MAICA: an ICA-based method for source separation in a low-channel EEG recording

Izabela Rejer and Paweł Górski

Faculty of Computer Science and Information Technology, West Pomeranian University of Technology Szczecin, Szczecin, Poland
E-mail: irejer@wi.zut.edu.pl

Received 15 October 2018, revised 17 June 2019
Accepted for publication 30 July 2019
Published 11 September 2019

Abstract
Objective. The paper aims to present a method that enables the application of independent component analysis (ICA) to a low-channel EEG recording. The idea behind the method (called moving average ICA or MAICA) is to extend the original low-sensor matrix of signals by applying a set of zero-phase moving average filters to each of the recorded signals. Approach. The paper discusses the theoretical background of the MAICA algorithm and verifies its usefulness under three exemplary settings: (i) a pure mathematic system composed of ten source sinusoids; (ii) real EEG data recorded from 64 channels; (iii) real EEG data recorded from five subjects during 200 trials with motor imagery brain–computer interface. Main results. The first system shows that MAICA is able to decompose two mixed signals (composed of ten source sinusoids) into ten components with an extremely high correlation between the source patterns and identified components (99%–100%). The second system shows that when used over five channels, MAICA is able to recognize more artefact components than those recognized by classic ICA used over 64 channels. Finally, the third system demonstrates that MAICA is capable of working in an online mode without significant delays; the additional time needed to run MAICA for one trial was less than 6ms in the survey reported in the paper. Significance. The method presented in the paper should have a significant impact on all areas of medical signal processing where a large number of known and/or unknown patterns have to be retrieved in real time from complex signals recorded from a small number of external/internal body sensors.

Keywords: single channel ICA, independent component analysis, motor imagery, brain computer interface, EEG signal analysis, MAICA

1. Introduction

One of the crucial but as yet insufficiently addressed aspects of brain function monitoring is the issue of recording EEG signals outside of controlled laboratory conditions and without the supervision of qualified medical staff. Constant monitoring of cognitive functions in a noisy environment can be highly beneficial in the case of professions demanding highly focused and sustained attention, such as airline pilots [1] or drivers [2] or in the neuromarketing research [3]. Of course, the continuous monitoring of EEG signals outside of scientific laboratories, research centres or medical clinics is not an easy task. There are numerous issues that need to be addressed, and the two with the highest priority are the low signal quality in an outside-lab environment and insufficient user comfort.

We can usually increase the quality of a signal by applying spatial filtering algorithms [4] that allow us to find the estimated sources of the signal recorded from a user’s scalp. There are many different spatial filters. Some of them are nonparametric (e.g. the common average reference, surface Laplacian derivation, or bipolar derivation [5]), others have parameters that can be adapted to the user data (e.g. principal component analysis (PCA) [6, 7], minimum noise fraction (MNF) [7], common spatial patterns (CSP) [8, 9] and
independent component analysis (ICA) [10–14]). Although filters from both classes can improve the signal quality, much better results are obtained when data-driven filters are applied [5].

The linear mapping from sensors to sources provided by data-driven filtering techniques is usually more precise when more sensors are used. For example, Friman, Volosyak, and Gröser showed that in a brain–computer interface based on steady state visual evoked potentials (SSVEP-BCI), the classification accuracy increased from 69% to 84% when a MEC (minimum energy combination) spatial filter was applied over two and six channels, respectively [5]. Although in a scientific laboratory we can freely apply an arbitrary dense matrix of sensors distributed over the whole user’s head, we do not have this luxury in a home environment. Each additional sensor not only decreases user comfort but also makes the EEG sensors application less straightforward and more prone to errors. Thus, in a home environment, the EEG signal should be recorded using the smallest possible number of sensors, preferably distributed only over a small part of the user’s head. In summary, on the one hand, a large number of widely distributed sensors is needed to apply data-driven spatial filters increasing the signal quality, on the other hand only a few sensors located close to each other should be used to increase user comfort and to ensure the most convenient application of sensors. To fulfil both requirements, we need to find a way of applying spatial filtering techniques to a signal recorded from a sparse sensor setup. In this paper, we introduce a method that can be used to achieve this goal. The proposed method is called moving average independent component analysis (MAICA) and is an extension of classical ICA.

ICA is a linear spatial filter often used in EEG signal analysis to trace the artefacts. Although a great deal of research has confirmed that ICA enhances the signal-to-noise ratio (SNR) of EEG signal recorded from a high number of channels, for example 16 channels [15], 19–20 channels [16], 71 channels [17], much fewer studies have analysed the outcomes of ICA in a low-channel setup [18–29]. The reason for this is that to solve the linear ICA model, two conditions need to be fulfilled [18]:

1. The sources must be statistically independent, with at most one source with a Gaussian distribution;
2. The number of signals in the sensors’ domain should be at least as large as the number of sources that we want to extract.

The second condition implies that while for a 64-channel setup we are able to retrieve up to 64 components, in the case of a three-channel setup, only three components can be extracted from the data using classical ICA. Since the signals recorded by the sensors include not only brain activity, related and unrelated to the task at hand but also many types of internal and external artefacts, it is unreasonable to assume that decomposition into only three components would be sufficient to allow a decision on the origin of these components.

This leads to the underdetermined ICA model $(m > n$, where $m$ is the number of components and $n$ is the number of sensors). Several approaches to solving this model have been proposed in the literature. For example, in [30] a method was proposed for estimating a mixing matrix and inferring the most probable sources based on the expectation-maximisation algorithm [31]. In [32], the same problem was tackled by employing a sparse prior probability (in Laplacian form) on the basis coefficients to remove the redundancy inherent in an overcomplete representation. Even for the most extreme case of the underdetermined ICA model, it is for a single-channel model, several solutions have been proposed. One of the first, called simply SCICA (single channel ICA), was provided by Davies and James [22]. The authors proposed to solve the single-channel ICA model by dividing the signal recorded from a single sensor into a sequence of contiguous blocks and then perform ICA algorithm on the matrix composed of these blocks. Another proposition was provided in [19, 23], where the authors solved the single-channel ICA model by combining a non-linear dynamical systems framework with a standard implementation of ICA. The core of the proposed method, called the dynamical embedding method (DE), is to represent a single channel recording by a dynamical embedding matrix composed of a series of delayed vectors taken from this recording and then submit the matrix of delayed vectors to one of the ICA algorithms.

The next method, called EEMD-ICA (ensemble empirical-mode decomposition ICA), was described in [24, 25]. The method combines a modified version of EMD algorithm (EEMD) and FastICA. The EEMD algorithm defines a set of IMFs (intrinsic-mode functions) averaged over a given number of trials. In each trial, the EMD algorithm is applied on the original signal with added noise of a standard deviation (SD) being a fraction of the SD of the original signal. The set of averaged IMFs returned by EEMD constitutes a multi-channel representation of the original single-channel recording and is an input matrix for the FastICA algorithm.

One more approach to solving the underdetermined ICA model that should be mentioned here combines ICA and discrete wavelet transform (DWT) [24, 26–29]. Like in the previously mentioned methods, also this algorithm, called WICA (wavelet ICA), is composed of two parts. The first part aims to find a multi-channel representation of a single (or a few) channel recording by using wavelet transform. In [27] it is proposed to use a 3-level decomposition that provides a four-components representation of a single channel, where each component roughly corresponds to one of the four classic EEG subbands (delta, theta, alpha, and beta). As with the other approaches, the multi-component representation obtained as an output of the wavelet decomposition is then submitted to one of the ICA algorithms.

The aim of ICA is to decompose the matrix of input signals to the matrix of original sources. The ICA model, that is an example of a classic linear decomposition, assumes that the input signals are the mixtures of sources. To be more precise this model assumes that each signal is composed of the same sources but combined with different proportions. Although this ideal assumption does not hold in general for real signals, still most of the signals recorded from the sensors are composed of most of the source signals. When WICA or EEMD-ICA decompose the signal recorded from a single...
sensor into components, the ICA input matrix is composed of signals that are not related to each other. Hence, although both methods provide additional dimensions needed by ICA model to return more output components, they use ICA to transform a set of more or less independent components to another set of independent components. Such a transformation, without reinforcement of sources in the ICA input matrix, is less effective than the transformation performed on the input matrix composed of mixtures of original sources.

In this paper, we present a very simple yet fast and effective algorithm to solve the underdetermined ICA problem. The idea behind our approach, called MAICA, is to create artificial input signals by applying a set of zero-phase moving average filters to the recorded EEG channels and to use ICA on the dataset composed of original EEG channels and their filtered versions. The main advantage of MAICA is that it fulfills the assumption of preserving the mixtures of sources in the input matrix. By using moving average filters with the smallest possible windows, MAICA repeats most of the information that is contained in the signals recorded from sensors. In this way, not only the dimension of the input matrix is extended but also the linear transformation is much more effective (this will be shown in section 4.1).

The MAICA algorithm can be successfully applied to find signal components even in the case of a very sparse channel setup and under strict time constraints. Obviously, when the original data are recorded from a very small number of channels, we are not able to localise the sources spatially. Furthermore, we do not claim that all the estimated components are entirely independent. This, however, is not our concern; what we want to achieve by our method is the extraction of ‘interesting’ components, that is, components that can be recognised either as artefacts or as true brain sources corresponding to the task at hand.

The MAICA method is based on the multi-filtering algorithm that was first proposed in [33]. The current paper extends the original algorithm by proposing the approach for the automatic computation of filters’ parameters, provides the motivation for the whole method and compares its results with four other methods that can be used for creating a multi-channel representation of a low-channel signal. The main hypothesis of this paper can be formulated as follows: **MAICA is able to decompose a low-channel EEG signal into components in the same way as ICA does for a multi-channel signal.** The verification of this hypothesis will be carried out in three stages using three exemplary systems:

1. an artificial system composed of ten sinusoid components,
2. an EEG signal recorded from 64 channels,
3. an EEG signal recorded from two channels.

2. MAICA

The ICA problem can be stated as follows. We assume that there are $n$ linear mixtures $x_1, \ldots, x_n$ of $n$ independent components. A vector $x$ (observed signals) can be written as [10]:

$$x = As,$$  \hspace{1cm} (1)

where $A$ represents a mixing matrix with size $n \times n$, and $s$ is the vector of independent components. The aim of ICA is to find a matrix $W$ (i.e. an inverse of the matrix $A$) to reverse the mixing effect. Then, after computing the matrix $W$, the estimated independent components $y$ can be obtained by [34, 35]:

$$y = Wx \simeq s.$$ \hspace{1cm} (2)

The major drawback of classical ICA, which limits its use in the case of low-sensor recordings, is that the number of components returned by ICA algorithms is the same as the number of sensors. Hence, if we use a three-channel setup, we are able to retrieve at most three output components. In the extreme case when only one sensor is available, classical ICA is unable to perform any linear transformation on this signal and hence is unable to recover any of the underlying sources. This feature of ICA stems from linear algebra. ICA is simply a linear filter that transforms the dataset acquired by the matrix of sensors to another orthogonal base using rotating and scaling operations. Although it can do a lot with the input data, it is not capable of adding new dimensions to the matrix of sensors. This means that the only way to obtain more signal components after decomposing the observed data set using ICA is to increase the dimension of the input matrix.

Since we want to maintain the low-sensor setup, we cannot solve this problem simply by adding new sensors. Instead, we need to find a way to create new input signals artificially, using the existing matrix of sensors. In other words, to apply ICA to a sparse channel setup, it is first necessary to form a multi-channel representation of a low-channel recording.

2.1. Filtering procedure

To form a multi-channel data representation, we propose to use a set of zero-phase moving average filters on the matrix of data recorded from the sensors. A classical moving average filter operates by averaging $N$ points from the input signal $x$ to produce each point in the output signal $y$:

$$y[n] = \frac{1}{N} \sum_{i=0}^{N-1} x[n-i].$$ \hspace{1cm} (3)

To design the filter, only the value of the parameter $N$, which denotes the size of the averaging window, needs to be established. The filtering process, as defined in (3), introduces a phase shift into the signal $y$. To reverse this effect and to obtain the zero-phased filtered signal $z$, the filtering process needs to be performed once again on the $y$ signal in the reverse direction. The simplest approach to achieve this is to (i) reverse the order of the data in $y$, obtaining $y_{\text{rev}}$; (ii) apply (3) to $y_{\text{rev}}$; and (iii) reverse the order of the data in $y_{\text{rev}}$, obtaining $z$.

The filtering procedure performed by the MAICA algorithm can be described as follows. We assume that $x$ is a random column vector containing observations recorded from $ch$ channels, $x = [x_1, x_2, \ldots, x_{ch}]$. To create additional input dimensions, each $x_i$ ($i = 1, 2, \ldots, ch$) is filtered $k$ times with a zero-phase moving average filter. The size of the averaging...
window \( N \) is calculated separately for each filter, using an additional procedure described in section 2.2, and is stored in a vector \( L = [N_1, N_2, \ldots, N_k] \). The output from the filtering operation (vector \( g \)) contains both the \( ch \) original signals from vector \( x \) and \( k \) filtered versions of each signal. The pseudocode, given in figure 1, describes the filtering operation in detail.

Vector \( g \), obtained after the filtering procedure, is our multi-dimensional representation of a low-channel recording. This vector can be later directly introduced as an input vector to one of the classical ICA algorithms.

2.2. Computing filter parameters

Before the procedure described in section 2.1 can be applied, first two parameters need to be defined, the number of necessary filters \( k \) and the vector \( L \). Both parameters strongly depend on the analysed system and should be defined according to its characteristics. In this subsection, we therefore first set out general rules that should be followed when deciding on both parameters; and next, provide an automatic algorithm that can be used to compute their values.

Let us start the discussion with the second parameter. The best option, which limits the loss of information induced by the filtering process, is to use the smallest possible averaging window (\( N = 2 \)). Using a moving average filter with a window of this size ensures that each of the artificially created components from the vector \( g \) has almost the same frequency spectrum as one of the signals from the vector \( x \). The only part of the spectrum that is lost in this operation is composed of the highest frequency bins. Moreover, due to the use of small window size, the time waveforms also change only slightly, preserving much of the original information from the time domain.

The vector \( L \) that contains the smallest possible \( N \) values for the successive filters is the best choice from the point of view of preserving information. However, this choice is not always possible. To solve the ICA model, the \( A \) matrix must be invertible, and hence must be a full rank matrix. If two of the ICA input signals (two components from the vector \( g \)) are too highly correlated, this condition is violated, since the two columns of matrix \( A \) are linearly dependent. This means that in a real scenario, we need to choose the \( L \) vector in such a way to ensure that the components of the vector \( g \) will have almost the same frequency spectrum as their unfiltered ancestors, but at the same time will not be entirely correlated with them or with the other components of the vector \( g \).

Let us now move to the first parameter, the number of components introduced to the vector \( g \). Since the general rule states that the higher the dimension of the input vector, the more components can be returned by the ICA procedure, then to obtain more components at the ICA output we need more components in vector \( g \). Of course, the number of components in vector \( g \) cannot be arbitrarily large because the components must be as similar as possible to the original signals from vector \( x \) (without violating the rule of matrix \( A \) being full rank). This means that we should stop the process of adding new components to vector \( g \) when the filtered signals start to drift too far away from the original signals in terms of their similarity.

In summary, two conditions should be taken into consideration when defining the number of necessary filters (\( k \)) and the components of vector \( L \):

1. The similarity between each new input vector created in the filtering process and the corresponding original signal from vector \( x \) should be as high as possible.
2. The similarity between each pair of input components from vector \( g \) cannot be equal to 100%.

Both conditions can be addressed with Pearson’s correlation coefficient, which is defined as [36]:

\[
\rho_{ab} = \frac{\text{cov}(a, b)}{\sigma_a \sigma_b} \tag{4}
\]

where \( a, b \) are the two signals; \( \text{cov}(a, b) \) is the covariance between \( a \) and \( b \), and \( \sigma_a, \sigma_b \) are the standard deviations of \( a \) and \( b \), respectively. Due to the ambiguity in regard to the sign of the output components in ICA procedure [37], in the proposed algorithm we are interested only in the strength of the correlation, and hence the absolute value of the \( \rho \) coefficient is used.

The pseudocode shown in figure 2 illustrates the algorithm used to choose the parameters of the filtering process, which incorporates both of the rules described above. The output of this algorithm is a vector \( L \) composed of the window sizes of \( k \) moving average filters that should be applied when filtering data from the system under consideration. In our approach, we look for the vector \( L \) using only the data stored in the \( x_i \) channel and then apply the same vector to the other channels. In the case of more than one sensor, we could theoretically define the parameters of the filtering process with respect to each of the signals from vector \( x \); in practice, however, this would not be a correct choice, because it would favour some input signals over others. By using the same set of filters for all signals from vector \( x \), we ensure that the spectral and temporal information contained in each of the original signals is enhanced to the same degree.

The algorithm has two separate stop conditions. The first corresponds to the first similarity rule mentioned above: the similarity between each new input vector created in the filtering process and the corresponding original signal from the vector \( x \) should be as high as possible. The second verifies the correlation of the output components returned by ICA. According to this condition, the procedure of choosing the components of vector \( L \) ends when any of the two output
ICA components are highly correlated. The high correlation of the output ICA components (which should be independent of each other) usually means that ICA is not able to recover any further components, and has instead started to recover their mixtures. In other words, a high correlation of two output components indicates that the number of filters is too high and that the procedure of adding new input signals should be stopped.

The first three lines of the pseudocode in figure 2 introduce three correlation parameters: MinOriginal, MaxOriginal, and MaxOutputs. Each of these parameters represents one of the concepts mentioned before: MinOriginal determines the minimal acceptable correlation between the original signal $x_1$ and its filtered versions; MaxOriginal corresponds to the requirement of matrix $A$ to be a full rank matrix; MaxOutputs defines the maximal correlation between output components that should ensure their independence. The proposed values of these parameters (MinOriginal: 0.6, MaxOriginal: 0.98, and MaxOutputs: 0.4) were derived directly from the general rules described earlier and were then tested in artificial and real settings, giving good separation results.

It should be noted here that while the MaxOriginal parameter should not be changed at all (it decides whether the matrix $A$ is invertible), the two other parameters (MinOriginal and MaxOutputs) have quite a big error margin. In [33] was shown that the main effect of changes in both parameters is the changing number of components returned by ICA. That means that we can use them to control the number of output components. If we expect more sources, we can lower MinOriginal and/or increase MaxOutputs. On the other hand, if we want to obtain fewer output components, we can increase MinOriginal and lower MaxOutputs. Of course, usually, we do not know the exact number of sources influencing the system at hand. In such a case it is better to relax both parameters to allow for more filters. As we showed in [33], via the sinusoid system that will be discussed in section 3.1, if there are more output components than sources, most of the sources will be still recovered with a high accuracy but some of them might be doubled (with a changed phase) or might be mixtures of the other components. The reverse case, with more restrictive parameters, is much more difficult for ICA to resolve since there are not enough components to recover all the original sources. In such case, the strongest sources will also be recovered but with a smaller accuracy.

An additional remark about the parameters of the filtering process should be made here. Since these parameters depend on the internal complexity of the observed system, the procedure of finding these parameters needs to be performed only once per task at hand. In general, the approximate number of sources influencing the recorded signals is more or less fixed over time. Of course, certain aspects of neural or artefactual activity may appear or vanish over time. However, even if the true number of sources $t$ differs from $s$ in some trials, this will only cause that the weakest sources will be omitted ($t > s$) or some mixtures of the strongest sources will be returned as additional ICA components ($t < s$).

Hence, the pseudocode in figure 2 needs to be run only once for a given system. Later, when new data are recorded from the system, it is sufficient to filter them (using the pseudocode in figure 1) and to pass the new vector $g$ to one of the ICA algorithms. In our implementation, we use the FastICA algorithm [8], but other algorithms may be applied as well. Of course, if only one segment of the data is recorded from the system, it is sufficient to use the algorithm in figure 2, which returns not only the vector $L$ but also the set of output components $b$ obtained for the first segment of data.

3. Methods

To demonstrate the potential of the MAICA method, we applied it to the following systems:

- **System 1**, an artificial system composed of 10 sinusoid components (single-trial data).
- **System 2**, an EEG signal recorded from 64 channels (data from one continuous two-minute session composed of alternating motor tasks).
- **System 3**, an EEG signal recorded from two channels; data from MI-BCI sessions (multi-trial data).

3.1. System 1—sinusoidal components

To carry out a preliminary verification of the MAICA method described in section 2, we used an artificial system composed of ten sinusoid waveforms $y_1...y_{10}$ of the same length (1000 samples) but different frequencies (table 1). The sampling frequency used in the system was equal to 1000 Hz.

The sinusoids from table 1 were combined to form two linear compositions. The first of these contained 100% of the first five...
I Rejer and P Górski

6

sinusoids and 10% of the last five sinusoids, while the second contained 10% of the first five sinusoids and 100% of the last five (5). To compare the MAICA with the state-of-the-art methods, the mixtures $M_1$ and $M_2$ were submitted to five algorithms—DE, WICA, EMD + ICA, EEMD-ICA, and MAICA.

$$M_1 = 100\% (y_1 + y_2 + y_3 + y_4 + y_5) + 10\% (y_6 + y_7 + y_8 + y_9 + y_{10})$$

$$M_2 = 10\% (y_1 + y_2 + y_3 + y_4 + y_5) + 100\% (y_6 + y_7 + y_8 + y_9 + y_{10})$$

(5)

3.2. System 2—PhysioBank EEG dataset

The second stage of our research aimed to verify whether the MAICA algorithm applied to the data recorded from a low–channel setup would return the same artefact components as classical ICA applied to the data recorded from a multi-channel setup. To achieve this, we used one of the benchmark files from the EEG Motor Movement/Imagery Dataset [38] stored in PhysioBank Databases [39] (data set: s001r03.edf). The file contained EEG signals recorded during one continuous two–minute session composed of alternating motor tasks performed by a subject. The EEG data were recorded with 64 electrodes located according to the International 10–10 system (sampling rate—160 Hz).

We began the analysis by performing classical FastICA on the entire 64-channel data set. We then visually inspected the set of 64 output components returned by FastICA and identified the most obvious artefact-components. These components were later used as ground truth for the components returned by MAICA. Next, we ran MAICA on the data set composed of signals recorded from five channels distributed over the whole scalp (Fz, Cz, Oz, T7, T8). While the two algorithm parameters were set to their default levels ($\text{MinOriginal} = 0.4$, and $\text{MaxOriginal} = 0.98$), the parameter $\text{MaxOutputs}$ was relaxed (from 0.6 to 0.4) to allow for more output components and to compensate the length of the EEG signals (19200 samples).

Table 1. Source sinusoidal waveforms.

| $y_1 = \sin(320\pi)$ | $y_2 = \sin(200\pi)$ | $y_3 = \sin(9\pi)$ | $y_4 = \sin(100\pi)$ | $y_5 = \sin(400\pi)$ |
|----------------------|----------------------|---------------------|----------------------|----------------------|
| $y_6 = \sin(2\pi)$   | $y_7 = \sin(120\pi)$ | $y_8 = \sin(300\pi)$| $y_9 = \sin(250\pi)$ | $y_{10} = \sin(60\pi)$|

Figure 3. Source sinusoidal signals.

Figure 4. Linear combinations of source sinusoids from figure 3, mixed in the proportions defined in (5).

Figure 5. Set of components returned by MAICA.
Finally, we compared the components returned by MAICA with the components found for the whole 64-channel set. The analysis scheme presented above was used to show visually the similarity of the results obtained with MAICA applied for a low-channel setup and those obtained with classic ICA applied for a rich-channel setup. To prove that this similarity was not just a ‘lucky trial’ effect and to show that MAICA consistently returns similar components as 64-channel ICA, we run the four algorithms discussed in Introduction (DE, EMD, EEMD-ICA, and WICA) together with classic ICA and MAICA on 109 data sets stored in EEG Motor Movement/Imagery Dataset. Each data set was recorded from a different subject performing the same task (opening and closing left or right fist). For each set, the classic ICA was run over the whole matrix of channels, and the five other preprocessing algorithms were run over only five channels: Fz, Cz, Oz, T7, and T8. The components returned by each preprocessing algorithm for each data set were correlated with the components returned by classic ICA. For the comparison purpose the 109 sets of five correlation coefficients of the highest value were chosen for each algorithm.

### Table 2. Correlation between MAICA components and source sinusoids.

| Source signal | $y_1$ | $y_2$ | $y_3$ | $y_4$ | $y_5$ | $y_6$ | $y_7$ | $y_8$ | $y_9$ | $y_{10}$ |
|---------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Component IC5 | IC9   | IC2   | IC8   | IC10  | IC1   | IC6   | IC7   | IC4   | IC3   |        |
| Similarity index | 0.99  | 0.99  | 0.99  | 0.99  | 1     | 1     | 1     | 1     | 1     | 0.99   |

### Table 3. Correlation between DE components and source sinusoids ($s = 30, d = 4$).

| Source signal | $y_1$ | $y_2$ | $y_3$ | $y_4$ | $y_5$ | $y_6$ | $y_7$ | $y_8$ | $y_9$ | $y_{10}$ |
|---------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Component IC8 | IC5   | IC7   | IC1   | IC5   | IC10  | IC9   | IC4   | IC9   | IC6   |        |
| Similarity index | 0.65  | 0.63  | 0.97  | 0.99  | 0.62  | 0.78  | 0.48  | 0.9   | 0.72  | 0.78   |

3.3. System 3—motor imagery classification

The last stage of our research aimed to determine whether MAICA could be used for real-time EEG classification. To achieve this, we used EEG data sets recorded at our laboratory during the MI-BCI experiment. Five subjects (three male, two female, mean age 26.4, range 22–32 years) participated in the experiment. All of them were right-handed and all had normal vision. None of the subjects had previous experiences with MI-BCI and none reported any mental disorders. The experiment was conducted according to the Helsinki declaration on proper treatments of human subjects. Before the experiment onset, written consent was obtained from all subjects.

The details of the experiment were as follows. The subject was placed in a comfortable chair and EEG electrodes were applied to his or her head. A brief sound announced the start of the experiment. The main body of the experiment was divided into 200 trials. During each trial, an arrow pointing to the left or the right was presented to the subject. The subject’s task was to imagine a wrist rotation with the hand corresponding to the arrow direction. The directions for the succeeding trials were chosen randomly. The length of each trial was fixed at 10s. The experiment was run in a trial-by-trial mode which means that there were no breaks between trials. The end of one trial was also the start of the next trial. The experiment was divided into four sessions of 50 trials each. There were five-minute breaks between sessions.

EEG data were recorded from two monopolar channels at a sampling frequency of 256 Hz. The electrodes were attached to the subject’s scalp at the C3 and C4 positions. The reference and ground electrodes were located at Fpz and the right mastoid, respectively. The EEG signal was acquired with a Discovery 20 amplifier (BrainMaster) and preliminarily...
filtered in the 1–30 Hz band with a Butterworth filter of the 4th order. The four central seconds were extracted from each trial (seconds: 4–7) and passed to further analysis.

To determine whether MAICA can enhance the classification accuracy, two datasets were prepared (using EEG data from sessions 1–3). The first set was composed of two-channel EEG data, while the second contained channels reconstructed from components returned by MAICA run separately on each trial. Obviously, we did not reconstruct the original channels from all components (that would lead to the same channels as in the beginning) but we first detected and removed the ‘worst’ component pairs from each trial. To decide which components should be discarded, MAICA was performed separately on both channels. Next, the components obtained for C3 channel were correlated with the components obtained for C4 channel and those pairs of components that fulfilled the following conditions were discarded:

- the correlation of components in the pair exceeded 90%;
- both components from the pair had a similar impact on both channels (the corresponding weights in the mixing matrices $A$ differed by a maximum of 10%).

Two factors motivated this approach. Firstly, a high correlation between the two components extracted from different channels is likely to mean that they represent the same source. Secondly, if this source is more or less equally represented in both C3 and C4 channels, it does not correspond to the current brain activity related to the hand motor imagery task.

Both datasets were processed using the standard BCI processing pipeline: preprocessing, feature extraction, and classification. Each stage of the processing pipeline was performed with the same algorithms and parameters for both sets. Since the 10-fold cross-validation scheme was used in the classification process, the preprocessing and feature extraction steps were performed in a 10-iteration loop, separately for each of the ten training sets. In the preprocessing step, the CSP algorithm was used to find a pair of spatial filters and reorganise the data. Next, the band power features were extracted from both patterns. Four frequency bands were used in this task: a classic alpha band (8–13 Hz) and three beta sub-bands: 13–18 Hz, 18–24 Hz, and 24–30 Hz. The 135 trials (from sessions 1–3), each described by eight power features and a corresponding class label, were submitted to the SVM classifier of a linear kernel and soft-margin parameter $c$ equal to 1. After the training, the performance of the classifier was evaluated on the basis of the remaining 15 trials. The whole procedure was repeated ten times for each training subset. The final performance metric for the classifier was the accuracy averaged over ten validation sets.

The two classifiers were built and tested during the break after the third session. In the last session, both classifiers were used in an online mode to validate their performance against entirely fresh data. The output from the classifiers was delivered as a feedback for the user in the form of a small blue arrow displayed just above the leading arrow. Although there were two outputs in each trial (first—from the classifier trained over the original EEG data, second—from the classifier trained over the channels reconstructed from MAICA components), to facilitate a fair comparison between methods, we randomly chose and presented to the user only one of them.

When the experiment was finished, we performed an offline analysis on the data recorded in cross-validation part of the experiment (150 trials from the first three sessions). The aim of this analysis was to compare MAICA with the four other preprocessing algorithms discussed previously: DE, WICA, EMD + ICA and EEMD-ICA. The experimental scheme for each algorithm was exactly the same as for MAICA and classic ICA.

4. Results

4.1. System 1—sinusoidal components

Figure 3 shows the original set of ten sinusoids used to verify the MAICA algorithm. After combining the sinusoids in the
shown in table 3. As can be seen from the table, even for the full separation. The best separation, obtained for \( s = 30 \), is the delay, given in numbers of samples), we did not obtain \( s = 26 \) different settings for \( s = 2:2:50 \) is the length of the number of components as sources, and hence we set \( n = 1 \) and \( M = 2 \) mixtures. Initially, we tried to retrieve the same method (DE), we constructed a matrix composed of delayed source sinusoid) with the highest similarity index. presents the ten combinations (one combination per each tion of MAICA components and original sinusoids. Table 2 tion coefficient given in (4) was computed for each combina-

| Source signal | \( y_1 \) | \( y_2 \) | \( y_3 \) | \( y_4 \) | \( y_5 \) | \( y_6 \) | \( y_7 \) | \( y_8 \) | \( y_9 \) | \( y_{10} \) |
|--------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| Component    | IC9      | IC6      | IC10     | IC8      | IC9      | IC4      | IC1      | IC5      | IC3      | IC4      |
| Similarity index | 0.69    | 0.97     | 1        | 0.99     | 0.66     | 0.75     | 0.62     | 0.96     | 0.83     | 0.52     |

Figure 8. Set of components returned by EMD + ICA. proportions defined in (5), two linear combinations of signals were obtained, as shown in figure 4. The task of MAICA was to take these two combinations as inputs and return the output components, which were then compared with the true sources, it is with the original sinusoids from figure 3. The components returned by MAICA are shown in figure 5. As can be seen, the algorithm not only recovered the correct number of sources but also returned components that were very similar to the source sinusoids. To calculate the similarity index, the correlation coefficient given in (4) was computed for each combination of MAICA components and original sinusoids. Table 2 presents the ten combinations (one combination per each source sinusoid) with the highest similarity index.

To compare MAICA with the dynamical embedding method (DE), we constructed a matrix composed of delayed \( M1 \) and \( M2 \) mixtures. Initially, we tried to retrieve the same number of components as sources, and hence we set \( d \) to four (where \( d \) is the number of delayed signals). Although we tried 26 different settings for \( s \) (where \( s = 2:2:50 \) is the length of the delay, given in numbers of samples), we did not obtain full separation. The best separation, obtained for \( s = 30 \), is shown in table 3. As can be seen from the table, even for the ‘best’ case only three sources were recovered with a correlation equal of higher than 90%.

Due to unsatisfactory results obtained with \( d = 4 \), we tried to increase the number of output components by changing the parameter \( d (d = 2:2:50) \) with \( s = 2 \). In this way, we managed to retrieve five of the sources with a correlation higher than 90% (figure 6, table 4) but with a significantly larger number of delays (\( d = 42 \)). With these parameters, the ICA algorithm had to be performed on 86 input signals (instead of ten), which significantly increased the computation time.

Next, we introduced the \( M1 \) and \( M2 \) mixtures to WICA algorithm. We tried to recover the original sources with different: wavelets families (Daubechies, Coiflets, Symlets, and Biorthogonal), wavelets parameters, and decomposition levels (from 3 to 6). The best decomposition was obtained for 4-level decomposition with Symlet17 wavelet. The set of components returned by FastICA for the input matrix composed of wavelet components is presented in figure 7 and their correlation with ten original sinusoids—in table 5. As it can be noticed in the table only four out of ten original sources were retrieved after WICA application. Symlet17 is a filter of a quite high order. However, when we tried to use filters of lower orders, we were able to retrieve only two–three original sources. The filters of higher orders also did not allow to recover more than four sources.

Finally, we run the two EMD algorithms, the classic version and the modified EEMD-ICA version, on the \( M1 \) and \( M2 \) mixtures. The set of output components obtained for the input matrix composed of IMFs returned by a classic EMD algorithm is presented in figure 8 and their correlation with ten original sinusoids—in table 6. For the EEMD-ICA algorithm we tried different levels of noise parameter \( (np = 0.1:0.1:1) \) and different numbers of trials \( (n = 10:10:200) \). The best separation (shown in figure 9 and table 7) was obtained for \( np = 1 \) and \( n = 100 \).

The last table in this section (table 8) presents the summary of the results obtained with all five tested methods. The succeeding columns of this table contain: the method name, the set of parameters providing the best separation results, the number of sources retrieved with the correlation equal or higher than 90%, and the computation time needed to perform the algorithm. All the computations were performed on a machine with the following parameters: Intel CORE i7-6700HQ 2.60 GHz CPU, 16 GB RAM, Windows 10, ×64.

4.2. System 2—PhysioBank EEG dataset

During this part of our research, the components returned by FastICA run over EEG data from 64 channels (from the dataset s001r03.edf) were compared to the components returned by MAICA run over EEG data recorded from five channels (Fz, Cz, Oz, T7, T8) from the same dataset. The EEG signals
To make the figures in this section more readable, only a 15 s recorded from the chosen channels are shown in figure 10.

To extend the number of channels submitted to FastICA, MAICA was run over the five chosen channels. The algorithm returned the vector \( L \), containing window sizes for 11 filters, \( L = [3, 7, 15, 28, 49, 79, 118, 171, 229, 289, 355] \). The filters of the averaging windows defined in \( L \) were then used to filter the data from each of the five channels. The new dataset was composed of five original EEG signals and their 55 filtered versions. The set was submitted to the FactICA algorithm, which returned 60 output components.

As a preliminary confirmation of the correctness of the components obtained after applying MAICA, the output components were paired with the original EEG signals. The pairing process was carried out in two steps. In the first step, the correlation between each output component and the original channel was calculated using (4), and the correlation matrix was built. In the second step, each output component was paired with one channel according to the maximum rule. Figure 11 shows six of these pairs, four with high correlation and two with low correlation. It should be emphasised that the high correlation can be observed not only for the channels used as MAICA input (Fpz, T8) but also for the channels that were not directly introduced to the MAICA procedure (Iz, AF8).

The two next figures in this section, figures 12 and 13, show five of the most characteristic output components returned by FastICA for the set of 64 original channels (figure 12) and for the set of 60 components created with MAICA (figure 13). As can be seen, both sets contain components that can be attributed to well-known artefacts. For example, \( IC_{17} \_fastica \) and \( IC_{31} \_MAICA \) appear to have captured the power-line noise; \( IC_{4} \_fastica \), \( IC_{24} \_MAICA \), and \( IC_{47} \_MAICA \) strongly resemble EOG artefacts (clearly visible in channel Fpz in figure 10); \( IC_{11} \_fastica \) and \( IC_{19} \_MAICA \) captured a rhythmic artefact, which is likely to be an ECG artefact; and \( IC_{8} \_MAICA \) looks like a classic EMG artefact. By comparing both figures, it can be seen that although the mentioned artefacts are visible in both sets of components, they are much more apparent in the set returned by FastICA after applying MAICA.

The results of the comparison analysis performed on 109 data sets stored in EEG Motor Movement/Imagery Dataset are presented in table 9. To create each single data point shown in the first row of the table, the highest correlation coefficients (presenting similarity of the results returned by the classic ICA and one of the other algorithms) were averaged over 109 data sets. The data in the next rows of table 9 were created in the same fashion but for the second, third, fourth, and fifth highest correlation coefficients. The methods’ parameters were set to the levels ensuring the similar number of components (about 50 components):

| Component | \( y_1 \) | \( y_2 \) | \( y_3 \) | \( y_4 \) | \( y_5 \) | \( y_6 \) | \( y_7 \) | \( y_8 \) | \( y_9 \) | \( y_{10} \) |
|-----------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| Similarity | IC9 | IC9 | IC1 | IC8 | IC9 | IC3 | IC6 | IC7 | IC7 | IC10 |
| Index     | 0.62 | 0.36 | 0.90 | 0.82 | 0.63 | 0.97 | 0.97 | 0.70 | 0.70 | 0.98 |

Figure 9. Set of components returned by EEMD-ICA for 100 trials with \( np \) parameter equal to 1.

1. MAICA: \( \text{MinOriginal} = 0.4, \text{MaxOriginal} = 0.98, \text{MaxOutputs} = 0.4; \)
2. DE: \( s = 5, d = 9; \)
3. EMD + ICA: 9 IMFs;
4. EEDE-ICA: \( np = 1, \text{trials} = 100; \)
5. WICA: wavelet: simlet17, 9 decomposition levels.

4.3. System 3—motor imagery classification

The two systems discussed so far involved only a single trial. Hence, both stages of the MAICA procedure (finding the \( L \) vector and filtering the data) could be performed at the same time. The situation was very different in the classification example, as it involved 200 trials per subject. Since all trials were recorded from the same system (meaning that the complexity of each trial was more or less the same), the procedure for finding the vector \( L \) was run only for the first trial. The output of this procedure is presented in table 10. To support our approach of ‘denoising’ EEG signal presented in section 3.3, the MAICA algorithm was run separately for both channels. In this way, two separate \( L \) vectors were
created for each subject: one for channel C3, and one for channel C4.

As can be seen in table 10, both, the number of filters and the window sizes of the successive filters, were highly similar for both channels, for almost all subjects. This high level of similarity means that the complexity of the analysed system did not change significantly across subjects. Only the signal recorded from channel C4 for Subject 3 allowed for more filters than the other signals. After the analysis of C4 signal, we found that the reason for the higher number of filters was an imprecise montage of C4 electrode that caused the appearance of a slow wave artefact (trend) in the signal. As a result, filters of wider filtering window returned signals of sufficiently high correlation with the original C4 signal and hence more filters were created.

After defining the filter parameters, two classifiers were trained for each subject, using the procedure described in section 3.3. Table 11 presents the classification accuracy of these two classifiers, the classifier without (C_classic) and with MAICA filtering (C_MAICA). The results of two sessions, the cross-validation session and the test session, are presented in the table. The two first columns under each session contain the classification accuracy for the C_classic classifier (first column) and the C_MAICA classifier (second column). The third column shows the number of trials that were corrected as a result of applying the proposed denoising approach. Finally, the last column (Accuracy difference [%]) presents the relative increase in the classification accuracy after applying MAICA. Additionally, the last row of the table contains the mean value of each column.

When analysing the classification accuracy with and without MAICA preprocessing, it can be seen that although the increase in accuracy was not significant (according to t-student test at $p = 0.05$) for the validation data, it was significant for the test data. For each subject, the increase was...
equal to at least 5%, achieving the highest value (25%) for Subject 2. This significant increase in the classification accuracy, obtained for data that were not known at any stage of the classifier training process, is a very important feature of the proposed approach. Although we did not anticipate this behaviour when beginning the experiment, it seems to be quite reasonable. By decomposing the signal from each new trial into an individual set of components, MAICA can separate artefact components that are different from those in the training set. If any of these components fulfill the conditions for removal, it is removed from the set of components and a cleaner signal is reconstructed.

As can be seen from table 11, only about 25% of the trials in both sets were corrected after the application of MAICA. This value was slightly higher for the test set (26%) than for the validation set (24%). It is difficult to detect any correlation between the number of corrected trials and classification accuracy. In general, in the case of validation data, the more trials were corrected, the higher the accuracy increase was obtained (although for Subject 1, no accuracy increase was observed despite 30 corrected trials). In the case of the test session, there was no correlation between the results.

The application of the MAICA algorithm (in conjunction with the proposed artefact reduction approach) extends the processing time. Table 12 shows the amount of time needed to perform the entire BCI processing pipeline for one trial, both with ($C_{\text{MAICA}}$ [ms]) and without ($C_{\text{classic}}$ [ms]) MAICA preprocessing. As can be seen from the last column of table 12, which shows the difference between the processing times with and without MAICA preprocessing, the extra time needed for MAICA calculations was very small (less than 6 ms) and was stable across subjects.

The last table in this section (table 13) presents the comparison of the classifiers trained over the components obtained with the six analysed preprocessing algorithms: classic ICA, MAICA, DE, WICA, EMD + ICA and EEMD-ICA. Because the length of the signal from each trial (1024 samples) was similar as in System 1 (1000 samples), the parameters of all methods (except the DE method) remained at the same levels (the parameters are presented in table 13). Addressing the DE method, we decreased the number of delays ($d$) from 42 to 8 to ensure similar processing time of all the methods.

5. Discussion

Each of the three systems presented in this paper illustrated a different advantage of the MAICA application. The first system, composed of a mixture of sinusoids, proved that the proposed method provides an additional space needed to extract more source components than input mixtures. As shown in section 4.1, when the default levels of MAICA parameters were used in the analysis, the decomposition was almost perfect. Not only were all ten source sinusoids retrieved (figure 5), but the correlation between the ICA output components and the initial sinusoids was also extremely high (99%–100%, table 2).

![Figure 12. The most characteristic output components returned by FastICA for the set of 64 original signals.](image1)

![Figure 13. The most characteristic output components returned by FastICA for the set of 60 components created with MAICA.](image2)

During the tests performed on the sinusoid system, we also compared the decomposition obtained after application of MAICA with the decomposition obtained with the DE, WICA, EMD + ICA, and EEMD-ICA methods. Although we tried many different sets of parameters for each method (apart from EMD + ICA), we were unable to retrieve all the source sinusoids with the same high correlation as that provided by MAICA (table 8). Among the four methods used for comparison, the best set of components was obtained with the DE method (for an input set composed of the two original mixtures of sinusoids and their 42 delays). When ICA was run on this rich set of input components, it was able to retrieve ten output components showing quite high correlation with the sources (75%–99%). It is likely that even the full decomposition could be obtained with the DE method if many more delays were used; however, due to a limited signal length (1000 samples), we had to limit the number of delays used during the tests.
Table 9. The correlation coefficients averaged over 109 sets (the table presents the five first ranks), standard deviations are reported in square brackets.

| Coefficient rank | MAICA [%] | DE [%] | WICA [%] | EMD + ICA [%] | EEMB-ICA [%] |
|------------------|-----------|--------|----------|---------------|--------------|
| 1                | 55.32 (1.7) | 49.33 (1.9) | 45.76 (2.2) | 44.04 (3.3) | 51.66 (1.4) |
| 2                | 54.74 (2.2) | 45.72 (2.5) | 44.21 (2.3) | 42.33 (3.7) | 50.20 (2.2) |
| 3                | 50.21 (3.1) | 45.05 (3.4) | 43.77 (2.7) | 41.67 (3.3) | 48.37 (1.8) |
| 4                | 48.09 (1.9) | 46.30 (2.8) | 43.53 (2.3) | 39.21 (4.1) | 47.83 (2.6) |
| 5                | 47.24 (2.2) | 42.89 (2.4) | 41.90 (2.6) | 37.06 (3.7) | 46.25 (2.3) |
| Mean             | 51.12 (3.7) | 45.86 (2.3) | 43.83 (1.4) | 40.86 (2.7) | 48.86 (2.1) |

Table 10. Vectors $L$ returned for each channel (C3, C4) and each subject.

| S | Channel | Size of the averaging window of successive filters |
|---|---------|-------------------------------------------------|
| 1 | C3      | 7 10 13 17 22 27 31                             |
|   | C4      | 7 10 14 18 23 28 32 36                         |
| 2 | C3      | 9 13 18 24 29 34 39                             |
|   | C4      | 9 13 18 23 28 33 38                             |
| 3 | C3      | 10 16 22 27 32 36 40 44                        |
|   | C4      | 14 19 25 30 35 39 43 47 51 55 59 63 68 73 78 83 |
| 4 | C3      | 10 13 17 21 26 30                             |
|   | C4      | 10 13 17 22 27 32 36                           |
| 5 | C3      | 11 14 17 17 21 26 30                          |
|   | C4      | 12 15 19 24 29 33 37                           |

Table 11. Classification accuracy of the classifier without (C_CLASSIC) and with MAICA filtering (C_MAICA) in the cross-validation and test session for each subject.

| S  | C_CLASSIC [%] | C_MAICA [%] | No. of corr. trials | Acc. increase [%] | C_CLASSIC [%] | C_MAICA [%] | No. of corr. trials | Acc. increase [%] |
|----|---------------|--------------|---------------------|-------------------|---------------|--------------|---------------------|-------------------|
| 1  | 76.00         | 76.00        | 30                  | 0.0               | 70.00         | 74.00        | 13                  | 5.7               |
| 2  | 78.00         | 80.70        | 36                  | 3.5               | 64.00         | 80.00        | 13                  | 25.0              |
| 3  | 83.30         | 84.70        | 28                  | 1.7               | 80.00         | 84.00        | 9                   | 5.0               |
| 4  | 66.00         | 68.00        | 56                  | 3.0               | 65.80         | 70.00        | 13                  | 6.4               |
| 5  | 74.70         | 76.00        | 34                  | 1.7               | 82.00         | 86.00        | 19                  | 4.9               |
| M  | 75.60         | 77.10        | 36.8                | 2.0               | 72.40         | 78.80        | 13.4                | 9.4               |

Table 12. Processing time needed for one trial classification, with and without application of the MAICA algorithm.

| Subject | C_CLASSIC [ms] | C_MAICA [ms] | Difference [ms] |
|---------|----------------|--------------|-----------------|
| 1       | 14.869 220     | 20.688 880   | 5.819 660       |
| 2       | 13.486 405     | 19.220 645   | 5.734 240       |
| 3       | 30.542 290     | 36.273 030   | 5.730 740       |
| 4       | 14.818 445     | 20.547 605   | 5.729 160       |
| 5       | 18.101 060     | 23.835 120   | 5.734 060       |
| Mean    | 18.363 484     | 24.113 056   | 5.749 572       |
In the introduction, we underlined that the main benefit of MAICA is that it enhances the sources present in the input ICA matrix by repeating the original signal with slight modifications. The same feature is shared by the DE method. Also here the original (slightly shorter) signal is introduced many times to the ICA input matrix.

Hence, the problem with the DE method does not set with the decomposition accuracy but rather with the processing time. When this factor is compared for both methods, MAICA appears to be much better. Although the amount of time needed to generate an additional artificial input component (by applying a moving average filter or delaying the mixed signal) is negligible, the time needed by the ICA algorithm to find the mixing matrix increases significantly with each additional input. When we compared the computation time of both methods, the DE algorithm was 35 times slower than MAICA. Of course, this is not a problem during offline processing but may affect the online analysis.

Using the second system, we aimed to show that the components returned by ICA after applying the set of zero-phase moving average filters are correct not only for an artificially created mathematical system but also for a real multi-dimensional dataset. As we have shown in section 4.2 MAICA used over only five channels recognised artefacts similar to those of classical ICA used over 64 channels (figures 12 and 13). Moreover, the artefact components returned by MAICA were more straightforward than those returned by the classical ICA. The comparison of MAICA and other four preprocessing algorithms over 109 EEG data sets stored in EEG Motor Movement/Imagery Dataset (table 9) showed that all five methods provided quite stable results across subjects. However, the degree of correlation between the components returned by classical ICA and the five other algorithms differed slightly across algorithms. The highest averaged correlation coefficients were obtained for MAICA and EEDM-ICA, the smallest for WICA and EMD + ICA.

Finally, the last system (motor imagery classification) was used to show that MAICA is capable of working in an online mode. We managed to achieve this goal since the delay incurred by preparing the two sets of filtered signals (for C3 and C4), applying ICA to each of the two sets, and discarding probable artefacts was smaller than 6ms for each trial (table 12). This small increase in the processing time did not introduce any noticeable delays to the interface classification process.

Table 13 compares the classification results obtained after preprocessing of the raw EEG signals with the six algorithms: classic ICA, MAICA, DE, WICA, EMD + ICA, and EEMD-ICA. As it can be noticed in the table, only MAICA was able to improve classification results obtained with classic ICA for all five subjects. The two other algorithms WICA and EEMD-ICA performed better only for two (EEMD-ICA) or one (WICA) subjects. The classification accuracy provided by the two remaining algorithms (DE and EMD + ICA) was smaller than that obtained with classic ICA for all subjects. Comparing MAICA and EEMD-ICA it should be mentioned that although the mean accuracy obtained after applying MAICA was about 2% higher than after applying EEMD-ICA, the EEMD-ICA algorithm occurred to be slightly better than MAICA for one subject (S2).

Generally, MAICA can be used on all types of signals that could be introduced to the ICA itself. In other words, if we can apply ICA to the signal matrix, we can also use MAICA to extend this matrix to facilitate the source extraction. However, apart from the question about the ‘possibility’, there is also a question about the ‘reasonability’ of such a solution. There are two cases when the application of MAICA is not reasonable and both depends on the size of the input matrix:

(1) input matrix is composed of very short signals—probably no filters will be created and hence no gain from using MAICA will be obtained;
(2) input matrix is composed of tens of signals—since ICA works well with high-dimensional data, the MAICA application probably will only increase the complexity of the calculations.

The one possible limitation of MAICA is that it needs some insight from its user. It has two parameters (MinOriginal and MaxOutputs) that have to be set correctly to obtain the optimal separation of sources. Although it is not difficult to provide the right parameters using the general rules that we discussed in the paper, still it is not a pure out-of-the-shelf solution.

### 6. Conclusion

This paper demonstrates that the MAICA algorithm enables the decomposition of a low-dimensional signal into a high-dimensional set of components. The three systems presented in this paper show that there is a high probability that at least some of the components in this set are equivalent to the true sources of the signal recorded from the sensors. In the sinusoid example, the ten output components were almost identical to the sinusoid functions used in the two input mixtures.
Moreover, in the second survey, in which only five of the 64 recorded EEG signals were introduced to the MAICA algorithm, the detected artefacts were not only similar to those returned by classical ICA applied over 64 channels but were also much more readable. The system used in the second survey directly supports our thesis that MAICA can decompose a low-channel EEG signal into components in the same way as ICA does for a multi-channel signal. The last stage of the experiment revealed three more benefits of the proposed method: a short processing time, the stability of the filter parameters, and the ability to work correctly with the entirely new data. All three features are very important since they allow the use of this algorithm for the online classification of brain activity. Moreover, due to the stability of the filter parameters, they need to be calculated once before the beginning of the online classification, and can then be applied in subsequent online trials, without affecting the classification accuracy even in the test trials.

Acknowledgments

This work was supported by the National Science Centre, Poland, Grant No. 2017/27/B/HS4/01216.

ORCID iDs

Izabela Rejer

https://orcid.org/0000-0002-9003-4129

References

[1] Hankins T C and Wilson G F 1998 A comparison of heart rate, eye activity, EEG and subjective measures of pilot mental workload during flight Aviat. Space Environ. Med. 69 360–7 (https://www.ncbi.nlm.nih.gov/pubmed/9561283)
[2] Liang S F, Lin C T, Wu R C, Chen Y C, Huang T Y and Jung T P 2006 Monitoring driver’s alertness based on the driving performance estimation and the EEG power spectrum analysis 27th Annual Conf. 2005 IEEE Eng. Med. Biol. (Shanghai: IEEE) pp 5738–41
[3] Chynal P, Sobecki J, Rymarz M and Kiliński A B 2016 Shopping behaviour analysis using eye tracking and EEG 2016 9th Int. Conf. on Human System Interactions pp 458–64
[4] Nakanishi M, Wang Y, Hsu S H, Wang Y T and Jung T P 2017 Independent component analysis-based spatial filtering improves template-based SSVEP detection 39th Annual Int. Conf. IEEE Eng. Med. Biol. Soc. pp 3620–3
[5] Friman O, Volosyak I and Gräser A 2007 Multiple channel detection of steady-state visual evoked potentials for brain–computer interfaces IEEE Trans. Biomed. Eng. 54 742–50
[6] Aviyente S 2006 Analysis of event related potentials using PCA and matching pursuit on the time-frequency plane IEEE Conf. Eng. Med. Biol. Soc. pp 2454–7
[7] Luo G, Chen G, Tian L, Qin L and Qian S E 2016 Minimum noise fraction versus principal component analysis as a preprocessing step for hyperspectral imagery denoising Can. J. Remote Sens. 42 106–16
[8] Eshoaeifar S, Took C C, Park C and Mandic D C 2017 Quaternion common spatial patterns IEEE Trans. Neural Syst. Rehabil. Eng. 25 1278–86
[9] Kim Y, Ryu J, Kim K K, Took C C, Mandic D P and Park C 2016 Motor imagery classification using mu and beta rhythms of EEG with strong uncorrelating transform based complex common spatial patterns Comput. Intell. Neurosci. 2016 1480962
[10] Hyvärinen A and Oja E 2000 Independent component analysis: algorithms and applications Neuronal Netw. 13 411–30
[11] Oja E and Yuan Z 2006 The FastICA algorithm revisited: convergence analysis IEEE Trans. Neural. Netw. 17 1370–81
[12] Srinivasulu A and Reddy M S 2012 Artifacts removing from EEG signals by ICA algorithms IOSR J. Electron. Electron. Eng. 2 11–6
[13] Kim C S, Sun J, Liu D, Wang Q and Paek S G 2017 Removal of ocular artifacts using ICA and adaptive filter for motor imagery-based BCI IEEE/CCAA J. Automat. Sin. pp 1–8
[14] Augustyniak P 2017 Localization of noise sources in a multilead electrophysiological record Polish Conf. on Biocybernnetics and Biomed. Eng. (Berlin: Springer) pp 56–67
[15] Xue Z, Li J, Li S and Wan B 2006 Using ICA to remove eye blink and power line artifacts in EEG 1st Int. Conf. on Innovative Computing, Information and Control Volume I (ICICIC’06) vol 3 (Beijing: IEEE) pp 107–10
[16] Jung T P, Humphries C, Lee T W, Makeig S, McKeown M J, Iragui V and Seijinoski T J 1998 Extended ICA removes artifacts from electroencephalographic recordings Advances in Neural Information Processing Systems NIPS: Proc. 1997 (Cambridge, MA: MIT Press) pp 894–900 (https://papers. nips.cc/paper/1343-extended-ica-removes-artifacts-from-electroencephalographic-recordings.pdf)
[17] Delorme A, Palmer J, Onton J, Oostenveld R and Makeig S 2012 Independent EEG sources are dipolar PLoS One 7 e30135
[18] Woon W L and Lowe D 2004 Can we learn anything from single-channel unaveraged MEG data? Neural Comput. Appl. 13 360–8
[19] James C J and Gibson O J 2003 Temporally constrained ICA: an application to artifact rejection in electromagnetic brain signal analysis IEEE Trans. Biomed. Eng. 50 1108–16
[20] Rejér I and Górski P 2015 Benefits of ICA in the case of a few channel EEG 2015 37th Annual Int. Conf. IEEE Eng. Med. Biol. Soc. (Milan: IEEE) pp 7434–7
[21] Rejér I and Górski P 2017 Independent component analysis in a motor imagery brain computer interface IEEE EUROCON 2017-17th Int. Conf. on Smart Technol. (Mcadonia: IEEE) pp 126–31
[22] Davies M E and James C J 2007 Source separation using single channel ICA Signal Process. 87 1819–32
[23] James C J, Gibson O and Davies M 2006 On the analysis of single versus multiple channels of electromagnetic brain signals Artif. Intell. Med. 37 131–43
[24] Mijovic B, De Vos M, Gigorijevic I, Taelman J and Van Huffel S 2010 Source separation from single-channel recordings by combining empirical-mode decomposition and independent component analysis IEEE Trans. Biomed. Eng. 57 2188–96
[25] Taelman J, Mijovic B, Van Huffel S, Devuyst S and Dutoit T 2011 ECG artifact removal from surface EMG signals by combining empirical mode decomposition and independent component analysis Biosignals pp 421–4 (https://pdlse. semanticscholar.org/6f34/0589d3419b6b7b27e9617bdecd17918102f7.pdf)
[26] Inuso G, La Foresta F, Mannone N and Morabito F C 2007 Wavelet-ICA methodology for efficient artifact removal from electroencephalographic recordings 2007 Int. Joint Conf. on Neural Networks (Florida: IEEE) pp 1524–9
[27] Mannone N, La Foresta F and Morabito F C 2012 Automatic artifact rejection from multichannel scalp EEG by wavelet ICA IEEE Trans. Sens. J. 12 533–42
[28] Mahajan R and Morshed B I 2015 Unsupervised eye blink artifact denoising of EEG data with modified multiscale
sample entropy, kurtosis, and wavelet-ICA. IEEE J. Biomed. Health Inf. 19 158–65
[29] Azzerboni B, Carpentieri M, La Foresta F and Morabito F C 2004 Neural-ICA and wavelet transform for artifacts removal in surface EMG 2004 IEEE Int. Joint Conf. on Neural Networks (IEEE Cat. No. 04CH37541) (Hungary: IEEE) pp 3223–8
[30] Girolami M 2001 A variational method for learning sparse and overcomplete representations Neural Comput. 13 2517–32
[31] Dempster A P, Laird N M and Rubin D B 1977 Maximum likelihood from incomplete data via the EM algorithm J. R. Stat. Soc. B 39 1–22 (https://www.jstor.org/stable/2984875?seq=1#page_scan_tab_contents)
[32] Lewicki M S and Sejnowski T J 2000 Learning overcomplete representations Neural Comput. 12 337–65
[33] Rejer I and Górska P 2019 A multi-filtering algorithm for applying ICA in a low-channel EEG ICAISC 2019 Proc. ed L Rutkowski et al (Berlin: Springer) 11509
[34] Langlois D, Chartier S and Gosselin D 2010 An introduction to independent component analysis: InfoMax and FastICA algorithms Tutorials Quant. Methods Psychol. 6 31–8
[35] Tichavský P, Koldovský Z and Oja E 2006 Performance analysis of the FastICA algorithm and Cramér–Rao bounds for linear independent component analysis IEEE Trans. Signal Process. 54 1189–203
[36] Asuero A G, Sayago A and Gonzalez A G 2006 The correlation coefficient: an overview Crit. Rev. Anal. Chem. 36 41–59
[37] Djuwari D, Kumar D K and Palaniswami M 2006 Limitations of ICA for artefact removal 27th Annual Conf. 2005 IEEE Eng. Med. Soc. (Shanghai: IEEE) pp 4685–8
[38] Schalk G, McFarland D J, Hinterberger T, Birbaumer N and Wolpaw J R 2004 BCI2000: a general-purpose brain–computer interface (BCI) system IEEE Trans. Biomed. Eng. 51 1034–43
[39] Goldberger A L, Amaral L A, Glass L, Hausdorff J M, Ivanov P C, Mark R G, Mietus J E, Moody G B, Peng C K and Stanley H E 2000 PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals Circulation 101 e215–20