A NEW METHOD TO DETECT EVENT-RELATED POTENTIALS BASED ON PEARSON'S CORRELATION

William Giroldini*, Luciano Pederzoli*, Marco Bilucaglia*, Simone Melloni* and Patrizio Tressoldi°

*EvanLab, Via dei Ricci, 22 - 50023 Impruneta, 50023 Firenze, Italy, °Dipartimento di Psicologia Generale, Università di Padova, via Venezia, 8, 35131 Padova, Italy

Corresponding author:

Patrizio Tressoldi
Dipartimento di Psicologia Generale-Università di Padova
Via Venezia 8, 35131 PADOVA-ITALY
Tel+390498276500
Fax+390498276600
Email: patrizio.tressoldi@unipd.it

Abstract

Background: Event-Related Potentials (ERPs) are widely used in Brain Computer Interfaces applications and in neurology and psychology for the study of cognitive process, mental disorders, attention deficit, schizophrenia, autism, etc. Because the spontaneous noisy EEG activity is uncorrelated with the ERP waveform, the noise will decrease in the order of $1/\sqrt{(N)}$ (inverse of square root of N), where N is the number of averaged epochs. Since the background EEG activity has a higher amplitude than ERPs waveform, the averaging technique highlights ERPs and attenuates the noise. This is the easiest strategy currently used to detect ERPs.

New Method: In this paper, a new method is proposed, called GW6, in order to calculate the ERP using a mathematical routine based only on Pearson's Correlation.

Results: The result is a graphic with the same time-resolution of the classic ERP that shows positive peaks representing the increase of correlation of the EEG signal in correspondence to the stimuli.

Comparison with Existing Methods: the GW6 method allows highlighting other components of ERP response, usually hidden in the standard and simple method based on the averaging of all the phase and time-locked epochs. For this reason, this new method could be very helpful to investigate these hidden components of the ERP response and to develop applications for medical purposes.

Conclusions: The method we are proposing can be directly used in the form of software written in Visual Basic and easily and quickly implemented in any other programming language.

Highlights

- a new method is proposed, called GW6, in order to calculate the ERP (Event-Related Potential) using a mathematical routine based only on Pearson's Correlation
- The result is a graphic with the same time-resolution of the classic ERP that shows positive peaks representing the increase of correlation of the EEG signal in correspondence to the stimuli
- The method we are proposing can be directly used in the form of software written in Visual Basic and easily and quickly implemented in any other programming language.

Keywords: Event-Related Potentials; Brain-Computer-Interfaces; Pearson’s correlation;

Introduction

The Event-Related Potential (ERP) is an electroencephalographic (EEG) signal recorded from multiple brain areas, in response to a single short stimulus such as visual, auditory, or muscle movement (Linden, 2005; Sanei and Chambers, 2013; Aydin, 2008).

ERPs are widely used in Brain Computer Interfaces (BCIs) applications and in neurology and psychology for the study of cognitive process, mental disorders, attention deficit, schizophrenia, autism, etc.

ERPs are weak signals in comparison with the spontaneous EEG activity, with very low signal-to-noise ratio (SNR) (Linden, 2005; Croft and Barry, 2000), and are typically constituted by two to four waves of low amplitude (4-10 microvolts) with a characteristic positive wave called P300, which occurs with a latency of about 300 milliseconds in response to the stimulus. The detection of ERPs is an important problem and several methods exist to distinguish these weak signals. Indeed, ERP analysis has become a major part of brain research today, especially in the design and development of BCIs (Sano and Bakardjian, 2009).

Since ERP is considered a reproducible response to a stimulus, with relatively stable amplitude, waveform and latency, the standard method to extract ERPs is based on the repeated presentation of the stimulus for about 80-100 times, with a random inter-stimulus time of few seconds. This strategy allows calculating the ERPs averaging several epochs that are time-locked and phase-locked.

Because the spontaneous noisy EEG activity is uncorrelated with the ERP waveform, the noise will decrease in the order of 1/SQR(N) (inverse of square root of N), where N is the number of averaged epochs. Since the background EEG activity has a higher amplitude than ERPs waveform, the averaging technique highlights ERPs and attenuates the noise. This is the easiest strategy currently used to detect ERPs.

In this paper, we are going to illustrate a new method to detect ERPs, also in particularly noisy signals and with significant variations of the latency, known as “latency jitter”.

Our method, named GW6, is less restrictive regarding the issue of jitter. It also allows detecting an ERP when the standard approach, based on the average, fails or gives unsatisfactory result.

Yet the GW6 method do not reproduce the typical biphasic waveform of the ERP but rather an always-positive waveform. For this reason, this procedure is useful together with the classic averaging technique, rather as an alternative to the latter.

In general, to calculate ERPs by the averaging method, essentially three conditions or hypotheses must be satisfied:

1) The signal is time-locked and waveform-locked.

2) The noise is uncorrelated with the signal.

3) The latency is relatively stable (low jitter).

The GW6 method uses extensively the Pearson's Correlation among all EEG signals recorded by a multichannel EEG device. By using the averaging method, it is possible to work also with a single EEG channel, while the GW6 method works with a multichannel EEG device, from a minimum of 10 channels. It is also possible to calculate successively the ERP of each channel.

We are going to describe also a useful signal preprocessing method, named Artefacts Reduction (by means of) Standard Deviation (ARSD).

The EEG preprocessing

The average of epochs is nevertheless only the last step of ERPs calculation. Several preprocessing stages are usually necessary because strong artifacts very easily affect the EEG signals, due to eye movements, the heartbeat (ECG artifacts), head movements, bad electrode-skin contacts, etc. All these artifacts can be several times larger (also 10-20 times or more) than the underlying ERPs, therefore they are able to destroy every average with random waves and peaks which can hide the true ERP waveform.

The first preprocessing step includes the use of a band-pass filter in the range of 1 to 20 Hz obtained with a digital filter, which *must not change the phase of signals.* The Reverse Fourier Transform is suitable for this purpose, among other methods. Many researchers have suggested that the P300 component is primarily formed by transient oscillatory events in delta, theta and alpha band and therefore a 1 to 14 Hz band-pass could be sufficient (Wastell, 1977). In our analysis, we used a 1 to 20 Hz band-pass filtration.

The successive step includes a variety of methods: among the most used, it is included the Independent Component Analysis (ICA) algorithm (Makeig, 1996; Vorobyov and Cichocki, 2002), which allows separating true EEG signal from its undesirable components (twitch, heartbeat, etc.). In general, this method requires a decision-making on what signal component (after separation) is considered undesirable and what should not.

Blind Source Separation (Joyce, Gorodnitsky and Kutas, 2004) is a technique based on the hypothesis that the observed signals from a multichannel recording are generated from a mixture of several distinct source signals. Using this method, it is possible to isolate the original source signal by applying some kind of complex transform to the set of observed signals.

Discrete Wavelet Transform is another method that can be used to analyze the temporal and spectral properties of non-stationary signals (Wang, MaierLeopold, Logothetis and Liang, 2007; Quiroga and Garcia, 2003; Hu et al., 2011).

The artificial neural network, known as Adaptive Neuro-Fuzzy Inference System, was described for P300 detection (Ramírez-Cortes, Alarcon-Aquino, Rosas-Cholula, Gomez-Gil and Escamilla-Ambrosio, 2010). Moreover, the Adaptive Noise Canceller (Ahirwal, Kumar and Singh, 2014) can detects ERPs. You should note that these algorithms are generally very complicated.
The so-called winsorization is a routine which reduces the effects due to eye blinks, voluntary and involuntary eye movements, muscle activity, or subject's movements, that can cause large amplitude outliers in the recorded signal. To reduce the effects of such artifacts, the data from each channel are elaborated and the amplitude values exceeding a lower and an higher percentile threshold, are replaced respectively by the lower and upper percentile (Dixon and Tukey, 1968).

In our software, we used two fast and uncomplicated routines that are:

- Normalization of signals (see Appendix 1, in the final section of this paper), which reduces the effects due to EEG signal's strong fluctuations and artifacts. A useful variant is the ARSD routine also described in the same section. With our experimental data, the ARSD routine followed by Normalization obtains the best results, but normalization, in general, is sufficient.
- Exclusion from the average of all epochs with an amplitude which overcomes a fixed threshold, for example a threshold of 80 microvolts. A drawback of this technique is that a large number of signals could be discarded and consequently the average could be calculated on insufficient data.

Materials and Methods

In this paper, the GW6 method is described using a routine written in Visual Basic language. Finally the method is applied to true EEG signals recorded using a low-cost EEG device, the Emotiv EPOC® EEG Neuroheadset. This headset is wireless and consists of 14 electrodes and two reference electrodes, located and labeled according to the international 10-20 system. Channel names are: AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8 and AF4. The acquired EEG signals are transmitted wirelessly to the computer by means of weak radio signals in the 2.4 GHz frequency band. The Emotiv 's sampling frequency is 128 Hz for every channel and the signals are encoded with a 14-bit definition.
Moreover, the Emotiv hardware operates on the signals with a preliminary band-pass filter in the 0.1 to 43 Hz band, consequently the output signals are relatively free from the 50 Hz power-line frequency, but they are often rich in artifacts.

The Emotiv EPOC® headset was successfully used to record ERPs (Badcock, 2013) although it is not considered a medical-grade device.

We collected and recorded the raw signals from the Emotiv EPOC® headset using a homemade software and a special data-type format based on the .CSV format. The same homemade software was used to give the necessary auditory and/or visual stimulus to the subject.

ERPs were induced by an auditory stimulus (pure 500 Hz sine wave), and a contemporary light flash given using an array of 16 red high-efficiency LEDs. The stimulus length was of 1 second, and the stimuli were repeated from 100 up to 128 times.

Participants were all healthy volunteers, preliminarily informed about the experimentation's purpose. Each participant granted a written consensus. Using the native EEG reference electrode of Emotiv EPOC® headset, we recorded a first set of ERPs. Another set of EEG files, of better quality, were recorded with the reference electrodes connected to the earlobes, a variant that assures better quality of the signals, rather than in the standard configuration of Emotiv EPOC® headset, where the reference electrodes are located in an active zone of the head.

The new algorithm

In many papers describing a mathematical method to analyze something, complex mathematical formulas are usually given, which must be successively translated in a computer-language, for example C, C++, Visual Basic, Java, Python, or else. This step could be very complicated and limit the diffusion and application of some useful methods. In this paper, we will describe this new algorithm directly in an easy and simple well know language, Visual Basic 6, in order to facilitate its application.

We describe the basic idea of this new method in Fig. 1 and 2:

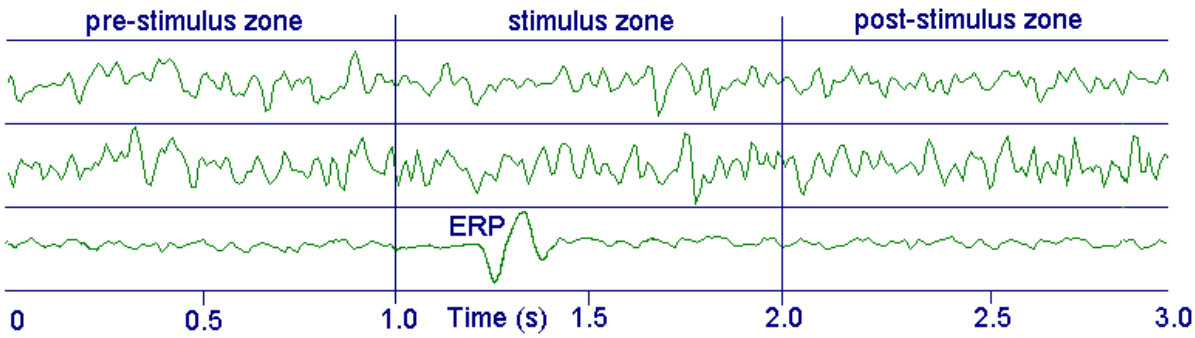


Fig. 1: The upper tracks represents the raw signals of two EEG channels, in time-locked epochs, whereas the lower track is the average of a sufficient number (about 100) of epochs for each channel (ERP is not in scale). The figure shows a positive peak about 300 ms after the stimulus

onset (P300 wave). The ERP has typically a length of about 300-500 ms, depending on the kind of stimulus and band-pass filtering of the signal.

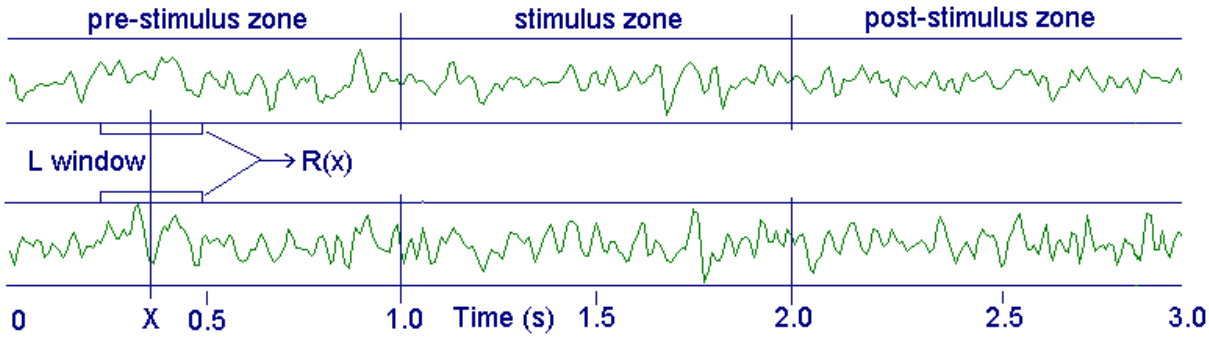


Fig. 2: The data-windows of length L is shifted progressively along the two tracks, and the corresponding Pearson's Correlation is calculated and stored in the vector R(x)

Let us now consider the data-window of length L (about 270 ms) centered in the point X of the signal. We can calculate the Linear Pearson's Correlation between these two data-segments and the result will be a number *r* represented by the vector R(x), which can be calculated for every point X simply by shifting progressively the windows of one unity of sample (running windows). In general, the averaged value of R(x) will change from the pre-stimulus zone to the stimulus zone because the (auditory or/and visual) stimulus changes the correlation between the two EEG signals, which represent the activity of different parts of the brain. An interval about 270 ms long was selected because it represents the typical delay for a conscious response corresponding to the P300 wave.

This change of correlation can appear or as an increase or a decrease with respect to the baseline (the zone preceding the stimulus). Let us consider a true example, based on the Emotiv EPOC® , where the number of channels is NC = 14, the sampling frequency is 128 samples/second, the stimulus length is one second, and the epoch length is 3 seconds. In this case, it is possible to calculate the vector R(x) in a number of pair combinations $Nt = NC*(NC - 1)/2 = 91$.

The result could be expressed using an array R(I, X) where I = 1... 91, and X = 1... 384.

This last number arises from an epoch of 3-seconds length and 128 samples/second, with the stimulus given at sample number 128, and stopped at sample number 256, after one second. In general, we can represent the raw signals as a time-locked array of V(C, X, J) type, where C = 1... 14 are the EEG channels, X = 1... 384 are the samples along 3 seconds, and J is the number of stimulations given to the subject, usually about 100. The entire GW6 procedure is better described in the Visual Basic routine (see Appendix 3).

The following are the stages of elaboration based on the Emotiv EPOC® device, but not limited to this specific device (the numbers here described are only examples):

- **Stage 1**: collection of the raw EEG data in a time-locked way, V(C, X, J); C = 1... 14 , J = 1... 100 is the typical number of stimulation, X = 1… 384 are the samples. Our customized software collects the raw signals and performs this stage.
- **Stage 2**: filtration in a selected band (example: 1-20 Hz, and Normalization or ARSD or both, see Appendix) giving the filtered array: V(C, X, J) + filtration → W(C, X, J).
- **Stage 3**: calculus of the simple average of W(C, X, J) among all epochs, giving the final array Ev(C, X), where: C= 1... 14 and X= 1… 384, which is the simple and classic ERP for each channel.
- **Stage 4**: calculus of all the Pearson's Correlations combinations using a running-window 270 ms long. The result is the array R(I, X), I = 1..91, which is already the average of all stimuli.
- **Stage 5**: calculus of the mean value of baseline for each Nt combination (baseline is balanced and calculated as average from X=1 to 128 + average from X=257 to 384), and subtraction of this baseline from the array R(I, X) taking the absolute value R'(I, X) = Abs[R(I, X – Baseline)].
- **Stage 6**: average along all the Nt combinations (and all the stimulation), giving the final array Sync1(X), which represents the global variation of the EEG correlations during an epoch of 3 seconds. It is also possible to calculate an equivalent array Sync2(C, X) for each channel C.

The global array Sync1(X) and also Sync2(C, X) will show a positive peak in ERP's zone, as represented in Fig. 3. These peaks are examples of the typical results provided by this method:

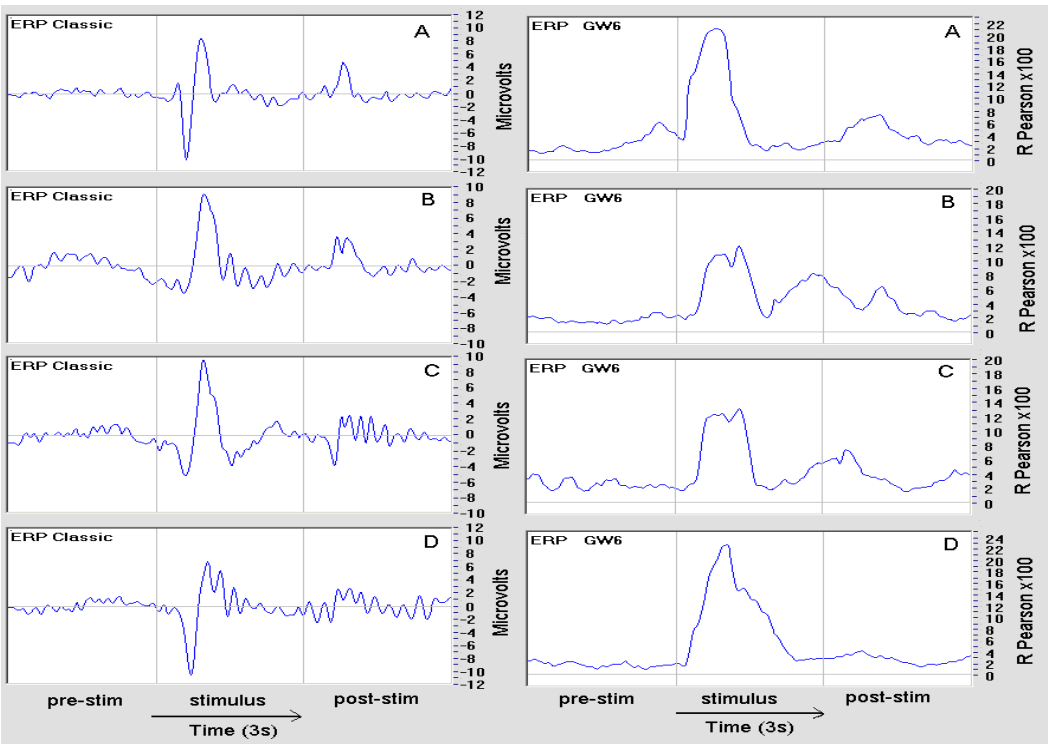


Fig.3: In these pictures, given as example, on the left is shown the classic ERP (amplitude in microvolts). On the right is shown the corresponding GW6 graphic; the result is expressed as R-Pearson value multiplied by 100. All these graphics are the global average of 14 EEG channels and about 120 stimuli; the EEG data were filtered in the band 1-20 Hz and submitted to the routines Normalization or ARSD, or both. In all cases a positive peak is observed in coincidence with the P300 maximum peak, but in several cases, the majority, the positive peak of GW6 graphic is larger than the corresponding classic ERP (see, for example, cases B, C and D).

In order to better inquire the properties of GW6 routine, we wrote an emulation software. In this software, a simple artificial ERP's waveform was added to a random noise, and suitably filtered (low-pass filter) in order to reproduce the typical frequency distribution of the EEG signal.
The artificial ERP signal was mixed with a variable amount of this random signal and submitted both to the classic average and to the GW6 routines (Fig. 4).

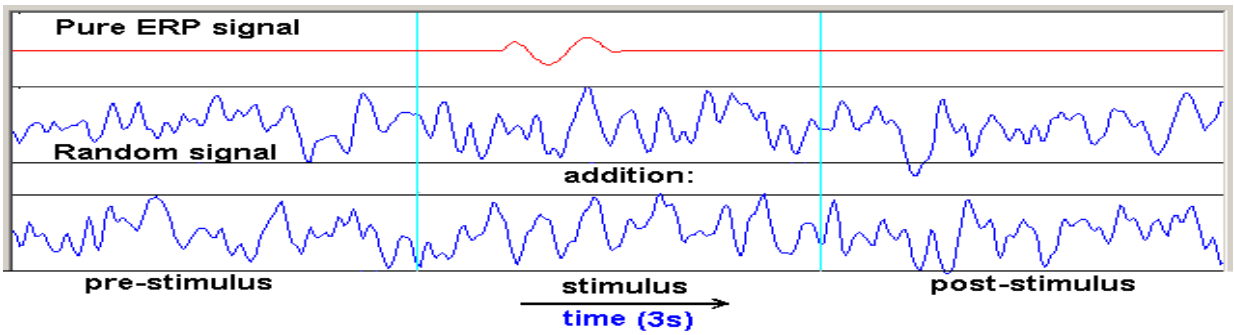
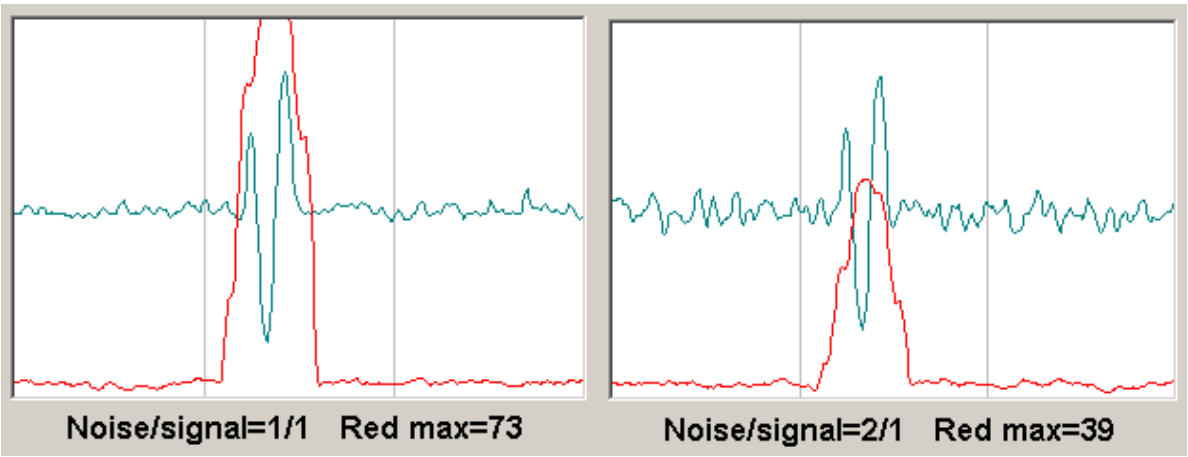


Fig. 4: Artificial ERP signal mixed with a variable amount of a random signal and submitted both to the classic average and to the GW6 routines



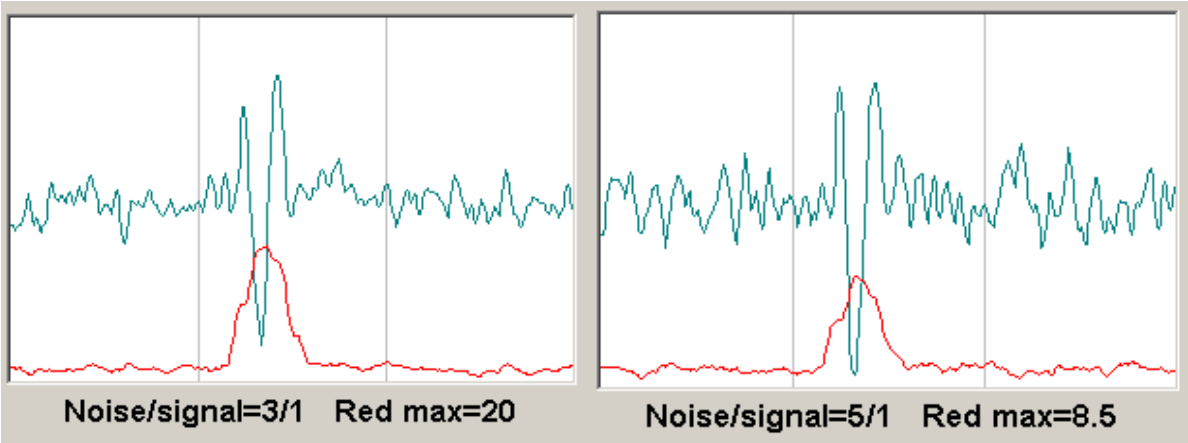


Fig.5. Example of the ERP + Random signal emulation for 4 levels of noise-to-signal ratio.

Fig.5 shows the results of the classical Average routine and of the GW6 routine for a progressive increase of the noise-to-signal ratio, average of 100 ERPs on a single channel. Whereas the final amplitude of the ERP waveform do not change, but becomes progressively noisier, the GW6 graphic amplitude (red curve) drops progressively but with stable residual noise. Very interesting is the emulation of these two routines in presence of the so called "latency jitter", which is an unstable ERP time latency that in some cases could affect the ERPs.

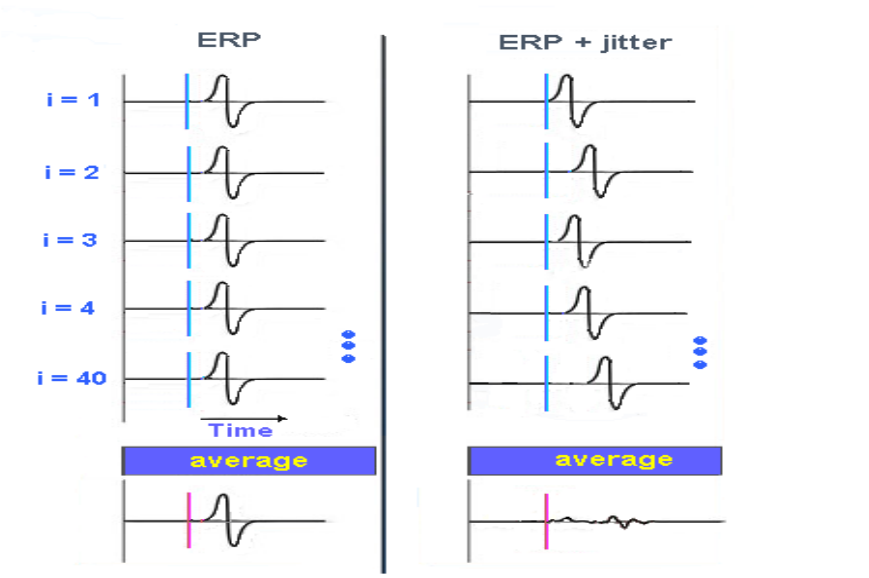


Fig. 6: ERP stable latency on the left and with latency jiitter on the right

When the ERP latency is stable (Fig. 6, left picture), also its average is stable and shows the maximum amplitude. Nevertheless, if latency jitter is present (due to some physiological cause) the corresponding average decreases because each ERP do not combine with the same phase and consequently ERPs show the tendency to cancel reciprocally. This effect is more pronounced as the jitter is increasing.

In the software emulation of Fig. 7 a stable noise-to-signal ratio (3 / 1) was used, but with a random jitter progressively incremented. Moreover, the jitter was random between the ERPs, but was constant for all the channels in each ERP. The results show that the GW6 routine is more resistant to jitter than the simple classic average.

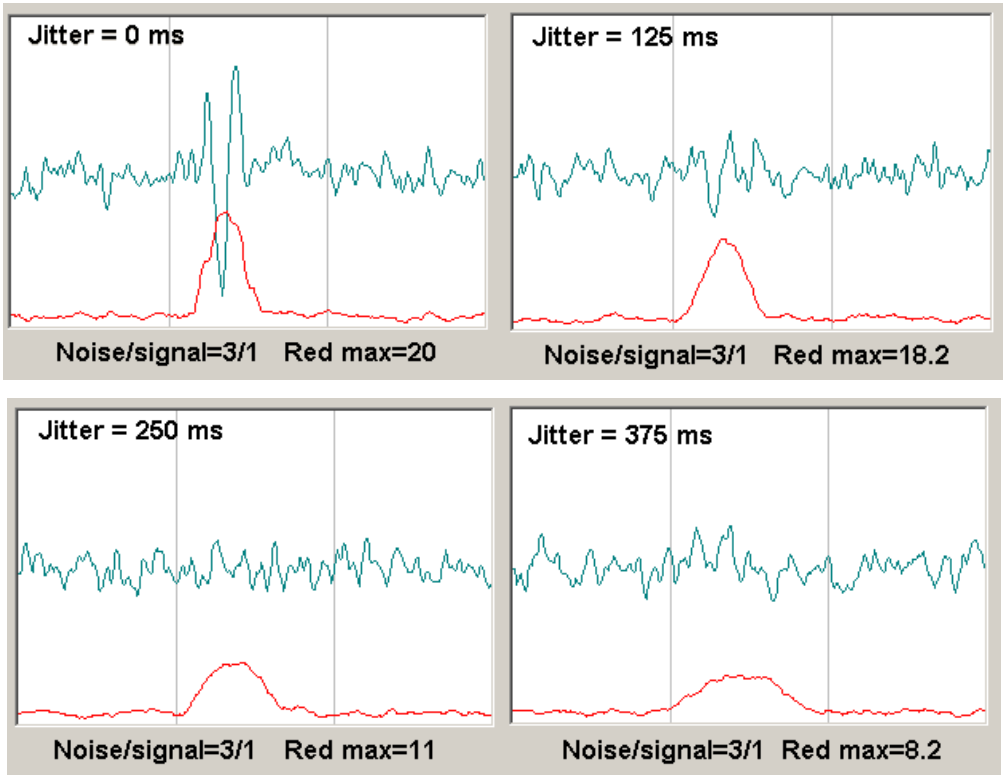


Fig. 7: A stable noise-to-signal ratio (3/1) but with a random jitter progressively incremented.

Whereas the classic ERP waveform disappears rapidly increasing the jitter, the GW6 routine gives a result still identifiable (the red curve), where the amplitude is decreasing but not so rapidly, and the width of the curve is increasing.

This interesting property is very important, because suggests some other possibility about the large GW6 peaks observed in Fig. 3, in particular in B, C and D cases.

Following an intuition, we added a new and simple routine in our software used to analyze the true ERP with the classic and the GW6 routine. At the end of the elaboration, which gives the typical result shown in Fig. 3, we created another routine where to the set of EEG signals W(C, X, J), the classic ERP average was subtracted (see the routine SWXX in Appendix) giving a new array : W'(C, X, J) = W(C, X , J) - Ev(C, X) , then this new data-set was submitted to the stages 3, 4, 5 and 6 previously described. Including this strange operation in our emulation software, and successively performing the same 3, 4, 5 and 6 stages, as a result no any ERP appears, and no any significant GW6 peak appears. This is obvious because we have canceled from the random noise

the ERP component, and consequently nothing is waiting to appear, but that is true only if jitter is zero (Fig. 8 and Fig. 9).

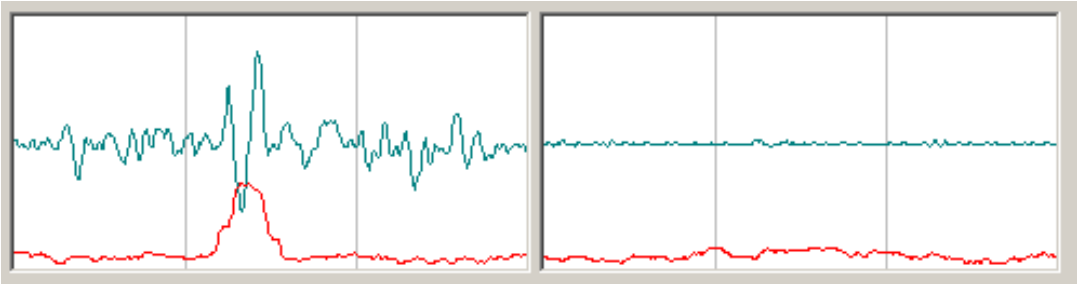


Fig.8 - Case 1. Left: W(C, X, J) from ERP pure wave + random noise, Jitter = 0, average of 100 ERPs. Right: with the same elaboration of the corresponding W'(C,X,J) array both graphics disappear.

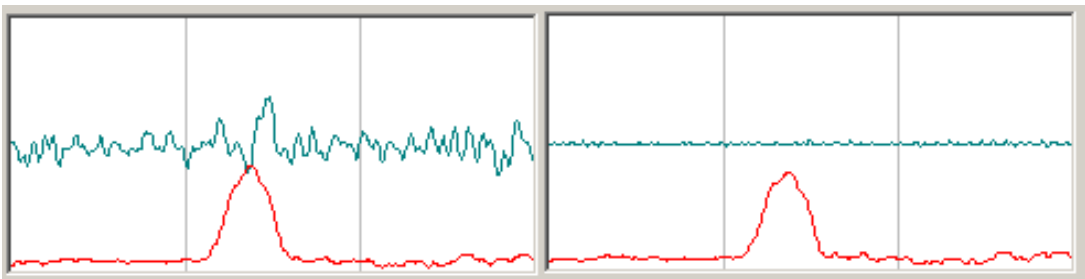


Fig. 9 - Case 2. Left: W(C, X, J) from ERP pure wave + random noise, Jitter = 78ms (from 0 to 78 ms, random), average of 100 ERPs. Right: with the same elaboration of the corresponding W'(C,X,J) array only the classic ERP disappears. In presence of Jitter, the GW6 routine always shows an ERP.

We performed a new variant in the emulation software: beside the pure ERP wave + random signal, we added also a random common signal (RCS) to every channel only in a limited zone near the ERP, *but this RCS is random among the ERPs.* In this variant of emulation, we hypothesized that the stimulus given to the subject could cause not only a simple brain response based on a stable waveform with low jitter (the classic ERP) *but also a non-stable waveform very similar or identical in all the EEG channels.* The simple average does not reveal this kind of electric response, because waveform is near random, but it is easily revealed by the GW6 routine, *which is based on the computation of the variation of correlation among all the EEG channels during the stimulus.*

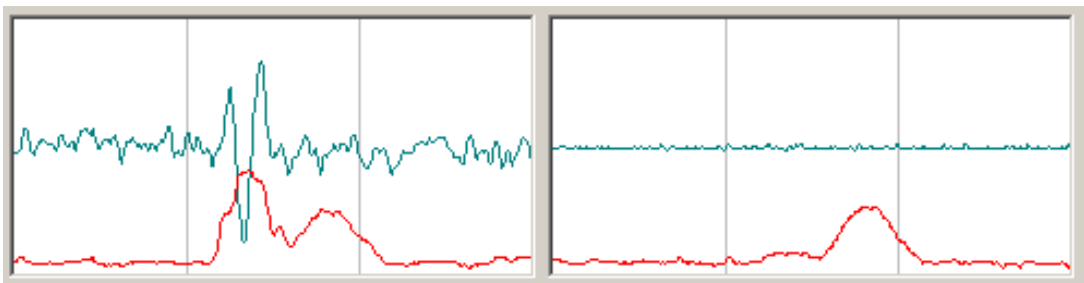


Fig. 10 - Case 3. Left: W(C, X, J) from ERP pure wave + random noise, Jitter = 0, CRS width about 400 ms, average of 100 ERPs. Right: with the same elaboration of the W'(C,X,J) array only the classic ERP disappears, not that due to CRS.

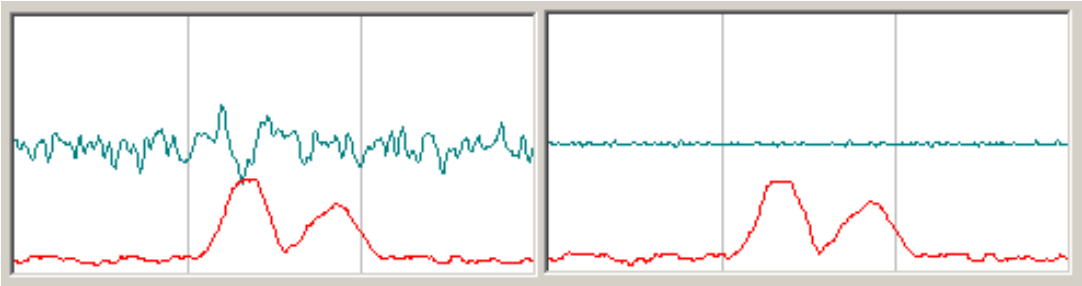


Fig. 11 - Case 4. Left: W(C, X, J) from ERP pure wave + random noise, Jitter = 78 ms, CRS width about 400 ms, average of 100 ERPs. Right: with the same elaboration of the W'(C,X,J) array now both peaks are visible.

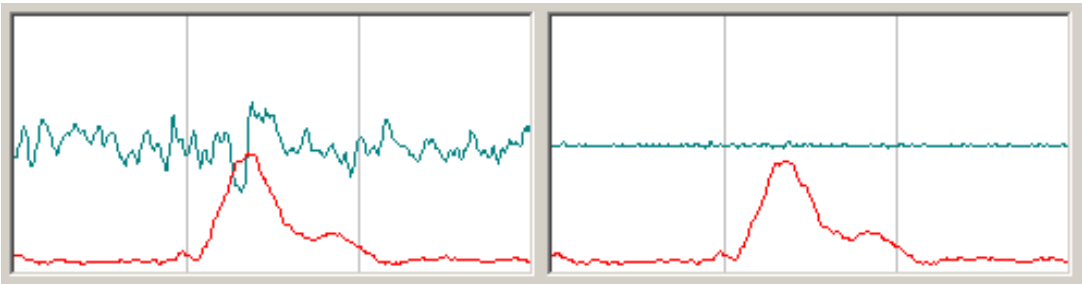


Fig. 12 - Case 5. Left: W(C, X, J) from ERP pure wave + random noise, Jitter =78ms, CRS width about 860 ms, average of 100 ERPs. Right: with the same elaboration of the W'(C,X,J) array now both peaks are overlapped and visible.

We suggest that the two last cases (4 - Fig. 11 and 5 - Fig. 12) are the most representative of the true experimental ERPs. It is easily possible to calculate a great number of combinations and situations using our demonstrative software, which can be freely downloaded at the address www.evanlab/org/.

Now, if we submit our true experimental ERPs to the same procedure, i.e. analysis of the W(C, X, J) data followed by the transform into the W'(C , X, J) data-set and a new analysis, we obtain these typical results (Fig.13):

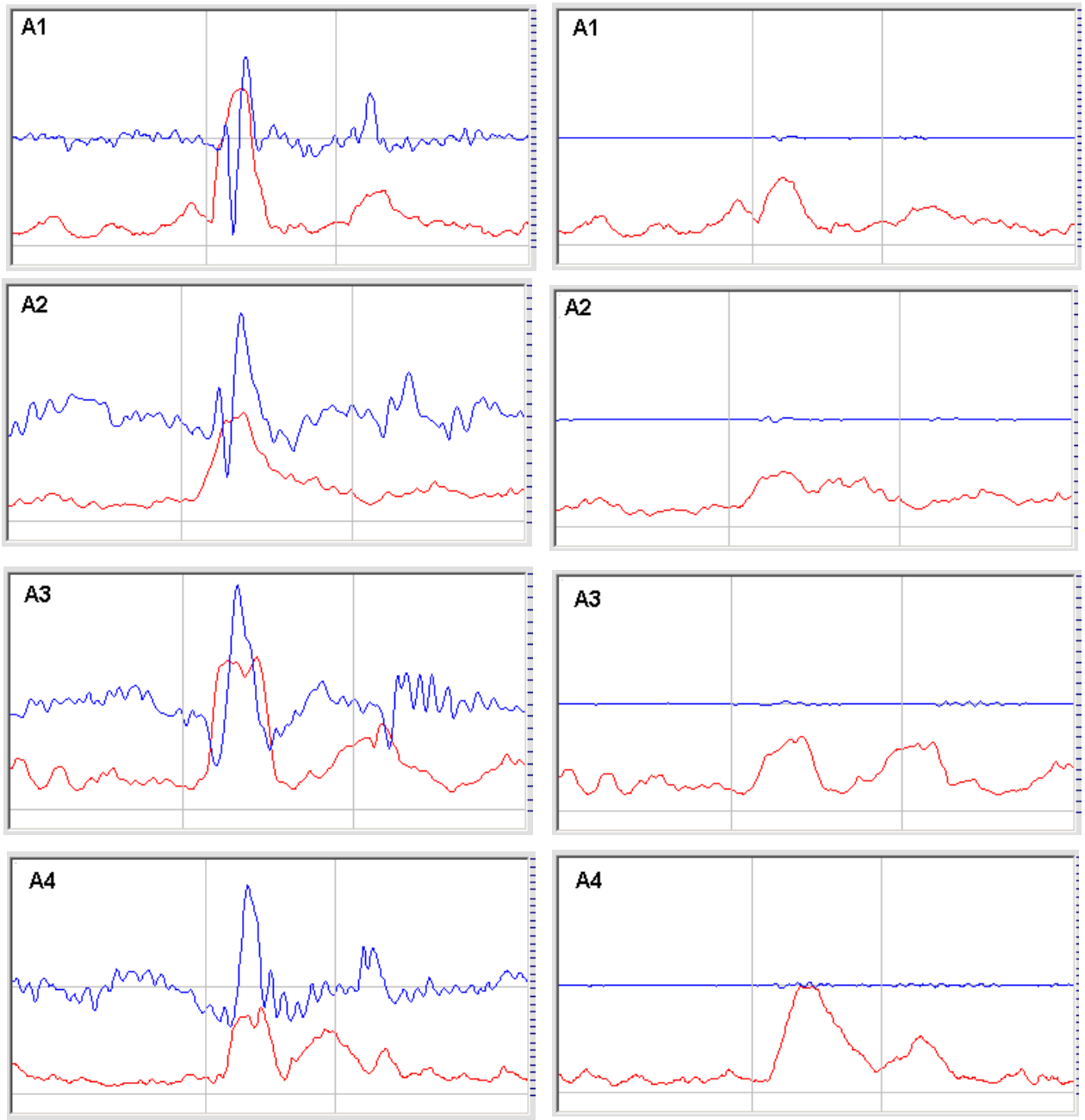


Fig.13: Results of the analysis of the true experimental ERPs analysis of the W(C, X, J) data followed by the transform into the W'(C , X, J) data-set.

In the greater part of cases, after the subtraction of the classic ERP waveform from the EEG data, the GW6 routine (red graphic) shows a reduction in amplitude in correspondence with the standard ERP wave, but other peaks are little changed, and in several cases, the entire graphic is not much changed.

Conclusions

This result could mean that there is some amount of jitter in coincidence with the classic ERP, and/or also other non-phase-locked components, like those hypothesized in the emulation software, both under the classic ERP peak and in other zones of the graphic.

The purpose of this paper nevertheless in not to inquire accurately about the EEG response to a stimulus, but only to propose a new routine for the ERPs' detection, that could become very important for future research about the nature, origin and characteristic of ERPs, on the light of the preliminary result here presented.

In particular, this new method could be very useful to inquire about the hidden components of the ERP response, with possible important application for medical purposes.

Moreover, we emphasize also the choice to give the routines directly in a simple programming language, together with a demonstrative software complete of its source code.

Acknowledgments: this study was partially supported by the research grant n. 124/12 of Bial Foundation. We thank Elena Prati and Claudia Evangelista for English revision.

REFERENCES

Ahirwal, M. K., Kumar, A., & Singh, G. K. (2014). Adaptive filtering of EEG/ERP through noise cancellers using an improved PSO algorithm. *Swarm and Evolutionary Computation*, *14*, 76-91.

Aydin, S. (2008). Comparison of basic linear filters in extracting auditory evoked potentials. *Turkish Journal of Electrical Engineering & Computer Sciences, 16*(2), 111-123.

Badcock, N. A., Mousikou, P., Mahajan, Y., de Lissa, P., Thie, J., & McArthur, G. (2013). Validation of the Emotiv EPOC® ® EEG gaming system for measuring research quality auditory ERPs. *PeerJ*, *1*, e38.

Croft, R. J., & Barry, R. J. (2000). Removal of ocular artifact from the EEG: a review. *Neurophysiologie Clinique/Clinical Neurophysiology*, *30*(1), 5-19.

Dixon, W. J., & Tukey, J. W. (1968). Approximate behavior of the distribution of Winsorized t (Trimming/Winsorization 2). *Technometrics*, *10*(1), 83-98.

Hu, L., Liang, M., Mouraux, A., Wise, R. G., Hu, Y., & Iannetti, G. D. (2011). Taking into account latency, amplitude, and morphology: improved estimation of single-trial ERPs by wavelet filtering and multiple linear regression. *Journal of Neurophysiology, 106*(6), 3216-3229.

Joyce, C. A., Gorodnitsky, I. F., & Kutas, M. (2004). Automatic removal of eye movement and blink artifacts from EEG data using blind component separation. *Psychophysiology, 41*(2), 313-325.

Linden, D. E. (2005). The P300: where in the brain is it produced and what does it tell us?. *The Neuroscientist, 11*(6), 563-576.

Makeig, S., Bell, A. J., Jung, T. P., & Sejnowski, T. J. (1996). Independent component analysis of electroencephalographic data. *Advances in neural information processing systems*, 145-151.

Quiroga, R. Q., & Garcia, H. (2003). Single-trial event-related potentials with wavelet denoising. *Clinical Neurophysiology*, *114*(2), 376-390.

Ramírez-Cortes, J. M., Alarcon-Aquino, V., Rosas-Cholula, G., Gomez-Gil, P., & Escamilla-Ambrosio, J. (2010). P-300 rhythm detection using ANFIS algorithm and wavelet feature extraction in EEG signals. In *Proceedings of the World Congress on Engineering and Computer Science* (Vol. 1, pp. 963-968). San Francisco: International Association of Engineers.

Sanei, S., & Chambers, J. A. (2013). *EEG signal processing*. John Wiley & Sons.

Sano, A., & Bakardjian, H. (2009). Movement-related cortical evoked potentials using four-limb imagery. *International Journal of Neuroscience, 119*(5), 639-663.

Vorobyov, S., & Cichocki, A. (2002). Blind noise reduction for multisensory signals using ICA and subspace filtering, with application to EEG analysis. *Biological Cybernetics*, *86*(4), 293-303.
Wang, Z., Maier, A., Leopold, D. A., Logothetis, N. K., & Liang, H. (2007). Single-trial evoked potential estimation using wavelets. *Computers in Biology and Medicine*, *37*(4), 463-473.

Wastell, D. G. (1977). Statistical detection of individual evoked responses: an evaluation of Woody's adaptive filter. *Electroencephalography and Clinical Neurophysiology*, *42*(6), 835-839.

APPENDIX

Routines of data elaboration written in VB6 language.

Note: the symbol ':' represents the separation between two successive instructions, the declarations about arrays and variables are sometime omitted (in general, they are of Currency or Integer type). Several instructions are written on the same line for compactness reasons.
Now suppose that the W(C, X, J) array represents all the data to be analyzed, where C is the number of channels (14 in the Emotiv Epoc), X the samples from X = 1 to 384, equivalent to 3 seconds at 128 samples/second, J = 1 to Ns is the number of stimulus given to the subject (usually Ns = 100). Moreover, we suppose that the stimulus is given at X = 128 and stops at X = 256 (duration: 1 second).
Some other support array and variables are used in this routine. Data in the W(C, X, J) array must be with zero-average (easy to implement, here omitted). The unity of the W(C, X, J) data could be in microvolts, or raw data at 10, 12, 16 bit, etc.

Appendix 1: preprocessing of EEG data with the method here called "Normalization"

```
Private Sub Normalize()
Dim A As Currency, Sd As Currency, X As Integer, C As Integer, J As Integer
' This normalization step create epochs with a shape identical to that of the original EEG
' signal but transformed into a uniform scale to allow epochs to be easy combined and analyzed
' across the subjects. Moreover, the original signal amplitude is recovered using a constant K
' where K is the Standard Deviation of a good quality signal , experimentally found with the
' specific instrument used. Here the number K = 20 is used.
' The array W(C, X, J) must be at zero-average in each channel. See the Appendix 2.

N1 = 384 :  NC = 14 :  Ns = 100  'definition of the value of data
For J = 1  To  Ns
    For  C = 1  To  NC  :  A = 0  :  Sd = 0
       For  X = 1  To  N1  :  A = A + W(C, X, J)  :  Next X
      A = A / N1   'average of signal
         For  X  =  1  To  N1  :  Sd = Sd + (W(C, X, J) - A) ^ 2  :  Next X
        Sd = Sd / N1  :  Sd = Sqr(Sd)  'standard deviation of each channel
       If  Sd = 0 Then  Sd = 1
       For  X = 1  To  N1  :  W(C, X, J) = 20 * (W(C, X, J) - A) / Sd  :  Next X
  Next  C
Next  J
End Sub
```

Appendix 2: preprocessing of EEG data with the method named "ARSD"

```
Private Sub ArtefARSD()
Dim A As Currency, B As Currency, Pm(20) As Single
Dim Iter As Integer, Fux  As  Integer, X  As  Integer, C  As  Integer,  J  As  Integer
' This routine works like the Normalization but with some differences. In several cases it
' works better or is very similar to Normalization. This routine reduces big artifacts
' to smaller artifacts. Good quality signals are unchanged.
' The best is to use the  ARSD, followed by Normalization.
N1 = 384 : NC = 14 : Ns = 100 'definition of the data-value
' first step: zero average of all the EEG channel, could be omitted if already done.
For J = 1  To  Ns
  For C = 1  To  NC  :  A = 0
     For X = 1  To  N1  :  A = A + W(C, X, J)  :  Next X
     A =  A / N1
     For X = 1  To  N1  :  W(C, X, J) = W(C, X, J) – A  :  Next X
  Next  C
Next  J
For  J = 1  To  Ns
    For C = 1  To NC
        For  Iter = 1  To 20
        A = 0 :  B = 0 :  Pm(C) = 0
        For X = 1 To N1  :  A = A + W(C, X, J)  :  Next X
       A = A / N1  :  B = 0
       For  X = 1  To  N1  :  B = B  +  (W(C, X, J) - A) ^ 2  :  Next X
       B = B / N1  :  Pm(C) = Sqr(B)   'Standard Deviation for each channel
' the original data are corrected until a threshold <= 34 where the number 34
' is about 1.7* Sd ( Standard Deviation) of a good signal experimentally found with
' the device used. In our device, good  Sd is  20  and then 1.7*20 = 34
      Fux = 0
      If  Pm(C) >  34 Then
         Fux = 1
         For  X = 1  To  N1
         If  Abs(W(C, X, J) ) >=  Pm(C)  Then  W(C, X, J)  =  0.7 * W(C, X, J )
         Next X
       End If
      If  Fux  =  0  Then  Exit  For
      Next  Iter
      ' the resulting signal is amplified by the factor 1.4 to recover the correct amplitude
      For  X = 1  To  N1  :  W(C, X, J ) = 1.4 * W(C, X, J)  :  Next X
   Next  C
Next  J
End Sub
```

Observation: The use of ARSD + Normalization (in this order) greatly reduces the artifacts in the EEG signals without changing the phase and with a minimum change in the frequency distribution. This combination is very fast, efficient and easy to implement in any software language

Appendix 3: the essential core of the GW6 routine

Note: in this routine we must suppose that the W(C, X, J) array previously described, should have data also in the range from X= - 30 to (N1 + 30), so the precedent routines should be modified in order to fill the entire set of data.

```
Private Sub GW6()
Dim  V1(800) As Single,  V2(800) As Single,  A1(800) As Single
Dim F1  As Single,  F2 As Single,  F3 As Single,  I As Integer,  J As Integer
Dim XM  As Single,  YM As Single,  R  As  Single,  X  As Integer
Dim  Ax  As Integer,  Bx  As Integer,  Np As Integer,  C  As Integer
Dim Nz  As Integer,  A  As Integer,  A2(200) As Single,  U As Integer
' the array Cm(92, 800) and the variable Nt should be declared in the Option Explicit of VB6
' moreover each element of the array Cm(92, 800) must be put to zero before this routine
N1  =  384  :  NC = 14  :  Ns = 100 'definition of the value of data
Np  =  34      ' 34 is a windows of about 270 ms, at 128 samples/s , could be changed
Nt  =  (NC ^ 2 - NC) / 2 'Nt = 91 in our case is the number of combination with 14 channels
For  J  =  1  To  Ns   'for all the stimuli given
   U = 0  :  I = 0
   For  Ax  =  1  To  (NC - 1)
      For Bx  =  (Ax + 1)  To  NC
      I  =  I + 1  'counter of the progressive combinations of two channels
         For  U = 1  To  N1
         X1  =  U - (Np / 2)  :  X2 = U + (Np / 2)  :  A = 0
         For  X  =  X1  To  X2  :  A = A + 1
        V1(A)  =  W(Ax, X, J ) :  V2(A) = W(Bx, X, J)  :  Next X
        Nz  =  A
        GoSub  Correlas
        A1(U) = R  :  A = 0
        Next U
        For  X = 1  To  N1  :  Cm(I, X) = Cm(I, X) + A1(X)  :  Next X
      Next  Bx
   Next  Ax
Next  J
For  I = 1   To   Nt  'averaging along all the Ns stimuli
  For  X = 1  To  N1  :  Cm(I, X) = Cm(I, X) /Ns  :  Next X
Next  I
' now we have the array of output   Cm(I , X) where I =  1...91,   X = 1...384
' This array could be submitted to some low-pass filtration, see the routine ClassicERP
Exit Sub
Correlas:  'Pearson's correlation
F1 = 0 :  F2 = 0 :  F3 = 0  :  XM = 0 :  YM = 0
For  X = 1  To  Nz  :  XM = XM + V1(X)  :  YM = YM + V2(X)  :  Next X
XM = XM / Nz  :  YM = YM / Nz
For   X = 1  To  Nz  :   F1 = F1 + (V1(X) - XM) * (V2(X) - YM)
F2 = F2  +  (V1(X) - XM) ^ 2  :  F3 = F3 + (V2(X) - YM) ^ 2
Next  X
F1 = F1 / Nz  :  F2 = F2 / Nz  :  F3 = F3 / Nz
If  F2 = 0   Or   F3 = 0  Then  R = 0  :  Return
R = 100 * F1 / Sqr(F2 * F3) ' the r of Pearson is multiplied by 100
Return
End Sub
```

Appendix 4: successive elaboration of the GW6 output array Cm(I, X)

```
Private Sub FinalElab()
Dim Bs(100) As Single , B1 As Single, B2 As Single , C As Integer, J As Integer,
Dim X As Integer, I As Integer
'The array Sync1(X) and Sync2(C, X) should be defined in the Option Explicit section of VB6
' Nt = 91 is defined in the previous routine, this variable should be declared in Option Explicit
B1= 128  :  B2 = B1+ 128  :  N1 = 384
' the values B1 and  B2 are the  definition of the zone for balanced baseline calculation
' in particular, the first zone is the pre-stimulus (from X=1 to B1, a second zone is
' the post-stimulus zone, it is important for good results to calculate baseline from both zones.
' each element of the array Sync1(X) and Sync2(C,X) are put to zero
For  X = 1  To  N1  :  Sync1(X) = 0 : Next X
For  C = 1  To   NC
   For X = 1  To   N1  :  Sync2(C, X) = 0 :  Next X
Next C
For   J = 1   To   Nt  :  Bs(J)=0 :  A=0
   For   X = 1   To   B1  :  A = A + 1  :  Bs(J) = Bs(J) + Cm(J, X)  :  Next X
   For   X = B2  To  N1  :  A = A + 1  :  Bs(j) = Bs(J) + Cm(J, X)  :  Next X
   Bs(J) = Bs(J) / A  ' baseline for each combination
Next  J
For   J = 1   To  Nt
  For  X = 1  To  N1   :  Sync1(X) = Sync1(X) + Abs(Cm(J, X) - Bs(J))  :  Next X
Next   J
For  X = 1  To  N1  :  Sync1(X) = Sync1(X) / Nt  :  Next X
' Now the array Sync1(X) is the average (global average) of Correlation in all the Nt combinations
' and  for all the  Ns number  of  ERPs
' Calculus of the array Sync2(C, X) for each channel
I = 0
For   Ax = 1  To   (NC – 1)
    For   Bx = (Ax + 1)  To  NC
       I =  I + 1   'counter of all the combinations of the channels
       For  U = 1  To  NC
      If  (U = X)   Or   (U = Bx)  Then
         For  X = 1  To   N1   :   Sync2(U, X) = Sync2(U, X) + Abs(Cm(J, X) – Bs(J))  :  Next X
     End If
     Next U
  Next Bx
Next Ax
For  C = 1   To   NC
   For  X = 1  To  N1  :  Sync2(C, X)  =  Sync2(C, X) / (NC-1)  :  Next X
Next  C
' Now  the  array  Sync2(C, X) is  the  Correlation  for  each  channel.
' Each  channel  is  the average  of  (NC-1)  data.
End Sub
```

Appendix 5: classic calculus of ERP by simple averaging, from the array W(C, X, J)

```
Private Sub ClassicERP()
Dim J As Integer, C As Integer, X As Integer, K As Single
' The array Ev(C,X) should be declared in the Option Explicit section of VB6
For C = 1 To NC
   For X = 1 To N1 : Ev(C, X) = 0 : Next X
Next C
For J = 1 To Ns        'for all the stimuli
   For C = 1 To NC    'for all the chanels
      For X = 1 To N1 : Ev(C, X) = Ev(C, X) + W(C, X, J) : Next X
   Next C
Next J
For C = 1 To NC
   For X = 1 To N1 : Ev(C, X) = Ev(C, X) / Ns : Next X
Next C
' Now the array Ev(C, X) is the classic ERP ; this array could be also submitted to some low-pass
' filter like this here described: the value 0.5 is arbitrary, more higher, more lower-pass filtered.
K = 0.5
For C = 1 To NC : Ev(C, 0) = Ev(C, 1) : Ev(C, N1+1) = Ev(C, N1)
  For X = 1 To N1
   Ev(C, X) = (Ev(C, X) + K*Ev(C, X-1) + K*Ev(C, X + 1)) / (1 + 2*K)
  Next X
Next C
End Sub
```

Appendix 6: subtraction of the classic ERP from the array W(C, X, J)

(described only for research purpose, this routine is not to be used normally with the GW6)

```
Private Sub SWXX()
Dim X As Integer, J As Integer, C As Integer
' very simple routine to subtract the classic ERP component from the EEG data
For J = 1 To Ns      'all the erps
   For C = 1 To NC   'all the channels
      For X = 1 To N1 : W'(C, X, J) = W(C, X, J) - Ev(C, X) : Next X
    Next C
Next J
End Sub
```