linear modulation scheme over a doubly-selective channel as given in Section 3.5. The sparsity arises from the fine grid that we introduce on the parameter space. Thus, the ESpaM algorithm is no longer directly estimating the finite impulse response of the channel, but rather the coefficients corresponding to the basis or grid points that we have chosen. The third example is an OFDM transmission over a doubly-selective channel as presented in Section 3.6. As in the second example, the sparsity arises from the grid on the space of Doppler frequencies and delays. If we assume that only a few paths of the transmission channel are relevant, then there will only be very few Doppler frequencies and delays where the corresponding coefficient will be different from zero. For this application, we will also present a semiblind version of the ESpaM algorithm, that makes additional use of a small number of (known) pilot symbols that are introduced into the transmitted sequence.

The rest of Chapter 5 is organized as follows. We will first give an overview of the state of the art in Blind Identification as well as in Compressed Sensing. Then we formalize ML estimation and describe the EM algorithm. We will then introduce the ESpaM algorithm. We will discuss several implementations of the sparsity constraint into the EM algorithm to justify our approach to the ESpaM algorithm. A further justification will be given in the following Section 5.3.2 with theoretical results showing that the ESpaM algorithm converges and that the true parameter value is a fixed point of the ESpaM algorithm under some conditions on the sparsity and regularity of the model. We present as well a semiblind version of the ESpaM algorithm and apply it to the OFDM transmission. We will conclude with Monte-Carlo experiments for the ESpaM algorithm. The performance results on the standard EM algorithm may be found in Chapter 4 in Section 4.7 where we concentrate more on different solutions to implement the expectation step via various smoothing algorithms.

5.1 State of the Art

For over 30 years, Blind Identification has received plenty of attention in the signal processing community. Many of the earlier approaches are based on order statistics. Xu et al. [1995] and Wang et al. [2001] use for example second order statistics of the received symbol. Even earlier, Godard [1980] developed the Constant Modulus Algorithm (CMA) which is based on higher order statistics. For a long time, it has been considered as the state of the art or at least as a good benchmark for new algorithms. A more recent review of the CMA is Johnson et al. [1998]. Other work on high order statistics includes Giannakis and Mendel [1989], Porat and Friedlander [1991] and Tugnait [1987]. Another approach is to use the cyclostationarity of the signal, see for example Giannakis [1998] and Slock [1994].

The Kalman Filter by Kalman and Bucy [1961] is a filtering approach for linear Gaussian models. It has been extensively used for all kind of estimation purposes in digital communications. A good reference is Anderson and Moore [2005]. Exemplarily for the application in digital communications we cite Komninakis et al. [1999] for the estimation of frequency selective channels. In the presented model, Kalman filtering is not applicable since the state space is finite and the transition of the states is thus obviously not Gaussian.

Blind Identification of time-varying channels is a considerably more difficult task. If a Markov model for the channel coefficients is reasonable, the Kalman filter is the optimal method if the model is linear and Gaussian. Otherwise, Particle Filtering is a very powerful alternative and has been applied numerously for this problem, see 4.1 for more details.

One approach to circumvent the time dependence of the channel are Basis Expansion Models (BEM) first looked at in detail by Giannakis and Tepedelenlioglu [1998]. These methods introduce time-varying basis functions on the parameter space such that the time-varying channel may be written as a linear combination of these basis functions. The coefficients of the basis functions are thus constant in time over a block of signals. BEM methods for identification of doubly-selective channels have for example been considered by Leus and Moonen [2003] and Leus [2005]. Recently, Tugnait et al. [2010] proposed a method of tracking these basis coefficients over several blocks of signals. Another recent work on MLSE and MAP equalization using BEM is by Barhumi and Moonen [2009].

A lot of work has been devoted to the problem of fast changing channels in OFDM systems, see e.g. Schniter [2004]. Very often Blind Identification has been considered when only one Doppler shift and one delay are present, see for example van de Beek et al. [1997]. The thesis by Hijazi [2008] is a recent and comprehensive work on OFDM estimation presenting different approaches. On the one hand, Hijazi and Ros [2010] uses a Kalman filter. A polynomial estimation technique is used in Hijazi and Ros [2009].

A common approach to parameter estimation is Maximum Likelihood (ML) estimation. In the proposed model, the ML estimate of the parameters is however not analytically available. In hidden Markov models with Gaussian noise, the Expectation-Maximization algorithm by Dempster et al. [1977] is a well-studied iterative method that allows to approach the ML estimate. For an introduction, we refer to the tutorials by Bilmes [1997] and Rabiner [1989]. A thorough review on the algorithm and its theoretical properties is given in Cappé et al. [2007].

In digital communications, the EM algorithm has also as well been applied very often for Blind Identification, see for example Georghiades and Han [1997], Anton-Haro et al. [1997] and Kaleh and Vallet [1994]. Al-Naffouri et al. [2002] used an EM approach for parameter estimation in an OFDM transmission. Very recently, Hwang and Schniter [2009] combined the EM algorithm with basis expansion for estimation on doubly selective channels.

Furthermore, Herzet et al. [2007] propose an EM algorithm for blind synchronization in digital communications. They derive a sum product algorithm based on factor graphs to reduce the complexity of the EM algorithm. This improvement is however specific to the model and therefore not applicable to this work. Furthermore, they consider an Additive Gaussian White

Noise (AWGN) channel model which is considerably simpler than the frequency-selective channel models used here. Bayesian Maximum Likelihood has also been considered by Roufarshbaf and Nelson [2008] and Nelson and Singer [2006]. The algorithm by Ben Salem and Salut [2004] is another iterative ML method combining basis expansion, Kalman filtering and Particle Filtering. ML estimation may also be implemented with the help of the Viterbi algorithm by Viterbi [1967]. The Expectation-Maximization Viterbi algorithm (EMVA) by Nguyen and Levy [2003] is an iterative method that is very close to the standard EM algorithm. We will describe it in more detail in 5.2.1. The estimate obtained by using the Viterbi algorithm inside the EM algorithm is biased. An approach to remove the bias is the Adjusted Viterbi Training by Lember and Koloydenko [2007]. A second approach is to use a priority-driven search algorithm by Druck et al. [2007] using underestimates of the true cost function. The algorithm is more efficient than the Viterbi algorithm and is almost optimal most of the time. Seshadri [1994] proposed a suboptimal Viterbi trellis search algorithm that updates the channel estimates inside the Viterbi decoding after each time step.

The field of Compressed Sensing has evolved in the recent years to tackle under-determined, but sparse linear systems, see for example the groundbreaking works by Candes and Tao [2005, 2006]. Many methods to solve this problem have been proposed, for example the Matching Pursuit (MP) by Mallat and Zhang [1993], the Orthogonal Matching Pursuit (OMP) by Pati et al. [1993] and Tropp2004, or a minimization with respect to the L1-norm as in Tropp [2005] or the Basis Pursuit in Chen et al. [1998] or more recently in Pfander and Rauhut [to appear] or the Lasso as in Tibshirani [1996], Donoho [2006], Candes and Tao [2006].

Recently, Compressed Sensing has been applied to channel identification in a communications setting. In many cases introducing an appropriate basis will make this non-linear parameter-estimation problem a linear sparse one. One of the first applications has been by Fuchs [1997] to cater for an unknown number of channel paths. Bajwa et al. [2010] give a comprehensive overview of recent advances in what they call Compressed Channel Sensing. Bajwa et al. [2008a,b] as well as Tauböck et al. [2010] have proposed methods for more general channels, including doubly-selective channels, see Bajwa et al. [2008c]. Rapidly time-varying sparse channels have been covered by Lui and Borah [2003], Li and Preisig [2007]. Even if the channel is not sparse, a benefit from sparse methods is possible by introducing an over-complete basis to improve the estimation accuracy as demonstrated by Sharp and Scaglione [2008]. OFDM channels have for example been considered in Tauböck and Hlawatsch [2008]. Since all of these methods only consider training based identification and therefore essentially assume the measurement matrix to be known, we refer to Bajwa et al. [2010], Taubock et al. [2010] for a more thorough overview of Compressed Channel Sensing. The focus for training based methods is furthermore not only on the reconstruction or estimation of the channel, but also on the sensing or the design of the training sequence. Since we consider Blind Identification, we do not have influence on the design and concentrate thus solely on the reconstruction of the channel.

Compressed Sensing has also been cast in a Bayesian framework by Tipping [2001], Tzagkarakis and Tsakalides [2010], where an artificial prior distribution is introduced on the unknown parameters. The problem may then be solved by the Relevance Vector Machine by Tipping [2001]. The different proposed methods differ mainly in the (artificial) choice of the prior.

Asif et al. [2009] published one of the first and only results on a blind Compressive Sensing approach to deconvolution in communications. However, the method they propose is for the case without noise and of a randomly precoded signal where each symbol is drawn from $\mathbb{R}$. Their algorithm is an optimization procedure over the joint space of the signal and the channel impulse response. This space is however not convex if the symbol alphabet is finite and therefore their method is not applicable in the two applications we consider here.

5.2 ML Estimation and the Expectation-Maximization Algorithm

Before we formalize the likelihood and the EM algorithm, we introduce some further terminology and fix some requirements. The received observation vector Y is considered to be given and fixed. In the terminology introduced by Dempster et al. [1977] Y is referred to as the *incomplete data*, while the couple of the unknown S and Y is called *complete data*. Furthermore, since we are treating the Blind Identification problem, the model or modulation is fixed as well, i.e. the state space $\mathcal{S}$ of the hidden Markov chain is considered known. We will consider the (known) parameters α of the matrix $\Psi(\alpha)$ fixed and write $\Psi = \Psi(\alpha)$ as a short form.

The *complete data likelihood* is defined as the joint probability density $f(S, Y; \theta)$ as given in (3.9) and instead of viewing f as a function of S and Y, the complete data likelihood is to be seen as a function of θ while S and Y are fixed. Since S is not observable, the complete data likelihood is not available. We define therefore the *incomplete data likelihood* which is the marginal of the complete data likelihood by integrating over the space of the symbol matrices.

Definition 2 *The incomplete data likelihood L is the function*

$$L : \Theta \to \mathbb{R}$$

that is given for $\theta \in \Theta$ by

$$L(\theta) = \int_{\mathscr{S}} f(S, Y; \theta) d\mathbb{P}(S). \tag{5.3}$$

We recall that $\mathscr{S}$ is the state space of S.

The maximum likelihood (ML) estimate of θ is hence given by

$$\hat{\theta} = \arg\max_{\theta \in \Theta} L(\theta). \tag{5.4}$$

Since f is a probability density, the likelihood $L(\theta)$ is positive and we may as well consider and maximize the *log-likelihood* which is defined as

$$l(\theta) = \log L(\theta). \tag{5.5}$$

Since $\hat{\theta}$ may not be derived analytically in general, we have to resort to an iterative procedure to derive an estimate of it. The Expectation-Maximization (EM) algorithm by Dempster et al. [1977] is a well known method for ML estimation in incomplete data models with noise stemming from the exponential family. Instead of maximizing the likelihood function directly the EM algorithm maximizes the intermediate quantity in each iteration step.

Definition 3 *The intermediate quantity $Q(\cdot, \theta')$ is a function on Θ indexed by θ' which is defined for $\theta \in \Theta$ by*

$$\begin{aligned} Q(\theta, \theta') &= \mathbb{E}_S \left[\log\left(f(S, Y; \theta)\right) | Y; \theta'\right] \\ &= \int_{\mathscr{S}} \log\left(f(S, Y; \theta)\right) p(S|Y; \theta') d\mathbb{P}(S). \end{aligned} \tag{5.6}$$

By $\mathbb{E}_S[\cdot|Y; \theta']$ we denote expectation with respect to S conditional on Y and given the parameter value θ'. Hence, the intermediate quantity is the logarithm of the joint density smoothed with respect to the conditional distribution of S given the parameter value θ'. In other words, since it is not possible to integrate the joint density f directly with respect to $\mathbb{P}$, we integrate it with respect to some other measure, namely $p(\cdot|Y; \theta')d\mathbb{P}(S)$ that simplifies enormously the integration as we will show subsequently.

If $\theta = (x, \sigma)$ and S are given sequentially as the Markov chain $(s_k)_k$ as described in (3.4) and if additionally the noise is Gaussian then (5.6) simplifies to

$$\begin{aligned} Q(\theta, \theta') &= \int_{\mathcal{S}} \log \left(\prod_{k=1}^{K} \frac{1}{\pi\sigma^2} e^{-\frac{1}{\sigma^2}\|s_k\Psi_k x - y_k\|^2} \prod_{k=2}^{K} q\left(s_k \left| s_{k-1}\right.\right) \right) p_k(s_k|Y;\theta') d\mathbb{P}(s_k) \\ &= -\frac{1}{\sigma^2} \sum_{k=1}^{K} \int_{\mathcal{S}} \|s_k\Psi_k x - y_k\|^2 p_k(s_k|Y;\theta') d\mathbb{P}(s_k) - K\log(\sigma^2) + \text{const}, \end{aligned} \qquad (5.7)$$

where the constant term does not depend on the unknown parameter vector $\theta = (x, \sigma)$ since the transition probabilities $q\left(\cdot\left|\cdot\right.\right)$ are known.

Although in the presented examples the quantities in this section are well defined, this is not the case in general. Therefore, we need the following assumptions that guarantee that the subsequent proposition holds true.

Assumption 3

1. *The parameter space Θ is an open subset of $\mathbb{R}^{Q+1}$.*
2. *For any $\theta \in \Theta$, $L(\theta)$ is positive and finite.*
3. *For any $(\theta, \theta') \in \Theta \times \Theta$, $\int |\nabla_\theta \log p(S|Y;\theta)| p(S|Y;\theta') d\mathbb{P}(S)$ is finite.*

By ∇_θ we denote the gradient with respect to θ.

The following proposition is the key property of the intermediate quantity that justifies the EM algorithm.

Proposition 1 *Under Assumption 3, for any $(\theta, \theta') \in \Theta \times \Theta$,*

$$l(\theta) - l\left(\theta'\right) \geq Q(\theta, \theta') - Q(\theta', \theta').$$

The proof is not difficult. It is a direct application of Jensen's inequality and may for example be found in Cappé et al. [2007]. In other words, an increase in the intermediate quantity, increases as well the likelihood by at least as much.

We are now able to formulate the EM algorithm. It alternately carries out the expectation (E) step and the maximization (M) step. The E step calculates the intermediate quantity for a new parameter estimate, while the M step updates the parameter estimate by maximizing the intermediate quantity.

After defining an initial guess of the parameter $\hat{\theta}_0$, the EM algorithm then consists of the two iterative steps

1. **Expectation:** Calculate $Q(\theta, \hat{\theta}_i)$.
2. **Maximization:** Calculate $\hat{\theta}_{i+1} = \arg\max_\theta Q(\theta, \hat{\theta}_i)$.

The EM algorithm is known to converge to a critical point of the likelihood function, see for example Cappé et al. [2007]. If the likelihood function has several local maxima, then the EM algorithm has to be set up with several different initial parameter values to increase the probability to converge to the global maximum. In certain cases as the time-invariant channel model in Section 3.4, the convergence to the global maximum can be ensured by a certain choice of initial values, see Section 5.2.1 and Nguyen and Levy [2005].

In contrast to the likelihood function, the intermediate quantity $Q\left(\theta, \hat{\theta}_i\right)$ can be maximized more easily with respect to θ and thus the M step of the EM algorithm is feasibly implemented.

The solutions are given by setting the partial derivatives of Q with respect to x and σ equal to zero:

$$\begin{aligned} \nabla_x Q\Big((x,\sigma),\hat{\theta}_i\Big) &= 0 && (5.8)\\ \nabla_{\sigma^2} Q\Big((x,\sigma),\hat{\theta}_i\Big) &= 0 && (5.9) \end{aligned}$$

The best channel estimate is thus given as the solution of the following system of linear equations:

$$\mathrm{E}_{\mathrm{sy}}\big(\hat{\theta}_i\big) = \mathrm{E}_{\mathrm{ss}}\big(\hat{\theta}_i\big)\hat{x}_{i+1}, \tag{5.10}$$

where

$$\begin{aligned} \mathrm{E}_{\mathrm{sy}}\big(\hat{\theta}_i\big) &= \mathbb{E}_S\left[(S\Psi)^H Y \middle| Y;\hat{\theta}_i\right]\\ &= \int_{\mathscr{S}} \Big((S\Psi)^H Y\Big) p\left(S \middle| Y;\hat{\theta}_i\right) d\mathbb{P}(S),\\ \mathrm{E}_{\mathrm{ss}}\big(\hat{\theta}_i\big) &= \mathbb{E}_S\left[(S\Psi)^H S\Psi \middle| Y;\hat{\theta}_i\right]\\ &= \int_{\mathscr{S}} \Big((S\Psi)^H S\Psi\Big) p\left(S \middle| Y;\hat{\theta}_i\right) d\mathbb{P}(S). \end{aligned}$$

We denote by ^{H} the Hermitian of a complex matrix.

If the model is a HMM according to (3.4), then these quantities write

$$\begin{aligned} \mathrm{E}_{\mathrm{sy}}\big(\hat{\theta}_i\big) &= \sum_{k=1}^K y_k \int_{\mathcal{S}} (s_k\Psi_k)^H p_k\left(s_k \middle| Y;\hat{\theta}_i\right) d\mathbb{P}(s_k),\\ \mathrm{E}_{\mathrm{ss}}\big(\hat{\theta}_i\big) &= \sum_{k=1}^K \int_{\mathcal{S}} (s_k\Psi_k)^H (s_k\Psi_k) p_k\left(s_k \middle| Y;\hat{\theta}_i\right) d\mathbb{P}(s_k). \end{aligned}$$

Luckily, (5.10) is independent of σ, such that the estimate of x may be updated independently of σ.

If the intermediate quantity is given as in (5.7), then the partial derivative with respect to σ is given by

$$\nabla_{\sigma^2} Q\Big((x,\sigma),\hat{\theta}_i\Big) = \frac{1}{(\sigma^2)^2}\sum_{k=1}^K \int_{\mathcal{S}} \|s_k\Psi_k x - y_k\|^2 p_k\left(s_k \middle| Y;\hat{\theta}_i\right) d\mathbb{P}(s_k) - K\frac{1}{\sigma^2} \tag{5.11}$$

and hence equating to 0 gives

$$\begin{aligned} \hat{\sigma}^2_{i+1} &= \frac{1}{K}\sum_{k=1}^K \int_{\mathcal{S}} \|s_k\Psi_k\hat{x}_{i+1} - y_k\|^2 p_k\left(s_k \middle| Y;\hat{\theta}_i\right) d\mathbb{P}(s_k)\\ &= \frac{1}{K}\Big((\hat{x}_{i+1})^H \mathrm{E}_{\mathrm{ss}}\big(\hat{\theta}_i\big)\hat{x}_{i+1} - 2(\hat{x}_{i+1})^H \mathrm{E}_{\mathrm{sy}}\big(\hat{\theta}_i\big) + Y^H Y\Big) && (5.12)\\ &= \frac{1}{K}\Big(-(\hat{x}_{i+1})^H \mathrm{E}_{\mathrm{sy}}\big(\hat{\theta}_i\big) + Y^H Y\Big) && (5.13) \end{aligned}$$

The expectation step consists in calculating the marginal smoothing probabilities $p(S|Y;\theta)$ given some parameter estimate θ which may for example be implemented by the Baum-Welch algorithm by Baum et al. [1970] in a HMM with a small finite state space.

5.2.1 Expectation-Maximization Viterbi Algorithm and Viterbi Decoding Techniques

Since we will extensively use the Expectation-Maximization Viterbi Algorithm (EMVA) by Nguyen and Levy [2003] as a comparison in the simulation and we will also show how to improve it, we present it here in more detail. The EMVA is a variant of the standard EM algorithm. The expectation step is replaced by a Viterbi decoding by Viterbi [1967]. The survivor paths of the Viterbi algorithm are then considered as a representative sample of all data sequences with high posterior probability such that the posterior distribution is estimated based on these survivor paths. In Nguyen and Levy [2005], the same algorithm has been applied to linear modulation schemes in frequency-selective channels.

For each x_K at time K in the state space $\mathcal{S}$, let $W_k(x_k)$ be the shortest path to x_K. How to derive them has been described in detail in Section 4.2.2.

Now Nguyen and Levy [2003] consider that the joint smoothing distribution

$$p(\cdot|Y;\theta) = \mathbb{P}(s_{1:K} = \cdot|Y;\theta)$$

of all possible trajectories $x_{1:K} \in \mathcal{S}^K$ may be well approximated by only considering the survivor paths $W_k(x_k)$ of the Viterbi search. The idea is that only very few trajectories will have a considerable posterior probability and these trajectories are often amongst the survivor paths of the Viterbi. Similar to Particle Filtering and Particle Smoothing as in Section 4, $p(\cdot|Y;\theta)$ is approximated by discretizing the state space:

$$\hat{p}(x_{1:K}|Y;\theta) = \sum_{z_K} p(W_K(z_K)|Y;\theta)\delta_{W_K(z_K)}(x_{1:K}), \tag{5.14}$$

where δ is again the index function as in (4.7).

The EMVA implements thus the expectation step of the standard EM algorithm by replacing the exact smoothing distribution by its approximation $\hat{p}_{1:K}(\cdot|Y;\theta)$. The maximization step remains unaltered.

Furthermore, Nguyen and Levy [2005] discuss how to initialize the EM algorithm, i.e. how to choose the initial parameter estimate in the first iteration. This is essential, since a 'bad' initial estimate may cause the algorithm to converge to a local maximum far from the global maximum. Their Blind Channel Acquisition (BCA) method ensures convergence to the global maximum. It consists of several trial runs of the EM algorithm each time with a different initial parameter choice. At each run, the estimate of the unknown parameter x is initialized by setting all components except for two equal to zero. More precisely, the initial estimate has only one nonzero real-valued tap and one nonzero purely imaginary tap. If the channel is initialized with non-zero components located at the proper position, of correct sign and unit magnitude, then the EM algorithm will converge. For an arbitrary complex channel of order L, there are $(2L)^2$ possibilities for such an initializer, which implies a significant increase in the complexity. For a real channel, the imaginary tap is not needed, and the number of possibilities is L (when ignoring the sign). We will use the BCA method in the simulations. However, in praxis we may not advise to use it due to the very large number of necessary initializations. Furthermore, as soon as the smoothing has to be implemented by an approximate method like an approximate Viterbi search or a Particle Smoothing method, convergence using the BCA method is no longer ensured.

5.3 The Expectation Sparse Maximization Algorithm

We now introduce the Expectation Sparse Maximization (ESPaM) algorithm which is a variant of the EM algorithm for the case that the parameter vector x is sparse. For the moment, we only consider the vector x and not the standard deviation σ which is obviously not sparse. Hence, at each iteration the ESpaM algorithm will provide a means to obtain a sparse estimate of x.

For reasons that we explain in Section 5.3.1, the intermediate quantity of the ESpaM algorithm is defined as for the EM algorithm, i.e. without a sparsity constraint. Thus, at each iteration the ESpaM algorithm maximizes the intermediate quantity Q that has been defined in (5.6).

As in (5.10), the M step of the ESpaM algorithm consists in maximizing

$$\mathrm{E}_{\mathrm{sy}}\big(\hat{\theta}_i\big) = \mathrm{E}_{\mathrm{ss}}\big(\hat{\theta}_i\big)x,$$

but now with an additional sparsity constraint. That is, the algorithm is solving one of the following problems:

$$\min_x \|x\|_0 \quad \text{s.t.} \quad \mathrm{E}_{\mathrm{sy}}\left(\hat{\theta}_i\right) = \mathrm{E}_{\mathrm{ss}}\left(\hat{\theta}_i\right)x, \tag{5.15}$$

the Lasso problem as given in Tibshirani [1996], Donoho [2006], Candes and Tao [2006]:

$$\min_x \|x\|_1 + \lambda \left\|\mathrm{E}_{\mathrm{sy}}\left(\hat{\theta}_i\right) - \mathrm{E}_{\mathrm{ss}}\left(\hat{\theta}_i\right)x\right\|^2, \tag{5.16}$$

for $\lambda > 0$ or the Basis pursuit problem as presented in Chen et al. [1998]:

$$\min_x \|x\|_1 \quad \text{s.t.} \quad \mathrm{E}_{\mathrm{sy}}\left(\hat{\theta}_i\right) = \mathrm{E}_{\mathrm{ss}}\left(\hat{\theta}_i\right)x. \tag{5.17}$$

There are several reasons to consider one of the previous sparse problems (5.15), (5.16) or (5.17). If the covariance matrix E_{ss} has full rank, i.e. $\mathrm{rk}\left(\mathrm{E}_{\mathrm{ss}}(\hat{\theta}_i)\right) = Q$, then the solution $\hat{x}_i$ to the maximization problem in (5.10) is unique and given by $\hat{x}_i = \left(\mathrm{E}_{\mathrm{ss}}(\hat{\theta}_i)\right)^{-1}\mathrm{E}_{\mathrm{sy}}\big(\hat{\theta}_i\big)$. Thus, the M step of the EM algorithm is easily implemented.

However, if x is sparse then in many applications the column rank of the symbol matrix S is smaller than the length Q of x and thus E_{ss} does not have full rank. This for example the case in the examples given in Section 3.5 and Section 3.6 where a fine grid is introduced on the state space, and each coefficient of x corresponds to one grid point. However, the column rank of S corresponds to the number of symbols that influence each received signal and this is in practice much smaller than the number of grid points.

The rank of $\mathrm{E}_{\mathrm{ss}}(\hat{\theta}_i)$ is given by $\mathrm{rk}\left(\mathrm{E}_{\mathrm{ss}}(\hat{\theta}_i)\right) = \min(Q, L)$ and the solution to the equation as in (5.10) is hence only unique if $L = Q$ as it is the case in the example of Section 3.4. In general $L \ll Q$ and the solution to (5.10) is a subspace of dimension $Q - L$. We propose thus to use a sparse algorithm to select the sparsest vector in this subspace as the new parameter estimate.

Even if $L = Q$, sparse algorithms may be used to improve the robustness of the EM algorithm, see for example the simulation results in Section 5.4 for the time-invariant model as given in Section 3.4.

Hence, we propose to solve the maximization problem of the intermediate quantity by applying a sparse algorithm like the Matching Pursuit (MP) Mallat and Zhang [1993], the Orthogonal Matching Pursuit (OMP) Pati et al. [1993], Tropp [2004] or a L1-regularization to one of the problems (5.15), (5.16) or (5.17) with the new parameter estimate $\hat{\theta}_{i+1}$.

Since the standard deviation σ is not sparse and even more important the update of the vector x is independent of σ, the parameter update for σ remains the same compared to the EM algorithm. However, most sparse algorithms are not exact, especially the greedy algorithms like MP or OMP, which we recommend to use for reasons that we will explain later on. If the solution is not exact then $\mathrm{E}_{\mathrm{sy}}\big(\hat{\theta}_i\big) \neq \mathrm{E}_{\mathrm{ss}}\big(\hat{\theta}_i\big)\hat{x}_{i+1}$. Hence, the transformation from (5.12) to (5.13) is no longer valid. Thus, the parameter update is given by (5.12).

The Expectation and Sparse Maximization (ESpaM) algorithm on the initial parameter estimate $\hat{\theta}^{(0)}$ is thus given by iterating the following steps:

1. Calculate $p\left(S\,\middle|Y;\hat{\theta}_i\right)$ or in the sequential model: $p_k\left(s_k\,\middle|Y;\hat{\theta}_i\right)$ for $k=1,\ldots,K$.

2. Derive $\mathrm{E}_{\mathrm{sy}}(\hat{\theta}_i)$ and $\mathrm{E}_{\mathrm{ss}}(\hat{\theta}_i)$.

3. Solve $\hat{x}_{i+1} = \mathrm{Sparse}\left(\mathrm{E}_{\mathrm{sy}}(\hat{\theta}_i),\, \mathrm{E}_{\mathrm{ss}}(\hat{\theta}_i)\right)$.

4. Update $\hat{\sigma}^2_{i+1} = \dfrac{1}{K}\left((\hat{x}_{i+1})^H\,\mathrm{E}_{\mathrm{ss}}(\hat{\theta}_i)\hat{x}_{i+1} - 2(\hat{x}_{i+1})^H\,\mathrm{E}_{\mathrm{sy}}(\hat{\theta}_i) + Y^HY\right)$

5. Set $\hat{\theta}_{i+1} = (\hat{x}_{i+1}, \hat{\sigma}^2_{i+1})$

The function $\mathrm{Sparse}(\cdot,\cdot)$ denotes the specific sparse algorithm that takes as input the matrix $\mathrm{E}_{\mathrm{ss}}(\hat{\theta}_i)$ and the vector $\mathrm{E}_{\mathrm{sy}}(\hat{\theta}_i)$ to solve one of the problems (5.15), (5.16) or (5.17).

5.3.1 Discussion on the Sparse ML Estimation

Instead of adding a sparsity constraint to the intermediate quantity as given in (5.7), the sparse EM algorithm described above uses a sparsity constraint for the derivative of the intermediate quantity. But since the necessary condition for a minimum of (5.7) is that the derivative is equal to 0, the sparsest solution to (5.10) will also be the sparsest minimum of (5.7).

Both the MP and the OMP do not guarantee to find exact sparse solutions to (5.10), but since the intermediate quantity is in general sufficiently smooth, a point having a gradient close to zero will also be close to the maximum of the intermediate quantity. Furthermore, it is not necessary to find the exact maximum, since as long as Q increases, the likelihood will also increase and so the sequence of iterative parameter estimates will still converge to a critical point of the likelihood function.

If the symbol matrix S is known, ML estimation of the channel x is equivalent to minimizing

$$\|S\Psi x - Y\|_2^2 = \textstyle\sum_{k=1}^{K} \|s_k\Psi_k x - y_k\|_2^2 \tag{5.18}$$

with respect to x. A solution to this problem under additional sparsity constraints is readily available since sparse algorithms like the MP and OMP directly apply to (5.18). Unfortunately, this may not be generalized to the blind convolution problem. The maximization step of the EM algorithm now consists in minimizing Problem (5.7) with respect to x, such that $\|x\|_0 \leq \kappa$ for some $\kappa < Q$. Observe that, if the symbol sequence is known, the probability distribution in (5.7) becomes a point mass and (5.7) reduces to (5.18).

It is not possible to rewrite (5.7) in matrix form such that the sparse algorithms like the MP and the OMP may be applied. L1-regularization is still applicable. Its implementation is very complex because evaluating the right hand side of (5.7) is very costly. We have however run Monte-Carlo experiments for a moderate channel order L and the numerical results did not show any performance improvement with respect to the sparse EM algorithm presented in Section 5.3. Therefore, we strongly recommend to use the ESpaM algorithm.

A different sparse Expectation-Maximization type algorithm that allows to directly apply the MP and OMP to the maximization criterion may be established by using a slightly different maximization criterion. Instead of maximizing the intermediate quantity, one could maximize

$$\left\|\mathbb{E}_S[S|Y;\theta']\Psi x - Y\right\|_2^2 \tag{5.19}$$

where the conditional expectation of S is given by

$$\mathbb{E}_S[S|Y;\theta'] = \int_{\mathcal{X}^{K\times L}} S\,p\left(S|Y;\theta'\right)d\mathbb{P}(S). \tag{5.20}$$

This is analogous to (5.18), where S is replaced by its expectation based on the current parameter estimate. It is also similar to the Maximization-Maximization algorithm, where the expectation step is replaced by a Viterbi search Viterbi [1967]. Now the sparse minimization methods like MP or OMP are readily applicable. However, in contrast to the EM algorithm convergence is not assured and Monte-Carlo experiments on the time-invariant model showed that the performance is clearly lower compared to the ESpaM algorithm.

5.3.2 Convergence Properties of the ESpaM Algorithm

We now discuss the theoretical convergence properties of the ESpaM algorithm.

Proposition 2 *Let $\left(\hat{\theta}_i\right)_i$ be a sequence of estimates of x obtained by the ESpaM algorithm. Then for every i*

$$L\left(\hat{\theta}_{i+1}\right) \geq L\left(\hat{\theta}_i\right).$$

This proposition directly follows from the fact that the ESpaM algorithm does not alter the principle of the EM algorithm. Indeed, the intermediate quantity is still maximized. The ESpam algorithm only provides a criterion which solution in the subspace of maximal values of the intermediate quantity is preferable, such that Proposition 2 is a direct consequence of Proposition 1.

We have thus established that the ESpaM algorithm will converge (or possibly diverge if the parameter space is not compact). As for the EM algorithm, convergence to the global maximum of the likelihood function is not ensured. In general if the measurement matrix Ψ is quadratic and of full rank, the likelihood function has isolated local maxima, i.e. the maximization of the intermediate quantity is unique. An almost immediate consequence is that the true parameter is a fixpoint of the algorithm. However, this is not obvious if Ψ is not of full rank, since in this case the solutions maximazing the intermediate quantity form a subspace of the parameter space.

We will now give a sufficient condition for which the true parameter x still remains a fixpoint of the ESpaM algorithm.

Assumption 4 *We assume that S is not degenerated such that $\mathbb{E}_x[S^H S]$ has full rank. Since this is the covariance matrix of S we just require that there is no affine relation between two of the columns of S.*

Assumption 5 *We assume that every combination of $2p$ columns of Ψ is linearly independent. p denotes again the number of non-zero coefficients in the true parameter vector x. This requires obviously that $L \geq 2p$.*

Proposition 3 *Let Assumptions 4 and 5 be true. Then the true parameter vector x is a fixpoint of the ESpaM algorithm, i.e. there exists no sparser solution.*

Proof: Let $\theta = (x, \sigma)$. Note, that standard results of the EM algorithm give that the correct parameter value θ maximizes $Q(\cdot, \theta)$, i.e. it is a solution of (5.10) such that

$$\mathrm{E}_{\mathrm{sy}}(\theta) = \mathrm{E}_{\mathrm{ss}}(\theta)\, x. \tag{5.21}$$

To show that x is as well a fixpoint of the proposed ESpaM algorithm, we first show that every combination of $2p$ rows of $\mathrm{E}_{\mathrm{ss}}(\theta)$ is also independent. Secondly we show that x is a fixed point, i.e. that every other solution to (5.15) has more non-zero components than p.

1. We recall that the measurement matrix Ψ is of size $Q \times L$. Let T be a subset of $\{1, \ldots, Q\}$ of size $2p$ and let T^- be the remaining indices. For a matrix A let A_T denote the submatrix of A consisting of the columns with indices corresponding to T. Without loss of generality we assume for the ease of presentation that $T = \{1, \ldots, 2p\}$.

   Then $\mathrm{E}_{\mathrm{ss}}(\theta)$ may be decomposed as

   $$\mathrm{E}_{\mathrm{ss}}(\theta) = \begin{bmatrix} \Psi_T^H \mathbb{E}_x[S^H S] \Psi_T & 0 \\ 0 & \Psi_{T^-}^H \mathbb{E}_x[S^H S] \Psi_{T^-} \end{bmatrix} \tag{5.22}$$

   With Assumption 4 $\mathbb{E}_x[S^H S]$ has full rank L and with Assumption 5 the matrix Ψ_T has rank $2p$ with $L \geq 2p$. Thus, the upper left block in the block decomposition in (5.22) has also rank $2p$. Hence, the columns of $\mathrm{E}_{\mathrm{ss}}(\theta)$ with indices T are independent. Since this is true for every T, every combination of $2p$ rows is independent.

2. Let x' be a second solution of (5.21), and assume that $x \neq x'$ and that $\|x'\|_0 \leq p$. Then

   $$\mathrm{E}_{\mathrm{ss}}\big(x - x'\big) = 0.$$

   Since $(x - x')$ has less than $2p$ non-zero components, but any $2p$ columns of $\mathrm{E}_{\mathrm{ss}}(\theta)$ are independent, it follows that $(x - x') = 0$. This is a contradiction.

■

This result is a lot stronger than the equivalent result in non-blind sparse models, since then it only holds true in the noiseless case. However, in the ESpaM algorithm, the noise is already taken into account in the expectation step. Hence, the Problem (5.15) is correct. The solution will satisfy the equality constraints.

5.3.3 Semiblind ESpaM algorithm

We will now adapt the ESpaM algorithm such that it is applicable to semi-blind channel identification. Let $(k_1, \ldots, k_U)$ denote the U (equally spaced) time steps at which pilot symbols are introduced that are known to the receiver. We denote the pilot symbols itself by P_{k_u}. This knowledge may be used to improve the blind channel estimation. The adaptation of the blind sparse EM algorithm is straightforward. The sparse M step remains the same with the only difference that we now have more knowledge on the symbols, i.e. the corresponding probability distributions will be more peaked.

The Baum-Welch algorithm where these distributions are derived has thus to be adapted. For each pilot u, the symbol a_{k_u} is known and thus the first component of S_{k_u} is not random. Hence, the forward iteration of the Baum-Welch algorithm for k_u is derived according to the decomposition of the filtering probabilities

$$p_k(s|y_{1:k};\theta) = g_k\left(y_k|s;x\right) \sum_{s'} q\left(s\left|s'\right.\right) p_{k-1}(s'|y_{1:k-1};\theta) \tag{5.23}$$

for those states s_{k_u} that coincide with the pilot symbol P_{k_u} in their first coefficient, while the remaining probabilities are irrelevant:

$$\bar{p}(s_{k_u}|y_{1:k_u};x) = \begin{cases} p_k(s|y_{1:k};\theta), & \text{if } S_{k_u}[0] = P_{k_u} \\ 0, & \text{otherwise,} \end{cases} \tag{5.24}$$

where $s_{k_u}[0]$ denotes the first component of s_{k_u}. The filtering probabilities are then again normalized. We stress that (5.24) yields the exact filtering probabilities given the additional knowledge that S_{k_u} is only partly random. The forward iteration for the remaining time steps is given by Decomposition (5.23) as before. The backward iterations are adapted accordingly.

5.4 Simulations

The focus of this section is on the evaluation of the performance of the ESpaM algorithm. We have already provided several results for Monte-Carlo experiments on the standard algorithm in Section 4.7 in Chapter 4, where the focus is rather on the expectation step. In other words, in Section 4.7, we compare different solutions for the E step, while in this section we compare different solutions for the M step. The E step is the same for the EM as well as the ESpaM algorithm.

We apply the ESpaM algorithm as mentioned in the introduction of this chapter to three concrete examples of blind and semi-blind channel estimation in digital communications. The first example is a time-invariant but sparse multi-path channel as presented in Section 3.4. For this model the basis matrix Ψ is the identity matrix, such that the solution to the maximization equation (5.10) is unique. Thus, it is possible to compare the ESpaM algorithm to the EM algorithm. We will show, that for large channel orders but sparse impulse responses the ESpaM algorithm is clearly superior.

The second and third examples are doubly-selective channels with a linear and an OFDM modulation respectively. For these examples, the sparsity arises from introducing a fine grid on the parameter space. Consequently, the maximization problem as in (5.10) is no longer unique, such that we may not apply the EM algorithm. As a comparison we will use the algorithm by Ben Salem and Salut [2004] for the linear modulation and a semi-blind as well as a non-blind method for the OFDM transmission.

5.4.1 Sparse Time-Invariant Multipath Channel

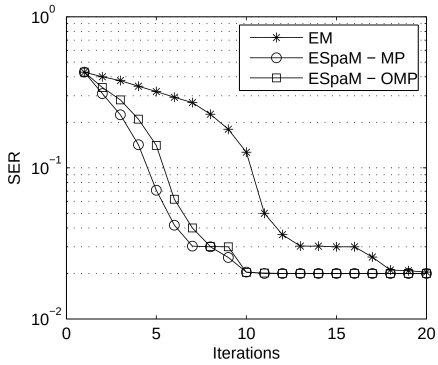

Figure 5.1: Time-invariant channel: SER over iterations of EM using the maximization methods, $L = 8$, $p = 2$, SNR 12dB

We start by considering the time-invariant channel model as in Section 3.4, since in this case we have a comparable method readily at hand by using the standard non-sparse maximization step of the EM algorithm as the solution of (5.10). This is possible because only in this case the number of relevant symbols L is equal to the taps of the channel Q, such that E_{ss} has full rank.

We used a QPSK modulation and measured the performance in terms of the symbol error rate (SER) and the mean-squared error $\mathrm{MSE}(\hat{x}) = \mathbb{E}\left(\|\hat{x} - x\|^2\right)$ of the channel. In the figures we refer to the exact solution to (5.10) as the EM, to the ESpaM using matching pursuit and the orthogonal matching pursuit solving (5.10) as ESpaM - MP and ESpaM - OMP respectively. For the MP and OMP we used p+3 iterations, where p is again the number of active components in the channel impulse response. This is to show, that it is not necessary to exactly know p, the

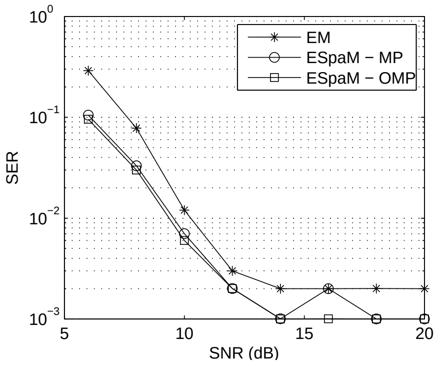

Figure 5.2: Time-invariant channel: SER of EM using the maximization methods, $L = 8$, $p = 2$, over different SNR

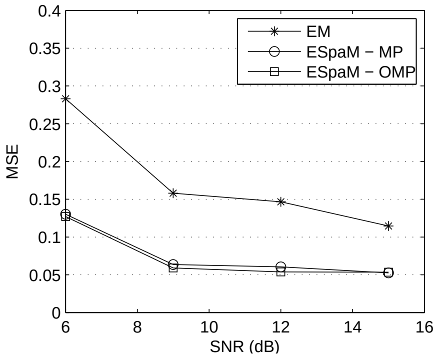

Figure 5.3: Time-invariant channel: MSE of channel of EM using the maximization methods, $L = 7$, $p = 2$, over different SNR

algorithms work well even if they are run with more iterations. However, as we explain later the number of iterations should not be chosen too large.

Since the algorithms work completely blindly, i.e. no symbol is known, there are obviously symmetries for the estimated sequence, which are removed before calculating the SER and the MSE. The support of the sparse channel as well as its coefficients are drawn from a uniform distribution for each Monte-Carlo run. For each method, we use one single random initial parameter estimate $\hat{\theta}^{(0)}$.

The first simulations were run with channel order $L = 8$ and $p = 2$ non-zero components. Fig. 5.1 shows the MSE over the first 20 EM iterations at SNR 12dB. The sparse methods MP and OMP converge considerably faster than the exact method. The SER after convergence is also slightly smaller for the MP and OMP, see Fig. 5.2. In Fig. 5.3 we compare the MSE of the maximization methods after 20 iterations of the EM for different SNR (with $L = 7$). Even after convergence, the OMP shows still better performance than the exact method. The MP has a slightly higher MSE if the SNR is large. This might indicate that the MP introduces a slight bias, which is however not significant for the estimation of the SER.

These first simulations showed that for small channel orders the exact maximization still provides satisfactory results. We now turn to larger channel orders and replace the Baum-Welch algorithm by Particle Smoothing. Fig. 5.4 compares the approximate EM involving Particle Smoothing (with different particle sizes) to the plain EM with the Baum-Welch algorithm both using the OMP maximization method. We have thus verified that there is no essential difference between the exact smoothing and the Particle Smoothing, i.e. the loss of using Particle Filtering

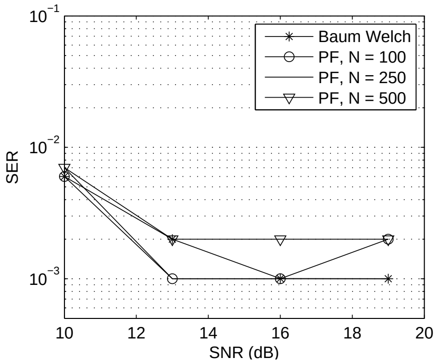

Figure 5.4: Time-invariant channel: SER of ESpaM using Baum-Welch algorithm vs. Particle Filtering with different particle sizes, $L = 8$, $p = 2$, OMP maximization

is very moderate.

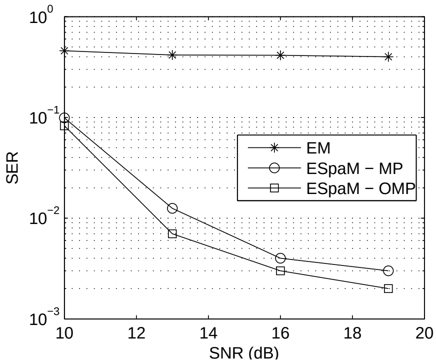

Figure 5.5: Time-invariant channel: SER of Particle Filtering using the maximization methods, $L = 15$, $p = 3$, $N = 100$, vs. SNR

Fig. 5.5 shows the SER for the four maximization methods using Particle Smoothing for a significantly higher channel order $L = 15$ with $p = 3$ active components. It appears that the non-sparse likelihood function has now many local maxima such that the EM algorithm is not robust anymore, while the OMP and the MP still have a very low SER. The same behavior is apparent in Fig. 5.5 showing the MSE after 20 iterations.

Finally, we used the channel order $L = 20$ with $p = 4$ active coefficients. Fig. 5.7 shows the MP and the OMP are the only methods capable of tracking a channel of such a large order. The development of the MSE in Fig. 5.8 reveals that the exact maximization method does not converge in contrast to the MP and OMP.

Since E_{ss} has full rank, the correct sparsest solution to (5.10) coincides with the solution of the standard EM algorithm. Thus, the EM-algorithm coincides with an ESpaM algorithm that uses a sparse algorithm that gives the exact sparsest solution, unless there are numerical instabilities with the standard EM algorithm. However, the huge performance increase of the ESpaM algorithm coupled with OMP and MP comes from the fact that these greedy algorithms if stopped after a few number of iterations force the solution to be exactly sparse. The EM algorithm always converges, but its problem is the convergence to local maxima. Restricting the parameter space such that it only contains sparse vectors obviously avoids or eliminates many of these local maxima, such that the ESpaM algorithm with OMP or MP is much more robust with respect to convergence to local maxima. This shows, that it is important to choose a small enough number of iterations of the OMP or MP, otherwise the convergence will be similar to

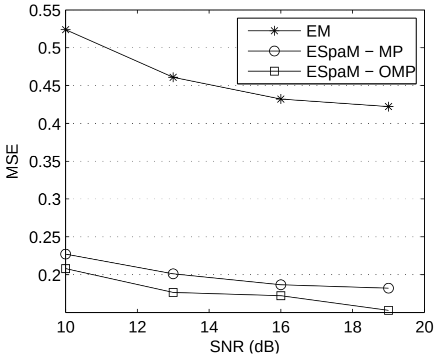

Figure 5.6: Time-invariant channel: MSE of channel of Particle Filtering using the maximization methods, $L = 15$, $p = 3$, $N = 100$, vs. SNR

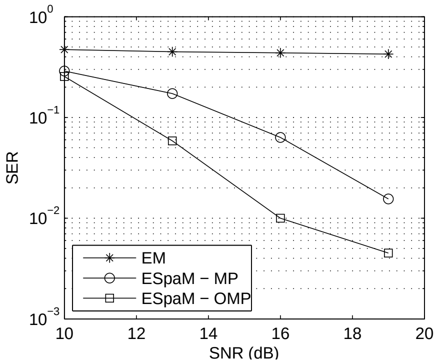

Figure 5.7: Time-invariant channel: SER of Particle Filtering using the maximization methods, $L = 20$, $p = 4$, $N = 100$, vs. SNR

the EM algorithm.

5.4.2 Doubly-Selective Multipath Channel

We now turn to the doubly-selective channel model as in Section 3.5. In contrast to the time-invariant model 3.4, the plain EM algorithm may not be applied since the matrix E_{ss} does not have full rank. This is because the number of grid points Q is much larger than the number of relevant symbols L for each observation. Therefore, the ESpaM algorithm does not only improve the performance, but is the only applicable ML method. We use the OMP algorithm in the M step.

For the following Monte-Carlo experiments, we assume that the channel consists of two paths, each with a random attenuation, a random delay and a random Doppler frequency which are drawn independently at each Monte-Carlo iteration. The number of observations is $K = 100$.

The SER of the ESpaM algorithm is compared to a genius bound where the Doppler frequencies and the delays are assumed to be known, i.e. by using the expected symbols $\mathbb{E}_x[s_k|y_{1:K}]$. Furthermore, we use the mean-squared error (MSE) of the channel impulse response which is now averaged over time. As a performance bound we will use the same sparse method but the symbol matrix is now assumed to be known. Then the problem reduces to the sparse minimization problem

$$\min_x \|S\Psi(\theta)x - Y\|_2^2$$

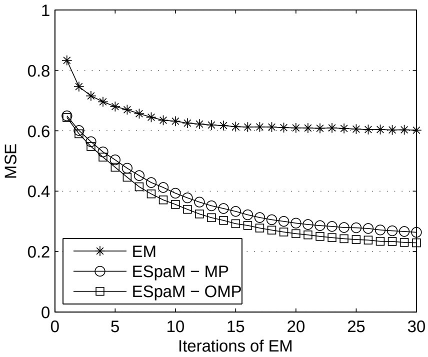

Figure 5.8: Time-invariant channel: MSE over Iterations of EM using the maximization methods, $L = 20$, $p = 4$, $N = 100$, SNR 16dB

to which we also apply the OMP algorithm.

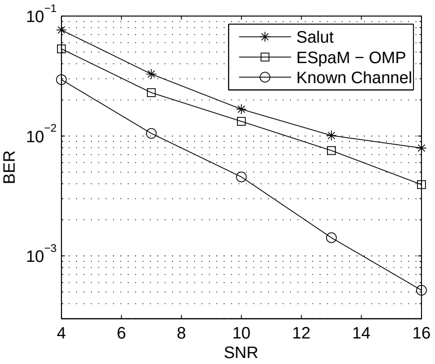

Figure 5.9: Doubly selective channel: BPSK, BER over different SNRs, 2 random delays and Doppler frequencies off grid

As mentioned before, the algorithm by Salut Ben Salem and Salut [2004] with 16 particles is used as a comparison. The initial channel estimate was chosen such that the coefficients corresponding to the time-constant basis vector were random, while the coefficients of the remaining basis vectors were set to 0. This was clearly superior to a completely random initialization.

We start with a BPSK modulation and set the maximal Doppler spread to $\omega_{\max} = 1e^{-2}/T$ and the maximal delay to $2T$ such that the channel order $L = 4$ is sufficient. The grid step size for the estimation is fixed to $1e^{-3}/T$ for the Doppler frequencies and $0.33T$ for the delays. For each Monte-Carlo run, the delays and doppler frequencies are chosen uniformly randomly in the complete range, i.e. they do not lie on the grid points. The ESpaM algorithm is run over 30 iterations until convergence. Salut's algorithm as well as the ESpaM algorithm are started with a set of two different initial parameter estimates. Fig. 5.9 shows the BER over different SNRs. The ESpaM algorithm is thus slightly superior to Salut's algorithm and not too far away from the BER for known channel parameters. The MSE is given in Fig. 5.10. The ESpaM algorithm is thus even more superior to Salut's algorithm regarding the estimation of the channel. Furthermore, in contrast to Salut's algorithm the ESpaM algorithm also gives an estimate of the Doppler frequencies and delays as well as the number of paths of the channel.

The next simulations have been run with a QPSK modulation, while the remaining parameters as Doppler frequencies, delays and channel order have been kept the same. Fig. 5.11 shows again the SER over different SNRs. Salut's algorithm is not adapted to this more complicated model, while the ESpaM algorithm still maintains a low SER.

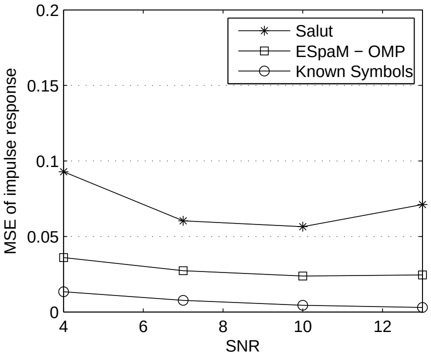

Figure 5.10: Doubly selective channel: BPSK, MSE of channel impulse response averaged over time over different SNR, 2 random delays and Doppler frequencies off grid

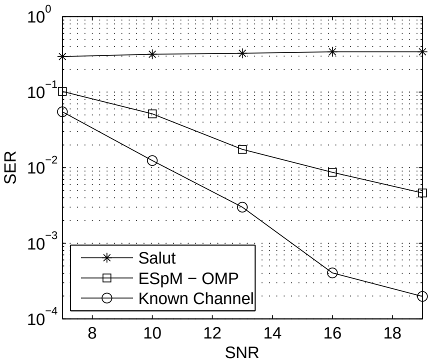

Figure 5.11: Doubly selective channel: QPSK, BER over different SNRs, 2 random delays and Doppler frequencies off grid

We regard again the QPSK modulation but with a higher maximal delay of $5T$. This is still small enough, such that the relevant number of symbols $L = 6$ is sufficiently small to apply the Baum-Welch algorithm. If it is larger, the Baum-Welch algorithm may be replaced by a Particle Smoothing algorithm (see for example Barembruch et al. [2009], Doucet et al. [2000] and many others). The maximal bound on the Doppler spread is again set to $\omega_{\max} = 1e^{-2}/T$. The grid step size for the estimation is fixed to $1e^{-3}/T$ for the Doppler frequencies and $0.33T$ for the delays. For the following simulation, one single random initialization is used for the ESpaM algorithm.

The delays and Doppler frequencies are randomly sampled in the range between minimal and maximal values. i.e. not on the grid. Fig. 5.12 shows the SER after 25 iterations of the ESpaM algorithm using one single random initialization. The true delays as well as the Doppler frequencies have been generated on the grid points. Obviously, the SER of the ESpaM algorithm is larger than for the known channel, but it is still satisfactorily small. The MSE, see Fig. 5.13, quickly converges over the iterations of the ESpaM algorithm. It can be seen that an even better performance could be achieved by using more EM iterations. Almost always the performance of the EM algorithm may be considerably improved by using several initial values, such that these results should be understood as a benchmark of the capacity of the algorithm and not as the lowest MSE or SER achievable in an actual real world implementation.

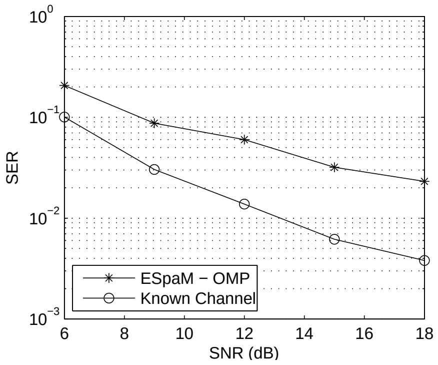

Figure 5.12: Doubly selective channel: QPSK, SER over different SNRs, 2 random delays and Doppler frequencies off grid, maximal delay $5T$, after 25 iterations of the ESpaM algorithm

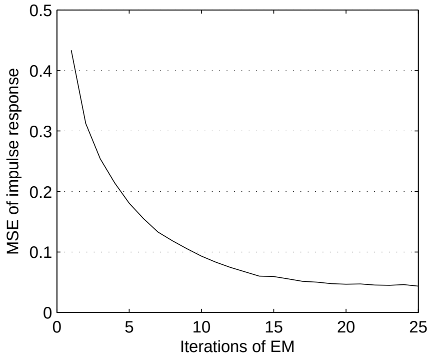

Figure 5.13: Doubly selective channel: QPSK, MSE of channel impulse response averaged over time, over iterations of ESpaM, 2 random delays and Doppler frequencies off grid, maximal delay $5T$, SNR 21dB

5.4.3 OFDM Transmission with Doubly-Selective Multipath Channel

We consider an OFDM system with $K = 128$ subcarriers. The considered doubly-selective channel consists of two paths, for each of which the attenuation, the doppler frequency and the delay are drawn randomly at each Monte-Carlo run.

We use the blind sparse EM algorithm, as well as the semi-blind version with a different number of pilots, namely every 5th symbol and every 10th symbol, which we indicate in the brackets in the following figures. A smaller number of pilots did not increase the performance in comparison to the blind algorithm. We use the OMP algorithm for the sparse M step.

We evaluate the bit error rate (BER) based on the estimated symbol sequence $\mathbb{E}_{\hat{\theta}}[s_k|y_{1:K}]$ where $\hat{\theta}$ is the final parameter estimate. The BER of the blind and semi-blind sparse EM algorithms is compared to a genius bound based on $\mathbb{E}_x[s_k|y_{1:K}]$ where x is the true channel response. Furthermore, we use the mean-squared error (MSE) of the channel impulse response which is averaged over time. As a non-blind performance bound we apply the OMP algorithm to the same grid with known symbol matrix. Then the problem reduces to the sparse minimization problem $\min_x \|S\Psi(\theta)x - Y\|_2^2$.

We start with a BPSK modulation and set the maximal Doppler spread to $\omega_{\max} = 4e^{-2}/K$ and the maximal delay to $4T_s$ such that the channel order $K{+}1 = 5$ is sufficient. For each Monte-Carlo run, the delays and doppler frequencies are randomly chosen between 0 and the upper bounds as multiples of $1/K$ and T_s respectively. Due to the very strong Doppler and

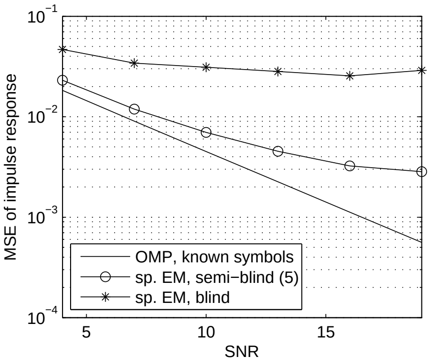

Figure 5.14: OFDM: BPSK, MSE of channel impulse response averaged over time after 10 iterations of sparse EM, 5 initializations, 2 random delays and Doppler frequencies

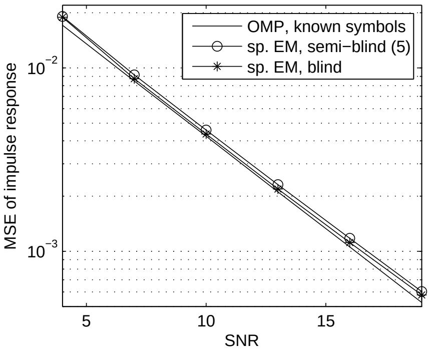

Figure 5.15: OFDM: BPSK, median MSE of channel impulse response averaged over time after 10 iterations of sparse EM, 2 random delays and Doppler frequencies

delay shifts that we consider, the likelihood function exhibits several local maxima. To ensure convergence to the global maximum, we therefore used 5 different initializations for x each of which is chosen completely randomly.

The MSEs in Fig. 5.14 show that despite the very strong Doppler and delay the sparse EM algorithm converges well. The blind algorithm is almost equivalent to the semi-blind algorithm where every 10th symbol is known. Furthermore, the sparse EM algorithm only needs the knowledge of every 5th symbol to yield almost comparable performance to the non-blind OMP. The difference between the 4 schemes is mainly due to the initializations. When the sparse EM algorithm has been badly initialized, it converges to a local maximum, such that the MSE of the channel is very high. However, the median (instead of the mean value) of the channel MSEs over the different Monte-Carlo runs, see Fig. 5.15 reveals that the four schemes are equivalent, i.e. in most cases the MSE of the blind sparse EM algorithm is as low as the non-blind performance bound.

Fig. 5.16 shows the mean BER of the four schemes. Unexpectedly, the BER of the semi-blind scheme with every 10th symbol known is higher than for the blind algorithm. This is in fact due to symmetries in the blind model. Obviously, if x is a maximum of the likelihood then so is e.g. $-x$ as well as shifts in x. These symmetries have been removed for the calculation of the MSE and BER. However, the semi-blind algorithm sometimes converged to a symmetry such that even if the MSE was low, the BER was higher than for the blind algorithm. If each 5th symbol is known, this behavior does not occur. Apparently, enough symbols are known to prevent convergence to a symmetry.

Finally, Fig 5.17 shows the rate of convergence where a lower performance bound is given by

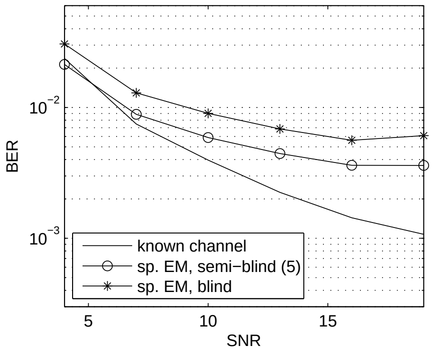

Figure 5.16: OFDM: BPSK, BER over different SNRs after 10 iterations of sparse EM, 5 initializations, 2 random delays and Doppler frequencies, sp. EM = sparse EM algorithm, semi-blind(5) = pilot at every 5th symbol

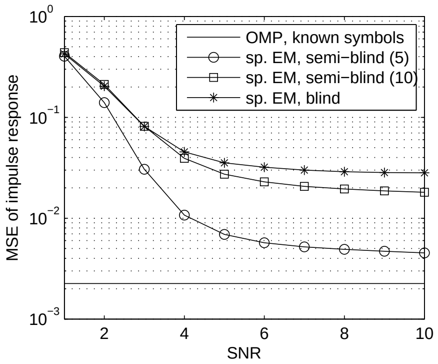

Figure 5.17: OFDM: BPSK, MSE of channel impulse response at SNR 13dB over the first 10 iterations of the EM algorithm, non-blind OMP as lower performance bound, 2 random delays and Doppler frequencies

the non-blind OMP algorithm. The sparse blind and semi-blind EM algorithms converge very quickly within about five iterations.

Chapter 6

Blind Classification

Blind Modulation Classification characterizes the estimation of the modulation scheme used at the transmitter without knowledge neither on the channel nor on the sent symbols. We consider more specifically blind estimation of linear modulation schemes. In this case, the model is given as in Chapter 3 and what Blind Classification is effectively estimating is the symbol alphabet of the modulation scheme or in other words the state space of the model. Thus, what is known as Blind Classification in the signal processing community, is called model estimation or model testing in the statistics community.

Blind Classification is at the top level of the described chain of algorithms. We are in a situation where almost all parameters are unknown. Obviously, this includes the modulation alphabet $\mathcal{X}$, as well as the channel state information x, the noise level σ^2 and the symbols S or s_k respectively. We consider as well the case where the order L of the finite impulse response of the channel is unknown.

The methods in the literature may be split up into two categories, feature-based methods and maximum likelihood (ML) methods. An elaborated and recent survey of existing Classification methods is given in Dobre et al. [2007]. Algorithms of the first category use features of the model like all forms of moments of the amplitude, the frequency, the phase or wavelet transforms to identify the modulation scheme or test for a specific modulation scheme. Unfortunately, all existing algorithms are restricted to single-path fading channels or even to a model of additive-white-Gaussian-noise (AWGN) channels.

We will therefore concentrate on the second category and consider only likelihood based model estimators. Their common ground is a list of potential modulations schemes, such that each modulation scheme defines a potential model. Therefore, we will use the terms modulation scheme and model synonymously in this chapter - with a little confinement: the channel order may also be part of the model. The likelihood of the received observation sequence has to be calculated or estimated for each of these potential models. In a second step a model estimator establishes a decision on the model based on these likelihoods. If only two potential models are considered, then the problem may be reformulated as a statistical test. The hypothesis that the first model is true is tested against the hypothesis that the second model is true. The test statistics are based on the ratio of the likelihoods in both models and are therefore called likelihood ratio tests (LRT). If more than two potential models are considered, then we speak rather of model estimators. But they are equally based on the likelihoods of the models and on the same decision criteria. Hence, even if we speak about LRTs, we note that every statement easily extends to the estimation of several potential models.

The proposed ML Classification methods in the literature differ in how the likelihood in each model is calculated. In general, it is not possible to calculate the exact likelihood because of the unknown parameters. Here again two different concepts exist. On the one hand, many methods use an averaged likelihood ratio test (ALRT) which considers the unknown parameters as random quantities. The likelihood of the model is then obtained by marginalization, i.e. by integration over the unknown parameters. This is feasible in models like AWGN channels, but in the presented frequency-selective multi-path channel model, the parameter space has a large dimension and numeric integration is therefore computationally infeasible. The second concept are generalized likelihood ratio tests (GLRT) which consider the parameters as unknown constants. Consequently, GLRTs estimate the parameters in each model and base their likelihood estimate on the estimators of the unknown parameters. This is the concept we will focus on.

The digital communications system is mathematically modelled as a hidden Markov model (HMM) as given in (3.4). The formulation of the GLRTs is possible even in a very general model. It is however not obvious and often not possible to derive the ML estimate of the unknown parameters in the general model. However, in 5 we discuss ML estimators with which the GLRTs are implementable. The order of the finite impulse response of the channel is a priori not known and will as well be considered as an uncertainty of the model, such that each combination of a potential channel order and modulation constellation size is a potential model.

Hong [2006] recently proposed an estimator for BSPK and QPSK modulation employing the same channel model. The channel parameters are estimated using the expectation-maximization (EM) algorithm coupled with the Baum-Welch algorithm as described in 5. While all parts of the estimator are very well known for decades, it is not applicable to larger constellations due to the exploding computational complexity. We will therefore discuss an approximate ML estimator based on a Particle Smoothing method and present several ideas to improve the estimator.

If the channel order is fixed, the number of parameters is fixed over the different models, such that even the simple decision for the model with the maximal likelihood is consistent. In this case, there is thus no need of a more involved decision rule. If the channel order is unknown and included in the model estimation we propose to use the Bayesian Information Criterion (BIC) in order to avoid overestimation of the channel order. The BIC follows the principle of “Minimum Description Length” (MDL) by Rissanen [1978] which suggests to choose the model giving the shortest description of the data. A model comprising less parameters is therefore preferable over a model with more parameters if the data is equally described by both models.

Since the concept of the GLRT is well known in this context, we will rather focus on the practical implementation and the issues of complexity and robustness that one faces for large modulation schemes. After having read the previous chapters, it becomes quite obvious how the different ML estimation procedures may be used as an implementation and we will show how each step fits into the model estimation.

An important point of the (approximate) EM algorithm (in each model) as implementation of the ML estimation of the unknown parameters is the choice of the initial value of the unknown parameters, i.e. the channel coefficients and the variance of the noise. Without any prior knowledge, they have to be chosen randomly. We improve the choice for large constellation sizes significantly by employing the estimates for smaller constellation sizes. The large modulation schemes may be closely approximated by (biased) macro-states models, i.e. by melting down a certain number of neighboring points to one single macro-state. For example, by melting down each 4 neighboring points of a 64-QAM alphabet we obtain a rescaled 16-QAM model. Therefore, the channel estimate in the 16-QAM model, which has to be calculated anyway, may serve as an initial value for the estimation in the 64-QAM model. We present simulation results that show the potential of these methods and the gain of robustness in the macro-states model that even outweighs the introduced bias. This concept often called multilevel strategies has already been used by Guo et al. [2004], Eyuboglu et al. [1988], Aggarwal and Wang [2007] in a slightly different way to reduce the complexity of Particle Filtering. Before sampling particles in the large space, pilot samples are generated in different levels - corresponding to the macro-states.

Finally, we address the problem that Particle Filtering are hardly able to identify states with high smoothing weights if the channel is not in minimum phase (i.e. the last channel coefficient is large). We suggest a partial solution of this prevalent problem that consist of reversing the direction of the HMM, i.e. reversing the order of the observation sequence. This allows to estimate channels in maximum phase. We apply this method to half of the initializations of the approximate EM algorithm and the standard method to the remaining ones, such that our algorithm is able to estimate channels in maximum and minimum phase.

Unfortunately, little theoretical justification is known for GLRTs in the Blind Classification context. In general, hypothesis tests estimating the order of a state space in HMMs or even in Gaussian mixture models are not consistent due to the non-identifiability. On the other hand, we will show that, due to the restrictive structure of the linear modulation schemes, the estimation is consistent in this specific application.

Furthermore, in addition to interpreting the Classification of testing for two distinct models, we use a different approach, which makes use of the specific structure of QAM schemes. It is therefore not readily extendable to other modulation schemes. We place the two potential QAM schemes in a larger model enwrapping the two modulation schemes. The resulting interpretation is thus that the Classification is rather a parameter estimation problem. Furthermore, the underlying model in this case has a very restrictive structure that we will exploit to show the identifiability in this model and to develop asymptotic properties of the proposed tests.

In our approach, we place the estimation in the context of the larger of the two QAM constellations that shall be tested and consider the proportion of each of the symbols as a new unknown parameter. If the true model corresponds to the larger QAM, all of the proportions will be equal, whereas if it is the smaller QAM, a certain part of the proportions will be equal to zero. We show that this model is identifiable for both of the possible constellations. To simplify the theoretical analysis, we assume a flat fading channel without inter-symbol interference, such that the HMM reduces to a Gaussian mixture model. We note, that the proposed test statistics may as well be applied to HMMs, though in this case theoretical results are hard to obtain. Gassiat and Boucheron [2003] pioneered the theoretical analysis for order estimation in HMMs. Even in their work on finite emission alphabets, they have to resort to large deviation methods as in Dembo and Zeitouni [1999].

The standard GLRT as in Hong [2006] reconsidered in this new framework is a simple hypothesis test, i.e. we test that a certain part of the proportions is equal to zero versus all proportions are equal.

The second test statistic that we propose considers the null hypothesis that the true model corresponds to the larger modulation scheme, i.e. that all proportions are equal. We test against the alternative hypothesis, that the proportions are not equal. The advantage of this test is, that the asymptotic distribution of the statistic under the null hypothesis is well known and equal to a Chi-Squared distribution, which does not depend on the choice of the parameters. In contrast to the first test, this second one is thus very easy to calibrate.

Finally, the third test statistic we propose is a composite hypothesis test and tests the null hypothesis that the true model is the smaller model, i.e. that a certain part of the proportions is equal to zero. The alternative hypothesis is thus that these proportions are unequal to zero. The asymptotic distribution under the null hypothesis is somewhat more intricate since the parameter under the null hypothesis is on the boundary of the parameter space. We show however, following Andrews [2001], that it is a mixture of a Chi-Squared distribution and a point mass distribution.

6.1 State of the Art

Blind Classification is a rather recent research area in digital communications. Dobre et al. [2007] have recently given a very comprehensive survey of Blind Classification techniques. Two different concepts are prevalent. On the one hand, there are likelihood based methods. These methods decide for a model that maximizes the likelihood of the given observation sequence. They are optimal in the Bayesian sense since they minimize the probability of false classification. The difference in the proposed ML Classification algorithms is mainly in how to treat the unknown parameters x.

The Average Likelihood Ratio Test (ALRT) treats the unknown parameters as random variables with some known distribution and integrates the likelihood with respect to this distribution. This has been applied to modulation Classification by Wei and Mendel [2000] for digital amplitude-phase modulations, by Huan and Polydoros [1995] for MPSK modulations in additive white Gaussian noise (AWGN), by Abdi et al. [2004] for multi-antenna arrays and many others. The second concept is the Generalized Likelihood Ratio Test (GLRT) where the unknown parameters are estimated in each model based on a ML criterion beforehand to the Classification. Applications to Classification include the work by Panagiotou et al. [2000] on AWGN channels, the work by Lay and Polydoros [1995] employing per-survivor processing techniques under inter-symbol interference (ISI), as well as plenty other works. The concepts have also been combined in hybrid techniques and many algorithms have been proposed that reduce the complexity by using approximate methods. A recent summarizing work on these Quasi Hybrid Likelihood Ratio Test is by Hameed et al. [2009]. Unfortunately, most existing algorithms are restricted to single-path fading channels or even to a model of additive-white-Gaussian-noise (AWGN).

Hong [2006] proposed an GLRT estimator for BSPK and QPSK modulations employing the same channel model that is used in this thesis. The channel parameters are estimated using the expectation-maximization (EM) algorithm by Dempster et al. [1977] coupled with the Baum-Welch algorithm by Baum et al. [1970]. While all parts of the estimator are very well known for decades, it is not applicable to larger constellations due to the exploding computational complexity. Hence, the standard algorithm proposed in this thesis, is inspired by Hong [2006], replacing the Baum-Welch algorithm by approximate techniques and optimizing the algorithm.

Likelihood Ratio Tests are very well studied in the statistics community. A very comprehensible and comprehensive book on that topic is Lehmann and Romano [2005].

In contrast to ML based CLassification, feature based approaches use a variety of different properties of the received signal, like the variance of the centered signal, the variance of the zero-crossing interval, wavelet transforms of the signal, higher order statistics and many others. Since all of these criteria concern frequency-flat fading and none of them is applicable or easily generalizable to frequency-selective or even doubly-selective channels, we do not consider the approach of basing the Classification on these features of the model. We refer the reader to Dobre et al. [2007] for a review and survey of these methods. A novel approach by Ahmadi and Berangi [2008] considers the constellation of the received symbols. An essential prerequisite is however an AWGN channel.

6.2 Standard Maximum Likelihood Estimator for Blind Classification in Frequency-Selective Channels

We will now discuss the Blind Classification implementation for frequency-selective multipath channels with a focus on linear modulations with large constellations, such that standard methods as in Hong [2006] are not feasible anymore.

We will focus on the frequency-selective channel model as described in Section 3.4. In the next section we will revisit the model again to introduce some additional notations that are necessary for the model estimation procedure. The model estimator is also valid for the more general models. However, the solutions that we will discuss to solve the implementation issues are mainly adapted to the frequency-selective channel.

After having discussed the model, we will briefly introduce the concept of model estimation by the principle of minimum description length. We will then go more into detail to speak about the time reversed hidden Markov chain, which facilitates the estimation if the channel is minimum phase. We will then discuss a macro-states model and how this can help to profit from estimates in the 'small' models (like 4-QAM and 16-QAM) for the estimation for the 'large' models (like 64-QAM and larger).

6.2.1 Model Description

Let $\mathcal{X}_p$ be the alphabet of size $d_p = |\mathcal{X}_p|$ of the p-th of P potential modulation schemes. Let $\mathcal{X}$ denote the true alphabet that coincides with one of the $\mathcal{X}_p$.

Suppose, that we have a sequence $\{a_k\}_{k\geq 0}$ of symbols that are uniformly and independently drawn from $\mathcal{X}$. Let L be the (unknown) channel order and denote by $h = (h_0, \ldots, h_{L-1})^T$ the channel coefficients.

Analogously to Section 3.4, the observation sequence is then given by:

$$y_k = \textstyle\sum_{l=0}^{L-1} a_{k-l} h_l + \varepsilon_k, \quad \varepsilon_k \sim \mathcal{N}_{\mathbb{C}}\left(0, \sigma^2\right),$$

where $\{\varepsilon_k\}_k$ is a sequence of i.i.d. complex circular Gaussian variables with variance σ^2. We rewrite the model in matrix notation to obtain a Hidden Markov Model (HMM):

$$y_k = h^T s_k + \varepsilon_k, \quad \varepsilon_k \sim \mathcal{N}_{\mathbb{C}}\left(0, \sigma^2\right),$$

where $s_k = [a_k, a_{k-1}, \ldots, a_{k-L+1}]^T$ and $\mathbf{w}_k = [a_k, 0, \ldots, 0]^T$. For more details, we refer to 3.4.

Each potential model is a combination of a potential alphabet $\mathcal{X}_p$ and a potential channel order L. We assume that an upper bound on the channel order is known. The state space of the Markov chain $\{s_k\}_k$ in the p-th model is $\mathcal{S}_{p,L} = (\mathcal{X}_p)^L$ and is of size $(d_p)^L$. The vector of unknown parameters in each model is denoted $\theta_L = (h, \sigma)$ of which the parameter space does not depend on the modulation p, but only on the channel order. The transition probabilities from state $s = [a_0, \ldots, a_{L-1}]^T$ to $s' = [a'_0, \ldots, a'_{L-1}]^T$ are given by

$$q_{p,L}\left(s' \,|\, s\right) = \begin{cases} \frac{1}{d_p} & \text{if } [a_0, \ldots, a_{L-2}] = [a'_1, \ldots, a'_{L-1}] \\ 0 & \text{otherwise.} \end{cases}$$

Furthermore, the likelihood of the observation y conditional to the current state s may be expressed as

$$g_L\left(s, y; \theta\right) = \frac{1}{\pi\sigma^2} \exp\left(-\frac{1}{\sigma^2}\left|h^T s - y\right|^2\right).$$

For ease of presentation, we will assume in this chapter that the data likelihood is constant over time. Observe that the likelihood function only depends on the channel order and not on the modulation.

We assume as in the previous chapters, that an observation sequence of K symbols $Y = y_{1:K} = (y_1, \ldots, y_K)$ is given and fixed.

Finally, for $1 \leq k \leq j \leq K$, we introduce the short-hand notations $s_{k:j} = (s_k, \ldots, s_j)$ for a sequence of states and similarly, $y_{k:j} = (y_k, \ldots, y_j)$ for a sequence of observations.

6.2.2 Model Estimation

Assume that we are interested in the Classification of the P linear modulation schemes with alphabets $\mathcal{X}_p$ for $p \in \{1, \cdots, P\}$. Furthermore, we assume a minimal potential channel order $L_{\min}$ and the maximal potential channel order $L_{\max}$. The set Θ_L of all possible parameters for a given channel order L is given by

$$\Theta_L = \{\theta = (\theta_0, \cdots, \theta_L) |\, \theta_0, \cdots, \theta_{L-1} \in \mathbb{C},\, \theta_L \in \mathbb{R}^+\}$$

such that the first L components describe the channel coefficients and θ_L the standard deviation of the noise, independently of the modulation $\mathcal{X}_p$. The dimension $\mathrm{Dim}(\Theta_L)$ of the parameter space is thus equal to $L+1$ and it does not depend on the modulation scheme. We define a model as the set of the (parameterized) likelihood functions describing this model, i.e. $\mathcal{M}_{p,L} = \{l_{p,L}(\cdot, \theta_L) | \theta_L \in \Theta_L\}$, where $l_{p,L}(\cdot, \theta_L)$ is the likelihood function in the model given by p and L with parameter θ_L taking as the first argument an observation sequence of length K.

We follow the principle of "Minimum Description Length" (MDL) Rissanen [1978] which suggests to choose the model giving the shortest description of the data. A model comprising less parameters is therefore preferable over a model with more parameters if the data is equally described by the two models.

The vector of unknown parameters consists of the channel coefficients and the standard deviation of the noise. The exact calculation of the data likelihood is therefore not possible since it requires the knowledge of the channel and of the noise. We replace the unknown parameters by their ML estimates for the given sequence Y. We denote the ML estimate in the model $\mathcal{M}_{p,L}$ by $\hat{\theta}_{p,L}$.

A common and well-studied criterion implementing the MDL principle is the Bayesian Information Criterion (BIC) introduced by Schwarz [1978] which is given by

$$\mathrm{BIC}(\mathcal{M}_{p,L}) = -\log l\left(Y, \hat{\theta}_{p,L}\right) + \tfrac{\mathrm{Dim}(\Theta_L)}{2} \log K$$

The criterion is hence a penalized log-likelihood. The larger the dimension of the underlying model, the stronger is the penalty.

The model estimator is hence given by

$$\hat{\mathcal{M}}_{\mathrm{BIC}} = \arg\min_{\mathcal{M}_{p,L}} \left\{ -\log l\left(Y, \hat{\theta}_{p,L}\right) + \tfrac{L+1}{2} \log K \right\}.$$

With this penalty term, the model estimator is consistent for $K \to \infty$, although it is not the smallest possible penalty term having this property. On the other hand, the dimension of the parameter space in this context is very small and does not depend on the modulation scheme. Furthermore, if the channel order is assumed to be known, the penalty term disappears completely. If the channel order is not known, its estimation is less important than that of the modulation, since the assumption of a finite, fixed channel order L is a simplification of the real model. The smaller the channel order is chosen, the more easily it is estimated, the larger it is chosen, the more precise is the model. Its estimation is thus a compromise between the accuracy of the estimation and the accuracy of the model.

6.2.3 Approximate ML-Estimation of Unknown Parameters

In this section, we consider the ML estimation of the unknown parameters - channel coefficients and standard deviation - for a given model $\mathcal{M}_{p,L}$. To keep the notations concise, we neglect omit the dependence on p and L when there is no ambiguity.

The Maximum-Likelihood (ML) estimate of the unknown parameters $\theta = (h, \sigma)$ is the maximum of the function $\theta \to l(Y; \theta)$. The Expectation-Maximization (EM) algorithm Dempster et al. [1977] is a well-known method to derive the ML estimator in incomplete data models. We discuss the EM algorithm in detail in Section 5.2. It consists of two steps, which are repeated iteratively: the expectation (E) step and the maximization (M) step. In the E step in this model, the smoothing distributions $p_k(\cdot|Y; \theta) = \mathbb{P}\left(s_k = \cdot \,|\, Y; \theta\right)$ of the signal s_k are calculated, given the observations Y and the current fit θ' of the parameters θ. The M step updates the parameter estimate θ', in this model it only requires to solve a small system of linear equations.

The Baum-Welch algorithm Baum et al. [1970] as described in Section 4.2 is an efficient implementation of the E step if the state space is small enough. Otherwise, approximate smoothing methods have to be applied to reduce the complexity. We discuss Particle Smoothing algorithms as well as approximations of the Viterbi algorithm in Chapter 4. Amongst these algorithms, we concentrate here on the fixed-interval smoothing algorithm Doucet et al. [2000], Godsill et al. [2004], which is the superior to other approximate smoothing algorithms in this context Barembruch et al. [2009]. The exact algorithm is given as in Algorithm 3.

6.2.4 Reversed HMM Model

The two previous methods have a common drawback. Since the points of the support of the smoothing distributions are simulated from a Particle Filtering algorithm, the approximations will fail, if the correct smoothing distribution contains points with high smoothing probabilities but small filtering probabilities. This is the case, if the channel is minimum phase. Otherwise, the forward particle filter is less likely to identify these states. On the contrary, a backward particle filter would be easily able to identify them if the channel is maximum phase. We will now propose a method to diminish this effect and to improve the estimation of the unknown parameters with the approximate EM algorithm.

In this context, the time reversed chain (s_k, y_k) for $k = K, K-1, \cdots, 1$ is as well a hidden Markov model. Indeed, the transition probabilities form as well a probability distribution on s_k, keeping s_{k+1} fixed, i.e. $\sum_{s \in \mathcal{S}} q\left(s' \,|s\right) = 1$ for each $s' \in \mathcal{S}$. Hence, the (marginal) Particle Filtering algorithm can be applied backwards in time to estimate the backward filtering distribution $p_k(s|y_{k:K};\theta) = \mathbb{P}\left(s_k = \cdot|y_{k:K};\theta\right)$ for $k = K, K-1 \cdots, 1$ and $s \in \mathcal{S}$, based on the decomposition:

$$\begin{aligned} p_k(s|y_{k:K};\theta) &= \frac{\mathbb{P}\left(s_k = s|y_{k+1:K}\right) g_k\left(y_k|s;\theta\right)}{p(y_k|y_{k+1:K})} \\ &\propto \mathbb{P}\left(s_k = s|y_{k+1:K}\right) g_k\left(y_k|s;\theta\right) \\ &\propto g_k\left(y_k|s;\theta\right) \sum_{s' \in \mathcal{S}} q\left(s' \,|s\right) p_{k+1}(s'|y_{k+1:K};\theta). \end{aligned} \tag{6.1}$$

Similarly to Decomposition (4.27), the smoothing probabilities $p_k(s|Y;\theta)$ for $s \in \mathcal{S}$ may also be decomposed in:

$$p_k(s|Y;\theta) = p_k(s|y_{k:K};\theta) \sum_{s' \in \mathcal{S}} \frac{q\left(s\,|s'\right) p_{k-1}(s'|Y;\theta)}{\sum_{s'' \in \mathcal{S}} q\left(s''\,|s'\right) p_k(s''|y_{k:K};\theta)} \tag{6.2}$$

It is thus as well possible to approximate the marginal smoothing probabilities by a backward particle filter and a forward weights correction, equivalently to the original fixed-interval Particle Smoothing. The backward particle filter is the same particle filter used in Alg. 3, but now applied to the reversed hidden Markov model, i.e. to the observation sequence $y_{K:1} = (y_K, \cdots, y_1)$. The forward weights correction is then the equivalent of Equation (4.27), but now calculated in forward iterations from $k = 2$ to $k = K$. We will refer to this particle algorithm as reversed fixed-interval smoothing. It is clear that the theoretical properties of this algorithm are exactly equal to the standard fixed-interval smoothing, as well for the approximation of the smoothing probabilities as for the convergence of the EM algorithm replacing the E step by these algorithms.

We now dispose of two smoothing algorithms of which one is very powerful if the channel is minimum phase and the other one when it is maximum phase. In practice we do not know the channel and we are hence not able to decide which of these algorithms to employ. But in practice, the EM algorithm is initialized anyway several times with different values to ensure the convergence to the global maximum of the likelihood function. We propose hence to use the standard fixed-interval smoothing for half of the initializations of the algorithm and the backward fixed-interval smoothing for the remaining half. By doing so, the practical convergence of the estimate of the unknown parameters is significantly improved compared to only using the

Algorithm 4: Approximate EM algorithm

```
Input  : $I$, $(\hat{h}^i, \hat{\sigma}^i)$ for $i \in \{0, \cdots, I-1\}$, $y_{1:K}$, $L$, $N$, $\mathcal{X}$
Output: $\hat{h}^i$, $\hat{\sigma}^i$, $\hat{l}^i$   Parameter and likelihood estimate for $i \in \{0, \cdots, I-1\}$
for $i \in \{0, \cdots, I-1\}$ (each initialization) do
    if $i \mod 2 = 0$ (every second initialization) then
        $y_{1:K} \leftarrow y_{1:K} \rightsquigarrow$ forward fixed-interval smoothing
    else
        $y_{1:K} \leftarrow y_{K:-1:1} = (y_K, \cdots, y_1) \rightsquigarrow$ backward fixed-interval smoothing
    end
    repeat
        • E-step: fixed-interval smoothing: run Algorithm 3 :
          Input  : $y_{1:K}$, $L$, $N$, $\mathcal{X}$, $(\hat{h}^i, \hat{\sigma}^i)$,
          Output: $\hat{p}_k(\cdot|Y;\theta)$ pour $k \in \{1, \cdots, K\}$, $\hat{l}$
        • M-step: Update of parameters:
          $$(\hat{h}^i,\ \hat{\sigma}^i) \leftarrow \underset{h\in\mathbb{C}^L, \sigma\in\mathbb{R}^+}{\arg\max} \left(-K\log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2}\sum_{k=0}^{K}\mathbb{E}_{\hat{p}_k(\cdot|Y;\theta)}\left[\left|h^T s_k - y_k\right|^2 \middle| Y\right]\right)$$
    until stop condition;
    if $i \mod 2 = 1$ (adaptation of channel estimate if observation reversed) then
        $\hat{h}^i_{0:L-1} = \left(\hat{h}^i_{L-1}, \cdots, \hat{h}^i_0\right)$ (reverse the channel coefficients)
    end
end
```

standard fixed-interval smoothing as we demonstrate in Section 6.2.6 with the help of several Monte-Carlo experiments.

Alg. 4 shows how to implement the approximate EM algorithm employing the two fixed interval smoothing algorithms with I different initializations and initial estimates $(\hat{h}^i, \hat{\sigma}^i)$ of the unknown parameters for each initialization $i \in \{0, \cdots, I-1\}$.

6.2.5 Macro-States Model

In this section, we consider a macro-states model which will serve as an improvement of the initial parameter estimates of the large modulation schemes (64-QAM, 128-QAM, 256-QAM). We show in Section 6.2.6 that with the help of the macro-states model, the computational complexity of the parameter estimation for the large modulation schemes is significantly reduced and the convergence is even improved. This reduces thus as well the complexity of the model Classification.

Definition 4 *Let $\mathcal{X} \subset \mathbb{C}$ be a grid in the complex plane consisting of d points. A grid $\mathcal{X}_M$ of size $d_M < d$ is called a* ***macro-states model*** *for $\mathcal{X}$ with a set of* ***residues*** *$\mathcal{X}_R$ of size d_R if*

1. $\forall x \in \mathcal{X} : \exists! x_M \in \mathcal{X}_M \exists! x_R \in \mathcal{X}_R : \quad x = x_M + x_R$,
   i.e. each point in $\mathcal{X}$ is the unique sum of a point in $\mathcal{X}_M$ and one point in $\mathcal{X}_R$,
2. $d = d_M d_R$.

Fig. 6.1 shows such a macro-states model for the example of a 64-QAM modulation alphabet. The macro-states alphabet stems thus from a rescaled 16-QAM model and the residues are equal

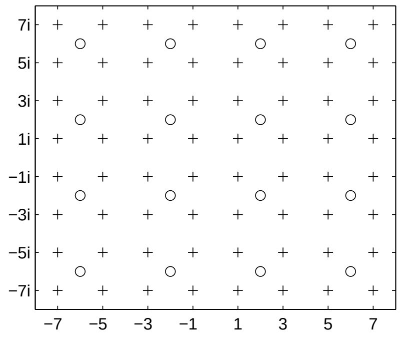

Figure 6.1: Symbols of 64-QAM (+) and of rescaled 16-QAM (○)

to the alphabet of a 4-QAM. The alphabets of the QAM models allow thus for two chains of macro-states models. The first chain consists of the 4-QAM, the 16-QAM, the 64-QAM and the 256-QAM, while the second chain includes the 8-QAM, the 32-QAM and the 128-QAM.

Recall the hidden Markov model for the alphabet $\mathcal{X}$ of size d:

$$\begin{aligned} a_k &\sim \mathcal{U}(\mathcal{X}) \\ y_k &= \sum_{l=0}^{L-1} a_{k-l} h_l + \varepsilon_k, \quad \varepsilon_k \sim \mathcal{N}_{\mathbb{C}}\left(0, \sigma^2\right), \end{aligned} \tag{6.3}$$

where $\mathcal{U}(\mathcal{X})$ describes a uniform distribution on $\mathcal{X}$.

We define the random variable $\tilde{a}_k$ as the sum of a random variable $z_k \in \mathcal{X}_M$ and of a variable $\omega_k \in \mathcal{X}_R$:

$$\begin{aligned} \tilde{a}_k &= z_k + \omega_k \\ z_k &\sim \mathcal{U}\left(\mathcal{X}_M\right) \\ \omega_k &\sim \mathcal{U}(\mathcal{X}_R), \end{aligned} \tag{6.4}$$

such that a_k and $\tilde{a}_k$ follow the same distribution. The Model (6.3) is thus equivalent to

$$\begin{aligned} z_k &\sim \mathcal{U}\left(\mathcal{X}_M\right), \\ \omega_k &\sim \mathcal{U}(\mathcal{X}_R), \\ y_k &= \sum_{l=0}^{L-1} z_{k-l} h_l + \sum_{l=0}^{L-1} \omega_{k-l} h_l + \varepsilon_k, \quad \varepsilon_k \sim \mathcal{N}_{\mathbb{C}}\left(0, \sigma^2\right). \end{aligned}$$

We define

$$Z_k = \sum_{l=0}^{L-1} z_{k-l} h_l$$

and

$$W_k = \sum_{l=0}^{L-1} \omega_{k-l} h_l.$$

Note that the hidden Markov model (z_k, y_k) is not equivalent to a rescaled QAM model of size d_M since $W_k + \varepsilon_k$ is not normally distributed. On the other hand, the influence of W_k is smaller by an order of magnitude than Z_k. For example, for the example of the 64-QAM with a macro-states alphabet of 16 points, the average energy $\mathbb{E}\left[\|Z_k\|^2\right]$ is 20 times larger than $\mathbb{E}\left[\|W_k\|^2\right]$. By estimating the channel coefficients in a rescaled QAM model of size d_M we introduce thus a small bias, since we changed the model, but on the other hand we are considering a much smaller model and the variance of the channel estimator will therefore decrease. We demonstrate this by simulations in Section 6.2.6. Furthermore, the computational complexity decreases significantly as well.

We propose to use the macro-states model to improve the initial estimate of the channel coefficients in the 'true' model with alphabet $\mathcal{X}$ by using the (rescaled) estimate of the channel that has been obtained in the biased macro-states model with alphabet $\mathcal{X}_M$. Exemplarily for a 4-QAM (we denote the alphabet $\mathcal{X}_4$), a 16-QAM ($\mathcal{X}_{16}$) and a 64-QAM ($\mathcal{X}_{64}$), we present the Alg. 5 - the complete model Classification algorithm. To keep the notations concise, we have assumed a fixed channel order L. We use the notation $\hat{l}^i_j$ for the likelihood estimate of the sequence Y in the model j-QAM and for the ith initialization of the approximate EM algorithm and equivalently $\hat{h}^i_j$ for the channel estimate and $\hat{\sigma}^i_j$ for the estimate of the standard deviation. Note, that for a fixed channel order the BIC criterion of the model estimator reduces to maximizing the likelihoods in each model since the penalty term is constant for each model.

Algorithm 5: Macro-States Estimation Chain

```
Input  : $Y$, $L$, $I$, $N$
Output: $\hat{\sigma}$, $\hat{h}$, $\hat{l}$   Parameter and likelihood estimate

• Estimation 4-QAM;
    ∗ Initial Values: For $i \in \{0, \cdots, I-1\}$, create $\tilde{h}^i_4$ and $\tilde{\sigma}^i_4$.
    ∗ Approximate EM algorithm : run Algorithm 4 with
      Input  : $I$, $(\tilde{h}^i_4, \tilde{\sigma}^i_4)_{i<I}$, $y_{0:K}$, $L$, $N$, $\mathcal{X}_4$
      Output: $\hat{l}^i_0(y_{0:K})$, $\hat{h}^i_4$, $\hat{\sigma}^i_4$ for $i \in \{0, \cdots, I-1\}$
• for $j \in \{16, 64\}$ (16-QAM, 64-QAM) do
        ∗ Initial values: For $i \in \{0, \cdots, I-1\}$, put $\tilde{h}^i_j = 0.5\hat{h}^i_{j/4}$ and create $\tilde{\sigma}^i_j$.
        ∗ Approximate EM algorithm : run Algorithm 4 with
          Input  : $I$, $(\tilde{h}^i_j, \tilde{\sigma}^i_j)_{i<I}$, $y_{0:K}$, $L$, $N_j$, $\mathcal{X}_j$
          Output: $\hat{l}^i_j(y_{0:K})$, $\hat{h}^i_j$, $\hat{\sigma}^i_j$ pour $i \in \{0, \cdots, I-1\}$
  end
• Maximization of the model ;
    ∗ Maximization over the different initializations for $j \in \{4, 16, 64\}$:
      Calculate
          $\hat{\imath}_j = \arg\max_{i<I} \hat{l}^i_j.$
      Put $\hat{l}_j(y_{0:K}) = \hat{l}^{\hat{\imath}_j}_j$, $\hat{h}_j = \hat{h}^{\hat{\imath}_j}_j$ and $\hat{\sigma}_j = \hat{\sigma}^{\hat{\imath}_j}_j$.
    ∗ Model maximization:
      Calculate
          $\hat{\jmath} = \arg\max_{j \in \{4,16,64\}} \hat{l}_j.$
      Put $\hat{l} = \hat{l}_{\hat{\jmath}}$, $\hat{h} = \hat{h}_{\hat{\jmath}}$, $\hat{\sigma} = \hat{\sigma}_{\hat{\jmath}}$ and $\hat{\mathcal{M}}$ = QAM-$\hat{\jmath}$.
```

6.2.6 Simulations Results

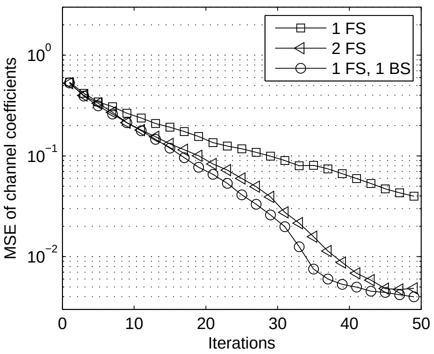

Figure 6.2: Convergence of channel coefficients, 16-QAM, $L = 3$, for different number of initializations of the approximate EM, FS = Forward Fixed-Interval Smoothing, BS = Backward Fixed-Interval Smoothing

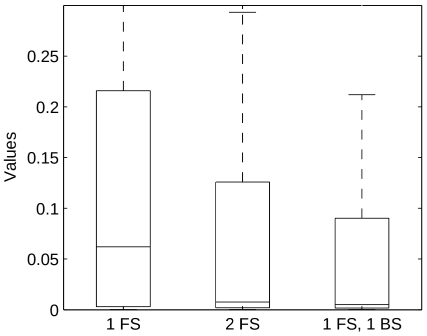

Figure 6.3: MSE of channel coefficients after 40 iterations, 16-QAM, $L = 3$, for different number of initializations of the approximate EM, FS = Forward Fixed-Interval Smoothing, BS = Backward Fixed-Interval Smoothing

We consider first of all the performance of the backward fixed-interval smoothing (BS) algorithm compared to the forward fixed-interval smoothing (FS) in terms of the convergence of approximate EM algorithm incorporating the smoothing algorithms. It is measured in terms of the mean squared error $\text{MSE}(\hat{h}) = \mathbb{E}\left(\sum_{l=0}^{L-1}\left|\hat{h}_l - h_l\right|^2\right)$ of the current estimate of the channel coefficients $\hat{h}$ compared to the true coefficients h. We only consider the MSE of the channel coefficients and do not include the standard deviation. The main issue was the convergence of the channel, since the convergence of the standard deviation did not pose any serious problems. Furthermore, a good channel estimation is more vital for the subsequent symbol detection than the standard deviation. The alphabets of the modulation schemes are normalized such that the averaged energy of the symbols is equal to one. We fix the number of observations to $K = 300$, for which the assumption of a constant channel is reasonable.

The approximate EM algorithm has been initialized a varying number of times such that we are able to compare the performance of only using the FS compared to replacing the FS by the BS for half of the initializations. The convergence of the channel coefficients in a 16-QAM modulation with $L = 3$ is given in Fig. 6.2. The channel coefficients have been generated randomly for each Monte-Carlo run, such that the channel has been about as often in maximum

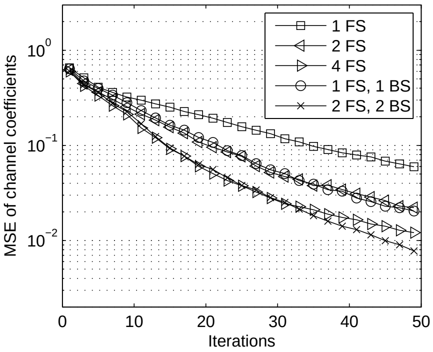

Figure 6.4: Convergence of channel coefficients, 64-QAM, $L = 4$, for different number of initializations of the approximate EM, FS = Forward Fixed-Interval Smoothing, BS = Backward Fixed-Interval Smoothing

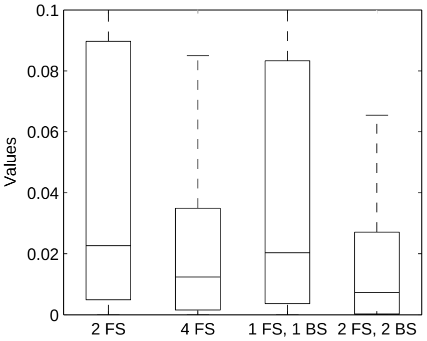

Figure 6.5: MSE of channel coefficients after 50 iterations, 64-QAM, $L = 4$, for different number of initializations of the approximate EM, FS = Forward Fixed-Interval Smoothing, BS = Backward Fixed-Interval Smoothing

phase as in minimum phase. Already in this small and relatively easy model, the convergence of the EM algorithm improves by using one time the BS and one time the FS instead of two times the FS. The variance of the MSE after convergence, see Fig. 6.3 is also smaller if the BS has been used one time.

The differences are even more significant in a more complex model with a 64-QAM modulation scheme and channel order $L = 4$. In Fig. 6.4, we display as well the convergence if the FS has been used four times and the FS and BS have been each used two times. The last choice is clearly superior in terms of the average MSE, as well as in terms of the variance of the MSE after convergence (Fig. 6.5.

Now we examine the performance of the macro-states model. As mentioned in Section 6.2.5, we use the macro-states model to improve the initial parameter estimate for the estimation in the true model. Since we consider the estimation of the unknown parameters as part of the model Classification, it is necessary to derive a channel estimate in each potential model anyway. If the a macro-states model of the true model is as well a potential model, using the macro-states channel estimate as initial value in the true model does not increase the computational complexity. Fig. 6.6 shows the convergence of the channel coefficients in terms of the MSE for the true constellation being a 64-QAM and a macro-states model consisting of 16 states. We fixed again $K = 300$, $L = 4$ and used channel coefficients randomly chosen in each Monte-Carlo run. The first half of the iterations correspond to the estimation in the macro-states model.

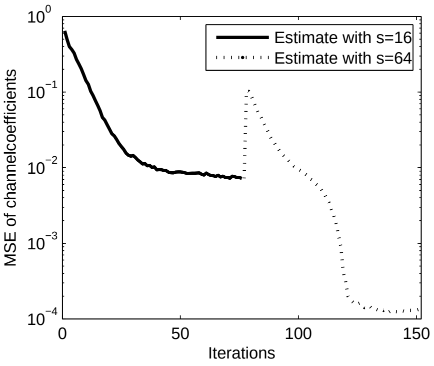

Figure 6.6: Convergence of channel coefficients, true constellation: 64-QAM, firstly: 75 iteration in macro-states model (16-QAM), then 75 iterations in true model, $N = 500$

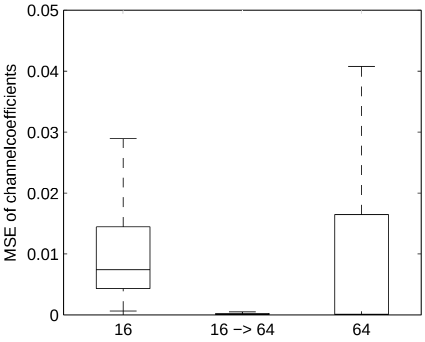

Figure 6.7: MSE after convergence, true constellation: 64-QAM, $N = 500$, (16) in macro-states model (16-QAM), (16->4) firstly macro-states model, then true model, (64) in true model

The successive iterations are conducted in the true model. Already in the macro-states model, the channel estimate converges closely to the true channel coefficients. At the switch over to the true model, the MSE is increasing. This might be due to the introduced bias of the macro-states model.

Fig. 6.7 shows the MSE after the convergence of the approximate EM and supports the previous figure. It can be seen that the macro-states model (corresponding to the left hand bar) introduces a small bias, although the MSE is already very small. On the other hand, the variance of the MSE is much smaller compared to the pure estimation in the true model (corresponding to the right hand bar). The combination of the prior estimation in the macro-states model and a subsequent estimation in the true model (corresponding to the middle bar) combines the advantage of a small variance and the unbiasedness, i.e. the performance is clearly superior to the pure estimation in the true model.

We repeated the same experiment in a 16-QAM model with a macro-states model corresponding to a 4-QAM model. The results, see Fig. 6.8 coincide with the previous figures, although the 16-QAM model is already sufficiently robust, such that the variance of the pure estimation in the true model (corresponding to the right hand bar) is not as large as in the 64-QAM model. The bias in the macro-states model is again very small, the channel coefficients are already closely approximated

We turn now to the Blind Classification of the modulation scheme. Since to our knowledge this problem has not yet been satisfactorily covered in the literature, we are not able to compare our algorithm to existing methods. In the first simulation we have considered the 4-QAM, 16-

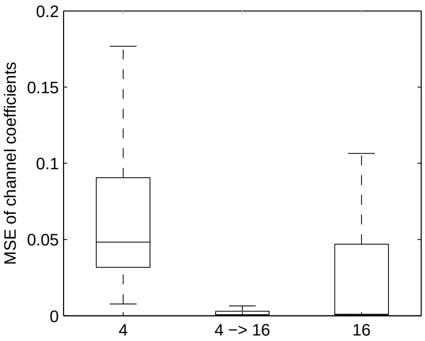

Figure 6.8: MSE after convergence, true constellation: 16-QAM, $N = 500$, (4) in macro-states model (4-QAM), (4->16) firstly macro-states model, then true model, (16) in true model

	True model					
	4-QAM	16-QAM	32-QAM	64-QAM	128-QAM	256-QAM
Probability	98.29 %	98.33 %	65.84 %	99.19%	90.60%	73.80 %

Table 6.1: Model Classification, BUx, $K = 300$, percentage of correct estimation of the true model

QAM, 32-QAM, 64-QAM, 128-QAM and 256-QAM models as potential modulation schemes. The transmission model and data simulation correspond to a bad urban environment model (BUx) defined in COST207 [1989], ETSI [2002]. We have fixed the channel order to $L = 3$, which was sufficient to cover the important peaks of the channel impulse response. The length of the observation sequence was again chosen to $K = 300$. Increasing the number of observations does in fact not significantly improve the estimation of the channel, as we verified in simulations. The SNR was chosen such that the theoretic BER over a AWGN channel is equal to 1e-3 in each model. We used 4 initializations and 60 maximal iterations for the approximate EM algorithm. Furthermore, the Particle Filtering algorithm was given $N = 50$ for the small modulations (16-QAM, 32-QAM) and $N = 200$ for the remaining, larger modulations schemes, while we used the plain EM algorithm coupled with the Baum-Welch algorithm in the 4-QAM model.

We evaluated the performance in the terms of the probability to estimate the correct model, i.e. in Table 6.1 we give the percentage of simulations for which the true model is chosen by the model estimator. The probability of correct estimation in the 4-QAM, 16-QAM and 64-QAM model is very high, while it is much lower for the 32-QAM modulation. This is mainly due to the fact, that we do not use a macro-states model to improve the initial parameter estimate and we use as well a small particle size for the 32-QAM.

6.3 Generalized Likelihood Ratio Tests Based on Proportions Parameter

We will now place ourselves in the context of flat-fading single-path channels such that the model reduces to a Gaussian mixture model. The model estimation may then be interpreted as testing the number of relevant components of the mixture. We introduce new test statistics that tackle the problem from a different point of view, namely a parameter estimating problem, as we described in the introduction of this chapter.

In the remainder, we present the model, followed by a description of the three test statistics including a justification of the asymptotic distributions of the second and the third test. We will then mention briefly the estimation via an expectation-maximization (EM) algorithm Dempster et al. [1977] of the unknown parameters including the proportions. The information matrix is approximated by Monte-Carlo simulations. We conclude with a numerical comparison of the three test statistics, including a discussion on the sensitivity of the choice of initial parameter values for the EM algorithm when including the proportions as an unknown parameter.

For the ease of presentation we present the methods mainly for the example of testing a 4-QAM model versus a 16-QAM model. But it is straightforward to use these statistics for other QAM models. It is as well straightforward to extend the test statistics to model estimation when more than two potential models are given. Furthermore, the presented statistics extend as well to the case of frequency-selective multi-path channels. However in multi-path channels, the underlying model is hidden Markov model (HMM) and not much is known about theoretical properties of statistical tests in general HMMs. First approaches are for example by Gassiat and Boucheron [2003], Bickel et al. [1998], Giudici et al. [2000].

6.3.1 Model Description

We consider two potential QAM constellations, the smaller one having the alphabet $\mathcal{X}_s$ consisting of d_s and the larger constellation with symbol alphabet $\mathcal{X}_l$ of d_l states. We assume that d_l is a multiple of d_s. For the ease of presentation, the alphabets are not normalized such that $\mathcal{X}_s$ is a subset of the larger alphabet $\mathcal{X}_l$, i.e. $\mathcal{X}_s \subset \mathcal{X}_l$.

The state space $\mathcal{X}$ of our model is given by the larger alphabet $\mathcal{X}_l = \mathcal{X}$. In a linear modulation scheme, each symbol in the alphabet is transmitted with the same probability (we do not consider coding or consider the coding scheme to be unknown). We generalize this and define two subsets $\mathcal{X}_0$ and $\mathcal{X}_1$ of $\mathcal{X}_l$, where the subsets have different probabilities or proportions for the submission of their corresponding symbols. We define $\mathcal{X}_0 = \mathcal{X}_s$, i.e. the smaller alphabet, and $\mathcal{X}_1 = \mathcal{X}_l \setminus \mathcal{X}_s$. The two subsets are thus of order $d_0 = d_s$ and $d_1 = d_l - d_s$.

At time step k, the symbol s_k is drawn from $\mathcal{X}$ with probabilities equal to $\frac{p_0}{d_0}$ for all states in $\mathcal{X}_0$ and $\frac{p_1}{d_1}$ for all states in $\mathcal{X}_1$. More precisely, $s_k \in \mathcal{X}_0$ with probability p_0 and in $\mathcal{X}_1$ with probability p_1. All the symbols in $\mathcal{X}_0$ are equiprobable, as well as all states in $\mathcal{X}_1$. For this to be meaningful, we assume

(A1) $p_0, p_1 \geq 0$, and

(A2) $p_0 + p_1 = 1$.

As we do not consider coding schemes, the symbols at different time steps are considered to be independent. The observation at time step k is the sum of the faded symbol and an independent Gaussian noise term:

$$y_k = hs_k + \varepsilon_k,$$

where h is the scaling or fading coefficient of the channel and $\varepsilon_k \sim \mathcal{N}_{\mathbb{C}}(0, \sigma^2)$ has a normal complex distribution with zero mean and variance σ^2.

The vector of unknown parameters θ consists thus of the proportion p_1, the channel coefficient h and the standard deviation of the noise σ. The parameter space is denoted Θ.

The density of the distribution of the observation y conditional to the current state s for a given θ may be expressed as

$$g\left(y|s;\theta\right) = \frac{1}{\pi\sigma^2} \exp\left(-\frac{1}{\sigma^2}\left|hs - y\right|^2\right).$$

The unconditional distribution of observation y_k is hence a mixture of Gaussian distributions with density function

$$f(y,\theta) = \frac{p_0}{d_0} \sum_{s\in\mathcal{X}_0} g\left(y|s;\theta\right) + \frac{p_1}{d_1} \sum_{s\in\mathcal{X}_1} g\left(y|s;\theta\right) \tag{6.5}$$

for $y \in \mathbb{C}$.

In the remainder, a sequence of K observations $y_{1:K} = (y_1, \cdots, y_K)$ is fixed.

Example 2 *4-QAM and 16-QAM*

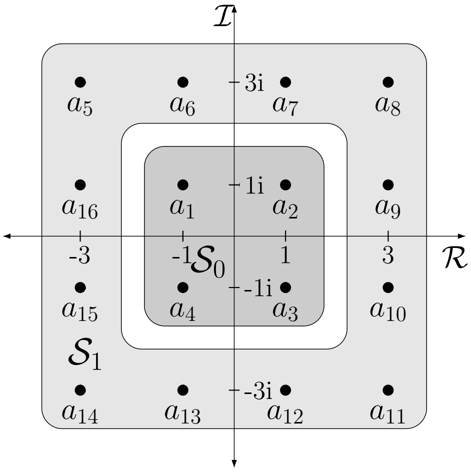

Figure 6.9: 16-QAM constellation split up in $\mathcal{X}_0$ and $\mathcal{X}_1$

We consider in more details the example of a testing a 4-QAM constellation versus a 16-QAM constellation. Then the alphabets become

$$\mathcal{X}_0 = \{1+i,\ 1-i,\ -1-i,\ -1+i\}$$

with $d_0 = 4$ *and*

$$\mathcal{X}_1 = \{\pm a \pm bi :\ a,b \in \{1,3\}, a = 3 \vee b = 3\}.$$

with $d_1 = 12$ *as given in Figure 6.9.*

Hence, to generate a signal from the 4-QAM constellation, we set $p_0 = 1$ *and* $p_1 = 1$*. A 16-QAM signal is created by setting* $p_0 = 1/4$ *and* $p_1 = 3/4$*.*

6.3.2 Likelihood Ratio Tests

As before, let $\mathcal{X}_s$ denote the alphabet of symbols of the small constellation (d_s-QAM) and $\mathcal{X}_l$ the states of the large constellation (d_l-QAM).

If the true model corresponds to the d_l-QAM, than $p_0/d_0 = p_1/d_1$, while we have $p_1 = 0$ if it corresponds to the smaller constellation. Based on these proportions we may now define the test statistics to test for the constellation size .

The likelihood as a function of $\theta \in \Theta$ given a sequence of observations $y_{0:K}$ is given by

$$L_K(\theta) = \prod_{k=1}^{K} f(y_k, \theta) \tag{6.6}$$

We denote the log-likelihood by

$$l_K(\theta) = \log L_K(\theta). \tag{6.7}$$

In the following we will present three possible likelihood ratio tests based on (6.7). Before that we establish the consistency of the Maximum Likelihood (ML) estimator for all the possible constellations.

Consistency/Identifiability of the ML Estimator

We will first show the identifiability of the ML estimator in Model 6.2.1 under the assumption that the true model corresponds to the small constellation d_s-QAM. For the ease of presentation, we assume that $d_s = 4$ and $d_l = 16$. The same holds for the inverse case, when the true constellation is the larger one. We will state that afterwards. The proof is analogous and omitted for brevity.

Definition 5 *The Maximum Likelihood (ML) estimator $\hat{\theta}_K$ in Model 6.2.1 for a given sequence $y_{1:K}$ is given by*

$$\hat{\theta}_K = \arg\max_{\theta \in \Theta} l_K(\theta)$$

Let $f_T(y) = f(y, \theta_T)$ for $y \in \mathbb{C}$ denote the density of the distribution of y_k in the true model. It is well known that for $\theta \in \Theta$

$$l_K(\theta) = \frac{1}{K} \sum_{k=1}^{K} \log f(y_k, \theta) \underset{\mathbb{P}_T}{\longrightarrow} l(\theta) \tag{6.8}$$

where $\underset{\mathbb{P}_T}{\longrightarrow}$ denotes convergence in probability in the true model and

$$\begin{aligned} l(\theta) &= \int \log\big(f(y,\theta)\big) f_T(y)dy = \mathbb{E}_{\mathbb{P}_T}\left[\log f(y_k, \theta)\right] \\ &= -\operatorname{KL}\left(f_T \| f(\cdot,\theta)\right) + \int \log\big(f_T(y)\big) f_T(y)dy, \end{aligned} \tag{6.9}$$

where the second term does not depend on θ and $\operatorname{KL}(f_T \| f(\cdot,\theta))$ denotes the Kullback-Leibler divergence between f_T and $f(\cdot,\theta)$. The function $\theta \mapsto l(\theta)$ is thus maximal at θ^* if $f_T = f(\cdot, \theta^*)$.

Theorem 6 *Let the true model be a d_s-QAM with parameter $\theta_T = (p_{1,T}, h_T, \sigma_T)$ where $p_{1,T} = 0$. Then the estimator $\hat{\theta}_K$ is consistent, i.e. $\hat{\theta}_K$ converges in probability to θ_T.*

Proof: Let $a_s \in \mathcal{X}_s$. In order to assure $f_T = f(\cdot, \hat{\theta}_K)$, there must exist one point $a_l \in \mathcal{X}_l$, such that $a_l h^* = a_s h_T$, i.e. one mode of $f(\cdot,\theta^*)$ has to coincide with the mode of f_T corresponding to a_s. Observe that as soon as this state together with θ^* is given, all other modes of $f(\cdot,\theta^*)$ are automatically determined. Assume $a_l \in \mathcal{X}_1$, i.e. a_l is in the the outer ring. But then $\operatorname{KL}(f_T \| f(\cdot,\theta^*)) > 0$, since the mixture weight p_1 corresponding to this mode is not larger than $\frac{1}{12}$, while the mixture weight in f_T is equal to $\frac{1}{4}$. Hence, $a_l \in \mathcal{X}_0$, but then $\theta^* = \theta_T$, apart from symmetries.

Without loss of generality, we may impose an upper limit on the channel and standard deviation, such that θ^* is a global maximum of $l(\theta)$.

Following the arguments in Andrews [1999], the convergence in (6.8) together with the global maximization yields a sufficient condition for the consistency. ■

This reveals that the consistency depends crucially on the number of proportion parameters and the choice of the sets $\mathcal{X}_0$ and $\mathcal{X}_1$. For example, the estimator is not longer consistent, if each state $a_i \in \mathcal{X}$ is assigned an own proportion parameter p_i. Neither is it possible to assign one proportion parameter to the four inner points, one to the four outer corner points and one to the remaining points.

We now turn to the consistency of the estimation if the true model is an $\mathcal{X}_l$-QAM.

Theorem 7 *Let the true model be a d_l-QAM with parameter $\theta_T = (p_{1,T}, h_T, \sigma_T)$ where $p_{1,T} = \frac{d_l}{d}$. Then the estimator $\hat{\theta}_K$ is consistent, i.e. $\hat{\theta}_K$ converges in probability to θ_T.*

As mentioned before, the proof is equivalent to the proof of Theorem 6 and omitted for brevity.

Simple Hypothesis Test

The first test statistic we consider is a simple GLRT statistic. The null-hypothesis is defined by

$$\mathrm{H}_0 : p_1 = 0$$

corresponding to the d_s-QAM model. It is tested versus the alternative hypothesis

$$\mathrm{H}_1 : p_1 = \frac{d_l}{d}$$

corresponding to a d_l-QAM.

The ML estimates corresponding to both hypotheses are given by

$$\hat{\theta}_{K,0} = \arg\max_{\theta \in \Theta : p_1 = 0} l_K(\theta)$$

and

$$\hat{\theta}_{K,1} = \arg\max_{\theta \in \Theta : p_1 = \frac{d_l}{d}} l_K(\theta).$$

The simple generalized likelihood ratio test statistic is then defined by

$$T_K^1 = l_K(\hat{\theta}_{K,0}) - l_K(\hat{\theta}_{K,1}). \tag{6.10}$$

Unfortunately the asymptotic distribution of T_K^1 is not easily accessible such that it is not possible to calibrate the test. This test statistic is equivalent to testing the two QAM models as presented in Hong [2006] for BSPK and QPSK symbols. Marginalization of Theorems 6 and 7 provide however the consistency of this hypothesis test, which has not been considered in Hong [2006].

Composite Hypothesis Test with $H_0 : p_0 = p_1$

The second test is a composite generalized likelihood ratio test with null hypothesis

$$\mathrm{H}_0 : p_0 = \frac{1}{d_0}$$

and alternative hypothesis

$$\mathrm{H}_1 : p_0 \neq \frac{1}{d_0}.$$

The null hypothesis corresponds thus to the d_l-QAM which is tested versus the hypothesis that the true model does not coincide with the d_l-QAM.

The generalized likelihood ratio test compares the likelihood under the null hypothesis to the likelihood in the general model, i.e. for $\theta \in \Theta$, see Lehmann and Romano [2005] for an in-depth introduction.

The corresponding ML estimates are thus given by

$$\hat{\theta}_{K,0} = \arg\max_{\theta \in \Theta : p_0 = \frac{1}{d_0}} l_K(\theta)$$

and

$$\hat{\theta}_K = \arg\max_{\theta \in \Theta} l_K(\theta).$$

The test statistic is given by

$$T_K^2 = l_K(\hat{\theta}_K) - l_K(\hat{\theta}_{K,0}). \tag{6.11}$$

Contrarily to T_K^1, the asymptotic distribution of T_K^2 under the null hypothesis is well known by the Theorem 12.4.2 in Lehmann and Romano [2005].

Theorem 8 *Under the null hypothesis,*

$$2\,T_K^2 \xrightarrow{d} \chi_1^2,$$

where χ_1^2 denotes a random variable with central chi-squared distribution with one degree of freedom.

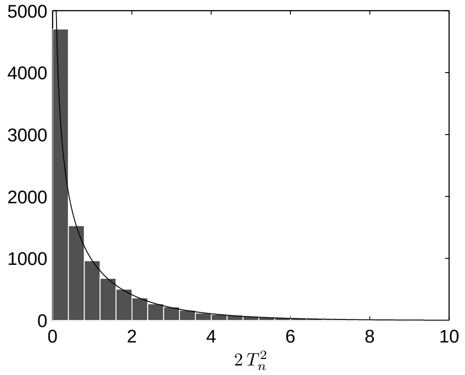

Figure 6.10: Asymptotic Distribution of $2\,T_K^2$, true model: 4-QAM, $d_s = 4$, $d_l = 16$, histogram for $K = 10000$ and SNR $= 4dB$, line corresponds to χ_1^2-distribution

Composite Hypothesis Test with $H_0 : p_1 = 0$

The last test statistic considers the d_s-QAM as the null hypothesis, i.e.

$$\mathrm{H}_0 : p_1 = 0.$$

The alternative hypothesis then becomes

$$\mathrm{H}_1 : p_1 \neq 0.$$

The corresponding ML estimates are in this case given by

$$\hat{\theta}_{K,0} = \arg\max_{\theta\in\Theta:p_1=0} l_K(\theta)$$

and

$$\hat{\theta}_K = \arg\max_{\theta\in\Theta} l_K(\theta).$$

The test statistic writes

$$T_K^3 = l_K(\hat{\theta}_K) - l_K(\hat{\theta}_{K,0}). \tag{6.12}$$

In contrast to T_K^2, the Theorem 12.4.2 of Lehmann and Romano [2005] does not apply to T_K^3, since the true parameter p_1 lies on the boundary of the parameter space. The asymptotic distribution is a mixture of a chi-squared distribution and a distribution with point mass in zero as in Andrews [1999, 2001], Shapiro [1985].

Theorem 9 *Under the null hypothesis,*

$$2\,T_K^3 \xrightarrow{d} \bar{\chi}^2.$$

The random variable $\bar{\chi}^2$ is given by

$$\bar{\chi}^2 = \frac{1}{2}\left(\chi_1^2 + \chi_0^2\right),$$

where χ_1^2 is a random variable with a central chi-squared distribution with one degree of freedom and where $\chi_0^2 \equiv$ has unit mass at zero.

Proof: The result follows from Theorem 4 in Andrews [2001]. It suffices to verify the necessary assumptions.

1. The consistency follows from Theorem 6.

2. The score Z_{n,p_1} with respect to p_1 is given by

$$Z_{n,p_1}(\theta) = \frac{\partial}{\partial p_1} l_K(\theta) = \sum_{k=1}^{K} \frac{1}{f(y_k,\theta)} \left(-\frac{1}{d_0} \sum_{s \in \mathcal{X}_0} g_(\,(,|s;\,)\,y_k,\theta) + \frac{1}{d_1} \sum_{s \in \mathcal{X}_1} g_(\,(,|s;\,)\,y_k,\theta) \right). \tag{6.13}$$

   This holds as well for the right partial derivative on the boundary with $\theta = (0, h, \sigma)$. We have thus that $\mathbb{E}_0[Z_{n,p_1}(\theta_T)] = 0$, where $\mathbb{E}_0$ denotes expectation under H_0 and $\theta_T = (0, h_T, \sigma_T)$ the true values under H_0.

3. The Assumption 2^{2*} in Andrews [2001] is verified since the domain of $l_K(\cdot)$ is already a union of orthants. We have omitted to provide the explicit expressions for the partial derivatives of order 2, but these are also continuous and even differentiable, such that $2^{2*}(c)$ is verified. The function $l_K(\cdot)$ has thus a quadratic expansion in θ about θ_T.

4. Assumption 3* follows from the convergence in distribution of the normalized score and of the information matrix. Let Z_{p_1} denote the limit of $\sqrt{K} Z_{n,p_1}$ for $K \to \infty$ and $\mathcal{I}_{p_1}$ the limit of

$$\mathcal{I}_{K,p_1} = -\frac{1}{K} \frac{\partial^2}{\partial^2 p_1} l_K(\theta).$$

5. Assumption 5 to 9 assure that the parameter space Θ is locally approximated by a product of convex cones. This holds obviously since Θ suffices these conditions.

6. Assumption 10 holds obviously since all parameters are identified under H_0.

Following Theorem 4 in Andrews [2001], we have thus that

$$T_K^3 \xrightarrow{d} \max\{Z_{p_1}, 0\} \mathcal{I}_{p_1} \max\{Z_{p_1}, 0\} = \left(\max\{\sqrt{\mathcal{I}_{p_1}} Z_{p_1}, 0\} \right)^2. \tag{6.14}$$

Since Z_{p_1} is normally distributed with mean 0 the asymptotic distribution is the mixture with equal weights of a chi-squared distribution and a point mass at 0.

■

Note that a theoretical extension of the parameter space, i.e. allowing $p_1 < 0$, may imply that that the maximum of the likelihood function is not attained in 0 but for a negative value even if the true value is $p_1 = 0$. This is due to the fact, that $p_1 < 0$ allows $p_0 > 1$ and if the SNR is large enough, the observations will be close to the points corresponding to $\mathcal{X}_0$ and the overall likelihood will thus be higher. Figure 6.11 displays the likelihood depending on p_1 where we assume the channel and standard deviation to be known. The SNR is 8dB and the number of observations $K = 10000$. The figure shows that the optimal value in the actual parameter space is as expected at $p_1 = 0$, while the likelihood is almost linearly increasing on the extended parameter space for $p_1 < 0$. Only when the number of observations becomes very large ($K = 1e6$) and by focussing on an interval very close to $p_1 = 0$ (i.e. $-6e-4 < p_1 < 6e-4$),

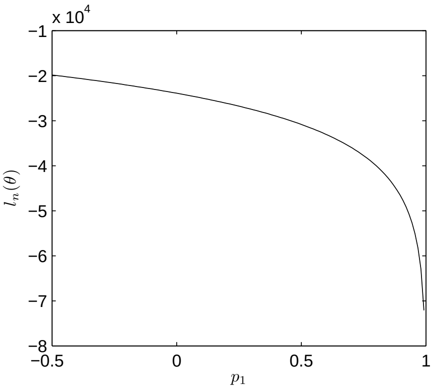

Figure 6.11: Log likelihood as function of p_1, true model: $p_1 = 0$, $K = 1e4$ and SNR $= 8dB$

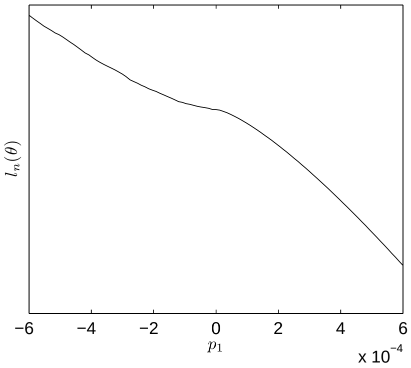

Figure 6.12: Log likelihood as function of p_1 close to $p_1 = 0$, true model: $p_1 = 0$, $K = 1e6$ and SNR $= 0dB$

one can see that there is a critical point at $p_1 = 0$, see Figure 6.12. This coincides thus with the expected score being equal to 0 in $p_1 = 0$.

For smaller sample sizes and on the whole parameter space, this critical point is not significant, such that the empirical score is rather negative than equal to 0. Hence, the mixing weight corresponding to χ_0^2 in Theorem 9 will be larger than that of χ_1^2, since the empirical version of Z_{p_1} in (6.13) for small sample sizes will be often negative.

Figure 6.13 shows the empirical cumulative distribution function (CDF) obtained in 10^6 Monte-Carlo runs of T_K^3 in comparison to the CDF of $\bar{\chi}^2(\theta_T) = 0.5\chi_0^2 + 0.5\chi_1^2$. The empirical CDF in this case corresponds closely to the mixture $0.9\chi_0^2 + 0.1\chi_1^2$. A threshold for the test set according to the asymptotic distribution of the test statistic is thus a conservative choice. The level of the test for smaller sizes will even be higher. Furthermore, this simplifies the maximum likelihood estimation of the parameters in comparison to Test 2, see Section 6.3.3, since the likelihood function around the true parameter value is more peaked. Hence, the EM algorithm converges in less iterations.

6.3.3 ML Estimation with the EM Algorithm

An explicit solution for the maximization of the function $\theta \mapsto l_K(\theta)$ does not exist. The expectation-maximization (EM) algorithm Dempster et al. [1977] is however a well-known, it-

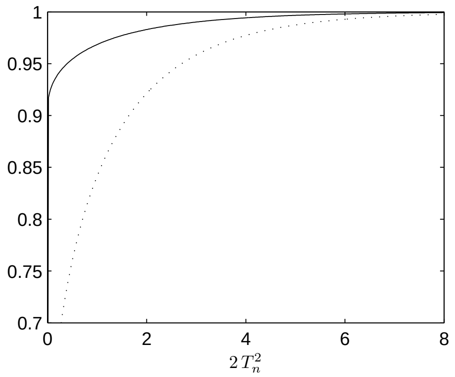

Figure 6.13: Empirical CDF of $2\,T_K^3$ $(-)$ and CDF of $\bar{\chi}^2(\theta_T)$ $(\cdots)$, $K = 10000$ and SNR $= 4dB$, true model: $p_1 = 0$

erative algorithm to derive an ML estimate in incomplete date models, see for example Cappé et al. [2007]. It maximizes iteratively the intermediate quantity $Q(\theta|\theta^{(t)}) = \mathbb{E}_{\theta^{(t)}}\left[l_K(\theta)\right]$ where $\theta^{(t)}$ denotes the parameter estimate at the t-th iteration.

The EM algorithm splits up into two steps, the expectation (E) step and the maximization (M) step. In the E step, the intermediate quantity $Q(\cdot|\theta^{(t)}$ is calculated for a given parameter estimate $\theta^{(t)}$. In the mixture model, i.e. a flat-fading channel, this is very fast and direct. At each time step k, the likelihood function $g_(\,(,|s;\,)\,y_k, \theta^{(t)})$ is calculated for each symbol s in the state space $\mathcal{X}$. If the channel has however multiple paths and is frequency selective, then the model is an HMM and the E step becomes more involved. It can be implemented by the Baum-Welch algorithm Baum et al. [1970] or a Particle Smoothing algorithm Doucet et al. [2000].

The M step consists of maximizing the function $Q(\cdot|\theta^{(t)}$. The maximization with respect to the channel coefficient and the standard deviation is well known and given in Chapter 5. We now establish the maximization with respect to the new proportion parameter p_0. Define

$$D_i\big(\theta^{(t)}\big) = \sum_{b_j \in \mathcal{X}_i} \sum_{k=1}^{K} \mathbb{P}\left(s_k = b_j | y_k, \theta^{(t)}\right)$$

for $i \in \{0, 1\}$. Then

$$\mathbb{E}_{\theta^{(t)}}\left[l_K(\theta)\right] = \log p_0 D_0\big(\theta^{(t)}\big) + \log p_1 D_1\big(\theta^{(t)}\big) + \text{const},$$

where the constant combines the terms that do not involve p_0 and p_1. The optimal proportions are then given by

$$p_0^{(t+1)} = \frac{D_0\big(\theta^{(t)}\big)}{d_0}$$

and

$$p_1^{(t+1)} = \frac{D_1\big(\theta^{(t)}\big)}{d_1}.$$

When discussing the EM algorithm, it is inevitable to discuss its sensibility to the initial choice of the parameters, i.e. its convergence to local maxima different from the global maximum if inadequately initialized. For example in the case where $p_1 = 0$, the likelihood function $l_K(\theta)$ has two clear distinct local maxima, see Figure 6.14. The standard deviation has been marginalized out for better visualization. The global maximum is at the true value, i.e. $h = 1$ and $p_1 = 0$, whereas there is a second local maximum at $h = 1/3$ and $p_1 = 1/12$. This maximum

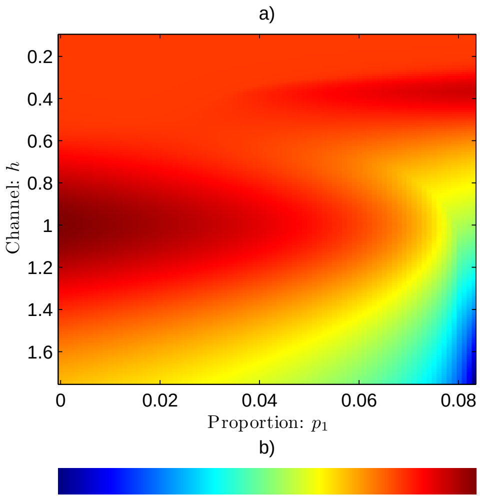

Figure 6.14: a) Likelihood function $l_K(\theta)$ as function of h and p_1, $K = 10000$ and SNR $= 8dB$, true model: $p_1 = 0$, b) Colormap from blue (small likelihood) to red (large likelihood)

corresponds to the case of rescaling the 16 states such that the four outer corner points coincide with the four true states. But then $p_1 \leq 1/12$ and it is clear that the corresponding likelihood is smaller than for the global maximum (with $p_0 = 1/4$).

6.3.4 Computational Results on Classification Performances

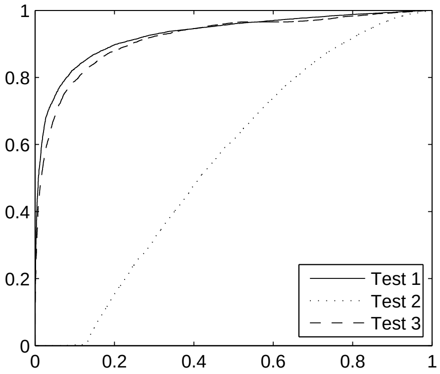

Figure 6.15: ROC curve of test for $K = 50$ and SNR $= 0dB$

In this section we evaluate the capability of Classification with the help of receiver operating characteristic (ROC) curves. A ROC curve has the advantage to yield an performance comparison without knowledge of the asymptotic properties of the test statistic. This enables us to compare the three tests for finite sample sizes with the help of Monte-Carlo experiments.

We consider testing the 4-QAM constellation versus the 16-QAM constellation. This means however for the composite tests statistics T_K^2 and T_K^3 that the performance of these tests is evaluated by considering one simple alternative as an example of all possible alternatives. For example for T_K^2, we consider $p_1 = 0$ as an example of the composite alternative $p_1 \neq p_0$. Furthermore, by restricting ourselves to these simple hypotheses, we are able to compare the

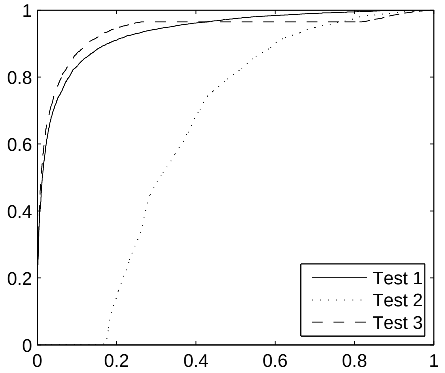

Figure 6.16: ROC curve of test for $K = 20$ and SNR $= 2dB$

three statistics which test essentially different assertions.

The alphabets are defined as in Figure 6.9, namely

$$\mathcal{X}_0 = \{1+i,\, 1-i,\, -1-i,\, -1+i\}$$

and

$$\mathcal{X}_1 = \{\pm a \pm bi :\ a, b \in \{1,3\}, a = 3 \vee b = 3\}.$$

The signal-to-noise ratio does not consider the norm of the states in order to obtain the same variance for observations in the 4-QAM and in the 16-QAM model, i.e.

$$\text{SNR}[dB] = 10 \log \frac{1}{\sigma^2},$$

where σ^2 denotes the variance of the noise.

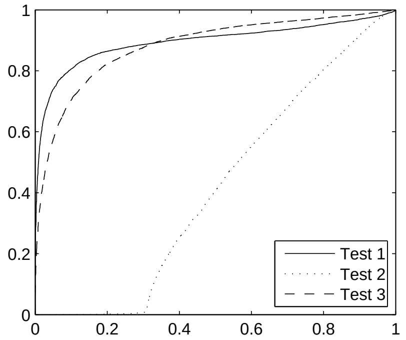

Figure 6.17: ROC curve of test for $K = 100$ and SNR $= -2dB$

Figure 6.15 shows the ROC curve for $K = 50$ and SNR $= 0dB$, while we have chosen $K = 20$ and SNR $= 2dB$ for Figure 6.16. The third Figure 6.17 finally uses $K = 100$ and SNR $= -2dB$. The three figures reveal that especially Test 1 and Test 3 make do with very little observations, e.g 20 observations for SNR $= 2dB$. We do not show ROC curves for larger number of observations, since the two tests are then able to separate the two hypotheses in 100% of the cases. Test 1 and Test 3 appear to be equivalent in general, with slight advantages for Test 1 when the SNR is small, while on the other hand Test 3 is slightly superior if the SNR is large.

Test 2 is clearly inferior. This is supposedly due to the fact, that the estimation in the model with $p_1 = 0$ (i.e. the 4-QAM) is more robust than the estimation in the larger model, i.e. the likelihood function is more less peaked in the larger model. While Test 2 estimates in a model with 16 states for each of the hypotheses, the two other tests use the smaller model for one of the hypotheses.

6.3.5 Conclusion

We have established the identifiability of maximum likelihood Classification for linear modulation schemes. Furthermore, we have proposed two new statistical tests, which have the advantage that their asymptotic distribution is known, such that it is easy to calibrate the test. One of these tests is furthermore equivalent to the simple hypothesis test in terms of the ROC performance. The same wrapping model may also be used for model estimation if more than two potential modulation schemes are given. The test statistics can also be easily extended to frequency-selective multi-path channels. However, the theoretical evaluation then becomes much more difficult.

CHAPTER 7

CONCLUSION - LIMITATIONS AND OUTLOOK

We have presented several methods and algorithms on each of the different levels of the Blind Classification problem. For a summary of our main contributions we refer to Chapter 2 and the respective sections in the other chapters.

In this chapter we will rather concentrate on practical limitations of the presented algorithms. The aim is to provide a feeling in which situation the algorithm work well and in which they do not. We will furthermore give an outlook of ideas how to further improve the proposed methods or apply them to other problems.

In Section 2.3, we have given a survey of theoretical results concerning the involved algorithms. We refer to this chapter for an outlook of what is possible and necessary to extend and establish theoretical results for the presented algorithms.

7.1 CLASSIFICATION AND IDENTIFICATION IN FREQUENCY-SELECTIVE CHANNELS

The algorithms we have proposed have certainly pushed the limits for Blind Identification and Blind Classification of linear modulations schemes in frequency-selective channels a considerable step forward. We have presented several results which showed the excellent performance for channel acquisition especially if the modulation scheme is not too large (up to 16-QAM), but for considerably complex channels. The combination of the EM algorithm with the Fixed-Interval Smoothing or the Joined Two-Filter Smoothing coupled with a random selection scheme showed clearly better results than the existing methods.

However, there are some practical issues that limit the use or degrade the performance of the proposed methods. The first issue is the curse of dimensionality. The larger the constellation of the used modulation scheme is, the more difficult the channel acquisition gets. On the other hand, the larger the constellation is, the better the channel estimate has to be in order to be able to recover the transmitted symbol sequence. There are several reasons why the channel acquisition is more complex for a large constellation (256-QAM). First of all, the likelihood function may have more local maxima in which case the convergence of the EM algorithm to the global maximum is more unlikely. Furthermore, since the state space is much larger, at each iteration, the EM algorithm moves the channel estimate much slower. This is due to the fact that in a large state space it is more easy to explain bad channel estimates compared to a small state space, such that the EM algorithm is moving more slowly away from these estimates. Hence, the EM algorithm needs more iterations to converge. There is another time issue: Since the state space is larger, we need more particles for the particle smoothing. Furthermore, at each time step we explore the offsprings of each particle, which means 256 for 256-QAM instead of 16 for a 16-QAM. For these reasons it is crucial to use a macro-states model as presented in Section 6.2.5 in order to obtain a good initial estimate of the parameters. This reduces the number of iterations as well as the probability to converge to a local maximum. Blind Identification followed by symbol recovery for a 256-QAM is in practice only possible if the channel is fairly simple. Otherwise, the presented algorithms fail or take up an unreasonable amount of time. On the other hand, it turns out that Blind Classification is a much easier task. In order to estimate the modulation scheme, the channel estimate does not have to be as accurate as for symbol recovery. For that reason, the computational results for Blind Classification in Section 6.2.6 are promising even for the 256-QAM with a reasonable amount of computational power.

Another issue is inherent to Particle Smoothing. All the Particle Smoothing algorithms in the literature as well as the Joined Two-Filter Smoothing that we proposed do not actually sample their particle positions from the smoothing distribution, but from the filtering distribution - even if it is from the forward as well as the backward filtering distribution for the Two-Filter Smoothing algorithms. The existing algorithms that sample from the smoothing distribution are computationally not feasible for this problem. As we explained before, a forward Particle Filter may easily identify channels that are maximum phase, i.e. where the first coefficient is the strongest and equivalently the backward Particle Filter works well for channels that are minimum phase. However, real world channels will never be maximum phase. The finite impulse response of the frequency-selective channel is the delayed and attenuated pulse shaping filter, maybe the superposition of several delayed versions of the filter if the channel is multi-path. Since the pulse shape is not a perfect pulse, there will certainly be some contributions before the strongest coefficient. A forward particle filter tends to cut off all the coefficients that come before the first real strong coefficient. Thus, the channel estimate of a particle filter will not be completely accurate. Again, this is more serious for large modulation schemes since they require a better estimate of the channel, while those cut off contributions normally not pose any problems for small constellations.

A critical issue of the EM algorithm is computational time. The algorithm requires a high number of calculations. This limits the practical use of the Blind Classification if a the number of potential modulation schemes is too large and at the same time the channel order is unknown. Thanks to the rapid increase in computational power, this problem will hopefully decrease over the next years.

We have presented new model estimators that involve an additional proportions parameter. We have derived their asymptotic distribution such that it is possible to calibrate these tests in the case of flat-fading channels. These results have to be extended to frequency-selective channels. The derivation will however be much more involved and was therefore not possible during the time of this project.

7.2 Expectation Sparse Maximization Algorithm

We like to devote an own section to the conclusion of the ESpaM algorithm. This algorithm is novel and we have formulated in a very general manner. Therefore, it is not only applicable to Blind Classification or Identification in digital communications, but we are confident that it may be useful in many other areas. On the one hand, the concept of Compressive Sensing has been applied to numerous research areas of which digital communications is not necessarily the most important. Whenever there is some uncertainty about the measurement matrix in a Compressive Sensing problem, the ESpaM algorithm may be used to gain information on the measurement matrix and the parameter vector at the same time.

On the other hand, the EM algorithm has as well been applied to numerous different areas.

In many cases, there might be some additional knowledge that the parameter vector is sparse, which had not been used yet in the EM algorithm. Hence, whenever this is the case the ESpaM algorithm may be used to provide a more robust estimator of the sparse parameter vector.

We have briefly discussed semiblind channel acquisition for the OFDM transmission. However, the EM algorithm as well as the ESpaM algorithm in their current forms are not well adapted to this problem. In the maximization step, each observation is weighted equally, i.e. is considered to carry the same amount of information. If some of the transmitted symbols are known, then there is more information on the posterior distribution of the symbol vectors at the corresponding time steps. We propose therefore to look into a weighted version of the EM algorithm, where the influence of each observation on the parameter update is weighted according to the amount of information that is available about this observations.

Appendix A

Simulations numériques sur des canaux séléctifs en fréquence plus réalistes

Dans ce chapitre, nous évaluons la performance des algorithmes de classification et identification aveugles sur des canaux séléctifs en fréquence, mais simuler de façon plus réaliste que dans les chapitres précédents. Les canaux et les observations sont simulé par rapport aux standards définie par COST207 [1989], ETSI [2002]. Dans le cadre de ces simulations, nous testons également la robustesse des algorithmes par rapport à quelques hypothèses aux modèles. Nous testons notamment sur des canaux variant en temps, des erreurs d'estimation de la période symbole et de la fréquence de la porteuse.

Nous avons mis au point de nouveaux algorithmes de déconvolution autodidactes, basés sur des approximations particulaires de l'estimateur du maximum de vraisemblance des paramètres des canaux. Ces méthodes sont itératives et basées sur une version stochastique de l'algorithme EM (Expectation-Maximisation), qui revient à remplacer le calcul exact des espérances sous les lois de lissage des symboles par des approximations de type échantillonnage d'importance (nous retenons un ensemble de valeurs possibles des symboles que nous pondérons à l'aide de poids d'importance). Les méthodes de calcul des symboles et des poids s'inspirent des méthodes de traitement ńper-survivor processingż, dans le sens où un certain nombre de chemins (dépendant de la complexité de la modulation et du canal) sont conservés dans une passe avant. Ils s'en distinguent par l'ajout d'une passe arrière, dont l'objectif est d'estimer des poids d'importance pour estimer les probabilités de lissage marginales. La complexité de ces algorithmes croît de façon linéaire avec le nombre d'états du treillis et du nombre de chemins conservés lors de la passe avant : elle dépend donc fortement de la complexité du problème que l'on se pose. Nous avons testé plusieurs de ces algorithmes dont le meilleur est l'algorithme de lissage à horizon fixe couplé à la méthode de sélection des poids minimisant la norme $L2$. Nous avons encore amélioré cet algorithme en changeant la direction de l'estimation d'une initialisation à l'autre.

Nous avons montré que l'estimation dans de grandes modulations (MAQ-64, MAQ-128 et MAQ-256) peut être améliorée on estimant d'abord les coefficients de canal à l'aide d'une méthode de quasi maximum de vraisemblance dans lequel nous remplaçons la vraisemblance exacte par la vraisemblance associée à une modulation ayant un plus petit nombre d'états (par exemple MAQ-16). Nous avons démontré à la fois théoriquement et pratiquement que cette façon de procéder permettait d'obtenir des estimateurs robustes qui permettent d'obtenir de bonnes valeurs d'initialisation pour l'estimation dans les vrais modèles. Chaque état dans le modèle biaisé peut être vu comme un macro-état regroupant un certain nombre d'états du vrai modèle. Nous avons conçu et mis en oeuvre un algorithme innovant d'estimation de l'ordre de la modulation, qui combine des critères issus de la théorie de l'information (critère MDL), les nouveaux algorithmes d'estimation des coefficients du canal par maximum de vraisemblance approché, et l'estimation par macro-états.

Nous avons mis en œuvre des simulations qui montrent la performance des nouveaux algorithmes. Nous avons déterminé à l'aide de méthodes de Monte-Carlo les taux d'erreurs binaires dans un nombre important de scénarios applicatifs en utilisant par le simulateur développé dans le cadre du projet. Pour démontrer la performance de nos algorithmes par rapport à l'état de l'art, nous considérons dans ce rapport un scénario particulier. Nous supposons que le rapport signal sur bruit (RSB) est choisi de telle sorte que pour les différentes constellations considérés le taux d'erreur théorique soit égal à $1e-3$ (la puissance du bruit varie donc en fonction de la modulation). Nous commençons par des résultats de l'estimation des taux d'erreurs des symboles pour des canaux à temps fixe et nous comparons nos algorithmes à l'algorithme CMA.

Nous considérons dans ces simulations les taux d'erreurs pour des données engendrées par le simulateur développé dans le cadre du projet. Les alphabets des modèles MAQ (MAQ-4, MAQ-16, MAQ-32, MAQ-64, MAQ-128, MAQ-256) sont normalisés pour que la puissance moyenne des symboles soit égale à 1. Les canaux considéré sont RAx, TUx, BUx et HTx. Nous fixons le rapport signal sur bruit (RSB) de chaque constellation pour que le taux d'erreur théorique soit égal à $1e-3$. L'excès de bande est fixé à $\lambda = 0.5$.

Nous commençons par présenter les résultats d'estimation des taux d'erreurs des symboles pour des canaux à temps fixe. Nous comparons nos algorithmes à l'algorithme CMA par blocs, qui est considéré comme l'état de l'art dans le contexte de la déconvolution autodidacte. Afin de pouvoir employer le CMA, il est nécessaire que le nombre d'états de la modulation soit connu. Par contre, l'ordre du canal n'a pas besoin d'être connu a priori et il est estimé parmi les ordres potentiels $L = 3, 5, 8$. Afin de réduire la complexité des simulations, nous réduisons la liste des ordres potentiels. Notons que l'algorithme CMA ajuste directement un filtre de déconvolution et ne procède donc pas à proprement parler à une estimation du canal.

Les résultats dépendent de façon très significative des choix de paramètres comme le nombre de particules, le nombre d'itérations et le nombre maximal d'itérations de l'algorithme EM approché. Les taux d'erreurs s'améliorent si on augmente le nombre de particules et le nombre d'itérations de l'algorithme EM. Mais l'augmentation de ces quantités augmente la complexité, de sorte que le choix des paramètres est un compromis entre le temps de calcul et la performance des méthodes d'estimation, mesurée au moyen du taux d'erreurs. Par exemple, il est préférable d'utiliser des ordres faibles du canal (par exemple $L = 3$), car l'estimation dans un tel modèle est plus simple et on peut donc diminuer à la fois le nombre de particules et le nombre d'itérations de l'algorithme EM. Les taux d'erreurs doivent donc être comparés en prenant en considération le choix correspondant des paramètres utilisés pour la simulation.

A.1 Taux d'erreurs pour le modèle MAQ-16

Le choix des paramètres pour la modulation MAQ-16 est donné dans la table A.1. Dans les tables suivantes (voir les tables A.2 , A.3, A.4 et A.5), les colonnes correspondent à des ordres du canal avec lesquelles l'estimation a été effectuée. $L = \hat{L}$ veut dire que, pour chaque simulation, l'ordre du canal est estimé (parmi une liste d'ordre de modèles possibles) et que le taux d'erreur est ensuite estimé en utilisant cet estimateur du canal. Elles montrent la probabilité d'avoir un taux d'erreurs plus faible que 0.01 pour différents modèles de canaux et pour une longueur de la trame égal à $K = 500$.

Nous présentons uniquement les résultats obtenus par l'algorithme EM approché basé sur une estimation automatique de l'ordre du canal. Pour le canal Rax (voir la table A.6), nous avons uniquement étudié les performances pour un ordre du canal $L = 3$ lorsque la taille du bloc excède $K = 1000$, ce qui explique la baisse de la performance de l'algorithme EM approché pour $K = 1000$, $K = 1500$ et $K = 2000$. Pour des valeurs de longueur de trames aussi longues (on peut s'interroger sur leur pertinence "pratique"),les algorithmes EM approchés sont très

Paramètre	**Ordre du canal**		
	$L = 3$	$L = 5$	$L = 8$
N	100	500	500
I	2	4	4
η	50		

Table A.1: Paramètres de simulation pour la modulation MAQ-16: N est le nombre de particules, I est le nombre d'initialisations et η est le nombre maximal des itérations

Algorithme	**Ordre du canal**			
	$L = 3$	$L = 5$	$L = 8$	$L = \hat{L}$
CMA	88.59 %			
Algorithme Particulaire	74.77 %	79.88%	48.35 %	94.89%
EMVA	75.14 %	40.68 %	51.98%	

Table A.2: RAx, MAQ-16, $n = 500$, probabilité d'un taux d'erreur $<$0.01

Algorithme	**Ordre du canal**			
	$L = 3$	$L = 5$	$L = 8$	$L = \hat{L}$
CMA	97.18 %			
Algorithme Particulaire	80.51 %	82.20%	51.69 %	98.87%
EMVA	75.14 %	40.68 %	51.98%	%

Table A.3: TUx, MAQ-16, $n = 500$, probabilité d'un taux d'erreur $<$0.01

Algorithme	**Ordre du canal**			
	$L = 3$	$L = 5$	$L = 8$	$L = \hat{L}$
CMA	79.74 %			
Algorithme Particulaire	93.81 %	98.50%	62.10 %	100 %
EMVA	44.84 %	42.21 %	40.52%	%

Table A.4: BUx, MAQ-16, $n = 500$, probabilité d'un taux d'erreur $<$0.01

Algorithme	**Ordre du canal**			
	$L = 3$	$L = 5$	$L = 8$	$L = \hat{L}$
CMA	63.11 %			
Algorithme Particulaire	11.17 %	83.50%	64.32 %	94.90 %

Table A.5: HTx, MAQ-16, $n = 500$, probabilité d'un taux d'erreur $<$0.01

complexes et gourmands en mémoire.

Les résultats montrent que l'algorithme EM approché est supérieur à l'algorithme CMA, particulièrement pour les canaux BUx et HTx. Les résultats suivants mettent en évidence la dépendance de la performance des algorithmes par rapport à la longueur de la trame. L'algorithme CMA par bloc est particulièrement pénalisé lorsque la taille des blocs est courte, mais sa performance sur des blocs de taille plus importante est satisfaisante.

Les résultats pour le canal TUx sont donnés dans la table A.6.

Pour le canal BUx (voir la table A.8), nous avons uniquement étudié les ordres du canal $L = 3,5$ à partir de $K = 1000$. Pour tous les canaux, on peut remarquer que la performance de l'algorithme particulaire est satisfaisante pour des trames courtes. Pour des trames longues, l'algorithme CMA redevient compétitif, car les algorithmes particulaires sont alors pénalisés par leur complexité qui croît linéairement en fonction du nombre d'observations.

A.2 Taux d'erreurs pour le modèle MAQ-32

Nous considérons maintenant les résultats pour la modulation MAQ-32. Les paramètres de simulation sont donnés dans la table A.9. Dans la table A.10, nous donnons le pourcentage de simulations pour lesquels le taux d'erreurs est inférieur à 0.01 pour une longueur de trame égale à 500, différents choix de longueur de canal et différents types de canaux. Comme pour le modèle MAQ-16, l'algorithme particulaire est clairement supérieur à l'algorithme CMA.

De nouveau, nous montrons maintenant la dépendance du taux d'erreurs en fonction de la longueur de la trame. Pour des raisons de temps (disponibilité des machines), tous les résultats pour le canal RAx pour les méthodes de maximum de vraisemblance approchés ont été obtenus on utilisant uniquement $L = 3$, ce qui explique les performances significativement inférieures à l'algorithme CMA pour des tailles de trame importantes, voir la table A.11 (les simulations complémentaires seront effectués avec l'estimateur adaptatif de l'ordre; une implémentation temps-réel devrait être envisagée). Pour le canal TUx, nous avons étudié différents choix de paramètres rappelés dans la table A.12. Pour $K = 200, 300$ et 1000, nous avons effectué uniquement 2 initialisations pour $L = 3$ et 4 initialisations pour $L = 5$. À partir de $K = 1500$, nous n'avons plus utilisé $L = 5$. On peut voir que le taux d'erreur dépend fortement du nombre d'initialisations. Même si on utilise uniquement $L = 3$ ($K = 1500, 2000$) le taux d'erreur est fortement supérieur à l'estimation avec $L = 3,5$ mais avec moins d'initialisations. Dans la simulation du canal BUx, voir la table A.13, le nombre d'initialisations est choisi égal à 2 pour $L = 3$ et 4 pour $L = 5$ pour $K = 200, 300$. Tous ces résultats montrent à nouveau que les performances de l'algorithme EM approché restent satisfaisantes même lorsque la longueur de la trame est faible. Le taux d'erreur est déjà raisonnable pour des trames courtes, contrairement à l'algorithme CMA qui requiert des trames beaucoup plus importantes pour converger. Par contre, l'algorithme EM approché dépend assez fortement du modèle de canal; nous essaierons de caractériser les raisons de cette dépendance de façon plus approfondie au cours de la prochaine phase de l'étude.

A.3 Taux d'erreurs pour le modèle MAQ-4

Pour le modèle MAQ-4, l'algorithme EM exact basé sur l'algorithme Baum-Welch peut être utilisé. Les paramètres de simulation sont donnés dans la table A.14. Les résultats obtenus pour l'algorithme estimant l'ordre du canal de façon automatique sont rassemblés dans la table A.15 où AP désigne l'algorithme particulaire.

Algorithme	**Nombre d'observations**					
	$K = 200$	$K = 300$	$K = 500$	$K = 1000$	$K = 1500$	$K = 2000$
CMA	6.48 %	100.00%	97.18 %	95.80 %	97.66%	95.80 %
Algorithme Particulaire	96.67 %	95.00 %	98.31 %	69.20 %	73.63%	69.20 %

Table A.6: RAx, MAQ-16, probabilité d'un taux d'erreur <0.01

Algorithme	**Nombre d'observations**					
	$K = 200$	$K = 300$	$K = 500$	$K = 1000$	$K = 1500$	$K = 2000$
CMA	12.00%	51.48%	97.18 %	100.00%	100.00%	100.00%
Algorithme Particulaire	98.00%	99.41%	98.31 %	98.01 %	98.01%	94.43 %

Table A.7: TUx, MAQ-16, probabilité d'un taux d'erreur <0.01

Algorithme	**Nombre d'observations**					
	$K = 200$	$K = 300$	$K = 500$	$K = 1000$	$K = 1500$	$K = 2000$
CMA	3.81 %	23.29%	79.74 %	91.53%	93.45%	96.15%
Algorithme Particulaire	99.75 %	99.82%	100.00%	99.73%	99.72%	100.00%

Table A.8: BUx, MAQ-16, probabilité d'un taux d'erreur <0.01

Paramètre	**Ordre du canal**	
	$L = 3$	$L = 5$
N	200	800
I	4	6
η	75	

Table A.9: Paramètres de simulation pour la modulation MAQ-32

Algorithme	CMA	**Algorithme particulaire**		
		$L = 3$	$L = 5$	$L = \hat{L}$
RAx	61.02%	73.04%	63.02%	80.59%
TUx	74.02%	91.48%	73.80%	91.48%
BUx	43.18%	79.80%	72.73%	89.90%
HTx	43.00%	56.20%	39.20%	70.80%

Table A.10: MAQ-32, $n = 500$, probabilité d'un taux d'erreur <0.01

Algorithme	**Nombre d'observations**					
	$K = 200$	$K = 300$	$K = 500$	$K = 1000$	$K = 1500$	$K = 2000$
CMA	0.20 %	9.80%	61.02 %	96.75%	97.33%	97.42%
Algorithme Particulaire	67.00 %	66.80%	73.04%	67.86%	72.33%	71.58%

Table A.11: RAx, MAQ-32, probabilité d'un taux d'erreur <0.01

Algorithme	**Nombre d'observations**					
	$K = 200$	$K = 300$	$K = 500$	$K = 1000$	$K = 1500$	$K = 2000$
CMA	1.40 %	15.80%	74.02%	100.00%	100.00%	100.00%
Algorithme Particulaire	53.35 %	50.40%	91.48%	55.80%	91.00%	87.74%

Table A.12: TUx, MAQ-32, probabilité d'un taux d'erreur <0.01

A.4 Taux d'erreurs pour le modèles MAQ-64 et MAQ-128

Dans le modèle MAQ-64, le canal est d'abord estimé en utilisant un modèle de macro-états; le modèle le plus approprié est le MAQ-16. Cet estimateur est ensuite choisi comme valeur initiale pour l'algorithme EM approché pour le vrai modèle MAQ-64. Les paramètres de simulation de l'estimation en macro-états sont donnés dans la table A.16 et les paramètres de l'estimation dans le modèle MAQ-64 sont rappelés dans la table A.17.

Dans la table A.18 nous donnons le pourcentage de simulations pour lesquels le taux d'erreurs est inférieur à 0.01 pour une longueur de trame égale à 500, différents choix de longueur de canal et différents types de canaux comme pour le modèle MAQ-32. La performance de l'algorithme EM approché est très satisfaisante sur le canal BUx, par contre elle est décevante pour le canal RAx (cette chute de performance doit être analysée). La table A.19 montre la dépendance de la performance de la longueur de la trame dans le modèle MAQ-64 sur le canal BUx. Pour cette simulation, nous avons uniquement utilisé l'ordre du canal égal à $L = 3$. Comme pour les modulations à faible nombre d'états, la performance de l'algorithme EM approché est satisfaisant même lorsque la longueur de trames est courte. Par contre, l'algorithme CMA nécessite au moins 1500 ou même 2000 observations pour être capable d'égaliser de façon satisfaisante le canal de propagation.

Dans le modèle MAQ-128, le canal est d'abord estimé en utilisant un modèle de macro-états, notamment le MAQ-32. Cet estimateur est choisi comme valeur initiale pour l'algorithme EM approché pour le modèle MAQ-128. Les paramètres de simulation de l'estimation en macro-états sont donnés dans la table A.20 et les paramètres de l'estimation dans le modèle MAQ-64 dans la table A.21. Dans la table A.22 nous donnons le pourcentage de simulations pour lesquels le taux d'erreurs est inférieur à 0.01 pour une longueur de trame égale à 500 et différents choix de longueur de canal.

A.5 Influence d'erreur de l'estimation de la période symbole et de la fréquence de la porteuse

Nous étudions maintenant l'influence d'une erreur d'estimation des paramètres qui doivent être estimés avant de mettre en œvre les méthodes d'identification, notamment la période de symboles et la fréquence de la porteuse. Les erreurs d'estimation dans les deux cas sont de l'ordre $\mathcal{O}(K^{-3/2})$ ou K et la longueur de la trame. Pour les simulations, nous avons ajusté le facteur linéaire de l'erreur ε de sorte que pour une longueur de la trame égale à $K = 300$ l'erreur soit égale à $\varepsilon = 10^{-4}$. Nous commençons par le modèle MAQ-4 en choisissant les paramètres d'estimation mentionnés dans la table A.23. Comme le nombre d'états de la modulation est faible, nous pouvons encore mettre on œvre l'algorithme EM exact. Nous donnons les résultats pour le canal BUx dans la table A.24.

Pour le modèle MAQ-16, les paramètres d'initialisation sont donnés dans la table A.1, mais nous avons uniquement utilisé $L = 3, 5$. Nous donnons les résultats pour le même canal BUx dans la table A.25.

Dans le modèle MAQ-64, nous avons uniquement effectué les simulations pour un ordre de canal égal à $L = 3$ avec des paramètres $I = 4$, $\eta = 60$ et $N = 250$ et 40 itérations dans le modèle de macro-états (MAQ-16). Les résultats sont donnés dans la table table:performance-periodesymboleMAQ64. Nous avons fait les mêmes simulation afin d'évaluer l'influence d'une erreur sur la fréquence porteuse. Comme cette erreur n'est pas prise en compte dans nos modèles (qui supposent un canal fixe), il n'est pas étonnant que le taux d'erreur atteigne une valeur importante 90%. Dans le futur, il sera clairement nécessaire de prendre en compte les variations des coefficients des canaux.

Algorithme	**Nombre d'observations**		
	$K = 200$	$K = 300$	$K = 500$
CMA	0 %	6.58 %	41.92%
Algorithme Particulaire	59.61 %	66.45%	89.90%

Table A.13: BUx, MAQ-32, probabilité d'un taux d'erreur <0.01

Paramètre	**Ordre du canal**	
	$L = 3$	$L = 5$
I	2	
η	50	

Table A.14: Paramètres de simulation pour la modulation MAQ-4

Algorithme	$K = 200$		$K = 300$		$K = 500$	
	CMA	AP	CMA	AP	CMA	AP
RAx	96.88%	97.71%	94.60%	95.32%	96.60%	96.80%
TUx	100 %	100 %	100 %	100 %	100 %	100 %
BUx	96.60%	99.60%	98.60%	100 %	98.18%	100 %
HTx	99.60%	98.80%	99.60%	99.40%		

Table A.15: MAQ-4, probabilité d'un taux d'erreur <0.01, AP = algorithme particulaire

Paramètre	**Ordre du canal**	
	$L = 3$	$L = 5$
N	100	500
I	4	6
η	40	

Table A.16: Paramètres de simulation en macro-états (MAQ-16) pour la modulation MAQ-64

Paramètre	**Ordre du canal**	
	$L = 3$	$L = 5$
N	200	800
I	4	6
η	60	

Table A.17: Paramètres de simulation pour la modulation MAQ-64

Algorithme	CMA	**Algorithme particulaire**		
		$L = 3$	$L = 5$	$L = \hat{L}$
RAx	0 %	12.98%	15.63%	25.66%
TUx	0 %	32.19%	30.48%	51.03%
BUx	0.34 %	92.86%	57.14%	96.60%
HTx	0 %	0 %	48.41%	48.41%

Table A.18: MAQ-64, $n = 500$, probabilité d'un taux d'erreur <0.01

Algorithme	**Nombre d'observations**				
	$K = 200$	$K = 300$	$K = 500$	$K = 1500$	$K = 2000$
CMA	0 %	0 %	0.34 %	13.37%	39.11%
Algorithme Particulaire	83.74 %	83.33%	92.86%	89.53%	93.07%

Table A.19: TUx, MAQ-64, probabilité d'un taux d'erreur <0.01

Paramètre	**Ordre du canal**	
	$L = 3$	$L = 5$
N	200	800
I	6	8
η	50	

Table A.20: Paramètres de simulation en macro-états (MAQ-32) pour la modulation MAQ-128

Paramètre	**Ordre du canal**	
	$L = 3$	$L = 5$
N	400	1000
I	6	8
η	50	75

Table A.21: Paramètres de simulation pour la modulation MAQ-128

Algorithme	CMA	**Algorithme particulaire**		
		$L = 3$	$L = 5$	$L = \hat{L}$
BUx	0 %	34.61%	52.40%	64.90%

Table A.22: MAQ-128, $n = 500$, probabilité d'un taux d'erreur <0.01

Paramètre	**Ordre du canal**	
	$L = 3$	$L = 5$
I	2	
η	40	

Table A.23: Paramètres pour la modulation MAQ-4, avec une erreur de l'estimation de la période de symboles

Algorithme	**Nombre d'observations**				
	$K = 200$	$K = 300$	$K = 500$	$K = 1500$	$K = 2000$
CMA	0.40 %	0.15 %	0.16 %	0.05%	0.08%
Algorithme Particulaire	0 %	0 %	0 %	0 %	0 %

Table A.24: taux d'erreur global; BUx, MAQ-4, erreur de période de symboles: $\mathcal{O}(K^{-3/2})$

Algorithme	**Nombre d'observations**				
	$K = 200$	$K = 300$	$K = 500$	$K = 1500$	$K = 2000$
CMA	17.02 %	5.58 %	1.74 %	0.48%	0.73%
Algorithme Particulaire	0.81 %	0.01 %	0.10 %	0.01 %	0.37 %

Table A.25: taux d'erreur global; BUx, MAQ-16, erreur de période de symboles: $\mathcal{O}(K^{-3/2})$

Algorithme	**Nombre d'observations**			
	$K = 200$	$K = 500$	$K = 1000$	$K = 2000$
CMA	71.74 %	30.28 %	9.35 %	3.75%
Algorithme Particulaire	1.44 %	0.30 %	0.10 %	0 %

Table A.26: taux d'erreur global; BUx, MAQ-64, erreur de période de symboles: $\mathcal{O}(K^{-3/2})$

A.6 Choix de modèle

Nous étudions maintenant la performance de la procédure de choix de modèle. Il n'existe aucun autre algorithme dans la littérature capable d'estimer l'ordre de la modulation et du canal pour des modulations autres que la QPSK et BPSK. Nous rapportons dans ce document donc uniquement les résultats pour l'algorithme que nous avons proposé. Dans la simulation, nous avons étudié des modulations MAQ-4, MAQ-16, MAQ-32,MAQ-64, MAQ-128 et MAQ-256 sur le canal BUx et pour une longueur de la trame égale à $K = 300$.

Les paramètres de simulation sont donnés dans la table A.27. Pour le modèle MAQ-4, nous mettons en oeuvre l'algorithme EM exact, et c'est pourquoi le nombre de particules n'est pas indiqué. La performance du choix de modèle est évalué en termes de la probabilité d'estimation du vrai modèle.

Dans la table A.28 nous donnons le pourcentage de simulations pour lesquels le vrai modèle est estimé pour le canal BUx et dans la table A.29 pour le canal TUx.

Nous avons choisi un nombre de particules, un nombre d'itérations et un nombre d'initialisations assez limités et les résultats sont déjà encourageants. Les probabilités de détection pourraient être améliorées en augmentant la complexité de calcul.

A.7 Canaux à temps variables

Finalement, nous présentons les performances des algorithmes pour des canaux à temps variables. Nous supposons que la vitesse du mobile est égale à 30m/s. Rappelons que les algorithmes sont développés pour des modèles de canaux constants et que la variation du canal n'est pour l'instant pas prise explicitement en compte. Il n'est pas surprenant que le taux d'erreur devrait augmente en fonction de la longueur de la trame, car sur des trames plus longues, la variation du canal est plus sensible que sur des trames courtes. Le choix des paramètres est le même que pour les canaux constants.

Nous montrons les résultats pour la modulation MAQ-4 dans la table A.30 et les résultats pour la modulation MAQ-16 dans la table A.31.

Paramètre	**Modulation**					
	MAQ-4	MAQ-16	MAQ-32	MAQ-64	MAQ-128	MAQ-256
L	3					
N	-	50	50	200	200	200
I	4					
η	60					

Table A.27: Paramètres de simulation pour le choix de modèle

Algorithme	**Vrai modèle**					
	MAQ-4	MAQ-16	MAQ-32	MAQ-64	MAQ-128	MAQ-256
Choix de modèle	98.29 %	98.33 %	65.84 %	99.19%	90.60%	73.80 %

Table A.28: BUx, K, probabilité de l'estimation du vrai modèle

[hbtp]

Algorithme	**Vrai modèle**					
	MAQ-4	MAQ-16	MAQ-32	MAQ-64	MAQ-128	MAQ-256
Choix de modèle	100 %	100 %	33.33 %	100%	69.60%	49.30 %

Table A.29: BUx, K, probabilité de l'estimation du vrai modèle

Algorithme	**Nombre d'observations**				
	$K = 200$	$K = 300$	$K = 500$	$K = 1500$	$K = 2000$
CMA	11.04 %	9.42 %	11.35 %	23.90%	41.58%
Algorithme Particulaire	3.81 %	4.32 %	6.91 %	18.51 %	38.36 %

Table A.30: taux d'erreur global; BUx, MAQ-4, canaux variables, vitesse : 30m/s

Algorithme	**Nombre d'observations**				
	$K = 200$	$K = 300$	$K = 500$	$K = 1500$	$K = 2000$
CMA	48.74 %	42.01 %	43.06 %	58.30%	74.52%
Algorithme Particulaire	18.27 %	21.47 %	31.23 %	56.93 %	80.44 %

Table A.31: taux d'erreur global; BUx, MAQ-16, canaux variables, vitesse : 30m/s

Bibliography

A. Abdi, O. Dobre, R. Choudhry, Y. Bar-Ness, and W. Su. Modulation classification in fading channels using antenna arrays. In *Military Communications Conference, 2004. MILCOM 2004. IEEE*, volume 1, pages 211 – 217 Vol. 1, oct.-3 nov. 2004. doi: 10.1109/MILCOM.2004.1493271.

P. Aggarwal and X. Wang. Multilevel sequential monte carlo algorithms for MIMO demodulation. *Wireless Communications, IEEE Transactions on*, 6(2):750–758, Feb. 2007. ISSN 1536-1276. doi: 10.1109/TWC.2007.05453.

N. Ahmadi and R. Berangi. Modulation classification of QAM and PSK from their constellation using genetic algorithm and hierarchical clustering. *Information and Communication Technologies: From Theory to Applications, 2008. ICTTA 2008. 3rd International Conference on*, pages 1–5, April 2008. doi: 10.1109/ICTTA.2008.4530242.

T. Y. Al-Naffouri, A. Bahai, and A. Paulraj. Semi-blind channel identification and equalization in OFDM: An expectation-maximization approach. In *VTC*, 2002.

B. D. O. Anderson and B. Moore, J. *Optimal Filtering.* Dover Books on Engineering, 2005.

J. Anderson and S. Mohan. Sequential coding algorithms: A survey and cost analysis. *Communications, IEEE Transactions on [legacy, pre - 1988]*, 32(2):169–176, Feb 1984. ISSN 0096-2244.

D. Andrews. Estimation when a parameter is on a boundary. *Econometrica*, 67:1341–1383, 1999. Cowles Foundation Paper No. 988.

D. Andrews. Testing when a parameter is on the boundary of the maintained hypothesis. *Econometrica*, 69:683:734, 2001. Cowles Foundation Paper No. 1021.

C. Anton-Haro, J. Fonollosa, and J. Fonollosa. Blind channel estimation and data detection using hidden Markov models. *Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on]*, 45(1):241–247, 1997. ISSN 1053-587X.

M. S. Asif, W. Mantzel, and J. Romberg. Random channel coding and blind deconvolution. *Allerton Conference on Communication, Control, and Computing*, 2009.

L. Bahl, J. Cocke, F. Jelinek, and J. Raviv. Optimal decoding of linear codes for minimizing symbol error rate (corresp.). *Information Theory, IEEE Transactions on*, 20(2):284–287, March 1974. ISSN 0018-9448.

W. Bajwa, J. Haupt, G. Raz, and R. Nowak. Compressed channel sensing. In *Information Sciences and Systems, 2008. CISS 2008. 42nd Annual Conference on*, pages 5–10, March 2008a. doi: 10.1109/CISS.2008.4558485.

W. Bajwa, A. Sayeed, and R. Nowak. Compressed sensing of wireless channels in time, frequency, and space. In *Signals, Systems and Computers, 2008 42nd Asilomar Conference on*, pages 2048–2052, Oct. 2008b. doi: 10.1109/ACSSC.2008.5074792.

W. Bajwa, A. Sayeed, and R. Nowak. Learning sparse doubly-selective channels. In *Communication, Control, and Computing, 46th Annual Allerton Conference on*, pages 575–582, Sept. 2008c. doi: 10.1109/ALLERTON.2008.4797610.

W. U. Bajwa, J. Haupt, A. M. Sayeed, and R. Nowak. Compressed channel sensing: A new approach to estimating sparse multipath channels. *to appear in Proc. IEEE*, 2010.

S. Barembruch. A comparison of approximate Viterbi techniques and particle filtering for data estimation in digital communications. In *Proc. IEEE International Conference on Speech, Audio and Signal Processing*, 2010.

S. Barembruch, A. Garivier, and E. Moulines. On approximate maximum likelihood methods for blind identification: How to cope with the curse of dimensionality. *Proc. 9th IEEE Workshop Signal Processing Advances Wireless Communications*, pages 639 – 643, 2008a.

S. Barembruch, A. Garivier, and E. Moulines. On optimal sampling for particle filtering in digital communication. *Proc. 9th IEEE Workshop Signal Processing Advances Wireless Communications*, pages 634 – 638, 2008b.

S. Barembruch, A. Garivier, and E. Moulines. On approximate maximum likelihood methods for blind identification: How to cope with the curse of dimensionality. *IEEE Transactions on Signal Processing*, 57(11):4247 – 4259, 2009.

S. Barembruch, E. Moulines, and A. Scaglione. The expectation and sparse maximization algorithm. *submitted to Journal on Communications and Networks*, 2010a.

S. Barembruch, E. Moulines, and A. Scaglione. Maximum likelihood blind deconvolution for sparse systems. In *Proc. IEEE Conference on Cognitive Information Processing*, 2010b.

S. Barembruch, E. Moulines, and A. Scaglione. A sparse EM algorithm for blind and semi-blind identification of doubly selective OFDM channels. In *Proc. 11th IEEE Workshop Signal Processing Advances Wireless Communications*, 2010c.

I. Barhumi and M. Moonen. Mlse and map equalization for transmission over doubly selective channels. *Vehicular Technology, IEEE Transactions on*, 58(8):4120 –4128, oct. 2009. ISSN 0018-9545. doi: 10.1109/TVT.2009.2024537.

L. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics*, 41:164–171, 1970.

F. Ben Salem and G. Salut. Deterministic particle receiver for multipath fading channels in wireless communications. part I: FDMA. *Traitement du Signal*, 21(4):347–358, 2004.

T. Bertozzi, D. Le Ruyet, G. Rigal, and V.-T. Han. Trellis-based search of the maximum a posteriori sequence using particle filtering. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, volume 6, pages 693–696, 2003a.

T. Bertozzi, D. Le Ruyett, G. Rigal, and H. Vu-Thien. On particle filtering for digital communications. In *Signal Processing Advances in Wireless Communications, 2003. SPAWC. 4th IEEE Workshop on*, pages 570–574, 2003b.

P. Bickel, Y. Ritov, and T. Rydén. Asymptotic normality of the maximum-likelihood estimator for general hidden Markov models. *Annals of Statistics*, 26(4):1614–1635, 1998.

J. Bilmes. A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. Technical Report TR-97-021, ICSI, 1997.

C. Bordin and M. Bruno. Particle filters for joint blind equalization and. decoding in frequency-selective channels. *IEEE Transactions on Signal Processing*, 56(6):2395–2405, 2008.

M. Briers, A. Doucet, and S. Maskell. Smoothing algorithms for state-space models. Technical report, Cambridge University Engineering Department Technical Report, CUED/F-INFENG/TR.498, 2004.

E. Candes and T. Tao. Decoding by linear programming. *Information Theory, IEEE Transactions on*, 51(12):4203–4215, Dec. 2005. ISSN 0018-9448. doi: 10.1109/TIT.2005.858979.

E. Candes and T. Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *Information Theory, IEEE Transactions on*, 52(12):5406–5425, Dec. 2006. ISSN 0018-9448. doi: 10.1109/TIT.2006.885507.

O. Cappé, E. Moulines, and T. Rydén. *Inference in Hidden Markov Models.* Springer, 2nd edition edition, 2007.

J. Carpenter, P. Clifford, and P. Fearnhead. Improved particle filter for nonlinear problems. *Radar, Sonar and Navigation, IEE Proceedings* -, 146(1):2–7, 1999. ISSN 1350-2395.

E. Charniak, S. Goldwater, and M. Johnson. Edge-based best-first chart parsing. In *Proceedings of the Sixth Workshop on Very Large Corpora*, 1998. URL citeseer.ist.psu.edu/article/charniak98edgebased.html.

R. Chen, X. Wang, and J. Liu. Adaptive joint detection and decoding in flat-fading channels via mixture Kalman filtering. *Information Theory, IEEE Transactions on*, 46(6):2079–2094, 2000. ISSN 0018-9448.

S. Chen, D. Donoho, and M. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1998. doi: 10.1137/S1064827596304010. URL http://link.aip.org/link/?SCE/20/33/1.

M. Collins. Discriminative training methods for hidden Markov models: theory and experiments with perceptron algorithms. In *EMNLP '02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, pages 1–8, Morristown, NJ, USA, 2002. Association for Computational Linguistics. doi: http://dx.doi.org/10.3115/1118693.1118694.

J. Cornebise. *Adaptive Sequential Monte Carlo Methods.* PhD thesis, Universite de Pierre et Marie Curie, 2009.

COST207. Digital land mobile radio communications. Technical report, Report Office for Official Publications of the European Communities, 1989. ISBN 92-825-9946-9.

S. Cotter and B. Rao. Sparse channel estimation via matching pursuit with application to equalization. *Communications, IEEE Transactions on*, 50(3):374–377, Mar 2002. ISSN 0090-6778. doi: 10.1109/26.990897.

D. Crisan and A. Doucet. Convergence of sequential Monte Carlo methods. Technical Report Cambridge University, CUED/FINFENG /TR381, 2000.

D. Crisan and A. Doucet. A survey of convergence results on particle filtering for practitioners. *IEEE Trans. Signal Processing*, 50(3):736–746, 2002.

P. Del Moral. *Feynman-Kac Formulae.* Springer Heidelberg, 2004.

A. Dembo and O. Zeitouni. *Large Deviations Techniques and Applications (Stochastic Modelling and Applied Probability).* Springer-Verlag, 1999.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum-likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

P. Djuric, Z. Jianqiu, G. Tadesse, H. Yufei, and K. Jayesh. Applications of particle filtering to communications: A review. volume II, pages 169–172, 2002.

P. Djuric, M. Vemula, and M. Bugallo. Signal processing by particle filtering for binary sensor networks. In *Digital Signal Processing Workshop, 2004 and the 3rd IEEE Signal Processing Education Workshop. 2004 IEEE 11th*, pages 263–267, 2004.

O. Dobre and F. Hameed. Likelihood-based algorithms for linear digital modulation classification in fading channels. *Electrical and Computer Engineering, 2006. CCECE '06. Canadian Conference on*, pages 1347–1350, May 2006. doi: 10.1109/CCECE.2006.277525.

O. Dobre, A. Abdi, Y. Bar-Ness, and W. Su. Survey of automatic modulation classification techniques: classical approaches and new trends. *Communications, IET*, 1(2):137–156, April 2007. ISSN 1751-8628. doi: 10.1049/iet-com:20050176.

D. L. Donoho. For most large underdetermined systems of equations, the minimal L1-norm near-solution approximates the sparsest near-solution. *Comm. Pure Appl. Math*, 59:907–934, 2006.

R. Douc and E. Moulines. Limit theorems for weighted samples with applications to sequential Monte Carlo methods. *Annals of Statistics*, 36(5):2344–2376, 2008.

A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000. ISSN 0960-3174. doi: http://dx.doi.org/10.1023/A:1008935410038.

A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo in Practice*. Springer-Verlag, 2001.

G. Druck, M. Narasimhan, and P. Viola. Learning a* underestimates : Using inference to guide inference. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS 07)*, pages 99–106, 2007.

G. Ehret, P. Reichenbach, U. Schindler, C. Horvath, S. Fritz, M. Nabholz, and P. Bucher. DNA binding specificity of different STAT proteins. *J. Biol. Chem.*, 276(9):6675–6688, 2001. doi: 10.1074/jbc.M001748200. URL `http://www.jbc.org/cgi/content/abstract/276/9/6675`. application of Viterbi algorithm in bioinformatics.

ETSI. Digital cellular telecommunications system (phase 2+) ; radio transmission and reception. Technical report, European Telecommunications Standards Institute, 2002. ETSI TS 100 910 v8.14.0 ou 3GPP TS 05.05 version 8.14.0 Release 1999.

M. Eyuboglu, S. Qureshi, and M. Chen. Reduced-state sequence estimation for trellis-coded modulation on intersymbol interference channels. *Global Telecommunications Conference, 1988, and Exhibition. 'Communications for the Information Age.' Conference Record, GLOBECOM '88., IEEE*, pages 878–882 vol.2, Nov-1 Dec 1988. doi: 10.1109/GLOCOM.1988.25963.

P. Fearnhead and P. Clifford. On-line inference for hidden Markov models via particle filters. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(4):887–899, 2003. doi: 10.1111/1467-9868.00421. URL `http://www.blackwell-synergy.com/doi/abs/10.1111/1467-9868.00421`.

P. Fearnhead, D. Wyncoll, and J. Tawn. A sequential smoothing algorithm with linear computational cost. *submitted*, 2008.

M. Feder and J. Catipovic. Algorithms for joint channel estimation and data recovery-application to equalization in underwater communications. *Oceanic Engineering, IEEE Journal of*, 16(1): 42–55, 1991. ISSN 0364-9059.

J. Forney, G. Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference. *Information Theory, IEEE Transactions on*, 18(3):363–378, May 1972. ISSN 0018-9448.

J. Forney, G.D. The Viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, March 1973. ISSN 0018-9219.

J.-J. Fuchs. Multipath time-delay estimation. In *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, volume 1, pages 527–530 vol.1, Apr 1997. doi: 10.1109/ICASSP.1997.599691.

E. Gassiat and S. Boucheron. Optimal error exponents in hidden Markov models order estimation. *Information Theory, IEEE Transactions on*, 49(4):964–980, April 2003. ISSN 0018-9448. doi: 10.1109/TIT.2003.809574.

C. Georghiades and J. C. Han. Sequence estimation in the presence of random parameters via the em algorithm. *Communications, IEEE Transactions on*, 45(3):300–308, Mar 1997. ISSN 0090-6778. doi: 10.1109/26.558691.

T. Ghirmai, M. F. Bugallo, J. Miguez, and P. M. Djuric. A sequential Monte Carlo method for adaptive blind timing estimation and data detection. *IEEE Transactions on Signal Processing*, 53(8):2855–2865, 2005. Part 1.

G. Giannakis. *The Digital Signal Processing Handbook*, chapter Cyclostationary Signal Analysis. CRC Press, 1998.

G. Giannakis and J. Mendel. Identification of nonminimum phase systems using higher order statistics. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 37(3):360 –377, mar 1989. ISSN 0096-3518. doi: 10.1109/29.21704.

G. Giannakis and C. Tepedelenlioglu. Basis expansion models and diversity techniques for blind identification and equalization of time-varying channels. *Proceedings of the IEEE*, 86(10): 1969–1986, Oct 1998. ISSN 0018-9219. doi: 10.1109/5.720248.

P. Giudici, T. Rydén, and P. Vandekerkhove. Likelihood-ratio tests for hidden Markov models. *Biometrics*, 56:742–747, 2000.

D. Godard. Self-recovering equalization and carrier tracking in two-dimensional data communication systems. *Communications, IEEE Transactions on*, 28(11):1867–1875, Nov 1980. ISSN 0090-6778.

S. Godsill, A. Doucet, and M. West. Maximum a posteriori sequence estimation using Monte Carlo particle filters. *Annals of the Institute of Statistical Mathematics*, 53:82–96(15), March 2001. URL http://www.ingentaconnect.com/content/klu/aism/2001/00000053/00000001/00356898.

S. J. Godsill, A. Doucet, and M. West. Monte Carlo smoothing for nonlinear time series. *Journal of the American Statistical Association*, 99:156–168(13), March 2004. doi: doi: 10.1198/016214504000000151. URL http://www.ingentaconnect.com/content/asa/jasa/2004/00000099/00000465/art00016.

N. Gordon, D. Salmond, and A. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, 140(2):107–113, 1993. ISSN 0956-375X.

D. Guo, X. Wang, and R. Chen. Multilevel mixture Kalman filter. *EURASIP J. Appl. Signal Process.*, 2004(1):2255–2266, 2004. ISSN 1110-8657. doi: http://dx.doi.org/10.1155/S1110865704403229.

F. Hameed, O. Dobre, and D. Popescu. On the likelihood-based approach to modulation classification. *Wireless Communications, IEEE Transactions on*, 8(12):5884 –5892, december 2009. ISSN 1536-1276. doi: 10.1109/TWC.2009.12.080883.

C. Herzet, V. Ramon, and L. Vandendorpe. A theoretical framework for iterative synchronization based on the sum-product and the expectation-maximization algorithms. *Signal Processing, IEEE Transactions on*, 55(5):1644–1658, 2007.

H. Hijazi. *Estimation de canal radio-mobile à évolution rapide dans les systèmes à modulation OFDM*. PhD thesis, Grenoble-INP, 2008.

H. Hijazi and L. Ros. Polynomial estimation of time-varying multipath gains with intercarrier interference mitigation in OFDM systems. *Vehicular Technology, IEEE Transactions on*, 58 (1):140–151, Jan. 2009. ISSN 0018-9545. doi: 10.1109/TVT.2008.923653.

H. Hijazi and L. Ros. Joint data QR-detection and Kalman estimation for OFDM time-varying Rayleigh channel complex gains. *Communications, IEEE Transactions on*, 58(1):170–178, January 2010. ISSN 0090-6778. doi: 10.1109/TCOMM.2010.01.080296.

L. Hong. Maximum likelihood BPSK and QPSK classifier in fading environment using the EM algorithm. *System Theory, 2006. SSST '06. Proceeding of the Thirty-Eighth Southeastern Symposium on*, pages 313–317, March 2006. doi: 10.1109/SSST.2006.1619049.

L. Hong and K. Ho. BPSK and qPSK modulation classification with unknown signal level. *MILCOM 2000. 21st Century Military Communications Conference Proceedings*, 2:976–980 vol.2, 2000. doi: 10.1109/MILCOM.2000.904076.

C.-Y. Huan and A. Polydoros. Likelihood methods for mpsk modulation classification. *Communications, IEEE Transactions on*, 43(234):1493 –1504, feb/mar/apr 1995. ISSN 0090-6778. doi: 10.1109/26.380199.

S.-J. Hwang and P. Schniter. EM-based soft noncoherent equalization of doubly selective channels using tree search and basis expansion. In *SPAWC '09. IEEE 10th Workshop on*, pages 6–10, 2009. doi: 10.1109/SPAWC.2009.5161736.

M. Isard and A. Blake. A smoothing filter for condensation. In *Proc 5th European Conf. Computer Vision*, volume 1, pages 767–781, 1998.

F. Jelinek. *Statistical Methods for Speech Recognition.* Cambridge, MA: The MIT Press (Language, speech, and communication series), 1997.

J. Johnson, R., P. Schniter, T. Endres, J. Behm, D. Brown, and R. Casas. Blind equalization using the constant modulus criterion: a review. *Proceedings of the IEEE*, 86(10):1927–1950, 1998. ISSN 0018-9219. doi: 10.1109/5.720246.

G. K. Kaleh and R. Vallet. Joint parameter estimation and symbol detection for linear or nonlinear unknown channels. *Communications, IEEE Transactions on*, 42(7):2406–2413, 1994. ISSN 0090-6778.

R. E. Kalman and R. Bucy. New results in linear filtering and prediction theory. *Transactions of the ASME. Series D, Journal of Basic Engineering*, 83:95–108, 1961.

G. Kitagawa. The two-filter formula for smoothing and an implementation of the Gaussian-sum smoother. *Annals of the Institute of Statistical Mathematics*, 46(4):605–623, Dec. 1994. URL http://dx.doi.org/10.1007/BF00773470.

G. Kitagawa. Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5(1):1–25, 1996. ISSN 10618600. URL http://www.jstor.org/stable/1390750.

C. Komninakis, C. Fragouli, A. Sayed, and R. Wesel. Channel estimation and equalization in fading. In *Signals, Systems, and Computers, 1999. Thirty-Third Asilomar Conference on*, volume 2, pages 1159–1163 vol.2, 1999.

A. Kong, J. S. Liu, and W. H. Wong. Sequential imputations and bayesian missing data problems. *Journal of the American Statistical Association*, 89:278–288, 1994.

N. Lay and A. Polydoros. Modulation classification of signals in unknown isi environments. In *Military Communications Conference, 1995. MILCOM '95, Conference Record, IEEE*, volume 1, pages 170 –174 vol.1, nov 1995. doi: 10.1109/MILCOM.1995.483293.

E. Lehmann and J. Romano. *Testing Statistical Hypotheses.* Springer Science+Business Media, Inc., 2005.

F. Lehmann. Blind estimation and detection of space-time trellis coded transmissions over the Rayleigh fading MIMO channel. *IEEE Transactions on Communications*, 56(3):334–338, 2008.

J. Lember and A. Koloydenko. Adjusted Viterbi training for hidden Markov models. Technical report, Eurandom, Eindhoven, The Netherlands, 2007.

G. Leus. Semi-blind channel estimation for rapidly time-varying channels. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on*, volume 3, pages iii/773 – iii/776 Vol. 3, march 2005. doi: 10.1109/ICASSP.2005.1415824.

G. Leus and M. Moonen. Deterministic subspace based blind channel estimation for doubly-selective channels. In *Signal Processing Advances in Wireless Communications, 2003. SPAWC 2003. 4th IEEE Workshop on*, pages 210 – 214, june 2003.

W. Li and J. Preisig. Estimation of rapidly time-varying sparse channels. *Oceanic Engineering, IEEE Journal of*, 32(4):927–939, Oct. 2007. ISSN 0364-9059. doi: 10.1109/JOE.2007.906409.

J. S. Liu and R. Chen. Blind deconvolution via sequential imputations. *Journal of the American Statistical Association*, 90:567–576, 1995.

J. S. Liu and R. Chen. Sequential monte carlo methods for dynamic systems. *Journal of American Statistical Association*, 93:1032–1044, 1998.

Y. Lui and D. Borah. Estimation of time-varying frequency-selective channels using a matching pursuit technique. In *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, volume 2, pages 941–946 vol.2, March 2003. doi: 10.1109/WCNC.2003.1200498.

S. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *Signal Processing, IEEE Trans. on*, 41(12):3397–3415, 1993. ISSN 1053-587X. doi: 10.1109/78.258082.

D. Q. Mayne. A solution of the smoothing problem for linear dynamic systems. *Automatica*, 4(2):73–92, Dec. 1966. URL http://www.sciencedirect.com/science/article/B6V21-47TFXV1-2/1/6f4d63f3092f3631d40d4d116a8bee2c.

J. Nelson and A. Singer. Bayesian ML sequence detection for ISI channels. In *Information Sciences and Systems, 2006 40th Annual Conference on*, pages 693–698, March 2006. doi: 10.1109/CISS.2006.286556.

H. Nguyen and B. Levy. Blind and semi-blind equalization of CPM signals with the EMV algorithm. *IEEE Transactions on Signal Processing*, 51(10):2650–2664, October 2003. ISSN 1053-587X. doi: 10.1109/TSP.2003.816876.

H. Nguyen and B. Levy. The expectation-maximization Viterbi algorithm for blind adaptive channel equalization. *IEEE Transactions on Communications*, 53(10):1671–1678, October 2005. ISSN 0090-6778. doi: 10.1109/TCOMM.2005.857162.

S. Nikolopoulos, A. Pitsillides, and D. Tipper. Addressing network survivability issues by finding the k-best paths through a trellis graph. In *INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 1, pages 370 –377 vol.1, apr 1997. doi: 10.1109/INFCOM.1997.635161.

J. J. Odell, V. Valtchev, P. C. Woodland, and S. J. Young. A one pass decoder design for large vocabulary recognition. In *HLT '94: Proceedings of the workshop on Human Language Technology*, pages 405–410, Morristown, NJ, USA, 1994. Association for Computational Linguistics. ISBN 1-55860-357-3. doi: http://dx.doi.org/10.3115/1075812.1075905.

J. Olsson, O. Cappé, R. Douc, and E. Moulines. Sequential Monte Carlo smoothing with application to parameter estimation in non-linear state space models. *Journal of the Bernoulli Society*, 14:1:155–179, 2008.

P. Panagiotou, A. Anastasopoulos, and A. Polydoros. Likelihood ratio tests for modulation classification. In *MILCOM 2000. 21st Century Military Communications Conference Proceedings*, volume 2, pages 670–674 vol.2, 2000. doi: 10.1109/MILCOM.2000.904013.

Y. Pati, R. Rezaiifar, and P. Krishnaprasad. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In *Signals, Systems and Computers, 1993. The 27th Asilomar Conf. on*, pages 40–44 vol.1, Nov 1993. doi: 10.1109/ACSSC.1993.342465.

G. Pfander and H. Rauhut. Sparsity in timeŨfrequency representations. *Journal of Fourier Analysis and Applications*, to appear.

M. Pitt and N. Shephard. Filtering via simulation: auxiliary particle filters. Economics Papers 1997-W13, Economics Group, Nuffield College, University of Oxford, 1999. available at http://ideas.repec.org/p/nuf/econwp/9713.html.

B. Porat and B. Friedlander. Blind equalization of digital communication channels using high-order moments. *Signal Processing, IEEE Transactions on*, 39(2):522 –526, feb 1991. ISSN 1053-587X. doi: 10.1109/78.80846.

E. Punskaya. *Sequential Monte Carlo Methods for Digital Communications*. PhD thesis, Cambridge Univ., Cambridge, U.K., 2003.

L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, pages 257–286, 1989.

J. Reeve and K. Amarasinghe. A parallel viterbi decoder for block cyclic and convolution codes. *Signal Processing*, 86:273–278, 2006.

J. Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, Sept. 1978. URL http://www.sciencedirect.com/science/article/B6V21-47WVRPB-46/1/ca6eb21e11f6bf2826ef3db007a99d37.

B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman filter: particle filters for tracking applications*. Artech House, 2004.

B. Roark. Probabilistic top-down parsing and language modeling. *Comput. Linguist.*, 27(2): 249–276, 2001. ISSN 0891-2017. doi: http://dx.doi.org/10.1162/089120101750300526.

P. Robertson, E. Villebrun, and P. Hoeher. A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain. *Communications, 1995. ICC 95 Seattle, Gateway to Globalization, 1995 IEEE International Conference on*, 2:1009–1013 vol.2, June 1995. doi: 10.1109/ICC.1995.524253.

H. Roufarshbaf and J. Nelson. Bayesian MLSD for multipath Rayleigh fading channels. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 2845–2848, 31 2008-April 4 2008. doi: 10.1109/ICASSP.2008.4518242.

P. Schniter. Low-complexity equalization of OFDM in doubly selective channels. *Signal Processing, IEEE Transactions on*, 52(4):1002–1011, 2004. ISSN 1053-587X. doi: 10.1109/TSP.2004.823503.

G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6, No. 2:461–464, 1978.

N. Seshadri. Joint data and channel estimation using blind trellis search techniques. *Communications, IEEE Transactions on*, 42(234):1000–1011, Feb/Mar/Apr 1994. ISSN 0090-6778. doi: 10.1109/TCOMM.1994.580208.

N. Seshadri and C.-E. Sundberg. List Viterbi decoding algorithms with applications. *Communications, IEEE Transactions on*, 42(234):313–323, Feb/Mar/Apr 1994. ISSN 0090-6778. doi: 10.1109/TCOMM.1994.577040.

A. Shapiro. Asymptotic distribution of test statistics in the analysis of moment structures under inequality constraints. *Biometrika*, 72(1):133–144, 1985. doi: 10.1093/biomet/72.1.133. URL http://biomet.oxfordjournals.org/cgi/content/abstract/72/1/133.

M. Sharp and A. Scaglione. Estimation of sparse multipath channels. In *Proc. IEEE Military Communications Conference (MILCOM)*, pages 1–7, November 2008.

S. Simmons. Breadth-first trellis decoding with adaptive effort. *Communications, IEEE Transactions on*, 38(1):3–12, Jan 1990. ISSN 0090-6778. doi: 10.1109/26.46522.

D. Slock. Blind fractionally-spaced equalization, perfect-reconstruction filter banks and multichannel linear prediction. In *Acoustics, Speech, and Signal Processing, 1994 IEEE International Conference on*, volume IV, pages 585 – 588, 1994.

G. Tauböck and F. Hlawatsch. A compressed sensing technique for OFDM channel estimation in mobile environments: Exploiting channel sparsity for reducing pilots. In *Acoustics, Speech and Signal Processing, 2008. IEEE International Conference on*, pages 2885–2888, 2008.

G. Tauböck, F. Hlawatsch, D. Eiwen, and H. Rauhut. Compressive estimation of doubly selective channels in multicarrier systems: Leakage effects and sparsity-enhancing processing. *Selected Topics in Signal Processing, IEEE Journal of*, 4(2):255 –271, april 2010. ISSN 1932-4553. doi: 10.1109/JSTSP.2010.2042410.

G. Taubock, F. Hlawatsch, D. Eiwen, and H. Rauhut. Compressive estimation of doubly selective channels in multicarrier systems: Leakage effects and sparsity-enhancing processing. *Selected Topics in Signal Processing, IEEE Journal of*, 4(2):255 –271, april 2010. ISSN 1932-4553. doi: 10.1109/JSTSP.2010.2042410.

R. Tibshirani. Regression shrinkage and selection via the Lasso. *J. Royal. Statist. Soc B.*, 58: 267–288, 1996.

M. E. Tipping. Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244, 2001.

J. Tropp. Greed is good: algorithmic results for sparse approximation. *Information Theory, IEEE Transactions on*, 50(10):2231–2242, Oct. 2004. ISSN 0018-9448. doi: 10.1109/TIT.2004.834793.

J. A. Tropp. Recovery of short, complex linear combinations via l1 minimization. *Information Theory, IEEE Transactions on*, 51(4):1568 –1570, april 2005. ISSN 0018-9448. doi: 10.1109/TIT.2005.844057.

D. Tse and P. Viswanath. *Fundamentals of Wireless Communication.* Cambridge University Press, 2005.

J. Tugnait. Detection and estimation for abruptly changing systems. *Decision and Control including the Symposium on Adaptive Processes, 1981 20th IEEE Conference on*, 20:1357–1362, December 1981. doi: 10.1109/CDC.1981.269460.

J. Tugnait. Identification of linear stochastic systems via second- and fourth-order cumulant matching. *Information Theory, IEEE Transactions on*, 33(3):393 – 407, may 1987. ISSN 0018-9448.

J. Tugnait, S. He, and H. Kim. Doubly selective channel estimation using exponential basis models and subblock tracking. *Signal Processing, IEEE Transactions on*, 58(3):1275 –1289, march 2010. ISSN 1053-587X. doi: 10.1109/TSP.2009.2036047.

G. Tzagkarakis and P. Tsakalides. Bayesian compressed sensing imaging using a Gaussian scale mixture. *submitted in IEEE Int. Conf. on Acoustics, Speech and Sig. Proc. (ICASSP)*, 2010.

J. van de Beek, M. Sandell, and P. Borjesson. ML estimation of time and frequency offset in OFDM systems. *Signal Processing, IEEE Trans. on*, 45(7):1800–1805, 1997. ISSN 1053-587X. doi: 10.1109/78.599949.

A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *Information Theory, IEEE Transactions on*, 13(2):260–269, April 1967. ISSN 0018-9448.

W. Wang and Q. Zhu. Sequential Monte Carlo localization in mobile sensor networks. *Wireless Networks*, 15(4):481–495, 2009. URL http://dx.doi.org/10.1007/s11276-007-0064-3.

Y. Wang, E. Serpedin, P. Ciblat, and P. Loubaton. Blind cyclostationary statistics based carrier frequency offset and symbol timing delay estimators in flat-fading channels. In *Military Communications Conference, 2001. MILCOM 2001. Communications for Network-Centric Operations: Creating the Information Force. IEEE*, volume 2, pages 1389 – 1393 vol.2, 2001. doi: 10.1109/MILCOM.2001.986083.

W. Wei and J. Mendel. Maximum-likelihood classification for digital amplitude-phase modulations. *Communications, IEEE Transactions on*, 48(2):189 –193, feb 2000. ISSN 0090-6778. doi: 10.1109/26.823550.

G. Xu, H. Liu, L. Tong, and T. Kailath. A least-squares approach to blind channel identification. *Signal Processing, IEEE Transactions on*, 43(12):2982 –2993, dec 1995. ISSN 1053-587X. doi: 10.1109/78.476442.

D. Yee, J. P. Reilly, and T. Kirubarajan. A blind sequential Monte Carlo detector for OFDM systems in the presence of phase noise, multipath fading, and channel order uncertainty. *Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on]*, 55(9):4581–4598, 2007. ISSN 1053-587X.

List of Figures

List of Tables