université PARIS-SACLAY

NNT: 2018SACLT018

Thèse de doctorat

TELECOM ParisTech

Non-linear models for neurophysiological time series

Thèse de doctorat de l'Université Paris-Saclay
préparée à Télécom ParisTech

École doctorale n° 580 STIC
Spécialité de doctorat: Traitement du Signal et des Images

Thèse présentée et soutenue à Télécom ParisTech, le 26/11/2018, par

Tom DUPRE LA TOUR

Composition du jury:

Alexandre GRAMFORT Inria Saclay	Directeur de thèse
Yves GRENIER Télécom ParisTech	Directeur de thèse
Maureen CLERC Inria Sophia	Examinateur
Mike X COHEN Radboud University	Examinateur
Eric MOULINES Ecole polytechnique	Examinateur
Guido NOLTE University Medical Center Hamburg-Eppendorf	Rapporteur
Dimitri VAN DE VILLE École polytechnique fédérale de Lausanne	Rapporteur

NON-LINEAR MODELS FOR NEUROPHYSIOLOGICAL TIME SERIES

by TOM DUPRE LA TOUR

PhD dissertation

Université Paris-Saclay
Sciences et technologies de l'Information et de la Communication

October 2015 - November 2018
Télécom ParisTech, Paris, France

PH.D. COMMITTEE

DIRECTORS:
Alexandre Gramfort, Inria, Saclay, France
Yves Grenier, Télécom ParisTech, Paris, France

REVIEWERS:
Guido Nolte, UKE, Hamburg, Germany
Dimitri Van De Ville, ÉPFL, Lausanne, Switzerland

EXAMINERS:
Maureen Clerc, Inria, Sophia, France
Mike X Cohen, Radboud University, Nijmegen, Netherlands
Éric Moulines, École polytechnique, Palaiseau, France

Contents

1 Introduction **11**
- 1.1 Neuroscientific challenges . . . 11
- 1.2 Cross-frequency coupling . . . 15
- 1.3 Non-linear autoregressive models . . . 18
- 1.4 Convolutional sparse coding . . . 21
- 1.5 Chapters summary . . . 23

2 Driven autoregressive models **29**
- 2.1 Driven autoregressive models . . . 30
- 2.2 Stability in DAR models . . . 36

3 DAR models and PAC **49**
- 3.1 DAR models and PAC . . . 50
- 3.2 Model selection . . . 55
- 3.3 Statistical significance . . . 67
- 3.4 Discussion . . . 70

4 Extensions to DAR models **73**
- 4.1 Driver estimation in DAR models . . . 74
- 4.2 Multivariate PAC . . . 80
- 4.3 Spectro-temporal receptive fields . . . 86

5 Convolutional sparse coding **93**
- 5.1 Convolutional sparse coding . . . 94
- 5.2 CSC with alpha-stable distributions . . . 103
- 5.3 Multivariate CSC with a rank-1 constraint . . . 110

Conclusion **123**

A Appendices **125**
- A.1 Convolutional sparse coding . . . 125

Bibliography **129**

"Creating a life that reflects your values and satisfies your soul is a rare achievement. In a culture that relentlessly promotes avarice and excess as the good life, a person happy doing his own work is usually considered an eccentric, if not a subversive. Ambition is only understood if it's to rise to the top of some imaginary ladder of success. Someone who takes an undemanding job because it affords him the time to pursue other interests and activities is considered a flake. A person who abandons a career in order to stay home and raise children is considered not to be living up to his potential – as if a job title and salary are the sole measure of human worth.

You'll be told in a hundred ways, some subtle and some not, to keep climbing, and never be satisfied with where you are, who you are, and what you're doing. There are a million ways to sell yourself out, and I guarantee you'll hear about them.

To invent your own life's meaning is not easy, but it's still allowed, and I think you'll be happier for the trouble."

– Bill Watterson

Abstract

Brain analysis mainly relies on complex recording techniques and on advanced signal processing tools used to interpret these recordings. In neurophysiological time series, as strong neural oscillations are observed in the mammalian brain, the natural processing tools are centered on spectral analysis, Fourier decomposition, and on linear filtering into canonical frequency bands. While this approach has had significant impact in neuroscience, it may give a misleading representation of the signal. Indeed, it is standard to see neuro-scientists consider small subsets of coefficients, implicitly assuming that the signals of interest are narrow-band, which turns out to be too reductive. Multiple warnings have been raised about this fallacy, and about the need of more appropriate methods to represent the signals.

More generally, a large number of neuroscientific studies use ad-hoc recipes to analyze time series and describe their properties. Importantly, these methods are heavily based on narrow-band filtering and on custom correlation metrics, and they fail to give a goodness of fit. Therefore, setting the parameters of these methods can only be driven by how much they lead to a strong value of the metric. As a consequence, even though these metrics give reasonable information, a legitimate and controlled comparison of methods and parameters, and therefore of the results, is impossible. This is the case for instance for a phenomenon known as phase-amplitude coupling (PAC), which consists in an amplitude modulation of a high frequency signal, time-locked with a specific phase of a slow neural oscillation.

In this work, we first propose to use driven autoregressive models (DAR) on neurophysiological time-series. These models give a spectral representation of the signal conditionally to another signal, and thus are able to capture PAC in a probabilistic model of the signal, for which statistical inference is fast and well-posed. Giving a proper model to the signal enables easy model selection and clear hypothesis-testing by using the likelihood of the given model with respect to the data. This data-driven approach is fundamentally different from the traditional PAC metrics, and constitutes a major improvement in PAC estimation by adopting a principled modeling approach.

We first present DAR models in a signal processing perspective, describing how they provide a non-linear spectral estimation of the signal. In particular, we discuss the stability of these models, and propose fast inference algorithms for the different parametrizations considered. Then, we present how to use these models for PAC analysis. We demonstrate the advantage of the model-based approach on three datasets acquired in rats and in humans. Using the powerful model selection enabled by the model-based approach, we further provide novel neuroscientific insights on previously reported PAC phenomena. We discuss the influence of the amplitude of the slow driving oscillation, we provide a directionality estimation through a delay parameter, and we describe

spectral properties of the slow driving oscillation. We also show that DAR models are statistically more robust to small samples than non-parametric approaches.

In a subsequent chapter, we explore different extensions to DAR models, relying on an estimation of the driving signal from the data. We describe preliminary findings on estimating PAC in multivariate signals, estimating virtual channels jointly with DAR models. We also show that DAR models can be used as encoding models, where the brain activity is predicted from the stimulus, leading naturally to spectro-temporal receptive fields estimation.

The last chapter covers a different class of models, which focuses on the temporal morphology of neurophysiological time-series, assuming that the signal is composed of a small number of prototypical temporal waveforms. Using a principled mathematical formulation of this problem, namely convolutional sparse coding (CSC), we show that we obtain a rich signal representation in a completely unsupervised way, avoiding the misleading Fourier decomposition. As CSC models were primarily developed for natural image processing, we extend their formulation to tackle the unusual challenges of neurophysiological time-series, using heavy-tail noise distribution and multivariate decompositions. We develop efficient inference algorithms for each formulation, and describe the potential of such representations on simulations and on empirical recordings.

Our DAR and CSC models offer novel and more robust possibilities to analyze neurophysiological time-series, paving the way for new insights on how our brain functions via spectral interactions or prototypical waveforms.

Keywords Time series – modeling – autoregressive – convolutional sparse coding – cross-frequency coupling – multivariate – encoding – neurophysiology – magnetoencephalography

Notation

General

$\mathbb{R}$	Set of real-valued numbers
$\mathbb{R}^N$	Set of real-valued vectors of size N
a_i	i^{th} element of a
$\hat{a}$	Estimated value of a
$A^\top$	Transpose of matrix A
$\|v\|_q$	Norm q of vector v. $\|v\|_q = \left(\sum_i \|v_i\|^q\right)^{1/q}$
$[\![a, b]\!]$	Set of integers x such that $a \leq x \leq b$
$\mathcal{U}, \mathcal{E}, \mathcal{N}, \mathcal{S}$	Uniform, exponential, Gaussian, and α-stable distributions

Time series

$x[t]$	Value of x at time t
$x^{\reflectbox{7}}$	Reversed-time series. If $x \in \mathbb{R}^T$, then $x^{\reflectbox{7}}[t] = x[T - t + 1]$.
$z * d$	Linear convolution. If $z \in \mathbb{R}^{T+L-1}$ and $d \in \mathbb{R}^L$, then $z * d \in \mathbb{R}^T$. If $D \in \mathbb{R}^{P \times L}$, then $z * D \in \mathbb{R}^{P \times T}$ is obtained by convolving every row of D by z.
$D \tilde{*} D'$	For $D' \in \mathbb{R}^{P \times L}$, $D \tilde{*} D' \in \mathbb{R}^{2L-1}$ is obtained by summing the convolution between each row of D and D': $D \tilde{*} D' = \sum_{p=1}^{P} D_p * D'_p$.

1 Introduction

"The truth is, most of us discover where we are headed when we arrive."

– Bill Watterson

Contents

1.1 Neuroscientific challenges 11
1.2 Cross-frequency coupling 15
1.3 Non-linear autoregressive models 18
1.4 Convolutional sparse coding 21
1.5 Chapters summary 23

1.1 Neuroscientific challenges

The scientific study of the brain has lead to remarkable advances since the middle of the twentieth century, including for instance the description of the visual cortex (Hubel and Wiesel, 1962) or the discovery of place cells (O'Keefe and Dostrovsky, 1971). However, many questions remain unsolved, and understanding the brain constitutes a major scientific challenge of our time. This challenge mainly relies on advanced techniques used to record the brain activity, and on signal processing tools used to interpret these recordings.

Measurements of neural activity Brain imaging techniques come with different temporal and spatial resolutions, along with different degrees of invasiveness. Electroencephalography (EEG), invented in the ninetieth century, measures the electrical activity of the brain with electrodes placed along the scalp. Magnetoencephalography (MEG) (Cohen, 1968), invented in the late sixties, measures the magnetic field produced by the brain, using very sensitive sensors. Both EEG and MEG are non-invasive and have a high temporal resolution (around 1 ms), making them extremely useful to monitor dynamic changes in the brain. Their spatial resolution is however limited (around 2 cm), and they suffer from a low signal-to-noise ratio (SNR). Note that there is hope to bring MEG sensors closer to the brain with new optically pumped magnetometers (Boto

et al., 2018), leading to better SNR and better spatial resolution. A complementary non-invasive technique is functional magnetic resonance imaging (fMRI) (Ogawa et al., 1990), which has a limited temporal resolution (typically around 1 s), but with a better spatial resolution (below 5 mm), enabling precise functional brain mappings.

In order to obtain at the same time good temporal and spatial time scale, electrodes can also be placed closer to the brain, with much more invasive techniques. For example, in electrocorticography (ECoG) (Jasper and Penfield, 1949) electrodes are placed on the cortical surface, below the skull. Another technique consists in placing micro-electrodes directly inside the brain to record so-called local field potential (LFP) (Einevoll et al., 2013). These techniques are much more difficult to implement, but provides extremely valuable recordings with excellent SNR. One limitation is however is that they offer a limited coverage of the brain. Spatial and temporal resolutions of these methods are summarized in Figure 1.1.

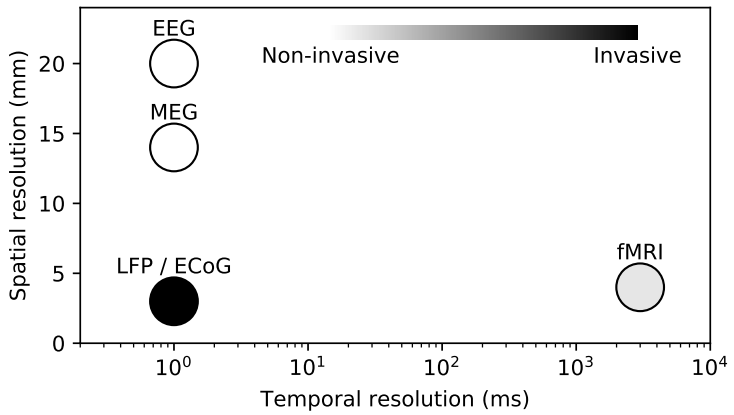


Figure 1.1 – Spatial and temporal resolutions of neural activity recording techniques.

Neural oscillations In neurophysiological time series, *i.e.* recordings with a temporal dimension, neural oscillations are observed in the mammalian brain at different temporal and spatial scales. Figure 1.2 presents an example of neural oscillations in a rodent striatum LFP recording. We clearly see a strong oscillation around 3 Hz, along with some weaker oscillations around 80 Hz. It leads naturally to some spectral analysis, and a large part of the traditional analysis is thus centered around the Fourier decomposition and linear filtering, leading to canonical frequency bands. In human, the main frequency bands are called delta (1-4 Hz), theta (4-8 Hz), alpha (8-12 Hz), beta (15-30 Hz), gamma (30-90 Hz), and high gamma (>50 Hz) (Buzsáki, 2006).

In each frequency band, signal power can be quantified, and used for instance to characterize cognitive states. For example, strong alpha rhythms are present in the visual cortex, and are usually associated with wakeful relaxation with closed eyes, while strong beta rhythms are associated with normal waking consciousness, and are located notably in the motor cortex (Buzsáki, 2006). Some rhythms have strong power in the same frequency band but are inherently different. The most remarkable example is the mu rhythm, which peaks in the same frequency band as the alpha rhythm, but which have a very different waveform in the shape of the Greek letter μ. They are located

predominantly in the motor cortex when the body in physically at rest (Buzsáki, 2006, Cole and Voytek, 2017).

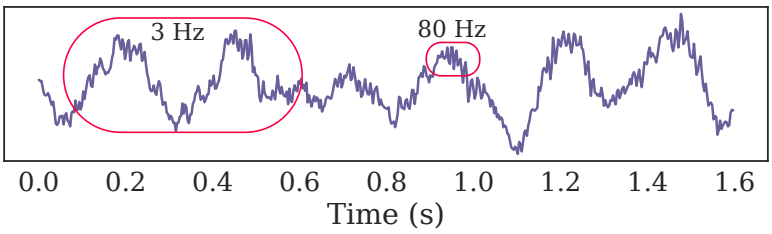


Figure 1.2 – Example of neural oscillations, in rodent striatal LFP recordings. We observe strong oscillations around 3 Hz and around 80 Hz.

Non-sinusoidal waveforms While spectral analysis based on Fourier basis and linear filtering has had significant impact in neuroscience, it may give a misleading representation of the signal. Indeed, it is standard to see neuro-scientists consider small subsets of coefficients, implicitly assuming that the signals of interest are narrow-band. This hypothesis is rather reductive (Mazaheri and Jensen, 2008), and wide-band waveforms have also been reported as key features in neurophysiological signals (Jones, 2016, Cole et al., 2016). Typically, a classic Fourier analysis fails to distinguish alpha rhythms from mu rhythms, which have the same peak frequency at around 10 Hz, but whose waveforms are different (Cole and Voytek, 2017, Hari and Puce, 2017). Jasper warned about this misconception as early as in 1948: "Even though it may be possible to analyze the complex forms of brain waves into a number of different sine-wave frequencies, this may lead only to what might be termed a "Fourier fallacy", if one assumes *ad hoc* that all of the necessary frequencies actually occur as periodic phenomena in cell groups within the brain." (Jasper, 1948). There is now a debate regarding whether neural activity consists more of transient bursts of isolated events rather than rhythmically sustained oscillations (van Ede et al., 2018). Going beyond classic Fourier analysis and linear filtering requires designing new tools to describe neurophysiological time series.

Non-stationary signals Another challenge is the non-stationarity of the neurophysiological signals. Many analysis tools rely on the hypothesis that the signal's characteristics are stable over time. Then, to measure fluctuations over time, the standard technique is to apply the same analysis multiple time on a sliding time windows. However, this technique reduces dramatically the robustness of the analysis, since for each computation the data size is reduced to a small window (Leonardi and Van De Ville, 2015). The robustness of an analysis is critical in neurophysiological signals, since the SNR can be extremely low, and the dynamic changes very fast. To overcome this limitation, different assumptions can be made instead of the too general non-stationarity assumption.

A prominent technique is to precisely synchronize multiple time windows with stimulus onsets, and to average these synchronized signals. This technique reveals the so-called event-related potential (ERP) (Davis et al., 1939), where the synchronized activity emerges from the noise through averaging. This powerful technique is however limited to temporally well defined stimuli, and only extracts the part of the response which is synchronized with the stimuli.

Another weaker assumption is to consider that the signal switches between different states with different characteristics. The signal changes over time from one state to another, but the states themselves are stable in time. This is the idea behind Hidden Markov models (HMM) (Baum and Petrie, 1966), which have been used for instance to estimate fast transient brain states in MEG data (Woolrich et al., 2013, Baker et al., 2014, Vidaurre et al., 2017). This assumption is also used in switching autoregressive models, as described later in this introduction. Here again, the key element of this assumption is to take into account multiple time windows to estimate each state's characteristics. Therefore, the analysis uses many more time points and can be much more robust and statistically efficient.

Another approach is to assume the signal is a linear sum of local temporal patterns, which are repeated multiple times over the signals. The patterns are assumed to be stable over time, but the decomposition of the signal fluctuates, allowing dynamic changes in the signals. This assumption is the basis of convolutional dictionary learning approaches as described later in this introduction. Here again, the goal is to use multiple realizations of the same pattern to get a more robust analysis. Note that contrary to the ERP, there is no assumption of synchronization with some stimuli onsets.

Proper signal models The need of new analysis tools lead to the design of many ad-hoc metrics describing the signal without a proper signal model. Using ad-hoc recipes, these metrics are able to quantify some phenomena present in the signals, but do not provide any measure of goodness of fit. Therefore, setting the parameters of these methods, such as filtering parameters, can only be driven by how much they lead to a strong value of the metric. As a consequence, even though current metrics give reasonable information, a legitimate and controlled comparison of methods and parameters, and therefore of the results, is impossible. Additionally, while simulations provide better control, they do not fully solve this issue, since a simulation may approximate at best, or miss at worst, the real structure of neurophysiological signals.

On the contrary, a model-based approach allows computing the likelihood of a recorded neural signal with respect to the model. This likelihood can be interpreted as a measure of the goodness of fit of the model, and often corresponds to the classical measure of explained variance. Such an evaluation metric is a natural criterion to compare models, and a first step towards an automatic model parameters selection on empirical data. Indeed, it enables out-of-sample evaluation, *i.e.* an evaluation of the model on some data which was not used during model inference and parameter selection. This evaluation is standard in the machine learning community (Bishop, 2006), and quantifies the model's ability to generalize the structure of the signals on unseen data.

Importantly, a signal modeling approach can be entirely statistical, as opposed to biophysical modeling approaches (Hyafil et al., 2015, Chehelcheraghi et al., 2017). The goal of statistical model is to better explain and describe the empirical data themselves in the absence of any assumption regarding the neural mechanisms that have generated them.

Computational efficiency Finally, another critical aspect of analysis techniques is the computationally efficiency. Indeed, the typical neurophysiological recording can be very long (tens of minutes to hours) and spread over multiple channels (tens to hundreds). The data can easily reach several gigabytes for each subject. Moreover, because of recent concerns about predictive power and reproducibility in neuroscience

(Carp, 2012, Munafò et al., 2017), the number of subjects in brain study is also increasing. Whereas in the past a typical study used to have 20 to 30 subjects, some recent datasets contain thousand of subjects, such as the human connectome project (HCP) (Van Essen et al., 2013), or the Cambridge center for aging and neuroscience (Cam-CAN) dataset (Taylor et al., 2017). Therefore, the analysis must be computationally efficient to be able to process such increasingly large amount of data.

1.2 Cross-frequency coupling

Cross-frequency coupling (CFC) The characterization of neural oscillations have given rise to important mechanistic hypotheses regarding their functional role in neurosciences (Buzsáki, 2006, Fries, 2015). One working hypothesis suggests that the coupling across neural oscillations may regulate and synchronize multi-scale communication of neural information within and across neural ensembles (Buzsáki, 2010, Fries, 2015). The coupling across different oscillatory activity is generically called *cross-frequency-coupling* (CFC) and has started receiving much attention (Jensen and Colgin, 2007, Lisman and Jensen, 2013, Canolty et al., 2006, Canolty and Knight, 2010, Hyafil et al., 2015). The most frequent instance of CFC consists in the observation that the power of high frequency activity is modulated by fluctuations of low-frequency oscillations, resulting in *phase-amplitude coupling* (PAC). This can be observed for instance in Figure 1.2, where at the peak of each 3 Hz oscillation we see a increase of energy at 80 Hz. Other instances of CFC include phase-phase coupling (Tort et al., 2007, Malerba and Kopell, 2013), amplitude-amplitude coupling (Bruns and Eckhorn, 2004, Shirvalkar et al., 2010), and phase-frequency coupling (Jensen and Colgin, 2007, Hyafil et al., 2015). By far, PAC is the most reported CFC in the literature. Figure 1.3 presents these different couplings on simple sinusoids.

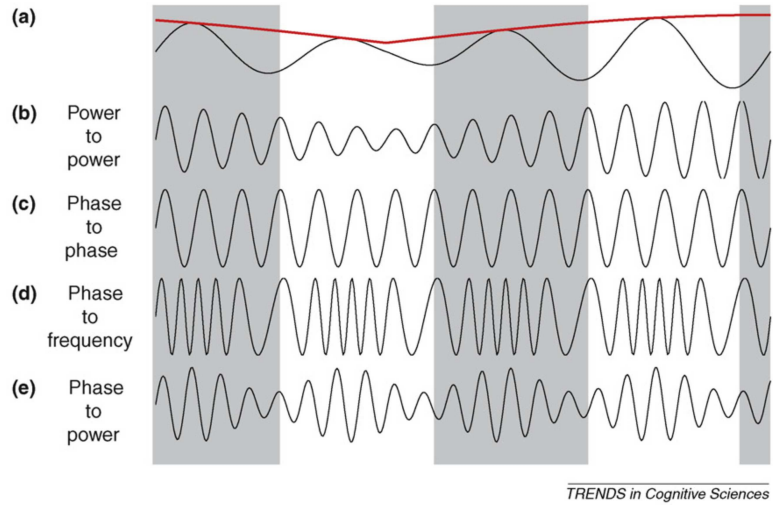


Figure 1.3 – Different sort of cross-frequency coupling (CFC), including amplitude-amplitude coupling (a/b), phase-phase coupling (a/c), phase-frequency coupling (a/d), and phase-amplitude coupling (a/e). Reproduced from Jensen and Colgin (2007).

Phase-amplitude coupling (PAC) Seminally, PAC was described in LFP of rodents displaying a modulation of gamma band power (40-100 Hz) as a function of the phase of their hippocampal theta band (5-10 Hz) (Bragin et al., 1995, Tort et al., 2008). Parallel recordings in different brain areas in behaving rodents have also highlighted differences in PAC between brain areas (*e.g.*, hippocampus and striatum) at specific moments during a goal-oriented behavior, both in terms of which high-frequency range and how narrow-band the low frequency is (Tort et al., 2008). PAC may promote cellular plasticity underlying memory formation (Axmacher et al., 2006). In humans, theta (4-8 Hz)/gamma (80-150 Hz) PAC was described in auditory cortex during speech perception (Canolty et al., 2006). In more recent works, theta/gamma PAC was reported during the processing of auditory sequences in both humans and monkeys (Kikuchi et al., 2017), during working memory maintenance in human hippocampus (Axmacher et al., 2010), and during serial memory recall using non-invasive human magnetoencephalography (MEG) (Heusser et al., 2016). PAC has been proposed to support the maintenance of information and to play an important role in long distance communication between different neural populations, considering that slow oscillations can propagate at larger scales than fast ones (Jensen and Colgin, 2007, Khan et al., 2013, Lisman and Jensen, 2013, Hyafil et al., 2015, Bonnefond et al., 2017). Consistent with this notion, PAC has also been reported across distinct brain regions (Sweeney-Reed et al., 2014). In sum, PAC has been proposed as a canonical mechanism for neural syntax (Buzsáki, 2010).

PAC estimation techniques Given the growing interest in CFC, and in PAC more specifically, developing adequate and unbiased tools to quantify the posited signatures of neural computations has motivated a number of contributions. We describe here these techniques, denoting y the signal containing the high-frequency activity, and x the signal with low-frequency oscillations. When a signal x results from a band-pass filtering step, we note the central frequency of the filter f_x and the bandwidth Δf_x.

To estimate PAC, the typical pipeline reported in the literature consists in four main processing steps:

1. Bandpass filtering is performed in order to extract the narrow-band neural oscillations at both low frequency f_x and high frequency f_y;
2. A Hilbert transform is applied to get the complex-valued analytic signals of x and y;
3. The phase ϕ_x of the low-frequency oscillation and the amplitude a_y of the high-frequency signals are extracted from the complex-valued signals;
4. A dedicated approach is used to quantify the correlation between the phase ϕ_x and the amplitude a_y signals.

For instance, the modulation index (MI) described in the pioneering work of Canolty et al. (2006) is the mean over time of the composite signal $z = a_y e^{\phi_x}$. The stronger the coupling between ϕ_x and a_y, the more the MI deviates from zero. This index has been further improved with a better normalization by Özkurt and Schnitzler (2011). Another approach (Lakatos et al., 2005, Tort et al., 2010) has been to partition $[0, 2\pi]$ into smaller intervals to get the time points t when $\phi_x(t)$ is within each interval, and to compute the mean of $a_y(t)$ on these time points. PAC was then quantified by looking at how much the distribution of a_y with respect to ϕ_x differs from uniformity. For instance, a simple height ratio (Lakatos et al., 2005), or a Kullback-Leibler divergence

as proposed by Tort et al. (2010), can be computed between the estimated distribution and the uniform distribution. Alternatively, it was proposed in Bruns and Eckhorn (2004) to use direct correlation between x and a_y. As this method yielded artificially weaker coupling values when the maximum amplitude a_y was not exactly on the peaks or troughs of x, this method was later extended to generalized linear models (GLM) using both $\cos(\phi_x)$ and $\sin(\phi_x)$ by Penny et al. (2008). This approach offers a metric which is independent of the phase at which the maximum amplitude occurs. Other approaches employed a measure of coherence (Colgin et al., 2009) or the phase-locking value (Lachaux et al., 1999).

Limitations As one can see, there is a long list of methods to quantify CFC in neural time series. Yet, a number of limitations which can significantly affect the outcomes and interpretations of neuroscientific findings exist with these approaches. For example, a systematic bias rises where one constructs the so-called *comodulogram*. A comodulogram is obtained by evaluating the chosen metric over a grid of frequency f_x and f_y. This bias emerges from the choice of the bandpass filter, which involves the critical choice of the bandwidth Δf_y. It has been reported several times that to observe any amplitude modulation, the bandwidth of the fast oscillation Δf_y has to be at least twice as high as the frequency of the slow oscillations f_x: $\Delta f_y > 2f_x$ (Berman et al., 2012, Dvorak and Fenton, 2014). As a comodulogram uses different values for f_y, many studies have used a variable bandwidth, by taking a fixed number of cycles in the filters. The bandwidth is thus proportional to the center frequency: $\Delta f_y \propto f_y$. This choice leads to a systematic bias, as it hides any possible coupling below the diagonal $f_y = 2f_x/\alpha$, where $\alpha = \Delta f_y/f_y$ is the proportionality factor. Other studies have used a constant bandwidth Δf_y; yet this also biases the results towards the low driver frequency f_x, considering that it hides any coupling with $f_x > \Delta f_y/2$. A proper way to build a comodulogram would be to take a variable bandwidth $\Delta f_y \propto f_x$, with $\Delta f_y > 2f_x$. However, this is not common practice as it is computationally very demanding, because it implies to bandpass filter y again for each value of f_x.

Another common issue arises with the use of the Hilbert transform to estimate the amplitude and the phase of real-valued signals. Such estimations rely on the hypothesis that the signals x and y are narrow-band, *i.e.* almost sinusoidal. However, numerous studies have used this technique on very wide-band signals such as the entire gamma band (80-150 Hz) (Canolty et al., 2006) (see other examples in Chavez et al. (2006)). The narrow-band assumption is debatable for high frequency activity and, consequently, using the Hilbert transform may yield non-meaningful amplitude estimations, and potentially poor estimations of PAC (Chavez et al., 2006, Dvorak and Fenton, 2014). Note also that, in this context, wavelet-based filtering is equivalent to the Hilbert transform (Quiroga et al., 2002, Bruns, 2004), and therefore does not provide a more valid alternative option.

Besides these issues of filtering and inappropriate use of Hilbert transforms, Hyafil (2015) also warned that certain choices of bandwidth Δf_y might mistake phase-frequency coupling for PAC, such as presented in Figure 1.4. See also the more recent work in Aru et al. (2015) for discussion and more practical recommendations for PAC analysis.

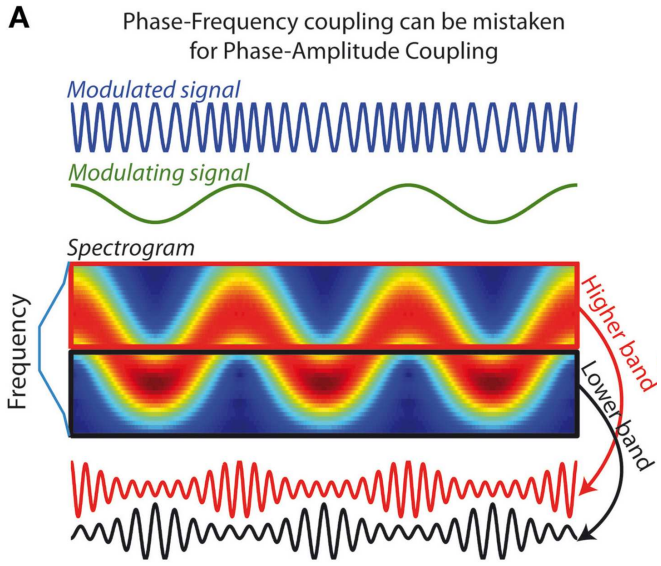


Figure 1.4 – Misidentification of phase-frequency coupling as phase-amplitude coupling. Reproduced from Hyafil (2015).

1.3 Non-linear autoregressive models

Given these limitations of PAC estimation metrics, we propose to use autoregressive (AR) models to capture PAC in neurophysiological time-series. Indeed, we first note that PAC corresponds to a modulation of the power-spectral density (PSD) of the signal. This remark naturally leads to AR models, which are stochastic processes that naturally estimate the PSD of signals. As we not only want to estimate the PSD, but also the PSD modulation, we develop non-linear AR models which estimate the PSD *conditionally* to a slowly-varying oscillation.

Giving a proper signal model to the signal enables easy model selection and clear hypothesis-testing by using the likelihood of the given model. This data-driven approach is fundamentally different from the traditional PAC metrics, and constitutes a major improvement in PAC estimation by adopting a principled modeling approach. We first limit ourselves to univariate signals, yet a multivariate extension is proposed later in this manuscript.

Autoregressive models (AR) AR models, also known as linear prediction modeling (Makhoul, 1975), are stochastic signal models, which aim to forecast the signal values based on its own past. More precisely, an AR model specifies that y depends linearly on its own p past values, where p is the *order* of the model:

$$\forall t \in [p+1, T] \qquad y(t) + \sum_{i=1}^{p} a_i y(t-i) = \varepsilon(t) \ , \tag{1.1}$$

where, T is the length of the signal and ε is the *innovation* (or *residual*) modeled with a Gaussian white noise: $\varepsilon(t) \sim \mathcal{N}(0, \sigma(t)^2)$.

AR models are mainly used for spectral estimation. Indeed, since an AR model is a linear filter that is fitted to somehow whiten a particular signal y, the PSD of this filter is close to the inverse of the PSD of the signal, providing a robust spectral estimation

of the signal. For a linear AR model, the PSD of y at a frequency f is given by:

$$\text{PSD}(f) = \sigma^2 \left| \sum_{i=0}^{p} a_i e^{-2j\pi fi} \right|^{-2}, \tag{1.2}$$

where $j^2 = -1$ and $a_0 = 1$. This estimation converges to the true PSD of y when $p \to \infty$, but in practice, using $p \in [10, 100]$ gives satisfying results in most applications (Kay and Marple, 1981).

AR models have been successfully used for spectral estimation in a wide variety of fields, such as geophysics, radar, radio astronomy, oceanography, speech processing, and many more (Kay and Marple, 1981). Importantly, they can be applied on any signal, even deterministic, as they can be seen as linear predictions based on the p past elements, independently from the way the signal was generated. However, in the specific case of a noisy sum of perfect sinusoids, the peak amplitudes in AR spectral estimates are proportional to the square of the power (Kay and Marple, 1981), unlike conventional Fourier spectral estimates where the peak amplitudes are proportional to the power. The method of Pisarenko (1973), later extended in the MUSIC algorithm (Schmidt, 1986), might be more suited for spectral estimation of such "ideal" signals. However, as neurophysiological signals are not perfect sinusoids, AR model are well suited for their analysis. For a good overview of spectral estimation techniques, including AR models, we refer the reader to Kay and Marple (1981).

Interestingly, AR models yield a spectral estimation with an excellent frequency resolution. Here, frequency resolution is not defined as the distance between two points in the PSD, which would go to zero as we increase zero-padding in the AR filter. Rather, the frequency resolution is defined as the smallest distance that can be detected between two spectral peaks. With this definition, spectral estimation with AR models has a better frequency resolution than Fourier-based methods (Marple, 1977).

Another benefit of spectral estimation with AR models is the theoretical guarantee in term of maximum entropy. Entropy is an information theory concept which quantifies the information contained in a signal (Shannon and Weaver, 1963). Maximizing the entropy of the spectrum consists in introducing as little as possible artificial information in the spectrum (Ables, 1974). Assuming that we have p auto-correlations of a wide-sense stationary signal, maximizing the entropy leads to the maximum likelihood estimate of an AR model (Ables, 1974). In other words, the maximum likelihood estimate of an AR model is the best spectral estimator in term of maximal entropy.

In neurophysiology, AR models have been successfully used to address multiple problems such as spectral estimation (Spyers-Ashby et al., 1998), temporal whitening (Mahan et al., 2015), and connectivity measures like Granger causality (Granger, 1988, Valdés-Sosa et al., 2005, Haufe et al., 2010) or coherence (Baccalá and Sameshima, 2001).

Standard AR models are statistically efficient given their low number of parameters, but they are linear, and therefore they cannot directly model non-linear phenomena like PAC. The challenge here is to consider a signal model which is rich enough to capture phenomena such as PAC, and simple enough to be statistically robust and computationally efficient. In order to extend AR models to cope with such situations of non-linearity and non-stationarity in signals, various advanced AR models have been proposed in other research fields such as audio signal processing and econometrics.

Non-linear AR models A first category of AR models extension are the time-varying AR models (TVAR) (Dahlhaus, 1996), where an instantaneous AR model is updated at each time point, with typically a single gradient descent step. However, as we would like the models to capture PAC, we can assume that the different spectral states of PAC are repeated multiple times on the signal. As TVAR models are unstructured, they do not make use of this assumption and are thus expected to get a poorer estimation of PAC.

A second category of non-linear autoregressive models are based on conditional heteroskedasticity (ARCH (Engle, 1982), GARCH (Bollerslev, 1986)), and are extremely popular in econometrics. In these models, the innovation variance $\sigma^2(t)$ is varying and modeled with an AR model. Therefore, they are used to model signals whose overall amplitude varies as a function of time. In the context of CFC and PAC, however, one would like to model variations in the PSD itself, such as shifts in peak frequencies (a.k.a. frequency modulations) or changes in amplitude only within certain frequency bands (a.k.a. amplitude modulations).

To achieve these spectral modulations, we consider a third category of models, that we call here the switching AR models. These models switch over time between different instantaneous AR models, and the switching mechanism can be either deterministic or probabilistic.

Switching AR models The seminal work of Tong and Lim (1980) introduced the threshold AR (TAR) models, where a driving time series x acts as a switching mechanism between several AR models applied on the signal y. Several extensions have been developed to add a smooth transition between regimes, like the exponential AR (EAR) (Haggan and Ozaki, 1981) or the smooth transition AR (STAR) (Chan and Tong, 1986) models. The general formulation can be written as:

$$\forall t \in [p+1, T] \qquad y(t) + \sum_{i=1}^{p} a_i(x(t)) y(t-i) = \varepsilon(t) \ , \tag{1.3}$$

where $a_i(x)$ can have different expressions. In TAR models, it is a piecewise constant function. In EAR models, it is a sum of negative squared exponential $a_i(x) = b_i + \sum_{k=0}^{m} a_{ik} e^{-\gamma_k x^2}$. In STAR models, it is a sum of sigmoid functions $a_i(x) = \sum_{k=0}^{m} a_{ik}(1 + e^{-\gamma_k (x - c_k)})^{-1}$.

Concerning the driver x, some models consider it to be hidden, assuming for instance a Markov chain structure (Hamilton, 1989). Such probabilistic inference is computationally intensive and cannot be evaluated on a validation set, since the driver has to be re-estimated on the validation set. In other models, a parametric approach enables model evaluation on a validation set, which makes model comparison easy. For instance, the driver can be a function of the signal y itself, as in self-exciting TAR (SETAR) (Tong and Lim, 1980, Dijk et al., 2002) models. A typical choice is $x(t) = y(t-d)$ with a positive delay $d > 0$. The driver can also be optimized as a weighted average of several potential drivers (Chen and So, 2006, Wu and Chen, 2007), before being used in a deterministic (Chen and So, 2006) or a probabilistic (Wu and Chen, 2007) TAR models. The set of potential drivers can also be used directly to linearly parametrize the AR coefficients (Grenier, 1983, Jachan et al., 2007, Spiridonakos and Fassois, 2014).

Driven AR models The models considered in this work are called driven AR (DAR) models (Grenier, 2013, Dupré la Tour et al., 2017a). They are deterministic continuously switching AR models, with a polynomial expression for $a_i(x)$:

$$a_i(x) = \sum_{k=0}^{m} a_{ik} x^k \ , \tag{1.4}$$

and where the driver x is an exogenous signal, *i.e.* it is considered to be known. We show in particular how they give a spectral estimation conditionally to the exogenous driver, which robustly captures PAC when using the low-frequency oscillation as the driver x. DAR models are also extremely efficient computationally, thanks to the polynomial parametrization.

DAR models provide efficient modeling of PAC thanks to a strong assumption. Indeed, they assume that the coupling is stationary, and that the PSD fluctuations are driven by a given signal. In other word, they assume that the PSD is non-stationary, but that the fluctuations have a one-dimensional trajectory which is in already known. This powerful assumption seems legitimate for PAC characterization, but is too strong for general analysis.

1.4 Convolutional sparse coding

Another approach to analyze non-sinusoidal and non-stationary neurophysiological signals is based on sparse representation learning. It consists in learning patterns that offer sparse approximations, in the sense that signals are well approximated by as few patterns as possible. Contrary to Fourier or wavelet bases, the patterns are not predefined, but are learned from the signal itself, in an unsupervised fashion. In particular, such learned representations are not limited to narrow frequency bands. Sparse analyses have been quite successful in multiple fields such as computer vision (Heide et al., 2015, Bagnell and Bradley, 2009, Kavukcuoglu et al., 2010), biomedical imaging (Pachitariu et al., 2013), genomic (Mairal et al., 2010), and audio signal processing (Grosse et al., 2007, Févotte et al., 2009).

Sparse representations Fourier representations can be extremely useful but are generally not sparse. Indeed, in most applications the signals have a continuous spectrum, and thus are not well approximated by a small number of sinusoids. To provide sparser representations in image and signal processing, multiple bases were proposed (Mallat and Zhang, 1993, Donoho and Johnstone, 1994, Candès et al., 2006). Among the most successful ones, wavelet bases are composed of shifted and dilated versions of a small set of patterns. A large effort has been dedicated to find the best wavelet set adapted for instance to natural images (Daubechies, 1988, Simoncelli et al., 1992, Starck et al., 2002, Do and Vetterli, 2005).

A different approach has been proposed by Olshausen and Field (1996), where the basis (or *dictionary*) was adapted to the dataset itself. They applied this approach on image patches and successfully *learned* a dictionary inducing sparse representations, from their dataset. The dictionary learning approach was later shown to outperform predefined bases for signal approximation (Elad and Aharon, 2006, Mairal et al., 2008). For more details about sparse representations, we refer the reader to Mairal et al. (2014).

Shift-invariant representations Since natural images have often millions of pixels, their natural representation have a very large number of dimensions. Therefore, the original dictionary learning was not applied on large images, but on small image patches (Olshausen and Field, 1996). In this case, a single pattern can be learned multiple times if it appears with different shifts with respect to the patch grid. In other words, a patch decomposition imposes an arbitrary alignment with the signal structure, which can artificially inflates the pattern diversity (Lewicki and Sejnowski, 1999). To overcome this issue, several *shift-invariant* dictionary learning methods have been proposed.

The first mention to shift-invariance sparse representations was described in Lewicki and Sejnowski (1999), using predefined patterns (a.k.a. *kernels* or *atoms*). Later, Grosse et al. (2007) proposed a formulation with both dictionary learning and shift-invariant sparse representations, with a now canonical formulation called *convolutional sparse coding* (CSC).

Convolutional sparse coding (CSC) CSC is a mathematically principled formulation which reads:

$$\begin{aligned} \min_{\{d_k\},\{z_k^n\}} \sum_{n=1}^{N} \frac{1}{2} \left\| x^n - \sum_{k=1}^{K} z_k^n * d_k \right\|_2^2 + \lambda \sum_{k=1}^{K} \|z_k^n\|_1 \ , \\ \text{s.t.} \quad \|d_k\|_2^2 \leq 1 \quad \forall k, \end{aligned} \tag{1.5}$$

where $\{x^n\}_{n=1}^N \subset \mathbb{R}^T$ are N observed signals, $\{d_k\}_{k=1}^K \subset \mathbb{R}^L$ are the K temporal atoms we aim to learn, $\{z_k^n\}_{k=1}^K \subset \mathbb{R}^{T-L+1}$ are K signals of activations (a.k.a. the code) associated with x^n, and $\lambda > 0$ is the regularization parameter.

This formulation uses an L1 regularization to induce sparsity in the activations (Tibshirani, 1996), making the problem convex in $\{z_k^n\}$, contrary to the original formulation of Olshausen and Field (1996). CSC was later extended to two-dimensional atoms (Zeiler et al., 2010), and gained popularity in computer vision (Heide et al., 2015, Wohlberg, 2016b, Šorel and Šroubek, 2016, Kavukcuoglu et al., 2010), biomedical imaging (Pachitariu et al., 2013), and audio signal processing (Grosse et al., 2007, Mailhé et al., 2008).

Note that sparsity is critical in the CSC problem, since there is many more unknown variables than known ones. Indeed, there are $K(T-L+1)N + KL$ unknown variables, for only NT points in the signals $\{x^n\}$. For example, without the sparsity constraint, a single dirac atom $d_0[t] = \delta_{0,t}$ and signals of activations identical to the signals $z_0^n[t] = x^n[t]$ would lead to a perfect reconstruction (except at the edges). Sparsity is here the key element to avoid such trivial solutions.

Other methods CSC is not the only formulation of shift-invariant dictionary learning. For instance, Jost et al. (2006) proposed the MoTIF algorithm, which uses an iterative strategy based on generalized eigenvalue decompositions, where the atoms are assumed to be orthogonal to each other and learnt one by one in a greedy way. More recently, the sliding window matching (SWM) algorithm (Gips et al., 2017) was proposed for learning time-varying atoms by using a correlation-based approach that aims to identify the recurring patterns. Even though some success has been reported with these algorithms, they have several limitations: SWM uses a slow stochastic search inspired by simulated annealing and MoTIF poorly handles correlated atoms, simultaneously activated, or having varying amplitudes; cases which often occur in practical applications.

In a similar spirit, Brockmeier and Príncipe (2016) used the matching pursuit algorithm combined with a rather heuristic dictionary update, which is similar to the MoTIF algorithm. In a very recent study, Hitziger et al. (2017) proposed the AWL algorithm, which presents a mathematically more principled approach close to CSC for modeling neural signals. Yet, as opposed to classical CSC approaches, the AWL algorithm imposes additional combinatorial constraints, which limit its scope to certain data that contain spike-like atoms. Also, since these constraints increase the complexity of the optimization problem, the authors had to resort to dataset-specific initializations and many heuristics in their inference procedure.

Challenges with neural signals While the current state-of-the-art CSC methods have a strong potential for modeling neural signals, they might also be limited as they consider an ℓ_2 reconstruction error, which corresponds to assuming an additive Gaussian noise distribution. While this assumption could be reasonable for several signal processing tasks, it turns out to be very restrictive for neural signals, which often contain heavy noise bursts and have low signal-to-noise ratio.

Another challenge is that CSC was essentially developed for univariate signals, whereas neural signals generally have tens to hundreds of channels. Interestingly, images can be multivariate such as color or hyper-spectral images, yet most CSC methods used in computer vision only consider gray scale images. To the best of our knowledge, the only reference to multivariate CSC is Wohlberg (2016a), where the author proposes two models well suited for 3-channel images, but not for hundreds of channels. Multivariate sparse coding was also developed in Barthélemy et al. (2012, 2013), yet their proposed optimization techniques are not specific to shift-invariant models, and not scalable to long signals.

Finally, neural signals can be very long (tens of minutes to hours sampled at 1 kHz), therefore a computational challenge also rises, and optimization techniques used in CSC are not well suited for long signals. A traditional workaround is to split long signals into blocks, but the optimization problem is then changed, introducing edge-artifacts which can bias the estimation.

1.5 Chapters summary

We describe here the contents of each chapter, emphasizing on the contributions, and listing the associated publications.

Chapter 2: Driven autoregressive models In this chapter, we introduce driven autoregressive (DAR) models. DAR models are deterministic continuously switching AR models, with a polynomial expression $a_i(x) = \sum_{k=0}^{m} a_{ik} x^k$, and where the driver x is an exogenous signal. We also parametrize the log of the residual variance as a polynomial expression of the driver $\log(\sigma(x)) = \sum_{k=0}^{m} b_k x^k$.

In Section 2.1, we describe the parametrization and propose an efficient estimation method for DAR models. Estimating the coefficients a_{ik} leads to a simple system of linear equations, which can be solved analytically. To estimate the residual's variance coefficients b_k, we propose a Newton scheme and a heuristic to have a good initialization. We empirically validate the polynomial expression through a comparison with a piecewise-constant expression and with a sum of sigmoids. We also discuss an alternative parametrization with an asynchronous driving behavior and a faster estimation.

In Section 2.2, we discuss the stability of DAR models. Assuming that each instantaneous AR model is stable, and that the system is slowly-varying, we derive a global stability property. Then, we introduce a different parametrization for DAR models, based on lattice representation and log-area-ratio, to force each instantaneous AR model to be stable. To estimate these log-area-ratio coefficients, we propose a Newton scheme with an good initialization based on intermediate lattice coefficients. Finally, we present an experiment to explore the limit of quickly-varying drivers in DAR models.

This work lead to the following conference publication:

- Dupré la Tour, T., Grenier, Y., and Gramfort, A. (2017a). Parametric estimation of spectrum driven by an exogenous signal. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4301–4305

As the main contributor to this project, I conducted the formal analysis, the model development, the software implementation, the experiment conceptions and visualizations, and the article writing. I am the only student who worked on the development and validation of the method.

Chapter 3: DAR models and phase-amplitude coupling In this chapter, we propose to use DAR models to capture and characterize phase-amplitude coupling (PAC). To do so, we consider the low-frequency oscillation as the exogenous driver x. This allows the instantaneous power spectral density (PSD) of the signal to be a function of the low-frequency oscillation. Thus, DAR models are statistical signal models that do not model PAC explicitly but which are able to capture it.

We advocate that DAR models address a number of the limitations of traditional PAC metrics. They do *not* use bandpass filter or Hilbert transform on the high frequencies y. More importantly, they introduce a measure of *goodness of fit*, through the use of a probabilistic signal model. Thus, DAR model's quality can be assessed by evaluating the *likelihood* of the data under the model, enabling legitimate model selection. This feature is unique to our modeling approach, and constitutes a major improvement compared to past approaches that estimate CFC.

In Section 3.1, we describe how to use DAR models to capture PAC. We first describe the models and detail the necessary preprocessing steps. Then, we adapt the parametrization to make the model invariant to the phase of the coupling. We also describe how to quantify PAC from the DAR model, and how to build a comodulogram. Finally, we discuss that DAR models are not specific to PAC, but can also measure phase-frequency coupling, and are related to amplitude-amplitude coupling.

In Section 3.2, we describe in detail how using DAR models enable model selection, which presents a major improvement compared to past approaches that estimate PAC. We start with a review of classic model selection techniques. Then, we use model selection to choose the best filter extracting the driver. We validate this approach on simulations, and apply it on three empirical datasets. This analysis lead to an original hypothesis concerning an asymmetrical PSD of the driver. Then, we use model selection to select the best delay between the driver and the signal. We discuss the directionality interpretation of such delay, emphasizing the difference between this delay and a simple phase difference. We validate this approach on simulations, and apply it on three empirical datasets. Finally, we use model selection to answer an hypothesis concerning the role of the driver's amplitude. On three empirical datasets, we find that the driver's amplitude improves the model likelihood, which advocates that the driver's amplitude

is incorrectly discarded in most PAC analyses.

In Section 3.3, we discuss techniques to assess statistical significance in PAC estimation. We first present standard permutation-based techniques, along with model selection techniques that can be used with DAR models. Then, using PAC simulations, we show that parametric methods such as DAR models are more robust to small samples than non-parametric methods. Finally, we discuss the effect of non-sinusoidal waveforms of PAC measures.

This work lead to the following journal publication:

- Dupré la Tour, T., Tallot, L., Grabot, L., Doyère, V., van Wassenhove, V., Grenier, Y., and Gramfort, A. (2017b). Non-linear auto-regressive models for cross-frequency coupling in neural time series. *PLOS Computational Biology*, 13(12)

As the main contributor to this project, I conducted the formal analysis, the model development, the software implementation, the experiment conceptions and visualizations, and the article writing. I am the only student who worked on the development and validation of the method.

Chapter 4: Extensions to DAR models In this chapter, we propose different extensions to DAR models, centered on the driver estimation. Indeed, in previous chapters, we consider the driver x to be known, but this strong assumption can be soften by optimizing the driver to better fit the spectral trajectory of the modeled signal. Different applications directly stem from such driver estimation.

In Section 4.1, we first describe different potential approaches to optimize the driver. Then, we propose a parametric approach to allow model evaluation on a validation set, modeling the driver as a weighted sum of potential drivers. We develop an optimization scheme based on gradient descent, and show the performances on both simulations and empirical recordings.

In Section 4.2, we tackle PAC estimation in multivariate recordings. To leverage the information contained in multiple channels, we develop an estimation based on a joint estimation of two virtual channels and a DAR model. This method builds upon the generalized eigenvalue decomposition (GED) method developed in Cohen (2017), and extend it to DAR models. In particular, it uses the driver estimation scheme described in previous section.

In Section 4.3, we show that DAR models can be considered as encoding models, where the brain activity is predicted from the stimulus. Applying DAR models on ECoG channels recorded jointly with the audio stimulus, we discuss preliminary findings and show that driver estimation naturally leads to a spectro-temporal receptive field (STRF) estimation.

This work lead to the following conference publication:

- Dupré la Tour, T., Grenier, Y., and Gramfort, A. (2018a). Driver estimation in non-linear autoregressive models. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE

As the main contributor to this project, I conducted the formal analysis, the model development, the software implementation, the experiment conceptions and visualizations, and the article writing. I am the only student who worked on the development and validation of the method.

Chapter 5: Convolutional sparse coding In this chapter, we address the present need in the neuroscience community to better capture the complex morphology of brain waves. Our approach is based on convolutional sparse coding (CSC) models, which are dictionary learning models using shift-invariant representations and strong sparsity assumptions.

We propose efficient optimization schemes leading to state-of-the-art performances. We then extend CSC models to cope with brain recordings challenges, such as severe artifacts, low signal-to-noise ratio, and long multivariate signals. These extensions are critical to be able to use CSC models on brain recordings.

In Section 5.1, we first present our CSC formulation, adding a positivity constraint on the activation to take into account the polarity of neural activations. We then propose efficient optimization schemes based on the quasi-Newton L-BFGS algorithm, which outperform previously proposed state-of-the-art ADMM-based algorithms. We also highlight critical details such as initialization schemes or possible parallel computations. Using simulation, we demonstrate the state-of-the-art speed of our optimization method. We also demonstrate the better performances of CSC models on simulated signals, compared to other state-of-the-art dictionary learning with shift-invariant representations. With empirical LFP recordings, we show that CSC models are able to extract prototypical waveforms, such as spikes or cross-frequency coupling atoms.

In Section 5.2, we propose an extension to CSC models to cope with heavy-tailed noise that can be present in brain recordings. Our models, that we call αCSC, are based on a probabilistic formulation and on α-stable distributions. To estimate these models, we develop an inference strategy based on a Monte Carlo expectation-maximization algorithm. The expectation step is performed with a Metropolis-Hasting algorithm while the maximization step corresponds to a weighted CSC estimation. Results on LFP recordings demonstrate that such algorithms can be robust to the presence of strong transient artifacts and thus reveal insights on neural time-series without supervision.

In Section 5.3, we extend CSC model to multivariate time-series, using a rank-1 constraint on the atoms to account for the instantaneous spreading of an electromagnetic source over all the channels. We also propose efficient optimization strategies, namely a locally greedy coordinate descent, and a projected gradient descent with precomputation steps for faster gradient computations. We provide multiple numerical evaluations of our method, which show the highly competitive running time on both univariate and multivariate models. The algorithm scales sub-linearly with the number of channels which means it can be employed even for dense sensor arrays with 200-300 sensors. We also demonstrate the estimation performance of the multivariate model by recovering patterns on low signal-to-noise ratio data. Finally, we illustrate our method with non-sinusoidal atoms learned on multivariate MEG data, that thanks to the rank-1 model can be localized in the brain for clinical or cognitive neuroscience studies.

This work lead to the following conference and preprint publications:

- Jas, M., Dupré la Tour, T., Şimşekli, U., and Gramfort, A. (2017). Learning the morphology of brain signals using alpha-stable convolutional sparse coding. In *Advances in Neural Information Processing Systems 30 (NIPS)*, pages 1099–1108
- Dupré la Tour, T., Moreau, T., Jas, M., and Gramfort, A. (2018b). Multivariate convolutional sparse coding for electromagnetic brain signals. In *Advances in Neural Information Processing Systems (NIPS)*

These projects were more collaborative than previous ones, and the work was not centered on a single student. In particular, in the alpha-CSC project, I was a key element in model development, software implementation, experiment conceptions and visualizations, and article writing. In the multivariate CSC project, I contributed equally with Thomas Moreau, conducting the formal analysis, the model development, the software implementation, the experiment conceptions and visualizations, and the article writing.

2

Driven autoregressive models

"What do you do for living, my lad?"
"I learn things, and I love Chloe."

– *Boris Vian*

Contents

2.1 Driven autoregressive models 30
2.1.1 Model definition 30
2.1.2 A polynomial parametrization 31
2.1.3 DAR models estimation 31
2.1.4 Model's power spectral density 34
2.1.5 An asynchronous parametrization 35
2.2 Stability in DAR models 36
2.2.1 Stability in stationary models 36
2.2.2 Stability in non-stationary AR models 38
2.2.3 Stable DAR models definition 41
2.2.4 Stable DAR models estimation 42
2.2.5 Model comparison 44
2.2.6 Model selection experiment 46

In this chapter, we introduce driven autoregressive (DAR) models. DAR models are deterministic continuously switching AR models, which use a polynomial expression $a_i(x) = \sum_{k=0}^{m} a_{ik} x^k$, and where the driver x is an exogenous signal. We also parametrize the log of the residual variance as $\log(\sigma)(x) = \sum_{k=0}^{m} b_k x^k$. We first define DAR models and develop an efficient estimation algorithm. We show how to use DAR models for conditional spectral estimation. We also discuss a different asynchronous parametrization with a more efficient estimation but with a more difficult interpretation.

Then we discuss stability in DAR model. We first recall stability properties of stationary models, then we derive a stability criterion on slowly-varying systems. We also propose another parametrization which enforces the instantaneous model to be stable at each time, and develop an efficient optimization scheme. Finally, we present an experiment to explore the limit of quickly-varying drivers in DAR models.

This chapter covers the following publication:

- Dupré la Tour, T., Grenier, Y., and Gramfort, A. (2017a). Parametric estimation of spectrum driven by an exogenous signal. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4301–4305

2.1 Driven autoregressive models

2.1.1 Model definition

Let y be a univariate locally stationary signal, as defined in Dahlhaus (1996). An auto-regressive (AR) model specifies that y depends linearly on its own p past values, where p is the *order* of the model:

$$\forall t \in [\![p+1, T]\!] \qquad y(t) + \sum_{i=1}^{p} a_i y(t-i) = \varepsilon(t) \ . \tag{2.1}$$

Here, T is the length of the signal and ε is the *innovation* (or *residual*) modeled with a Gaussian white noise: $\varepsilon(t) \sim \mathcal{N}(0, \sigma^2)$. To extend this AR model to a non-linear model, one can assume that the AR coefficients a_i are non-linear functions of a given exogenous signal x, here called the *driver*:

$$\forall i \in [\![1, p]\!] \qquad a_i(t) = g_i(x(t)) \ . \tag{2.2}$$

In TAR models (Tong and Lim, 1980), $g_i(x)$ is a piecewise constant function. In EAR models (Haggan and Ozaki, 1981), it is a sum of negative squared exponential $a_i(x) = b_i + \sum_{k=0}^{m} a_{ik} e^{-\gamma_k x^2}$. In STAR models (Chan and Tong, 1986), it is a sum of sigmoid functions $a_i(x) = \sum_{k=0}^{m} a_{ik}(1 + e^{-\gamma_k (x - c_k)})^{-1}$.

As proposed in Grenier (2013), we will consider the non-linear functions g_i as polynomials:

$$g_i(x) = \sum_{k=0}^{m} a_{ik} x^k \ , \tag{2.3}$$

where x^k is x to the power k. Inserting the time-dependent AR coefficients (2.3) into the AR model (2.1), we obtain the following equation:

$$\forall t \in [\![p+1, T]\!] \qquad y(t) + \sum_{i=1}^{p} \sum_{k=0}^{m} a_{ik} x(t)^k y(t-i) = \varepsilon(t) \ . \tag{2.4}$$

This can be simplified and rewritten as:

$$\forall t \in [\![p+1, T]\!] \qquad y(t) + A^\top \tilde{Y}(t) = \varepsilon(t) \ , \tag{2.5}$$

where $A \in \mathbb{R}^{p(m+1)}$ is a vector composed of the scalars a_{ik} and $\tilde{Y}(t) \in \mathbb{R}^{p(m+1)}$ is a vector composed of the regressors $x(t)^k y(t-i)$.

We also consider a time-varying innovation variance $\sigma(t)^2$ driven by x. This corresponds to the assumption that the power of the signal at time t depends on the driver at this same instant. Since the standard deviation is necessarily positive, we use the following polynomial model for its logarithm:

$$\forall t \in [\![p+1, T]\!] \qquad \log(\sigma(t)) = \sum_{k=0}^{m} b_k x(t)^k = B^\top X(t) \ , \tag{2.6}$$

where $B \in \mathbb{R}^{m+1}$ is a vector composed of the scalars b_k, and $X(t) \in \mathbb{R}^{m+1}$ is a vector composed of the successive powers $x(t)^k$.

We call this model a driven auto-regressive (DAR) model.

2.1.2 A polynomial parametrization

The first benefit of a polynomial parametrization is that the model estimation is computationally very efficient, as described in Subsection 2.1.3. In contrast, EAR and STAR models are more expensive to estimate. Let's recall for instance the STAR parametrization (Chan and Tong, 1986, Dijk et al., 2002), where $a_i(t)$ is a sum of sigmoid functions:

$$\forall t \in [\![p+1, T]\!] \qquad a_i(x) = \sum_{k=0}^{m} \frac{a_{ik}}{1 + e^{-\gamma_k(x - c_k)}} \quad . \tag{2.7}$$

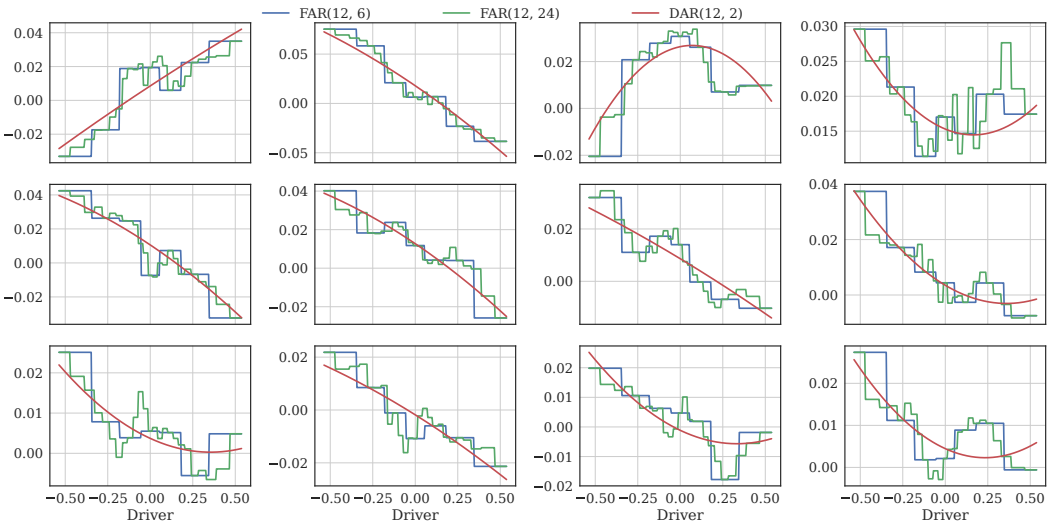
The parameters a_{ik} are easy to optimize conditionally to the rest, but there is no good initialization procedure for the transitions parameters γ_k and c_k. The non-linear optimization problem may be slow to solve and have several local minima, and the cost of an initial grid-search is large, or even prohibitive as soon as $m \geq 2$, since the grid-search is in a space of dimension $2m$.

Another benefit is the robustness of the estimation, since DAR models use the entire signal to estimate the coefficients. In contrast, a more general approach is to parametrize with piecewise constants functions, such as in functional AR (FAR) models (Chen and Tsay, 1993) and multistate TARX models (Tong, 2011). To estimate a piecewise constant AR parametrization with s-steps, the authors divide the driver's values into s equally distributed bins, then they fit s independent linear AR models on the time samples of each bin. This estimation is very fast, yet each AR model uses only T/s samples, whereas the polynomial approach uses all the T samples. The polynomial parametrization also uses fewer coefficients since a low order m is sufficient. DAR models are therefore more robust.

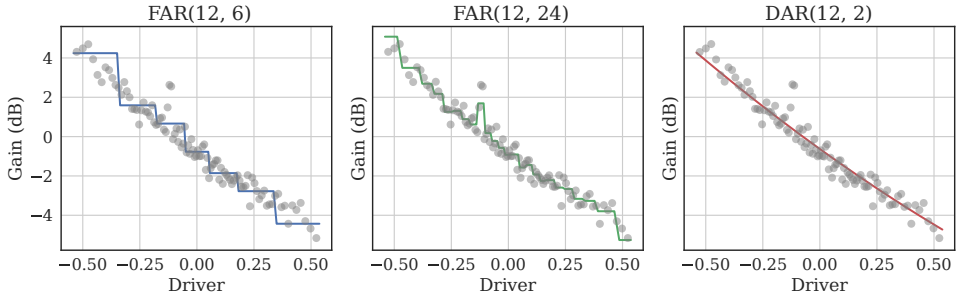
To illustrate the choice of a polynomial parametrization, we compared the functions $a_i(x)$ on an empirical ECoG signal recorded on the human auditory cortex (see datasets details in Subsection 3.1.5). We compared three parametrizations of these functions: two piecewise constant functions with respectively $s = 7$ and $s = 25$ steps, and one polynomial function with order $m = 2$, as used in the rest of this work. The models are fitted using $p = 12$. We see that an order-2 polynomial is sufficient to approximate the trend of both the AR coefficients and the innovation variance, as estimated by the piecewise constant functions. Yet the polynomial uses much fewer coefficients, since the polynomial uses $(p+1)(m+1) = 39$ coefficients, while the piecewise constant functions use $(p+1)(s+1) = 91$ (resp. 325) coefficients for $s = 7$ (resp. 25).

2.1.3 DAR models estimation

Estimation of A DAR model equation (1.4) is non-linear with respect to the given signals x and y, yet it is linear with respect to the regressors $x(t)^k y(t-i)$. Therefore after computing the regressors, it is possible to obtain an analytical expression of the parameters in A.



(a) AR coefficients $a_i(x)$ as functions of the driver x.



(b) Innovation variance $\sigma^2(x)$ as functions of the driver x. The gray circles are the mean squared-residual over 100 bins of the driver values.

Figure 2.1 – Empirical validation of the polynomial basis in DAR models. The polynomial parametrization of DAR models seems sufficient to model the trajectory of AR coefficients. It also uses much fewer degrees of freedom than piecewise-constant parametrizations.

Indeed, as the innovation $\varepsilon(t)$ is assumed to be a Gaussian white noise, the model likelihood L is obtained via:

$$L = \prod_{t=p+1}^{T} \frac{1}{\sqrt{2\pi\sigma(t)^2}} \exp\left(-\frac{\varepsilon(t)^2}{2\sigma(t)^2}\right), \tag{2.8}$$

or

$$-2\log(L) = T\log(2\pi) + \sum_{t=p+1}^{T} \frac{\varepsilon(t)^2}{\sigma(t)^2} + 2\sum_{t=p+1}^{T} \log(\sigma(t)).$$

We estimate DAR models with a maximum likelihood estimate (MLE). Here, if the innovation variance $\sigma(t)^2$ is considered fixed, maximizing L boils down to minimizing the sum of squares of $\varepsilon(t)$. Since $\varepsilon(t)$ is the residual, the inference of the parameters amounts to maximizing the variance explained by the model. To start the maximization, we first assume that the innovation variance is fixed and equal to the signal's empirical

variance. The likelihood then reads:

$$-2\log(L) = C + \sum_{t=p+1}^{T} \frac{\varepsilon(t)^2}{\sigma(t)^2} = C + \sum_{t=p+1}^{T} \frac{1}{\sigma(t)^2}\Big(y(t) + A^\top \tilde{Y}(t)\Big)^2 , \tag{2.9}$$

where C is a constant. We can thus estimate the DAR coefficients $A \in \mathbb{R}^{p(m+1)}$ by solving the following linear system (a.k.a. *normal equations*):

$$\left(\sum_{t=p+1}^{T} \frac{1}{\sigma(t)^2}\tilde{Y}(t)\tilde{Y}(t)^\top\right)\hat{A} = -\left(\sum_{t=p+1}^{T} \frac{1}{\sigma(t)^2}\tilde{Y}(t)y(t)\right) . \tag{2.10}$$

The complexity of forming the linear system is $\mathcal{O}(Tp(m+1))$, and the complexity of solving the system is $\mathcal{O}(p^3(m+1)^3)$, which is very fast since p is usually below 50, and m below 2.

Side note A common practical situation is to estimate all the models of order lower or equal to p, and to select the best model according to a specific criterion (see Section 3.2). In this case, a way to speed up the previous computation would be to use the matrix inversion lemma, also known as the Woodbury matrix identity (Woodbury, 1950):

$$(A + CBC^\top)^{-1} = A^{-1} - A^{-1}C(B^{-1} + C^\top A^{-1}C)^{-1}C^\top A^{-1} \tag{2.11}$$

where $A \in \mathbb{R}^{a\times a}$, $B \in \mathbb{R}^{b\times b}$ and $C \in \mathbb{R}^{a\times b}$. Using this formula, we could estimate the coefficients at order p, $\hat{A}^{(p)}$, using the coefficients at previous order $\hat{A}^{(p-1)}$, which only costs $\mathcal{O}((p-1)^2(m+1)^3)$. Therefore, we could estimate all models of order lower or equal to p for the same computational price as the last one. However, we do not use this method in practice, since we witnessed that accumulating numerical errors across orders would sometimes cause the estimate to diverge.

Estimation of B Given $\hat{A}$, one can then estimate the vector $B \in \mathbb{R}^{m+1}$ from the residual $\varepsilon(t)$, maximizing the likelihood using off-the-shelf optimization techniques adapted to differentiable problems. In practice, we use a Newton procedure, since the gradient and Hessian of the log-likelihood are easily obtained:

$$\frac{\partial \log L}{\partial b_k} = \sum_{t=p+1}^{T}\left(\frac{\varepsilon(t)^2}{\sigma(t)^2} - 1\right)x(t)^k \tag{2.12}$$

$$\frac{\partial^2 \log L}{\partial b_k \partial b_{k'}} = -2\sum_{t=p+1}^{T}\frac{\varepsilon(t)^2}{\sigma(t)^2}x(t)^{k+k'} \tag{2.13}$$

To give a good initialization of the Newton procedure, we use the following heuristic. We first sort the driver values $\{x(t)\}_{t=1}^{T}$, and split them into $S = 3(m+1)$ equally populated bins $\{X_s\}_{s=1}^{S}$. Then, we compute the median driver value on each bin $\tilde{x}_s = \text{median}(X_s)$. We also define the set of time points corresponding to each bin $\Omega_s = \{t : x(t) \in X_s\}$, and compute the mean squared residual on each of these sets $\tilde{v}_s = \text{mean}_{t\in\Omega_s}(\varepsilon(t)^2)$.

We now have a set of S couples $(\tilde{x}_s, \tilde{v}_s)$, which sample the function $\sigma^2 : \tilde{x}_s \mapsto \tilde{v}_s$. To obtain a feasible approximation by our polynomial parametrization (and obtain $\hat{B}$), we perform a linear regression of $\log(\tilde{v})$ on the basis $[1,\ x,\ \ldots,\ x^m]$. This procedure

Algorithm 2.1: Estimation of DAR models.

Input : Signal y, driver x, orders (p, m), number of iterations n
Initialize $\sigma(t)$ with $\sigma(t)^2 = 1/(T-p-1)\sum_{t=p+1}^{T}(y(t)-\bar{y})^2$,
for $i = 1$ **to** n **do**
  Estimate the coefficients A with (2.10),
  Compute $\varepsilon(t)$ with (2.4),
  Initialize B with the binning heuristic,
  Refine B with a Netwon procedure, using (2.12) and (2.13),
  Update $\sigma(t)$ with (2.6),
return (A, B)

gives us a good guess of the solution, making the Newton algorithm converge in few iterations.

Given such an estimate $\hat{B}$, we can update the instantaneous variance $\sigma(t)^2$ using (2.6). One can then iterate between the estimations of the coefficients A and B. In our experiments, we observed that one or two iterations are sufficient. The algorithm is summarized in Algorithm 2.1.

2.1.4 Model's power spectral density

AR models are mainly used for spectral estimation. Indeed, since an AR model is a linear filter that whiten a particular signal y, the power spectral density (PSD) of this filter is close to the inverse of the PSD of the signal, providing a parametric spectral estimation of the signal. For a linear AR model, the PSD at a frequency f is given by:

$$\mathrm{PSD}(f) = \sigma^2 \left| \sum_{i=0}^{p} a_i e^{-2j\pi f i} \right|^{-2}, \tag{2.14}$$

where $j^2 = -1$ and $a_0 = 1$. This estimation is perfect when $p \to \infty$, but in practice, using $p \in [\![10, 100]\!]$ gives satisfying results in most applications.

In the case of DAR models, the PSD is a function of the driver's value x, so we note it $\mathrm{PSD}(x)$. For a given complex driver's value $x(t_0)$, we compute the associated AR coefficients $a_i(t_0)$ using (2.2), along with the associated innovation's standard deviation $\sigma(t_0)$ using (2.6). Since AR models with time-varying coefficients are locally stationary (Dahlhaus, 1996), we can compute the PSD at a frequency f with:

$$\mathrm{PSD}(t_0)(f) = \sigma(t_0)^2 \left| \sum_{i=0}^{p} a_i(t_0) e^{-j2\pi f i} \right|^{-2}. \tag{2.15}$$

We thus have a different PSD for each driver value $x(t_0)$, *i.e.* for each time instant, as the driver fluctuates in time. We illustrate this conditional PSD in Figure 2.2, where we plot the PSD for different value of x. The four panels correspond to four models, with fixed or driven AR coefficient a_i, and fixed or driven innovation's standard deviation σ. We see how the varying AR coefficients $a_i(t)$ parametrize the fluctuations of the PSD, whereas the varying innovation's standard deviation $\sigma(t)$ is responsible for the absolute fluctuations in power over all frequencies. Interestingly, when both are driven by x, the two effects cancel each other in the non-modulated frequency bands (*e.g.* $[120 - 166]$ Hz). The model were fitted on rodent striatal LFP recordings

(see datasets details in Subsection 3.1.5), using $(p, m) = (17, 2)$. We used the fast Fourier transform (FFT) in (2.15) to compute the PSD for a broad range of frequencies $f \in [0, f_s/2]$ very quickly.

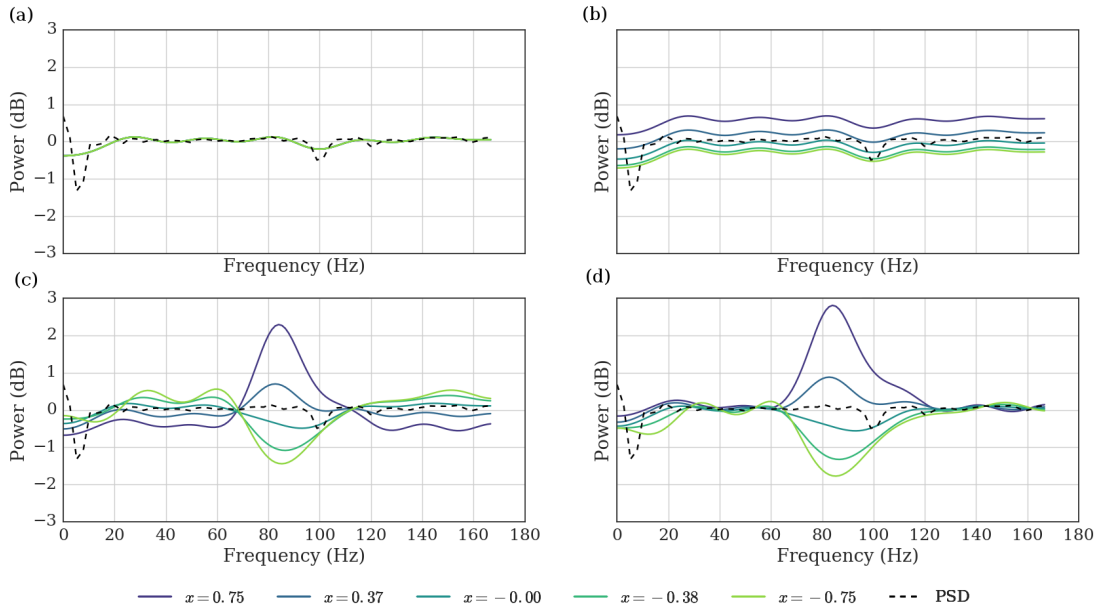


Figure 2.2 – Power spectral densities of different models. (a) With fixed a_i and fixed σ. (b) With fixed a_i and driven $\sigma(x)$. (c) With driven $a_i(x)$ and fixed σ. (d) With driven $a_i(x)$ and driven $\sigma(x)$. Each colored line corresponds to a particualr driver value. The black dashed lines is the PSD computed with a Welch method (Welch, 1967).

2.1.5 An asynchronous parametrization

Let's recall the proposed DAR parametrization:

$$\forall t \in [\![p+1, T]\!] \qquad y(t) + \sum_{i=1}^{p} \sum_{k=0}^{m} a_{ik} x(t)^k y(t-i) = \varepsilon(t) \ . \tag{2.16}$$

We see that the driver is *synchronous* with the AR model, in the sense that at time t, the AR model only depend on $x(t)$. We also studied another parametrization, where the driver's effect is asynchronous:

$$\forall t \in [\![p+1, T]\!] \qquad y(t) + \sum_{i=1}^{p} \sum_{k=0}^{m} a_{ik} x(t-i)^k y(t-i) = \varepsilon(t) \ . \tag{2.17}$$

We call this model an asynchronous DAR model.

Asynchronous DAR models are slightly more difficult to interpret than DAR models, since the relationship between the PSD and the driver is not instantaneous, and in the rest of this work, we only used DAR models. However, their parametrization leads to a faster estimation method, since we can rewrite them into simple vector AR (VAR) models:

$$\forall t \in [\![p+1, T]\!] \qquad Y(t) + \sum_{i=1}^{p} A_i Y(t-i) = \varepsilon(t) \ , \tag{2.18}$$

where $A_i = [a_{i0}, \dots, a_{im}]$ and $Y(t) = [y(t), y(t)x(t), \dots, y(t)x(t)^m]^\top$. VAR models can be efficiently estimated with Levinson-Durbin recursion (Durbin, 1960), where the linear system inversion has a complexity of $\mathcal{O}(n^2)$ instead of $\mathcal{O}(n^3)$ (with $n = p(m+1)$). This complexity improvement can be critical for large models, but as we used models with limited size ($p(m+1) < 200$), we preferred to keep the interpretability of synchronous DAR models.

2.2 Stability in DAR models

In this section, we discuss stability in DAR models. We first recall standard stability definitions and criteria for stationary AR models. We then extend them to non-stationary AR models, assuming that all instantaneous AR models are stable, and that the system is slowly varying. Finally, we describe an alternative DAR parametrization, which forces all instantaneous AR models to be stable. For these models, we propose an efficient estimation scheme based on a Netwon procedure with a good initialization.

2.2.1 Stability in stationary models

Let's first define the notion of stability:

Definition 2.1. *(Rugh, 1996, Definition 6.5)*
Suppose a non-stationary linear system:

$$X(t+1) = A(t)X(t), \quad X(t_0) = X_0, \tag{2.19}$$

where $X \in \mathbb{R}^n$ is a state vector, and $A \in \mathbb{R}^{n\times n}$ is a transition matrix.
This system is called uniform exponentially stable *if there exist $\gamma > 0$, $\lambda > 0$ such that for any t_0 and X_0, the solution satisfies:*

$$\|X(t)\| \leq \gamma e^{-\lambda(t-t_0)}\|X_0\|. \tag{2.20}$$

Another notion of stability can also be used in the context of input-output systems:

Definition 2.2. *(Rugh, 1996, Definition 12.1)*
Suppose a non-stationary linear system:

$$\begin{aligned} X(t+1) &= A(t)X(t) + B(t)U(t), \quad X(t_0) = X_0, \\ Y(t) &= C(t)X(t). \end{aligned} \tag{2.21}$$

where $X \in \mathbb{R}^n$ is a state vector, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$ are transition matrices, $U \in \mathbb{R}^m$ is the input vector, and $Y \in \mathbb{R}^p$ is the output vector.
This system is called uniform bounded-input bounded-output (BIBO) stable *if there exists a finite $\eta > 0$ such that for any input signal $U(t)$, the solution satisfies:*

$$\sup_{t\geq t_0} \|Y(t)\| \leq \eta \sup_{t\geq t_0} \|U(t)\|. \tag{2.22}$$

Filter poles A standard way to estimate stability in (stationary) AR models is to compute the poles of the AR linear filter $H(z) = \sigma^2/P(z)$ with $P(z) = 1 + \sum_{i=1}^p a_i z^{-i}$. A linear filter is stable if and only if all its poles λ_i are strictly inside the unit disk: $\forall i \in [1, p],\ |\lambda_i| < 1$. However, poles computation can be quite expensive, and we do not use this method in practice.

Lattice representation Another standard technique to assess stability in AR models is to consider the partial correlation coefficients k_p, introduced in the fast Levinson-Durbin algorithm (Durbin, 1960). This algorithm estimates the a_i coefficients recursively from previous orders: the a_i at order p (noted here $a_i^{(p)}$) can be computed from k_p and from the AR coefficients at previous order $a_i^{(p-1)}$:

$$a_p^{(p)} = k_p; \qquad \forall i \in [\![1, p-1]\!],\ a_i^{(p)} = a_i^{(p-1)} + k_p a_{p-i}^{(p-1)}. \tag{2.23}$$

Therefore, considering the sets $\{a_i\}_{i=1}^p$ and $\{k_i\}_{i=1}^p$, the knowledge of one set allows to compute the other. The latter is called the lattice representation, and brings a simple necessary and sufficient condition for stability (Makhoul, 1977):

$$\forall i \in [\![1, p]\!] \qquad -1 < k_i < 1. \tag{2.24}$$

This criterion is very useful, as the standard lattice parameter estimations naturally lead to parameters in that range. We will also use this criterion in Subsection 2.2.3 to define locally stable DAR models.

Estimation of lattice parameters The lattice parameters are generally estimated through the forward and backward residuals ε^+ and ε^-. At the output of the i-th lattice cell, the residuals are given by:

$$\begin{array}{c} \forall i \in [\![1, p]\!] \\ \forall t \in [\![i+1, T]\!] \end{array} \quad \left\{ \begin{array}{rcl} \varepsilon_i^+(t) & = & \varepsilon_{i-1}^+(t) + k_i \varepsilon_{i-1}^-(t-1), \\ \varepsilon_i^-(t) & = & \varepsilon_{i-1}^-(t-1) + k_i \varepsilon_{i-1}^+(t), \end{array} \right. \tag{2.25}$$

where we initialize with $\varepsilon_0^+(t) = \varepsilon_0^-(t) = y(t)$. Let's also define the following quantities:

$$\forall i \in [\![1, p]\!] \qquad C_i = \mathbf{E}[\varepsilon_i^+(t)\varepsilon_i^-(t-1)], \tag{2.26}$$

$$\forall i \in [\![1, p]\!] \qquad F_i = \mathbf{E}[\varepsilon_i^+(t)^2], \tag{2.27}$$

$$\forall i \in [\![1, p]\!] \qquad B_i = \mathbf{E}[\varepsilon_i^-(t)^2], \tag{2.28}$$

where $\mathbf{E}[x(t)]$ is the expected value of $x(t)$. To estimate the lattice parameter k_i of cell i, several methods exist (Makhoul, 1977):

- Minimizing F_i leads to $k_i^f = -C_{i-1}/B_{i-1}$
- Minimizing B_i leads to $k_i^b = -C_{i-1}/F_{i-1}$
- Minimizing $F_i + B_i$ leads to $k_i^{fb} = -2C_{i-1}/(F_{i-1} + B_{i-1})$
- Another common definition is $k_i^s = -C_{i-1}/\sqrt{F_{i-1}B_{i-1}}$ (not linked to a least square minimization)

The last two definitions guarantee stability since $|k_i^{fb}| \leq |k_i^s| < 1$ (Makhoul, 1977).

State-space representation Another way to assess stability is to write an AR model through a state-space representation:

$$X(t+1) = AX(t) + B\varepsilon(t), \tag{2.29}$$

$$y(t) = CX(t). \tag{2.30}$$

where $A \in \mathbb{R}^{p\times p}$, $B \in \mathbb{R}^{p\times 1}$, $C \in \mathbb{R}^{1\times p}$, $X(t) \in \mathbb{R}^{p\times 1}$, $y(t) \in \mathbb{R}$, and $\varepsilon(t) \in \mathbb{R}$.

Such a state-space model is stable if and only if the spectral radius of A is strictly lower than 1, *i.e.* all the eigenvalue of A are in the interior of the unit disk. This holds for both uniformly exponentially stable (Rugh, 1996, Theorem 22.11) and uniformly BIBO stable (Rugh, 1996, Theorem 27.9).

The stationary AR model is equivalent to two canonical forms of state-space representation: the observable and the controllable form.

The *observable* canonical form is:

$$A(t) = \begin{bmatrix} -a_1 & 1 & 0 & \cdots & 0 \\ -a_2 & 0 & 1 & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ -a_{p-1} & 0 & \cdots & 0 & 1 \\ -a_p & 0 & \cdots & 0 & 0 \end{bmatrix}, \quad B(t) = \begin{bmatrix} \sigma(t)^2 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \quad C(t)^\top = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \tag{2.31}$$

and the *controllable* canonical form is:

$$\bar{A}(t) = \begin{bmatrix} -a_1 & -a_2 & \cdots & -a_{p-1} & -a_p \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}, \quad \bar{B}(t) = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \quad \bar{C}(t)^\top = \begin{bmatrix} \sigma(t)^2 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}. \tag{2.32}$$

The eigenvalues of A are the roots of the polynomial:

$$\det(\lambda I_p - A) = \lambda^p + \sum_{i=1}^{p} a_i \lambda^{p-i}, \tag{2.33}$$

where we recognize the AR filter polynomial $P(\lambda)$ multiplied by λ^p. We see that this is nothing more than considering the poles of the linear AR filter. A is sometime called the *companion matrix* of P, as its *characteristic polynomial* is P. Note that in the stationary case, $A = \bar{A}^\top$, so both characteristic polynomials are identical. We will use this representation in the next subsection to discuss stability in non-stationary AR models.

2.2.2 Stability in non-stationary AR models

Counter-examples The stability properties described in the previous section are only valid in stationary AR models. In particular, the overall system might be unstable even though all instantaneous filters are stable. For instance, in the case of state-space systems, we have the counter-example (Rugh, 1996, Example 24.1):

$$A(t) = \begin{cases} \begin{pmatrix} 0 & 2 \\ 1/4 & 0 \end{pmatrix}, \text{if } t \text{ is odd}, \\ \begin{pmatrix} 0 & 1/4 \\ 2 & 0 \end{pmatrix}, \text{if } t \text{ is even}. \end{cases} \tag{2.34}$$

This system is not stable, yet the poles $\pm 1/\sqrt{2}$ are inside the unit disk at all time.

Properties of slowly-varying linear systems In order to cope with stability in non-stationary AR models, we will make use of the following results:

Theorem 2.3. *(Desoer, 1970) or (Rugh, 1996, Theorem 24.8)*
Suppose a non-stationary linear system:

$$X(t+1) = A(t)X(t), \quad X(t_0) = X_0. \tag{2.35}$$

Suppose that there exists $\alpha > 0$ and $0 \leq \mu < 1$ such that for all t $\|A(t)\| \leq \alpha$ and for all eigenvalues $\lambda_i(t)$ of $A(t)$, $|\lambda_i(t)| \leq \mu$. Then there exists $\beta > 0$ such that if $\|A(t+1) - A(t)\| < \beta$ for all t, then the system (2.35) is uniformly exponentially stable.

In other words, if the system is bounded and instantaneously stable at all time, with poles poles bounded away from the unit circle, and if the system is varying sufficiently slowly, then we have uniform exponential stability. This result is complemented by the following theorem:

Theorem 2.4. *(Rugh, 1996, Theorem 27.4)*
Suppose a non-stationary linear system:

$$\begin{aligned} X(t+1) &= A(t)X(t) + B(t)u(t), \quad X(t_0) = X_0, \\ y(t) &= C(t)X(t). \end{aligned} \tag{2.36}$$

If the system is uniformly exponentially stable, and if $\|B(t)\| \leq \gamma$ and $\|C(t)\| \leq \zeta$ for all t, then (2.36) is also uniformly BIBO stable.

This theorem extends uniform exponential stability to uniform BIBO stability, under appropriate bounding hypotheses. To apply these two theorems to non-stationary AR models, we will also use a result on perturbed systems:

Theorem 2.5. *(Rugh, 1996, Theorem 24.7)*
Suppose a non-stationary linear system:

$$X(t+1) = A(t)X(t), \quad X(t_0) = X_0. \tag{2.37}$$

If the system is uniformly exponentially stable, then there exists $\beta > 0$ such that for a perturbation $F \in \mathbb{R}^{p \times p}$, if $\sup \|F(t)\| \leq \beta$, then the perturbed system:

$$X(t+1) = (A(t) + F(t))X(t), \quad X(t_0) = X_0, \tag{2.38}$$

is uniformly exponentially stable.

Essentially, this theorem states that uniform exponential stability is still valid on a neighborhood of the system.

Stability in non-stationary AR models Given previous theorems, let's go back to the *observable* state-space representation of the non-stationary AR model:

$$A(t) = \begin{bmatrix} -a_1(t+1) & 1 & 0 & \cdots & 0 \\ -a_2(t+2) & 0 & 1 & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ -a_{p-1}(t+p-1) & 0 & \cdots & 0 & 1 \\ -a_p(t+p) & 0 & \cdots & 0 & 0 \end{bmatrix}, \quad B(t) = \begin{bmatrix} \sigma(t)^2 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \quad C(t)^\top = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}. \tag{2.39}$$

Importantly, this observable representation is equivalent to the non-stationary AR model (Grenier, 1984) (but not the controllable one). Let's assume we have a non-stationary AR model whose instantaneous AR models are stable at all time. We can be tempted to use the Theorem 2.3 and Theorem 2.4, so as to prove uniform exponential and uniform BIBO stability. However, one can see that the polynomial is *not* the same as in the instantaneous filter:

$$P(t,z) = 1 + \sum_{i=1}^{p} a_i(t) z^{-i}, \tag{2.40}$$

$$\det(\lambda_i I_p - A(t)) = \lambda^p + \sum_{i=1}^{p} a_i(t+i)\lambda^{p-i}. \tag{2.41}$$

In the first we have $a_i(t)$, and in the second $a_i(t+i)$. Thus, even if the poles of the instantaneous filter are in the unit disk for all time, the poles of $A(t)$ can be outside the unit disk (see an example in (Grenier, 1984, Part 2 - Annexe 8)).

However, we can assume that if the $a_i(t)$ are changing slowly, the roots of both polynomials should be close. This assumption is not trivial, since polynomial's roots can be *very* sensitive to a variation in polynomial's coefficients (*e.g.* Wilkinson's polynomial (Wilkinson, 1984)).

In order to evaluate the variations of the roots of the two polynomials, let's define a new matrix $\tilde{A}$:

$$\tilde{A}(t) = \begin{bmatrix} -a_1(t) & 1 & 0 & \cdots & 0 \\ -a_2(t) & 0 & 1 & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ -a_{p-1}(t) & 0 & \cdots & 0 & 1 \\ -a_p(t) & 0 & \cdots & 0 & 0 \end{bmatrix}, \tag{2.42}$$

and the perturbation matrix:

$$F(t) = A(t) - \tilde{A}(t) = \begin{bmatrix} a_1(t) - a_1(t+1) & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_p(t) - a_p(t+p) & 0 & \cdots & 0 \end{bmatrix}. \tag{2.43}$$

Under necessary hypotheses, Theorem 2.3 states that the system $X(t+1) = \tilde{A}(t)X(t)$ is uniformly exponentially stable. Then Theorem 2.5 states that if $\|F(t)\|$ is sufficiently small, *i.e.* the coefficients $a_i(t)$ vary sufficiently slowly, then the system $X(t+1) = A(t)X(t) = (\tilde{A}(t) + F(t))X(t)$ is also uniformly exponentially stable. With appropriate hypotheses, Theorem 2.4 extends it to uniform BIBO stability.

In other words, if the non-stationary AR model is instantaneously stable at all time, with poles bounded away from the unit circle, and *if the system is varying sufficiently slowly*, then the overall system is stable.

Controllable form

Note that in non-stationary AR models, the *controllable* canonical form uses

$$\bar{A}(t) = \begin{bmatrix} -a_1(t) & -a_2(t) & \cdots & -a_{p-1}(t) & -a_p(t) \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}, \tag{2.44}$$

which involves the same poles than the instantaneous AR filter, which could be promising. However, there is no equivalence between the controllable form and the AR representation in the general case. A sufficient condition for equivalence is if this form is also observable (Grenier, 1984, Part 1 - Chapter 3 - Section 4.2.3).

Variant: non-instantaneous DAR models Let's recall that in niDAR models, we have changed the coefficients $a_i(t)$ into $a_i(t-i)$. Then the *observable* state-space is changed into:

$$A(t) = \begin{bmatrix} -a_1(t) & 1 & 0 & \cdots & 0 \\ -a_2(t) & 0 & 1 & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ -a_{p-1}(t) & 0 & \cdots & 0 & 1 \\ -a_p(t) & 0 & \cdots & 0 & 0 \end{bmatrix}, \quad B(t) = \begin{bmatrix} \sigma(t)^2 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \quad C(t)^\top = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}. \tag{2.45}$$

Here we have a more instantaneous spectrum for $A(t)$. However, this asynchronous AR parametrization is not compatible with the log-area ratios described in next subsection, so we are unable to guarantee the poles of the instantaneous filter to be strictly inside the unit disk.

2.2.3 Stable DAR models definition

In previous subsection, we use the hypothesis that the instantaneous AR models are stable at each time. However, this is not a property of DAR models presented earlier. In fact, the polynomial functions $a_i(x)$ are not bounded, so a large driver value $x(t)$ can easily lead to an unstable AR model. This is not a problem if we use DAR models only for analysis, but can be an issue if we use fitted DAR models to generate new signals.

To tackle this issue, we present here a different parametrization based on Grenier and Omnes-Chevalier (1988), which uses lattice coefficients (*c.f.* Section 2.2.1), and log-area ratios (Wakita, 1973), which were developed in acoustic models of the physics of sound propagation in the vocal tract.

First, let's recall the correspondence between AR coefficients and lattice coefficients:

$$a_p^{(p)} = k_p; \qquad \forall i \in [\![1, p-1]\!],\ a_i^{(p)} = a_i^{(p-1)} + k_p a_{p-i}^{(p-1)}. \tag{2.46}$$

We can define a different non-stationary parametrization, where the $k_i(t)$ are directly driven by the exogenous signal x, instead of the $a_i(t)$:

$$\forall i \in [\![1, p]\!] \qquad k_i(t) = \sum_{j=0}^{m} k_{ij} x(t)^j = K_i^\top X(t). \tag{2.47}$$

This parametrization is used as an intermediate step during model estimation. We recall that such lattice representation brings a simple condition (necessary and sufficient) for stability: $-1 < k_i < 1$ (Benesty et al., 2007). To enforce this condition, we use the log-area ratios γ_i (Wakita, 1973), as suggested in (Grenier and Omnes-Chevalier, 1988):

$$\gamma_i = \log\left(\frac{1+k_i}{1-k_i}\right) \iff k_i = \frac{e^{\gamma_i}-1}{e^{\gamma_i}+1}. \tag{2.48}$$

We use the log-area ratios coefficients $\gamma_i(t)$ in order to force the $k_i(t)$ to be within $]-1,1[$, which enforces the stability of the instantaneous model for all t. The corresponding non-stationary parametrization is:

$$\forall i \in [\![1,p]\!] \qquad \gamma_i(t) = \sum_{j=0}^{m} \gamma_{ij} x(t)^j = \Gamma_i^\top X(t). \tag{2.49}$$

We call this model a stable DAR model. Note that this parametrization only enforces stability of each *instantaneous* AR models. The global stability is only given for models which are varying sufficiently slowly.

2.2.4 Stable DAR models estimation

Once again, we estimate the log-area ratios coefficients Γ_i by maximizing the likelihood. As in DAR models, we first assume that the innovation variance is constant: $\sigma(t)^2 = \sigma^2$, and equal to the signal's empirical variance. Then, we maximize the likelihood iteratively from order 1 to order p, by adding each time a lattice cell. So for each lattice cell from $i = 1$ to p, we estimate the log-area ratios coefficients $\Gamma_i \in \mathbb{R}^{m+1}$ by minimizing a least-square criterion over the forward and backward residuals:

$$J(\Gamma_i) = \sum_{t=i+1}^{T} \frac{\varepsilon_i^+(t)^2 + \varepsilon_i^-(t)^2}{\sigma(t)^2}, \tag{2.50}$$

where the forward and backward residuals at the output of the i-th lattice cell are given by (as (2.25)):

$$\begin{array}{c} \forall i \in [\![1,p]\!] \\ \forall t \in [\![i+1,T]\!] \end{array} \qquad \left\{ \begin{array}{rcl} \varepsilon_i^+(t) & = & \varepsilon_{i-1}^+(t) + k_i \varepsilon_{i-1}^-(t-1), \\ \varepsilon_i^-(t) & = & \varepsilon_{i-1}^-(t-1) + k_i \varepsilon_{i-1}^+(t). \end{array} \right. \tag{2.51}$$

Note that maximizing the likelihood corresponds to minimizing the least-square criterion only on the forward residual. However, minimizing both residual is a standard way to improve the robustness of the estimation (Makhoul, 1977).

We minimize $J(\Gamma_i)$ with a Newton procedure since the gradient and Hessian can be computed easily.

Gradient The first partial derivative G_i is obtained via:

$$[G_i]_j = \frac{\partial J}{\partial \gamma_{ij}} = 2 \sum_{t=i+1}^{T} \frac{1}{\sigma(t)^2} \left(\varepsilon_i^+(t) \frac{\partial \varepsilon_i^+(t)}{\partial k_i(t)} + \varepsilon_i^-(t) \frac{\partial \varepsilon_i^-(t)}{\partial k_i(t)} \right) \frac{\partial k_i(t)}{\partial \gamma_{ij}} \tag{2.52}$$

$$= 2 \sum_{t=i+1}^{T} \frac{1}{\sigma(t)^2} \left(\varepsilon_i^+(t) \varepsilon_{i-1}^-(t-1) + \varepsilon_i^-(t) \varepsilon_{i-1}^+(t) \right) \frac{\partial k_i(t)}{\partial \gamma_{ij}}, \tag{2.53}$$

with:

$$\frac{\partial k_i(t)}{\partial \gamma_{ij}} = \frac{\partial k_i(t)}{\partial \gamma_i(t)} \frac{\partial \gamma_i(t)}{\partial \gamma_{ij}} = \frac{1}{2}(1 - k_i(t)^2)x(t)^j. \tag{2.54}$$

We can rewrite it in a more compact vector form:

$$G_i = \sum_{t=i+1}^{T} \frac{g_i(t)}{\sigma(t)^2} X(t), \tag{2.55}$$

with:

$$g_i(t) = \left(\varepsilon_i^+(t)\varepsilon_{i-1}^-(t-1) + \varepsilon_i^-(t)\varepsilon_{i-1}^+(t)\right)(1 - k_i(t)^2). \tag{2.56}$$

Since $\varepsilon_i^+(t)$ and $\varepsilon_i^-(t)$ are not known, we need to express them as functions of $\varepsilon_{i-1}^+(t)$ and $\varepsilon_{i-1}^-(t)$. These functions are given in (2.51). We have then:

$$g_i(t) = \left(2\varepsilon_{i-1}^+(t)\varepsilon_{i-1}^-(t-1) + k_i(t)(\varepsilon_{i-1}^-(t-1)^2 + \varepsilon_{i-1}^+(t)^2)\right)(1 - k_i(t)^2) \tag{2.57}$$

To evaluate this formula, $\tilde{k}_i(t)$ is computed from $\hat{\gamma}_i(t) = \hat{\Gamma}_i^\top X(t)$ with the log-area ratios formula (2.48).

Hessian The second partial derivative H_i is obtained via:

$$[H_i]_{jj'} = \frac{\partial^2 J}{\partial \gamma_{ij} \partial \gamma_{ij'}} = \frac{\partial}{\partial \gamma_{ij}} \left(\sum_{t=i+1}^{T} \frac{g_i(t)}{\sigma(t)^2} x(t)^{j'} \right) \tag{2.58}$$

$$= \sum_{t=i+1}^{T} \frac{1}{\sigma(t)^2} \frac{\partial g_i(t)}{\partial k_i(t)} \frac{\partial k_i(t)}{\partial \gamma_{ij}} x(t)^{j'}, \tag{2.59}$$

with:

$$\begin{aligned} \frac{\partial g_i(t)}{\partial k_i(t)} = &\left(2\varepsilon_{i-1}^+(t)\varepsilon_{i-1}^-(t-1) + k_i(t)(\varepsilon_{i-1}^-(t-1)^2 + \varepsilon_{i-1}^+(t)^2)\right)(-2k_i(t)) \\ &+ \left(\varepsilon_{i-1}^-(t-1)^2 + \varepsilon_{i-1}^+(t)^2\right)(1 - k_i(t)^2). \end{aligned} \tag{2.60}$$

That we can rewrite:

$$H_i = \sum_{t=i+1}^{T} \frac{h_i(t)}{\sigma(t)^2} X(t)X(t)^\top, \tag{2.61}$$

with:

$$\begin{aligned} h_i(t) = (1 - k_i(t)^2)((&\varepsilon_{i-1}^-(t-1)^2 + \varepsilon_{i-1}^+(t)^2)(1 - 3k_i(t)^2) \\ &- 4\varepsilon_{i-1}^+(t)\varepsilon_{i-1}^-(t-1)k_i(t)). \end{aligned} \tag{2.62}$$

Once again, to evaluate this formula, $\tilde{k}_i(t)$ is computed from $\hat{\gamma}_i(t) = \hat{\Gamma}_i^\top X(t)$ using the log-area ratios formula (2.48).

Initialization To start with a good initialization, we approximate the problem with a two-step algorithm. In the first step, we solve the linear problem $\operatorname{argmin}_{K_i} J(K_i)$ using the lattice parametrization (2.47). It leads to a *normal equation*:

$$\left(\sum_{t=i+1}^{T} \frac{(\varepsilon_{i-1}^{-}(t-1)^2 + \varepsilon_{i-1}^{+}(t)^2)}{\sigma(t)^2} X(t)X(t)^\top\right)\hat{K}_i = \\ -\sum_{t=i+1}^{T} \frac{2}{\sigma(t)^2}\varepsilon_{i-1}^{+}(t)\varepsilon_{i-1}^{-}(t-1)X(t). \tag{2.63}$$

In the second step, we estimate Γ_i with a regression over the trajectories $\gamma_i(t)$ and $\hat{k}_i(t)$: After clipping $\hat{k}_i(t) = \hat{K}_i^\top X(t)$ inside $[-1+\eta, 1-\eta]$, with $\eta = 10^{-6}$ for example, we measure the quality of the approximation with a second least-squares criterion:

$$J_2(\Gamma_i) = \sum_{t=i+1}^{T} \left(\Gamma_i^\top X(t) - \log\left(\frac{1+\hat{k}_i(t)}{1-\hat{k}_i(t)}\right)\right)^2. \tag{2.64}$$

Again this linear system leads to a normal equation:

$$\left(\sum_{t=i+1}^{T} X(t)X(t)^\top\right)\hat{\Gamma}_i = \sum_{t=i+1}^{T} X(t)\log\left(\frac{1+\hat{k}_i(t)}{1-\hat{k}_i(t)}\right). \tag{2.65}$$

These two steps are used to initialize the Newton procedure, and largely speed-up the optimization.

We iterate this optimization over each lattice cell i: For $i > 1$, the residuals $\varepsilon_i^+(t)$ and $\varepsilon_i^-(t)$ are obtained with (2.51) using the $\tilde{k}_i(t)$ derived from $\hat{\gamma}_i(t) = \hat{\Gamma}_i^\top X(t)$ using (2.48). We then obtain the DAR coefficients Γ_i for every order i from 1 to p. The entire algorithm is given in Algorithm 2.2.

2.2.5 Model comparison

During our experiments, we compared extensively DAR models, stable DAR models, and asynchronous DAR models. We did not notice any significant differences in modeling between DAR models and stable DAR models. Asynchronous DAR models yielded slightly different likelihood scores with no clear trend, but they were more difficult to interpret than DAR models.

As an example, we present in Figure 2.3 the PSD of four models, fitted on rodent striatal LFP recordings (see datasets details in Subsection 3.1.5), using $(p, m) = (17, 2)$. We compared a standard DAR model as in (2.4), an asynchronous DAR model as in (2.17), a DAR model with a lattice parametrization as in (2.47), and a stable DAR model as in (2.49). The PSD are almost identical.

However, the computational costs were quite different (see Table 2.1 for some example of fitting times). DAR models were the fastest, beating slightly asynchronous DAR models. Their poorer complexity in p might inverse the trend for large p though. Stable DAR models were significantly slower.

Due to speed results and non-significant modeling differences, we gradually moved to using only standard DAR models. In the rest of this manuscript, we only present results on standard DAR models.

Algorithm 2.2: Estimation of log-area ratios coefficients Γ_i in stable DAR models.

Input : Signal y, driver x, orders (p, m), threshold η, threshold h
Compute $X = [1,\ x, \ldots,\ x^m]$ and $\sigma(t)$,
Initialize $\varepsilon_0^+ = \varepsilon_0^- = y$,
for $i = 1$ **to** p **do**
  Step 1: Compute $\hat{K}_i$ with normal equation (2.63),
  Expand $\hat{k}_i(t) = \hat{K}_i^\top X(t)$ for all $t \geq i$,
  Crop $\hat{k}_i(t)$ inside $[-1 + \eta, 1 - \eta]$ for all $t \geq i$,
  Step 2: Initialize $\hat{\Gamma}_i$ with normal equation (2.65),
  repeat
    Expand $\hat{\gamma}_i(t) = \hat{\Gamma}_i^\top X(t)$ for all $t \geq i$,
    Compute $\tilde{k}_i(t)$ using (2.48) and $\hat{\gamma}_i(t)$, for all $t \geq i$,
    Compute the gradient G_i with (2.55), and the Hessian H_i with (2.61),
    Update $\hat{\Gamma}_i$ with a Newton step,
  until $\max_j \Delta\hat{\Gamma}_{ij} < h$;
  Expand $\hat{\gamma}_i(t) = \hat{\Gamma}_i^\top X(t)$ for all $t \geq i$,
  Compute $\tilde{k}_i(t)$ using (2.48) and $\hat{\gamma}_i(t)$, for all $t \geq i$,
  Compute $\varepsilon_i^+(t)$ and $\varepsilon_i^-(t)$ using (2.51), for all $t \geq i$,
return $\{\hat{\Gamma}_i\}_{i=1}^p$

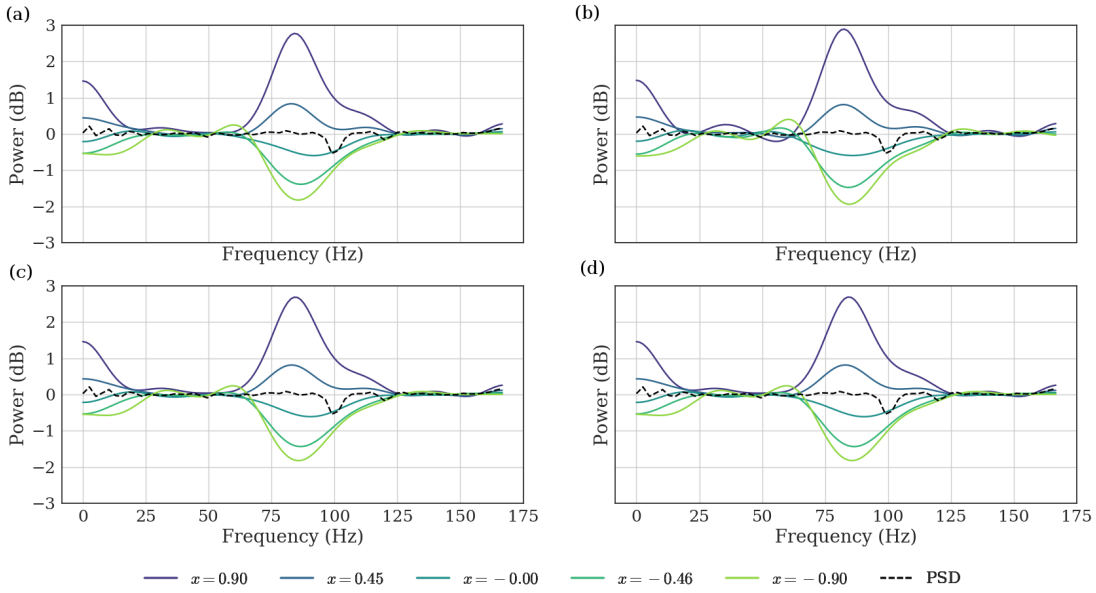


Figure 2.3 – Power spectral densities of different models. All models give extremely similar results. (a) A DAR model (2.4). (b) An asynchronous DAR model (2.17). (c) A DAR model with lattice parametrization (2.47). (d) A stable DAR model (2.49). Each colored line corresponds to a particualr driver value. The black dashed lines is the PSD computed with a Welch method (Welch, 1967).

Table 2.1 – Computational fitting times (in seconds) of several models and orders (p, m), fitted on a long rodent striatal LFP recording with $T = 600,000$ points.

Model \ order	**(17, 1)**	**(17, 2)**	**(100, 1)**	**(100, 2)**
DAR	0.19	0.28	1.48	2.53
Asynchronous DAR	0.29	0.44	1.59	2.55
Lattice DAR	0.94	1.41	5.49	7.76
Stable DAR	1.58	2.04	9.20	11.49

2.2.6 Model selection experiment

In Subsection 2.2.2, we describe sufficient conditions for non-linear AR models to be stable. One of the conditions is that the driver needs to vary sufficiently slowly. On the one hand, this condition is rather reasonable. On the other hand, we can imagine examples of stable models with a quickly varying driver. A better question would be: does extending the stationary AR model really make sense with a quickly-varying driver? To empirically tackle this question, we developed a simulation study based on model selection.

Simulations First, we generated a driving signal x for $T = 10^4$ time points, by upsampling a Gaussian white noise of length Tf_x/f_s (where $f_s = 10^3$ Hz is the sampling frequency). Such a driver only contains frequencies below f_x.

Then, we created stable DAR models based on the following procedure. We drew a set of log-area ratios coefficients $\gamma_i(t)$ using random walks, starting from a real number $\gamma_i(0) \in]-1, 1[$, and computing $\gamma_i(t+1) = \gamma_i(t) + \epsilon_i(t)$ with $\epsilon_i(t) \sim \mathcal{N}(0, 0.3)$. Note that the log-area ratios coefficients do not need to be in $]-1, 1[$ to make the instantaneous AR models stable (*c.f.* Subsection 2.2.3). However, to avoid having very large coefficients during a long period of time, we rescaled the trajectory to be inside $[-4, 4]$. To make the log-area ratios coefficients dependent of the driver x, we projected them on the basis $X(t) = [1,\ x(t),\ \ldots,\ x(t)^m]^\top$:

$$\hat{\Gamma}_i = \underset{\Gamma_i \in \mathbb{R}^{m+1}}{\operatorname{argmin}} \sum_{t=1}^{T} (\Gamma_i^\top X(t) - \gamma_i(t))^2. \tag{2.66}$$

With the obtained stable DAR coefficients $\hat{\Gamma}_i$, we computed the instantaneous log-area ratios $\tilde{\gamma}_i(t) = \hat{\Gamma}_i X(t)$, and the corresponding lattice coefficients $\tilde{k}_i(t)$. We generated a signal $y(t)$ by feeding the lattice filters with a Gaussian white noise. To focus only on the spectral fluctuations, we used a constant innovation gain: $\sigma(t) = \sigma$.

We repeated this procedure 100 times, generating 100 signals from 100 DAR models with $p = 10$ and $m \in [0, 1, 2]$. We also tested it for different driver's maximal frequency f_x.

Model selection Then, on each simulated signal, we fitted multiple stable DAR models on y, using the true driver x, with $p \in [1, 20]$ and $m \in [0, 3]$. We used the Bayesian information criterion (BIC) to select the best model and the corresponding best hyper-parameters are noted $\hat{p}$ and $\hat{m}$. Model selection is described in details in Section 3.2.

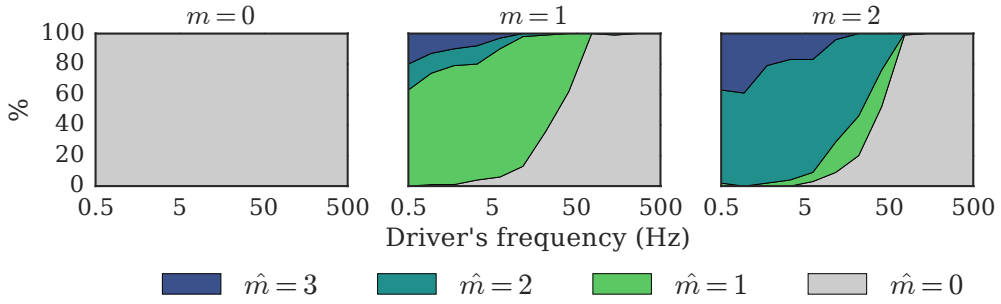


Figure 2.4 – Model selection with the Bayesian information criterion (BIC). The graphs show the proportion of each $\hat{m}$ selected, with respect to the driver's frequency. We simulated 100 signals from 100 DAR models with $p = 10$ and $m = 0$ (left), $m = 1$ (middle), $m = 2$ (right). We then estimated some DAR models on these signals, and we selected $\hat{p}$ and $\hat{m}$ that minimized the BIC. The graphs show the proportion of each $\hat{m}$ selected. The hyper-parameter m is correctly estimated in most cases if the driver's frequency is not too high ($f_x < 50\,\text{Hz}$).

Results The results are presented in Figure 2.4, where we show the proportion on each $\hat{m}$ selected by the BIC, for different driver's maximal frequencies, and different true parameter m.

For $m = 0$, the BIC selects the correct order $\hat{m} = 0$ on 100% of the simulations. In other words, with signals simulated with AR models, DAR models might better fit the signals since they have more degrees of freedom than AR models, but the BIC, which penalizes model complexity, correctly selects AR models ($\hat{m} = 0$).

For $m = 1$ and $m = 2$, we can see that for low driver's frequency, *i.e.* for slowly-varying drivers, the BIC selects the correct m in most cases. However, for high driver's frequency, *i.e.* for quickly-varying drivers, the BIC selects $\hat{m} = 0$, which corresponds to a stationary AR model. Importantly, this happens on signals simulated with DAR models, and fitted with the true drivers. These results empirically demonstrate that using DAR models with quickly-varying drivers hardly makes sense, and that stationary AR models can be preferred.

Note that when the driver's frequency is too low, the BIC sometimes overestimates m since the time length is too short to see many driver's oscillations. The BIC also selects p correctly at ± 2 (not shown) for all driver's frequencies.

3

DAR models and phase-amplitude coupling

"Then you will say to them
"Yes, the stars always make me laugh!"
and they will think you are crazy."

– Antoine de Saint-Exupéry

Contents

3.1 DAR models and PAC 50
3.1.1 Preprocessing 50
3.1.2 Phase invariant parametrization 52
3.1.3 Conditional power spectral density 52
3.1.4 Comodulogram 53
3.1.5 Datasets and simulated PAC signals 54
3.2 Model selection 55
3.2.1 Generative models 55
3.2.2 Driver filter selection 56
3.2.3 Delay selection and directionality 61
3.2.4 Amplitude of the driver 65
3.3 Statistical significance 67
3.3.1 Quantifying significance 67
3.3.2 Robustness to small samples 68
3.3.3 Spurious PAC 69
3.4 Discussion 70

In this chapter, we propose to use DAR models defined in the previous chapter to capture and characterize phase-amplitude coupling (PAC) in neurophysiological time-series. Indeed, these models are especially designed to capture spectral fluctuations coupled with a slowly-varying signal.

We first show how to adapt DAR models to analyze PAC. Then we describe in details how to perform model selection, a unique feature of PAC analysis with DAR models. Finally, we discuss statistical significance and robustness of PAC analysis with DAR models. Our method is fully available as an open source package that comes with documentation, tests, and examples: `https://pactools.github.io`.

This chapter covers the following publication:

- Dupré la Tour, T., Tallot, L., Grabot, L., Doyère, V., van Wassenhove, V., Grenier, Y., and Gramfort, A. (2017b). Non-linear auto-regressive models for cross-frequency coupling in neural time series. *PLOS Computational Biology*, 13(12)

3.1 DAR models and PAC

3.1.1 Preprocessing

In this section, we describe the preprocessing applied to the raw signal z to obtain the driver x and the signal y. The raw signal z is a univariate signal, for instance one channel of an ECoG recording.

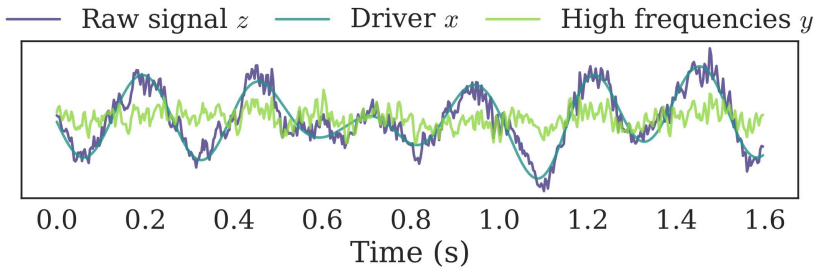


Figure 3.1 – Example of a preprocessed signal. The driver x is extracted from the raw signal z with a bandpass filter, and then subtracted from z to give y. A PAC effect is present as we see stronger high frequency oscillations at the peaks of the driver. The signal y is presented here as it looks before the temporal whitening step.

First, the raw signal is down-sampled to an appropriate sampling frequency f_s (we used 333 Hz and 625 Hz in our examples), as we only consider frequencies up to $f_s/2$. Then, power line noise and its harmonics are removed as follow. As power line frequency fluctuates over time, we decompose the signal in chunks of 5 seconds. On each chunk, the power line frequency is estimated by projecting the signal onto multiple sinusoids, and the maximum-power projection is subtracted from the raw signal. When necessary, we also use a high-pass filter to detrend the signal (Bigdely-Shamlo et al., 2015), typically at 1 Hz.

A bandpass filter is then used to extract the driver x. The choice of center frequency f_x and bandwidth Δf_x of this filter will be discussed in details in Subsection 3.2.2. The filter equation is:

$$w(t) = b(t)\cos(2\pi f_x t) \ , \tag{3.1}$$

where $b(t)$ is a Blackman window of order $\lfloor 1.65 * f_s/\Delta f_x \rfloor * 2 + 1$, chosen to have a bandwidth of Δf_x at -3 dB. In other words, the filter attenuation is 50% at $f_x \pm \Delta f_x/2$. Importantly, the filter is zero-phase since it is symmetric.

Then, the driver x is subtracted from the raw signal z to create the modeled signal $y = z - x$. Note that the signal y now contains a frequency gap around f_x, which can be a nuisance for the AR estimate that provides a compact model for the broad band power spectrum density of the signal. To solve this issue, we fill this gap by adding a Gaussian white noise filtered with the same filter w, and adjusted in energy to have a smooth power spectral density (PSD) in y. Such a preprocessing step is also commonly used when working with vector auto-regressive models (VAR) on neurophysiological signals corrected for power line noise.

Finally, we whiten y with a linear AR model, by applying the inverse AR filter to the signal. This temporal whitening step is not necessary, yet it reduces the need for high order p in DAR and therefore reduces both the computational cost and the variance of

the model. After this whitening step, the PSD of y is mostly flat, and DAR models only contain the modulation of the PSD. Figure 3.1 shows a time sample of a preprocessed signal, taken from a rodent striatal LFP signal (see datasets details in Subsection 3.1.5). One can see how the slowly varying driving signal follows the original raw signal and how the high frequencies bursting on the peak of the slow oscillation remain in the processed signal y. The general processing pipeline is summarized in Figure 3.2 (a).

In the case of analyzing PAC between two different channels A and B, we can simply perform the previous steps on both raw signals z_A and z_B, and fit a DAR model on y_A driven by x_B. It could also be possible to not subtract x_A from z_A, if the low frequency bands x_A and x_B are assumed to be uncorrelated, which might not be the case. In such cases, the other preprocessing steps should still be performed on z_A.

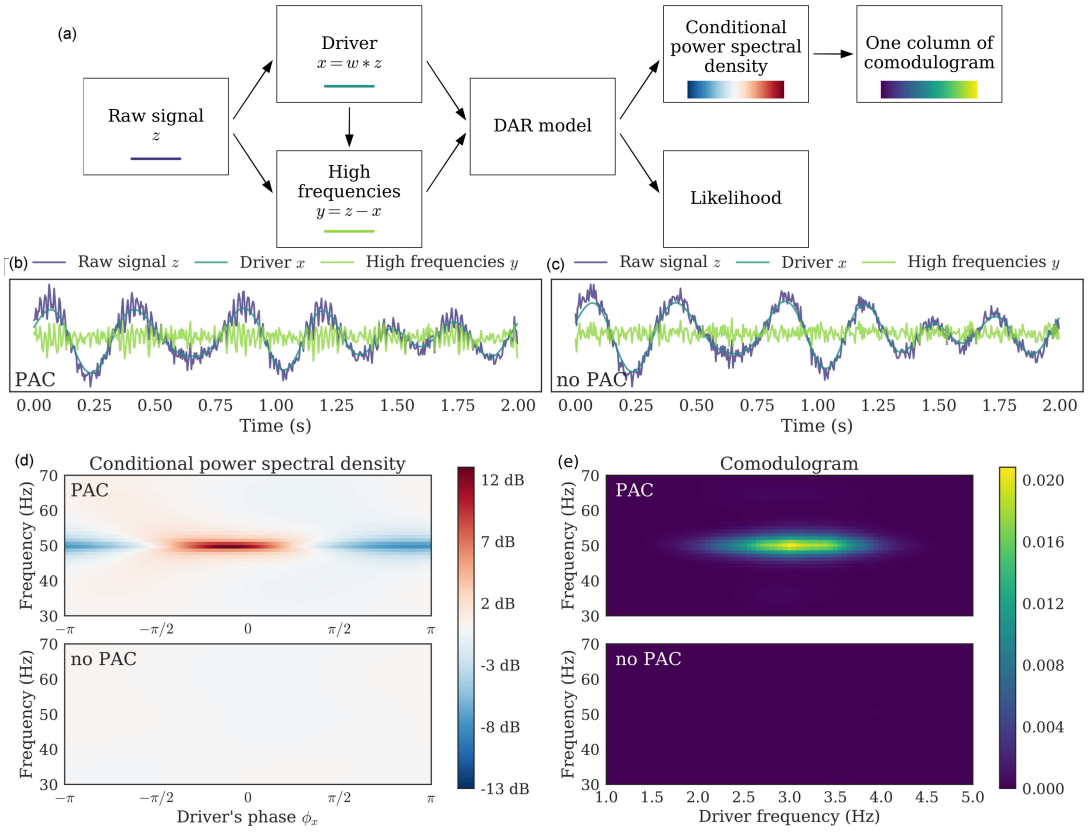


Figure 3.2 – Pipeline, signals, conditional PSD, and comodulograms. (a) Pipeline of the method. We applied it with $(p, m) = (10, 1)$ on two simulated signals: (b) Simulated signal with PAC and (c) Simulated signal without PAC. (d) From a fitted model, we computed the PSD conditionally to the driver's phase ϕ_x; each line is centered independently to show amplitude modulation. PAC can be identified in the fluctuation of the PSD as the driver's phase varies: around 50 Hz, the PSD has more power for one phase than for another. This figure corresponds to a single driver's frequency $f_x = 3.0\,\text{Hz}$. (e) Applying this method to a range of driver's frequency, we build a comodulogram, which quantifies the PAC between each pair of frequencies.

3.1.2 Phase invariant parametrization

Let's denote by ϕ_0 the *preferred phase* of the coupling, *i.e.* the phase of the driver that corresponds to the maximum amplitude of the high frequencies. When $\phi_0 = 0$, the high frequency bursts happen in the peaks of the driver, whereas when $\phi_0 = \pi$, they happen in the troughs.

The driver extracted as described in previous subsection is real-valued, and the filter used to extract it is based on a cosine. As the cosine has the same value for ϕ and $\pi - \phi$, the PAC estimation is biased and underestimated when ϕ_0 is not equal to 0 or π. Indeed, the model mixes the contribution of ϕ and $\pi - \phi$, which attenuates the modulation effect. This bias was already reported in PAC estimation by (Bruns and Eckhorn, 2004), and the solution proposed by (Penny et al., 2008) was to use both cosine and sine to have PAC estimates that are invariant to the preferred phase ϕ_0.

The same technique can be used on DAR models. To do so, we not only filter the raw signal with $w_1(t) = b(t)\cos(2\pi f_x t)$ to obtain x_1, but also with $w_2(t) = b(t)\sin(2\pi f_x t)$ to obtain x_2, creating a complex-valued driver $x = x_1 + jx_2$ where j denotes a complex number, $j^2 = -1$. With a complex-valued driver, DAR models are naturally extended by adding more regressors:

$$\forall i \in [1, p] \qquad a_i(t) = \sum_{0 \leq k+l \leq m} a_{ikl}\, x_1(t)^k x_2(t)^l \ . \tag{3.2}$$

Note that we not only add the powers of x_2, but also the cross-terms $x_1 x_2$, $x_1^2 x_2$, etc. Indeed, cross-terms turned out to be critical to properly remove the bias when $m \geq 2$. The only case that does not need cross-terms to correctly remove the bias is the case $m = 1$.

Instead of $p(m+1)$ regressors, we now have $p(m+1)(m+2)/2$ regressors, and the number of degrees of freedom of the model is now $d = (p+1)(m+1)(m+2)/2$. Typical values for m are below 3, so the number of parameters stays within a reasonable range despite the squared dependence in m. Note that it remains much lower than the number of time samples and the estimation problem stays therefore well-posed. The innovation variance model is also updated accordingly:

$$\log(\sigma(t)) = \sum_{0 \leq k+l \leq m} b_{kl}\, x_1(t)^k x_2(t)^l \ . \tag{3.3}$$

To validate this parametrization, we simulated some signals containing PAC, as described in Subsection 3.1.5, introducing a phase difference ϕ_0 in the modulation. For each value of ϕ_0, we fitted a DAR model with a real driver, and a DAR model with a complex driver, and compared their negative log-likelihood by time sample (the lower the better). We also fitted an AR model to serve as a baseline. The parameters were set to $(p, m) = (10, 1)$.

The results are presented in Figure 3.3. A bias is visible around $\phi_0 = \pm\pi/2$, since the real-valued driver DAR model does not fit better than the AR model. As expected, this bias disappears when we update the model to a complex-valued driver DAR model. We also verified that this result holds with $m > 1$.

3.1.3 Conditional power spectral density

After preprocessing and model estimation, we can compute the conditional power spectral density (PSD) of the DAR model, as described in Subsection 2.1.4.

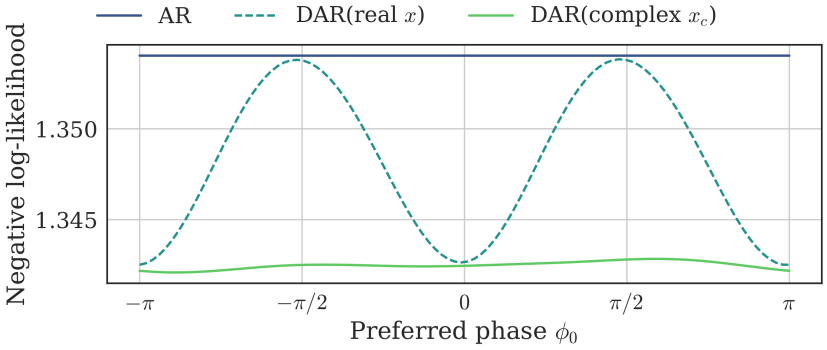


Figure 3.3 – Removing the preferred phase bias with a complex-valued driver. Using DAR models with a real-value driver is biased if the preferred phase is not 0 or π. Using a complex-valued driver fixes this bias.

In Figure 3.2 (b and c), one can see two simulated signals: one with PAC on the left (b) and one without PAC on the right (c). The simulation process, which does not make use of DAR model, will be detailed and discussed in Subsection 3.1.5.

In Figure 3.2 (d), one can see an example of PSD(x), with artificial driver's values $x_n = \rho \exp(j\phi_n)$. In practice we set ρ to the median of the empirical driver's amplitude, $\rho = \text{median}(|x|)$, and ϕ_n spans the entire interval $[-\pi, \pi]$. Note that the PSD is a function of the driver value and not only its phase, unless the driver has been normalized in amplitude. Computing the PSD on a circle allows to visualize the PSD fluctuations with respect to the driver's phase, giving a representation similar to other PAC metrics. The PAC is identified by the modulation of the spectrum with respect to the driver's phase ϕ. Other trajectories could be chosen to visualize also the dependency on the driver's amplitude. Note that with DAR models, the amplitude modulation of y is modeled jointly on the entire spectrum, as opposed to frequency band by frequency band in most other PAC metrics.

3.1.4 Comodulogram

We now detail how comodulograms can be derived from DAR models. To quantify the coupling at a given frequency f, we first compute PSD(x)(f) on a range of N artificial driver's values x_n located on a circle of radius $\rho = \text{median}(|x(t)|)$, and we normalize it to sum to 1:

$$\forall n \in [1, N] \qquad p_f(n) = \frac{\text{PSD}(\rho e^{j2\pi n/N})(f)}{\sum_{n=1}^{N} \text{PSD}(\rho e^{j2\pi n/N})(f)} \quad . \tag{3.4}$$

Then, we use the same method as (Tort et al., 2010) to measure the fluctuation of $p_f(n)$: we compute the Kullback-Leibler divergence between $p_f(n)$ and the uniform distribution $q(n) = 1/N$, and we normalize it with the maximum entropy $\log(N)$:

$$M(f) = \frac{1}{\log(N)} \sum_{n=1}^{N} p_f(n) \log\left(\frac{p_f(n)}{q(n)}\right) \quad . \tag{3.5}$$

This metric M is between 0 and 1 and allows us to quantify the non-flatness of $p_f(n)$, *i.e.* the amplitude modulation, frequency by frequency. We can then build a *comodulogram*, a figure commonly used in the literature which depicts the PAC metric

for a grid of frequencies (f_x, f). When derived from DAR models as described here, the comodulogram is not affected by the systematic biases presented in the introduction. Indeed, we do not need to filter the high frequencies, and DAR models estimate the amplitude modulations for all frequencies at the same time.

Figure 3.2 (e) shows the comodulogram of the two simulated signals, one with and one without PAC. As expected the DAR model reports no coupling when there is none, and reports a strong coupling here between 3 and 50 Hz when it is actually simulated.

3.1.5 Datasets and simulated PAC signals

Datasets We tested DAR models on three different datasets, with invasive recordings in humans and rodents:

1. A rodent local field potential (LFP) recording in the dorso-medial striatum from Dallérac et al. (2017, Supplementary figure 2), (1,800 seconds down-sampled at 333 Hz).
2. A rodent LFP recording in the hippocampus from Khodagholy et al. (2015) collected during rapid eye movement (REM) sleep (100 seconds down-sampled at 625 Hz).
3. A human electro-corticogram (ECoG) channel in auditory cortex from Canolty et al. (2006) (730 seconds down-sampled at 333 Hz).

We refer the reader to corresponding articles for more details on the recording modalities of these neurophysiological signals.

Simulations In our experiments, methods comparisons and model evaluations, we make an extensive use of simulated signals showing some PAC. The simulations are generated as follows: we simulate a coupling between $f_x = 3$ Hz and $f_y = 50$ Hz, with a sampling frequency $f_s = 240$ Hz, during T time points (T depends on the experiment). We do not use a perfect sinusoidal driver, as using such an ideal oscillatory signal is over-simplistic. Drivers can indeed have a larger band as suggested by our experiments. Finally, strong empirical evidence suggests that neurophysiological signals have complex morphologies that can be overlooked when studied as ideal sinusoids (Cole and Voytek, 2017).

To construct a time-varying driver peaking at 3 Hz, we bandpass filter a Gaussian white noise at a center frequency $f_x = 3$ Hz and with a bandwidth $\Delta f_x = 1$ Hz, using the same filter detailed in the preprocessing Subsection 3.1.1. The smaller the bandwidth, the closer the driver is to a perfect sinusoid. We normalize the driver to have unit standard deviation $\sigma_x = 1$. We then modulate the amplitude a_y of a sinusoid at, for example, $f_y = 50$ Hz, using a sigmoid on the driver x:

$$a_y(t) = \frac{1}{1 + \exp(-\lambda x(t))} \quad , \tag{3.6}$$

with a sharpness set to $\lambda = 3$. By doing so, the amplitude varies between 0 and 1 depending on the driver's value. We normalize the signal y to have a standard deviation $\sigma_y = 0.4$. We then add to the signal y both the driver x and a Gaussian white noise with a standard deviation $\sigma_\varepsilon = 1$. Note that this simulation procedure is not based on a DAR model. In other words, we do not validate our model using signals that fall perfectly into the category of stochastic signals that are synthesized with a DAR model.

3.2 Model selection

3.2.1 Generative models

DAR models are generative, which offers a significant advantage over traditional PAC metrics. Indeed, we can use the likelihood of the model L (2.8), which quantifies the *goodness of fit* of the data given the model, to compare different choice of hyper-parameters. To select automatically hyper-parameters, we typically perform a grid-search and select the hyper-parameters which yield the highest likelihood on left-out data. Importantly, the highest likelihood does not necessarily correspond to the highest PAC score, but rather to the model with the highest goodness of fit or explained variance. This detail is crucial, as optimizing for the best fit is a legitimate data-driven approach, whereas optimizing for the highest PAC score is statistically more questionable.

Importantly, to avoid overfitting the signal used during model estimation, the likelihood needs to be estimated on *another* signal. This out-of-sample evaluation is very standard in the machine learning community (Bishop, 2006), as it reduces the risk of overfitting the signal used during model estimation. To have a more robust model selection, we can split the dataset multiple times into a training and a testing signal, and aggregate the likelihood scores of each split. This procedure is called cross-validation (CV).

Another benefit of a model selection based on CV is that it naturally penalizes the complexity of the model. Indeed, we may want to compare models with different number of parameters (*i.e.* degrees of freedom), for instance with different order p in DAR models. In that case, adding more degrees of freedom usually leads to a better fit on the training signal. However, it also increases the variance of the estimation, which may lead to a poorer fit on out-of-sample signals if the model is too complex. Thus, as CV selects the model which generalizes the best on unseen data, it naturally avoids too complex models. Note that CV requires a large amount of data to be left-out for model testing, and that the two datasets be separated by a minimum delay to ensure data independence (Arlot et al., 2010). We give an example of model and order selection with CV in Subsection 3.2.4.

Another approach to compare models with different degrees of freedom is to use a so-called information criteria, such as the Akaike Information Criterion (AIC) (Akaike, 1998) or the Bayesian Information Criterion (BIC) (Schwarz et al., 1978). These criteria modify the log-likelihood with an additive term which penalizes the complexity of the model. AIC and BIC read respectively $\text{AIC} = -2\log(L) + 2d$ and $\text{BIC} = -2\log(L) + d\log(T)$, where d is the number of degrees of freedom of the model. In DAR models with a complex driver as used for PAC analysis, we have $d = (p+1)(m+1)(m+2)/2$. Notably, the BIC has been used extensively for order selection in AR models. We describe an experiment using model selection based on the BIC in Subsection 2.2.6.

Another selection approach uses a likelihood ratio test to compare *nested* models. This test compares a model H_1 and a restriction of this model H_0 where we set k coefficients to a fixed value (*e.g.* 0) before the estimation. Wilk's theorem (Wilks, 1938) states that the likelihood ratio $-2\log(L_0/L_1)$ asymptotically follows a $\chi_2(k)$ distribution under the null hypothesis H_0. It is then possible to compute a p-value and decide whether to reject or not the null hypothesis H_0.

In practice, we chose to use model selection with CV in almost all our experiments. In the next three subsections, we present in details the benefit of hyper-parameter selection, describing three possible applications.

3.2.2 Driver filter selection

Our first example of hyper-parameter selection concerns the driver's filter. During the preprocessing, the driver x is extracted from the preprocessed signal z through a band-pass filter, with center frequency f_x and bandwidth Δf_x. To select automatically these two parameters, we perform a grid-search over hyper-parameters and select the one leading to the best likelihood on out-of-sample signals.

For a fair comparison of the drivers, we need to evaluate the model fitting on the exact same signal y. To do so, we remove from y all the possible drivers tested on this grid-search using a high-pass filter above maximum center frequency f_x. As during preprocessing, we fill the frequency gap by adding a Gaussian white noise filtered with a low-pass filter complementary to the high-pass filter.

Simulation study To validate our approach, we tested on simulated signals whether both center frequency f_x and bandwidth Δf_x were correctly estimated. For this, we simulated signals using $f_x(simu) = 4\,\text{Hz}$ and $\Delta f_x(simu) \in [0.2, 0.4, 0.8, 1.6, 3.2]\,\text{Hz}$. To mimic the duration of our three real signals, we used $T = \lfloor 100 f_s \rfloor$ time points, for a duration of 100 seconds.

As the noise mainly affected the amplitude modulation but barely altered the driver, we also added a Gaussian white noise low-pass filtered at 20 Hz, and scaled to have a PSD difference of 10 dB at $f_x = 4\,\text{Hz}$ between the driver and the noise. By ding so, the driver was also altered by noise.

Given the simulated signal, we performed a grid-search over f_x and Δf_x, extracting the driver, fitting the model and computing the likelihood of the model. Results are presented in Figure 3.4. As expected, the center frequency f_x was correctly estimated for all bandwidths Δf_x. More importantly, the negative log-likelihood was minimal at the correct simulated bandwidth.

This simulation study confirms that the likelihood can be used to estimate the correct parameters for the driver's filtering step, and that it does not present any obvious bias in the estimation.

Driver estimates on human and rodent recordings The outcome of the model selection procedure on the three neurophysiological signals are reported in Figure 3.5. Two general observations can be made. First, for all three signals, the optimal bandwidth was relatively large (3.2 Hz). Second, the optimal center frequency changed as we increased the bandwidth. Interestingly, this phenomenon was not observed in the simulation study, suggesting that in real data the optimal driver is wide-band and has an asymmetrical spectrum. In other words, the driver's frequency is not precisely defined, and a large band-pass filter should be preferred to extract the driver.

These observations thus question the practical choice of parameters. The classical approach is to build a comodulogram to select the best driver frequency f_x, choosing arbitrarily the bandwidth Δf_x. This bandwidth is typically quite narrow, *e.g.* 0.4 Hz, and chosen to clearly isolate the maximum frequency in the comodulogram. However, this approach relies on the assumption that the driver is nearly sinusoidal, which is unrealistic (Cole and Voytek, 2017). On the contrary, our data-driven approach selects the frequency and bandwidth that lead to the highest goodness of fit of our model.

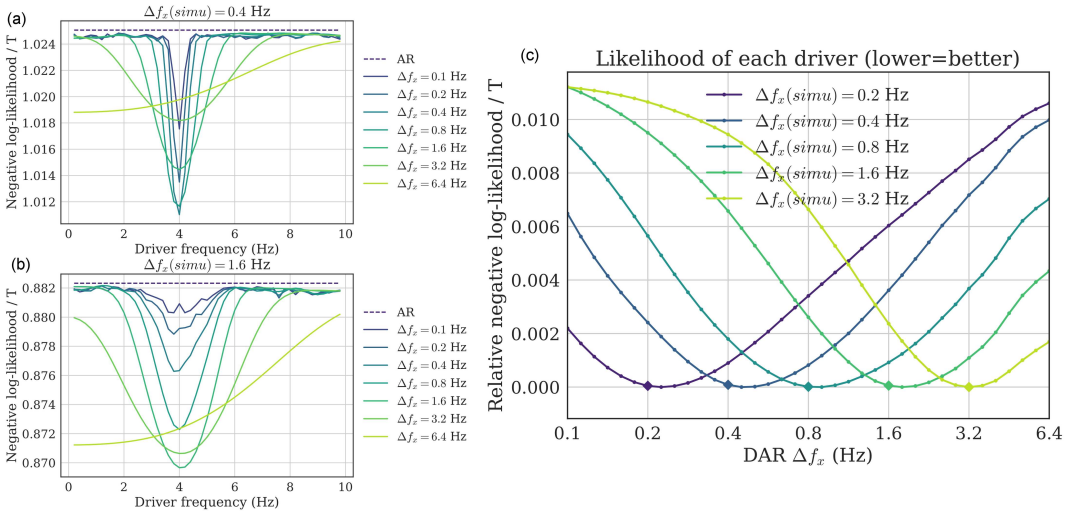


Figure 3.4 – Driver's frequency and bandwidth selection (simulations). (a,b) Two examples of grid-search over both f_x and Δf_x, over simulated signals with (a) $\Delta f_x(simu) = 0.4\,\text{Hz}$ and (b) $\Delta f_x(simu) = 1.6\,\text{Hz}$. For all bandwidths (except 6.4 Hz), the negative log-likelihood is minimal at the correct frequency $f_x = 4\,\text{Hz}$. (c) To see more precisely the bandwidth estimation, we plot the negative log-likelihood (relative to each minimum for readability), for several ground-truth bandwidth $\Delta f_x(simu)$, with $f_x = 4\,\text{Hz}$. For each line, the minimum correctly estimates the ground-truth bandwidth (depicted as a diamond), showing empirically that the parameter selection method gives satisfying results.

To further describe the implications of our observations, we focused on the rodent striatal LFP signal. We compared an arbitrary choice of driver bandwidth (0.4 Hz) and the optimal choice with respect to the model likelihood (3.2 Hz). For these two bandwidths, we selected the optimal center frequencies, 2.2 Hz and 4.0 Hz respectively. Figure 3.5(b) shows a time sample of the raw signal and the two extracted drivers. The wide-band driver (in green) followed very well the raw signal, which was not a perfect sinusoid. On the contrary, the narrow-band driver (in blue) seemed poorly related to the slow oscillation of the raw-signal. To determine which of these two slow varying signals best captured the temporal amplitude modulations of the high frequencies, we fitted a model using each of these two drivers. We used the *same* high-pass filtered signal y, *i.e.* where we removed all frequencies below 16 Hz.

We found that the model with the wide-band driver explained more variance in the high frequencies than the model with the narrow-band driver, that is, that the model better explained the amplitude fluctuations of the high frequencies. As we used the same signal y for both models and as the driver lied in different frequency intervals, it should be noted that the correspondence to the slow oscillation visible in Figure 3.5(b) and the quality of the fit of the amplitude modulation in the high frequencies were two independent observations. By optimizing for the latter, one observes in Figure 3.5(b) that it leads to a more realistic extraction of the driving neural oscillation.

More generally, our results emphasize the complexity of the choice of parameters in PAC analysis. With our method, we were able to easily compare different parameters, even on non-simulated data, which offered a principled way to set them and helped avoiding misinterpretation due to bad choices. We now investigate the conditional PSDs

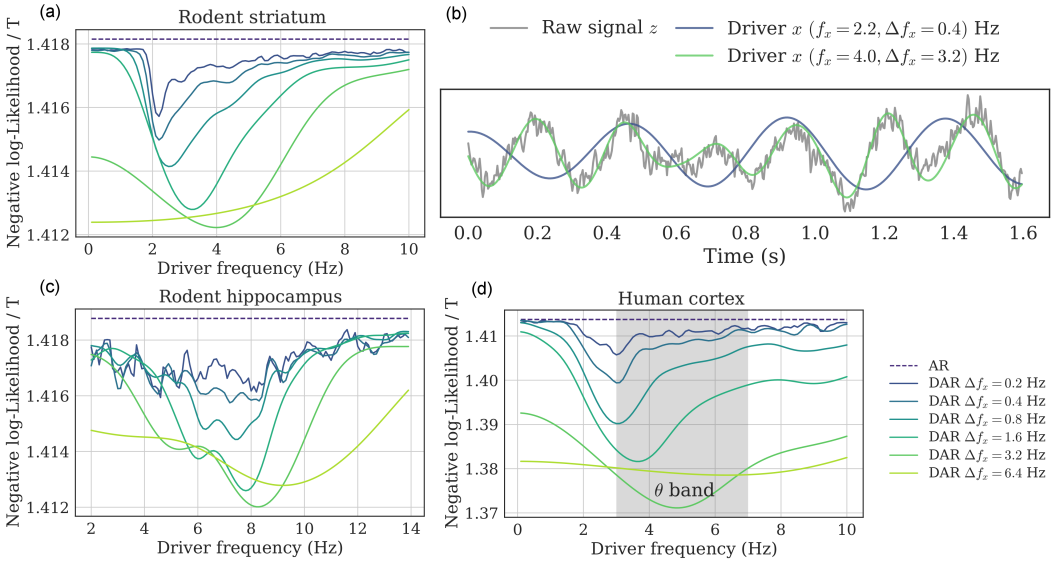


Figure 3.5 – Driver's frequency and bandwidth selection (real signals). (a,c,d) Negative log-likelihood of the fitted model for a grid of filtering parameters f_x and Δf_x: (a) rodent striatum, (c) rodent hippocampus, (d) human auditory cortex. The optimal bandwidth was very large (3.2 Hz), and the optimal center frequency changed as the bandwidth increased, suggesting that the optimal driver had a wide asymmetrical spectrum. (b) This portion of the rodent striatal signal shows two examples of driver with different bandwidths: The wide-band driver better follows the raw signal, and *independently* also leads to a better fit in DAR models.

and the comodulograms estimates on these three datasets.

PSDs and comodulograms estimates of human and rodent recordings Figure 3.6 shows for each signal the PSD (in dB) depending on the driver's phase, estimated through DAR models. The hyper-parameters (p, m) for the rodent striatal, the rodent hippocampal and the human cortical data were $(90, 2)$, $(15, 2)$, and $(24, 2)$, respectively. These parameters were chosen by cross-validation with an exhaustive grid search: $p \in [1, 100]$ and $m \in [0, 3]$. The filtering parameters $(f_x,\ \Delta f_x)$ were chosen to maximize the likelihood as described in the previous section, and are respectively $(8.2,\ 3.2)$ Hz, $(5.2,\ 3.2)$ Hz and $(4.0,\ 3.2)$ Hz (Figure 3.6 bottom).

For comparison, we also show the PSD obtained with a bandwidth $\Delta f_x = 0.4$ Hz, with the center frequency chosen to maximize the likelihood. We used $(6.4,\ 0.4)$ Hz, $(3.2,\ 0.4)$ Hz and $(2.2,\ 0.4)$ Hz (Figure 3.6 top), respectively.

We observed two kinds of phenomena. In the rodent striatal and hippocampal data, the coupling was mainly concentrated around a given high frequency (*i.e.* 80 Hz and 125 Hz, respectively), whereas in the human cortical data, the coupling was observed at all frequencies, with a maximum around 20 Hz. The smoothness of the figures depends on the parameter p: a low value leads to a smoother PSD. Please note that the interpretation of the results depends on the filtering parameters.

The two phenomena can also be visualized in Figure 3.7, where we plotted comodulograms for all three signals, computed with four different PAC metrics: the mean vector length first proposed in Canolty et al. (2006) and updated by Özkurt and Schnitzler (2011),

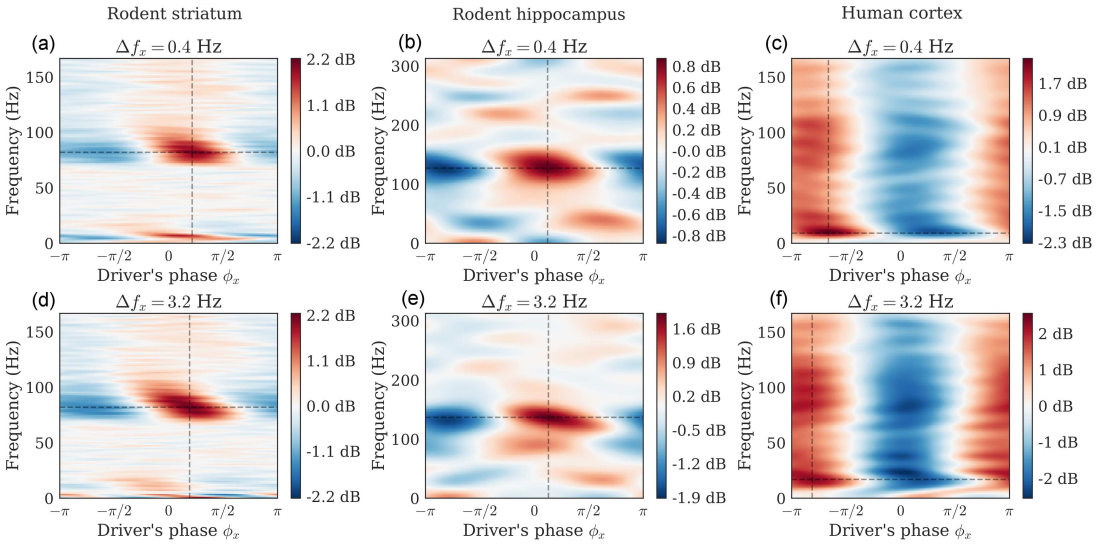


Figure 3.6 – PSD conditional to the driver's phase. Dataset: Rodent striatum (a,d), rodent hippocampus (b,e), human auditory cortex (c,f). We derive the conditional PSD from the fitted DAR models. In one plot, each line shows at a given frequency the amplitude modulation with respect to the driver's phase. The driver bandwidth Δf_x is 0.4 Hz (top row) and 3.2 Hz (bottom row). Note that the maximum amplitude is not always at a phase of 0 or π (*i.e.* respectively the peaks or the troughs of the slow oscillation). In figure (d), we can also observe that the peak frequency is slightly modulated by the phase of the driver (phase-frequency coupling).

the parametric model of Penny et al. (2008) based on GLM, the modulation index of Tort et al. (2010) that uses the Kullback-Leibler divergence, and our method based on DAR models.

In DAR models, the hyper-parameters used are the cross-validated values described above. With other methods, high frequencies were extracted with a bandwidth Δf_y twice the highest driver frequency used: 14 Hz, 28 Hz, and 20 Hz respectively. For all methods, the drivers were extracted with the bandwidth $\Delta f_x = 3.2$ Hz, optimal for DAR models. The white lines crop the regions with a p-value $p < 0.01$, as describe in Subsection 3.3.1.

The resulting comodulograms did not look like typically reported comodulograms, as we used a much larger bandwidth (3.2 Hz) than what was found in the literature. With an arbitrary choice of parameters, we could have obtained more classical comodulograms, as shown in Figure 3.8. However, these results could be misleading because they suggest a coupling between sinusoidal oscillators, although the corresponding drivers are not perfectly sinusoidal (see Figure 3.5(b)).

On these comodulograms, the four methods yielded comparable results as the data we used were very long: 1800, 100, and 730 seconds for the rodent striatal, the rodent hippocampal and the human cortical data, respectively. However, as we will see in Subsection 3.3.2, differences emerge between methods when the signals are shorter.

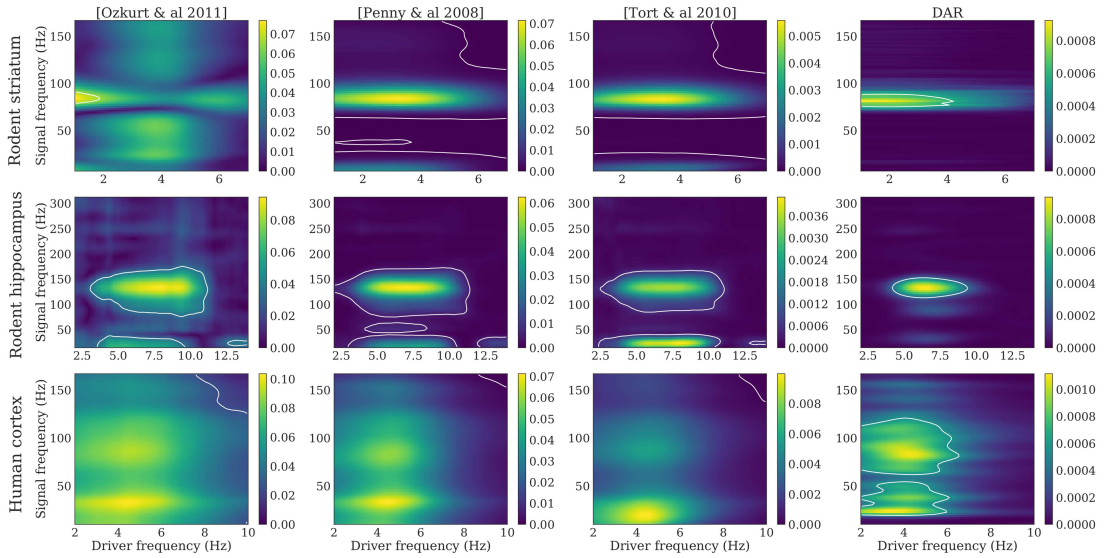


Figure 3.7 – Comodulograms with parameters maximizing the likelihood. Dataset: rodent striatum (top), rodent hippocampus (middle) and human auditory cortex (bottom). Methods, from left to right: Özkurt and Schnitzler (2011), Penny et al. (2008), Tort et al. (2010), DAR models (ours). The DAR model parameters and the driver bandwidth are chosen to be optimal with respect to the likelihood. White lines outline the regions with a p-value $p < 0.01$, as described in Subsection 3.3.1.

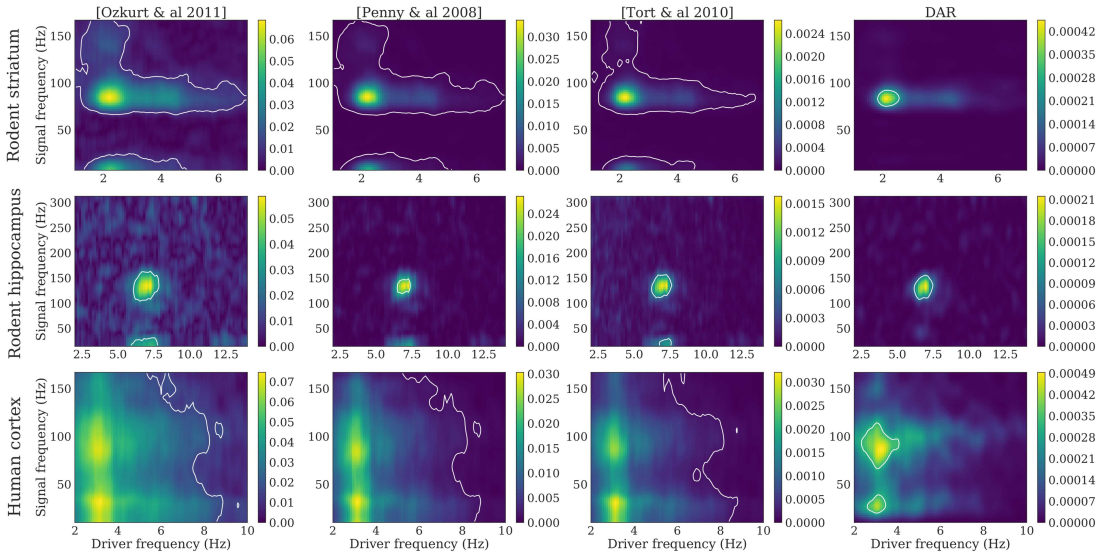


Figure 3.8 – Comodulograms with parameters maximizing the comodulgoram sharpness. Dataset: rodent striatum (top), rodent hippocampus (middle) and human auditory cortex (bottom). Methods, from left to right: Özkurt and Schnitzler (2011), Penny et al. (2008), Tort et al. (2010), DAR models (ours). The driver bandwidth is chosen to have a well defined maximum in driver frequency: $\Delta f_x = 0.4\,\text{Hz}$. The DAR model parameters are chosen to give similar results than the other methods: $(p, m) = (10, 2)$. White lines outline the regions with a p-value $p < 0.01$, as described in Subsection 3.3.1

3.2.3 Delay selection and directionality

Our second example of hyper-parameter selection concerns a delay parameter. So far, DAR models use the driver at time t to parametrize the AR coefficients with the assumption that, in PAC, high frequency activity is modulated by the driving signal without any delay. However, one could legitimately ask whether slow fluctuations in oscillatory activity precede the amplitude modulation of the fast activity or, conversely, whether high frequency activity precedes low frequency fluctuations. This is what we will refer to as *directionality estimation*.

To assess the problem of directionality estimation with DAR models, we capitalize on the likelihood function. After observing some PAC between y and x, we introduce a delay τ in the driver so as to estimate the goodness of fit of a DAR model between y and x_τ, where $x_\tau(t) = x(t-\tau)$. Using a grid of values for the delay, one can report the value τ_0 that corresponds to the maximum likelihood. In practice, as we use non-causal filters to extract the driver, and because the AR model uses samples from the past of the signal y, one might wonder if there is any positive or negative bias in our delay estimation. To address this issue, we apply our analysis on both forward and time-reversed signals, and sum the two models log-likelihood. By doing so, the filtering bias is strongly attenuated because it similarly affects both forward and time-reversed models. If the best delay is positive, it means that the past driver yields a better fit than the present driver, *i.e.* that the slow oscillation precedes the amplitude modulation of the fast oscillation. Inversely, a negative delay means that the amplitude modulation of the fast oscillation follows changes in the slow oscillation. It is noteworthy that in DAR models, we arbitrarily call *driver* the slow oscillation although the model makes no assumption on the directionality of the coupling.

Preferred phase and temporal delay It is worth noting that the estimation of the delay τ_0 shall not be considered as an alternative way to estimate the preferred phase ϕ_0 defined in Subsection 3.1.2. Although the preferred phase and the delay would be identical if the driver were a perfect stationary sinusoid, in non-stationary neural systems in which the driver's instantaneous frequency may fluctuate with time, the preferred phase and delay will likely differ. Such a scenario can occur either because signal waveforms are not perfect sinusoids (Cole et al., 2016) or because the driver actually changes (Tort et al., 2008). In other words, τ_0 is a time delay which is identical at all time and corresponds to different phase shifts $\phi(t) = \tau_0(2\pi f(t))$ which depend on the instantaneous frequency $f(t)$. On the contrary, ϕ_0 is a phase shift that is constant over time and which corresponds to different time delays $\tau(t) = \phi_0/(2\pi f(t))$ which also depend on the instantaneous frequency. Figure 3.9 illustrates these specific points and disentangles the two distinct notions.

Like most PAC metrics, DAR models are invariant with respect to the preferred phase. Both the strength of the coupling and the model likelihood are unchanged with respect to ϕ_0 (*c.f.* Subsection 3.1.2). On the contrary, all PAC metrics including DAR models are strongly affected by a time delay τ when the driver is not a perfect sinusoid. This delay attenuates the coupling and may artificially modify the preferred phase. When the delay is too large, all metrics would measure zero coupling. This is in fact what justifies the use of surrogate techniques that introduce a large time shift to quantify the variance of the measure in the absence of coupling (Canolty et al., 2006, Tort et al., 2010, Aru et al., 2015). Figure 3.9(e-h) provides the delay estimated with DAR models, obtained by maximizing the likelihood over a grid of delays. In Figure 3.9(i-l), we show

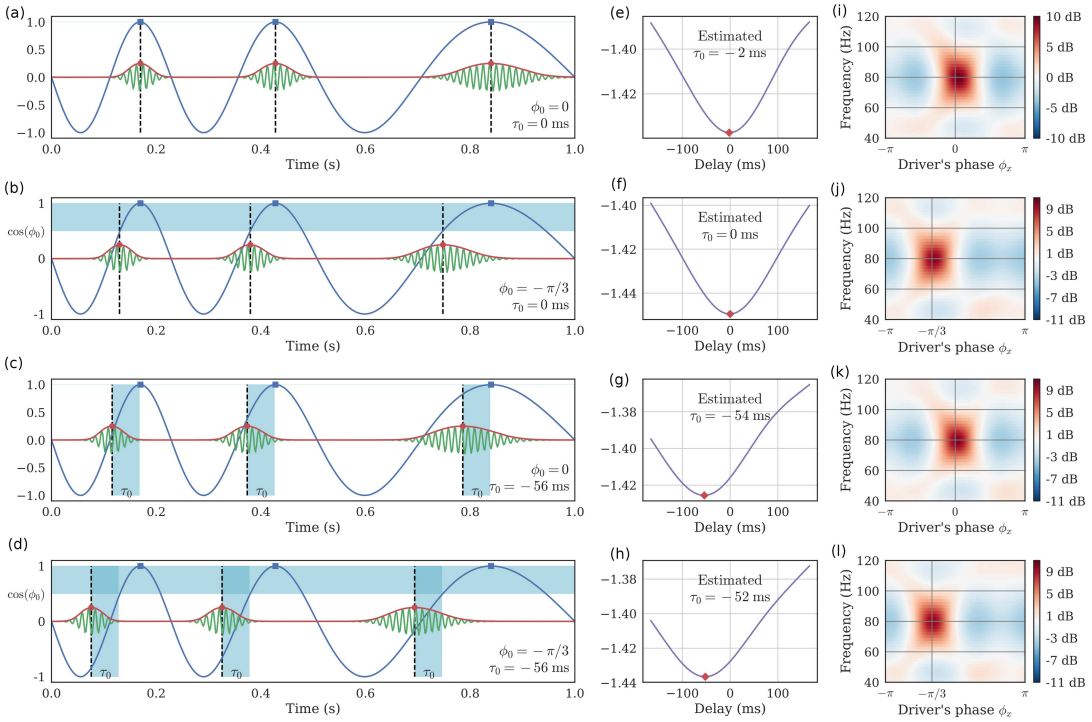


Figure 3.9 – The temporal delay τ is distinct from the preferred phase ϕ_0. (a) When both are equal to zero, the high frequency bursts happen in the driver's peaks. (b) When $\tau = 0$ and $\phi \neq 0$, the bursts are shifted in time with respect to the driver's peaks, and this shift *varies* depending on the instantaneous frequency of the driver. (c) When $\tau \neq 0$ and $\phi = 0$, the bursts are shifted in time with respect to the driver's peaks, and this shift is *constant* over the signal. In this case, note how the driver's phase corresponding to the bursts varies depending on the instantaneous frequency of the driver. (d) τ and ϕ_0 can also be both non-zero. (e-h) Negative log-likelihood of DAR models, fitted with different delays between the driver and the high frequencies. The method correctly estimates the delay even when $\phi_0 \neq 0$. (i-l) PSD conditional to the driver's phase, estimated through a DAR model with the best estimated delay. The maximum amplitude occurs at the phase ϕ_0.

the conditional PSD of the model obtained with the best estimated delay as well as the estimated preferred phase ϕ_0. Using the likelihood, our method is able to estimate this delay, thus improving PAC estimation and extending PAC analysis.

Other directionality approaches The question of delay estimation has been previously addressed in the literature with measures borrowed from information theory. For instance, transfer entropy (TE) (Schreiber, 2000, Wibral et al., 2013, Park et al., 2015) has been adapted to PAC by applying it on the driver and the envelop of the fast oscillations (Besserve et al., 2010). The TE from a signal a to a signal b is defined as the mutual information between the present of b and the past of a, conditioned on the past of b. It is similar in this way to Granger causality (GC) (Granger, 1969), which also compares how the present of b can be better predicted by the past of both a and b, than by only the past of b, using AR models. Note that when variables follow a Gaussian distribution, TE and GC are actually equivalent (Barnett et al., 2009). In the case of PAC, both TE and GC need to be evaluated on the driver x and the *envelope* of a

bandpass filtered y_f. Interestingly, GC has been shown to fail when the signal-to-noise ratios of the two signals are different (Nolte et al., 2010), which is often the case for the driver and the envelop of the high frequencies. Hence, it remains to be investigated how TE/GC compares with DAR models in terms of performances.

Our method is also different from these other methods, since we model directly y and not its envelop, making our method more specific to PAC. This is made possible since DAR models use the delayed driver x_τ not to predict y, but to modulate how y is predicted from its own past. PAC directly arises from this non-linear interaction in DAR models. Inherently to this PAC specificity, our method is also asymmetrical with respect to x and y.

Directionality in PAC has also been addressed with another method called Cross-Frequency Directionality (CFD) (Jiang et al., 2015), which is based on the phase slope index (PSI) (Nolte et al., 2008). PSI is a measure of the phase slope in the cross-spectrum of two signals, and is also used to infer causal relations between signals. CFD adapts this measure by applying it on the driver and the envelop of the fast oscillations. Contrary to TE, CFD is not designed to measure a delay; thus, we modified it so as to compare it *qualitatively* to our approach (see below).

If such delay estimation results may not reflect pure causality (Aru et al., 2015), for instance because of the transitivity of correlation, they nevertheless improve the analysis of PAC going one step further by estimating the delay between the coupled components.

Simulation study To validate the directionality estimations approach, we simulated signals with $T = 1024$ time points (4 seconds), as described in Subsection 3.1.5. We introduced a delay between the slow oscillation and the amplitude modulation of the fast oscillation, and verified that our method could correctly estimate the delays. Importantly, we did not use a perfect sinusoidal driver, as the delay would only end up in a phase shift, and would not change the strength of the PAC. We used a bandwidth $\Delta f_x = 2.0\,\text{Hz}$ and a noise level $\sigma_\varepsilon = 0.4$.

We compared our method, using a DAR model with $(p, m) = (10, 1)$, to the cross-frequency directionality (CFD) approach described in (Jiang et al., 2015). This method makes use of the phase slope index (PSI) (Nolte et al., 2008) for PAC estimation. In (Jiang et al., 2015), the PSI over a frequency band F is defined as:

$$\text{PSI}(F) = \Im\Big(\sum_{f \in F} C_{xy}(f)^* C_{xy}(f + \delta_f)\Big) \ , \tag{3.7}$$

where C_{xy} is the complex coherence, $\Im$ is the imaginary part, and δ_f is the frequency resolution. As the PSI was not designed to provide an estimation of the delay, we *modified* it into:

$$\tau_{\text{PSI}}(F) = \frac{\text{PSI}(F)}{2\pi\delta_f n_F} \ , \tag{3.8}$$

where n_F is the number of frequencies in set F. Note that this delay estimator is correct only if the coherence is almost perfect $C_{xy}(f) \approx 1$ and if the phase slope is small enough to have $\sin(\phi(f + \delta_f) - \phi(f)) \approx \phi(f + \delta_f) - \phi(f)$. As these assumptions are rarely met in practice, we only used this estimator to compare *qualitatively* with our own estimator.

Results presented in Figure 3.10 show the mean estimated delay with ± 1 standard deviation computed over 20 simulations. The delay was correctly estimated in both time directions, without any visible bias. As expected, the modified CFD was biased, and showed a higher variance than the DAR-based approach.

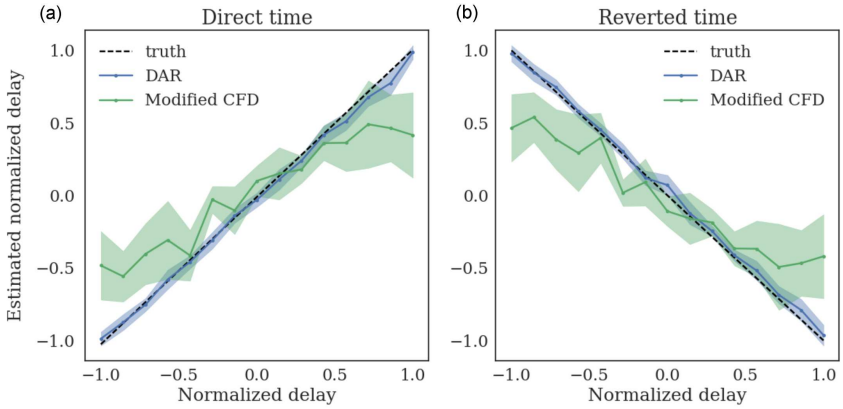


Figure 3.10 – Estimating delays over 20 simulations for each delay. The delays are estimated on both direct time (a) and reverted time (b). The delays are normalized: $\tau = 1$ corresponds to one driver oscillation, *i.e.* $1/f_x$ sec. The modified CFD shows a bias and only serves for qualitative comparison. The delay estimation based on DAR models correctly estimates the delays, without any visible bias.

Delay estimates in human and rodent recordings We applied this delay estimation method on the three experimental recordings. Once again, note that the comparison with the CFD (Jiang et al., 2015) is only qualitative, since the CFD was not designed as a delay estimator. The temporal delay was estimated with DAR models with the parameters obtained from the cross-validation selection. The driver we used was obtained with the best filtering parameters found in our grid-search: $(f_x,\ \Delta f_x)$ are respectively $(4.0,\ 3.2)$ Hz, $(8.2,\ 3.2)$ Hz, and $(5.2,\ 3.2)$ Hz.

The results are presented in Figure 4.13, where we display the delay estimated with maximum likelihood over multiple DAR models. We also display error bars which correspond to the standard deviation obtained with a block-bootstrap strategy (Carlstein, 1986). This strategy consists in splitting the signal into $n = 100$ non-overlapping blocks of equal length, to draw at random with replacement n blocks (using the same n), and to evaluate the delay on this new signal. We repeated this process 20 times and computed the standard deviation of the 20 estimated delays. Such strategy is used to estimate empirically the variance of a general statistic from stationary time series.

First, one can observe that the results were qualitatively similar between the two methods. The CFD was biased toward zero for large delays, as was previously shown in the simulations. One can also note that the directionality was not always the same across datasets. For the rodent striatal and the human cortical data, the delay was negative and the best model fit happened between the signal y and the *future* driver x. Inversely, in the rodent hippocampal data, the delay was positive, and the best model fit was between the signal y and the *past* driver x. Further experiments need to be performed to better understand the origin of such delays but we demonstrate here the usefulness of DAR models to estimate them.

Note that our method does not select the delay that leads to the maximum PAC, which would not be statistically valid. On the contrary, we select the delay that leads to the best fit of the model on the data. This approach is more rigorous since we maximize the variance explained by our model, and not the effect of interest.

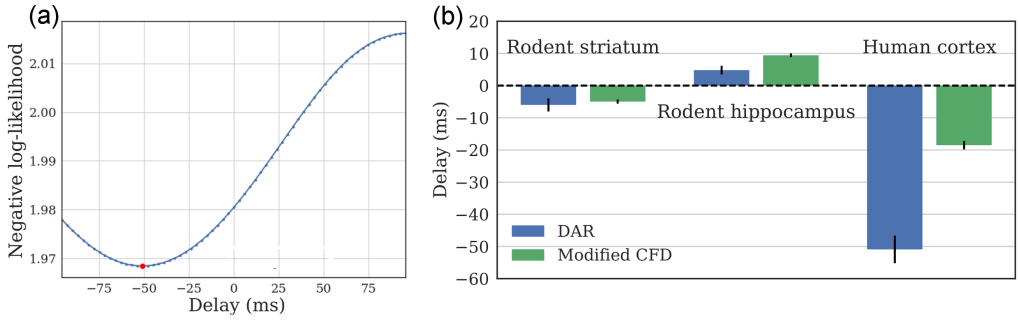


Figure 3.11 – Estimated delays on rodent and human datasets. (a) Negative log likelihood for multiple delays, with the human cortical signal. (b) Optimal delay for the three signals, computed with model selection on DAR models, and with a CFD method modified to provide a delay. The error bars indicate the standard deviation obtained with a block-bootstrap strategy. For the rodent hippocampal data, the delay is positive: the low frequency oscillation precedes the high frequency oscillation. For the human cortical data and the rodent striatal data, the delay is negative: the high frequency oscillation precedes the low frequency oscillation.

3.2.4 Amplitude of the driver

Our last example of model selection concerns the driver's amplitude. One can note that in DAR models, the driver contains not only the phase of the slow oscillation, but also its amplitude. As the driver is not a perfect sinusoid, its amplitude fluctuates with time. On the contrary, most PAC metrics discard the amplitude fluctuations of the slow oscillation and only consider its phase.

To evaluate these two options, we compared two drivers using DAR models: the original (complex) driver $x(t)$, and the normalized driver $\tilde{x}(t) = x(t)/|x(t)|$. This normalized driver only contains the phase information, as in most traditional PAC metrics. Using cross-validation, we compared the log-likelihood of the two fitted models, and found a difference always in favor of the non-normalized driver $x(t)$, as it can be visualized in Figure 3.12.

More precisely, for all three dataset, (a) rodent striatum, (b) rodent hippocampus, and (c) human auditory cortex, we split the signal in half, we fitted the models on the first half, and evaluated the model likelihood on the second half. We compared four different models on a grid of parameter $p \in [0, 100 - 200]$ and $m \in [0, 3]$:

1. AR: a linear AR model
2. Heteroskedastic AR (HAR): an hybrid model between a linear AR model and a DAR model, where the innovation variance σ^2 is driven by x, but the AR coefficients are constant in time.
3. Phase DAR (PDAR): a DAR model, with a normalized driver: $x/|x|$. By doing so, we only consider the phase of the slow oscillation, as in most PAC metrics.
4. DAR: a DAR model, where both the innovation variance σ^2 and the AR coefficients are driven by x.

The figures present the negative log likelihood (lower is better) by time sample, computed on left-out data. Each line in Figure 3.12 corresponds to a given model with the parameter m corresponding to the best couple of parameter (p, m), as listed in Table 3.1. One can observe that the curves of negative log-likelihood are not convex, yet they exhibit rather

Table 3.1 – Best couple of parameter (p, m) as selected by maximum likelihood through cross-validation, for different models and datasets: (a) rodent striatum, (b) rodent hippocampus, and (c) human auditory cortex. See also Figure 3.12.

Model	**Dataset (a)**	**Dataset (b)**	**Dataset (c)**
AR	$(179, 0)$	$(80, 0)$	$(0, 0)$
HAR	$(179, 2)$	$(80, 2)$	$(0, 2)$
PDAR	$(82, 2)$	$(15, 1)$	$(29, 1)$
DAR	$(90, 2)$	$(15, 2)$	$(24, 2)$

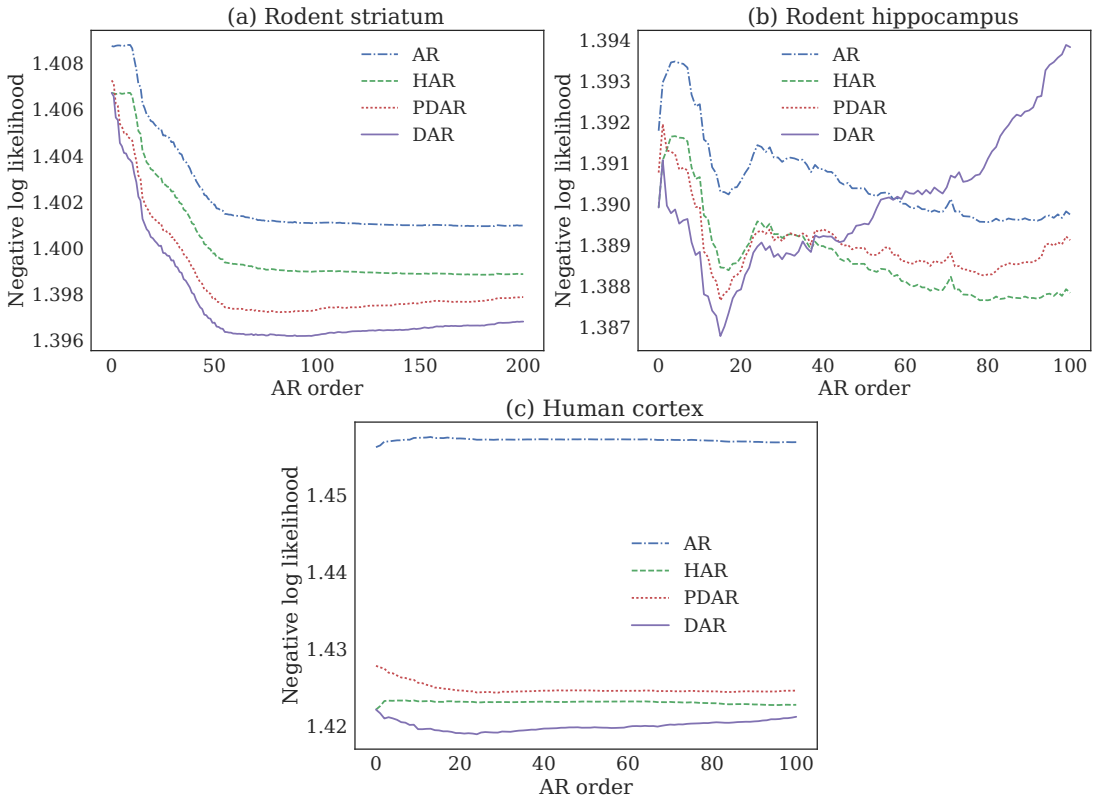


Figure 3.12 – Negative log-likelihood (lower is better) of different models with different AR order p, on three datasets: (a) rodent striatum, (b) rodent hippocampus, and (c) human auditory cortex. The negative log likelihood is computed on left-out data. DAR models always outperform PDAR models, which suggests that the coupling phenomenon is associated with amplitude fluctuations in the driver.

clear minima used to define the optimal parameters.

This result shows that the coupling phenomenon is associated with amplitude fluctuations, a kind of phase/amplitude-amplitude coupling, as it was previously observed in van Wijk et al. (2015). In their work, the GLM parametric method (Penny et al., 2008) was improved when taking into account the amplitude of the slow oscillation. Here, we use our generative model framework to provide an easy comparison tool through the likelihood, to validate this neuroscientific insight directly from the signals.

3.3 Statistical significance

3.3.1 Quantifying significance

A general challenge of PAC analysis is to quantify the significance of a result. Indeed, a PAC metric generally gives a value which is large when there is a coupling, and close to zero when there is no coupling, but we need to quantify what it means to be close to zero.

Permutation techniques A standard way to tackle this issue is to use permutation techniques to estimate the distribution of the metric in the absence of coupling (Canolty et al., 2006, Aru et al., 2015). To do so, a standard technique is to add a large random time shift between x and y, assuming no coupling is possible for large time shifts. The PAC value between the shifted signals is thus close to zero. Repeating this process multiple times (*e.g.* 1000 times), we obtain an empirical probability distribution, and we can estimate the probability of observing a given coupling by chance, *i.e.* as a false positive. This probability is called a p-value, and it is usually compared to an arbitrary threshold usually set at $\alpha = 0.05$, though multiple critics advocate to use a smaller one.

This approach is sometimes applied directly on the comodulogram. In this case, as a comodulogram contains multiple couple of frequencies (f_x, f_y), the p-value needs to be corrected to avoid suffering from multiple testing issues. To correct it, the Bonferroni adjustment is very popular. It simply divides the threshold α by the number of tests (Bonferroni, 1936), thus controlling the error rate on the entire collection of test. However, as this adjustment is very strict, another common strategy is to estimate a false discovery rate (FDR). The FDR quantifies the percentage of false positive associated with a given threshold (Benjamini and Hochberg, 1995).

Another approach to avoid multiple testing issue is to consider the distribution of the maxima. After adding a random time shift, the idea is to compute the entire comodulogram, and to keep only the maximum value, repeating this process multiple times to estimate the distribution of maxima in the case of uncoupled time shifted signals. Then, we can take the 99-percentile of this distribution to obtain the threshold associated with the p-value $p = 0.01$. This method does not suffer from multiple testing issues contrary to the previous approach that estimates a different null distribution for each frequency couple in a comodulogram. This is the method we used in our experiments.

Importantly, we avoid using a z-score, since we cannot assume the distributions to be Gaussian. A z-score is computed by standardizing the PAC value, using the mean and standard deviation of the PAC distribution on uncoupled time shifted signals. This standard step, for instance used in Canolty et al. (2006), assumes that the distribution is Gaussian, which is not true in general. Since we have an estimate of the distribution, a p-value can be estimated directly from the distribution, without the use of a z-score.

We present in Figure 3.13 an example of comodulogram with contours lines delineating the significant values. The comodulogram was computed on a signal with simulated PAC between 5 Hz and 50 Hz, with $T = 1000$ points sampled at $f_s = 200$ Hz, using DAR models. We computed 200 comodulograms with random time shifts between x and y, to estimate the distribution of values when there is no coupling. In the left image, we transformed the PAC value into z-scores, corresponding to the distribution of each couple of frequency, *i.e.* each point of the comodulogram. We then derived a

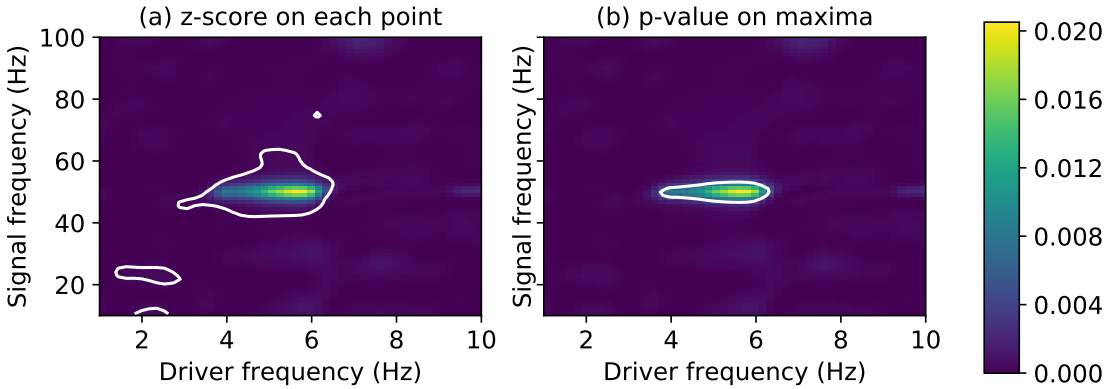


Figure 3.13 – Significance contours on a comodulogram (white lines). (a) Significance computed with a z-score on each couple of frequency. (b) Significance computed with a p-value on the distribution of maxima.

threshold from the z-scores. In the right image, we computed a single threshold from the distribution of comodulogram maxima. We observe that the latter is more resistant to false positives.

Model selection techniques Analyzing PAC with DAR models enables a different way to evaluate if there is significant PAC. Indeed, as described in details in Section 3.2, we can use the model likelihood to perform model selection. This approach is not possible with standard PAC metrics, since they do not offer a goodness of fit score.

For example, comparing DAR models and AR models, we can evaluate if there is a significant spectral modulation with respect to a given driver. If the AR models are selected, it means that the potential spectral modulations were not strong enough to better fit the signal with a DAR model than with an AR model. On the contrary, if DAR models are selected, it means that there is a spectral modulation coupled with the given driver, strong enough to better fit than linear AR models. The fitted DAR model can then be analyzed to describe which kind of spectral modulation is present, using for instance conditional PSD representations as in Figure 3.6.

It is worth recalling that DAR models and AR models are nested, in the sense that AR models are a special case of DAR models. Thus, a DAR model would only have more degrees of freedom, and would thus always better fit the training signal. Therefore, it is critical to estimate the fitting not on the training signal, but on left-out signals, *i.e.* data not used during model estimation.

3.3.2 Robustness to small samples

Given that DAR models are parametric with a limited number of parameters to estimate, less time samples may be needed to estimate PAC as compared to non-parametric methods. We tested this assumption using simulated signals of varying duration. We computed their comodulograms (as in Figure 3.7) and selected the frequencies of maximum coupling. For each duration, we simulated 200 signals, selected the 200 frequency pairs corresponding to the maxima, and plotted them in a 2D representation. We then compared the following four methods: DAR models with $(p, m) = (10, 1)$, the GLM-based model (Penny et al., 2008), and two non-parametric methods (Tort et al., 2010, Özkurt and Schnitzler, 2011). Results shown in Figure 3.14 show that

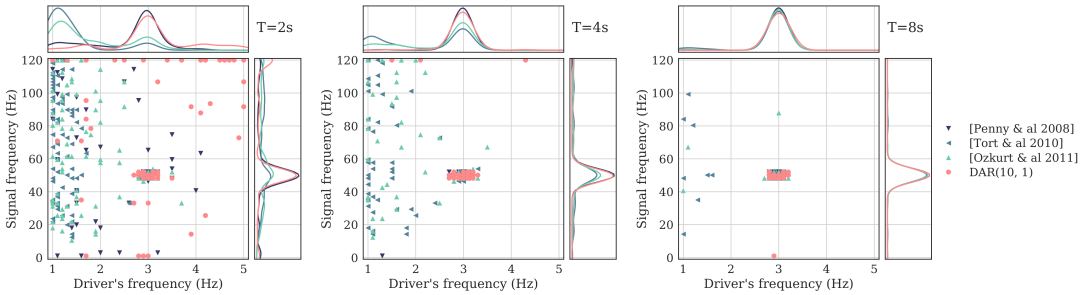


Figure 3.14 – Robustness to small samples. Frequencies of the maximum PAC value, with four methods: a DAR model with $(p, m) = (10, 1)$, the GLM-based model from Penny *et al.* (Penny et al., 2008), and two non-parametric models from Tort *et al.* (Tort et al., 2010) and Ozkurt *et al.* (Özkurt and Schnitzler, 2011). Each point corresponds to one signal out of 200. Kernel density estimates are represented above and at the right of each scatter plot. From left to right, the simulated signals last $T = 2$, 4 and 8 seconds. The signals are simulated with a PAC between 3 Hz and 50 Hz. The DAR models correctly estimate this pair of frequency even with a short signal length, as well as the GLM-based metric (Penny et al., 2008), while the two other metrics (Tort et al., 2010, Özkurt and Schnitzler, 2011) are strongly affected by the small length of the signals, and do not estimate the correct pair of frequency.

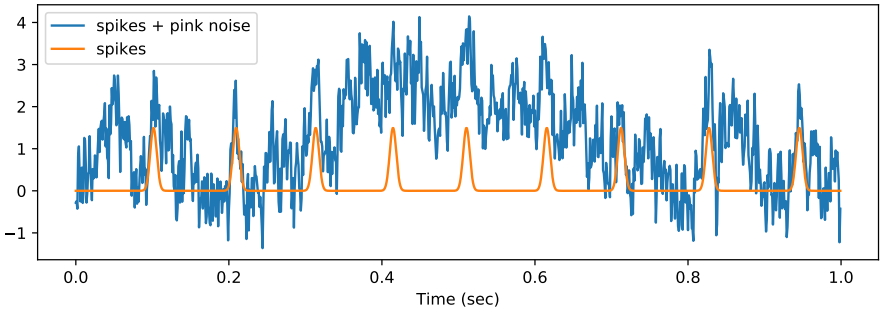
parametric approaches (DAR models and GLMs) provided a more robust estimation of PAC frequencies with short signals ($T = 2$ sec) than non-parametric methods.

The robustness to small sample size is a key feature of parametric models, as it significantly improves PAC analysis during shorter experiments. When undertaking a PAC analysis across time using a sliding time window, parametric models should therefore provide more robust PAC estimates. Note that the specific time values in these simulations should not be taken as general guidelines as they depend on the simulation parameters such as the signal-to-noise ratio. However, across all tests we ran, parametric methods consistently provided more accurate results than non-parametric ones.

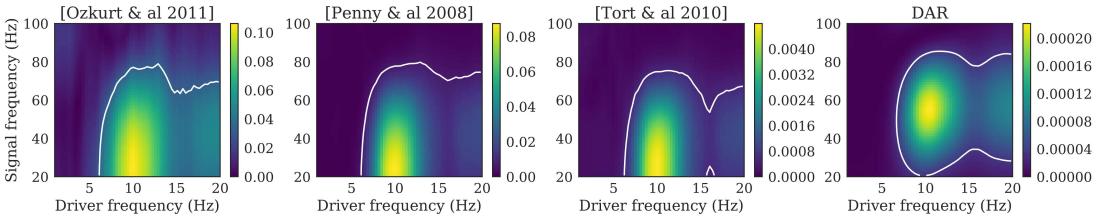
3.3.3 Spurious PAC

A recent concern in PAC analysis is that all PAC metrics may detect a coupling even though the signal is not composed of two cross-frequency coupled oscillators (Kramer et al., 2008, Lozano-Soldevilla et al., 2016, Amiri et al., 2016, Gerber et al., 2016, Vaz et al., 2017). It may happen for instance with sharp slow oscillations, described in humans intracranial recordings (Cole et al., 2016). Sharp edges are known not to be well described by a Fourier analysis, which decomposes the signal in a linear combination of sinusoids. Indeed, such sharp slow oscillations create artificial high frequency activity at each sharp edge, and these high frequencies are thus artificially coupled with the slow oscillations. This false positive detection is commonly referred to as "spurious" coupling (Jensen et al., 2016).

Figure 3.15a shows some simulated spurious PAC dataset, generated using a spike train at 10 Hz and pink noise, as described in (Gerber et al., 2016), and Figure 3.15b shows comodulograms computed on this signal. The figure shows that all four methods, including the proposed one, detect some significant PAC, even though there is no nested oscillations in the signal. Even though our method does not use filtering in the



(a) Simulated signal with spurious PAC.



(b) Comodulograms obtained on a simulated signal with spurious PAC.

Figure 3.15 – Spurious PAC. (a) Spurious PAC was generated using a spike train at 10 Hz and pink noise, as described in (Gerber et al., 2016). (b) All four methods, including the proposed one, detect some significant PAC, even though there is no nested oscillations.

high frequencies, it does not solve this issue and is affected in the same way as other traditional PAC metrics. Indeed, our work shed light on the wide-band property of the slow oscillations, but DAR models cannot cope with full-band slow oscillations, which contain strong harmonic components in the high frequencies. However, we consider that such "spurious" PAC can also be a relevant feature of a signal, as stated in (Cole et al., 2016). In their study, they show that abnormal beta oscillations (13-30 Hz) in the basal ganglia and motor cortex underlie some "spurious" PAC, but are actually a strong feature associated with Parkinson's disease. A robust way to disentangle the different mechanisms that lead to similar PAC results remains to be developed.

3.4 Discussion

Cross-frequency coupling Cross-frequency coupling (CFC) and phase-amplitude coupling (PAC) more specifically have been proposed to play a fundamental role in neural processes ranging from the encoding, maintenance and retrieval of information (Buzsáki, 2010, Jensen and Colgin, 2007, Lisman and Jensen, 2013, Axmacher et al., 2010, Fell and Axmacher, 2011, Kaplan et al., 2014, Hyafil et al., 2015), to large-scale communication across neural ensembles (Canolty and Knight, 2010, Jirsa and Müller, 2013, Khan et al., 2013, Florin and Baillet, 2015). While a steady increase in observations of PAC in neural data has been seen, how to best detect and quantify such phenomena remains difficult to settle.

DAR models We argue that a method using DAR models, as described here, is rich enough to capture the time-varying statistics of brain signals in addition to provide efficient inference algorithms. These non-linear statistical models are probabilistic,

allowing the estimation of their goodness of fit to the data, and allowing for an easy and fully controlled comparison across models and parameters. In other words, they offer a unique principled data-driven model selection approach, an estimation strategy of phase/amplitude-amplitude coupling based on the approximation of the actual signals, a better temporal resolution of dynamic PAC and the estimation of coupling directionality.

Model and parameter selection One of the main features of PAC estimation through our method is the ability to compare models or parameters on non-synthetic data. On the contrary, traditional PAC metrics *cannot* be compared on non-synthetic data, and two different choices of parameters can lead to different interpretations. There is no legitimate way to decide which parameter shall be used with empirical data using traditional metrics. The likelihood of the DAR model that can be estimated on left-out data offers a rigorous solution to this problem.

Experiments We presented results on both simulated signals and empirical neurophysiological signals. The simulations gave us an illustration of the phenomenon we want to model, and helped us understand how to visualize a fitted DAR model. They also served a validation purpose for the bandwidth selection approach that we performed on real data. Using the data-driven parameter selection on non-synthetic signals, we showed how to choose sensible parameters for the filtering of the slow oscillation. All empirical signals are different, and it was for example reported in the neuroscience literature that peak frequencies vary between individuals (Haegens et al., 2014) and that this should not be overlooked in the analysis of the data. The parameter selection based on fitted DAR models makes it possible to fit parameters on individual datasets. Our results also shed light on the asymmetrical and wide-band properties of the slow oscillation, which could denote crucial features involved in cognition (Cole and Voytek, 2017).

Driver's amplitude fluctuations The second novelty of our method stands in considering the amplitude fluctuations of the slow oscillation in the PAC measure and not only its phase. Using the rodent and human data, we showed that the instantaneous amplitude of the slow oscillation influences the coupling in PAC, as it was previously suggested in (van Wijk et al., 2015). The amplitude information should therefore not be discarded as it is done by existing PAC metrics. For instance, the measure of alpha/gamma coupling reported during rest (Osipova et al., 2008, Roux et al., 2013) should incorporate alpha fluctuations when studied in the context of visual tasks (Voytek et al., 2010), as an increase of alpha power is often concomitant with a decrease of gamma power (Fries et al., 2001). The comparison between DAR models considering or not these low-frequency power fluctuations would inform on the nature of the coupling: purely phase-amplitude, or rather phase/amplitude-amplitude. In Tort et al. (2008), both theta power changes and modulation of theta/gamma PAC were reported in rats having to make a left or right decision to find a reward in a maze. The use of our method could decipher whether the changes in coupling were related to the changes in power, informing on the underlying mechanisms of decision-making. Moreover, as our method models the entire spectrum simultaneously, a phase-frequency coupling could potentially be captured in our models. Therefore, our method is not limited to purely phase-amplitude coupling, and extends the traditional CFC analysis.

Robustness to small samples Furthermore, in those types of experiments, changes in PAC can be very fast depending on the cognitive state of the subject. Therefore, the need for dynamic PAC estimates is growing (Tort et al., 2008). We showed with simulations that DAR models are more robust than non-parametric methods when estimating PAC on small time samples. This robustness is critical for time-limited experiments and also when analyzing PAC across time in a fine manner, typically when dynamic processes are at play.

Delay estimation Last but not least, likelihood comparison can also be used to estimate the delay between the coupled components, which would give new insights on highly debated questions on the role of oscillations in neuronal communication (Fries, 2005, Bastos et al., 2015). For example, a delay close to zero could suggest that the low and high frequency components of the coupling might be generated in the same area, whereas a large delay would suggest they might come from different areas. As an alternative interpretation, the two components may come from the same area, but the coupling mechanism itself might be lagged. In this case, a negative delay would suggest that the low frequency oscillation is driven by the high frequency oscillations, whereas a positive delay would suggest that the low frequency oscillation drives the high frequency amplitude modulation. In any case, this type of analysis will provide valuable information to guide further experimental questions.

Multivariate signals The method we presented here uses univariate signals obtained invasively in rodents or humans. As a lot of neurophysiological research uses non-invasive MEG or EEG recordings containing multiple channels, a multivariate analysis could be of high interest. One way to use data from multiple channels is to estimate a single signal using a spatial filter such as in (Cohen, 2017). Such a method is therefore complementary to univariate PAC metrics like ours which can be applied to the output of the spatial filter. The method from (Cohen, 2017) builds spatial filters that maximize the difference between, say, high-frequency activity that appears during peaks of a low-frequency oscillation *versus* high-frequency activity that is unrelated to the low-frequency oscillation. Again, from the signal obtained with the spatial filter, it is straightforward to adapt most PAC metrics such as our method. In Section 4.2, we describe how to derive these spatial filters directly from DAR models, for multivariate PAC analysis.

Conclusion Neurophysiological signals have all the statistical properties to make them a challenge from a signal processing perspective. They contain non-linearities, non-stationarities, they are noisy and they can be long, hence posing important computational challenges. Our method based on DAR models offer novel and more robust possibilities to analyze neurophysiological signals, paving the way for new insights on how our brain functions via spectral interactions using local or distant coupling mechanisms.

4

Extensions to DAR models

"I'm killing time while I wait for life to shower me with meaning and happiness."
– *Bill Watterson*

Contents

4.1 Driver estimation in DAR models . . . 74
4.1.1 Potential approaches . . . 74
4.1.2 Driver estimation . . . 76
4.1.3 Experiments . . . 78
4.2 Multivariate PAC . . . 80
4.2.1 Multivariate PAC with GED . . . 80
4.2.2 Multivariate PAC with DAR models . . . 82
4.2.3 Model estimation . . . 83
4.2.4 Experiments . . . 85
4.3 Spectro-temporal receptive fields . . . 86
4.3.1 Spectro-temporal receptive fields . . . 86
4.3.2 STRF and DAR models . . . 88
4.3.3 Experiments . . . 88

In this chapter, we describe different extensions to DAR models, centered on the driver estimation. We first describe how to estimate the driver as a weighted sum over a set of potential drivers. Then we show how to use such estimated driver in two applications. In the case of multivariate signals, we describe how to estimate virtual channels on which we can apply DAR models to describe PAC. We also use driver estimation in an encoding setting, modeling brain activity conditionally to the stimuli using DAR models, leading to a spectro-temporal receptive field (STRF) estimation.

This chapter covers the following publication:

- Dupré la Tour, T., Grenier, Y., and Gramfort, A. (2018a). Driver estimation in non-linear autoregressive models. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE

4.1 Driver estimation in DAR models

In Chapter 2, we introduced DAR models, a class of deterministic continuously switching AR models which use polynomial parametrizations with respect to a driving signal x. In these models, the driver is assumed to be known, and do not need to be estimated.

In Chapter 3, we described how to use DAR models on neurophysiological signals to model spectral modulation driven by a given slow oscillation, a phenomenon known as phase-amplitude coupling (PAC). In this case, the driver x was extracted from the signal using band-pass filtering, and we discussed in details how to use model selection to choose filtering parameters among a discrete grid of parameters.

However, due to the finite set of grid points, such driver may not correspond perfectly to the spectral dynamic of the modeled signal. Here, we propose to optimize the driver, to better fit the spectral trajectory of the modeled signal. To do so, a critical element is to use the entire signal for the spectral estimation, and not only a small time window. Indeed, we assume that the spectral states are repeated multiple times in the signal, and we should use the multiple repetitions to improve the estimation of the spectral states.

4.1.1 Potential approaches

Time-varying AR models To estimate the spectral trajectory without the driver, on possible approach could be to use unstructured time-varying AR models (TVAR) (Dahlhaus, 1996). In these models, one updates the AR model at each time point, with a single gradient descent step. Obviously, the step size is here critical, as it defines the smoothness of the parameter evolution. One solution is to use different step sizes, and to aggregate the instantaneous AR estimators, for example as described in Giraud et al. (2015).

TVAR models are very general, and do not use the repetition of spectral states, as present in PAC. To use this powerful property, we could approximate the obtained trajectory (in $\mathbb{R}^p$) of the instantaneous AR coefficients into a low-rank trajectory $A \approx X\tilde{A}$, where $A \in \mathbb{R}^{(T,p)}$, $X \in \mathbb{R}^{(T,m)}$, and $\tilde{A} \in \mathbb{R}^{(m,p)}$, with $m \ll p$. This idea could be formulated with the nuclear norm (Recht et al., 2010), using for instance a singular value thresholding scheme to estimate X (Cai et al., 2010). The low-rank trajectory X would be the estimated driver.

This idea is similar to the approach briefly described in Dimitriadis et al. (2016), where PAC is estimated on a sliding time-window, independently on each electrode. Then, at group level, a low-rank approximation of the time-varying PAC patterns is estimated, using the neural gas algorithm (Martinetz et al., 1991) to detect a low number of PAC *micro-states*. In Dimitriadis et al. (2016), the micro-states are defined by the pattern on the 64 electrodes, whereas in our case, we would define the states with the p AR coefficients.

Hidden Markov models There exists other attempts to define micro-states, making efficient use of the data when the same patterns occur recurrently at other points of time to improve estimation. Hidden Markov models (HMM) (Baum and Petrie, 1966) are probabilistic models which infer a number of states in the data, modeling with a Markov chain the switch between states. Each state is defined by a different probability distribution, from which the data is observed. The hidden states probabilities and

the probability distributions are typically inferred jointly, for example through an EM algorithm (Rabiner, 1989, Cappé et al., 2006, Rukat et al., 2016).

HMM have been used to estimate fast transient brain states in MEG data. The different states were defined with the noise means and covariance matrices (Woolrich et al., 2013, Baker et al., 2014, Vidaurre et al., 2017), or with vector AR (VAR) coefficients (Vidaurre et al., 2016).

Using AR models to define the different states, we could estimate PAC as composed of such a succession of spectral states. Such approach is similar to the work of Hamilton (1989), which uses non-linear AR models assuming a Markov chain structure in the driver. We could alternatively define the states with DAR models, which would then lead to the estimation of different PAC states.

Sum of potential drivers If the previous approaches could be successful, in the case of PAC, we can also use the strong assumption that the spectral dynamic is actually present in the signal. Indeed, we know that some low frequency bands contain a lot of information about the spectral modulation trajectory.

Moreover, inline with the philosophy of previous chapter, we would like to be able to evaluate our models on out-of samples data. Therefore, we cannot use methods that need to re-estimate the model on the validation data, such as TVAR models or HMM.

To leverage the low-frequency information, and to enable evaluation on a validation set, we propose a different approach, using multiple drivers in DAR models. In fact, one could potentially add more drivers directly into DAR models, just as we used two drivers in (3.2). This is similar to the work of Grenier (1983), Jachan et al. (2007), Spiridonakos and Fassois (2014), but it could lead to a very large number of degrees of freedom, especially if we use the cross terms between drivers as in (3.2). Estimation would have high variance, increasing the risk of model overfit. We would also lose the nice interpretability of the single driver, which is critical in neuroscience applications.

Instead, we propose to build a weighted average of potential drivers as in Chen and So (2006), Wu and Chen (2007), where the obtained driver was used on a threshold AR model and on a probabilistic switching AR model respectively. In our case, we use the weighted sum as a single driver in the polynomial parametrization of DAR models (*c.f.* Chapter 2). We decompose the driver into:

$$x(t) = \sum_{n=1}^{N} \alpha_n x_n(t). \tag{4.1}$$

The set of potential drivers $\{x_n\}$ can be for instance, a Fourier basis $x_n(t) = \exp(j2\pi nt)$ as in Jachan et al. (2007), or a Gabor dictionary (Feichtinger and Strohmer, 2012). Another choice is to use a set of delayed signals $x_n(t) = z(t-n)$ with $-M \leq n \leq M$. In this case, the coefficients α_n define a linear filter applied to z. We used this set in our experiments.

4.1.2 Driver estimation

We estimate the optimal driver weights α_n by maximizing the likelihood L of the model:

$$L = \prod_{t=p+1}^{T} \frac{1}{\sqrt{2\pi\sigma(t)^2}} \exp\left(-\frac{\varepsilon(t)^2}{2\sigma(t)^2}\right) \tag{4.2}$$

$$-2\log(L) = T\log(2\pi) + \sum_{t=p+1}^{T} \frac{\varepsilon(t)^2}{\sigma(t)^2} + 2\sum_{t=p+1}^{T} \log(\sigma(t))$$

Using an alternating optimization approach, we optimize the DAR model coefficients while keeping the driver fixed (*c.f.* Subsection 2.1.3), and we optimize the driver weights α_n while keeping the DAR model fixed. As this problem is non-convex, weights initialization is key to find good local minima.

Optimizing the driver weights can be done with various optimization algorithms. Here, we choose the quasi-Newton L-BFGS algorithm (Byrd et al., 1995), which only requires to compute the gradients. The gradient with respect to the weights reads:

$$\frac{\partial \log L}{\partial \alpha_n} = -\sum_{t\in\Theta} \left(\frac{\varepsilon(t)}{\sigma(t)^2}\frac{\partial\varepsilon(t)}{\partial\alpha_n} + (1 - \frac{\varepsilon(t)^2}{\sigma(t)^2})\frac{\partial \log\sigma(t)}{\partial\alpha_n} \right)$$

where $\Theta = [p+1, T]$ in the general case. In our experiments, we restricted the sum to $\Theta = [\max(p+1, M), T-M]$ to remove filtering issue at the edges. In particular, when multiple values of M are compared, we need to restrict the comparison to $\Theta = [\max(p+1, M_{max}), T - M_{max}]$.

The partial derivatives read:

$$\frac{\partial\varepsilon(t)}{\partial\alpha_n} = x_{re,n}(t)\frac{\partial\varepsilon(t)}{\partial x_{re}} + x_{im,n}(t)\frac{\partial\varepsilon(t)}{\partial x_{im}} \tag{4.3}$$

$$\frac{\partial \log\sigma(t)}{\partial\alpha_n} = x_{re,n}(t)\frac{\partial \log\sigma(t)}{\partial x_{re}} + x_{im,n}(t)\frac{\partial \log\sigma(t)}{\partial x_{im}} \tag{4.4}$$

Let's note $x\_$ when an expression is similar for both x_{re} and x_{im}. From equations (2.1), (3.2), and (3.3), we obtain:

$$\frac{\partial\varepsilon(t)}{\partial x\_} = \sum_{i=1}^{p} A_i^\top \frac{\partial X(t)}{\partial x\_} y(t-i) \tag{4.5}$$

$$\frac{\partial \log\sigma(t)}{\partial x\_} = B^\top \frac{\partial X(t)}{\partial x\_} \tag{4.6}$$

Finally, we can rewrite:

$$\frac{\partial \log L}{\partial \alpha_n} = -\sum_{t\in\Theta} \Big(x_{re,n}(t) g_{re}(t) + x_{im,n}(t) g_{im}(t) \Big) \tag{4.7}$$

with

$$g\_(t) = \left(\frac{\varepsilon(t)}{\sigma(t)^2}\frac{\partial\varepsilon(t)}{\partial x\_} + (1 - \frac{\varepsilon(t)^2}{\sigma(t)^2})\frac{\partial \log\sigma(t)}{\partial x\_} \right) \tag{4.8}$$

Computing the gradient involves $\mathcal{O}(Tp\tilde{m})$ operations to compute $g\_$, and $\mathcal{O}(TN)$ operations to compute the gradient in (4.7). In the special case $x\_{,n}(t) = z\_(t-n)$, we can rewrite (4.7) into a convolution, which can be performed in $\mathcal{O}(T\log(T))$ using the fast Fourier transform (FFT).

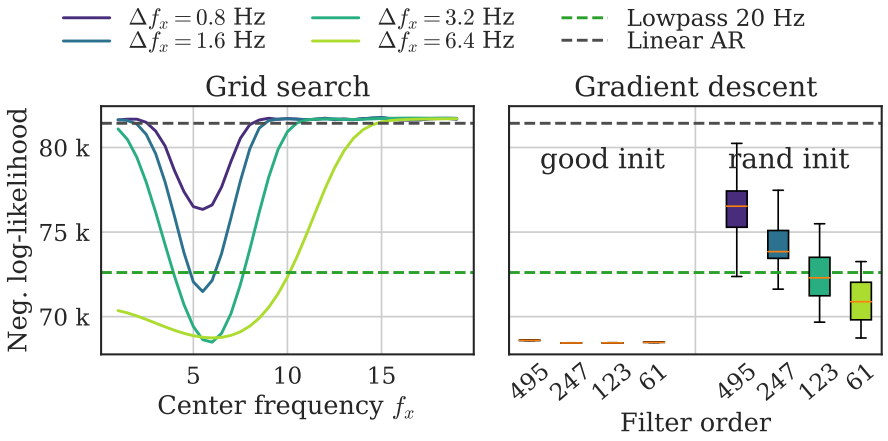


Figure 4.1 – Negative log-likelihood of DAR models fitted with different drivers (lower is better) and evaluated on a validation set. (Left) Grid search: The drivers were bandpass filtered at center frequency f_x with a bandwidth Δf_x. (Right) Gradient descent: The filters extracting the drivers were optimized by gradient descent, using either several bandpass filter initializations or some random initializations. All bandpass filter initializations with center frequency ranging from 2 Hz to 8 Hz gave optimal and comparable likelihoods. Filter order (495, 247, 123, 61) respectively correspond to bandwidths (0.8, 1.6, 3.2, 6.4) Hz.

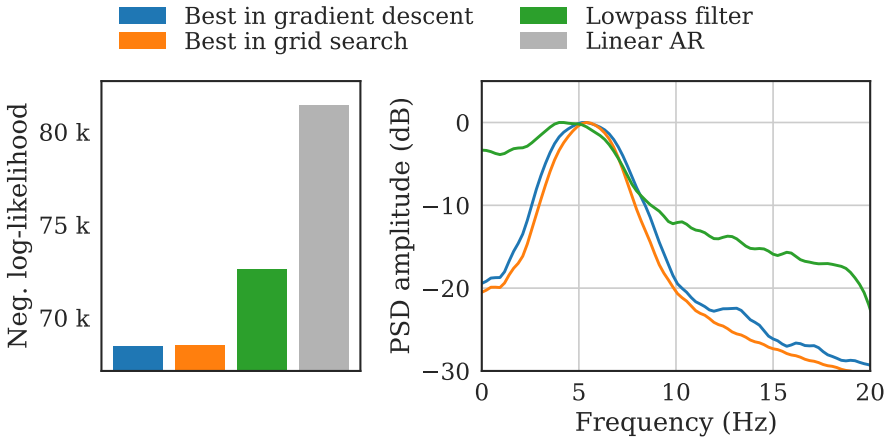


Figure 4.2 – Comparison of 4 models: 3 DAR fitted with different drivers, and 1 linear AR for reference. Both gradient descent and grid search strategies give comparable results, which are much better than when using the driver on the entire band [0, 20] Hz. (Left) Negative log-likelihood on a validation set (lower is better). (Right) Power spectral density of the best driver for each strategy.

Adding a symmetry constraint In the special case $x_n(t) = z(t-n)$, if we want to make sure the filter is zero-phase, we just need to make the filter symmetric. We can rewrite the driver as $x = 2\alpha_0 x_0 + \sum_{n=1}^{M} \alpha_n (x_n + x_{-n})$, where $N = 2M + 1$. A factor 2 is added in front of α_0 to have the same scale as the other weights. The gradient is updated into $\frac{\partial x}{\partial \alpha_n} = x_n + x_{-n}$ if $n > 0$ and $\frac{\partial x}{\partial \alpha_n} = 2x_0$ if $n = 0$.

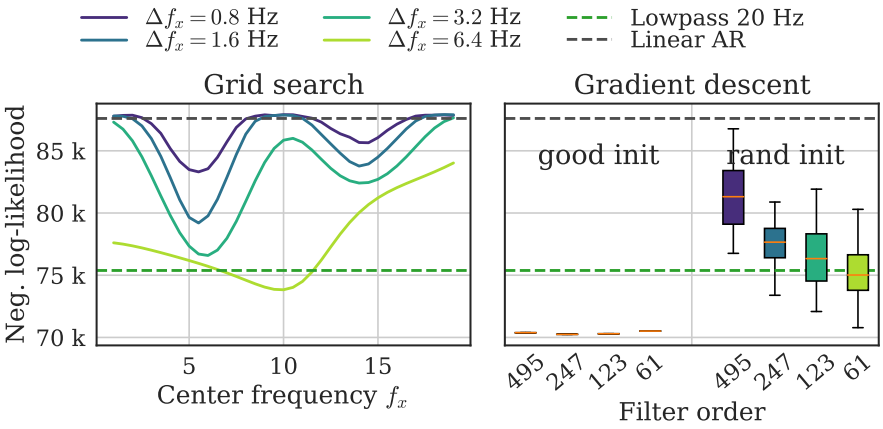


Figure 4.3 – Same as Figure 4.1, but using a bimodal driver at 5 and 14 Hz. The gradient descent strategy gave better results than grid-search, when using a good initialization.

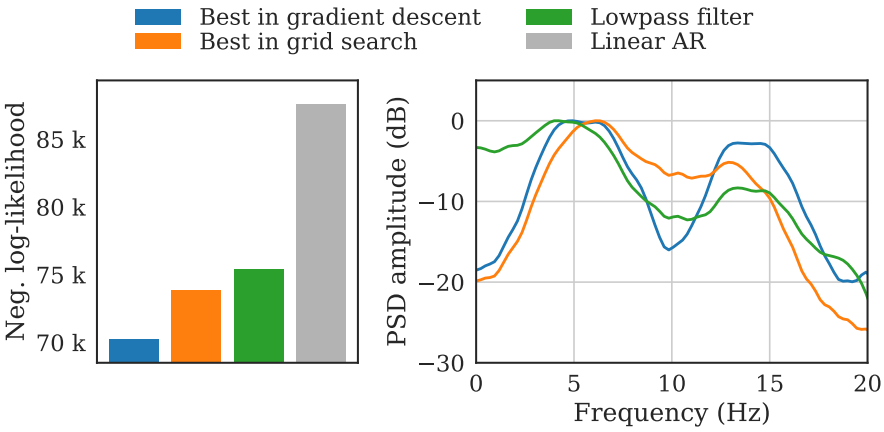


Figure 4.4 – Same as Figure 4.2, but using a bimodal driver. With a more complex spectral structure, the gradient descent strategy gives much better results than the grid search one, which is limited to single mode bandpass filters.

4.1.3 Experiments

Simulated data We created simulated signals with artificial coupling between a driver and a sinusoid. The signals were sampled at $f_s = 240$ Hz, and have a length $T = 10^5$. We first created a driver x by filtering a Gaussian white noise with a filter $w(t) = b(t)\exp(2j\pi f_x t)$, where b is a Blackman window of order $2\lfloor 1.65 f_s/\Delta f_x \rfloor + 1$, chosen to have a bandwidth of Δf_x at -3 dB. This driver x was then used to modulate the amplitude of a sinusoid $y(t) = s(x(t))\sin(2\pi f_y t)$ where s is a sigmoid function. The modulated sinusoid and the driver were summed up, along with some noise. The noise was pink with a frequency slope f^{-2} above 3 Hz and a plateau below 3 Hz, to mimic neurophysiological signals. The amplitude of the three signals were chosen to have a signal-to-noise ratio (SNR) of 5 dB at f_x and of 20 dB at f_y. Importantly, we do not use a DAR model to simulate such data.

We compared different choices of driver, using DAR models of order $(p, m) = (10, 2)$, and comparing their negative log-likelihood on a validation set using cross-validation. We split the signal into 10 parts of equal size, fitted a DAR model on 5 random parts, and estimating the negative log-likelihood on the 5 other parts, and repeating this

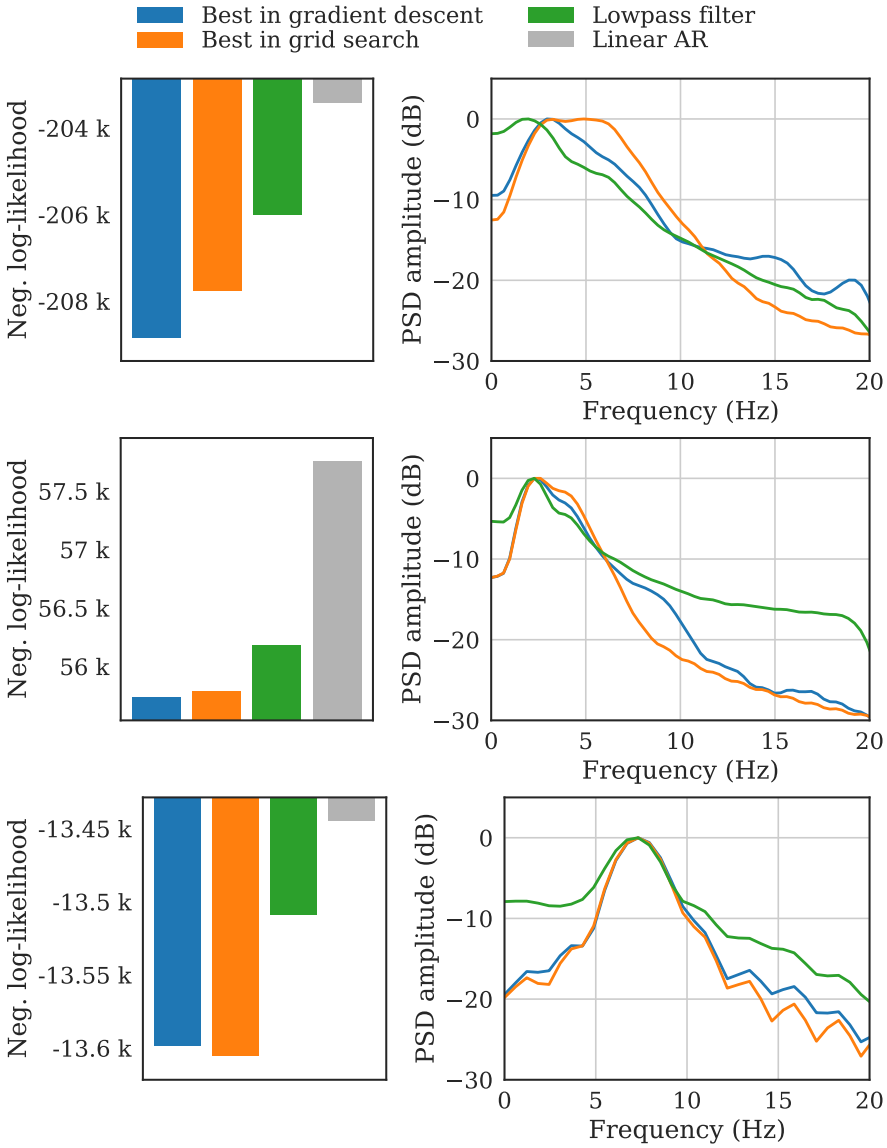


Figure 4.5 – Same as Figure 4.2, using three neurophysiological dataset: (Top) Human cortex ECoG, (Middle) rodent striatal LFP, (Bottom) rodent hippocampal LFP.

process 10 times. To fit the models, we first separated the low frequencies from the high frequencies using a low-pass filter at 20 Hz, which gave z and y respectively. We extracted the driver x from z using different strategies described below, and fitted DAR models on signal y with driver x.

The first strategy was grid-search, which searched over a set of bandpass filters as described above. The second strategy used the proposed gradient descent to optimize freely the filter extracting the driver. In this strategy, we used different initializations, since the problem is non-convex and thus may lead to different local minima. Initial filters were either bandpass filters as in the first strategy with center frequency ranging from 2 Hz to 8 Hz, or random filters generated with Gaussian white noise. We also compared with the entire low-pass filter z, and with a linear AR which uses no driver.

As a sanity check of the gradient descent strategy, the first simulation used a single-band $(f_x, \Delta f_x) = (5, 3)$ ground-truth driver, and results are presented in Figure 4.1 and Figure 4.2. Both strategies gave the same best results. We also observed that gradient descent converged to about the same log-likelihood for a large set of reasonable initializations. However, if the initialization does not capture CFC, the optimization leads to poorer results, yet better than the linear AR, even on the validation set.

To present the differences between the two strategies, the second simulation used a bimodal ground-truth driver, built as the sum of two drivers $x = x_1 + 0.4x_2$, filtered from Gaussian white noise with respectively $(f_{x_1}, \Delta f_{x_1}) = (5, 3)$ and $(f_{x_2}, \Delta f_{x_2}) = (14, 3)$. Results are presented in Figure 4.3 and Figure 4.4. In this case, the grid-search strategy could not correctly capture the two bands, and chose a large filter centered at 10 Hz. It performed only marginally better than the full low-pass signal z. In contrast, the optimization by gradient descent correctly captured the two bands, leading to much better results.

Empirical data We also validated our approach on three empirical neurophysiological signals containing CFC (*c.f.* Subsection 3.1.5. The results presented in Figure 4.5 show that the gradient descent strategy leads to a lower negative log-likelihood than the grid-search strategy for the human cortical signal. In this case, the difference could be related to an asymmetrical shape of the driver spectral peak at 4 Hz. This asymmetrical PSD was also proposed to explain the shift in the minimum in Figure 3.5.

For the rodent striatal signal, the likelihood gain is more subtle, but we witness the same difference in the PSD shape as in the human cortical signal. Indeed, we observe that the best filter obtained by gradient descent is asymmetrical, as its PSD decreases more slowly between 3 and 10 Hz than the predefined filter.

For the rodent hippocampal signal, the gradient descent leads to a poorer likelihood than the grid-search strategy, which probably accounts for some over-fitting. Indeed, we see in the PSD plot that the best filters are very similar in both strategies. In this case, the obtained driver might be close to be optimal, and the gradient descent strategy is more prone to over-fitting since it has more degrees of freedom than the grid-search strategy. Another possible interpretation is that the driver changes overtime, and it is thus better to have a more crude estimation of the filter.

4.2 Multivariate PAC

4.2.1 Multivariate PAC with GED

Most electrophysiological recordings are multivariate, in the sense that multiple channels are recorded at the same time through multiple sensors. For instance, EEG and MEG recordings can contain more than 300 channels. The information is often highly redundant across channels, but the multiple channels can be used to extract more knowledge from low signal-to-noise ratio. Indeed, if a signal is spread with high similarity on many channels, we can expect some part of the noise to be independent. The goal of multivariate analysis is to leverage this similarity to improve the robustness to noise.

As a lot of neurophysiological research uses non-invasive MEG or EEG recordings containing multiple channels, a multivariate PAC analysis could be of high interest. PAC is defined as a relation between two univariate signals, or between two frequency

bands of a single univariate signal. In the case of PAC, one way to use data from multiple channels is to estimate a single signal using a spatial filter such as in Cohen (2017). Such reconstructed univariate signal is often referred as a *virtual channel*. This method is complementary to univariate PAC metrics which can be applied to the virtual channel. DAR models can also be applied on such virtual channels.

The method from Cohen (2017) builds spatial filters that maximize the difference between, say, high-frequency activity that appears during peaks of a low-frequency oscillation *versus* high-frequency activity that is unrelated to the low-frequency oscillation. More specifically, with a multivariate signal over C channels, it builds two covariance matrices, $R \in \mathbb{R}^{C \times C}$ and $S \in \mathbb{R}^{C \times C}$, computed on the two different regimes we want to distinguish. Then it computes the generalized eigen-decomposition (GED) of R and S, which produces eigenvectors $w \in \mathbb{R}^C$ and eigenvalues $\lambda \in \mathbb{R}$ such that:

$$Sw = \lambda Rw \tag{4.9}$$

Such eigenvectors are easily interpretable when R is invertible, as eigenvectors of $R^{-1}S$. The largest eigenvalues correspond to directions which most differentiate S and R. The eigenvectors are also called *spatial filters*.

In Cohen (2017), GED is notably applied to estimate PAC. First, a low-frequency band is chosen. Then the covariance matrix S is computed on the filtered time series, and the covariance matrix R on the unfiltered time series. The GED is used on S and R to find a spatial filter corresponding to the low-frequency component. The spatial filter is the eigenvector corresponding to the highest eigenvalue. It is then used to create the low-frequency virtual channel.

To create the trough-modulated virtual channel, the troughs of the low-frequency virtual channel are extracted. Then, the covariance matrix S is computed on the time points surrounding these troughs, and the covariance matrix R on the entire time series. The GED is used on S and R to find a spatial filter corresponding to the trough-modulated component. Figure 4.6 presents a graphical overview of the creation of the trough-modulated virtual channel. Indeed, it selects the spatial filter corresponding to the highest variance in peri-trough data, compared to the entire dataset. The two obtained virtual channels can then be used in any PAC metric, or can be modeled with DAR models, to quantify the PAC.

One can note that the second spatial filter, corresponding to the trough-modulated component, is extracted based on the variance of the signal. However, a spectral modulation is not always associated with a broadband energy modulation. To solve this weakness, we build upon the GED framework to estimate spatial filters using DAR models

Using DAR models also remove the need to assume that the spectral modulation happens between peaks and troughs. Indeed, we discussed in Subsection 3.1.2 that the preferred phase, *i.e.* the phase corresponding to the largest amplitude of the modulated bands, might be different from 0 or π. DAR models make no assumption in this matter, and are able to estimate PAC with any preferred phase.

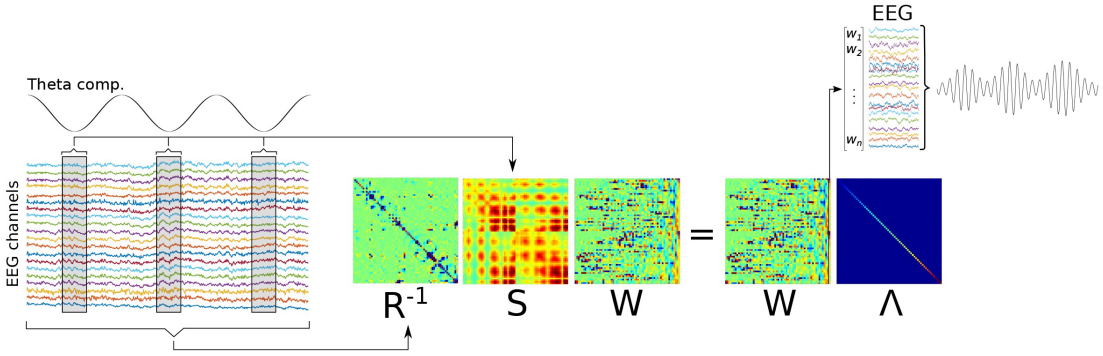


Figure 4.6 – Graphical overview of the method to create a trough-modulated virtual channel. A low-frequency component is identified ('Theta comp.'), and covariance matrices are computed on the basis of the multichannel data surrounding each trough, and using all data (respectively, S and R matrices). A GED of these matrices provides a set of eigenvectors (matrix W). The eigenvector with the largest eigenvalue (diagonal of matrix Λ) is used as weights to combine data from all channels linearly, which produces the component that best differentiates trough-related from non-trough-related activity. Reproduced from Cohen (2017).

4.2.2 Multivariate PAC with DAR models

Model definition Similarly to Cohen (2017), our approach forms two virtual channels on which we apply the DAR model. The virtual channels are composed of weighted sums of C signals, corresponding to the C channels.

$$y = \sum_{c=1}^{C} w_c y_c \qquad \text{and} \qquad x = \sum_{c=1}^{C} v_c x_c, \tag{4.10}$$

where x, y, x_c and y_c are univariate signals, and w_c and v_c are scalars. This formulation is based on the assumption that the source signal is instantaneously and identically spread over all sensors, with different gains expressed with the weights w_c and v_c. This assumption is particularly relevant for MEG multivariate recordings (Hari and Puce, 2017). Note that we estimate two different filters for the low frequency driver x and the broad-band signal y.

A naive idea would be to to maximize the likelihood of the model on y, but this idea might lead the weights to select the uncorrelated part of each channel. Indeed, selecting weights that cancel a strong source signal and keep only small uncorrelated noise would lead to a very low variance, and thus a very high likelihood in comparison with a filter selecting the strong source signal and removing the noise. To tackle this issue, we maximize instead the likelihood ratio between the DAR model and a linear AR model:

$$J = \log(L) - \log(\tilde{L}) \ , \tag{4.11}$$

where the tilde $\tilde{}$ indicates that it is related to the AR model, and the absence of tilde indicates that it is related to the DAR model. We have:

$$-2\log(L) = T\log(2\pi) + \sum_{t=p+1}^{T} \frac{\varepsilon(t)^2}{\sigma(t)^2} + 2\sum_{t=p+1}^{T} \log(\sigma(t)) \tag{4.12}$$

$$-2\log(\tilde{L}) = T\log(2\pi) + \frac{1}{\tilde{\sigma}^2}\sum_{t=p+1}^{T} \tilde{\varepsilon}(t)^2 + 2(T-p)\log(\tilde{\sigma}) \ . \tag{4.13}$$

Maximizing the likelihood ratio corresponds to maximizing the variance explained by the DAR model which was not explained by the AR model. Thus, the likelihood of the AR model is used as a normalization, to avoid constructing a low variance virtual channel.

4.2.3 Model estimation

We propose to estimate this model iteratively using three steps. First, we estimate the DAR model as in Chapter 2. Then we estimate the driver spatial filter v as in Section 4.1. Indeed, the spatial filter formulation of (4.10) is identical to the formulation of (4.1), so we can directly reuse the same gradient descent scheme. Finally, we estimate the signal spatial filter w with a dedicated procedure described below. The three steps are repeated multiple times before convergence.

Signal spatial filter estimation We estimate the signal spatial filter w by maximizing the log-likelihood difference (4.11):

$$J = K'' + \frac{1}{2}\sum_{t=p+1}^{T}\frac{\tilde{\varepsilon}(t)^2}{\tilde{\sigma}^2} - \frac{1}{2}\sum_{t=p+1}^{T}\frac{\varepsilon(t)^2}{\sigma(t)^2}\ , \tag{4.14}$$

where K'' is constant. Note that the AR and DAR models are here fixed. In particular, σ and $\tilde{\sigma}$ are constant and do not depend on the filter w. Using the linearity of AR models, we can write:

$$\begin{aligned}\varepsilon(t) &= y(t) + \sum_{i=1}^{p} a_i(t)y(t-i) &(4.15)\\ &= \sum_{c=1}^{C} w_c\left(y_c(t) + \sum_{i=1}^{p} a_i(t)y_c(t-i)\right) &(4.16)\\ &= \sum_{c=1}^{C} w_c\varepsilon_c(t). &(4.17)\end{aligned}$$

Using this expression in (4.14), we obtain:

$$\sum_{t=p+1}^{T}\frac{\varepsilon(t)^2}{\sigma(t)^2} = \sum_{c=1}^{C}\sum_{c'=1}^{C} w_c w_{c'}\sum_{t=p+1}^{T}\frac{\varepsilon_c(t)\varepsilon_{c'}(t)}{\sigma(t)^2} = w^\top R w, \tag{4.18}$$

where $R \in \mathbb{R}^{C\times C}$ is defined by:

$$R_{c,c'} = \sum_{t=p+1}^{T}\frac{\varepsilon_c(t)\varepsilon_{c'}(t)}{\sigma(t)^2}. \tag{4.19}$$

If we also define $S \in \mathbb{R}^{C\times C}$ in a similar way:

$$S_{c,c'} = \frac{1}{\tilde{\sigma}^2}\sum_{t=p+1}^{T}\tilde{\varepsilon}_c(t)\tilde{\varepsilon}_{c'}(t), \tag{4.20}$$

then we can rewrite the log-likelihood difference:

$$J = K'' + w^\top S w - w^\top R w \tag{4.21}$$

Algorithm 4.1: Estimation of virtual channels to model PAC on multivariate sig.

Input : Multivariate raw signal $z \in \mathbb{R}^{C \times T}$, temporal filter h, orders (p, m)
Extract signals y and drivers x from z, using filter h as in Subsection 3.1.1,
Compute covariance matrices R and S computed respectively on x and y,
Initialize driver spatial filter v with GED on R and S,
Initialize signal spatial filter w with Gaussian white noise,
for $n = 1$ **to** N **do**
  Estimate DAR (p, m) model coefficients (A, B) as in Algorithm 2.1,
  Estimate AR $(p, 0)$ model coefficients $(\tilde{A}, \tilde{B})$ as in Algorithm 2.1,
  Compute the corresponding multivariate residuals ε and $\tilde{\varepsilon}$,
  Compute covariance matrices R and S computed respectively on ε and $\tilde{\varepsilon}$,
  Estimate the signal spatial filter w with GED on R and S,
  Estimate the driver spatial filter v with gradient descent, as in Section 4.1,
return v, w, A, B

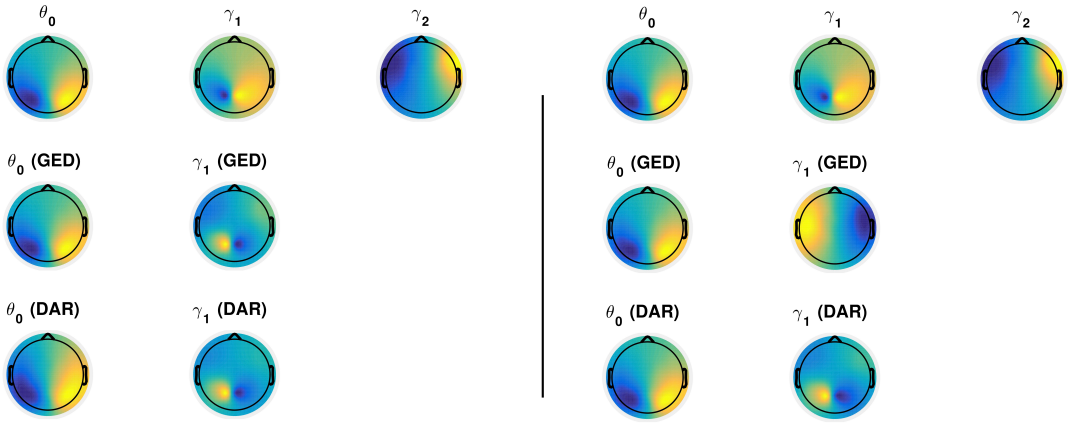
Therefore, the solution w of our problem $\text{argmax}_w J(w)$ is an eigenvector of the generalized eigen-decomposition of S and R. More precisely, the log-likelihood difference J is maximum for the largest eigenvalue (Fukunaga, 2013). The entire algorithm is detailed in Algorithm 4.1.

We see that the estimation through GED makes our approach very close to the method of Cohen (2017). The difference is in the covariance matrices used. In Cohen (2017), the covariance matrices are typically computed on the raw signal, using the peri-peak and peri-trough time points. In our case, the covariance matrices are computed on the residuals $\varepsilon(t)$ and $\tilde{\varepsilon}(t)$ using all the time points. Therefore, the method of Cohen (2017) assumes a variance difference between the peaks and the troughs of the slow oscillation, whereas our method assumes a variance difference between an AR model and a DAR model.

Driver's filter In Section 4.1, we demonstrated how to estimate the driver based on a set of potential drivers. We used it notably for estimating a *temporal* filter h which was used to extract the driver from the low-pass original time-series z_x. We used $x = h * z_x = \sum_{n=1}^{N} h_n z_x(t-n)$. In that case, the potential drivers were the delayed signals $x_n(t) = z_x(t-n)$.

In this section, we reuse this method, but this time to estimate a *spatial* filter, to extract the driver as a virtual channel. In this case, the potential drivers are the band-pass filtered signals on each channel $x_c(t) = h * z_c(t)$, where z_c is the original signal of channel c. We use $x = \sum_{c=1}^{C} v_c(h * z_c(t))$. Note that we assume here that the temporal filter h is given. To estimate h, we resort to a grid-search strategy over a bank of parametric bandpass filters.

Model initialization Our model is non-convex and is thus sensitive to the initialization. Different initializations may lead to different local minima. In particular, the initial spatial weights need to lead to virtual channels with PAC. Indeed, if the first DAR model fail to capture some PAC, the subsequent spatial weights refinements will not improve the model.



(a) Preferred phase $\phi_0 = 0$.

(b) Preferred phase $\phi_0 = \pi/2$.

Figure 4.7 – Spatial filters associated with the three dipole sources θ_0, γ_1 and γ_2: (top) simulated filters, (middle) filters estimated with GED, (bottom) filters estimated with DAR models. With a preferred phase $\phi_0 = 0$, both methods correctly estimate the spatial filters, without being distracted by the strong γ_2 dipole nearby. With a preferred phase $\phi_0 = \pi/2$, the GED method fails to estimate γ_1 contrary to the method using DAR models.

To tackle this issue, a first possibility is to use different random weights initializations, and to select the results with the highest log-likelihood difference J. Another possibility is to use the method of Cohen (2017) as an initialization, and use our method as a refinement step instead of simply using DAR models on the two initial virtual channels.

In our experiments, we saw that using random weights for the initial driver spatial filter v was not robust. Therefore, we used GED to estimate this initial spatial filter. Note that this makes no assumption on the coupling, except about the temporal filter h used to bandpass the drivers on each channel.

To initialize the spatial filter w, we prefer not using the GED as in Cohen (2017), since it makes an assumption on the preferred phase of the amplitude modulation, and since it supposes a modulation of the variance of the signal. On the contrary, our method based on DAR models makes no assumption on the preferred phase, and supposes a modulation of the PSD. Therefore, we initialize the spatial filter w with Gaussian white noise. We noted on our experiments that the filter w was well estimated even with a poor initial filter, provided that the initial driver spatial filter v was good.

4.2.4 Experiments

To validate our approach, we reused the simulation settings described in Cohen (2017, Method 1), using their MATLAB code available online. The method first simulates three source signals: a theta rhythm θ_0 at 6 Hz, with small amplitude and frequency fluctuations, a gamma rhythm γ_1 at 40 Hz modulated in amplitude by the theta rhythm, and a gamma rhythm γ_2 at 50 Hz modulated in amplitude by an independent signal. The third signal was serving as "distractor", and had twice the average amplitude of the second signal. Each signal was associated with a dipole localization, and projected on 64 MEG sensors. A pink noise, *i.e.* noise with a $1/f$ PSD, was added to all sensors.

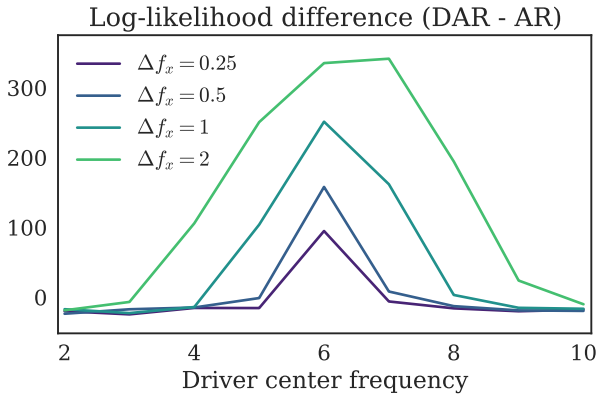


Figure 4.8 – Log-likelihood difference J over a grid of parameters f_x and Δf_x, computed on left-out data using cross-validation.

Then we compared the spatial filters obtained with the GED method of Cohen (2017), and with our method based on DAR models. For DAR models, we used a grid-search over the driver filter's center frequency f_x and bandwidth Δf_x, with $f_x \in [\![2, 10]\!]$ and $\Delta f_x \in \{0.25, 0.5, 1, 2\}$, and selected automatically the best parameters based on the log-likelihood difference computed over cross-validation (see Figure 4.8). For the GED method, we assumed that the ground-truth parameters were known. We can see in Figure 4.7 that the spatial filters are correctly estimated by both methods, with a preferred phase $\phi_0 = 0$, *i.e.* when the amplitude modulation is maximal at the troughs. With a preferred phase $\phi_0 = \pi/2$, the GED method fails to estimate the filter associated with γ_1, since it assumes that the modulation happens at the troughs. The method based on DAR models correctly estimate this filter since it is invariant with the preferred phase.

Additional experiments are required to compare in depth the two methods. Note however that the DAR approach is slower than the GED method, but brings several advantages in term of model selection, invariance to the preferred phase, and possibility to detect frequency modulation without overall variance modulation.

4.3 Spectro-temporal receptive fields

4.3.1 Spectro-temporal receptive fields

In cognitive neuroscience, predictive modeling comes with two flavors, *encoding* models and *decoding* models. Encoding models consider stimulus features to predict brain activity, while decoding models consider brain activity features to predict the stimulus. A nice introduction to both approaches is available in (Holdgraf et al., 2017).

Decoding models have grown popularity in natural image reconstruction from fMRI data (Kay et al., 2008), or in speech reconstruction from ECoG data (Brumberg et al., 2010, Pasley et al., 2012). Other decoding models aim to classify the stimuli instead of reconstructing them. Some successes have for instance been observed in classifying phonemes (Chang et al., 2010), words (Kellis et al., 2010), or semantic information (Wang et al., 2011) from ECoG recordings. Encoding models have also had remarkable successes in predicting fMRI features from visual stimuli (Naselaris et al., 2011), and in

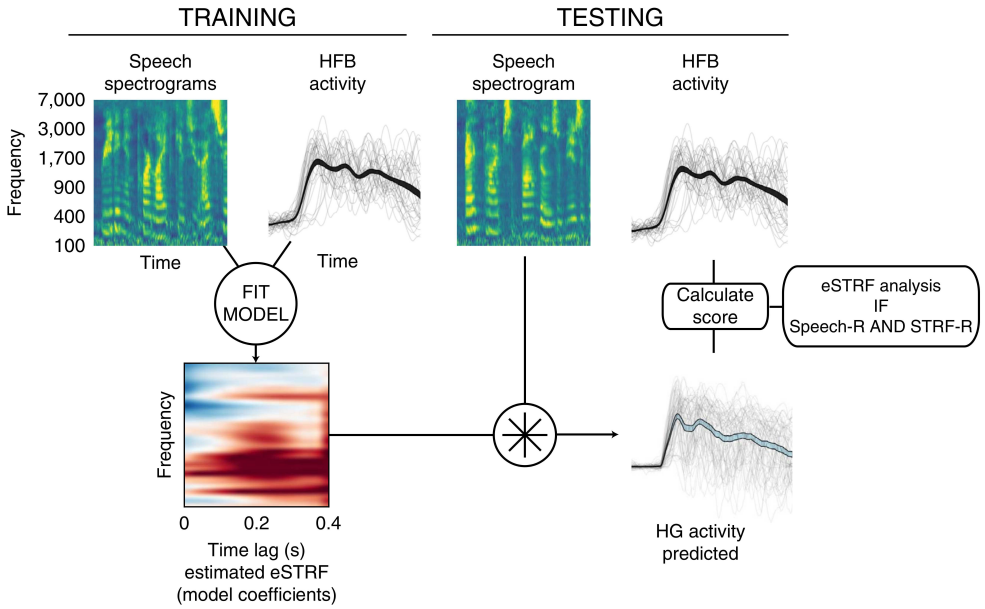


Figure 4.9 – Example of model fitting procedure. Auditory spectrograms of sound and evoked high-frequency broadband (HFB) activity (top, first/second columns) is used to fit a linear regression model, resulting in a set of model coefficients (STRF, lower left). This STRF is convolved with a held-out auditory spectrogram (top, third column) to generated a predicted HFB activity trace (lower right). The goodness of fit (cross-validated R2) is calculated between the predicted response and the actual HFB activity in the held-out trial (top, fourth column). Reproduced from (Holdgraf et al., 2016).

predicting ECoG features from audio stimuli (Mesgarani et al., 2014).

In this study, we consider encoding models of ECoG signals, recorded jointly with audio stimuli. The typical approach starts by building a spectrogram, which decomposes the audio stimuli into a time-frequency representation. In practice, a number (*e.g.* 128) of linear filters are applied on the signal to build this spectrogram. The center frequencies of the filters are usually logarithmically spaced, to mimic the human auditory system (Chi et al., 2005). Then, the spectrogram is concatenated multiple times with a range of time lags, to account for non-instantaneous effects. These features are then used in linear regression models to predict the gamma band envelop of ECoG signals (Holdgraf et al., 2016).

An interesting outcome of such modeling is the set of coefficients of the fitted linear regression models. Indeed, each spectro-temporal audio feature is weighted by the model with respect to its predictive power. The coefficients thus form a so-called *spectro-temporal receptive field* (STRF), which maps the neural response to the relevant acoustic features. STRFs has been used to track which brain region responds to which acoustic feature (Theunissen et al., 2000, Depireux et al., 2001), or to describe tuning shifts when comparing different stimulus conditions (Holdgraf et al., 2016). Figure 4.9 presents a typical STRF pipeline, reproduced from (Holdgraf et al., 2016).

Linear models are used in encoding approaches for their simplicity and their interpretability. However, such analysis is limited to predicting a slowly varying time-series such as the gamma rhythm envelop. A different approach would be to model the entire

ECoG signals, without using any filtering or envelop extraction. Reformulating the problem, we would like to model a neurophysiological time-series which may contain amplitude modulations related to a known driving stimulus. We see that DAR models are natural candidates for such modeling approach.

4.3.2 STRF and DAR models

To tackle this encoding problem, we simply use DAR models on the ECoG signal y. Different audio features can be use as the driver signal x.

A first natural candidate for the driver x is the raw audio signal itself z. However, the sampling frequency of the audio signal (*e.g.* 48 kHz) is much higher than the sampling frequency of the ECoG signal (*e.g.* 1 kHz). To be able to use in a DAR model, we need to have the same sampling frequency. We cannot simply low-pass filter and down-sample the audio signal, since we would lose a large part of the audio information, contained above 1 kHz. A better driver candidate is the envelop of the audio signal. Indeed, the envelop does not contain much information in the high frequencies, and can thus safely be low-pass filter and down-sampled.

Another approach is to avoid defining arbitrarily a driver signal, but to estimate it from a set of potential drivers, as described in Section 4.1. The driver x is then composed of a weighted sum of potential drivers x_n:

$$x(t) = \sum_{n=1}^{N} \alpha_n x_n(t). \tag{4.22}$$

A natural set consists in multiple band-pass filtered versions $x_n = w_n * z$ of the audio signal z, where w_n is a band-pass filter. Each potential driver can also be present multiple times in the set, with different time lags τ. The driver reads:

$$x(t) = \sum_{n=1}^{N} \sum_{\tau=0}^{\tau_{\max}} \alpha_{n,\tau} w_n * z(t-\tau). \tag{4.23}$$

This leads to a very strong parallel with the STRF described earlier. Indeed, the obtained weights $\alpha_{n,\tau}$ form a spectro-temporal representation of relevant audio features to drive the DAR model on the ECoG signal. Importantly, we model all the frequencies at the same time, without any filtering or envelop extraction step.

For simplicity, we will note the weights α_n in the rest of this study.

4.3.3 Experiments

We present here a number of exploratory experiments, using data from Holdgraf et al. (2016). The data consists in a grid of 32 ECoG channels, localized in the vicinity of the auditory cortex, recorded simultaneously with the audio stimulus. The audio stimulus was a succession of speech samples, which last a few seconds each, interlaced with a few seconds of silence.

In the first experiment, we applied univariate DAR models separately on each ECoG channel, using the audio envelop as the driver x. More specifically, we low-pass filtered (at 10 Hz) the absolute value of the audio signal, and used the Hilbert transform to obtain a complex-valued driver. For simplicity, we fixed the parameters to $p = 10$, $m = 2$, and $\tau = 0$, and compared with cross-validation the performances of DAR models

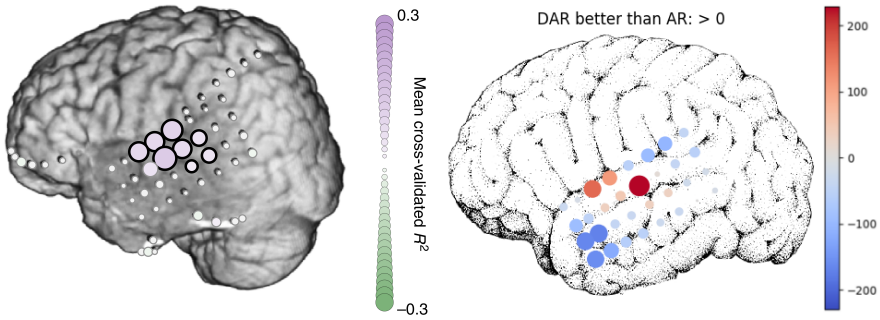


Figure 4.10 – Cross-validated goodness of fit on each channel of the ECoG grid. (Left) R^2 score of the linear regression model, trained on the delayed audio spectrogram to predict the gamma rhythm envelop (reproduced from Holdgraf et al. (2016)). (Right) Log-likelihood difference of DAR models versus AR models, trained on the raw ECoG channels using the audio envelop as the driver. Both methods consistently select the channels located on the auditory cortex.

versus AR models. The log-likelihood difference on out-of-sample data is presented in Figure 4.10 (right). We see that the log-likelihood difference is positive only on the channels located on the auditory cortex. A negative log-likelihood difference is observed on the other channels, showing that DAR models are over-fitting compared to AR models. For comparison, we also show in Figure 4.10 (left) the cross-validated R^2 score of the linear regression model, trained on the delayed spectrogram to predict the gamma rhythm envelop, as reported in Holdgraf et al. (2016).

Then we used a grid-search on parameters (p, m, τ), using cross-validation to select the best parameters. Figure 4.13 presents the results on all 32 channels, where we plot the log-likelihood difference between DAR and AR models. The horizontal axis corresponds to the delay parameter τ, each line corresponds to a parameter $m \in \{0, 1\}$, and we only show the results for the parameters p leading to the highest score. The auditory channels (3, 4, 5, 11, 12, 13, 18, 19) lead to a strong performance of DAR models versus AR models, which means that they do contain a spectral modulation correlated with the speech stimulus driving signal.

We then focused the investigation on channels with a strong log-likelihood difference, *e.g.* channel number 5. On this channel, we compared the log-likelihood of DAR models using different drivers, with a cross-validation scheme. The different drivers were: the speech envelop; a binary signal separating speech from silence; a weighted driver estimated over a set of potential drivers from the delayed spectrogram. The results are presented in Figure 4.11, where we show the modulated PSD with respect to the drivers, with the log-likelihood score in the title (with in brackets the standard deviation over cross-validation folds).

The first observation concerns the obtained PSD modulation. In this modulation, we can see a low frequency decrease and a high frequency increase in the ECoG signal in the presence of speech. This effect has already been reported, for instance by Pasley et al. (2012). Note that we did not use filtering on the ECoG signal to create this figure; the PSD modulation is directly derived from the fitted DAR model.

The second observation concerns the similarity of the results. Indeed, all three drivers lead to extremely similar PSD modulation, and the log-likelihood scores is not significantly different. In particular, the binary driver performs slightly better than the speech

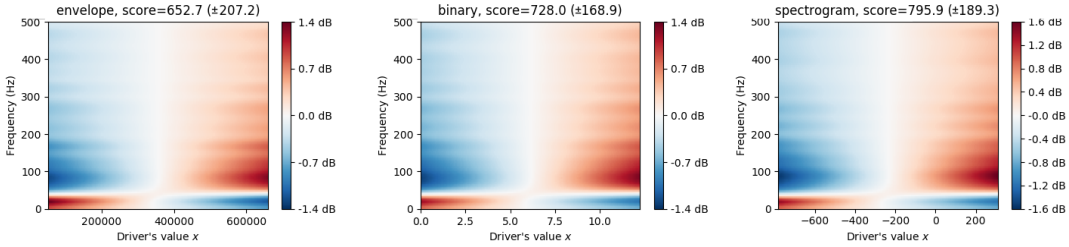


Figure 4.11 – Power spectral density (PSD) modulation with respect to the driver. (Left) Speech envelop driver, (middle) binary speech driver, (right) weighted driver over a spectrogram (see Figure 4.12). The log-likelihood score is shown in the title, with in brackets the standard deviation over cross-validation folds. Note that all drivers perform equally well, leading to extremely similar PSD modulations.

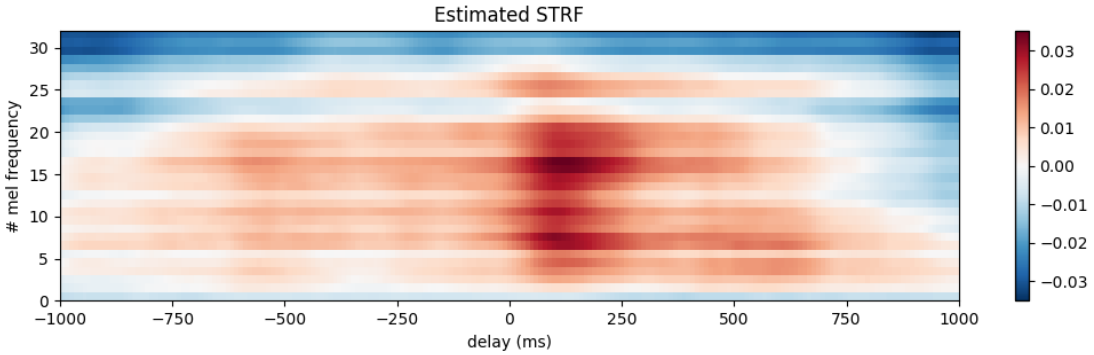


Figure 4.12 – Spectro-temporal receptive field (STRF) estimated with a DAR model. Note that this figure was cherry-picked over multiple restarts and optimization parameters.

envelop, which means that the extracted effect does not use much information in the speech intensity. The driver obtained from a weighted sum over the spectrogram is not significantly better, despite a much heavier optimization procedure. This might show the limit of our approach, showing that DAR models only focus on the main PSD modulation, and that more subtle effects are difficult to capture.

Figure 4.12 shows the weights estimated to create the third driver, thus creating a DAR-estimated STRF. Note that this figure was cherry-picked over multiple restarts and optimization parameters, since the estimating procedure is still quite unstable and non-convex.

Estimating STRF reliably through DAR models still needs improvement to be on par with the simplicity and robustness of linear regression model. Indeed, the high number of parameters for the driver makes the non-convex estimation not very reliable, in addition to the higher computational cost. A possible improvement could be found using some sort of regularization. However, DAR models perfectly fit in this encoding modeling approach, and we believe that they provide an excellent framework for legitimate and reproducible studies.

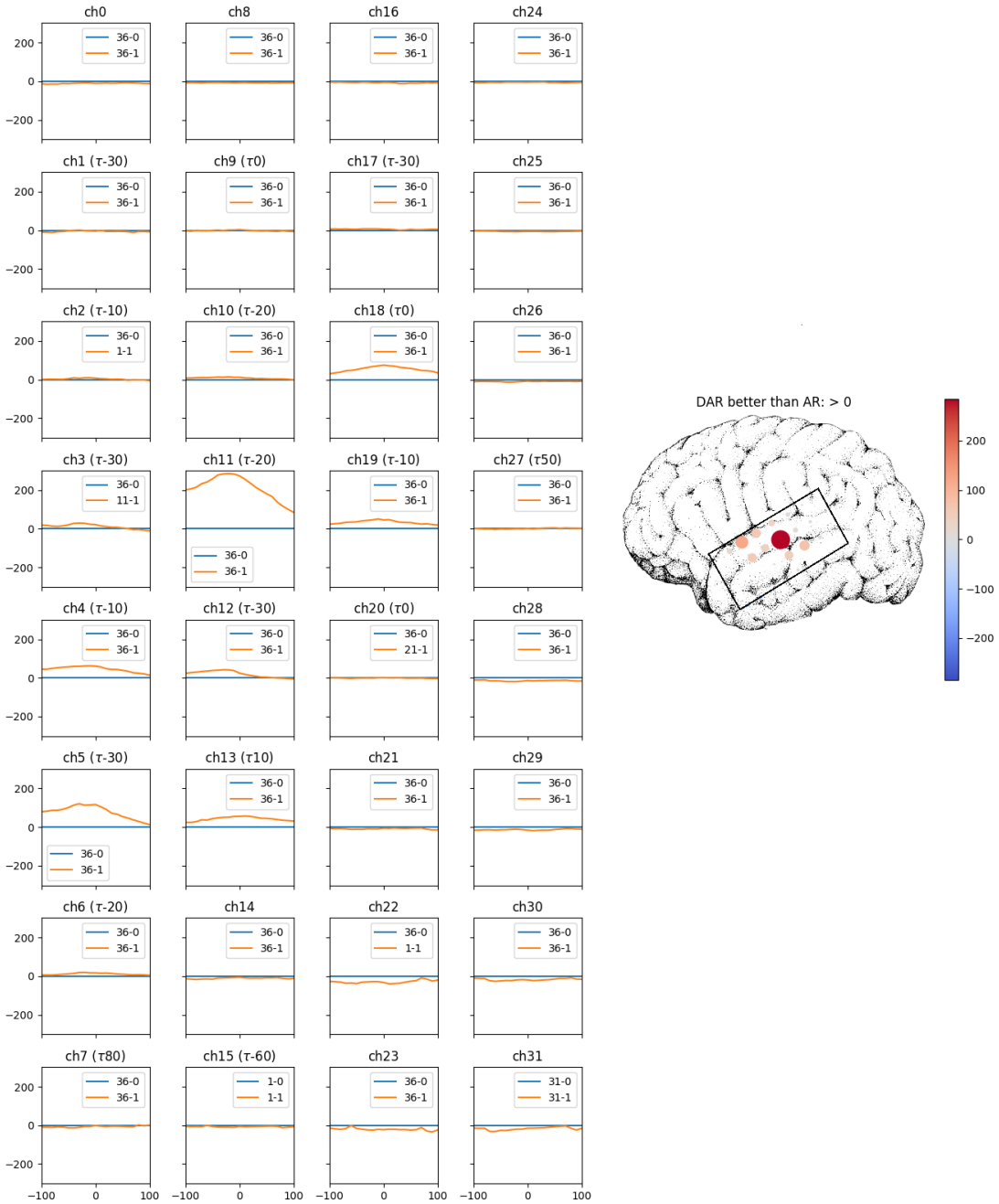


Figure 4.13 – Log-likelihood difference between DAR models and AR models, over a grid-search of parameters. The horizontal axis corresponds to the delay parameter τ, each line corresponds to a parameter $m \in \{0, 1\}$, and we only show the results for the parameter p leading to the highest score. The channel layout is displayed on the right panel.

5

Convolutional sparse coding

"Given the pace of technology, I propose we leave math to the machines and go play outside."

– Bill Watterson

Contents

5.1 Convolutional sparse coding . . . 94
5.1.1 Convolutional sparse coding model . . . 94
5.1.2 Model estimation: Z-step . . . 95
5.1.3 Model estimation: D-step . . . 96
5.1.4 Model initialization . . . 98
5.1.5 Experiments . . . 99
5.2 CSC with alpha-stable distributions . . . 103
5.2.1 Alpha-stable distributions . . . 104
5.2.2 Alpha-stable CSC . . . 105
5.2.3 Model estimation: maximum a posteriori (MAP) inference . . . 106
5.2.4 Model estimation: E-Step . . . 106
5.2.5 Model estimation: M-Step . . . 108
5.2.6 Model initialization . . . 108
5.2.7 Experiments . . . 109
5.3 Multivariate CSC with a rank-1 constraint . . . 110
5.3.1 Model definitions . . . 110
5.3.2 Model estimation: Z-step . . . 111
5.3.3 Model estimation: D-step . . . 113
5.3.4 Model initialization . . . 116
5.3.5 Experiments . . . 116

In this chapter, we address the present need in the neuroscience community to better capture the complex morphology of brain waves. Our approach is based on convolutional sparse coding (CSC) models, which are dictionary learning models using shift-invariant representations and strong sparsity assumptions.

We propose efficient optimization schemes leading to state-of-the-art performances. We then extend CSC models to cope with brain recordings challenges, such as severe artifacts, low signal-to-noise ratio, and long multivariate signals. These extensions are critical to be able to use CSC models on brain recordings.

The source code of all the methods described in this chapter is publicly available at https://alphacsc.github.io/, with documentation, tests, and multiple examples.

This chapter covers the following publications:

- Jas, M., Dupré la Tour, T., Şimşekli, U., and Gramfort, A. (2017). Learning the morphology of brain signals using alpha-stable convolutional sparse coding. In *Advances in Neural Information Processing Systems 30 (NIPS)*, pages 1099–1108
- Dupré la Tour, T., Moreau, T., Jas, M., and Gramfort, A. (2018b). Multivariate convolutional sparse coding for electromagnetic brain signals. In *Advances in Neural Information Processing Systems (NIPS)*

5.1 Convolutional sparse coding

5.1.1 Convolutional sparse coding model

Convolutional sparse coding (CSC) is a mathematically principled formulation of dictionary learning with shift-invariant sparse representations. It was introduced by Grosse et al. (2007), and consists in minimizing the following expression:

$$\begin{aligned} \underset{\{d_k\},\{z_k^n\}}{\operatorname{argmin}} \sum_{n=1}^{N} \frac{1}{2} \left\| x^n - \sum_{k=1}^{K} z_k^n * d_k \right\|_2^2 + \lambda \sum_{k=1}^{K} \|z_k^n\|_1 \ , \\ \text{s.t.} \quad \|d_k\|_2^2 \leq 1, \quad \forall k, \end{aligned} \tag{5.1}$$

where $\{x^n\}_{n=1}^N \subset \mathbb{R}^T$ are N observed signals of length T, $\{d_k\}_{k=1}^K \subset \mathbb{R}^L$ are the K temporal atoms of length L we aim to learn, $\{z_k^n\}_{k=1}^K \subset \mathbb{R}^{T-L+1}$ are K signals of activations (a.k.a. the code) associated with x^n, and $\lambda > 0$ is the regularization parameter.

The objective function (5.1) has two terms, an ℓ_2 data fitting term that corresponds to assuming an additive Gaussian noise model, and a regularization term that promotes sparsity with an ℓ_1 norm (Tibshirani, 1996). The regularization parameter is called $\lambda > 0$. The model enforces that d_k lies within the unit sphere, which prevents the scale ambiguity between d_k and z_k^n. Without this constraint, any solution (d_k, z_k^n) could be improved using $(\gamma d_k, \gamma^{-1} z_k^n)$ for any scalar $\gamma > 1$, which gives smaller regularization cost for the same data fitting term.

In this work, we also assume that the entries of z_k^n are positive, which means that the temporal patterns are present each time with the same polarity. This positivity constraint is not present in the original CSC model (Grosse et al., 2007). Our model reads:

$$\begin{aligned} \underset{\{d_k\},\{z_k^n\}}{\operatorname{argmin}} \sum_{n=1}^{N} \frac{1}{2} \left\| x^n - \sum_{k=1}^{K} z_k^n * d_k \right\|_2^2 + \lambda \sum_{k=1}^{K} \|z_k^n\|_1 \ , \\ \text{s.t.} \quad \|d_k\|_2^2 \leq 1, \text{ and } z_k^n \geq 0, \quad \forall k, n. \end{aligned} \tag{5.2}$$

General estimation scheme The model (5.2) is not jointly convex in d_k and z_k^n, which means that multiple local minima may exist, and that we cannot be sure that a given minimum is a global minimum. Therefore, different initialization may lead to different local minima, which can be compared with the loss function defined in (5.2).

However, we can note that the model is convex in both block of variables $\{d_k\}$ and $\{z_k^n\}$. Such property naturally lead to a block-coordinate descent, where we alternatively fix one block of variables and decrease the loss function with updates on the second block (Grosse et al., 2007). We call *Z-step* the update of the activations $\{z_k^n\}$ while keeping the atoms fixed, and *D-step* the update of the atoms $\{d_k\}$ while keeping the activations fixed.

5.1.2 Model estimation: Z-step

Given a set $\{d_k\}$ of K fixed atoms and a regularization parameter $\lambda > 0$, the Z-step aims to retrieve the NK activation signals $z_k^n \in \mathbb{R}_+^{\widetilde{T}}$ associated to the signals $x^n \in \mathbb{R}^{P \times T}$ by solving the following ℓ_1-regularized optimization problem:

$$\begin{aligned} \underset{\{z_k^n\}}{\operatorname{argmin}} & \sum_{n=1}^{N} \frac{1}{2} \left\| x^n - \sum_{k=1}^{K} z_k^n * d_k \right\|_2^2 + \lambda \sum_{k=1}^{K} \|z_k^n\|_1 \ , \\ \text{s.t.} \quad & z_k^n \geq 0, \quad \forall k, n. \end{aligned} \tag{5.3}$$

This problem is jointly convex in all the activation signals z_k^n and can thus be efficiently solved. In Chalasani et al. (2013), the authors proposed an algorithm based on FISTA (Beck and Teboulle, 2009) to solve it. Bristow et al. (2013) introduced a method based on ADMM (Boyd et al., 2011) to compute efficiently the activation signals z_k^n. These two methods are detailed and compared by Wohlberg (2016b), which also used the fast Fourier transform (FFT) to accelerate the computations. Finally, Kavukcuoglu et al. (2010) adapted the greedy coordinate descent (GCD) to solve this convolutional sparse coding problem.

In this work, we propose two different approaches to solve this problem. First, we make use of L-BFGS (Byrd et al., 1995) to improve on first order methods, which leads to state of the art computational speed. We also propose to use locally greedy coordinate descent (LGCD) (Moreau et al., 2018), to alleviate linear dependence in the signal length T. It makes efficient use of the activations sparsity to greatly reduce the computational cost. This method is described in Subsection 5.3.2, as it is designed for our multivariate CSC model, but it can also be applied on univariate CSC.

Note that problem (5.3) is independent for each signal x^n. The computation of each z^n can thus be parallelized, independently of the technique selected to solve the optimization. Therefore, we omit the superscript n in this subsection to simplify the notation.

L-BFGS-B The L-BFGS algorithm (Byrd et al., 1995) is a quasi-Newton method which estimates an approximation of the Hessian using gradient computations in order to speed up the iterative descent algorithm. It however requires the problem to be differentiable.

The first term in (5.3) is easily differentiable. Indeed, we can note that the convolution operator is linear, so we can rewrite $z_k * d_k = D_k z_k$, where D_k is a large Toeplitz matrix that we never construct in practice. This notation is just a way to understand this problem as a linear problem.

The second term in (5.3) is an ℓ_1 norm, which is not differentiable in zero. This non-differentiability is usually tackled with proximal algorithms like FISTA (Beck and Teboulle, 2009). However, as we added the additional positivity constraint $z_k \geq 0$, we can consider the ℓ_1 norm as differentiable on the constraint set $[0, \infty[$. We can

Algorithm 5.1: Calling L-BFGS-B for the Z-step

Input	: Signal x, atoms $\{d_k\}$, initialization $\{z_k\}$
Define `func`	: Given $\{z_k\}$, return the loss function, computed with (5.3)
Define `grad`	: Given $\{z_k\}$, return the gradient ∇_{z_k} for all k, computed with (5.4)
Define B	: all variables $z_k[t]$ are constrained in the box $[0, \infty[$
Run	: `scipy.optimize.fmin_l_bfgs_b` (`func`, $\{z_k\}$, `grad`, B)

thus use the L-BFGS-B algorithm, *i.e.* the L-BFGS algorithm with a box constraint: $0 \leq z_k[t] < \infty$.

We only need to compute the gradient ∇_{z_k} of the objective function with respect to z_k:

$$\nabla_{z_k} = d_k^{\tilde{}} * \left(x - \sum_{l=1}^{K} z_l * d_l \right) + \lambda \sum_{l=1}^{K} z_l \ . \tag{5.4}$$

where the left convolution is performed on the "valid" set to obtain the correct dimension of the gradient $\nabla_{z_k} \in \mathbb{R}^{\widetilde{T}}$. Using the L-BFGS algorithm, we are able to solve the Z-step problem with state-of-the-art computational speed, as we will show in the experiments Subsection 5.1.5. In our experiments, we used the implementation of L-BFGS-B available in SciPy (Jones et al., 2001), and we give more details on how to call this function in Algorithm 5.1.

5.1.3 Model estimation: D-step

Given KN *fixed* activation signals $z_k^n \in \mathbb{R}^{\widetilde{T}}$, associated to signals $x^n \in \mathbb{R}^T$, the D-step aims to update the K temporal patterns $d_k \in \mathbb{R}^L$, by solving:

$$\begin{aligned} \underset{\{d_k\}}{\operatorname{argmin}} \sum_{n=1}^{N} \frac{1}{2} \left\| x^n - \sum_{k=1}^{K} z_k^n * d_k \right\|_2^2 , \\ \text{s.t.} \quad \|d_k\|_2^2 \leq 1, \quad \forall k. \end{aligned} \tag{5.5}$$

This optimization problem turns out to be a constrained least-squares problem. This problem can be solved either in the time domain or in the Fourier domain (Grosse et al., 2007, Heide et al., 2015, Wohlberg, 2016b). The Fourier transform simplifies the convolutions that appear in least-squares problem, but it also induces several difficulties, such as that the atoms d_k have to be in a finite support L, an important issue ignored in the seminal work of (Grosse et al., 2007) and addressed with an ADMM solver in (Heide et al., 2015, Wohlberg, 2016b). Interestingly, the dual problem is also a smooth constraint problem yet with a simpler positivity box constraint. We thus propose to optimize the dual problem with L-BFGS-B.

In our multivariate CSC model described in Subsection 5.3.2, we also propose a projected gradient descent (PGD) method, since using L-BFGS on the dual problem is not possible. The PGD method can also be used on univariate CSC, yet during our experiments, we found that using the quasi-Newton L-BFGS solver turned out to be more efficient than any accelerated first order method in either the primal or the dual.

L-BFGS-B on the dual problem As when solving for the activations z_k, we can rewrite the convolution and the summation over the atoms as a large linear operation $Z^n d$, where d is the concatenation of all d_k. It leads to the simpler formula:

$$\underset{d}{\operatorname{argmin}} \sum_{n=1}^{N} \frac{1}{2}\|x^n - Z^n d\|_2^2, \quad \text{s.t. } \|d_k\|_2^2 \leq 1, \quad \forall k. \tag{5.6}$$

If not for the constraint, this is a classic least square problem which has a known closed-form solution:

$$d^* = \left(\sum_{n=1}^{N} Z^{n\top} Z^n\right)^{-1} \sum_{n=1}^{N} Z^{n\top} x^n \tag{5.7}$$

Given the constraint, we need to use iterative algorithm to approximate the solution. This problem can also be solved on the dual space, which is derived from the following Lagrangian:

$$g(d, \beta) = \sum_{n=1}^{N} \frac{1}{2}\|x^n - Z^n d\|_2^2 + \sum_{k=1}^{K} \beta_k(\|d_k\|_2^2 - 1) \quad \text{s.t. } \beta_k \geq 0, \quad \forall k, \tag{5.8}$$

where $\beta = (\beta_1, \beta_2, ..., \beta_K)$ are the dual variables. Therefore, the dual problem is:

$$\underset{d,\beta}{\operatorname{argmax}}\, g(d, \beta) = \underset{\beta}{\operatorname{argmax}}\, g(d^*(\beta), \beta) \tag{5.9}$$

where $d^*(\beta)$, the primal optimal, is given by:

$$d^*(\beta) = \left(\sum_{n=1}^{N} Z^{n\top} Z^n + \bar{\beta}\right)^{-1} \sum_{n=1}^{N} Z^{n\top} x^n \tag{5.10}$$

with $\bar{\beta} = \operatorname{diag}([\mathbb{1}\beta_1, \mathbb{1}\beta_2, ..., \mathbb{1}\beta_K]) \in \mathbb{R}^{KL}$, and $\mathbb{1} \in \mathbb{R}^L$. The gradient for the dual variable β_k is then given by:

$$\frac{\partial g(d^*, \beta)}{\partial \beta_k} = \|{d^*}_k\|_2^2 - 1, \tag{5.11}$$

with ${d^*}_k$ computed from (5.10). We can solve this iteratively using L-BFGS-B taking into account the positivity constraint $\beta_k \geq 0$ for all k. It amounts to computing the primal optimal at each step, then the dual gradients according to the updated primal, then updating the dual using the gradient and continuing this way until convergence. Here again, we used the implementation of L-BFGS-B available in SciPy (Jones et al., 2001), and we give more details on how to call this function in Algorithm 5.2.

Block coordinate descent (BCD) What we have described so far solves for all the atoms simultaneously. However, it is also possible to estimate the atoms sequentially one at a time using a block coordinate descent (BCD) approach, as in the work of (Mairal et al., 2010). In each iteration of the BCD algorithm, a residual r_k^n is computed as given by:

$$r_k^n = x^n - \sum_{k' \neq k} Z_{k'}^n d_{k'} \tag{5.12}$$

and corresponding subproblem (5.6) becomes:

$$\underset{d_k}{\operatorname{argmin}} \sum_{n=1}^{N} \frac{1}{2}\|r_k^n - Z_k^n d_k\|_2^2, \quad \text{s.t. } \|d_k\|_2^2 \leq 1, \quad \forall k. \tag{5.13}$$

Algorithm 5.2: Calling L-BFGS-B for the D-step

Input	: Signal x, activations $\{z_k\}$, initialization β
Define `func`	: Given β, compute the primal optimal $d^*(\beta)$ with (5.10) Return the Lagrangian $g(d^*(\beta), \beta)$, computed with (5.8)
Define `grad`	: Given β, compute the primal optimal $d^*(\beta)$ with (5.10) Return the gradient of the Lagrangian, computed with (5.11)
Define B	: All variables β_k are constrained in the box $[0, \infty[$
Run	: `scipy.optimize.fmin_l_bfgs_b` (`func`, β, `grad`, B)

Table 5.1 – Computational complexities of each step.

Step	Method	Computation	Computed	Complexity
Z-step	L-BFGS	Gradient evaluation	multiple times	$NKTL$
D-step	L-BFGS	Precomputation	once	NK^2TL^2
D-step	L-BFGS	Gradient evaluation	multiple time	K^3L^3
D-step	L-BFGS with BCD	Precomputation	once	$NKTL$
D-step	L-BFGS with BCD	Gradient evaluation	multiple time	KL^2

which is solved in the same way as subproblem (5.6).

In the simultaneous case, we construct one linear problem in $\mathcal{O}(NK^2TL^2)$ and one iteration costs $\mathcal{O}(L^3K^3)$, since we need to inverse a matrix in $\mathbb{R}^{KL\times KL}$. However, in the BCD strategy, we construct K linear problems in $\mathcal{O}(NL^2T)$ and one iteration costs only $\mathcal{O}(L^3)$, since we only need to inverse matrices in $\mathbb{R}^{L\times L}$. Actually, we can even use the fact that for one atom k, the matrix $\sum_{n=1}^{N}(Z_k^n)^T Z_k^n$ has a Toeplitz structure, *i.e.* it is constant along each diagonal. In this case, we can construct K linear problems in only $\mathcal{O}(NLT)$ and one iteration costs only $\mathcal{O}(L^2)$. Computational complexities are summarized in Table 5.1.

5.1.4 Model initialization

As the CSC problem is non-convex, different initializations might lead to different local optima. Therefore, the initialization is a critical part of the model estimation. As we start the optimization with a Z-step, the initialization only concerns the atoms, and the activations are simply initialized with zeros.

Chunk initialization The activations sub-problem (Z-step) is regularized with an ℓ_1-norm which induces sparsity: The higher the regularization parameter λ, the higher the sparsity. Therefore, there exists a value λ_{max} above which the sub-problem solution is always zeros (Hastie et al., 2015). As λ_{max} depends on the atoms d_k and on the signals x^n, its value changes after each D-step. In particular, its value might change a lot between the initialization and the first D-step. This is problematic since we cannot use a regularization λ above this initial λ_{max}, even though the following λ_{max} might be higher.

The standard strategy to initialize CSC methods is to generate random atoms with Gaussian white noise. However, as these atoms generally poorly correlate with the signals, the initial value of λ_{max} is low compared to the following ones. For example, on the MEG dataset described later in Subsection 5.3.5, we found that the initial λ_{max} is about 1/3 of the following ones (using $L = 32$).

To fix this problem, we propose to initialize the dictionary with random chunks of the signal. We noticed on the MEG dataset that the initial λ_{max} was then about the same values as the following ones, which allows to use higher regularization parameters, leading to sparser activations in the results.

KMC2 initialization A potential limitation of the chunk initialization is when a same pattern is present multiple times in the initial atoms. Therefore, we also investigated a different initialization based on a Markov chain Monte-Carlo (MCMC) method called AFK-MC2 (Bachem et al., 2016). This method was developed to initialize clustering algorithms. It iteratively adds a new initial centroid using random sampling, with a random rejection criterion based on the distance of the new centroid to the previous ones. Therefore, the method tends to select centroids that are far from each other, for a given distance.

In our case, we would like to use a distance which takes into account the shift invariance of the CSC model. Therefore, we used the AFK-MC2 algorithm on temporal chunks of the signal, using a custom convolutional distance:

$$f(d, d') = \min_{\tau \in [\![-L/2, L/2]\!]} \|d_\tau - d'\|_2 \ , \tag{5.14}$$

where $d_\tau[t] = d[t - \tau]$.

This strategy leads to better initializations than the random chunk strategy, as evaluated by the loss function after one Z-step. Indeed, in our experiments, the initial atoms given by the AFK-MC2 strategy were more diverse than with random chunks. However, it did not consistently lead to better minima in the sense of the loss function, nor lead to faster convergence. The qualitative evaluation of the obtained minima did not reveal a noticeable improvement. Therefore, we did not used this scheme in the presented results. Note that this conclusion could be different with a different evaluation, *e.g.* with a subsequent classification task based on the learned atoms or activations.

5.1.5 Experiments

In order to evaluate our approach, we conducted several experiments on both synthetic and empirical data. First, using simulations, we compared the CSC approach with two competing dictionary learning methods with shift-invariant sparse representations (Jost et al., 2006, Brockmeier and Príncipe, 2016). Then, we showed that our proposed optimization scheme for CSC provides significant improvements in terms of convergence speed over the state-of-the-art CSC methods. Finally, we considered LFP data, where we illustrated that our algorithm can reveal interesting properties in electrophysiological signals without supervision.

Dictionary learning with shift-invariant sparse representations In this synthetic data experiment, we illustrate the robustness of CSC in the presence of corrupted observations, compared to two competing state-of-art methods previously applied to

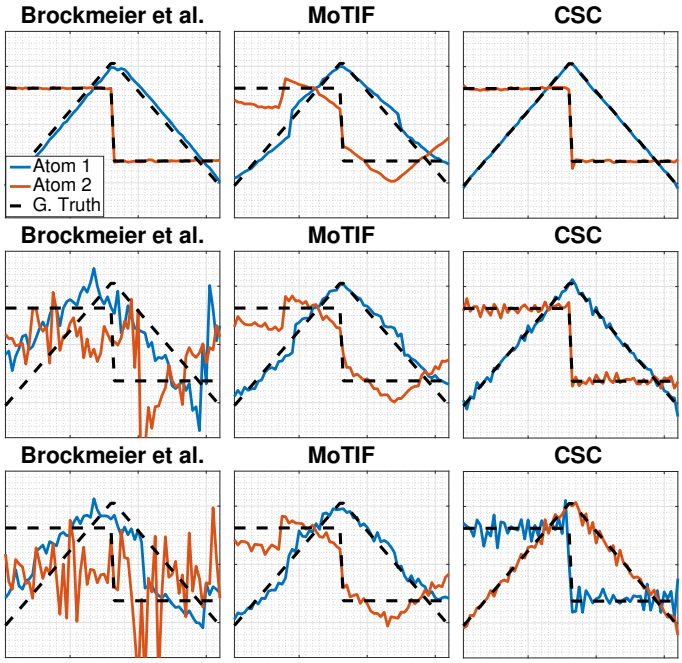


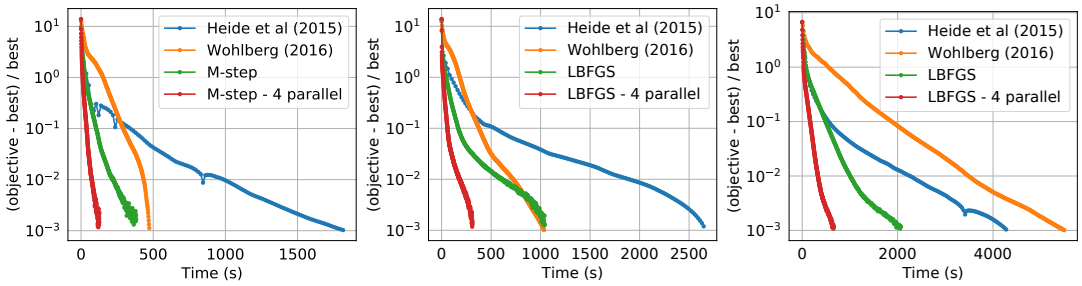
Figure 5.1 – Simulation to compare state-of-the-art dictionary learning methods applied to neural time series against CSC. From top to bottom row, we used 0%, 10%, and 20% of high noise trials.

neural time series: MoTIF (Jost et al., 2006) and a method base on matching pursuit (Brockmeier and Príncipe, 2016).

We simulated $N = 100$ trials of length $T = 512$ by first generating $K = 2$ zero-mean and unit-norm atoms of length $L = 64$. The simulated atoms are shown in dashed lines in Figure 5.1. The activation instants were integers drawn from a uniform distribution in $[\![0, T - L]\!]$. The amplitude of the activations were drawn from a uniform distribution in $[0, 1]$. Atoms were activated only once per trial and were allowed to overlap. The activations were then convolved with the generated atoms and summed up as in the CSC model definition (5.2). We corrupted a fraction of the trials (0%, 10%, or 20%) with strong Gaussian noise of standard deviation 0.1, *i.e.* one order of magnitude higher than in other trials. We used a regularization parameter of $\lambda = 0.2$.

Starting from 10 random initializations, the estimated atoms with the smallest ℓ_2 distance with the true atoms are shown in Figure 5.1. In the artifact-free scenario, all algorithms perform equally well, except for MoTIF (Jost et al., 2006) that suffers from the presence of activations with varying amplitudes. This is because it aligns the data using correlations before performing the eigenvalue decomposition, without taking into account the strength of activations in each trial. The performance of the method described in Brockmeier and Príncipe (2016) degrades as the level of corruption increases. On the other hand, CSC is clearly more robust to the increasing level of corruption and recovers reasonable atoms even when 20% of the trials are corrupted.

CSC speed performances We then show that our optimization strategy based on L-BFGS-B outperforms state-of-the-art CSC solvers in terms of convergence speed.



(a) $K = 2$, $L = 32$. (b) $K = 2$, $L = 128$. (c) $K = 10$, $L = 32$.

Figure 5.2 – Comparison of state-of-the-art methods with our approach. The y-axis shows the objective function relative to the obtained minimum for each run: $(f(x) - f(x^*))/f(x^*)$. Each curve is the geometrical mean over 24 different random initializations.

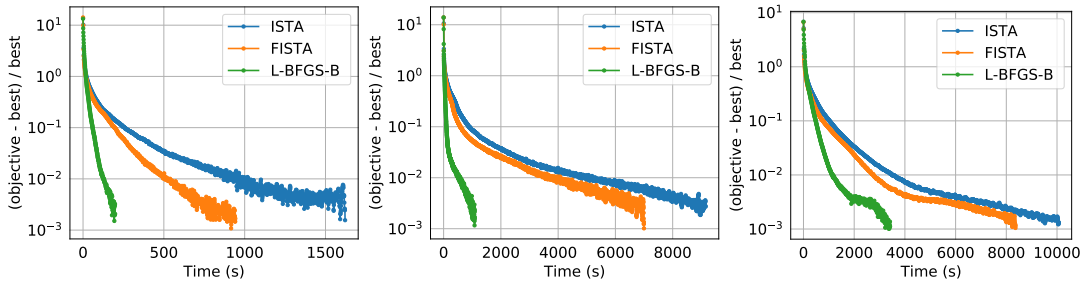
Using the same synthetic dataset as in previous experiment, we set $N = 100$, $T = 2000$ and $\lambda = 1$, and used different values for K and L. The Gaussian noise standard deviation was set to 0.01 We monitored the convergence of ADMM-based methods by Heide et al. (2015) and Wohlberg (2016b) against our CSC algorithm, using both a single-threaded and a parallel version for the Z-step. In this benchmarks, we used the L-BFGS method for the Z-step, and the L-BFGS with BCD method for the D-step. All algorithms used a single thread, except "LBFGS - 4 parallel" which uses 4 threads during the Z-step.

As the problem is non-convex, even if two algorithms start from the same point, they are not guaranteed to reach the same local minimum. Hence, for a fair comparison, we use a multiple restart strategy with averaging across 24 random seeds. Note that CSC can be viewed as a biconvex problem, for which global convergence guarantees can be shown under certain assumptions (Agarwal et al., 2014, Gorski et al., 2007). However, we observed that it is required to use multiple restarts, implying that these assumptions are not satisfied in this particular problem.

During our experiments we have observed that the ADMM-based methods do not guarantee the feasibility of the iterates. In other words, the norms of the estimated atoms might be greater than 1 during the iterations. To keep the algorithms comparable, when computing the objective value, we projected the atoms to the unit ball and scaled the activations accordingly. To be strictly comparable, we also imposed a positivity constraint on these algorithms. This is easily done by modifying the soft-thresholding operator $\text{soft}(x, \lambda) = \text{sign}(x)(|x| - \lambda)$ to be a rectified linear function $\text{ReLu}(x, \lambda) = \max(x - \lambda, 0)$.

In Figure 5.2, we illustrate the convergence behaviors of the different methods. Note that the y-axis is the precision relative to the objective value obtained upon convergence. In other words, each curve is relative to its own local minimum. Our method consistently performed better and the difference is even more striking for more challenging setups.

We also compared convergence plots of different solvers for the Z-step: ISTA, FISTA, and L-BFGS-B. The rationale for choosing a quasi-Newton solver for the Z-step becomes clear in Figure 5.3 as the L-BFGS-B solver turns out to be computationally advantageous on a variety of setups.



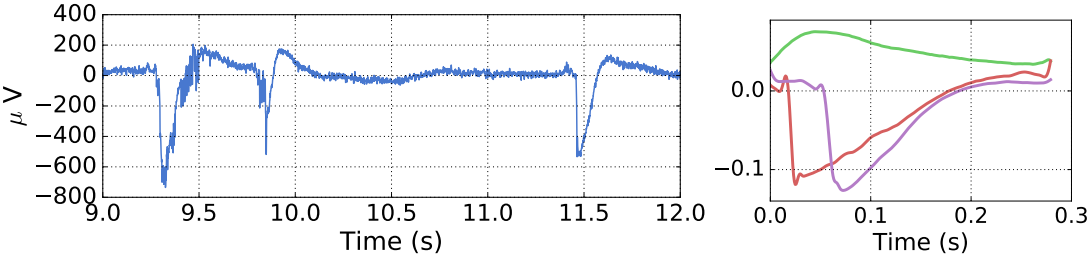
(a) $K = 2$, $L = 32$. (b) $K = 2$, $L = 128$. (c) $K = 10$, $L = 32$.

Figure 5.3 – Comparison of optimization solver in the Z-step. The y-axis shows the objective function relative to the obtained minimum for each run: $(f(x) - f(x^*))/f(x^*)$. Each curve is the geometrical mean over 24 different random initializations.

Results on spike recordings In this experiment, we considered empirical neural recordings from two different datasets. We first applied CSC on an LFP dataset previously used in (Hitziger et al., 2017) and containing epileptiform spikes as shown in Figure 5.4a. The data was recorded in the rat cortex, and is free of artifact. We segmented the data into $N = 300$ trials of length $T = 2500$ samples, windowed each trial with a tapered cosine function, and detrended the data with a high-pass filter at 1 Hz in order to remove drifts in the signal. We set $\lambda = 6$, $L = 350$, and $K = 3$. Atoms were initialized with Gaussian white noise.

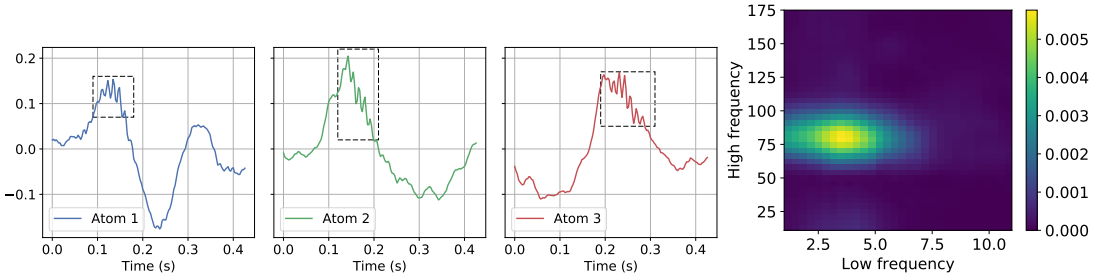
The recovered atoms by our algorithm are shown in Figure 5.4b. We can observe that the estimated atoms resemble the spikes in Figure 5.4a. These results show that, without using any heuristics, our approach can recover similar atoms to the ones reported in (Hitziger et al., 2017), even though it does not make any assumptions on the shapes of the waveforms, or initializes the atoms with template spikes in order to ease the optimization.

Results on LFP recording with phase-amplitude coupling The second dataset is an LFP channel in a rodent striatum from (Dallérac et al., 2017). We segmented the data into $N = 70$ trials of length $T = 2500$ samples, windowed each trial with a tapered cosine function, and detrended the data with a high-pass filter at 1 Hz. We set $\lambda = 10$, $L = 150$, and $K = 3$. Atoms were initialized with Gaussian white noise.



(a) LFP spike data from (Hitziger et al., 2017) (b) Estimated atoms

Figure 5.4 – Atoms learnt by αCSC on LFP data containing epileptiform spikes with $\alpha = 2$.



(a) Three atoms learnt by CSC. (b) Comodulogram.

Figure 5.5 – (a) Three atoms learnt from a rodent striatal LFP channel, using CSC. The atoms capture the cross-frequency coupling of the data (dashed rectangles). (b) Comodulogram presents the cross-frequency coupling intensity computed between pairs of frequency bands on the entire cleaned signal, following (Tort et al., 2010).

As opposed to the first LFP dataset, this dataset contains strong artifacts. Since CSC is not robust to these artifacts, we first *manually* identified and removed the corrupted trials. We also developed an alternative model which is robust to these artifacts, as presented in Section 5.2.

In Figure 5.5a, we illustrate the estimated atoms with CSC on the manually-cleaned data. We observe that the estimated atoms correspond to canonical waveforms found in the signal. In particular, the high frequency oscillations around 80 Hz are modulated in amplitude by the low-frequency oscillation around 3 Hz, a phenomenon known as cross-frequency coupling (CFC). We can observe this CFC by computing a comodulogram on the entire signal (Figure 5.5b), which measures the correlation between the amplitude of the high frequency band and the phase of the low frequency band. CFC is described in details in Chapter 3.

5.2 CSC with alpha-stable distributions

In previous section, we advocate that CSC methods have a strong potential for modeling neural signals. However, they might also be limited as they consider an ℓ_2 reconstruction error, which corresponds to assuming an additive Gaussian noise distribution. While this assumption could be reasonable for several signal processing tasks, it turns out to be very restrictive for neural signals, which often contain heavy noise bursts and have low signal-to-noise ratio.

In this section, we aim to address the aforementioned concerns and propose a novel probabilistic CSC model called αCSC, which is better-suited for neural signals. αCSC is based on a family of *heavy-tailed* distributions called α-stable distributions (Samorodnitsky and Taqqu, 1994) whose rich structure covers a broad range of noise distributions. The heavy-tailed nature of the α-stable distributions renders our model robust to impulsive observations. We develop a Monte Carlo expectation maximization (MCEM) algorithm for inference, with a weighted CSC model for the maximization step. We propose efficient optimization strategies, and we illustrate the benefits of the proposed approach on both synthetic and real datasets.

The symbols $\mathcal{U}$, $\mathcal{E}$, $\mathcal{N}$, $\mathcal{S}$ denote the univariate uniform, exponential, Gaussian, and α-stable distributions, respectively.

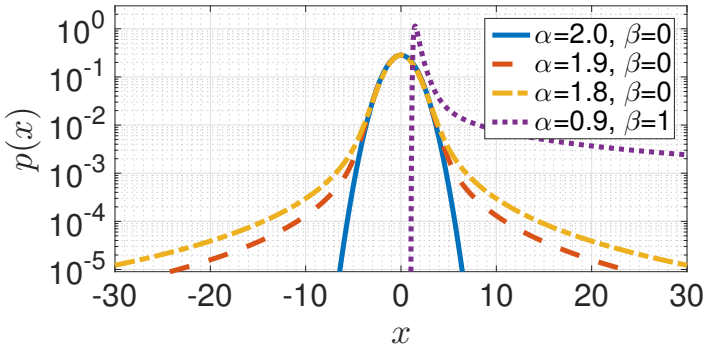


Figure 5.6 – PDFs of α-stable distributions. The distributions are more heavy-tailed than the Gaussian distribution (special case $\alpha = 2$ and $\beta = 0$).

5.2.1 Alpha-stable distributions

The α-stable distributions have become increasingly popular in modeling signals that might incur large variations (Kuruoglu, 1999, Mandelbrot, 2013, Şimşekli et al., 2015, Wang et al., 2016, Leglaive et al., 2017) and have a particular importance in statistics since they appear as the limiting distributions in the generalized central limit theorem (Samorodnitsky and Taqqu, 1994). They are characterized by four parameters:

1. $\alpha \in (0, 2]$ is the *characteristic exponent* and determines the tail thickness of the distribution: the distribution will be heavier-tailed as α gets smaller.
2. $\beta \in [-1, 1]$ is the *skewness* parameter. If $\beta = 0$, the distribution is symmetric.
3. $\sigma \in (0, \infty)$ is the *scale* parameter and measures the spread of the random variable around its mode (similar to the standard deviation of a Gaussian distribution).
4. $\mu \in (-\infty, \infty)$ is the *location* parameter (for $\alpha > 1$, it is simply the mean).

The probability density function of an α-stable distribution cannot be written in closed-form except for certain special cases. However, the characteristic function can be written as follows:

$$x \sim \mathcal{S}(\alpha, \beta, \sigma, \mu) \iff \mathbb{E}[\exp(j\omega x)] = \exp(-|\sigma\omega|^{\alpha}(1 + j\text{sign}(\omega)\beta\psi_{\alpha}(\omega)) + j\mu\omega) \ , \tag{5.15}$$

where $\psi_{\alpha}(\omega) = \log|\omega|$ for $\alpha = 1$, $\psi_{\alpha}(\omega) = \tan(\pi\alpha/2)$ for $\alpha \neq 1$, and $j^2 = -1$. As an important special case of the α-stable distributions, we obtain the Gaussian distribution when $\alpha = 2$ and $\beta = 0$, *i.e.* $\mathcal{S}(2, 0, \sigma, \mu) = \mathcal{N}(\mu, 2\sigma^2)$. In Figure 5.6, we illustrate the (approximately computed) probability density functions (PDF) of the α-stable distribution for different values of α and β. The distribution becomes heavier-tailed as we decrease α, whereas the tails vanish quickly when $\alpha = 2$.

The moments of the α-stable distributions can only be defined up to the order α, *i.e.* $\mathbb{E}[|x|^p] < \infty$ if and only if $p < \alpha$, which implies the distribution has infinite variance when $\alpha < 2$. Furthermore, despite the fact that the PDFs of α-stable distributions do not admit an analytical form, it is straightforward to draw random samples from them (Chambers et al., 1976).

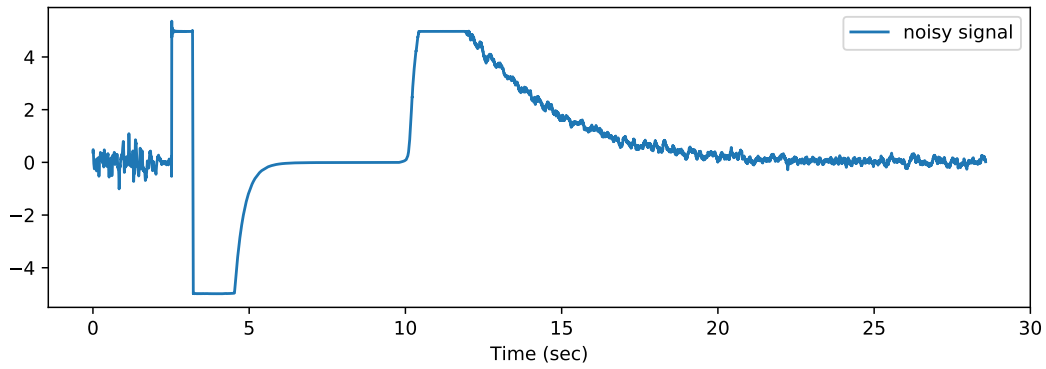


Figure 5.7 – Illustration of a striatal LFP channel, which contains severe artifacts.

5.2.2 Alpha-stable CSC

From a probabilistic perspective, the CSC problem can be also formulated as a maximum a-posteriori (MAP) estimation problem on the following probabilistic generative model:

$$z_k^n[t] \sim \mathcal{E}(\lambda), \quad x^n[t]|z,d \sim \mathcal{N}(\hat{x}^n[t],1), \quad \text{where} \quad \hat{x}^n \triangleq \sum_{k=1}^{K} z_k^n * d_k \ . \tag{5.16}$$

It is easy to verify that the MAP estimate for this probabilistic model, which reads $\text{argmax}_{d,z} \log p(d,z|x)$, is identical to the original optimization problem defined in (5.2). Note that the positivity constraint on the activations is equivalent to an exponential prior for the regularization term rather than the more common Laplacian prior.

It has been long known that, due to their light-tailed nature, Gaussian models often fail at handling noisy high amplitude observations or outliers (Huber, 1981). As a result, the "vanilla" CSC model turns out to be highly sensitive to outliers and impulsive noise that frequently occur in electrophysiological recordings, as illustrated in Figure 5.7. Possible origins of such artifacts are movement, muscle contractions, ocular blinks or electrode contact losses.

In this study, we aim at developing a probabilistic CSC model that would be capable of modeling challenging electrophysiological signals. We propose an extension of the original CSC model defined in (5.16) by replacing the light-tailed Gaussian likelihood (corresponding to the ℓ_2 reconstruction loss in (5.2)) with heavy-tailed α-stable distributions. We define the proposed probabilistic model (αCSC) as follows:

$$z_k^n[t] \sim \mathcal{E}(\lambda), \quad x^n[t]|z,d \sim \mathcal{S}(\alpha,0,1/\sqrt{2},\hat{x}^n[t]) \ . \tag{5.17}$$

While still being able to capture the temporal structure of the observed signals via convolution, the proposed model has a richer structure and would allow large variations and outliers, thanks to the heavy-tailed α-stable distributions. Note that the vanilla CSC defined in (5.16) appears as a special case of αCSC, as the α-stable distribution coincides with the Gaussian distribution when $\alpha = 2$.

5.2.3 Model estimation: maximum a posteriori (MAP) inference

Given the observed signals x, we are interested in the maximum a posteriori (MAP) estimate, defined as follows:

$$\underset{d,z}{\operatorname{argmax}} \sum_{n=1}^{N} \sum_{t=1}^{T} \left(\log p(x^n[t]|d,z) + \sum_{k=1}^{K} \log p(z_k^n[t]) \right). \tag{5.18}$$

As opposed to the Gaussian case, unfortunately, this optimization problem is not amenable to classical optimization tools, since the PDF of the α-stable distributions does not admit an analytical expression. As a remedy, we use the product property of the symmetric α-stable densities (Samorodnitsky and Taqqu, 1994, Godsill and Kuruoglu, 1999) and re-express the αCSC model as conditionally Gaussian. It leads to:

$$z_k^n[t] \sim \mathcal{E}(\lambda), \quad \phi^n[t] \sim \mathcal{S}\left(\frac{\alpha}{2}, 1, 2(\cos\frac{\pi\alpha}{4})^{2/\alpha}, 0\right), \quad x^n[t]|z,d,\phi \sim \mathcal{N}\left(\hat{x}^n[t], \frac{1}{2}\phi^n[t]\right), \tag{5.19}$$

where ϕ is called the *impulse* variable that is drawn from a *positive* α-stable distribution (*i.e.* $\beta = 1$), whose PDF is illustrated in Figure 5.6. It can be shown that both formulations of the αCSC model are identical by marginalizing the joint distribution $p(x,d,z,\phi)$ over ϕ (Samorodnitsky and Taqqu, 1994, Proposition 1.3.1).

The impulsive structure of the αCSC model becomes more prominent in this formulation. The variances of the Gaussian observations are modulated by stable random variables with infinite variance, where the impulsiveness depends on the value of α. It is also worth noting that when $\alpha = 2$, $\phi^n[t]$ becomes deterministic and we can again verify that αCSC coincides with the vanilla CSC.

The conditionally Gaussian structure of the augmented model has a crucial practical implication: if the impulse variable ϕ were to be known, then the MAP estimation problem over d and z in this model would turn into a weighted CSC problem, which is a much easier task compared to the original problem. In order to be able to exploit this property, we propose an expectation-maximization (EM) algorithm, which iteratively maximizes a lower bound of the log-posterior $\log p(d,z|x)$, and algorithmically boils down to computing the following steps in an iterative manner:

$$\text{E-Step:} \quad \mathcal{B}^{(i)}(d,z) = \mathbb{E}\left[\log p(x,\phi,z|d)\right]_{p(\phi|x,z^{(i)},d^{(i)})}, \tag{5.20}$$

$$\text{M-Step:} \quad (d^{(i+1)}, z^{(i+1)}) = \operatorname{argmax}_{d,z} \mathcal{B}^{(i)}(d,z). \tag{5.21}$$

where $\mathbb{E}[f(x)]_{q(x)}$ denotes the expectation of a function f under the distribution q, i denotes the iterations, and $\mathcal{B}^{(i)}$ is a lower bound to $\log p(d,z|x)$ which is tight at the current iterates $z^{(i)}$, $d^{(i)}$.

5.2.4 Model estimation: E-Step

In the expectation step of our algorithm, we need to compute the EM lower bound $\mathcal{B}$ that has the following form:

$$\mathcal{B}^{(i)}(d,z) =^{+} -\sum_{n=1}^{N} \left(\|\sqrt{w^{n(i)}} \odot (x^n - \sum_{k=1}^{K} z_k^n * d_k)\|_2^2 + \lambda \sum_{k=1}^{K} \|z_k^n\|_1 \right), \tag{5.22}$$

Algorithm 5.3: α-stable Convolutional Sparse Coding

Input : Regularization $\lambda \in_+$, K initial atoms d_k of length L,
Number of iterations I, J, M.

for $i = 1$ **to** I **do**
  / E-step: */*
  for $j = 1$ **to** J **do**
    Draw $\phi^n[t]^{(i,j)}$ via MCMC (5.25)
  Compute $w^{(i)}$ with (5.24)

  / M-step: weighted CSC */*
  for $m = 1$ **to** M **do**
    $z^{(i)}$ = L-BFGS-B as in Algorithm 5.1 but with (5.27)
    $d^{(i)}$ = L-BFGS-B on the dual problem as in Algorithm 5.2 but with (5.10)
return $w^{(I)}$, $d^{(I)}$, $z^{(I)}$

where $=^+$ denotes equality up to additive constants, $\odot$ denotes the Hadamard (element-wise) product, and the square-root operator is also defined element-wise. Here, $w^{n(i)} \in \mathbb{R}_+^T$ are the *weights* that are defined as follows:

$$w^n[t]^{(i)} \triangleq \mathbb{E}\left[1/\phi^n[t]\right]_{p(\phi|x,z^{(i)},d^{(i)})}. \tag{5.23}$$

As the variables $\phi^n[t]$ are expected to be large when $\hat{x}^n[t]$ cannot explain the observation $x^n[t]$ – typically due to a corruption or a high noise – the weights will accordingly suppress the importance of the particular point $x^n[t]$. Therefore, the overall approach will be more robust to corrupted data than the Gaussian models where all weights would be deterministic and equal to 0.5.

Unfortunately, the weights $w^{(i)}$ cannot be computed analytically, therefore we need to resort to approximate methods. In this study, we develop a Markov chain Monte Carlo (MCMC) method to approximately compute the weights, where we approximate the intractable expectations with a finite sample average, given as follows:

$$w^n[t]^{(i)} \approx \frac{1}{J}\sum_{j=1}^{J}\frac{1}{\phi^n[t]^{(i,j)}}, \tag{5.24}$$

where $\phi^n[t]^{(i,j)}$ are some samples that are ideally drawn from the posterior distribution $p(\phi|x, z^{(i)}, d^{(i)})$. Directly drawing samples from the posterior distribution of ϕ is not tractable either, and therefore, we develop a Metropolis-Hastings algorithm (Chib and Greenberg, 1995), that asymptotically generates samples from the target distribution $p(\phi|\cdot)$ in two steps. In the j-th iteration of this algorithm, we first draw a random sample for each n and t from the prior distribution (*c.f.* (5.19)), *i.e.* $\phi^n[t]' \sim p(\phi^n[t])$. We then compute an acceptance probability for each $\phi^n[t]'$ that is defined as follows:

$$\text{acc}(\phi^n[t]^{(i,j)} \to \phi^n[t]') \triangleq \min\left(1, \frac{p(x^n[t]|d^{(i)}, z^{(i)}, \phi'^n[t])}{p(x^n[t]|d^{(i)}, z^{(i)}, \phi^n[t]^{(i,j)})}\right), \tag{5.25}$$

where j denotes the iteration number of the MCMC algorithm.

Finally, we draw a uniform random number $u^n[t] \sim \mathcal{U}([0,1])$ for each n and t. If $u^n[t] < \text{acc}(\phi^n[t]^{(i,j)} \to \phi^n[t]')$, we accept the sample and set $\phi^n[t]^{(i+1)} = \phi^n[t]'$; otherwise we reject the sample and set $\phi^n[t]^{(i+1)} = \phi^n[t]^{(i)}$. This procedure forms a Markov chain that

leaves the target distribution $p(\phi|\cdot)$ invariant, where under mild ergodicity conditions, it can be shown that the finite-sample averages converge to their true values when J goes to infinity (Liu, 2008).

5.2.5 Model estimation: M-Step

Given the weights w^n that are estimated during the E-step, the objective of the M-step is to solve a weighted CSC problem, which is much easier compared to our original problem. The problem reads:

$$\begin{aligned} \underset{\{d_k\},\{z_k^n\}}{\operatorname{argmin}} \sum_{n=1}^{N} \frac{1}{2} \left\| \sqrt{w^n} \odot \left(x^n - \sum_{k=1}^{K} z_k^n * d_k\right) \right\|_2^2 + \lambda \sum_{k=1}^{K} \|z_k^n\|_1 \ , \\ \text{s.t.} \quad \|d_k\|_2^2 \leq 1, \text{ and } z_k^n \geq 0, \quad \forall k, n. \end{aligned} \tag{5.26}$$

Here, similarly to the vanilla CSC approach described in Section 5.1, we develop a block coordinate descent strategy, where we solve the problem for either $\{d_k\}$ or $\{z_k^n\}$, by keeping the other block of variable fixed. Note that in the weighted case, it is not clear how to solve this problem in the Fourier domain. We thus perform all the computations in the time domain, using the L-BFGS solvers described in Subsection 5.1.2 and Subsection 5.1.3.

For the Z-step, we just need to update the gradient is updated into:

$$\nabla_{z_k} = \overleftarrow{d_k} * \left(w \odot \left(x - \sum_{l=1}^{K} z_l * d_l \right) \right) + \lambda \sum_{l=1}^{K} z_l \ . \tag{5.27}$$

Thus, the weighted Z-step only adds a component-wise product over NT time points, compared to the vanilla CSC Z-step.

For the D-step, we need to update the primal optimal computation into:

$$d^*(\beta) = \left(\sum_{n=1}^{N} (w^n \odot Z^{n\top}) Z^n + \bar{\beta} \right)^{-1} \sum_{n=1}^{N} (w^n \odot Z^{n\top}) x^n \tag{5.28}$$

Unfortunately, this linear system does not have a Toeplitz structure as in the non-weighted case. Therefore, building and solving the system is slightly more expensive. Using the BCD strategy, building the system has a complexity of $\mathcal{O}(NKTL^2)$ instead of $\mathcal{O}(NKTL)$, and each gradient step costs $\mathcal{O}(KL^3)$ instead of $\mathcal{O}(KL^2)$.

Our entire EM approach can be summarized in the Algorithm 5.3. Note that during the alternating minimization, thanks to convexity we can warm start the d update and the z update using the solution from the previous update. This significantly speeds up the convergence of the L-BFGS-B algorithm, particularly in the later iterations of the overall algorithm.

5.2.6 Model initialization

As for standard CSC, αCSC models are non-convex, and the initialization is therefore a decisive step of the optimization. We initialize the atoms and activations in the say fashion than for standard CSC, as described in Subsection 5.1.4.

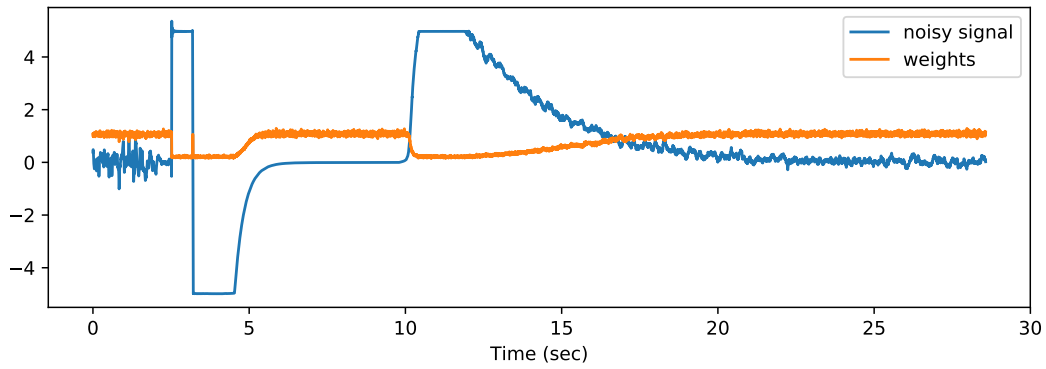


Figure 5.8 – The initial weights estimated with an E-step. The subsequent M-step will be less affected by the strong artifacts since they are down-weighted.

A good initialization of the weights $w^n[t]$ is also critical. If we initialize the weights with a constant value, the ℓ_2-norm of the weighted CSC model puts more importance in the large values of the signals, which are generally artifacts. Therefore, the obtained atoms often corresponds to these artifacts, and αCSC models do not perform better than CSC models.

To solve this issue, we can initialize the weights with the inverse of the standard deviation of each trial. Therefore, initial weights are small in trials with large amplitude artifacts, making the model robust to these artifacts. The subsequent iterations of the EM algorithm refine the weights, atoms and activations, with respect to the α-stable distribution, but the initial setup largely determine the final optimum, and the robustness to artifacts.

Actually, a better way to tackle this issue is to start with a weights estimation, using only zeros in the initial estimate of the reconstructed signals, $\hat{x}^n[t] = 0$ for all n and t. It simply corresponds to doing an E-step first, instead of an M-step. The initial weights then corresponds roughly to $w^n[t] \approx \max(1/x^n[t], g(\alpha))$, where $g(\alpha)$ only depends on the parameter α of the distribution. This threshold effect is due to the positive α-stable distribution of $\phi^n[t] \sim \mathcal{S}\left(\frac{\alpha}{2}, 1, 2(\cos\frac{\pi\alpha}{4})^{2/\alpha}, 0\right)$, as shown in Figure 5.6 (with $\beta = 1$). An example of such weights initialization is presented in Figure 5.8.

5.2.7 Experiments

To demonstrate the robustness of αCSC models, we performed the same experiment as in Subsection 5.1.5, using an LFP channel in a rodent striatum from (Dallérac et al., 2017). We initialized the weights w_n to the inverse of the variance of the trial x_n, and the atoms with Gaussian white noise.

We compared the atoms of three different setups:

1. A CSC model applied on clean data, *i.e.* where we removed manually the artifacts,
2. A CSC model applied on the full data,
3. An αCSC model applied on the full data, using $\alpha = 1.2$.

We illustrate the estimated atoms in Figure 5.9. Even though CSC is able to provide excellent results on the clean dataset, its performance heavily relies on the manual removal of the artifacts. It can be observed that in the presence of strong artifacts, CSC

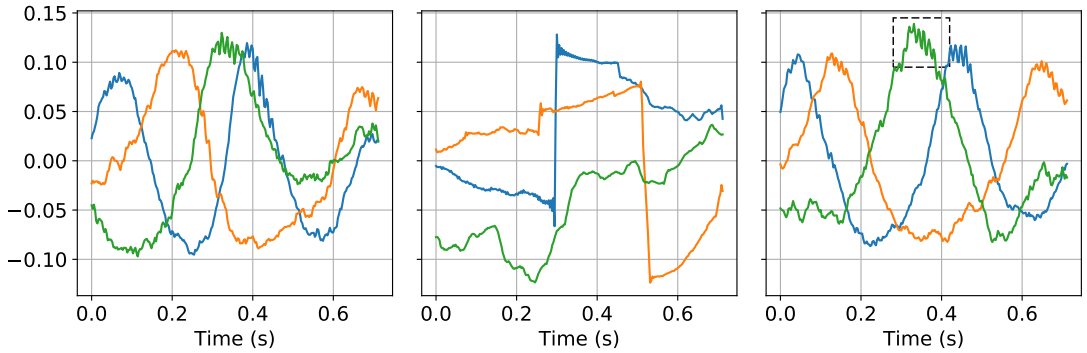


Figure 5.9 – Three atoms learnt respectively by: CSC (clean data), CSC (full data), αCSC (full data). The atoms contains cross-frequency coupling (dashed rectangle). The CSC model is not able to obtain these atoms since it is too much affected by large amplitude artifacts. On the other hand, the αCSC model is robust to these artifacts.

is not able to recover the atoms anymore. On the contrary, we see that αCSC can still recover atoms as observed in the artifact-free regime. In particular, the cross-frequency coupling phenomenon is still clearly visible.

5.3 Multivariate CSC with a rank-1 constraint

The CSC approach was essentially developed for univariate signals. Interestingly, images can be multivariate such as color or hyper-spectral images, yet most CSC methods only consider gray scale images. To the best of our knowledge, the only reference to multivariate CSC is Wohlberg (2016a), where the author proposes two models well suited for 3-channel images.

In the case of EEG and MEG recordings, neural activity is instantaneously and linearly spread across channels, due to Maxwell's equations (Hari and Puce, 2017). The same temporal patterns are reproduced on all channels with different intensities, which depend on each activity's location in the brain. To exploit this property, we propose to use a rank-1 constraint on each multivariate atom. This idea has been mentioned in (Barthélemy et al., 2012, 2013), but was considered less flexible than the full-rank model. Moreover, their proposed optimization techniques are not specific to shift-invariant models, and not scalable to long signals.

5.3.1 Model definitions

Univariate CSC Let's recall the univariate CSC model as defined in (5.2):

$$\underset{\{d_k\},\{z_k^n\}}{\operatorname{argmin}} \sum_{n=1}^{N} \frac{1}{2} \left\| x^n - \sum_{k=1}^{K} z_k^n * d_k \right\|_2^2 + \lambda \sum_{k=1}^{K} \|z_k^n\|_1 \ , \quad \text{s.t.} \quad \|d_k\|_2^2 \leq 1 \text{ and } z_k^n \geq 0, \quad \forall k, n, \tag{5.29}$$

where $\{x^n\}_{n=1}^N \subset \mathbb{R}^T$ are N observed signals of length T, $\lambda > 0$ is the regularization parameter, $\{d_k\}_{k=1}^K \subset \mathbb{R}^L$ are the K temporal atoms of length L we aim to learn, and $\{z_k^n\}_{k=1}^K \subset \mathbb{R}^{T-L+1}$ are K signals of activations a.k.a. the code associated with x^n. We note $T = T - L + 1$.

Multivariate CSC The multivariate formulation uses an additional dimension on the signals and on the atoms, since the signal is recorded over P channels (mapping to space locations):

$$\begin{aligned} \underset{\{D_k\},\{z_k^n\}}{\operatorname{argmin}} \sum_{n=1}^{N} \frac{1}{2} \left\| X^n - \sum_{k=1}^{K} z_k^n * D_k \right\|_2^2 + \lambda \sum_{k=1}^{K} \|z_k^n\|_1, \\ \text{s.t.} \quad \|D_k\|_2^2 \leq 1 \text{ and } z_k^n \geq 0 \quad \forall k, n, \end{aligned} \tag{5.30}$$

where $\{X^n\}_{n=1}^N \subset \mathbb{R}^{P \times T}$ are N observed multivariate signals, $\{D_k\}_{k=1}^K \subset \mathbb{R}^{P \times L}$ are the spatio-temporal atoms, and $\{z_k^n\}_{k=1}^K \subset \mathbb{R}^{\widetilde{T}}$ are the sparse activations associated with X^n.

Multivariate CSC with rank-1 constraint This model is similar to the multivariate case but it adds a rank-1 constraint on the dictionary, $D_k = u_k v_k^\top \in \mathbb{R}^{P \times L}$, with $u_k \in \mathbb{R}^P$ being the pattern over channels and $v_k \in \mathbb{R}^L$ the pattern over time. The optimization problem boils down to:

$$\begin{aligned} \underset{\{u_k\},\{v_k\},\{z_k^n\}}{\operatorname{argmin}} \sum_{n=1}^{N} \frac{1}{2} \left\| X^n - \sum_{k=1}^{K} z_k^n * (u_k v_k^\top) \right\|_2^2 + \lambda \sum_{k=1}^{K} \left\| z_k^n \right\|_1, \\ \text{s.t.} \quad \|u_k\|_2^2 \leq 1 \ , \ \|v_k\|_2^2 \leq 1 \text{ and } z_k^n \geq 0, \quad \forall k, n. \end{aligned} \tag{5.31}$$

The rank-1 constraint is consistent with Maxwell's equations and the physical model of electrophysiological signals like EEG or MEG, where each source is linearly spread instantaneously over channels with a constant topographic map (Hari and Puce, 2017). Using this assumption, one aims to improve the estimation of patterns under the presence of independent noise over channels. Moreover, it can help separating overlapping sources which are inherently rank-1 but whose sum is generally of higher rank. Finally, as explained below, several computations can be factorized to speed up computational time.

General estimation scheme Problems (5.29), (5.30) and (5.31) share the same structure. They are convex in each variable but not jointly convex. The resolution is done by using a block coordinate descent approach which minimizes alternatingly the objective function over one block of the variables. In the following subsection, we describe this approach on the multivariate with rank-1 constraint case (5.31), updating iteratively the activations z_k^n, the spatial patterns u_k, and the temporal pattern v_k.

5.3.2 Model estimation: Z-step

Given K *fixed* multivariate atoms $D_k \in \mathbb{R}^{P \times L}$ and a regularization parameter $\lambda > 0$, we recall that the Z-step aims to retrieve the NK activation signals $z_k^n \in \mathbb{R}^{\widetilde{T}}$ associated to the signals $X^n \in \mathbb{R}^{P \times T}$ by solving the following ℓ_1-regularized optimization problem:

$$\begin{aligned} \underset{\{z_k^n\}}{\operatorname{argmin}} \sum_{n=1}^{N} \frac{1}{2} \left\| X^n - \sum_{k=1}^{K} z_k^n * D_k \right\|_2^2 + \lambda \sum_{k=1}^{K} \|z_k^n\|_1 \ , \\ \text{s.t.} \quad z_k^n \geq 0, \quad \forall k, n. \end{aligned} \tag{5.32}$$

Algorithm 5.4: Locally greedy coordinate descent (LGCD)

Input : Signal x, atoms d_k, number of segments M, stopping parameter $\epsilon > 0$, z_k initialization

Initialize $\beta_k[t]$ with (5.33).

repeat

  for $m = 1$ **to** M **do**

    Compute $z'_k[t] = \max\left(\frac{\beta_k[t]-\lambda}{\|D_k\|_2^2}, 0\right)$ for $(k,t) \in \mathcal{C}_m$

    Choose $(k_0, t_0) = \underset{(k,t)\in\mathcal{C}_m}{\operatorname{argmax}} |z_k[t] - z'_k[t]|$

    Update β with (5.34)

    Update the current point estimate $z_{k_0}[t_0] \leftarrow z'_{k_0}[t_0]$

until $\|z - z'\|_\infty < \epsilon$;

This problem can be solved with techniques such as FISTA, ADMM, L-BFGS, or GCD, as described in Subsection 5.1.2. For long signals, these techniques can be quite slow due to the computation of the gradient (FISTA, ADMM, L-BFGS) or the choice of the best coordinate to update in GCD, which are operations that scale linearly in T. A way to alleviate this limitation is to use a locally greedy coordinate descent (LGCD) strategy, presented recently in Moreau et al. (2018).

Coordinate descent (CD) The key idea of coordinate descent is to update our estimate of the solution one coordinate $z_k[t]$ at a time. For (5.32), it is possible to compute the optimal value $z'_k[t]$ of one coordinate $z_k[t]$ given that all the others are fixed. Indeed, the problem restricted to one coordinate has a closed-form solution given by:

$$z'_k[t] = \max\left(\frac{\beta_k[t]-\lambda}{\|D_k\|_2^2}, 0\right), \text{ with } \beta_k[t] = \left[D_k^{\backprime} \tilde{*} \left(x - \sum_{l=1}^{K} z_l * D_l + z_k[t] e_t * D_k\right)\right][t] \tag{5.33}$$

where $e_t \in \mathbb{R}^{\widetilde{T}}$ is the canonical basis vector with value 1 at index t and 0 elsewhere. A proof of this closed-form solution is given in Appendix A.1.1.

When updating the coefficient $z_{k_0}[t_0]$ to the value $z'_{k_0}[t_0]$, β is updated with:

$$\beta_k^{(q+1)}[t] = \beta_k^{(q)}[t] + (D_{k_0}^{\backprime} \tilde{*} D_k)[t - t_0](z_{k_0}[t_0] - z'_{k_0}[t_0]), \qquad \forall (k,t) \neq (k_0, t_0) \ . \tag{5.34}$$

The term $(D_{k_0}^{\backprime} \tilde{*} D_k)[t - t_0]$ is zero for $|t - t_0| \geq L$. Thus, only $K(2L-1)$ coefficients of β need to be changed (Kavukcuoglu et al., 2010). A proof of this update is given in Appendix A.1.2. The CD algorithm updates at each iteration a coordinate to this optimal value. The coordinate to update can be chosen with different strategies:

- The cyclic CD (Friedman et al., 2007), which iterates over all coordinates.
- The randomized CD (Nesterov, 2010, Richtárik and Takáč, 2014), which chooses a coordinate at random for each iteration.
- The greedy CD (Osher and Li, 2009), which chooses the coordinate the farthest from its optimal value.

Locally greedy coordinate descent (LGCD) The choice of a coordinate selection strategy results of a tradeoff between the computational cost of each iteration and the improvement it provides. For cyclic and randomized strategies, the iteration complexity is $\mathcal{O}(KL)$ as the coordinate selection has a constant complexity. The greedy selection of a coordinate is more expensive as it is linear in the signal length $\mathcal{O}(K\widetilde{T})$. However, greedy selection is more efficient iteration-wise (Nutini et al., 2015).

Moreau et al. (2018) proposed to consider a locally greedy selection strategy for CD. The coordinate to update is chosen greedily in one of M subsegments of the signal, *i.e.* at iteration q, the selected coordinate is:

$$(k_0, t_0) = \underset{(k,t)\in\mathcal{C}_m}{\operatorname{argmax}} |z_k[t] - z'_k[t]| \ , \qquad m \equiv q \pmod M + 1 \ , \tag{5.35}$$

with $\mathcal{C}_m = [\![1, K]\!] \times [\![(m-1)\widetilde{T}/M, m\widetilde{T}/M]\!]$. With this strategy, the coordinate selection complexity is linear in the length of the considered subsegment $\mathcal{O}(K\widetilde{T}/M)$. By choosing $M = \lfloor \widetilde{T}/(2L-1) \rfloor$, the complexity of update is the same as the complexity of random and cyclic coordinate selection, $\mathcal{O}(KL)$. We detail the steps of LGCD in Algorithm 5.4.

Using the LGCD algorithm is not faster than L-BFGS in the general case, since it is fast only when few coefficients need to be updated. However, this algorithm is particularly efficient when the z_k are sparse. Indeed, in this case, only few coefficients need to be updated in the signal, resulting in a low number of iterations to reach convergence. Therefore, LGCD is particularly efficient when the regularization parameter λ is large. The difference with L-BFGS can be decisive for long time series, as shown in the experiments Subsection 5.3.5. Computational complexities are detailed in Table 5.2.

Precomputation To speed up the LGCD iterations during the Z-step, we can precompute $D_k^{\curvearrowleft} \tilde{*} D_l \in \mathbb{R}^{2L-1}$. We have:

$$(D_k^{\curvearrowleft} \tilde{*} D_l)[t] = \sum_{p=1}^{P} \sum_{\tau=1}^{L} D_{k,p}[\tau] D_{l,p}[t+\tau-1], \quad \forall t \in [\![1, 2L-1]\!]. \tag{5.36}$$

In the case of the rank-1 constraint model, we can factorize the computation with:

$$(D_k^{\curvearrowleft} \tilde{*} D_l)[t] = \left(\sum_{p=1}^{P} u_{k,p} u_{l,p} \right) \sum_{\tau=1}^{L} v_k[\tau] v_l[t+\tau-1], \quad \forall t \in [\![1, 2L-1]\!]. \tag{5.37}$$

The computational complexities are respectively $\mathcal{O}\left(K^2L^2P\right)$ and $\mathcal{O}\left(K^2L(L+P)\right)$.

5.3.3 Model estimation: D-step

Given KN *fixed* activation signals $z_k^n \in \mathbb{R}^{\widetilde{T}}$, associated to signals $x^n \in \mathbb{R}^T$, the D-step aims to update the K temporal patterns $d_k \in \mathbb{R}^L$, by solving:

$$\begin{aligned} \underset{\{u_k\},\{v_k\}}{\operatorname{argmin}} E = \underset{\{u_k\},\{v_k\}}{\operatorname{argmin}} \sum_{n=1}^{N} \frac{1}{2} \left\| X^n - \sum_{k=1}^{K} z_k^n * (u_k v_k^\top) \right\|_2^2 \\ \text{s.t.} \quad \|u_k\|_2^2 \leq 1 \text{ and } \|v_k\|_2^2 \leq 1, \quad \forall k. \end{aligned} \tag{5.38}$$

In Subsection 5.1.3, we solve the univariate problem with L-BFGS in the dual problem. But here the D-step problem is not convex, so we must resort to a block coordinate descent, first updating $\{u_k\}$ while keeping $\{v_k\}$, and then updating $\{v_k\}$ while keeping $\{u_k\}$ fixed. We use in both cases a projected gradient descent (PGD) algorithm.

Algorithm 5.5: Projected gradient descent for updating $\{u_k\}$ and $\{v_k\}$.

Input : Signals X^n, activations z_k^n, stopping parameter $\epsilon > 0$,
initial estimate $\{u_k\}$ and $\{v_k\}$

Initialize Φ_k with (5.41) and Ψ_k with (5.42) .

repeat

  Compute with (5.43) for $k \in [\![1, K]\!]$, $G_k = \nabla_{u_k} E$,

  Update the estimate with $\{u_k^{(q+1)}\} \leftarrow$ **to** `Armijo`$(\{u_k^{(q)}\}, G_k, E)$

until $\sum_{k=1}^K \left\| u_k^{(q+1)} - u_k^{(q)} \right\|_1 < \epsilon$;

Set $\{u_k\} \leftarrow \{u_k^{(q)}\}$

repeat

  Compute with (5.44) for $k \in [\![1, K]\!]$, $G_k = \nabla_{v_k} E$,

  Update the estimate with $\{v_k^{(q+1)}\} \leftarrow$ **to** `Armijo`$(\{v_k^{(q)}\}, G_k, E)$

until $\sum_{k=1}^K \left\| v_k^{(q+1)} - v_k^{(q)} \right\|_1 < \epsilon$;

Set $\{v_k\} \leftarrow \{v_k^{(q)}\}$

return $\{u_k\}_k$ and $\{v_k\}_k$

Projected gradient descent (PGD) PGD is an algorithm which alternates between a gradient descent step and a projection on the constraint set. In the case of (5.38), the projection on the unit ball is a simple scaling operation:

$$\text{proj}(u_k) = \frac{u_k}{\max(\|u_k\|_2, 1)} \tag{5.39}$$

The projection is identical for v_k. To compute the gradients $\nabla_{u_k} E$ and $\nabla_{v_k} E$ of (5.38) relative to u_k and v_k, we first compute the gradient $\nabla_{D_k} E$ relative to the multivariate atom $D_k = u_k v_k^\top \in \mathbb{R}^{P \times L}$. We also introduce some constants Φ_k and $\Psi_{k,l}$, which are constant during the entire D-step:

$$\nabla_{D_k} E = \sum_{n=1}^N (z_k^n)^\curlywedge * \left(x^n - \sum_{l=1}^K z_l^n * D_l \right) = \Phi_k - \sum_{l=1}^K \Psi_{k,l} * D_l \tag{5.40}$$

where $\Phi_k \in \mathbb{R}^L$ and $\Psi_{k,l} \in \mathbb{R}^{2L-1}$ are computed with:

$$\Phi_k[t] = \sum_{n=1}^N \sum_{\tau=1}^{\widetilde{T}} z_k^n[\tau] x^n[t + \tau - 1], \quad \forall t \in [\![1, L]\!], \tag{5.41}$$

$$\Psi_{k,l}[t] = \sum_{n=1}^N \sum_{\tau=1}^{\widetilde{T}} z_k^n[\tau] z_l^n[t + \tau - 1], \quad \forall t \in [\![1, 2L-1]\!]. \tag{5.42}$$

Details of this calculation are available in Appendix A.1.3. Note that in the last equation (5.42), the sum only concerns the defined terms, *i.e.* the time points τ such that $(t + \tau - 1) \in [\![1, \widetilde{T}]\!]$.

Then, the gradients relative to u_k and v_k are obtained using the chain rule:

$$\nabla_{u_k} E = (\nabla_{D_k} E) v_k \quad \in \mathbb{R}^P \ , \tag{5.43}$$

$$\nabla_{v_k} E = u_k^\top (\nabla_{D_k} E) \quad \in \mathbb{R}^L \ , \tag{5.44}$$

Algorithm 5.6: Power-iteration is used to find the maximum eigenvalue of a linear operator.

Input : Linear operator A, initial vector u, tolerance parameter ϵ
repeat
  $u \leftarrow Au/\|Au\|$
  $\lambda' \leftarrow \lambda$
  $\lambda \leftarrow u^\top Au$
until $(\lambda - \lambda')/\lambda' < \epsilon$;
return eigenvalue λ and eigenvector u

Algorithm 5.5 details the different step used in our algorithm to update $\{u_k\}$ and $\{v_k\}$. Note that E can also be computed efficiently, up to a constant term C, with the following:

$$E = \sum_{k=1}^{K} u_k^\top (\nabla_{D_k} E) v_k + C \ . \tag{5.45}$$

Step-size strategy To find a good step size during the gradient descent, we can estimate the Lipschitz constant L of each sub-problem. To do so, we use a *power-iteration* (Trefethen and Bau III, 1997) scheme on the linear operators A and B, where:

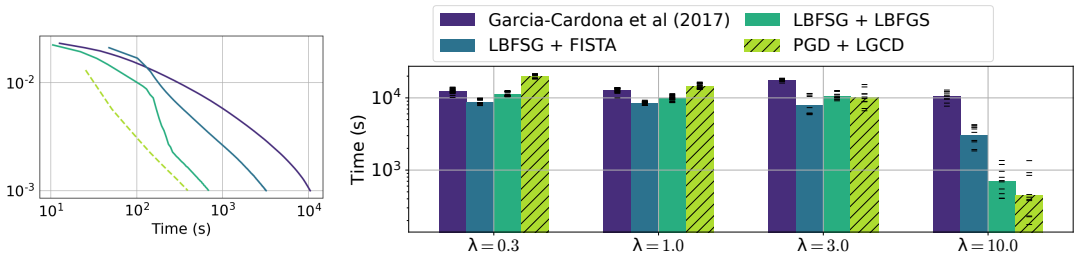
$$(Au)_k = \left(\sum_{l=1}^{K} \Psi_{k,l} * D_l \right) v_k, \qquad \text{and } (Bv)_k = u_k^\top \left(\sum_{l=1}^{K} \Psi_{k,l} * D_l \right). \tag{5.46}$$

Power-iteration consists in applying a linear operator multiple times to a random initial vector, using normalizations step to stay in reasonable numerical scales. Considering the eigenvector decomposition of the random initial vector, each eigenvector is scaled by λ_i^N, where λ_i is the associated eigenvalue, and N is the number of times the operator is applied. As the result is scaled to be unit norm, each eigenvector component goes to zero, except the one associated with the largest eigenvalue. The algorithm is detailed in Algorithm 5.6. After estimating the Lipschitz constant L, we can use the optimal step-size $1/L$ in the PGD to solve the D-step.

Note that both Lipschitz constants L_u and L_v have to be updated after each Z-step. The power iteration can be warm-started by storing the eigenvectors u and v associated with the largest eigenvalues, and using it as the initial vectors in the subsequent power iteration procedures.

Table 5.2 – Computational complexities of each step.

Step	Method	Computation	Computed	Complexity
Z-step	LGCD	β initialization	once	$NKTL$
Z-step	LGCD	Precomputation	once	K^2L^2
Z-step	LGCD	M coordinate updates	multiple times	MKL
D-step	PGD	Φ precomputation	once	$NKTL$
D-step	PGD	Ψ precomputation	once	NK^2TL
D-step	PGD	Gradient evaluation	multiple times	K^2L^2
D-step	PGD	Function evaluation	multiple times	K^2L^2



(a) $\lambda = 10$ (univariate). (b) Time to reach a precision of 10^{-3} (univariate).

Figure 5.10 – Comparison of state-of-the-art univariate methods. (a) Convergence plot with the objective function relative to the obtained minimum, as a function of computational time. (b) Time taken to reach a relative precision of 10^{-3}, for different regularization parameters λ.

An alternative method is to use the Armijo backtracking line-search (Wright and Nocedal, 1999), which dynamically adapts the step-size at each PGD iteration. To do so, the Armijo line-search needs to evaluate the function $E(u, v)$ multiple times, but this evaluation is computationally cheap when using precomputed constants Φ_k and $\Psi_{k,l}$, and (5.45). In our experiments, this strategy was computationally more efficient than estimating the Lipschitz constants.

5.3.4 Model initialization

We already discussed the need of a good initialization in Subsection 5.1.4. In particular, we saw that random atoms poorly correlate with the signal, making the initial value of λ_{max} very low compared to the following ones. This phenomenon is even stronger on multivariate atoms, where we witnessed initial λ_{max} as low as $1/20$ of the following ones, when using Gaussian white noise atoms.

Once again, to fix this problem, we propose to initialize the dictionary with random chunks of the signal. To deal with the rank-1 constraint of the multivariate atoms, we project each multivariate chunk on a rank-1 approximation using singular value decomposition (SVD), and keeping only the largest eigenvalue.

5.3.5 Experiments

Speed performance on univariate CSC To illustrate the performance of our optimization strategy, we monitored its convergence speed on a MEG dataset, the somatosensory dataset from the MNE software (Gramfort et al., 2013, 2014), which contains responses to median nerve stimulation. We considered only the gradiometers channels, which measure the gradient of the magnetic field, and we used the following parameters: $T = 134\,700$, $N = 2$, $K = 8$, and $L = 128$.

First we compared our PGD and LGCD solvers against three state-of-the-art *univariate* CSC solvers. The first was developed by Garcia-Cardona and Wohlberg (2017) and is based on ADMM, while the second and third were our methods described in Section 5.1, using L-BFGS in the dual for the D-step, and respectively FISTA and L-BFGS for the Z-step. All solvers shared the same objective function, but as the problem is non-convex, the solvers are not guaranteed to reach the same local minima, even though we started from the same initial settings. Hence, for a fair comparison, we computed the convergence curves relative to each local minimum, and averaged them over 10 different initializations.

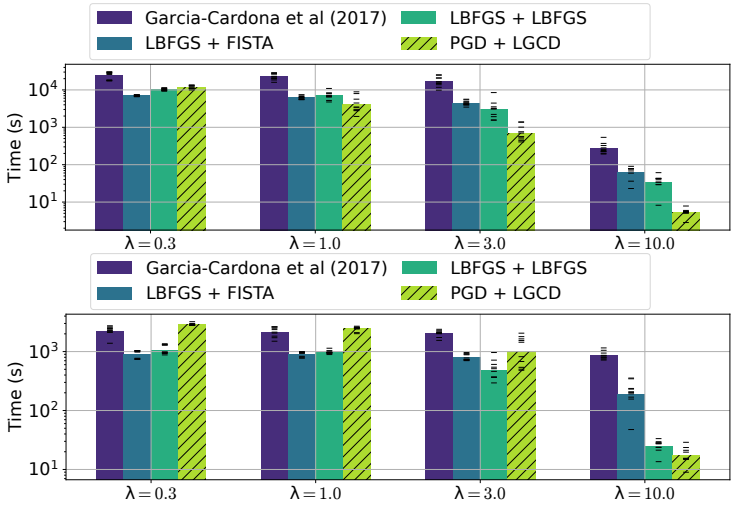


Figure 5.11 – Comparison of state-of-the-art univariate methods. (Top) With shorter ($L = 16$ instead of $L = 128$) atoms. (Bottom) With shorter signals ($T = 13\ 470$ instead of $T = 134\ 700$).

The results, presented in Figure 5.10(a, b), demonstrate the competitiveness of our method, for reasonable choices of λ. Indeed, a higher regularization parameter leads to sparser activations z_k^n, on which the LGCD algorithm is particularly efficient.

We also tried shorter atoms of length $L = 16$ instead of $L = 128$, and independently shorter signals of length $T = 13\ 470$ instead of $T = 134\ 700$. Results are presented in Figure 5.11. They confirm the competitiveness of our method, especially when using large regularization parameters. As the maximum possible regularization λ_{max} was around 90, we question the practical use of low regularization values, which would poorly enforce the sparsity constraint.

Speed performance on multivariate CSC Then, we also compared our method against a multivariate ADMM solver developed by Wohlberg (2016a). As this solver was quite slow on these long signals, we limited our experiments to $P = 5$ channels. The results, presented in Figure 5.12 and Figure 5.13, show that our method is faster than the competing method for large λ. The difference is even more prominent on larger number of channels.

Scaling with the number of channels The multivariate model involves an extra dimension P but its impact on the computational complexity of our solver is limited. Figure 5.14 shows the average running times of the Z-step and the D-step. Timings are normalized with respect to the timings for a single channel. The running times are computed using the same signals from the somatosensory dataset, with the following parameters: $T = 26\ 940$, $N = 10$, $K = 2$, $L = 128$. We can see that the scaling of these three operations is sub-linear in P. For the Z-step, only the initial computations for the first β_k and the constants $D_k^{\boldsymbol{\curvearrowleft}} \tilde{*} D_l$ depend linearly on P so that the complexity increase is limited compared to the complexity of solving the optimization problem (5.32). For the D-step, the computation of the gradients is linear with P, but the most expensive operations here are the computation of the constant Ψ_k, which does not depend on P.

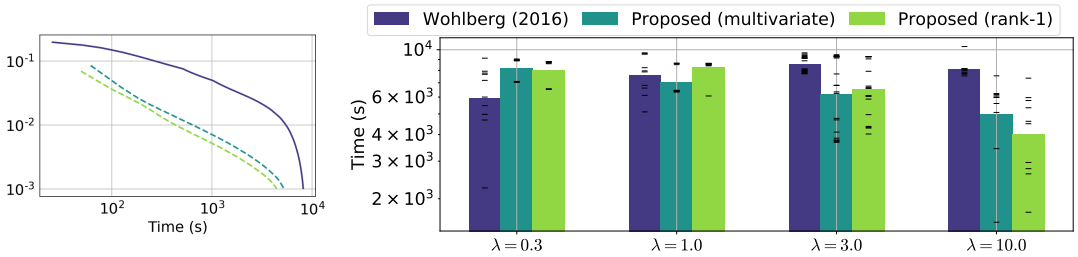
Finding patterns in low SNR signals Since the multivariate model has access to more data, we would expect it to perform better compared to the univariate model especially for low SNR signals. To demonstrate this, we compared the two models when varying the number of channels P and the SNR of the data. The original dictionary contained two patterns, a square and a triangle, presented in Figure 5.15. The signals were obtained by convolving the atoms with activation signals z_k^n, where the activation locations were sampled uniformly in $[\![1, \widetilde{T}]\!] \times [\![1, K]\!]$ with 5% non-zero activations, and the amplitudes were uniformly sampled in $[0, 1]$. Then, a Gaussian white noise with variance σ was added to the signal. We fixed $N = 100$, $L = 64$ and $\widetilde{T} = 640$ for our simulated signals. We can see in Figure 5.15a the temporal patterns recovered for $\sigma = 10^{-3}$ using only one channel and using 5 channels. While the patterns recovered with one channel are very noisy, the multivariate model with rank-1 constraint recovers the original atoms accurately. This can be expected as the univariate model is ill-defined in this situation, where some atoms are superimposed. For the rank-1 model, as the atoms have different spatial maps, the problem is easier. Indeed, the spatial maps are different for each atom, their sum is not of rank-1. The constraint is thus helping to separate the patterns as soon as there is more than one channel.

Then, we evaluated the learned temporal atoms. Due to permutation and sign ambiguity, we computed the ℓ_2-norm of the difference between the temporal pattern $\widehat{v}_k$ and the ground truths, v_k or $-v_k$, for all permutations $\mathfrak{S}(K)$ *i.e.*

$$\text{loss}(\widehat{v}) = \min_{s \in \mathfrak{S}(K)} \sum_{k=1}^{K} \min \left(\|\widehat{v}_{s(k)} - v_k\|_2^2, \|\widehat{v}_{s(k)} + v_k\|_2^2 \right) \quad . \tag{5.47}$$

Multiple values of λ were tested and the best loss is reported in Figure 5.15b for varying noise levels σ. We observe that independently of the noise level, the multivariate rank-1 model outperforms the univariate one. This is true even for good SNR, as using multiple channels disambiguates the separation of overlapping patterns.

Examples of atoms in real MEG signals We also tested our algorithm on experimental data, using the MNE "somatosensory" dataset (Gramfort et al., 2013, 2014). We first extracted $N = 103$ trials from the data. Each trial lasts 6 s with a sampling frequency of 150 Hz ($T = 900$). We selected only gradiometer channels, leading to $P = 204$ channels. The signals were notch-filtered to remove the power-line noise, and high-pass filtered at 2 Hz to remove the low-frequency trend. The purpose of the



(a) $\lambda = 10$ (multivariate). (b) Time to reach a precision of 10^{-3} (multivariate).

Figure 5.12 – Comparison of state-of-the-art multivariate methods, using $P = 5$ channels. (a) Convergence plot with the objective function relative to the obtained minimum, as a function of computational time. (b) Time taken to reach a relative precision of 10^{-3}, for different regularization parameters λ.

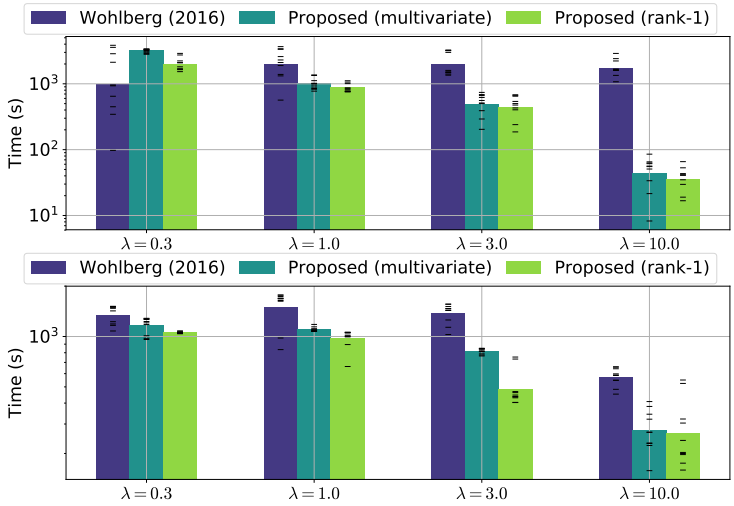
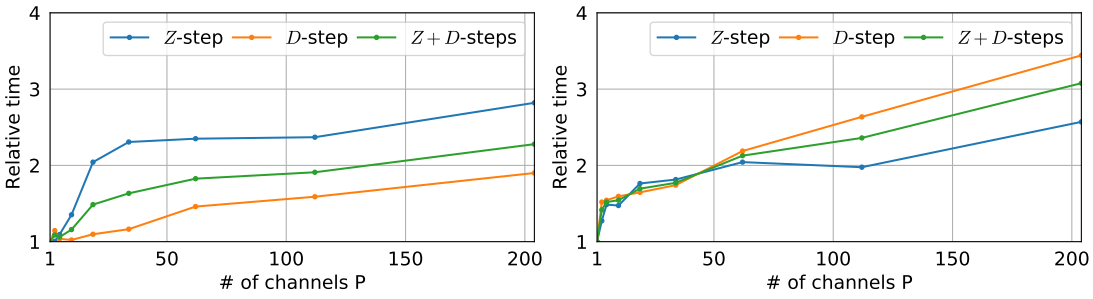


Figure 5.13 – Comparison of state-of-the-art multivariate methods. (Top) With shorter ($L = 16$ instead of $L = 128$) atoms. (Bottom) With shorter signals ($T = 13\,470$ instead of $T = 134\,700$).

temporal filtering is to remove low frequency drift artifacts which contribute a lot to the variance of the raw signals.

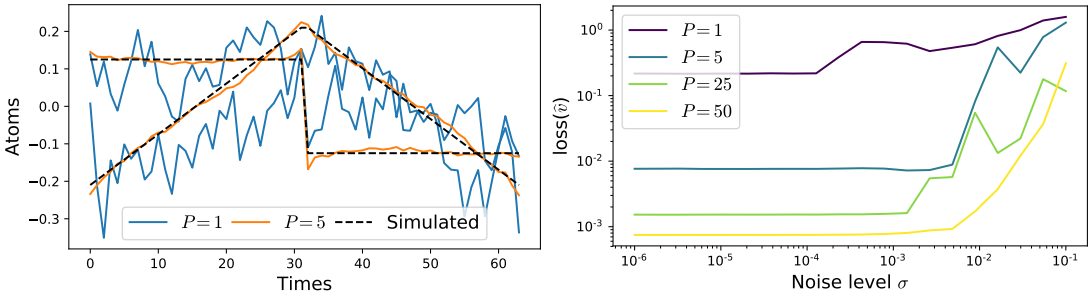
Figure 5.16a shows a recovered non-sinusoidal brain rhythm which resembles the well-known mu-rhythm. The mu-rhythm has been implicated in motor-related activity (Hari, 2006) and is centered around 9–11 Hz. Indeed, while the power is concentrated in the same frequency band as the alpha, it has a very different spatial topography (Figure 5.16b). In Figure 5.16c, the power spectral density (PSD) shows two components of the mu-rhythm – one at around 9 Hz, and a harmonic at 18 Hz as previously reported in (Hari, 2006). Based on our analysis, it is clear that the 18 Hz component is simply a harmonic of the mu-rhythm even though a Fourier-based analysis could lead us to falsely conclude that the data contained beta-rhythms.

Finally, due to the rank-1 nature of our atoms, it is straightforward to fit an equivalent current dipole (Tuomisto et al., 1983) to interpret the origin of the signal. Figure 5.16d shows that the atom does indeed localize in the primary somatosensory cortex, or the so-called S1 region with a 59.3% goodness of fit. Intriguingly, we also found such atoms in the *secondary* somatosensory region, also known as S2. One such atom is shown in



(a) $\lambda = .005\lambda_{\max}$ (b) $\lambda = .001\lambda_{\max}$

Figure 5.14 – Timings of Z and D updates when varying the number of channels P. The scaling is sublinear with P, due to the precomputation steps in the optimization.

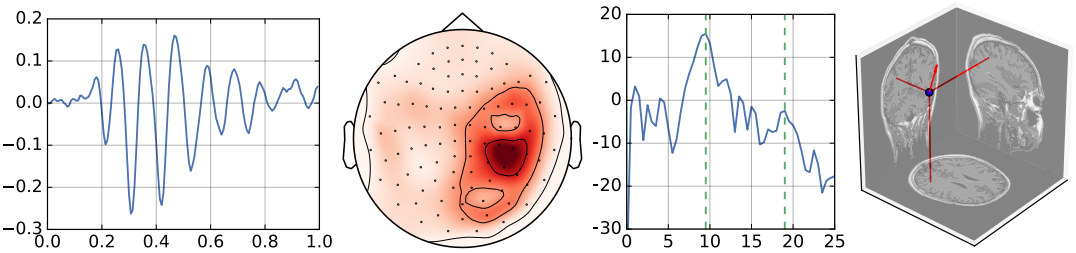


(a) Patterns recovered with 1 and 5 channels. (b) Loss with respect to noise level σ and P.

Figure 5.15 – (a) Temporal patterns recovered with $P = 1$ and $P = 5$. The signals were generated with the two simulated temporal patterns and with $\sigma = 10^{-3}$. (b) Evolution of the recovery loss (lower is better) with σ for different values of P. Using more channels improves the recovery of the original patterns.

Figure 5.17.

Sample dataset In addition to the MNE somatosensory dataset, we also analyzed the MNE "sample" dataset (Gramfort et al., 2013, 2014). In this case, we used $N = 1$, and the number of time points $T = 41\,584$ corresponds to 278 s of recording sampled at 150.15 Hz. The magnetometer channels are selected so the number of channels is $P = 102$. We learned $K = 25$ atoms. The sample data was lowpass filtered at 40 Hz, and highpass filtered at 1 Hz. Results are presented in Figure 5.18.



(a) Temporal atom (b) Spatial pattern (c) PSD (dB) (d) Dipole fit

Figure 5.16 – Atoms revealed using the MNE somatosensory data. Note the non-sinusoidal comb shape of the mu rhythm.

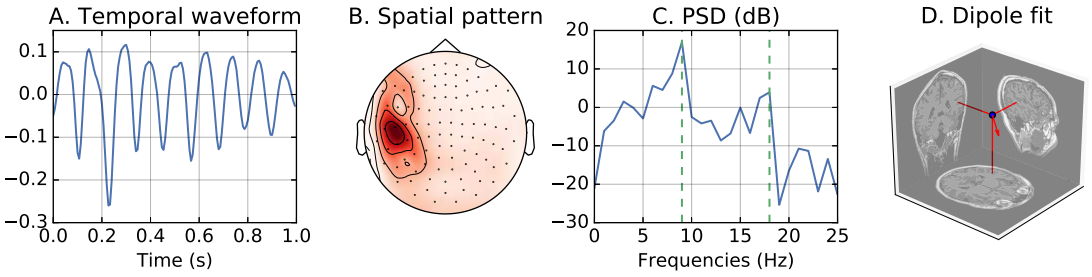


Figure 5.17 – Atom in the S2 region revealed in the MNE somatosensory data. A. The temporal waveform, and its corresponding B. Spatial pattern, C. The Power Spectral Density (PSD), and D. the dipole fit in the S2 region.

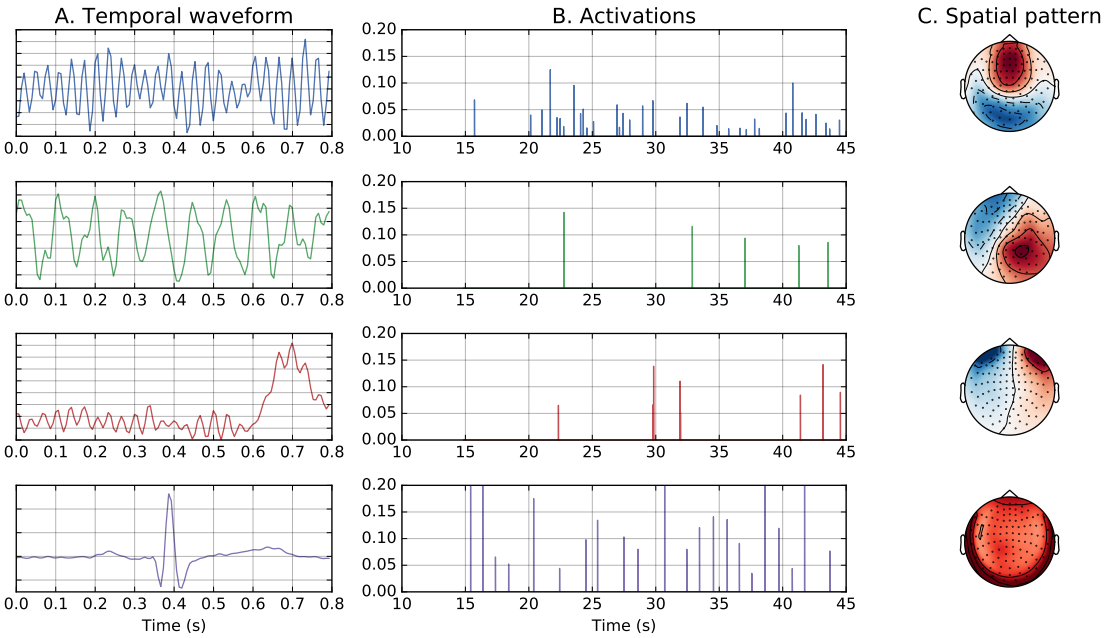


Figure 5.18 – A selection of A. temporal waveforms of the atoms learned on the MNE sample dataset, and their corresponding B. activations, and C. spatial patterns

Figure 5.18A shows the temporal waveforms of these atoms and Figure 5.18C shows the corresponding spatial pattern for a selection of the total atoms. As expected, we are able to recover latent components corresponding to ocular (3rd row) and cardiac artifacts (4th row). Indeed, the ocular artifacts displays the prototypical dipolar pattern in the frontal channels. In Figure 5.18B, we also show the sparse activations associated with the atoms.

More interestingly, we also recover an oscillatory waveform (first row) which appears to originate due to a dipole below the parietal channels at around a frequency of 30 Hz. We confirm this in Figure 5.19 using a dipole fit. Indeed, the atom does originate in the parietal lobe which suggests that what we observe is probably a motor rhythm. The dataset under consideration did in fact contain a button press task which could explain the presence of such an atom.

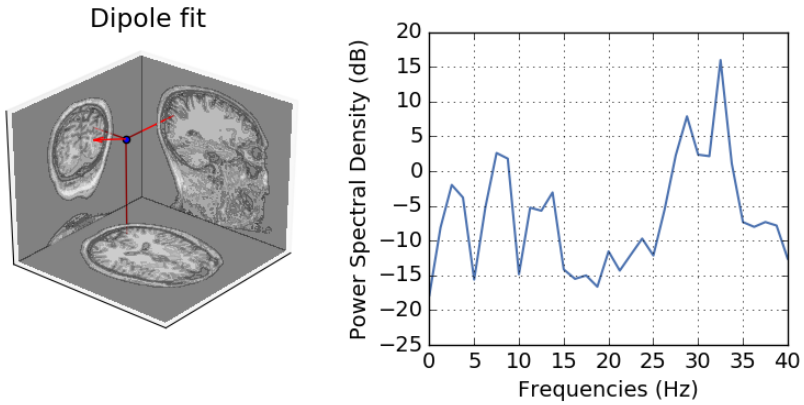


Figure 5.19 – Dipole fit and power spectral density computed on MNE somato sample dataset for the atom in first row in Figure Figure 5.18.

Conclusion

"It's a magical world, Hobbes, ol'd buddy...
...let's go exploring!"

– Bill Watterson

Neurophysiological time-series are extremely challenging from a signal processing perspective. They contain complex structures that need dedicated analyzing tools to be properly described, and their poor signal-to-noise ratio requires the design of sophisticated methods to aggregate the weak signal of interest over multiple time windows or multiple channels. Yet in many situations, the standard approach is based on linear filtering, ad-hoc correlation metrics, and surrogate methods for assessing significance. Such pipelines are arbitrary, often slow and statistically weak, and cannot be accurately compared except on over-simplistic simulations.

We advocate that a better approach consists in defining appropriate signal models. Such models need to be rich enough to capture the structure of the signal of interest, and simple enough to provide a robust and efficient estimation method. With a well-defined loss function, estimating a model resorts to an optimization problem, which can be solved using standard optimization algorithms. More importantly, if we can evaluate the goodness of fit of a model on a validation set, we enable model selection through cross-validation, leading to legitimate parameter selection and to reproducible data-driven analyses.

As in many situations, we do not need to design such models from scratch. We can stand on the shoulders of giants and reuse models developed in other contexts for other applications. Following this idea, the main contribution of this work is based on non-linear autoregressive models, which were developed in scientific communities focusing on speech processing and on econometry. Yet these models required some adjustments to properly fit to our cross-frequency coupling application.

These models describe the modulation of power-spectral density of a signal conditionally to another. They are fast to estimate, statistically powerful, and easy to compare. The model comparison is particularly powerful as discussed in details in this work. We also proposed a number of more sophisticated extensions, from adding a simple delay parameter, to estimating couplings on multivariate signals.

Using a similar modeling approach, we also explored another class of models, which focus on the temporal morphology of neurophysiological time-series. While previous models were based on spectral estimation, these models are based on temporal representations, assuming that the signal is composed of a small number of prototypical waveforms. Here again, we advocate that representing the model through a loss function makes the problem properly-defined and naturally leads to standard optimization techniques.

The natural mathematical formulation of this problem, namely convolutional sparse coding, was also inspired by existing work in another domain, image processing. To tackle the unusual challenges of neurophysiological time-series, we adapted these models

to heavy-tail noise distribution and multivariate signals. We showed that we obtain a rich signal representation in a completely unsupervised way, avoiding a Fourier decomposition that hides some relevant properties of the morphology of the neurophysiological signals..

Developing these models, we focused on building easily reusable tools and methods, to foster their use in the community. Indeed, re-implementing recent technical methods is a major pain point amongst researchers, especially in interdisciplinary research. As a result, we strove to make the code of our research projects not only available online, but also tested with continuous integration, extensively documented with examples, and carefully designed to be as simple to use as possible (https://pactools.github.io/, https://alphacsc.github.io/).

Our models offer novel and more robust possibilities to analyze neurophysiological time-series, paving the way for new insights on how our brain functions via spectral interactions or prototypical waveforms.

A

Appendices

Contents

A.1 Convolutional sparse coding 125
A.1.1 The optimal coordinate descent update 125
A.1.2 The β update 126
A.1.3 Details of the D-step PGD update 127

A.1 Convolutional sparse coding

A.1.1 The optimal coordinate descent update

The following proposition gives the optimal update of on coordinate during coordinate descent (CD) in the Z-step of the convolutional sparse coding (CSC) model described in Chapter 5.

Proposition A.1. *The optimal update $z'_{k_0}[t_0]$ of the coefficient (k_0, t_0) is given by*

$$z'_{k_0}[t_0] = \frac{1}{\|D_{k_0}\|_2^2} \max\left(\beta_{k_0}[t_0] - \lambda, 0\right) \ ,$$

with $\beta_{k_0}[t_0] = D_{k_0}^{\tilde{\uparrow}} \tilde{*} \left(X - \sum_{k=1}^K z_k * D_k + z_{k_0}[t_0] e_{t_0} * D_{k_0}\right)[t_0]$ and where e_{t_0} is the canonical vector in $\mathbb{R}^{\widetilde{T}}$ with value 1 in t_0 and value 0 elsewhere.

Proof. For $y \in \mathbb{R}^+$, we will denote $e_{k_0,t_0}(y)$ the cost difference between our current solution estimate z_k and the signal $z_k^{(1)}$ where the coefficient $z_{k_0}[t_0]$ has been replaced by y, *i.e.*

$$z_k^{(1)}[t] = \begin{cases} y, & \text{if } (k,t) = (k_0, t_0) \\ z_k[t], & \text{elsewhere} \end{cases} \ .$$

Let $\alpha_{k_0}[t] = (X - \sum_{k=1}^K z_k * D_k)[t] + D_{k_0}[t - t_0] z_{k_0}[t_0]$ for all $t \in [\![0, T-1]\!]$. This quantity denotes the residual when $z_{k_0}[t_0]$ is set to 0. It is important to note that it can be re-written as,

$$\alpha_k[t] = \left(X - \sum_{k=1}^K z_k * D_k + z_{k_0}[t_0] e_{t_0} * D_{k_0}\right)[t]$$

and thus, $\beta_{k_0}[t_0] = \left(D_{k_0}^{\curvearrowleft} \tilde{*} \alpha_{k_0}\right)[t_0]$. The cost difference $e_{k_0,t_0}(y)$ is,

$$\begin{aligned}
e_{k_0,t_0}(y) &= \frac{1}{2}\sum_{t=0}^{T-1}\left(X - \sum_{k=1}^{K} z_k * D_k\right)^2[t] + \lambda\sum_{k=1}^{K}\|z_k\|_1 - \frac{1}{2}\sum_{t=0}^{T-1}\left(X - \sum_{k=1}^{K} z_k^{(1)} * D_k\right)^2[t] + \lambda\sum_{k=1}^{K}\|z_k^{(1)}\|_1 \\
&= \frac{1}{2}\sum_{t=0}^{T-1}\left(\alpha_{k_0}[t] - D_{k_0}[t-t_0]z_{k_0}[t_0]\right)^2 - \frac{1}{2}\sum_{t=0}^{T-1}\left(\alpha_{k_0}[t] - D_{k_0}[t-t_0]y\right)^2 + \lambda(|z_{k_0}[t_0]| - |y|) \\
&= \frac{1}{2}\sum_{t=0}^{T-1} D_{k_0}[t-t_0]^2(z_{k_0}[t_0]^2 - y^2) - \sum_{t=0}^{T-1}\alpha_{k_0}[t]D_{k_0}[t-t_0](z_{k_0}[t_0] - y) + \lambda(|z_{k_0}[t_0]| - |y|) \\
&= \frac{\|D_{k_0}\|_2^2}{2}(z_{k_0}[t_0]^2 - y^2) - \underbrace{(D_{k_0}^{\curvearrowleft} \tilde{*} \alpha_{k_0})[t_0]}_{\beta_{k_0}[t_0]}(z_{k_0}[t_0] - y) + \lambda(|z_{k_0}[t_0]| - |y|)
\end{aligned}$$

Using this result, we can derive the optimal value $z'_{k_0}[t_0]$ to update the coefficient (k_0, t_0) as the solution of the following optimization problem:

$$z'_{k_0}[t_0] = \arg\max_{y\in\mathbb{R}^+} e_{k_0,t_0}(y) \sim \arg\min_{u\in\mathbb{R}^+} \frac{\|D_{k_0}\|_2^2}{2}\left(y - \frac{\beta_{k_0}[t_0]}{\|D_{k_0}\|_2^2}\right)^2 + \lambda y \ . \tag{A.1}$$

Simple computations show the desired result, *i.e.*

$$z'_{k_0}[t_0] = \frac{1}{\|D_{k_0}\|_2^2}\max(\beta_{k_0}[t_0] - \lambda, 0) \ .$$

□

A.1.2 The β update

After a coordinate update in CD during the Z-step of a CSC model, we need to update the variables β, using the following proposition:

Proposition A.2. *When updating the coefficient $z_{k_0}[t_0]$ to the value $z'_{k_0}[t_0]$, β is updated with:*

$$\beta_k^{(q+1)}[t] = \beta_k^{(q)}[t] + (D_{k_0}^{\curvearrowleft} \tilde{*} D_k)[t-t_0](z_{k_0}[t_0] - z'_{k_0}[t_0]), \qquad \forall (k,t) \neq (k_0,t_0) \ . \tag{A.2}$$

Proof. The value of $\beta_{k_0}[t_0]$ is independent of the value of $z_{k_0}[t_0]$. Indeed, the term $z_{k_0}[t_0]e_{t_0} * D_{k_0}$ cancels the contribution of $z_{k_0}[t_0]$ in the convolution $z_{k_0} * D_{k_0}$. Thus, when updating the value of the coefficient $z_{k_0}[t_0]$, $\beta_{k_0}[t_0]$ is not updated.

We denote $z_k^{(q+1)}$ the activation signal where the coefficient $z_{k_0}[t_0]$ as been updated to $z'_{k_0}[t_0]$, *i.e.*,

$$z_k^{(q+1)}[t] = \begin{cases} z'_{k_0}[t_0], & \text{if } (k,t) = (k_0,t_0) \\ z_k[t], & \text{elsewhere} \end{cases} .$$

For $(k,t) \neq (k_0,t_0)$,

$$
\begin{aligned}
\beta_k^{(q+1)}[t] &= \left[D_k^{\Lsh} \tilde{*} \left(X - \sum_{l=1}^{K} z_l^{(1)} * D_l + z_k[t] e_t * D_k \right) \right][t] \\
&= \left[D_k^{\Lsh} \tilde{*} \left(X - \sum_{l=1}^{K} z_l * D_l + z_k[t] e_t * D_k + (z_{k_0}[t_0] - z'_{k_0}[t_0]) e_{t_0} * D_k \right) \right][t] \\
&= \left[D_k^{\Lsh} \tilde{*} \left(X - \sum_{l=1}^{K} z_l * D_l + z_k[t] e_t * D_k \right) \right][t] + \left[D_k^{\Lsh} \tilde{*} \left((z_{k_0}[t_0] - z'_{k_0}[t_0]) e_{t_0} * D_k \right) \right][t] \\
&= \beta_k^{(q)}[t] + (z_{k_0}[t_0] - z'_{k_0}[t_0]) \left[D_k^{\Lsh} \tilde{*} \left(e_{t_0} * D_k \right) \right][t] \\
&= \beta_k^{(q)}[t] + (D_k^{\Lsh} \tilde{*} D_k)[t - t_0](z_{k_0}[t_0] - z'_{k_0}[t_0])
\end{aligned}
$$

With this relation, it is possible to keep β_k up to date with few operation after each coordinate update. □

A.1.3 Details of the D-step PGD update

First, let's recall the objective function, as introduced in Subsection 5.3.3:

$$
\begin{aligned}
\underset{\{u_k\},\{v_k\}}{\operatorname{argmin}} E = \underset{\{u_k\},\{v_k\}}{\operatorname{argmin}} \sum_{n=1}^{N} \frac{1}{2} \left\| X^n - \sum_{k=1}^{K} z_k^n * (u_k v_k^\top) \right\|_2^2 \\
\text{s.t.} \quad \|u_k\|_2^2 \leq 1 \text{ and } \|v_k\|_2^2 \leq 1.
\end{aligned} \tag{A.3}
$$

To compute the gradient relatively to a full atom $D_k = u_k v_k^\top \in \mathbb{R}^{P \times L}$, we introduce some constants Φ_k and $\Psi_{k,l}$, which are constant during the entire D-step:

$$
\nabla_{D_k} E = \sum_{n=1}^{N} (z_k^n)^{\Lsh} * \left(X^n - \sum_{l=1}^{K} z_l^n * D_l \right) = \Phi_k - \sum_{l=1}^{K} \Psi_{k,l} * D_l \tag{A.4}
$$

Indeed, we have:

$$\nabla_{D_k} E[t] = \sum_{n=1}^{N} \left((z_k^n)^{\backprime} * \left(X^n - \sum_{l=1}^{K} z_l^n * D_l \right) \right)[t] \tag{A.5}$$

$$= \sum_{n=1}^{N} \sum_{\tau=1}^{\widetilde{T}} z_k^n[\tau] \left(X^n - \sum_{l=1}^{K} z_l^n * D_l \right)[t+\tau-1] \tag{A.6}$$

$$= \sum_{n=1}^{N} \sum_{\tau=1}^{\widetilde{T}} z_k^n[\tau] \left(X^n[t+\tau-1] - \sum_{l=1}^{K} \sum_{\tau'=1}^{L} z_l^n[\tau'] D_l[t+\tau-\tau'] \right) \tag{A.7}$$

$$= \Phi_k[t] - \sum_{l=1}^{K} \sum_{\tau'=1}^{L} \left(\sum_{n=1}^{N} \sum_{\tau=1}^{\widetilde{T}} z_k^n[\tau] z_l^n[t+\tau-\tau'] \right) D_l[\tau'] \tag{A.8}$$

$$= \Phi_k[t] - \sum_{l=1}^{K} \sum_{\tau'=1}^{L} \Psi_{k,l}[t+1-\tau'] D_l[\tau'] \tag{A.9}$$

$$= \Phi_k[t] - \sum_{l=1}^{K} (\Psi_{k,l} * D_l)[t] \tag{A.10}$$

where $\Phi_k \in \mathbb{R}^{P \times L}$ are computed with:

$$\Phi_k[t] = \sum_{n=1}^{N} \sum_{\tau=1}^{\widetilde{T}} z_k^n[\tau] X^n[t+\tau-1], \quad \forall t \in [\![1, L]\!], \tag{A.11}$$

and where $\Psi_{k,l} \in \mathbb{R}^{2L-1}$ are computed with:

$$\Psi_{k,l}[t] = \sum_{n=1}^{N} \sum_{\tau=1}^{\widetilde{T}} z_k^n[\tau] z_l^n[t+\tau-1], \quad \forall t \in [\![1, 2L-1]\!]. \tag{A.12}$$

Note that in the last equation (A.12), the sum only concerns the defined terms, *i.e.*$(t + \tau - 1) \in [\![1, \widetilde{T}]\!]$. The computational complexities of Φ_k and $\Psi_{k,l}$ are respectively $\mathcal{O}\left(NLTKP\right)$ and $\mathcal{O}\left(NLTK^2\right)$.

Bibliography

Ables, J. (1974). Maximum entropy spectral analysis. *Astronomy and Astrophysics Supplement Series*, 15:383.

Agarwal, A., Anandkumar, A., Jain, P., Netrapalli, P., and Tandon, R. (2014). Learning sparsely used overcomplete dictionaries. In *Conference on Learning Theory*, pages 123–137.

Akaike, H. (1998). Information theory and an extension of the maximum likelihood principle. In *Selected Papers of Hirotugu Akaike*, pages 199–213. Springer.

Amiri, M., Lina, J.-M., Pizzo, F., and Gotman, J. (2016). High frequency oscillations and spikes: separating real hfos from false oscillations. *Clinical Neurophysiology*, 127(1):187–196.

Arlot, S., Celisse, A., et al. (2010). A survey of cross-validation procedures for model selection. *Statistics surveys*, 4:40–79.

Aru, J., Aru, J., Priesemann, V., Wibral, M., Lana, L., Pipa, G., Singer, W., and Vicente, R. (2015). Untangling cross-frequency coupling in neuroscience. *Current opinion in neurobiology*, 31:51–61.

Axmacher, N., Henseler, M. M., Jensen, O., Weinreich, I., Elger, C. E., and Fell, J. (2010). Cross-frequency coupling supports multi-item working memory in the human hippocampus. *Proceedings of the National Academy of Sciences*, 107(7):3228–3233.

Axmacher, N., Mormann, F., Fernández, G., Elger, C. E., and Fell, J. (2006). Memory formation by neuronal synchronization. *Brain research reviews*, 52(1):170–182.

Baccalá, A. L. and Sameshima, K. (2001). Partial directed coherence: a new concept in neural structure determination. *Biological Cybernetics*, 84(6):463–474.

Bachem, O., Lucic, M., Hassani, H., and Krause, A. (2016). Fast and provably good seedings for k-means. In *Advances in Neural Information Processing Systems*, pages 55–63.

Bagnell, J. A. and Bradley, D. M. (2009). Differentiable sparse coding. In *Advances in neural information processing systems*, pages 113–120.

Baker, A. P., Brookes, M. J., Rezek, I. A., Smith, S. M., Behrens, T., Smith, P. J. P., and Woolrich, M. (2014). Fast transient networks in spontaneous human brain activity. *Elife*, 3:e01867.

Barnett, L., Barrett, A. B., and Seth, A. K. (2009). Granger causality and transfer entropy are equivalent for gaussian variables. *Physical review letters*, 103(23):238701.

Barthélemy, Q., Gouy-Pailler, C., Isaac, Y., Souloumiac, A., Larue, A., and Mars, J. I. (2013). Multivariate temporal dictionary learning for EEG. *J. Neurosci. Methods*, 215(1):19–28.

Barthélemy, Q., Larue, A., Mayoue, A., Mercier, D., and Mars, J. I. (2012). Shift & 2d rotation invariant sparse coding for multivariate signals. *IEEE Transactions on Signal Processing*, 60(4):1597–1611.

Bastos, A. M., Vezoli, J., and Fries, P. (2015). Communication through coherence with inter-areal delays. *Current opinion in neurobiology*, 31:173–180.

Baum, L. E. and Petrie, T. (1966). Statistical inference for probabilistic functions of finite state markov chains. *The annals of mathematical statistics*, 37(6):1554–1563.

Beck, A. and Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202.

Benesty, J., Sondhi, M. M., and Huang, Y. (2007). *Springer handbook of speech processing*. Springer.

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the royal statistical society. Series B (Methodological)*, pages 289–300.

Berman, J. I., McDaniel, J., Liu, S., Cornew, L., Gaetz, W., Roberts, T. P., and Edgar, J. C. (2012). Variable bandwidth filtering for improved sensitivity of cross-frequency coupling metrics. *Brain connectivity*, 2(3):155–163.

Besserve, M., Schölkopf, B., Logothetis, N. K., and Panzeri, S. (2010). Causal relationships between frequency bands of extracellular signals in visual cortex revealed by an information theoretic analysis. *Journal of computational neuroscience*, 29(3):547–566.

Bigdely-Shamlo, N., Mullen, T., Kothe, C., Su, K.-M., and Robbins, K. A. (2015). The PREP pipeline: standardized preprocessing for large-scale EEG analysis. *Frontiers in neuroinformatics*, 9:16.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of econometrics*, 31(3):307–327.

Bonferroni, C. (1936). Teoria statistica delle classi e calcolo delle probabilita. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commericiali di Firenze*, 8:3–62.

Bonnefond, M., Kastner, S., and Jensen, O. (2017). Communication between brain areas based on nested oscillations. *eNeuro*, 4(2):ENEURO–0153.

Boto, E., Holmes, N., Leggett, J., Roberts, G., Shah, V., Meyer, S. S., Muñoz, L. D., Mullinger, K. J., Tierney, T. M., Bestmann, S., et al. (2018). Moving magnetoencephalography towards real-world applications with a wearable system. *Nature*, 555(7698):657.

Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122.

Bragin, A., Jandó, G., Nádasdy, Z., Hetke, J., Wise, K., and Buzsáki, G. (1995). Gamma (40-100 Hz) oscillation in the hippocampus of the behaving rat. *The Journal of Neuroscience*, 15(1):47–60.

Bristow, H., Eriksson, A., and Lucey, S. (2013). Fast convolutional sparse coding. In *Computer Vision and Pattern Recognition (CVPR)*, pages 391–398.

Brockmeier, A. J. and Príncipe, J. C. (2016). Learning recurrent waveforms within EEGs. *IEEE Transactions on Biomedical Engineering*, 63(1):43–54.

Brumberg, J. S., Nieto-Castanon, A., Kennedy, P. R., and Guenther, F. H. (2010). Brain–computer interfaces for speech communication. *Speech communication*, 52(4):367–379.

Bruns, A. (2004). Fourier-, hilbert-and wavelet-based signal analysis: are they really different approaches? *Journal of Neuroscience methods*, 137(2):321–332.

Bruns, A. and Eckhorn, R. (2004). Task-related coupling from high-to low-frequency signals among visual cortical areas in human subdural recordings. *International Journal of Psychophysiology*, 51(2):97–116.

Buzsáki, G. (2006). *Rhythms of the Brain*. Oxford University Press.

Buzsáki, G. (2010). Neural syntax: cell assemblies, synapsembles, and readers. *Neuron*, 68(3):362–385.

Byrd, R. H., Lu, P., Nocedal, J., and Zhu, C. (1995). A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208.

Cai, J.-F., Candès, E. J., and Shen, Z. (2010). A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982.

Candès, E. J., Romberg, J., and Tao, T. (2006). Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on information theory*, 52(2):489–509.

Canolty, R. T., Edwards, E., Dalal, S. S., Soltani, M., Nagarajan, S. S., Kirsch, H. E., Berger, M. S., Barbaro, N. M., and Knight, R. T. (2006). High gamma power is phase-locked to theta oscillations in human neocortex. *Science*, 313(5793):1626–1628.

Canolty, R. T. and Knight, R. T. (2010). The functional role of cross-frequency coupling. *Trends in cognitive sciences*, 14(11):506–515.

Cappé, O., Moulines, E., and Rydén, T. (2006). *Inference in hidden Markov models*. Springer Science & Business Media.

Carlstein, E. (1986). The use of subseries values for estimating the variance of a general statistic from a stationary sequence. *The Annals of Statistics*, pages 1171–1179.

Carp, J. (2012). The secret lives of experiments: methods reporting in the fmri literature. *Neuroimage*, 63(1):289–300.

Chalasani, R., Principe, J. C., and Ramakrishnan, N. (2013). A fast proximal method for convolutional sparse coding. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1–5.

Chambers, J. M., Mallows, C. L., and Stuck, B. W. (1976). A method for simulating stable random variables. *Journal of the american statistical association*, 71(354):340–344.

Chan, K. S. and Tong, H. (1986). On estimating thresholds in autoregressive models. *Journal of Time Series Analysis*, 7(3):179–190.

Chang, E. F., Rieger, J. W., Johnson, K., Berger, M. S., Barbaro, N. M., and Knight, R. T. (2010). Categorical speech representation in human superior temporal gyrus. *Nature neuroscience*, 13(11):1428.

Chavez, M., Besserve, M., Adam, C., and Martinerie, J. (2006). Towards a proper estimation of phase synchronization from time series. *Journal of Neuroscience methods*, 154(1):149–160.

Chehelcheraghi, M., van Leeuwen, C., Steur, E., and Chie, N. (2017). A neural mass model of cross frequency coupling. *PLoS ONE 12(4): e0173776.*

Chen, C. W. and So, M. K. (2006). On a threshold heteroscedastic model. *International Journal of Forecasting*, 22(1):73–89.

Chen, R. and Tsay, R. S. (1993). Functional-coefficient autoregressive models. *Journal of the American Statistical Association*, 88(421):298–308.

Chi, T., Ru, P., and Shamma, S. A. (2005). Multiresolution spectrotemporal analysis of complex sounds. *The Journal of the Acoustical Society of America*, 118(2):887–906.

Chib, S. and Greenberg, E. (1995). Understanding the Metropolis-Hastings algorithm. *The American Statistician*, 49(4):327–335.

Cohen, D. (1968). Magnetoencephalography: evidence of magnetic fields produced by alpha-rhythm currents. *Science*, 161(3843):784–786.

Cohen, M. X. (2017). Multivariate cross-frequency coupling via generalized eigendecomposition. *eLife*, 6:e21792.

Cole, S. R., Peterson, E. J., van der Meij, R., de Hemptinne, C., Starr, P. A., and Voytek, B. (2016). Nonsinusoidal oscillations underlie pathological phase-amplitude coupling in the motor cortex in parkinson's disease. *bioRxiv*, page 049304.

Cole, S. R. and Voytek, B. (2017). Brain oscillations and the importance of waveform shape. *Trends in Cognitive Sciences.*

Colgin, L. L., Denninger, T., Fyhn, M., Hafting, T., Bonnevie, T., Jensen, O., Moser, M.-B., and Moser, E. I. (2009). Frequency of gamma oscillations routes flow of information in the hippocampus. *Nature*, 462(7271):353–357.

Dahlhaus, R. (1996). On the Kullback-Leibler information divergence of locally stationary processes. *Stochastic Processes and their Applications*, 62(1):139–168.

Dallérac, G., Graupner, M., Knippenberg, J., Martinez, R. C. R., Tavares, T. F., Tallot, L., El Massioui, N., Verschueren, A., Höhn, S., Bertolus, J. B., et al. (2017). Updating temporal expectancy of an aversive event engages striatal plasticity under amygdala control. *Nature Communications*, 8:13920.

Daubechies, I. (1988). Orthonormal bases of compactly supported wavelets. *Communications on pure and applied mathematics*, 41(7):909–996.

Davis, H., Davis, P. A., Loomis, A. L., Harvey, E. N., and Hobart, G. (1939). Electrical reactions of the human brain to auditory stimulation during sleep. *Journal of Neurophysiology*, 2(6):500–514.

Depireux, D. A., Simon, J. Z., Klein, D. J., and Shamma, S. A. (2001). Spectro-temporal response field characterization with dynamic ripples in ferret primary auditory cortex. *Journal of neurophysiology*, 85(3):1220–1234.

Desoer, C. A. (1970). Slowly varying discrete system xi+1=Ai xi. *Electronics Letters*, 6(11):339–340.

Dijk, D. v., Teräsvirta, T., and Franses, P. H. (2002). Smooth transition autoregressive models–a survey of recent developments. *Econometric reviews*, 21(1):1–47.

Dimitriadis, S. I., Sun, Y., Thakor, N., and Bezerianos, A. (2016). Mining cross-frequency coupling microstates (CFCμstates) from EEG recordings during resting state and mental arithmetic tasks. In *Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the*, pages 5517–5520. IEEE.

Do, M. N. and Vetterli, M. (2005). The contourlet transform: an efficient directional multiresolution image representation. *IEEE Transactions on image processing*, 14(12):2091–2106.

Donoho, D. L. and Johnstone, J. M. (1994). Ideal spatial adaptation by wavelet shrinkage. *biometrika*, 81(3):425–455.

Dupré la Tour, T., Grenier, Y., and Gramfort, A. (2017a). Parametric estimation of spectrum driven by an exogenous signal. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4301–4305.

Dupré la Tour, T., Grenier, Y., and Gramfort, A. (2018a). Driver estimation in non-linear autoregressive models. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Dupré la Tour, T., Moreau, T., Jas, M., and Gramfort, A. (2018b). Multivariate convolutional sparse coding for electromagnetic brain signals. In *Advances in Neural Information Processing Systems (NIPS)*.

Dupré la Tour, T., Tallot, L., Grabot, L., Doyère, V., van Wassenhove, V., Grenier, Y., and Gramfort, A. (2017b). Non-linear auto-regressive models for cross- frequency coupling in neural time series. *PLOS Computational Biology*, 13(12).

Durbin, J. (1960). The fitting of time-series models. *Review of the Int. statistical institute*, pages 233–244.

Dvorak, D. and Fenton, A. A. (2014). Toward a proper estimation of phase–amplitude coupling in neural oscillations. *Journal of Neuroscience methods*, 225:42–56.

Einevoll, G. T., Lindén, H., Tetzlaff, T., Łeski, S., and Pettersen, K. H. (2013). Local field potentials. *Principles of Neural Coding*, page 37.

Elad, M. and Aharon, M. (2006). Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing*, 15(12):3736–3745.

Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica: Journal of the Econometric Society*, pages 987–1007.

Feichtinger, H. G. and Strohmer, T. (2012). *Gabor analysis and algorithms: Theory and applications*. Springer Science & Business Media.

Fell, J. and Axmacher, N. (2011). The role of phase synchronization in memory processes. *Nature reviews. Neuroscience*, 12(2):105.

Févotte, C., Bertin, N., and Durrieu, J.-L. (2009). Nonnegative matrix factorization with the itakura- saito divergence: With application to music analysis. *Neural computation*, 21(3):793–830.

Florin, E. and Baillet, S. (2015). The brain's resting-state activity is shaped by synchronized cross-frequency coupling of neural oscillations. *NeuroImage*, 111:26–35.

Friedman, J., Hastie, T., Höfling, H., and Tibshirani, R. (2007). Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2):302–332.

Fries, P. (2005). A mechanism for cognitive dynamics: neuronal communication through neuronal coherence. *Trends in cognitive sciences*, 9(10):474–480.

Fries, P. (2015). Rhythms for cognition: communication through coherence. *Neuron*, 88(1):220–235.

Fries, P., Reynolds, J. H., Rorie, A. E., and Desimone, R. (2001). Modulation of oscillatory neuronal synchronization by selective visual attention. *Science*, 291(5508):1560–1563.

Fukunaga, K. (2013). *Introduction to statistical pattern recognition*. Elsevier.

Garcia-Cardona, C. and Wohlberg, B. (2017). Convolutional dictionary learning. *arXiv preprint arXiv:1709.02893*.

Gerber, E. M., Sadeh, B., Ward, A., Knight, R. T., and Deouell, L. Y. (2016). Non-sinusoidal activity can produce cross-frequency coupling in cortical signals in the absence of functional interaction between neural sources. *PloS one*, 11(12):e0167351.

Gips, B., Bahramisharif, A., Lowet, E., Roberts, M., de Weerd, P., Jensen, O., and van der Eerden, J. (2017). Discovering recurring patterns in electrophysiological recordings. *J. Neurosci. Methods*, 275:66–79.

Giraud, C., Roueff, F., Sanchez-Perez, A., et al. (2015). Aggregation of predictors for nonstationary sub- linear processes and online adaptive forecasting of time varying autoregressive processes. *The Annals of Statistics*, 43(6):2412–2450.

Godsill, S. and Kuruoglu, E. (1999). Bayesian inference for time series with heavy-tailed symmetric α-stable noise processes. *Proc. Applications of heavy tailed distributions in economics, eng. and stat.*

Gorski, J., Pfeuffer, F., and Klamroth, K. (2007). Biconvex sets and optimization with biconvex functions: a survey and extensions. *Mathematical Methods of Operations Research*, 66(3):373–407.

Gramfort, A., Luessi, M., Larson, E., Engemann, D. A., Strohmeier, D., Brodbeck, C., Goj, R., Jas, M., Brooks, T., Parkkonen, L., et al. (2013). MEG and EEG data analysis with MNE-python. *Frontiers in neuroscience*, 7:267.

Gramfort, A., Luessi, M., Larson, E., Engemann, D. A., Strohmeier, D., Brodbeck, C., Parkkonen, L., and Hämäläinen, M. S. (2014). Mne software for processing meg and eeg data. *Neuroimage*, 86:446–460.

Granger, C. W. (1969). Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: Journal of the Econometric Society*, pages 424–438.

Granger, C. W. (1988). Some recent development in a concept of causality. *Journal of econometrics*, 39(1):199–211.

Grenier, Y. (1983). Time-dependent ARMA modeling of nonstationary signals. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 31(4):899–911.

Grenier, Y. (1984). Modélisation de signaux non-stationnaires.

Grenier, Y. (2013). Estimating an AR model with exogenous driver. Technical Report 2013D007, Telecom ParisTech.

Grenier, Y. and Omnes-Chevalier, M.-C. (1988). Autoregressive models with time-dependent log area ratios. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 36(10):1602–1612.

Grosse, R., Raina, R., Kwong, H., and Ng, A. Y. (2007). Shift-invariant sparse coding for audio classification. In *23rd Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 149–158. AUAI Press.

Haegens, S., Cousijn, H., Wallis, G., Harrison, P. J., and Nobre, A. C. (2014). Inter-and intra-individual variability in alpha peak frequency. *Neuroimage*, 92:46–55.

Haggan, V. and Ozaki, T. (1981). Modelling nonlinear random vibrations using an amplitude-dependent autoregressive time series model. *Biometrika*, 68(1):189–196.

Hamilton, J. D. (1989). A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica: Journal of the Econometric Society*, pages 357–384.

Hari, R. (2006). Action–perception connection and the cortical mu rhythm. *Progress in brain research*, 159:253–260.

Hari, R. and Puce, A. (2017). *MEG-EEG Primer*. Oxford University Press.

Hastie, T., Tibshirani, R., and Wainwright, M. J. (2015). *Statistical Learning with Sparsity*. CRC Press.

Haufe, S., Tomioka, R., Nolte, G., Müller, K.-R., and Kawanabe, M. (2010). Modeling sparse connectivity between underlying brain sources for EEG/MEG. *Bio. Eng., IEEE Trans.*, 57(8):1954–1963.

Heide, F., Heidrich, W., and Wetzstein, G. (2015). Fast and flexible convolutional sparse coding. In *Computer Vision and Pattern Recognition (CVPR)*, pages 5135–5143. IEEE.

Heusser, A. C., Poeppel, D., Ezzyat, Y., and Davachi, L. (2016). Episodic sequence memory is supported by a theta- gamma phase code. *Nature neuroscience*.

Hitziger, S., Clerc, M., Saillet, S., Benar, C., and Papadopoulo, T. (2017). Adaptive waveform learning: A framework for modeling variability in neurophysiological signals. *IEEE Transactions on Signal Processing*.

Holdgraf, C. R., De Heer, W., Pasley, B., Rieger, J., Crone, N., Lin, J. J., Knight, R. T., and Theunissen, F. E. (2016). Rapid tuning shifts in human auditory cortex enhance speech intelligibility. *Nature communications*, 7:13654.

Holdgraf, C. R., Rieger, J. W., Micheli, C., Martin, S., Knight, R. T., and Theunissen, F. E. (2017). Encoding and decoding models in cognitive electrophysiology. *Frontiers in Systems Neuroscience*, 11:61.

Hubel, D. H. and Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106–154.

Huber, P. J. (1981). *Robust Statistics*. Wiley.

Hyafil, A. (2015). Misidentifications of specific forms of cross- frequency coupling: three warnings. *Frontiers in Neuroscience*, 9.

Hyafil, A., Giraud, A.-L., Fontolan, L., and Gutkin, B. (2015). Neural cross-frequency coupling: Connecting architectures, mechanisms, and functions. *Trends in Neurosciences*, 38(11):725–740.

Jachan, M., Matz, G., and Hlawatsch, F. (2007). Time-frequency ARMA models and parameter estimators for underspread nonstationary random processes. *IEEE Transactions on Signal Processing*, 55(9):4366–4381.

Jas, M., Dupré la Tour, T., Şimşekli, U., and Gramfort, A. (2017). Learning the morphology of brain signals using alpha-stable convolutional sparse coding. In *Advances in Neural Information Processing Systems 30 (NIPS)*, pages 1099–1108.

Jasper, H. and Penfield, W. (1949). Electrocorticograms in man: effect of voluntary movement upon the electrical activity of the precentral gyrus. *Archiv für Psychiatrie und Nervenkrankheiten*, 183(1-2):163–174.

Jasper, H. H. (1948). Charting the sea of brain waves. *Science*, 108(2805):343–347.

Jensen, O. and Colgin, L. L. (2007). Cross-frequency coupling between neuronal oscillations. *Trends in cognitive sciences*, 11(7):267–269.

Jensen, O., Spaak, E., and Park, H. (2016). Discriminating valid from spurious indices of phase- amplitude coupling. *eneuro*, pages ENEURO–0334.

Jiang, H., Bahramisharif, A., van Gerven, M. A., and Jensen, O. (2015). Measuring directionality between neuronal oscillations of different frequencies. *Neuroimage*, 118:359–367.

Jirsa, V. and Müller, V. (2013). Cross-frequency coupling in real and virtual brain networks. *Frontiers in computational neuroscience*, 7:78.

Jones, E., Oliphant, T., Peterson, P., et al. (2001). SciPy: Open source scientific tools for Python.

Jones, S. R. (2016). When brain rhythms aren't 'rhythmic': implication for their mechanisms and meaning. *Curr. Opin. Neurobiol.*, 40:72–80.

Jost, P., Vandergheynst, P., Lesage, S., and Gribonval, R. (2006). Motif: an efficient algorithm for learning translation invariant dictionaries. In *Acoustics, Speech and Signal Processing (ICASSP)*, volume 5. IEEE.

Kaplan, R., Bush, D., Bonnefond, M., Bandettini, P. A., Barnes, G. R., Doeller, C. F., and Burgess, N. (2014). Medial prefrontal theta phase coupling during spatial memory retrieval. *Hippocampus*, 24(6):656–665.

Kavukcuoglu, K., Sermanet, P., Boureau, Y.-L., Gregor, K., Mathieu, M., and Le Cun, Y. (2010). Learning convolutional feature hierarchies for visual recognition. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1090–1098.

Kay, K. N., Naselaris, T., Prenger, R. J., and Gallant, J. L. (2008). Identifying natural images from human brain activity. *Nature*, 452(7185):352–355.

Kay, S. M. and Marple, S. L. (1981). Spectrum analysis–a modern perspective. *Proceedings of the IEEE*, 69(11):1380–1419.

Kellis, S., Miller, K., Thomson, K., Brown, R., House, P., and Greger, B. (2010). Decoding spoken words using local field potentials recorded from the cortical surface. *Journal of neural engineering*, 7(5):056007.

Khan, S., Gramfort, A., Shetty, N., M., K., Ganesan, S., Moran, J., Lee, S., Gabrieli, J., Tager-Flusberg, H., Joseph, R., Herbert, M., Hämäläinen, M., and Kenet, T. (2013). Local and long-range functional connectivity is reduced in concert in autism spectrum disorders. *Proc. Natl. Acad. Sci.*

Khodagholy, D., Gelinas, J. N., Thesen, T., Doyle, W., Devinsky, O., Malliaras, G. G., and Buzsáki, G. (2015). Neurogrid: recording action potentials from the surface of the brain. *Nature neuroscience*, 18(2):310–315.

Kikuchi, Y., Attaheri, A., Wilson, B., Rhone, A. E., Nourski, K. V., Gander, P. E., Kovach, C. K., Kawasaki, H., Griffiths, T. D., Howard III, M. A., et al. (2017). Sequence learning modulates neural responses and oscillatory coupling in human and monkey auditory cortex. *PLoS biology*, 15(4):e2000219.

Kramer, M. A., Tort, A. B., and Kopell, N. J. (2008). Sharp edge artifacts and spurious coupling in EEG frequency comodulation measures. *Journal of Neuroscience methods*, 170(2):352–357.

Kuruoglu, E. E. (1999). *Signal processing in α-stable noise environments: a least Lp-norm approach*. PhD thesis, University of Cambridge.

Lachaux, J.-P., Rodriguez, E., Martinerie, J., Varela, F. J., et al. (1999). Measuring phase synchrony in brain signals. *Human brain mapping*, 8(4):194–208.

Lakatos, P., Shah, A. S., Knuth, K. H., Ulbert, I., Karmos, G., and Schroeder, C. E. (2005). An oscillatory hierarchy controlling neuronal excitability and stimulus processing in the auditory cortex. *Journal of neurophysiology*, 94(3):1904–1911.

Leglaive, S., Şimşekli, U., Liutkus, A., Badeau, R., and Richard, G. (2017). Alpha-stable multichannel audio source separation. In *ICASSP*, pages 576–580.

Leonardi, N. and Van De Ville, D. (2015). On spurious and real fluctuations of dynamic functional connectivity during rest. *Neuroimage*, 104:430–436.

Lewicki, M. S. and Sejnowski, T. J. (1999). Coding time-varying signals using sparse, shift- invariant representations. In *Advances in neural information processing systems*, pages 730–736.

Lisman, J. E. and Jensen, O. (2013). The theta-gamma neural code. *Neuron*, 77(6):1002–1016.

Liu, J. (2008). *Monte Carlo strategies in scientific computing*. Springer.

Lozano-Soldevilla, D., ter Huurne, N., and Oostenveld, R. (2016). Neuronal oscillations with non-sinusoidal morphology produce spurious phase-to-amplitude coupling and directionality. *Frontiers in Computational Neuroscience*, 10.

Mahan, M. Y., Chorn, C. R., and Georgopoulos, A. P. (2015). White noise test: detecting autocorrelation and nonstationarities in long time series after ARIMA modeling. In *Proceedings 14th Python in Science Conference (Scipy 2015), Austin, TX*.

Mailhé, B., Lesage, S., Gribonval, R., Bimbot, F., and Vandergheynst, P. (2008). Shift-invariant dictionary learning for sparse representations: extending K-SVD. In *16th Eur. Signal Process. Conf.*, pages 1–5. IEEE.

Mairal, J., Bach, F., Ponce, J., et al. (2014). Sparse modeling for image and vision processing. *Foundations and Trends® in Computer Graphics and Vision*, 8(2-3):85–283.

Mairal, J., Bach, F., Ponce, J., and Sapiro, G. (2010). Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11(Jan):19–60.

Mairal, J., Elad, M., and Sapiro, G. (2008). Sparse representation for color image restoration. *IEEE Transactions on image processing*, 17(1):53–69.

Makhoul, J. (1975). Linear prediction: A tutorial review. *Proceedings of the IEEE*, 63(4):561–580.

Makhoul, J. (1977). Stable and efficient lattice methods for linear prediction. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 25(5):423–428.

Malerba, P. and Kopell, N. (2013). Phase resetting reduces theta–gamma rhythmic interaction to a one-dimensional map. *Journal of mathematical biology*, pages 1–26.

Mallat, S. G. and Zhang, Z. (1993). Matching pursuits with time-frequency dictionaries. *IEEE Transactions on signal processing*, 41(12):3397–3415.

Mandelbrot, B. B. (2013). *Fractals and scaling in finance: Discontinuity, concentration, risk. Selecta volume E.* Springer Science & Business Media.

Marple, L. (1977). Resolution of conventional fourier, autoregressive, and special ARMA methods of spectrum analysis. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'77.*, volume 2, pages 74–77. IEEE.

Martinetz, T., Schulten, K., et al. (1991). A" neural-gas" network learns topologies. *Artificial Neural Networks*, page 397–402.

Mazaheri, A. and Jensen, O. (2008). Asymmetric amplitude modulations of brain oscillations generate slow evoked responses. *The Journal of Neuroscience*, 28(31):7781–7787.

Mesgarani, N., Cheung, C., Johnson, K., and Chang, E. F. (2014). Phonetic feature encoding in human superior temporal gyrus. *Science*, 343(6174):1006–1010.

Moreau, T., Oudre, L., and Vayatis, N. (2018). Dicod: Distributed convolutional sparse coding. In *International Conference on Machine Learning (ICML)*.

Munafò, M. R., Nosek, B. A., Bishop, D. V., Button, K. S., Chambers, C. D., du Sert, N. P., Simonsohn, U., Wagenmakers, E.-J., Ware, J. J., and Ioannidis, J. P. (2017). A manifesto for reproducible science. *Nature Human Behaviour*, 1(1):0021.

Naselaris, T., Kay, K. N., Nishimoto, S., and Gallant, J. L. (2011). Encoding and decoding in fmri. *Neuroimage*, 56(2):400–410.

Nesterov, Y. (2010). Efficiency of coordinate descent methods on huge- scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362.

Nolte, G., Ziehe, A., Krämer, N., Popescu, F., and Müller, K.-R. (2010). Comparison of granger causality and phase slope index. In *NIPS Causality: Objectives and Assessment*, pages 267–276.

Nolte, G., Ziehe, A., Nikulin, V., Schlögl, A., Krämer, N., Brismar, T., and Müller, K.-R. (2008). Robustly estimating the flow direction of information in complex physical systems. *Physical Review Letters*, 100:234101.

Nutini, J., Schmidt, M., Laradji, I. H., Friedlander, M. P., and Koepke, H. (2015). Coordinate descent converges faster with the gauss- southwell rule than random selection. In *International Conference on Machine Learning (ICML)*, pages 1632–1641.

Ogawa, S., Lee, T.-M., Kay, A. R., and Tank, D. W. (1990). Brain magnetic resonance imaging with contrast dependent on blood oxygenation. *Proceedings of the National Academy of Sciences*, 87(24):9868–9872.

O'Keefe, J. and Dostrovsky, J. (1971). The hippocampus as a spatial map: Preliminary evidence from unit activity in the freely-moving rat. *Brain research*.

Olshausen, B. A. and Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607.

Osher, S. and Li, Y. (2009). Coordinate descent optimization for ℓ_1 minimization with application to compressed sensing; a greedy algorithm. *Inverse Problems and Imaging*, 3(3):487–503.

Osipova, D., Hermes, D., and Jensen, O. (2008). Gamma power is phase-locked to posterior alpha activity. *PloS one*, 3(12):e3990.

Özkurt, T. E. and Schnitzler, A. (2011). A critical note on the definition of phase–amplitude cross-frequency coupling. *Journal of Neuroscience methods*, 201(2):438–443.

Pachitariu, M., Packer, A. M., Pettit, N., Dalgleish, H., Hausser, M., and Sahani, M. (2013). Extracting regions of interest from biological images with convolutional sparse block coding. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1745–1753.

Park, H., Ince, R. A., Schyns, P. G., Thut, G., and Gross, J. (2015). Frontal top-down signals increase coupling of auditory low-frequency oscillations to continuous speech in human listeners. *Current Biology*, 25(12):1649–1653.

Pasley, B. N., David, S. V., Mesgarani, N., Flinker, A., Shamma, S. A., Crone, N. E., Knight, R. T., and Chang, E. F. (2012). Reconstructing speech from human auditory cortex. *PLoS Biol*, 10(1):e1001251.

Penny, W., Duzel, E., Miller, K., and Ojemann, J. (2008). Testing for nested oscillation. *Journal of Neuroscience methods*, 174(1):50–61.

Pisarenko, V. F. (1973). The retrieval of harmonics from a covariance function. *Geophysical Journal International*, 33(3):347–366.

Quiroga, R. Q., Kraskov, A., Kreuz, T., and Grassberger, P. (2002). Performance of different synchronization measures in real data: a case study on eeg signals. *Physical Review E*, 65(4):041903.

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Recht, B., Fazel, M., and Parrilo, P. A. (2010). Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review*, 52(3):471–501.

Richtárik, P. and Takáč, M. (2014). Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(1-2):1–38.

Roux, F., Wibral, M., Singer, W., Aru, J., and Uhlhaas, P. J. (2013). The phase of thalamic alpha activity modulates cortical gamma-band activity: evidence from resting- state MEG recordings. *Journal of Neuroscience*, 33(45):17827–17835.

Rugh, W. J. (1996). *Linear system theory*, volume 2. prentice hall Upper Saddle River, NJ.

Rukat, T., Baker, A., Quinn, A., and Woolrich, M. (2016). Resting state brain networks from EEG: Hidden Markov states vs. classical microstates. *arXiv preprint arXiv:1606.02344*.

Samorodnitsky, G. and Taqqu, M. S. (1994). *Stable non-Gaussian random processes: stochastic models with infinite variance*, volume 1. CRC press.

Schmidt, R. (1986). Multiple emitter location and signal parameter estimation. *IEEE transactions on antennas and propagation*, 34(3):276–280.

Schreiber, T. (2000). Measuring information transfer. *Physical review letters*, 85(2):461.

Schwarz, G. et al. (1978). Estimating the dimension of a model. *Ann. Stat.*, 6(2):461–464.

Shannon, C. E. and Weaver, W. (1963). The mathematical theory of communication. 1949. *Urbana, IL: University of Illinois Press.*

Shirvalkar, P. R., Rapp, P. R., and Shapiro, M. L. (2010). Bidirectional changes to hippocampal theta–gamma comodulation predict memory for recent spatial episodes. *Proceedings of the National Academy of Sciences*, 107(15):7054–7059.

Simoncelli, E. P., Freeman, W. T., Adelson, E. H., and Heeger, D. J. (1992). Shiftable multiscale transforms. *IEEE transactions on Information Theory*, 38(2):587–607.

Şimşekli, U., Liutkus, A., and Cemgil, A. T. (2015). Alpha-stable matrix factorization. *IEEE SPL*, 22(12):2289–2293.

Šorel, M. and Šroubek, F. (2016). Fast convolutional sparse coding using matrix inversion lemma. *Digital Signal Processing.*

Spiridonakos, M. and Fassois, S. (2014). Non-stationary random vibration modelling and analysis via functional series time-dependent ARMA (FS-TARMA) models–a critical survey. *Mechanical Systems and Signal Processing*, 47(1):175–224.

Spyers-Ashby, J., Bain, P., and Roberts, S. (1998). A comparison of fast fourier transform (FFT) and autoregressive (AR) spectral estimation techniques for the analysis of tremor data. *Journal of neuroscience methods*, 83(1):35–43.

Starck, J.-L., Candès, E. J., and Donoho, D. L. (2002). The curvelet transform for image denoising. *IEEE Transactions on image processing*, 11(6):670–684.

Sweeney-Reed, C. M., Zaehle, T., Voges, J., Schmitt, F. C., Buentjen, L., Kopitzki, K., Esslinger, C., Hinrichs, H., Heinze, H.-J., Knight, R. T., et al. (2014). Corticothalamic phase synchrony and cross-frequency coupling predict human memory formation. *Elife*, 3:e05352.

Taylor, J. R., Williams, N., Cusack, R., Auer, T., Shafto, M. A., Dixon, M., Tyler, L. K., Henson, R. N., et al. (2017). The cambridge centre for ageing and neuroscience (cam-can) data repository: structural and functional mri, meg, and cognitive data from a cross-sectional adult lifespan sample. *Neuroimage*, 144:262–269.

Theunissen, F. E., Sen, K., and Doupe, A. J. (2000). Spectral-temporal receptive fields of nonlinear auditory neurons obtained using natural sounds. *Journal of Neuroscience*, 20(6):2315–2331.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288.

Tong, H. (2011). Threshold models in time series analysis—30 years on. *Statistics and its Interface*, 4(2):107–118.

Tong, H. and Lim, K. S. (1980). Threshold autoregression, limit cycles and cyclical data. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 245–292.

Tort, A. B., Komorowski, R., Eichenbaum, H., and Kopell, N. (2010). Measuring phase-amplitude coupling between neuronal oscillations of different frequencies. *J. Neurophysiol.*, 104(2):1195–1210.

Tort, A. B., Kramer, M. A., Thorn, C., Gibson, D. J., Kubota, Y., Graybiel, A. M., and Kopell, N. J. (2008). Dynamic cross-frequency couplings of local field potential oscillations in rat striatum and hippocampus during performance of a t-maze task. *Proc. Natl. Acad. Sci.*, 105(51):20517–20522.

Tort, A. B., Rotstein, H. G., Dugladze, T., Gloveli, T., and Kopell, N. J. (2007). On the formation of gamma-coherent cell assemblies by oriens lacunosum-moleculare interneurons in the hippocampus. *Proceedings of the National Academy of Sciences*, 104(33):13490–13495.

Trefethen, L. N. and Bau III, D. (1997). *Numerical linear algebra*, volume 50. Siam.

Tuomisto, T., Hari, R., Katila, T., Poutanen, T., and Varpula, T. (1983). Studies of auditory evoked magnetic and electric responses: Modality specificity and modelling. *Il Nuovo Cimento D*, 2(2):471–483.

Valdés-Sosa, P. A., Sánchez-Bornot, J. M., Lage-Castellanos, A., Vega-Hernández, M., Bosch-Bayard, J., Melie-García, L., and Canales-Rodríguez, E. (2005). Estimating brain functional connectivity with sparse multivariate autoregression. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 360(1457):969–981.

van Ede, F., Quinn, A. J., Woolrich, M. W., and Nobre, A. C. (2018). Neural oscillations: Sustained rhythms or transient burst-events? *Trends in Neurosciences*.

Van Essen, D. C., Smith, S. M., Barch, D. M., Behrens, T. E., Yacoub, E., Ugurbil, K., Consortium, W.-M. H., et al. (2013). The wu-minn human connectome project: an overview. *Neuroimage*, 80:62–79.

van Wijk, B., Jha, A., Penny, W., and Litvak, V. (2015). Parametric estimation of cross-frequency coupling. *Journal of neuroscience methods*, 243:94–102.

Vaz, A. P., Yaffe, R. B., Wittig, J. H., Inati, S. K., and Zaghloul, K. A. (2017). Dual origins of measured phase-amplitude coupling reveal distinct neural mechanisms underlying episodic memory in the human cortex. *Neuroimage*, 148:148–159.

Vidaurre, D., Abeysuriya, R., Becker, R., Quinn, A. J., Alfaro-Almagro, F., Smith, S. M., and Woolrich, M. W. (2017). Discovering dynamic brain networks from big data in rest and task. *NeuroImage*.

Vidaurre, D., Quinn, A. J., Baker, A. P., Dupret, D., Tejero-Cantero, A., and Woolrich, M. W. (2016). Spectrally resolved fast transient brain states in electrophysiological data. *Neuroimage*, 126:81–95.

Voytek, B., Canolty, R. T., Shestyuk, A., Crone, N., Parvizi, J., and Knight, R. T. (2010). Shifts in gamma phase–amplitude coupling frequency from theta to alpha over posterior cortex during visual tasks. *Frontiers in human neuroscience*, 4:191.

Wakita, H. (1973). Direct estimation of the vocal tract shape by inverse filtering of acoustic speech waveforms. *IEEE Transactions on Audio and Electroacoustics*, 21(5):417–427.

Wang, W., Degenhart, A. D., Sudre, G. P., Pomerleau, D. A., and Tyler-Kabara, E. C. (2011). Decoding semantic information from human electrocorticographic (ecog) signals. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, pages 6294–6298. IEEE.

Wang, Y., Qi, Y., Wang, Y., Lei, Z., Zheng, X., and Pan, G. (2016). Delving into α-stable distribution in noise suppression for seizure detection from scalp EEG. *J. Neural. Eng.*, 13(5):056009.

Welch, P. (1967). The use of fast fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms. *IEEE Transactions on audio and electroacoustics*, 15(2):70–73.

Wibral, M., Pampu, N., Priesemann, V., Siebenhühner, F., Seiwert, H., Lindner, M., Lizier, J. T., and Vicente, R. (2013). Measuring information-transfer delays. *PloS one*, 8(2):e55809.

Wilkinson, J. H. (1984). The perfidious polynomial. *Studies in numerical analysis*, 24:1–28.

Wilks, S. S. (1938). The large-sample distribution of the likelihood ratio for testing composite hypotheses. *The Annals of Mathematical Statistics*, 9(1):60–62.

Wohlberg, B. (2016a). Convolutional sparse representation of color images. In *IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI)*, pages 57–60.

Wohlberg, B. (2016b). Efficient algorithms for convolutional sparse representations. *Image Processing, IEEE Transactions on*, 25(1):301–315.

Woodbury, M. A. (1950). Inverting modified matrices. *Memorandum report*, 42(106):336.

Woolrich, M. W., Baker, A., Luckhoo, H., Mohseni, H., Barnes, G., Brookes, M., and Rezek, I. (2013). Dynamic state allocation for MEG source reconstruction. *Neuroimage*, 77:77–92.

Wright, S. and Nocedal, J. (1999). *Numerical optimization*, volume 35. Springer Science.

Wu, S. and Chen, R. (2007). Threshold variable determination and threshold variable driven switching autoregressive models. *Statistica Sinica*, 17(1):241.

Zeiler, M. D., Krishnan, D., Taylor, G., and Fergus, R. (2010). Deconvolutional networks. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2528–2535. IEEE.

université PARIS-SACLAY
ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre: Modèles non-linéaires pour les séries temporelles neurophysiologiques

Mots clés: Séries temporelles – modélisation – autoregressive – Codage parcimonieux convolutionnel – couplage cross-fréquence – multivarié – encodage – neurophysiologie – magnéto-encéphalographie

Résumé: Dans les séries temporelles neurophysiologiques, on observe de fortes oscillations neuronales, et les outils d'analyse sont donc naturellement centrés sur le filtrage à bande étroite. Puisque cette approche est trop réductrice, nous proposons de nouvelles méthodes pour représenter ces signaux. Nous centrons tout d'abord notre étude sur le couplage phase-amplitude (PAC), dans lequel une bande haute fréquence est modulée en amplitude par la phase d'une oscillation neuronale plus lente. Nous proposons de capturer ce couplage dans un modèle probabiliste appelé modèle autoregressif piloté (DAR). Cette modélisation permet une sélection de modèle efficace grâce à la mesure de vraisemblance, ce qui constitue un apport majeur à l'estimation du PAC. Nous présentons différentes paramétrisations des modèles DAR et leurs algorithmes d'inférence rapides, et discutons de leur stabilité. Puis nous montrons comment utiliser les modèles DAR pour l'analyse du PAC, et démontrons l'avantage de l'approche par modélisation avec trois jeux de donnée. Puis nous explorons plusieurs extensions à ces modèles, pour estimer le signal pilote à partir des données, le PAC sur des signaux multivariés, ou encore des champs réceptifs spectro-temporels. Enfin, nous proposons aussi d'adapter les modèles de codage parcimonieux convolutionnels pour les séries temporelles neurophysiologiques, en les étendant à des distributions à queues lourdes et à des décompositions multivariées. Nous développons des algorithmes d'inférence efficaces pour chaque formulations, et montrons que l'on obtient de riches représentations de façon non-supervisée.

Title: Non-linear models for neurophysiological time series

Keywords: Time series – modeling – autoregressive – convolutional sparse coding – cross-frequency coupling – multivariate – encoding – neurophysiology – magneto-encephalography

Abstract: In neurophysiological time series, strong neural oscillations are observed in the mammalian brain, and the natural processing tools are thus centered on narrow-band linear filtering. As this approach is too reductive, we propose new methods to represent these signals. We first focus on the study of phase-amplitude coupling (PAC), which consists in an amplitude modulation of a high frequency band, time-locked with a specific phase of a slow neural oscillation. We propose to use driven autoregressive models (DAR), to capture PAC in a probabilistic model. Giving a proper model to the signal enables model selection by using the likelihood of the model, which constitutes a major improvement in PAC estimation. We first present different parametrization of DAR models, with fast inference algorithms and stability discussions. Then, we present how to use DAR models for PAC analysis, demonstrating the advantage of the model-based approach on three empirical datasets. Then, we explore different extensions to DAR models, estimating the driving signal from the data, PAC in multivariate signals, or spectro-temporal receptive fields. Finally, we also propose to adapt convolutional sparse coding (CSC) models for neurophysiological time-series, extending them to heavy-tail noise distribution and multivariate decompositions. We develop efficient inference algorithms for each formulation, and show that we obtain rich unsupervised signal representations.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France