(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
08 October 2020 (08.10.2020)

WIPO | PCT

(10) International Publication Number
WO 2020/198787 A1

(51) International Patent Classification:
A61B 5/0484 (2006.01)

(21) International Application Number:
PCT/AU2020/050311

(22) International Filing Date:
30 March 2020 (30.03.2020)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
2019901078 29 March 2019 (29.03.2019) AU

(71) Applicants: **AUSTRALIAN HEARING SERVICES** [AU/AU]; Level 5, 15 University Avenue, Macquarie University, New South Wales 2109 (AU). **UNIVERSITY OF GRANADA** [ES/ES]; University of Granada, Technology Transfer Office, Gran Via de Colon 48, 3rd Floor, ES-18071 Granada (ES).

(72) Inventors: **SEGURA LUNA, Jose Carlos**; c/o University of Granada, Technology Transfer Office Gran Via de Colon 48, 3rd Floor, ES-18071 Granada (ES). **DE LA TORRE VEGA, Angel**; c/o University of Granada, Technology Transfer Office Gran Via de Colon 48, 3rd Floor, ES-18071 Granada (ES). **VALDERRAMA VALENZUELA, Joaquin Tomas**; c/o Australian Hearing Services, Level 5, 16 University Avenue, Macquarie University, New South Wales 2109 (AU).

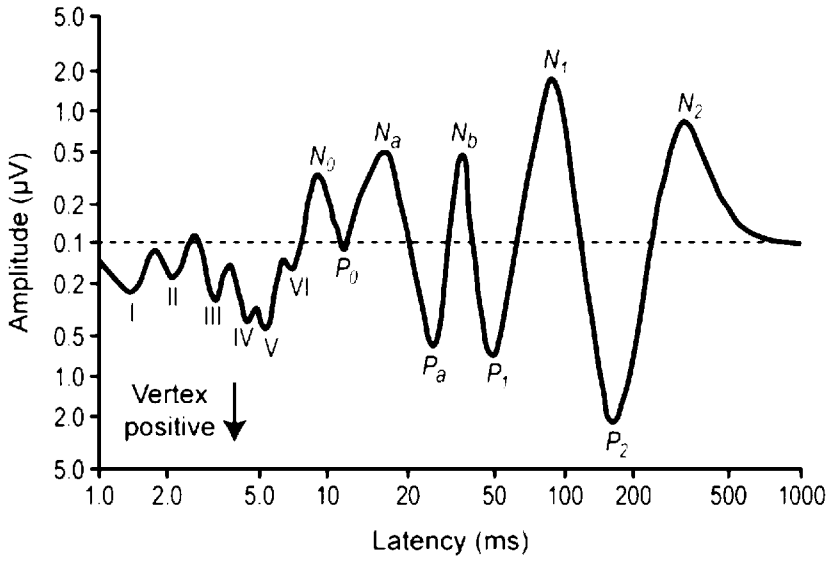
(74) Agent: **FAL PATENTS PTY LTD**; Level 14 / 114 William Street, Melbourne, Victoria 3000 (AU).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ,

WO 2020/198787 A1

(54) Title: METHOD FOR FLEXIBLE DECONVOLUTION OF AUDITORY EVOKED POTENTIALS

FIGURE 2



(57) Abstract: The invention generally relates to a method and system of estimating the transient auditory evoked potential ('AEP') responses of a subject, the method comprising: generating a digital auditory stimulus signal consisting of at least one auditory stimulus type; presenting the at least one auditory stimulus type to a subject via a transducer; recording an electroencephalogram signal ('EEG') including the neural response of the subject to the at least one auditory stimulus type; synchronizing the digital auditory stimulus signal with the recorded EEG; and deconvolving the overlapping AEP responses of the subject from the EEG by applying an iterative randomized stimulation and averaging ('IRSA') technique, wherein the step of applying an IRSA technique is performed with matrix operations in the representation spaces of the AEP and the EEG.

[Continued on next page]

OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

METHOD FOR FLEXIBLE DECONVOLUTION OF AUDITORY EVOKED POTENTIALS

FIELD

[0001] The present invention is generally directed to a system and method for flexible deconvolution of auditory evoked potentials.

BACKGROUND OF THE INVENTION

[0002] Auditory evoked potentials (AEPs) are a set of low-amplitude voltage waves that represent the synchronous activity of neurons in different stages of the auditory pathway in response to a sound stimulus. The recording of AEPs has at least two important applications. Firstly, evaluation of the morphology of these waves allows investigation of the neural structures that encode and process the sounds we perceive, which in turn helps our understanding of the working of the auditory system. Secondly, AEPs are widely used in the clinic as a physiological measure of the state of an auditory system, which in turn allows evaluation of subjects that cannot provide a reliable behavioural response to a sound stimulus, like newborns or adults with dementia.

[0003] Figure 1 shows three different transient AEPs: [left] auditory brainstem responses (ABR); [centre] middle latency responses (MLR); and [right] cortical auditory evoked potentials (CAEP). The main components of the ABRs are waves I, III and V, representing neural activity from the cochlea, brainstem and midbrain, respectively. The MLR components originate from the thalamus, the medial geniculate body and primary auditory cortex. In CAEPs, the P_1-N_1-P_2 complex is elicited in the primary and secondary auditory cortex.

[0004] The recording process of AEPs consists of placing surface electrodes on different positions on the head of a subject and presenting a large number of repetitions of a specific auditory stimulus to the subject, typically using insert earphones. The signal recorded at the electrodes is known as electroencephalogram (EEG), and includes the neural response to the stimulus (the signal of interest) and noise artefacts of different nature (electrophysiological, electromagnetic, electronic, myogenic, etc.). Since the signal-to-noise ratio (SNR) of the signal of interest in the raw EEG is very low (usually around -20 or -30 dB), it is a common practice to (1) filter the EEG within the frequency range in which the AEPs are present (ABRs: [100-3000] Hz; MLRs: [10-300] Hz; CAEPs: [1-30] Hz), and (2) average the segments of the EEG that contain the signal of interest to increase the SNR of the response.

[0005] The conventional method of stimulation consists of delivering the auditory stimuli periodically, i.e. with a fixed inter-stimulus interval (ISI). The conventional stimulation method has the important limitation that the ISI must be greater than the duration of the AEP to avoid

contamination of the recording by adjacent responses; otherwise it would not be possible to recover the overlapping AEP. Taking into account that the typical duration of ABR, MLR and CAEP signals is around 10 ms, 100 ms, and 400 ms respectively (see Figure 1); ABRs, MLRs and CAEPs cannot be recorded with the conventional method at rates higher than 100 Hz, 10 Hz, and 2.5 Hz, respectively.

[0006] However, the recording of these AEPs at faster rates (when the evoked responses are overlapping) presents several advantages. First, the possibility of obtaining transient AEPs without limiting to a minimum ISI provides a large degree of flexibility when designing research audiology experiments. Second, the recording of AEPs at high stimulus rates allows evaluation of the auditory system under a stressed condition (also known as “neural adaptation”), which can provide useful clinical information (e.g. subjects with autism spectrum disorders tend to manifest a deficit in neural adaptation). Further, recording transient AEPs in conditions in which the neural responses are overlapping is critical to understand and measure the neural response of the auditory system to more ecologically valid stimuli, like real running speech.

[0007] The mathematical process that disentangles overlapping responses is known as deconvolution. One characteristic of deconvolution algorithms is that the ISI must not be fixed, rather it must present a certain amount of dispersion from a periodic presentation, also known as jitter.

[0008] Moreover, it is also common that conventional electrophysiology experiments present a single type of stimulus, and it is assumed that all stimuli evoke a response with the same morphology (time-invariant assumption). However, some experiments may require the assumption of a multi-response model in which different types of stimuli evoke different types of responses. For example, if two different auditory stimuli types are presented simultaneously (e.g. clicks at 80 and 30 dBHL), it is reasonable to assume that each type of stimulus would evoke a different type of neural response. Deconvolution of overlapping responses with different morphology (multi-response deconvolution) is not possible with most existing deconvolution methods.

[0009] In addition, since ABR, MLR and CAEP components represent the neural activity of different stations of the ascending auditory pathway, it would be desirable to have a representation of these components in a single plot (like the diagram shown in Figure 2), rather than separated in three different signals (as exemplified by Figure 1). Unfortunately, since each portion of the response is characterized by a different latency and bandwidth (i.e. ABR components contain frequencies in the range 100-3000 Hz, MLRs in 10-300 Hz, and CAEP in

1-30 Hz), obtaining a real signal like the one shown in Figure 2 is not straightforward when using existing processing methods.

The EEG model

[0010] The digital EEG $y(n)$ recorded in an evoked-potential recording procedure is usually modelled as a convolutional process:

$$y(n) = s(n) * x(n) + n_0(n) \quad (1)$$

where n is the index of the samples (with $n \in \{0, \dots, N-1\}$, N being the number of samples of the EEG); $s(n)$ is the stimulation signal consisting of one impulse at the beginning of each stimulation event; $x(n)$ represents the evoked response to each stimulus (with $x(n)$ null for $n > (J-1)$, J being the length of the evoked response); $n_0(n)$ represents the noise affecting the EEG; and the asterisk $(*)$ represents convolution. If the stimulation signal contains K events at the samples m_k, the stimulation signal can be written as:

$$s(n) = \sum_{k=0}^{K-1} \delta(n - m_k) \quad (2)$$

Where $\delta(n)$ is the unitary impulse at $n = 0$. Taking into account that $x(n) * \delta(n - m_k) = x(n - m_k)$, the EEG can be rewritten as:

$$y(n) = \sum_{k=0}^{K-1} x(n - m_k) + n_0(n) \quad (3)$$

Overlapping responses – 40 Hz-ASSR

[0011] The 40 Hz-ASSR is a steady state evoked response resulting from overlapping MLRs presented at 40 stimuli per second (Galambos et al., 1981 - Ref 8; Bohorquez et al., 2008 - Ref 4).

[0012] Figure 3 shows the effect of using an ISI lower than the averaging window using synthesized signals. The first signal shows MLR signals evoked by stimuli whose ISI is 333 ms (stimulus rate of 3 Hz, i.e. 3 stimuli per second). The top signal shows the instants in which the stimuli are presented. This example shows that the responses do not overlap because the ISI is longer than the averaging window (100 ms in MLRs).

[0013] At 8 Hz (second signal, with ISI = 125 ms) the responses do not overlap; however, at rates greater than 10 Hz (with ISIs lower than 100 ms) the responses do overlap.

[0014] These examples show that overlapping responses result in a periodic signal (of a period equal to the stimulus period) in which the original response is contaminated with

adjacent responses. Depending on the stimulus rate, this contamination can be constructive or destructive. Figure 3 shows that the stimulus rate of 40 Hz produces a constructive interference that makes that the resulting signal presents a greater amplitude than the original one, which may facilitate neural response detection. This constructive interference is a consequence of the Na-Pa component overlapping with the Nb-Pb component of the adjacent response, resulting in a greater-amplitude signal (Galambos et al., 1981 - Ref 8; Bohorquez et al., 2008 - Ref 4). This auditory evoked potential evoked by a stimulus rate of 40 Hz is known as 40 Hz auditory steady-state response (ASSR). In contrast to transient responses, which are analysed in the time domain, ASSR signals are typically analysed in the frequency domain.

[0015] The main limitations of the 40 Hz-ASSR are that:

while steady-state signals are useful in neural response detection, they do not show the neural activity of the different generators (as transient responses do), and therefore, they cannot be used to determine the site of lesion or to understand how a specific section of the auditory pathway responds to a stimulus;

the stimulus sequence is fixed to this particular stimulus rate (40 stimuli per second), and therefore it lacks the flexibility required in the design of some specific research and clinical tests; and

since the neural generators cannot be determined in steady-state signals, it could be the case that these signals are driven by a stimulus artefact (rather than by a neural response), potentially leading to a misleading analysis.

Deconvolution methods

[0016] A number of mathematical processes have been developed to estimate the transient evoked response $x(n)$ from a stimulus signal $s(n)$ whose ISI is lower than the duration of the evoked response (i.e., the neural responses are overlapping). As mentioned earlier, these methods require a certain amount of jitter (or variation in the ISI distribution). The most relevant methods are: continuous loop averaging deconvolution (CLAD, Bohorquez and Ozdamar, 2006 - Ref 5), quasi-periodic sequence deconvolution (QSD, Jewett et al., 2004 - Ref 11), maximum length sequences (MLS, Eysholdt and Schreiner, 1982 - Ref 7), and least-squares deconvolution (LS, Bardy et al., 2014a - Ref 1). The general approach of these methods is outlined below.

[0017] The EEG model presented in equation (1) can also be presented in the frequency domain as

$$Y(f) = S(f) * X(f) + N_0(f) \quad (3a)$$

and the evoked response can be estimated in the frequency domain by working out equation (3b)

$$X(f) = \frac{Y(f)}{S(f)} + \frac{N_0(f)}{S(f)} \quad (3b)$$

[0018] Finally, the transient evoked response can be converted back to the time domain by applying the Inverse Fourier Transform to $X(f)$, i.e. $x(n) = IFFT\{X(f)\}$.

[0019] The CLAD, QSD, MLS and LS algorithms describe different methods to obtain a stimulus signal $s(n)$ that minimize the possibility of obtaining frequency components near zero, which would significantly increase the noise in the deconvolution process, as shown in the term $N_0(f)/S(f)$ in equation (3b).

[0020] One important common limitation of these methods is that, since stimulus signals must accomplish several constraints to avoid increasing noise in the deconvolution process, generating efficient sequences is a sensitive process which may reduce flexibility when designing certain experiments, particularly when the ISI is significantly lower than the duration of the evoked response and the amount of jitter is small.

[0021] From the aforementioned methods, only the LS algorithm allows multi-response deconvolution.

ADJAR-Level 1 and randomised stimulation and averaging (RSA)

[0022] ADJAR-Level 1 (Woldorff, 1993 - Ref 20) and randomized stimulation and averaging (RSA, Valderrama et al., 2012 - Ref 14) provide an estimate of the response by synchronous averaging of the EEG:

$$\hat{x}(j) = \frac{1}{K}\sum_{k=0}^{K-1} y(j + m_k) \quad (4)$$

with j ∈ $\{0, \ldots, J-1\}$. When the length of the response J is smaller than the minimum ISI, this estimation is only affected by the noise, and averaging a large enough number of responses provides an accurate estimate.

[0023] The limitation of these methods is that if the responses are overlapping, the interference associated with adjacent responses degrades the estimation, and the estimated response is not reliable.

ADJAR-Level 2 and iterative randomized stimulation and averaging (IRSA)

[0024] With a similar approach, ADJAR-Level 2 (Woldorff, 1993 - Ref 20) and iterative randomized stimulation and averaging (IRSA, Valderrama et al., 2014a - Ref 15) aim to overcome the effect of the aforementioned interference. The main idea of IRSA is that the interference can be estimated using the estimated response $\hat{x}(j)$ and therefore, a more accurate response can be iteratively estimated by averaging a modified EEG in which the interference associated with adjacent responses is suppressed. By using the estimated response at iteration i, $\hat{x}_i(j)$, an interference-free EEG for the k^{th} stimulus (i.e. in which the interference from all the stimuli except the k^{th} stimulus is suppressed) can be derived as:

$$y_{k,i}(n) = y(n) - \sum_{k'=0,k'\neq k}^{k-1} \hat{x}(n - m_{k'}) \quad (5)$$

and the evoked response can be estimated at iteration $i+1$ by averaging the EEG portions without interference:

$$\hat{x}_{i+1}(j) = \frac{1}{k}\sum_{k=0}^{k-1} y_{k,i}(j - m_k) \quad (6)$$

[0025] Applying this approach, each iteration provides a better estimate of the evoked response, and a more accurate suppression of the interference is therefore obtained in the next iteration. As a result, the effect of the interference caused by overlapping responses is minimized iteratively.

[0026] In general, an evoked-potential recording session involves a large number of stimuli and a long EEG. Therefore, this existing IRSA technique can be unpractical, because at each iteration i, k EEGs should be calculated (each one including the k^{th} response and suppressing all the others), leading to a large amount of computation. However, the computation can be simplified, on the basis that suppressing all except one response is equivalent to suppressing all responses and then adding one:

$$y_{k,i}(n) = y(n)\sum_{k'=0}^{k-1} \hat{x}_i(n - m_{k'}) + \hat{x}_i(n - m_k) = r_i(n) + \hat{x}_i(n - m_k) \quad (7)$$

where $r_n(n) = y(n) - s(n) * \hat{x}_i(n)$ represents the residual of the EEG, i.e. the recorded EEG minus the EEG expected from the estimated response $\hat{x}_i(n)$ and the stimulation sequence $s(n)$. With this definition, the IRSA can be reformulated as:

$$\hat{x}_{i+1}(j) = \frac{1}{K}\sum_{k=0}^{K-1} r_i(j + m_k) + \frac{1}{K}\sum_{k=0}^{K-1} \hat{x}_i(n - m_k + m_k) + \frac{1}{K}\sum_{k=0}^{K-1} r_i(j + m_k) + \hat{x}_i(j) \quad (8)$$

or, if $z_i(j)$ is defined as the averaged residual:

$$z_i(j) = \frac{1}{K}\sum_{k=0}^{K-1} r_i(j + m_k) \quad (9)$$

the iterative estimation of the response can be written as:

$$\hat{x}_{i+1}(j) = z_i(j) + \hat{x}_i(j) \quad (10)$$

[0027] Even though this procedure usually converges to a stable solution, it has been found that the ADJAR-Level 2 and IRSA algorithms are sometimes unstable, depending on the distribution of the ISI in the stimulation sequence, and in this case the solution tends to oscillate (Woldorff, 1993 - Ref 20; Valderrama et al., 2014a - Ref 14). The risk of oscillation particularly increases for narrow ISI distributions. Including a convergence control parameter (α, in the range [0,1]) was found to be a simple solution in order to avoid this instability:

$$\hat{x}_{i+1}(j) = \hat{x}_i(j) + \alpha \cdot z_i(j) \quad (11)$$

[0028] According to Valderrama et al. (2014a - Ref 14), a small enough α guarantees convergence and stability of the algorithm, but requires more iterations to reach convergence.

[0029] The limitations of ADJAR-Level 2 are that many authors have reported this method to be difficult to implement, and that it does not include the convergence parameter α, which may result in uncontrolled instability issues.

[0030] The shared limitation of IRSA and ADJAR-Level 2 is their high computational load, because every iteration requires computations involving the whole EEG [equation 11]. The computational complexity increases linearly with the number of iterations, and is also influenced by the EEG length (N) and the product of response length times the number of stimuli ($J \times K$). Therefore, an evoked potential recording procedure including a large number of stimuli (and therefore a long EEG) and requiring a large number of iterations in IRSA, implies a prohibitive computational complexity that conventional IRSA difficult to be applied or unpractical in most clinical and research applications.

[0031] Taking into account the previous derivations, the IRSA algorithm / technique can be summarized as follows:

1. Initialization:

$$\hat{x}_0(j) = 0 \qquad x_0(j) = \frac{1}{K}\sum_{k=0}^{K-1} y(j + m_k) \qquad \forall j = 0, \ldots, J-1 \quad (12)$$

2. Response updating

$$\hat{x}_i(j) = \hat{x}_{i-1}(j) + \alpha \cdot z_{i-1}(j) \quad \forall j = 0, \ldots, J-1 \quad (13)$$

3. Residual estimation

$$r_i(n) = y(n) - \sum_{k=n}^{K-1} \hat{s}_i(n - m_k) \quad \forall n = 0, \ldots, N-1 \tag{14}$$

4. Averaged-residual estimation:

$$z_i(j) = \frac{1}{K} \sum_{k=0}^{K-1} r_i(j + m_k) \quad \forall j = 0, \ldots, J-1 \tag{15}$$

5. Steps 2 to 4 are repeated until convergence.

[0032] The energy of the averaged residual tends to decrease with the iterations, and different convergence criteria can be applied (for example, a minimum reduction of the averaged residual energy, or a relative reduction of the averaged residual energy with respect to that of the previous iteration). In this the present case a predefined number of iterations was used.

[0033] It can be noted that $z_0(j)$ in IRSA corresponds to the estimation provided by RSA. The computational complexity increases linearly with the number of iterations, and is also influenced by the EEG length (N) because of equation (14) and the product of the response length times the number of stimuli ($J \times K$) because of equations (14) and (15). The computational cost associated to equation (13) is negligible compared to that of the other equations. Therefore, an evoked potential recording procedure including a large number of stimuli (and therefore a long EEG) and requiring a large number of iterations in IRSA, implies a prohibitive computational complexity that makes conventional IRSA difficult to apply or unpractical in most applications.

[0034] Documentation detailing aspects of the IRSA technique include: Valderrama, et al., (2016 - Ref 17); Valderrama, et al., (2014a - Ref 15); Valderrama, et al., (2014b - Ref 16); Valderrama, et al., (2012 - Ref 14).

Comprehensive representation of AEPs

Cortical ERPs and brainstem FFRs recorded in the same stimulus sequence

[0035] Bidelman (2015 - Ref 3) proposed a stimulus paradigm for concurrent recording of the auditory brainstem frequency following response (FFR) and cortical event-related potentials (ERPs). This stimulus uses a clustered stimulus presentation and variable ISI, as presented in Figure 4.

[0036] The limitations of this technique are that (1) the steady-state response of the early components of the auditory pathway do not provide information of their neural generators [i.e., the neural activity of the cochlea, brainstem and midbrain cannot be determined]; and (2) early, middle and late components of the auditory pathway are not shown in a single plot.

Simultaneously-evoked auditory potentials (SEAP) with FFRs and CAEPs

[0037] Slugocki et al. (2017 - Ref 13) also combined the use of FFRs and CAEPs to simultaneously measure cortical and subcortical auditory-evoked activity. The SEAP stimulus consists of a pure-tone carrier of 500 Hz that has been amplitude-modulated at the sum of 37 and 81 Hz (depth 100%). The authors show that SEAP elicits a 500 Hz FFR (showing activity of the inferior colliculus); a 80 Hz FFR (subcortical activity); a 40 Hz FFR (primary auditory cortex); mismatch negativity (MMN) and P3a and the N1-P2 complex (secondary auditory cortex).

[0038] Similar to Bidelman (2015 - Ref 3), one important limitation of this method is that FFRs do not provide information of their neural generators, and therefore, the activity of the subcortical neural stations cannot be determined. For example, they claim that the 500 Hz FFR shows activity from the inferior colliculus, but, similar to the 40 Hz ASSR, it is very likely that the 500 Hz ASSR has multiple generators (not only activity from the midbrain), and that the steady-state signal is the result of the sum of different components from different overlapping responses.

Simultaneous recording of brainstem, middle and late responses using deconvolution

[0039] Holt and Ozdamar (2014 - Ref 10, 2016 - Ref 9) used the CLAD deconvolution method to record the impulsive response of the auditory pathway at increasing stimulus rates. Figure 5 shows grand-averages at different stimulus rates in which the ABR wave V; the MLR components N_a, P_a, N_b; and the CAEP P_1-N_1-P_2 complex can be identified.

[0040] The main limitation of this approach is that representing the deconvolved signals in the linear time domain prevents the early components to be correctly analysed. For example, the ABR wave V can be observed in all traces, but this type of presentation does not provide a good resolution for the remaining ABR components.

Logarithmic representation of AEPs

[0041] Michelini et al. (1982 - Ref 12) proposed a non-linear samples reduction of the digitized response and its representation in the logarithmic time-axis to efficiently display all components of the ascending auditory pathway in a single plot. Figure 6 shows an example of an AEP after sample reduction and representation in the logarithmic time-scale.

[0042] The main drawback of this method is that the proposed non-linear sample reduction process is done arbitrarily without any supporting physiological model, and that this process

does not implement a latency-dependent filtering like the one proposed in the present invention.

Comprehensive recording of AEPs by projecting over a base of functions.

[0043] Valderrama et al. (2017 - Ref 18) presented an abstract in the IERASG conference (Warsaw, May 2017) projecting the averaged response over a base of sinc functions uniformly distributed in the logarithmic time scale and then reconstructing the signal from the projected space provides a latency-dependent filtering appropriate to represent the activity of the main stations of the auditory pathway from the cochlea to the auditory cortex in a single plot.

Least-squares deconvolution

[0044] Bardy et al. (2014b - Ref 2) used the least-squares deconvolution algorithm to deconvolve overlapping CAEPs evoked by different stimuli. Results show that multi-response deconvolution is successfully achieved with this algorithm. As mentioned earlier, the search of efficient stimulus sequences can be arduous and may constrain the flexibility of some experiments.

Split-IRSA

[0045] Valderrama et al. (2016 - Ref 17) updated the IRSA formulation to allow multi-response deconvolution in ABR and MLR signals. This was used to prove that (1) wide jittered stimulation sequences could incur in a violation of the time invariant assumption; and that (2) the neurons of the auditory pathway are not only influenced by short-term adaptation (the influence that the previous stimulus has on each response), but also by long-term adaptation, i.e. the morphology of each response is also influenced by the overall stimulus rate of several previous stimuli.

[0046] Similar to IRSA, the limitation of Split-IRSA is the high computational load, as each iteration involves operations processing the whole EEG, which constrains the applicability of this method.

Objective

[0047] Despite the advances in deconvolution technologies are exemplified above it remains desirable to provide a further alternative or improved method of deconvolving AEP responses which may allow for one or more of:

deconvolution of overlapping AEP responses;

deconvolution of overlapping responses which present different morphologies

representation of the main neural components from the cochlea to the auditory cortex in a single plot;

improved computational efficiency when compared with existing deconvolution methodologies; and

providing a useful alternative to existing deconvolution methodologies.

[0048] The reference in this specification to any prior publication, or information derived from it, or to any matter which is known, is not, and should not be taken as an acknowledgement or admission or any form of suggestion that the prior publication, or information derived from it, or known matter forms part of the common general knowledge in the field of endeavour to which this specification relates.

SUMMARY

[0049] According to a first aspect of the invention, there is provided a method of estimating the transient auditory evoked potential ('AEP') responses of a subject, the method comprising:

generating a digital auditory stimulus signal consisting of at least one auditory stimulus type;

presenting the at least one auditory stimulus type to a subject via a transducer;

recording an electroencephalogram signal ('EEG') including the neural response of the subject to the at least one auditory stimulus type;

synchronizing the digital auditory stimulus signal with the recorded EEG; and

deconvolving the overlapping AEP responses of the subject from the EEG by applying an iterative randomized stimulation and averaging ('IRSA') technique,

wherein the step of applying an IRSA technique is performed with matrix operations in the representation spaces of the AEP and the EEG.

[0050] Optionally, the IRSA technique comprises the steps of: (a) initialisation, (b) response updating, and (c) averaged-residual estimation in which the steps of (b) response updating and (c) averaged-residual estimation are repeated until convergence and wherein steps (a)-(c) are performed using matrix operations.

[0051] Optionally, the least one auditory stimulus type includes a stimulus type having a jittered inter-stimulus interval less than the duration of the resulting auditory evoked potential to be detected.

[0052] Optionally, the at least one auditory stimulus type is selected from the group consisting of:

standard auditory stimuli such clicks and tone-bursts; and

complex auditory stimuli like multi-pattern stimuli, speech-like stimuli, or natural speech stimuli.

[0053] Optionally, the method comprises applying more than one auditory stimulus type, such that the different stimulus types evoke different AEP responses.

[0054] Optionally, the step of applying the IRSA technique comprises performing iterative matrix operations in segments limited to the duration of the AEP, rather than the duration of the EEG (that is, performing matrix operations in the representation space of the AEPs rather than in the representation space of the EEG).

[0055] Optionally, the step of applying the IRSA technique comprises configuring the matrix operations according to the symmetric-Toeplitz properties of generated matrices to thereby reduce the computational effort required to deconvolve the AEP responses.

[0056] Optionally, the step of applying the IRSA technique comprises calculating the matrix product used when implementing the iterations of the IRSA technique as a convolution.

[0057] Optionally, the method further comprises the step of calculating the autocorrelation of the digital auditory stimulus signal either as a cross-correlation or as a sum for all stimuli of the digital auditory stimulus signal.

[0058] Optionally, the step of applying the IRSA technique comprises calculating an averaged residual as either the normalised cross-correlation of the EEG and the digital auditory stimulus signal or as a sum for all stimuli of the digital auditory stimulus signal.

[0059] Optionally, the method comprises:

applying more than one auditory stimulus type, such that the different stimulus types evoke different AEP types; and

adapting the IRSA technique to deconvolve more than one AEP type ('multi-response deconvolution') in its matrix formulation.

[0060] Optionally, the method comprises performing an orthonormal transformation of the representation space, and performing IRSA operations in the transformed representation space.

[0061] Optionally, the step of applying an orthonormal transformation results in a transformed representation space of reduced dimensions.

[0062] Optionally, IRSA operations are performed in the transformed representation space derived from a matrix performing any one of the following steps: low-pass filtering; band-pass filtering; decimation; latency dependent filtering; or latency dependent decimation.

[0063] Optionally, IRSA operations are performed in the reduced representation space derived from an orthonormal matrix performing latency dependent filtering and latency dependent decimation.

[0064] Optionally, the method is used to estimate AEP responses to complex auditory stimuli, including multi-pattern stimuli, speech-like stimuli or natural speech stimuli, either in a single-response or multi-response approach.

[0065] Optionally, the method is used to estimate one or more of auditory brainstem responses, middle latency responses, or cortical auditory evoked potentials to complex auditory stimuli, including multi-pattern stimuli, speech-like stimuli or natural speech stimuli, either in a single-response or multi-response approach.

[0066] Optionally, the method is used to simultaneously estimate auditory brainstem responses, middle latency responses, and cortical auditory evoked responses to complex auditory stimuli, such as multi-pattern stimuli, speech-like stimuli or natural speech stimuli, either in a single-response or multi-response approach.

[0067] Optionally, the method is used to estimate one or more of auditory brainstem responses, middle latency responses, or cortical auditory evoked potentials, either in a single-response or multi-response approach.

[0068] Optionally, the method is used to simultaneously estimate auditory brainstem responses, middle latency responses, and cortical auditory evoked responses, either in a single-response or multi-response approach.

[0069] Optionally, the method further comprising graphically representing the estimated AEP.

[0070] Optionally, the method is performed at least in part on a computer.

[0071] According to a second aspect of the invention, there is provided a system configured to estimate the auditory evoked potential responses of a subject by implementing a method according to the first aspect of the invention, the system comprising:

a data processor;

a memory in data communication with the data processor;

wherein the system is configured to implement a method according to a first aspect of the invention.

[0072] According to a third aspect of the invention, there is provided a computer program comprising instructions to make a computer carry out a method according to a first aspect of the invention.

[0073] According to a fourth aspect of the invention, there is provided a computer-readable storage medium comprising program instructions capable of making a computer carry out a method according to a first aspect of the invention.

[0074] According to a fifth aspect of the invention, there is provided a transmissible signal comprising program instructions capable of making a computer carry out a method according to a first aspect of the invention.

[0075] Throughout this specification and the claims which follow, unless the context requires otherwise, the word “comprise” and variations thereof such as “comprises” and “comprising”, will be understood to include the inclusion of a stated integer or step or group of integers or steps but not the exclusion of any other integer or step or groups of integers or steps.

BRIEF DESCRIPTION OF THE FIGURES

[0076] FIGURE 1 shows an example of transient ABR, MLR and CAEP signals

[0077] FIGURE 2 shows a diagram (not a real response) demonstrating the main neural components of the ascending auditory pathway from the cochlea (ABR wave I) to the auditory cortex (P1-N1-P2 complex).

[0078] FIGURE 3 shows resulting ASSRs from a synthesized experiment at different ISIs. The 40 Hz-ASSR produces a constructive interference that facilitates neural response detection.

[0079] FIGURE 4 shows a schematic illustration of the clustered stimulus paradigm for simultaneously recording auditory brainstem FFRs and cortical ERPs.

[0080] FIGURE 5 show grand-average AEP signals at different stimulus rates.

[0081] FIGURE 6 shows AEPs after sample reduction and representation in the logarithmic time-scale.

[0082] FIGURE 7 shows functions of the basis with $K_{dec} = 10$ samples/decade, before orthonormalization, using $f_s = 25$ kHz. The plots in the left include all the functions in the base.

The plots in the right include a detail of three functions. The time axis is represented in linear scale in the top plots, and in logarithmic scale in the bottom plots.

[0083] FIGURE 8 shows functions of the basis with K_{dec} = 10 samples/decade, after orthonormalization, using f_s = 25 kHz. The plots in the left include all the functions in the base. The plots in the right include a detail of three functions. The time axis is represented in linear scale in the top plots, and in logarithmic scale in the bottom plots.

[0084] Figure 9 shows signal $\mathbf{x}_n$ (in blue) generated with a basis using K_{dec} = 15 samples/decade and contaminated with AGWN, and the latency dependent low-pass filtered signal $V'_r V_r \mathbf{x}$ (in red). From top to bottom, the whole signal (interval [0 ms - 400 ms]), and detail for different time intervals: [0.4 ms - 4 ms], [4 ms - 40 ms] and [40 ms - 400 ms].

[0085] Figure 10 shows synthesized signal generated with a basis using K_{dec} =15 samples/decade, and signals resulting from projecting it with $V_r^T V_r$ using more than 15 samples/decade in the transformation for dimensionality reduction. The SNR associated to the difference between the reference signal (with K_{dec} = 15 samples/decade) and the projected signals was 39.20 dB, 47.34 dB, 51.09 dB and 55.07 dB for K_{dec} = 20, 30, 40 and 50 samples/decade respectively.

[0086] FIGURE 11 shows portions of clean and noisy synthesized EEGs. In red, the stimulation signal; in blue the clean EEG (left panel) and the noisy EEG (right panel).

[0087] Figure 12 shows the original response and responses provided by IRSA after 50 iterations. In the left panel, the results of the conventional IRSA and matrix IRSA. In the right panel, the responses provided by matrix IRSA projected with $\mathrm{V_r^T V_r}$ and those provided by matrix IRSA performed in the reduced representation space.

[0088] Figure 13 shows the original response and responses provided by IRSA after 10000 iterations. In the left panel, the response provided by matrix IRSA projected with $\mathrm{V_r^T V_r}$ and those provided by matrix IRSA performed in the reduced representation space. In the right panel, difference between both results.

[0089] FIGURE 14 shows AEP responses estimated from real EEGs - comparison of responses estimated with 50 iterations. Left panel: comparison of conventional-IRSA and matrix-IRSA. Right panel: comparison of matrix-IRSA after projection with $\mathrm{V_r^T V_r}$ and matrix-IRSA performed in the reduced representation space.

[0090] FIGURE 15 shows AEP responses estimated from real EEGs. Comparison of responses estimated with 10000 iterations using matrix-IRSA and projection and matrix-IRSA-red. Right panel shows the difference between both results.

[0091] FIGURE 16 shows AEP responses estimated from real EEGs. Comparisons of the responses estimated with different resolutions in the dimensionality reduction. The responses for $K_{dec} = 70$ samples/decade were used as reference in the comparison presented in Table 8.

[0092] FIGURE 17 shows a comparison of responses estimated with 10000 iterations using real EEGs. Left panel: matrix-IRSA estimated responses using the standard and fast versions. Right panel: matrix-IRSA responses estimated in the reduced representation space using the standard and fast versions.

[0093] FIGURE 18 shows the difference between the responses estimated with the standard and the fast versions of matrix-IRSA. Left panel: for estimations in the complete representation space. Right panel: for estimations in the reduced representation space.

DETAILED DESCRIPTION

[0094] In broad terms, the invention relates to a method of estimating the auditory evoked potential responses of a subject by deconvolving overlapping AEP responses applying an IRSA technique, , and an apparatus configured to deconvolve an overlapping AEP responses applying an IRSA technique. The method of deconvolving an overlapping AEP responses may provide for reduced computation when compared to similar existing technologies. Within the broader concepts, embodiments of the above methods and apparatuses are described and defined below.

Matrix representation of the IRSA procedure

[0095] The signals involved in the convolutional model of the EEG and the equation (1) can be represented using a matrix notation:

$$\begin{pmatrix} y(0) \\ y(1) \\ y(2) \\ \vdots \\ y(N-1) \end{pmatrix} = \begin{pmatrix} s(0) & 0 & 0 & \cdots & 0 \\ s(1) & s(0) & 0 & \cdots & 0 \\ s(2) & s(1) & s(0) & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ s(J-2) & s(J-3) & s(J-4) & \cdots & 0 \\ s(J-1) & s(J-2) & s(J-3) & \cdots & s(0) \\ s(J) & s(J-1) & s(J-2) & \cdots & s(1) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ s(N-J) & s(N-J-1) & s(N-J-2) & \cdots & \vdots \\ 0 & s(N-J) & s(N-J-1) & \cdots & \vdots \\ 0 & 0 & s(N-J) & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & s(N-J) \end{pmatrix} \begin{pmatrix} x(0) \\ x(1) \\ \vdots \\ x(J-1) \end{pmatrix} + \begin{pmatrix} n_0(0) \\ n_0(1) \\ n_0(2) \\ \vdots \\ n_0(N-1) \end{pmatrix} \tag{16}$$

which can written in a compact form as:

$$\mathbf{Y} = S\mathbf{x} + \mathbf{n}_0. \tag{17}$$

where **y**, **n** and *S***x** are *N*-component column vectors, **x** is a *J*-component column vector and *S* is a matrix with *N* rows and *J* columns (an *N*×*J* matrix). It should be noted that $\mathrm{s(n)}$ is null for all the samples except for those corresponding to a stimulation event (at samples $\mathrm{m_k}$), and therefore, most of the elements in the stimulation matrix *S* are null.

[0096] Similarly, equation (4) can be rewritten in matrix notation as:

$$\hat{\mathbf{x}} = \frac{1}{K} S^T \mathbf{y} = S_K \mathbf{y} \qquad S_K \equiv \frac{1}{K} S^T \tag{18}$$

where S^T is the transposed of matrix S, and S_K is defined from S including transposition and normalization. The last equation provides the RSA solution in matrix notation. With these definitions, IRSA can easily be formulated with matrix notation:

1, $\hat{\mathrm{x}}_0 = 0 \quad \mathrm{z}_0 = \mathrm{S_K y}$ (initialisation) (20)

2. $\hat{\mathrm{x}}_\mathrm{l} = \hat{\mathrm{x}}_{\mathrm{l}-1} + \alpha \mathrm{z}_{\mathrm{l}-1}$ (updated response estimation) (21)

3. $\mathrm{r_i} = \mathrm{y} - \mathrm{S}\hat{\mathrm{x}}_\mathrm{i}$ (residual estimation) (22)

4. $\mathrm{z_l} = \mathrm{S_K r_i}$ (average-residual estimation) (23)

(steps 2 to 4 are repeated until convergence)

[0097] Using the matrix representation, steps 3 and 4 can be compacted into one step:

$$\mathrm{z_i} = \mathrm{S_K r_i} = \mathrm{S_K y} = \mathrm{z_o} - \mathrm{S_K S}\hat{\mathrm{x}}_\mathrm{i} \tag{23}$$

where R_S is a J × J square matrix resulting of the product of matrices S_K and S. The matrix R_S can also be obtained as the normalized autocorrelation matrix of the stimulation sequence s(n):

$$R_s = \frac{1}{K}\begin{pmatrix} r_s(0) & r_s(1) & r_s(2) & \cdots & r_s(J-1) \\ r_s(1) & r_s(0) & r_s(1) & \cdots & r_s(J-2) \\ r_s(2) & r_s(1) & r_s(0) & \cdots & r_s(J-3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_s(J-1) & r_s(J-2) & r_s(J-3) & \cdots & r_s(0) \end{pmatrix} \quad (24)$$

where $R_S(j)$ is the autocorrelation function of $s(n)$, and can be estimated as:

$$r_s(j) = \sum_{n=j}^{N-1} s(n)s(n-j) \qquad \forall j_1, j_2 \in \{0, \dots, J-1\} \quad (25)$$

And therefore R_s can be estimated as:

$$r_s(j_i, j_2) = \frac{1}{k} r_s(|j_1 - j_2|) \qquad \forall j = 0, \dots, J-1 \quad (26)$$

Matrix implementation of IRSA algorithm

[0098] The proposed combination of steps 3 and 4 using equation (23) provides an important reduction of the computational complexity of IRSA. The conventional IRSA requires an operation involving all the EEG and all the stimuli in order to calculate $r_i(n)$, and then an operation involving all the signal $r_i(n)$, (with the same duration of the EEG) and all the stimuli, in order to calculate $z_i(j)$. In contrast, the matrix based formulation of IRSA do not require calculating $r_i(n)$ since $z_i(j)$ is directly estimated from $z_0(j)$, $r_s(j)$ and $\hat{x}_i(j)$. Additionally, the estimation of $z_i(j)$ only involves the matrix operation $R_s\hat{\mathbf{x}}_i$ with a $J \times J$ matrix, and a summation of vectors with J elements, where the vector z_0 and the matrix R_s are used at all the iterations and can therefore be computed just once at the beginning of the algorithm.

[0099] With these considerations in mind, the matrix implementation of the IRSA algorithm is the following:

1. Initialisation

$$\hat{x}_0(j) = 0 \quad z_0(j) = \frac{1}{K}\sum_{k=j}^{K-1} y(j + m_k) \quad r_s(j) = \sum_{n=j}^{N-J} s(n)s(n-j) \quad \{\hat{\mathbf{x}}_0, \mathbf{z}_0, R_s\} \quad (27)$$

2. Response updating:

$$\hat{\mathbf{x}}_i = \hat{\mathbf{x}}_{i-1} + \alpha \mathbf{z}_{i-1} \quad (28)$$

3. Averaged-residual estimation

$$\mathbf{z}_i = \mathbf{z}_0 + R_s\hat{\mathbf{x}}_i \quad (29)$$

4. Steps 2 and 3 are merged until convergence.

[0100] According to the previous formulation, the matrix implementation of the IRSA algorithm requires an initialization (where two J-dimension vectors, $\hat{\mathbf{x}}_0$ and $\mathbf{z}_0$, and a $J \times J$ matrix R_s are estimated) and an iterative procedure, involving matrix and vector operations in a J-dimensional space. It should be noted that the matrix implementation of IRSA is mathematically equivalent to the conventional IRSA. However, the computational complexity is substantially smaller, because there are computations involving the whole EEG only at initialization (i.e. estimation of $\mathbf{z}_0$ and R_s), and the computations at the iterations just involve matrix or vector operations with dimensionality J. In other words, the computational complexity of the iterations depends on the length of the response (J) but not on the length of the EEG (N) nor the number of stimuli (K), and this provides a very efficient implementation of the IRSA algorithm even for experiments with a large number of stimuli or long EEGs.

Dimensionality reduction

Matrix formulation of IRSA in a transformed representation space

[0101] If V is an orthonormal transformation of the J-dimension representation space (i.e. V is a $J \times J$ square matrix whose rows are the components of J orthonormal functions or orthonormal vectors), the IRSA algorithm can equivalently be the IRSA algorithm can equivalently be implemented in the original or in the transformed representation space:

$$\hat{\mathbf{x}}_0^v = V\hat{\mathbf{x}}_0 = 0 \qquad \mathbf{z}_0^v = V\mathbf{z}_0 \qquad R_s^v = VR_sV^T \quad (30)$$

$$\hat{\mathbf{x}}_i^v = \hat{\mathbf{x}}_{i-1}^v + \alpha\hat{\mathbf{z}}_{i-1} \quad (31)$$

$$z_i^v = z_0^v + R_s^v\hat{x}_i^v \quad (32)$$

and the recovered evoked response after convergence can be transformed back to the original representation space by applying the inverse matrix of the orthonormal transformation:

$$\hat{x}_i = V^T\hat{x}_i^v \quad (33)$$

[0102] Since the orthonormal matrix verifies that $\mathrm{V^TV = I}$, it is easy to demonstrate that the IRSA algorithm provides identical results when it is implemented either in the original or in the transformed representation spaces:

$$V^T\mathbf{z}_i^v = V^T\mathbf{z}_0^v - V^T R_s^v\hat{\mathbf{x}}_i^v = V^TV\mathbf{z}_0 - V^TVR_sV^TV\hat{\mathbf{x}}_i = I\mathbf{z}_0 - R_sI\hat{\mathbf{x}}_i = \mathbf{z}_0 - R_s\hat{\mathbf{x}}_i = \mathbf{z}_i \quad (34)$$

Responses contained in a subspace: dimensionality reduction

[0103] If the orthonormal transformation V is selected in a way that some components are expected to be null when it is used for representing an evoked response, then a reduced transformation matrix V_r can be defined by excluding those rows corresponding to components with null amplitude. In this case, if the dimensionality of V_r is $J_r \times J$, with $J_r < J$, the application of V_r to a vector (or function, or signal) reduces its dimensionality from J to J_r components. A typical example of this dimensionality reduction can be found in low-pass filtering in the frequency domain. In this example, the orthonormal transformation would be the fast Fourier transform (FFT), the inverse transformation would be the inverse-FFT (iFFT), and the low-pass filtering in the frequency domain could be equivalently implemented by canceling those components with frequency above the cut-off frequency in the frequency domain, or by truncating the orthonormal transformation and applying the truncated FFT to the signal (which provides only the components expected to be non-null in the frequency domain) and then applying the truncated iFFT to this reduced representation domain (which provides the filtered signal in the time domain).

[0104] A substantial difference between the complete V and the reduced V_r orthonormal transformations is that V is invertible ($V^{-1} = V^T$), but not V_r. In fact, the product $V_r^TV_r$ (which is a $J \times J$ matrix) applied to a vector (or function, or signal) is a projector that provides a vector that is an element of the original J-dimension representation space but is contained in a subspace with dimension J_r. In other words, $V_r^TV_r \neq I$, or equivalently, in general $\mathbf{x} = V_r^TV_r\mathbf{x}$ (while the original and the recovered vectors are equal if V is invertible).

[0105] The interesting advantage of the reduced transformation V_r is that, if a vector $\mathbf{x}$ (or function, or signal) belongs to the J_r-dimension reduced subspace (for example, if $\mathbf{x}$ is a signal with components only below the cut-off frequency), even though $V_r^TV_r \neq I$, the equivalence $\mathbf{x} = V_r^TV_r\mathbf{x}$ is verified (in the example of the low-pass filtering, if $\mathbf{x}$ only contains low frequency components and the effect of applying $V_r^TV_r$ consists in a removal of high frequency components, $V_r^TV_r\mathbf{x}$ and $\mathbf{x}$ are identical because $\mathbf{x}$ contains no high frequency components to be removed).

Matrix IRSA algorithm in a reduced representation space

[0106] Let's suppose the matrix IRSA formulation described with equations (27), (28), (29) assuming evoked responses that can be represented in a reduced representation space, i.e. evoked responses verifying that $\mathbf{x} = V_r^TV_r\mathbf{x}$ using a reduced orthogonal transformation V_r with

$J_r < J$. The evoked response in the reduced representation space $x_i^{v_r}$ can be used to recover the evoked response:

$$V_r^T \hat{\mathbf{x}}_i^{v_r} = V_r^T V_r \hat{\mathbf{x}}_i = \hat{\mathbf{x}}_i \tag{35}$$

and therefore, the IRSA algorithm can be formulated in the reduced representation space. According to equation (28),

the evoked response at iteration i in the reduced representation space is obtained as:

$$\hat{\mathbf{x}}_i^{v_r} = V_r \hat{\mathbf{x}}_{i-1} + \alpha V_r \mathbf{z}_{i-1} = \hat{\mathbf{x}}_{i-1}^{v_r} + \alpha \mathbf{z}_{i-1}^{v_r} \tag{36}$$

and the averaged residual in the reduced representation space is, at iteration i:

$$\mathbf{z}_i^{v_r} = V_r \mathbf{z}_i = V_r \mathbf{z}_0 - V_r R_s \hat{\mathbf{x}}_i \tag{37}$$

[0107] Taking into account that $\hat{\mathbf{x}}_i = V_r^T V_r \hat{\mathbf{x}}_i$ (because $\hat{\mathbf{x}}_i$ is assumed to be appropriately represented in the reduced representation space), the averaged residual in the reduced representation space can be rewritten as:

$$\mathbf{z}_i^{v_r} = V_r \mathbf{z}_0 - V_r R_s V_r^T V_r \hat{\mathbf{x}}_i \tag{38}$$

and if we define the normalized correlation matrix in the reduced representation space as:

$$R_s^{v_r} = V_r R_s V_r^T \tag{39}$$

(which is a $Jr \times Jr$ matrix), the averaged residual in the reduced representation space can be estimated as:

$$\mathbf{z}_i^{v_r} = \mathbf{z}_0^{v_r} - R_s^{v_r} \hat{\mathbf{x}}_i^{v_r} \tag{40}$$

[0108] In summary, the matrix IRSA procedure can be formulated in the reduced representation space according to the following algorithm:

1. Initialization:

$$\hat{x}_0(j) = 0 \qquad z_0(j) = \frac{1}{K} \sum_{k=0}^{K-1} y(j + m_k) \qquad r_s(j) = \sum_{n=j}^{N-J} s(n) s(n-j) \qquad \{\mathbf{x}_0, \mathbf{z}_0, R_s\} \tag{41}$$

$$\mathbf{x}_0^{v_r} = V_r \mathbf{x}_0 = 0 \qquad \mathbf{z}_0^{v_r} = V_r \mathbf{z}_0 \qquad R_s^{v_r} = V_r R_s V_r^T \qquad \{\hat{\mathbf{x}}_0^{v_r}, \mathbf{z}_0^{v_r}, R_s^{v_r}\} \tag{42}$$

2. Response updating:

$$\hat{\mathbf{x}}_i^{v_r} = \hat{\mathbf{x}}_{i-1}^{v_r} + \alpha \mathbf{z}_{i-1}^{v_r} \tag{43}$$

3. Averaged-residual estimation:

$$\mathbf{z}_i^{(r)} = \mathbf{z}_0^{(r)} - R_s^{(r)} \mathbf{x}_i^{(r)} \tag{44}$$

4. Steps 2 and 3 are repeated until convergence.

5. The recovered evoked response after convergence is transformed back to the original representation space:

$$\hat{\mathbf{x}}_i = V_r^T \hat{\mathbf{x}}_i^{(r)} \tag{45}$$

[0109] There are two important differences between both matrix implementations of IRSA. On the one hand, the solutions provided by both algorithms are different because in the last algorithm the recovered evoked response is forced to be contained in the subspace of reduced dimensionality. For example, if dimensionality reduction implied a low-pass filtering, the implementation in the reduced representation space would provide a low-pass filtered solution, while the other implementation would provide a non-filtered solution (that would contain, for example, high frequency components associated to noise). On the second hand, the steps involved in the iterative process (steps 2 and 3) require matrix operations with R_v^{rs}, which is a $J_r \times J_r$ matrix, and with vectors of dimension J_r. If the dimensionality of the reduced representation space I significantly smaller than that of the original representation space (i.e. if $J_r \ll J$) the computation involved in the algorithm decreases significantly.

Matrix IRSA constrained to a subspace

[0110] In the previous derivation, the evoked response was assumed to belong to the reduced subspace (i.e. $\mathbf{x} = V_r^T V_r \mathbf{x}$). A different situation could be considered if the matrix IRSA algorithm is applied in a subspace defined by a truncated orthonormal transformation V_1 not verifying the previous assumption (i.e. $\mathbf{x} \neq V_1^T V_1 \mathbf{x}$). In that case, the step from equation (37) to equation (38) is not valid. The resolution of the IRSA algorithm constrained to the subspace defined by V_1 can be obtained by estimating $\hat{\mathbf{x}}_i$ in the complete representation space and then projecting it over the subspace ($V_1^T V_1 \hat{\mathbf{x}}_i$). However, this algorithm would require operations in a J-dimension representation space (not in a J_1- dimension subspace). A relevant question is whether the resolution in the subspace is equivalent or not to the resolution in the complete representation space followed by projection into the subspace, because if they are equivalent, a lot of computation can be saved even when the condition $\mathbf{x} = V_1^T V_1 \mathbf{x}$ is not verified. However, if they are not equivalent, the implementation in the reduced representation space requires the condition $\mathbf{x} \neq V_1^T V_1 \mathbf{x}$. Both procedures would be equivalent if $V_1 \mathbf{z}_i$ can be obtained in the reduced representation space, or if the following condition is verified:

$$V_1\mathbf{z}_0 - V_1R_s\hat{\mathbf{x}}_i = \mathbf{z}_0^{v_1} - R_s^{v_1}\hat{\mathbf{x}}_i^{v_1} \tag{46}$$

or equivalently, if:

$$V_1R_s\hat{\mathbf{x}}_i = R_s^{v_1}\hat{\mathbf{x}}_i^{v_1} \tag{47}$$

[0111] In order to analyze this condition, the dimensionality reduction V_1 will be represented as a truncated orthonormal matrix, i.e. a matrix containing J_1 orthonormal vectors (or functions), with $J_1 < J$, representing the J_1 components of the reduced representation space. This way, the complete orthonormal matrix V can be decomposed into two matrices (the projector over the reduced space V_1 and the projector over its orthogonal complement V_2):

$$V = \begin{pmatrix} \mathbf{v}_0^T \\ \mathbf{v}_1^T \\ \vdots \\ \mathbf{v}_{J_1-1}^T \\ \mathbf{v}_{J_1}^T \\ \vdots \\ \mathbf{v}_{J-1}^T \end{pmatrix} = \begin{pmatrix} \mathbf{v}_0^T \\ \mathbf{v}_1^T \\ \vdots \\ \mathbf{v}_{J_1-1}^T \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \mathbf{v}_{J_1}^T \\ \vdots \\ \mathbf{v}_{J-1}^T \end{pmatrix} = V_1 + V_2 \tag{48}$$

Verifying:

$$V^TV = I \qquad V_1^TV_2 = 0 \qquad V_2^TV_1 = 0 \tag{49}$$

With this decomposition of the orthogonal transformation, $VR_s\,\hat{x}_i$ can be expanded as:

$$VR_s\mathbf{x}_i = VR_sV^TV\mathbf{x}_i = (V_1 + V_2)R_s(V_1^T + V_2^T)(V_1 + V_2)\hat{\mathbf{x}}_i \tag{50}$$

where two terms were omitted because of equation (49). Finally, the term $V_1R_s\,\hat{x}_i$ can be written as:

$$V_1R_s\hat{\mathbf{x}}_i = R_s^{v_1}\mathbf{x}_i^{v_1} + V_1R_sV_2^T\hat{\mathbf{x}}_i^{v_2} \tag{52}$$

and comparing this decomposition with equation (47), it is clear that the condition is verified if the last term is null, i.e., if either $V_1R_sV_2^T$ is null or if $\hat{x}_i^{V_2}$ i is null. In general, $V_1R_sV_2^T$ is never null for an IRSA problem (because it requires that the autocorrelation function $r_s(j)$ only contains a non-null value at $j = 0$, and in this case, the IRSA algorithm makes non-sense because RSA is appropriate). The second option ($\hat{x}_i^{V_2} = V_2\hat{x}_i$ null) requires that the projection of the response over the orthogonal complement is null, which is equivalent to the condition $x = V_1^TV_1\,x$.

[0112] Therefore, in order to perform the matrix IRSA algorithm in a reduced representation space given by V_1, the condition $x = V_1^T V_1 x$ (or equivalently $V_2 x = 0$) is required. Otherwise, the term $V_1 R_s V_2^T \hat{x}_i^{V_2}$ would introduce a bias in the estimated solution.

[0113] In general, the dimensionality reduction is associated to the cancellation of some components in the response, for example, because of low-pass filtering (some spectral components could be removed if they are not expected to be present in the response), or because of the expected duration of the response (components after a given latency are assumed to be null). If the response is expected to be contained in the reduced representation space, the matrix IRSA algorithm can be performed in the reduced representation space. However, the reduced dimensionality matrix IRSA algorithm cannot be used to remove, for example, the stimulation artefact, because this contribution is expected to be observed in the response (in spite of its non-biological origin), and ignoring it in the reduced representation space produces a bias in the estimated response.

Selecting an appropriate representation space

Dimensionality reduction for down-sampling

[0114] The objective of the transformation V_r is to reduce the dimensionality without removing any component expected to be present in the response. In the case of a response without components above a cut-off frequency B, according to the sampling theorem, the response can be down-sampled (or decimated) at a sampling frequency $f_s > 2B$, and the reduction of the sampling frequency can be interpreted as a dimensionality reduction. Before decimation, a low-pass filtering is usually applied in order to remove high frequency components associated to noise (to prevent aliasing caused by noise). The low-pass filter can be used for both the decimation (or dimensionality reduction) and the interpolation (or reconstruction of the response at the original sampling frequency from the decimated response).

[0115] The down-sampling procedure can easily be described with a matrix H_r. If $h(l)$ is the impulsive response of the low-pass filter and $x(j)$ is the response to be filtered and down-sampled, the low-pass filtered response is:

$$x_{lp}(j) = h(j) * x(j) = \sum_l h(l)x(j-l) \tag{53}$$

and if the decimation factor is m, the decimated response is:

$$x^h(j_r) = x_{lp}(j_r \cdot m) = \sum_l h(l)x(j_r \cdot m - l) \tag{54}$$

The low-pass filtering can be represented with matrix notation as:

$$\mathrm{x}^{h} = H\mathrm{x} \tag{55}$$

where H is the J × J convolution matrix representing the filter (H(i; j) = h(i − j)):

$$\begin{pmatrix} x^h(0) \\ x^h(1) \\ \vdots \\ x^h(J-1) \end{pmatrix} = \begin{pmatrix} h_0 & h_{-1} & h_{-2} & h_{-3} & \cdots \\ h_1 & h_0 & h_{-1} & h_{-2} & \cdots \\ h_2 & h_1 & h_0 & h_{-1} & \cdots \\ h_3 & h_2 & h_1 & h_0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} x(0) \\ x(1) \\ \vdots \\ x(J-1) \end{pmatrix} \tag{56}$$

(h(l) = $\mathrm{h_l}$ as used for a compact notation). Similarly, the low-pass filtering and decimation can be represented with matrix notation, where the decimation in a factor m is performed by removing m − 1 rows of m in the matrix H:

$$\mathrm{x}^{h_r} = H_r\mathrm{x} \tag{57}$$

where $\mathrm{H_r}$ is the$\mathrm{J_r}$ × J matrix containing the rows $\mathrm{J_r}$ × m (with $\mathrm{J_r}$ = 0, … , $\mathrm{J_r}$ − 1 and $\mathrm{J_r}$ = J/m) of the matrix H. For example, if the number of samples of the response is J = 12, and the decimation factor is m = 3, then the matrix equation would be:

$$\begin{pmatrix} x^{h_r}(0) \\ x^{h_r}(1) \\ x^{h_r}(2) \\ x^{h_r}(3) \end{pmatrix} = \begin{pmatrix} h_0 & h_{-1} & h_{-2} & h_{-3} & \cdots & h_{-11} \\ h_3 & h_2 & h_1 & h_0 & \cdots & h_{-8} \\ h_6 & h_5 & h_4 & h_3 & \cdots & h_{-5} \\ h_9 & h_8 & h_7 & h_6 & \cdots & h_{-2} \end{pmatrix} \begin{pmatrix} x(0) \\ x(1) \\ x(2) \\ \vdots \\ x(11) \end{pmatrix} \tag{58}$$

and the matrix Hr provides both low-pass filtering and decimation, with a dimensionality reduction from $J = 12$ to $J_r = 4$.

[0116] The rows of the matrix H_r define a basis of the reduced subspace, i.e. J_r linearly independent functions covering the reduced subspace. These functions can be orthonormalized with the Gram–Schmidt process. Using the orthonormalized functions v_{j_r}, the transformation providing the dimensionality reduction is:

$$V_r = \begin{pmatrix} \mathrm{v}_0^T \\ \mathrm{v}_1^T \\ \vdots \\ \mathrm{v}_{J_r-1}^T \end{pmatrix} \tag{59}$$

[0117] This transformation can be used to obtain the reduced representation from the response: $\mathbf{x}^{v_r} = V_r\mathbf{x}$. It can be also used to recover the response in the original representation from the reduced representation: $\mathbf{x} = V_r^T\mathbf{x}^{v_r}$, where $V_r^T V_r \neq I$ (because V_r is an incomplete

orthonormal transformation) but $\mathbf{x} = V_r^T V_r \mathbf{x}$ (because $\mathbf{x}$ is assumed to belong to the subspace: since $\mathbf{x}$ contains no high frequency components, the sampling theorem guarantees that the reduced representation can be used to recover the original response).

Frequency content of auditory responses

[0118] Auditory evoked response, including ABR, MLR and CAEP, is associated to the activity of different elements of the auditory pathway. Each portion of the response is characterized by a specific latency and bandwidth. Waves of ABR present latencies in the range 1 ms - 10ms, and contains components in the frequency band 100 Hz - 3000 Hz. MLR latencies are in the range 10 ms - 100 ms with frequencies in the range 10 Hz - 300 Hz. CAEP latencies are in the range 100 ms - 1000 ms and the frequency components are in the band 1 Hz - 30 Hz. These latencies and frequency bands determine the configuration of the recording procedure when the responses are registered. For example, in order to record ABR, a band-pass filter is applied to remove frequency components out of the band 100 Hz - 3000 Hz. This filter allows sampling at a minimum sampling rate of 8 kHz or 10 kHz (typically ABRs are recorded at 20 kHz or 25 kHz) and removes MLR and CAEP contributions (which allows the use of an averaging window of 10 ms or 12 ms). When MLR is recorded, the EEG is band-pass filtered for removing the components out of the range 10 Hz - 300 Hz. This filter removes the ABR and CAEP contributions from the EEG, allowing the use of an averaging window of 100 ms and a sampling frequency of 800 Hz or 1 kHz (even though typical sampling frequencies are 10kHz or 20 kHz for MLR). For CAEP recording, the EEG is filtered with a 1 Hz - 30 Hz band-pass filter and this evoked response could be recorded using a 1000 ms window at a sampling frequency of 80 Hz or 100 Hz (even though typical sampling frequency is 1 kHz or 2 kHz).

Latency specific filtering and down-sampling

[0119] According to these frequency bands, the matrix IRSA with dimensionality reduction could be applied, for ABR recording, by band-pass filtering the EEG between 100 Hz and 3000 Hz and using a 10 ms window and a sampling frequency of 10 kHz, which corresponds to a reduced dimension J_r of 100 components. Similarly, dimensionality reduction for MLR can be achieved by filtering between 10 Hz and 300 Hz, down-sampling to 1 kHz and using a 100 ms window, which provides a reduced dimension J_r of 100 components. And finally, for CAEP, filtering between 1 Hz and 30 Hz, down-sampling to 100 Hz and using a response window of 1000 ms provide a reduced dimension J_r of 100 components. Therefore, an appropriate filtering and downsampling allows a dimensionality reduction useful for a fast matrix IRSA. This dimensionality reduction is specific for the type of response. However, as can be seen,

the more central the potential, the later the response is, and the lower the frequency content is. There is a progressive reduction of the frequency band necessary for representing the evoked potentials that suggests some kind of latency-specific filtering. The matrix-based filtering and down-sampling processes described with equations (56) and (57) provide an easy procedure for implementing this latency-specific filtering, since each row of the filtering matrix corresponds to a specific latency and provides the specific impulsive response (and therefore the corresponding frequency response) around this latency.

[0120] The frequency response can therefore be modified at the different rows of the filtering matrix in order to provide a latency dependent filtering. Similarly, the ratio of rows preserved or discarded can also be modified consistently in order to provide a latency specific down-sampling according to the latency specific frequency content. By using the matrix description of the filtering and down-sampling processes, the reduction of the bandwidth and the down-sampling factor can be progressively modified with the latency (instead of using a constant configuration for each portion of the response). The progressive filtering and down-sampling can also be applied for recovering simultaneously the potentials corresponding to the whole auditory pathway (including ABR, MLR and CAEP) with a reduced dimensionality in the matrix IRSA procedure. With a few hundreds of components in the reduced representation, a matrix IRSA providing ABR, MLR and CAEP simultaneously can be performed, while performing the matrix IRSA in the complete representation space would require, in this case, to deal with a dimensionality $J = 10.000$ (corresponding to 1000 ms for an appropriate representation of CAEP and a minimum sampling frequency of 10 kHz for an appropriate representation of BAER).

Design of the reduced transformation for latency dependent filtering and down-sampling

[0121] In order to perform the latency dependent filtering and down-sampling at K_{dec} samples per decade, a logarithmic compression of the time axis is performed and the signal to be processed is then low-pass filtered and uniformly sampled in the compressed time axis. The original response contains samples at the time values:

$$t_j = jT_s \tag{60}$$

where $j = 0, 1, \ldots, J-1$, and $T_s = 1/f_s$ is the sampling period. The samples in the compressed time axis can be related with the linear time with the equation:

$$j_r(t) = K_{dec} \log_{10}\left(\frac{t}{T_s} + \beta\right) - K_{dec} \log_{10}(\beta) \tag{61}$$

[0122] The constant β provides a linear behavior if $t/T_s \ll \beta$ and a logarithmic compression if $t/T_s \gg \beta$. In order to provide at least one sample in the reduced representation for each sample in the original representation, the following relation should be verified:

$$\left.\frac{d\, j_r(j)}{d\, j}\right|_{j=0} = \left.\frac{d\, j_r(t/T_s)}{d\,(t/T_s)}\right|_{t=0} = 1 \tag{62}$$

which provides the value of β:

$$\beta = \frac{K_{dec}}{\ln(10)} \tag{63}$$

where $ln()$ is the natural logarithm. The additive term of equation (61) sets a null index in the compressed representation for the first sample in the original representation, i.e.$j_r(0) = 0$. With this value of β, the equation (61) can be rewritten as:

$$j_r(t) = K_{dec} \log_{10}\left(\frac{t}{T_s}\frac{\ln(10)}{K_{dec}} + 1\right) \tag{64}$$

[0123] Finally, for large enough values of t (i.e., when $t = T_s \gg \ln(10)/K_{dec}$), and increase of a decade provides an increase of K_{dec} samples in $j_r(t)$:

$$j_r(10\,t) \approx K_{dec}\log_{10}\left(\frac{10\,t}{T_s}\frac{\ln(10)}{K_{dec}}\right) = K_{dec}\log_{10}\left(\frac{t}{T_s}\frac{\ln(10)}{K_{dec}}\right) + K_{dec}\log_{10}(10) = j_r(t) + K_{dec} \tag{65}$$

and therefore, the constant K_{dec} represents the number of samples per decade in the reduced representation.

[0124] Due to the compression, the sampling period of the compressed time axis is not constant. If we define $T'_s(t)$ as the sampling period of the compressed representation (which is a function of the latency t), it can be estimated as:

$$T'_s(t) = t(j_r+1) - t(j_r) \approx \frac{t(j_r+\epsilon) - t(j_r)}{j_r+\epsilon-j_r}(j_r+1-j_r) = \left.\frac{d\,t(j_r)}{d\,j_r}\right|_{j_r} = \frac{1}{d\,j_r(t)/d\,t|_t} \tag{66}$$

where the derivative, obtained from equation (64), is:

$$\frac{d\, j_r(t)}{d\, t} = \frac{1}{T_s}\left(\frac{t}{T_s}\frac{\ln(10)}{K_{dec}} + 1\right)^{-1} = \left(t\frac{\ln(10)}{K_{dec}} + T_s\right)^{-1} \tag{67}$$

and therefore, the sampling period and the sampling frequency in the compressed time-axis are, respectively:

$$T'_s(t) = T_s + t\frac{\ln(10)}{K_{dec}} \qquad f'_s(t) = \frac{1}{T'_s} \tag{68}$$

[0125] The maximum sampling frequency (or the minimum sampling period) of the reduced representation is obtained for the first sample (when t = 0):

$$\lim_{t\to 0} T'_s(t) = T_s \qquad \lim_{t\to 0} f'_s(t) = \frac{1}{T_s} = f_s \tag{69}$$

and as the latency increases, the sampling frequency is decreasing and the sampling period increasing according to equation (68). For large enough values of t (when $t/T_S \gg K_{dec}/\ln(10)$), the sampling period and frequency can be approached as:

$$T'_s(t) \approx t\frac{\ln(10)}{K_{dec}} \qquad f'_s(t) \approx \frac{1}{t}\frac{K_{dec}}{\ln(10)} \tag{70}$$

and they depend on the latency t and the number of samples per decade K_{dec} but not on the original sampling period. The bandwidth preserved at each latency depends on the local sampling frequency and the low-pass filter (that should be included in order to remove high-frequency components of the noise). The bandwidth is limited to $f'_s(t)/2$ by the sampling theorem, and a low-pass filter preserving a bandwidth $B'(t) = 0{:}45f'_s(t)$ is reasonable and easy to be implemented.

[0126] Table 1 shows the local sampling frequency $f'_s(t)$ and bandwidth $B'(t)$ (assuming $B'(t) = 0{:}45f'_s(t)$, as a function of the latency, for original sampling frequencies f_s = 25 kHz and f_s = 100 kHz, and using two different resolutions in the reduced representation space (K_{dec} = 40 and K_{dec} = 60 samples per decade). As can be observed, the preserved bandwidth depends on the original sampling frequency only for early latencies. According to the bandwidth assumed for the different types of evoked potentials (minimum 3 kHz at 1 ms for ABR, 300 Hz at 10 ms for MLR, 30 Hz at 100 ms for CAEP) a resolution of 40 samples per decade would be enough for an appropriate representation of the different waves.

Table 1

	K_{dec} = 40 samples/dec				K_{dec} = 60 samples/dec			
	f_s =25 kHz		f_s = 100 kHz		f_s =25 kHz		f_s = 100 kHz	
t	$f'_s(t)$	$B'(t)$	$f'_s(t)$	$B'(t)$	$f'_s(t)$	$B'(t)$	$f'_s(t)$	$B'(t)$
1 ms	10.2 kHz	4.61 kHz	14.8 kHz	6.66 kHz	12.8 kHz	5.74 kHz	20.7 kHz	9.30 kHz
2 ms	6.45 kHz	2.90 kHz	7.99 kHz	3.60 kHz	8.57 kHz	3.85 kHz	11.5 kHz	5.19 kHz
5 ms	3.05 kHz	1.37 kHz	3.36 kHz	1.51 kHz	4.31 kHz	1.94 kHz	4.95 kHz	2.23 kHz
10 ms	1.62 kHz	731 Hz	1.71 kHz	768 Hz	2.36 kHz	1.06 kHz	2.54 kHz	1.14 kHz
20 ms	839 Hz	378 Hz	861 Hz	388 Hz	1.24 kHz	557 Hz	1.29 kHz	579 Hz
50 ms	343 Hz	154 Hz	346 Hz	156 Hz	511 Hz	230 Hz	518 Hz	233 Hz
100 ms	173 Hz	77.6 Hz	173 Hz	78.0 Hz	258 Hz	116 Hz	260 Hz	117 Hz
200 ms	86.6 Hz	39.0 Hz	86.8 Hz	39.1 Hz	130 Hz	58.3 Hz	130 Hz	58.6 Hz
500 ms	34.7 Hz	15.6 Hz	34.7 Hz	15.6 Hz	52.0 Hz	23.4 Hz	52.1 Hz	23.4 Hz
1000 ms	17.4 Hz	7.81 Hz	17.4 Hz	7.82 Hz	26.0 Hz	11.7 Hz	26.1 Hz	11.7 Hz

[0127] For the low-pass filtering of the original signal, a raised-cosine filter in the compressed time axis is designed. The raised-cosine signal is commonly used in digital communications because it provides an appropriate limitation of the bandwidth with a relatively short duration in the impulsive response that can be controlled with a roll-off factor. The impulsive response of the filter is scaled in order to provide a constant bandwidth in the compressed time axis (and dependent on the latency according to the local sampling frequency $f'_s(t)$), and is sampled at the time instants t_j of the original time representation. The sampling functions (i.e., the functions used to obtain each sample in the reduced representation) are stored as the rows in the filtering and decimation matrix H_r. Finally, the sampling functions are orthonormalized with the Gram-Schmidt process in order to obtain an orthonormal basis for the reduced representation space.

[0128] The vectors of the basis are arranged in a $J_r \times J$ matrix, V_r, providing the reduced representation (equivalent to a latency-dependent low-pass filtering and a down-sampling). The V_r matrix verifies that $V_r V_r^T$ is a $J_r \times J_r$ identity matrix (because the sampling vectors are orthogonal), and that $V_r^T V_r$ is a $J \times J$ square matrix providing a latency dependent low-pass filtering.

Example of the basis for dimensionality reduction

[0129] The Figures 7 and 8 show the functions of the basis before and after the Gram-Schmidt orthonormalization, respectively. The roll-off factor of the raised-cosine functions was $\alpha = 0{:}25$, and the sampling period was 0.9 of the raised-cosine symbol period. The sampling frequency of the original representation was 25 kHz, and the reduced representation was configured to provide$K_{dec} = 10$ samples/decade. The plots in the top represent the functions of the basis versus the time in linear scale, while the plots in the bottom represent them as a

function of the logarithmically scaled time. As observed in these plots, the basis provides a non-uniform low-pass filtering and down-sampling, where the band-width and the sampling rate decrease with the latency. For small latencies, the sampling tends to be uniform (sampling frequency similar to the original 25 kHz). As the latency increases, the sampling tends to be uniform in the logarithmically compressed time axis, with 10 samples per decade (for example, 10 samples between 1 ms and 10 ms, or between 5 ms and 50 ms). The detailed representation of the sampling functions (plots in the right side of FIGURE 7) show the raised-cosine function (in the logarithmically scaled time axis) centered at the position of each sample.

[0130] The Gram-Schmidt orthonormalization (Figure 8) modifies both the shape and the amplitude of the functions, in order to make $\mathbf{v}_i^T \mathbf{v}_j = \delta_{i,j}$. Even though the position of the main lobe of each sampling function corresponds to the latency of each sample, the shape is affected by the orthonormalization process, mainly in the left-side of each function (because orthonormalization has been performed starting at early latencies). The amplitude decreases as the latency increases as a consequence of the normalization.

Latency dependent low-pass filtering

[0131] Figure 9 illustrates the latency dependent filtering provided by the reduced representation. Using a basis V_r for reducing the representation from $J = 10000$ samples (at $f_s = 25$ kHz, time interval [0 - 400 ms]) to $J_r = 47$ (with $K_{dec} = 15$ samples/decade), a synthetic signal was generated using 47 random values in the reduced representation by transforming them to the complete representation with $= V_r^T \mathbf{x}^{vr}$. Additive Gaussian White Noise (AGWN) at 15.2 dB was added to this synthetic signal ($\mathbf{x}_n = \mathbf{x} + \mathbf{n}$). The noisy signal was then filtered by transforming $\mathbf{x}_n$ to the reduced representation space, and then transforming the reduced representation back to the original representation: $\hat{\mathbf{x}} = V_r^T (V_r \mathbf{x}_n)$. Figure 9 shows both the noisy signal $\mathbf{x}_n$, and the recovered signal $\hat{\mathbf{x}}$. In the left side plots, the time is in linear scale, while it is logarithmically scaled in the right side plots. The plots in the top represent the whole time interval ([0 ms - 400 ms]), and the other plots represent a detail of different time intervals ([0.4 ms - 4 ms], [4 ms - 40 ms] and [40 ms - 400 ms]). As can be observed, the reduced representation provides a latency dependent filtering, with a decreasing bandwidth as the latency increases. Additionally, it can be observed that the representation using the logarithmically scaled time-axis provides a more comprehensive visualization of the different waves across the three decades.

[0132] The plots in Figure 10 illustrate that filtering a previously filtered signal has no significant effect (or equivalently, using a resolution better than the required has no effect on

the recovered signal). The original signal $\mathbf{x}_0$, generated with K_{dec} = 15 samples/decade, has been projected with 4 different transformations $\hat{\mathbf{x}}_k = V_{r,k}^T(V_{r,k}\mathbf{x}_0)$, $(k = 1,2,3,4)$ generated with $K_{dec} = 20,30,40,50$ samples/decade. As expected, since the reduced subspaces are included in the next ones as Kdec increases, the resulting recovered signals are identical to the original one. This result, obvious in the case of filters with constant band-width (a band-limited signal remains invariant when it is filtered with a band-width wider than that of the signal), takes also place when latency-dependent low-pass filters are applied. The error between the recovered and the original signal was measured, and SNR associated to this error was greater than 40 dB (with error mainly associated to numerical accuracy and truncation at the end of the signals).

Results

Results with synthetic signals - Configuration of the synthetic EEGs

[0133] The experiments with synthetic signals were performed with a sampling frequency fs = 25 kHz. A known pseudo response was used, with a duration of 400 ms (10000 samples). This pseudo response was that one used in Figures 3 and 4 (obtained by projecting a random signal with VT_rV_r using a transformation V_r with 15 samples/decade). A number of stimuli was generated with a random ISI following a uniform distribution between 30 ms and 100 ms. Three EEGs were synthesized by convolution of the response with the stimulation signal, using sequences of 2000 stimuli (duration of the EEG: 132 seconds, 3.30 millions of samples), 5000 stimuli (325 seconds, 8.14 millions of samples) and 10000 stimuli (649 seconds, 16.2 millions of samples). The EEGs were contaminated with AWGN at -6 dB. Figure 11 represents a portion of the clean and noisy EEGs used in the simulations. The EEGs are represented in blue, while the stimulation signal is represented in red.

Results with synthetic signals - Comparison of the responses estimated with the different IRSA implementations

[0134] Figure 12 compares the responses obtained with IRSA after 50 iterations with a convergence control parameter α =0.1. In this simulation, the EEG was synthesized with 10000 stimuli (649 seconds and 16.2 millions of samples in the EEG). The left panel includes the response provided by conventional IRSA and that provided by the matrix implementation of IRSA. The right panel shows the response provided by matrix-IRSA projected using the transformation $V_r^TV_r$ and the response provided by matrix-IRSA performed in the reduced representation space. The transformation reducing the dimensionality was prepared with 40

samples/decade. The pseudo response used in the experiments for preparing the synthetic EEG has been included as reference.

[0135] As can be observed, the conventional-IRSA and the matrix-IRSA provide indistinguishable results. The difference between both responses is associated to the accuracy of the numerical representation (the amplitude of the difference is around 10^{-15} the amplitude of the response). As can be observed, the estimated response tends to the reference, but 50 iterations are not enough to achieve convergence. The estimated signal is affected by the noise (due to the noise added to the synthetic EEG). The comparison of the estimated responses illustrates that conventional-IRSA and matrix-IRSA provide identical results.

[0136] The projection using $V_r^T V_r$ provides a latency-dependent filtering of the response (as can be seen in the right panel). Again, 50 iterations seems to be insufficient. The response obtained by projecting the result is very similar to that provided by matrix-IRSA in the reduced representation space. In this case both responses are hardly distinguished even though the amplitude of the difference is around 10^{-3} the amplitude of the response. The slight difference between both procedures seems to be associated to accidental correlation between the noise contaminating the signal and the stimulation sequence (which provides a small contribution to the estimated response out of the reduced representation space that propagates the error as described with equation (52)).

[0137] Figure 13 compares the results provided by matrix-IRSA after projection and those provided by matrix-IRSA in the reduced representation space after 10000 iterations. The left panel shows the response estimations together with the pseudo response (included as reference), while the right panel shows the difference between both estimations. As can be observed, after a large enough number of iterations, the estimated responses converged to the pseudo response. Both estimates are again hardly distinguished, and the difference between them is significantly smaller than the response, which illustrates that performing matrix-IRSA and projecting the result in the subspace is equivalent to performing the matrix-IRSA in the reduced subspace.

[0138] Table 2 evaluates the difference between the responses estimated by the different IRSA implementations. The responses are compared in terms of energy ratio of the reference response to the difference between both compared responses, expressed in dB (or, equivalently, SNR associated to the comparison of both responses). Since the simulation allows a comparison with the template used for preparing the synthetic EEGs, the responses provided by the different IRSA implementations are firstly compared with the template. Due to

the large execution time of the conventional IRSA, experiments were carried out with this implementation only for 50 iterations. The experiments for 100000 iterations were carried out only for the matrix-IRSA in the reduced representation space, in order to verify the convergence of the responses after 10000 iterations (matrix-IRSA in the complete representation space would be prohibitive for 100000 iterations).

Table 2 – Simulation results: Comparison of the responses estimated with the different implementations of IRSA.

	Experimental conditions:				
Comparison:	50 iter. 2000 stimuli	50 iter. 5000 stimuli	50 iter. 10000 stimuli	10000 iter. 10000 stimuli	100000 iter. 10000 stimuli
conv-IRSA vs. Template	10.88 dB	11.17 dB	11.14 dB	-	-
matrix-IRSA vs. Template	10.88 dB	11.17 dB	11.14 dB	27.11 dB	-
matrix-IRSA-proj vs. Template	11.36 dB	11.38 dB	11.24 dB	39.45 dB	-
matrix-IRSA-red vs. Template	11.35 dB	11.37 dB	11.23 dB	39.77 dB	39.77 dB
matrix-IRSA vs. conv-IRSA	301.63 dB	301.09 dB	300.41 dB	-	-
matrix-IRSA-red vs. matrix-IRSA-proj	54.77 dB	50.75 dB	54.98 dB	59.11 dB	-

[0139] The comparison of the responses provided by the different implementations of IRSA with the template shows that (1) 50 iterations are not enough for convergence (the SNR increases from 11 dB at 50 iterations to 27 dB or 39 dB at 10000 iterations); (2) 10000 iterations are enough for convergence (there is no increase of SNR from 10000 to 100000 iterations); (3) projection (either with matrix-IRSA-projection or with matrix-IRSA-red) provides a latency dependent filtering that improves the quality of the response (at 10000 iterations the latency dependent filtering increases the SNR from 27 dB to 39 dB); (4) the quality of the responses provided by conventional IRSA and matrix-IRSA is identical (identical SNR when compared with the template, and SNR associated to the difference close to 300 dB); (5) the quality of the responses provided by matrix-IRSA-proj and matrix-IRSA-red is very similar (very similar SNR when compared with the template, and SNR associated to the difference greater than 50 dB); (6) in the case of 50 iterations, the increase in the number of stimuli does not provide the expected improvement in the quality of the estimated responses (the expected improvement is 4 dB when the number of stimuli increases from 2000 to 5000, 3 dB when it increases from 5000 to 10000), because of the insufficient number of iterations (the error associated to insufficient number of iterations is greater than that associated to the additive noise in this condition).

[0140] The responses provided by conventional-IRSA and matrix-IRSA are essentially identical, as reflected by so large SNR (around 300 dB in all the comparisons). This small difference is caused by the accuracy in the internal numerical representation. The error between both IRSA procedures is much smaller than the error between the estimated responses and the template (approximately 11 dB after 50 iterations, and 27.11 dB after 10000

iterations), and therefore the responses provided by both implementations can be assumed to be identical.

[0141] The latency dependent filtering provides and evident improvement of quality (an increase of 12 dB in the response estimations in this simulation). This improvement is the result of a reduction of the noise associated to the latency dependent filtering, which removes noise components with frequency above the cut-off frequency associated to each latency and preserves the frequency components of the estimated response (because the pseudo response was generated with 15 samples/decade and the latency dependent filtering was defined with 40 samples/decade and therefore the response is not expected to be distorted by this latency dependent filtering).

[0142] The SNR associated to the comparison of matrix-IRSA after projection and matrix-IRSA performed in the reduced representation space is lower, but it is also very high (always higher than 50 dB). As discussed previously, the small differences are probably associated to accidental correlations between the noise and the stimulation signal, which propagates a difference between both estimations procedures according to equation (52). In any case, with so large SNR, in practice both results can be assumed to be identical, since the error between the estimated responses and the template is significantly greater and the associated SNR significantly lower (about 11 dB after 50 iterations, and close to 40 dB at convergence) than those corresponding to the comparison of both IRSA estimations (more than 50 dB at 50 iterations and close to 60 dB at 10000 iterations).

Results with synthetic signals - Comparison of the execution time for the IRSA implementation

[0143] Table 3 shows the results of the simulations in terms of computational load. The execution time for the different procedures (using a desktop computer with an Intel-Core i7-3770 CPU, 3.40 GHz, 8.00 GB RAM) are compared in this Table. The execution time was measured in different conditions in order to observe the influence of the EEG length or the number of iterations. The EEG length (associated to the number of stimuli in these experiments) increases all the execution times (initialization, iterations and total execution time) in conventional IRSA. Similarly the time required for the initialization increases with the EEG length for matrix-IRSA and matrix-IRSA-red. However, the time required for each iteration is not affected by the EEG length in the matrix implementations of the algorithm, as expected from the formulation. The time required for each iteration is almost constant in the case of matrix-IRSA (there are some fluctuations associated to the computer dedication to the algorithms, since the computer was running other processes simultaneously). The time

required for each iteration of matrix-IRSA-red was not accurately measured for 50 iterations because of its short duration and the small number of iterations. For a large number of iterations, the execution times for matrix-IRSA-red are more reliable (as can be observed in the Table for 10000 and 100000 iterations).

[0144] A reasonable estimation based on IRSA (with a small convergence control parameter) requires several thousands of iterations. If the execution times for 10000 iterations are compared, one can observe that the conventional IRSA algorithm would require a prohibitive execution time (the execution time in this case, 23690 s, or approximately 6h:35', was estimated from the results for 50 iterations). The matrix implementation of IRSA reduces the execution time to 487.9 s (approximately 8 minutes), and therefore, the execution time is reduced in a factor 50. The matrix-IRSA performed in the reduced representation space provides the response in less than 10 s, which corresponds to an additional reduction of the execution time in a factor 50 (the reduction was around 2500 with respect to the conventional IRSA). It is remarkable that most of the execution time of matrix-IRSA-red is devoted to the initialization, while most of the execution time of matrix-IRSA and conv-IRSA is devoted to the iterations, and therefore, the improvement in the execution time is more important as more iterations are performed. Obviously, the reduction of the execution time of the matrix-IRSA-red with respect to the matrix-IRSA is associated to the reduction of the dimensionality (in this case, from $J = 10000$ samples when the algorithm is performed in the complete representation space to $J_r = 110$ samples in the reduced representation space).

[0145] Table 3 shows the results of the simulations in terms of computational load. The execution time for the different procedures (using a desktop computer with an Intel-Core i7-3770 CPU, 3.40 GHz, 8.00 GB RAM) are compared in this Table. The execution time was measured in different conditions in order to observe the influence of the EEG length or the number of iterations. The EEG length (associated to the number of stimuli in these experiments) increases all the execution times (initialization, iterations and total execution time) in conventional IRSA. Similarly the time required for the initialization increases with the EEG length for matrix-IRSA and matrix-IRSA-red. However, the time required for each iteration is not affected by the EEG length in the matrix implementations of the algorithm, as expected from the formulation. The time required for each iteration is almost constant in the case of matrix-IRSA (there are some fluctuations associated to the computer dedication to the algorithms, since the computer was running other processes simultaneously). The time required for each iteration of matrix-IRSA-red was not accurately measured for 50 iterations because of its short duration and the small number of iterations. For a large number of

iterations, the execution times for matrix-IRSA-red are more reliable (as can be observed in the Table for 10000 and 100000 iterations).

Table 3 - Simulation results: Execution time required for the different implementations of IRSA.

Procedure	stimuli	iterations	t_{ini}	t_{iter}	t_{tot}
conv-IRSA	2000	50	0.198 s	0.467 s	23.53 s
matrix-IRSA	2000	50	4.307 s	53.2 ms	6.96 s
matrix-IRSA-red	2000	50	4.525 s	0.148 ms	4.53 s
conv-IRSA	5000	50	0.497 s	1.176 s	59.28 s
matrix-IRSA	5000	50	4.980 s	51.7 ms	7.56 s
matrix-IRSA-red	5000	50	5.127 s	0.188 ms	5.14 s
conv-IRSA	10000	50	1.062 s	2.369 s	119.5 s
matrix-IRSA	10000	50	9.612 s	51.9 ms	12.2 s
matrix-IRSA-red	10000	50	10.24 s	0.162 ms	10.3 s
conv-IRSA	10000	10000	*1.062 s*	*2.369 s*	*23690 s*
matrix-IRSA	10000	10000	9.86 s	47.8 ms	487.9 s
matrix-IRSA-red	10000	10000	9.82 s	5.08 μs	9.87 s
matrix-IRSA-red	10000	100000	10.40 s	5.85 μs	11.0 s

[0146] A reasonable estimation based on IRSA (with a small convergence control parameter) requires several thousands of iterations. If the execution times for 10000 iterations are compared, one can observe that the conventional IRSA algorithm would require a prohibitive execution time (the execution time in this case, 23690 s, or approximately 6h:35', was estimated from the results for 50 iterations). The matrix implementation of IRSA reduces the execution time to 487.9 s (approximately 8 minutes), and therefore, the execution time is reduced in a factor 50. The matrix-IRSA performed in the reduced representation space provides the response in less than 10 s, which corresponds to an additional reduction of the execution time in a factor 50 (the reduction was around 2500 with respect to the conventional IRSA). It is remarkable that most of the execution time of matrix-IRSA-red is devoted to the initialization, while most of the execution time of matrix-IRSA and conv-IRSA is devoted to the iterations, and therefore, the improvement in the execution time is more important as more iterations are performed. Obviously, the reduction of the execution time of the matrix-IRSA-red with respect to the matrix-IRSA is associated to the reduction of the dimensionality in this case, from $J = 10000$ samples when the algorithm is performed in the complete representation space to $J_r = 110$ samples in the reduced representation space).

Results with real EEGs - Recording session

[0147] The evaluation of the proposed IRSA optimizations using real EEGs was based on an AEP experiment in which 4 different stimulation rates were configured. The stimulation signal was prepared using a uniform distribution of ISI between 500 and 800 ms (for an average stimulation rate of 1.53 Hz), between 300 and 600 ms (for 2.22 Hz), between 100 and 300 ms

(for 5.00 Hz) and between 30 and 100 ms (for 15.38 Hz). The number of stimuli used for each configuration was increased with the stimulation rate (from 1500 stimuli at the slowest rate to 20000 stimuli at the fastest rate). The stimulation consisted in a sequence of rarefaction clicks presented at the instants defined by the stimulation sequence. The clicks were delivered diotically through ER-3A insert earphones at 60 dB HL. The recording electrodes were located at the upper forehead (F_z, active), at the mastoids (Tp9 and Tp10, references 1 and 2) and at the middle forehead (Fpz, ground). The EEGs were recorded using a BioSemi instrumentation pre-amplifier (BioSemi V.B., Amsterdam, Netherlands), with a [1-3000] Hz bandwidth and a sampling frequency of 16384 samples per second. The [Fz-Tp9] and [Fz-Tp10] were averaged to obtain a single EEG. Eye-blink artifacts were suppressed with the iterative template matching and suppression (ITMS), an algorithm that detects, models and suppresses blink-artifacts from a single-channel EEG (Valderrama et al., 2018 - Ref 19). Table 4 summarizes the configurations involved in this EEG recording session. Since one of the objectives of the inventors was the evaluation of the optimization procedures proposed for the IRSA algorithm, only one subject (male, 33 years) was considered in these experiments.

Table 4: Configuration of the EEG recording session.

Configuration	ISI	aver. stim. rate	K (stimuli)	EEG length (seconds)	EEG length (samples)
1	500 - 800 ms	1.53 Hz	1500	990	$16.22 \cdot 10^6$
2	300 - 600 ms	2.22 Hz	2000	924	$15.14 \cdot 10^6$
3	100 - 300 ms	5.00 Hz	5000	1021	$16.72 \cdot 10^6$
4	30 - 100 ms	15.38 Hz	20000	1322	$21.66 \cdot 10^6$

Results with real EEGs - Comparison of the responses estimated with the different IRSA implementations

[0148] The EEGs have been processed with different versions of the IRSA algorithm, including conventional IRSA, matrix-IRSA and matrix-IRSA performed in a reduced representation space. The length of the response was set to 1 second ($J = 16384$ samples) and the convergence parameter for IRSA was set to $\alpha = 0.02$ in order to avoid oscillations in the iterative algorithm. The dimensionality reduction was prepared with 40 samples per decade, and the dimension of the reduced representation space was $J_r = 119$. The conventional and matrix implementations of IRSA have been compared for 50 iterations, and the matrix implementations (in the complete representation space and in the reduced representation space) have been compared for both 50 and 10000 iterations. The matrix-IRSA in the reduced representation space has also been performed for 100000 iterations in order to verify convergence.

[0149] Figure 14 shows the response estimations provided by different IRSA implementations after 50 iterations. The time axis is logarithmically scaled in order to clearly show the different evoked potentials, including three decades between 1 ms and 1 s. The waves of the evoked potentials are indicated in the plots, including ABRs (waves $\mathrm{I}, \mathrm{II}, \mathrm{III}, \mathrm{V}$), MLRs (waves $N_0, P_0, N_a, P_a, N_b, P_b$) and CAEPs (waves P_1, N_1, P_2). The different plots correspond to different ISI configurations (or average stimulation rates), and the change in the latency and amplitude of the waves can be appreciated as the stimulation rate increases. The left panel compares the responses estimated with conventional IRSA and matrix-IRSA, while the right panel compares those estimated with matrix-IRSA followed by projection with $V_r^T V_r$ and matrix-IRSA performed in the reduced representation space. When the responses of the left and right panels are compared, a reduction of the noise can be appreciated as a consequence of the latency dependent filtering. On the other hand, the responses provided by conv-IRSA and matrix-IRSA, as well as those provided by matrix-IRSA-proj and matrix-IRSA-red, were found to be hardly distinguishable. Figure 15 shows the responses provided by matrix-IRSA-proj and matrix-IRSA-red after 10000 iterations (conventional IRSA for so many iterations is prohibitive and was not computed). In this Figure the amplitudes of the estimated responses are larger than those obtained for 50 iterations because convergence requires several thousands of iterations (50 iterations are clearly insufficient). The right panel of Figure 15 represents the difference between matrix-IRSA-proj and matrix-IRSA-red. It can be appreciated that the difference is small (the vertical scale for the responses is $1\mu\mathrm{V/div}$, while the scale for the differences is $0.02\mu V/\mathrm{div}$).

Table 5: Results with real EEGs: Comparison of the responses estimated with the different implementations of IRSA.

Compared methods		Configuration ISI (ms)	stimuli	EEG duration	50 iterations	10000 iterations
conv-IRSA vs. matrix-IRSA	1	500-800	1500	990 s	309.09 dB	-
	2	300-600	2000	924 s	309.16 dB	-
	3	100-300	5000	1021 s	305.13 dB	-
	4	30-100	20000	1322 s	303.17 dB	-
	Average				**306.64 dB**	-
matrix-IRSA-proj vs. matrix-IRSA-red	1	500-800	1500	990 s	53.76 dB	32.04 dB
	2	300-600	2000	924 s	49.17 dB	37.22 dB
	3	100-300	5000	1021 s	34.93 dB	26.55 dB
	4	30-100	20000	1322 s	48.24 dB	35.68 dB
	Average				**46.53 dB**	**32.87 dB**

[0150] Table 5 compares the responses estimated by conv-IRSA and matrix-IRSA (for 50 iterations) and those estimated by matrix-IRSA-proj and matrix-IRSA-red (for 50 and 10000 iterations). As in the case of the simulations, conventional and matrix implementations provides identical results (SNR associated to the comparisons around 300 dB, probably due

to the numerical precision). However, the comparison of matrix-IRSA-proj and matrix-IRSA-red provides a SNR lower than that observed in the simulations (46 dB in average for 50 iterations, 33 dB in average for 10000 iterations). The difference between both methods is small, and is probably associated to accidental correlation between the noise and the stimulation sequence that propagates the error according to equation (52). In order to verify convergence, the response estimations after 50, 10000 and 100000 iterations have been compared in Table 6. The comparison of responses for 50 and 10000 iterations (with SNR around 3 dB) reveals that convergence is not achieved for 50 iterations (as observed when Figures 14 and 15 are compared). On the other hand, 10000 iterations are enough for convergence (the SNR associated to the comparison of responses for 10000 and 100000 is greater than 170 dB in all the configurations).

Table 6: Results with real EEGs: Comparison of the responses estimated with 50, 10000 and 100000 iterations with matrix-IRSAred.

Compared conditions	Configuration				IRSA method matrix-IRSA-red
		ISI (ms)	stimuli	EEG duration	
50 iterations vs. 10000 iterations	1	500-800	1500	990 s	6.22 dB
	2	300-600	2000	924 s	4.54 dB
	3	100-300	5000	1021 s	3.10 dB
	4	30-100	20000	1322 s	1.59 dB
	Average				**3.86 dB**
10000 iterations vs. 100000 iterations	1	500-800	1500	990 s	302.27 dB
	2	300-600	2000	924 s	176.25 dB
	3	100-300	5000	1021 s	182.51 dB
	4	30-100	20000	1322 s	172.95 dB
	Average				**208.50 dB**

Results with real EEGs - Comparison of the execution time for the IRSA implementations

[0151] Table 7 shows the execution times associated to the different implementations of IRSA. The time required for initialization, for each iteration and the total execution time are estimated for different conditions and IRSA implementations. The execution times observed in this Table are consistent with those observed for the synthetic EEGs: (1) the time required for each iteration depends on the number of stimuli and EEG length for the conventional IRSA, but not for the matrix implementations; (2) matrix-IRSA provides a substantial reduction of the computational time with respect to conventional-IRSA; (3) for a reasonable number of iterations (several thousands) most of the execution time is devoted to the iterations in conventional-IRSA and matrix-IRSA; however in matrix-IRSA-red the time involved in each iteration is very small (several microseconds) and most of the execution time is devoted to the initialization; (4) for a reasonable number of iterations, matrix-IRSA-red provides a substantial reduction of the execution time with respect to matrix-IRSA or conventional-IRSA. The total

execution time for 10000 iterations was reduced from 28h:30' (conventional-IRSA, estimated from results for 50 iterations) to 1h:36' (matrix-IRSA) and to 1':17" (matrix-IRSA-red). The execution time is therefore reduced in a factor 18 due to the matrix implementation, and an additional factor 75 due to the implementation in the reduced representation space, providing a total time reduction in a factor 1300 when both optimizations are combined.

Table 7: Results with real EEGs. Time required for execution for different responses, procedures and conditions.

Procedure	iterations	Configuration				Execution time		
			ISI (ms)	stimuli	EEG dur.	t_{ini}	t_{iter}	t_{tot}
Conv-IRSA	50	1	500-800	1500	990 s	0.215 s	0.593 s	29.9 s
		2	300-600	2000	924 s	0.286 s	0.775 s	39.0 s
		3	100-300	5000	1021 s	0.708 s	1.804 s	90.9 s
		4	30-100	20000	1322 s	2.820 s	7.005 s	353.1 s
		Total execution time						**512.9 s**
Mat-IRSA	50	1	500-800	1500	990 s	14.73 s	0.131 s	21.3 s
		2	300-600	2000	924 s	14.50 s	0.130 s	21.0 s
		3	100-300	5000	1021 s	14.38 s	0.131 s	21.0 s
		4	30-100	20000	1322 s	25.78 s	0.140 s	32.8 s
		Total execution time						**96.1 s**
Mat-IRSA-red	50	1	500-800	1500	990 s	15.80 s	139 μs	15.8 s
		2	300-600	2000	924 s	15.90 s	134 μs	15.9 s
		3	100-300	5000	1021 s	15.55 s	195 μs	15.6 s
		4	30-100	20000	1322 s	25.58 s	435 μs	25.6 s
		Total execution time						**72.9 s**
Conv-IRSA (estimation)	10000	1	500-800	1500	990 s	0.215 s	0.593 s	5980 s
		2	300-600	2000	924 s	0.286 s	0.775 s	7800 s
		3	100-300	5000	1021 s	0.708 s	1.804 s	18180 s
		4	30-100	20000	1322 s	2.820 s	7.005 s	70620 s
		Total execution time						**102580 s**
Mat-IRSA	10000	1	500-800	1500	990 s	16.17 s	0.138 s	1399.2 s
		2	300-600	2000	924 s	15.61 s	0.139 s	1407.2 s
		3	100-300	5000	1021 s	15.88 s	0.141 s	1427.2 s
		4	30-100	20000	1322 s	37.68 s	0.152 s	1560.7 s
		Total execution time						**5794.3 s**
Mat-IRSA-red	10000	1	500-800	1500	990 s	17.25 s	5.24 μs	17.3 s
		2	300-600	2000	924 s	17.12 s	6.13 μs	17.2 s
		3	100-300	5000	1021 s	16.62 s	6.51 μs	16.7 s
		4	30-100	20000	1322 s	25.68 s	5.99 μs	25.7 s
		Total execution time						**76.9 s**
Mat-IRSA-red	100000	1	500-800	1500	990 s	16.80 s	6.18 μs	17.4 s
		2	300-600	2000	924 s	16.79 s	6.19 μs	17.4 s
		3	100-300	5000	1021 s	17.18 s	6.34 μs	17.8 s
		4	30-100	20000	1322 s	26.36 s	6.14 μs	27.0 s
		Total execution time						**79.6 s**

Results with real EEGs - Selecting an appropriate resolution for dimensionality reduction

[0152] An important contribution for the reduction of the computational load is associated to the reduction of the dimensionality. The dimensionality of the reduced representation space depends on the resolution used for the basis definition. A resolution of $K_{dec} = 40$ samples/decade provides a reasonable estimation of the responses. However, selecting an appropriate K_{dec} is important, since a greater K_{dec} involves more computational load, while a smaller one would reduce the accuracy of the estimated responses (since an appropriate representation of the response requires a minimum resolution).

[0153] In order to determine what is the appropriate resolution, a sensitivity study has been carried out by estimating the responses for different resolutions ranging between 5 and 90 samples per decade. Figure 16 shows the responses estimated at different resolutions with matrix-IRSA-red after 10000 iterations. As can be observed, a resolution smaller than 20 samples/decade is not enough for identifying all the ABR components. On the other hand, an excessive resolution (greater than 80 samples/decade) provides a too high cut-off frequency in the latency dependent filtering that causes small high frequency oscillations associated to noise. The responses shown in the Figure illustrate that a resolution between 40 and 60 samples/decade is appropriate for the representation of all the AEPs, with a good tradeoff between resolution, dimensionality reduction and noise filtering.

[0154] In order to quantitatively evaluate what is the appropriate resolution, the inventors have compared the responses estimated for different resolutions with those estimated for 70 samples/decade. The responses with $K_{dec} = 70$, used as reference, are marked in black in Figure 16. The SNR associated to the difference between the evaluated and the reference responses has been calculated for different resolutions and are expressed in dB in Table 8. The highest SNR is observed for 60 and 80 samples/decade since these are the estimations obtained in the conditions most similar to the reference. A slight reduction of the SNR is observed for 90, 50 and 40 samples/decade, probably associated to the different amount of noise in the estimated responses associated to the different resolutions. However, a progressive reduction of the SNR is observed for resolutions of 30 samples/decade or smaller. This analysis suggests that resolutions smaller than 30 samples/decade are not enough, while a resolution in the range 40-60 samples/decade are appropriate for representing AEP: these resolutions provide appropriate representation of ABRs, MLRs and CAEPs, optimal dimensionality reduction, optimal reduction of the computational time for matrix-IRSA-red, and optimal latency dependent filtering.

Table 8: Results with real EEGs. Evaluation of the estimated response as a function of the resolution (number of samples/decade). The responses with 70 samples/decade

were used as reference. The SNR measures the ratio of the response to the difference between the compared responses expressed in dB.

		Configuration				
K_{dec}	J_r	1	2	3	4	average
5	19	5.98 dB	7.69 dB	5.10 dB	1.74 dB	**5.13 dB**
10	35	9.76 dB	13.26 dB	12.03 dB	4.74 dB	**9.94 dB**
20	65	10.07 dB	21.32 dB	13.60 dB	14.10 dB	**14.77 dB**
30	92	11.01 dB	23.77 dB	14.37 dB	19.13 dB	**17.07 dB**
40	119	23.08 dB	28.02 dB	20.85 dB	26.46 dB	**24.60 dB**
50	143	20.24 dB	26.57 dB	25.60 dB	25.88 dB	**24.57 dB**
60	167	26.18 dB	26.77 dB	28.12 dB	27.36 dB	**27.11 dB**
70	191	-	-	-	-	-
80	213	29.30 dB	29.27 dB	29.34 dB	26.83 dB	**28.68 dB**
90	236	27.13 dB	27.52 dB	28.20 dB	24.88 dB	**26.93 dB**

Discussion and conclusions

[0155] The inventors proposed and evaluated two optimizations of the IRSA algorithm for AEP estimation. The first one is a matrix implementation of the IRSA algorithm. The second one is a reduction of the dimensionality of the algorithm based on a latency dependent low-pass filtering and decimation.

[0156] The matrix formulation was theoretically demonstrated and experimentally verified to be equivalent to the conventional formulation of the IRSA algorithm. The results provided by both algorithms are identical (with differences associated to the limited numerical resolution). The reduction of the computational load when IRSA is performed with the matrix formulation is due to the fact that conventional IRSA requires, at each iteration, computations involving the whole EEG (typically millions of samples), while matrix-IRSA requires computations involving the whole EEG only at initialization, but at each iteration the computations just involve operations with the length of the response (typically hundreds or thousands of samples). Therefore, the matrix formulation of IRSA is an equivalent implementation that provides a substantial reduction of the computational load.

[0157] The dimensionality reduction allows an additional reduction of the computational load of IRSA. The conditions that the dimensionality reduction should verify in order to make matrix-IRSA and matrix-IRSA-red equivalent have been studied: both are equivalent if the response to be estimated is contained in the reduced representation space. This condition suggests, for example, that a simple dimensionality reduction could be achieved by appropriate low-pass filtering and decimation. The particular nature of the AEP signals suggests a more challenging dimensionality reduction, because the bandwidth of the different components (ABR, MLR, CAEP) decreases as the latency increases. A procedure for designing an orthonormal basis

providing a latency-dependent low-pass filtering and decimation has been proposed. This basis provides a dimensionality reduction that preserves the AEP components when the resolution is large enough. An analysis of the preserved frequency content at each latency suggests that a resolution of 40 samples/decade should be enough (see Table 1). A sensitivity analysis with real EEGs has been performed in order to determine what is an appropriate resolution for optimally representing the AEP responses. A resolution of 40 samples/decade was found to be appropriate, with an evident degradation when a resolution equal or smaller than 30 samples/decade is applied (see Figure 16 and Table 8).

[0158] The dimensionality reduction is not very relevant when a specific portion of the response is considered: for example, if only ABR or only MLR or only CAEP responses are under consideration, after appropriate band-pass filtering and decimation of the EEG, 100 samples would be enough for representing the response, and the latency-dependent lowpass filtering and decimation would provide a dimensionality reduction from 100 samples to 40 samples, which would moderately improve the matrix-IRSA efficiency. However, the proposed dimensionality reduction is more relevant when several AEP portions are simultaneously under consideration. If all ABR, MLR and CAEP are considered, a minimum dimensionality of 10000 samples is required (minimum sampling frequency of 10 kHz for appropriate representation of ABR, minimum window length of 1 s for appropriate representation of CAEP), and the proposed latency-dependent low-pass filtering and decimation with a resolution of 40 samples/decade reduces the dimensionality from 10000 to 110, with the subsequent impact in the reduction of the computational load of the IRSA algorithm.

[0159] In this sense, the proposed implementation of matrix-IRSA in a reduced representation space provides a framework in which the whole AEP response (including ABR, MLR and CAEP) can be estimated. Without this dimensionality reduction, matrix-IRSA would require a prohibitive execution time to reach convergence, while matrix-IRSA performed in the reduced representation space provides the results in few seconds. This opens the possibility of a global analysis of the different components of the auditory response in a more comprehensive representation. The representation of the response including several decades and using a logarithmically scaled time axis (as in Figures 14 and 15) allows a compact representation of all the evoked potentials.

[0160] In addition to the optimization of the computational time of matrix-IRSA, the dimensionality reduction provides a latency dependent low-pass filtering that contributes to appropriately remove the noise affecting the estimated response. The transformation provided by the orthogonal basis preserves those components in which the response is expected to contain energy and removes those components in which the response is not expected,

cancelling the noise affecting the removed components. This improves the quality of the estimated responses.

[0161] Even though the resulting responses provided by matrix-IRSA followed by projection and matrix-IRSA-red are very similar, there are slight differences between both results. These differences are observed for simulations as well as for real EEGs. The differences are associated to the noise affecting the EEG, and the accidental correlations of the noise and the stimulation signal. The noise and the stimulation should be uncorrelated signals, but due to specific values of both signals and their limited length, they are not completely uncorrelated, and the estimation of the response from a noisy EEG contains energy out of the reduced representation space that propagates a small error according to equation (52). The impact of this error can be reduced by increasing the resolution in the dimensionality reduction. Using an increased resolution has three effects: it increases the dimensionality of the reduced representation space (the improvement in the execution time would be smaller); it reduces the error propagated by the term V1RsV T 2 ^xv2 i in equation (52); and it reduces the efficiency of noise reduction associated to the latency-dependent low-pass filtering (because of the use of an excessive resolution). A practical alternative to avoid the last effect would be to perform matrix-IRSA-red with a high resolution (for example $K_{dec} = 100$ samples/dec) in order to reduce the error, and after convergence, filtering the resulting response by applying $V_r^T V_r$ defined with the appropriate resolution (for example $K_{dec} = 40$ samples/dec). In any case, the sensitivity analysis reveals that 40 samples/dec is an appropriate resolution.

[0162] The suggested dimensionality reduction, in addition to the advantages previously described (reduction of the computational load of matrix-IRSA algorithm and latency-dependent low-pass filtering for noise reduction) provide a compact representation of the estimated responses, particularly useful if the whole response is under consideration. An AEP response, with a duration of 10000 samples, can be represented using only 100 or 200 samples. The compact representation is not a limitation, since the basis can be applied to recover the standard representation from the compact one (i.e. to transform the response from the reduced representation space to the complete representation space). This way, if the $J_r \times J$ transformation V_r is stored (or the procedure to obtain V_r is clearly defined), the representation of each response does not require J but only J_r samples. This can be exploited in different contexts: to reduce the size of a database of responses, for optimally transmitting the responses, or when using the responses in automatic classification, distance measurements among responses, automatic quality assessment, automatic detection of peaks or other artificial intelligence applications (a substantial dimensionality reduction preserving the relevant information is always useful as a pre-processing procedure in all these examples).

[0163] Regarding the impact of the proposed optimizations in the execution time, for the evaluation with the described experiment using real EEGs, completing 10000 iterations of IRSA required 28h:30m with the conventional implementation, 1h:36m with the matrix implementation and 1m:17s with the matrix implementation in the reduced representation space (see Table 7). Taking into account that the time involved in the recording session was 1h:11', the execution time required for conventional IRSA make this implementation unacceptable for a practical application. In spite of the time reduction provided by the matrix implementation, the processing time is larger than the acquisition time. In contrast, the matrix-IRSA performed in the reduced representation space required just a few seconds (most of them devoted to initialization) to complete the algorithm, providing a useful applicability for clinical or research applications.

Computational optimization of the matrix-IRSA method for recording evoked potentials: Fast implementation for complete and reduced representation spaces

[0164] The inventors have also studied the computational load of the matrix-IRSA algorithm (for both complete and reduced representations spaces). In the case of matrix-IRSA in the complete representation space, an important part of the computation is associated to the calculation of the cross correlation between the EEG and the stimulation signal (both signals with a long duration, typically several minutes and millions of samples) and the calculation of the autocorrelation of the stimulation signal. However, the most important factor affecting the requested memory and the execution time is a matrix product (involving the autocorrelation matrix of the stimulation signal and the response estimated at each iteration), where the size of the matrix is $J \times J$ (being J the length of the response). For AEP estimations including only a portion of the evoked response (i.e. only ABR, or only MLR, or only CAEP) the length of the response is several hundreds of samples and the matrix product is not an important problem (neither for memory nor execution time).

[0165] However, if the response includes the contributions of the whole auditory pathway (including ABR, MLR and CAEP simultaneously), the length of the response could be several thousands of samples (in the experiments with real EEGs discussed earlier) , the response length is J =16384 samples), and the matrix product limits the computational efficiency of the matrix-IRSA algorithm because of both, memory requirements (the matrix requi requires 268 millions of numbers, and more than 2 GB of memory for double precision representation) and execution time.

[0166] In the case of matrix-IRSA performed in the reduced representation space, the computation of the correlations during the initialization are affected by the same problem as

in the complete space. The initialization also requires the calculation of the autocorrelation matrix (of the stimulation signal) in the reduced space (i.e. transforming a $J \times J$ matrix into a $J_r \times J_r$ matrix where J_r is the dimensionality of the reduced space), with the subsequent problems of memory and execution time requirements. However, after the initialization, the matrix product performed at each iteration involves a $J_r \times J_r$ matrix (where J_r is typically between 100 and 200) and the iterations do not involve restrictive memory or execution time requirements.

[0167] Focusing the attention in the most critical computations of the matrix-IRSA algorithms (both in the complete and the reduced spaces), the inventors proposed optimizations to improve their efficiency. Two aspects are exploited in order to implement the optimizations. With respect to the computation of the correlations, the fact that they involve the stimulation signal (which is composed of a relatively small number of unitary isolated impulses) allows to simplify these calculations. With respect to the matrix products (involving the autocorrelation matrix of the stimulation sequence when matrix-IRSA is performed in the complete space, or the autocorrelation matrix transformed to the reduced space in the other case), the fact that the involved matrix is symmetric Toeplitz allows to store it using just J values (instead of $J \times J$ values) as well as to obtain the matrix product with a convolution operation, simplifying the calculation and reducing the memory requirements. Additionally, these optimizations do not involve any approximation, and they provide identical results to those from the previous versions of matrix-IRSA (both in the complete or in the reduced representation spaces).

[0168] The proposed optimizations of the matrix-IRSA algorithm and the inventors evaluated the improvements (in terms of computational efficiency) obtained when the optimized matrix-IRSA algorithms are applied to obtain AEP responses using real EEGs are now described.

METHODS

Optimization of matrix-IRSA in the complete representation space

The matrix-IRSA algorithm

[0169] The matrix-IRSA algorithm models the EEG $y(n)$ as a convolutional process involving the response to be estimated $x(n)$, the stimulation signal $s(n)$, and the noise $n_0(n)$:

$$y(n) = s(n) * x(n) + n_0(n) \tag{71}$$

where n is the index of the samples (with $n = 0; \ldots ; N-1$), N the length of the EEG, the response $x(n)$ is assumed to be null for $n > J$ (J is the length of the response), and the

asterisk $(*)$ represents convolution. The stimulation signal $s(n)$ contains K events at the samples m_k and can therefore be written as:

$$s(n) = \sum_{k=0}^{K-1} \delta(n - m_k) \tag{72}$$

where $\delta(n)$ is the unitary impulse at $n = 0$. With these definitions, the matrix-IRSA algorithm is the following

1. Initialization:

$$\hat{x}_0(j) = 0 \qquad z_0(j) = \frac{1}{K} \sum_{k=0}^{K-1} y(j + m_k) \qquad r_s(j) = \sum_{n=j}^{N-J} s(n)s(n-j) \qquad \forall j \in \{0, \ldots, J-1\} \tag{73}$$

$$R_s(j_1, j_2) = \frac{1}{K} r_s(|j_1 - j_2|) \qquad \forall j_1, j_2 \in \{0, \ldots, J-1\} \qquad \{\hat{\mathbf{x}}_0, \mathbf{z}_0, R_s\} \tag{74}$$

2. Response updating:

$$\hat{\mathbf{x}}_i = \hat{\mathbf{x}}_{i-1} + \alpha \mathbf{z}_{i-1} \tag{75}$$

3. Averaged-residual estimation:

$$\mathbf{z}_i = \mathbf{z}_0 - R_s \hat{\mathbf{x}}_i \tag{76}$$

4. Steps 2 and 3 are repeated until convergence.

[0170] The computational requirements of this algorithm are conditioned by the number of samples in the EEG (N) and the length of the response (J). Two operations involve signals with the length of the EEG during the initialization: the initialization of the averaged residual $z_0(j)$ and the estimation of the autocorrelation of the stimulation signal $r_s(j)$. Regarding the length of the response, the most critical point concerning the computational requirements is the management of the matrix R_s (that requires to store $J \times J$ values) at initialization, and the matrix product $R_s \hat{\mathbf{x}}_i$ at each iteration.

Cross-correlation of the EEG and the stimulation signal

[0171] Interestingly, the averaged residual $z_0(j)$ can be calculated as the normalized cross-correlation between the EEG $y(n)$ and the stimulation signal $s(n)$:

$$\frac{1}{K}\mathrm{xcorr}_{[y,s]}(j) = \frac{1}{K}\sum_{n=j}^{N-j} y(n)s(n-j) = \frac{1}{K}\sum_{n=j}^{N-j} y(n)\sum_{k=0}^{K-1}\delta(n-j-m_k) = \frac{1}{K}\sum_{k=0}^{K-1} y(j+m_k) = z_0(j) \tag{77}$$

[0172] This equivalence is associated to the fact that the stimulation signal consists in a number of unitary impulses.

[0173] Even though both procedures for computing $z_0(j)$ (using the cross-correlation or the sum for each individual impulse) are equivalent, the best procedure in terms of computational load depends on the length of the involved signals (N and J) and the number of stimuli (K) in the stimulation signal. In general, if the number of impulses in the stimulation sequence was very large, the calculation based in cross-correlation would be more efficient, because internally, cross-correlation is calculated with Fast Fourier Transform (FFT) and Inverse Fast Fourier Transform (IFFT) that are efficient algorithms in spite of the length of the involved signals. However, if the number of stimuli is small, computing $z_0(j)$ as a sum for each impulse is more efficient than using cross-correlation. For typical values of the EG length (N about several millions of samples), response length (J about several hundreds or a few thousands of samples) and number of stimuli (K about several thousands of stimuli), the calculation of $z_0(j)$ as a sum for all the stimuli is more efficient.

Autocorrelation of the stimulation signal

[0174] Similarly, the autocorrelation of the stimulation signal can be calculated either as a cross-correlation or as a sum for all the stimuli:

$$r_s(j) = \mathrm{xcorr}_{[s,s]}(j) = \sum_{n=j}^{N-j} s(n)s(n-j) = \sum_{k=0}^{K-1} s(j+m_k) \tag{78}$$

(the demonstration of this equivalence is similar to that in equation (77) where the signal y(n) is substituted by s(n)). Again, the computational efficiency can easily be improved by calculating the autocorrelation as a sum for all the stimuli.

Optimization of the matrix product

[0175] The most memory and time consuming part of the matrix-IRSA algorithm is the matrix product $R_s\hat{\mathbf{x}}_i$. This matrix product involves the $J \times J$ matrix R_s and the vector $\hat{\mathbf{x}}_i$ with a length of J samples:

$$\begin{pmatrix} p_0 \\ p_1 \\ p_2 \\ \vdots \\ p_{J-1} \end{pmatrix} = R_s\hat{\mathbf{x}}_i = \frac{1}{K}\begin{pmatrix} r_s(0) & r_s(1) & r_s(2) & \cdots & r_s(J-1) \\ r_s(1) & r_s(0) & r_s(1) & \cdots & r_s(J-2) \\ r_s(2) & r_s(1) & r_s(0) & \cdots & r_s(J-3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_s(J-1) & r_s(J-2) & r_s(J-3) & \cdots & r_s(0) \end{pmatrix}\begin{pmatrix} x_i(0) \\ x_i(1) \\ x_i(2) \\ \vdots \\ x_i(J-1) \end{pmatrix} \tag{79}$$

[0176] The matrix product requires the storage of a $J \times J$ matrix, J^2 products and J^2 sums, with the corresponding memory and execution time requirements. Taking into account the symmetric Toeplitz nature of the autocorrelation matrix R_s, it is a very redundant matrix that can be stored with just J values (instead of J^2), and the product can be written as:

$$p_j = \frac{1}{K} \sum_{j'=0}^{J-1} x_i(j') r_s(|j - j'|) \qquad \forall j = 0, \ldots, J-1 \tag{80}$$

and if we define an extended-normalized autocorrelation signal $r_{s.ext}$ as:

$$r_{s.ext}(j) \equiv \frac{1}{K} r_s(|j|) \qquad \forall j = -(J-1), \ldots, J-1 \tag{81}$$

the product can equivalently be written as:

$$p_j = \sum_{j'=0}^{J-1} x_i(j') r_{s.ext}(j - j') \qquad \forall j = 0, \ldots, J-1 \tag{82}$$

that is, the matrix product can equivalently be calculated as a convolution:

$$R_s \dot{\mathbf{x}}_i = \dot{\mathbf{x}}_i * \mathbf{r}_{s.ext} \tag{83}$$

[0177] Regarding the memory requirements, the product based on convolution is more efficient since a $2J - 1$ vector (instead of a $J \times J$ matrix) is used. Regarding the number of operations involved, explicit computation of the convolution product would require J^2 products and J^2 sums, as in the case of the matrix product. However, since the convolution is a built-in function optimized in MatLab and Octave, the calculations based on convolutions are faster than explicit products and sums. Additionally, a FFT-based convolution could be implemented. The MatLab and Octave convolution function do not perform FFT-based convolutions, but calculations are optimized taking into account only the non-null samples of the vectors to be convolved. Under specific circumstances, a FFT-based convolution would be faster.

Fast algorithm for matrix-IRSA

[0178] The algorithm for matrix-IRSA in the complete representation space including the proposed optimizations is as follows:

1. Initialization:

$$\hat{x}_0(j) = 0 \qquad z_0(j) = \frac{1}{K}\sum_{k=0}^{K-1} y(j+m_k) \qquad r_s(j) = \sum_{k=0}^{K-1} s(j+m_k) \qquad \forall j \in \{0,\ldots,J-1\} \tag{84}$$

$$r_{s,ext}(j) = \frac{1}{K} r_s(|j|) \qquad \forall j \in \{-(J-1),\ldots,J-1\} \qquad \{\hat{\mathbf{x}}_0, \mathbf{z}_0, \mathbf{r}_{s,ext}\} \tag{85}$$

2. Response updating:

$$\hat{\mathbf{x}}_i = \hat{\mathbf{x}}_{i-1} + \alpha \mathbf{z}_{i-1} \tag{86}$$

3. Averaged-residual estimation:

$$\mathbf{z}_i = \mathbf{z}_0 - \hat{\mathbf{x}}_i * \mathbf{r}_{s,ext} \tag{87}$$

4. Steps 2 and 3 are repeated until convergence.

Optimization of matrix-IRSA in the reduced representation space

The matrix-IRSA algorithm in a reduced representation space

[0179] The dimensionality reduction for the matrix-IRSA algorithm is described with a $J_r \times J$ incomplete orthonormal matrix V_r, with $J_r < J$, verifying that $x = V_r^T V_r \mathbf{x}$ (i.e. verifying that the expected AEP responses $\mathbf{x}$ are included in the reduced representation space), where V_r^T represents the transpose of V_r. Given the matrix V_r describing the dimensionality reduction, the matrix-IRSA algorithm in the reduced representation space is the following:

1. Initialization:

$$\hat{x}_0(j) = 0 \qquad z_0(j) = \frac{1}{K}\sum_{k=0}^{K-1} y(j+m_k) \qquad r_s(j) = \sum_{n=0}^{N-j} s(n)s(n-j) \qquad \{\hat{\mathbf{x}}_0, \mathbf{z}_0, R_s\} \tag{88}$$

$$\mathbf{x}_0^{v_r} = V_r \hat{\mathbf{x}}_0 = 0 \qquad \mathbf{z}_0^{v_r} = V_r \mathbf{z}_0 \qquad R_s^{v_r} = V_r R_s V_r^T \qquad \{\hat{\mathbf{x}}_0^{v_r}, \mathbf{z}_0^{v_r}, R_s^{v_r}\} \tag{89}$$

2. Response updating:

$$\hat{\mathbf{x}}_i^{v_r} = \hat{\mathbf{x}}_{i-1}^{v_r} + \alpha \mathbf{z}_{i-1}^{v_r} \tag{90}$$

3. Averaged-residual estimation:

$$\mathbf{z}_i^{v_r} = \mathbf{z}_0^{v_r} - R_s^{v_r} \mathbf{x}_i^{v_r} \tag{91}$$

4. Steps 2 and 3 are repeated until convergence.

5. The recovered evoked response after convergence is transformed back to the original representation space:

$$\hat{\mathbf{x}}_i = V_r^T \hat{\mathbf{x}}_i^{v_r} \tag{92}$$

[0180] As in the case of the matrix-IRSA performed in the complete representation space, there are two operations involving signals with the length of the EEG during the initialization: the initialization of the averaged residual $z_0(j)$ and the estimation of the autocorrelation of the stimulation signal $r_s(j)$. Also during the initialization, the transformation of the matrix R_s to the reduced representation space include matrix products involving the $J \times J$ matrix R_s. However, at each iteration, the size of the involved vectors and matrices is J_r and $J_r \times J_r$ respectively, since these calculations are performed in the reduced representation space. Therefore, the most critical point concerning the computational requirements in this algorithm is the manipulation of the $J \times J$ matrix R_s in order to obtain the autocorrelation matrix in the reduced representation space $R_s^{v_r}$.

Optimization of the cross-correlations

[0181] As in the case of the matrix-IRSA performed in the complete representation space, the averaged residual $z_0(j)$ can equivalently be calculated with a cross-correlation between the EEG $y(n)$ and the stimulation signal $s(n)$ or with a sum for each individual impulse:

$$z_0(j) = \frac{1}{K}\sum_{k=0}^{K-1} y(j+m_k) = \frac{1}{K}\sum_{n=j}^{N-j} y(n)s(n-j) = \frac{1}{K}\mathrm{xcorr}_{[y,s]}(j) \tag{93}$$

and for typical values of N (length of the EEG), J (length of the response) and K (number of stimuli), the implementation based on the sum for each individual impulse is more efficient.

[0182] Similarly, the autocorrelation of the stimulation signal can be calculated as an autocorrelation or as a sum for each individual impulse in the stimulation signal:

$$r_s(j) = \mathrm{xcorr}_{[s,s]}(j) = \sum_{n=j}^{N-j} s(n)s(n-j) = \sum_{k=0}^{K-1} s(j+m_k) \tag{94}$$

and the implementation as a sum for each individual impulse is usually more efficient.

Optimization of the matrix transformation

[0183] The most memory and time consuming part of the matrix-IRSA algorithm, when it is performed in the reduced representation space, is the matrix product required to obtain the transformed autocorrelation matrix:

since it involves a matrix product of a $J \times J$ matrix with a $J \times J_r$ matrix ($R_s V_r^T$), and another matrix product of involving a $J_r \times J$ matrix and a $J \times J_r$ matrix ($V_r(R_s V_r^T)$), it implies the memory management for a $J \times J$ matrix as well as $J^2 \cdot J_r$ products and sums for the first matrix product, and $J_r^2 \cdot J$ products and sums for the second matrix product (being the first matrix product the most critical one, because $J_r \times J$).

[0184] The matrix providing the dimensionality reduction V_r is composed of J_r row vectors of an orthonormal basis of functions:

$$V_r = \begin{pmatrix} \mathbf{v}_0^T \\ \mathbf{v}_1^T \\ \vdots \\ \mathbf{v}_{J_r-1}^T \end{pmatrix} \tag{96}$$

and its transpose V_r^T is composed of J_r column vectors:

$$V_r = \begin{pmatrix} \mathbf{v}_0 & \mathbf{v}_1 & \cdots & \mathbf{v}_{J_r-1} \end{pmatrix} \tag{97}$$

and therefore, the product $R_s V_r^T$ can be decomposed as J_r products of R_s with each one of the vectors in the basis:

$$A \equiv R_s V_r^T \qquad A = \begin{pmatrix} \mathbf{a}_0 & \mathbf{a}_1 & \cdots & \mathbf{a}_{J_r-1} \end{pmatrix} \qquad \mathbf{a}_j = R_s \mathbf{v}_j \quad \forall j = 0 \ldots J_r - 1 \tag{98}$$

[0185] Taking into account that R_s is a symmetric Toeplitz matrix, as in the case of the complete representation space, the autocorrelation matrix can be completely represented with J values of the autocorrelation of the stimulation signal $R_s(j)$, and the matrix products can be performed with convolutions:

$$a_j(j') = \sum_{j''=0}^{J-1} R_s(j', j'') v_j(j'') = \frac{1}{K} \sum_{j''=0}^{J-1} r_s(|j' - j''|) v_j(j'') = \sum_{j''=0}^{J-1} r_{s,ext}(j' - j'') v_j(j'') \tag{99}$$

with $j' = 0; \ldots ; J - 1$ and $j = 0; \ldots ; Jr - 1$. Therefore, the vectors $\mathbf{a}_j$ in the resulting matrix A can be obtained with J_r convolutions:

$$\mathbf{a}_j = \mathbf{r}_{s,ext} * \mathbf{v}_j \qquad \forall j = 0, \ldots, J_r - 1 \tag{100}$$

with the subsequent reduction in memory and execution time requirements, as in the case of the algorithm performed in the complete representation space. When the vectors a_j are calculated, they can be arranged in the $J \times J_r$ matrix A, and the transformed autocorrelation matrix can be obtained with the matrix product:

$$R_s^{v_r} = V_r A \tag{101}$$

which is a matrix operation involving matrices of $J_r \times J$ and $J \times J_r$, and with reasonable memory and execution time requirements if J_r is small.

1. Initialization:

$$\hat{x}_0(j) = 0 \qquad z_0(j) = \frac{1}{K}\sum_{k=0}^{K-1} y(j+m_k) \qquad r_s(j) = \sum_{k=0}^{K-1} s(j+m_k) \qquad \forall j \in \{0,\ldots,J-1\} \tag{102}$$

$$r_{s,ext}(j) = \frac{1}{K} r_s(|j|) \qquad \forall j \in \{-(J-1),\ldots,J-1\} \qquad \{\hat{\mathbf{x}}_0, \mathbf{z}_0, \mathbf{r}_{s,ext}\} \tag{103}$$

$$\mathbf{a}_j = \mathbf{r}_{s,ext} * \mathbf{v}_j \qquad A(j',j) = a_j(j') \qquad \forall j = 0,\ldots,J_r-1 \quad \forall j' = 0,\ldots,J-1 \tag{104}$$

$$\hat{\mathbf{x}}_0^{v_r} = V_r \hat{\mathbf{x}}_0 = 0 \qquad \mathbf{z}_0^{v_r} = V_r \mathbf{z}_0 \qquad R_s^{v_r} = V_r A \qquad \{\hat{\mathbf{x}}_0^{v_r}, \mathbf{z}_0^{v_r}, R_s^{v_r}\} \tag{105}$$

2. Response updating:

$$\hat{\mathbf{x}}_i^{v_r} = \hat{\mathbf{x}}_{i-1}^{v_r} + \alpha \mathbf{z}_{i-1}^{v_r} \tag{106}$$

3. Averaged-residual estimation:

$$\mathbf{z}_i^{v_r} = \mathbf{z}_0^{v_r} - R_s^{v_r} \hat{\mathbf{x}}_i^{v_r} \tag{107}$$

4. Steps 2 and 3 are repeated until convergence.

5. The recovered evoked response after convergence is transformed back to the original representation space:

$$\hat{\mathbf{x}}_i = V_r^T \hat{\mathbf{x}}_i^{v_r} \tag{108}$$

Experimental results

[0186] The inventors have proposed optimizations for the matrix-IRSA algorithm performed in the complete representation space, as well as optimizations for the matrix-IRSA algorithm performed in a reduced representation space. The proposed optimizations involve the calculation of correlations during the initialization and some matrix products. In order to evaluate the proposed optimizations, the inventors designed experiments to (a) show that the results are identical with the corresponding algorithms before and after the optimizations and (b) evaluate the reduction in the computational requirements (execution time and memory

requirements). The evaluation of the proposed optimizations is based on real EEGs registered during an evoked potential recording session.

AEP recording session

[0187] The EEGs used in this evaluation were the same as used earlier. These EEGs were recorded in an AEP recording session in which 4 different stimulation rates were configured. The stimulation signal was prepared using a uniform distribution of ISI between 500 and 800 ms (for an average stimulation rate of 1.53 Hz), between 300 and 600 ms (for 2.22 Hz), between 100 and 300 ms (for 5.00 Hz) and between 30 and 100 ms (for 15.38 Hz). The number of stimuli used for each configuration was increased with the stimulation rate (from 1500 stimuli at the slowest rate to 20000 stimuli at the fastest rate). The stimulation consisted in a sequence of rarefaction clicks resented at the instants defined by the stimulation sequence. The clicks were delivered diotically through ER-3A insert earphones at 60 dB HL. The recording electrodes were located at the upper forehead (Fz, active), at the mastoids (Tp9 and Tp10, references 1 and 2) and at the middle forehead (Fpz, ground). The EEGs were recorded using a BioSemi instrumentation pre-amplifier (BioSemi V.B., Amsterdam, Netherlands), with a [1-3000] Hz bandwidth and a sampling frequency of 16384 samples per second. The [Fz-Tp9] and [Fz-Tp10] were averaged to obtain a single EEG. Eye-blink artifacts were suppressed with the iterative template matching and suppression (ITMS), an algorithm that detects, models and suppresses blink-artefacts from a single-channel EEG (Valderrama et al, 2018 – Ref 19).

Table 9: Configuration of the EEG recording session

Configuration	ISI	aver. stim. rate	K (stimuli)	EEG length (seconds)	EEG length (samples)
1	500 - 800 ms	1.53 Hz	1500	990	$16.22 \cdot 10^6$
2	300 - 600 ms	2.22 Hz	2000	924	$15.14 \cdot 10^6$
3	100 - 300 ms	5.00 Hz	5000	1021	$16.72 \cdot 10^6$
4	30 - 100 ms	15.38 Hz	20000	1322	$21.66 \cdot 10^6$

[0188] Table 9 summarizes the configurations involved in this EEG recording session. Since the objective of this experiment is the evaluation of the optimization procedures proposed for the matrix-IRSA algorithm, only one subject (male, 33 years) was considered in these experiments.

Table 10: Comparison of the responses estimated with different versions of the matrix-IRSA algorithm. The ratio of the response energy (Er) to the energy of the difference between the compared responses (Ed) is expressed in dB.

Compared versions	Configuration				Energy ratio E_r/E_d
		ISI (ms)	stimuli	EEG duration	
matrix-IRSA vs. matrix-IRSA-fast	1	500-800	1500	990 s	294.47 dB
	2	300-600	2000	924 s	298.57 dB
	3	100-300	5000	1021 s	297.31 dB
	4	30-100	20000	1322 s	292.42 dB
	Average				**295.69 dB**
matrix-IRSA-red vs. matrix-IRSA-red-fast	1	500-800	1500	990 s	303.22 dB
	2	300-600	2000	924 s	292.83 dB
	3	100-300	5000	1021 s	287.57 dB
	4	30-100	20000	1322 s	279.26 dB
	Average				**290.72 dB**

Equivalence of optimized algorithms

[0189] Figure 17 compares the responses provided by the original and the optimized matrix-IRSA algorithms after 10000 iterations using a convergence parameter$\alpha = 0.02$. The different waves of the evoked responses are marked for the first recording configuration. The responses provided by the matrix-IRSA in the complete representation space with and without the proposed optimizations (algorithms matrix-IRSA and matrix-IRSA-fast) are compared in the left panel of the Figure 17. The plots in blue are the responses estimated with the original matrix-IRSA algorithm while the plots in red are those estimated with the optimized version. In the right panel, the responses estimated with the matrix-IRSA in the reduced representation space are compared (matrix-IRSA-red, in blue, for the original algorithm, matrix-IRSA-red-fast, in red, for the optimized algorithm). As can be observed in these plots, the responses provided by the original and the optimized versions are essentially identical.

[0190] The difference between the responses provided by the original and the optimized algorithms is represented in Figure 18 (left panel for matrix-IRSA in the complete representation space, right panel for matrix-IRSA in the reduced representation space). This figure illustrates the small difference in the responses provided by the original and the optimized algorithms. The amplitude of the responses is around $1\ \mu$V, while the amplitude of the difference is around $10^{-14}\mu$V.

[0191] In order to evaluate the difference between the original and the optimized algorithms the signal to noise ratio (SNR) associated to the difference (i.e. the ratio of the energy of the responses to the energy of the difference) was measured for the different responses and with the difference methods. Table 10 shows the SNR associated to the comparison of the responses provided by the different versions. All the SNRs are close to 300 dB, which clearly demonstrates that the results provided by the original and the optimized version are identical

in practice. The small differences are associated to the limited precision in the numerical representation and the slightly different numerical procedures involved in some computations (sums and products vs. FFTs and IFFTs for computing correlations, for example).

Computational efficiency

[0192] The computational efficiency provided by the proposed optimizations has been evaluated in terms of the reduction of memory requirements and execution time. The computations were performed in a desktop computer with an Intel-Core i7-3770 CPU, 3.40 GHz, 8.00 GB RAM. Table 11 shows the execution time (time required for the initialization, for each iteration and for the complete algorithm) and memory requirements with the different versions of the matrix-IRSA method (in the complete representation space without and with optimizations, and in the reduced representation space without and with optimizations).

Table 11: Execution time and memory requirements with different versions of the matrix-IRSA method.

Procedure	iterations	Configuration				Execution time			Memory requirements
			ISI (ms)	stimuli	EEG dur.	t_{ini}	t_{iter}	t_{tot}	
Mat-IRSA original	10000	1	500-800	1500	990 s	16.17 s	0.138 s	1399.2 s	2.41 GB
		2	300-600	2000	924 s	15.61 s	0.139 s	1407.2 s	2.39 GB
		3	100-300	5000	1021 s	15.88 s	0.141 s	1427.2 s	2.42 GB
		4	30-100	20000	1322 s	37.68 s	0.152 s	1560.7 s	2.49 GB
						Total:		**5794.3 s**	
Mat-IRSA fast	10000	1	500-800	1500	990 s	1.00 s	2.49 ms	25.9 s	260.3 MB
		2	300-600	2000	924 s	1.19 s	5.08 ms	52.0 s	243.0 MB
		3	100-300	5000	1021 s	2.64 s	30.4 ms	307.0 s	268.5 MB
		4	30-100	20000	1322 s	9.36 s	30.0 ms	309.4 s	347.7 MB
						Total:		**694.3 s**	
Mat-IRSA-red original	10000	1	500-800	1500	990 s	17.25 s	5.24 μs	17.3 s	2.42 GB
		2	300-600	2000	924 s	17.12 s	6.13 μs	17.2 s	2.40 GB
		3	100-300	5000	1021 s	16.62 s	6.51 μs	16.7 s	2.43 GB
		4	30-100	20000	1322 s	25.68 s	5.99 μs	25.7 s	2.51 GB
						Total:		**76.9 s**	
Mat-IRSA-red fast	10000	1	500-800	1500	990 s	1.20 s	6.61 μs	1.26 s	291.3 MB
		2	300-600	2000	924 s	1.48 s	4.71 μs	1.53 s	274.0 MB
		3	100-300	5000	1021 s	3.46 s	4.62 μs	3.5 s	299.4 MB
		4	30-100	20000	1322 s	10.14 s	5.17 μs	10.2 s	378.6 MB
						Total:		**16.5 s**	

[0193] Regarding the memory requirements, matrix-IRSA in both, the complete or the reduced representation space requires memory allocation for the $J \times J$ autocorrelation matrixR_s when the optimization are not included. Since $J = 16384$ and each number is represented in double precision format (8 Bytes), the allocation of the R_s matrix requires $2.1475 \cdot 10^9$ Bytes, i.e. more than 2 GB. The implementation of the matrix products as convolutions requires the allocation of only J numbers, and therefore, the memory requirements are reduced, due to the optimization, from more than 2 GB to less than 400 MB.

[0194] The execution time is also substantially reduced in the optimized versions with respect to the corresponding original versions. In the case of matrix-IRSA performed in the complete representation space, the time required for estimating the 4 responses was decreased from 5794.3 seconds to 694.3 seconds (i.e. the execution time is reduced in a factor 8.3). Similarly, in the case of matrix-IRSA performed in the reduced representation space, the time for estimating the 4 responses was reduced from 76.9 seconds to 16.5 seconds (i.e. the execution time is reduced in a factor 4.7).

[0195] Identifying what is the specific mechanism involved in the execution time reduction is difficult in a multi-task computer. The allocation of memory (in the case of the non-optimized versions) requires some time depending on the amount of memory installed in the system and the programs that are running simultaneously. The operations involving variables with so many elements (like the matrix product with R_s) also requires the management of large areas of memory that takes time to the operation system. The management of variables requiring 2 GB is a problem for the execution of the algorithm even in a system with 8 GB RAM. The memory allocation problem would be dramatic for systems with a smaller RAM memory. Finally, the execution time reduction associated to the reorganization of the calculations is difficult to be evaluated. The matrix product $R_s\hat{\mathbf{x}}_i$ requires J^2 products and sums either if it is computed as a matrix product or if it is computed as a convolution. However, some execution time reduction is obtained from internal optimizations of the MatLab convolution function (for example, it is based in a built-in function, only the non-null values in the vectors to be convolved are considered in order to improve efficiency, etc.).

Additional optimisations

[0196] An additional optimization could be included in the matrix-IRSA algorithms by computing convolutions in the Fourier domain. Default MatLab implementation of the convolution is done by explicit products and sums of the samples (where some execution time can be saved by identifying the null samples in the signals to be convolved). In general, for signals with length N the cost of the convolution is $O(N^2)$. A more efficient convolution can be performed in the FFT domain (it requires the FFT of the signals to be convolved, a product of the transformed signals and a IFFT of the product). In that case the cost of the convolution is $O(N\log_2(N))$. For small J (or for a large number of null values in the signals to be convolved) the standard convolution algorithm is more efficient, while for large J, the FFT-based convolution is more efficient.

[0197] In our matrix-IRSA algorithms, convolutions are related to the matrix products involving the symmetric Toeplitz matrix R_s, and therefore, the signal to be convolved is $r_{s:ext}$, i.e. an

extended and normalized version of r_s. This is a pair signal (it is the normalized autocorrelation function of the stimulation sequence), and therefore convolution and cross-correlation with this signal is equivalent.

[0198] Table 12 compares the execution time for different implementations of the matrix-IRSA algorithm (either in the complete or in the reduced representation spaces). The execution times are compared for the original implementation, for the fast version (using conventional convolutions for the matrix product) and for the fast version implementing FFTbased convolutions (see last three columns). By comparing the last two cases, the utility of the FFT-based convolution can be discussed: in some cases the FFT convolution provides a significant improvement (matrix-IRSA in the complete representation space for experiment configurations 2, 3 and 4), while in others the improvement is small (configuration 1in complete representation space or configurations 3 and 4 in reduced representation space) or even there is an increment in the execution time (configurations 1 and 2 in the reduced representation space). This is related to the number of null samples in the signals to be convolved (that changes depending on the stimulation sequence and therefore depends on the experiment configuration) and also depends on the number of samples in the response J (in all the experiments in this application $J = 16384$, but obviously FFT-based convolutions are preferable as J increases).

Table 12: Execution time and memory requirements with different versions of the matrix-IRSA method

Repres. space	Number of iter.	Configuration				Execution time		
			ISI (ms)	stimuli	EEG dur.	original	fast	FFT-conv
Complete	10000	1	500-800	1500	990 s	1399.2 s	25.9 s	22.3 s
		2	300-600	2000	924 s	1407.2 s	52.0 s	21.6 s
		3	100-300	5000	1021 s	1427.2 s	307.0 s	23.6 s
		4	30-100	20000	1322 s	1560.7 s	309.4 s	30.3 s
		Total:				**5794.3 s**	**694.3 s**	**97.8 s**
Reduced	10000	1	500-800	1500	990 s	17.3 s	1.26 s	1.36 s
		2	300-600	2000	924 s	17.2 s	1.53 s	1.57 s
		3	100-300	5000	1021 s	16.7 s	3.5 s	3.0 s
		4	30-100	20000	1322 s	25.7 s	10.2 s	9.7 s
		Total:				**76.9 s**	**16.5 s**	**15.6 s**

[0199] In this application, some computational optimizations of the algorithm matrix-IRSA are proposed. The optimizations include:

a fast procedure for computing correlations at the initialization (that makes use of the fact that the stimulation signal is null for most of the samples, and therefore correlations involving this signal can be calculated as sum for each stimulus) and

a procedure for computing matrix products as a convolution (that makes use of the fact that the involved matrix is Toeplitz, and therefore the product does not requires to store the

whole matrix but only a signal containing the first column and the first row, with the subsequent reduction of memory and execution time requirements).

[0200] The application describes the motivation and formulation of the optimizations, and provides updated functions for matrix- IRSA (in the complete and the reduced representation spaces) implementing these optimizations.

[0201] This application also includes an evaluation of the proposed optimizations, in terms of (a) equivalence of the estimated responses and (b) computational efficiency, using real EEGs registered in an AEP recording session. The analysis of the responses estimated with and without the proposed optimizations reveals that the algorithms with and without the optimizations are equivalent (the estimated responses are identical in practice, with a SNR associated to the difference around 300 dB). The analysis of computational requirements reveals a substantial reduction of the required memory (from more than 2 GB to less than 400 MB in the performed experiments) as well as a substantial reduction of the execution time (in a factor 8.3 when matrix-IRSA is performed in the complete representation space, and in a factor 4.7 when it is performed in the reduced representation space).

[0202] It will be understood to persons skilled in the art of the invention that modifications may be made without departing from the spirit and scope of the invention. The embodiments and/or examples as described herein are therefore to be considered as illustrative and not restrictive.

List of references

[0203] The disclosures of each of the below references are hereby incorporated in their entirety into the present specification:

Articles

Ref 1. Bardy F, Dillon H, Van Dun B. (2014a). Least-squares deconvolution of evoked potentials and sequence optimization for multiple stimuli under low-jitter conditions. Clinical Neurophysiology 125, 727–737.

Ref 2. Bardy F, Van Dun B, Dillon H, Cowan R. (2014b). Least-squares (LS) deconvolution of a series of overlapping cortical auditory evoked potentials: a simulation and experimental study. Journal of Neural Engineering 11, art. 046016.

Ref 3. Bidelman GM (2015). Towards an optimal paradigm for simultaneously recording cortical and brainstem auditory evoked potentials. Journal of Neuroscience Methods 241, 94-100.

Ref 4. Bohorquez J, Ozdamar O (2008). Generation of the 40-Hz auditory steady-state response (ASSR) explained using convolution. Clinical Neurophysiology 119, 2598-2607.

Ref 5. Bohorquez J, Ozdamar O (2006). Signal to noise ratio analysis of maximum length sequence deconvolution of overlapping evoked potentials. Journal of the Acoustical Society of America 119, 1073-1081.

Ref 6. de la Torre, A., Valderrama, J., Segura, J.C., Alvarez, I.M. Matrix-based formulation of the iterative randomized stimulation and averaging method for recording evoked potentials. Journal of the Acoustical Society of America (2019), vol. 146, no. 6, pp. 4545-4556.

Ref 7. Eysholdt U, Schreiner C (1982). Maximum length sequences: A fast method for measuring brain-stem-evoked responses. Audiology 21, 242–250.

Ref 8. Galambos R, Makeig S, Talmachoff PJ (1981). A 40-Hz auditory potential recorded from the human scalp. PNAS 78, 2643-2647.

Ref 9. Holt F, Ozdamar O (2016). Effects of rate (0.3-40/s) on simultaneously recorded auditory brainstem, middle and late responses using deconvolution. Clinical Neurophysiology 127, 1589-1602.

Ref 10. Holt FD, Ozdamar O (2014) Simultaneous acquisition of high-rate early, middle, and late auditory evoked potentials. 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (26-30 Aug 2014: Chicago, IL, USA).

Ref 11. Jewett DL, Caplovitz G, Baird B, Trumpis M, Olson MP, Larson-Prior LJ (2004). The use of QSD (q-sequence deconvolution) to recover superposed, transient evoked-responses. Clinical Neurophysiology 115, 2754–2775.

Ref 12. Michelini S, Arslan E, Prosser S, Pedrielli F (1982). Logarithmic display of auditory evoked potentials. Journal of Biomedical Engineering 4, 62-64.

Ref 13. Slugocki C, Bosnyak D, Trainor LJ (2017). Simultaneously-evoked auditory potentials (SEAP): A new method for concurrent measurement of cortical and subcortical auditory-evoked activity. Hearing Research 345, 30-42.

Ref 14. Valderrama J, Alvarez I, de la Torre A, Segura JC, Sainz M, Vargas JL (2012). Recording of auditory brainstem response at high stimulation rates using randomized stimulation and averaging. Journal of the Acoustical Society of America 132, 3856-3865.

Ref 15. Valderrama J, de la Torre A, Alvarez I, Segura JC, Thornton ARD, Sainz M, Vargas JL (2014a). Auditory brainstem and middle latency responses recorded at fast rates with randomized stimulation. Journal of the Acoustical Society of America 136, 3233-3248.

Ref 16. Valderrama, J., de la Torre, A., Alvarez, I., Segura, J.C., Thornton, A.R.D., Sainz, M., Vargas, J.L. A study of adaptation mechanisms based on ABR recorded at high stimulation rate. Clinical Neurophysiology (2014b), vol. 125, no. 4, pp. 805-813.

Ref 17. Valderrama J, de la Torre A, Medina C, Segura JC, Sainz M, Vargas JL (2016). Selective processing of auditory evoked responses with iterative-randomized stimulation and averaging: A strategy for evaluating the time-invariant assumption. Hearing Research 333, 66-76.

Ref 18. Valderrama J, de la Torre A, Van Dun B, Undurraga J, Segura JC, Dillon H, McAlpine D (2017). Comprehensive recording of auditory evoked potentials by projecting over a base of functions. Abstract presented at the XXV International Evoked Response Audiometry Study Group Biennial Symposium (Warsaw, Poland, 21-25 May).

Ref 19. Valderrama, J., de la Torre, A., Van Dun, B. (2018). An automatic algorithm for blink-artifact suppression based on iterative template matching. Journal of Neural Engineering 15, art. 016008, 15p.

Ref 20. Woldorff MG (1993). Distortion of ERP averages due to overlap from temporally adjacent ERPs: Analysis and correction. Psychophysiology 30, 98–119.

Patents

Ref 21. Thornton ARD, Chambers JD, Folkard TJ (1994). Deconvolution of MLS response data (WIPO Patent Application WO/1994/025925, and United States Patent 5734827). Medical Research Council (London).

Ref 22. Jewett, DL (2004). [1] QSD recovery of superposed transient responses (United States Patent 6831467). [2] Recovery of overlapped transient responses using QSD apparatus (WIPO Patent Application WO/2003/003918). [3] QSD apparatus and method for recovery of transient response obscured by superposition (United State Patent 6809526). Abratech Corporation (Sausalito, CA).

CLAIMS

1. A method of estimating the transient auditory evoked potential ('AEP') responses of a subject, the method comprising:

generating a digital auditory stimulus signal consisting of at least one auditory stimulus type;

presenting the at least one auditory stimulus type to a subject via a transducer;

recording an electroencephalogram signal ('EEG') including the neural response of the subject to the at least one auditory stimulus type;

synchronizing the digital auditory stimulus signal with the recorded EEG; and

deconvolving the overlapping AEP responses of the subject from the EEG by applying an iterative randomized stimulation and averaging ('IRSA') technique,

wherein the step of applying an IRSA technique is performed with matrix operations in the representation spaces of the AEP and the EEG.

2. A method according to claim 1, wherein the IRSA technique comprises the steps of: (a) initialisation, (b) response updating, and (c) averaged-residual estimation in which the steps of (b) response updating and (c) averaged-residual estimation are repeated until convergence and wherein steps (a)-(c) are performed using matrix operations.

3. A method according to either of claims 1 or 2, wherein the least one auditory stimulus type includes a stimulus type having a jittered inter-stimulus interval less than the duration of the resulting auditory evoked potential to be detected.

4. A method according to any one of the previous claims, wherein the at least one auditory stimulus type is selected from the group consisting of:

standard auditory stimuli such clicks and tone-bursts; and

complex auditory stimuli like multi-pattern stimuli, speech-like stimuli, or natural speech stimuli.

5. A method according to either of claims any of the previous claims, comprising: applying more than one auditory stimulus type, such that the different stimulus types evoke different AEP responses.

6. A method according to any one of the previous claims, wherein the step of applying the IRSA technique comprises performing iterative matrix operations in segments limited to the duration of the AEP, rather than the duration of the EEG (that is, performing matrix operations in the representation space of the AEPs rather than in the representation space of the EEG).

7. A method according to any one of the previous claims, wherein the step of applying the IRSA technique comprises configuring the matrix operations according to the symmetric-Toeplitz properties of generated matrices to thereby reduce the computational effort required to deconvolve the AEP responses.

8. A method according to any one of the previous claims, wherein the step of applying the IRSA technique comprises calculating the matrix product used when implementing the iterations of the IRSA technique as a convolution.

9. A method according to any one of the previous claims, further comprising the step of calculating the autocorrelation of the digital auditory stimulus signal either as a cross-correlation or as a sum for all stimuli of the digital auditory stimulus signal.

10. A method according to any one of the previous claims, wherein the step of applying the IRSA technique comprises calculating an averaged residual as either the normalised cross-correlation of the EEG and the digital auditory stimulus signal or as a sum for all stimuli of the digital auditory stimulus signal.

11. A method according to either of claims any of the previous claims, comprising:

applying more than one auditory stimulus type, such that the different stimulus types evoke different AEP types; and

adapting the IRSA technique to deconvolve more than one AEP type ('multi-response deconvolution') in its matrix formulation.

12. A method according to any of the previous claims, comprising performing an orthonormal transformation of the representation space, and performing IRSA operations in the transformed representation space.

13. A method according to claim 12, wherein the step of applying an orthonormal transformation results in a transformed representation space of reduced dimensions.

14. A method according to any one of the previous claims, wherein IRSA operations are performed in the transformed representation space derived from a matrix performing any one of the following steps: low-pass filtering; band-pass filtering; decimation; latency dependent filtering; or latency dependent decimation.

15. A method according to any one of the previous claims wherein IRSA operations are performed in the reduced representation space derived from an orthonormal matrix performing latency dependent filtering and latency dependent decimation.

16. A method according to any one of the previous claims, wherein the method is used to estimate AEP responses to complex auditory stimuli, including multi-pattern stimuli, speech-like stimuli or natural speech stimuli, either in a single-response or multi-response approach.

17. A method according to any one of the previous claims, wherein the method is used to estimate one or more of auditory brainstem responses, middle latency responses, or cortical auditory evoked potentials to complex auditory stimuli, including multi-pattern stimuli, speech-like stimuli or natural speech stimuli, either in a single-response or multi-response approach.

18. A method according to any one of the previous claims, wherein the method is used to simultaneously estimate auditory brainstem responses, middle latency responses, and cortical auditory evoked responses to complex auditory stimuli, such as multi-pattern stimuli, speech-like stimuli or natural speech stimuli, either in a single-response or multi-response approach.

19. A method according to any one of the previous claims, wherein the method is used to estimate one or more of auditory brainstem responses, middle latency responses, or cortical auditory evoked potentials, either in a single-response or multi-response approach.

20. A method according to any one of the previous claims, wherein the method is used to simultaneously estimate auditory brainstem responses, middle latency responses, and cortical auditory evoked responses, either in a single-response or multi-response approach.

21. A method according to any one of the previous claims, further comprising graphically representing the estimated AEP.

22. A method according to any one of the preceding claims, wherein the method is performed at least in part on a computer.

23. A system configured to estimate the auditory evoked potential responses of a subject according to a method of any one of the previous claims, the system comprising:

a data processor;

a memory in data communication with the data processor;

wherein the system is configured to implement a method according to any one of the previous claims.

24. A computer program comprising instructions to make a computer carry out a method according to any one claims 1 to 22.

25. A computer-readable storage medium comprising program instructions capable of making a computer carry out the method according to any one of claims 1-22.

26. A transmissible signal comprising program instructions capable of making a computer carry out the method according to any one of claims 1-22.

FIGURE 1

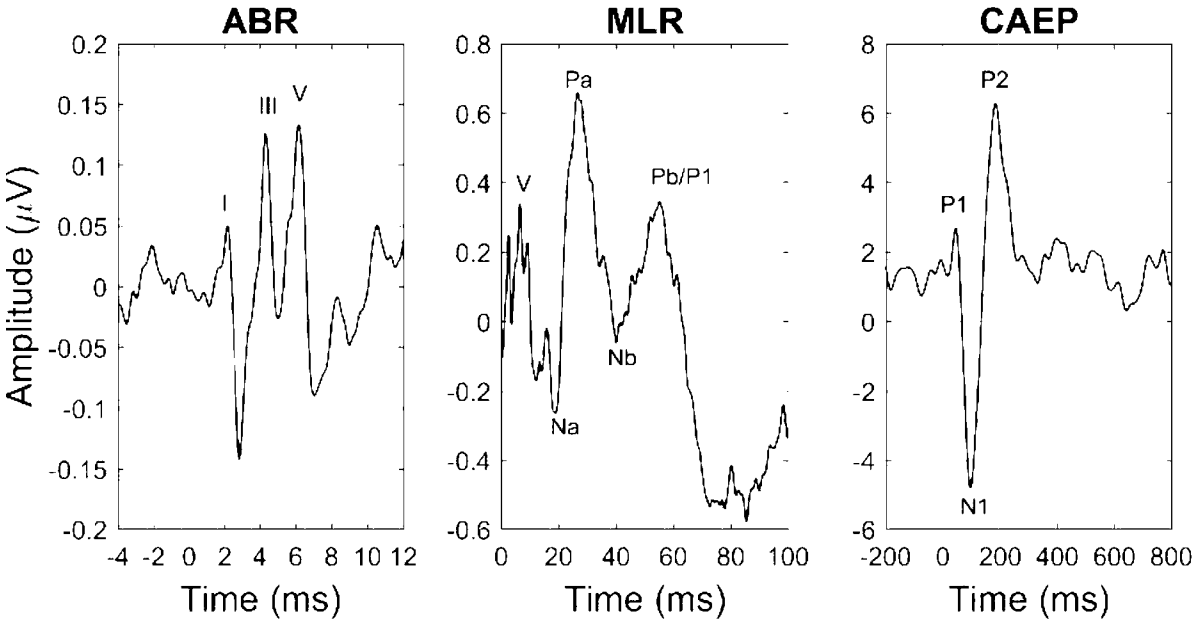


FIGURE 2

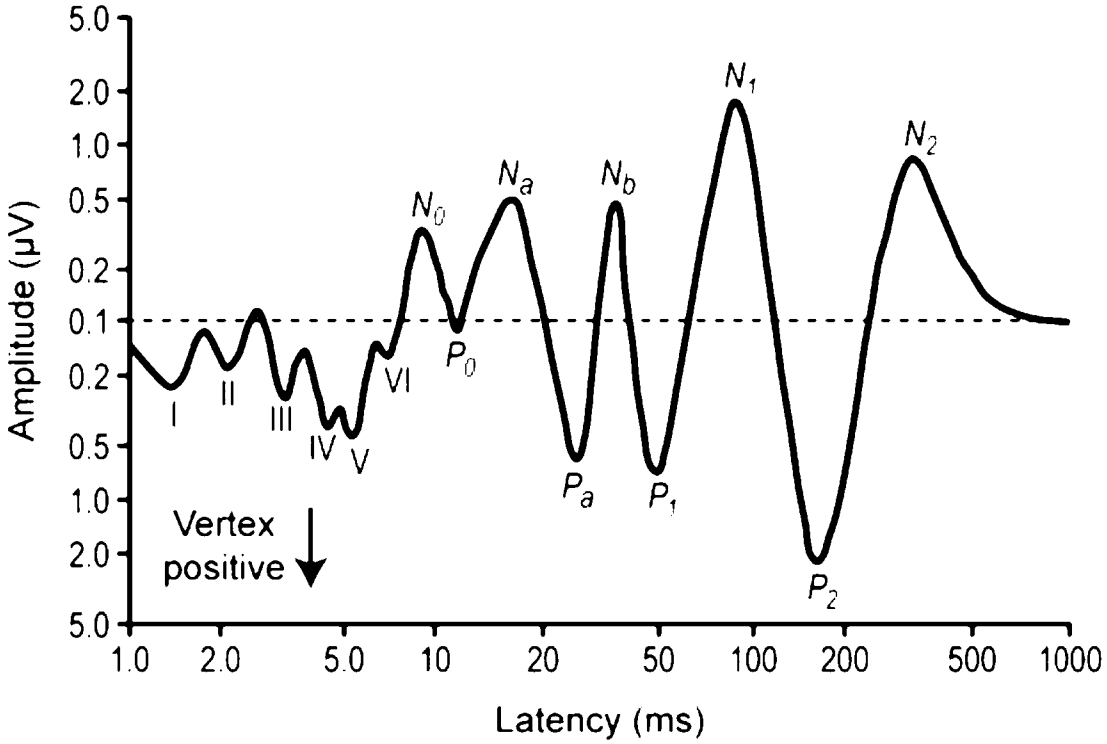


FIGURE 3

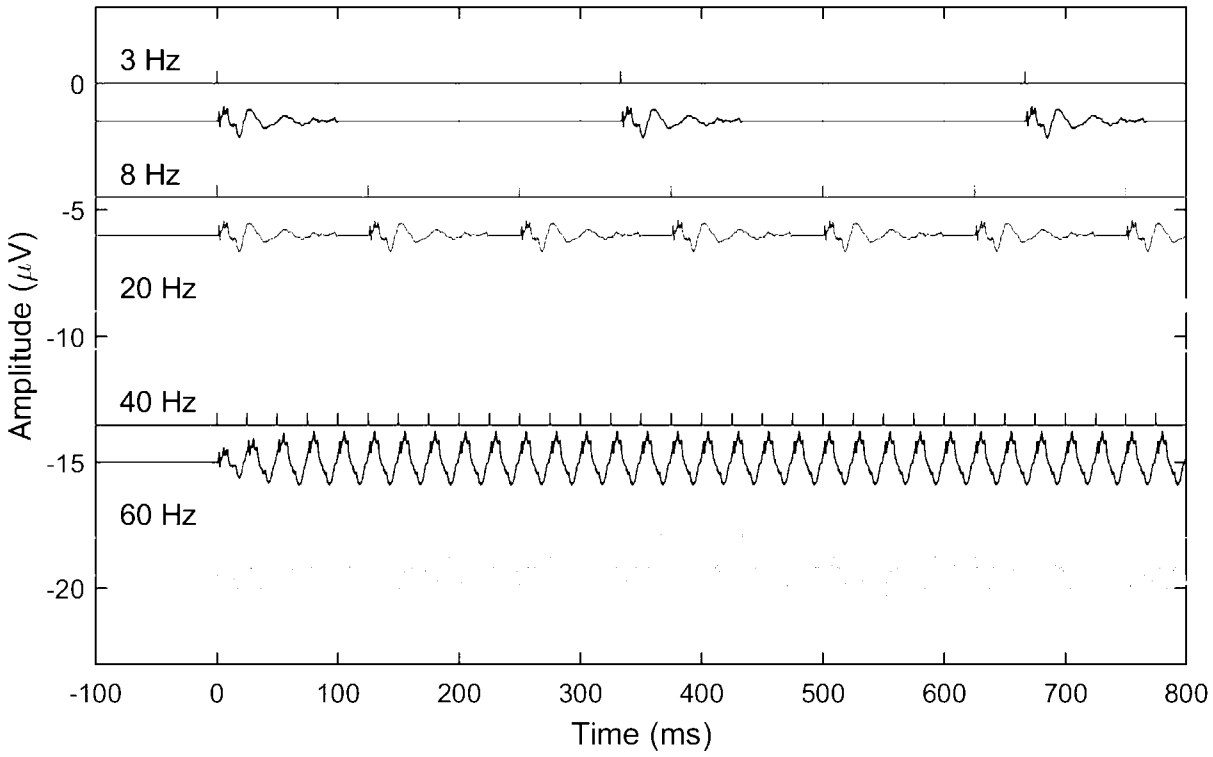


FIGURE 4

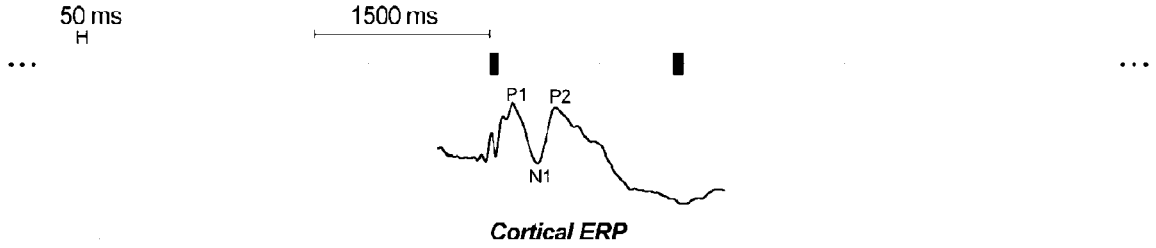


FIGURE 5

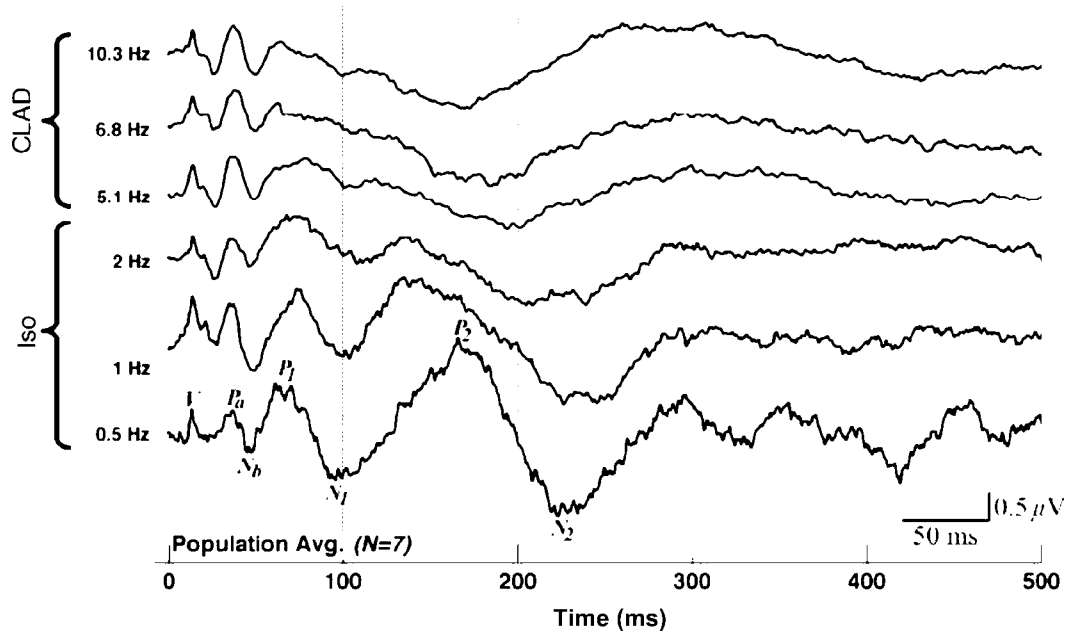


FIGURE 6

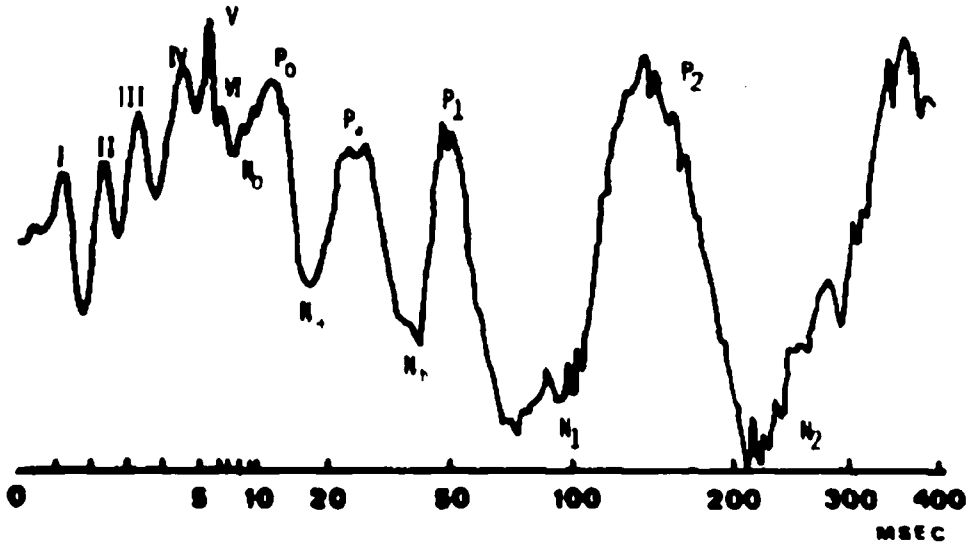
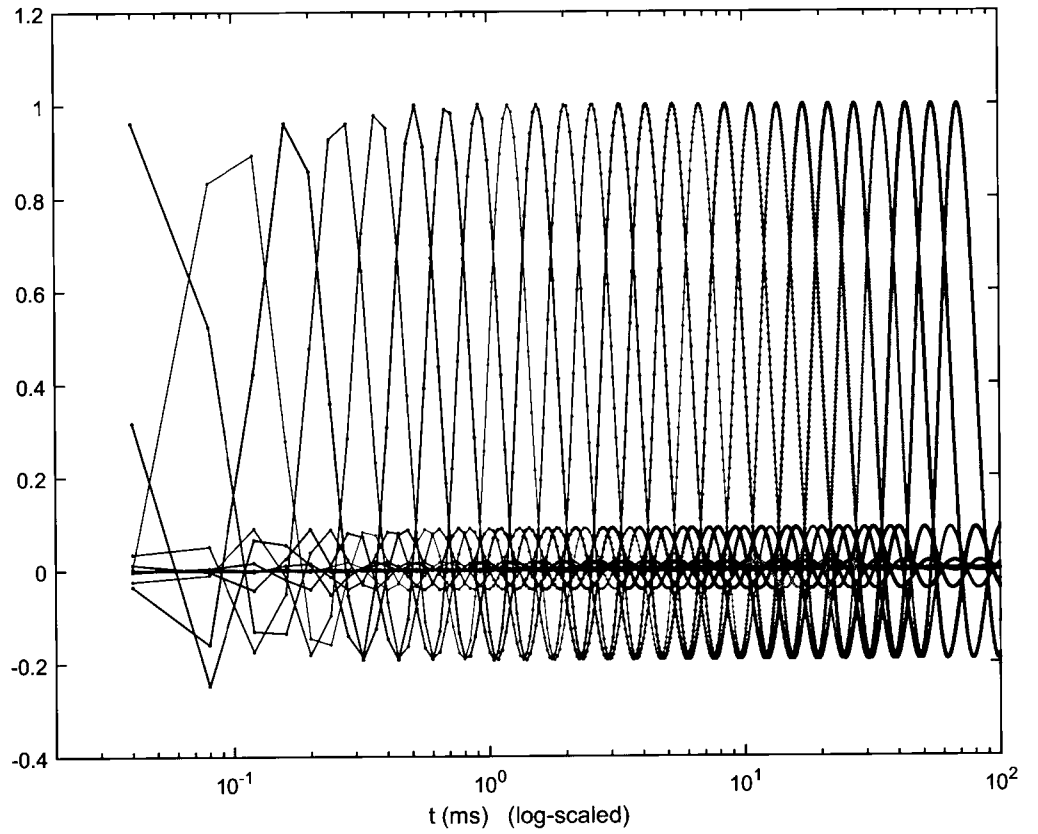


FIGURE 7


1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
10^{-1}
10^{0}
10^{1}
10^{2}
t (ms) (log-scaled)

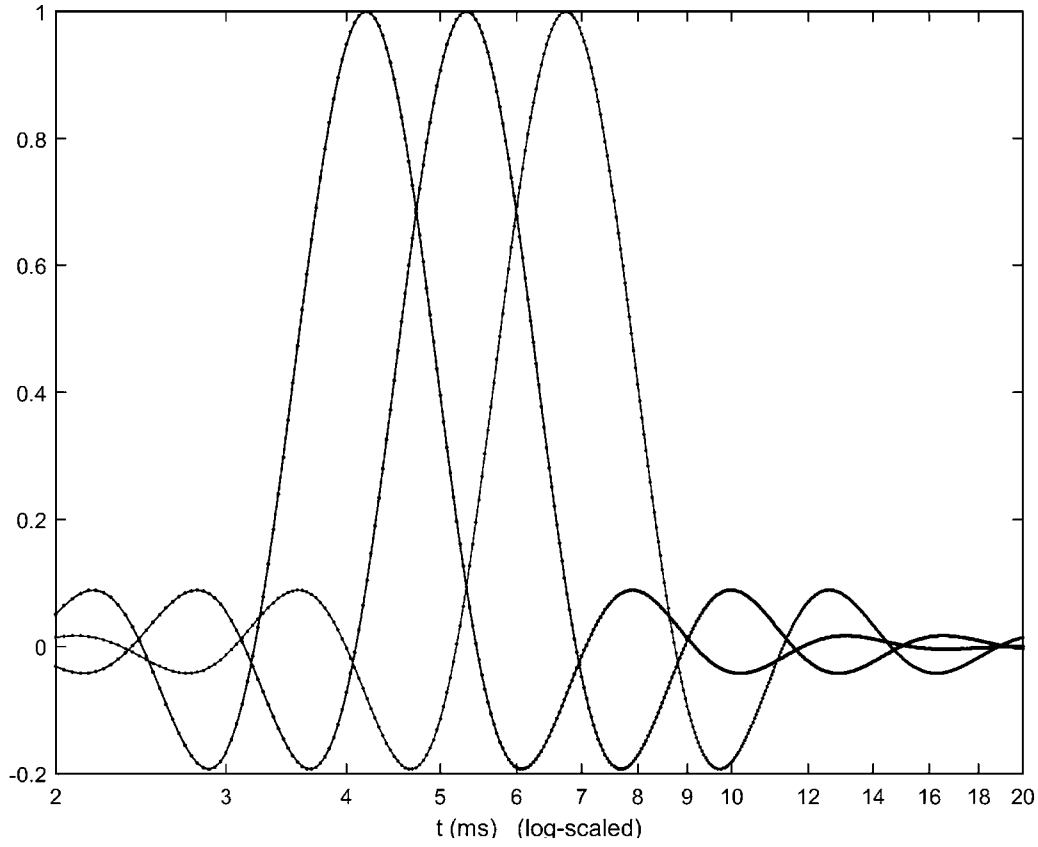
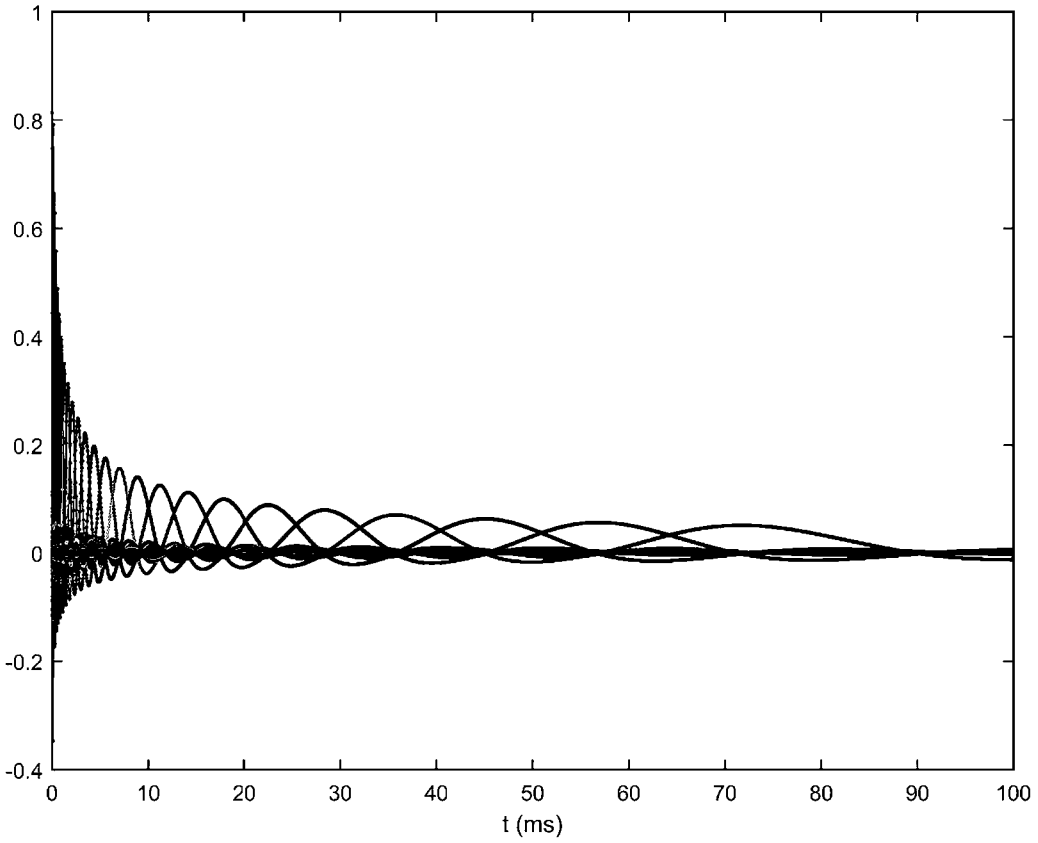

1
0.8
0.6
0.4
0.2
0
-0.2
2
3
4
5
6
7
8
9
10
12
14
16
18
20
t (ms) (log-scaled)

FIGURE 8


1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
0
10
20
30
40
50
60
70
80
90
100
t (ms)

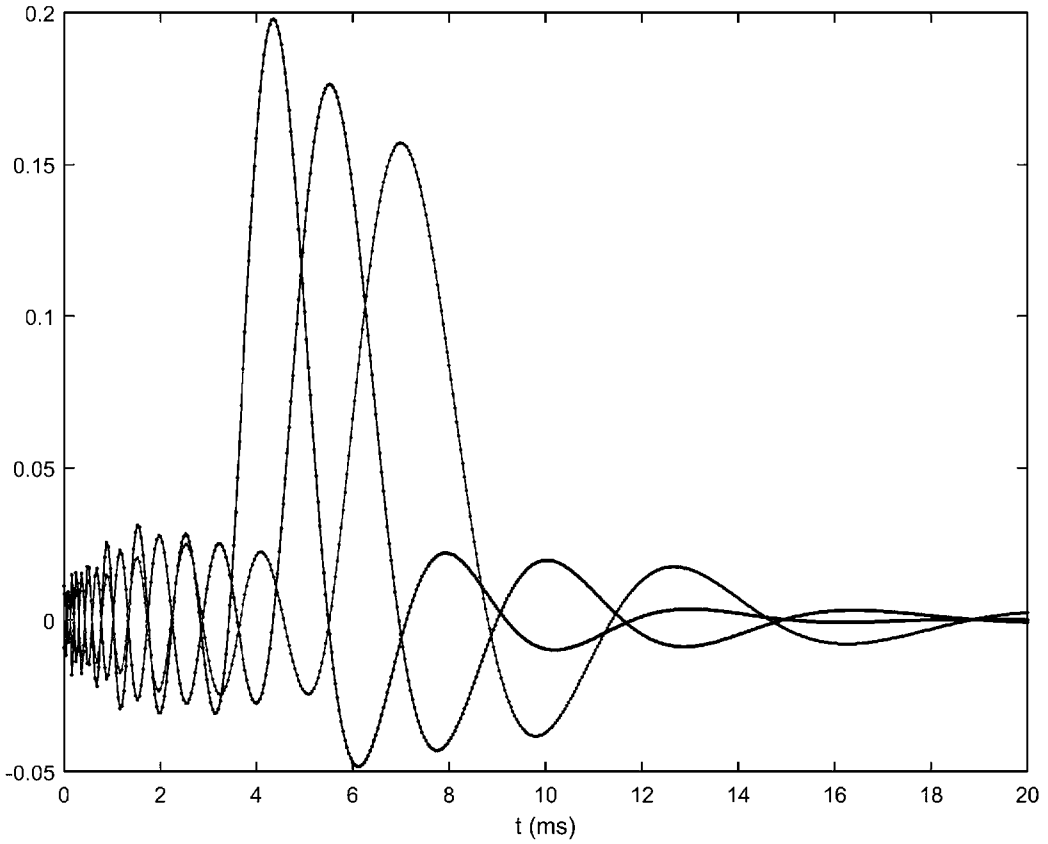
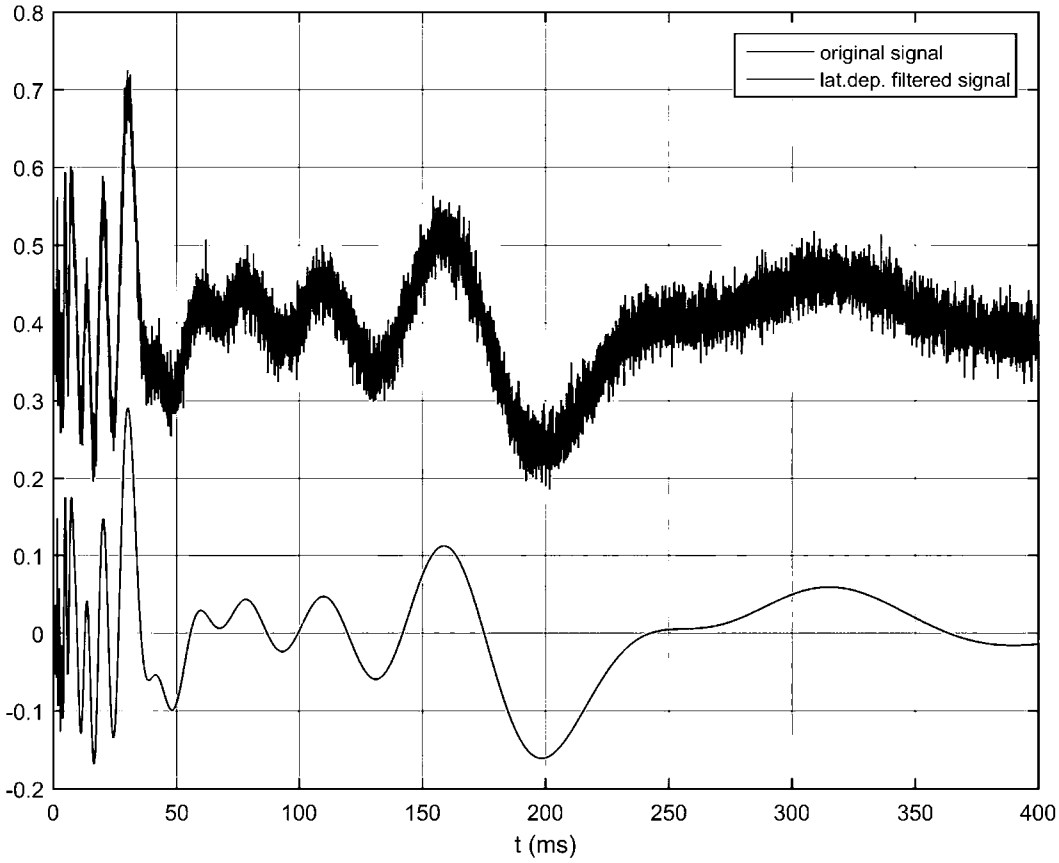

0.2
0.15
0.1
0.05
0
-0.05
0
2
4
6
8
10
12
14
16
18
20
t (ms)

FIGURE 9


original signal
lat.dep. filtered signal
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
0
50
100
150
200
250
300
350
400
t (ms)

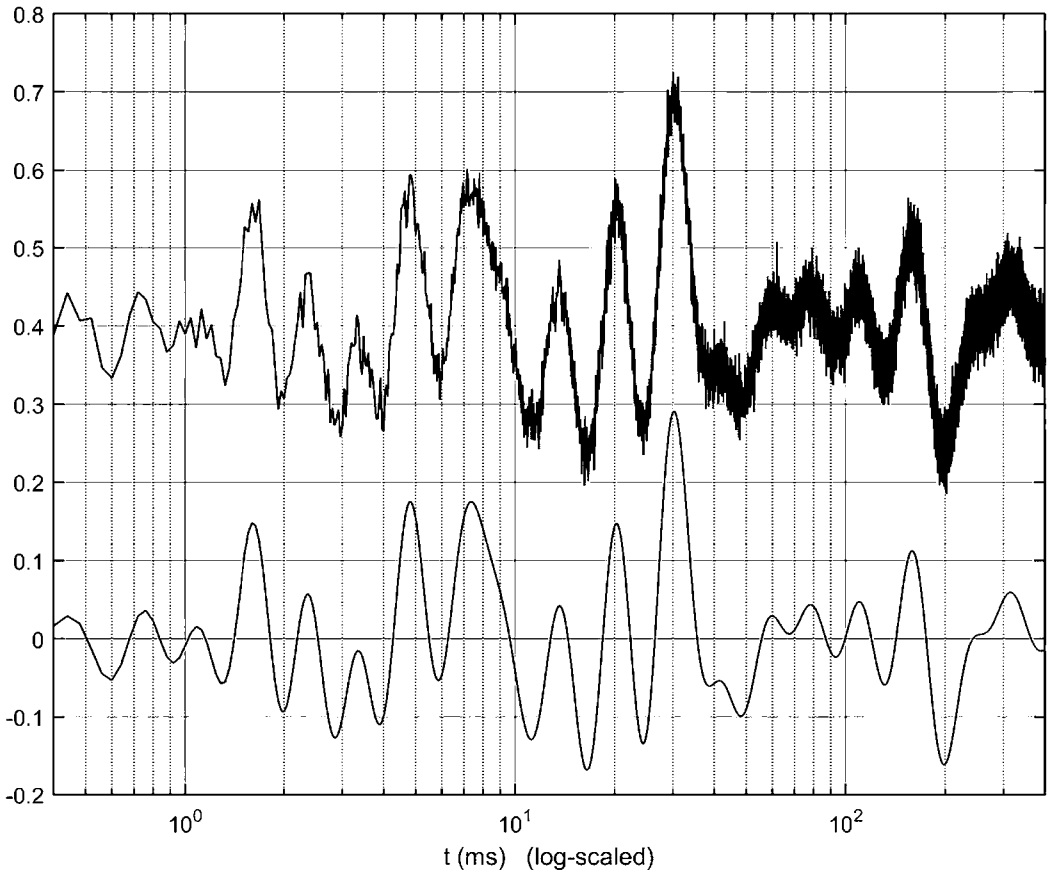
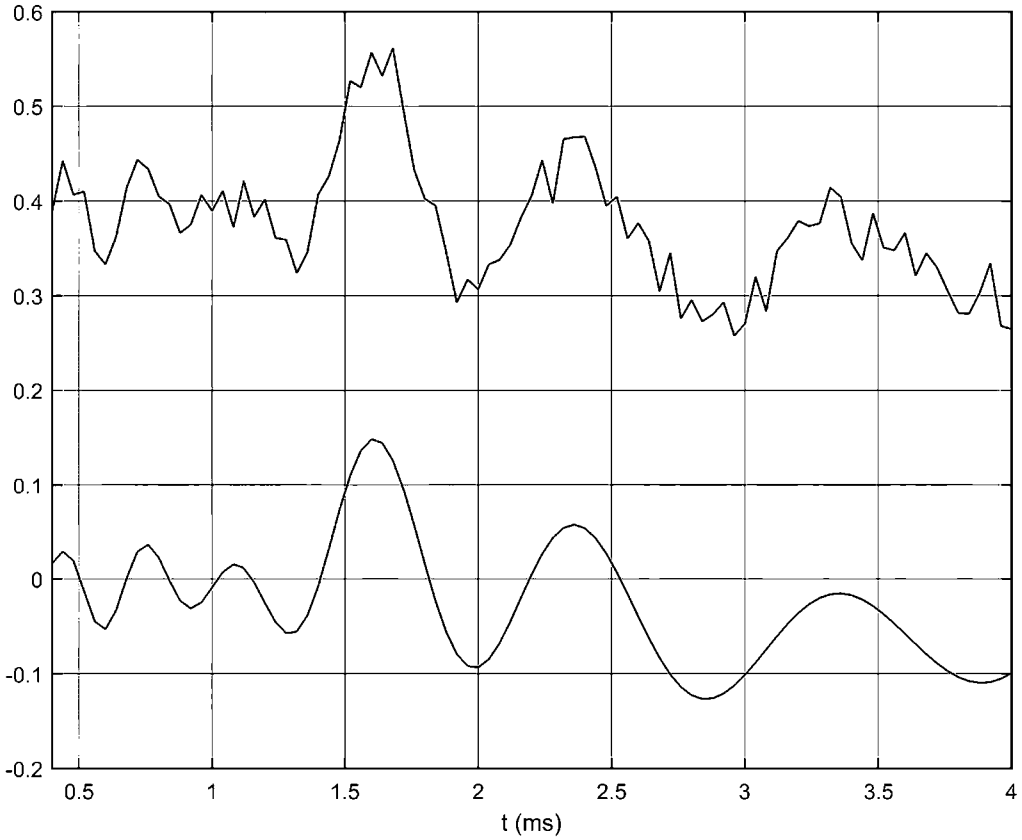

0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
10^0
10^1
10^2
t (ms) (log-scaled)

FIGURE 9 (Cont)



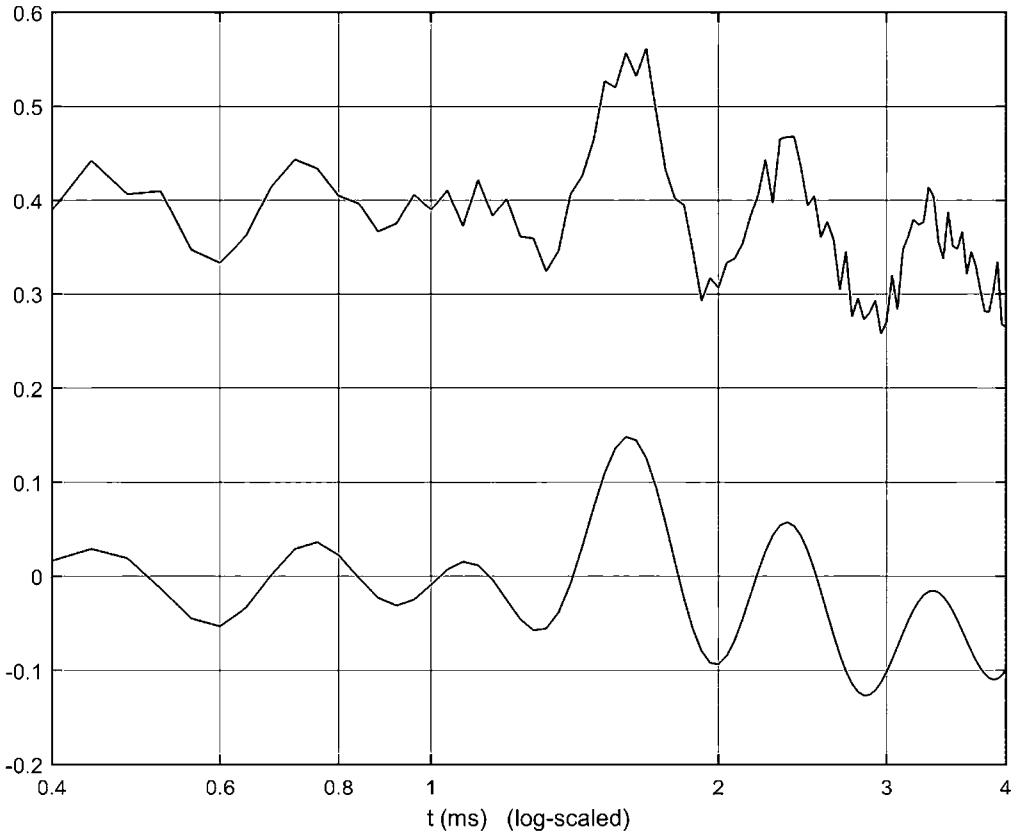
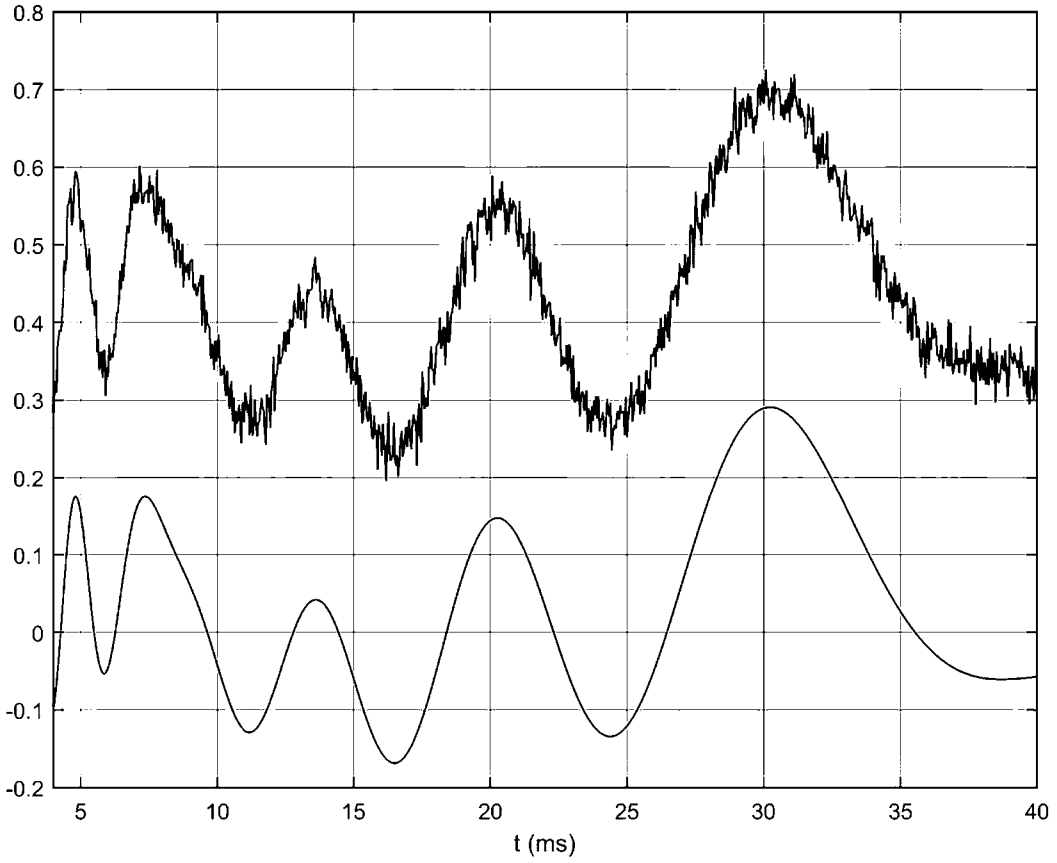


FIGURE 9 (Cont)



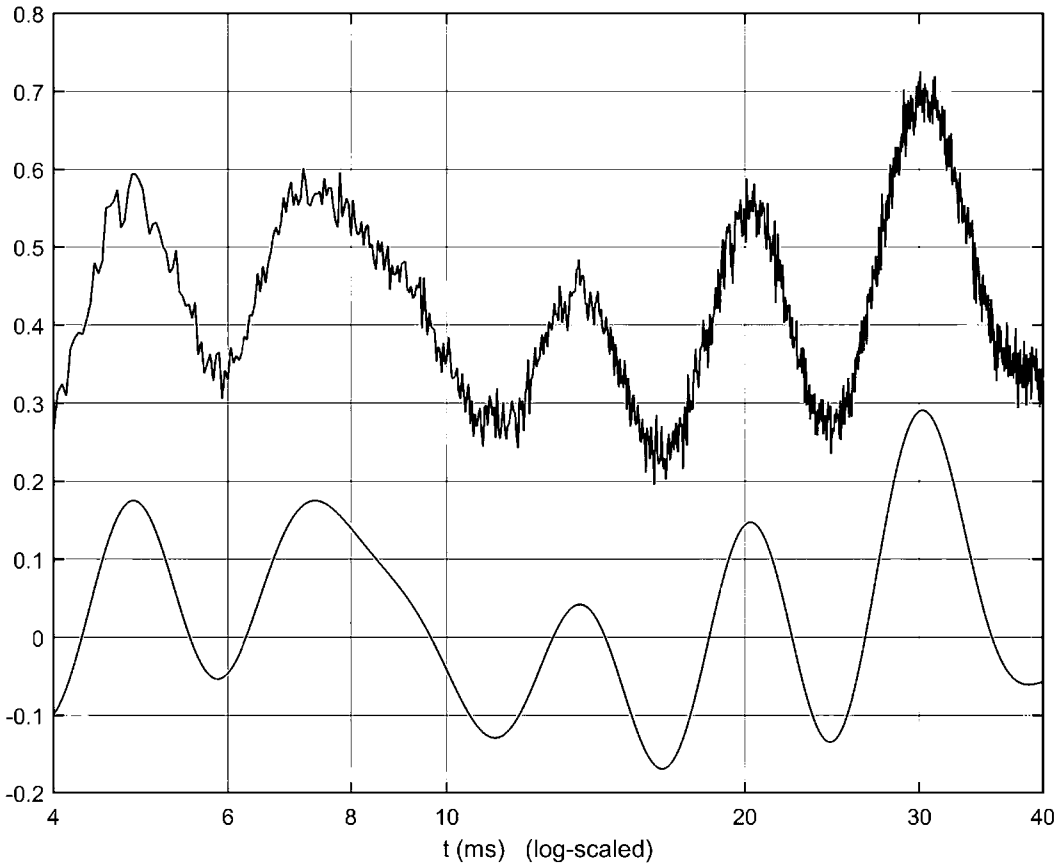
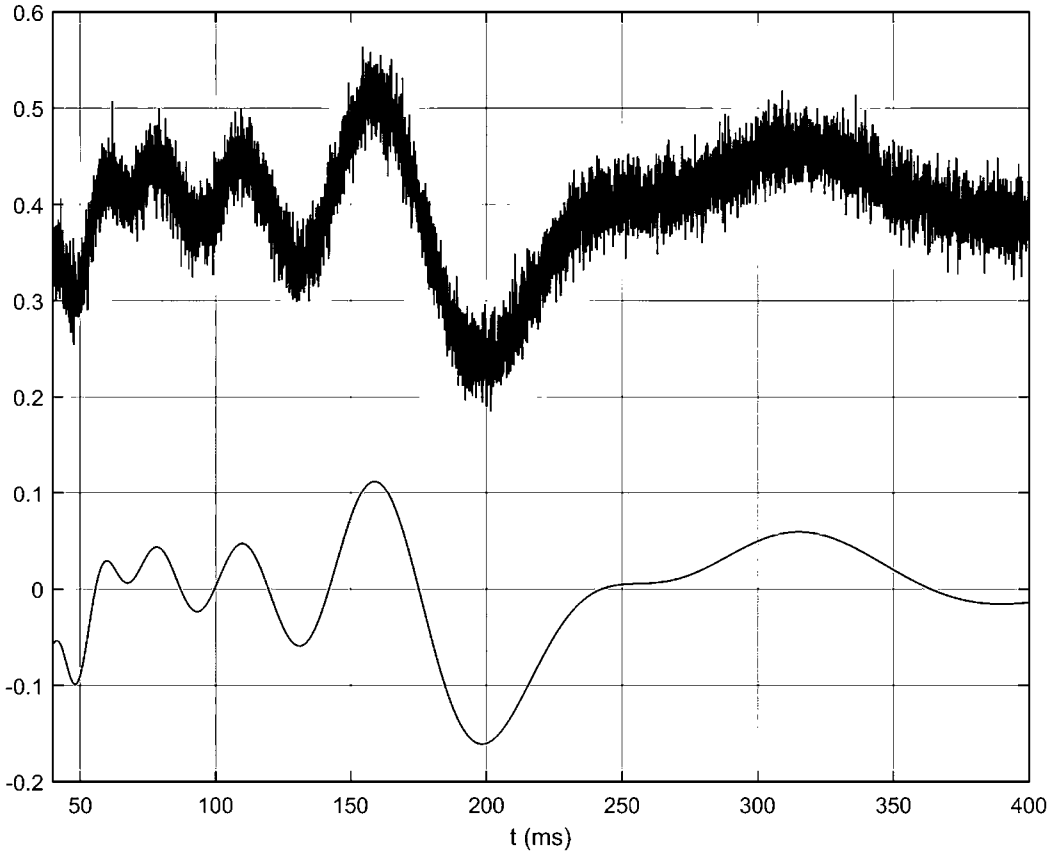


FIGURE 9 (Cont)



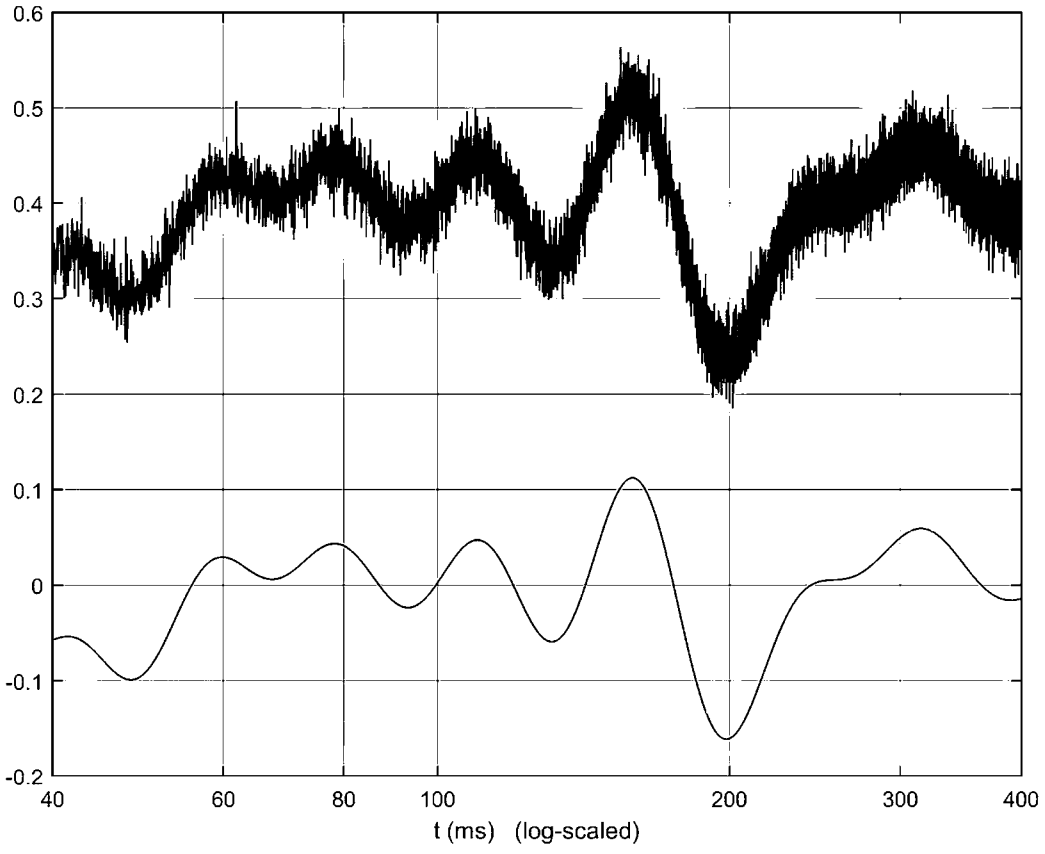


FIGURE 10

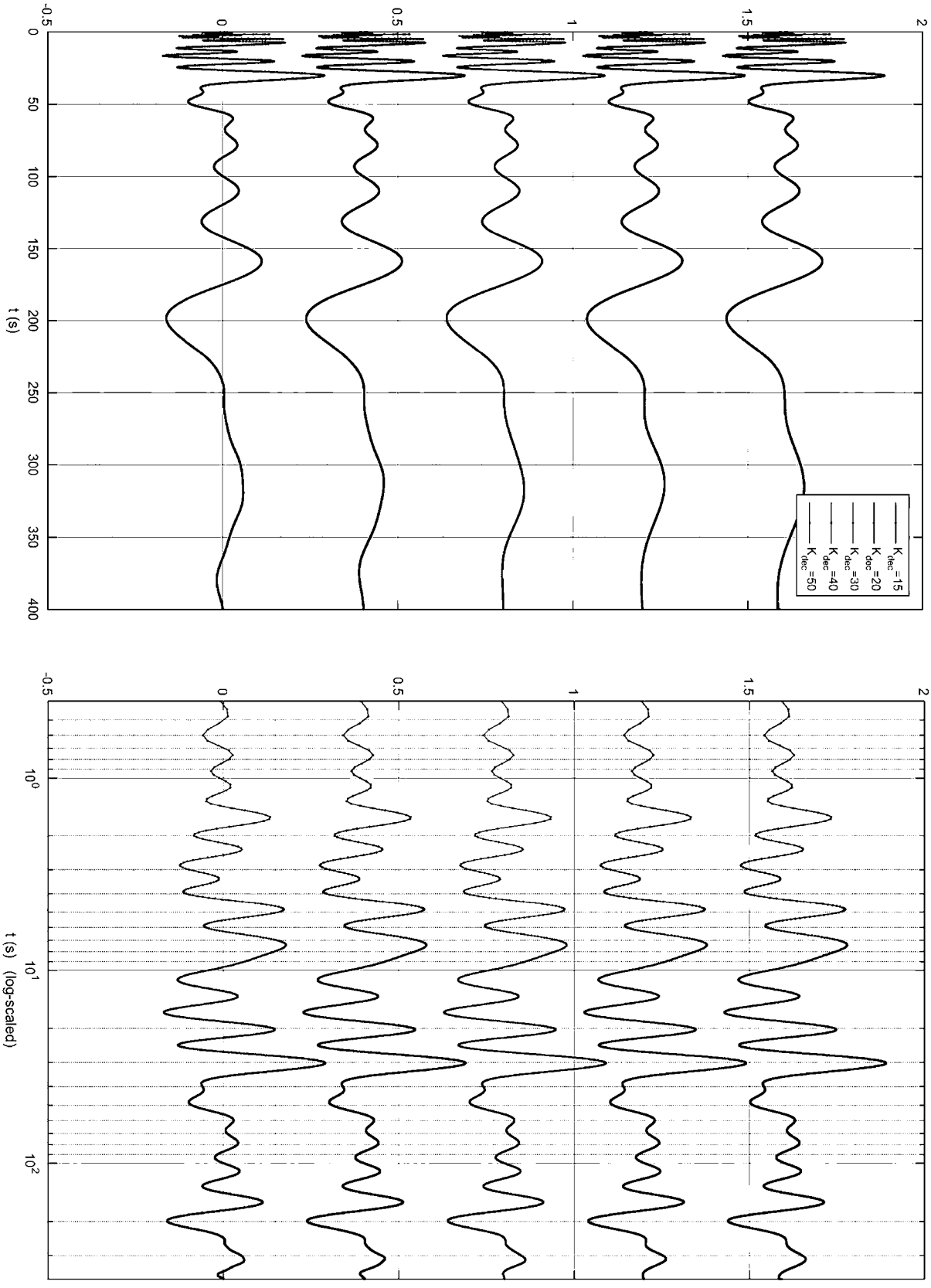

-0.5
0
0.5
1
1.5
2
0
50
100
150
200
250
300
350
400
t (s)
K dec =15
K dec =20
K dec =30
K dec =40
K dec =50
10^0
10^1
10^2
t (s) (log-scaled)

FIGURE 11

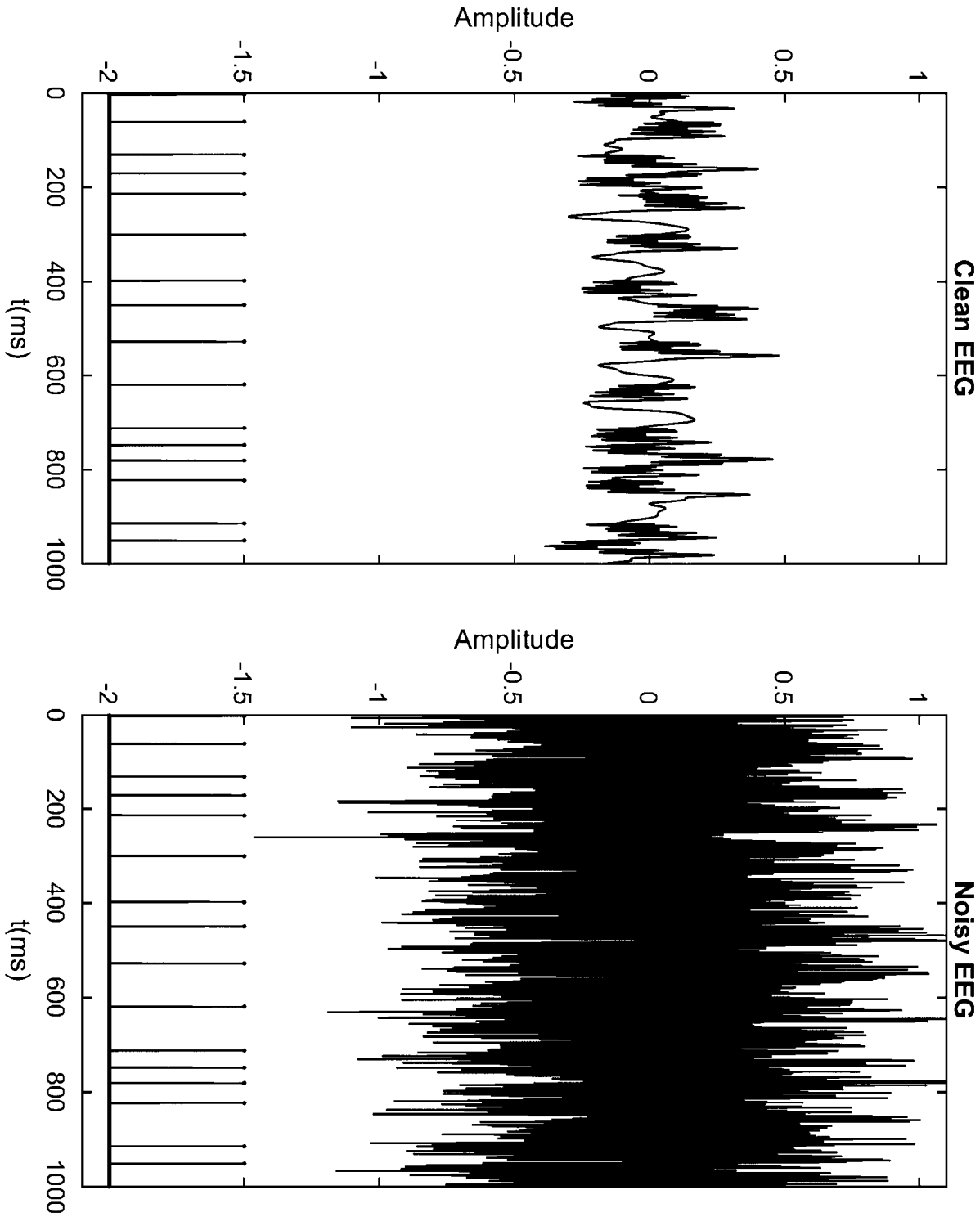
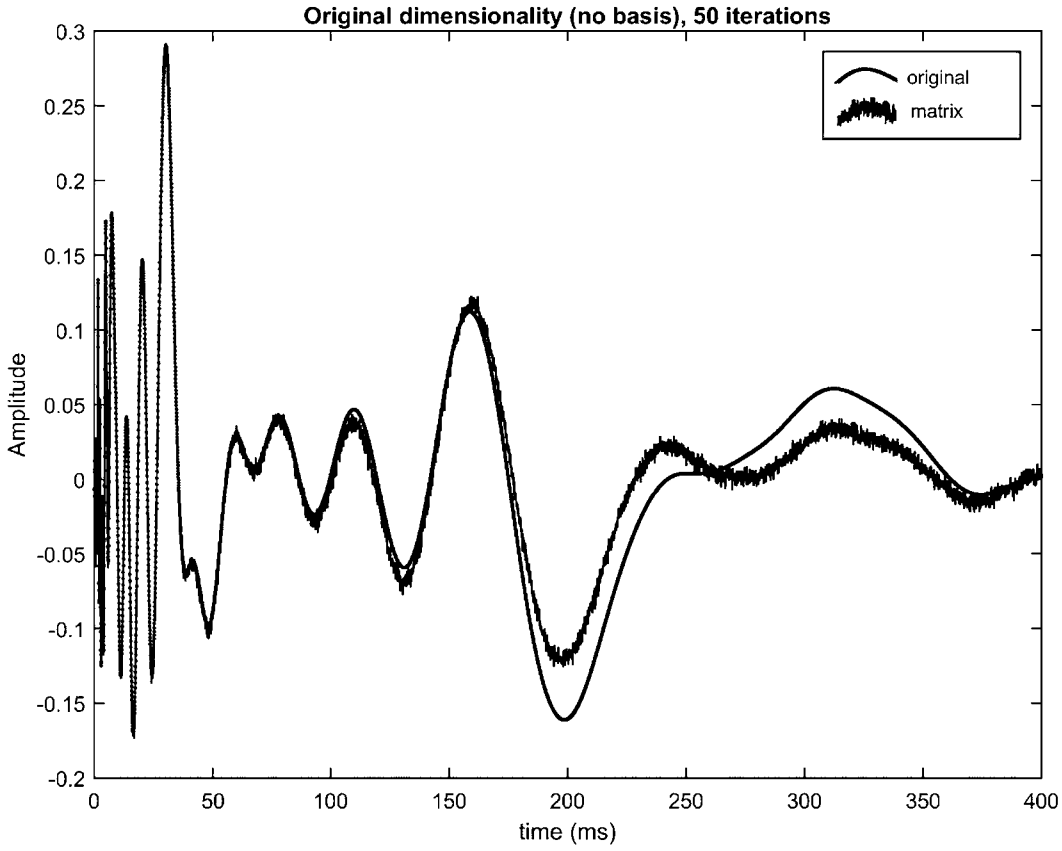

Amplitude
-2
-1.5
-1
-0.5
0
0.5
1
0
200
400
600
800
1000
t(ms)
Clean EEG
Amplitude
-2
-1.5
-1
-0.5
0
0.5
1
0
200
400
600
800
1000
t(ms)
Noisy EEG

FIGURE 12


Original dimensionality (no basis), 50 iterations
original
matrix
Amplitude
time (ms)
0.3
0.25
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
0
50
100
150
200
250
300
350
400

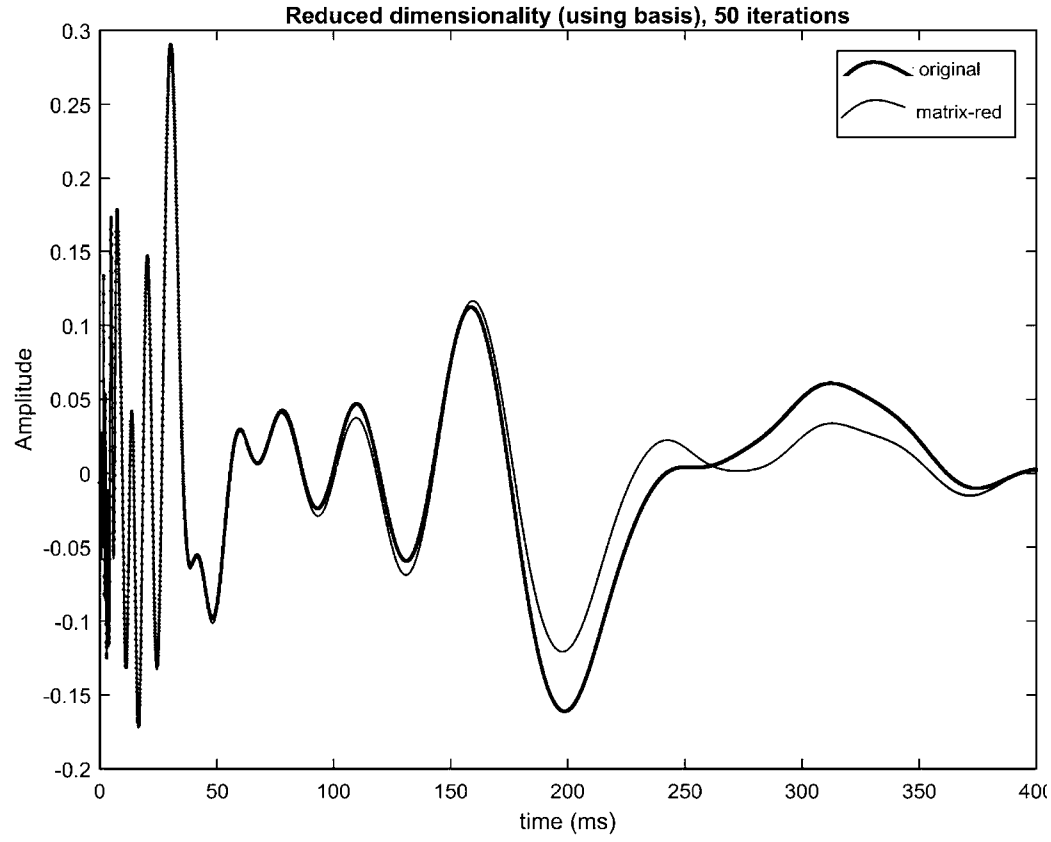
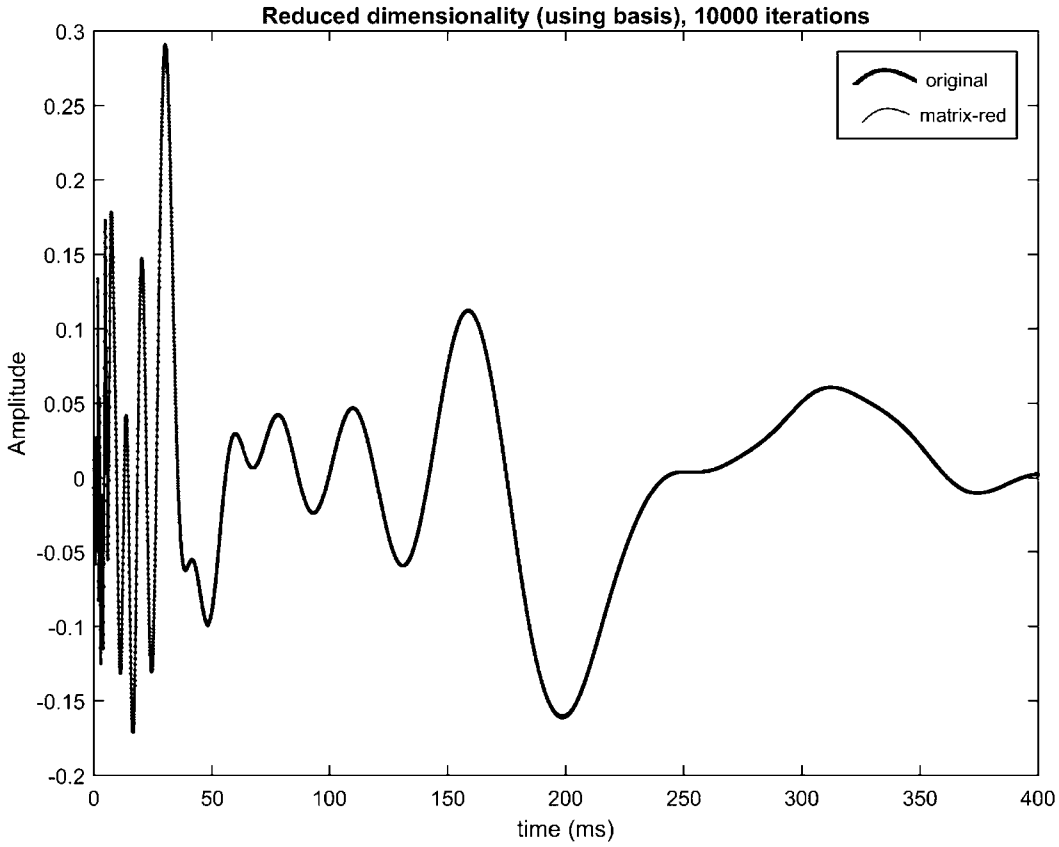

Reduced dimensionality (using basis), 50 iterations
original
matrix-red
Amplitude
time (ms)
0.3
0.25
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
0
50
100
150
200
250
300
350

FIGURE 13


Reduced dimensionality (using basis), 10000 iterations
original
matrix-red
Amplitude
time (ms)
0.3
0.25
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
0
50
100
150
200
250
300
350
400

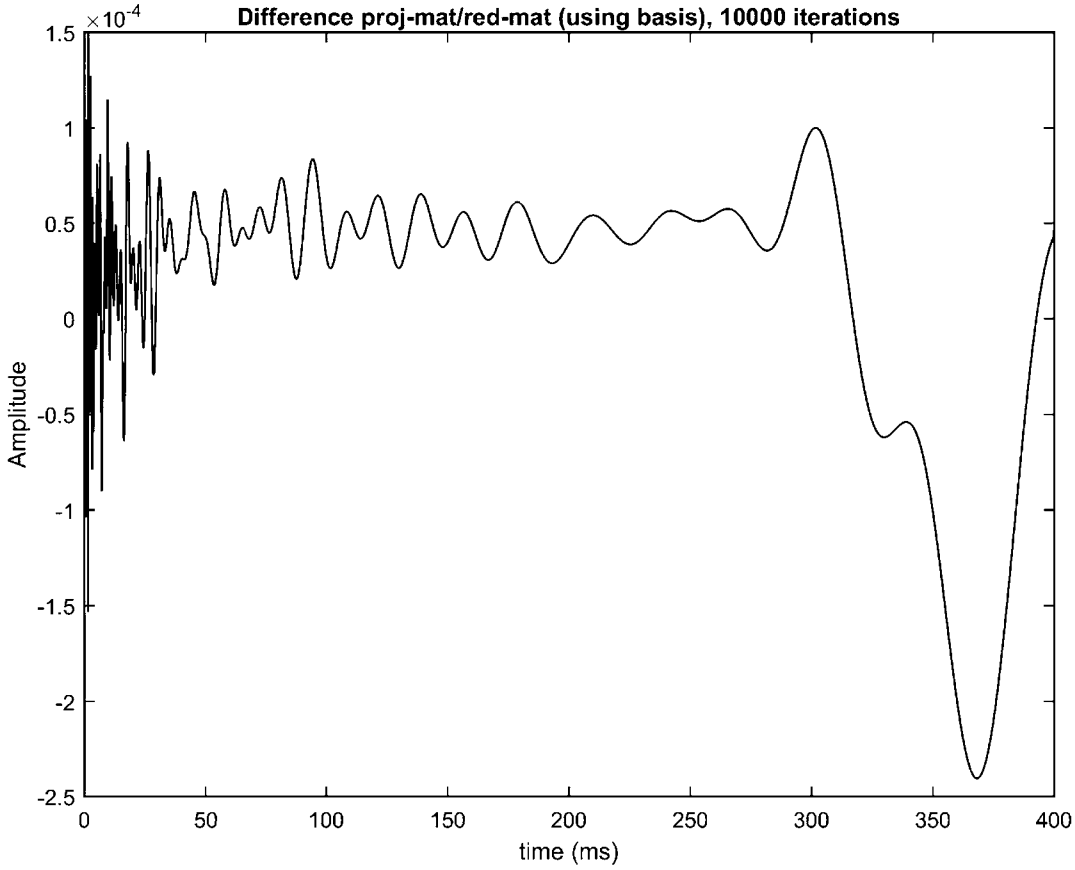

Difference proj-mat/red-mat (using basis), 10000 iterations
×10^-4
Amplitude
time (ms)
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
0
50
100
150
200
250
300
350
400

FIGURE 14

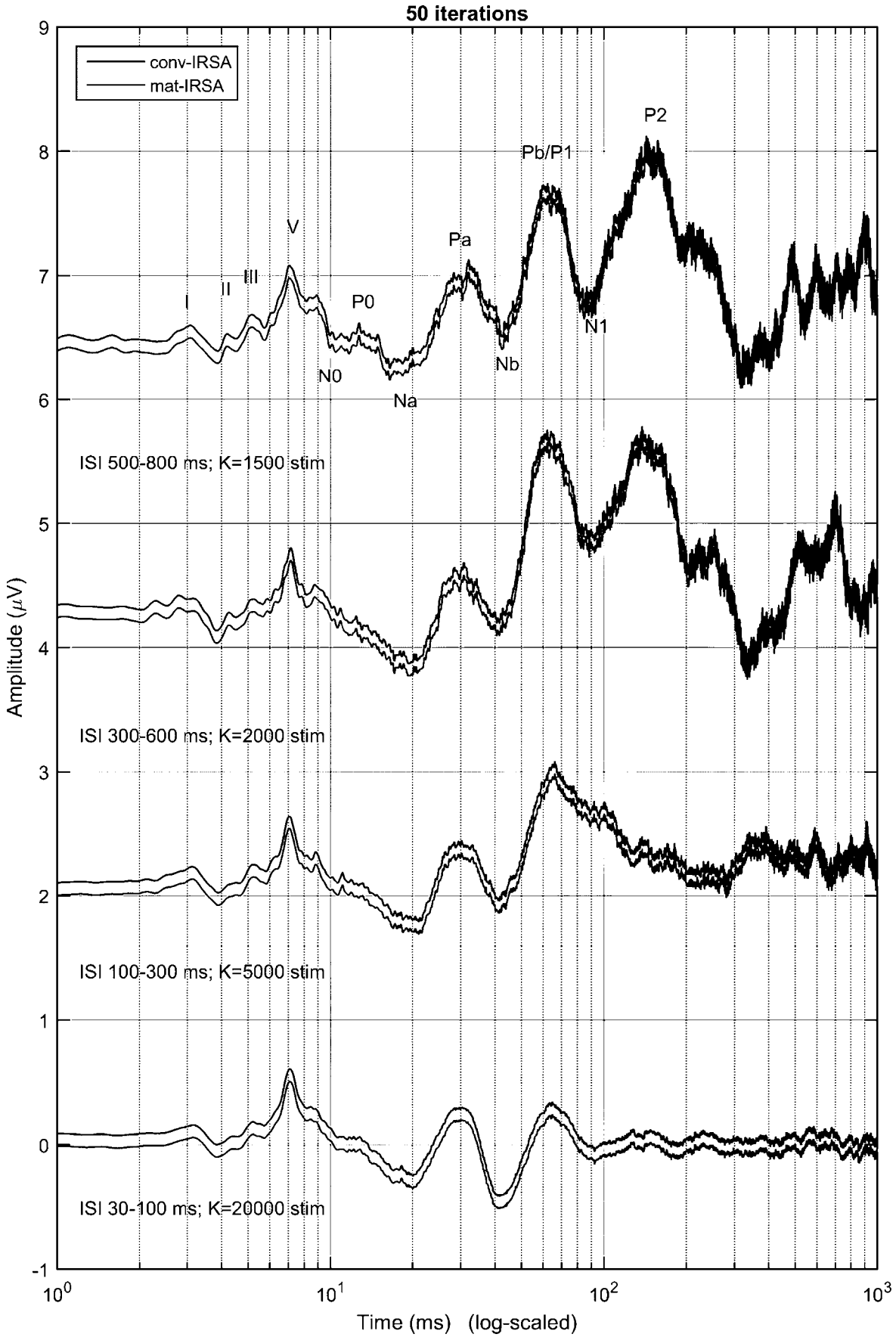

50 iterations
conv-IRSA
mat-IRSA
I
II
III
V
P0
N0
Na
Pa
Nb
Pb/P1
N1
P2
ISI 500-800 ms; K=1500 stim
ISI 300-600 ms; K=2000 stim
ISI 100-300 ms; K=5000 stim
ISI 30-100 ms; K=20000 stim
Amplitude (μV)
Time (ms) (log-scaled)

FIGURE 14 (cont)

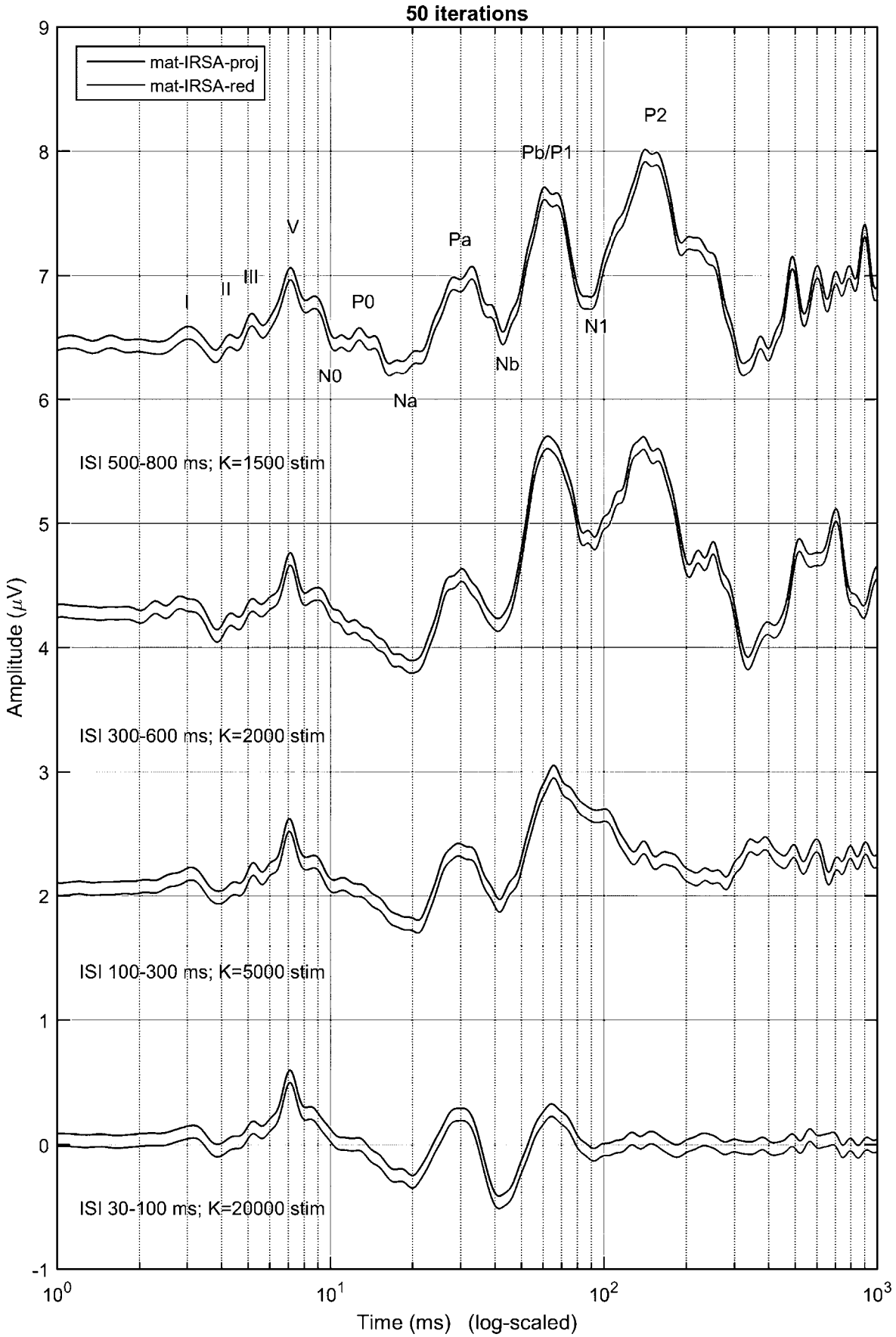


FIGURE 15

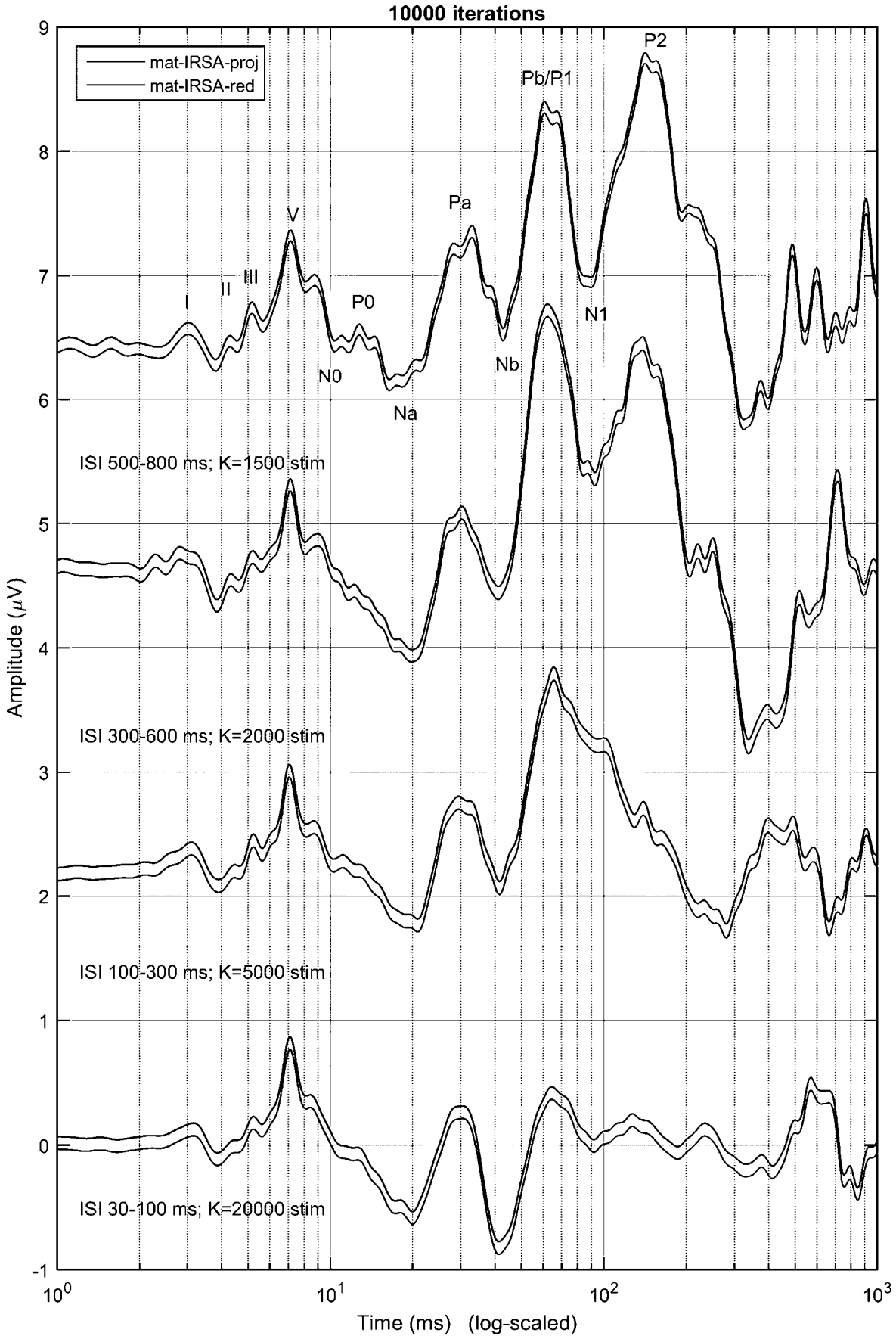

10000 iterations
mat-IRSA-proj
mat-IRSA-red
I
II
III
V
P0
N0
Na
Pa
Nb
Pb/P1
N1
P2
ISI 500-800 ms; K=1500 stim
ISI 300-600 ms; K=2000 stim
ISI 100-300 ms; K=5000 stim
ISI 30-100 ms; K=20000 stim
Amplitude (μV)
Time (ms) (log-scaled)

FIGURE 15 (cont)

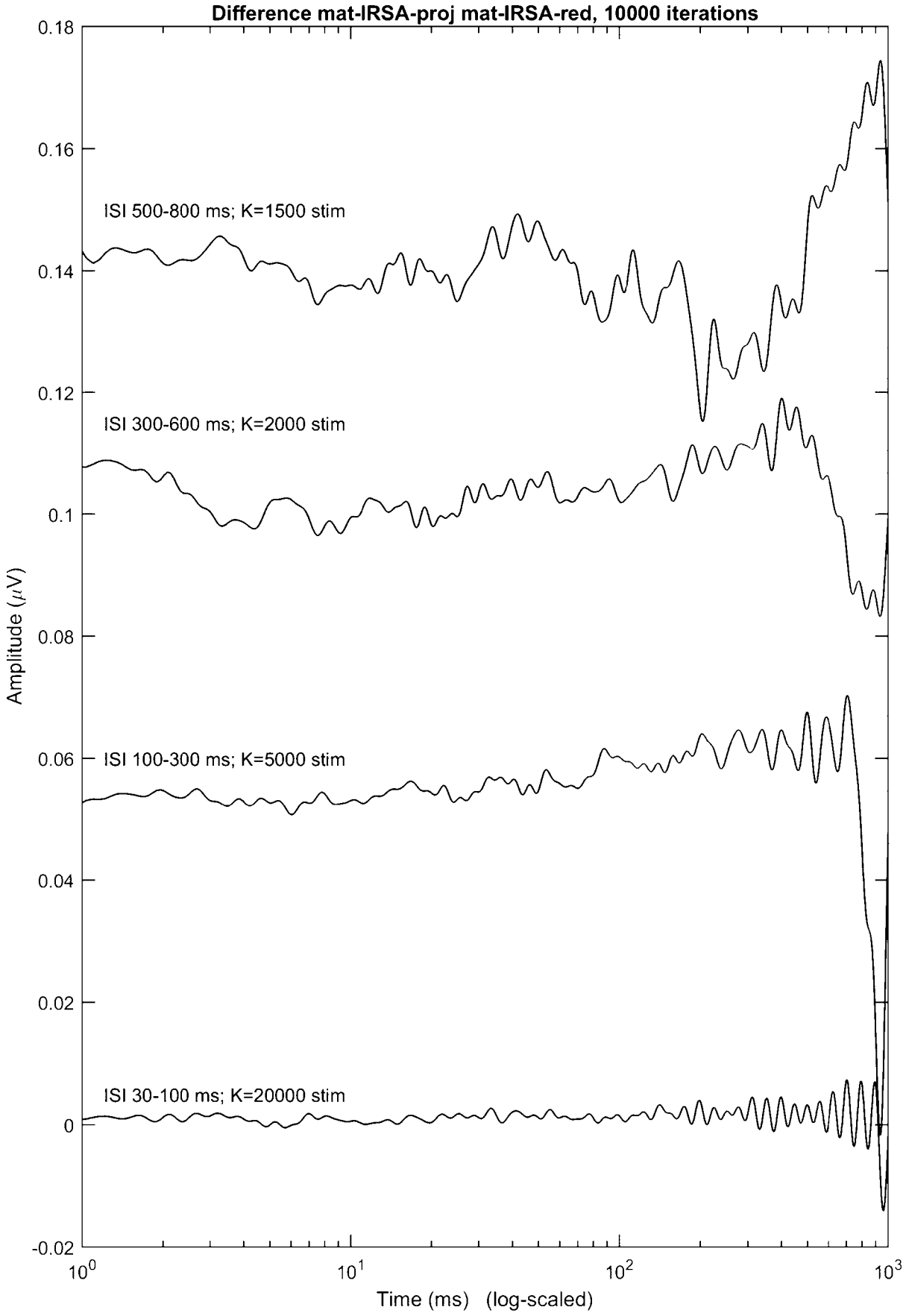


FIGURE 16

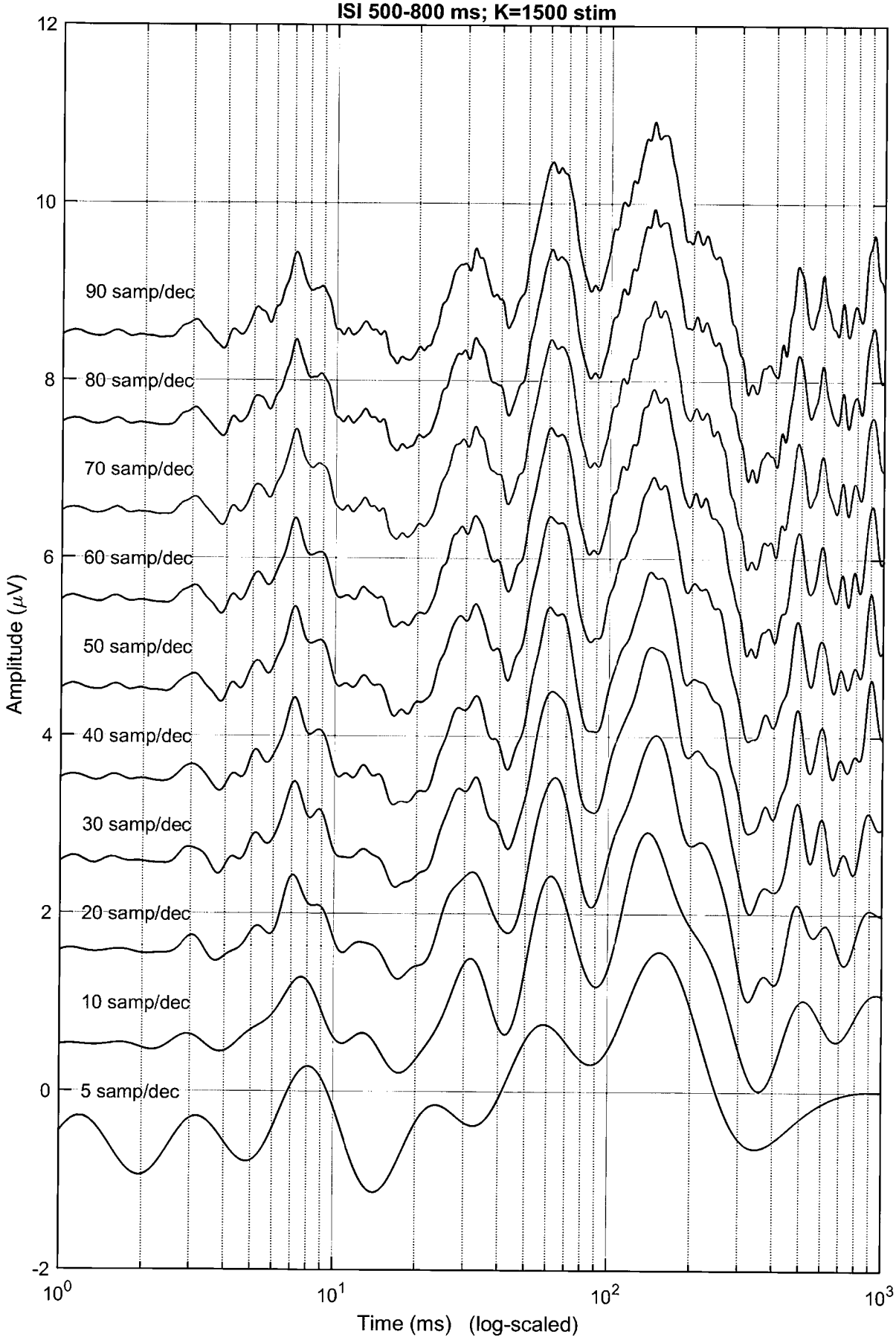

ISI 500-800 ms; K=1500 stim
90 samp/dec
80 samp/dec
70 samp/dec
60 samp/dec
50 samp/dec
40 samp/dec
30 samp/dec
20 samp/dec
10 samp/dec
5 samp/dec
Amplitude (μV)
Time (ms) (log-scaled)
12
10
8
6
4
2
0
-2
10^0
10^1
10^2
10^3

FIGURE 16 (cont)

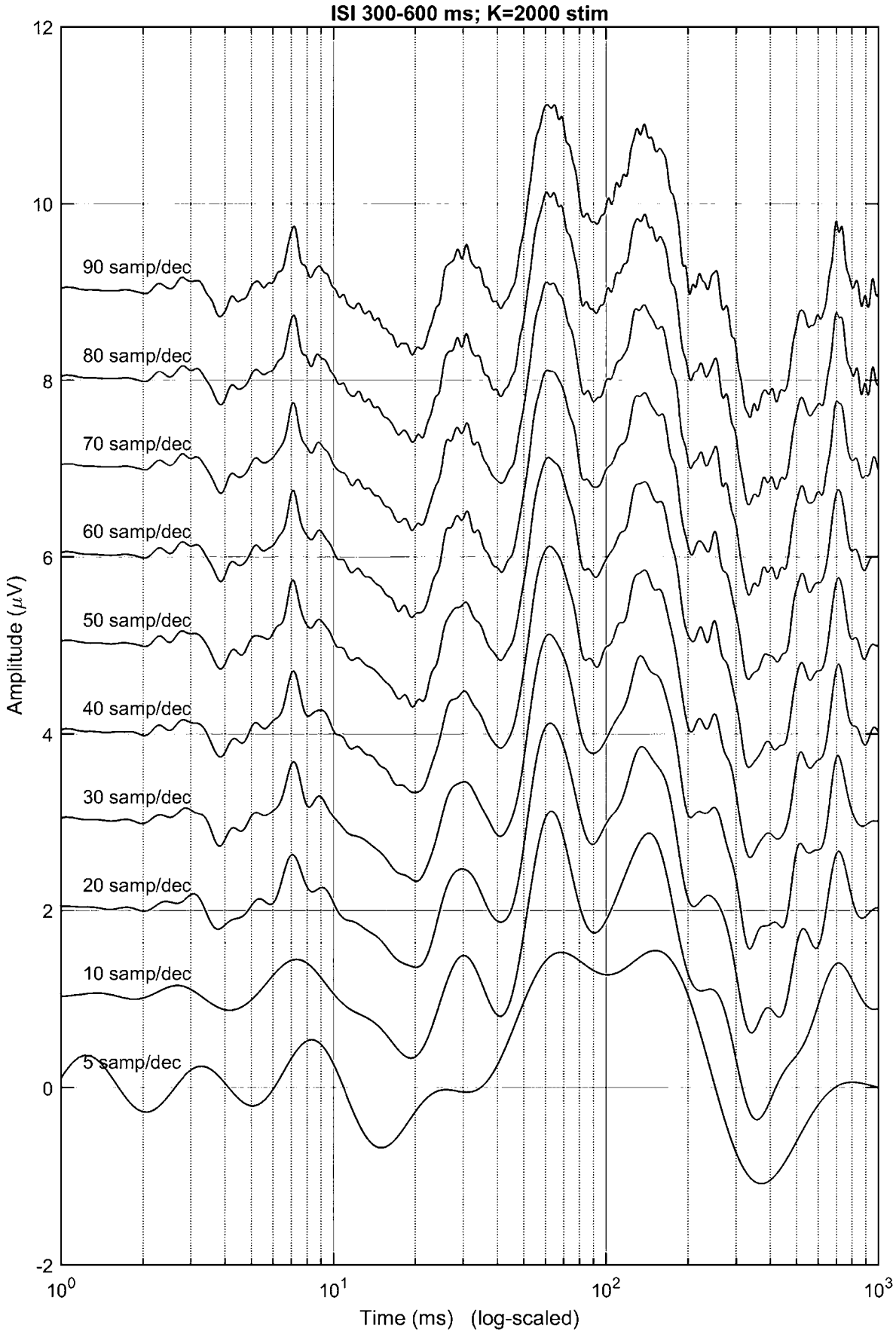


FIGURE 16 (cont)

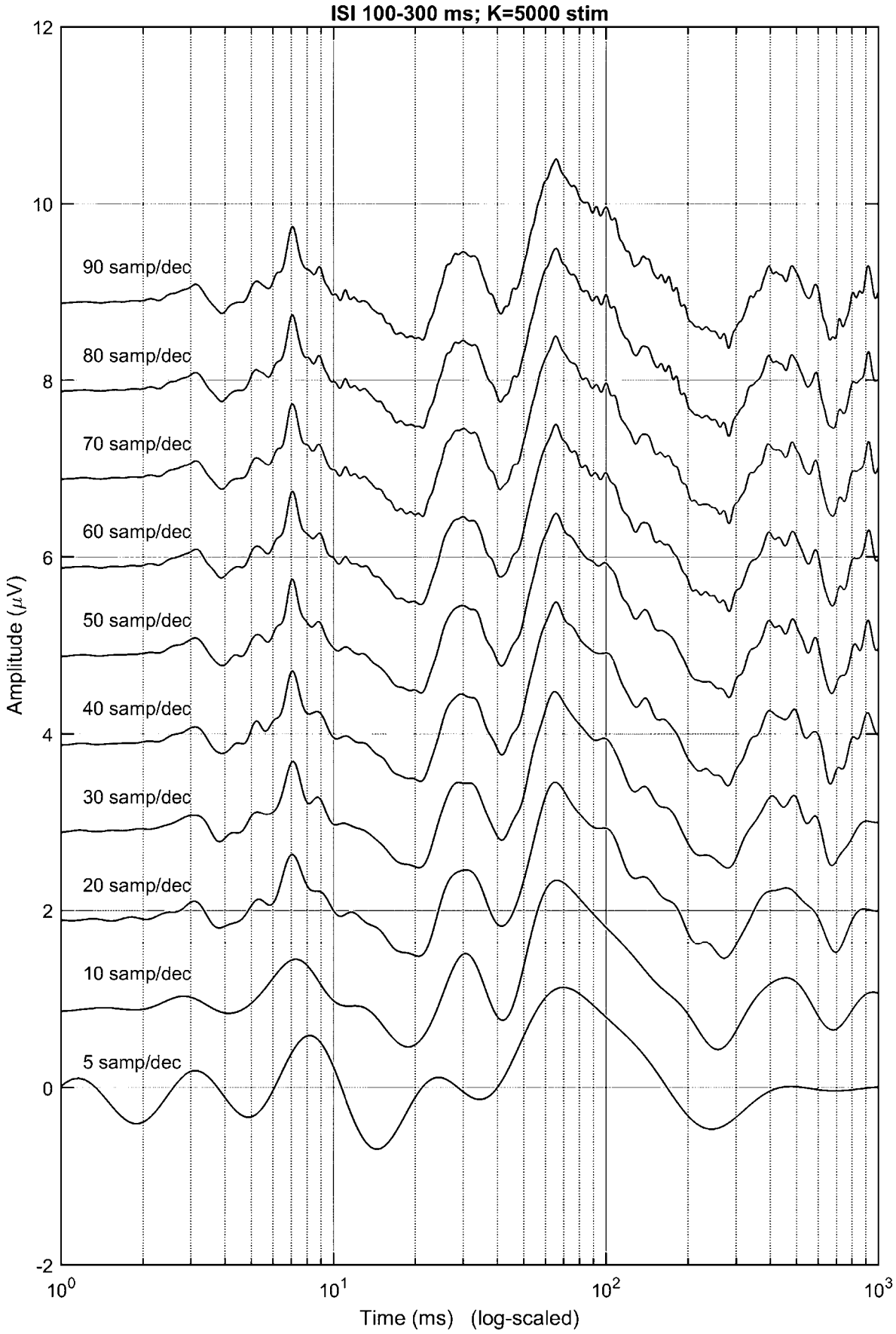


FIGURE 16 (cont)

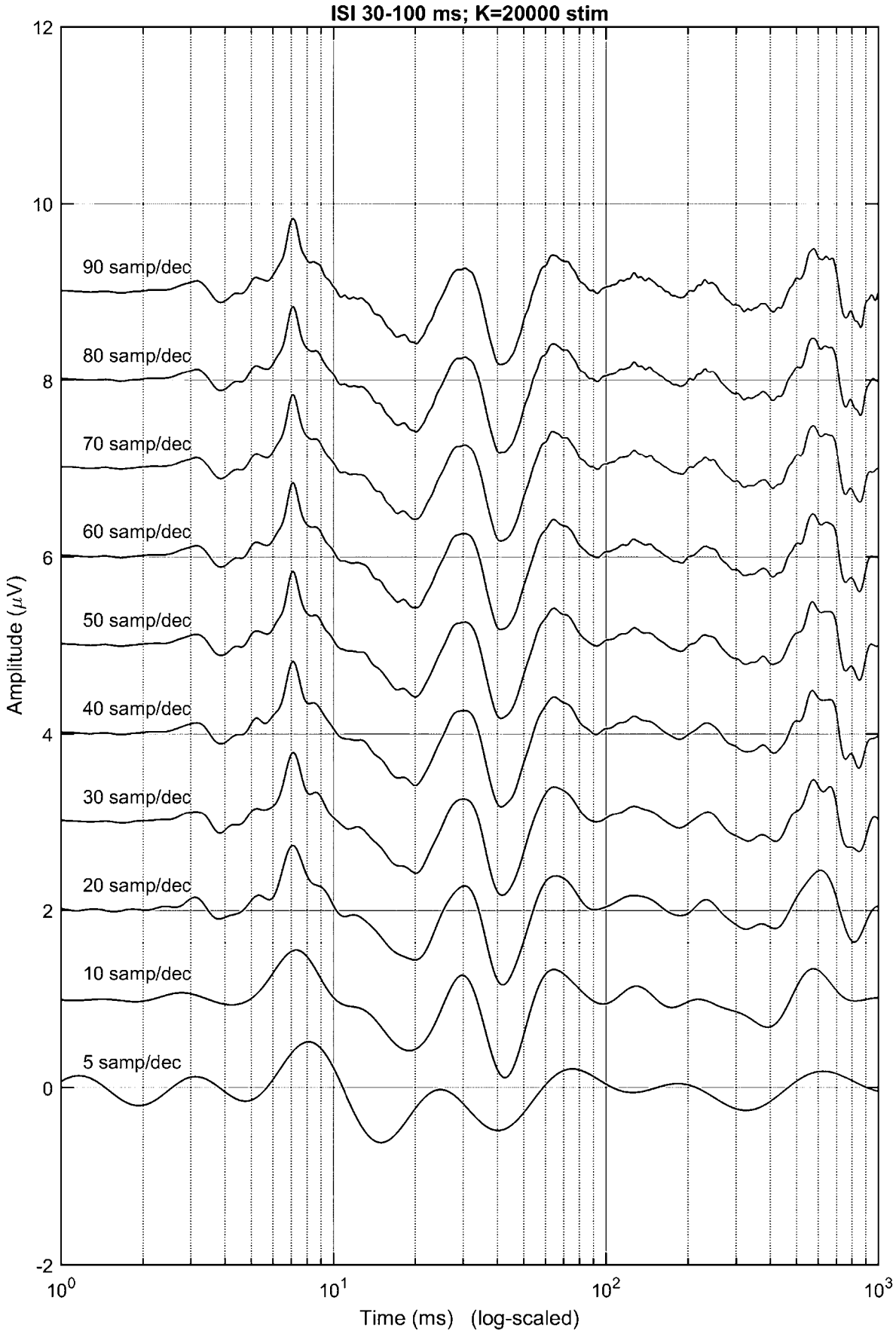


FIGURE 17

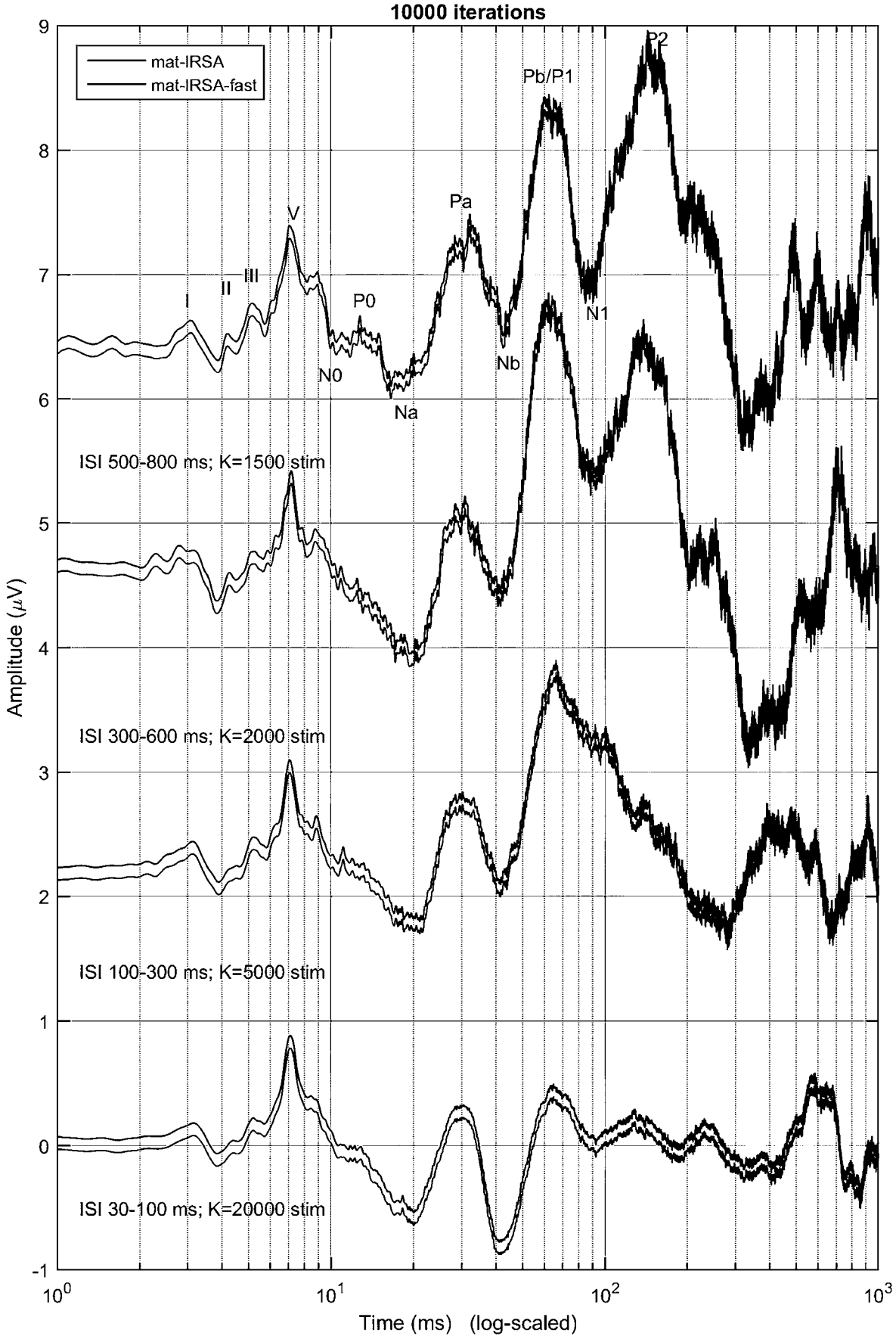

10000 iterations
mat-IRSA
mat-IRSA-fast
I
II
III
V
P0
N0
Na
Pa
Nb
Pb/P1
N1
P2
ISI 500-800 ms; K=1500 stim
ISI 300-600 ms; K=2000 stim
ISI 100-300 ms; K=5000 stim
ISI 30-100 ms; K=20000 stim
Amplitude (μV)
Time (ms) (log-scaled)

FIGURE 17 (cont)

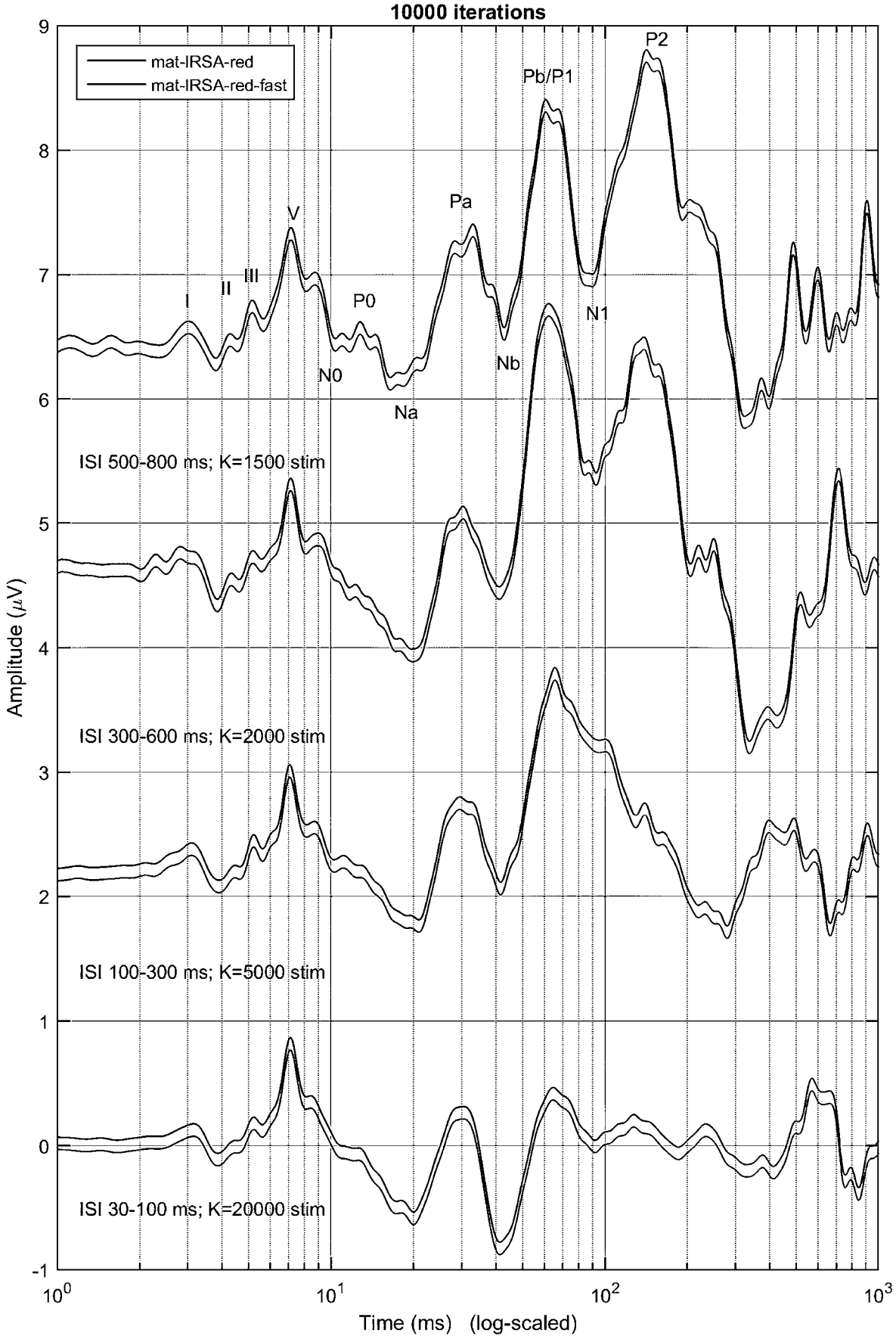
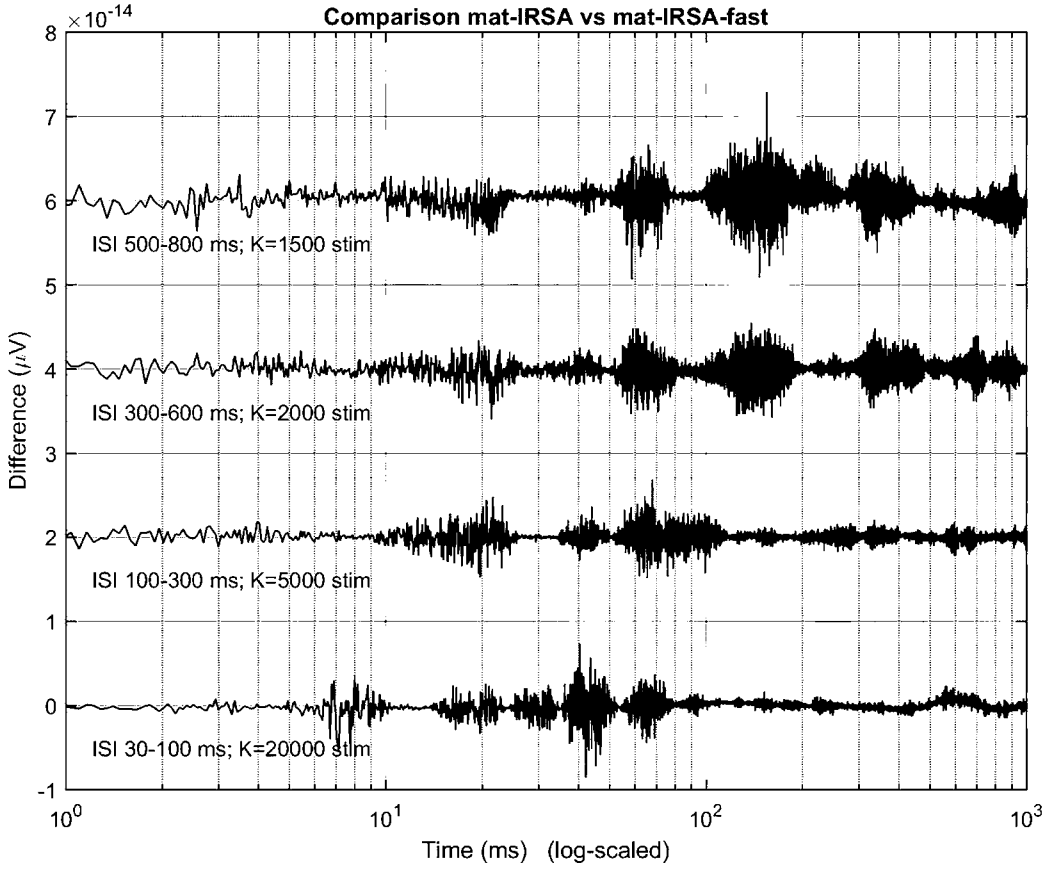
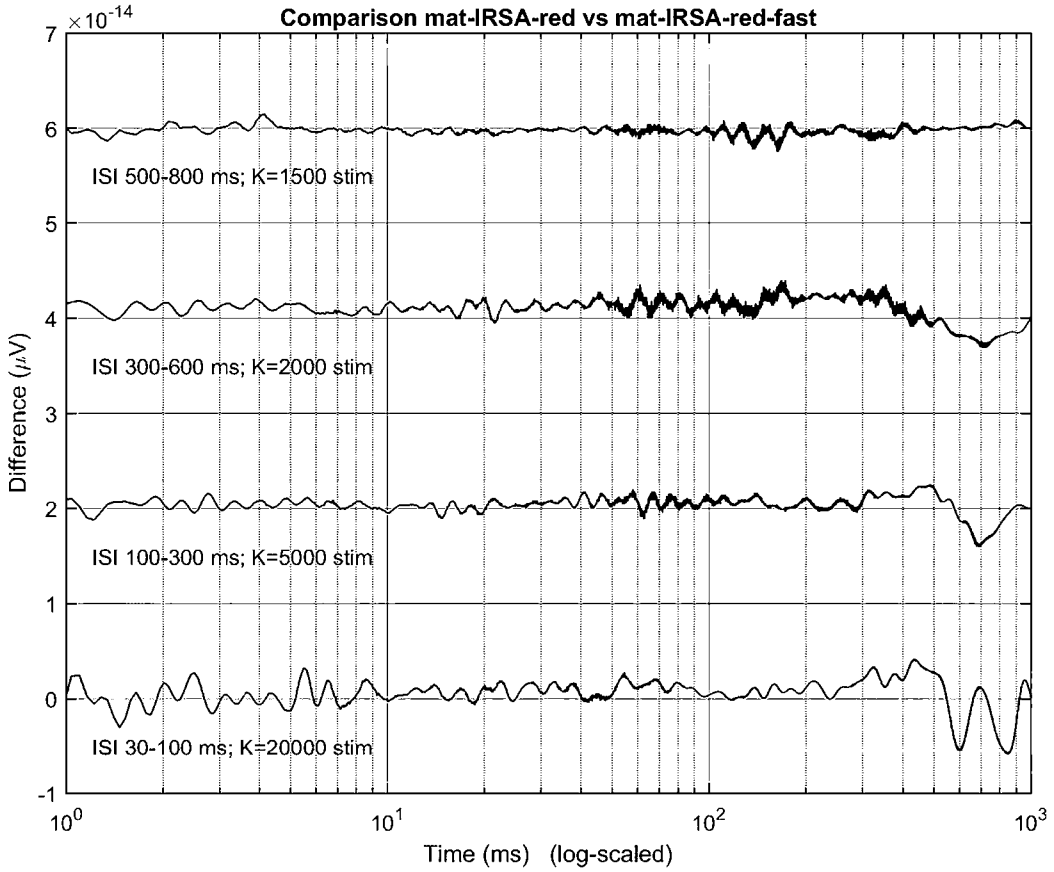


FIGURE 18


Comparison mat-IRSA vs mat-IRSA-fast
×10^-14
Difference (μV)
ISI 500-800 ms; K=1500 stim
ISI 300-600 ms; K=2000 stim
ISI 100-300 ms; K=5000 stim
ISI 30-100 ms; K=20000 stim
Time (ms) (log-scaled)


Comparison mat-IRSA-red vs mat-IRSA-red-fast
×10^-14
Difference (μV)
ISI 500-800 ms; K=1500 stim
ISI 300-600 ms; K=2000 stim
ISI 100-300 ms; K=5000 stim
ISI 30-100 ms; K=20000 stim
Time (ms) (log-scaled)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/AU2020/050311

A. CLASSIFICATION OF SUBJECT MATTER

A61B 5/0484 (2006.01)

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

Databases: PATENW, EPODOC, WPI, MEDLINE, INSPEC and NPL; classification symbols: A61B5/04/LOW, A61B5/12/LOW, A61B5/72/LOW, G06F17/10/LOW, A61B5/04845, A61B5/04001, A61B5/0484, A61B5/125, A61B5/4005, A61B5/4064, and A61B5/04012/LOW; keywords: auditory, acoustically, potentials, responses, overlapping, superposed, deconvolution, recover, iterative, repeated, random, arbitrary and like terms; applicants name 'AUSTRALIAN HEARING SERVICES' and 'UNIVERSITY OF GRANADA' searched; inventors name 'SEGURA LUNA, Jose Carlos', 'DE LA TORRE VEGA, Angel' and 'VALDERRAMA VALENZUELA, Joaquin Tomas' searched
Google Patents/Scholar keywords: auditory, acoustically, potentials, responses, overlapping, superposed, deconvolution, recover, iterative, repeated, random, arbitrary and like terms
Applicants/Inventors name searched in internal databases provided by IP Australia

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
	Documents are listed in the continuation of Box C	

[X] Further documents are listed in the continuation of Box C [ ] See patent family annex

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"D" document cited by the applicant in the international application
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
8 July 2020	08 July 2020
Name and mailing address of the ISA/AU	**Authorised officer**
AUSTRALIAN PATENT OFFICE PO BOX 200, WODEN ACT 2606, AUSTRALIA Email address: pct@ipaustralia.gov.au	Kevin Sivieng AUSTRALIAN PATENT OFFICE (ISO 9001 Quality Certified Service) Telephone No. +61262832609

INTERNATIONAL SEARCH REPORT C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		International application No. **PCT/AU2020/050311**
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	Valderrama JT, Alvarez I, de la Torre A, Segura JC, Sainz M, Vargas JL. 'Recording of auditory brainstem response at high stimulation rates using randomized stimulation and averaging'. *J Acoust Soc Am*. 2012;132(6):3856-3865. doi:10.1121/1.4764511 Whole document	1-26
A	Valderrama JT, de la Torre A, Alvarez IM, et al. 'Auditory brainstem and middle latency responses recorded at fast rates with randomized stimulation'. *J Acoust Soc Am*. 2014;136(6):3233. doi:10.1121/1.4900832 Whole document	1-26
A	Valderrama JT, de la Torre A, Medina C, Segura JC, Thornton, ARD (2016). 'Selective processing of auditory evoked responses with iterative-randomized stimulation and averaging: A strategy for evaluating the time-invariant assumption'. *Hearing Research*, *333*, 66-76. https://doi.org/10.1016/j.heares.2015.12.009 Whole document	1-26