2016

Extrema Based Signal Transforms For Biomedical Signal Analysis

Bhadhan Roy Joy
University of Alabama at Birmingham

EXTREMA BASED SIGNAL TRANSFORMS FOR BIOMEDICAL SIGNAL ANALYSIS

by

BHADHAN ROY JOY

ARIE NAKHMANI, COMMITTEE CHAIR
MOHAMMAD R. HAIDER
KARTHIKEYAN LINGASUBRAMANIAN

A THESIS

Submitted to the faculty of the University of Alabama at Birmingham,
in partial fulfillment of the requirements of the degree of
Master of Science

BIRMINGHAM, ALABAMA

2016

EXTREMA BASED SIGNAL TRANSFORMS FOR BIOMEDICAL SIGNAL ANALYSIS

BHADHAN ROY JOY

ELECTRICAL AND COMPUTER ENGINEERING

ABSTRACT

Signal transforms are very important tools to extract useful information from scientific, engineering, or medical raw data. Unfortunately, traditional transform techniques impose unrealistic assumptions on the signal, often producing erroneous interpretation of results. Well-known integral transforms, such as short time Fourier transform, though have fast implementation algorithms (e.g., FFT), are still computationally expensive. They have multiple parameters that should be tuned, and it is not readily clear how to tune them for long-duration nonstationary signals. To solve these problems, one needs a computationally inexpensive transform with no parameters that will highlight important data aspects. We propose a simple transform based on extrema points of the signal. The transform value at a given point is calculated based on the distance and magnitude difference of two extrema points it lies between, rather than considering every point around it. We discuss implementation of the developed algorithm and show examples of successfully applying the transform in detecting Delta waves in brain EEG signal. Ideas for improvement and further research are discussed.

Keywords: Parameterless Transform, Extrema Transform, EEG, Delta wave, Hilbert-Huang Transform, HHT, Empirical Mode Decomposition, EMD

DEDICATION

TO MY FAMILY AND FRIENDS

ACKNOWLEDGEMENTS

I would like to express my deepest thanks to my academic supervisor and committee chair, Dr. Arie Nakhmani, who have inspired me through every step of my research and academic work. In the very beginning, he helped me to select a research topic aligned with my interest and then guided me throughout the whole time during my thesis. Without his guidance this dissertation would not have been possible.

Thanks to Dr. Mohammad Haider and Dr. Karthik Lingasubramanian for serving on my committee. I am grateful to Dr. Haider for teaching me the art of writing good engineering papers. I am deeply indebted to Dr. Frank Skidmore for his valuable knowledge of Medical Imaging and EEG signal processing. I would also like to thank S. M. Mehedee Parvez, AKM Arifuzzman, Farig Sadeque, Isa Hatipoglu and Dr. Daisy Wong, who helped me with their insight and feedback during different stages of my research.

Many thanks to the faculties and staff in the ECE department. I very appreciate the financial support all through my graduate study in UAB. I am thankful to Ms. Janice Hitchcock Hofman and Ms. Sandra D Muhammad, who have offered me tremendous help with my paperwork.

Last but not least, I would like to express my deepest gratitude and appreciation to my family - my father Nikhil Chandra Roy, my mother Putul Rani Roy, my brother Monoshi Kumar Roy. Without their inspiration and support, it would have been impossible for me to pursue graduate studies.

Contents

List of Tables

List of Figures

1. INTRODUCTION

Human body mechanism is controlled by the nervous system. As the central part of the system, the brain receives outside world signal, processes it and triggers a reaction. Brain signal changes during cognitive activities and in different sleep stages. Neurological disorders such as epilepsy, or sleep disorder cause abnormalities in the brain signal [**1**]. Therefore, brain signal analysis is necessary for disease detection, cognitive brain and sleep study.

Multitude of other applications require analysis of nonstationary signals. Study of seismic wave reveals the energy density associated with an earthquake [**2**]. Speech signal analysis can lead to noninvasive and cheap detection method of Parkinson's disease [**3**]. Application of signal transform methods on data like global temperature or CO_2 concentration can reveal inherent periodicity and increasing or decreasing trend buried in the data [**4**]. Due to these and other applications in the broad range of fields, many different transform methods were proposed in the past.

1.1. Motivation Behind Development of a New Transform

Practical signals are nonstationary. EEG, ECG, seismic wave, financial data- frequency spectrum of these signals vary over time. But in many existing transform methods, the signal is considered to be periodic, stationary, or piecewise periodic. In conventional transforms, signal analysis is done by projecting it onto *a priori* selected basis vectors, which remain unchanged during analysis. But adaptability is a necessary criterion for the basis for expanding non-stationary signal [**5**]. As the spectrum of nonstationary data is not static while basis vectors remained unchanged during analysis, the methods may not capture the varying nature of signal efficiently under all conditions. In addition to that, real world signals are noisy, where the noise often can be orders of magnitude stronger than the signal itself. Often, signal and noise are mixed and difficult to separate. In these situations,

removing a lot of noise would also lead to the removal of some signal [**6**]. Designing suitable filter to remove noise introduces extra complexity for the signal analysis process. Moreover, the performance of most existing transform depends on the judicious selection of parameters. Often signals for analysis are of long duration. For example, sleep study requires analysis of EEG signal couple of hours long. Computationally expensive traditional methods are not ideal for fast analysis of long duration signals. In this dissertation, we address the shortcoming of traditional transforms and develop a parameterless and computationally inexpensive signal transform.

1.2. Literature Review

Traditionally, signal analysis field has been dominated by Fourier Transform [**7**], [**8**] to find energy distribution of the signal in the frequency domain. In Fourier transform, the signal is decomposed into a combination of sinusoids. Generally, when performing Fourier transform, one assumes periodic and stationary signal to make sense from the result. As a sinusoid is infinite in the time domain with fixed amplitude, more sinusoids in the time domain are necessary to represent a sudden jump in the signal. Next, sharp change and discontinuity result in spreading of Fourier spectra in the frequency domain. To overcome the shortcomings of Fourier transform, Short-Time Fourier Transform (STFT) has been developed [**9**], [**10**]. In STFT, time-frequency distribution is computed by applying and successively sliding a short-time duration window along the time axis. Data is assumed to be piecewise stationary during this short period. Here, the frequency resolution is controlled by the selection of window size at the beginning of analysis whereas the nature of the signal is generally not known a priori; that makes the choice of a suitable window size challenging. The window width must be small to localize an event in time. On the other hand, long time span is necessary to determine frequency resolution, thus requiring a compromise between two conflicting requirements. [**11**]. Moreover, type of the window and consecutive windows overlapping intervals should be chosen manually to get quality results.

In the late 1970's, J. Morlet developed an alternative to STFT, which consisted of first windowing a signal, and then computing its Fourier coefficients. Prudent choice of the window resulted in better extraction of signal information in both time and frequency domain. Works based on the idea led to the development of Wavelet transform [**12**]. In her seminal paper on wavelets [**13**], Daubechies constructed orthonormal bases of compactly supported wavelets to give the field a solid mathematical foundation. As a vast improvement of STFT, wavelet analysis has numerous applications in disparate fields even though it has some limitations. It uses a predefined set of basis vectors of finite length, and the selected set may not be suitable for analyzing the signal in consideration. In wavelet analysis, good frequency and poor time resolution are observed at low frequency and the contrary at high-frequency range [**14**].

More recently, a new signal analysis method, namely Hilbert-Huang Transform (HHT), has been proposed [**5**], [**15**], [**16**]. In HHT, frequency change in the signal is considered at a very local level, which makes it particularly suitable for non-stationary data analysis. HHT algorithm can be divided into two parts: (a) Empirical mode decomposition (EMD), (b) Hilbert transform to calculate time-frequency-energy from EMD signals [**17**]. In EMD, the signal is decomposed into a finite and generally small number of base functions called Intrinsic Mode Functions (IMF). Those functions have the following properties:

(1) The number of extrema points and the number of zero-crossings of IMF differ by at most one [**5**, **18**].
(2) At any point, the mean value of the envelope defined by local maxima and the envelope defined by local minima is zero [**5**, **18**].

The EMD algorithm employs an iterative sifting process which is as follows:

(1) Determine the local maxima and minima of the signal $x(t)$.
(2) Connect the maxima with an interpolation function to create an upper envelope $e_{max}(t)$ and repeat the procedure for the minima to create lower envelop $e_{min}(t)$.

(3) Compute the mean $m(t)$ of extrema envelopes.

$$m(t) = \frac{e_{max}(t) + e_{min}(t)}{2} \tag{1.1}$$

(4) Calculate residual $c(t) = x(t) - m(t)$.

(5) Iterate over the residual $c(t)$ until it meets the criteria of IMF.

Once one IMF is calculated, it is subtracted from the original signal and sifting process is repeated on reminder signal to extract the next IMF. This is repeated to extract n number of IMFs until the residual r_n is less than a predefined tolerance value, or when r_n becomes a monotonic function [**5**], [**19**].

After the signal $x(t)$ is fully decomposed, it can be expressed as the sum of the IMFs and final residue.

$$x(t) = \sum_{j=1}^{n} c_j + r_n \tag{1.2}$$

Here, n is the number of IMFs, c_j is a decomposed IMF, which has well-behaved Hilbert transform and r_n is the residue after the sifting process ends. For IMF c_j Hilbert transform is:

$$H[c_j(t)] = \tilde{y}_j(t) = \frac{1}{\pi} PV \int_{-\infty}^{\infty} \frac{c_j(t')}{t - t'} dt' \tag{1.3}$$

Here, PV is the Cauchy principal value [**4**]. For each IMF, we can calculate instantaneous frequency $\omega_j, j = 1, 2, ..., n$ and amplitude $a_j, j = 1, 2, ..., n$ using (1.4), (1.5).

$$c_j(t) + i\tilde{y}_j(t) = a_j(t) e^{i\theta(t)} \tag{1.4}$$

where $a_j(t) = \sqrt{c_j^2(t) + \tilde{y}_j^2}$, $\theta(t) = \arctan(\frac{\tilde{y}_j}{c_j})$ and $i = \sqrt{-1}$

$a_j(t)$ and $\theta(t)$ are the instantaneous amplitude and phase functions of IMF c_j [**4**]. The instantaneous frequency is the time derivative of the phase.

$$\omega_j = \frac{d\theta_j(t)}{dt} \tag{1.5}$$

From instantaneous frequency and amplitude original signal can be reconstructed using (1.6).

$$x(t) = Real \sum_{j=1}^{N} a_j(t) e^{i \int \omega_j(t) dt} \tag{1.6}$$

HHT is useful in the analysis of nonlinear non-stationary data. So it has been successfully applied in diverse fields. The method has been employed in distorted power quality analysis [**20**–**22**], speech analysis [**23**–**27**], financial [**18**, **28**, **29**] and geophysical data study [**19**, **30**, **31**], mechanical fault detection [**32**–**34**], signal filtering [**35**–**39**], to name a few. Because of human body signals being non-stationary in nature, HHT is particularly suitable for biosignal analysis. It has found application in heart signal analysis [**17**, **40**–**43**], blood flow and pressure measurement [**44**–**46**], EEG study [**47**–**53**]. Though versatile, it also suffers from some limitations. HHT is not effective on narrow-band signal [**54**] and is computationally expensive. Keeping in mind the shortcomings of existing signal analysis tool, we develop a new, parameterless, computationally inexpensive signal transform. The transform does not require *a priori* basis, takes frequency range to search for as input and produces large value at times when the provided frequency range is dominant in signal in consideration.

1.3. Organization of the Thesis

This thesis develops a new extrema based computationally inexpensive transform for signal analysis. The primary objective of this research is to develop the algorithm and explores the mathematical properties and application of the transform.

In Chapter 2, description and algorithm of the transformation are presented. In Chapter 3, general transform properties are discussed and proved. Some typical examples are provided to demonstrate those properties. We compare the performance of existing transforms to our transform in Chapter 4. In the first example, we consider a noise corrupted linear chirp signal to search the time and duration of a particular frequency range using developed transform and compare the performance with HHT. In the next example, we analyze the effect of

noise and demonstrate that the transform is robust to noise. We conclude the chapter with another example showing the application of signal in detecting delta wave in EEG signal. Limitations, future research ideas for improvement are discussed in Chapter 5.

2. TRANSFORM DEFINITION AND ALGORITHM

The proposed transform involves well-defined iterative steps based on extrema points applied to the signal $x(t)$. Let's assume that the signal is given by $x(t)$ for $t \geq 0$. All its discrete local extrema points are denoted by E_i, including the starting point $E_0 = x(0)$. Those extrema points appear at times $\{t_0 = 0, t_1, ..., t_i, ...\}$. The first level transform is defined by two discrete vectors $\{t_i\}$ and $\{|E_{i+1} - E_i|\}$, which is the vertical distance between the consecutive extrema points. The next transform level is computed by constructing piecewise linear signal $x_1(t)$ connecting the points $\{\frac{t_i + t_{i+1}}{2}, \frac{E_i + E_{i+1}}{2}\}$. Practically, this means a piecewise linear curve passing between the extrema points of the original signal, providing smoothing to the signal $x(t)$. For level 2, another pair of discrete vectors of time and vertical distances is computed from $x_1(t)$ using the same approach from the level 1. The process is continued for additional levels. Generally, less than 10 levels are needed for the analysis.

One important use of the proposed transform is the detection of signal in noise; more specifically, detection of nonstationary periodic signals in a predefined frequency range. For that, in each step, the signal component in the provided frequency range is searched and if found the regions are marked. The expanded transform is the summation of values in the marked region. The algorithm is described below:

(1) Given the range of desired frequencies $[f_{min}, f_{max}]$, calculate the range for a number of sample points, I_{min} and I_{max}, between two successive extrema:

$$\begin{aligned} I_{min} &= \frac{F_s}{2 f_{max}}, \\ I_{max} &= \frac{F_s}{2 f_{min}}. \end{aligned} \tag{2.1}$$

Here, F_s is the sampling frequency of the signal.

(2) Create a zero vector v_1 of the same size of original signal $x(t)$.

(3) Calculate all extrema points $\{E_0 = x(0), E_1, ..., E_i, E_{i+1}, ...\}$ at time $\{t_0 = 0, t_1, ..., t_i, t_{i+1}, ...\}$. If all the samples have the same value, then consider the first and last sample as two consecutive extrema points.

(4) If the distance between two successive extrema points E_i and E_{i+1} at time t_i and t_{i+1}, respectively is in between I_{min} and I_{max}, then put value $|E_{i+1} - E_i|$ for every data sample in between t_i and t_{i+1}. For sample k in between t_i and t_{i+1},

$$v_1(k) = \begin{cases} |E_{i+1} - E_i|, & \text{if } I_{min} \leq t_{i+1} - t_i \leq I_{max} \\ 0, & \text{otherwise} \end{cases} \tag{2.2}$$

(5) Update the signal $x(t)$ for next level with the average values of the successive extrema points placed at the mid point between them. All other non-extrema points are discarded.

$$\begin{aligned} t_i' &= \frac{t_i + t_{i+1}}{2}, \\ x_1(t_i') &= \frac{E_i + E_{i+1}}{2}. \end{aligned} \tag{2.3}$$

(6) Continue step 2 - 5 for n number of times for level $2, 3, ..., n$ until the total number of samples in the signal becomes 3 or less.

(7) After n number of iterations, we have vector $v_1, v_2, v_3, ..., v_n$ created in each iteration. The expanded transform $T[x(t)]$ is the summation of all the vectors.

$$T[x(t)] = \sum_{j=1}^{n} v_j \tag{2.4}$$

3. MATHEMATICAL PROPERTIES

Let the transform of signal $x(t)$ be defined as $X(t)$.

$$T[x(t)] = X(t) \tag{3.1}$$

3.1. Transform of a Constant

The Extrema transform of a constant signal $x(t) = c$ is 0.

$$T(c) = 0 \tag{3.2}$$

Proof: During the step 3 of first iteration, the extrema are $E_0 = E_1 = c$ located at t_0 and t_1 samples, where t_0 and t_1 are the first and last sample, respectively. As $E_1 - E_0 = 0$, so for both cases of (2.2),

$$v_1 = 0 \tag{3.3}$$

The updated point, form (2.3) is $x_1 = \frac{E_1+E_0}{2} = c$ at time $\frac{t_0+t_1}{2}$. As only one sample remains after update, so according to step 6, it is the final iteration and the transform is $T[x(t)] = v_1 = 0$.

3.2. Transform of a Sinusoid

Transform of sinusoid $x(t) = A\sin(2\pi f + \theta)$ is

$$T(x) = \begin{cases} 2A, & \text{if } f_{min} \leq f \leq f_{max} \\ 0, & \text{otherwise} \end{cases} \tag{3.4}$$

Proof: Let the extrema points of the sinusoid are $\{E_0 = x(0), E_1, ..., E_i, E_{i+1}, ...\}$ at time $\{t_0 = 0, t_1, ..., t_i, t_{i+1}, ...\}$. The sampling frequency is F_s. Here, $|E_{i+1} - E_i| = 2A$.

Assuming the sampling frequency high enough, the distance between two consecutive extrema points of same phase is $\frac{1}{f}$ sec $= \frac{F_s}{f}$ samples.

As there is an extrema of opposite phase at the midpoint of two same phase extrema, the number of samples between two consecutive extrema is $\frac{F_s}{2f}$ samples.

During first iteration of the algorithm, according to (2.2),

$$v_1(k) = \begin{cases} 2A, & \text{if } I_{min} \leq t \leq I_{max} \\ 0, & \text{otherwise} \end{cases} \tag{3.5}$$

Here, $I_{min} = \frac{F_s}{2f_{max}}$, $t = \frac{F_s}{2f}$ and $I_{max} = \frac{F_s}{2f_{min}}$. The inequality, after simplification becomes $f_{min} \leq f \leq f_{max}$. We get,

$$v_1 = \begin{cases} 2A, & \text{if } f_{min} \leq f \leq f_{max} \\ 0, & \text{otherwise} \end{cases} \tag{3.6}$$

In the next step, update of each consecutive extrema pair E_i and E_{i+1} at time t_i and t_{i+1} samples is calculated by (2.3),

$$\begin{aligned} t_i' &= \frac{t_i + t_{i+1}}{2}, \\ x_1(t_i') &= \frac{E_i + E_{i+1}}{2} = 0 \end{aligned} \tag{3.7}$$

Updated signal is a constant 0 valued signal. In second iteration, produced vector v_2 is zero vector according to (3.3) and iteration stops after second iteration.

$$v_2 = 0 \tag{3.8}$$

Transform is summation of v_1 and v_2.

$$T(sine) = v_1 + v_2 = \begin{cases} 2A, & \text{if } f_{min} \leq f \leq f_{max} \\ 0, & \text{otherwise} \end{cases} \tag{3.9}$$

In Fig. 1 (a) and 2 (a), frequency of the sinusoid is $f = 10$Hz and amplitude $A = 1$. Transform is applied on the same signal, in first case with $f_{min} = 9$Hz and $f_{max} = 11$Hz. $f_{min} < f < f_{max}$, so from (3.9) the transform is $T(sine) = 2$, which can be seen in Fig. 1 (c). In the next example, $f_{min} = 14$Hz and $f_{max} = 16$Hz while the signal remains unchanged. As $f > f_{max}$, the output of the transform in this case is 0, which is displayed in Fig. 2 (c).

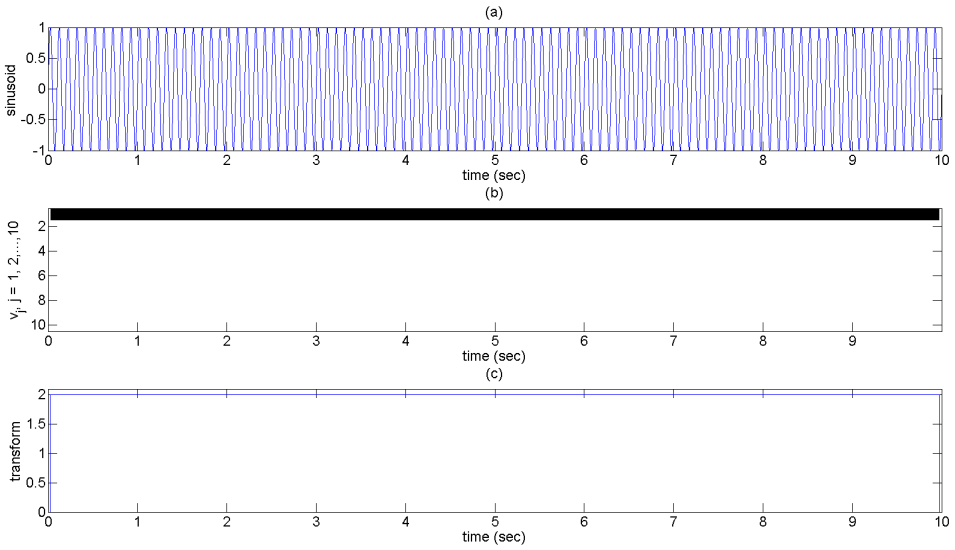

Figure 1: (a) Sinusoid of frequency 10Hz. (b) Image representation of transform: $f_{min} = 9$ and $f_{max} = 11$. (c) Extrema transform applied to sinusoid with the parameters.

3.3. Amplitude Scaling

Amplitude scaling of signal

$$T[a(x(t))] = aX(t) \tag{3.10}$$

Proof: Let, $\{E_0 = x(0), E_1, ..., E_i, E_{i+1}, ...\}$ at time $\{t_0 = 0, t_1, ..., t_i, t_{i+1}, ...\}$ are the extrema of $x(t)$ and for scaled signal $a(x(t))$ the extrema are $\{a.E_0 = a.x(0), a.E_1, ..., a.E_i, a.E_{i+1}, ...\}$ at time $\{t_0 = 0, t_1, ..., t_i, t_{i+1}, ...\}$. At the beginning of first iteration, maxima of the original and scaled signal are at the same location and second signal's maxima are scaled by factor a.

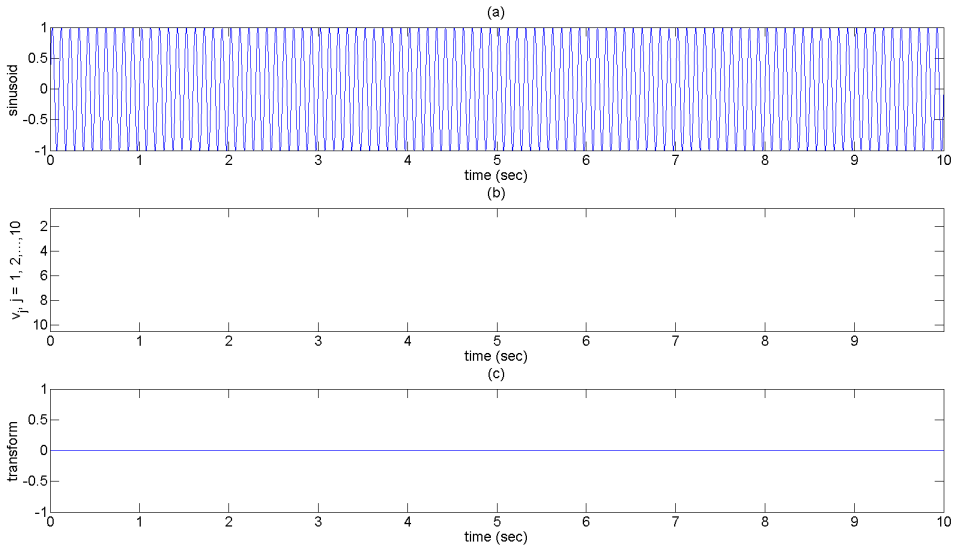

Figure 2: (a) Sinusoid of frequency 10Hz. (b) Image representation of transform: $f_{min} = 14$ and $f_{max} = 16$. (c) Extrema transform applied to the sinusoid with the parameters.

Let us suppose at the beginning of $j-th$ iteration, the maxima of both signals are at the same location. For signal $x_{j-1}(t)$ at $j-th$ iteration, any $k-th$ sample value of vector v_j between two consecutive extrema at time t_i and t_{i+1} is,

$$v_j(k) = \begin{cases} |E_{i+1} - E_i|, & \text{if } I_{min} \leq t_{i+1} - t_i \leq I_{max} \\ 0, & \text{otherwise} \end{cases} \tag{3.11}$$

Here, $t_i \leq k < t_{i+1}$, E_i and E_{i+1} are two consecutive extrema points at time t_i and t_{i+1} of $x_{j-1}(t)$.

Updated signal point for the extrema pairs is

$$\begin{aligned} t_i' &= \frac{t_i + t_{i+1}}{2}, \\ x_j(t_i') &= \frac{E_i + E_{i+1}}{2} \end{aligned} \tag{3.12}$$

On the other hand, for scaled signal $ax_{j-1}(t)$, at $j-th$ iteration, sample values of vector v_j^* between t_i and t_{i+1} is,

$$v_j^*(k) = \begin{cases} |a.E_{i+1} - a.E_i| = |a(E_{i+1} - E_i)|, & \text{if } I_{min} \leq t_{i+1} - t_i \leq I_{max} \\ 0 = a.0, & \text{otherwise} \end{cases} \tag{3.13}$$

The update point for the pair would be

$$\begin{aligned} t_i^* &= \frac{t_i + t_{i+1}}{2}, \\ x_j^*(t_i^*) &= \frac{a.E_i + a.Ei + 1}{2} \\ &= \frac{a(E_i + E_{i+1})}{2} \end{aligned} \tag{3.14}$$

Comparing (3.11) with (3.13) and (3.12) with (3.14) we get,

$$\begin{aligned} v_j^* &= a.v_j, \\ t_i^* &= t_i', \\ x_j^*(t_i^*) &= a.x_j(t_i') \end{aligned} \tag{3.15}$$

It can be noticed from (3.15) that at the beginning of $j+1-th$ iteration, the update of scaled signal would have samples at the same location as the updated original signal, and scaled by factor a. Following same steps in $j+1$-th iteration we would get

$$v_{j+1}^* = a.v_{j+1} \tag{3.16}$$

The stopping criteria does not include sample values, it depends on the number of points left after each iteration, so for both cases it would stop after the same, say n, number of iteration. Using (2.4) for scaled signal,

$$
\begin{aligned}
T[a(x(t))] &= \sum_{j=1}^{n} v_j^*, \\
&= a \sum_{j=1}^{n} v_j \\
&= a.X(t)
\end{aligned}
\tag{3.17}
$$

3.4. DC Offset

$$
T[(x(t)) + c] = X(t) \tag{3.18}
$$

Proof: Let, $\{E_0 = x(0), E_1, ..., E_i, E_{i+1}, ...\}$ at time $\{t_0 = 0, t_1, ..., t_i, t_{i+1}, ...\}$ are the extrema of $x(t)$ and for signal $x(t) + c$ the extrema are $\{E_0 + c, E_1 + c, ..., E_i + c, E_{i+1} + c, ...\}$ at time $\{t_0 = 0, t_1, ..., t_i, t_{i+1}, ...\}$. At the beginning of first iteration, the maxima of original and offset signal are at the same location and second signal's maxima values are shifted by c.

Let us suppose during the beginning of $j-th$ iteration, the maxima of both signals are at the same location. For $x_{j-1}(t)$ at $j-th$ iteration, the sample values of vector v_j between consecutive extrema at time t_i and t_{i+1} are expressed by (3.11) and updated signal point is calculated by (3.12).

On the other hand, for DC offset signal $x_{j-1}(t) + c$, at the beginning of j-th iteration, the sample values of v_j^* between two consecutive extrema at t_i and t_{i+1} are,

$$
v_j^*(k) = \begin{cases} |E_{i+1} + c - (E_i + c)| = |E_{i+1} - E_i|, & \text{if } I_{min} \leq t_{i+1} - t_i \leq I_{max} \\ 0, & \text{otherwise} \end{cases}
\tag{3.19}
$$

The updated point for the pair would be

$$\begin{aligned} t_i^* &= \frac{t_i + t_{i+1}}{2}, \\ x_j^*(t_i^*) &= \frac{E_i + c + E_{i+1} + c}{2} \\ &= \frac{E_i + E_{i+1}}{2} + c \end{aligned} \tag{3.20}$$

Comparing (3.11) with (3.19) and (3.12) with (3.20) we get,

$$\begin{aligned} v_j^* &= v_j, \\ t_i^* &= t_i', \\ x_j^*(t_i^*) &= x_j(t_i') + a \end{aligned} \tag{3.21}$$

It can be noticed from (3.21) that at the beginning of $j+1-th$ iteration, the update of offset signal would have samples at the same location as the updated original signal, and offset by value c. Following same steps in $j+1$-th iteration we would get

$$v_{j+1}^* = v_{j+1} \tag{3.22}$$

The stopping criteria depends on the number of points left after each iteration, for both cases it would stop after same number of iteration. As noticed in (3.22), in both original and offset signal, the vectors output after each iteration are the same. That implies in both cases the transform would yield the same result, as the transform is summation of th output vectors. We get,

$$T[x(t) + c] = X(t) \tag{3.23}$$

3.5. Time Shift

$$T[x(t - \tau)] = X(t - \tau) \tag{3.24}$$

Proof: Let, $\{E_0 = x(0), E_1, ..., E_i, E_{i+1}, ...\}$ at time $\{t_0 = 0, t_1, ..., t_i, t_{i+1}, ...\}$ are the extrema of $x(t)$ and $\{E_0 = x(0), E_1, ..., E_i, E_{i+1}, ...\}$ at time $\{t_0 + \tau, t_1 + \tau, ..., t_i + \tau, t_{i+1} + \tau, ...\}$ are

the extrema of $x(t-\tau)$. The two signals have the same extrema sequence, but the extrema of $x(t-\tau)$ are shifted by τ to the right.

Let us suppose at the beginning of j-th iteration, the extrema of the second signal are shifted by τ to the right, that is two consecutive extrema have values E_i and E_{i+1} at $t_i+\tau$ and $t_{i+1}+\tau$, respectively. For any arbitrary sample k^* in between $t_i+\tau$ and $t_{i+1}+\tau$,

$$v_j^*(k^*) = \begin{cases} |E_{i+1}-E_i|, & \text{if } I_{min} \le t \le I_{max} \\ 0, & \text{otherwise} \end{cases} \tag{3.25}$$

Here, $t = t_{i+1}+\tau-(t_i+\tau) = t_{i+1}-t_i$ and $t_i+\tau \le k^* < t_{i+1}+\tau$.

From comparing (3.11), (3.25) and the range of k and k^* of the equations, it can be deduced that v_j^* has same value as v_j when time shifted to the right by τ.

$$v_j^*(t^*) = v_j(t-\tau) \tag{3.26}$$

The update of i-th extrema pair, according to (2.3) is,

$$\begin{aligned} t_i^* &= \frac{t_i+\tau+t_{i+1}+\tau}{2} \\ &= \frac{t_i+t_{i+1}}{2}+\tau \\ x_j^*(t_i^*) &= \frac{E_i+E_{i+1}}{2} \end{aligned} \tag{3.27}$$

Comparing (3.12) with (3.27) we see the updated time shifted signal would also be equal to updated original signal, after time shifting by τ. So, at the beginning of $j+1$ iteration, the second signal would maintain its τ time shift and values and the number of points of both first and second updated signal would be the same. Following same steps in $j+1$ iteration we would get

$$v_{j+1}^*(t^*) = v_{j+1}(t-\tau) \tag{3.28}$$

It would take the same n number of iterations for both signals before stopping. For $x(t-\tau)$ the transform is

$$\begin{aligned} T[x(t-t_0)] &= \sum_{j=1}^{n} v_j^*(t^*) \\ &= \sum_{j=1}^{n} v_j^*(t-\tau) \\ &= X(t-\tau) \end{aligned} \tag{3.29}$$

4. RESULTS

4.1. Noisy Chirp Signal

We have tested our extrema-based transform algorithm on various synthetic signals with artificially added white Gaussian noise. The first example is the corrupted by noise ($SNR = -9db$) linear chirp signal ranging from $0Hz$ to $30Hz$. The sampling frequency is $F_s = 10KHz$. The original signal and corresponding noisy version with chirp superimposed on are shown in Fig. 3 (a), (b).

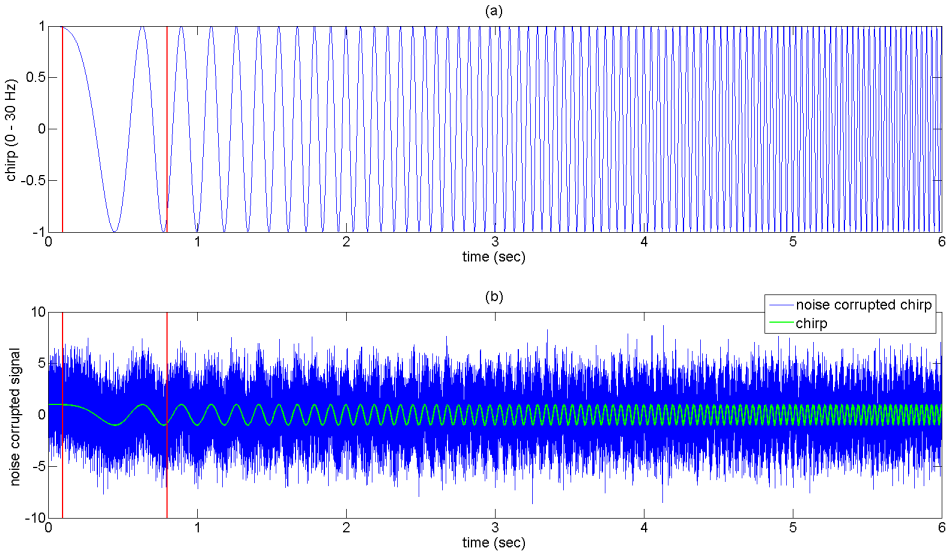

Figure 3: (a) Linear chirp signal. (b) Noise corrupted signal with original signal superimposed on it.

We are interested in capturing the time duration where original signal dominant frequency is in the $0.5-4$ Hz range. We have selected this frequency range because it happens to coincide with the delta brain wave frequency range, which is particularly difficult to identify with other transforms. We apply our proposed algorithm on the signal with $f_{min} = 0.5$, $f_{max} = 4$. After $n = 13$ iterations (Algorithm steps 6, 7) the number of samples left becomes 3. So the chirp signal transform is,

$$T[x_{chirp}(t)] = \sum_{j=1}^{13} v_j \tag{4.1}$$

In Fig. 4(a), (b) the graphical representation of chirp signal transform is displayed. Darker image colors demonstrate larger vertical distances between the successive extrema points; each row corresponds to a single level of transform; the sizes of a single color intervals show the time distance between the successive extrema. The region when original signal frequency is $0.5-4$ Hz is marked by two red vertical lines. It can be observed that the transform produces large value inside the bounded area as compared to other regions.

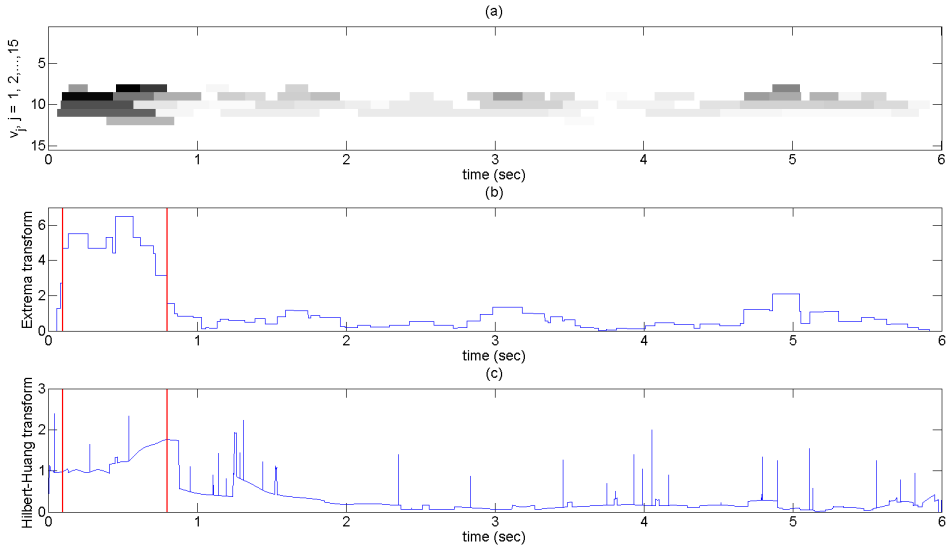

Figure 4: (a) Image representation of matrix created by vertical concatenation of v_1, v_2, ..., v_n. (b) Extrema transform applied the the noise corrupted chirp signal. (c) HHT of the signal.

We apply HHT method on the noisy chirp to compare the results between HHT and Extrema Transform. In HHT sifting process, the signal is decomposed into 15 IMFs, namely $c_1, c_2, ..., c_{15}$ and residue r_{15}. Instantaneous amplitude $a_1(t), a_2(t), ..., a_{15}(t)$ and corresponding instantaneous frequencies $\omega_1(t), \omega_2(t), ..., \omega_{15}(t)$ are calculated using (1.4) and (1.5). As we are interested in $0.5-4$ Hz frequency, so we kept only the amplitude values when instantaneous frequency is in between the range.

$$a_j^*(k) = \begin{cases} a_j(k), & \text{if } 0.5 \leq \omega_j(k) \leq 4 \\ 0, & \text{otherwise} \end{cases} \tag{4.2}$$

Here, $j = 1, 2, ..., 15$ and k is k-th sample of a_j and ω_j. We add the modified amplitudes to calculate total amplitude value of the decomposed signal in the range.

$$HHT(x_{chirp}(t)) = \sum_{j=1}^{15} a_j^* \tag{4.3}$$

In Fig. 4(c), we plot the HHT transform. It shows a comparable result to Extrema Transform, i.e. large values inside the marked region and smaller values outside. But in HHT, large value can also be observed around the surrounding area near of selected frequency range. Also significant spikes in different places can be noticed. We present a quantitative comparison between two methods in Table 1.

Table 1: Comparison of results between Extrema Transform and HHT

Measure	Extrema transform			HHT		
	0.5-4Hz	Other	Ratio	0.5-4Hz	Other	Ratio
Mean	5.0636	0.6305	8.0307	1.2463	0.2486	5.0136
Median	4.8412	0.5157	9.3875	1.1989	0.1627	7.3683
Std. Deviation	0.9256	0.4965	1.8644	0.2724	0.2790	0.9760
Area	3.5447	3.3422	1.0606	0.8724	1.3176	0.6621

From Table 1 it can be observed that Extrema transform produces relatively larger value in terms of mean, median, standard deviation and area for the time duration of the considered frequency range as compared to HHT.

4.2. Performance Analysis under Noisy Conditions

In this example, we study the effect of noise on the signal $x(t)$. The sampling frequency, time duration and amplitude of the signal are 10kHz, 60sec and 1, respectively. The frequency of the signal changes over time and we are interested in capturing the time duration when the frequency of the signal is $0.5 - 4$Hz.

In Fig. 5(a), the original signal is displayed. Fig. 5(b) and Fig. 5(c) demonstrate Extrema transform and STFT of the signal. It is evident from Fig. 5(a) and 5(b) that the mentioned frequency range is present in the signal during $5-10$sec and $30-40$sec.

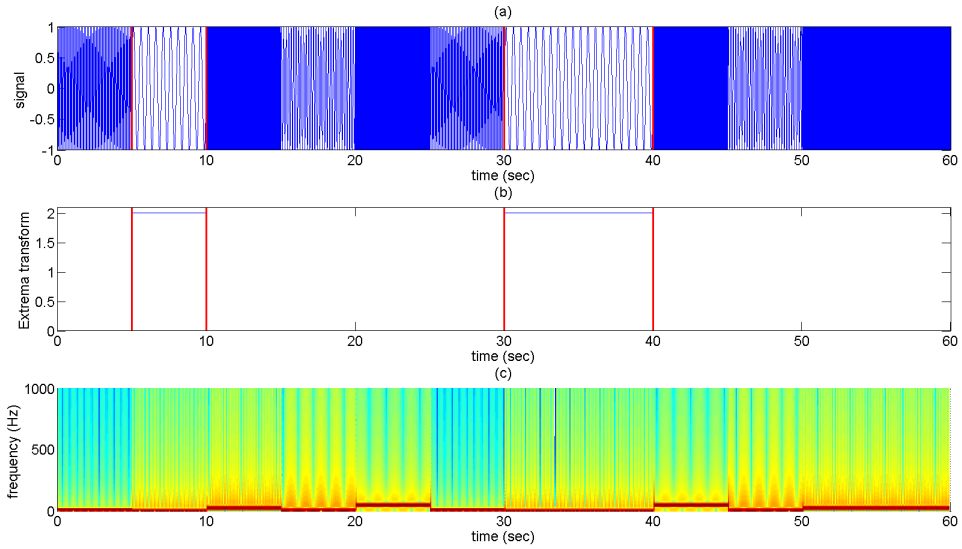

Figure 5: (a) Original signal. (b) Extrema transform of the signal. (c) STFT of the signal.

To analyze the performance of the Extrema Transform under noisy conditions, we add zero mean Gaussian noise of standard deviation σ.

$$x_{noisy}(t) = x(t) + \eta; \quad \eta \sim N(0, \sigma^2) \tag{4.4}$$

In Fig. 6(a), 7(a), 8(a), 9(a), signal corrupted by Gaussian noise of standard deviation σ of 1, 2, 5 and 7, respectively are displayed. SNR for σ of 1, 2, 5 and 7 are $-3db$, $-9db$, $-17db$ and $-20db$, respectively for the signal. It can be noticed from the transforms in Fig. 6(b), 7(b), 8(b), 9(b) that though the noise amount is increasing as σ increases, the transform produces larger value during $0.5-4$Hz duration even when noise is orders of magnitude greater than the signal. In each case, the $0.5-4$ Hz region are bounded by vertical lines and can be easily distinguished from other region by transform value. Also compared to the corresponding STFT, the transform produces better result and time of concerned frequency range can be pinpointed easily.

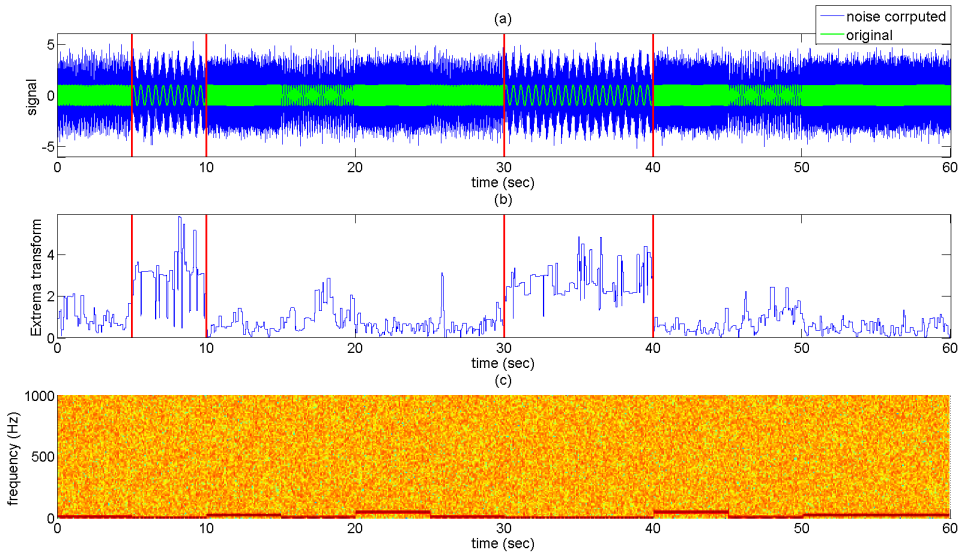

Figure 6: (a) Gaussian noise corrupted signal, $\sigma = 1$. (b) Extrema transform of the signal. (c) STFT of the signal.

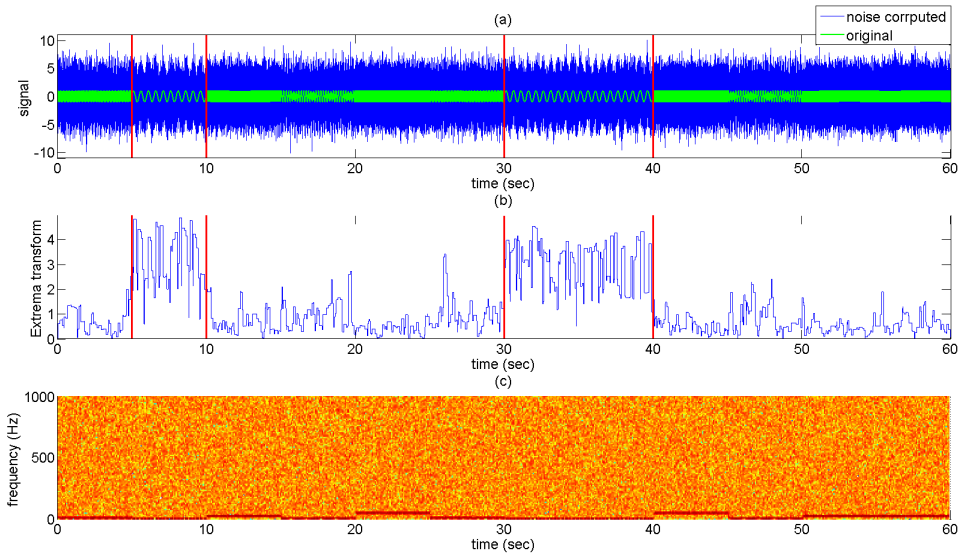

Figure 7: (a) Gaussian noise corrupted signal, $\sigma = 2$. (b) Extrema transform of the signal. (c) STFT of the signal.

In Fig. 10, we plot the ratio of mean transform values between considered frequency region and other region vs noise standard deviation.

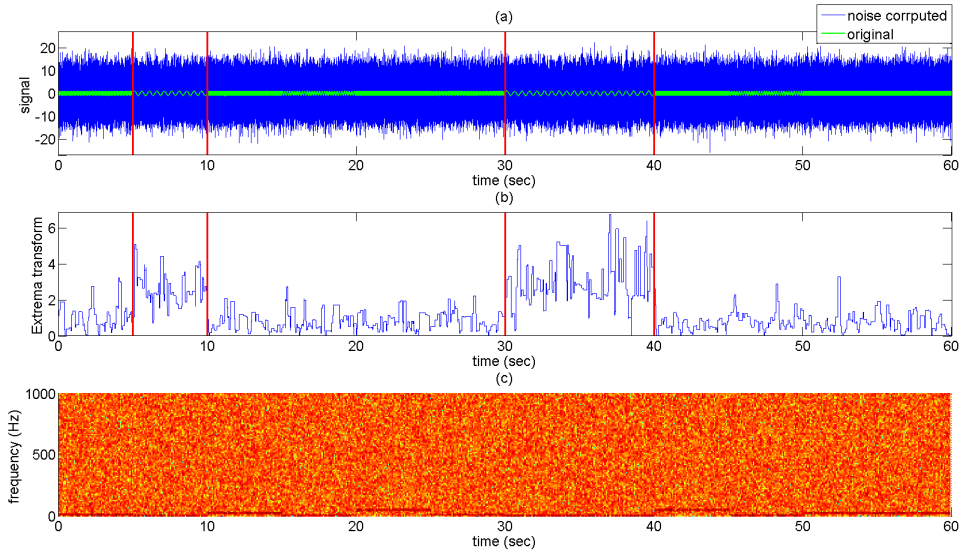

Figure 8: (a) Gaussian noise corrupted signal, $\sigma = 5$. (b) Extrema transform of the signal. (c) STFT of the signal.

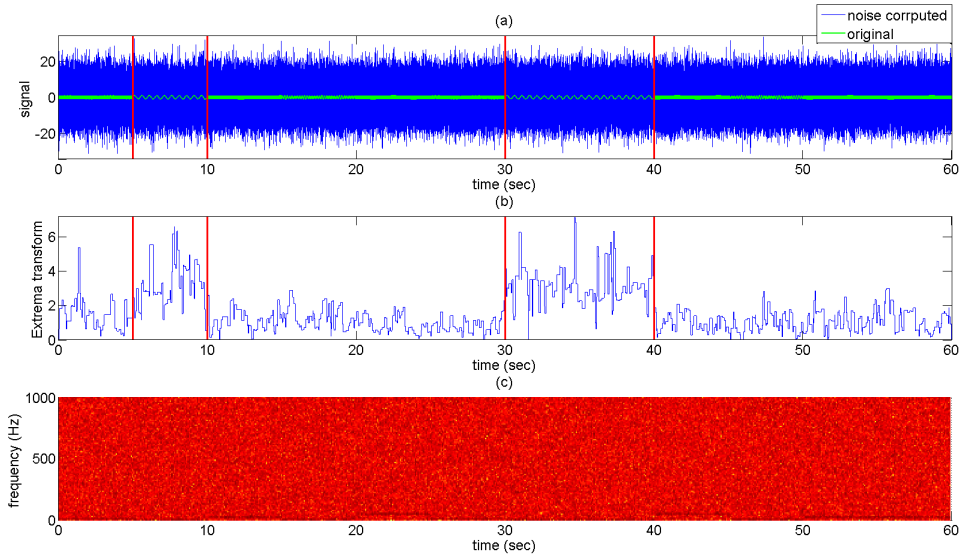

Figure 9: (a) Gaussian noise corrupted signal, $\sigma = 7$. (b) Extrema transform of the signal. (c) STFT of the signal.

$$mean_1 = mean(T[x_{noisy}(t)]), \quad if \quad 0.5 \leq freq(x(t)) \leq 4$$

$$mean_2 = mean(T[x_{noisy}(t)]), \quad if \quad freq(x(t)) \leq 0.5 \quad or \quad freq(x(t)) \geq 4 \tag{4.5}$$

$$ratio = \frac{mean_1}{mean_2}$$

Here, $x(t)$ is the original signal, $x_{noisy}(t)$ is the noise corrupted signal, $freq(x(t))$ is the frequency of original signal at time t, and $T[x_{noisy}(t)])$ is the Extrema Transform of noisy signal.

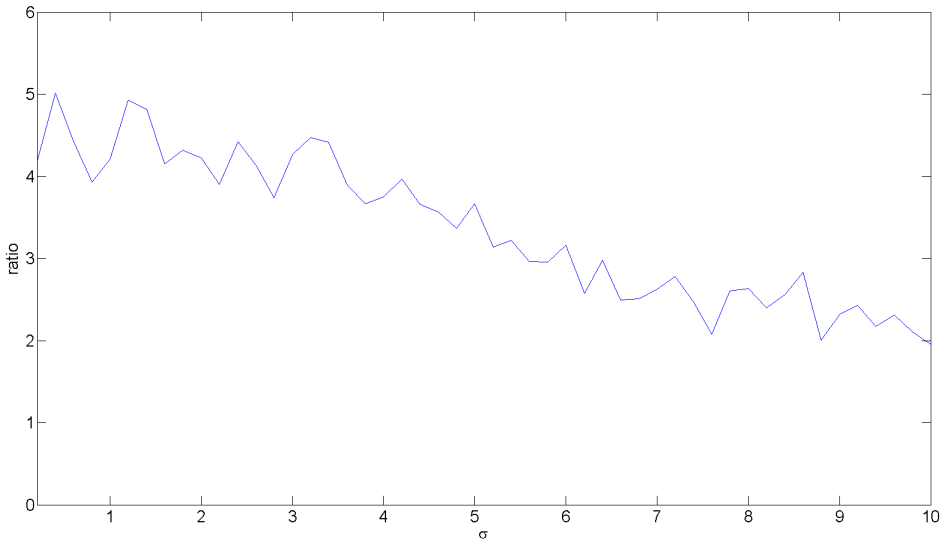

Figure 10: Mean transform amplitude ratio. vs noise standard deviation, σ

From Fig. 10, we observe that for standard deviation of noise between $0.2-2$, the Extrema transform produces on average 4 times larger value in the interested frequency region compared to other regions. With the increase of noise standard deviation, the ratio deceases slowly. Under very strong noise ($\sigma = 10$, $SNR = -23db$), the ratio becomes 2. It implies under extremely noisy condition the transform would produce on average twice as large value in the frequency range region in consideration. So, the transform would be useful in detecting the interested frequency range under extremely noisy condition.

4.3. Application to EEG signal to Detect Delta Wave

In this experiment, we explore the application of Extrema Transform to detect delta wave (0.5-4Hz) in EEG signal. Fig. 11(a) displays the EEG signal and 11(b) shows the STFT of the signal.

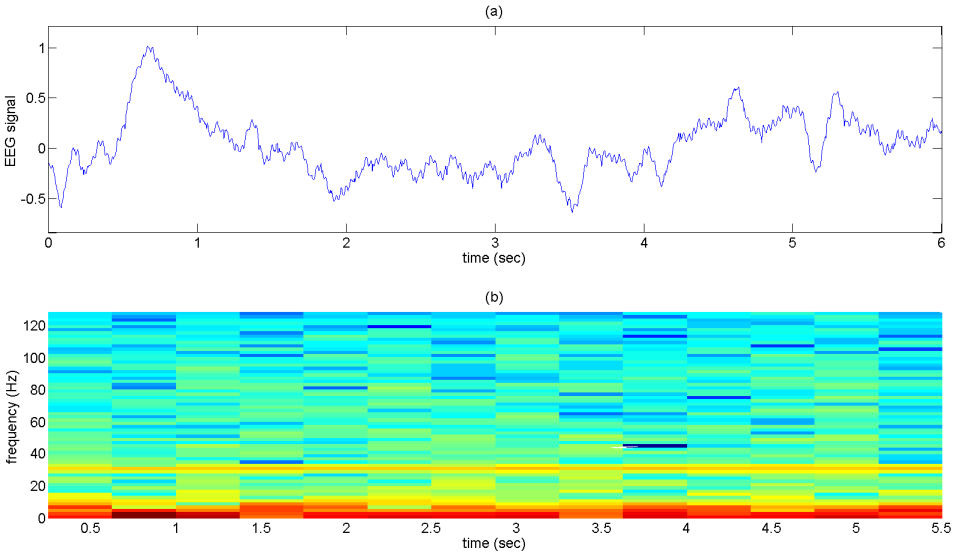

Figure 11: (a) EEG signal. (b) STFT of the signal.

In Fig. 12, the transform reveals delta activity in the whole duration of EEG signal apart from small regions around 2.5sec and 4sec.

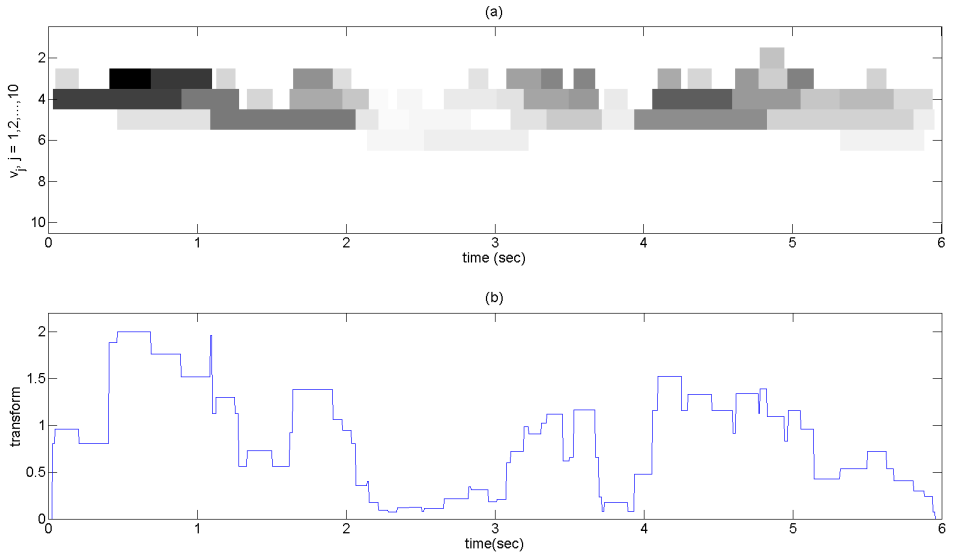

Figure 12: (a) Image representation of the Extrema transform of EEG signal. (b) Extrema transform of EEG signal.

We design a lowpass filter of 4Hz cutoff frequency, 0.01dB pick-to-pick ripple and 80dB stopband attenuation. The magnitude response is displayed in Fig. 13.

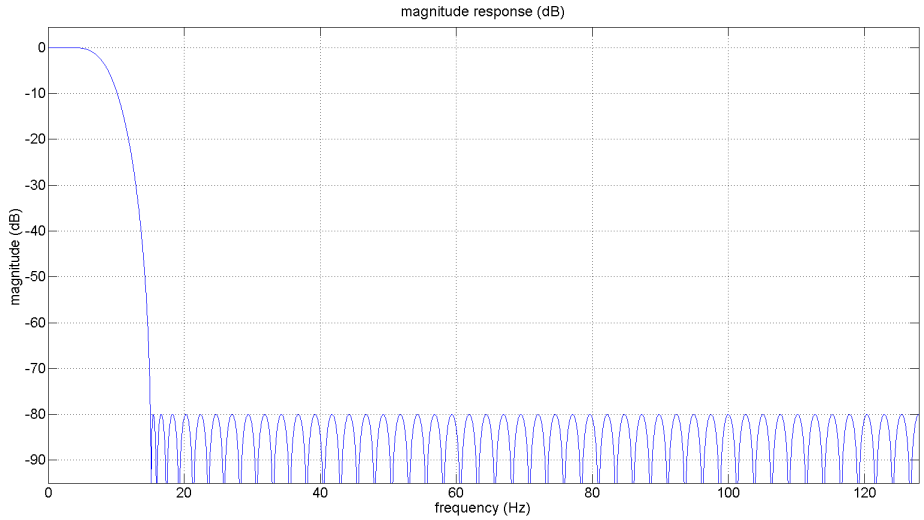

Figure 13: Magnitude response of the designed low-pass filter

Fig. 14 shows the output of filtered signal. It can be observed that filtering smooths the signal preserving the extrema.

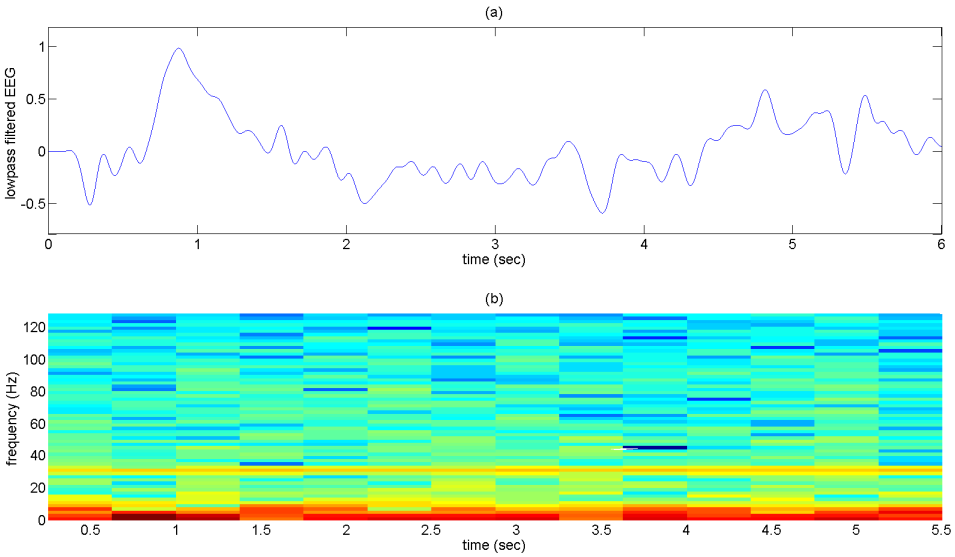

Figure 14: (a) Filtered EEG signal. (b) STFT of the filtered signal.

Fig. 15 displays the output of the Extrema Transform applied on the filtered signal. Comparing Fig. 13 and Fig. 15 we observe that the Extrema Transform on unfiltered and filtered signal produce almost identical result.

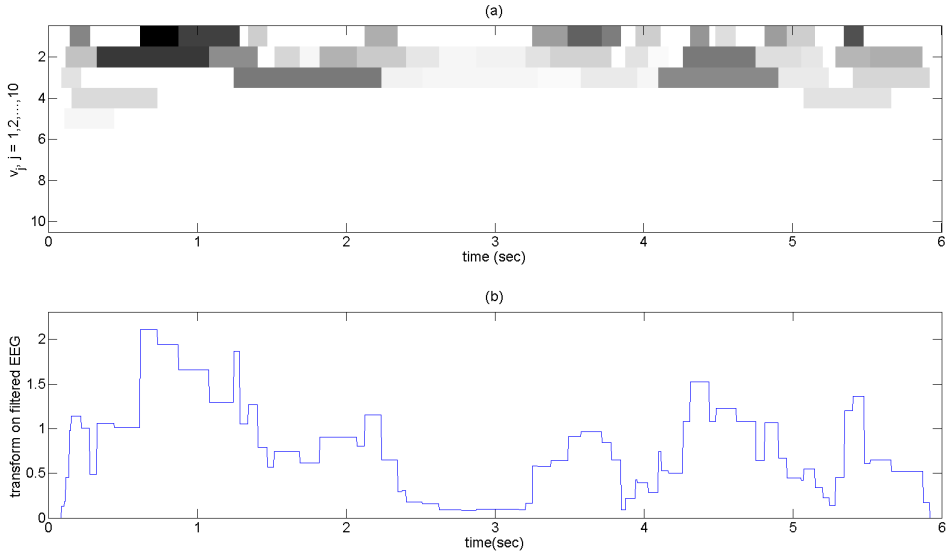

Figure 15: (a) Image representation of the Extrema transform of filtered signal. (b) Extrema transform of the filtered EEG signal.

5. CONCLUSIONS AND FUTURE RESEARCH

5.1. Summary

This dissertation focused on the development and application of a new signal analysis tool based on extrema points. We explained the motivation behind this work, discussed existing transform methods in the field, described the developed algorithm, presented and proved some mathematical properties of the transform, applied the transform on synthetic and real EEG signal and analyze the performance of the method.

First, we applied the transform in a chirp signal corrupted by strong white Gaussian noise where we knew the frequency of any given time of the signal. We were interested in finding the time and duration of occurrence of a particular frequency range. We analyzed the performance of the Extrema Transform with HHT and the results showed that the Extrema Transform produces better result than HHT in detecting interested frequency range.

Next, we tested the performance of the transform under varying noisy conditions. We demonstrated that the transform is robust to noise and separates concerned frequency range regions even when noise is orders of magnitude stronger than the signal itself. The performance of the transform degrades very slowly with the increase of noise power.

Finally, we applied the transform on a real EEG signal which is non-stationary in nature. It effectively found delta wave regions present in the signal. The example demonstrated that it can be applied to real world signals as a faster and efficient alternative to traditional signal analysis methods.

5.2. Limitations of Extrema Transform

Though Extrema Transform has great potential for analyzing non-stationary signals and produces excellent performance in the presence of strong noise, some certain issues should be addressed to broaden its applicability. A few limitations of the transform are presented:

- The transform performs poorly at the presence of multiple dominant frequencies of comparable energy at the same time.
- The transform is not sensitive to signal shape, it considers only the extrema points in each iteration to calculate signal samples for the next iteration. Because of that, signals of different shape but same extrema points would yield same transform output. For example, sinusoid and triangular waves of same signal, phase and amplitude would produce same result in the output.
- Extrema transform does not have an inverse, the original signal can not be re-constructed from the transformation. As our purpose is to analyze non-stationary signals, it is not necessary to have an inverse of the transform for analysis.
- All vector v_js are given same weight in the algorithm, whereas it is evident that the first few vectors are most important. On the other hand, the first few vectors contain only noise components in the presence of noise. Assigning weight to v_j vectors may increase performance of the transform.
- In the Extrema transform algorithm, based on the distance between two consecutive extrema, a value of either the difference of the extrema points or zero is placed in v_j in between the the extrema points sample location (2.2). This may lead to misidentification of frequency regions when frequencies are very close to f_{min} and f_{max} under noisy conditions.

5.3. Possible Improvement and Future Work

In this section we would discuss the possible improvements of the Extrema Transform to address the issues presented in the previous section.

- Apply Extrema Transform on IMFs of HHT to produce good result when more than one dominant frequency of comparable energy are present. Sifting process in HHT decomposes signal into IMFs of different frequency band, applying Extrema Transform on IMFs separately should enable us to analyze signals where multiple dominant frequencies of comparable energy are present simultaneously.

- Consider both the extrema points and the curvature of the signal between them in the transformation algorithm. This would make the transform shape variant.
- Explore the importance of an inverse of the transform and develop an approximate inverse.
- Given the amount of noise present in the signal is known, assign weight to different v_js to produce best result.
- Rather than assigning either the difference between two extrema values or zero in (2.2), we would assign a value considering how close the distance of the extrema is to I_{min} and I_{max}. In case the distance is in between I_{min} and I_{max}, the value would be larger in the middle and linearly decreasing but nonzero in both directions towards I_{min} and I_{max}. When the distance is either greater than I_{max} or less than I_{min}, the value would be larger around I_{max} and I_{min} and monotonically decreasing as distance increases or decreases, respectively.

REFERENCE

[1] M. Thomas, H. Sing, G. Belenky, H. Holcomb, H. Mayberg, R. Dannals, J. Wagner, D. Thorne, K. Popp, L. Rowland *et al.*, "Neural basis of alertness and cognitive performance impairments during sleepiness. i. effects of 24 h of sleep deprivation on waking human regional brain activity," *Journal of sleep research*, vol. 9, no. 4, pp. 335–352, 2000.

[2] N. E. Huang, C. C. Chern, K. Huang, L. W. Salvino, S. R. Long, and K. L. Fan, "A new spectral representation of earthquake data: Hilbert spectral analysis of station tcu129, chi-chi, taiwan, 21 september 1999," *Bulletin of the Seismological Society of America*, vol. 91, no. 5, pp. 1310–1338, 2001.

[3] A. Tsanas, M. A. Little, P. E. McSharry, J. Spielman, and L. O. Ramig, "Novel speech signal processing algorithms for high-accuracy classification of parkinson's disease," *Biomedical Engineering, IEEE Transactions on*, vol. 59, no. 5, pp. 1264–1271, 2012.

[4] B. L. Barnhart, "The hilbert-huang transform: theory, applications, development," 2011.

[5] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, and H. H. Liu, "The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis," in *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 454, no. 1971. The Royal Society, 1998, pp. 903–995.

[6] M. X. Cohen, *Analyzing neural time series data: theory and practice*. MIT Press, 2014.

[7] A. V. Oppenheim, A. S. Willsky, and S. H. Nawab, *Signals and systems*. Pearson, 2014.

[8] A. V. Oppenheim, R. W. Schafer, J. R. Buck *et al.*, *Discrete-time signal processing*. Prentice hall Englewood Cliffs, NJ, 1989, vol. 2.

[9] J. Allen, "Short-term spectral analysis, and modification by discrete fourier transform," *IEEE Transactions on Acoustics Speech and Signal Processing*, vol. 25, no. 3, pp. 235–238, 1977.

[10] J. B. Allen and L. R. Rabiner, "A unified approach to short-time fourier analysis and synthesis," *Proceedings of the IEEE*, vol. 65, no. 11, pp. 1558–1564, 1977.

[11] T. Schlurmann, "The empirical mode decomposition and the hilbert spectra to analyse embedded characteristic oscillations of extreme waves," in *Rogue Waves*, 2001, pp. 157–165.

[12] A. Grossmann and J. Morlet, "Decomposition of hardy functions into square integrable wavelets of constant shape," *SIAM journal on mathematical analysis*, vol. 15, no. 4, pp. 723–736, 1984.

[13] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Communications on pure and applied mathematics*, vol. 41, no. 7, pp. 909–996, 1988.

[14] H. Sharabaty, J. Martin, B. Jammes, and D. Esteve, "Alpha and theta wave localisation using hilbert-huang transform: empirical study of the accuracy," in *Information and Communication Technologies, 2006. ICTTA'06. 2nd*, vol. 1. IEEE, 2006, pp. 1159–1164.

[15] G. Rilling, P. Flandrin, P. Goncalves *et al.*, "On empirical mode decomposition and its algorithms," in *IEEE-EURASIP workshop on nonlinear signal and image processing*, vol. 3. IEEER, 2003, pp. 8–11.

[16] G. Rilling, P. Flandrin, P. Gonçalves, and J. M. Lilly, "Bivariate empirical mode decomposition," *Signal Processing Letters, IEEE*, vol. 14, no. 12, pp. 936–939, 2007.

[17] Z. Zhidong, Z. Zhijin, and C. Yuquan, "Time-frequency analysis of heart sound based on hht [hilbert-huang transform]," in *Communications, Circuits and Systems, 2005. Proceedings. 2005 International Conference on*, vol. 2. IEEE, 2005.

[18] N. E. Huang, M.-L. Wu, W. Qu, S. R. Long, and S. S. Shen, "Applications of hilbert–huang transform to non-stationary financial time series analysis," *Applied stochastic models in business and industry*, vol. 19, no. 3, pp. 245–268, 2003.

[19] N. E. Huang and Z. Wu, "A review on hilbert-huang transform: Method and its applications to geophysical studies," *Reviews of Geophysics*, vol. 46, no. 2, 2008.

[20] N. Senroy, S. Suryanarayanan, and P. F. Ribeiro, "An improved hilbert–huang method for analysis of time-varying waveforms in power quality," *Power Systems, IEEE Transactions on*, vol. 22, no. 4, pp. 1843–1850, 2007.

[21] N. C. Tse, J. Y. Chan, W.-H. Lau, and L. L. Lai, "Hybrid wavelet and hilbert transform with frequency-shifting decomposition for power quality analysis," *Instrumentation and Measurement, IEEE Transactions on*, vol. 61, no. 12, pp. 3225–3233, 2012.

[22] M. J. Afroni, D. Sutanto, and D. Stirling, "Analysis of nonstationary power-quality waveforms using iterative hilbert huang transform and sax algorithm," *Power Delivery, IEEE Transactions on*, vol. 28, no. 4, pp. 2134–2144, 2013.

[23] Z.-F. Liu, Z.-P. Liao, and E.-F. Sang, "Speech enhancement based on hilbert-huang transform," in *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on*, vol. 8. IEEE, 2005, pp. 4908–4912.

[24] X. Zou, X. Li, and R. Zhang, "Speech enhancement based on hilbert-huang transform theory," in *Computer and Computational Sciences, 2006. IMSCCS'06. First International Multi-Symposiums on*, vol. 1. IEEE, 2006, pp. 208–213.

[25] H. Huang and J. Pan, "Speech pitch determination based on hilbert-huang transform," *Signal Processing*, vol. 86, no. 4, pp. 792–803, 2006.

[26] G. Schlotthauer, M. E. Torres, and H. L. Rufiner, "A new algorithm for instantaneous f 0 speech extraction based on ensemble empirical mode decomposition," in *Signal Processing Conference, 2009 17th European*. IEEE, 2009, pp. 2347–2351.

[27] S. K. Roy and W.-P. Zhu, "Pitch estimation of noisy speech using ensemble empirical mode decomposition and dominant harmonic modification," in *Electrical and Computer Engineering (CCECE), 2014 IEEE 27th Canadian Conference on*. IEEE, 2014, pp. 1–4.

[28] L. Yu, S. Wang, K. K. Lai, and F. Wen, "A multiscale neural network learning paradigm for financial crisis forecasting," *Neurocomputing*, vol. 73, no. 4, pp. 716–725, 2010.

[29] K. Ju, D. Zhou, P. Zhou, and J. Wu, "Macroeconomic effects of oil price shocks in china: An empirical study based on hilbert–huang transform and event study," *Applied Energy*, vol. 136, pp. 1053–1066, 2014.

[30] Y.-Y. Sun, C.-H. Chen, J.-Y. Liu, C.-H. Wang, and D.-L. Chen, "Instantaneous phase shift of annual subsurface temperature cycles derived by the hilbert-huang transform," *Journal of Geophysical Research: Atmospheres*, vol. 120, no. 5, pp. 1670–1677, 2015.

[31] J. Han and M. van der Baan, "Empirical mode decomposition for seismic time-frequency analysis," *Geophysics*, vol. 78, no. 2, pp. O9–O19, 2013.

[32] A. G. Espinosa, J. A. Rosero, J. Cusido, L. Romeral, and J. A. Ortega, "Fault detection by means of hilbert–huang transform of the stator current in a pmsm with demagnetization," *Energy Conversion, IEEE Transactions on*, vol. 25, no. 2, pp. 312–318, 2010.

[33] G. Cheng, Y.-l. Cheng, L.-h. Shen, J.-b. Qiu, and S. Zhang, "Gear fault identification based on hilbert–huang transform and som neural network," *Measurement*, vol. 46, no. 3, pp. 1137–1146, 2013.

[34] Z. Su, Y. Zhang, M. Jia, F. Xu, and J. Hu, "Gear fault identification and classification of singular value decomposition based on hilbert-huang transform," *Journal of mechanical science and technology*, vol. 25, no. 2, pp. 267–272, 2011.

[35] A.-O. Boudraa and J.-C. Cexus, "Emd-based signal filtering," *Instrumentation and Measurement, IEEE Transactions on*, vol. 56, no. 6, pp. 2196–2202, 2007.

[36] Y. Kopsinis and S. McLaughlin, "Development of emd-based denoising methods inspired by wavelet thresholding," *Signal Processing, IEEE Transactions on*, vol. 57, no. 4, pp. 1351–1362, 2009.

[37] N. Chatlani and J. J. Soraghan, "Emd-based filtering (emdf) of low-frequency noise for speech enhancement," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 4, pp. 1158–1166, 2012.

[38] S. A. Elgamel and J. J. Soraghan, "Using emd-frft filtering to mitigate very high power interference in chirp tracking radars," *Signal Processing Letters, IEEE*, vol. 18, no. 4, pp. 263–266, 2011.

[39] A. Komaty, A.-O. Boudraa, B. Augier, and D. Daré-Emzivat, "Emd-based filtering using similarity measure between probability density functions of imfs," *Instrumentation and Measurement, IEEE Transactions on*, vol. 63, no. 1, pp. 27–34, 2014.

[40] H. Li, S. Kwong, L. Yang, D. Huang, and D. Xiao, "Hilbert-huang transform for analysis of heart rate variability in cardiac health," *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, vol. 8, no. 6, pp. 1557–1567, 2011.

[41] T.-H. Hung, C.-C. Chou, W.-C. Fang, A. H.-T. Li, Y.-C. Chang, B.-K. Hwang, and Y.-W. Shau, "Time-frequency analysis of heart sound signals based on hilbert-huang transformation," in *Consumer Electronics (ISCE), 2012 IEEE 16th International Symposium on*. IEEE, 2012, pp. 1–3.

[42] S.-Z. Fan, Q. Wei, P.-F. Shi, Y.-J. Chen, Q. Liu, and J.-S. Shieh, "A comparison of patients heart rate variability and blood flow variability during surgery based on the hilbert–huang transform," *Biomedical Signal Processing and Control*, vol. 7, no. 5, pp. 465–473, 2012.

[43] L. Xu, Y. Wang, Y. Yao, C. Feng, Y. Zhao, and M. Q. Meng, "Comparison of six envelope extraction methods based on abnormal heart sounds," in *Biomedical Engineering and Informatics (BMEI), 2010 3rd International Conference on*, vol. 2. IEEE, 2010, pp. 813–817.

[44] M.-T. Lo, K. Hu, Y. Liu, C.-K. Peng, and V. Novak, "Multimodal pressure-flow analysis: Application of hilbert huang transform in cerebral blood flow regulation," *EURASIP journal on advances in signal processing*, vol. 2008, no. 1, pp. 1–15, 2008.

[45] Q. Zhang, Y. Shi, D. Teng, A. Dinh, S.-B. Ko, L. Chen, J. Basran, D. Bello-Haas, Y. Choi *et al.*, "Pulse transit time-based blood pressure estimation using hilbert-huang transform," in *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*. IEEE, 2009, pp. 1785–1788.

[46] Y. Choi, Q. Zhang, and S. Ko, "Noninvasive cuffless blood pressure estimation using pulse transit time and hilbert–huang transform," *Computers & Electrical Engineering*, vol. 39, no. 1, pp. 103–111, 2013.

[47] R. J. Oweis and E. W. Abdulhay, "Seizure classification in eeg signals utilizing hilbert-huang transform," *Biomedical engineering online*, vol. 10, no. 1, pp. 38–52, 2011.

[48] J. Yan and L. Lu, "Improved hilbert–huang transform based weak signal detection methodology and its application on incipient fault diagnosis and ecg signal analysis," *Signal Processing*, vol. 98, pp. 74–87, 2014.

[49] Y. Li, F. Yingle, L. Gu, and T. Qinye, "Sleep stage classification based on eeg hilbert-huang transform," in *Industrial Electronics and Applications, 2009. ICIEA 2009. 4th IEEE Conference on*. IEEE, 2009, pp. 3676–3681.

[50] S. Li, W. Zhou, Q. Yuan, S. Geng, and D. Cai, "Feature extraction and recognition of ictal eeg using emd and svm," *Computers in biology and medicine*, vol. 43, no. 7, pp. 807–816, 2013.

[51] F. Duman, N. Özdemir, and E. Yildirim, "Patient specific seizure prediction algorithm using hilbert-huang transform," in *Biomedical and Health Informatics (BHI), 2012 IEEE-EMBS International Conference on*. IEEE, 2012, pp. 705–708.

[52] Y. Liu, L. Yan, B. Zeng, and W. Wang, "Automatic sleep stage scoring using hilbert-huang transform with bp neural network," in *Bioinformatics and Biomedical Engineering (iCBBE), 2010 4th International Conference on*. IEEE, 2010, pp. 1–4.

[53] V. Bajaj and R. B. Pachori, "Separation of rhythms of eeg signals based on hilbert-huang transformation with application to seizure detection," in *Convergence and Hybrid Information Technology*. Springer, 2012, pp. 493–500.

[54] Y. Chen and M. Q. Feng, "A technique to improve the empirical mode decomposition in the hilbert-huang transform," *Earthquake Engineering and Engineering Vibration*, vol. 2, no. 1, pp. 75–85, 2003.

APPENDIX

MATLAB CODE

Functions to determine extrema points of input signal:

```
function maxima = findmaxima(x)
%FINDMAXIMA  Find location of local maxima
%  From David Sampson
% Identify whether signal is rising or falling
upordown = sign(diff(x));
% Find points where signal is rising before, falling after
maxflags = [upordown(1)<0; diff(upordown)<0; upordown(end)>0];
maxima   = find(maxflags);
```

```
function [x,t] = findextrema(x,t)
%FINDEXTREMA  Find location of local extrema (both maxima and minima)
%  From David Sampson
% Identify whether signal is rising or falling
upordown_ind = sign(diff(x));
% Find points where signal is rising before, falling after
extremaflags = [1; diff(upordown_ind)~=0; 1];
ind = find(extremaflags);
t   = t(ind);
x = x(ind);
```

```
function [x,t] = scaledown(x,t)
% SCALEDOWN places the average of two signal samples in the middle
t = (t(1:end-1)+t(2:end))/2;
x = (x(1:end-1)+x(2:end))/2;
```

Code for generating Fig. 1

```
clear all;
clc;
close all;

fontsize = 18;
amplitude = 1;
stopTime = 10; % signal duration
frequency = 10;
phase = 0;
Fmin = 9;
Fmax = 11;
T = 1000;  % Samples per second

Imin = floor(T/Fmax/2); % minimal interval between two successive
% extrema
Imax = ceil(T/Fmin/2);  % maximum interval between two successive
% extrema

signal = sineSignal(amplitude, T, stopTime, frequency, phase);
N = numel(signal);
figure(1)
subplot(311)
```

```
plot(linspace(0, stopTime, N), signal)
set(gca,'FontSize',fontsize)
pos1 = get(gca, 'position');
title('(a)')
xlabel('time (sec)')
ylabel('sinusoid')
level = 10;
%%%%%%%%%%%%%%
mat = zeros(10,N);
t1 = 0:N-1;
for z=1:level
if numel(signal)>3
[x,t] = findextrema(signal,t1);  % extrema points and indices
dx = diff(x);
for m=1:numel(dx)
dist = abs(dx(m));
if (t(m+1)-t(m))>= Imin && (t(m+1)-t(m))<=Imax
mat(z,1+(round(t(m)):round(t(m+1))))= dist;
else
mat(z,1+(round(t(m)):round(t(m+1))))= 0;
end
end
[signal,t1] = scaledown(x,t);
end
end

transform = sum(mat);
subplot(312)
```

```
set(gca,'FontSize',fontsize)
imagesc(0:1/T:stopTime-1/T, 1:level, mat);

title('(b)')
colormap(flipud(gray))
caxis([0, 2])
colorbar
set(colorbar, 'fontsize', fontsize)
axis on;
%tickMarks = {'XTick',[]};
%set(gca, 'XTicksVisible', false);
xlabel('time (sec)')
ylabel('v_{j}, j = {1, 2,...,10}')
pos2 = get(gca, 'position');
set(gca, 'position', [pos2(1) pos2(2) (pos1(3)-0.025) pos2(4)])
subplot(313)
set(gca,'FontSize',fontsize)
plot(linspace(0, stopTime, N), transform)
axis([0 stopTime 0 2*amplitude+0.1])
title('(c)')
xlabel('time (sec)')
ylabel('transform')
axis([0 10 0 max(transform)+0.1])


function signal = sineSignal(amplitude, samplePerSecond, stopTime,...
         frequency, phase)
phase = (pi/180) * phase; % degree to radian
% samples per second
```

```
dt = 1/samplePerSecond;                    % seconds per sample
t = (0:dt:stopTime - dt)';     % seconds
signal = amplitude * sin(2*pi*frequency*t + phase);
end
```

Code for performance comparison with Hilbert-Huang transform:

```
clear all;
clc;
close all;

fontsize = 18;
load chirp_noisy

seconds = 6;
Fmin = 0.5; % min of delta wave
Fmax = 4; % max of delta wave
Imin = round(Fs/Fmax/2); %minimal interval in samples
Imax = round(Fs/Fmin/2); %maximal interval in samples
tt = linspace(0, seconds, Fs*seconds);
x1 = chirp(tt, start_frequency, seconds, end_frequency);

subplot(211)
plot(tt, x1)
set(gca,'FontSize',fontsize)
title('(a)')
xlabel('time (sec)')
ylabel('chirp (0 - 30 Hz)')
hold on;
line(0.1*ones(1,3), -1:1, 'color','r', 'linewidth', 2)
```

```
hold on;
line((4/5)*ones(1,3), -1:1, 'color','r', 'linewidth', 2)
hold off;

subplot(212)
plot(tt, signal)
set(gca,'FontSize',fontsize)
title('(b)')
xlabel('time (sec)')
ylabel('noise corrupted signal')
hold on;
plot(tt, x1, 'g', 'linewidth', 1.5)
hold on;
legend('noise corrupted chirp', 'chirp')
%legend('boxoff')
line(0.1*ones(1,3), -10:10:10, 'color','r', 'linewidth', 2)
hold on;
line((4/5)*ones(1,3), -10:10:10, 'color','r', 'linewidth', 2)
hold off;
pos1 = get(gca, 'position');
% subplot(313)
% spectrogram(x1, 128, 32, 128, Fs, 'yaxis')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
N = size(signal, 1);
mat = zeros(15,N);
t1 = 0:N-1;
x1 = signal;
label = 15;
```

```
%ind = 1:numel(t1); %[1:2500]
for z=1:label
if numel(x1)>3
[x,t] = findextrema(x1,t1);  % extrema points and indices
dx = diff(x);
for m=1:numel(dx)
dist = abs(dx(m));
if (t(m+1)-t(m))>Imin && (t(m+1)-t(m))<Imax
mat(z,1+(round(t(m)):round(t(m+1))))=(1)*dist;
else
mat(z,1+(round(t(m)):round(t(m+1))))=0*dist;
end
end
[x1,t1] = scaledown(x,t);
% x1 = x1*2;
end


end


figure;
subplot(311)
set(gca,'FontSize',fontsize)
imagesc(tt, 1:label, mat)
colormap(flipud(gray))
axis on;
colorbar
xlabel('time (sec)')
ylabel('v_{j}, j = {1, 2,...,15}')
```

```
title('(a)')
pos2 = get(gca, 'position');
set(gca, 'position', [pos2(1) pos2(2) (pos1(3)-0.025) pos2(4)])
set(colorbar, 'fontsize', fontsize)
subplot(312)
plot((0:N-1)/Fs, sum(mat))
set(gca,'FontSize',fontsize)
title('(b)')
xlabel('time (sec)')
ylabel('Extrema transform')
axis([0 seconds 0 7])
hold on;
line(0.1*ones(1,8), 0:7,  'color','r', 'linewidth', 2)
hold on;
line((4/5)*ones(1,8), 0:7, 'color','r', 'linewidth', 2)
hold off;

interpolation = 'spline'; % other options: linear, cubic, spline
imf = emd(signal, 'interp', interpolation);
[A,f,tt] = hhspectrum(imf(1:end-1,:));  % calculating imf signal
% amplitude, instantaneous frequency and time instances

f = f*Fs;
% %max(f)

mask1 = (f <=Fmax);
mask2 = (f >=Fmin);
mask = mask1.*mask2;
```

```
maskNotRange = ~mask;

%mask = (f <=4);

f1 = f.*mask; % retaining only the delta wave frequencies
A1 = A.*mask; % retaining only the delta amplitude

subplot(313)
plot(tt/Fs, sum(A1))
set(gca,'FontSize',fontsize)
title('(c)')
xlabel('time (sec)')
ylabel('Hilbert-Huang transform')
axis([0 seconds 0 3])
hold on;
line(0.1*ones(1,4), 0:3, 'color','r', 'linewidth', 2)
hold on;
line((4/5)*ones(1,4), 0:3, 'color','r', 'linewidth', 2)
hold off;

%comparing the values inside and outside of range
mattt = (1:N)/Fs;
startTime = 0.1; % when freq = 0.5
stopTime = 0.8; % when freq = 4

matStartIndex = find(mattt >= startTime, 1, 'first');
matStopIndex = find(mattt >= stopTime, 1, 'first');
```

```
matSum = sum(mat);

insiderSum = matSum(matStartIndex:matStopIndex);
outsiderSum = horzcat(matSum(1: matStartIndex - 1), ...
           matSum(matStopIndex + 1:end));

meanInsiderSum = mean(insiderSum);
medianInsiderSum = median(insiderSum);
stdInsiderSum = std(insiderSum);

meanOutsiderSum = mean(outsiderSum);
medianOutsiderSum = median(outsiderSum);
stdOutsiderSum = std(outsiderSum);

ratioMeanSum = meanInsiderSum/meanOutsiderSum;
ratioMedianSum = medianInsiderSum/medianOutsiderSum;
ratioStdSum = stdInsiderSum/stdOutsiderSum;

%%%%% are calculaltion
totalAreaSum = trapz(mattt, matSum);
insiderAreaSum = trapz(mattt(matStartIndex:matStopIndex), insiderSum);
outsiderAreaSum = totalAreaSum - insiderAreaSum;
ratioAreaSum = insiderAreaSum / outsiderAreaSum;
% Updated imfs
tt = tt/Fs;

startIndex = find(tt >= startTime, 1, 'first');
stopIndex = find(tt >= stopTime, 1, 'first');
```

```
sumA1 = sum(A1);
insiderImf = sumA1(startIndex:stopIndex);

outsiderImf = horzcat(sumA1(1: startIndex - 1),...
           sumA1(stopIndex + 1:end));
meanInsiderImf = mean(insiderImf);
medianInsiderImf = median(insiderImf);
stdInsiderImf = std(insiderImf);

meanOutsiderImf = mean(outsiderImf);
medianOutsiderImf = median(outsiderImf);
stdOutsiderImf = std(outsiderImf);

ratioMeanInf = meanInsiderImf/meanOutsiderImf;
ratioMedianImf = medianInsiderImf/medianOutsiderImf;
ratioStdImf = stdInsiderImf/stdOutsiderImf;

%%%%% are calculaltion
totalAreaImf = trapz(tt, sumA1);
insiderAreaImf = trapz(tt(startIndex:stopIndex), insiderImf);
outsiderAreaImf = totalAreaImf - insiderAreaImf;
ratioAreaImf = insiderAreaImf / outsiderAreaImf;
```

Code for generating Fig. 10: Performance of Extrema Transform under varying noise

```
clear all;
clc;
close all;
Fs = 10000; % sampling frequency
```

```
time = 5;
dt = 1/Fs;
tt = 0:dt:time-dt;
lnth = length(tt);

% frequency of different segments
deltaf = 2;
betaf = 25;
alphaf = 10;
thetaf = 6;
gammaf = 50;
delta = sin(2*pi*deltaf*tt);
beta = sin(2*pi*betaf*tt);
alpha = sin(2*pi*alphaf*tt);
theta = sin(2*pi*thetaf*tt);
gamma = sin(2*pi*gammaf*tt);

s = [alpha delta beta theta gamma alpha delta delta gamma theta ...
    beta beta]; % signal
N = length(s);
deltaIndex1 = 2;
deltaIndex2 = 7;
deltaIndex3 = 8;

Fmin = 0.5; % min of delta wave
Fmax = 4; % max of delta wave
Imin = round(Fs/Fmax/2); %minimal interval in samples
Imax = round(Fs/Fmin/2); %maximal interval in samples
```

```
totalNumber = 50; % do the iteration for this number of times
ratioArray = zeros(1,totalNumber); % array to hold ratio result
ratioIndex = 1;
factor = 0.2; % each time increase standard deviation (SD) by 0.2

for indx = 1:totalNumber
signal = s + indx*factor*randn(1, N); % adding gaussian noise of ...
% zero mean and standard deviation indx*factor
mat = zeros(15,N);
t1 = 0:N-1;
x1 = signal';
%ind = 1:numel(t1); %[1:2500]
for z=1:15
if numel(x1)>3
[x,t] = findextrema(x1,t1);  % extrema points and indices
dx = diff(x);
for m=1:numel(dx)
dist = abs(dx(m));
if (t(m+1)-t(m))>Imin && (t(m+1)-t(m))<Imax
mat(z,1+(round(t(m)):round(t(m+1))))=(1)*dist;
else
mat(z,1+(round(t(m)):round(t(m+1))))=0*dist;
end
end
end

[x1,t1] = scaledown(x,t);
```

```
end

transform = sum(mat);

% getting the regions when frequency is in delta range
deltaPortion1 = transform(((deltaIndex1 - 1)*lnth)+1 :...
             deltaIndex1*lnth);

deltaPortion2 = transform(((deltaIndex2 - 1)*lnth)+1 :...
             deltaIndex2*lnth);

deltaPortion3 = transform(((deltaIndex3 - 1)*lnth)+1 :...
             deltaIndex3*lnth);

deltaPortions = [deltaPortion1 deltaPortion2 deltaPortion3];
meanDelta = mean(deltaPortions);

% regions frequency not in delta range
otherPortions = [transform(1:lnth) transform(2*lnth+1:6*lnth)...
     transform(8*lnth+1:end)];
meanOther = mean(otherPortions);
ratioArray(indx) = meanDelta / meanOther;
end

plot((1:totalNumber)*factor, ratioArray)
set(gca,'FontSize',20)
xlabel('\sigma')
```

```
ylabel('ratio')
axis([1*factor (totalNumber*factor) 0 6])
```

Code of Extrema transform Applied on EEG signal:

```
clear all;
clc;
close all;

load EEG_cropped_frontal
fontsize = 18;

Fmax = 4; % max of delta wave
Fmin = 0.5; % min of delta wave
Imin = round(Fs/Fmax/2); %minimal interval between successive
       % extrema points
Imax = round(Fs/Fmin/2); %maximal interval between successive
       % extrema points

N = numel(signal);

figure;
subplot(211)
plot((0:N-1)/Fs, signal)
set(gca,'FontSize',fontsize)
pos1 = get(gca, 'position');
title('(a)')
xlabel('time (sec)')
ylabel('EEG signal')
axis([0 6 (min(signal)-0.2) (max(signal)+0.2)])
```

```
subplot(212)
spectrogram(signal, 128, 32, 128, Fs, 'yaxis')
colorbar
ylabel(colorbar, 'Power/frequency', 'fontsize', fontsize)
pos2 = get(gca, 'position');
set(gca, 'position', [pos2(1) pos2(2) (pos1(3)-0.03) pos2(4)])
set(gca,'FontSize',fontsize)
title('(b)')
xlabel('time (sec)')
ylabel('frequency (Hz)')

% transfrom without filtering
mat = zeros(10,N);
t1 = 0:N-1;
x1 = signal;
level = 10;
for z=1:level
if numel(x1)>3
[x,t] = findextrema(x1,t1);  % extrema points and indices
dx = diff(x);
for m=1:numel(dx)
dist = abs(dx(m));
if (t(m+1)-t(m))>Imin && (t(m+1)-t(m))<Imax
mat(z,1+(round(t(m)):round(t(m+1))))=1*dist;
else
mat(z,1+(round(t(m)):round(t(m+1))))=0*dist;
end
end
```

```
[x1,t1] = scaledown(x,t);
% x1 = x1*2;
end
end

transform = sum(mat);
figure;
subplot(211)
imagesc((0:N-1)/Fs,1:level, mat)
colormap(flipud(gray))
colorbar
pos2 = get(gca, 'position');
set(gca, 'position', [pos2(1) pos2(2) (pos1(3)-0.025) pos2(4)])
set(gca,'FontSize',fontsize)
axis on;
set(gca,'FontSize',fontsize)
title('(a)')
xlabel('time (sec)')
ylabel('v_{j}, j = 1,2,...,10')
subplot(212)
plot((0:N-1)/Fs, transform)
set(gca,'FontSize', fontsize)
title('(b)')
xlabel('time(sec)')
ylabel('transform')
axis([0 6 0 max(transform)+0.2])
% filtering
order   = 100;        % FIR filter order
```

```
Fp  = Fmax;       % 4Hz passband-edge frequency
%Fs  = 96e3;       % 96 kHz sampling frequency
Rp  = 0.00057565; % Corresponds to 0.01 dB peak-to-peak ripple
Rst = 1e-4;       % Corresponds to 80 dB stopband attenuation

NUM = firceqrip(order,Fp/(Fs/2),[Rp Rst],'passedge');
% NUM = vector of coeffs
fvtool(NUM,'Fs',Fs,'Color','White') % Visualize filter
set(gca,'FontSize', fontsize)
xlabel('frequency (Hz)')
ylabel('magnitude (dB)')
title('magnitude response (dB)')

signal_filtered = filter(NUM,1,signal);
figure;
subplot(211)
plot((0:N-1)/Fs, signal_filtered)
set(gca,'FontSize',fontsize)
title('(a)')
xlabel('time (sec)')
ylabel('lowpass filtered EEG')
axis([0 6 min(signal_filtered)-0.2 max(signal_filtered)+0.2])
subplot(212)
spectrogram(signal, 128, 32, 128, Fs, 'yaxis')
colorbar
ylabel(colorbar, 'Power/frequency', 'fontsize', fontsize)
pos2 = get(gca, 'position');
set(gca, 'position', [pos2(1) pos2(2) (pos1(3)-0.03) pos2(4)])
```

```
set(gca,'FontSize', fontsize)
title('(b)')
xlabel('time (sec)')
ylabel('frequency (Hz)')

% transfrom after filtering
mat = zeros(10,N);
t1 = 0:N-1;
x1 = signal_filtered;
%ind = 1:numel(t1); %[1:2500]
for z=1:10
if numel(x1)>3
[x,t] = findextrema(x1,t1);  % extrema points and indices
dx = diff(x);
for m=1:numel(dx)
dist = abs(dx(m));
if (t(m+1)-t(m))>Imin && (t(m+1)-t(m))<Imax
mat(z,1+(round(t(m)):round(t(m+1))))=1*dist;
else
mat(z,1+(round(t(m)):round(t(m+1))))=0*dist;
end
end
[x1,t1] = scaledown(x,t);
% x1 = x1*2;
end
end

transform = sum(mat);
```

```
figure;
subplot(211)
imagesc(mat)
%%%%%
imagesc((0:N-1)/Fs,1:level, mat)
colormap(flipud(gray))
colorbar
pos2 = get(gca, 'position');
set(gca, 'position', [pos2(1) pos2(2) (pos1(3)-0.025) pos2(4)])
set(gca,'FontSize',fontsize)
axis on;
set(gca,'FontSize',fontsize)
title('(a)')
xlabel('time (sec)')
ylabel('v_{j}, j = 1,2,...,10')
%%%%%%
subplot(212)
plot((0:N-1)/Fs, transform)
set(gca,'FontSize',fontsize)
title('(b)')
xlabel('time(sec)')
ylabel('transform on filtered EEG')
axis([0 6 0 max(transform)+0.2])
```