



HAL
open science

Modeling the variability of electrical activity in the brain

Sebastian Hitziger

► **To cite this version:**

Sebastian Hitziger. Modeling the variability of electrical activity in the brain. Other. Université Nice Sophia Antipolis, 2015. English. NNT: 2015NICE4015 . tel-01175851

HAL Id: tel-01175851

<https://theses.hal.science/tel-01175851>

Submitted on 13 Jul 2015

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

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

PhD THESIS

prepared at
Inria Sophia Antipolis

and presented at the
University of Nice-Sophia Antipolis
Graduate School of Information and Communication
Sciences

*A dissertation submitted in partial fulfillment
of the requirements for the degree of*

DOCTOR OF SCIENCE

Specialized in Control, Signal and Image Processing

Modeling the Variability of Electrical Activity in the Brain

Sebastian HITZIGER

<i>Advisors</i>	Théodore Papadopoulo Maureen Clerc	Inria Sophia Antipolis, France Inria Sophia Antipolis, France
<i>Reviewers</i>	Bruno Torrèsani Alain Rakotomamonjy	University of Aix-Marseille, France University of Rouen, France
<i>Examiners</i>	Laure Blanc-Féraud Christian G. Bénar	Inria Sophia Antipolis, France University of Aix-Marseille, France
<i>Invited</i>	Jean-Louis Divoux	Axononic, Vallauris, France

UNIVERSITÉ DE NICE-SOPHIA ANTIPOLIS

ÉCOLE DOCTORALE STIC
Sciences et Technologies de
l'Information et de la Communication

THÈSE

pour l'obtention du grade de

DOCTEUR EN SCIENCES

de l'Université de Nice-Sophia Antipolis

Discipline: Automatique, Traitement du Signal et des Images

présentée et soutenue par

Sebastian HITZIGER

Modélisation de la variabilité de l'activité électrique dans le cerveau

Date de soutenance : 14 avril 2014

Composition du jury:

<i>Directeurs</i>	Théodore Papadopoulo Maureen Clerc	Inria Sophia Antipolis, France Inria Sophia Antipolis, France
<i>Rapporteurs</i>	Bruno Torrèsani Alain Rakotomamonjy	Université d'Aix-Marseille, France Université de Rouen, France
<i>Examineurs</i>	Laure Blanc-Féraud Christian G. Bénar	Inria Sophia Antipolis, France Université d'Aix-Marseille, France
<i>Invité</i>	Jean-Louis Divoux	Axonic, Vallauris, France

To my grandmother.

Abstract

This thesis investigates the analysis of brain electrical activity. An important challenge is the presence of large variability in neuroelectrical recordings, both across different subjects and within a single subject, for example across experimental trials. We propose a new method called adaptive waveform learning (AWL). It is general enough to include all types of relevant variability empirically found in neuroelectric recordings, but it can be specialized for different concrete settings to prevent from overfitting irrelevant structures in the data.

The first part of this work gives an introduction into the electrophysiology of the brain, presents frequently used recording modalities, and describes state-of-the-art methods for neuroelectrical signal processing.

The main contribution of the thesis consists in three chapters introducing and evaluating the AWL method. We first provide a general signal decomposition model that explicitly includes different forms of variability across signal components. This model is then specialized for two concrete applications: processing a set of segmented experimental trials and learning repeating structures across a single recorded signal. Two algorithms are developed to solve these models. Their efficient implementations, based on alternate minimization and sparse coding techniques, allow the processing of large datasets.

The proposed algorithms are evaluated on both synthetic data and real data containing epileptiform spikes. Their performances are compared to those of PCA, ICA, and template matching for spike detection.

Keywords: electroencephalography (EEG), event-related potentials (ERP), epileptiform spikes, signal variability, dictionary learning, sparse coding

ABSTRACT

Résumé

Cette thèse explore l'analyse de l'activité électrique du cerveau. Un défi important de ces signaux est leur grande variabilité à travers différents essais et/ou différents sujets. Nous proposons une nouvelle méthode appelée *adaptive waveform learning* (AWL). Cette méthode est suffisamment générale pour permettre la prise en compte de la variabilité empiriquement rencontrée dans les signaux neuroélectriques, mais peut être spécialisée afin de prévenir l'overfitting du bruit.

La première partie de ce travail donne une introduction sur l'électrophysiologie du cerveau, présente les modalités d'enregistrement fréquemment utilisées et décrit l'état de l'art du traitement de signal neuroélectrique.

La principale contribution de cette thèse consiste en 3 chapitres introduisant et évaluant la méthode AWL. Nous proposons d'abord un modèle de décomposition de signal général qui inclut explicitement différentes formes de variabilité entre les composantes de signal. Ce modèle est ensuite spécialisé pour deux applications concrètes: le traitement d'une série d'essais expérimentaux segmentés et l'apprentissage de structures répétées dans un seul signal. Deux algorithmes sont développés pour résoudre ces problèmes de décomposition. Leur implémentation efficace basée sur des techniques de minimisation alternée et de codage parcimonieux permet le traitement de grands jeux de données.

Les algorithmes proposés sont évalués sur des données synthétiques et réelles contenant des pointes épileptiformes. Leurs performances sont comparées à celles de l'ACP, l'ICA, et du template-matching pour la détection des pointes.

Mots clés : électroencephalographie (EEG), potentiels évoqués (ERP), pointes épileptiformes, variabilité du signal, apprentissage de dictionnaires, codage parcimonieux

RÉSUMÉ

Acknowledgements

First of all, I would like to express my deep gratitude towards my supervisors Théo Papadopoulo and Maureen Clerc for accompanying me throughout the past three years with their expertise, thoughtful advice, and encouraging support. I would also like to thank Rachid Deriche and the Athena team at Inria for welcoming me and giving me such an inspiring environment.

I am very grateful to Alain Rakotomamonjy and Bruno Torr sani for taking a significant amount of their time to review this thesis and share with me their insightful suggestions and remarks. Furthermore, I would like to thank Laure Blanc-F raud and Christian B nar for their participation in the jury.

Much of the present work grew out of ideas coming from the fruitful exchange with the members of the two ANR¹ projects Coadapt and Multimodel. In particular, I would like to thank: Christian B nar for his great enthusiasm which encouraged me to pursue the detailed investigation of neurovascular coupling; Sandrine Sallet and the experimentalists for the great and rich datasets which allowed us to obtain many interesting findings; Bruno Torr sani for inspiring discussions on various signal processing topics; Alexandre Gramfort for his helpful suggestions on dictionary learning.

I am extremely grateful to all my friends and colleagues in the Athena and Neuromathcomp team who made this PhD a wonderful time: Emmanuel O., Emmanuel C., Sylvain, Anne-Charlotte, Joan, Auro, Dieter, Eoin, Hassan, Romain, James R., James I., Antoine, Gabriel and Sara, Elodie, Gonzalo, Kartheek, Li, Nathan l, Rutger, Marco, Demian, Lola. Very special thanks go to Christos for discussing multi-channel dictionary learning and sharing his results; to Jaime, Brahim, and Asya for putting up with me as an office mate; to Claire for kindly helping me through all those administrative issues; to James M. for those deep philosophical discussions and being the best flatmate one could imagine; and to my dear friend Rodrigo for sharing with me the pleasant bike rides to the office.

Finally, I would like to thank my family for all their love and support throughout this thesis.

¹L'Agence Nationale de la Recherche

ACKNOWLEDGEMENTS

Contents

Introduction en français	1
Contributions	3
Sommaire	4
Notations	8
1 Introduction	9
1.1 Contributions	10
1.2 Outline	12
1.3 Notation	14
2 Analyzing Brain Electrical Activity	15
2.1 Introduction	16
2.2 Electrophysiology of the brain	16
2.3 Recording electromagnetic activity	18
2.3.1 Electroencephalography (EEG)	18
2.3.2 Intracranial recordings	19
2.3.3 Magnetoencephalography (MEG)	21
2.3.4 Complementary measurements of EEG and MEG	21
2.4 Recording metabolism and hemodynamics	22
2.5 Neuroelectrical signals	23
2.5.1 Event-related potentials (ERPs)	24
2.5.2 Spontaneously repeating activity (SRA)	25
2.6 Signal analysis: goals and challenges	27
2.6.1 Assessing variability	27
2.6.2 Compact and interpretable representation	28
2.6.3 Flexible algorithm	29
2.7 Example dataset: multi-modal recordings	29
3 Neuroelectrical Signal Processing	33
3.1 Introduction and overview	34
3.2 Ensemble averaging	35

3.3	Models with temporal variability	37
3.3.1	Woody's method	37
3.3.2	Dynamic time warping	38
3.3.3	Variable response times	40
3.4	Linear multi-component models	40
3.4.1	Principal component analysis (PCA)	41
3.4.2	Independent component analysis (ICA)	42
3.5	Combined approaches	45
3.5.1	Differentially variable component analysis (dVCA)	45
3.6	Sparse representations	46
3.6.1	Sparse coding and time-frequency representations	46
3.6.2	Dictionary learning	50
3.6.3	Translation-invariant dictionary learning	50
3.7	Conclusion and outlook	52
4	Adaptive Waveform Learning (AWL)	53
4.1	Introduction	54
4.2	Modeling variability	54
4.2.1	Linear operations on signal components	55
4.2.1.1	Translations	56
4.2.1.2	Dilations	57
4.3	AWL model	57
4.3.1	General model	57
4.3.2	Sparse coefficients	58
4.3.3	Minimization problem	59
4.3.4	Explicit representation	59
4.4	Algorithm AWL	60
4.4.1	Initialization	60
4.4.2	Coefficient updates	61
4.4.3	Waveform updates	62
4.4.4	Waveform centering	63
4.5	Relation to previous methods	64
4.6	Hyperparameters and hierarchical AWL	65
5	Epoched AWL	67
5.1	Introduction	68
5.2	Model specifications	68
5.2.1	Explicit representation	69
5.3	Algorithm E-AWL	70
5.3.1	Coefficient updates	70
5.3.2	Waveform updates	71
5.3.3	Waveform centering	71
5.3.4	Discrete shifts and boundary issues	72
5.4	Evaluation on simulated data	73

5.4.1	Influence of varying amplitudes	76
5.4.2	Influence of varying latencies	76
5.4.3	Robustness to noise	78
5.4.4	Qualitative comparison	78
5.5	Evaluation on LFP recording	79
5.5.1	Preprocessing and epoching	80
5.5.2	Hierarchical representations with epoched AWL	80
5.5.3	Comparison of PCA, ICA, and AWL	82
6	Contiguous AWL	85
6.1	Introduction	86
6.2	Model specifications	86
6.3	Algorithm C-AWL	88
6.3.1	Coefficient updates	88
6.3.2	Waveform updates	89
6.3.3	Waveform centering	90
6.4	C-AWL for spike learning	91
6.4.1	Multi-class spike learning (MC-Spike)	92
6.4.1.1	Handling overlapping spikes	94
6.4.1.2	Initializing new spikes	94
6.4.1.3	Performance measures	95
6.4.1.4	Experiments	97
6.4.2	Adaptive duration spike learning (AD-Spike)	100
6.4.3	Comparing detection performance	102
6.5	Exploring neuro-vascular coupling	105
7	Conclusion	109
7.1	Strengths and weaknesses of AWL	110
7.2	Future work	111
	Conclusion en français)	115
	Points forts et faibles de AWL	116
	Travail futur	117
A	Implementation of MP and LARS	121
B	Publications of the Author	127

CONTENTS

List of Figures

1	Structure du document et dépendances entre les chapitres. . .	5
1.1	Document structure and dependencies between chapters . . .	12
2.1	Electrochemical communication between two neurons	16
2.2	Pyramidal neuron assemblies generate electromagnetic field .	17
2.3	First recording of a human EEG	18
2.4	EEG recording session and visualization on the scalp	19
2.5	EEG recording during epileptic seizure	20
2.6	Conoluted arrangement of the cerebral cortex	21
2.7	Spatial and temporal resolution of brain imaging techniques .	23
2.8	Generation of a P300 event-related potential	25
2.9	Brain activity during different sleep stages	26
2.10	Simultaneous LFP and CBF recording	30
3.1	Averaging over waveforms with latency jitter	36
3.2	Illustration of Woody’s method	38
3.3	Dynamic time warping: illustration of the path	39
3.4	Dynamic time warping applied to auditory evoked responses .	40
3.5	Artifact identification through ICA	44
3.6	Illustration of Gabor atoms	47
3.7	Enhancing induced activity through averaging in time-frequency	49
3.8	Algorithm scheme for dictionary learning	51
5.1	Illustration of waveform shifts on bounded domain	72
5.2	Simulated waveforms for synthetic experiment	73
5.3	Illustration of simulated noisy trials	74
5.4	Error curves for PCA, ICA, and E-AWL	77
5.5	Waveforms recoverd with PCA, ICA, and E-AWL	79
5.6	Illustration of epoched spikes from original dataset	80
5.7	Hierarchical waveform representation learned with E-AWL . .	81
5.8	Representations learned with PCA, ICA, and E-AWL	83

LIST OF FIGURES

6.1	Illustration of waveform shifts in a long signal	87
6.2	Two time windows of a typical spike	93
6.3	Subtracting overlapping spikes with matching pursuit	94
6.4	Spike representations learned with MC-Spike on LFP dataset	96
6.5	Spike representation with MC-Spike (five spike classes)	98
6.6	Dilatable spike representation learned with AD-Spike	101
6.7	Qualitative comparison between MC-Spike and AD-Spike	102
6.8	Detection accuracies: MC-Spike, AD-Spike, template matching	104
6.9	Epileptiform spikes synchronizing with CBF activity	106
7.1	Multi-channel extension	112
7.2	Extension multicanaux	119

List of Algorithms

1	Generic AWL	60
2	Hierarchical AWL	65
3	E-AWL	69
4	C-AWL	88
5	MC-Spike	93
6	AD-Spike	100
7	Matching pursuit	122

LIST OF ALGORITHMS

List of Acronyms

AD-Spike	Adaptive Duration Spike Learning
ANR	L'Agence Nationale de la Recherche
AP	Action Potential
AWL	Adaptive Waveform Learning
BCI	Brain Computer Interface
BOLD	Blood Oxygenation Level Dependent
BP	Bereitschaftspotential
BSS	Blind Source Separation
C-AWL	Contiguous AWL
CBF	Cerebral Blood Flow
DL	Dictionary Learning
DTW	Dynamic Time Warping
dVCA	Differentially Variable Component Analysis
E-AWL	Epoched AWL
ECoG	Electrocorticography
EEG	Electroencephalography
ERP	Event-Related Potential
FFT	Fast Fourier Transform
fMRI	functional Magnetic Resonance Imaging
GOF	Goodness of Fit
ICA	Independent Component Analysis
JADE	Joint Approximate Diagonalization of Eigenmatrices
LARS	Least Angle Regression
Lasso	Least Absolute Shrinkage and Selection Operator
LDF	Laser Doppler Flowmetry
LFP	Local Field Potential
LSR	Local Spiking Rate
MC-Spike	Multiclass Spike Learning
MEG	Magnetoencephalography
MOD	Method of Optimal Directions

LIST OF ACRONYMS

MP	Matching Pursuit
MUA	Multiple-Unit Spike Activity
NIRS	Near-Infrared Spectroscopy
OMP	Orthogonal Matching Pursuit
PCA	Principal Component Analysis
PET	Positron Emission Tomography
PSP	Postsynaptic Potential
sEEG	Stereoelectroencephalography
SNR	Signal-to-Noise Ratio
SPECT	Single Photon Emission Computed Tomography
SQUID	Superconducting Quantum Interference Device
SRA	Spontaneously Repeating Activity
SSVEP	Steady-State Visual-Evoked Potential
SVD	Singular Value Decomposition
VQ	Vector Quantization

Introduction en français

En 2013, deux projets de recherche d'une durée de 10 ans avec plus d'un milliard d'euros de budget chacun, le *Human Brain Project* et le *BRAIN Initiative*, ont été lancés respectivement par l'UE et les États-Unis. Visant à simuler le cerveau humain à l'aide de supercalculateurs, ces projets témoignent du grand intérêt scientifique à une meilleure compréhension des mécanismes sous-jacents au cerveau.

Les avancées résultant du progrès dans la recherche en neurosciences ont été nombreuses. Outre l'amélioration constante du traitement des troubles mentaux étudiés depuis longtemps comme l'épilepsie, la démence, la maladie de Parkinson ou la dépression, le développement technologique a conduit à de nombreuses nouvelles applications des neurosciences. À titre d'exemple, les interfaces cerveau-ordinateur (BCI) permettent d'utiliser les ondes cérébrales d'un utilisateur pour communiquer avec un dispositif externe. Le BCI peut ainsi aider les personnes handicapées ou non dans l'exécution de diverses tâches cognitives ou sensori-motrices.

À bien des égards, ces progrès sont dûs à la large gamme de modalités d'imagerie cérébrale qui sont aujourd'hui disponibles, comprenant (parmi plusieurs autres) l'électro- et la magnétoencéphalographie (EEG / MEG), l'imagerie par résonance magnétique fonctionnelle (IRMf), la tomographie par émission des positons (TEP), la tomographie d'émission monophotonique (TEMP), et la spectroscopie proche infrarouge (NIRS). Avec la possibilité d'acquérir à haute résolution des enregistrements multimodaux vient aussi le défi de l'interprétation de ces données de manière adéquate. Cela nécessite tout d'abord la conception de modèles physiologiques expliquant la génération de ces signaux. Des méthodes sophistiquées de traitement du signal sont ensuite nécessaires pour, sur la base de ces modèles, représenter des signaux avec des informations significatives et statistiquement pertinentes.

Le présent travail se concentre sur l'analyse des enregistrements neuroélectriques. L'une des techniques fondamentales de traitement de ces signaux se base sur la moyenne d'un grand nombre de signaux enregistrés dans des conditions similaires. Cette méthode, introduite par George D. Daw-

son (Dawson, 1954), s'est montrée utile pour détecter les potentiels évoqués (ERP) de petites amplitudes, qui seraient cachés sous le bruit et l'activité de fond des neurones avec un seul enregistrement. Bien que cette technique soit encore souvent utilisée pour analyser les ERPs, les hypothèses sous-jacentes au calcul de la moyenne ont été contestées et montrées inexactes dans de nombreux contextes (Coppola et al., 1978; Horvath, 1969; Truccolo et al., 2002).

Un problème principal est l'observation que les réponses neurales ont tendance à varier en amplitude, en latence et même en forme (Jung et al., 2001; Kisley and Gerstein, 1999). Cette variabilité existe entre différents sujets, mais aussi pour un même sujet entre différents essais expérimentaux.

Compenser la variabilité des signaux du cerveau peut permettre une meilleure caractérisation de la réponse neurale stéréotypée à un phénomène sous-jacent. En outre, la description et la quantification de la variabilité, inter ou intra-sujet, sont des sources précieuses d'information. Par exemple, la diminution des amplitudes de la réponse à travers une session d'enregistrement peut être un signe d'effets de fatigue ou d'accoutumance du sujet. Une des premières extensions de l'approche de Dawson est la méthode de Woody (Woody, 1967) qui prend en compte différentes latences pendant le calcul de la moyenne. Ces dernières décennies, un large éventail de modèles et de techniques a été proposé pour tenir compte des différents aspects de variabilité. Les techniques de décomposition linéaire comme l'ACP et l'ICA peuvent compenser des changements en amplitude alors que d'autres techniques comme le *dynamic time warping* considèrent des déformations plus générales de la forme du signal.

Outre la comptabilisation adéquate de la variabilité, il y a plusieurs autres caractéristiques qu'un outil idéal de traitement de signal neuroélectrique devrait posséder. Les enregistrements, qui sont souvent acquis dans plusieurs dimensions (par exemple, de multiples essais expérimentaux, canaux ou modalités), nécessitent l'extraction efficace d'informations pertinentes dans une représentation compacte qui permet une interprétation facile. Il est indispensable d'incorporer dans le modèle toutes les informations préalables afin d'éviter l'extraction de structures de signaux non pertinentes. D'un autre côté, les formes exactes des réponses neurales sont souvent inconnues, ce qui rend nécessaire l'apprentissage de ces formes d'onde aussi *aveuglement* (c'est-à-dire, sans *a priori*) que possible. En raison de ces différentes exigences complémentaires, la conception d'une méthode optimale est une tâche difficile.

L'objectif de cette thèse est de fournir un cadre de décomposition des signaux modélisant explicitement la variabilité rencontrée dans les enregistrements neuroélectriques. Grâce à sa formulation générale, ce cadre peut être adapté à une variété de différentes tâches de traitement du signal.

Contributions

Le résultat principal de cette thèse est la conception d'une méthode, appelé *adaptive waveform learning* (AWL), pour traiter des signaux neuroélectriques. La nouveauté de cette approche est la modélisation explicite de la variabilité des composantes des signaux par des transformations mathématiques.

La méthode AWL est tout d'abord présentée au Chapitre 4 de façon théorique. On considère un modèle très général de décomposition où la variabilité est représentée par des transformations linéaires arbitraires. Ce modèle sert de cadre général pour différentes applications neuroélectriques et peut être vu comme une généralisation de différents modèles existants. Dans ce cadre, nous ne fournissons pas d'implémentation concrète. Cependant, nous présentons un algorithme générique qui peut être utilisé comme une recette pour implémenter de telles applications concrètes. Comme AWL repose sur un modèle parcimonieux, l'algorithme générique peut être efficacement implémenté par minimisation alternée.

AWL est ensuite spécialisé dans les chapitres 5 et 6 pour deux applications fréquemment rencontrées dans l'analyse du signal neuroélectrique : le traitement des signaux *époqués* (c'est-à-dire segmentés, par exemple les potentiels évoqués) et le traitement des signaux contigus (c'est-à-dire non segmentés, par exemple des pointes épileptiformes). Les deux implémentations résultantes, E-AWL et C-AWL, sont évaluées sur des données simulées et réelles et comparées à l'APC et l'ICA. Les expériences montrent l'utilité de E-AWL et C-AWL comme des outils robustes de traitement du signal qui sont capables d'apprendre des représentations intéressantes et originales à partir des données.

Les deux spécialisations sont seulement des exemples d'implémentations possibles de AWL. Différentes variantes, comme une extension pour le traitement des enregistrements sur multiples canaux, peuvent aussi être dérivées en faisant des ajustements appropriés au cadre général. Cette question sera abordée dans le Chapitre 7.

En résumé, cette thèse propose les contributions suivantes :

- Chapitre 4: Introduction de la nouvelle méthode AWL dans un cadre général et conception d'un algorithme générique.
- Chapitre 5: Spécialisation de AWL pour traiter des enregistrements *époqués* (E-AWL) et implémentation à l'aide d'une modification de *least angle regression* (LARS). En particulier, E-AWL compense des différences de latences ainsi que des différences de phases à travers les *époques* et peut séparer les différentes composantes du signal en exploitant cette variabilité.
- Chapitre 6: Spécialisation de AWL pour traiter des enregistrements

contigus (C-AWL) et implementation efficace à l'aide de l'algorithme *matching pursuit* (MP). C-AWL est utilisé pour dériver deux techniques d'apprentissage de pointes épileptiformes : MC-Spike, qui fait l'hypothèse de différentes classes de pointes avec des formes immuables, et AD-Spike, qui représente les pointes au travers d'une seule forme de pointe de durée adaptative. Ces techniques permettent la détection des pointes ainsi que l'apprentissage de leurs formes et leurs variabilités à travers des données.

- Découvertes intéressantes dans le couplage de l'activité de pointes et des rythmes dans l'hémodynamique (Section 6.5). Ces découvertes résultent des représentations apprises avec MC-Spike et AD-Spike.
- Disponibilité du code pour les différentes méthodes et les expériences en ligne : <https://github.com/hitziger/AWL>.

Sommaire

Le contenu de cette thèse est structuré en trois parties : une partie introductive (chapitres 2 et 3), la partie principale (chapitres 4 - 6) et la conclusion (Chapitre 7). Le contenu de chacun des ces chapitres est résumé dans les paragraphes suivants. Afin de donner une meilleure orientation au lecteur, nous illustrons la structure du présent document dans l'organigramme Figure 1 qui montre les dépendances entre les chapitres.

Chapitre 2

Ce chapitre donne une introduction à l'analyse de l'activité électrique dans le cerveau. Nous commençons par expliquer des concepts fondamentaux de la génération des signaux électromagnétiques par des ensembles de neurones, qui sont mesurables à différents niveaux du cerveau. Plusieurs modalités pour enregistrer l'électromagnétisme, le métabolisme et l'hémodynamique dans le cerveau sont présentées. Nous donnons ensuite un aperçu des différentes caractéristiques des enregistrements neuroélectriques et définissons des défis et objectifs pour traiter ces signaux. Le chapitre conclut par la présentation d'une étude multimodale, conduite sur des rats anesthésiés et visant à l'exploration du couplage entre l'activité neuroélectrique et hémodynamique. Ce jeu de données est analysé plus finement ultérieurement (chapitres 5 et 6) avec la nouvelle méthode AWL.

Chapitre 3

Ce chapitre présente une revue des techniques de traitement du signal neuroélectrique, avec une attention particulière sur la variabilité à travers les signaux. Le chapitre commence en présentant une méthode de moyennage

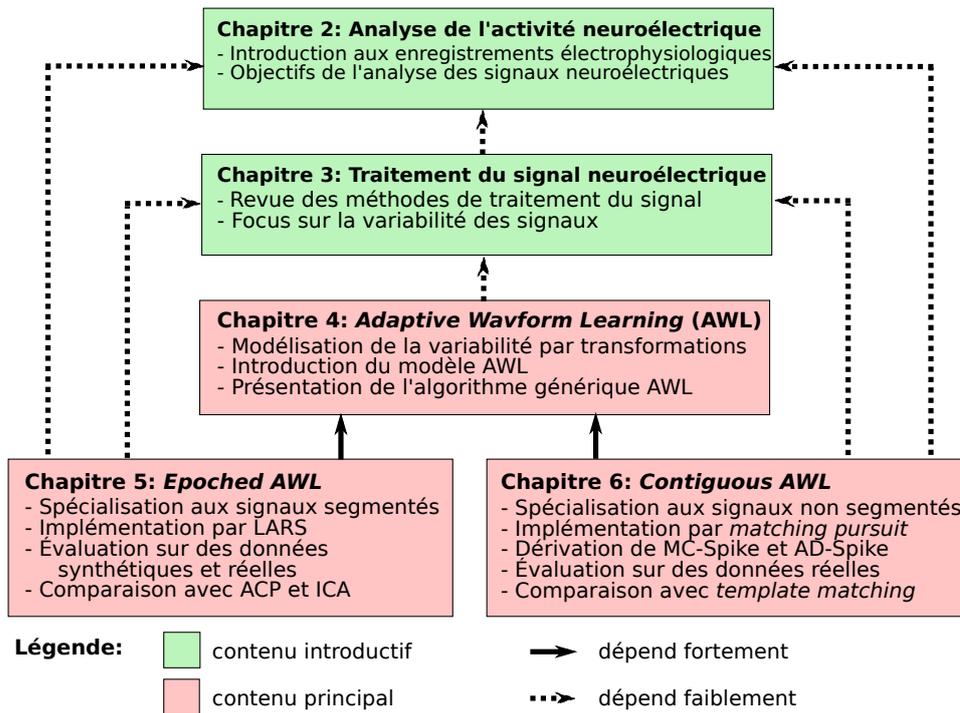


Figure 1: Structure du document et dépendances entre des chapitres. Le matériel introductif (vert) fournit les bases physiologiques et méthodologiques pour le traitement du signal neuroélectrique. La contribution principale de cette thèse, la méthode AWL, est introduite dans le chapitre 4 et spécialisée pour des signaux *époqués* et non *époqués* dans les chapitres 5 et 6, respectivement.

pour les potentiels évoqués introduite dans Dawson (1954). Nous présentons ensuite différentes extensions de cette méthode qui prennent en compte différents types de variabilité temporelle, telles que la compensation de différences de latences dans la méthode de Woody (Woody, 1967). Une autre approche utilise des modèles linéaires avec différentes composantes des signaux et mène aux méthodes ACP et ICA. Un autre groupe d'outils pour le traitement des signaux neuroélectriques consiste en représentations parcimonieuses, souvent en domaine temps-fréquence. Enfin, quelques approches intégratives sont présentées. Celles-ci permettent de prendre en compte différents types de variabilité dans les données multicanaux.

Chapitre 4

Ce chapitre est le premier des trois chapitres principaux qui présentent la nouvelle méthode *adaptive waveform learning* (AWL). Le modèle sous-jacent comprend explicitement des types généraux de variabilité des signaux. Cette variabilité est décrite par une famille d'opérateurs linéaires $\{\phi_p\}$ agissant sur les composantes des signaux. Concrètement, chaque signal \mathbf{x}_m est modélisé comme combinaison linéaire de formes d'onde \mathbf{d}_k modifiées par les transformations linéaires ϕ_p et s'écrit

$$\mathbf{x}_m = \sum_k \sum_p a_{kpm} \phi_p(\mathbf{d}_k) + \epsilon_m,$$

avec un bruit additif ϵ_m . L'apprentissage des quantités inconnues s'effectue par la minimisation d'un terme d'attache aux données quadratique plus un *a priori* de parcimonie, sous la forme

$$\min_m \sum \left(\left\| \mathbf{x}_m - \sum_k \sum_p a_{kpm} \phi_p(\mathbf{d}_k) \right\|^2 + \lambda \sum_{k,p} |a_{kpm}| \right).$$

De plus, nous considérons diverses autres contraintes de parcimonie qui doivent être adaptées au contexte considéré. Nous proposons un algorithme générique qui procède par minimisation alternée sur les formes d'onde \mathbf{d}_k et des coefficients de la regression a_{kpm} . Grâce aux hypothèses de parcimonie, l'estimation des coefficients peut être effectué efficacement par des techniques de codage parcimonieux. Après avoir introduit l'algorithme générique, nous examinons en détail ses différentes étapes. Celles-ci comprennent aussi l'initialisation des formes d'onde et l'estimation de l'ordre du modèle.

Le chapitre demeure dans un cadre général de sorte que AWL peut être considéré comme un métamodèle pour plusieurs des méthodes présentées dans le Chapitre 3. Cette généralité permet de dériver différents algorithmes concrets pour des applications spécifiques, telles que celles présentées dans les chapitres 5 et 6.

Chapitre 5

Ce chapitre aborde le traitement des signaux *époqués* (c'est-à-dire des signaux segmentés) qui contiennent des potentiels évoqués (ERPs) ou de l'activité répétée. À cette fin, l'algorithme général AWL du chapitre précédent est spécialisé pour calculer les composantes du signal aux latences variables à travers les *époques*. L'algorithme résultant, appelé E-AWL, est implémenté en adaptant le cadre itératif de AWL : à partir d'un ensemble de formes d'onde $\{\mathbf{d}_k\}$ initialisées par du bruit gaussien, ces formes et leurs amplitudes a_{kpm} et latences δ_{kpm} sont mises à jour efficacement à chaque étape de l'algorithme par la technique du *block coordinate descent* et LARS (décrit en annexe), respectivement.

E-AWL est d'abord évalué sur des données synthétiques pour différentes magnitudes de variabilité des amplitudes et des latences ainsi que pour différents niveaux de bruit. Les résultats sont comparés à ceux de l'analyse en composantes principales (ACP) et de l'analyse en composantes indépendantes (ICA). Comme attendu, E-AWL est capable de détecter et compenser la variabilité de latence, bien mieux que les deux autres approches.

Nous illustrons ensuite la capacité de E-AWL à produire des représentations pertinentes sur les données réelles présentées dans le Chapitre 2. À cette fin, nous proposons une version hiérarchique de E-AWL qui apprend un nombre croissant de formes d'onde. Comparé à ACP et ICA, E-AWL montre des résultats supérieurs en ce qui concerne la séparation des pointes épileptiformes et des formes d'onde oscillantes.

Chapitre 6

Ce chapitre décrit une approche alternative à celle de E-AWL en traitant des signaux contigus, c'est-à-dire sans les *époquer*. L'algorithme correspondant, appelé AWL contigu (C-AWL), peut lui aussi être dérivé de AWL en faisant des spécialisations appropriées.

Le traitement sans *époquage* préalable est plus difficile, car les latences des signaux d'intérêt sont inconnues et doivent être détectées à travers le signal entier. Ceci est abordé au travers d'une implémentation efficace basée sur *matching pursuit* (MP). Dans le cas présent, l'initialisation des formes d'onde est extrêmement importante afin d'éviter la détection de composantes non pertinentes.

L'implémentation concrète est illustrée sur l'exemple de deux modèles de pointes épileptiformes : MC-Spike, basé sur différentes classes de formes de pointes, et AD-Spike, qui consiste en un seul *template* de pointe de durée variable. Ces deux algorithmes sont appliqués aux pointes épileptiformes traitées dans le Chapitre 5, cette fois sans les *époquer a priori*. Un avantage d'éviter l'*époquage* est la possibilité de traiter des pointes superposées, qui étaient rejetées dans l'approche E-AWL. Ceci permet d'avoir une représenta-

tion des données plus complète. Les résultats produits par les deux méthodes sont assez comparables. Dans le cas de MC-Spike, il se pose le problème du choix du nombre de formes d'onde, qui est abordé par une représentation hiérarchique. Ce problème ne se pose pas pour AD-Spike.

AD-Spike et MC-Spike sont ensuite comparés au *template matching* en ce qui concerne la précision de détection dans les données bruitées. Dans la majorité des cas considérés les deux nouvelles méthodes donnent de meilleurs résultats que *template matching*, et AD-Spike est légèrement supérieur à MC-Spike.

Enfin, la représentation apprise avec MC-Spike est utilisée pour révéler des informations sur le couplage avec les données hémodynamiques, qui avaient été enregistrées simultanément aux décharges électriques.

Chapitre 7

Ce chapitre fait un bilan des méthodes développées et des résultats obtenus. Dans une section de futurs travaux, nous décrivons d'autres possibilités pour des applications de la méthode AWL et des extensions, telles qu'une version multicanaux.

Notations

Dans cette thèse, nous allons utiliser des caractères minuscules et gras pour représenter des vecteurs (par exemple $\mathbf{x}, \mathbf{y}, \mathbf{d}$) et des caractères majuscules et gras pour représenter des matrices (par exemple \mathbf{A}, \mathbf{D}). Sauf mention particulière, les vecteurs sont toujours considérés comme vecteurs colonnes. Les caractères minuscules non gras représentent des scalaires, par exemple, les éléments d'un vecteur $\mathbf{x} = (x_1, \dots, x_N)$ ou d'une matrice $\mathbf{A} = (a_{ij})$. Une exception à ces conventions est faite dans le Chapitre 4 où les caractères minuscules et gras $\mathbf{x}, \mathbf{y}, \mathbf{d}$ dénotent des signaux temporels définis sur les nombres réels \mathbb{R} .

Chapter 1

Introduction

In 2013, two 10-year research projects with over one billion Euros budget each, the *Human Brain Project* and the *BRAIN Initiative*, were launched by the EU and the USA, respectively. Aimed at simulating the human brain with the use of supercomputers, these projects are testimony of the great scientific interest in a better understanding of the mechanisms underlying the brain.

In fact, the benefits arising from advances in neuroscientific research have shown to be tremendous. Besides constantly improving treatment options for long-studied mental disorders such as epilepsy, dementia, Parkinson's disease, or depression, recent technological and software development has led to many new neuroscientific applications. As an example, brain computer interfaces (BCIs) allow to establish a direct communication pathway from the user's brain waves to an external device and can assist both abled and disabled persons with various cognitive or sensory-motor tasks.

To a large degree, these advances are owed to the broad range of brain recording and imaging modalities that are nowadays available, including (among many others) electro- and magnetoencephalography (EEG/MEG), functional magnetic resonance imaging (fMRI), positron emission tomography (PET), single photon emission tomography (SPECT), and near-infrared spectroscopy (NIRS). Along with the possibility of acquiring high-resolution multi-modal recordings also comes the challenge of adequately interpreting this data. First, this requires to design appropriate physiological models explaining the generation of these signals. Then, sophisticated signal processing techniques are needed, which, based on these models, can provide signal representations with meaningful and statistically relevant information.

The present work focusses on the analysis of neuroelectrical recordings. One of the fundamental processing techniques for these signals consists in averaging over a large number of recorded signals acquired under similar conditions. This method was introduced by George D. Dawson ([Dawson](#),

1954) and proved useful in detecting event-related potentials (ERPs) of small amplitudes, which, in a single recording, would be buried in noise and neural background activity. Although still widely used to enhance event-related potentials up to present days, the assumptions underlying the process of averaging have been challenged and shown inaccurate in many settings (Coppola et al., 1978; Horvath, 1969; Truccolo et al., 2002).

A principal problem is the observation that the neural responses tend to vary in amplitude, latency, and even shape (Jung et al., 2001; Kisley and Gerstein, 1999). This variability is found across different subjects, but also within a single subject across repeated experimental trials.

Compensating for variability in brain signals has shown to lead to an improved characterization of the stereotypic neural response to some underlying phenomenon. Moreover, the description and quantification of variability, inter- or intra-subject, is a precious source of information. For example, decreasing response amplitudes across a recording session can be a sign of fatigue or habituation effects in the subject. Among the first extensions to Dawson’s approach is Woody’s method (Woody, 1967), which accounts for different latencies during averaging. The past decades produced a wide spectrum of models and techniques that consider different aspects of waveform variability, ranging from linear amplitude changes of different signal components (e.g., PCA, ICA) to very general shape deformations (e.g., dynamic time warping).

Besides adequately accounting for variability, there are many other requirements that an ideal neuroelectrical signal processing tool should meet. The often complex and multi-variate recordings (e.g., multiple experimental trials, channels, or modalities) make it necessary to efficiently extract the relevant information into a compact signal representation that can easily be interpreted. While all prior information should be incorporated in order to avoid fitting irrelevant signal structures, the exact shape of the brain responses are often unknown, making it necessary to learn the relevant wave-shapes as *blindly* as possible. Due to these different and complementary requirements, the design of an optimal method is a challenging task.

The aim of this thesis is to provide a signal decomposition framework which explicitly models and compensates for variability encountered in neuroelectrical recordings. Due to its general formulation, this framework can be adapted to a variety of different signal processing tasks, by making the task-specific specializations in the algorithm.

1.1 Contributions

The main result of this thesis is the design of a method, called adaptive waveform learning (AWL), for processing neuroelectrical signals. The novelty of the approach is the explicit modeling of waveform variability through math-

emational transformations. The AWL method is developed on two different levels:

The first level, presented in Chapter 4, is an analytical framework, which considers a very general signal decomposition model with variability represented through arbitrary linear transformations. This model provides a common framework to different neuroelectrical processing applications and can be seen as a generalization of different existing models. For this analytical setting, we do not provide a concrete implementation. However, we present a generic algorithm that can be used as a recipe in order to implement concrete settings. Thanks to the formulation of AWL as a sparse model, the generic algorithm can be efficiently implemented through alternate minimization based on sparse coding.

On the second level, presented in chapters 5 and 6, general AWL is then specialized to two settings frequently encountered in neuroelectrical signal analysis: processing *epoched* (i.e., segmented) neuroelectrical signals (e.g., event-related potentials) opposed to processing contiguous (i.e., non-segmented) signals (e.g., epileptiform spikes). Both resulting implementations, E-AWL and C-AWL, are evaluated on simulated and real data and compared to PCA, ICA, and template matching. The experiments give proof of the utility of E-AWL and C-AWL as robust signal processing tools that can learn interesting data representations. The two specializations are only examples for possible implementations of AWL. Different variants, such as an extension to multiple recording channels, may be derived in future works by making the appropriate adjustments to the general framework. This will be discussed in the concluding remarks in Chapter 7. In summary, this thesis makes the following contributions:

- Introduction of the novel method AWL in an analytical setting and design of a generic algorithm. See Chapter 4.
- Specialization of AWL to process epoched datasets (E-AWL) and its efficient implementation using a modification of least angle regression (LARS). In particular, E-AWL compensates for latency shifts as well as phase differences across data epochs and can separate different signal components by exploiting this variability. See Chapter 5.
- Specialization of AWL to process contiguous datasets (C-AWL) and its efficient implementation using matching pursuit. C-AWL is used to derive two spike learning techniques: MC-Spike, which assumes different spike classes of constant shapes, and AD-Spike, which models spikes through a single template of adaptive duration. These techniques allow for both spike detection and the learning of interesting representations of the spike shapes and their variability across the data. See Chapter 6.

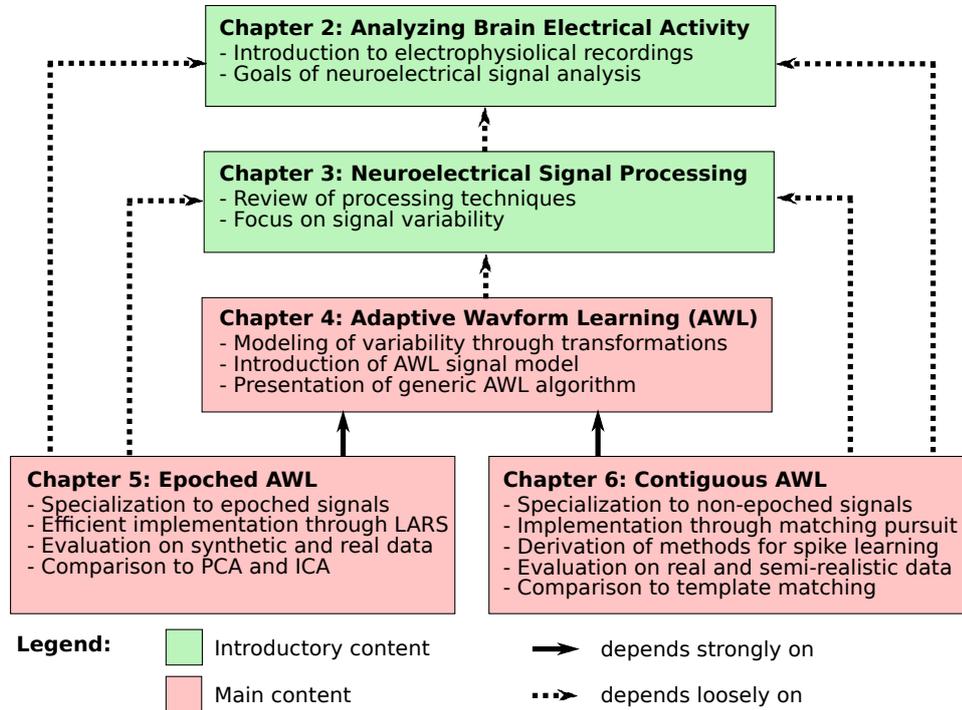


Figure 1.1: Illustration of the structure of the document and the dependencies between chapters. The introductory material (green) provides the physiological and methodological basics to neuroelectrical signal processing. The main contribution of this thesis, the method AWL, is introduced in Chapter 4 and specialized to an epoched (i.e., segmented) and a non-epoched signal setting in Chapter 5 and Chapter 6, respectively.

- Interesting findings in the coupling of neuroelectrical spiking activity and rhythms in the hemodynamics (Section 6.5). These findings resulted from the spike representations learned with MC-Spike and AD-Spike.
- Availability of the code for the different methods and the conducted experiments online for reproducibility: <https://github.com/hitziger/AWL>.

1.2 Outline

The content of this thesis is structured into an introductory part (chapters 2 and 3), the main part (chapters 4 - 6), and the conclusion (Chapter 7). The content of each of these chapters is briefly summarized in the following paragraphs. In order to provide an orientation to the reader, we illustrate the structure of this document in the dependency chart in Figure 1.1.

Chapter 2 gives an introduction to the analysis of brain electrical activity. We start by explaining fundamental concepts of the generation of electromagnetic signals by neuronal assemblies, which are measurable at different levels of the brain. Several modalities for recording the brain's electromagnetism, metabolism, and hemodynamics are presented. We then provide an overview over different characteristics of neuroelectrical recordings and define challenges and objectives for processing these signals. The chapter concludes by presenting an exemplary multi-modal study, conducted in anesthetized rats and aimed at exploring the relationships between neuroelectrical and hemodynamic changes. This dataset is later analyzed in detail (chapters 5 and 6) with the novel method AWL.

Chapter 3 provides a review of neuroelectrical signal processing techniques, with a main focus on signal variability. The chapter starts with a common averaging technique for event-related potentials introduced in Dawson (1954). It then shows different extensions to this technique that account for different types of temporal variability, such as latency compensation by Woody's method (Woody, 1967). A different approach uses linear models with different signal components and leads to the techniques PCA and ICA. Sparse representation techniques, often in time-frequency domain, form another group of popular tools for neuroelectrical signal processing. Finally, some recent integrative approaches are presented that allow to account for different types of variability in multi-variate data models.

Chapter 4 is the first of the three core chapters of this thesis presenting the novel method adaptive waveform learning (AWL). The underlying model explicitly includes general forms of signal variability which are described through mathematical operations on the signal components. A generic algorithm is proposed to compute the shapes of the waveforms and the variable parameters (i.e., amplitudes, latencies, and general morphological deformations) used in the model. This algorithm is based on sparsity assumptions about waveform occurrences and integrates sparse coding techniques in an alternate minimization framework. This means that waveform shapes and variability parameters are iteratively updated in subsequent steps. The chapter is kept in a very general setting, such that AWL can be seen as a metamodel for many of the techniques presented in Chapter 3. This generality allows to derive different concrete algorithms for specific applications, such as the ones presented in the following chapters 5 and 6.

Chapter 5 addresses the processing of a set of epoched (i.e., segmented) signals, containing event-related potentials (ERPs) or spontaneously repeating activity (SRA). For this purpose, the generic AWL algorithm from the preceding chapter is specialized to compute signal components occurring at variable latencies across epochs. The resulting E-AWL method is implemented by adapting the iterative AWL framework: starting with a set of waveforms initialized with white Gaussian noise, these waveforms and their amplitudes and latencies are efficiently updated in each algorithm step

through block coordinate descent and least angle regression (LARS), respectively. E-AWL is first evaluated on synthetic data for different amplitude and latency variability as well as different noise levels. We then illustrate the ability of E-AWL to produce meaningful representations of epoched epileptiform discharges (spikes) from the real data presented in Chapter 2. The results of both experiments are compared to those of principal component analysis (PCA) and independent component analysis (ICA).

Chapter 6 describes an alternative approach to E-AWL by processing contiguous signals, that is, without epoching. Again, the corresponding algorithm, C-AWL, can be derived from general AWL by making the appropriate specializations. This problem is more difficult compared to E-AWL in the sense that the time instants of the waveform occurrences are now completely unknown and have to be detected at arbitrary latencies throughout the entire signal. This is tackled through an efficient implementation based on matching pursuit. In addition, the initialization of waveforms is extremely important in order to avoid detecting irrelevant signal components. The concrete implementation is demonstrated for the example of two spike models: MC-Spike, which assumes spike shapes from different classes, and AD-Spike, consisting of single spike template with variable duration. These algorithms are applied to the epileptiform spikes processed in Chapter 5, however, now without prior epoching. This results in a more complete representation of the dataset, as overlapping spikes which had been discarded in the epoched setting, are now also taken into account. AD-Spike and MC-Spike are then compared to template matching in terms of detection accuracies in noisy data. Finally, the learned spike representations are used to reveal interesting insights into coupling with hemodynamic data, which had been recorded simultaneously with the electrical discharges.

1.3 Notation

Throughout this thesis, we will use bold lower case letters to denote vectors (e.g., $\mathbf{x}, \mathbf{y}, \mathbf{d}$), and bold upper case letters to denote matrices (e.g., \mathbf{A}, \mathbf{D}). If not specified otherwise, vectors are always understood as column vectors. Non-bold lower case letters denote scalars, e.g., entries in vectors $\mathbf{x} = (x_1, \dots, x_N)$ or matrices $\mathbf{A} = (a_{ij})$. An exception to this convention is made in Chapter 4 where the bold lower case letters $\mathbf{x}, \mathbf{y}, \mathbf{d}$ denote temporal signals defined on the real numbers \mathbb{R} .

Chapter 2

Analyzing Brain Electrical Activity

This chapter provides an introduction to electrophysiological brain activity and related recording modalities. We discuss the characteristics of neuroelectrical signals and define goals for their analysis. The chapter is concluded by an exemplary study investigating the coupling of hemodynamic and electrical brain activity.

Contents

2.1	Introduction	16
2.2	Electrophysiology of the brain	16
2.3	Recording electromagnetic activity	18
2.3.1	Electroencephalography (EEG)	18
2.3.2	Intracranial recordings	19
2.3.3	Magnetoencephalography (MEG)	21
2.3.4	Complementary measurements of EEG and MEG	21
2.4	Recording metabolism and hemodynamics	22
2.5	Neuroelectrical signals	23
2.5.1	Event-related potentials (ERPs)	24
2.5.2	Spontaneously repeating activity (SRA)	25
2.6	Signal analysis: goals and challenges	27
2.6.1	Assessing variability	27
2.6.2	Compact and interpretable representation	28
2.6.3	Flexible algorithm	29
2.7	Example dataset: multi-modal recordings	29

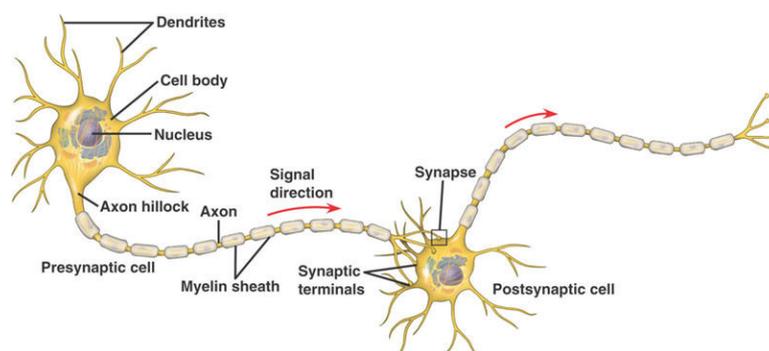


Figure 2.1: Electrochemical communication between two neurons. The left (presynaptic) neuron generates an electric *action potential* (AP) that travels along its axon. Through the synapse, a chemical interneural connection, the signal continues as *postsynaptic potential* (PSP) in the dendrites of the right neuron. Source: http://www.kurzweilai.net/images/neuron_structure.jpg.

2.1 Introduction

Understanding the mechanisms underlying the functioning of the human brain remains a great challenge in modern neuroscience. Sophisticated non-invasive modalities capable of recording electromagnetic, metabolic, and hemodynamic changes related to neuronal activity have enabled large-scale studies on living human subjects. Despite the technological advances, non-invasiveness produces significant limitations regarding signal quality, temporal and spatial resolution, as well as information on single cell activity. Invasive techniques mostly used in animal experiments provide important complementary information. There has been growing interest in simultaneous recordings with different modalities and studying how their underlying phenomena are related.

In this chapter, we focus on the electrophysiology of the brain and commonly used recording modalities. We also briefly describe modalities exploring the brain's hemodynamics and metabolism and give examples for simultaneous multi-modal recordings. We then discuss current challenges concerning the signal analysis and outline some general goals for successful neuroelectric signal processing. These will serve as a guideline for the comparison of existing techniques and the design of new methods in the following chapters. We conclude this chapter by discussing an exemplary multi-modal study in an animal model of epilepsy.

2.2 Electrophysiology of the brain

The human brain is estimated to contain around 100 billion neurons (Azevedo et al., 2009) which are permanently exchanging information along interneural

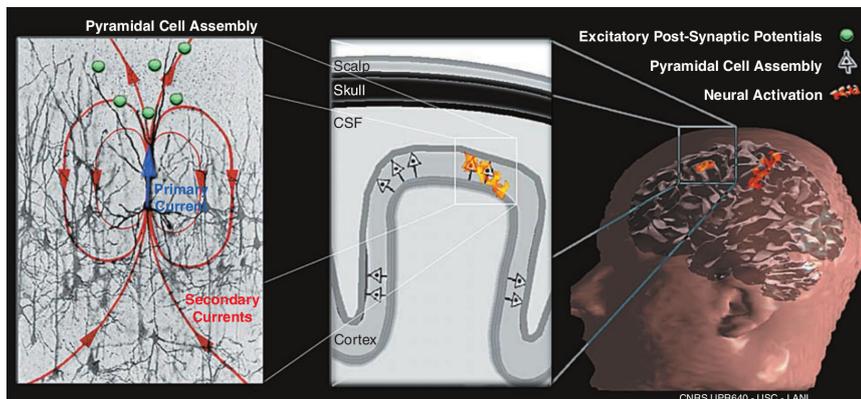


Figure 2.2: Large assemblies of cortical pyramidal neurons generate electromagnetic fields of sufficient magnitudes to be measured on the scalp level. **Left:** Postsynaptic potentials (PSPs) travel along the apical dendrites of pyramidal cells in the cortex, directed perpendicular to the cortical surface. They generate primary (intracellular) and secondary (extracellular) currents. **Center:** Thanks to the parallel arrangement of the apical dendrites, the electromagnetic fields produced by a synchronously active neural assembly sum up. The accumulated field can be modeled as a dipole with normal orientation with respect to the cortical surface. **Right:** If the assembly is sufficiently large, the resulting electromagnetic cortical source can be detected at the scalp level. Source: [Baillet et al. \(2001\)](#).

connections. This exchange is driven by electrochemical processes, through which neurons excite or inhibit other neurons to which they are connected through synapses, as depicted in Figure 2.1. When a neuron receives a large amount of excitatory signals in a short time interval, sufficient to sum above a certain threshold, the neuron *fires*: voltage-gated ion channels open, causing a transient depolarization in the neuron's membrane. This so called *action potential* (AP) travels along the neuron's axon and eventually continues as a *postsynaptic potential* (PSP) in the dendrites of connected neurons. The respective potential differences cause primary currents to flow inside the cell. The electric circuit is closed by secondary currents through all parts of the surrounding volume conductor, including the entire brain, skull, and scalp. In addition, both primary and secondary currents cause a change in the magnetic field.

In theory, the generated potentials have an impact on the electromagnetism at all three levels: intracellular, extracellular inside the head, and on the surface of the skull (i.e., even outside the head in case of the magnetic field). However, the electromagnetic field generated by a single neuron is small and decays with the square of the distance to the field's origin. Hence, it can only be empirically observed inside or in the immediate vicinity of the cell. In order to be measurable at the scalp level, the potentials of a synchronously firing neuron assembly need to add up.

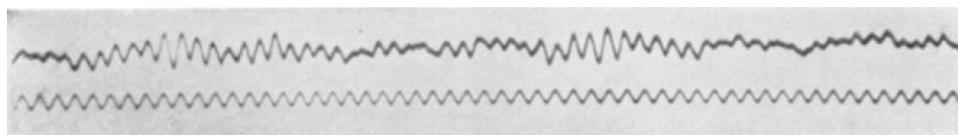


Figure 2.3: First human EEG recorded in 1924 and published in [Berger \(1929\)](#). The actual recorded signal is shown on top, while the second is a sinusoidal reference signal of 10 Hz, demonstrating the oscillatory measurement to represent alpha rhythms (cf. [Section 2.5](#)).

The conditions for such an electromagnetic field superposition are probably the most favorable for large assemblies of pyramidal cells in layers III and V of the cerebral cortex. These neurons have long apical dendrites perpendicular to the cortical surface. Hence, when a large number of PSPs travel synchronously along the dendrites, the generated parallel electromagnetic fields add up. This process is shown in [Figure 2.2](#). On the contrary, APs are believed to hardly contribute to the electromagnetic field at the scalp level. Although their relative membrane potential differences are larger (around 100 mV) than for PSPs (around 10 mV), they only last for several milliseconds (tens of milliseconds for PSPs). This makes the synchronicity constraint more stringent in order for APs to sum up.

More detailed information about the brain's electrophysiology can be found in [Speckmann and Elger \(2005\)](#).

2.3 Recording electromagnetic activity

2.3.1 Electroencephalography (EEG)

Ever since German physician Hans Berger obtained the first recordings of brain electrical activity in a human in 1924 (see [Figure 2.3](#)), electroencephalography (EEG) has been an extremely successful tool for studying the functioning of the brain. While Berger originally only used a single pair of electrodes to record potential differences between two locations on the scalp, a modern EEG protocol can measure up to 512 scalp potentials with sampling frequency of 10 000 Hz simultaneously. However, most setups only use less than 40 electrodes (see [Figure 2.4](#)). The equipment required for recording a modern EEG is simple and inexpensive, especially when compared to other brain imaging devices, and essentially consists of a set of electrodes, a signal amplifier, and a computer.

EEG almost instantaneously captures the electrical neural activity. This makes it the method of choice to study epilepsy, which is characterized by seizures involving abnormal electrical activity of neurons in the affected brain regions. A recording of such a seizure is shown in [Figure 2.5](#).

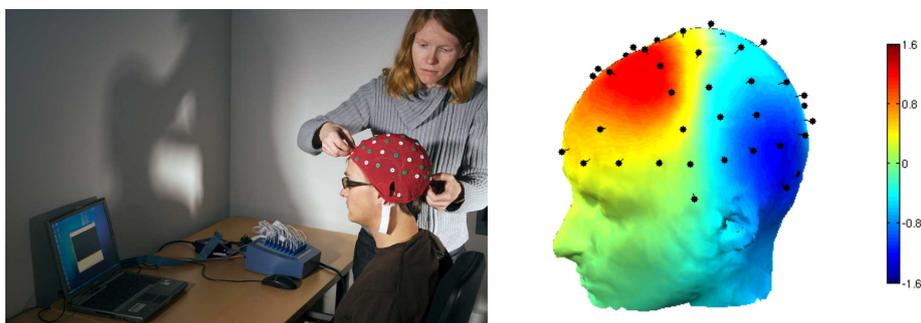


Figure 2.4: **Left:** EEG recording session with multi-electrode cap at Inria Sophia Antipolis. **Right:** Visualization of potential differences on the scalp, source: [Vallaghé \(2008\)](#).

The high temporal resolution around a millisecond provides a detailed temporal view of the brain activity. This furthermore allows direct real-time communication from the brain to an external device, known as brain-computer interface (BCI). These interfaces can help to study cognitive or sensory-motor functions and assist humans in the execution of these functions. A currently widely studied example is the P300 spelling device (cf. Section 2.5.1), enabling text production solely through the EEG. Other applications of EEG include the diagnosis of sleep disorders, encephalopathies, brain death, and the assessment of degrees of consciousness in coma.

In the past decades, much effort has been made to determine more precisely the spatial locations of the active neural sources by exploiting the different electrode positions. However, despite improvements, the spatial resolution of EEG remains limited (around 20 mm), staying behind that of more recent brain imaging techniques, notably functional magnetic resonance imaging (fMRI). Another shortcoming of EEG is the generally high noise level in the recordings caused by neural background activity. This often makes it necessary to acquire large amounts of data in order to extract useful information.

Nevertheless, EEG has hardly lost in popularity. Currently, much effort is made to maximize the usability of EEG devices through electrodes (or electrode caps) that do not require any conductive gel. These improvements could enable even more wide-spread use of EEG and the possibility of long-term monitoring outside clinics or laboratories.

2.3.2 Intracranial recordings

EEG usually refers to non-invasive recordings on the scalp. While these measurements outside of the head give a broad view of brain activity, their spatial resolution is limited. Different research studies and clinical applications, however, require to record activity directly at precise locations in the

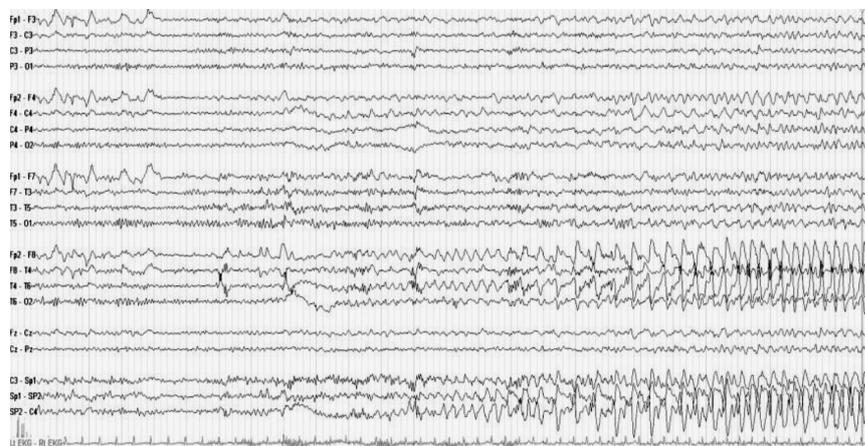


Figure 2.5: EEG recordings of a patient during epileptic seizure. The time series show potential differences between the two scalp electrodes marked on the left (e.g., Fp1-F3), named according to standard placement conventions. Source: <https://teddybrain.wordpress.com>.

brain by placing electrodes inside the skull.

In humans, the most common application of intracranial recordings is the exact localization of the cortical regions that generate epileptic seizures, often necessary prior to the treatment of epilepsy through surgery. For this purpose, single or multiple electrodes are either implanted directly onto the exposed surface of the cortex or even into deeper brain tissues. These techniques are known as electrocorticography (ECoG) and stereoelectroencephalography (sEEG), respectively.

In animals, intracranial recordings date back to as early as 1870 and are frequently used for research purposes. Depending on the size and impedance as well as the exact placement of the electrodes, neural activity can be measured at micro- and mesoscopic levels. When placed close (within about $50 \mu\text{m}$) to a neuron, a microelectrode can measure directly its unitary activity (Harris et al., 2000). In contrast, larger electrodes that are located in a greater distance from the active sources will capture the combined activity of populations of neurons. The high frequency components above 300 Hz contain multiple-unit spike activity (MUA), while the signal filtered below 300 Hz is referred to as local field potential (LFP) (Logothetis, 2003).

In order to investigate higher cognitive functions, animals with brain structures closest to humans such as non-human primates are preferred. However, for the study of the basic neural mechanisms – for example for a better understanding of the origins and progress of epileptic seizures – smaller animals such as rats are frequently used.

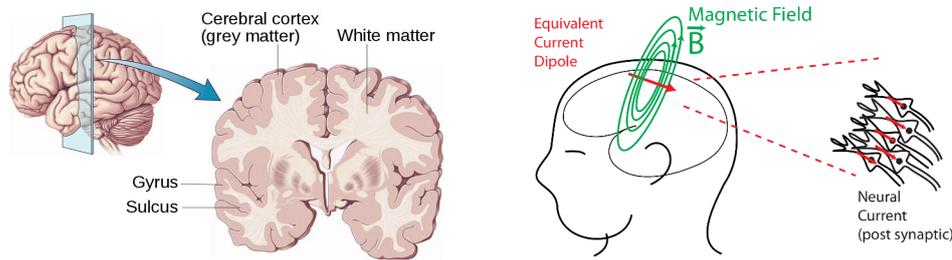


Figure 2.6: **Left:** A coronal slice through a human brain illustrating the cerebral cortex, that is, the brain's outer layer. Its convoluted arrangement leads to ridges (gyri) and fissures (sulci). Adapted from <http://www.slideshare.net/kbteh/human-brain>. **Right:** The magnetic field resulting from an electric current dipole lies in a plane that is perpendicular to the current flow. Source: Gramfort (2009).

2.3.3 Magnetoencephalography (MEG)

Magnetoencephalography (MEG) is the counterpart to EEG, exploiting the magnetic field generated by the active neurons. Introduced in 1968 by physicist David Cohen (Cohen, 1968), MEG was able to record signals of quality similar to EEG only after the invention of the first SQUID (superconducting quantum interference device) detectors (Zimmerman et al., 1970). In order to adequately measure the extremely small brain's magnetic field in the order of 10-100 femtotesla (fT), modern MEG uses sophisticated equipment and requires a magnetically shielded room.

An advantage of MEG compared to EEG is the higher spatial resolution, due to typically larger numbers of detectors and the higher insensitivity of the magnetic field to skull and scalp. However, the complicated recording conditions and high equipment costs limit its use.

2.3.4 Complementary measurements of EEG and MEG

While electric potentials and the magnetic field measured outside of the head both originate from the electrical neural activity, there are important qualitative differences between EEG and MEG recordings. As explained in Section 2.2, only large assemblies of simultaneously active pyramidal neurons are believed to significantly contribute to the electromagnetic field at scalp level. Since most electric current flows normally to the cortical surface, the direction of this field depends on the position of the source in the cortex, due to its convoluted arrangement in the brain (see Figure 2.6). In cortical regions parallel to the skull (in particular the gyri), most current flows to the nearest parts of the skull and can thus be detected by EEG. The magnetic field lines, however, are located in a plane tangential to the skull, and the electric source will be hardly visible to MEG. Sources in a cortical sulcus, in turn, can be better observed through MEG than through EEG. Hence,

EEG and MEG are actually complementary in the sense that they are each optimal for differently located sources.

2.4 Recording metabolism and hemodynamics

The recording modalities presented in the previous section capture the neural activity very directly, since the electromagnetic field propagates almost instantaneously. The greatest drawback is the limited spatial resolution of the non-invasive EEG and MEG.

Alternatively, neural activity can be measured indirectly through the brain's metabolism and hemodynamics. Active brain regions require an increased supply of nutrients such as oxygen and glucose, which in turn results in a change of the cerebral blood flow (CBF). The metabolic and hemodynamic changes, however, typically occur with a time lag of 1 second or more after the neural events. Many different techniques are used to measure the metabolism or CBF, leading to very complementary observations compared to the electromagnetic recordings.

Functional magnetic resonance imaging (fMRI) uses the blood-oxygen-level dependent (BOLD) contrast to obtain detailed spatial maps of neural activation, with a resolution as fine as a few millimeters. Since the discovery of the BOLD-contrast by Seiji Ogawa in the 1990s (Ogawa et al., 1990), the non-invasive fMRI has become the most widely-used tool for brain mapping research. The CBF and blood oxygenation can also be measured through near-infrared spectroscopy (NIRS) (Gibson et al., 2005) which exploits the different optical absorption spectra of oxygenated and deoxygenated hemoglobin. In contrast to fMRI, NIRS is more portable and can be used in infants. However, its use is limited to scan cortical tissue, whereas fMRI measures activity throughout the entire brain. Laser Doppler flowmetry (LDF), like NIRS, also involves the emission and detection of monochromatic light to and from neural tissue. The frequency of the red or near-infrared light is shifted according to the Doppler principle when scattered back from moving blood cells. This allows to monitor changes in the CBF. As a drawback, LDF cannot be calibrated in absolute units, as noted in Vongsavan and Matthews (1993). All three methods presented above are completely non-invasive.

In contrast, nuclear imaging, including positron emission tomography (PET) (Bailey et al., 2005) and single photon emission computed tomography (SPECT), require the injection of a radioactive tracer into the bloodstream. The tracer can be located as it produces measurable gamma rays either directly (SPECT) or indirectly through emitted positrons (PET). Depending on the specific tracer, the recordings can measure either oxygen or glucose levels in the blood, or directly the CBF.

The methods presented above are only a subset of brain recording and

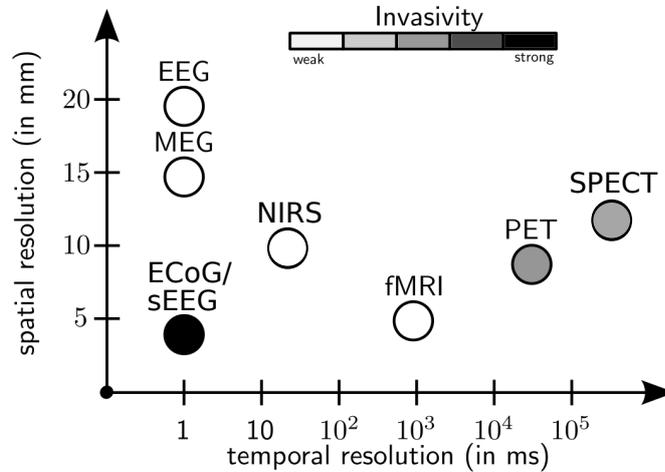


Figure 2.7: Comparison of spatial and temporal resolution of different brain imaging techniques. Adapted from [Olivi \(2011\)](#).

imaging techniques.

A comparison of the spatial and temporal resolution and the invasiveness of the presented electrophysiological and metabolic/hemodynamic recording modalities is shown in Figure 2.7. We note that the presented selection is not exhaustive as there exist many other techniques for observing activity in the brain. Because of the complementary information obtained through hemodynamic and metabolic recordings compared to electromagnetic recordings, it has been of interest to combine two or more of these modalities for simultaneous recordings. Such approaches can help to better understand the relationship between neural activity and blood flow changes, known as neurovascular coupling ([Vanzetta et al., 2010](#)).

2.5 Neuroelectrical signals

Neuroelectrical recordings can produce signals with very different characteristics. The first measured EEGs by Hans Berger showed strong oscillatory activity around 10 Hz. Oscillations of different frequencies have since been observed and divided into distinct frequency bands, associated with different types of mental activity: delta rhythm (< 4 Hz, e.g., deep sleep), theta rhythm (4 – 7 Hz, e.g., drowsiness), alpha rhythm (8 – 13 Hz, e.g., relaxed, eyes closed), beta rhythm (14 – 30 Hz, e.g., active thinking, certain sleep stages), and gamma rhythm (> 30 Hz, e.g., active information processing); see, for instance, [Sörnmo and Laguna \(2005, chapter 2\)](#). Other than rhythmic activity, transient waveforms such as increases or decreases in the electric potential are observed. Different types of rhythmic and transient activity are also observed during different sleep stages (see Figure 2.9).

The measured electrical activity is often divided into two categories: Event-related potentials (ERPs) are associated with the occurrence of a specific (and usually observable) event, such as muscle movements or the processing of a stimulus. All other activity can be subsumed as spontaneous activity.

The analysis of neuroelectrical signals is often very challenging due to low signal-to-noise ratios. This is especially the case for non-invasive recordings, where signals are buried in noise or neural background activity. In order to recover the signals of interest, typically, a large set of data samples has to be acquired. We will here introduce the main issues related to the processing of ERPs and spontaneous activity.

2.5.1 Event-related potentials (ERPs)

Event-related potentials (ERPs) are often studied in a setting where the underlying event (i.e., often a stimulus) can be controlled. Recordings are typically performed continuously while keeping track of the time instants (latencies) of the presented stimuli. This allows to automatically segment the contiguous dataset into short time windows around or after each stimulus, a process referred to as epoching. Taking the average over these windows, or epochs, results in an enhanced ERP waveform (Dawson, 1954).

However, averaging relies on the assumptions of a constant ERP response across all epochs and additive white noise. Both assumptions are highly questionable, and alternatives to averaging, which take into account variable response latencies and temporal scaling, are presented in the next chapter. Moreover, the practice of epoching the recorded signal may itself be challenged. In fact, as time intervals between presented stimuli are often short, epoching can introduce errors due to overlapping responses.

A widely studied ERP is the P300, describing a positive potential occurring approximately 250-500 ms after the target stimulus (Polich, 2007). It is typically observed in the so called *oddball* task, in response to a rare *target* stimulus occurring among frequent standard stimuli (Donchin et al., 1978; Pritchard, 1981). A main challenge of its assessment lies in the strong variability of its shape and latency (w.r.t. the stimulus) across subjects and task conditions. Another frequently studied evoked potential is the negative N100 which peaks about 80-120 ms after the stimulus onset. It is often followed by the positive P200. Figure 2.8 schematically illustrates an oddball task with a P300 response to the target stimulus and a N100-P200-N200 complex after non-targets.

The *Bereitschaftspotential* (German for readiness potential), or BP is a slow negative potential observed in the motor cortex, preceding human volutary movement (Deecke et al., 1976; Kornhuber and Deecke, 1965). The *Bereitschaftspotential* is about one or two magnitudes smaller than the alpha-rhythm measured in the EEG and only becomes apparent after

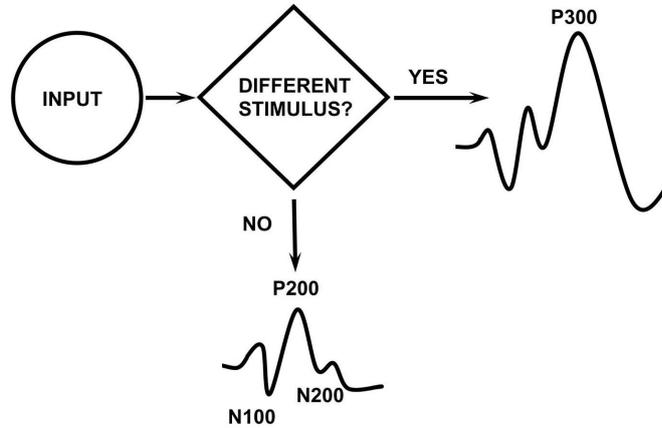


Figure 2.8: Illustration of the P300 ERP from (Polich, 2003). As long as the incoming stimuli remain unchanged, the measured responses are sensory evoked potentials with N100, P200, and N200 components. When the subject perceives a rare *target* stimulus as in the *oddball* task, the positive P300 potential can be observed in addition to the sensory evoked potentials.

averaging. Although the BP occurs before the muscle movement, it is still time-locked to this event and thus considered an ERP.

In contrast to these transient responses, the *steady-state* evoked potential is an ongoing response to a quickly repeating stimulus. The steady-state visual-evoked potential (SSVEP), for instance, is induced by a flickering at a constant frequency around 6 to 100 Hz and produces an increase in amplitude at the stimulated frequency.

When dealing with event-related oscillatory activity, it is important to distinguish between *evoked* and *induced* activity. Evoked activity refers to periodic activity that is phase-locked to the stimulus, while induced activity is not. Stimulus-locked averaging is suited to enhance the evoked activity. The varying phase of induced activity, however, leads to a decrease of amplitude in the average. In Tallon-Baudry and Bertrand (1999), the authors suggest averaging over time-frequency power maps; this technique will be discussed in the following chapter.

Introductions and detailed information about ERPs can be found, for instance, in Kutas and Dale (1997), Picton et al. (2000), or Luck (2012, 2014).

2.5.2 Spontaneously repeating activity (SRA)

All recorded activity that is neither noise introduced through the recording process nor directly attributed to a specific event is commonly referred to as spontaneous activity. While in the context of processing ERPs the

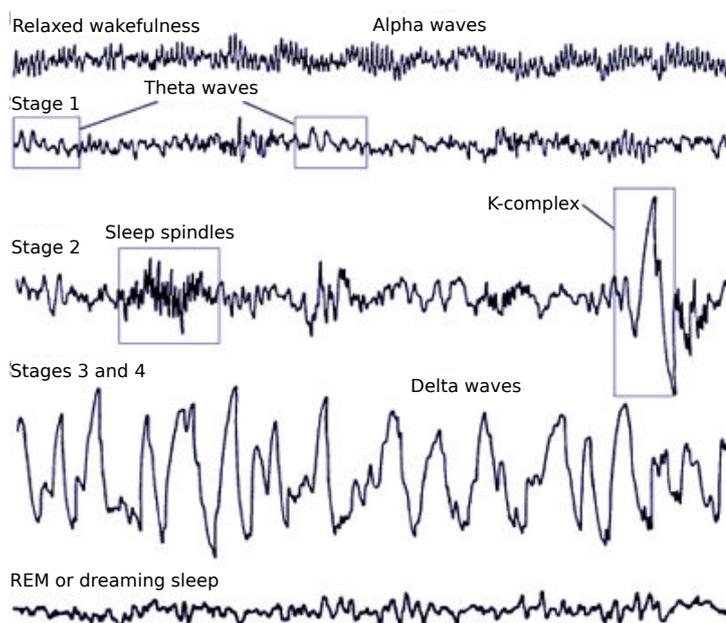


Figure 2.9: Brain activity during different sleep stages. The detection of transient sleep spindles and K-complexes, for example, can help in diagnosing stage 2 sleep.

spontaneous background activity is often considered as noise, it can contain interesting transients and rhythmic activity that are the subject of study in other applications. Polysomnography, a type of sleep study involving EEG recordings, can make use of ongoing rhythmic brain activity or transients such as sleep spindles or K-complexes (De Gennaro and Ferrara, 2003) to detect different sleep stages or diagnose sleep disorders (see Figure 2.9). Another application is the detection of inter-ictal (i.e., between seizures) spikes (see Figure 2.10) as a possible predictor of an epileptic seizure.

If recordings are reasonably short, the events of interest may be identified by a human expert. Epoching, as described in the previous section, then may be performed in order to further analyze the extracted waveforms. However, recordings often last for several hours or even days, making monitoring by a human expert infeasible, especially when real-time detection is required. In this case, automatic detection and classification techniques are needed that can process the entire contiguous recording.

For the purpose of this thesis, we shall introduce the term spontaneously repeating activity (SRA) as follows: by SRA we mean all activity of interest which repeats multiple times throughout the recorded signal, possibly with changing waveform amplitudes and shapes. Examples of SRA are, for instance, the inter-ictal spiking activity or the occurrences of sleep spindles mentioned above. SRA may be linked to some underlying phenomenon, but the time instants are unknown. In contrast, when referring to ERPs, we

will assume that the time instants of the triggering events and hence the approximate temporal locations of the waveforms are given.

2.6 Signal analysis: goals and challenges

As we have just seen, the range of brain imaging techniques and neurological research fields is broad. Consequently, the measured brain signals have very different characteristics and the purpose of their analysis varies across applications. Nevertheless, in this section, we discuss general objectives in the processing of neuroelectrical signals and outline the qualities of an "ideal" method. This will serve us as a reference for comparing existing methods in Chapter 3. It will also be the starting point for designing a new technique that addresses the challenges found across different applications (see Chapter 4).

A main issue addressed in this thesis is the occurrence of waveform variability encountered in many neuroelectrical signals. This variability needs to be adequately described so that it can be compensated for. Furthermore, it can serve as a valuable source of information. It is desired that the obtained data representations be compact and allow for easy and physiologically relevant interpretations. Finally, a good processing technique should be efficient, adaptive to input and application, and automatic. These aspects are discussed in detail in the following subsections.

2.6.1 Assessing variability

Large variability in brain electrical signals is often observed when comparing the responses of different subjects within the same experimental protocol. However, it may also occur within a single subject and recording session. This trial-to-trial variability has been acknowledged in many studies (Jung et al., 2001; Kisley and Gerstein, 1999). It concerns variable amplitudes and response latencies as well as more general shape deformations. Some of this variability can be explained, for instance, as results habituation or fatigue throughout a recording session or calibration errors in the recording device. However, the largest portion remains non-deterministic.

We will refer to variability as any change of a specific waveform across different occurrences. This may include a broad range of phenomena, such as changing latencies (e.g., w.r.t. a stimulus), rate of occurrence, amplitude, duration, frequency, phase, or even more general morphological deformations. For example, in the case of induced gamma activity (Tallon-Baudry and Bertrand, 1999), the oscillatory response is subject to phase shifts (cf. Section 2.5.1).

Regardless the origin of the variability, its description and quantification can provide an important source of information. A proper compensation for this variability furthermore leads to a better characterization of the "true"

or stereotypical response waveform (see Figure 3.1 in the next chapter for an illustration of averaging without jitter compensation). It can also help to separate different phenomena if their mutual variability is not entirely correlated. We suggest to assess variability through the following three guidelines:

- For each type of observed waveform (waveform class), determine a template waveform representing it.
- Detect all occurrences of each waveform class.
- Characterize variability in every waveform class and quantify the deviation of each occurrence from the template.

In this thesis, we will focus mainly on variability in two contexts: event-related potentials (ERPs, cf. Section 2.5.1) and spontaneously repeating activity (SRA, cf. Section 2.5.2). Both cases require the processing of a waveform that is subject to change. However, in the case of SRA, the waveform occurrences are in general unknown, hence the necessity of the detection step. For ERPs, the time instants of the stimuli are given; however, associated waveforms may occur at variable latencies.

The three steps above are actually interdependent: An adequate description of each template waveform should be a kind of variability-compensated average of the members of its class. Characterization of the variability, in turn, requires all instances of a class and is expressed relative to its representing template. This interdependence is essentially very similar to that encountered in vector quantization or clustering: the locations of the centroid points depend on the members of the cluster, while the class label of each vector depends on the nearest centroid point. The well-known K -means algorithm (MacQueen et al., 1967) solves this problem – very intuitively – through alternating updates between centroid locations and vector labeling. Another prominent technique that uses alternating updates is Woody’s method (Woody, 1967) in order to compensate for latency variability while calculating the average. This method will be presented in the following chapter.

2.6.2 Compact and interpretable representation

Naturally, a good data representation should be compact and interpretable. For the particular representation suggested above, the number of waveform classes should be kept as low as possible. In addition, the variability is best described in terms that can easily be quantified or parametrized. The representation should enable to reconstruct an approximation of the data that contains maximal signal and minimal noise variance.

Ideally, each waveform class represents some physiological phenomenon. In the case of multi-channel EEG (or MEG), waveforms could be associated

with active neural sources. In fact, much effort has been made to clearly separate the active sources functionally or spatiotemporally.

Compact and physiologically relevant data representations not only allow for easy interpretation but can also be used for further processing. An example is the relationship between multi-modal data: Often, these datasets are of too different nature to be compared directly. Meaningful features can instead be used to calculate dependencies or correlations.

A frequently used technique is sparse coding over large dictionaries of elementary signals, such as wavelets (cf. Section 3.6). However, the individual dictionary *atoms* often do not well represent the signals of interest. When using symmetric Gabor atoms, for instance, a large number of them will be necessary in order to approximate an asymmetric waveform.

2.6.3 Flexible algorithm

In order to be applicable to a wide range of signals and tasks, high flexibility and adaptivity of the algorithm is useful. It should be able to learn signal representations *blindly*, that is, with few prior assumptions. However, if reliable information is available (e.g., on waveform shapes or instances, variability, etc.), it should be exploited by the algorithm. Maximal use of prior information is crucial for low signal-to-noise ratios to avoid noise fitting.

Ideally, data should be processed as "raw" as possible. While certain pre-processing steps such as frequency filtering are often indispensable, their use comes with the risk of losing relevant information. For example, epoching of events to simplify averaging may introduce distortions if waveforms overlap. Therefore, it appears beneficial to process contiguous recordings as a whole. Multi-variate data (e.g., multiple recording channels, experimental trials, different modalities) should also be processed in one procedure, if possible. This way, possible coupling between the different datasets can be maximally exploited.

Finally, an optimal method should allow for efficient implementations, enabling fast computation and real-time detection of waveforms.

2.7 Example dataset: multi-modal recordings

To conclude this chapter, we illustrate different concepts addressed in this chapter on an exemplary study, which we will analyze in detail through different approaches in the following chapters.

The context of the study was the ANR¹ project Multimodel, which aimed to reveal hidden parameters through multi-modal data fusion with biophysical models (Blanchard et al., 2011; Hitziger et al., 2013; SAILLET et al., 2015;

¹L'Agence Nationale de la Recherche

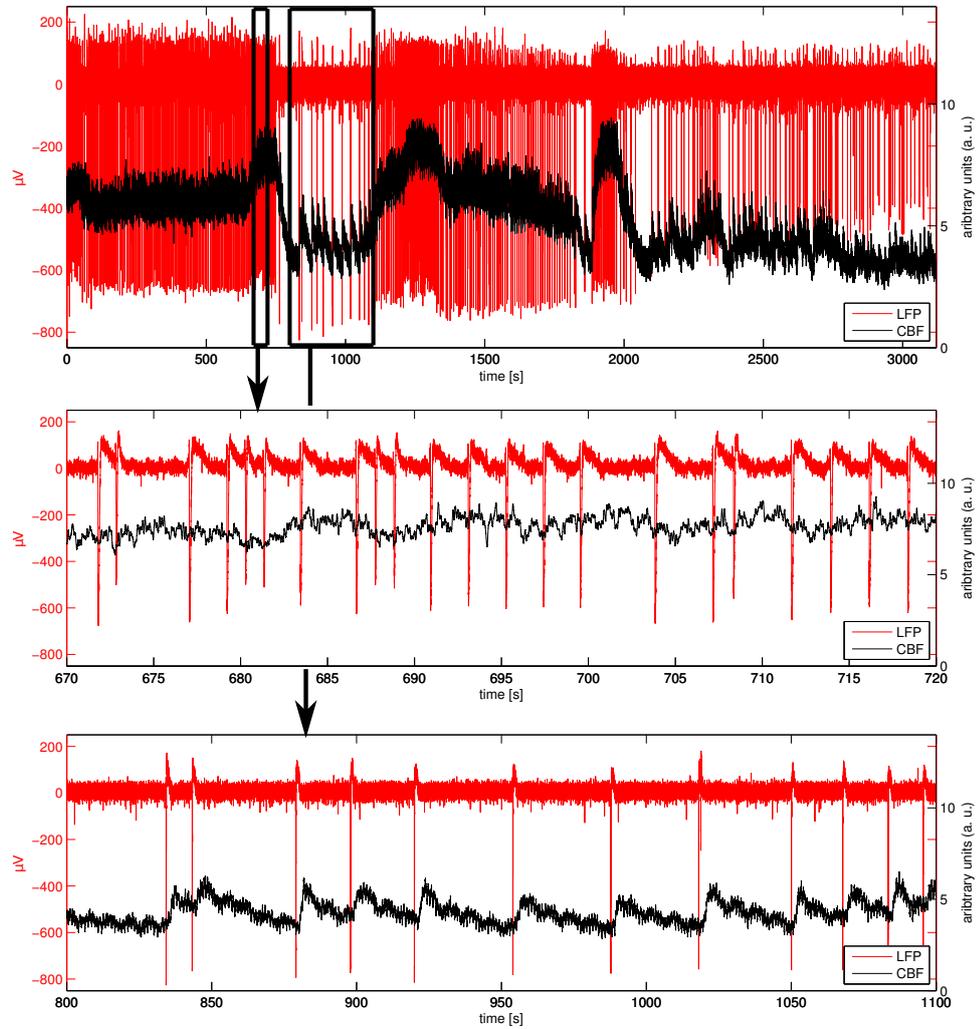


Figure 2.10: Local field potential (LFP, red) and cerebral blood flow (CBF, black) were recorded simultaneously in a rat's cortex. **Top row:** The approximately one hour long recording clearly shows an increase in the CBF level during increased spiking activity (red vertical lines) in the LFP. **Middle row:** During increased LFP activity, the change in CBF does not seem to be directly related to the individual spikes. **Bottom row:** When the distance between spikes is sufficiently large, a clear CBF response becomes visible. Note the different time scales in the three windows.

Voges et al., 2012). The particular dataset presented here consists of simultaneous recordings through several modalities in the cortex of anesthetized rats. Prior to and during the recording, bicuculline was injected locally to block inhibition, which caused neural electric discharges, simulating interictal (i.e., between epileptic seizures) spiking activity. The complete recording protocol which included six rats and other recording modalities can be found in Saillet et al. (2015), currently under review. The goal was to study activity of individual cells and neural assemblies, as well as their link with the hemodynamics.

Here, we will focus on the relationship between epileptiform spikes in the local field potential (LFP, cf. Section 2.3.2), measured by an intracortical electrode, and the cerebral blood flow (CBF), obtained through laser Doppler flowmetry (cf. Section 2.4). These simultaneous recordings are illustrated in Figure 2.10. The first row contains the entire time series of approximately one hour. The LFP (red) and CBF (black) are superposed in this plot to illustrate their relationship. The long vertical lines in the LFP are epileptiform discharges (spikes). Clearly, during increased spiking activity, the baseline of the CBF rises. The first outburst of spikes is depicted in the second row, showing only a short time window of the entire recording. Here, a relationship between individual spikes and CBF response cannot be directly seen, which is due to the short intervals between spikes (1-4 seconds) and possibly a saturation of the CBF. In contrast, during low spiking activity (last row), each LFP spike is followed by a clear CBF response.

When calculating the CBF response through conventional averaging, only a small subset of well-isolated spikes can be taken into account, since the typical response time is very long, that is, typically above 30 seconds (Vanzetta et al., 2010). This would exclude the periods of high spiking activity from the analysis, which could be of particular interest when investigating the coupling of neuroelectrical and hemodynamic phenomena. Response estimation through deconvolution (Vanzetta et al., 2010) can theoretically cope with response overlaps. However, during very high spiking rates, the CBF responses do not superpose linearly, leading to low reliability of these estimates.

As we will see throughout the further analysis of this dataset, there are several interesting dynamics and couplings between the datasets that cannot be observed by visual inspection. In a order to assess these hidden relationships, it can be beneficial to first describe the dynamics across the LFP spikes. In particular, this means variability in spike shapes, duration, energy, and occurrence rates. These parameters can then be correlated with the dynamics in the CBF in order to exploit the neurovascular coupling.

The novel method AWL introduced in Chapter 4 allows to learn data representations with variable waveforms. By making the appropriate specializations in chapters 5 and 6, we will show how AWL can produce complementary spike representations by learning on epoched and contiguous

recordings, respectively. The contiguous processing in Chapter 6 allows for a more complete representation of the dataset as it takes into account the entirety of epileptiform spikes (including superpositions). This allows to reveal interesting relationships between LFP spiking activity and dynamics in the CBF (cf. Section 6.5).

Chapter 3

Neuroelectrical Signal Processing

This chapter provides an overview of widely-used techniques for processing neuroelectrical data. It also introduces concepts used for the AWL algorithm presented in Chapter 4. For this purpose, we focus especially on aspects of signal variability.

Contents

3.1	Introduction and overview	34
3.2	Ensemble averaging	35
3.3	Models with temporal variability	37
3.3.1	Woody's method	37
3.3.2	Dynamic time warping	38
3.3.3	Variable response times	40
3.4	Linear multi-component models	40
3.4.1	Principal component analysis (PCA)	41
3.4.2	Independent component analysis (ICA)	42
3.5	Combined approaches	45
3.5.1	Differentially variable component analysis (dVCA)	45
3.6	Sparse representations	46
3.6.1	Sparse coding and time-frequency representations	46
3.6.2	Dictionary learning	50
3.6.3	Translation-invariant dictionary learning	50
3.7	Conclusion and outlook	52

3.1 Introduction and overview

This chapter has two aims: (i) describe important and widely-used techniques for processing neuroelectrical data and (ii) introduce concepts used for the algorithm proposed in Chapter 4. For this purpose, we limit the selection of presented methods to those relevant for this thesis and focus especially on aspects of signal variability. For a more exhaustive overview of neuroelectrical signal processing, see for instance [Michel and Murray \(2012\)](#).

As we have discussed in the previous chapter, the processing and interpretation of electrophysiological recordings is a challenging task. A single recorded signal in response to a presented stimulus, for instance, may not allow any conclusions concerning the event-related potential (ERP), as these recordings tend to be highly non-deterministic and noisy.

Acquiring a large number of experimental trials on several recording channels improves the possibility of separating the signal of interest from neural background activity or instrumental noise. However, a good estimation of the event-related part of the recordings requires adequate modeling of both the signal of interest and the noise. The most simple signal-plus-noise (SPN) model leads to the practice of ensemble averaging ([Dawson, 1954](#)), discussed in the next section. While this technique has remained common practice until today, many variations of the SPN model have been proposed, leading to more sophisticated techniques.

One of the main concerns with the SPN assumptions is the frequently observed variability of the event-related response itself. While amplitude changes do not largely affect the calculated average, variability in latency – if not accounted for – can lead to serious distortions and smoothing through averaging. Different time delay estimation techniques have been proposed to compensate for the temporal variability, a prominent approach being Woody’s method ([Woody, 1967](#)) described in Section 3.3.1. More general temporal rescaling can be obtained by dynamic time warping (Section 3.3.2).

A different approach consists in modeling the event-related response as a multi-component signal. These components may be of purely functional purpose, such as for dimensionality reduction of the data. This is the objective of principal component analysis (PCA) (Section 3.4.1), which proves capable of compressing the majority of the data variance in only a few components. Components may also be identified directly with active neural sources. Assuming statistical independence between sources allows their separation through independent component analysis (ICA) (Section 3.4.2). As both PCA and ICA are linear methods, they cannot compensate for temporal variability across trials. For this reason, they are most frequently applied to different recording channels, making use of their strict isochronicity.

The methods mentioned above each address one particular aspect of variability: Woody’s method, for instance, accounts for latency variability

of a single waveform while PCA and ICA consider varying amplitudes of different components. In Section 3.5 we describe an approach, differentially variable component analysis (dVCA), that explicitly combines these two aspects of variability in a single framework.

Sparse representations (Section 3.6) have proven useful for many signal and image processing tasks. Especially when shown in time-frequency domain, these representations give a good view of the activity in the neuroelectrical recordings. Sparse coding techniques (Section 3.6.1) allow to extract features, which have been previously defined in a *dictionary*, from the signals. Alternatively, dictionaries can be learned simultaneously with the code in a data-driven approach, which is discussed in Section 3.6.2. Some extensions to dictionary learning take into account different invariances such as translation invariance (Section 3.6.3). These techniques produce more compact representations that are better interpretable and can cope with temporal jitter.

3.2 Ensemble averaging

Given a number of experimental trials $\{\mathbf{x}_m\}_{m=1}^M$, the basic signal-plus-noise model (SPN) assumes an underlying deterministic signal \mathbf{d} (e.g., the event-related potential or ERP) that is unchanged across trials and background noise ϵ_m for each trial, leading to

$$\mathbf{x}_m = \mathbf{d} + \epsilon_m, \quad m = 1, \dots, M. \quad (3.1)$$

Averaging over m as suggested in Dawson (1954) leads to the estimation

$$\hat{\mathbf{d}} = \frac{1}{M} \sum_{m=1}^M \mathbf{x}_m = \mathbf{d} + \frac{1}{M} \sum_{m=1}^M \epsilon_m.$$

Assuming ϵ_m to be independent across trials and the noise variance σ_m to be the same for all trials, the noise amplitude in the average $\hat{\mathbf{d}}$ is reduced by a factor of $1/\sqrt{M}$ with respect to a single trial. However, in many experiments, this noise reduction cannot be observed empirically, showing the SPN to be overly simplistic. In fact, the assumptions of a constant event-related signal and the independence of background noise ϵ_m across trials are highly questionable.

In Figure 3.1, we illustrate the effects of averaging over trials \mathbf{x}_m if \mathbf{d} is not constant. In the case of varying amplitudes only, averaging still yields a very good estimate of \mathbf{d} (row 3). However, when the latencies of \mathbf{d} vary across trials, the amplitude in the average is reduced and the waveforms are smoothed (row 4, see figure caption for details). The subject of temporal variability across waveforms has always been a major concern in neuroelectrical signal processing and will be treated in the next section.

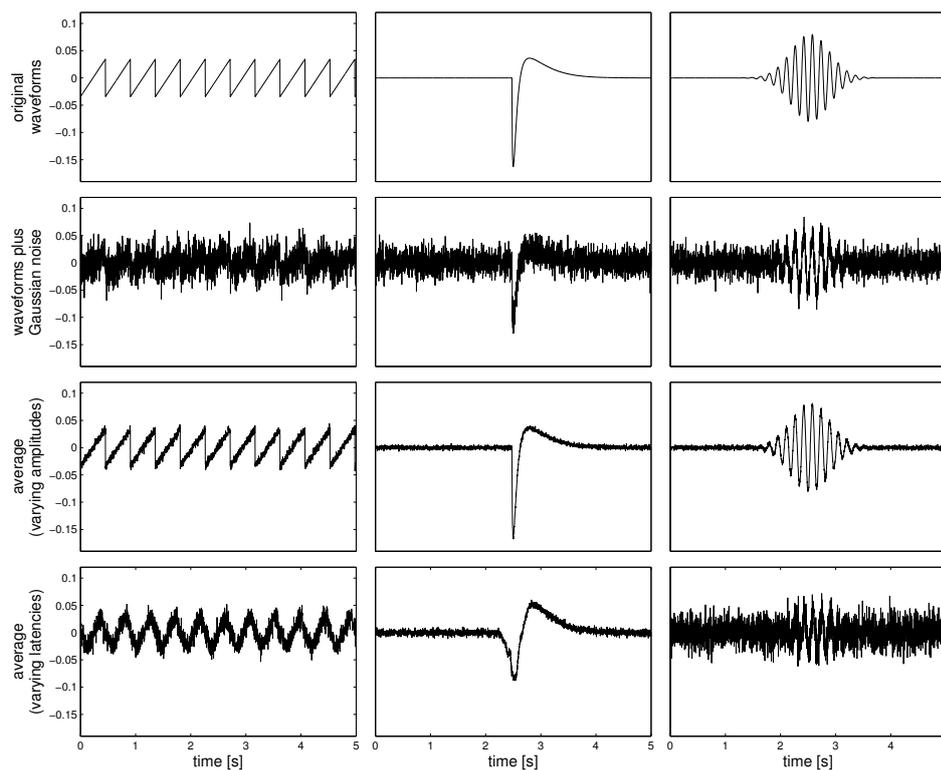


Figure 3.1: Three differently shaped, normalized waveforms are plotted without (top row) and with white Gaussian noise (second row, SNR: -2 dB). Row three shows averages over 100 noisy instances of each waveform, with amplitudes drawn from a Gaussian distribution (mean $\mu = 1$, standard deviation $\sigma = 0.5$). The last row shows the average over noisy waveforms with variable latencies ($\mu = 0$, $\sigma = 0.1$ seconds). All signals have been normalized for better comparison. Averaging without latency compensation leads to serious distortions and smoothing in the waveforms. The signal-to-noise ratio (SNR) of the third waveform even worsens after averaging.

3.3 Models with temporal variability

From the beginnings of ERP processing, it has been acknowledged that temporal variability across experimental trials may result in suboptimal averages. Adequate modeling and compensation for time-related variations in the signal shape has been of interest since the 1960s, and a great number of approaches and techniques have been proposed. Most generally, time-related variability can be described by rescaling functions $\tau_m : \mathbb{R} \rightarrow \mathbb{R}$, such that the SPN model (3.1) becomes

$$\mathbf{x}_m = \mathbf{d} \circ \tau_m + m, \quad m = 1, \dots, M,$$

where \circ denotes function composition. The simplest case considers constant time shifts or temporal jitter $\tau_m(t) = t - \delta_m$, which will be discussed in Section 3.3.1. More general scaling functions τ_m are used to represent linear or dynamic temporal stretches, in the latter case typically assuming monotonicity and sufficient smoothness of τ_m , see Section 3.3.2.

3.3.1 Woody's method

Extending the SPN model from the previous section to contain trial-dependent latencies δ_m of the signal of interest \mathbf{d} leads to

$$\mathbf{x}_m = \mathbf{d}(\cdot - \delta_m) + \epsilon_m, \quad m = 1, \dots, M. \quad (3.2)$$

The simultaneous estimation of \mathbf{d} and the latencies δ_m can be tackled in an iterative process, as proposed in [Woody \(1967\)](#). An initial signal estimation $\hat{\mathbf{d}}$ is used as a template. Based on the cross-correlations

$$\mathbf{c}_m(\delta) \equiv (\mathbf{d} \star \mathbf{x}_m)(\delta) \equiv \langle \mathbf{d}, \mathbf{x}_m(\cdot + \delta) \rangle$$

with $\langle \cdot, \cdot \rangle$ denoting the Euclidean scalar product, the latencies δ_m can first be estimated as

$$\hat{\delta}_m = \operatorname{argmax}_{\delta} (\mathbf{c}_m(\delta)).$$

This leads to the latency-corrected average

$$\hat{\mathbf{d}} = \sum_{m=1}^M \mathbf{x}_m(\cdot + \hat{\delta}_m),$$

which can now be used as new template for an improved estimation of latencies $\hat{\delta}_m$, and so on. This process is illustrated in [Figure 3.2](#).

No general convergence guarantees have been given for this procedure. As noted in [Woody \(1967\)](#) and several experimental studies after, the signal-to-noise ratio (SNR) has to be reasonably high in order to not align background activity. [John et al. \(1978\)](#) estimated a required root-mean-squared

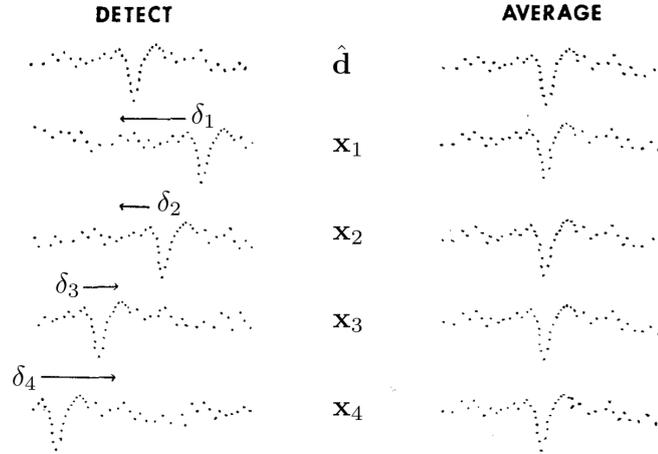


Figure 3.2: Illustration of Woody's method. The template $\hat{\mathbf{d}}$ is cross-correlated against the trials \mathbf{x}_m . The maximal values determine the respective latencies δ_m . After latency correction (right side) the signals are then averaged to obtain a new estimation $\hat{\mathbf{d}}$. Adapted from [Woody \(1967\)](#).

SNR of at least 0.5 in order for Woody's method to work. Otherwise, it is susceptible to align background noise instead of the signals. Several modifications for solving (3.2) have been proposed up to the present. [Möcks et al. \(1988\)](#) and [Pham et al. \(1987\)](#) suggested an algorithm in frequency domain with statistical testing on the presence of latency jitter as a main feature. [Cabasson and Meste \(2008\)](#) introduced a maximum-likelihood based variant to Woody's method that updates the template $\hat{\mathbf{d}}$ after each latency estimation of single trials, intended to improve and speed up convergence.

A different approach to the temporal alignment of ERPs is presented in [Gramfort et al. \(2010\)](#). The authors propose to use manifold learning techniques to first order trials with increasing response latencies. The optimal alignment is then determined by using a graph cuts algorithm.

3.3.2 Dynamic time warping

More general temporal variability can be obtained by considering non-decreasing warping functions τ_m , yielding

$$\mathbf{x}_m = \mathbf{d} \circ \tau_m + \epsilon_m, \quad m = 1, \dots, M. \quad (3.3)$$

In the speech processing community, a technique known as dynamic time warping (DTW) has been proposed to calculate such discrete warping functions τ_m using dynamic programming ([Itakura, 1975](#); [Sakoe and Chiba, 1971, 1978](#)). The general objective is to minimize the Euclidean distance $\|\mathbf{x}_m - \mathbf{d} \circ \tau_m\|_2$. Additionally, certain smoothness constraints on τ_m are imposed, preventing it from deviating too strongly from a linear function.

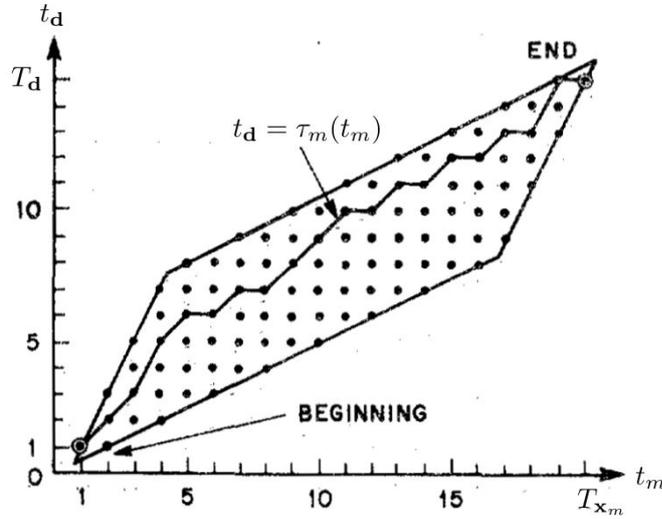


Figure 3.3: For two discrete signals with T_d and T_{x_m} sample points, respectively, dynamic time warping finds a path τ_m through the grid of sample pairs. The parallelogram restricts the path from deviating too strongly from a straight line. Adapted from [Itakura \(1975\)](#).

In the discrete case the problem can be formulated by finding a path that matches time samples of the two signals \mathbf{x}_m and \mathbf{d} , see illustration in Figure 3.3. The details of the implementation can be found for instance in [Rabiner and Juang \(1993, chapter 4\)](#).

In more recent works, it has been proposed to minimize additionally or instead the distance of the derivatives of the warped signals ([Keogh and Pazzani, 2001](#); [Wang et al., 1997](#)), as these were found to be a better measure of the actual shape of the time series than the absolute amplitudes.

Since the rate of human speech often varies non-uniformly, DTW has proven very useful for speech comparison up to the present. DTW has been found to be reasonably robust against white noise, though pre-smoothing is frequently used to minimize noise-related alignment errors ([Wang et al., 1997](#)). Recently, a multi-resolution approach to the problem has been proposed in [Salvador and Chan \(2007\)](#), performing linearly in computation time and storage – opposed to an originally quadratic complexity.

Despite the low signal-to-noise ratios (SNR) in single trial ERPs and the risk of warping noise and signals together, there has been interest for applying DTW to ERP processing. In [Picton et al. \(1988\)](#) for instance, the authors used DTW on auditory evoked potentials to obtain a template response across different subjects. This warped template has been found to better resemble the individual responses than does the grand-mean average (see Figure 3.4). It has also proven useful for combining individual evoked potentials. However, this approach was only found valid for root mean

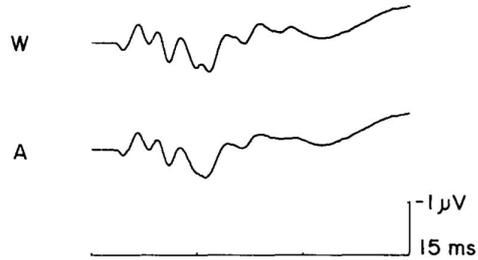


Figure 3.4: Comparison of combining auditory evoked responses from 32 subjects by dynamic time warping (W) and normal averaging (A), from [Picton et al. \(1988\)](#). The peaks of the warped template are of higher amplitudes and can be better distinguished.

squared SNRs of at least 0.5 (cf. previous section). A technique similar to DTW is based on cross-recurrences, that is, matches of similar data points between two signals ([Marwan et al., 2002](#)). This technique was used in [Ihrke et al. \(2009\)](#) to enhance the averaged ERP.

3.3.3 Variable response times

A particular case occurs when the duration of the ERP response is determined by an onset stimulus and a voluntary reaction response of the subject, for example, by pressing a button. In this case, traditional ERP averaging may be performed timelocked either to the stimulus or to the response. [Gibbons and Stahl \(2007\)](#) suggested instead to map the ERPs of different durations onto a common time interval prior to averaging.

The authors used fixed order monomials as rescaling functions τ_m , which in contrast to DTW did not depend on the shape of the recorded waveforms but only on the response times. This procedure thus remains applicable even for single trial ERPs with high noise levels. Applied to real data, the authors found the performance best for monomial orders 3 and 4, indicating higher temporal variability of later signal components.

3.4 Linear multi-component models

The methods described in the previous section were designed to recover a single waveform \mathbf{d} . In many applications, however, it can be of interest to extract different components $\{\mathbf{d}_k\}_{k=1}^K$. The simplest approach is to model the measured signals as linear combinations of these components, which reads

$$\mathbf{x}_m = \sum_{k=1}^K a_{km} \mathbf{d}_k + \epsilon_m, \quad m = 1, \dots, M, \quad (3.4)$$

with $a_{km} \in \mathbb{R}$ and ϵ_m again denoting background noise. However, this linear setting does not account for temporal variability across experimental trials,

and can produce smoothed waveforms as in Figure 3.1.

Instead of processing different trials, linear methods are often used on different channels, recorded simultaneously with EEG or MEG. In this case, a physiological interpretation can be given to (3.4): Modeling the neural sources as electric current dipoles (cf. Section 2.2) and assuming their spatial locations and orientations to be fixed in time, the relationship between the current amplitudes in sources and sensors are linear and given by the so called *leadfields*. Thus, for \mathbf{x}_m denoting the different recording channels and \mathbf{d}_k the neural sources, the leadfields correspond to the coefficients a_{km} in (3.4).

If \mathbf{x}_m and \mathbf{d}_k are discrete signals with $N \in \mathbb{N}$ sampling points, (3.4) can be formulated as a matrix factorization problem

$$\mathbf{X} = \mathbf{D} \mathbf{A} + \mathbf{E}, \quad (3.5)$$

where the columns of $\mathbf{X}, \mathbf{E} \in \mathbb{R}^{N \times M}$ and $\mathbf{D} \in \mathbb{R}^{N \times K}$ contain the vectors \mathbf{x}_m, ϵ_m , and \mathbf{d}_k , respectively, and $\mathbf{A} = \{a_{km}\} \in \mathbb{R}^{K \times M}$ is the coefficient matrix. An obvious degeneracy in the model lies in the fact that component amplitudes may be captured both in \mathbf{D} and in \mathbf{A} . For convenience, we will assume all components \mathbf{d}_k to be normalized in the l_2 -sense during this entire section, such that their amplitudes (or more precisely l_2 -norms) will be captured exclusively in the coefficients a_{km} . Even with this constraint, there exist many ways to realize such a decomposition. Different assumptions on the components \mathbf{d}_k lead to the techniques principal component analysis (PCA) and independent component analysis (ICA), which will be introduced in the following subsections.

In this section, we assume $K \leq M$, that is, the number of components to be extracted is less or equal to the number of provided channels or trials. The case of large K leads to sparse coding techniques and dictionary learning which will be addressed in Section 3.6.

3.4.1 Principal component analysis (PCA)

If the number of components K is smaller than the number of channels or trials M , it is a natural objective to maximize the data variance in the components $\{\mathbf{d}_k\}$. Principal component analysis (PCA), which was first proposed in Pearson (1901), solves this objective for every $K \leq M$ in a single decomposition. Formally, a full PCA of $\mathbf{X} \in \mathbb{R}^{N \times M}$ yields the decomposition $\mathbf{X} = \mathbf{D} \mathbf{A}$ into the orthogonal matrix $\mathbf{D} \in \mathbb{R}^{N \times M}$ and the coefficient matrix $\mathbf{A} \in \mathbb{R}^{M \times M}$, such that for any $K < M$ we have

$$\left(\mathbf{D}^{(K)}, \mathbf{A}^{(K)} \right) = \underset{\mathbf{D}' \in \mathbb{R}^{N \times K}, \mathbf{A}' \in \mathbb{R}^{K \times M}}{\operatorname{argmin}} \|\mathbf{X} - \mathbf{D}' \mathbf{A}'\|_F,$$

with the submatrices $\mathbf{D}^{(K)} = \{\mathbf{d}_1, \dots, \mathbf{d}_K\}$ and $\mathbf{A}^{(K)} = \{a_{km}\}_{k \leq K}$ and with $\|\cdot\|_F$ denoting the Frobenius norm. Hence, when performing a full PCA, any

number K of first principal components gives a set of normalized, orthogonal vectors maximizing the data variance. This allows to choose the number K *a posteriori* without the need to recalculate the principal components.

PCA is mathematically equivalent to a singular value decomposition (SVD),

$$\mathbf{X} = \mathbf{U} \mathbf{S} \mathbf{V}^t$$

with orthogonal matrices $\mathbf{U} \in \mathbb{R}^{N \times M}$ and $\mathbf{V} \in \mathbb{R}^{M \times M}$. The matrix $\mathbf{S} \in \mathbb{R}^{M \times M}$ contains the singular values on the main diagonal in decreasing order and is zero everywhere else. Now, $\mathbf{U} \equiv \mathbf{D}$ contains the principal components and $\mathbf{S} \mathbf{V}^t \equiv \mathbf{A}$ is the coefficient matrix .

The hierarchical ordering of principal components can be very useful for data compression and dimensionality reduction, as well as for separating signal (first components) from noise (last components). However, this separation of components does not correspond well to any physiological interpretation. On the contrary, variance maximization and orthogonality encourage mixing of signal components and noise if these are of similar amplitudes and linearly correlated. While being a popular ERP processing technique in the 80s and 90s, the limitations of PCA have frequently been reported. In a study with different simulated signals, [Wood and McCarthy \(1984\)](#) observed that the principal components contained large mixtures of the true components. [Lagerlund et al. \(1997\)](#) used PCA for spatial filtering and artifact removal from multi-channel EEG but acknowledged its inability to completely separate artifacts from signal, especially when amplitudes were similar. In addition, distortions of activities overlapping the removed artifacts were observed.

In order to enhance separation and reduce overlap between components, additional transformations such as Varimax ([Rösler and Manzey, 1981](#)) have often been used to post-process the principal components.

We will demonstrate PCA's performance for waveform separation in Chapter 5 (Figure 5.5 and 5.8), where it is compared to ICA and AWL.

3.4.2 Independent component analysis (ICA)

Another way to approach the decomposition (3.5) is to assume higher order statistical independence between the components $\{\mathbf{d}_k\}$. This problem was introduced as independent component analysis (ICA) in [Comon \(1994\)](#) in the engineering field of signal processing and belongs to the class of blind source separation (BSS) methods. ICA algorithms generally rely on the assumption that the components have non-Gaussian distributions. The most frequently used ICA implementations are the algorithms JADE ([Cardoso, 1999; Cardoso and Souloumiac, 1993](#)), infomax ICA ([Bell and Sejnowski, 1995](#)), and FastICA ([Hyvarinen, 1999](#)).

The most prominent example for blind source separation is the so called cocktail-party problem, where M microphones simultaneously record differ-

ent mixtures of speech signals from M speakers (Hyvärinen and Oja, 2000). ICA proved capable of well separating these signals based on their statistical independence and non-Gaussian distributions. It has furthermore been applied to a variety of problems from different domains. Some examples are: extraction of independent features from images, e.g., for face recognition (Bartlett, 2001, chapter 2); analysis of simultaneous econometric time-series, e.g., for predicting stock prices (Back and Weigend, 1997); and separating a user’s signal from interfering ones in mobile communications (Ristaniemi and Joutsensalo, 1999).

In a pioneering application to EEG data in Makeig et al. (1996), the authors found ICA well-suited for neural source identification, artifact detection, and separation of overlapping activity. In Vigário et al. (1998), ICA was successfully applied to MEG recordings. ICA has since become a very widely used EEG and MEG processing tool, mostly for the identification and removal of artifacts (Delorme et al., 2007; Gwin et al., 2010; Jung et al., 2000). An example from Jung et al. (2000) shown in Figure 3.5 illustrates the application of ICA to EEG time courses, isolating artifacts produced by a slow eye movement and muscle activity.

However, there are several limitations to the use of ICA in neuroelectrical signal processing. Most importantly, the validity of assuming independent and non-Gaussian neural source activity has frequently been challenged. In Hyvärinen et al. (2010), the authors argue that ICA’s reduced ability to find “interesting” neural sources comes mainly from the fact that the distributions of these sources are often not far from Gaussian. They suggest to first represent the data in time-frequency domain. In this domain, the most interesting signals have sparse representations and are therefore super-Gaussian, which would facilitate their identification through ICA. In addition, the strict isochronicity of neural sources in the linear model can be relaxed through the use of complex coefficients, which can capture phase shifts.

Another shortcoming of ICA is that the number of independent components is equal to the number of input signals, that is, $K = M$. However, there are often much fewer neural sources than recording channels. Some algorithms provide the option to calculate any desired number $K \leq M$ of components. The FastICA algorithm, for instance, proceeds by successively extracting components and can be stopped at any step. However, no guarantee is given that these components are the relevant ones, as there is no hierarchy across components such as in PCA. If only $K < M$ components shall be calculated, it can therefore be beneficial to first reduce the input dimension, for example, by performing a PCA.

Exhaustive discussions on the use of ICA in EEG signal processing can be found, for instance, in Onton et al. (2006) and Groppe et al. (2008).

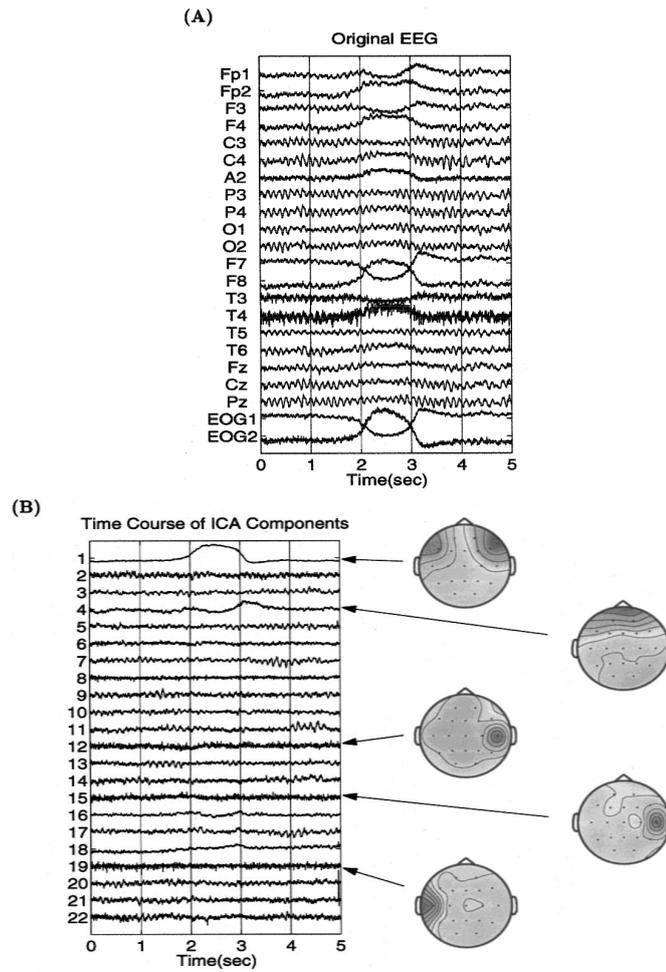


Figure 3.5: Illustration of artifact identification through ICA from Jung et al. (2000) (image adapted). The five second time window of multi-channel EEG recording (A) contains a prominent slow eye movement. After performing ICA (B), artifacts due to horizontal and vertical eye movement (components 1 and 4) and temporal muscle activity (components 12, 15, 19) could be identified.

3.5 Combined approaches

The methods presented above each address one particular aspect of variability in neuroelectrical signal processing: (i) temporal variability often encountered across experimental trials or subjects (Section 3.3) or (ii) varying amplitudes across components, which are well suited for multi-channel data due to their linearity (i.e., isochronicity). Each method is therefore optimized to capture either cross-trial/cross-subject or cross-channel variability, but will be suboptimal in combined multi-channel, multi-trial settings. Therefore, two or more steps are typically performed: for instance, a multi-channel ERP is first obtained by averaging over experimental trials which will then be decomposed into active sources/artifacts by means of ICA. The obvious problem of such a processing pipelines, is the risk of losing crucial information in a every step. Differentially variable component analysis (dVCA) therefore combines both aspects of variability and allows to process multiple trials and channels in a single framework.

3.5.1 Differentially variable component analysis (dVCA)

The model for differentially variable component analysis (dVCA) proposed in [Truccolo et al. \(2003\)](#) is given by

$$\mathbf{x}_m = \sum_{k=1}^K a_{km} \mathbf{d}_k(\cdot - \delta_m) + \epsilon_m, \quad m = 1, \dots, M$$

and thus combines the linear model (3.4) and Woody’s model (3.2). In [Truccolo et al. \(2003\)](#) the, the signals \mathbf{x}_m represented LFP trials (cf. Section 2.3.2) and \mathbf{d}_k the different components of the event-related response with variable latencies.

The authors proposed an alternate minimization approach, where in each iteration the estimates for latencies δ_m , waveforms \mathbf{d}_k , and finally amplitudes a_{km} are updated separately. In addition, they suggested learning the representation in a hierarchical manner, starting with a single component \mathbf{d}_1 and corresponding parameters a_{1m} , δ_m . As long as the residual signal still contains relevant structures, new components are added to the learning procedure.

The same authors extended the dVCA model in [Knuth et al. \(2006\)](#) to include multi-channel recordings. This extended model then reads

$$\mathbf{x}_{cm} = \sum_{k=1}^K b_{ck} a_{km} \mathbf{d}_k(\cdot - \delta_m) + \epsilon_{cm}, \quad m = 1, \dots, M, c = 1, \dots, C,$$

with c indexing different channels. In this model, \mathbf{d}_k represent the active sources in the brain. The coefficients b_{ck} describe the amplitude relation between sources and sensors, that is, the leadfields (cf. Section 3.4). Note

that the latencies δ_m vary only across trials but remain constant across channels, due to their isochronicity.

A variant to dVCA has been proposed in [Xu et al. \(2009\)](#) which solves the multi-component model in frequency instead of time domain. In addition, it uses an autoregressive model for ϵ_{mc} , in order to better describe ongoing background activity.

3.6 Sparse representations

As in the Section 3.4, we again consider linear representations of the signals

$$\mathbf{x}_m = \sum_{k=1}^K a_{km} \mathbf{d}_k + \epsilon_m. \quad (3.6)$$

However, we now assume the set $\mathbf{D} = \{\mathbf{d}_k\}_{k=1}^K$ to be large and often overcomplete, that is, $K > N$ for N denoting the number of discrete time samples in each signal. Such an overcomplete set \mathbf{D} is called a *dictionary* and the contained features \mathbf{d}_k are referred to as *atoms*. These representation are especially useful – both for interpretation and compression – if they are sparse, that is, most of the coefficients a_{km} are zero.

Sparsity has proven a very useful property when representing images or signals. This is especially the case for time-frequency representations, since natural images and signals contain only information from limited frequency spectra. In neuroelectrical image processing, sparsity can often be naturally motivated. The current amplitudes generated by a neural source, for example, have a sparse distribution over recording channels as they are only measurable in sensors nearby. Sparsity may not only occur across channels: it has been observed that certain response components are present only in a subset of recorded experimental trials.

Traditionally, the sparse decomposition were performed over predefined dictionaries that were known to yield sparse representations, such as windowed sinusoidal functions or different types of wavelets ([Mallat, 1999](#)). A more recent approach is to combine the optimal design of the dictionary and the calculation of the sparse representation, which is known as dictionary learning. Both concepts are presented in the following subsections.

3.6.1 Sparse coding and time-frequency representations

Sparsity can be measured simply by counting the number of non-zero entries in the coefficient vector $\mathbf{a} = \{a_1, \dots, a_K\}$, which is denoted by the l_0 -norm $\|\mathbf{a}\|_0$. When a dictionary $\mathbf{D} = \{\mathbf{d}_k\}$ is provided, the *sparse coding* problem for a single signal \mathbf{x} can be formulated as the minimization

$$\min_{\mathbf{a}} \left\| \mathbf{x} - \sum_{k=1}^K a_k \mathbf{d}_k \right\|_2^2, \quad \text{s.t. } \|\mathbf{a}\|_0 \leq L, \quad (3.7)$$

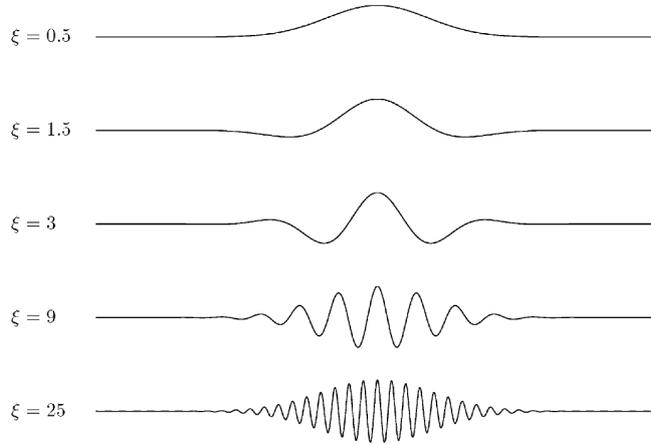


Figure 3.6: The real part of a Gabor atom is shown for different frequency modulations ξ while keeping the window size σ fixed. Source: [Béнар et al. \(2009\)](#).

where $L \ll K$ regulates the amount of sparsity. However, this problem is non-convex and NP-hard. Greedy algorithms such as matching pursuit (MP) ([Mallat and Zhang, 1993](#)) or orthogonal matching pursuit (OMP) ([Pati et al., 1993](#)) iteratively select the atoms \mathbf{d}_k most correlated with the signal \mathbf{x} . Alternatively, the problem can be relaxed by using the convex l_1 -norm, denoted $\|\cdot\|_1$, which is known to promote sparsity. The corresponding soft-constrained problem

$$\min_{\mathbf{a}} \left\| \mathbf{x} - \sum_{k=1}^K a_k \mathbf{d}_k \right\|_2^2 + \lambda \|\mathbf{a}\|_1, \quad (3.8)$$

with regularization parameter $\lambda > 0$, is convex and known as the Lasso ([Tibshirani, 1996](#)). As noted in [Mairal \(2010\)](#), the Lasso admits a closed form solution, if the indices and signs of the non-zero coefficients are known. Hence, one strategy is to search these *active* coefficients in successive steps, which is efficiently achieved by least angle regression (LARS) ([Efron et al., 2004](#)). Other popular algorithms to solve the Lasso include the proximal methods ISTA ([Chambolle et al., 1998](#)) and its accelerated version FISTA ([Beck and Teboulle, 2009](#)), as well as block coordinate descent using iterative thresholding ([Daubechies et al., 2004](#)).

We provide more details on MP and LARS in Appendix A. Both algorithms share the characteristic of successively selecting active atoms. This property will become very useful for the algorithm AWL introduced in the next chapter, as it allows to easily enforce additional sparsity and non-negativity constraints in each step.

The choice of the dictionary \mathbf{D} is crucial for the resulting representation. In image and signal processing, windowed Fourier bases and wavelets

(Mallat, 1999) have always been popular choices. Besides their good mathematical properties such as orthogonality and shift and scaling invariance, which allow for efficient calculation, they have shown to be suited for sparsely representing natural signals and images. In addition, they facilitate the interpretation of signals by representing them in the time-frequency domain.

A particularly popular choice is a dictionary of Gabor atoms, notably for processing EEG signals. These can be described by dilation γ , translation δ , and frequency modulation ξ of a Gaussian window function $g(t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}}$,

$$\mathbf{d}_{\gamma,\delta,\xi} = \frac{1}{\sqrt{\gamma}} g\left(\frac{1}{\gamma}(t - \delta)\right) e^{i\xi t}. \quad (3.9)$$

The real part of a Gabor atom is illustrated in Figure 3.6 for different frequencies ξ . In a pioneering work of applying sparse coding techniques to EEG signals, Durka and Blinowska (1995) used the MP algorithm with Gabor atoms to localize sleep spindles.

Note that the sparse coding formulas (3.7) and (3.8) address a single signal \mathbf{x} . Hence, when addressing multivariate data, similarities or coupling of the different trials or channels are not explicitly taken into account. More recent works have therefore adapted sparse coding techniques to exploit interdependencies between signals. A multi-channel version of MP proposed in (Durka et al., 2005) selects atoms that have the largest average correlation across all channels.

An advantage of using complex-valued time-frequency representation is the possibility to treat magnitude and phase of each frequency component separately. This is especially useful for cases of induced (i.e., not phase-locked) activity (cf. Section 2.5.1). In Tallon-Baudry and Bertrand (1999), the authors suggest to average over time-frequency power maps of single trials to reveal induced gamma activity, see Figure 3.7. The same idea is also employed in multi-variate versions of MP, which select atoms whose complex correlation coefficients have maximal average modulus (Tropp et al., 2006) or energy (Gribonval, 2003) across channels. Finally, Bénar et al. (2009) suggested a more general approach that allows for variations of all atom parameters across experimental trials, including amplitude, latency, frequency, and phase.

While sparse coding uses a model of linear combinations of signal components like PCA and ICA, it strongly differs from these techniques qualitatively. This is due to the fact that a dictionary typically contains a large number of atoms and is often shift- and scale-invariant. In this sense, these dictionaries are well suited to capture temporal variability such as latency jitter, change of duration, frequency or phase. However, in many cases the shapes of the atoms do not well represent the characteristic ERPs. These potentials are often asymmetric and not well-localized in time-frequency domain. Hence, in the case of a Gabor dictionary, for instance, a large number

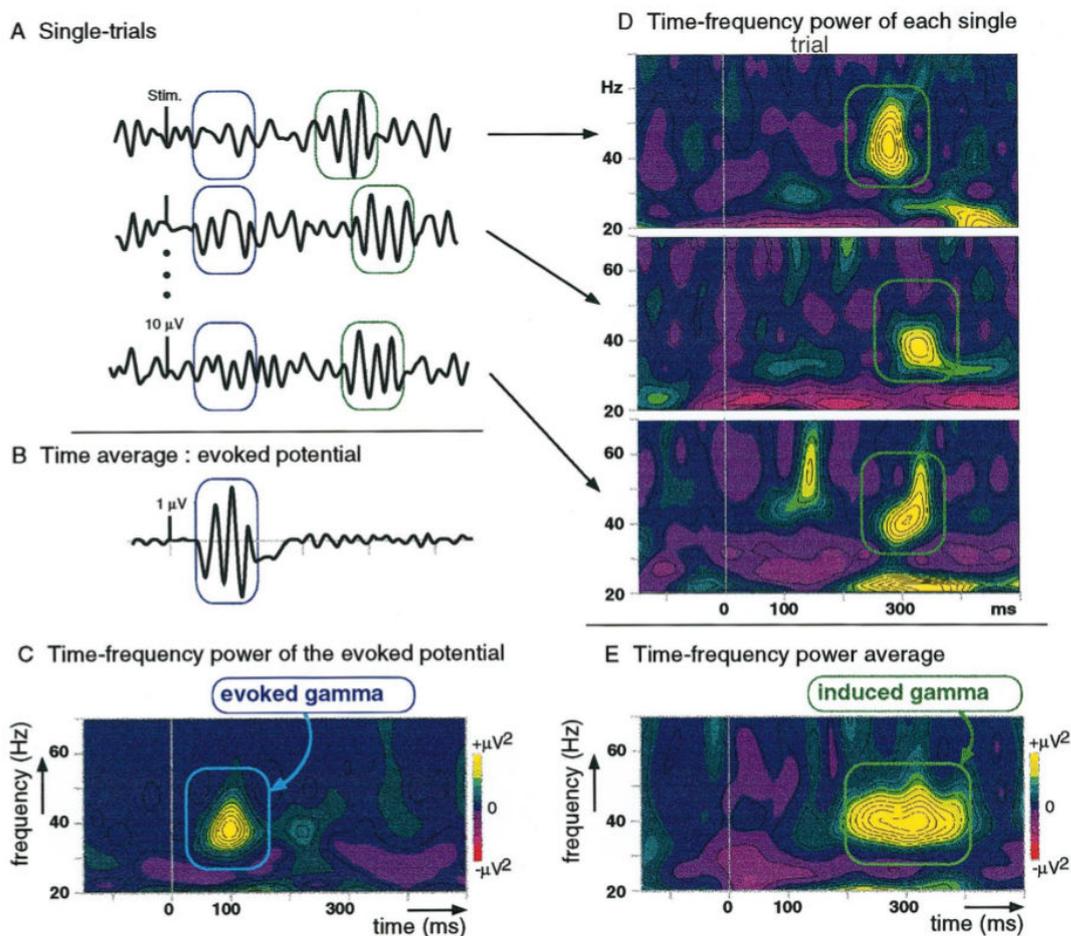


Figure 3.7: Illustration from Tallon-Baudry and Bertrand (1999), showing how to obtain evoked (i.e., phase-locked) and induced (i.e., not phase-locked) gamma activity through different averaging techniques. For evoked activity, single trials (A) are first averaged time-locked to the stimulus. The frequency content of the resulting evoked potential (B) can be clearly seen in the time-frequency power map (C). However, the induced activity is not visible in this approach, as averaging over the different phases diminishes the power of the frequency content. In contrast, averaging over time-frequency power maps of the individual trials (D) disregards the differences in phase and leads to a clear appearance of the induced gamma activity (E).

of these symmetric, time-frequency localized atoms is needed to encode the ERP shape.

As the shapes of the ERPs are usually not exactly known *a priori*, it is not clear how to design the optimal dictionary. Instead of defining it beforehand, it may therefore be beneficial to *learn* it together with the decomposition. This approach will be discussed in the following subsection.

3.6.2 Dictionary learning

The problem of designing a dictionary \mathbf{D} for given signals \mathbf{x}_m with the goal of yielding the sparsest decomposition was first discussed in [Olshausen and Field \(1997\)](#). The resulting dictionary learning problem can be stated as

$$\mathbf{D} = \underset{\mathbf{D}}{\operatorname{argmin}} \min_{\{\mathbf{a}_m\}} \sum_{m=1}^M \left(\frac{1}{2} \|\mathbf{x}_m - \mathbf{D}\mathbf{a}_m\|_2^2 + \lambda \|\mathbf{a}_m\|_1 \right), \quad (3.10)$$

where the $\mathbf{a}_m = (\mathbf{a}_{1m}, \dots, \mathbf{a}_{Km})$ denote the coefficient vectors. As before, we constrain the columns of \mathbf{D} to be normalized. This joint minimization is non-convex and can be seen as a generalization of the K -means clustering algorithm for vector quantization (VQ) which is also non-convex. This algorithm uses alternate minimization and inspired many dictionary learning algorithms, such as MOD ([Engan et al., 1999](#)) and K -SVD ([Aharon et al., 2006](#)). Their general procedure consists of alternating between updates of \mathbf{D} and $\mathbf{A} = \{\mathbf{a}_m\}$ while leaving the other fixed; this is illustrated in [Figure 3.8](#).

Dictionary learning has become a popular tool in image processing and has shown to yield state-of-the-art results in image compression, denoising, inpainting, and classification ([Elad and Aharon, 2006](#); [Mairal et al., 2008](#)). In order to cope with growing sizes of datasets, [Mairal et al. \(2009\)](#) suggested an efficient online version of the algorithm. Moreover, dictionary learning has successfully been applied to different types of medical signals and images ([Mailhé et al., 2009](#); [Tosic et al., 2010](#)) and audio data ([Jafari and Plumbley, 2009](#)). A recent overview of dictionary learning techniques and applications can be found in [Tosic and Frossard \(2011\)](#).

Despite the great success of dictionary learning techniques in many fields, they have been used only in few neuroelectrical applications, such as signal classification for brain-computer interfaces (BCI) in [Hamner et al. \(2011\)](#). One reason might be the large and generally unstructured learned dictionaries, which make interpretation of the resulting waveforms difficult. Exploiting invariances can help to reduce the number of waveforms that “generate” the dictionary.

3.6.3 Translation-invariant dictionary learning

Dictionary learning has not found much interest in neuroelectrical signal processing – despite its success in many other field. This may result from the

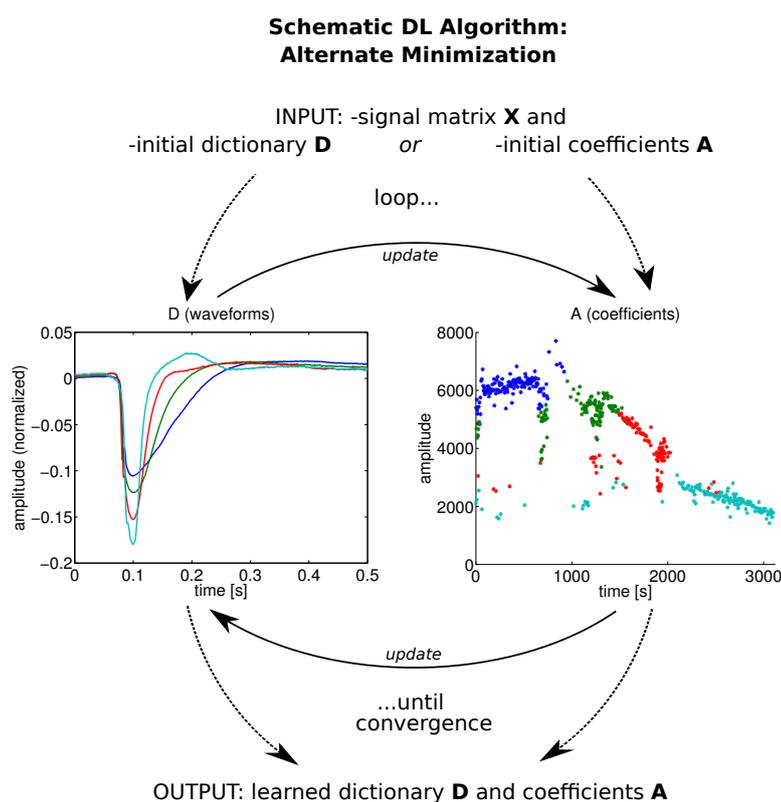


Figure 3.8: Typical algorithm scheme for dictionary learning using alternate minimization. After initializing either the dictionary \mathbf{D} or the coefficients \mathbf{A} , alternating updates are performed over the respective matrices. After numerical convergence (or a fixed number of iterations), the algorithm returns the final \mathbf{D} and \mathbf{A} . The figures used for the illustration correspond to the epileptiform spikes learned in Section 6.4.1.

following two inconveniences: (i) learned dictionaries are typically large and unstructured and (ii) they do not explicitly account for temporal variability. In particular, (ii) means that occurrences of a structure at different positions across signals have to be encoded by different atoms.

Exploiting invariances with respect to translations or other transformations can help to reduce the number of waveforms to be learned and obtain more structured or parametrized data representations. In addition, translation-invariance allows to compensate for temporal jitter. Various shift-invariant DL versions have therefore been proposed (Blumensath and Davies, 2006; Jost et al., 2006; Mailhé et al., 2008; Plumbley et al., 2006) and applied to images and audio data (Grosse et al.; Mørup et al., 2008). Rotation invariance has also been used for dictionaries learned on images (Wersing et al., 2003) and multi-variate signals (Barthélemy et al., 2012).

In Jost et al. (2005), the authors applied translation-invariant dictionary learning to EEG signals, resulting in oscillatory waveforms representing Alpha (8-12 Hz) and Beta (13-30 Hz) rhythms. A multi-variate shift-invariant version of dictionary learning is introduced in Barthélemy et al. (2013) to learn multi-channel EEG waveforms on BCI data, showing both the reconstruction qualities and ability to learn a meaningful representation of a P300 (cf. Section 2.5.1).

3.7 Conclusion and outlook

In this chapter, we presented different techniques that are commonly used to process neuroelectrical signals. In particular, we saw different forms of signal variability addressed by these methods, notably temporal variability (most prominently latency shifts) and varying amplitudes of different components in linear models. While some of the approaches explicitly combine different types of variability in a single model (e.g., dVCA and translation-invariant dictionary learning), most of them address only one particular aspect of variability.

In the following chapter, we introduce a new model, adaptive waveform learning (AWL), which generalizes the concepts of variability through the use of mathematical operators on signal components. The AWL model then contains many of the here presented models as special cases. Due to its generality, AWL can be specialized for different concrete settings, which is achieved by specifying implementational details in the generic AWL algorithm. Two concrete cases, processing of epoched and non-epoched recordings will be presented in chapters 5 and 6.

Chapter 4

Adaptive Waveform Learning (AWL)

In this chapter, we introduce a novel signal decomposition model, adaptive waveform learning (AWL), which explicitly includes variability of signal components. We propose a generic algorithm, based on sparse coding techniques, which will be concretely implemented in the following chapters.

Contents

4.1	Introduction	54
4.2	Modeling variability	54
4.2.1	Linear operations on signal components	55
4.3	AWL model	57
4.3.1	General model	57
4.3.2	Sparse coefficients	58
4.3.3	Minimization problem	59
4.3.4	Explicit representation	59
4.4	Algorithm AWL	60
4.4.1	Initialization	60
4.4.2	Coefficient updates	61
4.4.3	Waveform updates	62
4.4.4	Waveform centering	63
4.5	Relation to previous methods	64
4.6	Hyperparameters and hierarchical AWL	65

4.1 Introduction

In this chapter, we introduce a general signal decomposition model, from which we derive the generic adaptive waveform learning (AWL) algorithm. In particular, this framework serves as a means of processing event-related potentials (ERPs) as well as spontaneously repeating activity (SRA) (cf. Section 2.5). It may be applied to both epoched data and long, non-segmented recordings. The corresponding specifications and implementations are described in the following two chapters. A key ingredient to the approach is the explicit modeling of variability of signal components across time or different trials. This is described through linear operations on the component waveforms, such as translations and dilations, to model variability of waveform latencies and durations, respectively.

The generality of the framework allows to formulate most of the techniques from the previous chapter as special cases of our model. We will explicitly point out these relations, which will provide a better comparison of the different methods.

This chapter focusses primarily on analytical aspects of AWL. For this purpose, we assume a continuous setting, that is, all temporal signals \mathbf{x} considered in this chapter will be defined on \mathbb{R} . We can furthermore assume that

$$\mathbf{x} \in L^2 \equiv \left\{ f : \mathbb{R} \rightarrow \mathbb{R}; \int_{\mathbb{R}} f(t)^2 dt < \infty \right\},$$

as the actual recorded signals are of finite temporal duration and thus compactly supported. Aspects concerning discretization will mainly be treated in the following chapters, together with other implementational details.

4.2 Modeling variability

As we have seen in the previous chapters, variability in the signals of interest has been a major concern in neuroelectrical processing tasks. Variability of a signal component across time, experimental trials, or different subjects may be seen as a certain morphological deformation of this component at each of its instances. When modeling these deformations, it is indispensable to be at the same time (i) as general as necessary to capture all empirically relevant variability and (ii) specific enough to avoid noise fitting. The most simple case of component variability is a scalar multiplication allowing for amplitude changes. This is accounted for in linear models such as PCA, ICA, and sparse representations (cf. sections 3.4 and 3.6).

For practical reasons, it is useful to consider deformations with good mathematical properties. We will restrict ourselves to operators that describe linear transformations of signal components, as defined below in (4.1). This set is large enough to contain all relevant deformations for our purposes.

In fact, the signal deformations discussed in the previous chapter all involved shifting and rescaling of the time axis (cf. Section 3.3), which are linear operations on the signal components. In addition, operations should be of high rank, that is, close to invertible, in order to provide for a well-posed decomposition problem. Another important aspect is the possibility of efficient implementations of the operations in the discrete case. Especially desirable are orthogonal operations, leading to particularly simple solutions. Finally, for interpreting or post-processing the calculated signal representations, it can be very useful that transformations are easily parametrizable.

After considering general linear operations, we will focus on translations and dilations. These operations can capture a great variety of variability, such as differences in waveform latencies, phases, durations, and frequencies. In addition, translations and dilations are orthogonal operations, that is, they preserve the scalar product.

4.2.1 Linear operations on signal components

Consider a set Φ of linear operators on one-dimensional real-valued signals, that is, for every $\phi \in \Phi$, $\alpha, \beta \in \mathbb{R}$, and functions $\mathbf{x}, \mathbf{y} \in L^2$,

$$\phi(\alpha \mathbf{x} + \beta \mathbf{y}) = \alpha \phi(\mathbf{x}) + \beta \phi(\mathbf{y}) \quad (4.1)$$

holds. An important subset are linear operations that rescale the time axis (cf. Section 3.3). These are described by

$$\phi_\tau(\mathbf{x}) = \mathbf{x} \circ \tau$$

for some function $\tau : \mathbb{R} \rightarrow \mathbb{R}$ and \circ denoting function composition. We only consider temporal scaling functions τ that are strictly increasing and surjective. This implies that we assume the changes of each signal component to always occur in the same temporal order, but with possibly different durations across experimental trials or subjects.

With these constraints the rescaling functions τ are invertible and so are the corresponding *warping* operators $(\phi_\tau)^{-1} = \phi_{\tau^{-1}}$. Given the Euclidean scalar product $\langle \cdot, \cdot \rangle$ and using change of variables, we have

$$\begin{aligned} \langle \phi_\tau(\mathbf{x}), \phi_\tau(\mathbf{y}) \rangle &= \int_{\mathbb{R}} \phi_\tau(\mathbf{x}(t)) \phi_\tau(\mathbf{y}(t)) dt \\ &= \int_{\mathbb{R}} \mathbf{x}(\tau(t)) \mathbf{y}(\tau(t)) dt = \int_{\mathbb{R}} \mathbf{x}(T) \mathbf{y}(T) d\tau^{-1}(T). \end{aligned}$$

Hence, in the case of a constant derivative $\gamma := \frac{d\tau^{-1}(T)}{dT}$, we thus obtain

$$\langle \phi_\tau(\mathbf{x}), \phi_\tau(\mathbf{y}) \rangle = \gamma \langle \mathbf{x}, \mathbf{y} \rangle.$$

This shows that ϕ_τ preserves the scalar product up to multiplication with a constant if τ is affine, that is,

$$\tau = \frac{1}{\gamma}(\cdot - \delta)$$

for $\gamma, \delta \in \mathbb{R}, \gamma > 0$. This yields the affine orthogonal warps

$$\phi_{\delta, \gamma}(\mathbf{x}) = \frac{1}{\sqrt{\gamma}} \mathbf{x} \left(\frac{1}{\gamma}(\cdot - \delta) \right)$$

which are compositions of translations and dilations.

In the discrete case, any warping operator ϕ_τ may be represented by a sparse matrix (entries in the lower left and upper right are zero). In general, discretization and boundary effects lead to warping matrices which are not strictly invertible.

4.2.1.1 Translations

A significant part of the variability encountered in neurological signals are captured by translations

$$\phi_\delta(\mathbf{x}) = \mathbf{x}(\cdot - \delta).$$

On the one hand, translations may account for different onset latencies of transient events such as event-related potentials (Woody, 1967); on the other hand they can describe phase differences between oscillatory phenomena (Bastiaansen et al., 2012).

Besides their orthogonality, the translation operators exhibit another important property which allows for efficient implementation in the discrete case. A significant part of the computations required for obtaining the signal representations introduced in the following sections will consist in calculating correlations between transformed waveforms. In the case of translations, this means calculating cross-correlations, which may be computed using the fast Fourier transform (FFT) thanks to the property

$$\mathcal{F}\{\mathbf{x} \star \mathbf{y}\} = \mathcal{F}\{\mathbf{x}\}^* \cdot \mathcal{F}\{\mathbf{y}\},$$

analogous to the convolution theorem. Here \mathcal{F} denotes the Fourier transform, \star the cross-correlation (see Section 3.3.1) and $(\cdot)^*$ complex conjugation.

The AWL data model which will be introduced in Section 4.3 only admits a finite number of possible transformations of signal components. For this purpose, we suggest the selection of a set of translations about equispaced latencies $\{\delta_{-P}, \dots, \delta_P; P \in \mathbb{N}\}$, with $\delta_p = p \cdot dt$ for some fixed temporal distance $dt > 0$. In the case of discrete signals, we will typically choose dt equal to the sampling resolution of the signals.

4.2.1.2 Dilations

Dilations

$$\phi_\gamma(\mathbf{x}) = \frac{1}{\sqrt{\gamma}} \mathbf{x} \left(\frac{1}{\gamma} \cdot \right)$$

may capture the different durations of transient responses. They can also account for frequency changes of oscillatory events. Empirically, these forms of variability are probably less frequently encountered than latency and phase shifts. In addition, they are less efficient to implement and require interpolation in the discrete case.

Nevertheless, changes in waveform duration have been considered in the literature (cf. sections 3.3.2 and 3.3.3). We will illustrate the utility of dilations in the case of epileptiform spikes of varying width in Section 6.4.2.

When only a finite set of dilations is considered, we suggest logarithmically spaced stretches $\{\gamma_{-Q}, \dots, \gamma_Q; Q \in \mathbb{N}\}$, with $\gamma_q = \beta^q$ for some $\beta > 1$ which controls the resolution between stretches. Then, the maximal relative dilation factor is $\frac{\gamma_Q}{\gamma_{-Q}} = \beta^{2Q}$.

4.3 AWL model

In this section, we introduce our signal decomposition model, from which we derive the generic AWL algorithm. For this purpose, we will remain in a very general setting of a set of signals with repeating structures. This allows us to use the framework for a large number of neurological signal processing tasks; the specializations to the concrete settings of epoched and contiguous electrophysiological recordings will be made in chapters 5 and 6.

4.3.1 General model

Let $\{\mathbf{x}_m\}_{m=1}^M$ denote a set of temporal signals, with repeating structures across and possibly within each \mathbf{x}_m . We model these structures as waveforms $\{\mathbf{d}_k\}_{k=1}^K$, which may occur with different amplitudes, latencies, and changing shapes throughout the signals. Both latency and shape changes will be summarized in a set $\Phi = \{\phi_{-P}, \dots, \phi_P; P \in \mathbb{N}\}$ of linear operations (cf. Section 4.2), where we typically include pairs of an operator ϕ_p and its (possibly generalized) inverse $\phi_{-p} = \phi_p^{-1}$, as well as the identity $\phi_0 = \text{id}$. The finite set Φ can be seen as samples from a continuous distribution of possible deformations. For a good representation of this distribution, P may be chosen very large. The signals can now be modeled as linear combinations of the changing waveforms,

$$\mathbf{x}_m = \sum_{k=1}^K \sum_{p=-P}^P a_{kpm} \phi_p(\mathbf{d}_k) + \epsilon_m, \quad m = 1, \dots, M, \quad (4.2)$$

with ϵ_m the unmodeled part of each signal. Here, each non-zero coefficient $a_{kpm} \in \mathbb{R}$ reflects an occurrence of waveform \mathbf{d}_k in signal \mathbf{x}_m , with specific temporal location and deformation given by ϕ_p . For large P , this model will only be useful for interpretation if the coefficients a_{kpm} are *sparse* (i.e., mostly zero). Then, the non-zero a_{kpm} fulfill the function of selecting the appropriate latencies and deformations Φ of each waveform \mathbf{d}_k . Note that in order to prevent degeneracies of the model (cf. Section 3.4), we assume the \mathbf{d}_k and ϕ_p to be normalized,

$$\|\mathbf{d}_k\|_2 = 1 \quad \text{and} \quad \|\phi_p\|_2 \equiv \max_{\|\mathbf{d}_k\|_2=1} \|\phi_p(\mathbf{d}_k)\|_2 = 1. \quad (4.3)$$

Now, the energies of the waveforms (more precisely their l_2 -norms) are captured by the coefficients a_{kpm} .

4.3.2 Sparse coefficients

As already mentioned, the usefulness of model (4.2) depends on the sparsity of the coefficients a_{kpm} . There are different possibilities to impose sparsity. For the applications addressed in this thesis, we propose the following three types of constraints which encourage sparsity across coefficients. Depending on the application, any combination of these constraints may be enforced.

General sparsity:

If no further information about the structure of the sparsity is available, it can be imposed by assuming a general sparse distribution of coefficients \mathbf{a}_{kpm} . This leads to a non-convex l_0 -regularization of the corresponding minimization problem, cf. (3.7). Assuming instead a Laplacian distribution of the \mathbf{a}_{kpm} , leads to the convex Lasso problem with l_1 -prior, cf. (3.8). While different algorithms have been proposed to solve the Lasso exactly, their computational complexities exceed those of greedy algorithms, such as matching pursuit. For simplicity, we will only consider the Lasso formulation in this and the following chapters. Nevertheless, for the different implementations, we will use both l_0 - and l_1 -solvers.

Exclusivity:

In some cases, the occurrence of one waveform \mathbf{d}_k in a signal \mathbf{x}_m may exclude certain occurrences of the same or other waveforms. For example, in some situations, we want every waveform \mathbf{d}_k to occur at most once per signal \mathbf{x}_m (see Chapter 5). This is typically the case when processing event-related potentials (cf. Section 2.5.1), with \mathbf{d}_k representing the response components. Another exclusivity constraint can be obtained by imposing a minimal distance $\Delta > 0$ between waveform occurrences (see Chapter 6).

Since waveform occurrences are described through the non-zero coefficients, all exclusivity constraints can be formalized by defining a constraint set C that contains exclusive pairs of index combinations $(kp, k'p')$. Now the exclusivity constraint reads

$$\forall m \in \{1, \dots, M\} : (kp, k'p') \in C \Rightarrow (a_{kpm} = 0 \text{ or } a_{k'p'm} = 0). \quad (4.4)$$

Non-negativity:

In many settings (see chapters 5 and 6), we expect the waveforms \mathbf{d}_k to always occur with the same sign. This can be enforced using the non-negativity constraint

$$\forall k, p, m : a_{kpm} \geq 0. \quad (4.5)$$

Note, that this constraint also encourages sparsity, since negative coefficients will be set to zero.

4.3.3 Minimization problem

Since the set of possible waveform deformations Φ is given *a priori*, the unknowns in model (4.2) are the sparse coefficients a_{kpm} and the waveforms \mathbf{d}_k . They can be obtained by minimizing the residual signal. Considering the sparsity constraints above, we then have

$$\begin{aligned} \min_{a_{kpm}, \mathbf{d}_k} \sum_{m=1}^M \left(\left\| \mathbf{x}_m - \sum_{k=1}^K \sum_{p=-P}^P a_{kpm} \phi_p(\mathbf{d}_k) \right\|_2^2 + \lambda \sum_{k=1}^K \sum_{p=-P}^P |a_{kpm}| \right), \quad (4.6) \\ \text{s.t. } (4.4) \text{ and } (4.5) \text{ are satisfied,} \end{aligned}$$

with regularization parameter $\lambda > 0$.

Note that for $\Phi = \{\text{id}\}$ and without the constraints (4.4) and (4.5), the minimization reduces to the dictionary learning problem presented in Section 3.6.2. For the implementation of the minimization problem above, we therefore make use of the well-established dictionary learning algorithms based on alternate minimization (i.e., iteratively updating coefficients and waveforms separately). However, several adaptations are made to account for the presence of the transformations ϕ_p and the additional constraints. The resulting AWL method is summarized in Algorithm 1, which will be discussed in Section 4.4.

4.3.4 Explicit representation

The AWL representation of the signals \mathbf{x}_m is given by the set $\{a_{kpm}, \Phi, \mathbf{d}_k\}$. However, this formulation is sometimes not very practical, since the waveform occurrences are indirectly described through the non-zero coefficients a_{kpm} . We therefore propose an alternative representation $\{a_{klm}, \phi_{klm}, \mathbf{d}_k\}$.

Here, the a_{klm} denote only the coefficients of the actual waveform occurrences and ϕ_{klm} the corresponding latencies and deformations.

In the following implementation of the AWL algorithm, we will use the *sparse representation* $\{a_{kpm}, \Phi, \mathbf{d}_k\}$ during the update of the coefficients, as it allows to easily integrate sparse coding techniques. However, during the rest of the algorithm, we use the *explicit representation* $\{a_{klm}, \phi_{klm}, \mathbf{d}_k\}$, since it provides for more clarity in the formulas. A change between the two representations is denoted by

$$(\{a_{klm}\}, \{\phi_{klm}\}) = \text{SWITCH_REPRESENTATION}(\{a_{kpm}\}, \Phi).$$

4.4 Algorithm AWL

The general AWL method is described in Algorithm 1. Its iterative procedure consists of subsequent updates of coefficients and waveforms, described in the functions COEFFICIENT_UPDATES, WAVEFORM_UPDATES, and WAVEFORM_CENTERING, respectively. The iterations are terminated once a certain stopping criterion is satisfied, for example, after a fixed number of iterations or in the case of numerical convergence. Note that an initial set of waveforms has to be provided to start the iterative updates. This initialization and the three procedures are discussed in the following subsections.

Algorithm 1 Generic AWL

Input: $\{\mathbf{x}_m\}_{m=1}^M, \{\phi_p\}_{p=-P}^P, \{\mathbf{d}_k\}_{k=1}^K, \lambda \in \mathbb{R}_0^+, \text{constraint set } C.$

```

1: loop
2:   Set up dictionary  $\mathbf{D} = \{\mathbf{d}_k^p\}$  with  $\mathbf{d}_k^p = \phi_p(\mathbf{d}_k).$ 
3:    $\{a_{kpm}\} \leftarrow \text{COEFFICIENT\_UPDATES}(\{\mathbf{x}_m\}, \mathbf{D}, \lambda, C)$ 
4:    $(\{a_{klm}\}, \{\phi_{klm}\}) \leftarrow \text{SWITCH\_REPRESENTATION}(\{a_{kpm}\}, \Phi)$ 
5:   if stopping criterion reached then
6:     break
7:   end if
8:    $\{\mathbf{d}_k\} \leftarrow \text{WAVEFORM\_UPDATES}(\{\mathbf{x}_m\}, \{a_{klm}\}, \{\phi_{klm}\}, \{\mathbf{d}_k\})$ 
9:    $\{\mathbf{d}_k\} \leftarrow \text{WAVEFORM\_CENTERING}(\{a_{klm}\}, \{\phi_{klm}\}, \{\mathbf{d}_k\})$ 
10: end loop
    
```

Output: $\{a_{klm}\}, \{\phi_{klm}\}, \{\mathbf{d}_k\}.$

4.4.1 Initialization

The iterative updates in Algorithm 1 require a prior initialization of either waveforms or coefficients. We choose to initialize the waveforms, as in some cases estimates of their shapes are known *a priori*. Alternatively, estimations may be obtained from the data using a different method, for example, by performing a principal component analysis on $\{\mathbf{x}_m\}$.

As the constrained problem (4.6) is non-convex, the result will depend strongly on the initialization used. Hence, predefined waveforms may bias the result of the algorithm and may prevent it from learning significantly better representations. For these reasons, in our experiments, we generally preferred the use of random initializations with white Gaussian noise. An exception is the application to spike detection (cf. Section 6.4), for which we used an initial spike template.

4.4.2 Coefficient updates

The procedure COEFFICIENT_UPDATES in Algorithm 1 performs the minimization of (4.6) with constraints (4.4) and (4.5) for the coefficients a_{kpm} while leaving the waveforms \mathbf{d}_k fixed. As both the ϕ_p and the \mathbf{d}_k are fixed in this step, we can define the (very large) *dictionary* $\mathbf{D} = \{\mathbf{d}_k^p\}_{k=1, \dots, K}^{p=-P, \dots, P}$ of all transformed waveforms $\mathbf{d}_k^p \equiv \phi_p(\mathbf{d}_k)$. This eliminates the deformations ϕ_p and allows us to reformulate the minimization problem over the coefficients as a set of linear sparse coding problems.

```

1: procedure COEFFICIENT_UPDATES( $\{\mathbf{x}_m\}, \mathbf{D}, \lambda, C$ )
2:   for  $m = 1$  to  $M$  do
3:     Solve through sparse coding:
4:      $\{a_{kpm}\} \leftarrow \underset{\{a'_{kpm}\}}{\operatorname{argmin}} \left\| \mathbf{x}_m - \sum_{k=1}^K \sum_{p=-P}^P a'_{kpm} \mathbf{d}_k^p \right\|_2^2 + \lambda \sum_{k=1}^K \sum_{p=-P}^P |a'_{kpm}|$ ,
5:     s.t. (4.4):  $(kp, k'p') \in C \Rightarrow (a'_{kpm} = 0 \text{ or } a'_{k'p'm} = 0)$ 
6:     and (4.5):  $\forall k, p: a'_{kpm} \geq 0$ .
7:   end for
8: end procedure, return  $\{a_{kpm}\}$ 

```

Without the additional constraint, line 4 is exactly the Lasso problem, cf. (3.8), which can be solved by sparse coding techniques. In order to ensure the constraints (4.4) and (4.5), we suggest the use of an algorithm that selects the *active* (i.e., non-zero) coefficients for the most correlated atoms one by one. The simplest example is the greedy algorithm matching pursuit (MP) (Mallat and Zhang, 1993). But also least angle regression (LARS) (Efron et al., 2004) proceeds by iteratively building up the active set and has the advantage of exactly solving the Lasso problem. Details on both algorithms and efficient implementations are given in Appendix A.

Now the constraints can be ensured in each step of either MP or LARS: For non-negativity (4.5), we only select coefficients of atoms which are positively correlated with the data. After each activation of a coefficient a_{kpm} , we then mark all the coefficients which cannot be simultaneously active with a_{kpm} due to exclusivity (4.4). We can then simply exclude the marked coefficients from later activation. In the LARS algorithm, active coefficients

can also become deactivated in a later step. Hence, we need to verify which of the marked coefficients becomes admissible again for activation.

4.4.3 Waveform updates

Minimizing (4.6) for $\{\mathbf{d}_k\}$ while leaving the $\{a_{kpm}\}$ fixed is a convex problem as the l_1 -regularization and the additional constraints concern only the coefficients. We can solve this problem efficiently by performing block coordinate descent, that is, by iteratively minimizing for each \mathbf{d}_k . The update procedure is then described in WAVEFORM_UPDATES.

```

1: procedure WAVEFORM_UPDATES( $\{\mathbf{x}_m\}, \{a_{klm}\}, \{\phi_{klm}\}, \{\mathbf{d}_k\}$ )
2:   for  $k = 1$  to  $K$  do
3:      $\mathbf{d}_k \leftarrow \operatorname{argmin}_{\mathbf{d}_k} \sum_{m=1}^M \left\| \mathbf{x}_m - \sum_{k'=1}^K \sum_{l=1}^{L_{k'm}} a_{k'lm} \phi_{k'lm}(\mathbf{d}_{k'}) \right\|_2^2$ 
4:   end for
5: end procedure, return  $\{\mathbf{d}_k\}$ 
    
```

Note that we use the explicit representation described in Section 4.3.4, where $L_{k'm}$ denotes the number of occurrences (i.e., the non-zero coefficients) of waveform $\mathbf{d}_{k'}$ in signal \mathbf{x}_m .

By defining the linear transformations

$$\psi_{k'm} = \sum_{l=1}^{L_{k'm}} a_{k'lm} \phi_{k'lm},$$

we can rewrite line 3 as

$$\mathbf{d}_k = \operatorname{argmin}_{\mathbf{d}_k} \sum_{m=1}^M \left\| \mathbf{x}_m - \sum_{k'=1}^K \psi_{k'm}(\mathbf{d}_{k'}) \right\|_2^2.$$

Differentiating and satisfying the necessary condition for the minimum yields a closed form solution. It can be written as

$$\mathbf{d}_k = \left(\sum_{m=1}^M \psi_{km}^t \psi_{km} \right)^+ \sum_{m=1}^M \psi_{km}^t(\mathbf{r}_{km}), \quad (4.7)$$

where

$$\mathbf{r}_{km} = \mathbf{x}_m - \sum_{k' \neq k} \psi_{k'm}(\mathbf{d}_{k'}) \quad (4.8)$$

is the residual that results when subtracting all but waveform \mathbf{d}_k 's contribution from the signal \mathbf{x}_m . The term $(\cdot)^+$ denotes the Moore-Penrose pseudoinverse. Invertibility and conditioning of the operator $\sum \psi_{km}^t \psi_{km}$ are discussed later in Section 6.3.2 for translations and dilations.

Note that in block coordinate descent, several loops over the blocks usually need to be performed before convergence. However, in the cases we studied (see following chapters) one loop through the index set $\{1, \dots, K\}$ was sufficient for a good performance of Algorithm 1. This is due to the fact that the updates are initialized with waveforms calculated in the previous iteration of the algorithm, which are already good estimates for the new ones.

4.4.4 Waveform centering

We note that each \mathbf{d}_k only represents a template waveform which is subject to deformation and amplitude scaling at each of its occurrences. In order to avoid ambiguities in the concrete definition of this template, we suggest that each \mathbf{d}_k have unit norm according to (4.3) and be centered with respect to all deformations under which it occurs. The former constraint can simply be imposed by setting

$$\mathbf{d}_k \leftarrow \frac{\mathbf{d}_k}{\|\mathbf{d}_k\|_2}, \quad (4.9)$$

while the latter requires us to first define the *mean deformation* $\bar{\phi}_k$ of waveform \mathbf{d}_k with respect to all deformations $\{\phi_{klm}\}_{lm}$ previously calculated in UPDATE_COEFFICIENTS. A concrete definition of $\bar{\phi}_k$ can easily be given for the operations translation and dilation, as these are parametrizable over \mathbb{R} . This will be the subject of the following paragraph. A generalization to other operations is possible but will not be discussed here, as it is not relevant for this thesis.

Let $\phi \circ \phi'$ denote the composition of $\phi, \phi' : L^2 \rightarrow L^2$. Then, for $n \in \mathbb{N}$ we can recursively define $\phi^n = \phi^{n-1} \circ \phi$. Now considering compositions of translations and dilations

$$\phi_{\delta, \gamma}(\mathbf{x}) \equiv \frac{1}{\sqrt{\gamma}} \mathbf{x} \left(\frac{1}{\gamma} \cdot -\delta \right),$$

we can extend this definition to real exponents $\alpha \in \mathbb{R}$ by setting

$$\phi_{\delta, \gamma}^\alpha \equiv \phi_{\alpha \delta, \gamma^\alpha}.$$

This allows us to define for any $k \in \{1, \dots, K\}$ the weighted mean

$$\bar{\phi}_k = \left(\prod_{l,m} \phi_{klm}^{|a_{klm}|} \right)^{\frac{1}{\sum_{l,m} |a_{klm}|}},$$

over all deformations ϕ_{klm} calculated by UPDATE_WAVEFORMS (the product \prod is taken w.r.t. “ \circ ”). If each of these deformations is a composition of

a translation δ_{klm} and a dilation γ_{klm} , then so is $\bar{\phi}_k$ with translation and dilation given by

$$\bar{\delta}_k = \frac{\sum_{l,m} |a_{klm}| \delta_{klm}}{\sum_{l,m} |a_{klm}|},$$

$$\bar{\gamma}_k = \left(\prod_{l,m} \gamma_{klm}^{|a_{klm}|} \right)^{\frac{1}{\sum_{l,m} |a_{klm}|}},$$

respectively. Now waveform \mathbf{d}_k can be centered by setting

$$\mathbf{d}_k \leftarrow \bar{\phi}_k(\mathbf{d}_k). \quad (4.10)$$

We summarize the centering and normalization of the waveforms in the procedure WAVEFORM_CENTERING.

```

1: procedure WAVEFORM_CENTERING( $\{a_{klm}\}, \{\phi_{klm}\}, \{\mathbf{d}_k\}$ )
2:   for  $k = 1$  to  $K$  do
3:     Calculate centering transformation  $\bar{\phi}_k$ , set
4:      $\mathbf{d}_k \leftarrow \bar{\phi}_k(\mathbf{d}_k)$ ,
5:      $\mathbf{d}_k \leftarrow \frac{\mathbf{d}_k}{\|\mathbf{d}_k\|_2}$ .
6:   end for
7: end procedure, return  $\{\mathbf{d}_k\}$ 

```

Note that normalization (4.9) and centering (4.10) obviously alter the data representation $\{a_{klm}, \phi_{klm}, \mathbf{d}_k\}$. This has to be compensated by adjusting the coefficients and deformations to

$$\phi_{klm} \leftarrow \phi_{klm} \circ \bar{\phi}_k^{-1}, \quad \forall k, l, m, \quad (4.11)$$

$$a_{klm} \leftarrow \|\mathbf{d}_k\|_2 \cdot a_{klm}, \quad \forall k, l, m, \quad (4.12)$$

where $\|\mathbf{d}_k\|_2$ refers to the norm of \mathbf{d}_k before normalization (4.9). In this new representation the mean deformation $\bar{\phi}_k$ for each \mathbf{d}_k now equals the identity, which reduces the amount of large deformations across the ϕ_{klm} . This point will become important when considering discrete signals on a finite domain in chapters 5 and 6, where large deformations cause unwanted boundary effects.

Note that the adjustments (4.11) and (4.12) need not actually be performed explicitly as coefficients and deformations will be recalculated in UPDATE_COEFFICIENTS in the following iteration of Algorithm 1. This is also the reason why the stopping criterion in Algorithm 1 is placed after the coefficient updates.

4.5 Relation to previous methods

The proposed general model (4.2) includes many of the models presented in Chapter 3 as special cases. The generic Algorithm 1 also combines many

elements of previous methods, notably sparse coding and the alternating updates found in Woody’s algorithm as well as in dictionary learning.

Thanks to the iterative structure of the algorithm, other features of the previously presented techniques may easily be incorporated into AWL at any point in the algorithm. For instance, orthogonality or independence across waveforms may be enforced after each waveform update, using PCA or ICA, respectively.

When considering only translations as transformations on the component waveforms \mathbf{d}_k , model (4.2) is very similar to dVCA (cf. Section 3.5.1). However, model (4.2) is more general as it includes the possibility of multiple occurrences of a waveform \mathbf{d}_k across one signal. This will be useful to derive a non-epoched version of the algorithm in Chapter 6, which processes long signals with repeating waveforms. In addition, there is an important difference between the implementations of dVCA and AWL: while the former updates amplitudes and latencies in separate steps, AWL uses sparse coding techniques on all shifted waveforms to calculate amplitudes and latencies simultaneously.

4.6 Hyperparameters and hierarchical AWL

The number K of waveforms to be learned is often not known *a priori*. We therefore suggest a hierarchical learning approach which is summarized in Algorithm 2. This algorithm starts by learning a single waveform representation with AWL (Algorithm 1), which is stored in R_1 . Next, a second waveform is initialized and a new representation is learned, starting with these two waveforms. This process is repeated until a maximal number K_{max} of waveforms is reached. The output of the algorithm is a set of waveform representations R_K of increasing sizes K . These representations can then be compared *a posteriori*. Alternatively, the algorithm may be stopped at some number K if the residual signal does not contain any more relevant structures.

Algorithm 2 Hierarchical AWL

Input: $\{\mathbf{x}_m\}_{m=1}^M$, $\{\phi_p\}_{p=-P}^P$, $K_{max} \in \mathbb{N}$, $\lambda \in \mathbb{R}_0^+$, constraint set C .

- 1: **for** $K = 1$ to K_{max} **do**
- 2: Initialize \mathbf{d}_K and add to set $\{\mathbf{d}_k\}_{k=1}^{K-1}$.
- 3: $(\{a_{klm}\}, \{\phi_{klm}\}, \{\mathbf{d}_k\}) \leftarrow \text{AWL}(\{\mathbf{x}_m\}, \{\mathbf{d}_k\}_{k=1}^K, \{\phi_p\}, \lambda, C)$.
- 4: Save representation: $R_K \leftarrow (\{a_{klm}\}, \{\phi_{klm}\}, \{\mathbf{d}_k\})$.
- 5: **end for**

Output: $R_1, \dots, R_{K_{max}}$.

We found that the optimal number of waveforms to be learned is strongly dependent on the application and intended interpretation, as well as on

the chosen set of deformations Φ which influences the descriptive power of each waveform. Therefore, it is difficult and may even be misleading to give a general stopping criterion. Representations for different K may even provide different and even complementary information. For our synthetic and real data experiments, choosing K much larger than 5, however, did not provide any additional insight. Hence, comparing the representations for each $K = 1, \dots, 5$ remains a feasible task.

Nevertheless, to evaluate the quality of the representation, it may be instructive to calculate the residual variance of the data representation for each K , especially when an estimate of the noise variance is available for comparison. An overview of more sophisticated model selection procedures based on information criteria can be found in [Stoica and Selen \(2004\)](#).

The other hyperparameters of the algorithm are related to the sparsity of the representation, that is, the regularization parameter $\lambda > 0$ and the exclusive constraint set \mathcal{C} . We will illustrate appropriate choices in the following chapters, in the context of more concrete applications.

Chapter 5

Epoched AWL

In this chapter, we specialize AWL to epoched (i.e., segmented) datasets. We provide an efficient implementation E-AWL based on LARS. The algorithm is evaluated on synthetic and real data, and the results are compared to those produced by PCA and ICA.

Contents

5.1	Introduction	68
5.2	Model specifications	68
5.2.1	Explicit representation	69
5.3	Algorithm E-AWL	70
5.3.1	Coefficient updates	70
5.3.2	Waveform updates	71
5.3.3	Waveform centering	71
5.3.4	Discrete shifts and boundary issues	72
5.4	Evaluation on simulated data	73
5.4.1	Influence of varying amplitudes	76
5.4.2	Influence of varying latencies	76
5.4.3	Robustness to noise	78
5.4.4	Qualitative comparison	78
5.5	Evaluation on LFP recording	79
5.5.1	Preprocessing and epoching	80
5.5.2	Hierarchical representations with epoched AWL	80
5.5.3	Comparison of PCA, ICA, and AWL	82

5.1 Introduction

The focus of this chapter is on the processing of epoched datasets (cf. Section 2.5.1) with AWL. For this purpose, we will specialize the general AWL model introduced in (4.2) and give details for the implementation. The resulting epoched AWL algorithm (E-AWL) is evaluated on simulations and real data, and its performance is compared to that of PCA and ICA.

Throughout this chapter, the class of transformations on waveforms is limited to translations. The case of dilations (or their combination with translations) works similarly and will be treated in a non-epoched setting in the next chapter.

More general transformations may also be used, but the formulas which assume orthogonal transformations need to be reformulated, according to the general setting from Chapter 4.

5.2 Model specifications

We now consider the case of a set of epoched trials $\{\mathbf{x}_m\}$ from electrophysiological recordings, that is, short temporal signals containing event-related potentials (ERP, cf. Section 2.5.1) or spontaneously repeating activity (SRA, cf. Section 2.5.2). Assuming the ERP or SRA components to occur with a fixed shape but at different latencies $\{\delta_{-P}, \dots, \delta_P\}$ across epochs, the general model (4.2) now writes

$$\mathbf{x}_m = \sum_{k=1}^K \sum_{p=-P}^P a_{kpm} \mathbf{d}_k(\cdot - \delta_p) + \epsilon_m, \quad m = 1, \dots, M. \quad (5.1)$$

Note that in order to take into account all latencies in a given range with high resolution, P may be very large. As described in the previous chapter, the sparse coefficients a_{kpm} therefore fulfill the role of selecting the relevant latencies. In this model, we assume each ERP or SRA component \mathbf{d}_k to occur at most once per epoch \mathbf{x}_m . This leads to maximal sparsity *within* each \mathbf{x}_m , given by

$$\forall k, m : \|(a_{k(-P)m}, \dots, a_{kPm})\|_0 \leq 1,$$

where $\|\cdot\|_0$ denotes the l_0 -norm (cf. Section 3.6.1). This constraint is a special case of the exclusivity constraint (4.4) introduced in the previous chapter. In addition, we usually assume waveforms to occur with the same sign across epochs \mathbf{x}_m , leading to the non-negativity constraint on coefficients (4.5).

If the events of interest are known to occur rarely *across* epochs \mathbf{x}_m , general sparsity (cf. Section 4.3.2) can be imposed additionally. However, this introduces a bias towards rare events, which discourages the detection

of systematically occurring events of lower amplitudes. We therefore do not impose general sparsity in the present setting.

The waveforms \mathbf{d}_k and coefficients a_{kpm} in (5.1) are now obtained by solving

$$\min_{a_{kpm}, \mathbf{d}_k} \sum_{m=1}^M \left\| \mathbf{x}_m - \sum_{k=1}^K \sum_{p=-P}^P a_{kpm} \mathbf{d}_k (\cdot - \delta_p) \right\|_2^2 \quad (5.2)$$

$$\text{s.t. } \forall k, m : \quad \|(a_{k(-P)m}, \dots, a_{kPm})\|_0 \leq 1, \quad (5.3)$$

$$\forall k, p, m : \quad a_{kpm} \geq 0. \quad (5.4)$$

This problem can be tackled by making the appropriate specializations to the generic AWL algorithm (Algorithm 1), presented in the previous chapter. The resulting method is E-AWL, which is summarized in Algorithm 3. As before, the algorithm alternates between waveform and coefficient updates, which will be explained in detail in the following section.

5.2.1 Explicit representation

As mentioned in Chapter 4, the sparse representation $\{a_{kpm}, \delta_p, \mathbf{d}_k\}$ facilitates the implementation of sparse coding techniques for the coefficient updates. However, during the rest of the algorithm, we will change to the more explicit representation, denoted by

$$(\{a_{km}\}, \{\delta_{km}\}) \leftarrow \text{SWITCH_REPRESENTATION}(\{a_{kpm}\}, \{\delta_p\}).$$

Now, a_{km} and δ_{km} directly describe the coefficients and latencies of the actual waveform occurrences. Note that we could drop the index p due to the exclusivity constraint (5.3).

Algorithm 3 E-AWL

Input: $\{\mathbf{x}_1, \dots, \mathbf{x}_M\}, \{\mathbf{d}_1, \dots, \mathbf{d}_K\}, \{\delta_{-P}, \dots, \delta_P\}$.

```

1: loop
2:   Set up  $\mathbf{D} = \{\mathbf{d}_k^p\}$  with  $\mathbf{d}_k^p = \mathbf{d}_k (\cdot - \delta_p)$ .
3:    $\{a_{kpm}\} \leftarrow \text{COEFFICIENT\_UPDATES}(\{\mathbf{x}_m\}, \{\mathbf{d}_k^p\})$ 
4:    $(\{a_{km}\}, \{\delta_{km}\}) \leftarrow \text{SWITCH\_REPRESENTATION}(\{a_{kpm}\}, \{\delta_p\})$ 
5:   if stopping criterion reached then
6:     break
7:   end if
8:    $\{\mathbf{d}_k\} \leftarrow \text{WAVEFORM\_UPDATES}(\{\mathbf{x}_m\}, \{a_{km}\}, \{\delta_{km}\}, \{\mathbf{d}_k\})$ 
9:    $\{\mathbf{d}_k\} \leftarrow \text{WAVEFORM\_CENTERING}(\{a_{km}\}, \{\delta_{km}\}, \{\mathbf{d}_k\})$ 
10: end loop

```

Output: $\{a_{km}\}, \{\delta_{km}\}, \{\mathbf{d}_k\}$.

5.3 Algorithm E-AWL

In order to learn the waveform representation as *blindly* as possible, we generally initialize the waveforms \mathbf{d}_k in Algorithm 3 with white Gaussian noise (cf. Section 4.4.1). Then, coefficients and waveforms are iteratively updated in the procedures COEFFICIENT_UPDATES, WAVEFORM_UPDATES, and WAVEFORM_CENTERING, respectively, until a stopping criterion is reached (e.g., numerical convergence or maximal number of iterations).

5.3.1 Coefficient updates

For the coefficient updates, we have to minimize (5.2-5.4) with respect to the a_{kpm} while leaving the waveforms \mathbf{d}_k fixed. For this purpose, we first define the dictionary $\mathbf{D} = \{\mathbf{d}_k^p\}$ of all translated waveforms $\mathbf{d}_k^p = \mathbf{d}_k(\cdot - \delta_p)$. Now, the updates are described in the procedure COEFFICIENT_UPDATES.

```

1: procedure COEFFICIENT_UPDATES( $\{\mathbf{x}_m\}, \mathbf{D}$ )
2:   for  $m = 1$  to  $M$  do
3:     Solve using LARS:
4:      $\{a_{kpm}\} \leftarrow \underset{\{a'_{kpm}\}}{\operatorname{argmin}} \left\| \mathbf{x}_m - \sum_{k=1}^K \sum_{p=-P}^P a'_{kpm} \mathbf{d}_k^p \right\|_2^2$ ,
5:     s.t.  $\forall k: \|(a'_{k(-P)m}, \dots, a'_{kPm})\|_0 \leq 1$ ,
6:          $\forall k, p: a'_{kpm} \geq 0$ .
7:   end for
8: end procedure, return  $\{a_{kpm}\}$ 

```

Note that while line 4 by itself is an ordinary least squares problem, its solution is difficult due to the constraints in lines 5 and 6. Although we do not have any l_1 -regularization term in this problem as in Section 4.4.2, the LARS algorithm proves very useful in this situation. This is due to two reasons: First, when we follow the path of the LARS until the regularization parameter λ is zero (cf. Appendix A), LARS provides an exact solution to the unregularized least squares problem. Second, due to its stepwise activation and deactivation of coefficients, it allows to easily ensure the constraints in lines 5 and 6 in every step (cf. Section 4.4.2).

Note that in the actual implementation, we do not need to calculate the large dictionary $\mathbf{D} = \{\mathbf{d}_k^p\}$. In fact, the LARS algorithm only requires the correlations $\mathbf{D}^t \mathbf{x}$ as well as the covariance matrix $\mathbf{D}^t \mathbf{D}$. Both can be calculated efficiently through the fast Fourier transform (FFT) by using the shift-invariance of \mathbf{D} (cf. Appendix A).

5.3.2 Waveform updates

The waveforms are updated sequentially as described in Section 4.4.3. Using the explicit representation (cf. Section 5.2.1), the minimization problem for $k = 1, \dots, K$ reads

$$\min_{\mathbf{d}_k} \sum_{m=1}^M \left\| \mathbf{x}_m - \sum_{k'=1}^K a_{k'm} \mathbf{d}_{k'} (\cdot - \delta_{k'm}) \right\|_2^2.$$

Due to the orthogonality of the translation operator, the update formula (4.7,4.8) now simplifies to

$$\mathbf{d}_k = \left(\frac{1}{\sum_{m=1}^M a_{km}^2} \right) \sum_{m=1}^M a_{km} \mathbf{r}_{km} (\cdot + \delta_{km}), \quad (5.5)$$

where again

$$\mathbf{r}_{km} = \mathbf{x}_m - \sum_{k' \neq k} a_{k'm} \mathbf{d}_{k'} (\cdot - \delta_{k'm})$$

denotes the residual that results by subtracting all but waveform \mathbf{d}_k 's contribution from the signal \mathbf{x}_m . Hence, \mathbf{d}_k is obtained as a weighted average over the realigned residuals \mathbf{r}_{km} (note the inverse or adjoint translation $\mathbf{r}_{km} \mapsto \mathbf{r}_{km}(\cdot + \delta_{km})$). The scaling factor $\left(\frac{1}{\sum_m a_{km}^2} \right)$ may be omitted at this point, due to normalization in the following step. The updates are summarized in the procedure WAVEFORM_UPDATES.

```

1: procedure WAVEFORM_UPDATES( $\{\mathbf{x}_m\}, \{a_{km}\}, \{\delta_{km}\}, \{\mathbf{d}_k\}$ )
2:   for  $k = 1$  to  $K$  do
3:      $\mathbf{d}_k \leftarrow \sum_{m=1}^M a_{km} \mathbf{r}_{km} (\cdot + \delta_{km})$ , where
4:      $\mathbf{r}_{km} = \mathbf{x}_m - \sum_{k' \neq k} a_{k'm} \mathbf{d}_{k'} (\cdot - \delta_{k'm})$ .
5:   end for
6: end procedure, return  $\{\mathbf{d}_k\}$ 

```

5.3.3 Waveform centering

Waveform centering can become extremely important in the case of short epochs \mathbf{x}_m and relatively large latency jitter. Consider a randomly initialized waveform \mathbf{d}_k to coincidentally contain some structure in its left half which correlates well with a signal component located in the right half of the majority of epochs $\{\mathbf{x}_m\}$. Then, waveform \mathbf{d}_k has to be shifted mostly to the right, resulting in calculated latencies $\{\delta_{km}\}_m$ with positive average. On the one hand, this creates unnecessary boundary errors (cf. following subsection). On the other hand, if in some epochs $\{\mathbf{x}_m\}$ the component of interest

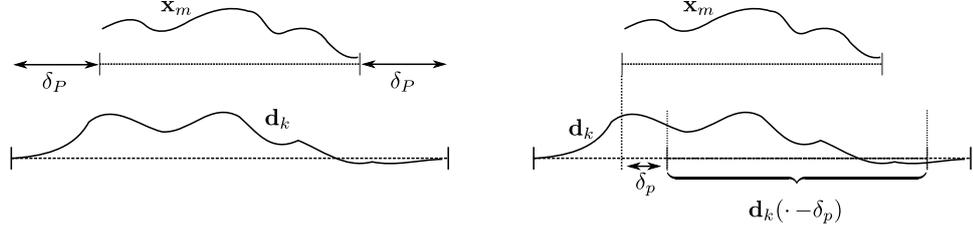


Figure 5.1: Implementation of waveform shifts on a discrete domain. **Left:** For the waveforms \mathbf{d}_k , the domain of \mathbf{x}_m is extended to the left and right about the length of the maximal shift δ_P . **Right:** Now, the shift operations $\mathbf{d}_k \mapsto \mathbf{d}_k(\cdot - \delta_p)$ are implemented by selecting a window in \mathbf{d}_k at latency δ_p , which has the length of each epoch \mathbf{x}_m . The opposite shifts $\mathbf{r}_{km} \mapsto \mathbf{r}_{km}(\cdot + \delta_{km})$ used in the waveform updates (cf. Section 5.3.2) are then implemented by embedding \mathbf{r}_{km} into the domain of \mathbf{d}_k and zero-padding everywhere else.

is located even further right than on average, it might not even be detected at all, due to bounded latencies $\{\delta_{-P}, \dots, \delta_P\}$ in the algorithm.

We thus proceed as described in Section 4.4.4 by calculating the mean latency

$$\bar{\delta}_k = \frac{\sum_m |a_{km}| \delta_{km}}{\sum_m |a_{km}|},$$

and realigning waveforms

$$\mathbf{d}_k \leftarrow \mathbf{d}_k(\cdot - \bar{\delta}_k).$$

In order to meet the normalization constraint, the waveforms are then projected onto the unit sphere. These updates are summarized in procedure WAVEFORM_CENTERING in Algorithm 3.

```

1: procedure WAVEFORM_CENTERING( $\{a_{km}\}, \{\delta_{km}\}, \{\mathbf{d}_k\}$ )
2:   for  $k = 1$  to  $K$  do
3:      $\bar{\delta}_k = \frac{\sum_m |a_{km}| \delta_{km}}{\sum_m |a_{km}|}$ ,
4:      $\mathbf{d}_k \leftarrow \mathbf{d}_k(\cdot - \bar{\delta}_k)$ ,
5:      $\mathbf{d}_k \leftarrow \frac{\mathbf{d}_k}{\|\mathbf{d}_k\|_2}$ .
6:   end for
7: end procedure, return  $\{\mathbf{d}_k\}$ 

```

5.3.4 Discrete shifts and boundary issues

In order to define the discrete set of admitted translations, we choose equispaced latencies $\{\delta_{-P}, \dots, \delta_P\}$ with $\delta_p = p \cdot dt$, as described in Section 4.2.1.1. The resolution between translations given by $dt > 0$ is taken equal to the

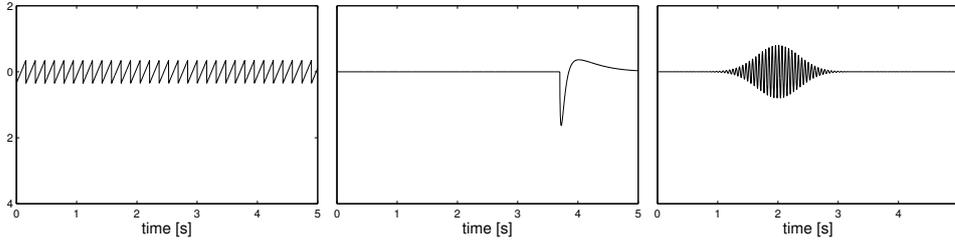


Figure 5.2: Three waveforms defined in order to generate trials according to model (5.1). Note the different shapes including ongoing and oscillatory as well as transient signals.

sampling resolution of the epochs $\{\mathbf{x}_m\}$, allowing efficient implementation via fast Fourier transform (cf. Appendix A). The choice of $P \in \mathbb{N}$ now depends on the desired maximal latency $\delta_P = P \cdot dt$, which can be based on the expected amount of waveform latency jitter across the epochs. In the presence of periodic activity which is not phase-locked to the epochs, it is useful to include shifts ranging over at least one full period of this activity.

Applying translations to signals on a finite domain naturally raises the question of how to proceed with parts “shifted out” of the domain. In the case of short transient events with small support in the center of the epochs, we may use circular shifts which can be implemented efficiently. However, when learning ongoing (e.g., oscillatory) activity as in Section 5.5, this approach will lead to undesired boundary artifacts.

In order to minimize information loss caused by signals shifted out of the domain, we choose the following solution. We define the waveforms \mathbf{d}_k on a slightly larger domain than the epochs \mathbf{x}_m , extended by the length of the maximal shift δ_P at each side. In this case, the shift operation $\mathbf{d}_k(\cdot - \delta_P)$ results in the selection of a window at latency δ_P in \mathbf{d}_k . This is illustrated in Figure 6.1. In turn, the opposite shifts $\mathbf{r}_{km} \mapsto \mathbf{r}_{km}(\cdot + \delta_{km})$ used for the waveform updates (5.5) are performed by zero-padding, such that no information is lost in this step.

5.4 Evaluation on simulated data

We first used simulated data in order to evaluate the capability of epoched AWL to identify multiple waveforms across a set of signals in the presence of waveform variability and noise. The levels of amplitude and latency variability as well as the signal-to-noise ratios (SNRs) were varied across different experiments, which is described in detail in the following subsections. The results are compared to those obtained by PCA and ICA (see sections 3.4.1 and 3.4.2).

We started by defining $K = 3$ waveforms of 2500 discrete samples, rep-

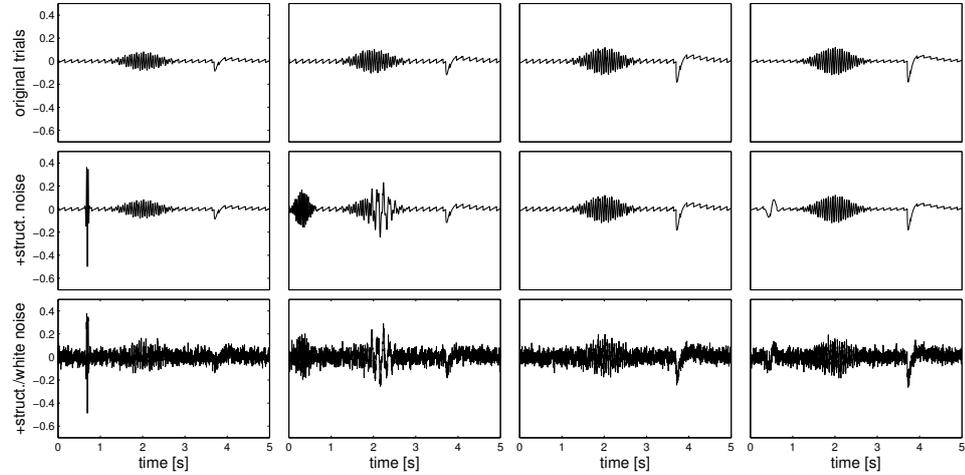


Figure 5.3: Four randomly chosen trials, generated according to model (5.1) from waveforms shown in Figure 5.2. First row: noiseless trials. Second row: trials plus structured noise. Third row: trials plus structured and white Gaussian noise.

representing temporal signals on a 5 second window with 500 Hz sampling rate. They are shown in Figure 5.2. Note that both transient and oscillatory waveforms are included. These waveforms were then used to create 200 trials according to model (5.1) with constraints (5.3) and (5.4). Amplitudes and latencies were drawn from different distributions, and two types of noise were used, Gaussian and structured noise. The exact distributions are specified in the following subsections for the individual experiments. Figure 5.3 illustrates some generated trials with and without noise.

The generated signals were then processed with PCA, ICA, and AWL in order to recover waveforms, amplitudes, and in the case of AWL also latencies. In order to provide for a comparison with the original waveforms, we considered the number $K = 3$ to be known *a priori*.

For the calculation through PCA, the waveforms were defined as the first three principal components, and the waveform amplitudes were given by the coefficient matrix of the PCA decomposition (cf. Section 3.4.1).

For the calculation with ICA (cf. Section 3.4.2), we used the Matlab software package FastICA¹, which implements the method described in Hyvarinen (1999). As discussed in Section 3.4.2, the number of independent components calculated by ICA normally equals the number of input components, while in the present experiment we only needed to obtain $K = 3$ waveforms from $M = 200$ trials. For this reason, we used an option provided in the FastICA algorithm, which allows to extract only a given number of output components. However, this approach, denoted later as ICA200, led

¹<http://research.ics.aalto.fi/ica/fastica/>

to very poor results. This comes from the fact that there is no hierarchy between independent components (cf. Section 3.4.2) such as between principal components (cf. Section 3.4.1). Hence, the components extracted by ICA200 were somewhat arbitrary and corresponded to noise rather than to the actual waveforms. An alternative procedure consists in first reducing the input dimension. This can be achieved, for instance, by performing a PCA and using only the first principal components as input for the ICA. Hence, besides the ICA performed on the full data (ICA200), we also used ICA with previous dimensionality reductions to 10 input channels (ICA10) and a maximal reduction to 3 input channels (ICA3).

For AWL, we used Algorithm 3 with translations $\{\delta_{-P}, \dots, \delta_P\}$ ranging from -0.1 to 0.1 seconds with a resolution equal to the sampling resolution of the epochs \mathbf{x}_m (cf. Section 5.3.4). After reconstruction, two performance measures were calculated for every method.

First, we evaluated each method's ability to capture the variance of the original noiseless data $\bar{\mathbf{x}}_m = \mathbf{x}_m - \epsilon_m$. For PCA and ICA, the corresponding estimates were given by

$$\hat{\mathbf{x}}_m = \sum_{k=1}^3 \hat{a}_{km} \hat{\mathbf{d}}_k,$$

with \hat{a}_{km} and $\hat{\mathbf{d}}_k$ the calculated coefficients and waveforms, respectively. Note that this error is the same for PCA and ICA3, since ICA3 was performed on the three components calculated by PCA. For AWL, we had

$$\hat{\mathbf{x}}_m = \sum_{k=1}^3 \hat{a}_{km} \hat{\mathbf{d}}_k(\cdot - \hat{\delta}_{km}),$$

with estimated latencies $\hat{\delta}_{km}$. The relative residual error was then obtained as

$$\epsilon_{\mathbf{x}} = \frac{\sum_{m=1}^M \|\bar{\mathbf{x}}_m - \hat{\mathbf{x}}_m\|_2}{\sum_{m=1}^M \|\bar{\mathbf{x}}_m\|_2}$$

for all three methods.

As a second measure, we investigated the similarity of the original and the calculated waveforms. Due to indeterminacies in the order and relative latencies of the waveforms, this required prior matching of calculated waveforms to the most similar original ones, followed by their realignment. The normalized waveform error is given by¹

$$\epsilon_{\mathbf{d}} = \frac{1}{\sqrt{2}K} \sum_{k=1}^K \|(\mathbf{d}_k - \hat{\mathbf{d}}_k)\|_2.$$

¹The factor $\frac{1}{\sqrt{2}}$ is used to normalize $\epsilon_{\mathbf{d}}$ for the “worst case”. That is, assuming \mathbf{d}_k and $\hat{\mathbf{d}}_k$ to be random vectors, normalized and with zero mean, then, if \mathbf{d}_k and $\hat{\mathbf{d}}_k$ are independent, the expected value of $\|(\mathbf{d}_k - \hat{\mathbf{d}}_k)\|_2$ is $\sqrt{2}$.

In the following subsections, we use these error measures to evaluate each method's performance for the cases of varying amplitudes and latencies of waveforms, as well as for varying signal-to-noise ratios (SNR) with respect to white Gaussian and structured noise. Finally, we show an example of the recovered waveforms by the three methods in the case of moderate variability and noise.

5.4.1 Influence of varying amplitudes

In order to investigate the effect of varying amplitudes, we first simulated trials with constant latencies of waveforms and a high SNR of 10 dB, with white Gaussian noise. Amplitudes were drawn from a Gaussian distribution with mean 1. The standard deviation (SD) σ_a was increased from 0 to 1 throughout simulations. Negative amplitudes were discarded. After the reconstruction with PCA, ICA, and AWL, the respective errors $\epsilon_{\mathbf{x}}$ and $\epsilon_{\mathbf{d}}$ were calculated; they are shown in the first row of Figure 5.4.

The plot on the left shows the reconstruction error $\epsilon_{\mathbf{x}}$ for increasing amplitude variability σ_a . The methods PCA, ICA3, and AWL all succeed in well-capturing the variance in the data for all levels of σ_a . ICA10 yielded good reconstructions for most values of σ_a . It usually succeeded in selecting the “right” independent components from the reduced 10 input channels. However, ICA200 which was performed on all 200 simulated trials, captured mostly the noise and led to completely useless results.

Concerning the correct separation of the waveforms, only ICA3 succeeds well for increasing amplitude variability σ_a (right plot) by exploiting the higher order statistical independence across waveforms. As the average amplitudes across the trials were equal for all three waveforms, PCA did not succeed in separating the waveforms. In fact, PCA orders principal components with respect to their variance, which leads to a mixing of components with similar variance (cf. Section 3.4.1). AWL improves with increasing σ_a but stays behind ICA. Again, the results for ICA200 and ICA10 were suboptimal.

5.4.2 Influence of varying latencies

Throughout the following simulations, latencies were drawn from a zero-mean Gaussian distribution with increasing standard deviation σ_δ in a range from 0 to 1 seconds (logarithmic sampling); the SNR was kept constant at 10 dB, again using white Gaussian noise. We maintained some amplitude variability ($\sigma_a = 0.3$) in order to allow PCA and ICA to separate the waveforms.

The results in row two of Figure 5.4 show that PCA and ICA3 slowly worsen in capturing the data variance with increasing σ_δ (left plot), while for AWL, ϵ_x remains relatively constant for $\sigma_\delta \leq 0.1$ seconds. For larger σ_δ ,

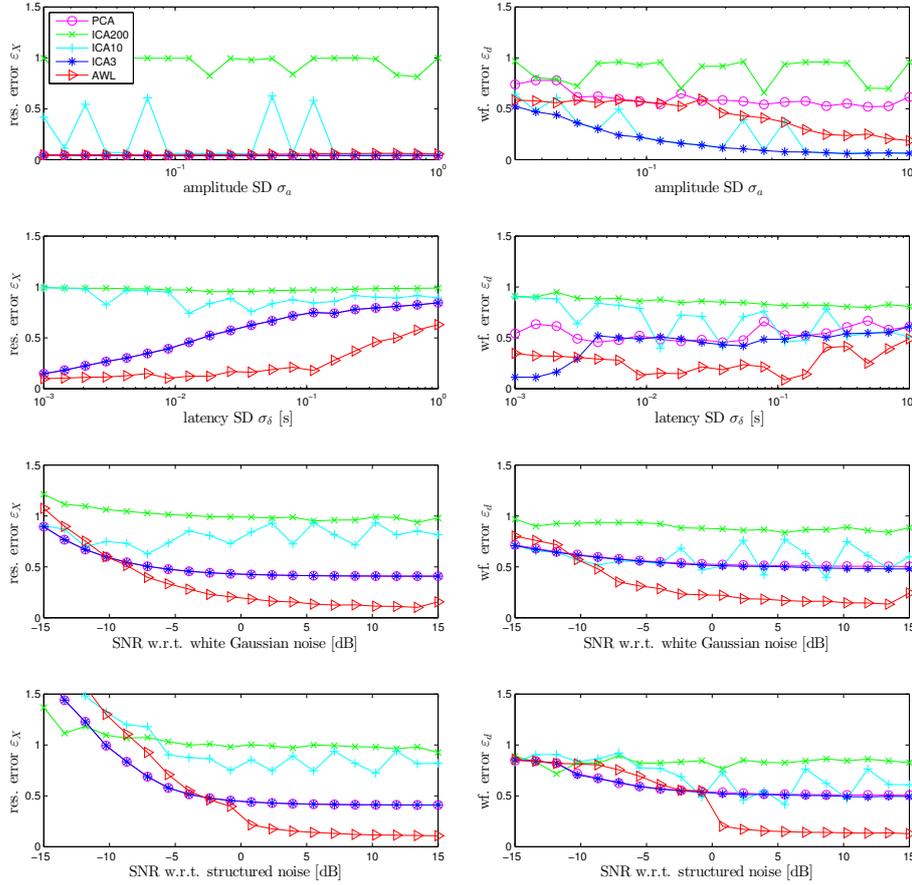


Figure 5.4: Error curves for methods PCA, ICA200, ICA10, ICA3, and epoched AWL (legend upper left) for increasing amplitude variability (first row), latency variability (second row), and increasing SNRs, for white Gaussian (third row) and structured noise (last row). The left column shows the residual errors ε_x of the reconstructed signals compared to the original (noiseless) signals. The errors ε_d of the reconstructed waveforms compared to the original ones are displayed in the right column. AWL always outperforms PCA and ICA3 in the presence of latency variability (> 0.003 s) and an SNR above -5 dB. Only in the case of constant latencies (first row) does ICA3 succeed in better separating waveforms than AWL. Note that ICA10 and ICA200 yield only suboptimal results (cf. Section 5.4.1).

the error also rises, since we translations $\{\delta_{-P}, \dots, \delta_P\}$ used in the AWL algorithm only ranged from -0.1 to 0.1 seconds.

Concerning waveform separation (right plot), ICA3's initially good performance suddenly decreases with latency variability σ_δ above 0.003 seconds. Around the same value for σ_δ , AWL's performance significantly increases. Again, ICA only yielded useful results for maximal dimensionality reduction (ICA3).

To sum up, the error curves show that while PCA and ICA do not account for the changing latencies and thus perform poorly in their presence, AWL not only compensates for them but even makes use of this variability for better waveform separation.

5.4.3 Robustness to noise

Finally, the performance with respect to two types of noise was studied: white Gaussian noise and more structured noise represented by real Gabor wavelets (cf. Section 3.6.1) with varying latencies, frequencies, durations, and amplitudes. The standard deviations of the waveforms' amplitudes and latencies were fixed to $\sigma_a = 0.3$ and $\sigma_\delta = 0.01$, respectively. The errors ϵ_x and ϵ_a for increasing SNR with respect to Gaussian and structured noise are shown in the third and fourth row of Figure 5.4, respectively.

The methods AWL, PCA, and ICA3 start to account for at least 50 per cent of the variance for an SNR between -10 and -5 dB (left column). While PCA and ICA3 do not improve significantly for higher SNRs anymore, AWL's performance steadily increases.

Concerning waveform separation (right column), the error ϵ_d of PCA and ICA3 remains above 0.7 for all SNRs. AWL starts to succeed in separating waveforms between -5 and 0 dB.

All methods showed to cope better with white Gaussian noise; in the case of structured noise, the SNR needed to be significantly higher to yield a similar performance. As before, ICA only yielded useful results when used with maximal dimensionality reduction (ICA3).

5.4.4 Qualitative comparison

For a qualitative comparison, we generated a set of trials with medium amplitude variability ($\sigma_a = 0.3$), small latency variability ($\sigma_\delta = 0.01$), and white Gaussian and structured noise, with resulting SNR of -2.67 dB. Four randomly chosen trials are displayed in Figure 5.3, the different rows show original and noisy signals (see caption).

The waveforms recovered with PCA, ICA, and AWL are shown in Figure 5.5. While PCA (first row) recovers mixtures of the three original waveforms (cf. Figure 5.2), ICA3 (fourth row) succeeds in well separating the third original waveform from the other two. However, in order to encode the

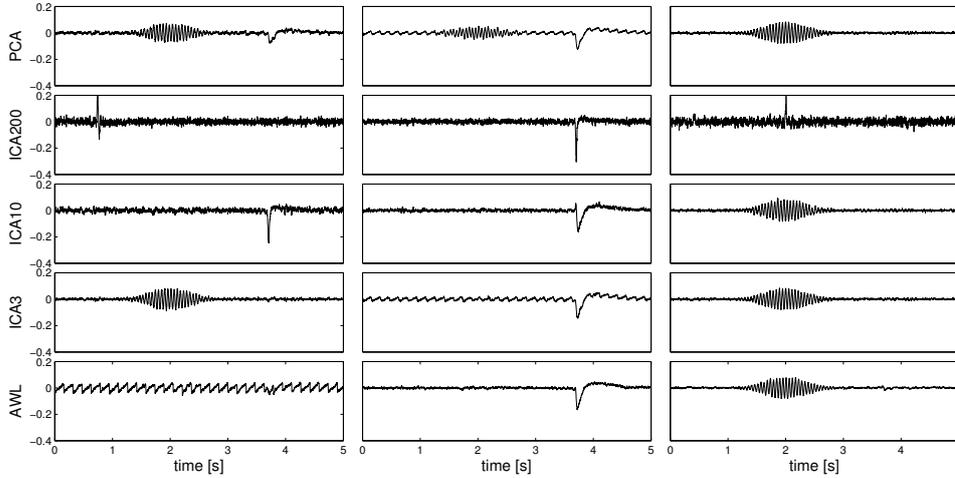


Figure 5.5: Waveforms recovered from noisy trials (Figure 5.3, last row) with PCA, ICA200, ICA10, ICA3, and AWL in the respective rows. PCA and ICA3 produce superpositions of the original waveforms (Figure 5.2). Without applying dimensionality reduction, ICA leads to useless results (ICA200). The results improve if the data variance is first compressed to ten channels (ICA10) but are best for maximal dimensionality reduction (ICA3). Only AWL separates all waveforms correctly.

different latencies across the data, it needs two differently shifted versions of this waveform (one and three). Without applying maximal dimensionality reduction before ICA, the results are suboptimal (ICA200, ICA10). Only AWL (last row) succeeds in separating all three waveforms correctly.

5.5 Evaluation on LFP recording

In this section, we demonstrate the utility of epoched AWL for processing the local field potential (LFP) dataset introduced in Section 2.7. The aim of this analysis was to obtain a compact representation of the large dataset (approximately one hour of recording), gain insight into shape and distribution of the epileptiform discharges (or spikes), and to detect other relevant signal structures. In order to apply Algorithm 3, the contiguous LFP recording first had to be segmented into epochs with well-isolated spikes, as explained in the following subsection. Note that in this section we do not process the simultaneously recorded cerebral blood flow (CBF), cf. Section 2.7. Since the CBF response is significantly slower than the LFP spikes, it cannot be well separated into epochs. Instead, the LFP-CBF relationship is assessed in a non-epoched framework in Chapter 6.

After preprocessing and epoching, waveform representations were learned hierarchically using Algorithm 3 for an increasing number of waveforms

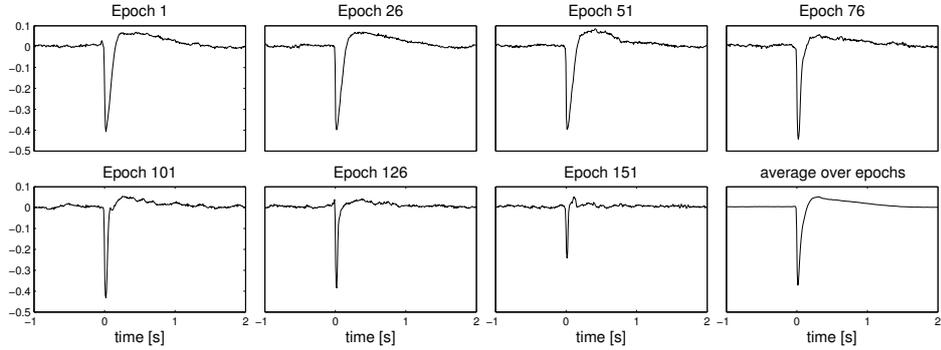


Figure 5.6: The first seven plots show sample epochs from the 169 epoched data segments. The average over all 169 epochs is plotted in the last figure. It becomes apparent that spikes are changing throughout the recording, decreasing in duration and amplitude towards the end. Note that for better visualization only 3 seconds of the 8 second long epochs are shown.

$K = 1, \dots, 5$. The results for $K = 5$ were then compared to those obtained with PCA and ICA.

5.5.1 Preprocessing and epoching

Since in this study we were mainly interested in the global shapes of the epileptiform discharges, the data was downsampled from 1250 Hz to 125 Hz. Prior to downsampling, lowpass filtering with cutoff frequency of 60 Hz was performed to avoid aliasing effects. Then 169 well-isolated spikes (at least 5 seconds peak-to-peak distance between neighboring spikes) were selected manually and segmented into 8 second time windows centered around the spikes. Epochs 1, 26, 51, 76, 101, 126, and 151 as well as the average over all 169 epochs are shown in Figure 5.6.

5.5.2 Hierarchical representations with epoched AWL

We used Algorithm 3 to learn representations with $K = 1, \dots, 5$ different waveforms. This was performed in a hierarchical manner, starting with a single waveform (cf. Section 4.6). Then a second waveform was initialized with white Gaussian noise, and Algorithm 3 was repeated on the increased set of waveforms. This incremental procedure was repeated until $K = 5$. Although the negative peaks of the spikes were already well-aligned thanks to exact manual epoching, we chose to allow translations $\{\delta_{-P}, \dots, \delta_P\}$ over the entire time window of the epochs, that is, from -4 to 4 seconds. This allowed for the detection of latent structures that were not exactly time- or phase-locked to the spikes.

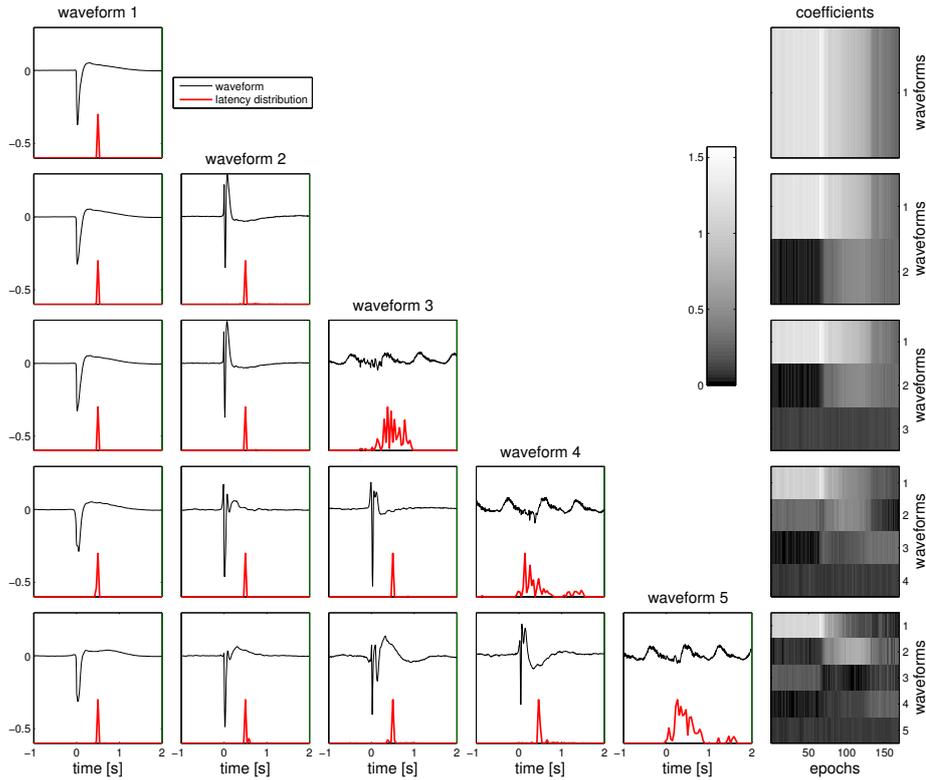


Figure 5.7: Hierarchical waveform representations obtained with AWL from 169 spike epochs (Figure 5.6). Each row corresponds to one learned set of waveforms $\{\mathbf{d}_k\}$ (note that we show only the central 3 seconds, which contain the relevant structure). The gray-value images in the last column show the coefficients $\{a_{km}\}$ of each waveform (row) across epochs (column); light colors correspond to large values. The red curve under each waveform illustrates the distribution of latencies $\{\delta_{km}\}$ used for this waveform across epochs. (Note that the red curves do not correspond strictly to the time axis; the true latency distributions were shifted 0.5 seconds to the right for the purpose of visualization, avoiding overlaps with the spikes.)

The five increasing representations are shown in the respective rows of Figure 5.7. The waveforms and their latency distributions are displayed in the first five columns, the gray values in the last column illustrate the waveforms' coefficients across epochs. The coefficients represent the waveforms' energies (or more precisely l_2 -norms).

For $K = 1$, the resulting waveform is simply a weighted average across epochs and resembles the average shown in Figure 5.6. The exact alignment of spikes across epochs is reflected by the clearly spiked latency distribution (red curve below waveform). The waveform's energy (upper right) appears to diminish towards the last epochs, with a transient increase around epoch 70.

Adding a second waveform (second row) to the learning process results in a two-component representation of the spikes. Interestingly, the second component appears to be systematically active only after the first 70 epochs. This reveals a morphological change in the spike form in the second half of epochs which could not be identified in the one component representation.

For $K = 3$, a new, oscillatory waveform appears. Its latency distribution extends quite uniformly over about 1 second, approximately corresponding to one full period. Hence, the phase of this oscillatory phenomenon seems to be uncorrelated with the spikes. Furthermore, the coefficient values indicate a constant intensity of this oscillation across epochs. Indeed, a further expert's analysis identified these oscillations as spike-unrelated artifacts, probably produced by the recording hardware.

By learning additional waveforms ($K = 4, 5$), the spike becomes further refined in different components. Interestingly, the energy of each spike component appears to be relatively constant for contiguous sets of epochs. This produces a segmentation into sets of epochs with similar spike morphology. However, while for $K = 4$ three segments can be properly distinguished (changes approximately at epochs 70 and 140), for $K = 5$ this segmentation is less clear. Further increasing K led to representations that were even harder to interpret (not shown here).

Note that each time a new waveform is added to the existing ones, the entire set of waveforms is relearned. However, until $K = 3$ the previously learned waveforms do not appear to change much when relearned. This clear waveform hierarchy may result from the non-convexity of the algorithm and its tendency to find a local minimum close to the initialization. However, the spike components clearly change for $K = 4$, suggesting the algorithm's capability to refine representations when necessary.

5.5.3 Comparison of PCA, ICA, and AWL

The result for $K = 5$ waveforms learned with AWL was compared with those produced by PCA and ICA. Here, we only show the results for ICA with maximal reduction of the data to five input components. As in the synthetic

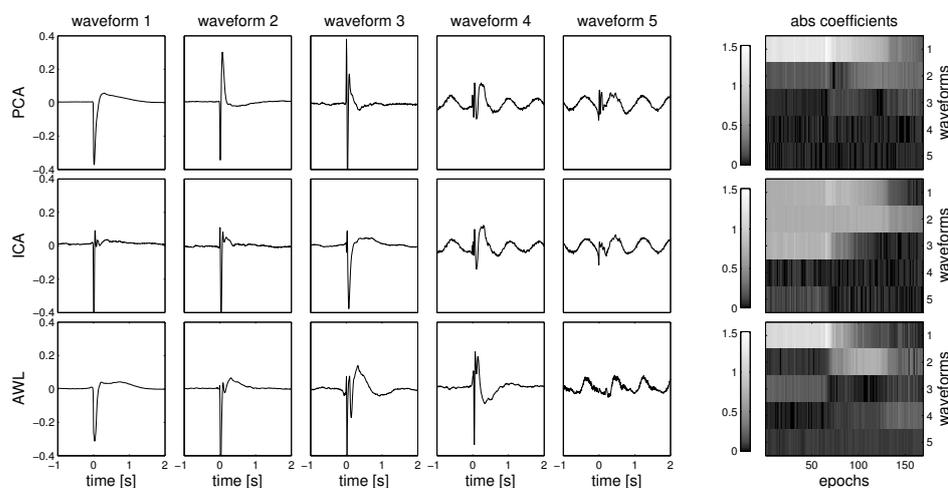


Figure 5.8: Representations with five waveforms are learned with PCA, ICA, and E-AWL, see respective rows. The last column shows the absolute coefficients of each waveform (row) used across the epochs (columns). Light colors correspond to large absolute values. All methods produce spike components and oscillations, however, only E-AWL clearly separates them and encodes the oscillations in a single waveform.

experiment before, ICA did not yield useful results without this reduction.

Figure 5.8 shows the learned waveforms and coefficients for PCA, ICA, and AWL in the respective rows. All three methods produce spike components as well as oscillatory waveforms. However, only AWL succeeds in clearly separating the oscillatory waveform from the spikes by exploiting its changing phase relative to the spikes. Since PCA and ICA cannot adapt the phase across epochs, they encode the oscillations as linear combinations of two periodic functions in phase quadrature (last two waveforms). However, this only allows us to explain the main frequency of the true oscillations, which is why these waveforms look very sinusoidal compared to the oscillations learned with AWL. In fact, when more components are extracted, PCA and ICA also produce harmonics of this main frequency (a sinusoidal function with double frequency first appears for eight extracted components).

In conclusion, AWL’s capability of capturing and compensating for latency variability showed clear benefits when compared to PCA and ICA. Yet, in this dataset, the spikes were exactly aligned and only the oscillatory artefact varied in phase. The advantages of latency compensation will become even more apparent when signals of interest are not as well aligned.

Chapter 6

Contiguous AWL

In this chapter, we specialize AWL to process non-epoched signals. We provide an efficient implementation C-AWL based on matching pursuit. We then present two approaches to spike processing, MC-Spike and AD-Spike, based on C-AWL. Both are evaluated on real data and compared to template matching in terms of detection accuracy. Finally, the spike representations are used to explore the relationship to the hemodynamics.

Contents

6.1	Introduction	86
6.2	Model specifications	86
6.3	Algorithm C-AWL	88
6.3.1	Coefficient updates	88
6.3.2	Waveform updates	89
6.3.3	Waveform centering	90
6.4	C-AWL for spike learning	91
6.4.1	Multi-class spike learning (MC-Spike)	92
6.4.2	Adaptive duration spike learning (AD-Spike)	100
6.4.3	Comparing detection performance	102
6.5	Exploring neuro-vascular coupling	105

6.1 Introduction

The method presented in the last chapter was designed to process a set of neuroelectric experimental trials. However, these trials are often the product of segmenting a contiguous recording into short epochs. When analyzing event-related potentials (ERPs), epoching is a standard procedure and can be automatically performed by locking the epochs to the stimuli. However, some experimental setups require inter-stimulus intervals to be relatively short, causing ERPs to overlap. In this case, epochs necessarily contain combinations of different responses, distorting the true ERP shape.

Moreover, in the case of spontaneously repeating activity (SRA), the occurrences of the waveforms of interest are not known *a priori*. Epoching then has to be performed either manually, which is time-consuming and only feasible for small datasets, or by using some automatic detection device with the risk of missing relevant events.

In this chapter, we present an alternative to the standard epoching approach by processing the contiguous dataset as a whole. This can be done by making the appropriate specifications in the generic Algorithm 1. The result, contiguous AWL (C-AWL), is a method that combines detection, shape learning, and parametrization of repeating structures across the dataset.

After discussing the implementation of C-AWL, we will make some additional specializations, allowing to use it as a spike processing tool. Two versions, a multi-class spike approach (MC-Spike) and a model with a single, adaptive spike (AD-Spike) are applied to the LFP data from Section 2.7. The learned representations are then used to reveal relationships between the LFP spikes and the simultaneously recorded cerebral blood flow.

6.2 Model specifications

Throughout this chapter, we consider a single long signal \mathbf{x} that contains repetitions of waveforms \mathbf{d}_k subject to deformations at different temporal locations. We will only consider linear changes of waveform duration (i.e., dilations), but the framework can be used for more general deformations (cf. Chapter 4). The different temporal locations δ_p are given by translations of the \mathbf{d}_k . This setting is a specific case of the general model (4.2), which now reads

$$\mathbf{x} = \sum_{k=1}^K \sum_{p=1}^P \sum_{q=-Q}^Q a_{kpq} \frac{1}{\sqrt{\gamma_q}} \mathbf{d}_k \left(\frac{1}{\gamma_q} (\cdot - \delta_p) \right) + \epsilon. \quad (6.1)$$

Note that we assume \mathbf{x} to be long and waveforms \mathbf{d}_k of significantly shorter duration. The translations $\mathbf{d}_k \mapsto \mathbf{d}_k(\cdot - \delta_p)$ are thus implemented by embedding \mathbf{d}_k into a time window in the domain of \mathbf{x} starting at δ_p , while the remaining signal is zero-padded; this is illustrated in Figure 6.1. The set $\{\delta_1, \dots, \delta_P\}$ thus includes $P = N + n - 1$ different latencies, where n

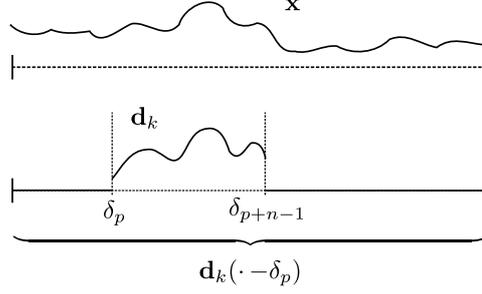


Figure 6.1: Illustration of the shift operation $\mathbf{d}_k \mapsto \mathbf{d}_k(\cdot - \delta_p)$: The waveform \mathbf{d}_k with n time samples is placed into the domain of \mathbf{x} , between time points δ_p and δ_{p+n-1} . The remaining signal is set to zero. The adjoint operation $\mathbf{x} \mapsto \mathbf{x}(\cdot + \delta_p) \cdot \mathbb{1}_{\text{supp}(\mathbf{d}_k)}$, with $\mathbb{1}_{\text{supp}(\mathbf{d}_k)}$ the indicator function for the support of \mathbf{d}_k , is then defined as selecting the window starting at δ_p in the signal \mathbf{x} .

and N denote the number of samples of \mathbf{d}_k and \mathbf{x} , respectively. The set $\{\gamma_{-Q}, \dots, \gamma_Q\}$ contains the logarithmically spaced dilation factors $\gamma_q = \beta^q$, where $\beta > 1$ controls the resolution between dilations and $\frac{\gamma_Q}{\gamma_{-Q}} = \beta^{2Q}$ is the maximal relative “stretch” (cf. Section 4.2.1.2).

If \mathbf{x} has many sample points, the set of coefficients $\{a_{kpq}\}$ will also be very large, such that sparsity is required for interpretable results. We therefore impose general sparsity, as discussed in Section 4.3.2. In addition, we implement the exclusivity constraint (4.4) by enforcing a minimal distance $\Delta > 0$ between two waveform occurrences. This leads to the minimization problem

$$\min_{\{a_{kpq}, \mathbf{d}_k\}} \left(\left\| \mathbf{x} - \sum_{k=1}^K \sum_{p=1}^P \sum_{q=-Q}^Q a_{kpq} \frac{1}{\sqrt{\gamma_q}} \mathbf{d}_k \left(\frac{1}{\gamma_q}(\cdot - \delta_p) \right) \right\|_2^2 + \lambda \sum_{k=1}^K \sum_{p=1}^P \sum_{q=-Q}^Q |a_{kpq}| \right), \quad (6.2)$$

$$\text{s.t. } a_{kpq} \neq 0 \Rightarrow (a_{k'p'q'} = 0 \text{ for all } k', q', p' \neq p : |\delta_p - \delta_{p'}| < \Delta). \quad (6.3)$$

Additionally, non-negativity (4.5) can be imposed if waveforms are supposed to respect the sign across the occurrences.

Note that while constraint (6.3) prohibits very close waveform occurrences, overlaps are still possible if Δ is smaller than the time window of the \mathbf{d}_k . Besides the difficulty in interpreting many close waveform occurrences, there is another practical reason to the minimal distance constraint: The waveforms of interest will often be smooth with most frequency content below sampling rate. Hence, if a waveform \mathbf{d}_k strongly correlates with the data at some latency δ_p , it will also show a strong correlation close to δ_p . This could lead to false multiple detections of \mathbf{d}_k in a short time interval.

The constraint (6.3) imposes the minimal distance also for different waveforms $\mathbf{d}_k, \mathbf{d}_{k'}$, which can be relaxed in cases where the waveforms have suffi-

ciently different shapes. However, in the following applications, we will learn very similar waveforms, hence the more restrictive formulation.

6.3 Algorithm C-AWL

We can solve the minimization problem (6.2, 6.3) by making the appropriate specifications to the generic Algorithm 1, which is summarized in Algorithm 4. Implementations of the procedures COEFFICIENT_UPDATES, WAVEFORM_UPDATES, and WAVEFORM_CENTERING will be given in the following subsections.

Algorithm 4 C-AWL

Input: \mathbf{x} , $\{\mathbf{d}_k\}_{k=1}^K$, $Q \in \mathbb{N}$, $\beta > 1$, $\lambda \in \mathbb{R}_0^+$, $\Delta > 0$.
 1: Calculate dilations $\{\gamma_{-Q}, \dots, \gamma_Q\}$ as $\gamma_q = \beta^q$.
 2: **loop**
 3: Set up $\mathbf{D} = \{\mathbf{d}_k^{pq}\}$ with $\mathbf{d}_k^{pq} = \frac{1}{\sqrt{\gamma_q}} \mathbf{d}_k \left(\frac{1}{\gamma_q} (\cdot - \delta_p) \right)$.
 4: $\{a_{kpq}\} \leftarrow \text{COEFFICIENT_UPDATES}(\mathbf{x}, \mathbf{D}, \lambda, \Delta)$
 5: $(\{a_{kl}\}, \{\delta_{kl}\}, \{\gamma_{kl}\}) \leftarrow \text{SWITCH_REPRESENTATION}(\{a_{kpq}\}, \{\delta_p\}, \{\gamma_q\})$
 6: **if** stopping criterion reached **then**
 7: **break**
 8: **end if**
 9: $\{\mathbf{d}_k\} \leftarrow \text{WAVEFORM_UPDATES}(\mathbf{x}, \{a_{kl}\}, \{\delta_{kl}\}, \{\gamma_{kl}\}, \{\mathbf{d}_k\})$
 10: $\{\mathbf{d}_k\} \leftarrow \text{WAVEFORM_CENTERING}(\{a_{kl}\}, \{\delta_{kl}\}, \{\gamma_{kl}\}, \{\mathbf{d}_k\})$
 11: **end loop**
Output: $\{a_{kl}, \delta_{kl}, \gamma_{kl}, \mathbf{d}_k\}$.

As explained in Section 4.3.4, we will use the sparse representation $\{a_{kpq}, \delta_p, \gamma_q\}$ for the coefficient update, but change to the explicit representation

$$(\{a_{kl}\}, \{\delta_{kl}\}, \{\gamma_{kl}\}) \leftarrow \text{SWITCH_REPRESENTATION}(\{a_{kpq}\}, \{\delta_p\}, \{\gamma_q\})$$

during the rest of the algorithm. This representation directly describes each occurrence of a waveform \mathbf{d}_k by its coefficient a_{kl} , time instant δ_{kl} , and duration γ_{kl} .

6.3.1 Coefficient updates

As in the previous chapters, we create a dictionary $\mathbf{D} = \{\mathbf{d}_k^{pq}\}$ with the dilated and translated waveforms $\mathbf{d}_k^{pq} = \frac{1}{\sqrt{\gamma_q}} \mathbf{d}_k \left(\frac{1}{\gamma_q} (\cdot - \delta_p) \right)$ before minimizing (6.2, 6.3) with respect to the coefficients $\{a_{kpq}\}$. The procedure COEFFICIENT_UPDATES is now given below.

Note that the dictionary \mathbf{D} contains $K \cdot P \cdot (2Q + 1)$ atoms and is thus typically very large. Calculation with LARS as in the previous chapter

```

1: procedure COEFFICIENT_UPDATES( $\mathbf{x}, \mathbf{D}, \lambda, \Delta$ )
2:   Solve with matching pursuit:
3:    $\{a_{kpq}\} \leftarrow \operatorname{argmin}_{\{a'_{kpq}\}} \left\| \mathbf{x} - \sum_{k=1}^K \sum_{p=1}^P \sum_{q=-Q}^Q a'_{kpq} \mathbf{d}_k^{pq} \right\|_2^2 + \lambda \sum_{k=1}^K \sum_{p=1}^P \sum_{q=-Q}^Q |a'_{kpq}|$ ,
4:   s.t.  $a'_{kpq} \neq 0 \Rightarrow (a'_{k'p'q'} = 0 \text{ for all } k', q', p' \neq p : |\delta_p - \delta_{p'}| < \Delta)$ .
5: end procedure, return  $\{a_{kpq}\}$ 

```

will therefore be infeasible. Instead we suggest to use matching pursuit (MP), which allows for a highly optimized implementation in the case of shift-invariant atoms (cf. Appendix A). However, the bottleneck in the calculations are the different dilations, which cannot be handled as efficiently as the translations. If a very high resolution across dilations is desired, it can be beneficial to use a multi-resolution approach: start each MP step with a medium resolution and refine the dilation factor after every waveform detection.

The minimal distance constraint in line 4 can be enforced after each MP step: we determine the temporal positions $\delta_{p'}$ close to the current waveform detection at δ_p and exclude all corresponding coefficients $\{a_{k'p'q'}\}$ from later activation. If desired, non-negativity (4.5) is easily enforced by selecting only atoms with a positive correlation coefficient.

As shown in Appendix A, the regularization parameter λ in the Lasso problem can be seen as a correlation threshold. Hence, the MP algorithm is stopped when the maximal correlation of all non-selected atoms with the residual signal is below λ .

The selection of λ is crucial for the outcome of the method. A possibility is to choose λ as a fraction of the largest correlation of all translated and dilated waveforms with the data, that is, $\lambda = \alpha \mathbf{D}^t \mathbf{x}$ with $0 < \alpha < 1$. Since \mathbf{D} changes in every iteration of Algorithm 4, λ also does. In this case, we instead provide the fixed parameter α , which can be determined by assumptions on the amount of amplitude variation of the signal components of interest across the recording.

6.3.2 Waveform updates

The waveforms are updated by solving the minimization problem (6.2) iteratively for each waveform \mathbf{d}_k while leaving the coefficients fixed. This is described in the procedure WAVEFORM_UPDATES.

The minimization problem in line 3 can be solved as follows: In analogy to Section 4.4.3, we first define linear transformations ψ_k with

$$\psi_k(\mathbf{d}_k) = \sum_{l=1}^{L_k} a_{kl} \frac{1}{\sqrt{\gamma_{kl}}} \mathbf{d}_k \left(\frac{1}{\gamma_{kl}} (\cdot - \delta_{kl}) \right), \quad (6.4)$$

```

1: procedure WAVEFORM_UPDATES( $\mathbf{x}, \{a_{kl}\}, \{\delta_{kl}\}, \{\gamma_{kl}\}, \{\mathbf{d}_k\}$ )
2:   for  $k = 1$  to  $K$  do
3:      $\mathbf{d}_k \leftarrow \underset{\mathbf{d}_k}{\operatorname{argmin}} \left\| \mathbf{x} - \sum_{k'=1}^K \sum_{l=1}^{L^{k'}} a_{k'l} \frac{1}{\sqrt{\gamma_{k'l}}} \mathbf{d}_{k'} \left( \frac{1}{\gamma_{k'l}} (\cdot - \delta_{k'l}) \right) \right\|_2^2$ 
4:   end for
5: end procedure, return  $\{\mathbf{d}_k\}$ 

```

such that line 3 becomes

$$\mathbf{d}_k \leftarrow \underset{\mathbf{d}_k}{\operatorname{argmin}} \left\| \mathbf{x} - \sum_{k'=1}^K \psi_{k'}(\mathbf{d}_{k'}) \right\|_2^2.$$

This results in the waveform update

$$\mathbf{d}_k = \left(\psi_k^t \psi_k \right)^+ \psi_k^t(\mathbf{r}_k) \quad (6.5)$$

where

$$\mathbf{r}_k = \mathbf{x} - \sum_{k' \neq k} \psi_{k'}(\mathbf{d}_{k'})$$

describes the residual that results when subtracting all waveform occurrences from the signal, except those of \mathbf{d}_k .

Note that the waveform update (6.5) describes a kind of deconvolution: The operator ψ_k defined in (6.4) represents a convolution of the sparse coefficient vector

$$(0, \dots, a_{k1}, \dots, a_{kl}, \dots, a_{k(l+1)}, \dots, a_{kL_k}, \dots, 0)$$

with the dilatable argument \mathbf{d}_k . The dots between a_{kl} and $a_{k(l+1)}$ denote zero-entries, corresponding to the distances $\delta_{k(l+1)} - \delta_{kl}$ between two occurrences of \mathbf{d}_k . The conditioning of ψ_k depends on the amount of overlap between occurrences, which can be controlled, for instance, through a large minimal distance parameter Δ . However, this would also limit the possibility of detecting close waveform occurrences. Alternatively, the update can be performed only on non-overlapping occurrences of \mathbf{d}_k , provided that enough of those are available. In this case, the update (6.5) reduces to a weighted average of rescaled signal segments that contain the waveform occurrences. If updating on strongly overlapping occurrences cannot be avoided, regularization should be considered.

6.3.3 Waveform centering

The procedure WAVEFORM_CENTERING performs centering operations with respect to latencies and dilations, as well as normalization. In the previous

chapters, we realigned each waveform such that the mean latency across epochs was close to 0. This notion does not make sense in the case of a single long signal. Instead, we suggest to ensure that the signal component of interest is approximately centered in the time window of \mathbf{d}_k . The concrete way of implementing this is application dependent. In our following evaluation on data with epileptiform spikes, for instance, we chose to align spikes with respect to their maximal peak.

Afterwards, centering with respect to dilations is performed as described in Section 4.4.4. The weighted geometric mean $\bar{\gamma}_k$ over the calculated dilations $\{\gamma_{kl}\}_l$ is obtained for each waveform \mathbf{d}_k as

$$\bar{\gamma}_k = \left(\prod_l^{L^k} \gamma_{kl}^{|a_{kl}|} \right)^{\frac{1}{\sum_l |a_{kl}|}}.$$

This is followed by the centering step

$$\mathbf{d}_k \leftarrow \frac{1}{\sqrt{\bar{\gamma}_k}} \mathbf{d}_k \left(\frac{1}{\bar{\gamma}_k}(\cdot) \right).$$

If a waveform \mathbf{d}_k varies much in duration across the recorded signal, it is important that the distribution of the $\{\gamma_{kl}\}_l$ is geometrically centered at 1. Otherwise, some large stretches or compressions of waveforms might not be included in the set $\{\gamma_{-Q}, \dots, \gamma_Q\}$. As before, the normalization constraint on waveforms is ensured by setting

$$\mathbf{d}_k \leftarrow \frac{\mathbf{d}_k}{\|\mathbf{d}_k\|_2}.$$

The above steps are summarized in the procedure WAVEFORM_CENTERING.

```

1: procedure WAVEFORM_CENTERING( $\{a_{kl}\}, \{\delta_{kl}\}, \{\gamma_{kl}\}, \{\mathbf{d}_k\}$ )
2:   for  $k = 1$  to  $K$  do
3:     Align  $\mathbf{d}_k$ , for instance at maximal peak.
4:      $\bar{\gamma}_k = \left( \prod_l^{L^k} \gamma_{kl}^{|a_{kl}|} \right)^{\frac{1}{\sum_l |a_{kl}|}}$ ,
5:      $\mathbf{d}_k \leftarrow \frac{1}{\sqrt{\bar{\gamma}_k}} \mathbf{d}_k \left( \frac{1}{\bar{\gamma}_k}(\cdot) \right)$ ,
6:      $\mathbf{d}_k \leftarrow \frac{\mathbf{d}_k}{\|\mathbf{d}_k\|_2}$ .
7:   end for
8: end procedure, return  $\{\mathbf{d}_k\}$ 

```

6.4 C-AWL for spike learning

The C-AWL algorithm introduced in the previous section is a tool that allows to learn repeating structures from a long contiguous dataset. In

particular, it can be used for processing datasets containing epileptiform spikes, such as the one presented in Section 2.7 and treated in Section 5.5 in an epoched setting. Here, we illustrate the advantages of directly processing the non-epoched data with C-AWL, including automatic detection of spike occurrences and the possibility of handling overlaps.

In order to use C-AWL for spike processing we will make a few specializations that address specific aspects about the spikes. For this purpose, we will assume the waveforms \mathbf{d}_k to exclusively capture the different spike shapes. Note that this was different when we applied E-AWL to the epoched data in Section 5.5, which produced spike shapes as well as other oscillatory waveforms. In fact, these waveforms were learned *blindly*, that is, without any prior assumptions about their shapes. However, the temporal information that was provided through the prior epoching step is not available in the contiguous setting. Hence, it is necessary to use some prior knowledge about the spike shapes in order to avoid fitting irrelevant signal structures.

We propose two different spike models: The first one, MC-Spike, is a multi-class model with constant shapes of the spike templates. That is, within each class, only amplitude variations are allowed across the different spike occurrences. The second one, AD-Spike, consists of a single adaptive spike template with variable duration. These approaches are presented in the following subsections and their respective advantages are illustrated on the contiguous spike dataset from Section 2.7. Finally, their detection performance is evaluated for different noise levels, and the results are compared to common template matching.

6.4.1 Multi-class spike learning (MC-Spike)

We now assume the signal \mathbf{x} to contain spikes from K different classes, given by $\{\mathbf{d}_k\}_{k=1}^K$. The shape of each \mathbf{d}_k is fixed, but it may occur with different amplitudes and latencies throughout \mathbf{x} . As before, we impose a minimal distance Δ between spike occurrences in order to well separate them. The corresponding model thus reads

$$\mathbf{x} = \sum_{k=1}^K \sum_{p=1}^P a_{kp} \mathbf{d}_k(\cdot - \delta_p) + \epsilon, \quad (6.6)$$

$$\text{with } a_{kp} \neq 0 \Rightarrow (a_{k'p'} = 0 \quad \forall k', p' \neq p : |\delta_p - \delta_{p'}| < \Delta), \quad (6.7)$$

$$\text{and } \forall k, p : a_{kp} \geq 0, \quad (6.8)$$

where the residual ϵ contains the spike-unrelated part of the signal. The minimal distance constraint (6.7) ensures that each spike occurrence is represented by a single waveform \mathbf{d}_k . If an estimate of each spike class is available, then the contiguous AWL algorithm may be applied directly to this initialization. However, as this is rarely the case, we will assume that only one basic spike template \mathbf{d}_1 is given and the number K of spike classes

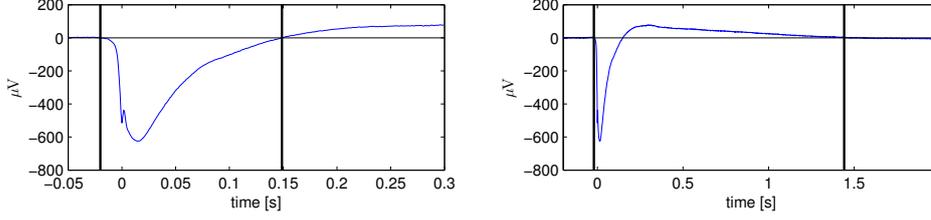


Figure 6.2: Two different time windows of a typical spike shape. **Left:** This window shows the negative wave of the spike, which lasts close to 0.2 seconds. **Right:** The entire spike, consisting of the fast negative wave followed by a slow positive wave, is approximately 1.5 seconds long.

is unknown. In this case, we propose the hierarchical learning approach (cf. Section 4.6) given in Algorithm 5. Note that we use the relative detection threshold α rather than the absolute threshold λ (cf. Section 6.3.1).

Algorithm 5 MC-Spike

Input: $\mathbf{x} \in \mathbb{R}^N$, $\mathbf{d}_1 \in \mathbb{R}^n$, $K_{max} \in \mathbb{N}$, $0 < \alpha < 1$, $\Delta > 0$.

- 1: **for** $K = 1$ to K_{max} **do**
- 2: **if** $K > 1$ **then**
- 3: $\mathbf{d}_K \leftarrow \text{INITIALIZE_SPIKE}(\mathbf{x}, \{a_{kl}\}, \{\delta_{kl}\}, \{\mathbf{d}_k\})$
- 4: **end if**
- 5: $(\{a_{kl}\}, \{\delta_{kl}\}, \{\mathbf{d}_k\})_{k=1}^K \leftarrow \text{C-AWL}(\mathbf{x}, \{\mathbf{d}_k\}_{k=1}^K, \alpha, \Delta)$
- 6: Save representation: $R_K \leftarrow (\{a_{kl}\}, \{\delta_{kl}\}, \{\mathbf{d}_k\})$.
- 7: **end for**

Output: $R_1, \dots, R_{K_{max}}$.

The algorithm starts by learning a single spike form representation with C-AWL (Algorithm 4), initialized with the provided template \mathbf{d}_1 . Then, an additional spike template \mathbf{d}_2 is initialized, and a new representation is learned starting from $\{\mathbf{d}_1, \mathbf{d}_2\}$. This process is repeated until the maximal representation size K_{max} is reached. Note that C-AWL is used without waveform dilations. For better clarity, we again note that the learned representations are *explicit*, with a_{kl} denoting only the non-zero coefficients, opposed to the sparse representation in model (6.6).

The result of Algorithm 5 is a set of spike representations $R_1, \dots, R_{K_{max}}$ of increasing cardinality. For reasonably small K_{max} , we suggest to qualitatively compare the different representations. Alternatively, model selection methods such as the Bayesian or Akaike information criterion (BIC, AIC, respectively) may be used to determine the most likely number K . An overview of model selection criteria can be found in [Stoica and Selen \(2004\)](#).

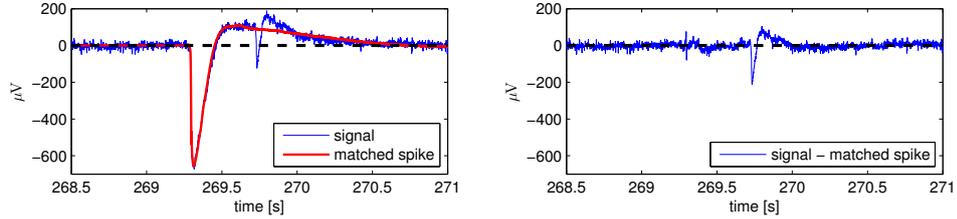


Figure 6.3: Illustration of spike subtraction in MC-Spike. **Left:** A spike template \mathbf{d}_k (red) is matched to the prominent spike during sparse coding with MP. Note that the negative peak of the closely following spike hardly reaches below the baseline due to the superposition with the slow positive wave of the first spike. **Right:** Subtraction of the matched spike template results in a baseline correction. The peak of the second spike now reaches a larger absolute minimum, which facilitates its detection.

6.4.1.1 Handling overlapping spikes

As mentioned before, an advantage of processing the signal without prior epoching is the possibility of appropriately handling overlaps. Note that (6.7) imposes a minimal distance only between the *onsets* of the spikes. Hence, overlaps are still possible if Δ is smaller than the total duration of the waveforms \mathbf{d}_k . On the other hand, a too small Δ can cause the encoding of a single spike through several \mathbf{d}_k .

In order to assess the appropriate choice of Δ , we investigate a typical spike shape found in the dataset, shown in Figure 6.2: the first negative wave (left plot) is about 0.2 seconds long while the entire spike, including the slow positive wave, lasts about 1.5 seconds (right plot). In order to provide for a clear spike separation, we choose to avoid overlaps between the fast negative waves of the spikes. Hence, we set $\Delta = 0.2$ seconds. This still allows large overlaps of the spikes whose total duration is longer.

The detection of spikes is performed through matching pursuit in the procedure `COEFFICIENT_UPDATES` (cf. Section 6.3.1). This means that after each spike detection, the contribution of this spike is subtracted from the residual signal (cf. Appendix A), which can improve the detection of a closely following spike. This is illustrated in Figure 6.3.

6.4.1.2 Initializing new spikes

When a new spike \mathbf{d}_K is added to the representation, its initialization is extremely important. If initialized with spike unrelated structures, \mathbf{d}_K may not be able to “compete” with the previously learned spike classes. That is, the $\mathbf{d}_1, \dots, \mathbf{d}_{K-1}$ will correlate much stronger with the spikes in the data, and will therefore be selected first in the coefficient updates. In this case, \mathbf{d}_K may not become active at all or only pick up noise. We implement the routine `INITIALIZE_SPIKE` by choosing the previously detected spike occur-

rence which is the least well explained by the $\{\mathbf{d}_k\}_{k=1}^{K-1}$. More concretely, we first calculate the residual

$$\mathbf{r} = \mathbf{x} - \sum_{k=1}^{K-1} \sum_{l=1}^{L_k} a_{kl} \mathbf{d}_k(\cdot - \delta_{kl}). \quad (6.9)$$

Now, we can describe the goodness of fit (GOF) of the detection at δ_{kl} by

$$\rho(\delta_{kl}) = 1 - \frac{\|\mathbf{r}(\cdot + \delta_{kl}) \cdot \mathbb{1}_{\text{supp}(\mathbf{d}_k)}\|_2}{\|\mathbf{r}(\cdot + \delta_{kl}) \cdot \mathbb{1}_{\text{supp}(\mathbf{d}_k)} + a_{kl} \mathbf{d}_k\|_2}. \quad (6.10)$$

Here, $\mathbb{1}_{\text{supp}(\mathbf{d}_k)}$ denotes the indicator function for the support of \mathbf{d}_k . The operation $\mathbf{r} \rightarrow \mathbf{r}(\cdot + \delta_{kl}) \cdot \mathbb{1}_{\text{supp}(\mathbf{d}_k)}$ thus corresponds to selecting a window in \mathbf{r} at time δ_{kl} (cf. Figure 6.1). After determining the latency at which this fit value is minimal,

$$\delta_{k'l'} = \underset{\{\delta_{kl}\}}{\text{argmin}}(\rho(\delta_{kl})),$$

we initialize the new spike at this position by setting

$$\mathbf{d}_K = \mathbf{r}(\cdot + \delta_{k'l'}) \cdot \mathbb{1}_{\text{supp}(\mathbf{d}_k)} + a_{k'l'} \mathbf{d}_{k'}.$$

Note, that we do not initialize \mathbf{d}_k directly in the signal \mathbf{x} but subtract the contributions of all spike detections except for the one at $\delta_{k'l'}$, possibly eliminating overlapping effects.

However, this initialization still bears the risk that the residual contains noise or artifacts, and \mathbf{d}_K would be susceptible to fit other artifacts during the following coding step. Hence, we only initialize the part of length Δ that corresponds to the sharp negative wave of the spike (cf. Figure 6.2). This is followed by normalization. If the newly initialized spike template detects too few spikes (< 3 in our implementation), we reinitialize it again with a random spike occurrence.

6.4.1.3 Performance measures

In order to compare the representations R_K , we define three performance measures. First, we define the average goodness of fit (GOF) $\bar{\rho}$ as the mean over all $\rho(\delta_{kl})$ defined in (6.10). Second, the relative root mean square (RMS) error is defined as

$$\varepsilon_{\mathbf{r}} = \frac{\|\mathbf{r}\|_2}{\|\mathbf{x}\|_2}$$

with residual \mathbf{r} (6.9). Finally, all spike occurrences were located in the data by visual inspection, resulting in a total number of 520 spikes. Their latencies were used to evaluate the detection performance of the methods.

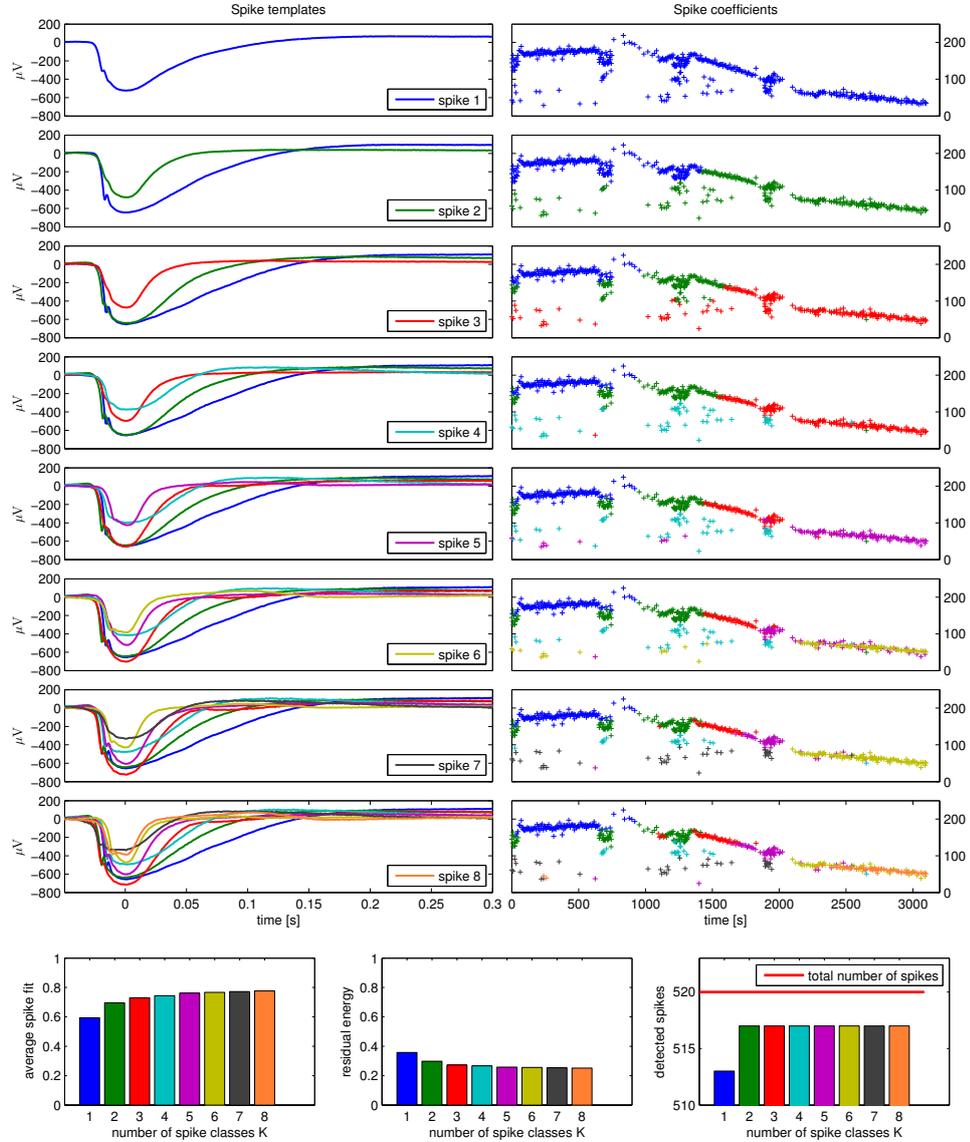


Figure 6.4: Spike representations learned with MC-Spike on LFP data (cf. Section 2.7). Each representation (row) is a result of adding a new spike class to the previous representation and relearning on all classes (cf. Algorithm 5). **Left column:** The learned spike templates \mathbf{d}_k are shown on a window around the negative waves. They are scaled with the average amplitude of the spikes that they represent in the data. The main difference across the templates appears to consist in their duration. **Right column:** The temporal locations δ_{kl} of the spike detections are plotted against their correlation coefficients a_{kl} . Interestingly, in most representations, the different spike classes appear to be clearly separable in this plot. **Bottom row:** Three different measures (columns) are shown for each representation: average spike fit $\bar{\rho}$, relative RMS error ε_r , and number of detected spikes (total: 520).

6.4.1.4 Experiments

We applied Algorithm 5 to the approximately one hour recording of local field potential (LFP) with epileptiform discharges presented in Section 2.7. This recording had a remarkably good signal-to-noise ratio (SNR), such that even small discharges were clearly visible. However, the small amplitude spikes often occurred during the slow positive waves of preceding spikes (cf. Figure 6.3), making them hard to detect when only looking at absolute amplitudes. In order to be able to learn high frequency components of the spikes, we processed the signal in its original resolution of 1250 Hz.

We initialized the first template spike \mathbf{d}_1 on a 2 second time window in the data that contained a clearly visible spike occurrence. We set $K_{max} = 8$ and used a minimal spike-to-spike distance $\Delta = 0.2$ seconds. Due to the high SNR, we set a low relative detection threshold, $\alpha = 0.1$ (cf. Section 6.3.1), in order to detect also the smaller amplitude spikes. The number of iterations in C-AWL was set to 10 for each representation R_K . The computation of all 8 representations took approximately 7 minutes on a laptop.

The results are shown in Figure 6.4. Each of the first eight rows in the figure corresponds to one representation R_K . The learned spike templates are shown in the left column. Note that while each spike class is scale-invariant and its template \mathbf{d}_k normalized, we here scaled the templates with the mean amplitude of the spikes they represented in the data. The main difference between the template shapes appears to be the different duration of the negative wave (note that we only show a 0.35 second time window for better illustration of this difference). The right column shows the coefficients a_{kl} plotted against the latencies δ_{kl} , corresponding to the positions and l_2 -norms of the detected spikes. It becomes clearly visible that the maximal spike energy decreases over time. Interestingly, the spike classes represented by the different colors are well-separated in this plot. In row three, for example, during the first third of the recording, the high energy spikes are represented by the blue template with long negative wave. The green spike class dominates the second third, while the short red template represents the last third as well as other low energy spikes across the recording. This clustering becomes finer with increasing number of templates.

The last row of Figure 6.4 shows the different performance measures from Section 6.4.1.3 for each representation: average GOF $\bar{\rho}$ of the detected spikes, relative RMS error ε_r , and the number of detected spikes, respectively. All detected spikes were true positives, due to the good signal-to-noise ratio and sufficiently large detection threshold α . The three measures show the most significant performance increase across the first representations R_1, R_2, R_3 .

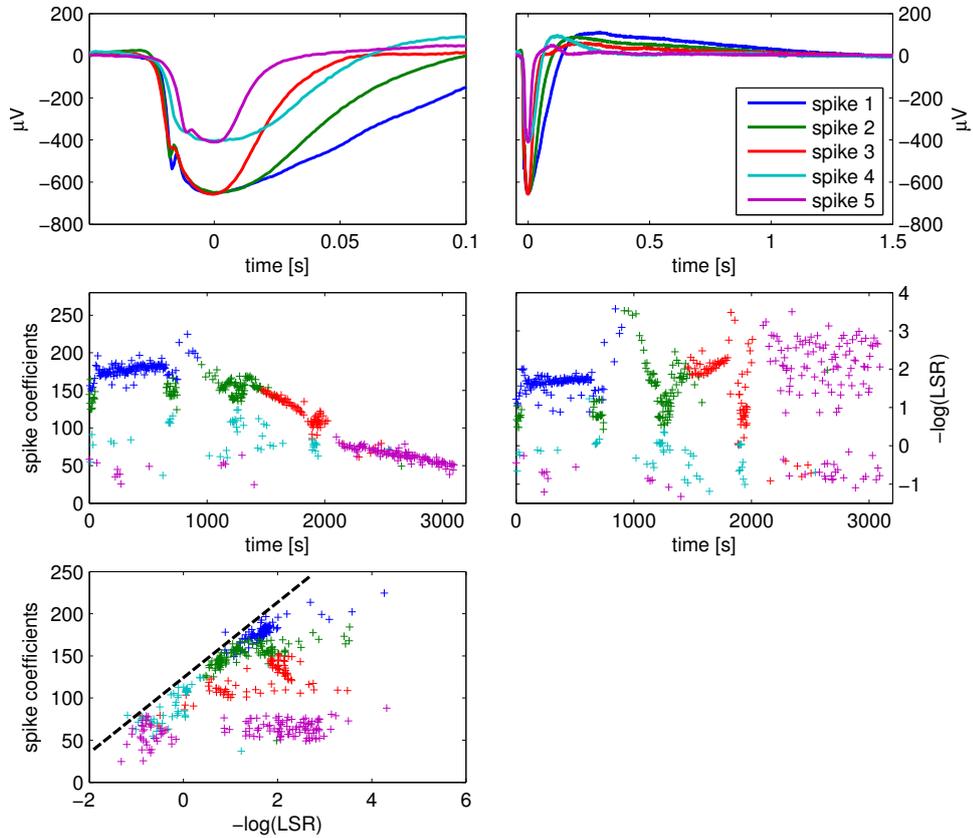


Figure 6.5: Spike representation learned with MC-Spike and $K = 5$ spike classes. **Top row:** The 5 spike templates are shown on two different time windows. Note that some of the spike shapes have a characteristic kink about 10-20 ms before the maximal negative peak (left). Besides that, the main differences consist in the duration and amplitude of the peak. On the longer time window (right), it is noticeable that there are also differences in the slow positive wave following the negative peak. **Middle row:** The two figures show coefficients and the negative logarithm of the local spiking rates (LSR) in time, respectively. In the first part of the recordings, these profiles look very similar, especially for the low energy spikes from classes 4 (light blue) and 5 (purple), suggesting that the $-\log(\text{LSR})$ can account for some of the local changes across spike energies. **Lower left:** This relationship becomes even more apparent when directly plotting the $-\log(\text{LSR})$ against coefficients. In fact, it seems that the potential energy of each spike is bounded clearly by a linear function of the $-\log(\text{LSR})$ (dashed line).

Investigating spiking rates

The energies of the spikes (Figure 6.4, right column) showed to decrease across the dataset. This can be explained by the diminishing effect of the inhibition blocker bicuculline, which was injected into the rat’s cortex prior to the recording (cf. Section 2.7). However, large variations across spike energies were also present in small time intervals.

In the original dataset, we were able to observe that a spike that closely followed another spike generally had significantly less energy than the previous one. We therefore investigated the relationship between the intervals between spikes and their energies, in order to explain the energy variations.

For this purpose, we first defined the *local spiking rate* (LSR) of a spike at latency δ_l as

$$\text{LSR}(\delta_l) = \frac{1}{\delta_l - \delta_{l-1}} \quad [\text{Hz}]. \quad (6.11)$$

Here, the δ_l denote the temporally ordered latencies of the spike occurrences, disregarding the different spike classes. For the following attempt to explain the energy variations, it proved useful to investigate the relationship with the negative logarithm of the LSR.

We now focus on the representation for $K = 5$ in more detail, which is shown in Figure 6.5. The top row shows the five templates on two different time windows. Notice that some of the spike templates clearly show a kink around 10 – 20 ms before the main negative peak, which is a characteristic property of the epileptiform spikes (Saillet et al., 2015).

The clear appearance of this fast peak is due to the processing in high signal resolution (1250 Hz) and gives proof of the exact alignment properties of MC-Spike. The global forms of the learned spike templates (right) show differences in the amplitude and durations also of the slow positive wave.

In the middle row of Figure 6.5, we now compare the learned spike coefficients (left), describing the spike energies (or more precisely l_2 -norms), with the negative log LSR (right). In the first part of the recording, the two profiles look very similar, especially for the low energy spikes of classes 4 (light blue) and 5 (purple). Hence, the $-\log(\text{LSR})$ seems to explain some of the local changes in spike energies.

The relationship can be observed more directly when plotting coefficients directly against the $-\log(\text{LSR})$ (lower left). The dashed line illustrates that the potential energy of a spike appears to depend linearly on the $-\log(\text{LSR})$. This can be interpreted as a need of the system (i.e., the neurons involved in the generation of the spike) for a certain time after each generated spike in order to recover its potential energy.

6.4.2 Adaptive duration spike learning (AD-Spike)

The results from the multi-class spike model showed that a major difference between the spike templates was their different duration. In this section, we therefore consider a model with a single spike template of adaptive duration (AD-Spike),

$$\mathbf{x} = \sum_{p=1}^P \sum_{q=-Q}^Q a_{pq} \frac{1}{\sqrt{\gamma}} \mathbf{d} \left(\frac{1}{\sqrt{\gamma}} \cdot -\delta_p \right) + \epsilon, \quad (6.12)$$

$$\text{with } a_{pq} \neq 0 \Rightarrow (a_{p'q'} = 0 \quad \forall q', p' \neq p : |\delta_p - \delta_{p'}| < \Delta), \quad (6.13)$$

$$\text{and } \forall p, q : a_{pq} \geq 0. \quad (6.14)$$

We can directly apply C-AWL (Algorithm 4) with $K = 1$ to this problem. For convenience, this is described in Algorithm 6. As for MC-Spike, we use the relative correlation threshold α instead of the absolute threshold λ (cf. Section 6.3.1).

Algorithm 6 AD-Spike

Input: $\mathbf{x} \in \mathbb{R}^N$, $\mathbf{d} \in \mathbb{R}^n$, $Q \in \mathbb{N}$, $\beta > 1$, $0 < \alpha < 1$, $\Delta > 0$.

1: $(\{a_l\}, \{\delta_l\}, \{\gamma_l\}, \mathbf{d}) \leftarrow \text{C_AWL}(\mathbf{x}, \mathbf{d}, Q, \beta, \alpha, \Delta)$

Output: $\{a_l\}, \{\delta_l\}, \{\gamma_l\}, \mathbf{d}$.

Experiments

We applied Algorithm 6 to the LFP-dataset from Section 2.7 with full sampling resolution (1250 Hz) in order to learn the AD-Spike representation. We used a total number of $2Q + 1 = 61$ dilation factors, and a maximal relative stretch of $\beta^{2Q} = 8$ in order to capture the full range of spike durations. To increase the resolution between dilations, we used a multi-resolution approach (cf. Section 6.3.1) with additional resolution factor of 11. As for MC-Spike, we set $\alpha = 0.1$, $\Delta = 0.2$ seconds and initialized \mathbf{d} on a 2 second window around a prominent spike in the data. Since the majority of spikes in the data were well isolated, we performed the updates only on these spikes, in order to avoid artifacts of overlapping spikes (cf. Section 6.3.2). We performed 4 iterations with C-AWL after which we observed convergence. The computation took approximately 2 minutes on a laptop.

The resulting representation is shown in Figure 6.6. The top row shows two different time windows of the learned template \mathbf{d} . Due to centering with respect to dilations in Algorithm 4, its width represents the average width of the detected spikes. Note that despite the temporal rescaling during the spike updates, the fast peak before the maximal negative peak is still visible. The bottom row shows the coefficients and dilations, respectively, plotted

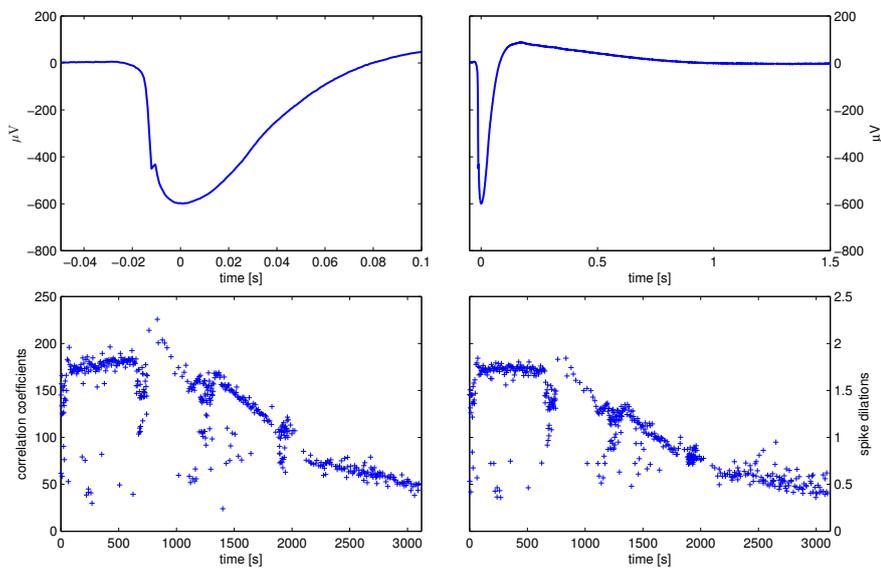


Figure 6.6: Dilatable spike representation learned with AD-Spike. **Top row:** The learned spike template is shown on two different time windows. Notice that the characteristic fast peak is clearly visible (kink around -0.15 seconds, left figure). **Bottom row:** The profiles of the spikes' coefficients (left) and dilation factors (right) across time are compared. The strong similarity shows that the different spike energies are explained primarily by their different durations.

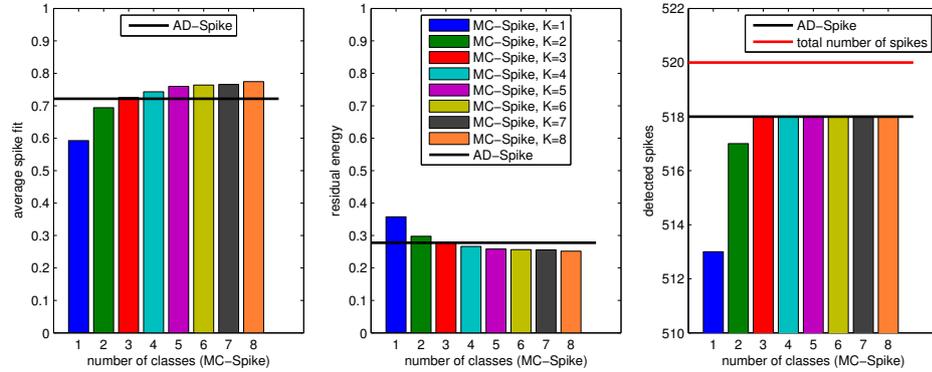


Figure 6.7: The qualities of the MC-Spike representations are compared to the AD-Spike representation by the measures goodness of fit $\bar{\rho}$, relative RMS error ε_r , and number of detected spikes. In these terms, the descriptive quality of AD-Spike seems to be similar to MC-Spike with $K = 3$ spike classes.

in time. These profiles are very similar, suggesting that the different spike energies mostly result from their different durations.

Figure 6.7 shows a comparison between the MC-Spike (cf. 6.4.1.4) and AD-Spike in terms of the measures average GOF $\bar{\rho}$, relative RMS error ε_r , and number of detected spikes (cf. Section 6.4.1.3). In these measures, AD-Spike is comparable to MC-Spike with $K = 3$. For $K > 3$, MC-Spike achieves a slightly higher fit $\bar{\rho}$ of spikes which is also reflected in the lower RMS error. This shows that not all of the spike variability can be explained by linear stretches.

An obvious advantage of AD-Spike over MC-Spike is the very compact representation in which all variability is parametrized in form of amplitude, latencies, and spike duration. In addition, no model order has to be selected.

6.4.3 Comparing detection performance

The LFP dataset that we processed in the previous subsections with MC-Spike and AD-Spike had an exceptionally high signal-to-noise ratio (SNR) with few artifacts. Hence, it was possible to detect even overlapping spikes of small amplitudes with high precision. Here, we evaluate the detection accuracy of MC-Spike and AD-Spike for different SNRs and compare the results to template matching based on cross-correlation. For this purpose, we added different levels of white Gaussian noise to the LFP dataset from Section 2.7, providing for a semi-realistic study. The “true” spike occurrences were determined manually in the original dataset, resulting in 520 marked spikes (cf. Section 6.4.1.3).

Template matching approaches have been used in different studies for

epileptiform spike detection (Dingle et al., 1993; Larsson et al., 2009). Typically, a given template is cross-correlated against the data. Given a certain correlation threshold, the correlation coefficients above this threshold indicate the detected spikes. The detection accuracy of such an approach depends strongly on the used template. As these are not always available, spikes previously extracted from the data are frequently used as templates.

For our evaluation of detection performance, we compared two settings: availability of a “good” template vs. the use of a noisy template. The good template was obtained as the average over the non-overlapping epoched spike occurrences from Section 5.5. The second template was directly taken from each noisy dataset. For this purpose, we first used the original data to define a time window containing a prominent, non-overlapping spike. Then, during the processing of each dataset, this window was extracted directly from the noisy dataset to define the template.

In order to avoid a bias due to threshold selection for the different methods, we assumed the number of spikes to be known *a priori*. While this is not a very realistic assumption, we found that it provided for a more objective comparison between the methods. Hence, all methods stopped detection after 520 temporal locations were determined. This overruled the threshold parameter α in MC-Spike and AD-Spike. In order to avoid multiple close detections of the same spike, we imposed a minimal spike-to-spike distance of $\Delta = 0.2$ seconds for different spike detections for all methods.

For MC-Spike we used the hierarchical Algorithm 5 with $K_{max} = 5$, producing 5 different representations. Here, we only show the results for $K = 1$ and $K = 5$ for better visibility.

The results are shown in Figure 6.8. The top row shows the detection accuracies for initialization with a the good template for low (left) and high SNRs (right). Note that the plot on the right only contains a narrow window of detection accuraries ($> 97\%$) as all methods showed almost perfect detection for a sufficiently high SNR. Below an SNR of -25 dB, template matching yielded best results, while above -25 dB, AD-Spike achieved highest accuracy (left plot). For a high SNR (right plot), AD-Spike and MC-Spike with $K = 5$ performed slightly better than template matching. Note that for MC-Spike, the single class model shows higher detection rate for lower SNRs while for high SNRs the model with $K = 5$ performs better.

When initializing the template directly from the noisy data (second row), template matching performed significantly worse than MC-Spike and AD-Spike, especially for low SNRs between -30 dB and -10 dB. AD-Spike shows best performance for all noise levels. Note the extremely steep performance increase for both MC-Spike and AD-Spike around -30 dB. Once the SNR is sufficiently high to detect enough spikes, these methods can benefit from their ability to relearn the bad template which produces the drastic improvement. In fact, above -25 dB, AD-Spike and MC-Spike completely compensate for the bad initialization, which becomes clear when comparing the two

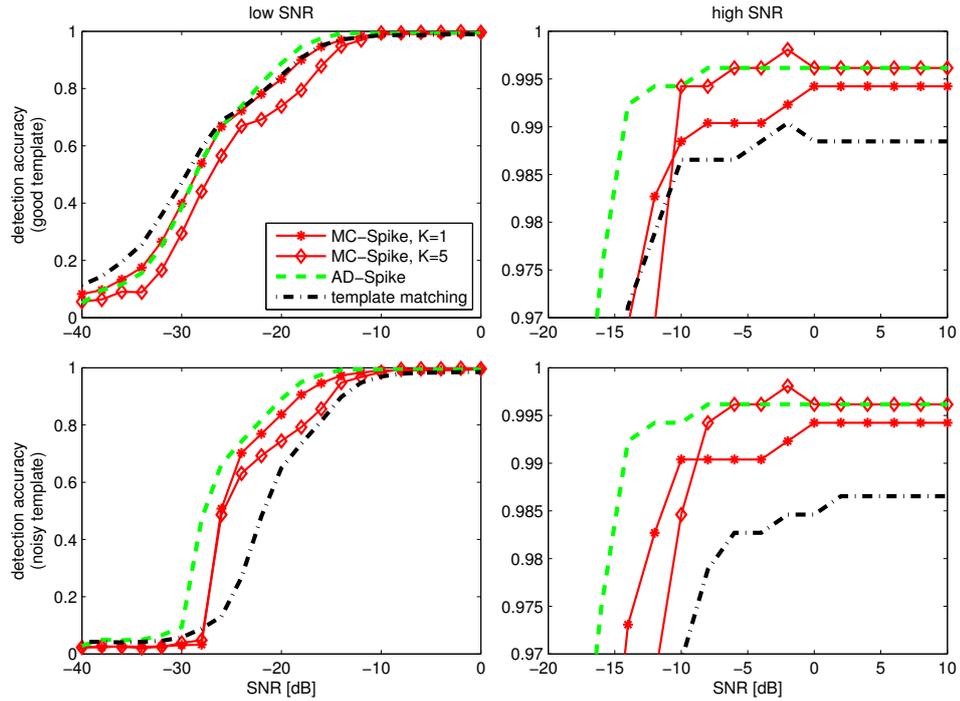


Figure 6.8: The performances of MC-Spike, AD-Spike, and template matching are compared in terms of detection accuracy. For this purpose, Gaussian noise of different amplitudes was generated and added to the original LFP dataset (cf. Section 2.7). **Top row:** Detection accuracies for increasing SNRs obtained when using a “good” spike template. Template matching performs better than AD-Spike and MC-Spike for SNRs below -25 dB (left). Above -25 dB the accuracy of AD-Spike is slightly better. All MC-Spike representations show worse results than template matching and AD-Spike, except for an SNR above -10 dB (right). **Bottom row:** When initializing the template directly in the noisy data, AD-Spike shows best performance for all SNRs. Template matching yields significantly lower accuracies for low SNRs.

rows.

In conclusion, AD-Spike shows a very good detection performance. MC-Spike shows a slightly inferior performance and differs depending on the number of classes. Template matching only performs better than the other methods in cases of high noise levels and a good template.

6.5 Exploring neuro-vascular coupling

The spike representations of the LFP recording learned in the previous section provide important parameters to assess the coupling with the hemodynamic activity. In Figure 6.5, we showed a correlation between the local spiking rates (LSR) of the spikes and their energies. We will now use the LSR to explore the relationship of the spiking activity with dynamics in the cerebral blood flow (CBF), which had been recorded simultaneously with the LFP-spikes (cf. Section 2.7).

The top row of Figure 6.9 shows the absolute values of the spectrogram of the CBF in a frequency range of 0 to 1.2 Hz, calculated by windowed Fourier transforms. Due to the large intensities around the zero frequency component, we show all values above 200 in the same color for better visibility. Note that while most frequency content is below 0.2 Hz, there is an ongoing rhythmic activity around 1 Hz. As supposed by the experimentalists of the study, this activity resulted from the rat's respiration.

The LSRs (6.11) for the spike representation learned with MC-Spike for $K = 5$ are now plotted on top of the CBF spectrogram in the middle row of Figure 6.9. For better visibility, we used a gray scale for the spectrogram and plotted all LSR values in red, regardless of their corresponding classes in the MC-Spike representation. While most spiking rates are around or below 0.2 Hz, some spikes are significantly faster. Especially around 700, 1300, and 1950 seconds, we can observe a concentration of spikes with LSR above 0.2 Hz. Interestingly, the LSRs of some of the spikes match exactly the frequency of the CBF activity around 1 Hz (green ellipses). In fact, around 700 seconds, we can even see clusters of LSRs around 0.5, 0.33, and 0.25 Hz, corresponding to the subharmonics of this activity (first ellipse).

An additional analysis showed that these spikes were in phase with the CBF rhythm. This is illustrated in the last row of Figure 6.9, where the original LFP data is compared to the CBF which was previously bandpass filtered (0.8 and 1.1 Hz). The peaks of the spikes always occur around the minima of the oscillations. The peak-to-peak distances between the spikes correspond to multiples of the cycle length of the CBF activity (between 1-4 cycles).

Within the ANR project Multimodel during which the dataset was acquired (cf. Section 2.7), this finding found great interest. Before, only the causal relationship from LFP spikes to the hemodynamic response had been

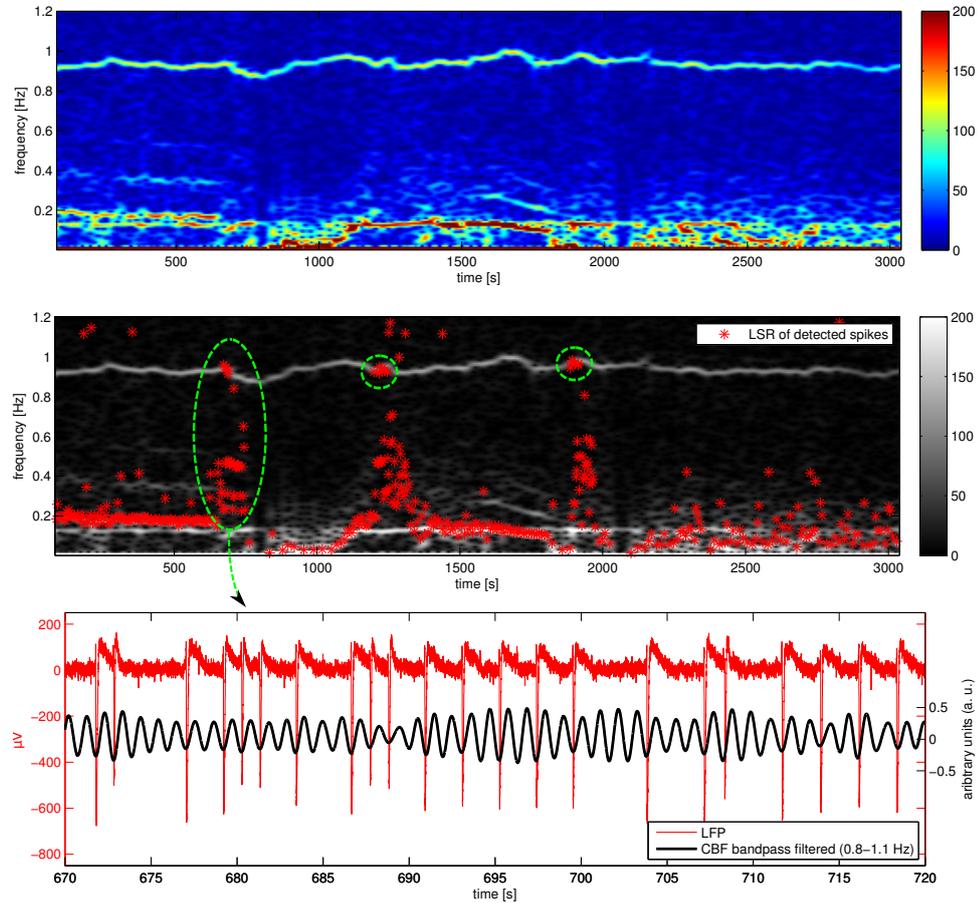


Figure 6.9: The figure illustrates the coupling between spiking rates and rhythmic activity in the CBF. **Top row:** The absolute values of the spectrogram of the CBF is shown for frequencies below 1.2 Hz. The highest intensities occur below 0.2 Hz. However, around 1 Hz, we can see an ongoing rhythmic activity across the entire time axis. **Middle row:** The local spiking rates (LSR) of the detected spikes are plotted on top of the spectrogram, which is shown in a gray scale for better visibility. Interestingly, during the three time periods of increased spiking activity around 750, 1300, and 1950 seconds, the fastest spikes appear to synchronize with the rhythmic activity in the CBF (green ellipses). In fact, around 700 seconds, we can even see clusters corresponding to the subharmonics of this activity, around 0.5, 0.33, and 0.25 seconds (first ellipse). **Bottom row:** This can also be observed in time domain when comparing the original LFP with the bandpass filtered CBF (0.8-1.1 Hz) in the time window 670-720 seconds. In fact, we observe that the spike peaks are clearly in phase with the CBF activity and occur with intervals between 1 and 4 cycles of this rhythm.

considered. However, the present finding suggests that there might also be a reversed causal relationship: The rhythmic activity in the CBF around 1 Hz remains relatively constant throughout the recording while the spikes alter their spiking rate to synchronize with this rhythm. Alternatively, this relationship could be induced by a third, latent variable.

In four of the five other datasets recorded in the same experimental setup in different rats, we could clearly see the rhythmic activity in the CBF described above. Its frequency varied across rats between 0.5 and 1.5 Hz. Unfortunately, the direct coupling between spiking rates and this activity could not be observed in the other rats. However, there generally appeared to be relationship between the spike density in a certain time interval and the energy of these oscillations.

As suggested by the experimentalists, in a future recording, it could be beneficial to directly measure the rat's respiration, in order to confirm the nature of the observed CBF activity. In addition, more empirical evidence is required to confirm the statistical relevance of the findings above.

Conclusion

In the present thesis, we addressed the challenges and benefits of adequately modeling variability in neuroelectrical signals. For this purpose, we developed a mathematical model that explicitly includes general types of signal variability, described through linear operators. The resulting method, adaptive waveform learning (AWL), provides a generic algorithm that can be specialized for different concrete settings. Thanks to the different notions of sparsity used in the AWL model, these specializations can be efficiently implemented through sparse coding techniques.

In particular, we studied two distinct settings for which we provided different AWL implementations: an epoched approach (E-AWL) for a set of short signal segments and a contiguous approach (C-AWL) which does not require prior epoching.

Evaluated on synthetic data, E-AWL proved a better performance than principal component analysis (PCA) and independent component analysis (ICA) in terms of waveform separation in the presence of latency jitter. Applied to real LFP recordings acquired in an anesthetized rat, E-AWL was able to clearly separate epileptiform spikes from an oscillatory artifact by exploiting its varying phase with respect to the spikes.

C-AWL was specialized for two spike learning algorithms: MC-Spike and AD-Spike. The representation learned on the LFP recording with MC-Spike provided a structural overview over the dataset by clustering spikes in a time-energy plot. They furthermore illustrated the different spike shapes throughout the data, which varied mainly in duration. AD-Spike led to a more compact representation by only using a single spike template with variable duration. In terms of spike detection, both methods showed robust results for signal-to-noise ratios above -20 dB and outperformed template matching when using a non-optimal template.

The insights obtained from analyzing the spike representations obtained with MC-Spike and AD-Spike then gave rise to study the relationships of

spiking rates and hemodynamic activity. This led to the interesting finding of spiking activity synchronized to the rhythmic activity in the cerebral blood flow (CBF) around 1 Hz in one of the studied rats. While this relationship could not be exactly reproduced in the other recorded rats, there appeared to be other couplings between this CBF rhythm and the spiking activity.

7.1 Strengths and weaknesses of AWL

The AWL model presented in Chapter 4 accounts for a broad range of signal variability through general linear operations on signal components. On the one hand, this allows the application of AWL to a variety of tasks. On the other hand, the generality bears the risk of lacking robustness, especially in the case of high noise levels. Hence, it is necessary to specialize AWL to the exact processing tasks it is applied to and include all available prior information into the learning process.

The incorporation of prior information is facilitated by the encapsulated structure of the generic AWL algorithm. For example, prior knowledge about waveform shapes can be directly integrated in the initialization step or the waveform updates. It is also possible to integrate elements of other methods in each step of the algorithm. For instance, performing a PCA or ICA after the waveform updates may provide for a better waveform separation in certain cases. Finally, the AWL framework can easily be extended to a multi-channel setting (see next section).

A major difficulty of AWL lies in its non-convexity. This is a common problem of dictionary learning methods and leads to a strong dependence of the results on the initializations used. Hence, it is necessary to carefully consider the initialization step. In this work, for instance, we found it beneficial to initialize learning on epoched datasets with Gaussian noise (cf. Chapter 5) in order to avoid biasing the results. However, when processing epileptiform spikes on long contiguous recordings (cf. Chapter 6), we initialized with a template to increase the detection accuracy.

When learning several waveform components with AWL, the number of components K normally has to be specified *a priori*. Although we provided a hierarchical approach which learns the waveform representations incrementally, this still requires to determine the number K_{max} of the maximal representation size. It would thus be desirable to have an automatic stopping criterion which determines the optimal representation sizes. However, due to the generality of the AWL framework, we found it difficult to formulate a global stopping criterion. Instead, we suggest that the optimal representation size should be chosen dependent on the application and the intended interpretation.

7.2 Future work

The generality of the AWL method allows its use for a variety of neuroelectrical signal processing tasks. While we could only focus on a few concrete settings in the scope of this thesis, we provide an overview over potential future work.

Application to different tasks

In the present work, we focussed on the thorough study of a single multi-modal dataset, which contained epileptiform spikes in the recorded LFP. This allowed us to see the different and complementary qualities of E-AWL and C-AWL. However, the analyzed data is only one example for a possible application of E-AWL and C-AWL. In future work, these algorithms could be applied to variety of neuroelectrical signal processing tasks, for example: learning of ERP components across epoched trials; detection and shape learning of transients such as sleep spindles or K-complexes; artifact rejection by exploiting different phases between artifacts and signals of interest (cf. Section 5.5).

More general deformations

The deformations in the concrete implementations of E-AWL and C-AWL were limited to latency shifts and linear dilations. In certain cases, a more general description of variability may be desired. For example, the linear dilations used in the AD-Spike model (cf. Section 6.4.2) were not able to completely explain the variability across different spike shapes. A possible generalization of dilations are non-linear warps which could be efficiently implemented by using dynamic time warping. In this case, instead of pre-defining the deformations, these warps could rather be learned together with the coefficients and the waveform shapes.

Multi-channel extension

AWL can also be extended to include several recording channels. As variability across channels typically differs from trial-to-trial variability, channels and trials should be modeled differently. For example, as described by the multi-channel dVCA model in Section 3.5.1, waveform latencies may vary across trials but are usually fixed across channels.

Concretely, in the current version of the E-AWL algorithm (Algorithm 3), the coefficient updates are performed separately across trials. In case of multiple channels, this update should be performed simultaneously on all channels of a specific trial to account for the isochronicity of the waveform occurrences across channels. This can be implemented similar to existing multi-channel versions of the matching pursuit algorithm (Béнар et al., 2009;

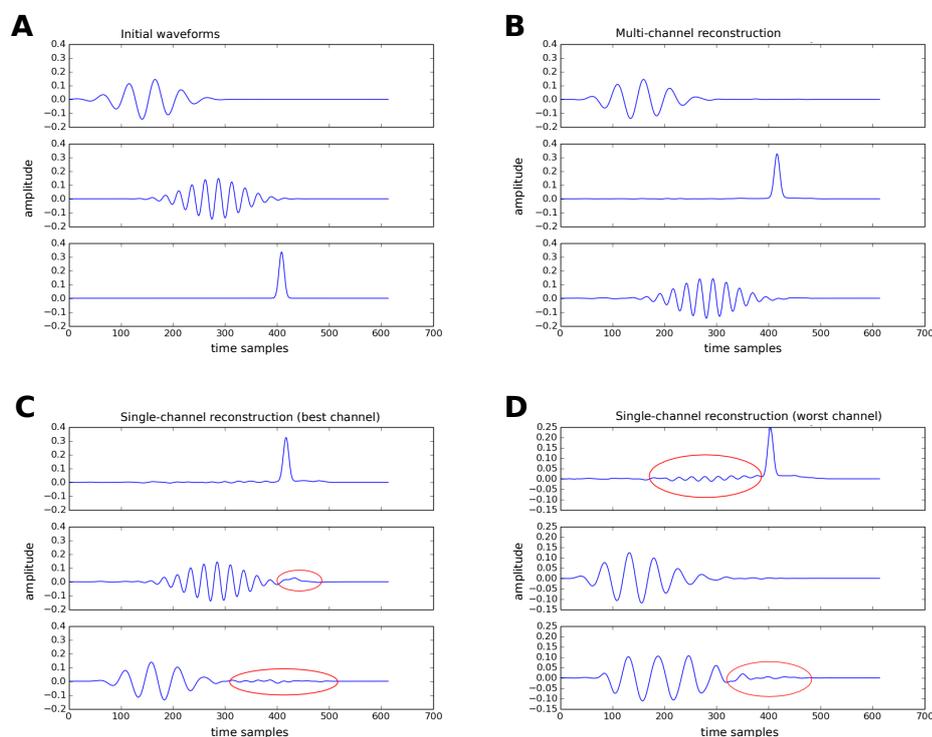


Figure 7.1: Waveforms reconstructed with single-channel and multi-channel learning approach from Papageorgakis (2014). The three initial waveforms are shown in (A). The multi-channel approach (B) recovered these waveforms almost perfectly. In contrast, when learning on each channel individually, the recovered waveforms still contained artifacts (shown in red) from other waveforms: (C) best channel, (D) worst channel.

Durka and Blinowska, 1995; Gribonval, 2003; Tropp et al., 2006) described in Section 3.6.1.

Preliminary results of multi-channel learning with E-AWL are shown in Figure 7.1. The waveforms learned with the multi-channel extension (B) almost perfectly recovered the original waveforms (A). Applying E-AWL to each of the six channels separately led to suboptimal results with artifacts, (C) and (D) show the best and the worst channel, respectively. The implementation, which included a modification of the LARS algorithm in E-AWL, and the simulations were conducted by Christos Papageorgakis during the final internship of his Master thesis (Papageorgakis, 2014) in our research team.

Spiking rates vs. CBF rhythms

The spike representations learned with AD-Spike and MC-Spike led to interesting findings concerning the coupling between spiking rates and rhythmic

activity in the cerebral blood flow (CBF) (cf. Section 6.5). Due to the current interest in neurovascular coupling (Blanchard et al., 2011; Sallet et al., 2015; Vanzetta et al., 2010; Voges et al., 2012), these findings could be of importance. However, more studies will be needed to confirm the statistical relevance of the findings. In addition, it has to be clarified if the origin of the observed CBF activity around 1 Hz is indeed caused by the respiration, as suggested by the experimentalists.

Conclusion en français

Dans cette thèse, nous avons abordé les défis et les gains que peut apporter la modélisation de la variabilité des signaux neuroélectriques. À cette fin, nous avons développé un modèle mathématique qui prend en compte différents types de variabilité des signaux, décrits par des opérateurs linéaires. La méthode résultante, *adaptive waveform learning* (AWL), fournit un algorithme générique qui peut être spécialisé pour des cas concrets. Grâce aux différentes notions de parcimonie utilisées dans le modèle AWL, ces spécialisations peuvent être implémentées de manière efficace par des techniques de codage parcimonieux.

En particulier, nous avons étudié deux cadres différents pour lesquels nous avons fourni différentes implémentations de AWL : une approche *époquée* (E-AWL) pour un ensemble de segments du signal, et une approche contiguë (C-AWL) qui ne nécessite pas de segmentation préalable.

Évalué sur des données synthétiques, E-AWL a montré une meilleure performance que l'analyse en composantes principales (ACP) et l'analyse en composantes indépendantes (ICA) en ce qui concerne la séparation des composantes du signal en présence de *jitter* sur les latences. Appliqué aux enregistrements réels du LFP acquis sur des rats anesthésiés, E-AWL était capable de séparer proprement les pointes épileptiformes des artefacts oscillatoires en exploitant la variabilité de la phase par rapport aux pointes.

C-AWL a été spécialisé en deux algorithmes d'apprentissage de pointes : MC-Spike et AD-Spike. Les représentations apprises sur les données LFP avec MC-Spike donnent un aperçu structurel du jeu de données en groupant les pointes dans un graphique temps-énergie. Par ailleurs, ces représentations illustrent les différentes formes de pointes à travers les données, qui varient principalement en durée. AD-Spike utilise une représentation plus compacte, en n'utilisant qu'un seul *template* de pointe avec une durée variable. Sur la détection des pointes, les deux méthodes ont montré des résultats robustes pour un rapport signal sur bruit au dessus de -20 dB et sont meilleures que l'algorithme *template matching* dans le cas où le *template* est sous-optimal.

L'analyse des représentations apprises avec MC-Spike et AD-Spike a ensuite motivé l'investigation de la relation entre l'activité de pointes et l'hémodynamique. Les résultats ont montré une synchronisation des pointes avec l'activité rythmique dans le flot sanguin cérébral (CBF) à environ 1 Hz dans un des rats étudiés. Alors que cette même relation n'est pas observable sur les autres rats, d'autres formes de couplages sont présentes entre le rythme dans le CBF et l'activité des pointes.

Points forts et faibles de AWL

Le modèle AWL présenté dans le Chapitre 4 prend en compte un large éventail de variabilités des signaux à travers des opérations linéaires générales sur les composantes du signal. D'un côté, ceci permet l'application de AWL à une grande variété de tâches. D'un autre côté, la généralité présente le risque de manquer de robustesse, particulièrement en cas de fort bruit. Par conséquent, il est nécessaire de précisément spécialiser AWL à la tâche de traitement à laquelle il est appliqué. De plus, il est important d'inclure toute l'information préalable dans le processus d'apprentissage.

L'incorporation de l'information préalable est facilitée par la structure encapsulée de l'algorithme AWL générique. Par exemple, des connaissances sur les formes d'onde peuvent être intégrées dans l'étape d'initialisation ou celle de mise à jour des formes d'onde. Par ailleurs, il est possible d'intégrer des éléments d'autres méthodes dans chaque étape de l'algorithme. Par exemple, en effectuant une ACP ou ICA après la mise à jour des formes d'onde, il est possible de mieux séparer les formes d'onde dans certains cas. Finalement, le cadre de AWL peut facilement être étendu au cas multicanaux (voir section suivante).

Une difficulté majeure de AWL consiste dans la non-convexité. Ceci est un problème commun aux méthodes d'apprentissage de dictionnaire et implique une forte dépendance des résultats à l'initialisation utilisée. Par conséquent, il est nécessaire de considérer l'étape d'initialisation soigneusement. Dans le présent travail, nous avons trouvé qu'il était avantageux d'initialiser l'apprentissage sur des données *époquées* avec du bruit gaussien (voir Chapitre 5) afin d'éviter de biaiser les résultats. Par contre, pour traiter les pointes épileptiformes dans un long enregistrement contigu (voir Chapitre 6), nous avons initialisé l'algorithme avec un *template* de pointe pour augmenter la précision de détection.

Pour apprendre plusieurs composantes du signal avec AWL, le nombre de composantes K doit normalement être précisé a priori. Bien que nous avons fourni une approche hiérarchique qui apprend les formes d'onde de manière incrémentale, il est quand-même nécessaire de spécifier le nombre K_{max} de la taille maximale des représentations. Il serait donc désirable d'avoir un critère d'arrêt automatique qui détermine la taille optimale de représen-

tation. En raison de la généralité du cadre de AWL, nous avons trouvé difficile de formuler un tel critère d'arrêt global. Nous proposons plutôt de choisir la taille optimale de représentation en fonction de l'application et l'interprétation prévue.

Travail futur

La généralité de AWL permet son usage pour une grande variété de tâches de traitement du signal neuroélectrique. Au lieu de seulement se focaliser sur les quelques applications concrètes étudiées dans cette thèse, nous fournissons un aperçu des possibilités de futurs travaux.

Application aux tâches différentes

Dans le présent travail, nous nous sommes concentrés sur l'étude exhaustive d'un seul jeu de données multimodales qui contient des pointes épileptiformes. Cette approche nous a permis de voir les qualités complémentaires de E-AWL et C-AWL. Par contre, les données analysées sont seulement un exemple d'application possible de E-AWL et C-AWL. Dans futurs travaux, ces algorithmes pourraient être appliqués à une variété de tâches de traitement du signal neuroélectrique, par exemple : l'apprentissage des composantes ERP à travers des essais expérimentaux; la détection et apprentissage des événements transitoires tels que des fuseaux de sommeil ou des complexes K; la séparation des artefacts en exploitant leurs différentes phases par rapport au signal (voir Section 5.5).

Déformations plus générales

Les déformations dans les implémentations concrètes de E-AWL et C-AWL sont limitées aux translations et dilatations linéaires. Dans certains cas, une description plus générale de la variabilité peut être désirée. Par exemple, les dilatations linéaires utilisées dans le modèle AD-Spike (voir Section 6.4.2) ne sont pas capables de complètement expliquer la variabilité à travers les différentes formes des pointes. Des *warps* non-linéaires représentent une généralisation qui pourrait être implémentée efficacement par *dynamic time warping*. Dans ce cas, les *warps* pourraient être appris en même temps que les coefficients et les formes d'onde, au lieu d'utiliser des déformations prédéfinies.

Extension multicanaux

AWL peut aussi être étendu pour inclure plusieurs canaux d'enregistrement. Comme la variabilité à travers des canaux est typiquement différente de la variabilité à travers les essais expérimentaux, les canaux et les essais doivent

être modélisés différemment. Par exemple, comme décrit par le modèle dVCA multicanaux à la Section 3.5.1, les latences des composantes du signal peuvent varier à travers les essais mais restent fixées à travers les canaux.

Concrètement, dans la version actuelle de l'algorithme E-AWL (Algorithme 3), la mise à jour des coefficients est effectuée séparément à travers les essais. En cas de plusieurs canaux, cette mise à jour doit être effectuée simultanément sur tous les canaux d'un essai spécifique pour prendre en compte la simultanéité des formes d'onde à travers les canaux. Ceci peut être implémenté de manière similaire aux versions multicanaux existantes de *matching pursuit* (Bénar et al., 2009; Durka and Blinowska, 1995; Gribonval, 2003; Tropp et al., 2006) décrites dans la Section 3.6.1.

Des résultats préliminaires de l'apprentissage multicanaux avec E-AWL sont montrés dans la Figure 7.2. Les formes d'onde apprises avec l'extension multicanaux (B) récupèrent les formes originales (A) presque parfaitement. L'application de E-AWL à chacun des six canaux individuels conduit à des résultats sous-optimaux avec des artefacts : (C) et (D) montrent les reconstructions respectivement sur le meilleur et le pire des canaux. L'implémentation, qui comprend une modification de l'algorithme LARS dans E-AWL, et les simulations ont été faites par Christos Papageorgakis pendant son stage final de master (Papageorgakis, 2014) dans notre équipe de recherche.

Activité des pointes et rythmes CBF

Les représentations des pointes apprises avec AD-Spike et MC-Spike ont mené à la découverte d'un couplage entre l'activité des pointes et l'activité rythmique dans le flot sanguin cérébral (CBF) (voir Section 6.5). En raison de l'intérêt actuel pour le couplage neurovasculaire (Blanchard et al., 2011; Sallet et al., 2015; Vanzetta et al., 2010; Voges et al., 2012), ces découvertes pourraient être importantes. Par contre, des études supplémentaires seront nécessaires pour confirmer la pertinence statistique de ces résultats. De plus, il est important de clarifier si l'activité autour de 1 Hz observée dans le CBF est effectivement causée par la respiration, comme suggéré par les expérimentateurs.

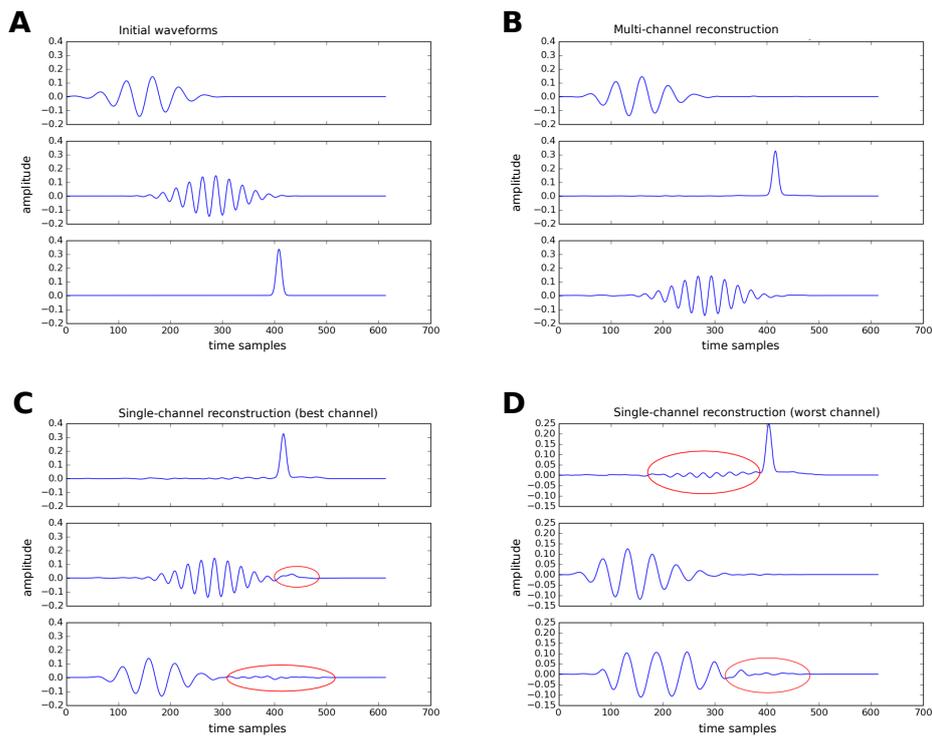


Figure 7.2: Formes d'onde reconstruites avec des approches d'apprentissage de dictionnaire sur des canaux individuels et l'apprentissage multicanaux de [Papageorgakis \(2014\)](#). Trois formes d'onde initiales are montrées dans (A). L'approche multicanaux (B) récupère les formes d'onde presque parfaitement. Par contre, l'apprentissage sur des canaux individuels produit des formes d'onde qui contiennent des artefacts (en rouge) des autres composantes : (C) meilleur canal, (D) pire canal.

Appendix A

Implementation of MP and LARS

In this appendix, we describe the sparse coding algorithms matching pursuit (MP) and least angle regression (LARS) in more detail. We also provide comments on their concrete implementations for E-AWL (Algorithm 3) and C-AWL (Algorithm 4). The code for the implementations used for the experiments in this thesis is available online at <https://github.com/hitziger/AWL>.

MP and LARS are designed to provide a sparse representation $\mathbf{a} \in \mathbb{R}^K$ of some signal $\mathbf{x} \in \mathbb{R}^N$ over a dictionary $\mathbf{D} \in \mathbb{R}^{N \times K}$ (cf. Section 3.6.1). Matching pursuit follows a straight forward approach to the problem by greedily searching for active atoms. The simplicity of the algorithm allows for a very efficient implementation. However, an optimal solution can only be ensured for orthogonal dictionaries. LARS on the other hand takes into account the correlations between non-orthogonal atoms and provides an exact solution to the Lasso problem (3.8). However, LARS comes with higher computational costs than MP and can become infeasible in the case of large dictionaries.

Both algorithms share the property of selecting *active* (non-zero) coefficients in subsequent steps. This property is very useful for the algorithms presented in this thesis, as it allows to easily ensure the additional constraints exclusivity (4.4) and non-negativity of coefficients (cf. Section 4.4.2). Furthermore, both algorithms do not explicitly require the dictionary \mathbf{D} or the signal \mathbf{x} as inputs, but only the correlations $\mathbf{D}^t \mathbf{x}$ and covariances $\mathbf{D}^t \mathbf{D}$. These can be efficiently calculated through fast Fourier transform since the dictionaries used in the methods E-AWL and C-AWL are shift-invariant.

Matching pursuit

For convenience, we restate the sparse coding problem (3.7) from Section 3.6.1 in matrix-vector notation,

$$\min_{\mathbf{a}} \|\mathbf{x} - \mathbf{D}\mathbf{a}\|_2^2, \quad \text{s.t. } \|\mathbf{a}\|_0 \leq L,$$

where the dictionary \mathbf{D} contains the atoms $\{\mathbf{d}_k\}_{k=1}^K$ as column vectors. Matching pursuit now greedily searches for atoms \mathbf{d}_k that have maximal correlation with the signal \mathbf{x} . Starting with initial correlations $\mathbf{y}^{(0)} = \mathbf{D}^t \mathbf{x}$, the first *active* (i.e., non-zero) coefficient is given by the maximal absolute value of $\mathbf{y}^{(0)}$, say $a_{k_0} = y_{k_0}$. Now, the contribution of the active atom is subtracted from the signal, giving the residual $\mathbf{r}^{(1)} = \mathbf{x} - a_{k_0} \mathbf{d}_{k_0}$. The atom that has maximal correlation with $\mathbf{r}^{(1)}$ will be selected next, and so on, until the desired number of active coefficients is found. Before giving the concrete implementation, we note that it is not necessary to explicitly calculate the residuals $\mathbf{r}^{(i)}$ since only the correlations $\mathbf{y}^{(i)} = \mathbf{D}^t \mathbf{r}^{(i)}$ are needed to select the next atoms. They can be written as

$$\begin{aligned} \mathbf{y}^{(i)} &= \mathbf{D}^t \mathbf{r}^{(i)} = \mathbf{D}^t \mathbf{r}^{(i-1)} - a_{k_i} \mathbf{D}^t \mathbf{d}_{k_i} \\ &= \mathbf{y}^{(i-1)} - a_{k_i} \mathbf{g}_{k_i}, \end{aligned}$$

where \mathbf{g}_k denotes the k -th column of the covariance matrix $\mathbf{G} \equiv \mathbf{D}^t \mathbf{D}$. The calculation details are now summarized in Algorithm 7. Note that the correlations $\mathbf{y}^{(0)} = \mathbf{D}^t \mathbf{x}$ and $\mathbf{G} \equiv \mathbf{D}^t \mathbf{D}$ need to be calculated only once in the beginning, such that the iterative selection can be performed very fast. If the dictionary \mathbf{D} is known beforehand, the covariance matrix \mathbf{G} can be precomputed, which is especially useful if several signals \mathbf{x}_m are encoded over the same \mathbf{D} .

Algorithm 7 MP

Input: $\mathbf{D} \in \mathbb{R}^{N \times K}$, $\mathbf{x} \in \mathbb{R}^N$, $L \in \mathbb{N}$

- 1: **Initialize**
- 2: $\mathbf{a} = (0, \dots, 0)^t \in \mathbb{R}^K$;
- 3: $\mathbf{y}^{(0)} = \mathbf{D}^t \mathbf{x}$;
- 4: $\mathbf{G} = \mathbf{D}^t \mathbf{D}$;
- 5: **for** $i = 1$ to L **do**
- 6: $k = \operatorname{argmax}(|\mathbf{y}^{(i-1)}|)$;
- 7: $a_k \leftarrow y_k$;
- 8: $\mathbf{y}^{(i)} \leftarrow \mathbf{y}^{(i-1)} - a_k \mathbf{g}_k$;
- 9: **end for**

Output: $\mathbf{a} \in \mathbb{R}^K$

In this thesis, we implemented matching pursuit for the coefficient up-

dates in C-AWL (Algorithm 4). In this case, \mathbf{D} consisted of translated and dilated waveforms $\{\mathbf{d}_k^{pq}\}$, resulting in approximately KQN atoms, with K the number of waveforms, Q the number of dilations, and N the sample size of \mathbf{x} . A naive calculation of $\mathbf{G} = \mathbf{D}^t \mathbf{D}$ would require $O(K^2 Q^2 N^3)$ multiplications, which is infeasible for long signal sizes N . However, by exploiting the redundancies in \mathbf{G} and efficiently calculating cross-correlations through the fast Fourier transform (FFT), the complexity could be reduced to $O(K^2 Q^2 n \log(n))$, with $n \ll N$ denoting the sample size of the waveforms \mathbf{d}_k . The calculation of $\mathbf{y}^{(0)} = \mathbf{D}^t \mathbf{x}$ could also be accelerated through FFT, however, this only holds for cases where $const \cdot \log(N) < n$. Note that choosing a high resolution across dilation factors, significantly increases the computational complexity. In fact, dilations were implemented through linear interpolation and could not be handled as efficiently as translations. After calculation of \mathbf{G} and $\mathbf{y}^{(0)}$, the most time-consuming part of the pursuit algorithm consisted in determining the maximum in line 6. This could be speeded up in our implementation by initially arranging $\mathbf{y}^{(0)}$ in descending order.

The additional constraints in the coefficient updates of Algorithm 4 were implemented as follows: For non-negativity of coefficients, we only selected atoms that were positively correlated with the residual signal, which results in replacing $|\mathbf{y}^{(i-1)}|$ with $\mathbf{y}^{(i-1)}$ in line 6 of Algorithm 7. After every MP iteration, we determined all coefficients whose activation would violate the exclusivity constraint (6.3). The corresponding entries in \mathbf{y} were then set to $-\infty$.

More details on efficient MP implementations for shift-invariant dictionaries can be found in Krstulovic and Gribonval (2006). The authors also provide the open source toolbox MPTK with Matlab interface¹.

Lasso and LARS

In this section, we provide some basic insights into the Lasso problem and illustrate the concepts of the LARS algorithm. This analysis roughly follows the lines of the compact description in Mairal (2010); the complete LARS algorithm can be found in Efron et al. (2004).

Writing the Lasso problem (3.8) from Section 3.6.1 in matrix-vector notation yields

$$\min_{\mathbf{a} \in \mathbb{R}^K} \left[f(\mathbf{a}) = \|\mathbf{x} - \mathbf{D}\mathbf{a}\|_2^2 + \lambda \|\mathbf{a}\|_1 \right], \quad (\text{A.1})$$

for some $\lambda \geq 0$. Now the atoms \mathbf{d}_k are the columns of the matrix \mathbf{D} , $\mathbf{a} = (a_1, \dots, a_K)^t$ is the coefficient vector, and $\|\mathbf{a}\|_1 = \sum_k |a_k|$ denotes the l_1 -norm. First, we investigate the properties of a solution vector \mathbf{a} . Since the absolute value function is not differentiable at 0, we consider the partial

¹<http://mptk.irisa.fr/>

subdifferentials

$$\frac{\partial}{\partial a_k} f = \{c \in \mathbb{R} \mid f(a') - f(a_k) \geq c(a' - a_k) \text{ for all } a' \in \mathbb{R}\}.$$

A necessary and sufficient optimality condition is that 0 is contained in the partial subdifferential for each k , which leads to

$$\mathbf{a}^* = \operatorname{argmin} f(\mathbf{a}) \Leftrightarrow \begin{cases} \text{if } a_k^* \neq 0 : & \mathbf{d}_k^t(\mathbf{x} - \mathbf{D}\mathbf{a}^*) = \lambda \operatorname{sgn}(a_k^*) \\ \text{if } a_k^* = 0 : & |\mathbf{d}_k^t(\mathbf{x} - \mathbf{D}\mathbf{a}^*)| \leq \lambda, \end{cases} \quad (\text{A.2})$$

where $\operatorname{sgn}(\cdot)$ denotes the signum function. This characterization of the solution provides two interesting insights into the problem:

1. The parameter λ serves as a correlation threshold: if atom \mathbf{d}_k correlates less than λ with the residual data $(\mathbf{x} - \mathbf{D}\mathbf{a}^*)$, its coefficient is set to zero.
2. If we know the indices and the signs of the *active* (i.e., non-zero) coefficients, say I^+ and $\operatorname{sgn}(\mathbf{a}_{I^+}^*)$, respectively, the solution can be calculated explicitly as¹

$$\mathbf{a}_{I^+}^* = \left(\mathbf{D}_{I^+}^t \mathbf{D}_{I^+}\right)^{-1} \left(\mathbf{D}_{I^+}^t \mathbf{x} - \lambda \operatorname{sgn}(\mathbf{a}_{I^+}^*)\right). \quad (\text{A.3})$$

Now, the crucial idea of LARS is to exploit the regularization path

$$\lambda \mapsto \mathbf{a}^*(\lambda),$$

that is, the behavior of the solution $\mathbf{a}^*(\lambda)$ for changing λ . By (A.2) and (A.3) it becomes clear that this path is linear, except for values of λ , for which the solution $\mathbf{a}^*(\lambda)$ contains a *critical entry* a_k^* , that is,

$$a_k^* = 0 \quad \text{and} \quad |\mathbf{d}_k^t(\mathbf{x} - \mathbf{D}\mathbf{a}^*)| < \lambda. \quad (\text{A.4})$$

It can be shown (Efron et al., 2004) that the number of values of λ for which $\mathbf{a}^*(\lambda)$ contains such a critical entry is finite. In the following, we will assume the *one-at-a-time* condition, meaning that for any $\lambda > 0$, the solution $\mathbf{a}^*(\lambda)$ has at most one critical entry. Without this condition, the problem is still solvable but becomes more subtle, details are given in Efron et al. (2004).

Hence, that regularization path is piecewise linear. We can now follow this path, starting at $\lambda = \max_k (|\mathbf{d}_k^t(\mathbf{x} - \mathbf{D}\mathbf{a}^*)|)$, for which the solution is trivial. Now, while decreasing λ , the solution $\mathbf{a}^*(\lambda)$ is always given through

¹For simplicity, we assume invertibility of $\mathbf{D}_{I^+}^t \mathbf{D}_{I^+}$. As noted in Mairal (2010), this assumption can be relaxed when including a small l_2 -prior in (A.1) which leads to the elastic net problem Zou and Hastie (2005).

(A.3), provided that we keep track of the active index set I^+ and the signs $\text{sgn}(\mathbf{a}_{I^+}^*)$. These can change whenever a coefficient a_k^* becomes critical: If a_k^* was zero before, it is now *activated*, with $\text{sgn}(a_k^*) = \text{sgn}(\mathbf{d}_k^t(\mathbf{x} - \mathbf{D}\mathbf{a}^*))$, and we add k to I^+ . However, if a_k^* was already active, it is now *deactivated*, and we remove k from I^+ . The algorithm stops when λ reaches the desired regularization parameter.

The complete LARS implementation is rather long and we refer the reader to [Efron et al. \(2004\)](#). However, we note a few aspects which are important for our implementation of the coefficient updates in E-AWL (Algorithm 3): If we follow the regularization path until $\lambda = 0$, LARS provides the exact solution to the unregularized least squares problem (5.2). Moreover, the additional constraints (5.3) and (5.4) can easily be enforced by making slight modifications to the activations and deactivations during the regularization path: After every activation of a coefficient a_k^* , we mark all inactive coefficients that cannot be simultaneously active with a_k^* due to the exclusivity constraint (5.3). These coefficients are excluded from later activation. Likewise, in case of the deactivation of an active coefficient a_k^* , we verify which of the marked coefficients becomes admissible again for activation. Furthermore, for non-negativity (5.4), we only activate a critical coefficient a_k^* if $\text{sgn}(\mathbf{d}_k^t(\mathbf{x} - \mathbf{D}\mathbf{a}^*)) = 1$, otherwise it remains zero.

As in MP, the actual signal \mathbf{x} and dictionary \mathbf{D} are not required for the calculations. LARS only uses the correlations $\mathbf{y} = \mathbf{D}^t\mathbf{x}$ and covariances $\mathbf{G} = \mathbf{D}^t\mathbf{D}$, which can be efficiently calculated in the shift-invariant case.

Our implementation of the coefficient updates in (Algorithm 3) uses the very efficient LARS implementation of the toolbox SPAMS¹, to which we made the changes described above in order to ensure the constraints (5.3) and (5.4).

¹<http://spams-devel.gforge.inria.fr/>

Appendix **B**

Publications of the Author

Journal papers:

S. Hitziger, M. Clerc, S. Saillet, C. Bénar, and T. Papadopoulo. *Adaptive waveform learning - application to single- and multi-modal neurological data*. In preparation.

S. Saillet, A. I. Ivanov, P. Quilichini, A. Ghestem, B. Giusiano, S. Hitziger, I. Vanzetta, C. Bernard, and C. G. Bénar. *Interneurons contribute to the hemodynamic/metabolic response to epileptiform discharges*. Submitted to Journal of Neurophysiology (second revision), 2015.

Conference papers:

S. Hitziger, M. Clerc, A. Gramfort, S. Saillet, C. Bénar, T. Papadopoulo. *Electro-metabolic coupling investigated with jitter invariant dictionary learning*. In Proceedings of International Human Brain Mapping Conference (HBM), 2014.

S. Hitziger, M. Clerc, A. Gramfort, S. Saillet, C. Bénar, and T. Papadopoulo. *Jitter-adaptive dictionary learning-application to multi-trial neuroelectric signals*. In Proceedings of the International Conference on Learning Representations (ICLR), 2013.

Bibliography

- M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, Nov. 2006. [50](#)
- F. A. Azevedo, L. R. Carvalho, L. T. Grinberg, J. M. Farfel, R. E. Ferretti, R. E. Leite, R. Lent, S. Herculano-Houzel, et al. Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain. *Journal of Comparative Neurology*, 513(5): 532–541, 2009. [16](#)
- A. D. Back and A. S. Weigend. A first application of independent component analysis to extracting structure from stock returns. *International journal of neural systems*, 8(04):473–484, 1997. [43](#)
- D. L. Bailey, D. W. Townsend, P. E. Valk, and M. N. Maisey. *Positron emission tomography*. Springer, 2005. [22](#)
- S. Baillet, J. C. Mosher, and R. M. Leahy. Electromagnetic brain mapping. *Signal Processing Magazine, IEEE*, 18(6):14–30, 2001. [17](#)
- Q. Barthélemy, A. Larue, A. Mayoue, D. Mercier, and J. I. Mars. Shift & 2d rotation invariant sparse coding for multivariate signals. *Signal Processing, IEEE Transactions on*, 60(4):1597–1611, 2012. [52](#)
- Q. Barthélemy, C. Gouy-Pailler, Y. Isaac, A. Souloumiac, A. Larue, and J. I. Mars. Multivariate temporal dictionary learning for eeg. *Journal of neuroscience methods*, 215(1):19–28, 2013. [52](#)
- M. S. Bartlett. *Face image analysis by unsupervised learning*. Springer, 2001, chapter 2. [43](#)
- M. Bastiaansen, A. Mazaheri, and O. Jensen. Beyond erps: oscillatory neuronal dynamics. *The Oxford handbook of event-related potential components*, pages 31–50, 2012. [56](#)

- A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009. [47](#)
- A. J. Bell and T. J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural computation*, 7(6):1129–1159, 1995. [42](#)
- C. Bénar, T. Papadopoulo, B. Torrèsani, and M. Clerc. Consensus matching pursuit for multi-trial eeg signals. *Journal of neuroscience methods*, 180(1):161–170, 2009. [47](#), [48](#), [111](#), [118](#)
- H. Berger. Über das elektrenkephalogramm des menschen. *European Archives of Psychiatry and Clinical Neuroscience*, 87(1):527–570, 1929. [18](#)
- S. Blanchard, T. Papadopoulo, C.-G. Bénar, N. Voges, M. Clerc, H. Benali, J. Warnking, O. David, and F. Wendling. Relationship between flow and metabolism in bold signals: insights from biophysical models. *Brain topography*, 24(1):40–53, 2011. [29](#), [113](#), [118](#)
- T. Blumensath and M. Davies. Sparse and shift-invariant representations of music. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14(1):50–57, 2006. [52](#)
- A. Cabasson and O. Meste. Time delay estimation: A new insight into the woody’s method. *Signal Processing Letters, IEEE*, 15:573–576, 2008. [38](#)
- J.-F. Cardoso. High-order contrasts for independent component analysis. *Neural computation*, 11(1):157–192, 1999. [42](#)
- J.-F. Cardoso and A. Souloumiac. Blind beamforming for non-gaussian signals. In *IEE Proceedings F (Radar and Signal Processing)*, volume 140, pages 362–370. IET, 1993. [42](#)
- A. Chambolle, R. De Vore, N. Lee, and B. Lucier. Nonlinear wavelet image processing: Variational problems, compression, and noise removal through wavelet shrinkage. *Image Processing, IEEE Transactions on*, 7(3):319–335, 1998. [47](#)
- D. Cohen. Magnetoencephalography: evidence of magnetic fields produced by alpha-rhythm currents. *Science*, 161(3843):784–786, 1968. [21](#)
- P. Comon. Independent component analysis, a new concept? *Signal processing*, 36(3):287–314, 1994. [42](#)

-
- R. Coppola, R. Tabor, and M. S. Buchsbaum. Signal to noise ratio and response variability measurements in single trial evoked potentials. *Electroencephalography and clinical neurophysiology*, 44(2):214–222, 1978. [2](#), [10](#)
- I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on pure and applied mathematics*, 57(11):1413–1457, 2004. [47](#)
- G. D. Dawson. A summation technique for the detection of small evoked potentials. *Electroencephalography and clinical neurophysiology*, 6:65–84, 1954. [2](#), [6](#), [9](#), [13](#), [24](#), [34](#), [35](#)
- L. De Gennaro and M. Ferrara. Sleep spindles: an overview. *Sleep medicine reviews*, 7(5):423–440, 2003. [26](#)
- L. Deecke, B. Grözinger, and H. Kornhuber. Voluntary finger movement in man: cerebral potentials and theory. *Biological cybernetics*, 23(2):99–119, 1976. [24](#)
- A. Delorme, T. Sejnowski, and S. Makeig. Enhanced detection of artifacts in eeg data using higher-order statistics and independent component analysis. *Neuroimage*, 34(4):1443–1449, 2007. [43](#)
- A. A. Dingle, R. D. Jones, G. J. Carroll, and W. Fright. A multistage system to detect epileptiform activity in the eeg. *Biomedical Engineering, IEEE Transactions on*, 40(12):1260–1268, 1993. [103](#)
- E. Donchin, W. Ritter, W. C. McCallum, et al. Cognitive psychophysiology: The endogenous components of the erp. *Event-related brain potentials in man*, pages 349–411, 1978. [24](#)
- P. Durka and K. Blinowska. Analysis of eeg transients by means of matching pursuit. *Annals of biomedical engineering*, 23(5):608–611, 1995. [48](#), [112](#), [118](#)
- P. Durka, A. Matysiak, E. Montes, P. Sosa, and K. Blinowska. Multichannel matching pursuit and eeg inverse solutions. *Journal of neuroscience methods*, 148(1):49–59, 2005. [48](#)
- B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004. [47](#), [61](#), [123](#), [125](#)
- M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *Image Processing, IEEE Transactions on*, 15(12):3736–3745, 2006. [50](#)

- K. Engan, S. Aase, and J. Hakon Husoy. Method of optimal directions for frame design. In *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, volume 5, pages 2443–2446. IEEE, 1999. [50](#)
- H. Gibbons and J. Stahl. Response-time corrected averaging of event-related potentials. *Clinical Neurophysiology*, 118(1):197–208, 2007. [40](#)
- A. Gibson, J. Hebden, and S. R. Arridge. Recent advances in diffuse optical imaging. *Physics in medicine and biology*, 50(4):R1, 2005. [22](#)
- A. Gramfort. *Mapping, timing and tracking cortical activations with MEG and EEG: Methods and application to human vision*. PhD thesis, PhD thesis, 2009. [21](#)
- A. Gramfort, R. Keriven, and M. Clerc. Graph-based variability estimation in single-trial event-related neural responses. *Biomedical Engineering, IEEE Transactions on*, 57(5):1051–1061, 2010. [38](#)
- R. Gribonval. Piecewise linear source separation. In *Optical Science and Technology, SPIE's 48th Annual Meeting*, pages 297–310. International Society for Optics and Photonics, 2003. [48](#), [112](#), [118](#)
- D. M. Groppe, S. Makeig, and M. Kutas. Independent component analysis of event-related potentials. *Cognitive science online*, 6(1):1–44, 2008. [43](#)
- R. Grosse, R. Raina, H. Kwong, and A. Y. Ng. Shift-Invariant Sparse Coding for Audio Classification. In *Uncertainty in Artificial Intelligence*, pages 149–158. [52](#)
- J. T. Gwin, K. Gramann, S. Makeig, and D. P. Ferris. Removal of movement artifact from high-density eeg recorded during walking and running. *Journal of neurophysiology*, 103(6):3526–3534, 2010. [43](#)
- B. Hamner, R. Chavarriaga, and J. d. R. Millán. Learning dictionaries of spatial and temporal eeg primitives for brain-computer interfaces. In *Workshop on Structured Sparsity: Learning and Inference, ICML 2011*, number EPFL-CONF-166740, 2011. [50](#)
- K. D. Harris, D. A. Henze, J. Csicsvari, H. Hirase, and G. Buzsáki. Accuracy of tetrode spike separation as determined by simultaneous intracellular and extracellular measurements. *Journal of neurophysiology*, 84(1):401–414, 2000. [20](#)
- S. Hitziger, M. Clerc, A. Gramfort, S. Sallet, C. Bénar, and T. Papadopoulo. Jitter-adaptive dictionary learning-application to multi-trial neuroelectric signals. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2013. [29](#)

- R. Horvath. Variability of cortical auditory evoked response. *J Neurophysiol*, 32(6):1056–1063, 1969. [2](#), [10](#)
- A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *Neural Networks, IEEE Transactions on*, 10(3):626–634, 1999. [42](#), [74](#)
- A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4):411–430, 2000. [43](#)
- A. Hyvärinen, P. Ramkumar, L. Parkkonen, and R. Hari. Independent component analysis of short-time fourier transforms for spontaneous eeg/meg analysis. *Neuroimage*, 49(1):257–271, 2010. [43](#)
- M. Ihrke, H. Schrobsdorff, and J. M. Herrmann. Recurrence-based synchronization of single trials for eeg-data analysis. In *Intelligent Data Engineering and Automated Learning-IDEAL 2009*, pages 118–125. Springer, 2009. [40](#)
- F. Itakura. Minimum prediction residual principle applied to speech recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 23(1):67–72, 1975. [38](#), [39](#)
- M. G. Jafari and M. D. Plumbley. Speech denoising based on a greedy adaptive dictionary algorithm. In *European Signal Processing Conference (EUSIPCO)*, 2009. [50](#)
- E. R. John, D. S. Ruchkin, and J. J. Vidal. Measurement of event-related potentials. *Event-Related Brain Potentials in Man, Academic Press, New York*, pages 93–138, 1978. [37](#)
- P. Jost, S. Lesage, P. Vandergheynst, and R. Gribonval. Learning redundant dictionaries with translation invariance property: the motif algorithm. Technical report, IEEE, 2005. [52](#)
- P. Jost, P. Vandergheynst, S. Lesage, and R. Gribonval. Motif: an efficient algorithm for learning translation invariant dictionaries. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 5, pages V–V. IEEE, 2006. [52](#)
- T. Jung, S. Makeig, M. Westerfield, J. Townsend, E. Courchesne, and T. Sejnowski. Analysis and visualization of single-trial event-related potentials. *Human brain mapping*, 14(3):166–185, 2001. [2](#), [10](#), [27](#)
- T.-P. Jung, S. Makeig, C. Humphries, T.-W. Lee, M. J. Mckeown, V. Iragui, and T. J. Sejnowski. Removing electroencephalographic artifacts by blind source separation. *Psychophysiology*, 37(02):163–178, 2000. [43](#), [44](#)

- E. J. Keogh and M. J. Pazzani. Derivative dynamic time warping. In *SDM*, volume 1, pages 5–7. SIAM, 2001. [39](#)
- M. A. Kiskey and G. L. Gerstein. Trial-to-trial variability and state-dependent modulation of auditory-evoked responses in cortex. *The Journal of neuroscience*, 19(23):10451–10460, 1999. [2](#), [10](#), [27](#)
- K. H. Knuth, A. S. Shah, W. A. Truccolo, M. Ding, S. L. Bressler, and C. E. Schroeder. Differentially variable component analysis: identifying multiple evoked components using trial-to-trial variability. *Journal of neurophysiology*, 95(5):3257–3276, 2006. [45](#)
- H. H. Kornhuber and L. Deecke. Hirnpotentialänderungen bei willkürbewegungen und passiven bewegungen des menschen: Bereitschaftspotential und refferente potentiale. *Pflüger's Archiv für die gesamte Physiologie des Menschen und der Tiere*, 284(1):1–17, 1965. [24](#)
- S. Krstulovic and R. Gribonval. MPTK: Matching Pursuit made tractable. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 3, pages III–496 – III–499, Toulouse, France, May 2006. [123](#)
- M. Kutas and A. Dale. Electrical and magnetic readings of mental functions. *Cognitive neuroscience*, 1974242, 1997. [25](#)
- T. Lagerlund, F. Sharbrough, and N. Busacker. Spatial filtering of multi-channel electroencephalographic recordings through principal component analysis by singular value decomposition. *Journal of Clinical Neurophysiology*, 14(1):73–82, 1997. [42](#)
- P. G. Larsson, J. Wilson, and O. Eeg-Olofsson. A new method for quantification and assessment of epileptiform activity in eeg with special reference to focal nocturnal epileptiform activity. *Brain topography*, 22(1):52–59, 2009. [103](#)
- N. K. Logothetis. The underpinnings of the bold functional magnetic resonance imaging signal. *The Journal of Neuroscience*, 23(10):3963–3971, 2003. [20](#)
- S. J. Luck. Event-related potentials. *APA handbook of research methods in psychology*, 1:523–546, 2012. [25](#)
- S. J. Luck. *An introduction to the event-related potential technique*. MIT press, 2014. [25](#)
- J. MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. California, USA, 1967. [28](#)

- B. Maill e, S. Lesage, R. Gribonval, F. Bimbot, P. Vandergheynst, et al. Shift-invariant dictionary learning for sparse representations: extending k-svd. In *16th European Signal Processing Conference (EUSIPCO'08)*, 2008. [52](#)
- B. Maill e, R. Gribonval, F. Bimbot, M. Lemay, P. Vandergheynst, and J.-M. Vesin. Dictionary learning for the sparse modelling of atrial fibrillation in eeg signals. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 465–468. IEEE, 2009. [50](#)
- J. Mairal. *Sparse coding for machine learning, image processing and computer vision*. PhD thesis,  cole normale sup rieure de Cachan-ENS Cachan, 2010. [47](#), [123](#), [124](#)
- J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. *arXiv preprint arXiv:0809.3083*, 2008. [50](#)
- J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online Learning for Matrix Factorization and Sparse Coding. 11:19–60, Aug. 2009. [50](#)
- S. Makeig, A. Bell, T. Jung, T. Sejnowski, et al. Independent component analysis of electroencephalographic data. *Advances in neural information processing systems*, pages 145–151, 1996. [43](#)
- S. Mallat. *A wavelet tour of signal processing*. Academic press, 1999. [46](#), [48](#)
- S. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *Signal Processing, IEEE Transactions on*, 41(12):3397–3415, 1993. [47](#), [61](#)
- N. Marwan, M. Thiel, and N. Nowaczyk. Cross recurrence plot based synchronization of time series. *arXiv preprint physics/0201062*, 2002. [40](#)
- C. M. Michel and M. M. Murray. Towards the utilization of eeg as a brain imaging tool. *Neuroimage*, 61(2):371–385, 2012. [34](#)
- J. M cks, W. K hler, T. Gasser, and D. T. Pham. Novel approaches to the problem of latency jitter. *Psychophysiology*, 25(2):217–226, 1988. [38](#)
- M. M rup, M. N. Schmidt, and L. K. Hansen. Shift invariant sparse coding of image and music data. *Submitted to Journal of Machine Learning Research*, 2008. [52](#)
- S. Ogawa, T.-M. Lee, A. S. Nayak, and P. Glynn. Oxygenation-sensitive contrast in magnetic resonance image of rodent brain at high magnetic fields. *Magnetic resonance in medicine*, 14(1):68–78, 1990. [22](#)
- E. Olivi. *Coupling of numerical methods for the forward problem in Magneto- and Electro-Encephalography*. PhD thesis, Universit  Nice Sophia Antipolis, 2011. [23](#)

BIBLIOGRAPHY

- B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: a strategy employed by V1? *Vision research*, 37(23):3311–25, Dec. 1997. [50](#)
- J. Onton, M. Westerfield, J. Townsend, and S. Makeig. Imaging human eeg dynamics using independent component analysis. *Neuroscience & Biobehavioral Reviews*, 30(6):808–822, 2006. [43](#)
- C. Papageorgakis. Dictionary learning for multidimensional data, 2014. [112](#), [118](#), [119](#)
- Y. Pati, R. Rezaifar, and P. Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on*, pages 40–44. IEEE, 1993. [47](#)
- K. Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901. [41](#)
- D. T. Pham, J. Möcks, W. Köhler, and T. Gasser. Variable latencies of noisy signals: estimation and testing in brain potential data. *Biometrika*, pages 525–533, 1987. [38](#)
- T. Picton, M. Hunt, R. Mowrey, R. Rodriguez, and J. Maru. Evaluation of brain-stem auditory evoked potentials using dynamic time warping. *Electroencephalography and Clinical Neurophysiology/Evoked Potentials Section*, 71(3):212–225, 1988. [39](#), [40](#)
- T. Picton, S. Bentin, P. Berg, E. Donchin, S. Hillyard, R. Johnson, G. Miller, W. Ritter, D. Ruchkin, M. Rugg, et al. Guidelines for using human event-related potentials to study cognition: recording standards and publication criteria. *Psychophysiology*, 37(02):127–152, 2000. [25](#)
- M. D. Plumbley, S. A. Abdallah, T. Blumensath, and M. E. Davies. Sparse representations of polyphonic music. *Signal Processing*, 86(3):417–431, 2006. [52](#)
- J. Polich. Overview of p3a and p3b. In *Detection of change: event-related potential and fMRI findings*, pages 83–98. J Polich, 2003. [25](#)
- J. Polich. Updating p300: an integrative theory of p3a and p3b. *Clinical neurophysiology*, 118(10):2128–2148, 2007. [24](#)
- W. S. Pritchard. Psychophysiology of p300. *Psychological bulletin*, 89(3): 506, 1981. [24](#)

-
- L. R. Rabiner and B.-H. Juang. *Fundamentals of speech recognition*, volume 14. PTR Prentice Hall Englewood Cliffs, 1993, chapter 4. [39](#)
- T. Ristaniemi and J. Joutsensalo. On the performance of blind source separation in cdma. In *Proc. Int. Workshop on Independent Component Analysis and Signal Separation (ICA 99)*, pages 437–441, 1999. [43](#)
- F. Rösler and D. Manzey. Principal components and varimax-rotated components in event-related potential research: some remarks on their interpretation. *Biological Psychology*, 13:3–26, 1981. [42](#)
- S. Sallet, A. I. Ivanov, P. Quilichini, A. Ghestem, B. Giusiano, S. Hitziger, I. Vanzetta, C. Bernard, and C. G. Bénar. Interneurons contribute to the hemodynamic/metabolic response to epileptiform discharges. *submitted to Journal of neurophysiology (second revision)*, 2015. [29](#), [31](#), [99](#), [113](#), [118](#)
- H. Sakoe and S. Chiba. A dynamic programming approach to continuous speech recognition. In *Proceedings of the seventh international congress on acoustics*, volume 3, pages 65–69, 1971. [38](#)
- H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 26(1):43–49, 1978. [38](#)
- S. Salvador and P. Chan. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 11(5):561–580, 2007. [39](#)
- L. Sörnmo and P. Laguna. *Bioelectrical signal processing in cardiac and neurological applications*. Academic Press, 2005, chapter 2. [23](#)
- E.-J. Speckmann and C. E. Elger. Introduction to the neurophysiological basis of the eeg and dc potentials. *Electroencephalography: Basic principles, clinical applications, and related fields*, pages 17–29, 2005. [18](#)
- P. Stoica and Y. Selen. Model-order selection: a review of information criterion rules. *Signal Processing Magazine, IEEE*, 21(4):36–47, 2004. [66](#), [93](#)
- C. Tallon-Baudry and O. Bertrand. Oscillatory gamma activity in humans and its role in object representation. *Trends in cognitive sciences*, 3(4): 151–162, 1999. [25](#), [27](#), [48](#), [49](#)
- R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996. [47](#)
- I. Tasic and P. Frossard. Dictionary Learning. *IEEE Signal Processing Magazine*, 28(2):27–38, Mar. 2011. [50](#)

- I. Tasic, I. Jovanovic, P. Frossard, M. Vetterli, and N. Duric. Ultrasound tomography with learned dictionaries. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 5502–5505. IEEE, 2010. [50](#)
- J. A. Tropp, A. C. Gilbert, and M. J. Strauss. Algorithms for simultaneous sparse approximation. part i: Greedy pursuit. *Signal Processing*, 86(3): 572–588, 2006. [48](#), [112](#), [118](#)
- W. Truccolo, K. H. Knuth, A. Shah, S. L. Bressler, C. E. Schroeder, and M. Ding. Estimation of single-trial multicomponent erps: Differentially variable component analysis (dvca). *Biological cybernetics*, 89(6):426–438, 2003. [45](#)
- W. A. Truccolo, M. Ding, K. H. Knuth, R. Nakamura, and S. L. Bressler. Trial-to-trial variability of cortical evoked responses: implications for the analysis of functional connectivity. *Clinical Neurophysiology*, 113(2):206–226, 2002. [2](#), [10](#)
- S. Vallaghé. *Modélisation du problème direct de la magnéto et électroencéphalographie: méthodes numériques et calibration*. PhD thesis, Université Nice Sophia Antipolis, 2008. [19](#)
- I. Vanzetta, C. Flynn, A. I. Ivanov, C. Bernard, and C. G. Bénar. Investigation of linear coupling between single-event blood flow responses and interictal discharges in a model of experimental epilepsy. *Journal of neurophysiology*, 103(6):3139–3152, 2010. [23](#), [31](#), [113](#), [118](#)
- R. Vigário, V. Jousmäki, M. Haemaelaenen, R. Haft, and E. Oja. Independent component analysis for identification of artifacts in magnetoencephalographic recordings. *Advances in neural information processing systems*, pages 229–235, 1998. [43](#)
- N. Voges, S. Blanchard, F. Wendling, O. David, H. Benali, T. Papadopoulo, M. Clerc, and C. Bénar. Modeling of the neurovascular coupling in epileptic discharges. *Brain topography*, 25(2):136–156, 2012. [31](#), [113](#), [118](#)
- N. Vongsavan and B. Matthews. Some aspects of the use of laser doppler flow meters for recording tissue blood flow. *Experimental physiology*, 78(1):1–14, 1993. [22](#)
- K. Wang, T. Gasser, et al. Alignment of curves by dynamic time warping. *The Annals of Statistics*, 25(3):1251–1276, 1997. [39](#)
- H. Wersing, J. Eggert, and E. Körner. Sparse coding with invariance constraints. In *Artificial Neural Networks and Neural Information Processing - ICANN/ICONIP 2003*, pages 385–392. Springer, 2003. [52](#)

- C. C. Wood and G. McCarthy. Principal component analysis of event-related potentials: Simulation studies demonstrate misallocation of variance across components. *Electroencephalography and Clinical Neurophysiology/Evoked Potentials Section*, 59(3):249–260, 1984. [42](#)
- C. Woody. Characterization of an adaptive filter for the analysis of variable latency neuroelectric signals. *Medical and Biological Engineering and Computing*, 5(6):539–554, 1967. [2](#), [6](#), [10](#), [13](#), [28](#), [34](#), [37](#), [38](#), [56](#)
- L. Xu, P. Stoica, J. Li, S. L. Bressler, X. Shao, and M. Ding. Aseo: a method for the simultaneous estimation of single-trial event-related potentials and ongoing brain activities. *Biomedical Engineering, IEEE Transactions on*, 56(1):111–121, 2009. [46](#)
- J. Zimmerman, P. Thiene, and J. Harding. Design and operation of stable rf-biased superconducting point-contact quantum devices, and a note on the properties of perfectly clean metal contacts. *Journal of Applied Physics*, 41(4):1572–1580, 1970. [21](#)
- H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005. [124](#)