Applied Statistics Project

Denoising physiological signals using greedy methods

Authors :
Alain Soltani, Loïc Tudela
Adil Salim, Laurent Lambert

Supervisors :
Alexandre Gramfort, Joseph Salmon

May 2014

ENSAE ParisTech

Abstract

One recurrent problem in physiological signal acquisition is the presence of additive noise, inherent to the experiment ; subject movements, devices whiffs all deteriorate the signal we intend to study. Our aim here is to develop efficient denoising algorithms, to extract clear signals from noisy electroencephalographic, magnetoencephalographic measurements.

We use in this study a famous signal processing algorithm : the *Matching Pursuit* algorithm, based on iterative research of highest correlations between the original signal and a dictionary of explanatory functions ; retained projections overlap to form a signal properly describing the original data.
We present a procedure for explanatory variables' selection - and particularly its stopping criterion - by drawing comparisons between our iterative algorithm and elementary regression models. We test the robustness of our model by various means - by ensuring that extracted noise is consistent with the assumptions of normality issued, and that our algorithm resists to simulated noise additions.

Various improvements are then detailed ; from additional orthogonal projections on the selected atoms (*Orthogonal Matching Pursuit*) to dictionary orthonormalization, we present successively the effects on our results in statistical and computational terms. Similitudes with classical signal processing tools, such as hard thresholding, are drawn. We finish by extending our work to the case of multichannel acquisitions, and developing a method for selecting the best dictionary.

Contents

Chapter 1

Problem definition

1.1 Introduction to functional neuroimaging methods

The past fifteen years have seen great progress in brain imaging methods ; the more precise the image of the brain, the better diagnosis from scientists and physicists, especially in areas such as epileptic reactions or tumors, where rapid and reliable information is a necessity.

Modern techniques presented in the following have the advantage of offering a wide temporal resolution - below 100 ms ; this allows us to conduct a very fine temporal analysis of the phenomena involved.

In this section, we will discuss two fundamental functional brain imaging methods, which usage ranges from comprehension of basic cognitive mechanisms to characterization of pathologies : *electroencephalography* (EEG) and *magnetoencephalography* (MEG). They both localize neural electrical activity using noninvasive measurements of external electromagnetic signals.

Electroencephalography is the recording of electrical activity along the scalp, by placing an electrode helmet on the patient. Electrically-charged neurons exchange ions with the extracellular milieu, and eventually reach the electrodes on the patient's scalp, via an effect volume conduction. These ionic current flows create voltage variations which, once recorded, form the EEG.

In clinical contexts, EEG refers to the recording of the brain's spontaneous electrical activity over a short period of time, usually 20–40 minutes, as recorded from multiple electrodes placed on the scalp.

Magnetoencephalography records magnetic fields produced by the aforementioned ionic currents ; in accordance with the Maxwell-Faraday equation, the previous electrical current will produce an orthogonal magnetic field, that we measure using very sensitive magnetometers.
The entire brain can be thought of as a current dipole, producing currents with a position, orientation, and magnitude, but no spatial extent. According to the right-hand rule, a current dipole gives rise to a magnetic field that flows around the axis of its vector component.

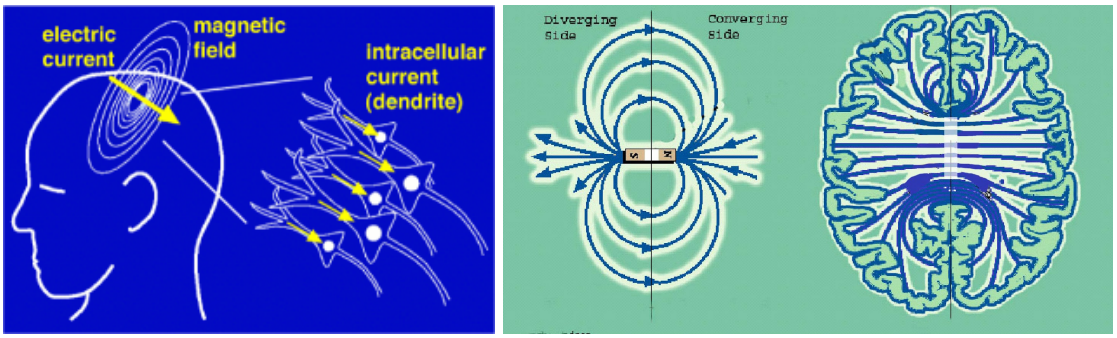

FIGURE 1.1: LEFT : *Dendrites propagate the ionic flow. This generates an electric current, and inductively, a magnetic field.*
RIGHT : *Brain as a dipole.*

It should be noted that these magnetic fields are weak. To generate a detectable signal, approximately 50,000 active neurons are needed. At 10^{-14} Tesla for cortical activity, the brain produces a considerably smaller field than the ambient magnetic noise in an urban environment - on the order of 10^{-7} T - or the terrestrial magnetic field - 47 μT.

We reach here a critical problem of magnetoencephalography : data cannot be surely acquired without any artifacts. As a matter of fact, physiological data can be corrupted by additive noise including, among others, background brain activity, electrical heart activity, eye-blinks and other electrical muscle activities.
As we shall see later, physiological signal denoising is a difficult but essential task. It involves several signal processing tools, as well as appropriate statistical criteria.
Let us now give a closer look at the data we manipulate.

1.2 Sample dataset & programming framework

Data sources, experimental protocol
We use in this study an extract from a sample dataset of recordings from one subject with combined MEG and EEG conducted at the Martinos Center of Massachusetts General Hospital.

In the experiment, auditory stimuli (delivered monaurally to the left or right ear) and visual stimuli (shown in the left or right visual hemifield) were presented in a random sequence.
To control for subject's attention, a smiley face was presented intermittently and the subject was asked to press a button upon its appearance.

Experiment phases
Our measurements include three phases of equal length :

- a pre-experiment phase (samples 0 to 180) : in which the subject is in idle state ;
- a experiment phase (samples 180 to 436) : consisting of the previously described stimuli ;
- a post-experiment phase (remaining samples) : the subject returns to the idle state.

Data format, physical measurements
Original data was acquired with a Neuromag VectorView MEG system with 306 sensors arranged in 102 triplets.
Our data has been restricted to MEG acquisitions from 203 sensor signals, sampled at 540 distincts moments on a 300-ms basis. Signal measurement and sampling times are stored in two arrays for the entire experiment.

As previously mentioned, the distribution of the brain's electric charges can be modeled as a current dipole ; thus the generated magnetic field presents a symmetry of revolution around the dipole axis. we put ourselves in a plane a plane passing through the axis of symmetry, and study the field in two orthogonal components.
For this reason, data is organized pair-wise ; odd and even sensors correspond to increases in these two orthogonal directions. Therefore, there is generally a great correlation between two successive sensor measurements of the same parity, but not between two successive sensors.

To avoid approximation error induced by the numerical methods of calculation, sampled data has been increased by a factor of 10^{12}.

Programming framework
The entire project was programmed in Python. It is available at the following address : https://github.com/Parveez/Stat_Appliquee.
All wavelet decompositions were built using Python wavelet library PyWavelets. Statistical tests were made using either classical NumPy features or statistical modeling and econometrics library StatsModels.

1.3 Single-channel model

1.3.1 Inverse linear problem

In the following, we consider, for each signal acquired from a unique sensor, the linear inverse problem :

$$y = D\lambda + \epsilon, \tag{1.1}$$

$y \in \mathbb{R}^n$ being the measured M/EEG signal, sampled at times $t_1, ...t_n$,
$D \in \mathcal{M}_{n,q}(\mathbb{R})$ the acquisition matrix and ϵ an error term.

Matrix D represents our explanatory dictionary ; its columns, seen as dictionary atoms, generate a subspace of $\mathbb{R}^n$ of dimension q. Sparse vector λ nonzero coefficients relates explanatory power of corresponding atoms.
D is typically a discrete signal processing transform, such as Discrete Wavelet Transform (DWT).[1] This ensures that D is injective (i.e. of rank q) ; thus $D^T D$ is invertible, and the number of coefficients is greater than the matrix rank : $n > q$.

[1] To see how to create a dictionary matrix from a wavelet transform, see Appendix A, subsection A.3.2.

1.3.2 Assumptions

In order to make classical econometric techniques such as ordinary least squares (OLS) applicable, we set our framework by assuming the following statements :

1. *Strict exogeneity.* The errors in the regression should have conditional mean zero : $\mathbb{E}[\epsilon|D] = 0$.
2. *No linear dependence.* The regressors in D must all be linearly independent. This means that the dictionary must have full column rank almost surely: $\mathbb{P}(\text{rank}(D) = q) = 1$.
3. *Spherical errors :* $\mathbb{V}[\epsilon|D] = \sigma^2 I_n$, where σ^2 is a parameter which determines the variance of each observation. If this assumption is violated, OLS estimates are still valid, but no longer efficient. It is customary to split this assumption into two parts :
   - *Homoscedasticity :* $\forall i \in \langle 1; n\rangle, \mathbb{E}[\epsilon_i^2|D] = \sigma^2$.
   - *Nonautocorrelation :* $\mathbb{E}[\epsilon_i \epsilon_j|D] = 0, i \neq j$.

   For further demonstrations - such as the tests involved in the atoms' selection process -, we need more *stringent* assumptions than 3. Therefore we can *replace* the latter by the more restrictive hypothesis :
4. *Error normality.* We assume that the errors have normal distribution conditional on the regressors : $\epsilon|D \sim \mathcal{N}(0, \sigma^2 I_n)$.

These are rather plausible assumptions : one can easily assume that the additive noise, such as eye-blink or environmental noise, is *i.i.d.*[2] In addition, our dictionaries corresponding to discrete signal transforms, the linear independence assumption is altogether logical.

The error variance parameter - representing at first glance a indication of the quality of our denoising - raises a true problematic : whether it should be seen as known or to be estimated. Hence we will present the two approaches :
· Firstly, a known-variance model, using the pre-experiment variance as the accurate value. In this case, the underlying assumption is that fields measured during the first (pre-experiment) phase are equivalent to the noise present in the second (actual experiment) phase.
· Secondly, we will present a more sophisticated model, that estimates the variance via the experiment signal.
It belongs to the experimenter to choose the preferred approach, depending on the expected error level and the available data.

1.3.3 OLS estimators

Using Assumption 2, $D^T D$ is inversible and the classic OLS estimator of λ can be written matrix-wise :

$$\hat{\lambda} = (D^T D)^{-1} D^T y \tag{1.2}$$

[2]However, a more developed model might take into account the possibility that the residues are heteroscedastic ; this goes beyond the scope of our study.

After estimating λ, the fitted values (or predicted values) from the regression becomes :

$$\hat{y} = D\hat{\lambda} = P_D y, \tag{1.3}$$

where P_D is the projection matrix onto the space spanned by the columns of D. We also define the annilihator matrix $M_D = I - P_D$; both matrices are symmetric and idempotent.
Hence we obtain the residuals from regression :

$$\hat{\epsilon} = y - D\hat{\lambda} = M_D y = M_D \epsilon \tag{1.4}$$

Using these residuals, we can estimate the value of σ^2 :

$$s^2 = \frac{\hat{\epsilon}^T \hat{\epsilon}}{n-q} = \frac{y^T M_D y}{n-q}, \tag{1.5}$$

where $n-q$ is number of degrees of freedom. Under previous assumptions, $\hat{\lambda}$ and s^2 are unbiased estimators of λ, σ^2 respectively.

Variance-covariance matrix of $\hat{\lambda}$ is equal to :

$$\mathbb{V}[\hat{\lambda}|D] = \sigma^2 (D^T D)^{-1} \tag{1.6}$$

The standard error of each coefficient $\hat{\lambda_i}$ becomes :

$$\text{s.d.}(\hat{\lambda_i}) = \sqrt{\sigma^2 [(D^T D)^{-1}]_{i,i}} \tag{1.7}$$

and can be estimated by :

$$\widehat{\text{s.e.}}(\hat{\lambda_i}) = \sqrt{s^2 [(D^T D)^{-1}]_{i,i}}. \tag{1.8}$$

1.4 Multi-channel model

The multi-channel model can be seen as analogous to the previous inverse linear problem, except the manipulated objects are no longer vectors but matrices ; as a matter of fact, one can build the following concatenated inverse problem :

$$Y = D\Lambda + E, \tag{1.9}$$

$Y = [y_1 ... y_r] \in \mathcal{M}_{n,r}(\mathbb{R})$ being a concatenation of the r measured M/EEG vectors to be denoised, sampled at times $t_1, ... t_n$, $D \in \mathcal{M}_{n,q}(\mathbb{R})$ the previous acquisition matrix, $\Lambda \in \mathcal{M}_{q,r}(\mathbb{R})$ sparse matrix and $E \in \mathcal{M}_{n,r}(\mathbb{R})$ a concatenated error term.

All assumptions previously made for the single-channel model remain valid for each single-channel subproblem of (1.8).

Chapter 2

Greedy algorithms

This chapter applies, unless stated otherwise, to the single-channel model. We will introduce algorithms and tests that will be generalized to the multi-channel model (Chapter 4).

2.1 Introduction

In the following, we define the l^0 pseudo-norm as :

$$\|\lambda\|_0 = \text{Card}\{i \in \{1; q\} \, | \lambda_i \neq 0\}.$$

Our goal is to minimize the distance between the observation y and the reconstructed signal $D\lambda$, under a sparsity constraint - as we want as few as possible non-zero coefficients in λ.
To recover an approximation of the signal, we use l^0-sparse optimization, which requires to solve either the sparsity constraint problem :

$$\min_{\|\lambda\|_0 \leqslant s} \|y - D\lambda\| \tag{2.1}$$

or the dual problem, called l^2-error constraint problem, of minimizing the number of non-zero coefficients under the constraint that denoised signal should be close enough to the original acquisition :

$$\min_{\|y - D\lambda\| \leqslant \eta} \|\lambda\|_0 \tag{2.2}$$

One needs to set-up either the sparsity parameter s or the error constraint η. These parameters should be tuned in accordance to the noise level, and when there is no noise - i.e. $\epsilon = 0$ - one should set $\eta = 0$, and solve the simplified problem :

$$\min_{y = D\lambda} \|\lambda\|_0 \, . \tag{2.3}$$

Note these are non-convex optimization problems ; this does not guarantee to find the global optimum.

An efficient strategy to solve these problems is to use heuristic greedy optimization methods, as the Matching Pursuit (MP) algorithms. We describe in this chapter the classical MP algorithm, and discuss the quality and robustness of our results.

2.2 Matching Pursuit

2.2.1 Classical approach

First introduced by S. Mallat and S. Zhang [1], the Matching Pursuit algorithm is a greedy procedure that progressively identifies the location of the "spikes" by looking at dictionary atoms that are maximally correlated with the current residual.

The problem can be stated as follows : given our original signal y and our dictionary D, we intend to approach y using a linear combination of D columns - i.e. selecting "significant" atoms in D.

The MP algorithm offers a simple approach :
· We initially set $R^{(0)}y = y$;
· At stage l, we find the atom $D_{i_0} \in \mathcal{M}_{n,1}(\mathbb{R})$ maximally correlated with residual $R^{(l)}y$:

$$D_{i_0} = \underset{D_i \in D}{\operatorname{argmax}} |\langle R^{(l)}y, D_i \rangle|. \tag{2.4}$$

· Then we subtract to $R^{(l)}y$ its projection on D_{i_0} to form $R^{(l+1)}y$:

$$R^{(l+1)}y = R^{(l)}y - \langle R^{(l)}y, D_{i_0} \rangle D_{i_0} \tag{2.5}$$

until the linear combination approaches well enough our signal.

Data: Original noisy measurement y.
Result: Denoised signal $D\lambda$.
Initialization: $\lambda^{(0)} = 0$;
while *signal approximation not efficient enough* **do**
  $\lambda^{(l+1)} = \lambda^{(l)} + \mu$
  with μ 1-sparse vector minimizing the error $\min_{\|\mu\|_0 = 1} \left\| y - D(\lambda^{(l)} + \mu) \right\|$.
end

Algorithm 1: Classic Matching Pursuit.

The current algorithm can be re-written as follows : at each step, we compute all the correlations between the atoms of D and the residual. We select the index of maximal correlation, by doing as follows :

Data: Original noisy measurement y.
Result: Denoised signal $D\lambda$.
Initialization: $\lambda^{(0)} = 0$;
while *signal approximation not efficient enough* **do**
  $[\lambda^{(l+1)}]_{i_0} = [\lambda^{(l)}]_{i_0} + [c]_{i_0}$
  with
  $c = D^T(y - D\lambda^{(l)})$ correlation vector, $i_0 = \underset{i}{\text{argmax}}|c_i|$.
end

Algorithm 2: Classic Matching Pursuit, matrix-wise.

One can show that the error $\left\|y - D\lambda^{(l)}\right\|$ converges toward zero when the number of atoms included in the model l increases. However, including too many atoms in our model may imply fitting noise in our signal approximation.
We discuss in the following section the usage of a statistical criterion to select the right number of atoms.

2.2.2 Towards a statistical criterion

As mentioned earlier, one can draw several comparisons between this Matching Pursuit algorithm and a classic linear regression model. This is especially helpful when it comes to controlling explanatory variables in our model.

As we iteratively add 1-sparse elements to vector λ, we are entitled to ask ourselves when to stop this inclusion ; in other words, when to stop adding explanatory variables to our model, in order to achieve a good and concise approximation of our signal y.

This amounts to testing, at each stage, the true value of this sparse vector λ ; more specifically, we test the true value of each newly-added coefficient via MP : whether it should be null - the associated column of D should not be part of the approximation signal $D\lambda$ - or not - the added explanatory variable is significant in our model.

For this, we used the following procedure :

- we add the new atom to a temporary dictionary, that contains only previously-selected atoms.
- we test, using classical econometric tests, the significance of the corresponding new coefficient in the sparse vector λ ; if the null hypothesis is not rejected (i.e. the new regressor isn't significant), we relegate it to noise.

In particular, our model being linear, we rely on OLS estimators to perform our tests. These are analogous to the parametric Wald test.

We denote by $D \in \mathcal{M}_{n,q}(\mathbb{R})$ our hypothetically-redundant dictionary, and $X \in \mathcal{M}_{n,l}(\mathbb{R})$ the temporary dictionary formed by selected columns of D via the first l MP steps.
We aim at testing significance of atom $x \in \mathcal{M}_{n,1}(\mathbb{R})$ once added to our model : we denote $Z = [X\ x]$ the eventual dictionary at step $l + 1$.

We assume X^TX is inversible - as D^T represents a discrete signal processing transform, X^T will be injective and thus the assumption is reasonable.

– Stage l : our statistical model goes as follows, with ϵ_l error term at stage l :

$$y = X\beta^{(l)} + \epsilon^{(l)}. \tag{2.6}$$

Note that we here use $\beta^{(l)}$, a l-size vector only formed of significant coefficients, instead of sparse vector $\lambda^{(l)}$. Both vectors are linked by the equation $X\beta^{(l)} = D\lambda^{(l)}$.

– Stage $l+1$: similarly, one can obtain :

$$y = Z\beta^{(l+1)} + \epsilon^{(l+1)} = Z\begin{pmatrix} \beta^{(l)} \\ \gamma \end{pmatrix} + \epsilon^{(l+1)}, \tag{2.7}$$

Using (1.2), OLS estimator of new vector $\beta^{(l+1)}$ becomes :

$$\begin{pmatrix} \hat{\beta}^{(l)} \\ \hat{\gamma} \end{pmatrix} = (Z^TZ)^{-1}Z^Ty = (Z^TZ)^{-1}\begin{pmatrix} X^Ty \\ x^Ty \end{pmatrix} \tag{2.8}$$

We test the null hypothesis $\mathcal{H}_0 : \gamma = 0$ against the two-sided alternative $\mathcal{H}_1 : \gamma \neq 0$. Conditionally on X, the general expression of our test statistic is :

$$T_{n,l+1} = \frac{\hat{\gamma}}{\text{s.d.}(\hat{\gamma})} \sim \mathcal{N}(0,1) \text{ under } \mathcal{H}_0. \tag{2.9}$$

Recall we work here with a known-variance model, explaining the asymptotic normality of $T_{n,l+1}$; an improved model using endogenous variance will be detailed later.

Considering $\text{s.d.}(\hat{\beta}_i) = \sqrt{\sigma^2[(Z^TZ)^{-1}]_{i,i}}$ and equation (2.8):

$$T_{n,l+1} = \frac{[(Z^TZ)^{-1}Z^Ty]_{l+1,1}}{\sqrt{\sigma^2[(Z^TZ)^{-1}]_{l+1,l+1}}} \tag{2.10}$$

This can be written equivalently :

$$T_{n,l+1} = \frac{(0_l\ 1)(Z^TZ)^{-1}Z^Ty}{\sqrt{\sigma^2(0_l\ 1)(Z^TZ)^{-1}\begin{pmatrix} 0_l \\ 1 \end{pmatrix}}} \tag{2.11}$$

One can solve the systems
$(Z^TZ)\begin{pmatrix} \beta \\ \gamma \end{pmatrix} = \begin{pmatrix} 0_l \\ 1 \end{pmatrix}$ and $(Z^TZ)\begin{pmatrix} \beta \\ \gamma \end{pmatrix} = \begin{pmatrix} X^Ty \\ x^Ty \end{pmatrix}$
to obtain the final, most practical expression of the statistic :

$$T_{n,l+1} = \frac{x^T(y - P_Xy)}{\sigma\sqrt{\|x\|_2^2 - x^TP_Xx}} \tag{2.12}$$

with $P_X = X(X^TX)^{-1}X^T$ projection matrix onto the space spanned by the columns of the temporary dictionary X, as defined in 1.3.3.

Region of rejection for hypothesis $\mathcal{H}_0$ becomes $\{|T_{n,l+1}| > q_{\mathcal{N}}^{1-\frac{\alpha}{2}}\}$, where $q_{\mathcal{N}}^{1-\frac{\alpha}{2}}$ is the $\frac{\alpha}{2}^{\text{th}}$ fractile of the normal distribution.

The selection process for explanatory variables in our model can now be understood as follows : starting with prior dictionary D, and a fixed confidence level α,

- Stage 1 : temporary dictionary is empty : $X = [\,]$, and we add x from D columns to the model; we compute the test statistic $T_{n,1}$ using $Z = [x]$.
  For every x such that $|T_{n,1}| > q_{\mathcal{N}}^{1-\frac{\alpha}{2}}$, we only retain the one providing the smallest *p-value*.
- Stage $l+1$: we assumed having built a temporary dictionary $X \in \mathcal{M}_{n,l}(\mathbb{R})$, and we add x from D columns not already included in X ; we compute the test statistic $T_{n,l+1}$ using $Z = [X\ x]$.
  For every x such that $|T_{n,l+1}| > q_{\mathcal{N}}^{1-\frac{\alpha}{2}}$, we only retain the one providing the smallest *p-value*.
- Algorithm stops when exists no more significant variables to add into our model, of which *p-value* stays under the confidence level α.

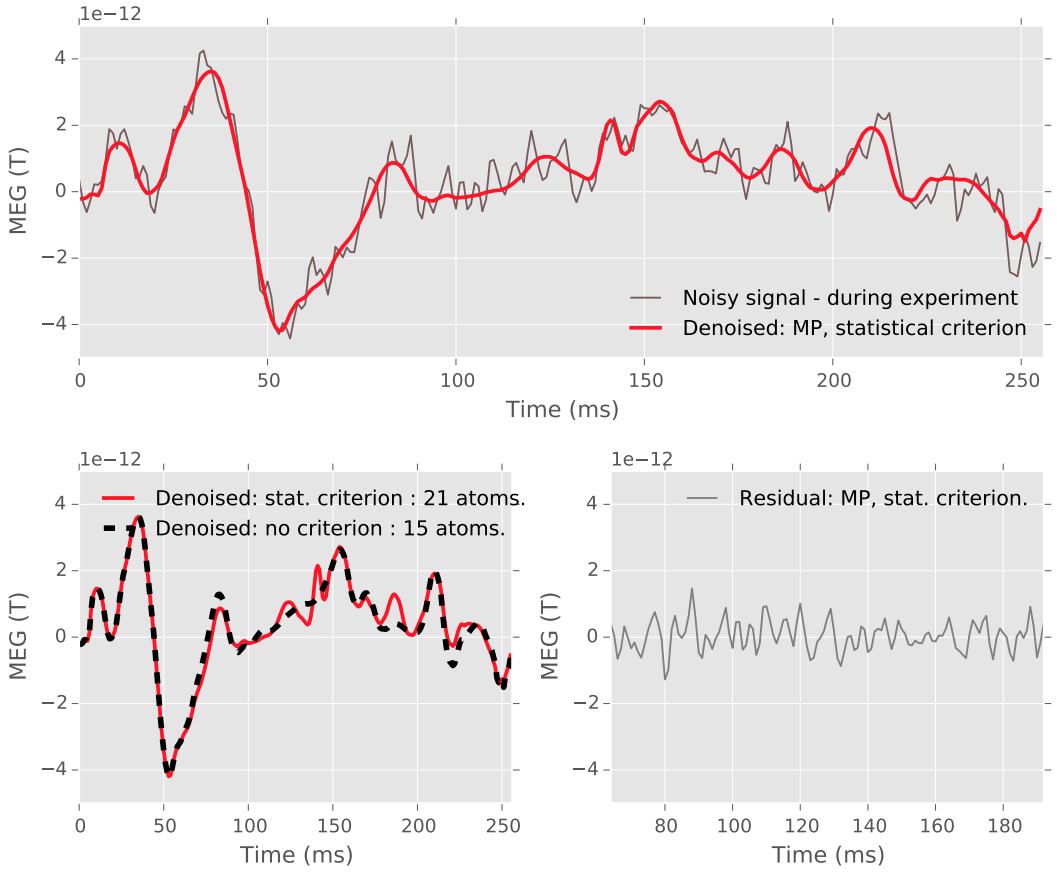

FIGURE 2.1: *Matching Pursuit results with and without statistical criterion.* 100^{th} *sensor,* `db5` *dictionary.*

Figure 2.1 shows an implementation of the MP algorithm using our statistical criterion for the 100^{th} sensor, as well as the original signal. As a guide, we also present a comparison with a MP using an arbitrary ending after 15 iterations. Last figure represents the extracted error term. For this example, we used a Daubechies wavelet dictionary, with 5 vanishing moments.

We can firstly observe that processing time is correct, with an average of 2.64×10^{-2} seconds. Secondly, the algorithm that uses our statistical criterion goes a little further than the arbitrary one ; as a matter of fact, no new atom is statistically significant at the level of 5% after 21 iterations, hence the matching pursuit algorithm naturally stops.

The Matching Pursuit algorithm smooths efficiently the signal, especially between sampling times 100 & 150, suggesting this interval is the site of several additional disturbances we properly free ourselves from.

Once our approximation obtained via Matching Pursuit, we test the relevance and robustness of our results. We begin by ensuring that the residuals computed are congruent with our statistical model.

2.2.3 Residuals characterization

One way to ensure the validity of our model is to characterize the residuals obtained after processing ; at the end of the algorithm, residuals and the additive noise we seek for should ideally match.
We investigated whether residuals followed a normal distribution or not, by using either a Quantile-Quantile diagram or a Kolmogorov-Smirnov test.

2.2.3.1 Quantile-quantile diagrams

A Quantile-Quantile diagram is a graphical method for comparing two probability distributions, by plotting their quantiles against each other.
Given an *i.i.d.* sample of length n $X = (X_1, ...X_n)$, the empirical distribution function is defined by : $F_n(x) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{\{X_i \leqslant x\}}$ from which we extract the empirical quantiles.

If the two distributions being compared match, the points in the Q–Q plot will approximately lie on the line $y = x$. If the two distributions are simply linearly related, the points will approximately lie on a different line.
This will certainly be the case : the underlying assumption here is that pre-experience signal and MP residuals both follow a normal distribution, in accordance with Assumption 4.

Figure 2.2 presents Q-Q diagrams using Daubechies wavelets with 1, 3 and 5 vanishing moments. Sample quantiles result from a MP applied to the 100^{th} sensor acquisition.

Let us specify here the reasons behind the choice of Daubechies wavelets[1]. It is mostly for their property of vanishing moments : this is very useful when aiming at reconstructing rapidly - that is to say, with a small number of atoms - regular functions[2].

[1] For more information about wavelets, see Appendix A.

[2] Although such results are not presented in this study, it is important to mention that all other wavelet families included in PyWavelet's `pywt.families()` produced equivalent results in terms of error normality.

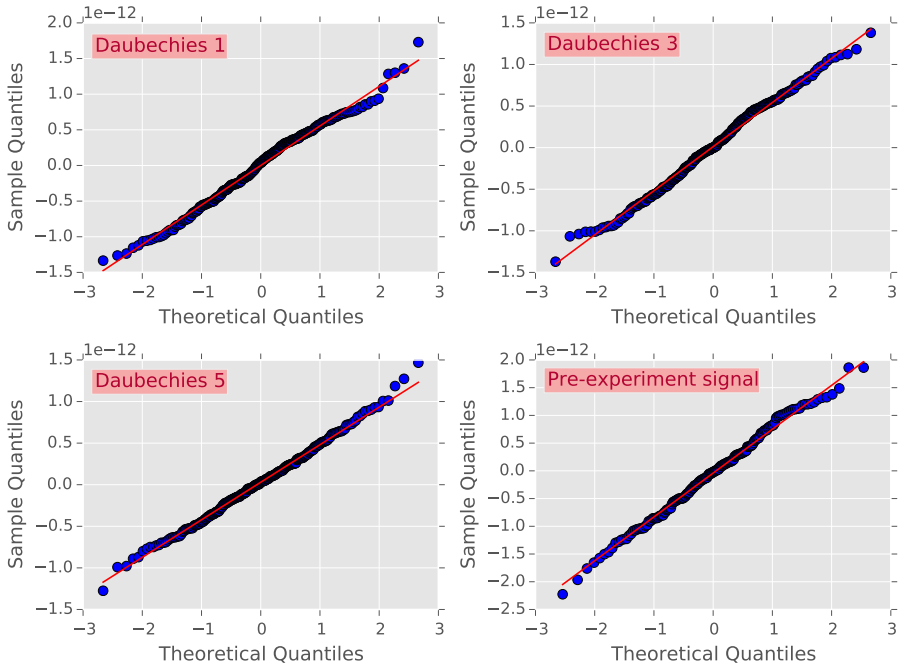

FIGURE 2.2: *Quantile-Quantile diagrams for Matching Pursuit outputs and pre-experiment signal, against data-scaled normal distribution.*

According to the four graphs, all residuals and pre-experiment signal seemingly follow a normal distribution of zero mean. This tends to legitimate assumption 4. made in 1.3.2., concerning error normality - it stands to reason, as this is a classic econometric assumption for linear models.

Although it legitimates the assumption that pre-experiment acquisition would follow a normal distribution, one can easily see that the value of the fitting line slope is clearly distinguishable from the residuals' ones. As a matter of fact, pre-experiment signal's variance is of the same order of residual variance - regardless of the number of vanishing moments - but constantly greater, as shown in Table 1 :

Signal	**Variance**
Pre-experiment	6.254×10^{-25}
MP Residual, Daubechies 1	2.071×10^{-25}
MP Residual, Daubechies 2	2.071×10^{-25}
MP Residual, Daubechies 3	2.159×10^{-25}
MP Residual, Daubechies 4	2.518×10^{-25}
MP Residual, Daubechies 5	1.622×10^{-25}
MP Residual, Daubechies 6	2.082×10^{-25}

TABLE 1. *Pre-experiment and residual variances. Unit : Tesla*^{2}.

This tends to invalidate the underlying assumption of the known-variance model - under which pre-experiment acquisitions would be equivalent to residual noise ϵ.
In this sense, a plausible explanation for this over-estimation phenomenon would be the following : during the pre-experiment phase, all neurons and all brain areas produce unnecessary signal - that is, noise.

During the experiment phase, neurons stimulated by the experiment react and emit the signal we seek for ; the rest, unaffected, can only emit noise. Assuming the independence of stimulated and non-stimulated neuronal emissions, higher variance for pre-experiment signal than residuals makes sense.

Not trying to draw any conclusions on this physical phenomenon, we kept in the following sections our fixed-variance algorithms, to which we add successive improvements. We also present alternative algorithms with estimated variance, later in Chapter 3.

In light of the graphical conviction provided by the previous diagrams, we decided to use a non-parametric test to confirm our results : to statistically validate the adequacy of residuals and pre-experiment signal to a normal distribution, we used the Kolmogorov-Smirnov test.

2.2.3.2 Kolmogorov-Smirnov test

The Kolmogorov–Smirnov statistic for a given cumulative distribution function $F(x)$ is

$$D_n = \sqrt{n} \sup_x |F_n(x) - F(x)|,$$

where $F_n(x)$ is the empirical distribution function.
The Kolmogorov–Smirnov test is constructed by using critical values of the Kolmogorov distribution ; we reject the hypothesis that empirical observations follow the specified distribution if $D_n > K_{95}$, such that $\mathbb{P}(K \leqslant K_{95}) = 0.05$, K following a Kolmogorov distribution.
The Python library used in this study actually implement the Lilliefors test, a normality test based on the Kolmogorov–Smirnov test, when the null hypothesis does not specify the expected value and variance of the distribution.

We test the null hypothesis under which the MP residuals follow a normal distribution; according to the results presented in Table 2, we cannot reject the hypothesis.

Signal	**K-statistic**	**P-value**
Pre-experiment	0.039	0.667
MP Residual, Daubechies 1	0.047	0.158
MP Residual, Daubechies 2	0.034	0.592
MP Residual, Daubechies 3	0.035	0.573
MP Residual, Daubechies 4	0.045	0.198
MP Residual, Daubechies 5	0.033	0.629
MP Residual, Daubechies 6	0.053	0.073

TABLE 2. *Lilliefors testing results for pre-experiment acquisition and experiment residuals.*

While first Daubechies wavelets tested confirm the insights provided by the Q-Q diagrams - the actual normality of residuals, we cannot help but notice that Daubechies 6 is a borderline case : although we can validate the normality of residuals it produces, we are at the limits of the rejection zone, giving a very high probability of a wrong decision Type I Error (i.e. a false positive).
We will see later other reasons that make Daubechies 6 a non-optimal dictionary for our study.

2.2.4 Model robustness

To validate the choice of the statistical criterion, one need to know whether the model is robust to several iterations of the algorithm. The goal here is to ensure that we kept only atoms explaining effectively the physiological signal.
Thus we conducted the following experiment :

1. We computed the MP algorithm on a single measurement, and obtained the associated denoised physiological signal.
2. We added a simulated gaussian white noise, on the same order of σ^2, to the MP output.
3. We computed the MP algorithm on the simulated noisy signal, and compared the two MP results.

Results are presented in Figure 2.4. Configuration is the same as in previous results.
The very slight difference in atoms included after 1^{st} and 2^{nd} Matching Pursuit is due to the fact that the atom selection is a non-convex problem ; the iterative research of significant atoms to be included in the model does not guarantee the uniqueness of the solution.
The difference between the two processed signals is of order of 6.10^{-15} Tesla, on average - a fairly decent magnitude, when compared to original measurement maximal amplitude, of order of 4.10^{-12}.

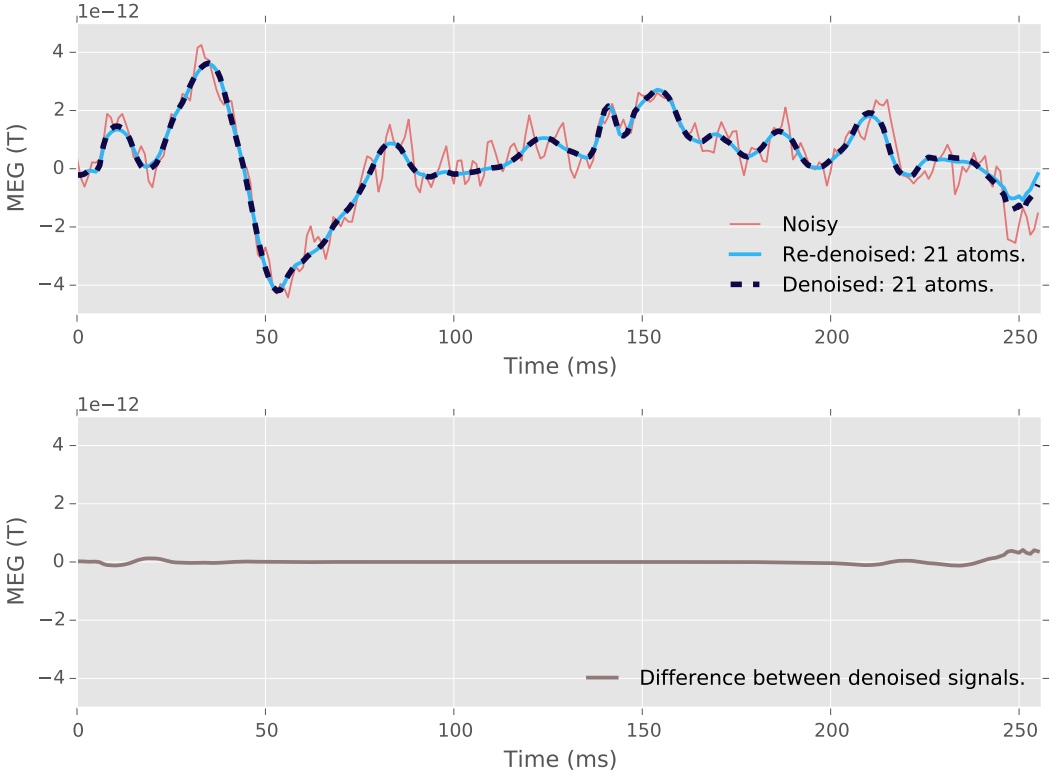

FIGURE 2.3: *Matching Pursuit on original and simulated noisy signals.*

Chapter 3

Refinements

In the following sections, we will present several improvements added to our model and algorithms ; the first one, known as Orthogonal Matching Pursuit (OMP) [2], consists of refining the classic Matching Pursuit algorithm, by adding orthogonal projections of the signal onto the set of selected atoms.

3.1 Orthogonal Matching Pursuit

Orthogonal Matching Pursuit improves over Matching Pursuit by reducing the error using an orthogonal projection on the subspace panned by the atoms retained. Signal reconstruction and residuals are orthogonal at each step.

At an iteration l of the algorithm, one computes a standard MP step :
we construct $\tilde{\lambda}$ such that

$$[\tilde{\lambda}]_{i_0} = [\lambda^{(l)}]_{i_0} + [c]_{i_0}$$

and the next iteration is computed by projecting $\tilde{\lambda}$ on the support :

$$\lambda^{(l+1)} = \underset{I(\lambda)=I(\tilde{\lambda})}{\operatorname{argmin}} \|y - D\lambda\|, \tag{3.1}$$

where $I(\lambda) = \{i \in \langle 1; q\rangle \mid \lambda_i \neq 0\}$ is the support of the solution.
This can be written matrix-wise as :

$$\lambda_I^{(l+1)} = D_I^+ y, \tag{3.2}$$

where D_I is the sub-matrix of columns indexed by I, $D_I^+ = (D_I^T D_I)^{-1} D_I^T$ its Moore-Penrose pseudo-inverse, and $\lambda_I^{(l+1)}$ the sub-vector formed by I indexes.

Data: Original noisy measurement y.
Result: Denoised signal $D\lambda$.
Initialization: $\lambda^{(0)} = 0$;
while *last atom included statistically significant* **do**
  $[\lambda^{(l+1)}]_{i_0} = [\lambda^{(l)}]_{i_0} + [c]_{i_0}$
  with
  $c = D^T(y - D\lambda^{(l)})$, $i_0 = \underset{i}{\operatorname{argmax}}|c_i|$;
  $\lambda_I^{(l+1)} = D_I^+ y$
  where I support of $\lambda^{(l+1)}$.
end

Algorithm 3: Orthogonal Matching Pursuit.

This extra step, as a matter of fact, amounts to project the signal on the subspace formed by the atoms already chosen ; it mechanically lowers the computed residuals, in comparison with the classic Matching Pursuit.

A statistical criterion for the inclusion of regressors in our model has also been set up. Test statistic, asymptotic distributions and selection processes are identical to the previous section.

Figure 3.1 shows the OMP algorithm implemented using our statistical criterion for the 100^{th} sensor, as well as the original signal. We compare it to a MP with similar characteristics. Again, we used a Daubechies wavelet dictionary, with 5 vanishing moments.

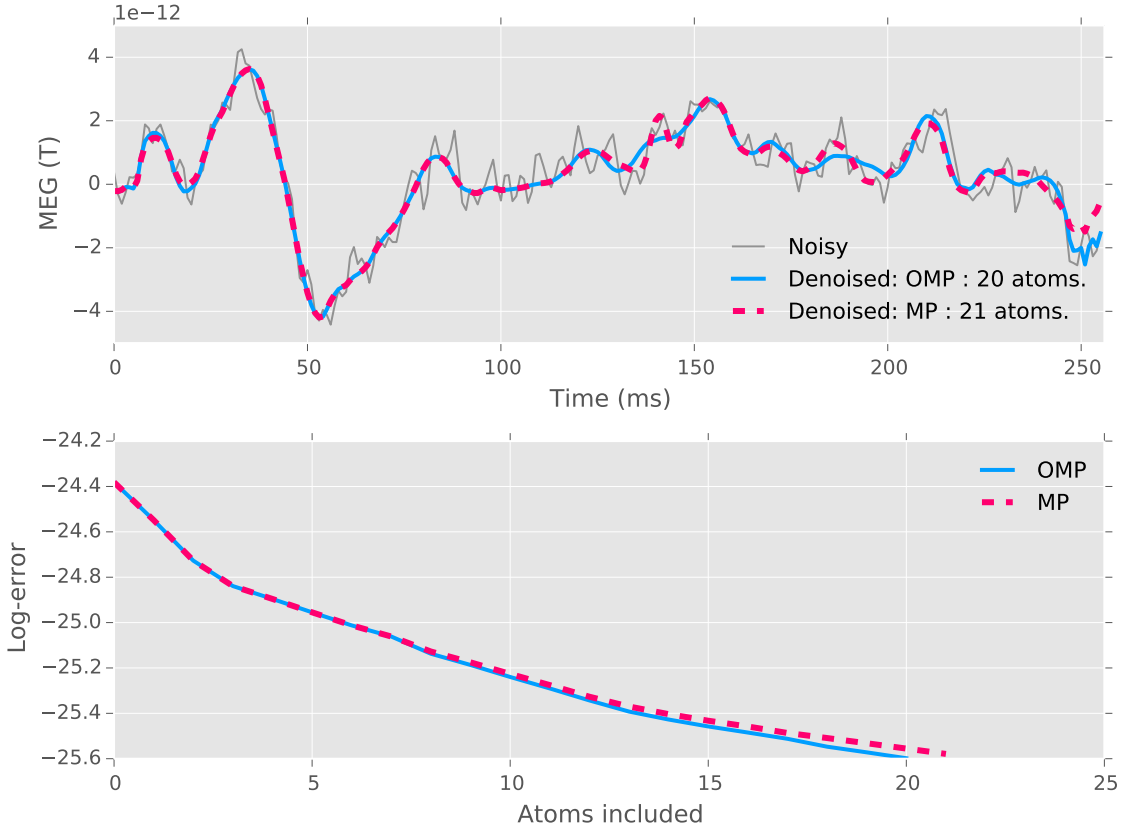

Figure 3.1: Upper graph : *Original & denoised signals via OMP, MP.*
Lower graph : *Logarithmic decrement of the error term (in norm) for OMP, MP.*

Although the numbers of atoms included are of equivalent order, OMP (with its statistical stopping criterion) often approximates the acquired signal in equal or fewer steps than the MP.
After every OMP step, all the coefficients extracted so far are updated ; this reduces the norm of the residuals while enabling the algorithm to converge more quickly than the standard version : through the modification of the sparse vector, it chooses more effectively each new atom to include in the model.
This is visible graphically, with a lower error for the OMP, when compared to MP.

The advantage of the OMP against the MP becomes stronger from a purely signal-processing standpoint : in the case where the sparsity constraint is set up manually, but not statistically - as this can be case in some signal processing studies - it appears that, using OMP, the error term becomes exactly null after a finite number of iterations. Here, the stopping criterion makes this distinction less visible, but OMP still remains an efficient improvement over the MP.

However, this has the disadvantage of producing additional computation, thus longer calculation time : 2.64×10^{-2} sec. on average for the MP, against 3.37×10^{-2} sec. for the OMP, on a single-channel implementation.
The effect is actually much more perceptible on wider ranges of data : for instance, when denoising 60 acquisitions via a multi-channel implementation, these figures rise to 4.88×10^{-1} and 8.90×10^{-1} sec, for MP and OMP respectively.
This doubling must be taken into account when denoising data of much higher spatial resolution, but will not be the subject of a detailed study here.

3.2 Orthonormal dictionary case

Another significant improvement is the usage of an orthonormal dictionary ; among the consequences such dictionaries may have on our results, we point out the simplification of testing procedures, and the equivalence with the well-known signal processing tools, such as hard thresholding.

3.2.1 Testing procedure

Let us recall the testing procedure revealed in 2.2.2. Following the previous notation, X contains already-selected columns of D, and x denotes a different atom we intend to add to our model.

If D is an orthogonal matrix - or semi-orthogonal matrix, in the case of non-squared matrices - previously-selected atoms and x are orthogonal : this directly implies $P_X x = 0$. Same goes for the scalar product between new atom x and the projection $P_X y$: $\langle x, P_X y \rangle = 0$.

If D is an orthonormal matrix, it also implies $||x|| = 1$.
Thus we obtain the following simplified test statistic :

$$T_{n,l+1} = \frac{x^T(y - P_X y)}{\sigma} = \frac{1}{\sigma}\langle x, y - P_X y\rangle = \frac{1}{\sigma}\langle x, y\rangle, \tag{3.3}$$

only depending on angle between new independant variable x and signal y.
It is important to note that this statistic does not use previously selected atoms to evaluate significance of atom x.
It then becomes possible to realize our approximation pursuit in a single step, without any iterations : it suffices to retain only the s atoms of maximum correlation, s being the sparsity constraint of problem (2.1), determined by usage of our statistical criterion.

Decomposition of the signal is then written in our orthonormal basis :

$$y = \sum_{j=1}^{q} \langle y, D_j \rangle D_j \tag{3.4}$$

and one has to only sort the atoms by correlation and operate a hard thresholding to retain the desired atoms. This will be the subject of the next section.

Remark. The Daubechies wavelet family is orthogonal : this means that the family generated by the mother wavelet is orthogonal[1] ; this definition does not imply that the dictionary is orthogonal (i.e. that its columns are orthogonal).
As a matter of fact, Daubechies dictionaries formed by `PyWavelet`'s wavelet transforms are not orthogonal - except Daubechies 1, corresponding to the Haar wavelet. Daubechies orthonormal dictionaries have been produced using the Gram-Schmidt algorithm.

Figure 3.2 presents the first results obtained computing an Orthogonal Matching Pursuit, alternately using the standard dictionary and its orthonormal version, for Daubechies 5. An OMP performed using orthonormal dictionary computes in 3.75×10^{-2} seconds.

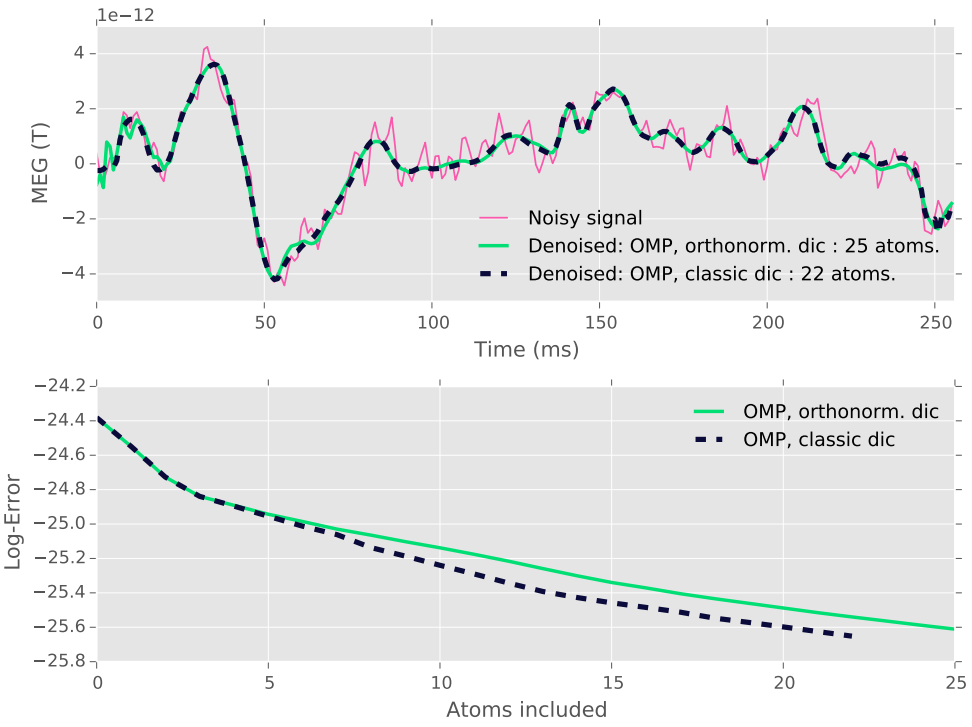

FIGURE 3.2: *OMP results, using classic and orthonormal basis.*

[1] This is done by translations and dilatations. For definitions & detailed properties, see Appendix A.

We can see from the upper graph that the use of an orthonormal dictionary does not seem to significantly modify our results - the difference between the two denoised signals revealing the non-orthogonality defaults.
However, the performance of the orthonormal dictionary - in terms of error convergence - seems poorer, and more atoms seem to be required to reach the final denoised signal. In fact, the orthonormalized dictionary is not optimal[2], and will not be used for the following sections. It has nevertheless a great advantage over the classic version : it is possible to implement it in a quick and simple one-step version, via the hard thresholding operator.

3.2.2 Hard thresholding equivalence

Let us now present specifically the equivalence between our iterative algorithms and the hard thresholding procedure, in the case of an orthonormal dictionary. As mentioned, we will use orthonormalized Daubechies dictionaries.
We will see here dictionaries as operators - something sensible, as they derive from discrete signal processing transforms. Our framework is the same as 3.2.1.

One can compute in closed form the solution to the signal recovery problem, by using the hard threshold operator :

$$D(\lambda) = D \circ S_T \circ D^T(y), \tag{3.5}$$

where $S_T(u)_i = \begin{cases} u_i \text{ if } |u_i| > T \\ 0 \text{ otherwise.} \end{cases}$
Threshold T should be selected so that either the sparsity or the error contraint is checked. Considering how we built our algorithms, we ensure here that our hard thresholding operator verifies the sparsity constraint.

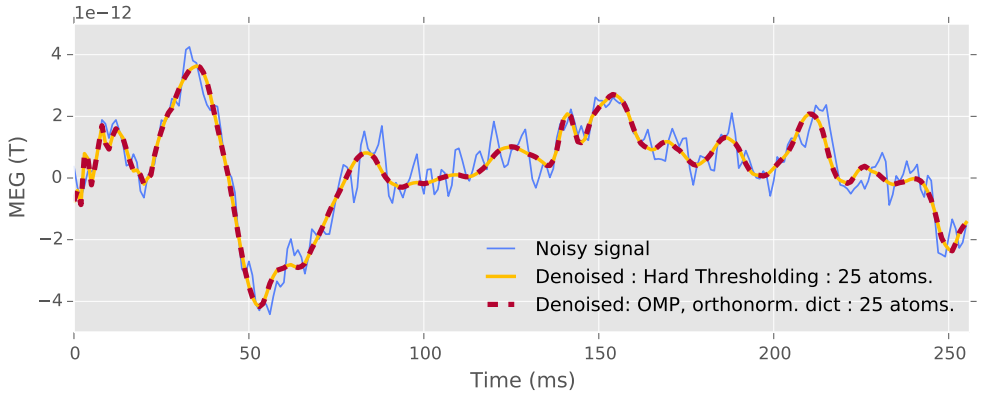

FIGURE 3.3: *Comparing OMP, using an orthonormal dictionary & hard thresholding.* `db5` dictionary.

As Figure 3.3 reveals, Matching Pursuit and hard thresholding become equivalent under the same sparsity constraint. It now becomes possible to easily implement our iterative procedures, with this one-step equivalent, provided the dictionary used is semi-orthogonal.

[2]We will see later criteria to quantify the performance of a standard dictionary, and compare two dictionaries.

3.3 Estimated variance model

Another approach possible is the development of algorithms requiring no known variance ; this makes sense in the case of experiments where idle state measurements are unavailable, for example.

Thus we suggest in this section a new version of the classic Matching Pursuit algorithm, which estimates the variance of the residuals at each stage using unbiased estimator s^2. This does not affect the research of maximally-correlated atoms, but changes the expression of our test statistic.

Let us place ourselves in the testing framework presented in 2.2.2.
We intend to add a new coefficient γ to our model, and we want to test its significance.

We test the null hypothesis $\mathcal{H}_0 : \gamma = 0$ against the two-sided alternative $\mathcal{H}_1 : \gamma \neq 0$. Conditionally on X, the expression of our test statistic is :

$$T_{n,l+1} = \frac{\hat{\gamma}}{\hat{\text{s.e.}}(\hat{\gamma})} \sim t_{n-l-1} \text{ under } \mathcal{H}_0. \tag{3.6}$$

Again, we can re-write this, knowing that $\hat{\text{s.e.}}(\hat{\gamma}) = \sqrt{s^2[(Z^TZ)^{-1}]_{l+1,l+1}}$:

$$T_{n,l+1} = \frac{[(Z^TZ)^{-1}Z^Ty]_{l+1,1}}{\sqrt{s^2[(Z^TZ)^{-1}]_{l+1,l+1}}} \tag{3.7}$$

and, finally :

$$T_{n,l+1} = \frac{x^T(y - P_Xy)}{\sqrt{s^2(\|x\|_2^2 - x^TP_Xx)}}. \tag{3.8}$$

Region of rejection for hypothesis $\mathcal{H}_0$ becomes $\{|T_{n,l+1|} > q_{n-l-1}^{1-\frac{\alpha}{2}}\}$, where $q_{n-l-1}^{1-\frac{\alpha}{2}}$ is the $\frac{\alpha}{2}^{\text{th}}$ fractile of a t-distribution, with $n - l - 1$ degrees of freedom.

Appendix B presents approximation results obtained via classic and variance-estimating algorithms, for the Matching Pursuit & Orthogonal Matching Pursuit.

The question of higher pre-experiment variance, presented in 2.2.3.1., resurfaces : estimated residuals variance being smaller than the pre-experiment one, the test statistic augments mechanically. Thus more atoms are added to our model, in comparison to the known-variance version.
This alternative algorithm and its stopping criterion nevertheless preserve their statistical legitimacy ; an argument in this favor is to consider the logarithmic error curves : we can see there that the algorithm selects exactly the same first atoms to denoise the signal - the estimated-variance version simply includes more elements.

As a matter of fact, the purpose of this study is not to decide on this variance issue: we only offer two alternatives to the experimenter; he will choose according to his expectations and the data in presence.

Chapter 4

Multi-channel model

We consider in this chapter the multi-channel linear inverse problem as stated in (1.4). We still use l^0-sparse optimization to recover an approximation of the signals in Λ - in the sense that all channels verify the same sparsity constraint :

$$\min_{\substack{\Lambda \\ \|\Lambda^j\|_0 \leqslant s,\ \forall j \in \langle 1;n \rangle}} \|Y - D\Lambda\| \tag{4.1}$$

where Λ^j denotes the j^{th} column of Λ. Our goal is to find atoms in D that explain the measured electromagnetic field, regardless of the sensor involved in the acquisition.

4.1 Multi-channel Matching Pursuit

As a matter of fact, the Matching Pursuit algorithm can be extended to a straightforward multi-sensor version, called Multi-channel Matching Pursuit (MMP) [3], deriving from the original MP presented in (2.2).

Recall that in the single-channel MP, we iteratively sought for the dictionary atom providing the highest absolute correlation with residuals : $D_{i_0} = \underset{D_i \in D}{\text{argmax}} |\langle R^l y, D_i \rangle|$, and subtract to residual $R^l y$ its projection on D_{i_0} to form next residual.

The multichannel algorithm is similarly constructed.
· At first stage, we set $R^{(0)}Y = Y$.
· At stage l, we seek for the atom maximizing the sum of squared products (i.e. energies) in all the r channels :

$$D_{i_0} = \underset{D_i \in D}{\text{argmax}} \sum_{j=1}^{r} |\langle R^{(l)} y_j, D_i \rangle|^2, \tag{4.2}$$

where y_j is the j^{th} channel from $Y = [y_1 ... y_r]$.
· We similarly subtract to matrix residual $R^l Y$ its projection on D_{i_0} to form residual at stage $l + 1$.

Note that the sum of squared products to be maximized in (4.1) is in fact the squared euclidean norm of row i, extracted from correlation matrix $C = D^T(Y - D\Lambda)$.

Thus the MMP algorithm can be written as :

Data: Original noisy matrix Y.
Result: Denoised, sparse matrix $D\Lambda$.
Initialization: $\Lambda^{(0)} = 0$;
while *last atom included statistically significant* **do**
  for $j \in \langle 1; r \rangle$ **do**
    $[\Lambda^{(l+1)}]_{i_0,j} = [\Lambda^{(l)}]_{i_0,j} + [C]_{i_0,j}$,
    with
    $C = D^T(Y - D\Lambda^{(l)})$ correlation matrix,
    $i_0 = \underset{i}{\operatorname{argmax}} \left\|C^i\right\|_2^2$, C^i i^{th} row.
  end
end

Algorithm 4: Multi-channel Matching Pursuit.

Again it comes to determining a statistical criterion for stopping the algorithm, this time concerning simultaneously all analyzed signals in Y.

At stage $l+1$, we test the null hypothesis $\mathcal{H}_0$ under which the correlation row C^i we add to $\Lambda^{(l)}$, to form $\Lambda^{(l+1)}$, is not significant - against its two-sided alternative.
This amounts to testing significance of new coefficients $[C]_{i_0,j}$.
One can form the previous test statistics from 2.2.2. for each channel j, noted $T^{(j)}_{n,l+1}$ and compute $K_{n,l+1} = \sum_{j=1}^{r} |T^{(j)}_{n,l+1}|^2$:

$$K_{n,l+1} \sim \chi^2_r \qquad \text{under } \mathcal{H}_0. \tag{4.3}$$

Region of rejection for hypothesis $\mathcal{H}_0$ becomes $\{|K_{n,l+1|} > q_r^{1-\frac{\alpha}{2}}\}$, where $q_r^{1-\frac{\alpha}{2}}$ is the $\frac{\alpha}{2}^{\text{th}}$ fractile of a χ^2-distribution, with r degrees of freedom.

Figure 4.1 presents results broadened to the study of the total signal, i.e. considering the entire set of the 203 acquisition sensors, via MP algorithm. We used a Daubechies wavelet dictionary, with 17 vanishing moments this time.[1] We arbitrary plot denoised signals from sensors 53 to 56 against their respective original acquisitions, as an insight of the approximations obtained via the multi-channel procedure.

This multi-channel denoising is compelling : residuals obtained obtained via MP present same normality qualities as in the single-channel section.[2] Due to the large number of signals studied, this procedure is very stable and provides a very efficient stop for atoms' inclusion.
The computation time is also very satisfactory, with an average of 1.368 seconds for a 203-channel Matching Pursuit.

Note that, at this stage, one can perform a mapping of the brain by representing respective areas of high and low noise. This would allow to distinguish brain zones stimulated by the experiment from the intact ones ; it becomes possible to draw conclusions about the physical nature of the experiments, and their influence on brain performance.
However, these conclusions are dependent on the choice of explanatory dictionary ; the next section allows us to choose the optimal dictionary for our study.

[1] The reason behind this choice is presented in the following sections.
[2] A reader interested in more detailed results may refer to Appendix C for figures and tests concerning sensors 50-59.

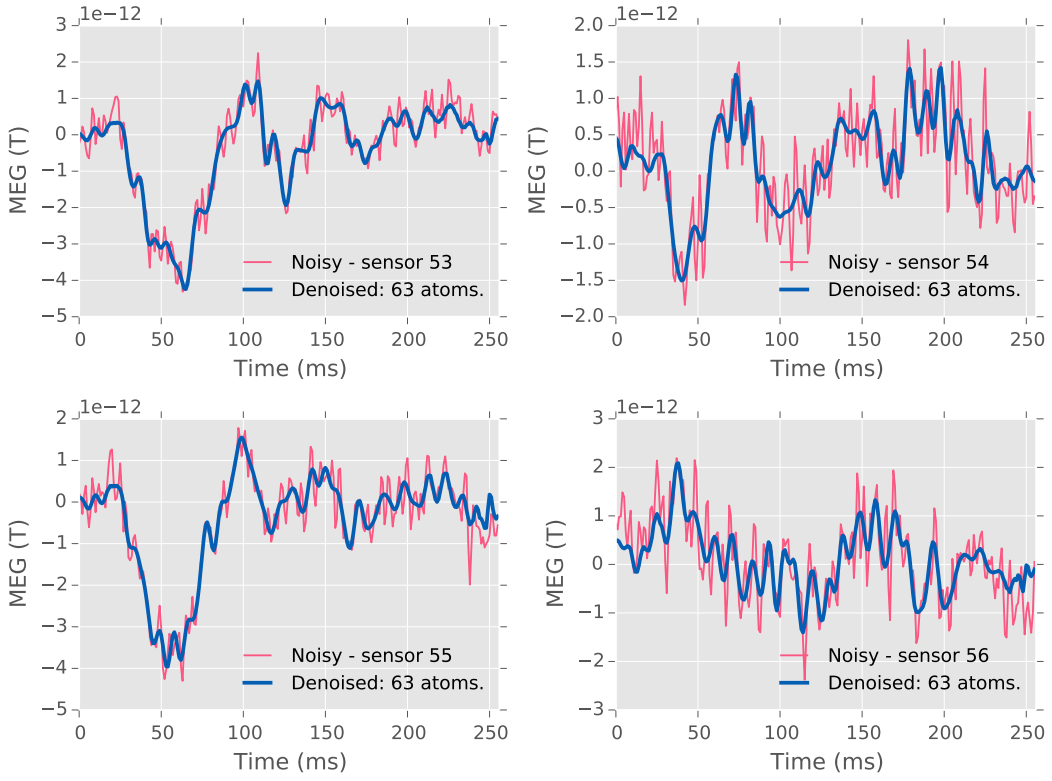

FIGURE 4.1: *4 denoised signals obtained via a 203-channel Matching Pursuit ; sensors 53-56,* `db17` *dictionary.*

4.2 Dictionary selection

One of the key points we have to consider in our denoising is the choice of the right dictionary. All wavelet families included in the PyWavelet library can be thought as legitimate candidates, in terms of error normality ; yet we deliberately restrict ourselves to the Daubechies wavelet family, for the aforementioned reasons.
The saying is that, in general, the Daubechies wavelets are chosen to have the highest number of vanishing moments.
In this section, we will develop criteria to select the best dictionary and see whether this traditional rule is legitimate or not.

4.2.1 Goodness of fit

Recalling the similarities between the single-sensor model and classic linear models, one may initially want to determine how well the combination of explanatory variables fit well our observations. A logical candidate for the evaluation of the model would then be the coefficient of determination R^2.
Note that it would never reach a maximal value of 1 : although the inverse linear problem is very similar to classic econometric models, our goal here is not to explain the largest share of our signals using dictionary atoms, but only to extract the part actually caused by the experience ; this is done via the statistical stopping criterion, allowing us to efficiently determine the number of regressors to include in our model.

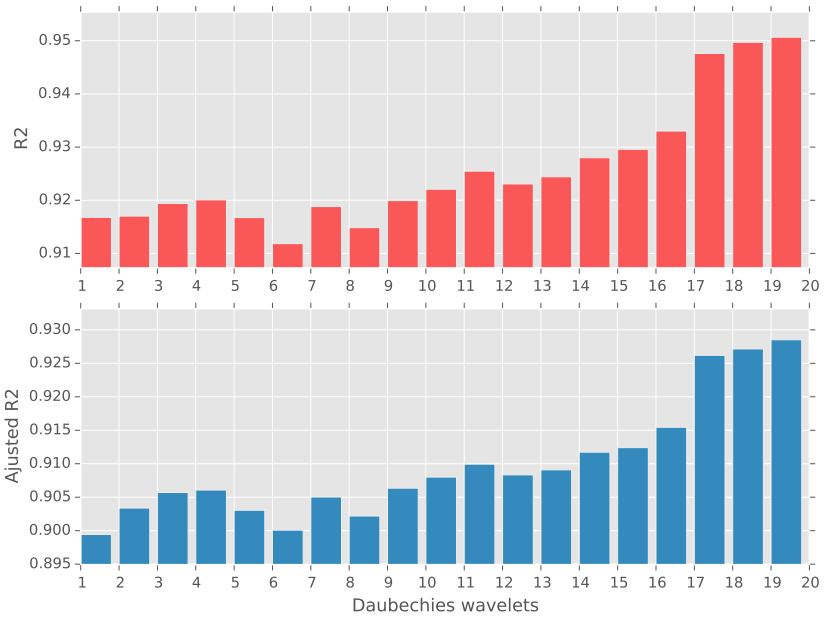

FIGURE 4.2: *Histogram of R^2 coefficients for the Daubechies wavelet family.*
UPPER GRAPH : *Classical R^2.*
LOWER GRAPH : *Ajusted R^2.*

Table 3 shows R^2 and ajusted R^2 obtained via various wavelet types.
As we can see, counter-intuitive results already appear ; for example, according to both graphs, several Daubechies wavelets with higher number of vanishing moments - something characterizing the ability to represent polynomial behaviour or information in a signal - do not necessarily provide the best results possible : for example, `db6` produces poorer R^2 results than `db4`. However, there is an overall trend towards improvement when augmenting vanishing moments. `db17` to `db20` seem to be the best dictionaries.

As a matter of fact, the R^2 is a basic but non-optimal tool for our study ; it allows us to only draw conclusions on the size of the produced residuals - something determined by the stopping criterion. Although the adjusted R^2 erases the mechanical increase produced by the addition of new variables, this coefficient still does not penalize the size of the sparse vector itself.
We present now two alternative methods for dictionary selection, both taking into account the constraints of our denoising problem.

4.2.2 Information criteria

Let us now introduce the concept of statistical information criterion ; a well-known and widely-used example is the Akaike Information Criterion (AIC).

The Akaike Information Criterion is a measure of the relative quality of a statistical model, for a given set of data. AIC deals with the trade-off between the goodness of fit of the model and its complexity.[3] The penalty discourages overfitting.

[3]AIC is founded on information entropy : it offers a relative estimate of the information lost using a given model ; something we would ideally represent by calculating the Kullback–Leibler divergence between the used model and the real (but usually unknown) process that generated the data.

For any statistical model :

$$AIC = 2l - 2\ln(L)$$

where $l \leqslant q$ is the number of parameters in the model, and L the maximized value of the likelihood function for the model.

Under our assumptions that errors are *i.i.d.* according to a normal distribution, and the boundary condition that log likelihood derivative with respect to the true variance is zero, this becomes :

$$AIC = 2l + n\ln(\sigma^2).$$

Given a set of candidate models for the data, the preferred model is the one with the minimum AIC value. Hence AIC not only rewards goodness of fit, but also includes a penalty that is an increasing function of the number of estimated parameters.

If we denote the AIC values of the candidate models by $AIC_1, AIC_2, AIC_3, ..., AIC_M$, then $\exp(\frac{AIC_{min} - AIC_i}{2})$ can be interpreted as the relative probability that the i^{th} model minimizes the estimated information loss.

We use in this study the AICc, that is AIC with a correction for finite sample sizes :

$$AICc = AIC + \frac{2l(l+1)}{n-l-1}$$

AICc is AIC with a greater penalty for extra parameters, and converges to AIC as n gets large.

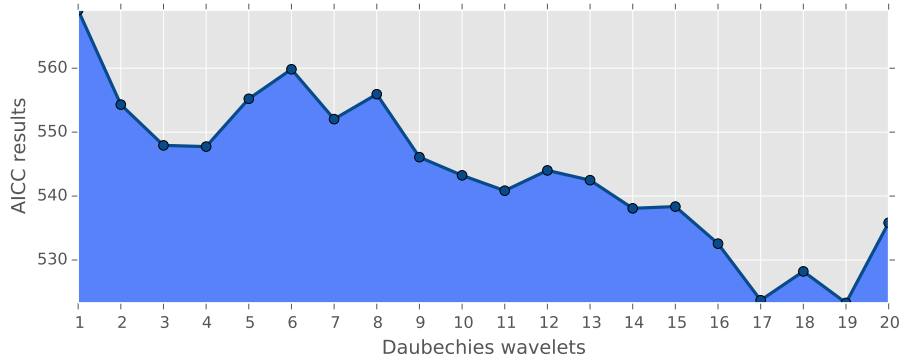

FIGURE 4.3: *AICc results for the Daubechies wavelet family, via MOMP.*

Figure 4.3 presents AICc results for the entire Daubechies wavelet family, via Multi-channel OMP (MOMP). As only the difference between two AICc's matter, we added an arbitrary constant to only consider positive values.
We can see here that, even when penalizing the sparse vector size, the overall trend stays. `db17` and `db19` clearly seem to be the best dictionaries for our study: for example, `db16` is $\exp(\frac{521-532}{2}) = 0.004$ times as probable as `db17` to minimize the information loss compared ; we therefore cannot select it, in comparison.[4]

We introduce a second criterion for selection of our dictionary : its ability to explain different portions of the signal using the same atoms. This is done by using 2-fold cross validations.

[4]We saw in subsection 2.2.3.1. that `db6` produced poor results in terms of normality.
AICc criterion shows that this dictionary is also non-optimal in terms of fit.

4.2.3 Cross-validation

The cross-validation is a technique for estimating the performance of a predictive model. It commonly involves partitioning a sample of data into distinct subsets, performing the analysis on one subset (called the *training* set), and validating the analysis on the other subset (called the validation set or *testing* set). To reduce variability, we exchange roles between subsets and perfom again the validation.

We perfom here a classic 2-fold cross-validation, also called holdout method :
· We arbitrary separate our experiment data in two signals of equal size Y_A & Y_B.

· In the first fold, we train on Y_A and test on Y_B : we perform an Orthogonal Matching Pursuit[5] on Y_A and obtain a list of atoms D_A selected to explain the signal.
Then we explain Y_B in two ways :

- firstly, via a similar Orthogonal Matching Pursuit on Y_B : this provides the expected denoised signal Z_B, such that $Y_B = Z_B + E_B$.
- secondly, only using the atoms D_A, via a hard thresholding : this provides an approximation of the denoised signal. We will denote this estimate by $\tilde{Z_B}$.

A good dictionary explains the bulk of the signal using the same atoms ; one can expect to find little difference between Z_B and $\tilde{Z_B}$ with an efficient explanatory basis.
To quantify this error, we use a risk function corresponding to the expected value of the quadratic loss : $R(\tilde{Z_B}) = ||Z_B - \tilde{Z_B}||_F^2$.[6]

· In the second fold, we exchange roles between Y_A and Y_B ; we train on Y_A and test on Y_B via the same procedure.

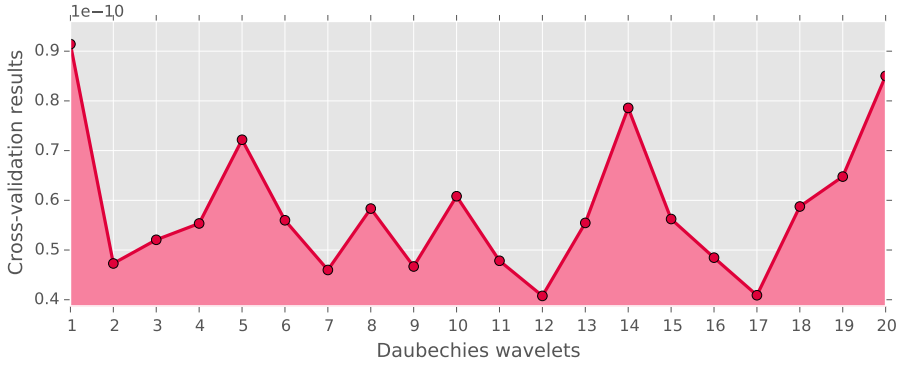

FIGURE 4.4: *Cross-validation results for the Daubechies wavelet family.*

Figure 4.4 provides cross-validation results for the Daubechies wavelet family. Here again, the `db17` dictionary gives the best results ; we can conclude from the two previous criteria that `db17` is an excellent dictionary for our study ; it will therefore be used in further results.
We can now say that the usual rule of thumb - of using the Daubechies wavelet with the highest number of vanishing moments - comes from a statistical reality. However, one must be cautious in its use: `db20` fails to provide the best possible results. A comprehensive study is desirable.

[5] If necessary, we orthonormalize the dictionary tested to preserve the equivalence with hard thresholding presented in 3.2.

[6] $||.||_F$ denotes the Frobenius norm.

4.3 Constructing the optimal dictionary : K-SVD

Finally, we present a method for building an optimal, data-dependent dictionary. We modify iteratively an initial dictionary to fit more accurately with the dataset employed ; for instance, one can start with the best wavelet dictionary available - found at the previous section - and optimize it until the optimum is reached.

We present here a learning algorithm built on a singular value decomposition approach, called K-SVD [4]. It is a generalization of the K-means clustering method, and it works by iteratively alternating between sparse coding our data based on the current dictionary, and updating the atoms in the dictionary to better fit the data.

We aim at finding the best possible dictionary and its corresponding sparse approximation, to represent the data samples by nearest neighbor ; thus we solve a more complex constraint problem than (4.1) :

$$\min_{\substack{D,\Lambda \\ \|\Lambda^j\|_0 \leqslant s,\ \forall j \in \langle 1;n \rangle}} \|Y - D\Lambda\| \tag{4.4}$$

where Λ^j denotes the j^{th} column of Λ (a single sparse signal).
As a matter of fact, we will minimize equivalently here the error energy term $\|Y - D\Lambda\|^2$.

Each iteration of the K-SVD algorithm can be divided in two steps :

1. We fix D, and find the best sparse approximation for our data Λ, using greedy algorithm. We will use here the Multi-channel Orthogonal Matching Pursuit to reconstruct the denoised signal.
2. After the sparse coding task, we optimize dictionary D, by updating one column at a time while fixing Λ. This updating phase, for example at iteration k, can be done by rewriting the penalty term as :

$$\begin{aligned} \|Y - D\Lambda\|^2 = \|Y - \sum_{j=1}^{q} D_j \Lambda^j\|^2 &= \|Y - \sum_{j=1, j\neq k}^{q} D_j \Lambda^j - D_k \Lambda^k\|^2 \\ &= \|Err - D_k \Lambda^k\|^2, \end{aligned} \tag{4.5}$$

   where D_k denotes the k^{th} column, Λ^k the k^{th} row vector and Err the error term removed from the influence of D_k.
   We can now solve the minimization problem by approximating the Err term with a rank-1 matrix, using a singular value decomposition (SVD), and then updating D_k with it.

   In practice, to enforce the sparsity constrain, we only look at the columns involved in the sparse coding step : after performing our MOMP, we shrink our vectors and matrices by removing the columns not included in the model :
   row vector Λ^k becomes Λ^k_R, its zero coefficients being discarded ;
   matrix Err becomes Err_R, with a restricted number of columns.

   The minimization problem aforementioned becomes $\|Err_R - D_k \Lambda^k_R\|^2$ and can be done by directly using SVD. SVD decomposes Err_R into $U \Delta V^T$.

   We update D_k with the first column of U, and the coefficient vector Λ^k_R with the first column of V multiplied by $[\Delta]_{1,1}$. After updating all the atoms included in the model, the process goes back to solve X.

Figure 4.5 compares the results obtained via K-SVD and Multi-Channel OMP methods, in terms of denoised signals and logarithmic error decrements. In the upper graph, we implemented the K-SVD method with 8 iterations, and plotted the last approximation obtained against an MOMP result, and the original noisy acquisition.
All sparse representations have been produced based on Daubechies 17 wavelets.

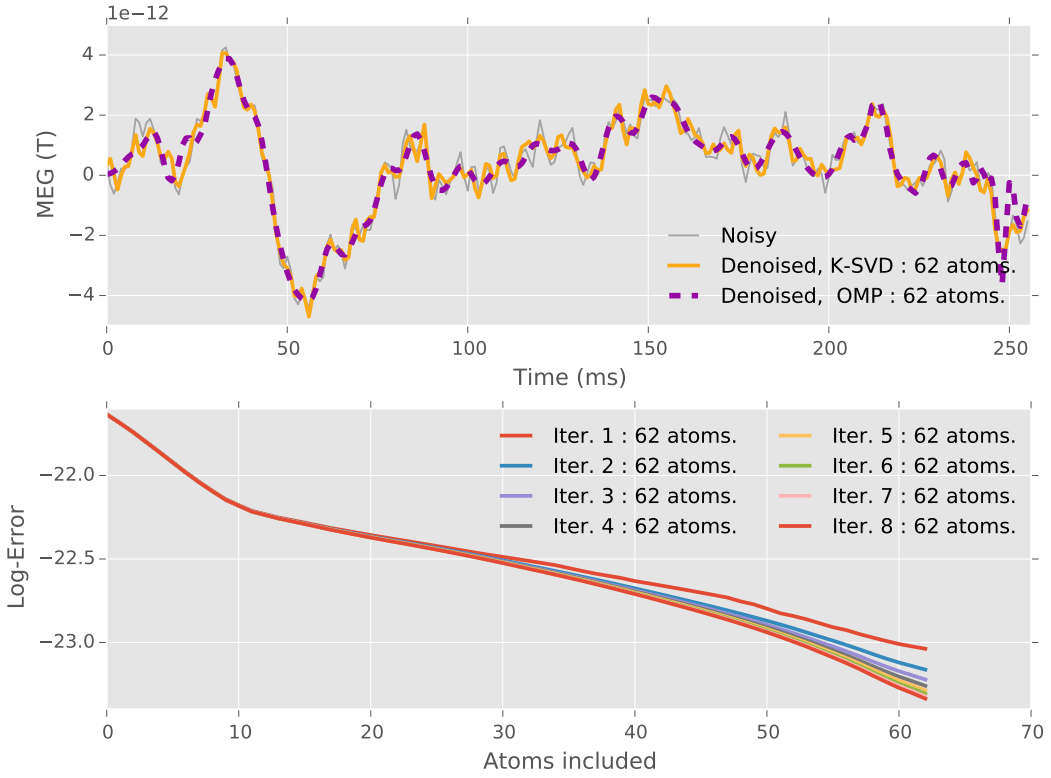

FIGURE 4.5: UPPER GRAPH : *Denoised signals, via K-SVD and OMP methods.*
LOWER GRAPH : *Associated logarithmic error decrements.*
Initial dictionary : `db17`.

We can observe here that, although we computed multi-channel denoisings, the approximation provided by the K-SVD method is very precise.[7] However, this automatically implies a diminution of the error term, in accordance.
Note the difference with the objective function to minimize in (4.1), for a unique Multi-channel Matching Pursuit : here we solve iteratively several linear inverse problems, aiming at minimizing the error term subject to the sparse constraint. Therefore it is not surprising to find a smaller error after each iteration.

Qualitative interpretation of the phenomenon then depends on the error threshold expected by the experimenter, in a known-variance model : it belongs to him to set up the number of iterations, following the experiment aim and the manipulated data.
In our study, a classic Matching Pursuit can be sufficient to explain and analyze, on the surface, most of the signal constructed. The K-SVD becomes suitable in the case of subtle phenomena analysis, where we seek to explore targeted local variations.

[7] This is why K-SVD is particularly suitable for compressed sensing, signal reconstruction purposes.

Chapter 5

Conclusion

The primary aim of this study was to provide denoised signals, from encephalographic acquisitions, which might be serve as a backdrop for further physical and comportemental analysis. We used MEG recordings from experiments conducted at the Martinos Center of Massachusetts General Hospital, captured using a set of 203 sensors.
The denoised signals must be removed from any additive noise and present the greatest clarity possible.

To develop an effective and rigorous denoising procedure, we placed ourselves in a linear framework, using a single explanatory dictionary for the entire set of sensor measurements ; indeed, acquisitions are explained using a unique set of relevant explanatory variables.
We also assumed that the extracted noise extract would satisfy suitable statistical requirements, such as normality or homoscedasticity.

We first used sparse optimization to initially recover the denoised approximation from a single-channel acquisition. Building the sparse vector was performed using heuristic greedy optimization methods, such as the classic signal processing algorithm Matching Pursuit. We developed for this algorithm a stopping criterion, to iteratively select only relevant variables for our model.
Residues faithfully met our normality requirements, and the algorithm has proven being very handy and robust.

We improved this first model by adding orthogonal projections on dictionary's selected atoms (Orthogonal Matching Pursuit) ; this allowed a more succinct explanation of the signal and a faster error convergence. We also presented several simplifications implied by dictionary orthonormalization ; in this case, our iterative algorithms become equivalent to a classic one-step procedure, the hard thresholding. A model with estimated variance has also been detailed.

We therefore generalized our procedures to the case of multichannel acquistions and presented rigorous selection criteria for best dictionary. We have seen in particular that the golden rule for the selection of the best wavelet to use reflects a statistical reality, but is only valid to a certain extent ; nothing replaces careful selection, accordingly to the study conducted. We present criteria to ensure the best trade-off possible between a good fitting of the data and a small model complexity ; our model is stable against cross-validations, addition of extra noise, and allows us to stay away from over-fitting problems.

As a graphic overview, we present a superposition of brain signals, on a short range of sensors, before and after denoising. Obtained signals are easy to characterize, from a graphical standpoint, and ready-to-use.
The use of Daubechies wavelets provides beneficial regularity for reconstructed signal ; an useful point when it comes to handling data during post-denoising procedures.

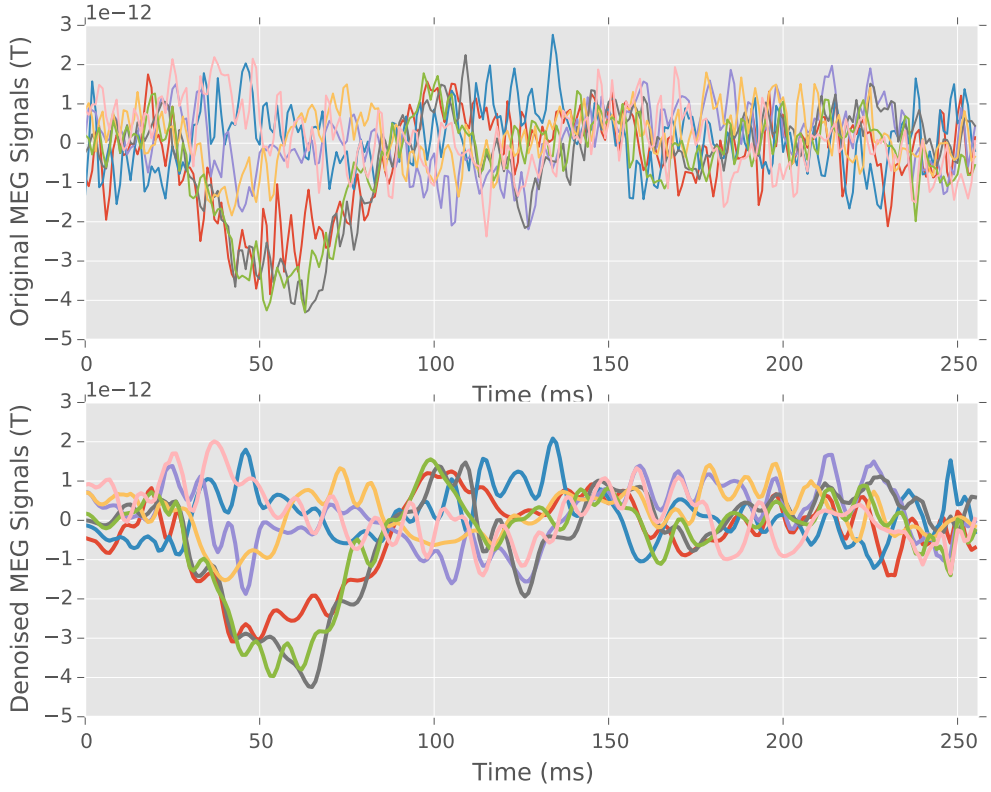

Figure 5.1: Upper Graph. *Original MEG acquisitions, sensors 50-57.*
Lower Graph. *Denoised MEG acquisitions, sensors 50-57.*

At the end of our procedures, we obtain quality signals that can be analyzed quickly by doctors and physicists, to draw conclusions on the influence of stimuli on brain performance. It becomes possible to distinguish brain areas stimulated during the experiment and develop brain mappings to medically quantify the patient behavior during the study; a purpose beyond the scope of this study, but raising scientific issues of great interest.

Appendix A

Wavelets Decomposition and Signal Processing

A.1 Wavelet theory genesis

Facing heat equation, Jean-Baptiste Fourier developed the idea that any periodic function - regular enough - could be seen as a combination of classic cosinus and sinus functions, of different frequencies. This remark would lay the first lines of a new theory: the harmonic analysis of signals. At this time, the only representation used was a temporal one, indicating the signal intensity at time t, but without information on its harmonic content.

Though the Fourier Analysis marks an important breakthrough concerning signal analysis, it still presents weaknesses that will be solved by the wavelet theory. Indeed, on one hand, this decomposition in Fourier series is only possible for periodic functions[1]. On the other hand, the coefficients of this decomposition are very sensible to local perturbations of the signal - a major drawback for its analysis.

Wavelet theory aims at building hilbertian basis of $\mathbb{L}^2(\mathbb{R})$, which behaves better than the Fourier one for the problems previously exposed.

Definition 1. Let H be a Hilbert Space. The family $(e_i)_{i\in I}$ is an Hilbertian basis of H if it is :
1. Orthonormal : $\forall i, j \in$ I, $\langle e_i, e_j \rangle = \delta_{ij}$
2. Total : $H = \overline{Vect(e_i)}$

Proposition 1. Let H be a separable Hilbert Space and $(e_n)_{n\in\mathbb{N}}$ an orthonormal family of H. The following propositions are equivalent.
1. $(e_n)_{n\in\mathbb{N}}$ is an Hilbertian basis
2. $\forall x \in H, x = \sum_n \langle x, e_n \rangle e_n$
3. $\{e_n \mid n \in \mathbb{N}\}^{\perp} = 0$
4. $\forall x \in$ H, $\|x\|^2 = \sum_n |\langle x, e_n \rangle|^2$. (Energy Conservation)

[1] Note that in practice, we observe signals on a finite time interval, so that we usually extend functions periodically.

A.2 Discrete Wavelet Transform

One can easily draw many parallels between the Fourier and the wavelet transforms. Imaginary exponentials[2] $e^{i\langle x,\Psi\rangle}$ from the former $\Psi \in \mathbb{R}^n$ are replaced by wavelets ϕ_Q indexed by the pavements $\mathrm{Q} \subset \mathbb{R}^n$. These wavelelets ϕ_Q are translations and dilatations of an initial function ϕ.
A largely used collection of pavements is the following :
$Q_{j,k} = \{x \in \mathbb{R}^n; 2^j x - k \in [0;1]^n\}$, called the dyadic scale.

In the following we will thus consider a family of $\mathbb{L}^2(\mathbb{R})$, $(\phi_{jk})_{j,k\in\mathbb{Z}}$ such that :

$$\phi_{jk}(t) = 2^{\frac{j}{2}}\phi(2^j t - k)\phi \in \mathbb{L}^2(\mathbb{R})$$

ϕ is called the *scaling function*.

Stéphane Mallat suggested an algorithm allowing a very fast wavelet decomposition computation.
The theoretical framework is based on *Multiresolution Analysis* ; We call Multiresolution Analysis a family $(V_j)_{j\in\mathbb{Z}}$ of $\mathbb{L}^2(\mathbb{R})$ subsets with the following properties :

$$\begin{aligned} V_j &= \{\sum_{k\in\mathbb{Z}} a_k\phi_{jk} : a_k \in \mathbb{R}\} \\ V_j &\subset V_{j+1} \\ \bigcap_{j\in\mathbb{Z}} V_j &= \{0\} \\ \overline{\bigcup_{j\in\mathbb{Z}} V_j} &= \mathbb{L}^2(\mathbb{R}). \end{aligned}$$

Since $\phi \in V_0 \subset V_1$, $\exists\ (m_0[k])_{k\in\mathbb{Z}}$ such that :

$$\phi(t) = 2\sum_{k\in\mathbb{Z}} m_0[k]\phi(2t - k).$$

We call the sequence (m_0) the first filter. This notion will be fundamental when processing fast wavelet decomposition.

Wavelets are seen as a way to describe *at level j* the difference between the two subsets V_j *and* V_{j+1}. It already underlines the fact that every decomposition depends on a level that we will have to specify when computing the discrete transform.

A particularly interesting level is the level 0 : $\exists W_0 | V_0 \oplus W_0 = V_1$.

And there exists a function ψ, called *mother wavelet*, that generates W_0 :

$$W_0 = \{t \mapsto \sum_{k\in\mathbb{Z}} d_k\psi(t - k) : d_k \in \mathbb{R}\}$$

[2] When studying univariate signals, we will only be interested in the case $n = 1$.

Since $\psi \in V_1$, there is a sequence (m_1) such that :

$$\psi(t) = \sum_{k\in\mathbb{Z}} m_1[k]\phi(2t-k)$$

(m_1) is the second filter.

The interesting thing about these filters is that their specification provides the whole information about the wavelet decomposition, through the following formulas:

$$\hat{\phi}(w) = \prod_{k=1}^{+\infty} m_0(\frac{w}{2^k})$$
$$\hat{\psi}(w) = m_1(\frac{w}{2})\hat{\phi}(\frac{w}{2})$$

Proposition 2. Let us note W_j such that $\forall j \in \mathbb{Z}, V_{j+1} = V_j \oplus W_j$.
Then,

$$\mathbb{L}^2(\mathbb{R}) = \bigoplus_{j=-\infty}^{+\infty} W_j = V_j \oplus \bigoplus_{k=j}^{+\infty} W_k$$

Remark 1. The principle of the discrete wavelet transform at level j consists in decomposing the signal in an adapted basis with respect to the second direct sum. The component of the signal on V_j is called the *approximate signal, at a level of resolution* j whereas the other component is the *detailed* signal. Note that, considering the way the $(V_j)_{j\in\mathbb{Z}}$ are constructed, when the resolution tends to $+\infty$, the approximation signal tends to contain all the information.

A.3 Computing the decomposition

A.3.1 Function decomposition in $\mathbb{L}^2(\mathbb{R})$

Now that we have correctly defined our framework, let us present the method to compute a fast discrete transform.[3]
As previously shown, there is obvious intricate connection between the subsets, making it completely natural to reason iteratively, and to compute at each step the following basis shift :

$$V_{j+1} \leftrightarrow V_j \oplus W_j$$

Since $(\phi_{j,k})_{k\in\mathbb{Z}}$ is a basis of V_j and $(\psi_{j,k})_{k\in\mathbb{Z}}$ a basis of W_j, the problem consists in finding the following transformation :

$$\{\phi_{j+1,k} : k \in \mathbb{Z}\} \leftrightarrow \{\phi_{j,k} : k \in \mathbb{Z}\} \cup \{\psi_{j,k} : k \in \mathbb{Z}\}$$

With $(a_{j+1,k})_{k\in\mathbb{Z}}$, (respectively $(a_{j,k})_{k\in\mathbb{Z}}$ and $(d_{j,k})_{k\in\mathbb{Z}}$), the coordinates of a signal in V_{j+1} (resp. V_j and W_j), the problem is equivalent to:

$$l_2(\mathbb{Z}) \to l_2(\mathbb{Z}) \times l_2(\mathbb{Z})$$
$$(a_{j+1,k})_{k\in\mathbb{Z}} \to [(a_{j,k})_{k\in\mathbb{Z}}; (d_{j,k})_{k\in\mathbb{Z}}]$$

[3]This transform is even faster than a classic Fast Fourier Transform (FFT).

The filters (m_0) and (m_1) will play a key role in the computation of the discrete wavelet transform and its inverse.
Indeed, for any $j \in \mathbb{Z}$, we have the following transform formulas :

Inverse wavelet transform:

$$a_{j+1,k} = \frac{1}{2} \sum_{l \in \mathbb{Z}} m_0[2l-k]a_{j,l} + m_1[2l-k]d_{j,l}$$

Forward wavelet transform:

$$a_{j,k} = 2 \sum_{l \in \mathbb{Z}} \tilde{m}_0[l]a_{j+1,2l-k}$$

$$d_{j,k} = 2 \sum_{l \in \mathbb{Z}} \tilde{m}_1[l]d_{j+1,2l-k}$$

Forward wavelet tranform uses *dual* filters: $(\tilde{m}_0)$ and $(\tilde{m}_1)$.
We will not go into further detail concerning these filters ; simply note that these dual filters only depend on (m_0) and (m_1).

A.3.2 Vector decomposition in $\mathbb{R}^n$

In practice, when processing signals, we only have sampled values at times [t_1,...,t_n] - so that we deal with elements of $\mathbb{R}^n$ instead of $\mathbb{L}^2(\mathbb{R})$. In this case, there is a level of resolution j such that the observed signal $y \in V_j$. In other words, there is a level of resolution which contains the whole information.

Suppose that we want to represent the vector y on level $j-2$.

- *First Step :* we project the signal on $V_{j-1} \oplus W_{j-1}$.
- *Second Step :* we project the signal on $V_{j-2} \oplus W_{j-2} \oplus W_{j-1}$.

We obtain 3 vectors cA (the signal projected on V_{j-2}), cD_{j-2} and cD_{j-1} (the signal projected respectively on W_{j-2} and W_{j-1}) ; we proceed iteratively to finally obtain vectors $cA, cD_1, ..., cD_{j-1}$.

Concatenated vector $\lambda = [cA, cD_1, ..., cD_{j-1}]^T$ represents the wavelet decomposition of the signal y, in our model.
This decomposition allows us to handily create the transposed wavelet dictionary, by implementing it on the canonical basis of $\mathbb{R}^n$: we obtain a matrix G, such that every decomposition λ of signal y can be written $\lambda = Gy$. G^T becomes our explanatory dictionary, using its column as atoms.

Operators for wavelet dictionary creation have been implemented, in the file `Classes.py`.

A.4 Example : Daubechies Wavelet

The Daubechies wavelets, based on the work of Ingrid Daubechies, are a family of orthogonal wavelets defining a discrete wavelet transform and characterized by a maximal number of vanishing moments for some given support.
With each wavelet type of this class, there is a scaling function which generates a multiresolution analysis.

Ingrid Daubechies searched a function ψ such that $(\psi_{j,k})_{j,k\in\mathbb{Z}}$ is an orthogonal basis of $\mathbb{L}^2(\mathbb{R})$. She classified the wavelets formed depending on the number of vanishing moments of the mother wavelet.
There are two naming schemes in use, DN using the length or number of taps, and dbA referring to the number of vanishing moments. So $D4$ and $db2$ are the same wavelet transform, for example.

Daubechies wavelets are of common use when dealing with regular signals. These are the most widespread class of wavelet when one want to process physiological signals.

To give a graphical insight of the wavelets used in this project, here is the scaling function and mother wavelet from Daubechies with 5 and 17 vanishing moments :

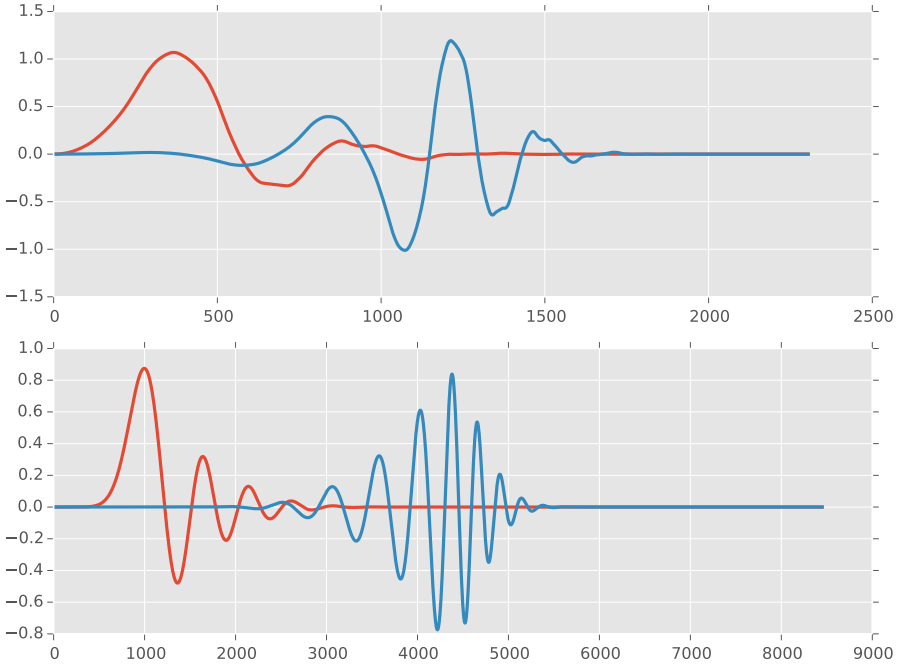

Figure A.1: Upper Graph : *Daubechies 3 scaling and mother wavelet functions.*
Lower Graph : *Daubechies 5 scaling and mother wavelet functions.*

Appendix B

Estimated variance model figures

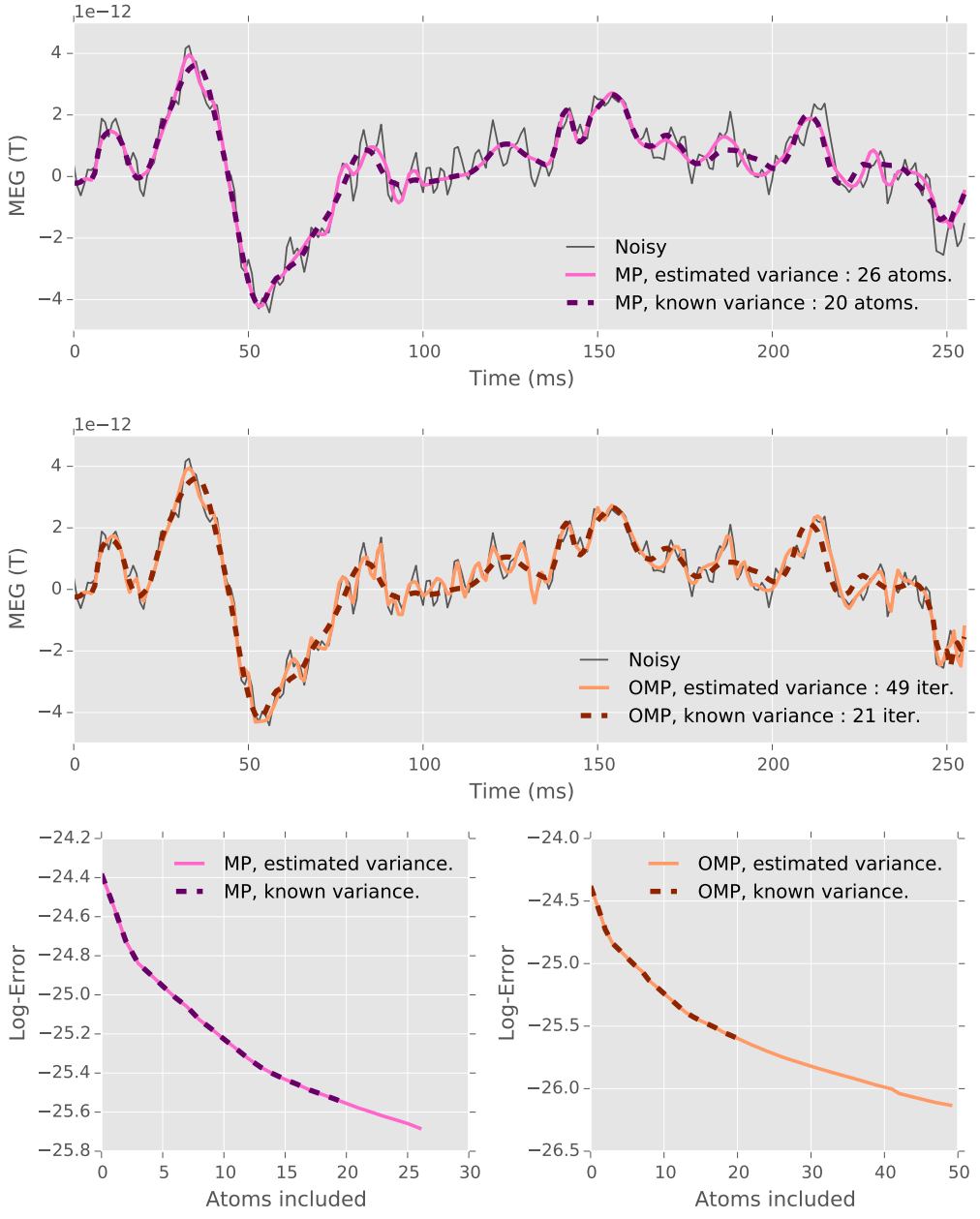

FIGURE B.1: UPPER GRAPHS : *Denoised results via estimated,known-variance models.* LOWER GRAPHS : *Associated logarithmic errors (in norm).* `db5` *dictionary.*

We present in this appendix figures obtained using our estimated-variance model. They represent single-channel processings, on the 100^{th} sensor, using the two main dictionaries of this study, `db5` and `db17`.

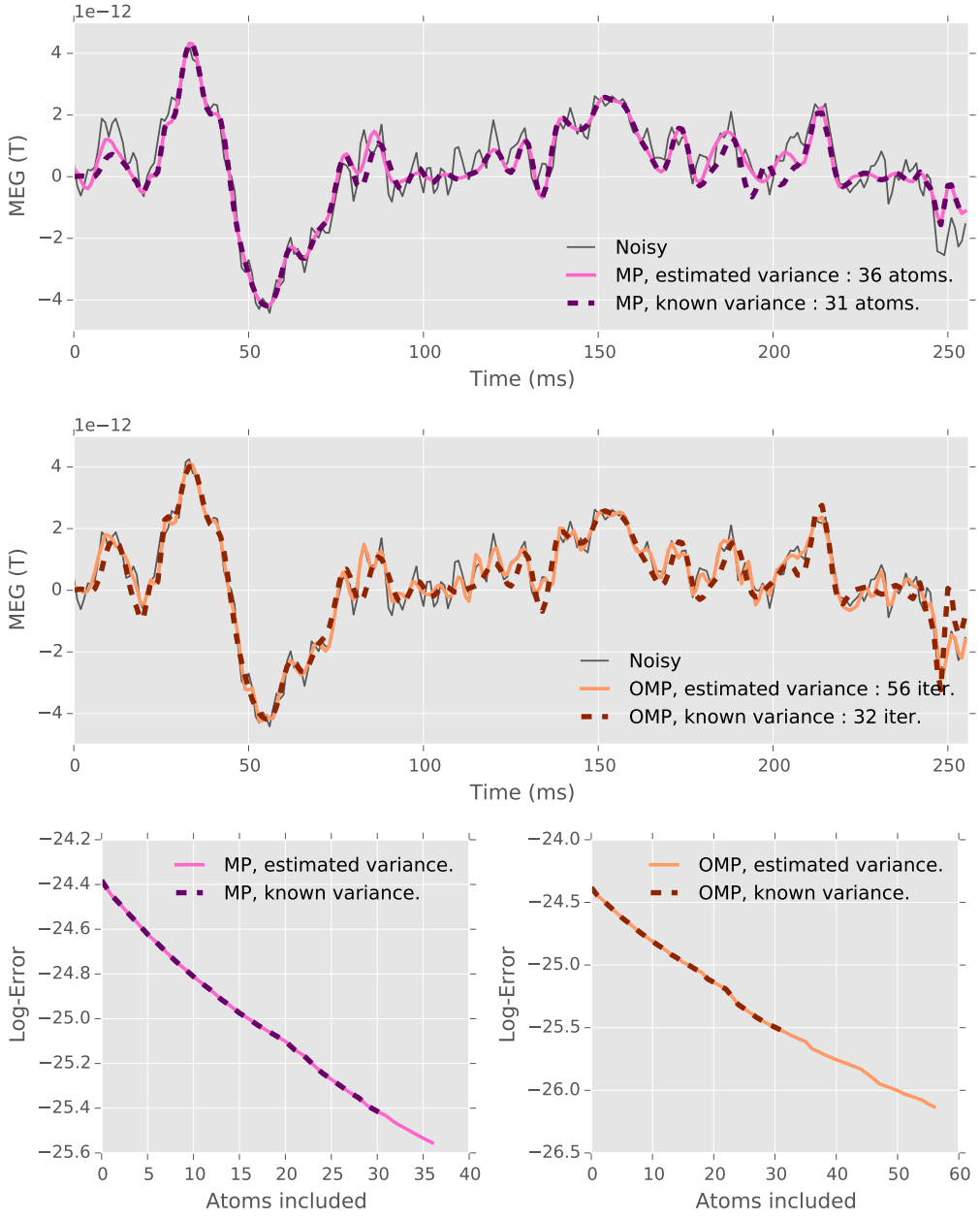

FIGURE B.2: UPPER GRAPHS : *Denoised results via estimated,known-variance models.* LOWER GRAPHS : *Associated logarithmic errors (in norm).* `db17` *dictionary.*

Appendix C

Detailed multi-channel results

Following results were obtained using Daubechies wavelets `db17` dictionary, and multi-channel Orthogonal Matching Pursuit, on the entire range of sensors. We present here results for sensors 50 to 59 for brevity.

C.1 Pre-experiment & residual variances

Sensor	Pre-exp.	Residual
Sensor 50	6.127	2.315
Sensor 51	7.500	2.430
Sensor 52	6.168	1.134
Sensor 53	6.429	1.032
Sensor 54	6.048	1.338

Sensor	Pre-exp.	Residual
Sensor 55	5.595	1.184
Sensor 56	7.565	2.294
Sensor 57	1.998	1.073
Sensor 58	5.414	0.986
Sensor 59	0.981	0.937

TABLE 1. *Pre-experiment & residual variances, multi-channel OMP, sensors 50-59. Unit :* 10^{-25} *Tesla*^{2}.

C.2 Lilliefors test results

Pre-exp.	K-statistic	P-value
Sensor 50	0.050	0.289
Sensor 51	0.058	0.128
Sensor 52	0.039	0.646
Sensor 53	0.048	0.347
Sensor 54	0.035	0.842
Sensor 55	0.058	0.126
Sensor 56	0.051	0.263
Sensor 57	0.045	0.456
Sensor 58	0.047	0.363
Sensor 59	0.071	0.256

Residual	K-statistic	P-value
Sensor 50	0.034	0.626
Sensor 51	0.031	0.760
Sensor 52	0.038	0.417
Sensor 53	0.039	0.396
Sensor 54	0.058	0.318
Sensor 55	0.038	0.428
Sensor 56	0.041	0.335
Sensor 57	0.030	0.798
Sensor 58	0.036	0.526
Sensor 59	0.034	0.604

TABLE 3, LEFT. *Lilliefors testing results for pre-experiment acquisition, sensors 50-59.*
TABLE 4, RIGHT. *Lilliefors testing results for multi-channel OMP residuals, sensors 50-59.*

C.3 Multi-channel Orthogonal Matching Pursuit figures

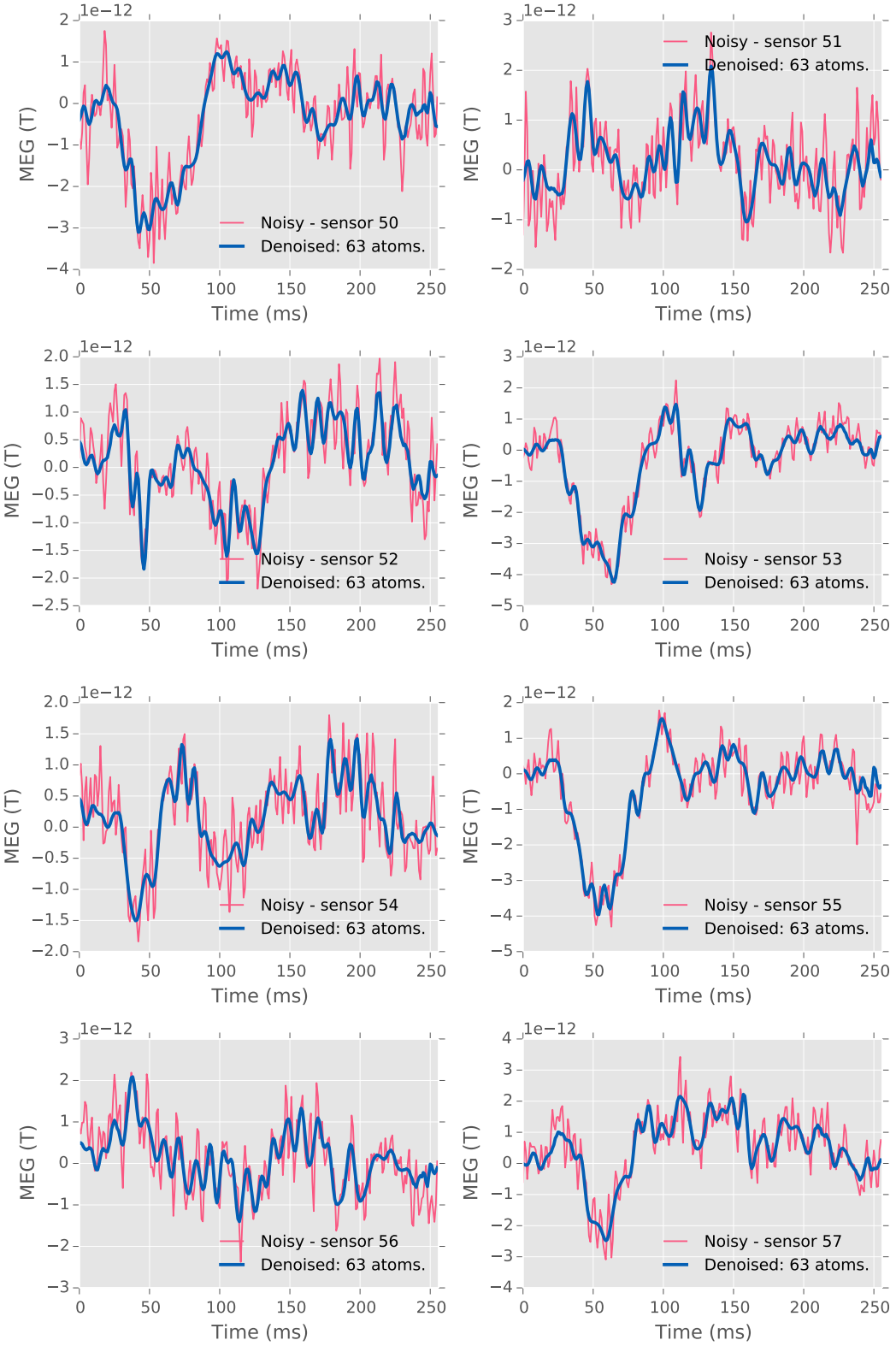

FIGURE C.1: *Original acquisitions & Multi-channel OMP denoised signals, sensors 50-57.*

Bibliography

[1] S. Mallat and S. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 1993.

[2] G. Peyre. Sparse regularization with matching pursuits. *A Numerical Tour of Signal Processing, Ceremade.*

[3] P. Durka, A. Matysiak, E. Martinez-Montes, P. Valdés-Sosa, and K. Blinowska. Multichannel matching pursuit and eeg inverse solutions. *Journal of Neuroscience Methods*, 2005.

[4] M. Aharon, M. Elad, and A. Bruckstein. K-svd: Design of dictionaries for sparse representation. *Israel Institute of Technology*, 2006.