Original Article

Noise Estimation in Electroencephalogram Signal by Using Volterra Series Coefficients

Malihe Hassani, Mohammad Reza Karami

Department of Electronic Engineering, Faculty of Electrical and Computer Engineering, Babol University of Technology, Babol, Iran

Submission: 09-11-2014 Accepted: 09-06-2015

ABSTRACT

The Volterra model is widely used for nonlinearity identification in practical applications. In this paper, we employed Volterra model to find the nonlinearity relation between electroencephalogram (EEG) signal and the noise that is a novel approach to estimate noise in EEG signal. We show that by employing this method we could considerably improve the signal to noise ratio by the ratio of at least 1.54. An important issue in implementing Volterra model is its computation complexity, especially when the degree of nonlinearity is increased. Hence, in many applications it is necessary to reduce the complexity of computation. In this paper, we use the property of EEG signal and propose a new and good approximation of delayed input signal to its adjacent samples in order to reduce the computation of finding Volterra series coefficients. The computation complexity is reduced by the ratio of at least 1/3 when the filter memory is 3.

Key words: *Electroencephalography, language, memory, noise, signal-to-noise ratio, Volterra model*

INTRODUCTION

An electroencephalogram (EEG) is a test that detects the electrical activity over a short period of time, usually 20–40 min in the brain using electrodes placed on the scalp. German physiologist and psychiatrist, Hans Berger (1873–1941) measured the first human EEG in 1924, although similar studies had been carried out in animals as early as 1870.

The common activity of millions of neurons, at the depth of several millimeters that have same spatial direction yields an electrical field, which is strong enough in order to measure from the people scalp.[1] The amplitude of an EEG signal is almost in the range of 40–100 μV with the five major frequency bands from 0 to 100 Hz.[1]

These frequency bands from low to high frequencies are respectively called delta (δ) 0.5–4 hz, theta (θ) 4–8 hz, alpha (α) 8–13 hz, beta (β) 13–30 hz, and gamma (γ) >30 Hz.

An important branch of studies regarding to EEG is artifact removal and noise canceling. EEG is nonstationary and highly vulnerable to a variety of noises, especially frequency interference, thus, eliminating the noise in the raw EEG data so as to obtain useful information that reflects the brain activities and states is an important subject for EEG.[2]

Various artifact removal methods were introduced in the literature; the most usual artifact elimination method is based on linear regression, and its purpose is to reject the most repeated artifact, that is, eye movement or blinking. This approach can approximate the concurrently measured electrooculogram (EOG) signals in the EEG and eliminate them.[3] However, because the EOG electrodes are located near the EEG electrodes, they involve a detailed amount of brain activity. Therefore, eliminating these signals may produce distortions in the EEG signals. More complete methods are established on a linear decomposition of the multichannel EEG recordings. The most common approach is the blind source separation (BSS).[4] The principal BSS assumption is based on the fact that the artifact sources are independent from brain sources, either normal or pathologic; the goal is to retrieve the original sources (brain and artifactual), by giving only sensor observations.

Normally, source separation algorithms outperform in no noise conditions, so denoising step (to remove additive noise) should be performed in the preprocessing stage of the EEG signal. A common solution for noise canceling from nonstationary signals is wavelet denoising (WD). The decomposition of a noisy signal on a wavelet basis (discrete orthogonal wavelet transform, discrete wavelet transform [DWT]) concentrates the main signal in a few wavelet coefficients that have large absolute values

Address for correspondence:
Mrs. Malihe Hassani, Department of Electronic Engineering, Faculty of Electrical and Computer Engineering,Babol University of Tecknology, Babol, Iran. E-mail: Mhassani@nit.ac.ir

without changing the noise random distribution. Therefore, denoising can be performed by putting a threshold on the wavelet coefficients.[5-8]

In this paper, the noise is considered dependent on the signal and has a nonlinear relation to it,[9] so we employed a nonlinear noise estimation approach based on Volterra series expansion to remove the noise. Recognition and compensation of undesired nonlinearity is one of the important subjects in the field of digital signal processing.[10] Adaptive Volterra filtering also has some applications to acoustic echo cancelation.[11] It was shown that the performance of the system was affected by unwanted nonlinearities in the system.[12] The influences of unwanted nonlinearities can be decreased by different ways.[13-15] Nonlinear models also have some applications in the speech processing domain and result in a better performance in this field.[16] Appling Volterra series to model a nonlinear system is widely used with great success. However, in the present conditions, methods of calculating the Volterra series coefficients are not general, despite the fact that they can be estimated for systems with known and fixed order. When the nonlinear system order is unknown, adaptive methods and algorithms are widely used for the Volterra kernel estimation. One important issue in kernel calculation is the speed of the process. If there is fast Volterra series coefficients estimation, we can use a higher order model for the system that leads to a better performance. This paper proposes a new implementation of the higher order Least mean squares (LMS) Volterra filter. A second order nonlinear system with memory is identified using the new implementation of the LMS algorithm for Volterra kernels estimation.

The new implementation uses the property of EEG signal and is based on an approximation of delayed input signal to its adjacent samples and then applying only linear adaptive LMS algorithm. After finding Volterra series coefficients, we employed Volterra model to find the nonlinearity relation between EEG signal and the noise, which is a new way to estimate noise in EEG signal. We show that, by employing this method we can considerably improve the signal to noise ratio (SNR).

THE VOLTERRA MODEL

For a discrete-time and causal nonlinear system with memory, the input-output relation of the nonlinear Volterra filter is given by:[17]

$$y\,[n]=\sum_{r=1}^{N}\sum_{k_1=0}^{M-1}\dots\sum_{k_r=0}^{M-1} h_r[k_1,....,k_r]x[n-k_1]....x[n-k_r] \qquad (1)$$

Where the functions h_r, r = 1,…,N represent the Volterra kernels, and N represents the nonlinearity degree of the model. By choosing $N = 2$, the Volterra series expansion can be expressed as:

$$\begin{aligned} y[n] = h_0 + \sum_{k_1=0}^{M-1} h_1[k_1]x[n-k_1] \\ + \sum_{k_1=0}^{M-1}\sum_{k_2=0}^{M-1} h_2[k_1,k_2]\,x[n-k_1]x[n-k_2] \end{aligned} \qquad (2)$$

Above equation contains one zero-order sentence, that is, h0, M sentence corresponding to first-order Volterra filter and M2 sentence corresponding to second-order Volterra filter. We can express the relation of input-output by applying nonlinear operators as illustrated below.

$$y[n]=h_0[x[n]]+H_1[x[n]]+H_2[x[n]] \qquad (3)$$

In above representations, the functions hi, i = 0, 1, 2 are the kernels corresponding to the nonlinear operators Hi [x[n]].

The model characterized by the Eq. 2 and Eq. 3 is called a second-order nonlinear Volterra model. In above expressions, filter memory is considered equal for all nonlinearity orders. In general form, Eq. 1 can have a different memory for each nonlinearity order. We can also consider symmetric Volterra coefficients that lead to more simplification.

The second-order Volterra kernel is an (M × M) matrix. The third-order kernel has M matrices with the size of (M × M). By considering symmetric kernels with memory *M*, we need to determine M (M + 1)/2 coefficients for the second-order Volterra kernel, while the third-order one needs M (M + 1) (M + 2)/6 coefficients.[15]

The accuracy of determining Volterra coefficients is an important issue in the practical applications. It was shown that homogeneity and non-orthogonality were two features of Volterra kernels. As a result of non-orthogonality, cross-correlation methods cannot be used to calculate Volterra kernels and the quantities of the Volterra kernels is dependent to the order of the applied Volterra filter.[18,19] However, for inputs that their amplitude density function is symmetric, like the Gaussian noise, the odd order Volterra functional is orthogonal to the even order Volterra functional. Hence, for this kind of input, a second-order Volterra model, with zero DC component, is an orthogonal model. Hence, we can apply cross-correlation technique in order to directly calculate Volterra kernel.[17] When the higher order Volterra model is used, adaptive techniques of kernel measurements are applied. Since the relation of the input-output is linear according to Volterra coefficients, using adaptive algorithms for the implementation of Volterra model is very easy. Here, we present the input vectors for different order filters. The first order input vector, when M = 3 is written as follows:

$$X^{(1)T}=\left[\,x\left[n\right]\;\; x\left[n-1\right]\;\; x\left[n-2\right]\right] \qquad (4)$$

By considering equal memories for different orders kernels, “the second order input vector” is given by:

$$X^{(2)} = X^{(1)} \times X^{(1)T} \qquad (5)$$

When the kernels are symmetric, just $x_{i,j}$ on the assumption of $i \geq j$, of $X^{(2)}$, are chosen in the input-output relation of the Volterra model. Therefore, “the second order input vector,” in the vector form is:

$$X^{(2)T} = \begin{bmatrix} x[n]^2 \quad x[n]x[n-1] \\ x[n]x[n-2] \quad x[n-1]^2 \\ x[n-1]x[n-2] \quad x[n-2]^2 \end{bmatrix} \tag{6}$$

And its length is 6. For “the third order input vector” it is proposed to express the multiple input delayed signal products in Eq. 2 by matrices elements.[11] Multiplying “the second order input vector” (Eq. 5) by the first order input vector can create these matrices. By considering equal memory for all filters in Eq. 2, $M = 3$, and symmetric kernels we have:[20]

$$X^{(2)} \times x[n] = \begin{bmatrix} x[n]^3 x[n]^2 x[n-1] x[n]^2 x[n-2] \\ \ldots x[n]x[n-1]^2 x[n]x[n-1]x[n-2] \\ \ldots\ldots x[n]x[n-2]^2 \end{bmatrix} \tag{7}$$

$$X^{(2)} \times x[n-1] = \begin{bmatrix} \ldots\ldots\ldots \\ \ldots x[n-1]^3 x[n-1]^2 x[n-2] \\ \ldots\ldots x[n-1]x[n-2]^2 \end{bmatrix} \tag{8}$$

$$X^{(2)} \times x[n-2] = \begin{bmatrix} \ldots\ldots\ldots \\ \ldots\ldots\ldots \\ \ldots\ldots x[n-2]^3 \end{bmatrix} \tag{9}$$

Therefore, “the third order input vector” composed of three matrices as illustrated in Eq. 7–9.[20] “The third order input vector” can be expressed in vector form as indicated below:

$$X^{(3)T} = \begin{bmatrix} x[n]^3 \quad x[n]^2 x[n-1] \ldots \quad x[n]x[n-2]^2 \\ x[n-1]^3 \ldots x[n-1]x[n-2]^2 \quad x[n-2]^3 \end{bmatrix} \tag{10}$$

Its size is (1 × 10). These input vectors are used to implement the LMS Volterra filter in nonlinear system identification.

ESTIMATING SECOND-ORDER VOLTERRA SERIES COEFFICIENTS

At the first step, it is required to find the nonlinear relation between noisy signal and noise. Then, we employ this relation to estimate noise present in the signal at next step. Consider the following block diagram [Figure 1]:

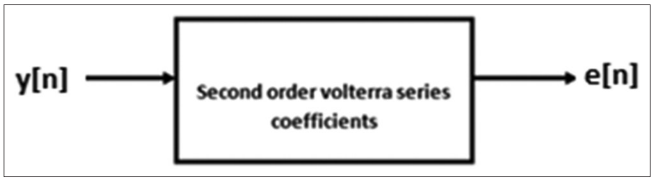

Figure 1: Relation between noisy signal and noise

In which y[n] is noisy signal and e[n] is the noise. As expressed in previous part, we have the following relation for above block diagram:

$$\begin{aligned} y[n] = h0 + \sum_{k1=0}^{M-1} h1[k1]\, e[n-k1] + \\ \sum_{k1=0}^{M-1}\sum_{k2=0}^{M-1} h2[k1,k2] \quad e[n-k1]\, e[n-k2] \end{aligned} \tag{11}$$

Where the nonlinearity degree of the model is 2 and M is the filter memory. h0, h1 (k1), h2 (k1, K2) are Volterra series coefficients. These coefficients will be obtained by LMS adaptive filter and the technique explained later is used for reducing its computation complexity. We employ these coefficients in nonlinear modeling of noise.

LEAST MEAN SQUARES ALGORITHM FOR TRUNCATED VOLTERRA SERIES MODEL

The development of a gradient type LMS adaptive algorithm for truncated Volterra series nonlinear models follows a similar method of development as for linear systems.[21]

The truncated r^{th} order Volterra series expansion is:

$$\begin{aligned} y(n) = h_0 + \sum_{k_1=0}^{M-1} h_1(k_1)\, x(n-k_1) + \\ \sum_{k_1=0}^{M-1}\sum_{k_2=0}^{M-1} h_2(k_1,k_2)\, x(n-k_1)\, x(n-k_2) + \ldots + \\ \sum_{k_1=0}^{M-1}\sum_{k_2=0}^{M-1} \ldots \sum_{k_r=0}^{M-1} h_r(k_1,\ldots,k_r)\, x(n-k_1)\, x(n-k_2) \ldots x(n-k_r) \end{aligned} \tag{12}$$

Assuming $h_0 = 0$ and $r = 2$, the weight vector can be expressed as:

$$H[n] = \begin{Bmatrix} h_1[0;n], h_1[1;n], \ldots, h_1[M-1;n], \\ h_2[0,0;n], h_2[0,1;n], \ldots, \\ h_2[0,M-1;n], h_2[1,1;n], \ldots, \\ h_2[M-1,M-1;n] \end{Bmatrix}^T \tag{13}$$

The input vector is:

$$X[n] = \begin{Bmatrix} x[n], x[n-1], \ldots, x[n-M+1], \\ x^2[n], x[n]x[n-1], \ldots, x[n]x[n-M+1], \\ x^2[n-1], \ldots, x^2[n-M+1] \end{Bmatrix}^T \tag{14}$$

Linear and quadratic coefficients are updated separately by minimizing the instantaneous square of the error:

$$J(n) = e^2(n) \quad where \quad e(n) = d(n) - \hat{d}(n) \tag{15}$$

Where $y(n)=\hat{d}(n)$ is the estimate of d (n). This results in the update equations:

$$h_1[k_1;n+1]=h_1[k_1;n]-\frac{\mu_1}{2}\frac{\partial e^2(n)}{\partial h_1[k_1;n]} \\ =h_1[k_1;n]+\mu_1 e(n)x(n-k_1) \tag{16}$$

$$h_2[k_1,k_2;n+1]=h[k_1,k_2;n]-\frac{\mu_1}{2}\frac{\partial e^2(n)}{\partial h_2[k_1,k_2;n]} \\ =h[k_1,k_2;n]+\mu_2 e(n)x(n-k_1)x(n-k_2) \tag{17}$$

Where μ_1 and μ_2 are step-sizes used to control the speed of convergence and ensure the stability of the filter.

Using the weight vector notation, H[n], we can combine the two update equations into one as the coefficient update equation:

$$e[n]=d[n]-H^T[n]X[n]$$

$$H[n+1]=H[n]+\mu X[n]e[n] \tag{18}$$

Where μ is chosen such that $0<\mu_1,\mu_2<\frac{2}{\lambda_{max}}$ and $\lambda_{\max}$ is the maximum eigenvalue of the autocorrelation matrix of the input vector X[n].

Reducing Computation Complexity to Find Volterra Series Coefficients

Consider the following second-order Volterra representation as an example:

$$y[n]=h0+\sum_{k1=0}^{2}h1[k1]x[n-k1] \\ +\sum_{k1=0}^{2}\sum_{k2=0}^{2}h2[k1,k2]\;x[n-k1]x[n-k2] \tag{19}$$

By extending above relation for M = 3 and considering symmetric kernel we have:

$$y[n]=h0+\sum_{k1=0}^{2}h1[k1]x[n-k1]+\sum_{k1=0}^{2}h2[k1]x[n-k1]^2 \\ +2h2[0,1]x[n]x[n-1] \\ +2h2[0,2]x[n]x[n-2] \\ +2h2[1,2]x[n-1]x[n-2] \tag{20}$$

In which h2 [k1] is the same as h2 [k1, k2] if k1 and k2 are equal.

An example of EEG signal is shown in terms of μV in Figure 2, Because difference between adjacent samples in EEG signal is negligible as shown in Table 1 and Figure 3,[22] we can consider x[n] x[n−1] equal by x[n]^2, x[n] x[n−2] equal by x[n−1]^2, and x[n−1]x[n−2] equal by x[n−2]^2, then we can rewrite above equation:

$$y[n]=h0+\sum_{k1=0}^{2}h1[k1]x[n-k1]+\sum_{k1=0}^{2}h2[k1]x[n-k1]^2 \\ +2h2[0,1]x[n]^2+2h2[0,2]x[n-1]^2 \\ +2h2[1,2]x[n-2]^2 \tag{21}$$

We can replace z[n] = x[n]^2:

$$y[n]=h0+\sum_{k1=0}^{2}h1[k1]x[n-k1]+\sum_{k1=0}^{2}h2[k1]z[n-\;\;k1] \\ +2h2[0,1]z[n]+2h2[0,2]z[n-1] \\ +2h2[1,2]z[n-2] \tag{22}$$

Instead of employing nonlinear adaptive filter that is very complicated, we can use the linear one and compute h2[0], h2[1], h2[2], and use these same coefficients instead of 2h2[0,1], 2h2[0,2], and 2 h2[1,2], respectively.

As a result we do not need to compute 9 coefficients, 3 coefficients need not to be computed due to symmetry,

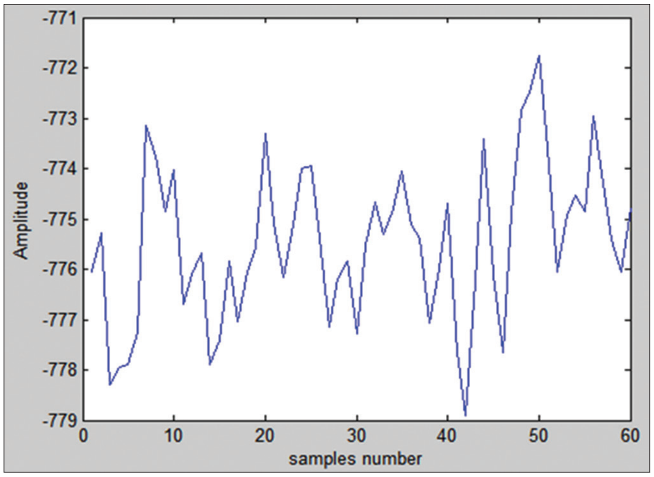

Figure 2: An example of electroencephalogram signal

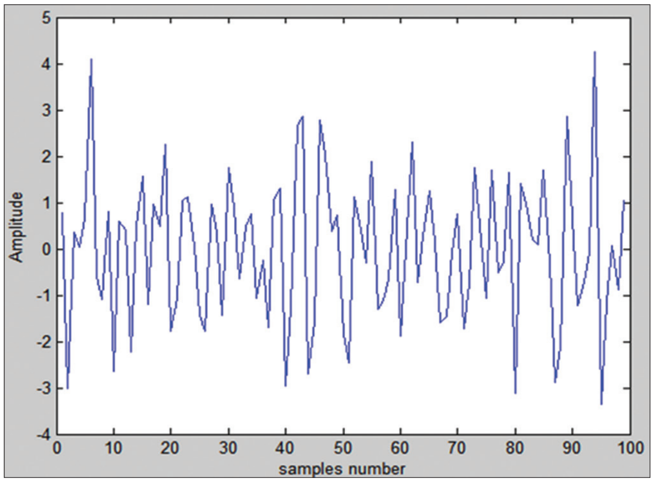

Figure 3: Difference between consecutive samples of an electroencephalogram signal

and 3 coefficients is obtained by sufficient approximation of x[n] with its adjacent samples.

Hence, instead of computing M^2 coefficients in second order kernel h2 [k1, k2], we only need to compute M coefficients. Then when M = 3, we can reduce the computation by the ratio of 1/3 even more because we do not consider the complexity of nonlinear adaptive filtering.

Estimating Noise Based on Second-Order Volterra Series Coefficients

After estimating the second-order Volterra series coefficients in the previous part, it is time to employ these coefficients for estimating noise. Consider the following system [Figure 4]:

In which y[n] is noisy signal, e[n] is estimated noise, and d[n] is denoised signal.

We use Volterra series coefficients h0, h1 (k1), h2 (k1, k2) obtained in previous part to estimate e[n], that is, noise present in the noisy signal y[n]. Then d[n] is obtained according to the following equation:

$$d[n] = y[n] - e[n] \tag{23}$$

The Evaluation Criteria

For denoising algorithms evaluation, we have used the classical criteria of the mean squared error (MSE) value and (SNR) value between the original signals and their denoised versions, as follow:

$$MSE = \frac{1}{N}\sum_{n=1}^{N}\left[x(n) - \hat{x}(n)\right]^2 \tag{24}$$

Table 1: An example of 10 consecutive sample of an EEG signal

N (n^{th} sample)	Value
1	−776.0625
2	−775.2813
3	−778.2813
4	−777.9375
5	−777.9063
6	−777.2500
7	−773.1563
8	−773.7500
9	−774.8438
10	−774.0313

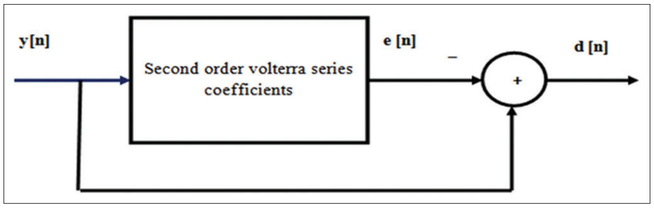

Figure 4: Block diagram of noise estimation by Volterra model

$$SNR = 10log\left|\frac{\sum_n \hat{x}(n)^2}{\sum_n\left[x(n) - \hat{x}(n)\right]^2}\right| \tag{25}$$

Where x(n) the original signal, $\hat{x}(n)$ its denoised version and N is the length of the signal. Furthermore, we have obtained the ratio of SNRs in denoised signal and noisy signal as a criterion of reduction of noise as follows:

$$R = \frac{SNR_{denoised\ \ signal}}{SNR_{noisy\ \ signal}} \tag{26}$$

Moreover, smaller MSE and larger SNR values indicated a better denoisng algorithm.

SIMULATION RESULTS

Synthetic Data

The synthetic noisy signal y has a length 500 and is analogous to EEG signal in no stationary points of view. It also has variable spectrum. The synthetic signal is a sine wave with a specific frequency in every 100 samples and has variance 0.5051, and the Gaussian noise is added to signal in two cases with mean zero and variance 0.1 and 0.3.[23,24]

As stated above, y can be expressed as follows:

$$y = \begin{Bmatrix} cos(2\pi f_1 n) + w(n) 0 \le n < 100 \\ cos(2\pi f_2 n) + w(n) 100 \le n < 200 \\ cos(2\pi f_3 n) + w(n) 200 \le n < 300 \\ cos(2\pi f_4 n) + w(n) 300 \le n < 400 \\ cos(2\pi f_5 n) + w(n) 400 \le n < 500 \end{Bmatrix} \tag{27}$$

Where w(n) is the Gaussian noise and $f_1, \ldots, f_5$ is the frequencies in the range of EEG frequency. Because EEG signal is not stationary and has different characteristics at different times, the synthetic signal is chosen so that it has the same behavior.

In order to find second-order Volterra coefficients between y[n] and e[n], we divide y[n] into periods of P points and compute the noise of current period by using Volterra coefficients of previous period, because adjacent points have approximately similar coefficients. Tables 2-4 show the results for noise variances (v) of 0.1, 0.3, and 0.5, respectively. As you see, SNR is more and more improved by decreasing P as it is expected.

In every case, several simulations were performed, and the results of denoising algorithms are shown as a mean ± standard deviation.

As nowadays classical solution for noise removal from nonstationary signals is WD, which has the capability of studying both frequency and time information simultaneously. The basic idea is simple: By decomposing the signal on a wavelet basis (DWT), we obtain a representation of the signal that concentrates most of its energy in few wavelet coefficients having large absolute values. On the contrary, most of the noise tends to be represented by the wavelet coefficients having small values, which means that its energy will not be retained by large value coefficients.

Consequently, performing a partial reconstruction of the signal using only these large coefficients (by inverse DWT) leads to an almost noise-free version of the signal.[25]

To do comparison between our proposed algorithm with other denoising algorithms, the results are computed for Daubechies, "db8," "db6," and "db4" wavelets and are shown in Tables 5-7 for noise variances of 0.1, 0.3, and 0.5, respectively. As you understand from the results, although WD is used for no stationary signals, in comparison with our nonlinear method, it does not perform well.

Table 2: Results from denoising simulated signals by changing the value of *P* (variance of noise is 0.1)

P	Snr (y[n])	Snr (d[n])	R=Snr (d[n])/Snr (y[n])	MSE
250	39.14±0.77	66.35±6.9	1.68±0.16	0.0104±3.92*10^(−4)
100	38.93±0.51	71.57±2.8	1.83±0.08	0.0102±3.96*10^(−4)
50	39.07±0.47	77.40±3.64	1.97±0.08	0.0101±5.76*10^(−4)
25	38.99±0.33	80.69±3.34	2.06±0.08	0.0103±5.09*10^(−4)
20	38.98±0.69	82.14±3.36	2.10±0.07	0.0102±7.04*10^(−4)
10	39.13±0.62	85.56±4.62	2.18±0.13	0.01±6.37*10^(−4)
5	38.82±0.7	91.33±8.13	2.34±0.22	0.0104±8.12*10^(−4)
2	39.49±0.82	97.82±7.03	2.47±0.17	0.0097±8.08*10^(−4)

Table 3: Results from denoising simulated signals by changing the value of *P* (variance of noise is 0.3)

P	Snr (y[n])	Snr (d[n])	R=Snr (d[n])/Snr (y[n])	MSE
250	17.43±0.66	32.78±2.97	1.87±0.15	0.0732±0.0077
100	16.96±0.54	40.14±2.67	2.36±0.12	0.0829±0.0029
50	17.10±0.65	45.90±3.14	2.67±0.16	0.0859±00.0059
25	17.21±0.46	51.95±2.64	2.92±0.28	0.0870±0.0043
20	17.00±0.51	53.26±1.78	3.12±0.11	0.0892±0.0041
10	16.93±0.50	60.02±1.87	3.54±0.15	0.0904±0.0049
5	17.18±0.37	64.81±4.51	3.78±0.22	0.0901±0.0044
2	17.02±0.86	70.07±8.54	4.11±0.50	0.0917±0.0074

Table 4: Results from denoising simulated signals by changing the value of *P* (variance of noise is 0.5)

P	Snr (y[n])	Snr (d[n])	R = Snr (d[n])/Snr (y[n])	MSE
250	6.87±0.27	13.18±1.56	1.91±0.17	0.16±0.01
100	6.77±0.36	20.45±2.99	3.01±0.35	0.2±0.018
50	6.79±0.47	27.03±3.17	3.96±0.26	0.21±0.01
25	6.85±0.78	30.56±1.51	4.49±0.43	0.23±0.02
20	7.05±0.59	32.99±3.83	4.67±0.42	0.23±0.01
10	6.49±0.71	39.02±3.46	6.02±0.39	0.25±0.02
5	6.99±0.74	46.83±4.83	6.72±0.72	0.24±0.018
2	6.89±0.68	53.46±4.56	7.79±0.87	0.24±0.019

Furthermore, we depicted the amount of R in different variances of 0.1, 0.3, and 0.5 for Daubechies, "db8," "db6," and "db4" wavelets and nonlinear modeling of noise when $P = 2$ in Figure 5.

We depicted noisy signal y[n], estimated noise e[n], and denoised signal d[n] for 100 sample length (in order to see the details) for the case that variance of noise is 0.1 in Figures 6 and 7 for $P = 2$ and 250, respectively.

Furthermore, in Figure 8 we depicted the histogram of estimated noise when $P = 250$ and variance of noise is 0.3, and then we calculated the goodness of fit by applying Chi-square test in different variances of noise and $P = 250$ for level significant of 0.05. As we know the $P > 0.05$ means, there is no significant difference between the estimated noise and the existing noise in the signal. We performed the simulation several times and Table 8 shows the percentage of times that the P value of Chi-square test is >0.05 in different variances of noise, that is, denoted by G. We can see, in most of the time the $P > 0.05$ and also G is increased by increasing variance of noise.

Table 5: Results from denoising simulated signals by using Daubechies wavelets (variance of noise is 0.1)

	Snr(y[n])	Snr(d[n])	R=Snr (d[n])/Snr(y[n])	MSE
Db4	39.20±0.57	54.34±1.18	1.38±0.02	$0.5008 \pm 2.66 \times 10^{-4}$
Db6	39.11±0.75	57.91±1.41	1.47±0.03	$0.5007 \pm 2.17 \times 10^{-4}$
Db8	39.29±0.51	59.95±1.46	1.52±0.02	$0.5007 \pm 1.41 \times 10^{-4}$

Table 6: Results from denoising simulated signals by using Daubechies wavelets (variance of noise is 0.3)

	Snr(y[n])	Snr(d[n])	R=Snr (d[n])/Snr(y[n])	MSE
Db4	17.50±0.31	37.60±1.28	2.14±0.09	0.5103±0.0017
Db6	17.17±0.78	37.38±2.01	2.18±0.011	0.5109±0.0023
Db8	16.92±0.72	38.20±2.33	2.25±0.019	0.5107±0.0027

Table 7: Results from denoising simulated signals by using Daubechies wavelets (variance of noise is 0.5)

	Snr(y[n])	Snr(d[n])	R=Snr (d[n])/Snr(y[n])	MSE
Db4	6.63±0.49	27.80±1.79	4.19±0.26	0.52±0.007
Db6	7.07±0.65	28.3±1.48	4.02±0.4	0.52±0.0053
Db8	7±0.7	28.34±1.65]	4.07±0.44	0.52±0.004

Table 8: Percentage of times that $P > 0.05$ in different variances of noise

Variance of noise	G
0.1	%75
0.3	%81
0.5	%85

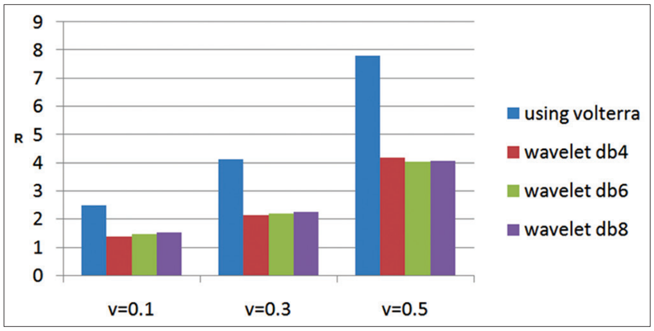

Figure 5: The amount of R in different variances of 0.1, 0.3 and 0.5 for Daubechies, “db8,” “db6,” and “db4” wavelets and nonlinear modeling of noise when *P* = 2

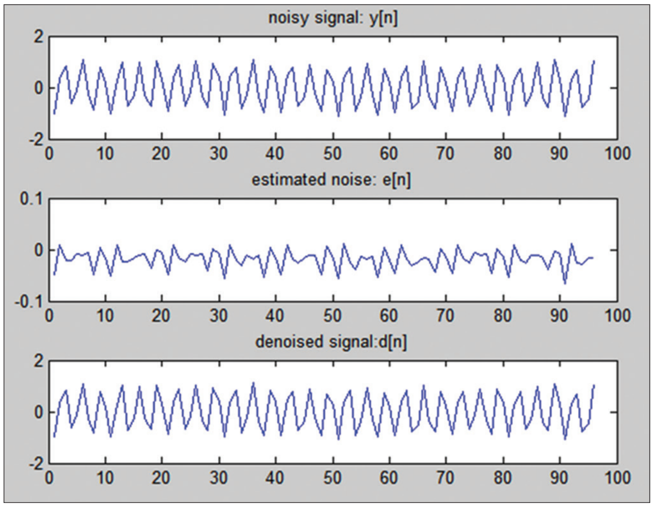

Figure 7: Noisy signal y[n], estimated noise e[n] (it is depicted bigger for the purpose of better representation), and denoised signal d[n], for *P* = 250

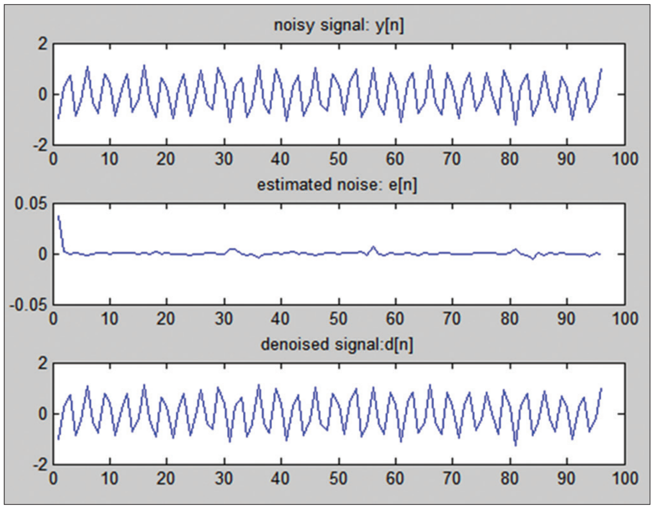

Figure 6: Noisy signal y[n], estimated noise e[n] (it is depicted bigger for the purpose of better representation), and denoised signal d[n], for *P* = 2

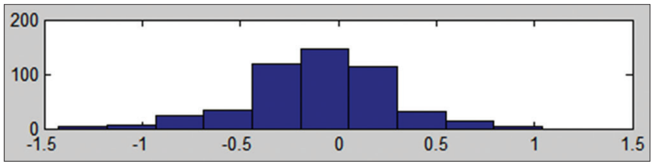

Figure 8: Histogram of estimated noise when *P* = 250 and variance of noise is 0.3

Real Data

EEG has small amplitude in the range of μV. There are five major frequency bands in the EEG. These frequency bands from low to high frequencies are called delta (δ), theta (θ), alpha (α), beta (β), and gamma (γ), respectively.[1]

Delta is the frequency range up to 4 Hz. It has the highest amplitude. It exists normally in adults in slow wave sleep. It is also seen normally in babies. Theta waves lie within the range of 4 Hz to 7 Hz. Theta is seen normally in young children, in drowsiness in older children and adults. Alpha is the frequency range from 7 Hz to 14 Hz. It is seen in normal adults during relaxed and mentally inactive awareness. Beta waves lie within the range of 15 Hz to about 30 Hz. It is associated with expectancy states and tension. Gamma is the frequency range approximately 30–100 Hz. Usually, it is not of clinical interests and thus often filtered out in many practical situations.

The real data were recorded from 32 electrodes placed at the standard positions of the 10–20 international system with a Biosemi Active Two system and are thus reference free.[22,26] They are not recorded from the scalp. The average signal from the two mastoid electrodes was used for referencing. Arbitrary referencing schemes can be implemented by subtracting the reference channel(s) from all other channels. The sampling rate is 2048 Hz. Note that the data might contain artifacts coming from eye-blinks, eye-movements, muscle-activity, etc. This means that the data are recorded under real conditions.

Human data based on five disabled and four healthy subjects. The disabled subjects (1–5) were all wheelchair-bound but had varying communication and limb muscle control abilities. The four healthy subjects (6–9) were all male PhD students, ages 30, and had no known neurological deficits. Signals were recorded at 2048 Hz sampling rate from 32 electrodes placed at the standard positions of the 10–20 international system.

We run the proposed algorithm for real data of length 0.48 s (1000 samples), but for the purpose of better representation, Figure 9 shows 100 samples of an EEG noisy signal, y[n], its estimated noise e[n] and denoised EEG signal d[n].

In order to compute and investigate the evaluation criteria for real data, at first we denoised it by our proposed method and consider it as an original signal and then add it Gaussian noise with mean zero and three different variances. Tables 9-11 show the results in the case of real data for noise variances of 0.1, 0.3, and 0.5, respectively.

In order to do a qualitative comparison of our denoising algorithm with other algorithms in the case of real data, we ask

Table 9: Results from denoising real data by changing the value of *P* (variance of noise is 0.1)

P	Snr (y[n])	Snr (d[n])	R=Snr (d[n])/Snr (y[n])	MSE
50	41.12±0.4	80.16±3.47	1.94±0.07	0.0096±5.5*10^(−4)
25	40.99±0.59	84.80±3.40	2.06±0.06	0.0097±5.9*10^(−4)
2	41.02±0.42	96.22±8.48	2.34±0.2	0.0095±5.11*10^(−4)

Table 10: Results from denoising real data by changing the value of *P* (variance of noise is 0.3)

P	Snr (y[n])	Snr (d[n])	R=Snr (d[n])/Snr (y[n])	MSE
50	19.01±0.6	38.24±1.08	2.03±0.04	0.0888±0.0051
25	18.66±0.56	49.62±1.88	2.61±0.05	0.0746±0.0036
2	18.95±0.6	67.38±5.16	3.54±0.22	0.0634±0.0019

Table 11: Results from denoising real data by changing the value of *P* (variance of noise is 0.5)

P	Snr (y[n])	Snr (d[n])	R=Snr (d[n])/Snr (y[n])	MSE
50	8.23±0.73	12.83±3.23	1.54±0.31	0.13±0.03
25	8.30±0.94	22.85±3.1	2.77±0.26	0.15±0.03
2	8.7±1.74	42.08±4.61	4.98±1.11	0.24±0.03

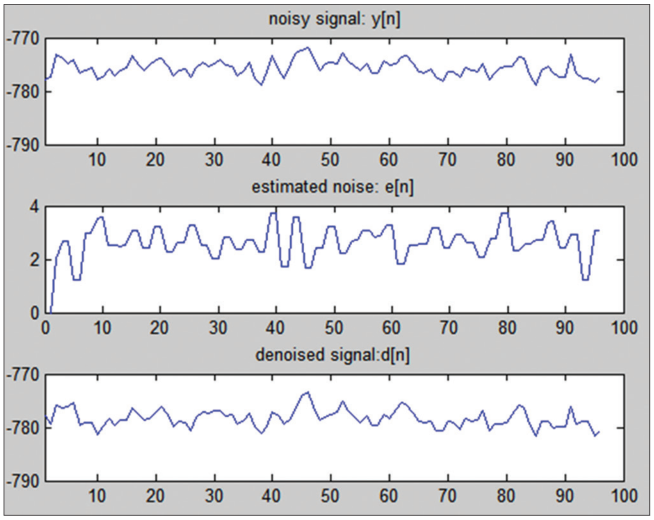

Figure 9: Noisy signal y[n], estimated noise e[n] (it is depicted bigger for the purpose of better representation), and denoised signal d[n], for *P* = 2, when the data is real

some expert to grade the quality of the denoised signal. We did it for all data and averaged the grades for final comparison. Note that we compared the results of our method when $P = 2$ in different variances of noise with wavelet db4.

Table 12 shows the rating system for the quality of denoised signal and Table 13 shows the mean grades for it.

Table 12: The rating system for the quality of denoised signal

Expert grade	Quality
5	Excellent (very well denoised)
4	Very good
3	Good
2	Bad
1	Very bad

Table 13: Mean grades for the quality of denoised signal

Method of denoising	Mean grade of experts
Wavelet (Db4)	3.7
Using Volterra (v=0.1)	4.1
Using Volterra (v=0.3)	4.4
Using Volterra (v=0.5)	4.7

CONCLUSION AND FUTURE WORK

This paper uses the property of EEG signal and proposes a new and good approximation of delayed input signal to its adjacent samples to reduce the complexity of Volterra series coefficients computation in nonlinear modeling of EEG signal. This method is important in practical applications in which computing higher order kernels by using nonlinear adaptive filtering is very complicated. By using the proposed technique, we only need to compute M coefficients instead of M^2 so reducing the computation by the ratio of at least 1/3, when M = 3. After finding Volterra series coefficients, we employed Volterra model to find the nonlinearity relation between EEG signal and the noise that is a new way to estimate noise in EEG signal. By employing this method, we can considerably improve the SNR in comparison with WD. We have to combine the linear and nonlinear method of estimating noise in the EEG signal in the future.

REFERENCES

1. Sanei S, Chambers J. EEG Signal Processing. Center of Digital Signal Processing Cardiff University, UK: John Wiley & Sons; 2007.
2. Daliri MR, Taghizadeh M, Niksirat KS. EEG Signature of Object Categorization from Event-related Potentials. J Med Signals Sens 2013;3:37-44.
3. Croft RJ, Barry RJ. Removal of ocular artifact from the EEG: A review. Neurophysiol Clin 2000;30:5-19.
4. Romo-Vázquez R, Vélez-Pérez H, Ranta R, Louis-Dorr V, Maquin D, Maillard L. Blind source separation, wavelet denoising and discriminant analysis for EEG artifacts and noise cancelling. Biomed Signal Process Control 2012;7:389-400.
5. Antoniadis A, Bigot J, Sapatinas T. Wavelet estimators in nonparametric regression: A comparative simulation study. J Stat Softw 2001;6:1-83. Available from: http://www-lmc.imag.fr/lmc-sms/Anestis.Antoniadis/HTTP/publis-anto.html. [Last accessed on 2015 Jul].
6. Antoniadis A. Wavelet methods in statistics: Some recent developments and their applications. Stat Surv 2007;1:16-55.
7. Serbes G, Aydin N. Denoising performance of modified dual-tree complex wavelet transform for processing quadrature embolic Doppler signals. Med Biol Eng Comput 2014;52:29-43.
8. Lavielle M, Lebarbier E. An application of MCMC methods to the multiple change-points problem. Signal Processing 2001;81:39-53.
9. Vahabi Z, Amirfattahi R, Mirzaei A. Enhancing P300 wave of BCI systems via negentropy in adaptive wavelet denoising. J Med Signals Sens 2011;1:165-76.
10. Stefanou A, Gielen G. A volterra seiries nonlinear model of the

sampling distorion in flash ADCs due to substrate noise coupling. IEEE Trans Circuits Syst 2011;58:877-81.

11. Azpicueta-Ruiz LA, Figueiras-vidal AR, Kellermann W, Arenas-Garcia J. Enhanced adaptive volterra filtering by automatic attenuation of memory regions and its application to acoustic echo cancellation. IEEE Trans Signal process 2013;61:2745-50.
12. Tsimbinos J. Identification and Compensation of Nonlinear Distortion, Thesis, Institute for Telecommunications Research, University of South Australia; 1995. p. 237. Available from: http://www.unisa.edu.au/html. [Last accessed on 2015 Jul].
13. Stenger A, Kellermann W. RLS-Adapted Polynomial Filter for Nonlinear Acoustic Echo Cancelling. Proceedings of the X EUSIPCO, Tampere, Finland; 2000. p. 1867-70.
14. Stenger A, Kellermann W. Adaptation of a memoryless preprocessor for nonlinear Acoustic echo cancelling. Signal Processing 2000;80:1747-60.
15. Küch F, Kellermann W. Nonlinear Echo Cancellation Using a Second Order Volterra Filter. ICCASSP; 2002.
16. Vladimir Despotović, Norbert Görtz, Zoran Perić. Improved Non-Linear Long-Term Predictors based on Volterra Filters. Elmar, Zadar: IEEE; 2012. p. 231-4.
17. Budura G, Naforniţa I, Kernels Measurement Techniques for Constructing Nonlinear Models. Vol. 1 Scientific Bulletin of the Politehnica, Romania: University of Timişoara; 2002. p. 190-5.
18. Schetzen M. The Volterra and Wiener Theories of Nonlinear Systems. New York: John Wiley and Sons; 1980.
19. Budura G, Botoca C. La construction d'un modèle non linéaire a l'aide de séries Volterra et Wiener, Revue Roumaine des Sciences Techniques, Serie Electrotechnique et Energetique. no. 4. 2005.
20. Budura G, Botoca C. Nonlinearities Identification Using The LMS Volterra Filter, 2005 WSEAS International Conference on Dynamical Systems and Control, Venice, Italy; November 2-4, 2005. p. 148-53.
21. Ogunfunmi T. Adaptive Nonlinear System Identification, The Volterra and Wiener Model Approaches. Santa Clara university, California: Springer Series on Signals and Communications Technology; 2007.
22. Hoffmann U, Vesin JM, Ebrahimi AT, Diserens K. An effcient P300-based brain computer interface for disabled subjects. Journal of Neuroscience Methods 2008;167:115-25.
23. Mert A, Akan A. Hilbert-Huang Transform Based Hierarchical Clustering for EEG Denoising, Signal Processing Conference (EUSIPCO), 2013 Proceedings of the 21st European; 2013.
24. Vorobyov S, Cichocki A. Blind noise reduction for multisensory signals using ICA and subspace filtering, with application to EEG analysis. Biol Cybern 2002;86:293-303.
25. Ranta R, Louis-Dorr V, Heinrich C, Wolf D. Iterative wavelet-based denoising methods and robust outlier detection. IEEE Signal Process Lett 2005;12:557-60.
26. Available from: http://www.bci.epfl.ch/p300. [Last accessed on 2015 Jul].

How to cite this article: Hassani M, Karami MR. Noise Estimation in Electroencephalogram Signal by Using Volterra Series Coefficients. J Med Sign Sence 2015;5:192-200.

Source of Support: Nil, **Conflict of Interest:** None declared

BIOGRAPHIES

Malihe Hassani has received her Bs.c & Ms.c in in electrical and electronic engineering 2002 & 2005 respectively. Now she is PhD candidate in Babol Noshirvani Institute of Tecknology,department of electrical and electronic engineering, she published more than 7 articled in journals and conferences. Her research interests are digital signal proessing,biomedical signal processing and image proessing.

E-mail: Mhassani@nit.ac.ir

Mohammad Reza Karami has received the Bs.c in electrical and electronic engineering in 1992, Ms.c of signal processing in 1994, and PhD in 1998 in biomedical engineering from I.N.P.L d'Nancy of France. He is now the Associate professor with the Department of Electrical and Computer Engineering, Babol University of Technology. Since 1998 his research is in signal and speech processing. He published mor than 100 articles in journals and conferences. He teaches Digital signal processing, Biomedical signal processing and speech processing in university. His research interests include Speech, Image and signal processing.

E-mail: mkarami@nit.ac.ir