JOURNAL OF APPLIED
COMPUTER SCIENCE
Vol. 26 No. 2 (2018), pp. 187-199

Normalization Effects in Matching Pursuit Algorithm with Gabor Dictionaries

Piotr T. Różański

College of Inter-Faculty Individual Studies
in Mathematics and Natural Sciences (MISMaP)
University of Warsaw
ul. Stefana Banacha 2C, 02-097 Warszawa
ptr@mimuw.edu.pl

Abstract. *The matching pursuit (MP) algorithm is a greedy method for signal decomposition used in video coding, data compression, and, particularly, analysis of EEG signals in various paradigms, including P300 and ER(D)S (motor imagery). An important issue for MP implementation is a correct treatment of normalization of atoms (functions) used in computations. Failing to account for normalization-related effects may affect both the numerical stability and the reliability of the algorithm. This paper describes these normalization effects, evaluates their impact on the algorithm's performance, and describe the proper approach together with a ready-to-use implementation, available under a General Public Licence (GPL). Several performance optimizations used as a part of this implementation are also described.*

Keywords: *matching pursuit, time-frequency, wavelet, EEG analysis.*

1. Introduction

1.1. Matching pursuit

The Matching Pursuit (MP) algorithm was first proposed by Mallat and Zhang [1] as a greedy solution to decompose a given signal x into a linear combination of functions (g_n) from a predefined set D, called a dictionary:

$$x = \sum_{n=0}^{M} \alpha_n g_n \,. \tag{1}$$

In general, problem (1) cannot be solved in polynomial time, if functions in the dictionary do not form an orthogonal set. However, a greedy procedure can be defined as

$$\begin{aligned} R_0 x &= x \\ R_{n+1} x &= R_n x - \langle R_n x, g_n \rangle \, g_n \\ g_n &= \operatorname{argmax}_{g_n \in D} \langle R_n x, g_n \rangle \ , \end{aligned} \tag{2}$$

where x is the signal being decomposed, $R_n x$ is a residual before n-th iteration (starting from $n = 0$), and g_n is the atom selected in n-th iteration. The above formulation assumes that all atoms $g \in D$ are L^2-normalized, i.e. $\langle g(t), g(t) \rangle = \int g(t)^2 dt = 1$.

In each step, the best approximation to the current residual $R_n x$ is chosen from the "dictionary" D and subtracted from the signal after being multiplied (fitted) with an adaptive scale factor.

It is worth noting, that since the matching pursuit does not provide an exact solution for (1), it should be referred to as "heuristic". However, it is traditionally referred to as an algorithm, and this article will refer to it as such.

The usual choice for the atoms (functions) forming the dictionary is the family of the Gabor atoms $g(s, f_0, t_0, \phi)$ defined as

$$g_{(s,f_0,t_0,\phi)}(t) = K_r \, e^{-\pi\left(\frac{t-t_0}{s}\right)^2} \cos(2\pi f_0 (t - t_0) + \phi) \,,$$

where K_r is the normalization constant.

There are two main reasons for the choice of Gabor atoms. Firstly, the Gabor atoms have the most compact representation on a time-frequency plane, and therefore are a natural candidate for using matching pursuit in calculating high-resolution estimates of the time-frequency distribution of the signal's energy. This approach has been successfully used to describe the time-frequency microstructure of EEG event-related (de-)synchronization in [2].

Secondly, the Gabor atoms and multivariate (multi-channel) matching pursuit have proven to be very useful in solving some advanced problems in EEG analysis, such as parametrization of the single-trial evoked potentials [3], providing efficient pre-processing for the inverse problem in EEG [4], or parametrization of EEG transients [5].

1.2. Dictionary construction

We start by defining a dictionary for the MP algorithm, by adapting the "optimal dictionary" construction from [6]. The construction is based on a single parameter ϵ, related to the density of the dictionary. Smaller values of ϵ correspond to more fine-grained dictionaries (consisting of a larger number of atoms). Generally, ϵ should be close to 0 (e.g. 0.1) for accurate decomposition.

Similarly to the original dyadic dictionary by Mallat [1], the scale parameter varies exponentially, starting from a pre-defined minimal scale s_0 (which could be problem-specific): $s_0, s_0 a, s_0 a^2, \ldots$ whereas the dilation factor a is defined as

$$a = \frac{1 + \epsilon\sqrt{(2-\epsilon^2)(\epsilon^4 - 2\epsilon^2 + 2)}}{(1-\epsilon^2)^2}\,. \tag{3}$$

For a given scale s, frequency (f_0) and postion (t_0) parameters form a regular, rectangular grid with spacing in both directions defined as

$$\Delta f = \frac{1}{s}\sqrt{-\frac{2}{\pi}\log(1-\epsilon^2)} \tag{4}$$

$$\Delta t = s\sqrt{-\frac{2}{\pi}\log(1-\epsilon^2)}\,. \tag{5}$$

The phase parameter ϕ is not taken into account due to the "phase-related equivalence" described in [6] and the reasons which shall be described later in this manuscript.

Such construction of the dictionary guarantees the constraints (in terms of the inner-product-related metric) between adjacent atoms, i.e.

$$\begin{aligned}\sqrt{1 - \langle g(s, f_0, t_0), g(sa, f_0, t_0)\rangle} &\le \epsilon \\ \sqrt{1 - \langle g(s, f_0, t_0), g(s, f_0 + \Delta f, t_0)\rangle} &\le \epsilon \\ \sqrt{1 - \langle g(s, f_0, t_0), g(s, f_0, t_0 + \Delta t)\rangle} &\le \epsilon\,.\end{aligned} \tag{6}$$

2. Normalization

Since in formula (2) it is assumed that all atoms g are L^2-normalized, it is crucial to fulfill this assumption for all atoms in the dictionary. The normalization factor for real Gabor atoms, which can be derived from $\langle g, g\rangle = 1$, is equal to

$$K_r = \frac{2^{3/4}}{\sqrt{s}} \left(1 + \cos(2\phi) \exp(-2\pi s^2 f_0^2)\right)^{-1/2} . \tag{7}$$

This value differs from the normalization value for the envelope function $e^{-\pi\left(\frac{t}{s}\right)^2}$ by a constant factor of $\sqrt{2}$ and introduces the additional factor involving an atom's phase. We can note that for sufficiently large frequencies ($sf_0 \gg 1$) the normalization constant is asymptotically phase-independent:

$$K_{r\mathrm{HF}} = \lim_{f_0\to\infty} K_r = \frac{2^{3/4}}{\sqrt{s}} . \tag{8}$$

However, in case of discrete-time signals, the above formulae are valid only in the range of low frequencies. Whenever f_0 approaches the Nyquist frequency f_N, the time discretisation effects come into account and the above formula is no longer accurate.

To derive the correct formula for a fast-oscillating atom with frequency $f_0 = f_N - \delta f$ (with $\delta f \ll f_N$), one can rewrite the oscillating factor of the Gabor atom as

$$\begin{aligned} &\cos(2\pi(f_N - \delta f)(t - t_0) + \phi) = \\ &= (-1)^n \cos(2\pi\delta f(t - t_0) + (2\pi f_N t_0 - \phi)) . \end{aligned}$$

Therefore, for $f \approx f_N$ there is a need to calculate "corrected" phase and frequency

$$f' = f_N - f \tag{9}$$

$$\phi' = 2\pi f_N t_0 - \phi \tag{10}$$

so these values can be used exclusively to calculate the corrected normalization factor

$$K_r' = \frac{2^{3/4}}{\sqrt{s}} \left(1 + \cos(2\phi') \exp(-2\pi s^2 f_0'^2)\right)^{-1/2} . \tag{11}$$

Plot 1 compares the different values of normalization factor: the analytically-derived value of K_r, the phase-independent approximation $K_{r\mathrm{HF}}$, and the corrected value of high-frequency normalization factor K_r'.

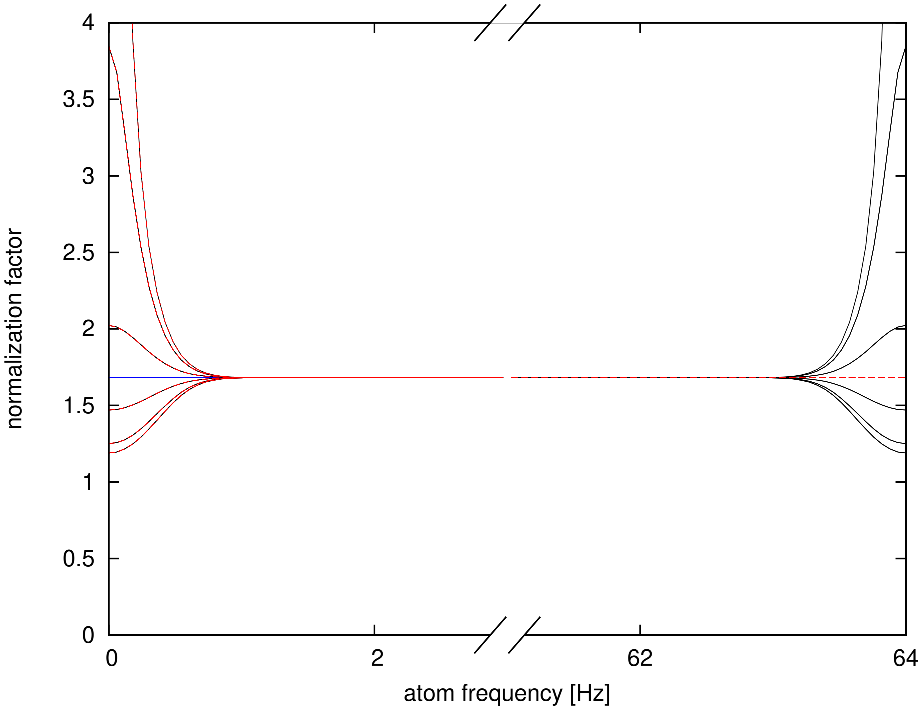

Figure 1: Values of normalization factor K for different normalization strategies (blue—$K_{r\mathrm{HF}}$, red—K_r, black—combined K_r and K_r') for $s = 1\,\mathrm{s}$ and sampling frequency of 128 Hz. Multiple branches of the same colour correspond to different values of ϕ.

Even though the difference between K_r and K'_r is significant only for $f \approx f_N$, atoms of such frequencies appear in real-world signal decompositions on a regular basis. If the normalization is not accounted for correctly, two effects may appear:

- selecting (and substracting) high-frequency atoms will still leave some non-zero residual in the signal, allowing it to be selected as the best match also in subsequent iterations,
- high-frequency atom may be selected instead of the correct one if the error in K_r estimation leads to over-estimating $\langle R_n x, g_n \rangle$ with the current residual.

The proper analytical treatment of normalization would be to use formula K_r for small frequencies ($f < \frac{1}{2} f_N$) and formula K'_r for large frequencies. Errors in normalization factor values obtained this way, relative to the values obtained numerically for the discretely-sampled functions are presented in plot 2. The proposed approximation is quite sufficient even for double-precision calculation, with relative errors not exceeding 10^{-12}.

2.1. Evaluation

As a test, 100 matching pursuit decompositions of randomly-generated white noise signal segments were performed. Each signal segment consisted of 2048 samples with sampling frequency of 128 Hz. Figure 3 visualizes decomposition accuracy, defined as

$$1 - \frac{\text{residual energy}}{\text{signal energy}} = 1 - \frac{\sum_i (R_n x)_i^2}{\sum_i x_i^2} \tag{12}$$

(where $\sum_i$ is a summation over all signal samples) as a function of the number of iterations n, both for correct and incorrect (based only on K_r) normalization strategies.

Without the correct treatment of atom normalization, all white noise decompositions demonstrated incorrect behaviour. After several initial steps, the algorithm usually started to select the same high-frequency atom in subsequent iterations.

To evaluate this effect on a more realistic example, a similar analysis was performed on eighteen 20-second segments of a single channel EEG recording. The signal was filtered with a high-pass filter to remove DC offset and drift. Results are presented in Figure 4. Although the difference is less pronounced than in the case of white noise decomposition, it is still significant.

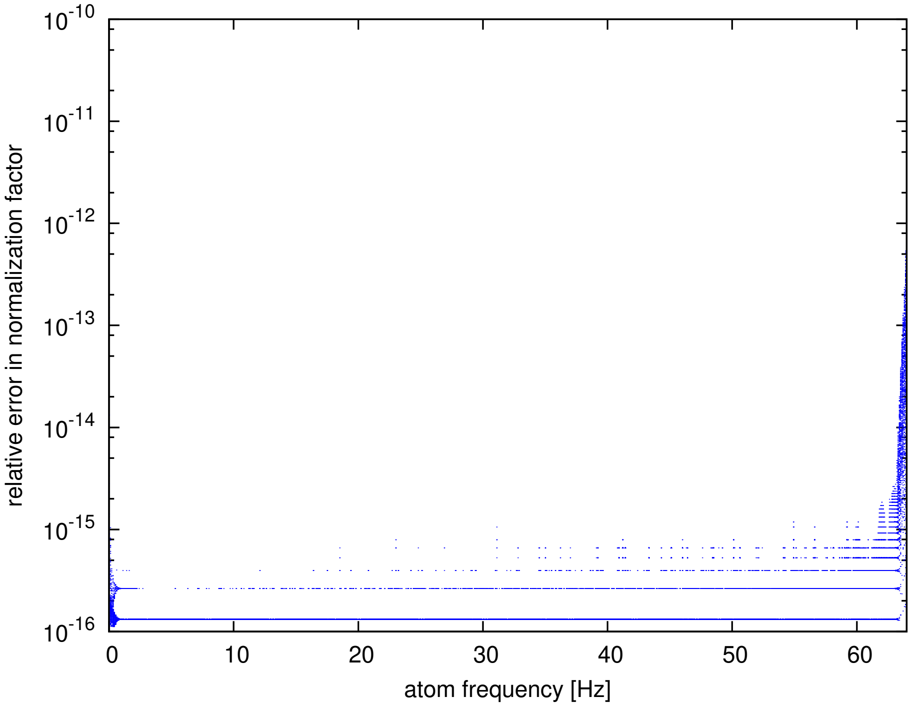

Figure 2: Relative error for the proposed normalization strategy for $s = 1\,\mathrm{s}$ and sampling frequency of 128 Hz.

2.2. Discussion

Since the standard approach in biomedical signal analysis is to apply low-pass filtering prior to the MP decomposition, the erroneous behavior is probably unnoticed in most cases, and therefore neglected in many implementations. However, even for the filtered signals, high frequency atoms can still appear because of accumulated numerical residues from previous iterations.

The alternative approach to this problem would be to numerically re-normalize the best atom found in a given iteration to acquire the correct value of α (see eq. 1). However, this may lead to selecting a sub-optimal atom in any given iteration, since un-normalized products $\langle R_n x, g_n \rangle$, used as a criterion for selecting the best atom, could not be properly compared.

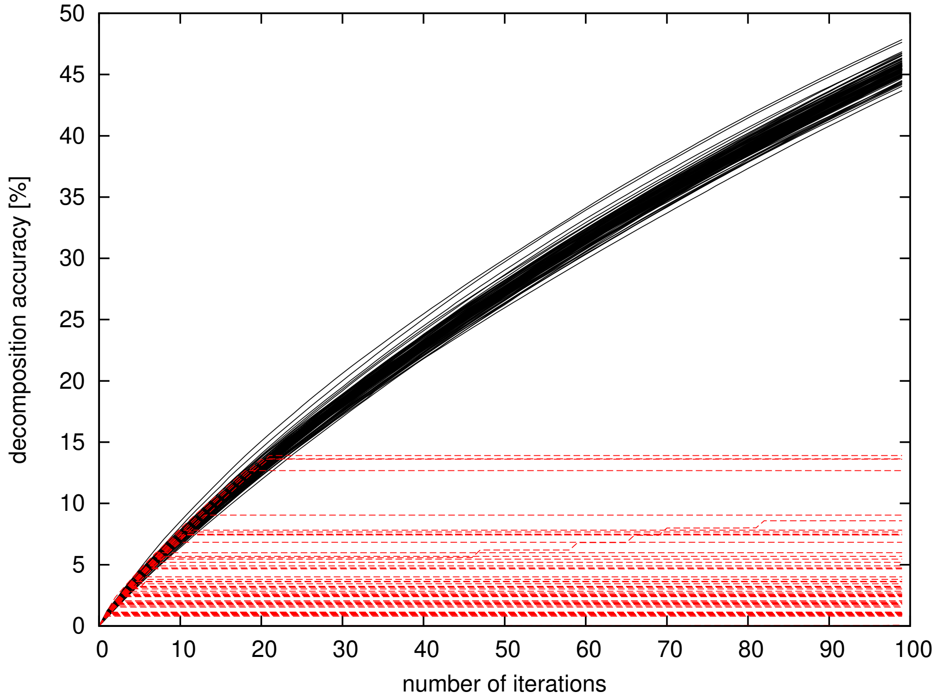

Figure 3: Decomposition quality for white noise signals for K_r-only (dashed red line) and correct (solid black line) normalization strategies.

3. Implementation

The normalization-aware version of MP algorithm has been implemented as an autonomous software package *empi*, written in C++ (2011 standard). The only external dependency is the FFTW library, chosen as the implementation of the Fast Fourier Transform due to its superior performance and compatibility with various architectures.

To take advantage of multi-threading (or multi-processor) architecture available currently in virtually all customer-grade personal computers, OpenMP parallelization has been introduced.

The software (current version: 0.4.1) is available from `https://github.com/develancer/empi` under a General Public Licence (version 3) as both C++

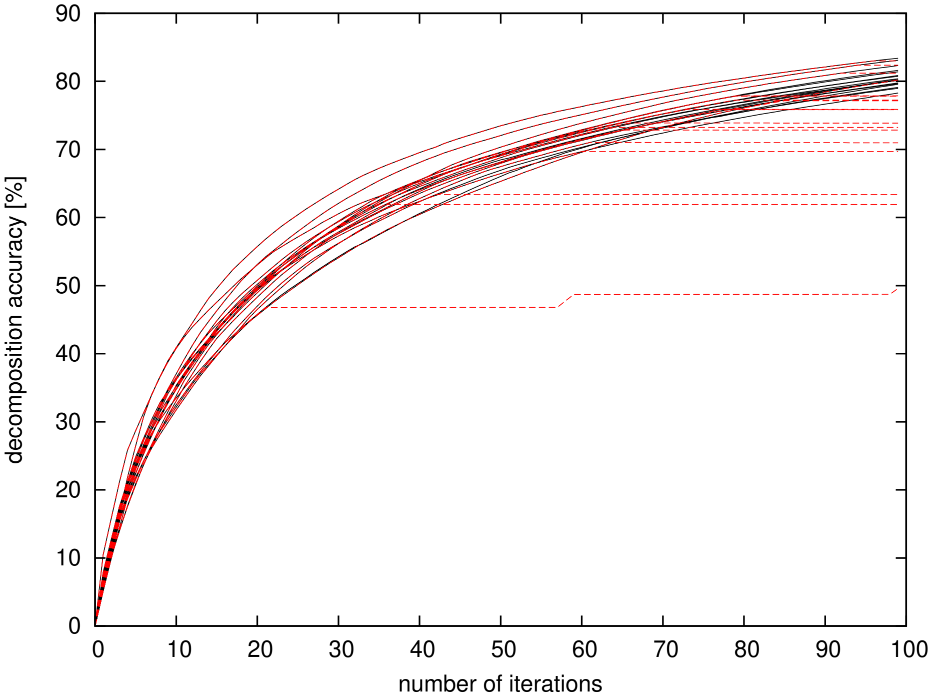

Figure 4: Decomposition quality for EEG signal segments for K_r-only (dashed red line) and correct (solid black line) normalization strategies.

source code and precompiled binaries for Linux and Microsoft Windows[1].

Apart from the correct treatment of atom normalization, a number of optimizations were implemented in *empi*. Some of the optimizations can be applied not only to matching pursuit with Gabor dictionaries, but to much wider spectrum of signal analysis algorithms (e.g. wavelet analysis), and therefore will be described in the following section.

3.1. Fast Fourier Transform

We introduce *complex* Gabor atoms as

$$G(s, f_0, t_0) = K_c \exp\left(-\pi\left(\frac{t-t_0}{s}\right)^2 + 2\pi \mathrm{i} f_0(t-t_0)\right), \tag{13}$$

[1] both 32-bit and 64-bit OS versions are supported

where K_c is a (much simpler) normalization constant, identical to the normalization factor of the envelope function alone:

$$K_c = \frac{2^{1/4}}{\sqrt{s}} . \tag{14}$$

The relation between complex and real Gabor atoms is pretty straightforward:

$$g(s, f_0, t_0, \phi) = \frac{K_r}{K_c} \,\Re(G(s, f_0, t_0) e^{\mathrm{i}\phi}) , \tag{15}$$

where $\Re$ stands for the real part of the complex number. This simple substitution allows to perform a major part of the computations on complex Gabor atoms.

This approach allows us to make an important optimization. The full formula for the scalar product of the signal with a complex Gabor atom can be written as

$$\langle x, G(s, f_0, t_0) \rangle = K_c \sum_t x(t) e^{-\pi\left(\frac{t-t_0}{s}\right)^2} e^{2\pi \mathrm{i} f_0 (t-t_0)} , \tag{16}$$

which is essentially the formula for a windowed discrete Fourier transform with a Gaussian window function. Therefore, the calculations may be performed with Fast Fourier Transform implementation, resulting in a significant decrease of the computation time. It is worth noting, that this technique may be applied not only to Gabor atoms, but any family of atoms consisting of an oscillating factor and the “envelope” factor with finite support.

3.2. Phase optimization

For given (s, f_0, t_0), the optimal phase ϕ_{opt} maximizing the inner product can be calculated as

$$\phi_{\mathrm{opt}} = \arg \langle x, G(s, f_0, t_0) \rangle . \tag{17}$$

With such phase,

$$\left| \left\langle x, g(s, f_0, t_0, \phi_{\mathrm{opt}}) \right\rangle \right| = \frac{K_r}{K_c} \left| \langle x, G(s, f_0, t_0) \rangle \right| . \tag{18}$$

Therefore, it is not necessary to include phase parameter in dictionary construction, as the optimal phase ϕ_{opt} can be found in a straightforward manner for any given set of parameters (s, f_0, t_0).

3.3. Priority queue

To improve the selection of the optimal atom in each step, similarly to the approach used in [7], a priority queue was introduced to store (and update) current values of $\langle R_n x, g \rangle$ for each atom g in the dictionary. In each iteration, the best fit can be selected in $O(1)$ time by the FIND-MAX operation introduced in the standard MAX-HEAP implementation from [8].

After selecting an optimal atom in each iteration, values stored in the priority queue have to be updated, with the help of an additional DECREASE-KEY operation. However, from the formula

$$\langle R_{n+1} x, g \rangle = \langle R_n x - \alpha g_n, g \rangle = \langle R_n x, g \rangle - \alpha \langle g_n, g \rangle \tag{19}$$

it is clear that the inner products between updated residual and all atoms g orthogonal to currently selected g_n ($\langle g_n, g \rangle \approx 0$) do not require to be recalculated between iterations.

This optimization is utilized by noticing that the absolute value of the inner product between Gabor atoms (both real and complex) is bounded by the inner product of their Gaussian envelopes. For every two complex Gabor atoms $G_1 = G(s_1, f_1, t_1)$ and $G_2 = G(s_2, f_2, t_2)$, this upper bound can be calculated as:

$$|\langle G_1, G_2 \rangle| \leq \sqrt{\frac{2 s_1 s_2}{s_1^2 + s_2^2}} \exp\left(\frac{-\pi (t_1 - t_2)^2}{s_1^2 + s_2^2} \right). \tag{20}$$

Therefore, if the above scalar product is estimated to be below given threshold (e.g. 10^{-15}), the pair of atoms may be treated as orthogonal and no inner product update have to be performed.

3.4. Multivariate Matching Pursuit

In case of analysing multichannel data, which is usually the case in EEG analysis, it is useful to take into account possible relations between simultaneous data in different channels. Therefore, the resulting decomposition of each channel will depend also on the data in every other channel. More specifically: in every iteration, atoms selected for all channels share the same values of parameters s, f_0 and t_0. Three variants of multivariate MP were implemented, as defined in [6]:

- MMP1: in every iteration, atoms selected for all channels share the same phase ϕ and the parameters maximize the sum of the moduli of inner products $\sum_c |\langle R^n x_c, g(s, f_0, t_0, \phi) \rangle|$.

- MMP2: in every iteration, atoms selected for all channels share the same phase ϕ and the parameters maximize the sum of the inner products $\sum_c \langle R^n x_c, g(s, f_0, t_0, \phi)\rangle$.

- MMP3: in every iteration, atoms selected for all channels are allowed to have different phases, and the parameters maximize the sum of the moduli of inner products $\sum_c |\langle R^n x_c, g(s, f_0, t_0, \phi_c)\rangle|$.

Due to the linearity of the inner product, decomposition in MMP2 variant can be performed on a single signal, constructed as a sum of all channels, since

$$\sum_c \langle R^n x_c, g(s, f_0, t_0, \phi)\rangle = \left\langle \left(\sum_c R^n x_c\right), g(s, f_0, t_0, \phi)\right\rangle .$$

This feature of MMP2 allows for a significant speed-up, compared to MMP1. After each iteration, the selected atom g_n has to be projected onto every channel to calculate the coefficients α (see eq. 1) for each channel. This additional step, however, is not computationally expensive.

4. Conclusions

This paper studies the effect of atom normalization on the performance and reliability of the matching pursuit algorithm. It was shown that the incorrect treatment of normalization may impede both the numerical stability of the algorithm, as well as its key feature—selecting the optimal atom at each step. A semi-analytical normalization strategy has been evaluated and shown to be accurate with the relative error not exceeding 10^{-12}. Also, a ready-to-use C++ implementation, combining the introduced normalization strategy with a range of described optimization, has been provided on an open-source licence.

5. Acknowledgements

This research was supported by the Polish National Science Centre (UMO-2015/17/B/ST7/03784).

References

[1] Mallat, S. G. and Zhang, Z., *Matching pursuits with time-frequency dictionaries*, IEEE Transactions on Signal Processing, Vol. 41, No. 12, Dec 1993, pp. 3397–3415.

[2] Durka, P. J., Ircha, D., Neuper, C., and Pfurtscheller, G., *Time-frequency microstructure of event-related electro-encephalogram desynchronisation and synchronisation*, Medical and Biological Engineering and Computing, Vol. 39, 2001, pp. 315–321.

[3] Sieluzycki, C., Konig, R., Matysiak, A., Kus, R., Ircha, D., and Durka, P. J., *Single-Trial Evoked Brain Responses Modeled by Multivariate Matching Pursuit*, IEEE Transactions on Biomedical Engineering, Vol. 56, No. 1, Jan 2009, pp. 74–82.

[4] Durka, P. J., Matysiak, A., Montes, E. M., Sosa, P. V., and Blinowska, K. J., *Multichannel matching pursuit and EEG inverse solutions.* Journal of neuroscience methods, Vol. 148 1, 2005, pp. 49–59.

[5] Durka, P. J., Malinowska, U., Zieleniewska, M., O'Reilly, C., Różański, P. T., and Żygierewicz, J., *Spindles in Svarog: framework and software for parametrization of EEG transients*, Front. Hum. Neurosci., 2015.

[6] Kuś, R., Różański, P. T., and Durka, P. J., *Multivariate matching pursuit in optimal Gabor dictionaries: theory and software with interface for EEG/MEG via Svarog*, BioMedical Engineering OnLine, Vol. 12, No. 1, Sep 2013, pp. 94.

[7] Krstulovic, S. and Gribonval, R., *MPTK: Matching Pursuit made Tractable*, In: Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP'06), Vol. 3, Toulouse, France, May 2006, pp. III–496 – III–499.

[8] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C., *Introduction to Algorithms, Third Edition*, The MIT Press, 3rd ed., 2009.