Biomedical Signal Processing and Control 20 (2015) 10–15

Contents lists available at ScienceDirect

Biomedical Signal Processing and Control

journal homepage: www.elsevier.com/locate/bspc

Short Communication

Detecting signal quality by ensemble empirical mode decomposition and Monte Carlo verification

Yun-Kai Lee[a], Sung-Chun Tang[b], Jia-Rong Yeh[c], Jiann-Shing Shieh[a,c,*]

[a] Department of Mechanical Engineering, and Innovation Center for Big Data and Digital Convergence, Yuan Ze University, Chung-Li 320, Taiwan
[b] Stroke Center and Department of Neurology, National Taiwan University Hospital, Taipei, Taiwan
[c] Center for Dynamical Biomarkers and Translational Medicine, National Central University, Chung-Li 32001, Taiwan

A R T I C L E I N F O

Article history:
Received 15 April 2014
Received in revised form
23 December 2014
Accepted 28 January 2015
Available online 22 April 2015

Keywords:
Bio-signal
Ensemble empirical mode decomposition (EEMD)
Intrinsic mode functions (IMFs)
Monte Carlo verification
White noise

A B S T R A C T

The measurements of bio-signals are always subject to interference from noise, which would be able to affect the research results. In the present study, we introduce a technique to detect the signal quality by using ensemble empirical mode decomposition (EEMD) and Monte Carlo verification. We first decompose the original signals into several intrinsic mode functions (IMFs) and calculate the average distances between the signal IMFs and the negative (−1) slope line in Monte Carlo verification. Then, the approximate amount of white noise percent level in original signal could be obtained via corresponding to the created curve of distance and noise percent. This new proposed technique makes the approximate white noise percent level could be obtained much easier via a simple distance index through the EEMD and Monte Carlo verification methods.

1. Introduction

Measurements of physiological signals are always subject to interference from noise, which might affect the research result. Also, in the natural systems, physiological signals are non-linear and non-stationary. Some linear and stationary assumptions might probably be inaccurate. In 1998, empirical mode decomposition (EMD) was proposed by Huang et al. [1] as an adaptive time–frequency data analysis algorithm that could be applied in both non-linear and non-stationary signals. EMD could decompose the signal to different intrinsic mode functions (IMFs) from the highest to the lowest frequency in time domain. This algorithm was widely applied in many research areas such as oceans, atmosphere, earthquake analyses, etc. Briefly, the steps of EMD algorithm included as follows. Firstly, it obtains the local maxima and minima as upper and lower envelopes by cubic spline. Also, the mean envelope could be obtained in this step. Then, it decomposes the signal via calculating the difference between the original signal and the mean envelope. The result is the first IMF. Finally, the decomposed signal would be treated as the input data. Then repeat the steps and decompose the signal until the signal becomes a monotonic function. The equation of EMD could be expressed as follows:

$$x(t) = \sum_{i=1}^{n} C_i(t) + R_n(t), \tag{1}$$

where $x(t)$ is the original recorded signal data in time domain, $C_i(t)$ is the ith IMF, and $R_n(t)$ is the residue.

Therefore, we could reconstruct the signal by choosing appropriate IMFs to filter the noises in the signal [2–6]. However, one of the major drawbacks of EMD algorithm is the mode-mixing problem, which is defined as a signal IMF consists either of signals on widely disparate scales, or a signal of a similar scale located in different IMF components [2]. To overcome the mode-mixing problem, Huang et al. [2], in 2009, proposed another noise-assisted algorithm named ensemble empirical mode decomposition (EEMD). The steps of EEMD included:

Step (1) Create a white noise background which would populate the whole time–frequency space and add the original signal into the white noise.

Step (2) Decompose the signal into IMFs.

Step (3) Repeat these two steps. In the end, the means of corresponding IMFs of the decompositions could be obtained as the final result [2,3].

* Corresponding author at: Department of Mechanical Engineering, Yuan Ze University, 135 Yuan-Tung Road, Chung-Li 32003, Taiwan. Tel.: +886 3 4638800x2470; fax: +886 3 4558013.
E-mail address: jsshieh@saturn.yzu.edu.tw (J.-S. Shieh).

http://dx.doi.org/10.1016/j.bspc.2015.01.010
1746-8094/

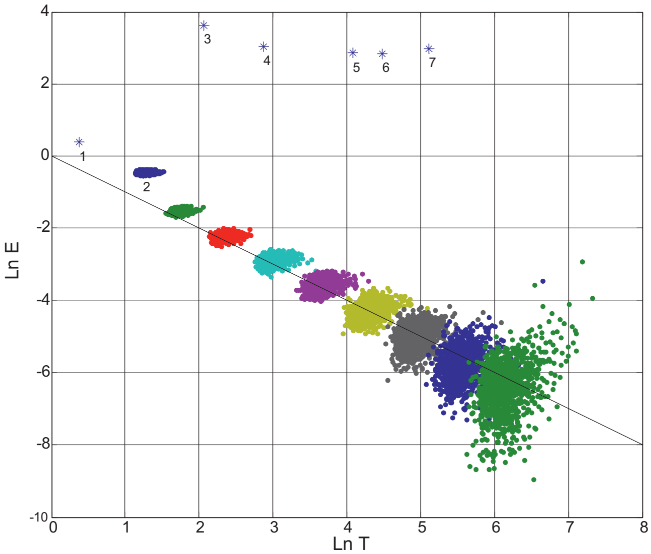

Fig. 1. Monte Carlo verification of 7 IMFs of an ECG signal to which has been added 10% white noise.

When the signal is added to the uniform white noise background, the bits of the signal would project onto the appropriate scale [3,4]. The purpose of creating a white noise background and adding it into the signal is to make the values of EMD upper and lower envelopes more easily obtained and further improve the mode-mixing problem.

Based on the numerical experiments in the previous study [7], Monte Carlo verification could be applied in determining the relationships within signal IMFs. For example, Fig. 1 shows the Monte Carlo verification of decomposition for an ECG signal with 10% standard deviation (SD) white noise added [7]. The groups of dots from the upper left to the lower right indicate the energy density as a function of the averaged period for white noise IMFs [3,4,7–9]. The significant asterisks with numbers on the top are the signal IMFs of the ECG signal with 10% white noise.

For white noise, the slope between energy density and averaged period is equal to −1. In other words, if the slope of IMF of the original signals is more closely approach −1, it means there are more noise and less signal information within this IMF [3,4,7–9]. The distance between the line and the IMF components would decrease along with an increasing percentage of white noise. In the present study, we introduce a new approach to detect the signal quality based on using EEMD and Monte Carlo verification to calculate this distance corresponding to the approximate amount of white noise percent in the original signal.

2. Methods

2.1. Creating a corresponding curve

To realize the white noise percent level, we first have to create a corresponding curve. The curve included the white noise percent in x-axis and the averaged distance index in y-axis so that the noise percent would be able to correspond with the average distance. The curve was created as the following steps:

Step (1) Add different white noise percent: in amounts ranging from 10% to 100% were added into the original signal.

Step (2) Decompose the signal via EEMD method: decompose the signal to 7 IMFs. In the present study, the amplitude of the added white noise background in EEMD is 0.2 × standard deviation (SD) of the original signal, same as the suggested amplitude of the added white noise background in the previous literature [2].

Step (3) Perform Monte Carlo verification: in this step, the characteristics of IMF components and the line could be seen.

Step (4) Normalize IMF 1 onto the negative (−1) slope line on Monte Carlo verification figure: in order to avoid the influence of the added white noise in EEMD, the added white noise background would probably further affect the average distance. In this step, we shifted all IMFs, and normalized IMF 1 onto the negative slope line so that the distance between IMF 1 and the negative slope line is 0. Normally, the frequency of noise is higher, and higher frequency would be decomposed in the first few IMFs. IMF 1 also included the added white noise in EEMD. Through the normalization, the effect of added white noise background in EEMD could be avoided. The positions of IMFs 2–7 were also been shifted following the shifted IMF 1.

Step (5) Calculate the average distance between the rest IMFs (IMFs 2–7) and the negative slope line in Monte Carlo verification of every percentage scale: finally, the curve could be created.

To verify the distance is decreasing along with the increasing noise percent, we first simulate sine and cosine waves. The white noise from 10% to 100% was added to the original sine and cosine waves. Then, we follow the steps 2–5 as described previously, and the averaged distance curve could be obtained. Because the signal is random, we repeat the steps for 10 times in each percent scale. Therefore, 10 averaged distance values in every percent scale would be obtained in the end. We then combine these 10 values to one curve by taking the mean in each percent scale. The decreased averaged distance with an increasing noise percent could be seen in Fig. 2. This figure demonstrates the tendency that the average distance will approach to a line with slope is equal to −1 if the original signal has more white noise.

2.2. Real ECG signal

To evaluate the real ECG signal, we have to create the corresponding curve of an actual ECG signal. Also, the ECG signal that has less interference is required. We first downloaded 30 fetal ECG signals of the first thorax electrode from non-invasive fetal electrocardiogram database in PhysioBank on PhysioNet website [10], and randomly took part of the signal data with length 10,000 of the downloaded 30 ECG signals to create the corresponding curve. The record numbers of thorax 1 electrode of the 30 ECG data are ecgca 102, 115, 127, 154, 192, 224, 252, 274, 290, 300, 308, 323, 368, 384, 392, 410, 416, 436, 444, 445, 473, 515, 571, 585, 595, 597, 621, 629, 649, 659 in the database. The methods of creating the corresponding curve were same as the steps mentioned previously in Section 2.1. Fig. 3 demonstrates the corresponding curve of 30 ECG data. After the procedures mentioned in Section 2.1 were completed, we combined the 30 ECG corresponding curves into one curve by taking the mean value in each percent scale. The averaged distance values and the corresponding curve are shown in Table 1 and Fig. 3.

Table 1
Mean distance value and standard deviation of the 30 data mean curve.

White noise (%)	Mean ± standard deviation (SD)
10	3.621 ± 0.023
20	2.792 ± 0.022
30	2.281 ± 0.021
40	1.928 ± 0.016
50	1.652 ± 0.02
60	1.449 ± 0.027
70	1.303 ± 0.027
80	1.206 ± 0.025
90	1.132 ± 0.041
100	1.033 ± 0.023