IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 25, Issue 3, Ser. II (May. – June. 2023), PP 25-32
www.iosrjournals.org

Improving The Hilbert Huang Transfrom By Using Fuzzy Logic

Thang Le Anh[*1], Trung Doang Van[1], Phuong Pham Kim[1], Huyen Le Xuan[2]
[1] *Faculty of the Information Technology, Hanoi University of Industry, Number 298, Cau Dien Street, Bac Tu Liem District, Ha Noi, Vietnam*
[2] *Faculty of the English Language, Hanoi University of Industry, Number 298, Cau Dien Street, Bac Tu Liem District, Ha Noi, Vietnam*

Abstract:
Hilbert-Huang Transform (HHT), proposed by N. E. Huang in 1998, is a novel algorithm for nonlinear and non-stationary signal processing. However, it still exists end effects problem which needs solving. In this paper, we propose to apply Fuzzy Logic for predicting extended data at two end sides of the original data before executing Empirical Mode Decomposition (EMD) process to solve the end effects problem. Our simulations showed that the method gets better results than some available methods.
Keywords: *HHT; EMD; IMF; Fuzzy Logic.*

Date of Submission: 28-05-2023 Date of Acceptance: 08-06-2023

I.Introduction

The Hilbert Huang Transform consists of two parts. The key part is the decomposition of a signal into a finite number of IMFs. During this part, the algorithm uses a cubic spline to connect all local maxima and minima of data to produce an upper and lower envelope of data. When endpoints are not extrema, the spline can swing widely. The effects not only influence the neighborhood of the endpoints but also propagate into the interior of the data. Recently, some researchers have applied Neural Networks (NN) for extension data before executing the EMD process to solve the end effect problem [5], [6], [7]. However, the training process of NN consumers much time. Therefore, it hardly applies to real-time applications. In this paper, we propose using Fuzzy Logic to predict extended data on both ends sides of the original data before carrying out the EMD process. Our proposed method is simple; it just performs a simple one-pass operation on the training set [8]. So, it consumes less time than the NN method does. The simulation results show the accurate prediction of this method. It also shows the comparison between the envelopes of data created by this method and those created by the Mirror Method (MM) and Improved Slope Base Method, respectively.

II. The empirical mode decomposition method (EMD)

The empirical mode decomposition method is necessary to deal with data from non-stationary and nonlinear processes [1]. The decomposition is based on the simple assumption that any data consists of different simple intrinsic modes of oscillations. Each intrinsic mode, linear or nonlinear, represents a simple oscillation, which will have the same number of extrema and zero-crossings. Furthermore, the oscillation will also be symmetric with respect to the “local mean.” At any given time, the data may have many different coexisting modes of oscillation, one superimposing on the others. The result is the final complicated data. Each of these oscillatory modes is represented by an intrinsic mode function (IMF) with the following definition:
(1) in the whole dataset, the number of extrema and the number of zero-crossings must either equal or differ at most by one.
(2) at any point, the mean value of the envelope defined by the local maxima and, the envelope defined by the local minima is zero.
With the above definition for the IMF, the algorithm to decompose any data set x(t) into IFM is illustrated in figure 1 below.

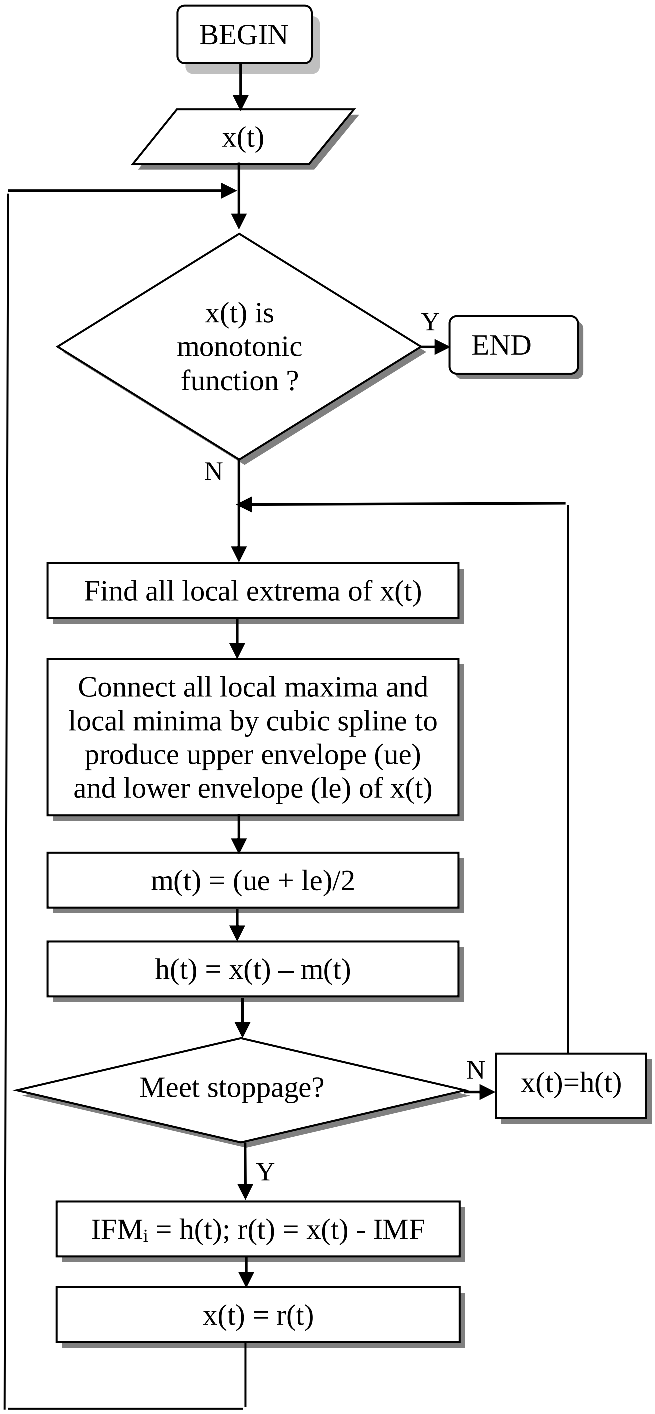

Figure 1. Flowchart of EMD algorithm

Where, the monotonic function is function that has maximum two extrema. There were two stoppage criteria. The first one was used by Huang et al. (1998). This stoppage criterion is determined by using a Cauchy type of convergence test. Specifically, the test requires the normalized squared difference between two successive sifting operations defined as:

$$SD_k = \frac{\sum_{t=0}^{T} \left| h_{k-1}(t) - h_k(t) \right|^2}{\sum_{t=0}^{T} h_{k-1}(t)^2} \qquad (1)$$

If this squared difference SD_k is smaller than a predetermined value, the sifting process will be stopped.

The second criterion was proposed by Huang et al. (1999, 2003). This criterion is based on the agreement of the number of zero-crossings and extrema. Specifically, an S-number is pre-selected. The sifting process will stop only after S consecutive times when the numbers of zero-crossings and extrema stay the same and are equal or differ at most by one. For optimal sifting, the range of S-numbers should be set between 4 and 8.

Summing up all IMFs and residue we obtain:

$$x(t) = \sum_{i=1}^{n} IFM_i + r_n \qquad (2)$$

III.Training of Fuzzy Logic system using a table–lookup scheme

Assume, we have a set of input and output data pairs:

$$(x_1^1, x_2^1; y^1), (x_1^2, x_2^2; y^2), ... \qquad (3)$$

The task here is to generate a set of fuzzy IF-THEN rules from the given input-output pairs of (3), and then use them to determine the logic system $f : (x_1, x_2) \to y$.

The approach divides into five steps as follows.

- ***Step 1***: Divide each domain interval of input and output variable into 2*N+1 regions (N can be different for each variable and the width of each region can be equal or unequal), denoted by SN (Small N), …, S1 (Small 1), CE (Center), B1 (Big 1), …, BN (Big N) and then assign each region to one fuzzy membership function.
- ***Step 2***: This step further divides into three small steps. First, calculate the degree of given x_1^i, x_2^i, y^i in different regions. Second, assign the given x_1^i, x_2^i, y^i to region with maximum degree. Finally, obtain one IF-THEN rule from one pair of given input-output data. For example, for a given input-output data pair (x_1^1, x_2^1, y^1), after the first and second steps, we specify x_1^1 in region S1, x_2^1 in region B2, and y^1 in region CE then we obtain rule 1: IF x_1 is S1 and x_2 is B2 THEN y is CE.
- ***Step 3***: Calculate the degree of each rule base on product strategy.

$$D(rule) = \mu_A(x_1) \times \mu_B(x_2) \times \mu_C(y) \qquad (4)$$

For example, to calculate the degree of rule: IF x_1 is S1 and x_2 is B2 THEN y is CE.

$$D(rule) = \mu_{S1}(x_1) \times \mu_{B2}(x_2) \times \mu_{CE}(y) = 0.8 \times 0.7 \times 1.0 = 0.56$$

where, x_1 has a degree of 0.8 in *S1*, x_2 has a degree of 0.7 in *B2* and *y* has a degree of 1.0 in *CE*.

- ***Step 4***: Create a combine Fuzzy rule base. This step avoids conflict rules and two identical rules exist in the combined fuzzy rule base. Conflict rules are rules which have the same IF part but a different THEN part. To solve conflict rules is to accept only the rule from the conflict group that has a maximum degree. We use table – lookup presents a fuzzy rule base. We fill cells of the rule base by the rules; if there is more than one rule on one cell of the fuzzy rule base, use the rule that has a maximum degree.
- ***Step 5***: Determine mapping bases on the Combined Fuzzy Rule Base. We use the following defuzzification strategy to determine the output control y for given input (x_1,x_2). First, we combine the antecedents of the *i*th fuzzy rule using product operations to determine the degree, $\mu_{O^i}^i$ of the output control corresponding to (x_1,x_2); that is,

$$\mu_{O^i}^i = \mu_{I_1^i}(x_1) \times \mu_{I_2^i}(x_2) \qquad (5)$$

Where O^i denotes the output region of rule *i*, and I_j^i denotes the input region of rule *i* for the *jth* component; for example, rule 1 gives

$$\mu_{CE}^1 = \mu_{S1}(x_1) \times \mu_{B2}(x2)$$

Then we use the center average defuzzification formula to determine the output

$$y = \frac{\sum_{i=1}^{M} \mu_{O^i}^i \times y^{-i}}{\sum_{i=1}^{M} \mu_{O^i}^i} \qquad (6)$$

Where y^{-i} denotes the center value of region O^i and M is number of rule in the combined fuzzy rule base.

IV.Our proposal

Our proposal is illustrated in Fig. 2. We create a combined rule base system that is based on steps 1 to 4 in session 3 and data pairs $(x(1),x(2),\dots x(p);x(p+1))$, $(x(2),x(3),\dots x(p+1);x(p+2))$, .., $(x(n-p-2),x(n-p),\dots x(n-1);x(n))$,. After that, we use the combined rule base system and step 5 above to predict and append extended data to the tail of the original data. We reserve original data and repeat the procedure similarly to extending tail data for the head. After appending data to the head and tail of the original data we carry out the EMD process.

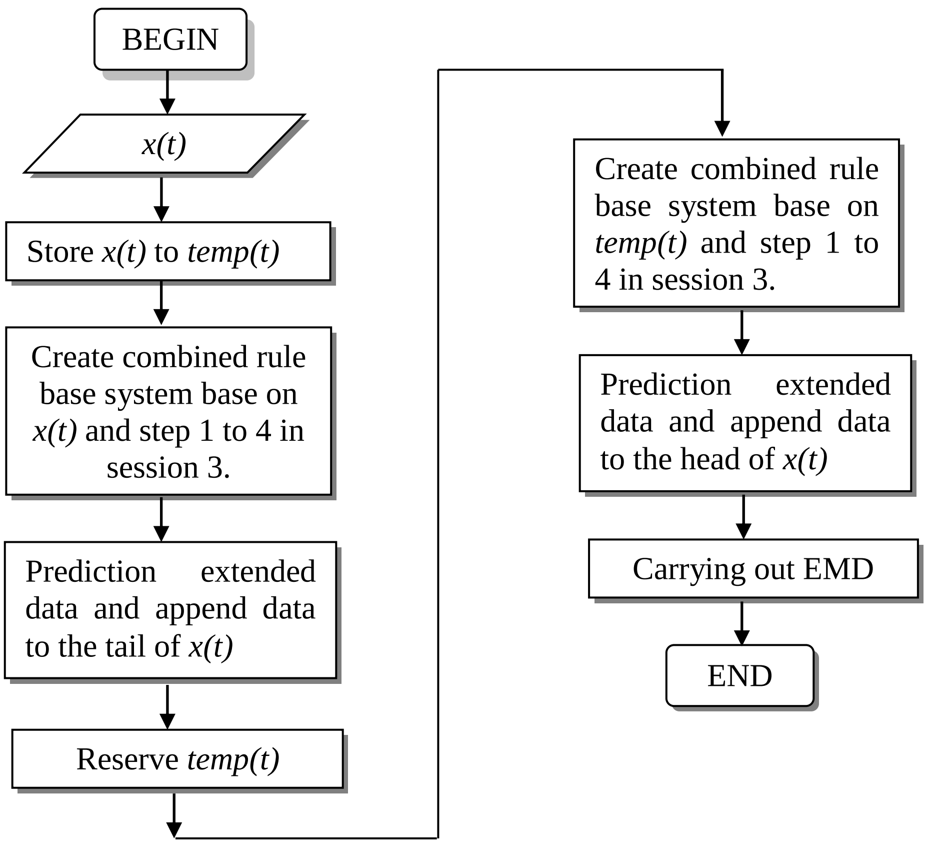

Figure 2. Extension data bases on fuzzy logic

V.Simulations

Consider time series x(k) is generated by the signal $x(t) = \sin(2*\pi*t) + \sin(4*\pi*t) + \sin(6*\pi*t)$ with sampling frequency 100 Hz. Fig.3 shows 800 points of this time series.

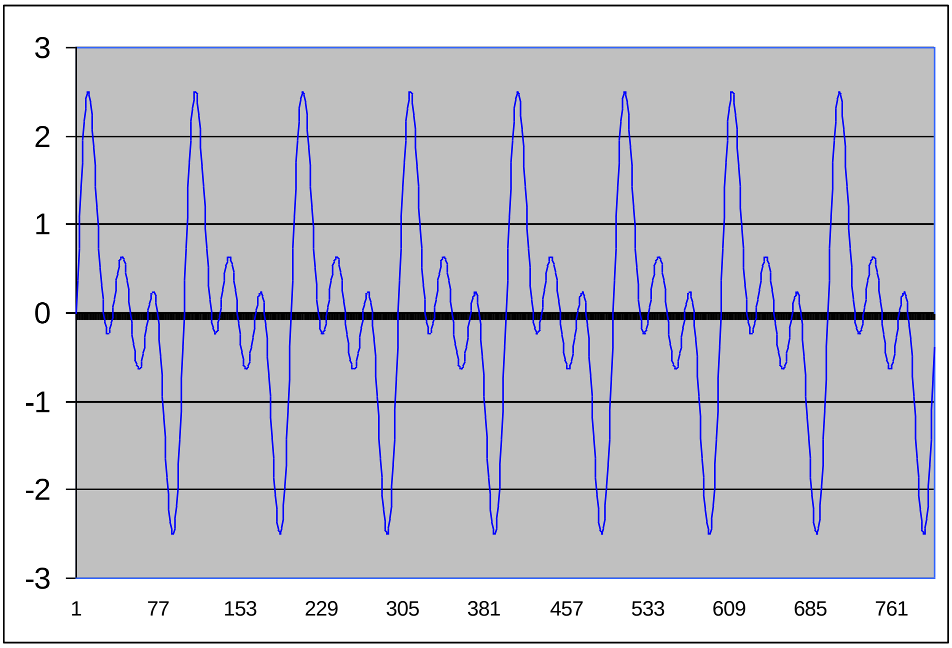

Figure.3. A session of time series

In our simulation we use nine points in the time series to predict next point in time series. The membership function for any input is shown in Fig.4. We divide domain interval of each input into 56 equals regions. We use first 700 points of time series as training set and final 100 points as test set. Fig.5 shows the result.

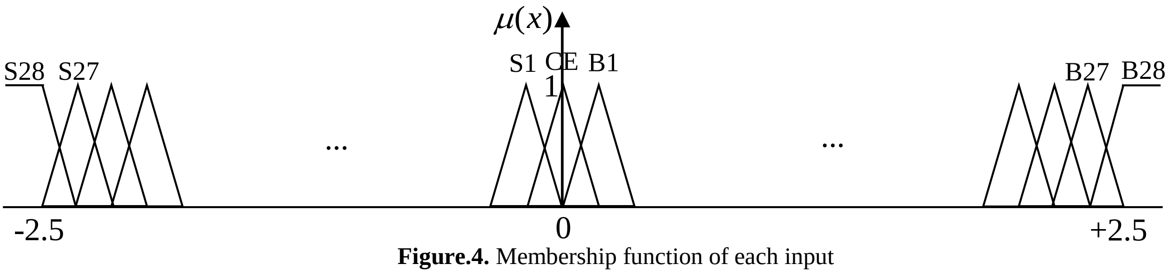

Figure.4. Membership function of each input

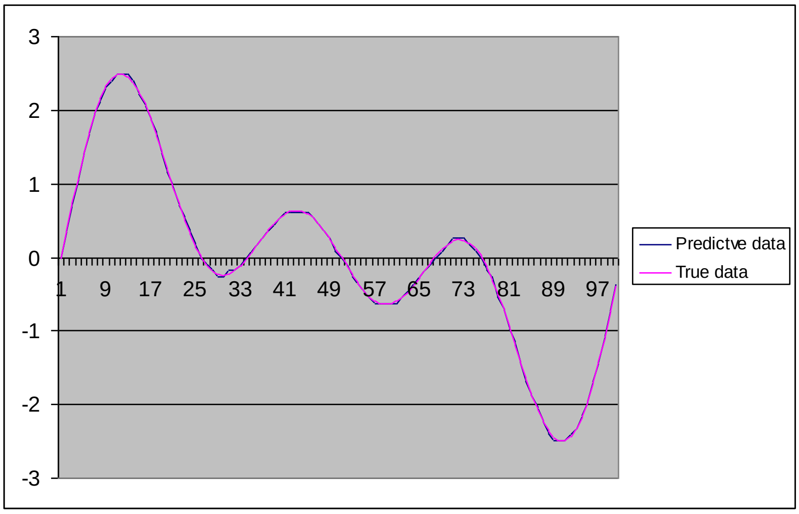

Figure.5. Prediction of summing up sine series

The mean square error of 100 test samples is 0.00052. In Fig.5 is shown that the predictive data and true data nearly overlap.

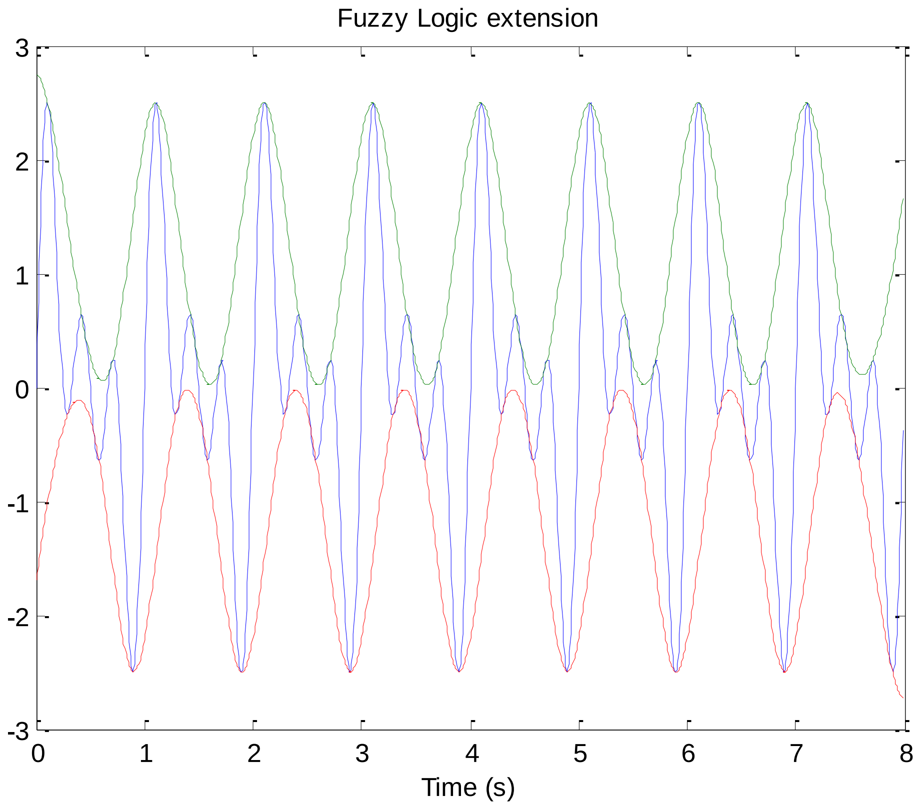

Figure.6. Upper and lower envelope use Fuzzy logic extension

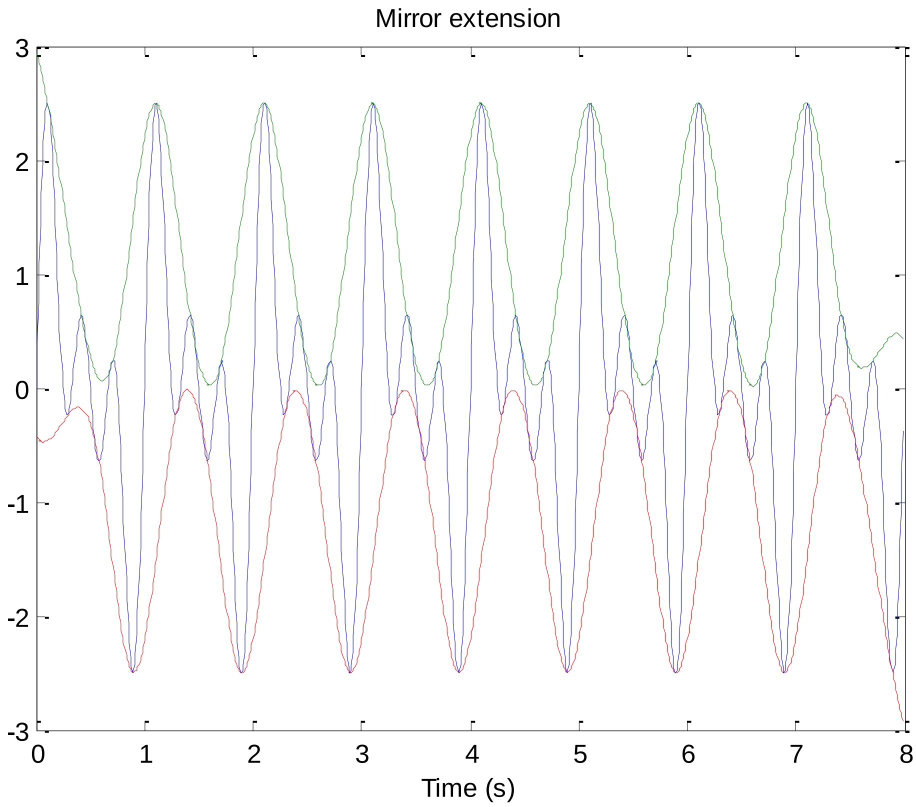

Figure.7. Upper and lower envelope use Mirror extension

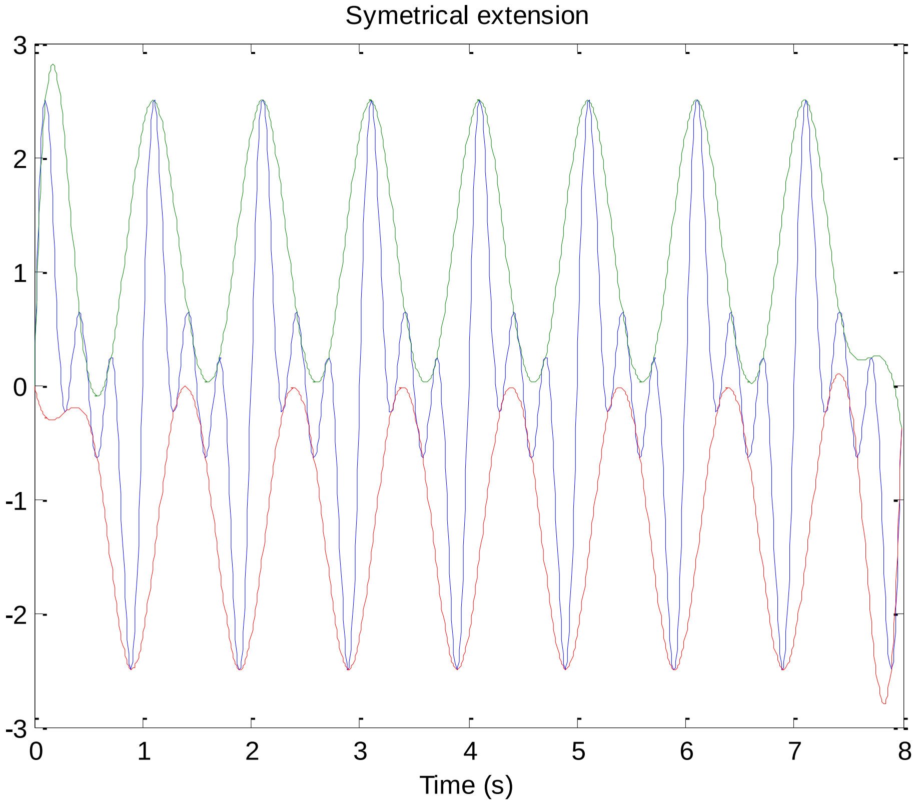

Figure.8. Upper and lower envelope use Symmetrical extension

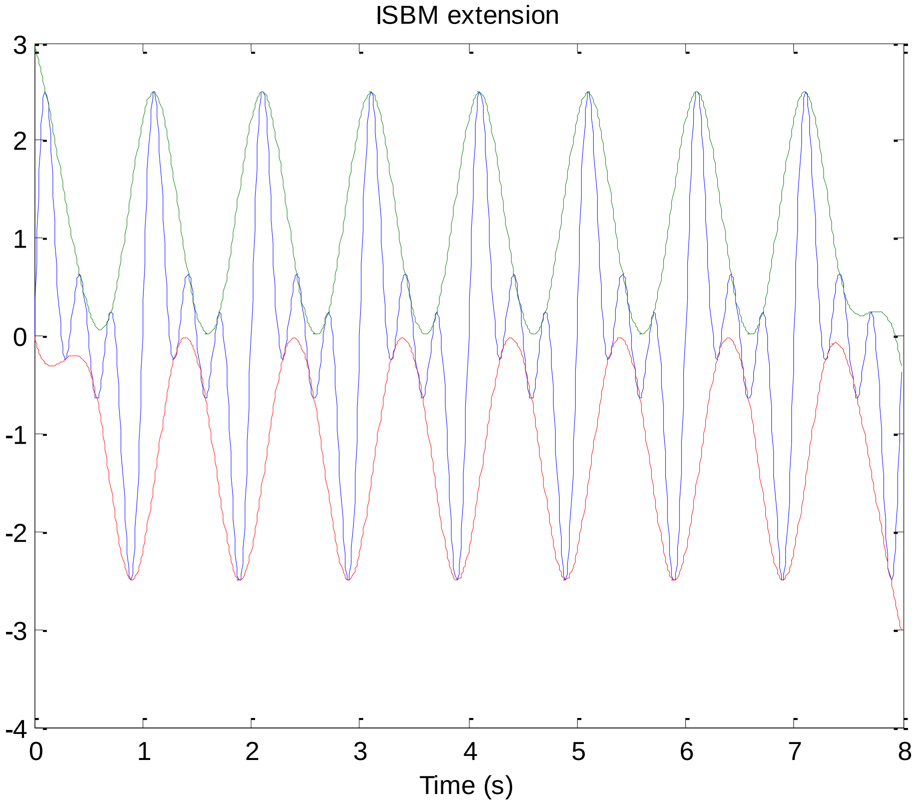

Figure.9. Upper and lower envelope use ISBM extension

Fig. 6, 7, 8, 9 shows lower and upper envelope that creates by some different extension methods. By comparison it is clear that our proposal method performs best.

VI.Conclusions

In this paper, we proposed to apply Fuzzy logic to solve the end effect of EMD. Because apply Fuzzy logic to extend data forward and backward, so accurate results are achieved. The simulation results show that our proposed method gets a better result than other methods.

REFERENCES

[1]. N. E. Huang, S. P. Shen, "Hilbert-Huang Transform and Its Application" World Scientific Pub Co Inc, 2005.
[2]. Gabriel Rilling, Patrick Flandrin and Paulo Gon¸calv`es, "on empirical mode decomposition and its algorithms". IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing, NSIP-03, 2003, pp. 1–5
[3]. Marcus Da¨ tig, Torsten Schlurmann, "Performance and limitations of the Hilbert–Huang transformation (HHT) with an application to irregular water waves", Ocean Engineering 31 (2004) 1783–1834.
[4]. Fangji Wu, Liangsheng Qu, "An improved method for restraining the end effect in empirical mode decomposition and its applications to the fault diagnosis of large rotating machinery", Journal of Sound and Vibration 314 (2008) 586–602.
[5]. Jian Xun, Shaoze Yan: "A revised Hilbert–Huang transformation based on the neural networks and its application in vibration signal analysis of a deployable structure" Mechanical Systems and Signal Processing 22 (2008) 1705–1723.
[6]. Jinghui Ma, Hong Jiang, Li Yao, Song Pu "End Effects Processing of Hilbert-Huang Transform Based on Genetic Algorithm and RBF Neural Network", Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference.
[7]. Chun-Yao Lee and Yao-chen Lee "BPNN Based Processing for End Effects of HHT", World Academy of Science, Engineering and Technology 72 2010.
[8]. L. Wang, "Adaptive fuzzy systems and control: Design and stability analysis", Prentice Hall International, 1994.