ADAPTIVE ITERATIVE FILTERING METHODS FOR NONLINEAR SIGNAL ANALYSIS AND APPLICATIONS

A Thesis
Presented to
The Academic Faculty

by

Jingfang Liu

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Mathematics

Georgia Institute of Technology
August, 2013

ADAPTIVE ITERATIVE FILTERING METHODS FOR NONLINEAR SIGNAL ANALYSIS AND APPLICATIONS

Approved by:

Professor Hao Min Zhou,
Committee Chair
School of Mathematics
Georgia Institute of Technology

Professor Hao Min Zhou , Advisor
School of Mathematics
Georgia Institute of Technology

Professor Sung Ha Kang
School of Mathematics
Georgia Institute of Technology

Professor Shui-Nee Chow
School of Mathematics
Georgia Institute of Technology

Professor Luca Dieci
School of Mathematics
Georgia Institute of Technology

Professor Edmond Chow
School of Computational Science and Engineering
Georgia Institute of Technology

Date Approved: April 29, 2013

This thesis is lovingly dedicated to

my family, my husband, Weizhe Zhang

for their support, encouragement, and constant love.

ACKNOWLEDGEMENTS

Many people contributed to this dissertation in different ways and I am grateful to all of them. First I would like to express my thanks to my advisor Dr. Hao Min Zhou. I am very appreciative for his advices and continuous support throughout my Ph.D study and research. I would not have written this thesis without discussion with him and getting inspiration from it. It is my fortune to be one of his graduate students and to work under his supervision.

I also thank the other members of my thesis committee: Dr. Shui-Nee Chow for his insightful suggestions on the relations between the decomposition method and the dynamical systems; Dr. Edmond Chow for his generosity of sharing his data with me; Dr. Luca Dieci and Dr. Sung Ha Kang for their distinctive suggestions on the numerical experiments and possible applications.

Furthermore, I am also very grateful to Dr. Antonio Cicone for motivating discussions, for his help on testing the algorithm and for proofreading this Thesis.

I thank the graduate students Yi Huang, Huijun Feng, Xiaolin Wang, Tianjun Ye, Ke Yin, Jun Lu, Yunlong He, Linwei Xin, Ruidong Wang, Ruodu Wang at the School of Mathematics for providing the support and friendship that I needed. I am grateful for the enjoyable time spent with them.

Finally, I would like to thank my family for all their love and encouragement. For my parents who raised me and helped me enthusiastically with my son when I was busy doing my study and research. For my supportive husband Weizhe Zhang who loved me so much and kept patient with me all the time. For my son who brought so much joy and turned into the largest motivation of my graduation. Thank you all.

TABLE OF CONTENTS

LIST OF FIGURES

SUMMARY

Time-frequency analysis for non-linear and non-stationary signals is extraordinarily challenging. To capture the changes in these types of signals, it is necessary for the analysis methods to be local, adaptive and stable. In recent years, decomposition based analysis methods were developed by different researchers to deal with non-linear and non-stationary signals. These methods share the feature that a signal is decomposed into finite number of components on which the time-frequency analysis can be applied. Differences lie in the strategies to extract these components: by iteration or by optimization. However, considering the requirements of being local, adaptive and stable, neither of these decompositions are perfectly satisfactory.

Motivated to find a local, adaptive and stable decomposition of a signal, this thesis presents an Adaptive Local Iterative Filtering (ALIF) algorithm. The adaptivity is obtained having the filter lengths being determined by the signal itself. The locality is ensured by the filter we designed based on a PDE model. The stability of this algorithm is shown and the convergence is proved. Moreover, we also propose a local definition for the instantaneous frequency in order to achieve a completely local analysis for non-linear and non-stationary signals. Examples show that this decomposition really helps in both simulated data analysis and real world application.

CHAPTER I

INTRODUCTION

1.1 Nonlinear Signal Analysis

Data and signal analysis has become increasingly important these days. Finding features and structures of the data is quite challenging especially when the data are generated by a non-linear system and that data are non-stationary. Time-frequency analysis of signals has been studied massively, readers can find relevant information in well-known books such as [20] and [37]. Traditionally, Fourier spectral analysis has been commonly used for signal and data analysis. Another well known approach is based on Wavelet transforms. Both of them are effective and easy to implement. However, there are some limitations. Fourier transform works well when systems are linear and the data are periodic or stationary, it cannot deal with non-stationary signals or data from non-linear systems. The wavelet transform is also a linear analysis tool. Both use predetermined bases and are not designed with effective data-adaptive properties.

To analyze non-linear and non-stationary signals efficiently, it is necessary that the analysis method is local, adaptive and stable. The event in time-frequency analysis should be only determined by the local behaviour of signal not by behaviours in faraway positions. Being adaptive means that the analysis procedure should be data-driven. Any analysis method with a rigid basis lose this property. Being stable means that the analysis results should be consistent for signals under small perturbation.

In the last decade, several decomposition techniques have been proposed to analyze non-linear and non-stationary signals. All these methods share the same approach: first they decompose a signal into simpler components and then apply a

time-frequency analysis to each component separately. The signal decomposition can be achieved in two ways: by iteration or by optimization. For example, Empirical Mode Decomposition [51] and Iterative Filtering method [78] are decompositions by iteration; adaptive data analysis via Sparse Time-Frequency Representation [47] and Synchrosqueezed Wavelet Transforms [22] are decompositions by optimization.

1.2 Empirical Mode Decomposition

The first iterative algorithm, Empirical Mode Decomposition (EMD), was introduced by Huang et al. in [51]. This method aims to iteratively decompose a signal into a finite series of intrinsic mode functions (IMFs) whose instantaneous frequencies are well behaved. Then based on Hilbert transform, Huang et al. compute the instantaneous frequencies for the IMFs to construct the spectrum for the signal [52, 131, 50].

For a given signal $x(t)$, $t \in \mathbb{R}$, its Hilbert transform is

$$H(x)(t) = \frac{1}{\pi}\text{p.v.}\int_{-\infty}^{\infty}\frac{x(\tau)}{t-\tau}d\tau, \tag{1}$$

provided this integral exists as a principal value. Then it is well known that

$$z(t) = x(t) + iH(x)(t) \tag{2}$$

is an analytic function and one can write it as

$$z(t) = x(t) + iH(x)(t) = a(t)e^{i\theta(t)}, \tag{3}$$

where $a(t)$, $\theta(t)$ are both real functions and $a(t)$ represents the amplitude $\theta(t)$ represents the phase of $z(t)$ respectively:

$$\begin{aligned} a(t) &= \sqrt{x^2(t) + H^2(x)(t)}, \\ \theta(t) &= \arctan\left(\frac{H(x)(t)}{x(t)}\right). \end{aligned} \tag{4}$$

The instantaneous frequency $w(t)$ for signal $x(t)$ is defined as

$$w(t) = \frac{d\theta(t)}{dt}. \tag{5}$$

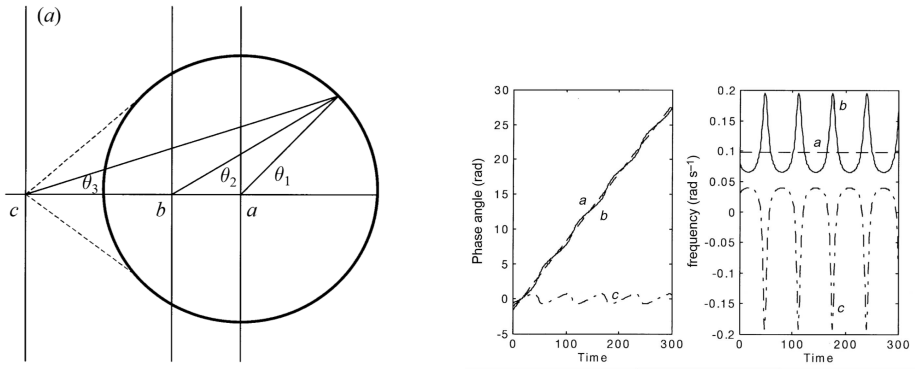
However, such definition of the instantaneous frequency may be controversial, mainly because it may lead to negative frequencies, which are not meaningful in practice. In order to obtain well behaved instantaneous frequency, which means the instantaneous frequency is in a narrow positive interval, Huang imposed some restrictive conditions on the signal. To illustrate these restrictions, consider one simple example. Let

$$x(t) = \sin t. \tag{6}$$

Its Hilbert transform $H(x)(t)$ is $\cos t$. Plot $(x(t), H(x)(t))$ on the phase plane, it is the unit circle shown in Figure 1(a) with origin $(0, 0)$ at point a . The unwrapped phase angle function is a straight line as shown in Figure 1(b) as the dashed line, and the instantaneous frequency, shown in Figure 1(b), is a constant. Let us move the mean of $x(t)$ from 0 to α, so

$$x(t) = \alpha + \sin t. \tag{7}$$

Plot $(x(t), H(x)(t))$ on the phase plane, it is still a circle with radius 1 since $H(x)(t)$ is again $\cos t$, but the center is shifted from $(0, 0)$ to $(\alpha, 0)$ like for instance point b. When $\alpha < 1$, the center $(\alpha, 0)$ is inside the circle. The phase angle is no longer a straight line but keeps increasing. Thus the instantaneous frequency behaves differently from the instantaneous frequency when $\alpha = 0$. However, it is still in a narrow positive interval, which is well behaved. When $\alpha > 1$, the center $(\alpha, 0)$ moves to the outside of the circle for example point c. As a result, the phase angle is not increasing any more. Instead, it is increasing and decreasing alternatively as the dashed dot line. Thus the instantaneous frequency yields negative values, as shown in Figure 1(b), which is meaningless.



(a) (b)

Figure 1: Physical interpretation of instantaneous frequency. [51]. (a) The phase plane for the signal $x(t) = \alpha + \sin t$. a: $\alpha = 0$; b: $\alpha < 1$; c: $\alpha > 1$. (b) The unwrapped phase function of $x(t)$ and the instantaneous frequency computed according to (5).

The simple example provided above illustrates that, for a signal such as the sinusoidal function, the instantaneous frequency is well behaved if the function is restricted to a zero mean. For general signals, in order to make the instantaneous frequency in a narrow positive interval, Huang considered two constraints which are expected to lead to a well behaved instantaneous frequency:

- in the whole data set, the number of extrema and the number of zero crossings must either equal or differ at most by one;
- at any point, the mean value of the envelope defined by the local maxima and the envelope defined by the local minima is zero.

The functions that satisfy these conditions are called intrinsic mode functions (IMFs), which represent the oscillation modes embedded in signals. The sinusoidal functions are IMFs, but IMFs are not limited to sinusoidal functions. Perturbations in both the amplitude and the frequency are allowed. For example, the signals with amplitude modulated and/or frequency modulated are IMFs. What is more important, the IMFs can be non-stationary. One example of IMF is shown in Figure 2.

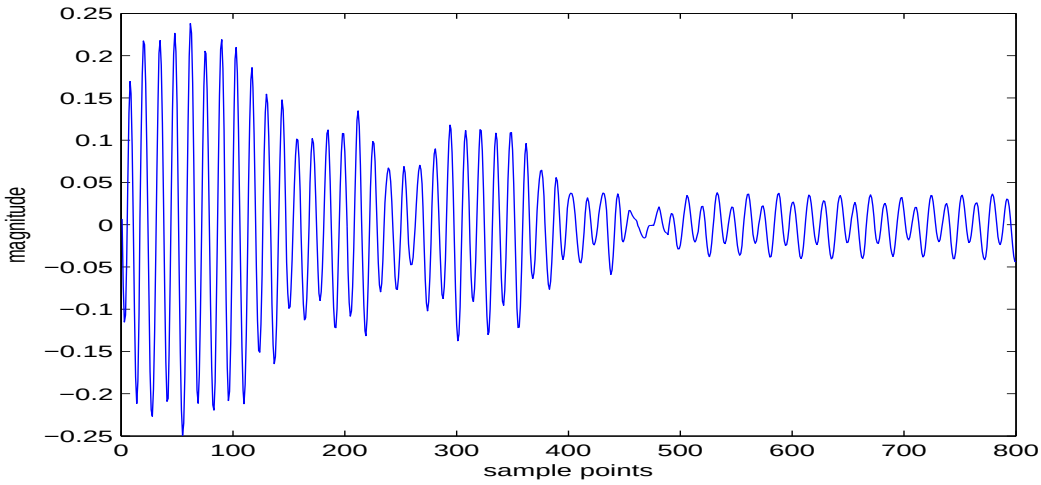


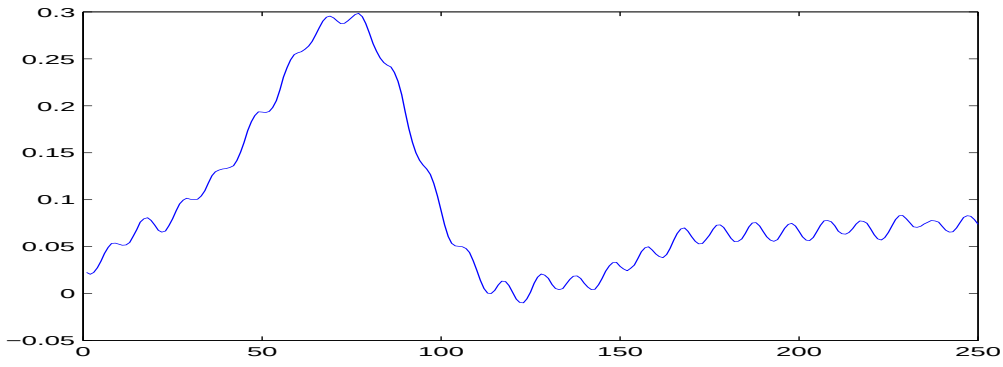
Figure 2: An example of the intrinsic mode function which has the same number of zero crossing points and local extrema and symmetric upper envelope and lower envelope.

The method Huang developed to compute IMFs is called the sifting process. It has an iterative structure, which can be described in the following. Given signal $x(t)$, the method generates an upper envelope as the cubic spline connecting the local maxima of $x(t)$ and a lower envelope as the cubic spline connecting the local minima of $x(t)$. Let $m_1(t)$ be the mean of the upper envelope and the lower envelope. Subtract $m_1(t)$ from $x(t)$, the fluctuation part is derived as

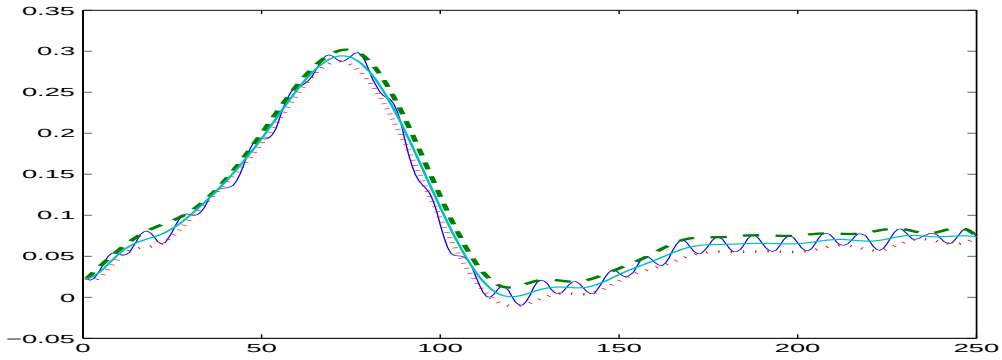
$$h_1(t) = x(t) - m_1(t). \tag{8}$$

The procedure is shown in Figure 3.

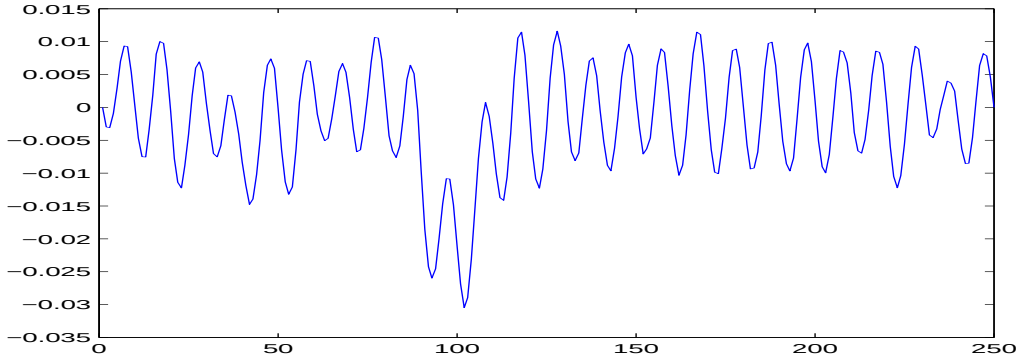
$h_1(t)$ contains oscillations. However, it may not satisfy the constraints of the IMF, i.e. the number of zero crossing points and the number of local extrema differ and the upper and lower envelops have not zero mean. In order to derive an IMF, it is necessary to iterate the steps described above. Treat $h_1(t)$ as the signal, construct its



(a)

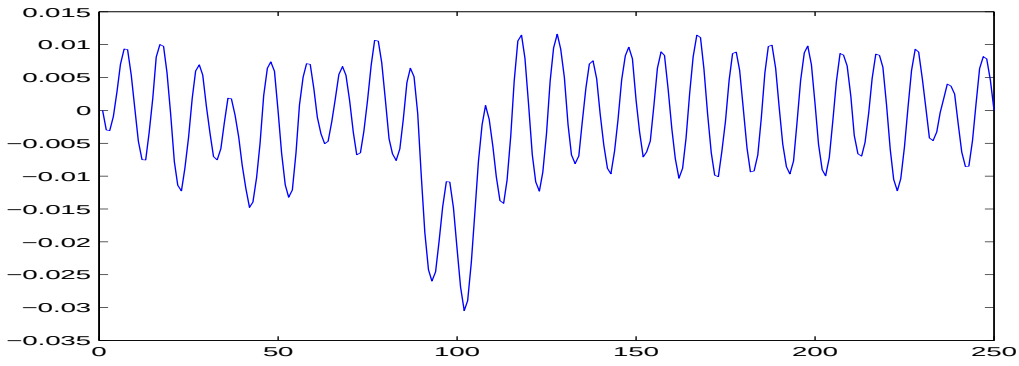


(b)

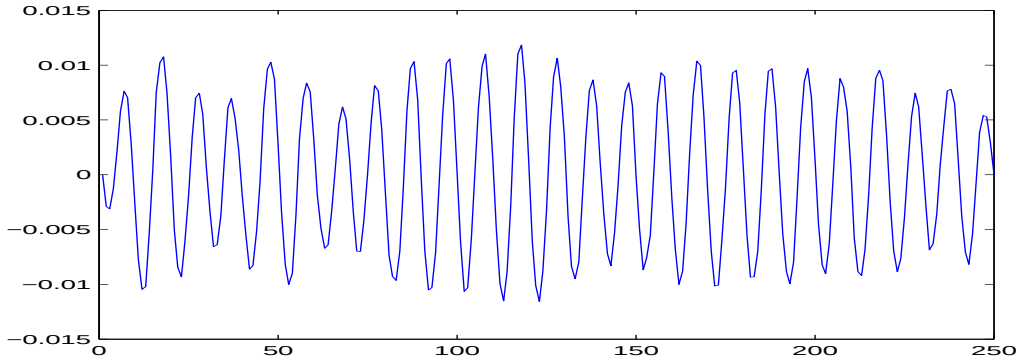


(c)

Figure 3: Illustration of the sifting process. (a) The original signal $x(t)$; (b) The upper and lower envelopes (dot-dashed lines) of $x(t)$ and their mean $m_1(t)$; (c) The difference of $x(t)$ and $m_t(t)$. This is not an IMF since the number of local extrema is greater than the number of zero crossings.



(a)



(b)

Figure 4: effects of the iterated sifting process. (a) After one sifting process as shown in Figure 3(c), it is not an IMF; (b) After eight siftings, the result is improved as an IMF.

upper and lower envelopes, subtract the mean of the envelopes $m_{11}(t)$ from $h_1(t)$:

$$h_1(t) - m_{11}(t) = h_{11}(x). \tag{9}$$

This process is repeated many times

$$h_{1(k-1)}(t) - m_{1k}(t) = h_{1k}(t) \tag{10}$$

until $h_{1k}(t)$ satisfies the two requirements of an IMF as shown in Figure 4.

In the implementation, the stopping criterion is related to the value of SD defined as

$$SD = \frac{\|h_{1(k-1)}(t) - h_{1k}(t)\|_{L^2}}{\|h_{1(k-1)}(t)\|_{L^2}}. \tag{11}$$

If SD is less than a pre-selected threshold, the iteration is stopped and $h_{1k}(t)$ is an IMF. A typical value of SD can be set between 0.2 and 0.3. Let

$$I_1(t) = h_{1k}(t). \tag{12}$$

This is the first IMF extracted from $x(t)$.

Then the sifting process is applied on the remaining signal

$$x(t) - I_1(t) \tag{13}$$

to get the subsequent IMFs. The sifting process is repeated until the remaining signal does not contain any oscillation, i.e. there is at most one extrema in the remaining signal. Denote the remaining signal by $y(t)$. The decomposition of signal $x(t)$ is

$$x(t) = I_1(t) + I_2(t) + ... + I_m(t) + y(t), \tag{14}$$

where m is the number of IMFs in the decomposition.

Since cubic splines are constructed based on the maxima and minima of the signal, the sifting process is local and adaptive. However, the sifting process is not stable since the cubic splines may behave quite differently under perturbations of the same signal. In order to make the sifting process more stable, Huang et al. developed Ensemble Empirical Mode Decomposition (EEMD) in [133]. In EEMD, to get one IMF, numbers of trials of sifting process are applied to different perturbations of the same signal. Taking the mean of the IMFs extracted in different trials, the IMF for the original signal without any perturbation is obtained. Another issue of using cubic splines is that the convergence of the sifting process is hard to deal with.

1.3 Iterative Filtering Algorithm

Recently, some other decomposition techniques have been developed inspired by Huang's EMD. Iterative Filtering (IF) algorithm is one of these methods devised by Wang et al. in [78]. IF makes use of the same algorithm frame as EMD.

Let $\mathcal{L}$ be an operator to get the 'moving average' of a signal x while $\mathcal{S}$ is an operator to take the fluctuation part

$$\mathcal{S}(x) = x - \mathcal{L}(x). \tag{15}$$

The moving average is referred as a replacing function as the mean of the envelopes in the sifting process. Then the first IMF is given by

$$I_1 = \lim_{n\to\infty} \mathcal{S}^n x. \tag{16}$$

Here the limit is reached so that applying $\mathcal{S}$ will not change the signal any more. The subsequent IMFs are obtained one after another by

$$I_k = \lim_{n\to\infty} \mathcal{S}^n(x - I_1 - ... - I_{k-1}). \tag{17}$$

The process stops when

$$y = x - I_1 - I_2 - ... - I_m \tag{18}$$

has at most one local maximum or one local minimum. By this step, the decomposition of $x(t)$ is

$$x = y + \sum_{j=1}^{m} I_j. \tag{19}$$

In the sifting process, the moving average is extracted by taking the average of the upper envelope and the lower envelope. In IF alorithm, the moving average is extracted by taking the convolution of the signal $x(n)$, $n \in \mathbb{Z}$ with a low pass filter $a(k), k = -m, -(m-1), ..., m-1, m$:

$$\mathcal{L}_a(x)(n) = \sum_{k=-m}^{m} x(n+k)a(k). \tag{20}$$

Wang et al. showed the convergence of IF algorithm with toeplitz filters for periodic signals.

Theorem 1. *[78] Let $x(n), n \in \mathbb{Z}$ be a periodic signal, i.e. there exists an $N > 0$ such that $x(n+N) = x(n)$ for all $n \in \mathbb{Z}$. Let $a(k), k = -m, -(m-1), ..., m-1, m$*

and

$$\mathcal{S}(x) = x - \mathcal{L}_a(x). \tag{21}$$

Denote

$$\hat{a}(\xi) = \sum_{k=-m}^{m} a(k) e^{2\pi i k \xi}. \tag{22}$$

Let $N > 2m$. *Then* $\mathcal{S}^n(x)$ *converges for all* $N-$*periodic* $X(n)$ *if and only if for all* $\xi \in \frac{1}{N}\mathbb{Z}$ *we have either* $\hat{a}(\xi) = 0$ *or* $|1 - \hat{a}(\xi)| < 1$. *Assume that* $\lim_{n\to\infty} \mathcal{S}^n(x) = y$, *then*

$$y = \sum_{k\in\Gamma} c_k E_k, \tag{23}$$

where $\Gamma = \{0 \le k < N : \hat{a}(k/N) = 0\}$, E_k *is given by* $E_k(n) = e^{\frac{2\pi i k n}{N}}$ *and*

$$c_k = \frac{1}{N} \sum_{j=0}^{N-1} x(j) e^{-\frac{2\pi i k j}{N}}. \tag{24}$$

The proof of Theorem 1 is presented in [78]. This theorem essentially gaurantees the convergence of obtaining the IMFs when the moving average is computed as the convolution of the signal and a filter with appropriate properties. Moreover, it is convergent to the projection onto the basis E_k, which depends on the chosen filter.

If $a(k)$ is symmetric, $\hat{a}(\xi) = 0$ or $|1 - \hat{a}(\xi)| < 1$ is equivalent to $0 \le \hat{a}(\xi) \le 2$ for all $\xi \in \mathbb{R}$ with $\hat{a}(\xi) \neq 2$ for $\xi \in \mathbb{Q}$. One filter satisfies these properties is the double average filter. So it is used in IF algorithm by Wang et al. The weight of double average filter for length $2m + 1$ is given by

$$a(k) = \frac{m + 1 - |k|}{(m + 1)^2}, \quad k = -m, ..., m. \tag{25}$$

It is clear that the double average filter satisfes the condition

$$0 \le \hat{a}(\xi) \le 1 \tag{26}$$

so that the convergence is guaranteed. Furthermore, the zeros of them are known:

$$\hat{a}_m(\xi) = 0 \quad \text{if and only if} \quad \xi = \frac{k}{m + 1}, 1 \le k \le m + 1. \tag{27}$$

The double average filter with half length 22 is shown in Figure 5.

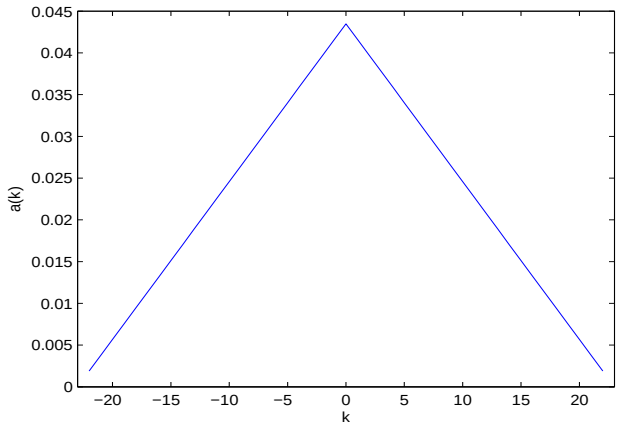


Figure 5: double average filter defined in (25) with $m = 22$

In the implementation of IF algorithm with uniform filter length where the moving average is computed by (20), the half filter length m is set to be

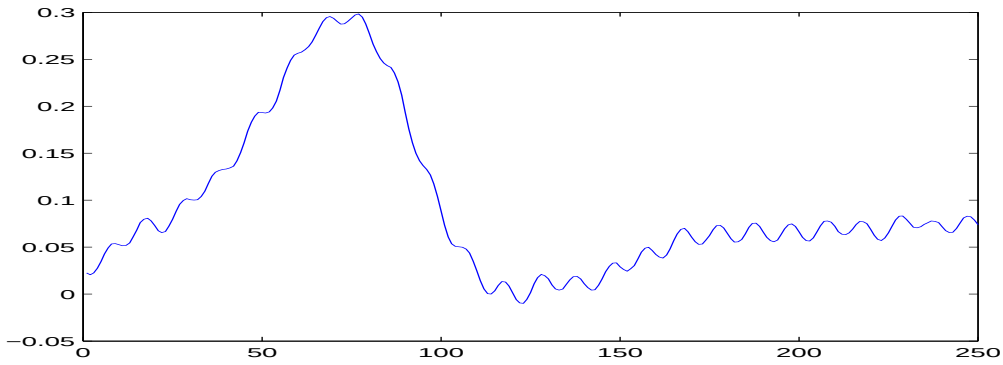
$$m = \left\lfloor \frac{2N}{k} \right\rfloor, \tag{28}$$

where N is the number of sample points of the signal $x(n)$ and k is the total number of local maxima and local minima in $x(n)$. Given the same signal as in the illustration of the sifting process in Figure 3, the effect of one step of the iterative filtering is illustrated in Figure 6. $\mathcal{S}^1(x)$ is shown in Figure 6(c). However, $\mathcal{S}^1(x)$ is not an IMF. Iterate the filtering process as $\mathcal{S}^n(x)$ and the IMF is obtained. In the numerical implementation, it is not necessary to grow n to a large enough number. Instead the stopping criteria of $\mathcal{S}^n(x)$ is also depending on a threshold of SD, which is similar to the sifting process. Let

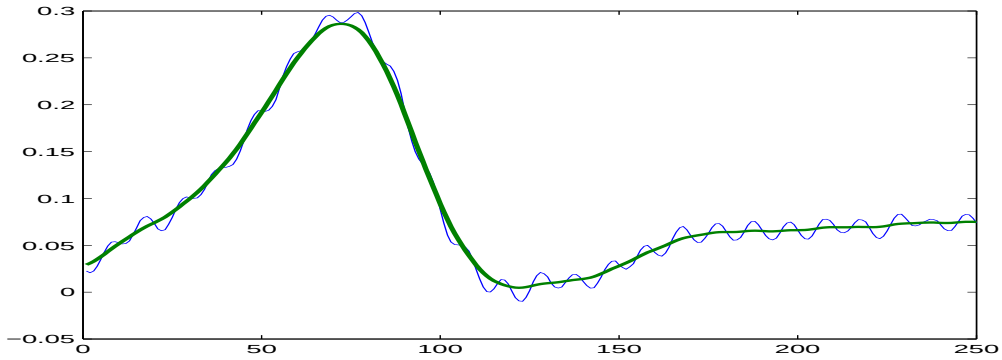
$$I_{k,j}(n) = \mathbb{S}_k^j(x(n) - I_1(n) - ... - I_{k-1}(n)), \tag{29}$$

then SD of iterative filtering is defined as

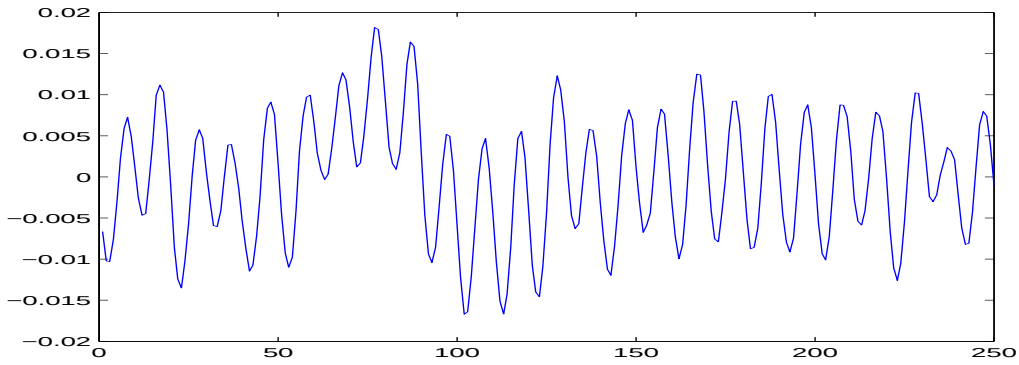
$$SD = \frac{\sum_{n=1}^{N} |I_{k,j}(n) - I_{k,j-1}(n)|^2}{\sum_{n=1}^{N-1} |I_{k,j-1}(n)|^2}. \tag{30}$$



(a)



(b)



(c)

Figure 6: Illustration of one step of Iterative Filtering (IF) algorithm. (a) The original signal $x(t)$; (b) $x(t)$ and its moving average computed by (20) using the double average filter. The thick solid line is the moving average of $x(t)$; (c) The difference of $x(t)$ and its moving average. This is not an IMF since the number of local extrema is greater than the number of zero crossings.

The iteration is stopped when SD is less than a pre-selected threshold. The smaller this threshold is the more iteration steps we take. In the implementation in [78], this threshold is ranging from 0.001 to 0.2. Usually it is set to be around 0.05. The first IMF derived from the given signal $x(t)$ following Figure 6 is shown in Figure 7.

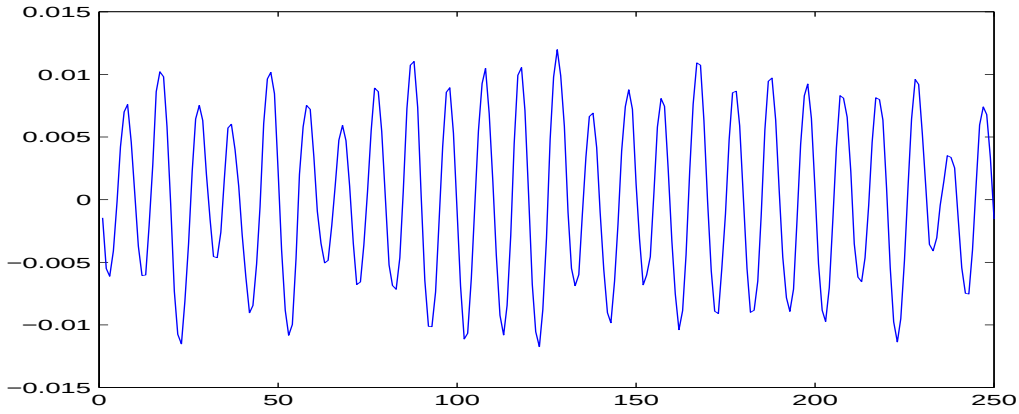


Figure 7: the first IMF derived from $x(t)$ where $x(t)$ is given in Figure 6(a).

The numerical experiment in Figure 7 shows that IF algorithm generates the similar IMF compared with the sifting process which is shown in Figure 4(b). What is more important, IF algorithm is much more stable under perturbation compared with EMD.

For the non-stationary signals, Wang et al. suggested to use filters with different lengths at different positions of the given signal, which we shall call adaptive filters from now on. Then the selection of these adaptive filter lengths plays an essential role in the performance of the decomposition. Wang et al. proposed a method of choosing the filter lengths at different points.

Assume there are k extreme points in the signal $x(n), n \in \mathbb{N}$. Let n_j be the local extrema point of $x(n)$. Then the filter length at $n = n_j$ is

$$l(n_j) = n_{j+2} - n_{j-2}. \tag{31}$$

So the filter lengths at the extreme points are obtained. Construct a cubic spline by connecting the points $(n_j, l(n_j))$. The filter length $l(n)$ for any other n within the sampling size of $x(n)$ is given by the interpolation of this spline. Then the half filter length $m(n)$ is obtained by

$$m(n) = \frac{l(n)}{2}. \tag{32}$$

The moving average $\mathcal{L}(x)$ with the adaptive filter is

$$\mathcal{L}(x)(n) = \sum_{k=-m(n)}^{m(n)} x(n+k)a(k). \tag{33}$$

In this case, $\mathcal{L}(x)$ is not the convolution of the signal and the filter any more. The convergence of IF algorithm with adaptive filters has not been established yet. Recently, by using some high order partial differential equations, Wang et al. developed mode decomposition evolution equations in [121] and [122] which can achieve similar results compared with IF algorithm.

1.4 Sparse Time-Frequency Representation

Inspired by EMD and the development in compressed (compressive) sensing theory, Hou et al. proposed an adaptive data analysis method via sparse time-frequency representation in [47] and [46] based on the observation that many multi-scale data usually have an sparse structure in the time-frequency plane.

Using the multicomponent amplitude modulate and frequency modulate (AM-FM) representation, which has been studied for example in[86] and [126], Hou assumed that each IMF could be written as an AM-FM function. A dictionary of IMFs is given as

$$\mathcal{D} = \{a(t)\cos\theta(t), \quad \theta'(t) \geq 0, \quad a(t) \in V(\theta)\}, \tag{34}$$

where $V(\theta)$ is the collection of all the functions that are smoother than $\cos\theta(t)$:

$$V(\theta) = \{g(t) : g(t) \text{ is smoother than } \cos\theta(t)\}. \tag{35}$$

Then the signal is decomposed over this dictionary by seeking the sparsest representation. The sparest decomposition of signal $f(t)$ is obtained by solving a non-linear optimization problem:

$$\begin{aligned} &\text{Minimize} \quad M, \\ &\text{Subject to} \quad f(t) = \sum_{k=1}^{M} a_k(t)\cos\theta_k(t), \quad a_k(t)\cos\theta_k(t) \in \mathcal{D}, \quad k = 1, ..., M. \end{aligned} \tag{36}$$

After this optimization problem is solved, the instantaneous frequency of the kth IMF is computed by

$$w_k(t) = \theta_k'(t). \tag{37}$$

A decomposition method based on a non-linear TV^3 minimization was developed as a tool to solve (36). Given signal $f(t)$, it is decomposed into its local median $a_0(t)$ and an IMF $a_1(t)\cos\theta_1(t)$ by solving the following optimization problem:

$$\begin{aligned} &\text{Minimize} \quad TV^3(a_0) + TV^3(a_1), \\ &\text{Subject to} \quad a_0(t) + a_1(t)\cos\theta_1(t) = f(t), \quad \theta_1'(t) \geq 0, \end{aligned} \tag{38}$$

where the third order total variation TV^3 is defined as

$$TV^3(g) = \int_R |g^{(4)}(x)|dx. \tag{39}$$

The non-linear minimization problem (38) is solved by a Newton type of iterative method. Numerical experiments in [46] shows that the sparse time-frequency representation generates similar components as EMD and its performance does not rely on the parameters such as the stopping criterion. However, there are two limitations of this approach. One is that the computational cost is high. The other is that it is sensitive to noise and a pre-denoising step by a low pass filter in the Fourier domain is required to remove the noise impact as in [46]. To get rid of both of these two issues, Hou et al. proposed another decomposition method based on the non-linear matching pursuit in [45] which yields a lower computational cost and stability to noise perturbation.

1.5 Synchrosqueezed Wavelet Transforms

Inspired to analyze EMD mathematically, Daubechies et al. proposed a different approach of constructing the IMFs from a signal in [22] based on optimizations. Using the multicomponent AM-FM representation, which is similar to sparse time-frequency representation introduced in Section 1.4, Daubechies consider a class of signals which are the superposition of AM-FM functions:

$$s(t) = \sum_{k=1}^{K} s_k(t), \tag{40}$$

where $s_k(t)$ is an AM-FM function and it can be written as:

$$s_k(t) = A_k(t)\cos(\Phi_k(t)), \quad \text{with} \quad A_k(t), \Phi_k'(t) > 0 \quad \forall t. \tag{41}$$

Locally, $s_k(t)$ can be viewed as a harmonic signal if the change of $A_k(t)$ and $\Phi_k'(t)$ is much slower than the change of $\Phi_k(t)$ itself. So at time t, the amplitude is $A_k(t)$ and the instantaneous frequency is $\Phi_k'(t)$.

Based on the continuous wavelet transform W_s of the signal $s(t)$ defined by

$$W_s(a,b) = \int s(t) a^{-1/2} \overline{\varphi(\frac{t-b}{a})} dt, \tag{42}$$

the synchrosqueezed wavelet transform (SWT) of $s(t)$ is defined as

$$\mathcal{T}_s(\omega, b) = \int_{A(b)} W_s(a,b) a^{-3/2} \delta(\omega(a,b) - \omega) da, \tag{43}$$

where $A(b) = \{a; W_s(a,b) \neq 0\}$ and $a \in A(b)$. It is proved in [22] that for any signal which can be written as a superposition of the AM-FM functions, SWT can decompose it successfully on the time-frequency plane. The adaptive time-frequency decomposition is achieved by solving the following minimization problem:

$$\begin{aligned} \underset{F(t,\omega)}{\operatorname{argmin}} \quad & \int \left| \operatorname{Re}\left[\int F(t,\omega) d\omega - s(t)\right] - s(t) \right|^2 dt \\ & + \mu \int\int \left| \partial_t F(t,\omega) - i\omega F(t,\omega) \right|^2 dt d\omega. \end{aligned} \tag{44}$$

The minimizer of (44) is close to the SWT of the signal $s(t)$.

1.6 Summary Of The Introduction

We introduced several decomposition methods which can perform time-frequency analysis afterwards in the previous sections. Each method has its strong points, but it is hard for each of them to deal with all kinds of signals. In order to develop an algorithm which handles general signals, we first consider some necessary properties of the decomposition.

For analysis results of non-linear and non-stationary signals, all events need to be localized on time and frequency axis in the analysis results. Being local means the frequency on one position should be affected only by local information while not be affected by any information from faraway positions. This is a basic requirement of the time-frequency analysis, which the decomposition method should also achieve.

Another necessary property is that the analysis method should be adaptive so that it can deal with all kinds of behaviours appearing in signals. The stationary signals are relatively easy to handle. However, there exist some non-stationary signals with sudden changes in the amplitude or (and) the frequency. The analysis results are reliable only if these transient behaviours are captured. It is necessary that the analysis method is adaptive so that it can catch possible transient behaviours as well as stationary behaviours if there are no sudden changes.

Besides locality and adaptivity, it is also necessary that the analysis method to be stable. Consider two signals which are perturbation of each other, it is natural that the decompositions of these two signals yield consistent components. By consistent components, we mean that the significant components showing up in the decomposition for one signal should also appear in the decomposition for the other signal. Consistent components can be achieved only when the decomposition is stable.

With the purpose of developing an adaptive, local, stable decomposition method, we first build a mathematical foundation of IF algorithm. Based on a splitting into an inner iteration and an outer iteration, where the inner iteration is intended to

extract a signal IMF, while the outer iteration extract all the IMFs from a signal, we present the convergence theorem of the inner iteration with uniform filters as well as the convergence theorems of both the inner iteration and the outer iteration with non-uniform filters.

we also present a new method, called Adaptive Local Iterative Filtering (ALIF) algorithm, to generalize the iterative filtering technique for general oscillatory signals with non-uniform filters. ALIF algorithm follows the same algorithm frame of EMD and IF. There are two main aspects in ALIF that are different from the existing IF algorithms. One is that we use Fokker -Planck equation, a second order partial differential equation (PDE), to construct smooth low pass filters which have compact support. The other is that we adapt the filter length according to an initial value problem for an ordinary differential equation (ODE) defined by the signal. The adaptation is highly non-linear and the data driven, yet, it ensures the convergence in the decomposition.

Moreover, we propose an alternative definition for the instantaneous frequency. In the existing instantaneous frequency analysis, Hilbert transform is used to build analytical signals. On the other hand, Hilbert transform is a global transform, which is not ideal to handle signals with transient features. To localize the analysis, we define the instantaneous frequency of an IMF, obtained by ALIF algorithm, as the rotation speed calculated by the normalized IMF and its derivative. We show that such definition for instantaneous frequency can better capture the frequency changes in non-linear signals.

The rest of this thesis is organized as follows: in Chapter 2, we present the mathematical fundamental of IF algorithm including the convergence Theorems for both uniform filters and adaptive filters for general signals; in Chapter 3, we develop ALIF algorithm based on the convergence Theorems and a PDE model; in Chapter 4, we propose a new local definition and computation of the instantaneous frequency; in

Chapter 5, we apply ALIF algorithm to a real world problem in the area of gas detection; we also apply ALIF algorithm to the solutions of several second order ordinary differential equations (ODEs) and the results provide a new understanding of the decomposition methods such as ALIF algorithm, EMD or IF algorithm.

CHAPTER II

FUNDAMENTALS OF IF ALGORITHM

In this chapter, we show the convergence of IF algorithm under proper assumptions. Before we explore the details, we present IF algorithm in the format of two layers of loops which are given as Algorithm 1 and Algorithm 2. For the convenience of later analysis, we call Algorithm 1 as the inner iteration and Algorithm 2 as the outer iteration of IF algorithm.

Algorithm 1 imf = get-one-IMF(f)

compute l // l is the length for filter mask
while the stopping criteria is not satisfied **do**
  get the filter w with length l
  $g = f - \int_{-l/2}^{l/2} f(x+t)w(t)dt$
  $f = f - g$
  compute l for this new f
end while
$imf = x$

Algorithm 2 IMF = get-all-IMFs(f)

IMF=$\emptyset$
while the number of extrema ≥ 2 **do**
  imf = get_one_IMF(x)
  IMF = IMF$\cup\{imf\}$
  $x = x - imf$
end while
IMF = IMF$\cup\{x\}$

The rest of this chapter is organized as follows: in Section 2.1, we present the convergence theorem of the inner iteration with uniform filters. there are certain requirements for the uniform filters based on the convergence theorem, we demonstrate that these requirements can be simply satisfied. In Section 2.2, we present the convergence theorems for both the inner iteration and the outer iteration with adaptive

filters. There are several constraints for the adaptive filters, which can be either satisfied by changing the filter length according to the signal during IF algorithm or be checked after each step of the iteration.

2.1 The Convergence Of The Inner Iteration With Uniform Filters

Let $f(x), x \in \mathbb{R}$ be a continuous signal. For any filter $w(x), x \in [-l, l]$ with compact support, the moving average of $f(x)$ using uniform filter $w(x)$ is

$$ave(x) = \int_{-l}^{l} f(x+t)w(t)dt. \tag{45}$$

Let

$$\mathcal{L}(f)(x) := ave(x) = \int_{-l}^{l} f(x+t)w(t)dt,$$

$$\mathcal{S}(f) := f - \mathcal{L}(f) = (I - \mathcal{L})f.$$

Then after one step of the inner iteration of IF algorithm, the signal is

$$\mathcal{S}(f) = (I - \mathcal{L})f; \tag{46}$$

after n steps of the inner iteration of IF algorithm, the signal is

$$\mathcal{S}^n(f) = (I - \mathcal{L})^n f. \tag{47}$$

If $\{\mathcal{S}^n(f)\}$ is convergent, the first IMF I_1 of $f(x)$ is

$$I_1 = \lim_{n\to\infty} \mathcal{S}^n(f). \tag{48}$$

It is proved in [78] that $\{\mathcal{S}^n(f)\}$ is convergent when f is a periodic signal. In this section, we discuss the convergence of the series $\{\mathcal{S}^n(f)\}$ for L^2 signals. Before getting to the convergence theorem, we need some preliminary analysis, which takes the symmetric property of the filter and the Fourier transform of $\mathcal{S}^n(f)$ into account.

Let the filter $w(x)$ be symmetric, i.e. $w(x) = w(-x), x \in [-l, l]$. Then the moving average of $f(x)$ computed by (45) is the convolution of $f(x)$ and $w(x)$:

$$\begin{aligned} ave(x) &= \int_{-l}^{l} f(x+t)w(t)dt = \int_{-l}^{l} f(x-t)w(t)dt \\ &= \int_{-\infty}^{\infty} f(x-t)w(t)dt = (f * w)(x). \end{aligned} \tag{49}$$

Since $\int_{-l}^{l} w(t)dt = 1$, $w(x) \in L^1(\mathbb{R})$. In addition, let $w(x)$ be a continuous function, then $w(x) \in L^p(\mathbb{R}), 0 < p < \infty$. So $w(x) \in L^2(\mathbb{R})$. The Fourier transform of $w(x)$ is defined by

$$\mathcal{F}(w)(\xi) = \int_{-\infty}^{\infty} w(x)e^{-2\pi i x\xi}dx, \xi \in \mathbb{R}. \tag{50}$$

If signal $f(x) \in L^2(R)$, the Fourier transform of $f(x)$ is

$$\mathcal{F}(f)(\xi) = \int_{-\infty}^{\infty} f(x)e^{-2\pi i x\xi}dx, \xi \in \mathbb{R}. \tag{51}$$

By the convolution theorem of Fourier transform, we have the Fourier transform of $ave(x)$ as:

$$\mathcal{F}(ave)(\xi) = \mathcal{F}(f)(\xi)\mathcal{F}(w)(\xi), \xi \in \mathbb{R}. \tag{52}$$

Thus

$$\begin{aligned} \mathcal{F}(\mathcal{L}f)(\xi) &= \mathcal{F}(f)(\xi)\mathcal{F}(w)(\xi), \quad \xi \in \mathbb{R}, \\ \mathcal{F}(\mathcal{S}^n(f))(\xi) &= \mathcal{F}((I-\mathcal{L})^n f)(\xi) = [1-\mathcal{F}(w)(\xi)]^n \mathcal{F}(f)(\xi), \quad \xi \in \mathbb{R}. \end{aligned} \tag{53}$$

Based on these preliminary analysis, we propose the convergence theorem of the series $\{\mathcal{S}^n(f)\}$.

Theorem 2. *Let $w(x), x \in [-l, l]$ be continuous and symmetric and let $f(x) \in L^2(\mathbb{R})$. If $|1 - \mathcal{F}(w)(\xi)| < 1$ or $\mathcal{F}(w)(\xi) = 0$, then $\{\mathcal{S}^n(f)\}$ converges and*

$$\lim_{n\to\infty} \mathcal{S}^n(f)(x) = \int_{-\infty}^{\infty} \mathcal{F}(f)(\xi)\chi_{\{\mathcal{F}(w)(\xi)=0\}} e^{2\pi i\xi x}d\xi. \tag{54}$$

Proof. $f(x) \in L^2(\mathbb{R})$, thus

$$\int_{-\infty}^{\infty} |\mathcal{F}(f)(\xi)|^2 d\xi = \int_{-\infty}^{\infty} |f(x)|^2 dx < \infty. \tag{55}$$

$|1-\mathcal{F}(w)(\xi)|<1$ or $\mathcal{F}(w)(\xi)=0$, thus

$$|1-\mathcal{F}(w)(\xi)|<1 \text{ or } 1-\mathcal{F}(w)(\xi)=1. \tag{56}$$

Then

$$|\mathcal{F}(\mathcal{S}^n(f))(\xi)| = |[1-\mathcal{F}(w)(\xi)]^n\mathcal{F}(f)(\xi)| = |1-\mathcal{F}(w)(\xi)|^n|\mathcal{F}(f)(\xi)|$$
$$\begin{cases} <|\mathcal{F}(f)(\xi)| & \text{if} \quad |1-\mathcal{F}(w)(\xi)|<1, \\ =|\mathcal{F}(f)(\xi)| & \text{if} \quad 1-\mathcal{F}(w)(\xi)=1, \end{cases}$$

and $\{\mathcal{F}(\mathcal{S}^n(f))\}$ is convergent as $n\to\infty$:

$$\lim_{n\to\infty}|\mathcal{F}(\mathcal{S}^n(f))| = \begin{cases} 0 & \text{if} \quad |1-\mathcal{F}(w)(\xi)|<1, \\ |\mathcal{F}(f)(\xi)| & \text{if} \quad 1-\mathcal{F}(w)(\xi)=1. \end{cases}$$

So $\{\mathcal{S}^n(f)\}$ is also convergent and

$$\lim_{n\to\infty}\mathcal{S}^n(f)(x) = \int_{-\infty}^{\infty}\mathcal{F}(f)(\xi)\chi_{\{\mathcal{F}(w)(\xi)=0\}}e^{2\pi i\xi x}d\xi.$$

□

By Theorem 2, to ensure the convergence of IF algorithm with uniform filters, the essential requirement of filter $w(x)$ is that

$$|1-\mathcal{F}(w)(\xi)|<1 \quad \text{or} \quad \mathcal{F}(w)(\xi)=0. \tag{57}$$

This requirement is not unrealistic. For example, the double average filter defined in (25) satisfies this requirement and $\mathcal{F}(a)(\xi)=0$ when $\xi=\frac{k}{m+1}, 1\le k\le m+1$. Moreover, the double average filter is not the only choice. The filters which satisfy the constraint $|1-\mathcal{F}(w)(\xi)|<1$ or $\mathcal{F}(w)(\xi)=0$ are easy to obtain. Let us demonstrate how to obtain such a filter based on the symmetric property.

For symmetric filters $w(x)$, $\mathcal{F}(w)(\xi)$ is real:

$$\begin{aligned}\mathcal{F}(w)(\xi) &= \int_{-\infty}^{\infty} w(x)e^{-2\pi ix\xi}dx \\ &= \int_{-\infty}^{\infty} w(x)\cos(-2\pi x\xi)dx + i\int_{-\infty}^{\infty} w(x)\sin(-2\pi x\xi)dx \\ &= \int_{-\infty}^{\infty} w(x)\cos(-2\pi x\xi)dx \\ &\in \mathbb{R}.\end{aligned} \tag{58}$$

In addition, since $\int_{-l}^{l} w(x)dx = 1$, we have

$$\begin{aligned}|\mathcal{F}(w)(\xi)| = |\int_{-\infty}^{\infty} w(x)\cos(-2\pi ix\xi)dx| &\leq \int_{-\infty}^{\infty} |w(x)\cos(-2\pi x\xi)|dx \\ < \int_{-\infty}^{\infty} |w(x)|dx &= \int_{-l}^{l} w(x)dx = 1.\end{aligned} \tag{59}$$

So for the symmetric filter $w(x)$, we have following property automatically:

$$-1 < \mathcal{F}(w)(\xi) < 1, \xi \in \mathbb{R}. \tag{60}$$

To obtain the filter $w(x), x \in [-l, l]$ such that $0 \leq \mathcal{F}(w)(\xi) < 1$, we make use of another symmetric filter $u(x), x \in [-l/2, l/2]$. Let $\mathcal{F}(u)(\xi)$ be the Fourier transform of $u(x)$, we have $-1 < \mathcal{F}(u)(\xi) < 1, \xi \in \mathbb{R}$. Let $w(x)$ be the convolution of $u(x)$ with itself

$$w(x) = u(x) * u(x). \tag{61}$$

Then $w(x)$ has the compact support $[-l, l]$ as well as the desired property: $0 < \mathcal{F}(w)(\xi) < 1$ or $\mathcal{F}(w)(\xi) = 0$, $\xi \in \mathbb{R}$.

2.2 The Convergence Of IF Algorithm With Adaptive Filters

In Theorem 2, $\chi_{\{\hat{w}(\xi)=0\}}$ is determined by the filter $w(x)$ itself not by the functions $\mathcal{S}^n(f), n \in \mathbb{N}$. If we apply IF algorithm with the same uniform filter on two different signals, $\chi_{\{\hat{m}(\xi)=0\}}$ is the same. This observation shows that using uniform filters in IF algorithm does not extract IMFs completely in an adaptive way. To get rid of this

issue, adaptive filters are explored where the filter lengths are different from point to point. In the computation of the moving average of $f(x)$, instead of using a fixed filter $w(x)$ with support $[-l, l]$, we make use of the function $w(x,t), t \in [-l(x), l(x)]$. For different x, the filter has different support.

Let $f_1(x) = f(x)$. Then after one step of the inner iteration of IF algorithm with adaptive filters, the signal is

$$f_2(x) = f_1(x) - \int_{-l_1(x)}^{l_1(x)} f_1(x+t) w_1(x,t) dt, \tag{62}$$

where $w_1(x,t), t \in [-l_1(x), l_1(x)]$ is the filter for signal $f_1(x)$ at point x with length $2l_1(x)$. Similarly, after n steps of the inner iteration of IF algorithm with adaptive filters, the signal is

$$f_{n+1}(x) = f_n(x) - \int_{-l_n(x)}^{l_n(x)} f_n(x+t) w_n(x,t) dt, \tag{63}$$

where $w_n(x,t), t \in [-l_n(x), l_n(x)]$ is the filter for signal $f_n(x)$ at point x with length $2l_n(x)$. If $\{f_n(x)\}$ is convergent, the first IMF $I_1(x)$ of $f(x)$ is

$$I_1(x) = \lim_{n \to \infty} f_n(x). \tag{64}$$

There is an equivalent formulation of (63) which is written in the following:

$$f_{n+1}(x) = f_n(x) - \int_{-L}^{L} f_n(x + g_n(x,y)) W(y) dy, \tag{65}$$

where $g_n(x,y) : [-L, L] \to [-l_n(x), l_n(x)]$ is a scaling function and $W(y), y \in [-L, L]$ is a fixed filter with length $2L$. We propose the convergence theorem of the inner iteration based on (63) and the convergence theorem of the outer iteration based on the equivalent formulation (65).

2.2.1 Rescaling The Horizontal X Axis

In (63) and (65), we propose two different formulas of updating steps in the inner iteration. (63) can be viewed as that the signal is fixed and that the filters are adjusted

to match up with the signal while (65) can be viewed as that the filter is fixed and that the signal is adjusted from point to point to match up with the filter. We can perform a scaling of the x axis on a fixed filter to get a filter with desired length. We can also perform a scaling of the x axis on the signal to get the same result as we rescale the filter. We demonstrate the underneath connection of (63) and (65) by the following example.

Let $W(t), t \in [-L, L]$ be a filter with fixed length. For any filter length $l(x)$, Let

$$y = \frac{l(x)}{L}t, \tag{66}$$

then the filter $w(y, l(x)), y \in [-l(x), l(x)]$ can be obtained by

$$w(y, l(x)) = \frac{L}{l(x)}W(\frac{L}{l(x)}y). \tag{67}$$

So filters with different lengths are obtained by changing the variable from t to y followed by multiplying proper constants. The moving average of $f(x)$ using $w(y, l(x))$ is

$$ave(x) = \int_{-l(x)}^{l(x)} f(x+y)w(y, l(x))dy. \tag{68}$$

By (67), the moving average of $f(x)$ is

$$ave(x) = \int_{-l(x)}^{l(x)} f(x+y)\frac{L}{l(x)}W(\frac{L}{l(x)}y)dy. \tag{69}$$

Since $y = \frac{l(x)}{L}t$, the moving average of $f(x)$ can also be written as

$$ave(x) = \int_{-L}^{L} f(x+\frac{l(x)}{L}t)W(t)dt. \tag{70}$$

In (67), we use linear scaling from the fixed filter $W(t)$ to general filter $w(y, l(x))$ with length $2l(x)$. In fact, other type of scaling method can also be used here as long as it keeps the filter $w(y, l(x))$ continuous and symmetric.

Let $g(x, t)$ be a scaling function

$$g(x, t) : R \times [-L, L] \to R \times [-l(x), l(x)], \tag{71}$$

where $[-L, L]$ is the support of the fixed filter $W(y)$ and $l(x)$ is the half mask length for the center x when computing the moving average of $f(x)$. Assume $g(x, -t) = -g(x, t)$, since $W(-t) = W(t), t \in [-L, L]$, thus $w(y, l(x)) = W(g(x, y)), y \in [-l(x), l(x)]$ is symmetric. Assume $g(x, t)$ is continuous in t, since $W(t)$ is continuous, thus $w(y, l(x)), y \in [-l(x), l(x)]$ is continuous in y. Using the scaling function $g(x, t)$, the moving average of $f(x)$ is

$$ave(x) = \int_{-L}^{L} f(x + g(x, t))W(t)dt. \tag{72}$$

(72) has the similar form of the moving average of $f(x)$ using uniform filter $W(t), t \in [-L, L]$ which is

$$ave(x) = \int_{-L}^{L} f(x + t)W(t)dt, \tag{73}$$

except the function $f(x + g(x, t))$ with scaling. Since $g(x, t)$ is not in the same range for different x, (72) is not the convolution of a signal and a filter. In this case, we cannot make use of the same analysis as we did for IF algorithm with uniform filters. Instead, we develop new techniques to analyze the convergence of IF algorithm with adaptive filters.

2.2.2 The Convergence Of The Inner Iteration

If the limit of $f_n(x)$ exits, then the moving average of $f_n(x)$ converges to zero, i.e. $\lim_{n\to\infty} \int_{-l_n(x)}^{l_n(x)} f_n(x + t)w_n(x, t)dt = 0$. In this case, $\lim_{n\to\infty} f_n(x)$ cannot be zero since $\lim_{n\to\infty} f_n(x) = 0$ corresponds to a zero component derived from the signal $f(x)$, which is not meaningful. The inner iteration is expected to end up with an IMF, which is an oscillatory function with the moving average as zero. Based on this analysis, we present the convergent theorem of the inner iteration of IF algorithm with adaptive filters.

Theorem 3. *Let $f(x), x \in \mathbb{R}$ be continuous and $f(x) \in L^\infty(\mathbb{R})$. Let*

$$\epsilon_n = \frac{\left\| \int_{-l_{n+1}(x)}^{l_{n+1}(x)} f_{n+1}(x+t) w_{n+1}(x,t) dt \right\|_{L^\infty}}{\left\| \int_{-l_n(x)}^{l_n(x)} f_n(x+t) w_n(x,t) dt \right\|_{L^\infty}}, \tag{74}$$

$$\delta_n = \frac{\left\| \int_{-l_{n+1}(x)}^{l_{n+1}(x)} |f_{n+1}(x+t)| w_{n+1}(x,t) dt \right\|_{L^\infty}}{\left\| \int_{-l_n(x)}^{l_n(x)} |f_n(x+t)| w_n(x,t) dt \right\|_{L^\infty}}. \tag{75}$$

If

$$\prod_{i=1}^{n} \epsilon_i \to 0, \quad \prod_{i=1}^{n} \delta_i \to c > 0, \quad \text{as } n \to \infty, \tag{76}$$

then $\{f_n(x)\}$ converges to an IMF.

Proof. Since $\prod_{i=1}^{n} \epsilon_n \to 0$ as $n \to \infty$ thus $\forall \epsilon > 0$, $\exists\ N$ such that when $n > N$ $\prod_{i=1}^{n} \epsilon_n < \epsilon$. By (74),

$$\prod_{i=1}^{n} \epsilon_i = \frac{\left\| \int_{-l_{n+1}(x)}^{l_{n+1}(x)} f_{n+1}(x+t) w_{n+1}(t) dt \right\|_{L^\infty}}{\left\| \int_{-l_1(x)}^{l_1(x)} f_1(x+t) w_1(t) dt \right\|_{L^\infty}} < \epsilon. \tag{77}$$

Then

$$\left\| \int_{-l_{n+1}(x)}^{l_{n+1}(x)} f_{n+1}(x+t) w_{n+1}(t) dt \right\|_{L^\infty} < \epsilon \left\| \int_{-l_1(x)}^{l_1(x)} f_1(x+t) w_1(t) dt \right\|_{L^\infty}. \tag{78}$$

So we have

$$\left\{ \left\| \int_{-l_{n+1}(x)}^{l_{n+1}(x)} f_{n+1}(x+t) w_{n+1}(t) dt \right\|_{L^\infty} \right\} \to 0 \quad \text{as} \quad n \to \infty. \tag{79}$$

As a result,

$$\int_{-l_{n+1}(x)}^{l_{n+1}(x)} f_{n+1}(x+t) w_{n+1}(t) dt \to 0 \quad \text{as} \quad n \to \infty. \tag{80}$$

So $\{f_n(x)\}$ is convergent. Let $F(x)$ denote the limit of $\{f_n(x)\}$ as following:

$$F(x) := \lim_{n \to \infty} f_n(x). \tag{81}$$

We consider the absolute area bounded by the curve $f_n(x)$ and the x axis. The limit of this area is a positive value as shown below:

$$\left\| \int_{-l_{n+1}(x)}^{l_{n+1}(x)} |f_{n+1}(x+t)| w_{n+1}(t) dt \right\|_{L^\infty} = \prod_{i=1}^{n} \delta_i \left\| \int_{-l_1(x)}^{l_1(x)} |f_1(x+t)| w_1(t) dt \right\|_{L^\infty} \tag{82}$$

and

$$\begin{aligned}
&\left\| \int_{-l(x)}^{l(x)} |F(x+t)|w(t)dt \right\|_{L^\infty} \\
=&\left(\lim_{n\to\infty} \prod_{i=1}^{n} \delta_i \right) \left\| \int_{-l_1(x)}^{l_1(x)} |f_1(x+t)|w_1(t)dt \right\|_{L^\infty} \\
=&c \left\| \int_{-l_1(x)}^{l_1(x)} |f_1(x+t)|w_1(t)dt \right\|_{L^\infty} > 0.
\end{aligned} \tag{83}$$

By (80), the moving average of $F(x)$ is zero; by (83), $F(x)$ itself is non zero. So we get that $F(x)$ is an IMF. □

Note that in order to satisfy the convergence condition in Theorem 2, it is not necessary $\epsilon_n < 1$ for each $n \in \mathbb{N}$. ϵ_n are allowed to be greater than 1 for some $n \in \mathbb{N}$. This is consistent with what we observe in the implementation of IF algorithm. There are signals whose L^∞ norm grow at the beginning of the inner iteration but eventually converge.

We should also remark that δ_n are not necessary to be greater than 1 to be convergent. For example if $\delta_n = 1 - \frac{1}{2^n}$, then

$$\prod_{n=1}^{\infty} \delta_n > 0. \tag{84}$$

Proof. Consider the function values of $\log(1-x)$ and $-2x$. We get that

$$\log(1-x) \geq -2x \quad \text{when} \quad x \leq \frac{1}{2}. \tag{85}$$

Then

$$\sum_{n=1}^{\infty} \log(1 - \frac{1}{2^n}) \geq -2 \sum_{n=1}^{\infty} \frac{1}{2^n} = -2. \tag{86}$$

So we can get

$$\prod_{n=1}^{\infty} (1 - \frac{1}{2^n}) = \exp\left\{ \sum_{n=1}^{\infty} \log(1 - \frac{1}{2^n}) \right\} \geq e^{-2} > 0. \tag{87}$$

□

This is only one special example of workable δ_n. There are actually many more cases of such δ_n. In IF algorithm, it is easy to check the values of δ_n or ϵ_n after each step in the inner iteration.

We consider the interpretion of the constraints (74) and (75). Let

$$M = \left\| \int_{-l_n(x)}^{l_n(x)} f_n(x+t) w_n(t) dt \right\|_{L^\infty}, \tag{88}$$

and M is reached at point x_0. Since

$$\left\| \int_{-l_{n+1}(x)}^{l_{n+1}(x)} f_{n+1}(x+t) w_{n+1}(t) dt \right\|_{L^\infty} = \epsilon_n \left\| \int_{-l_n(x)}^{l_n(x)} f_n(x+t) w_n(t) dt \right\|_{L^\infty}, \tag{89}$$

if $\epsilon_n < 1$, there exists a positive $\epsilon_0 < 1$ such that

$$\left\| \int_{-l_{n+1}(x)}^{l_{n+1}(x)} f_{n+1}(x+t) w_{n+1}(t) dt \right\|_{L^\infty} \leq (1-\epsilon_0) \left\| \int_{-l_n(x)}^{l_n(x)} f_n(x+t) w_n(t) dt \right\|_{L^\infty}. \tag{90}$$

At point x_0, it is equivalent to

$$\begin{aligned} & -(1-\epsilon_0)M \\ \leq & \int_{-l_{n+1}(x_0)}^{l_{n+1}(x_0)} \int_{-l_n(x_0+t)}^{l_n(x_0+t)} f_n(x_0+s+t) w_n(s) w_{n+1}(t) ds dt \\ & - \int_{-l_{n+1}(x_0)}^{l_{n+1}(x_0)} f_n(x_0+t) w_{n+1}(t) dt \\ \leq & (1-\epsilon_0)M. \end{aligned} \tag{91}$$

If $\int_{-l_n(x_0)}^{l_n(x_0)} f_n(x_0+t) w_n(t) dt > 0$ and $w_n(t)$ is close to $w_{n+1}(t)$, we have

$$\int_{-l_{n+1}(x_0)}^{l_{n+1}(x_0)} f_n(x_0+t) w_{n+1}(t) dt \approx M, \tag{92}$$

and

$$\begin{aligned} & \int_{-l_{n+1}(x_0)}^{l_{n+1}(x_0)} f_n(x_0+t) w_{n+1}(t) dt - (1-\epsilon_0)M \\ \leq & \int_{-l_{n+1}(x_0)}^{l_{n+1}(x_0)} \int_{-l_n(x_0+t)}^{l_n(x_0+t)} f_n(x_0+s+t) w_n(s) w_{n+1}(t) ds dt \\ \leq & \int_{-l_{n+1}(x_0)}^{l_{n+1}(x_0)} f_n(x_0+t) w_{n+1}(t) dt + (1-\epsilon_0)M. \end{aligned} \tag{93}$$

The right inequality is easy to be satisfied since the right part is close to $2M$ and the middle part is the moving average of a function with maximal value M. Consider the left inequality and let

$$v(t) = \int_{-l_n(x)}^{l_n(x)} w_n(s+t) w_{n+1}(t) ds. \tag{94}$$

Then the left inequality can be written as

$$\int_{-l_{n+1}(x_0)}^{l_{n+1}(x_0)} f_n(x_0+t) w_{n+1}(t) dt - (1-\epsilon_0)M \leq \int_{-l_n(x)-l_{n+1}(x)}^{l_n(x)+l_{n+1}(x)} f_n(x+t) v_n(t) dt. \tag{95}$$

When $w_{n+1}(x)$ is close to $w_n(x)$, the left side of (95) is close to $\epsilon_0 M$. When the weight of the filter is concentrated in the middle, $v_n(t)$ decays faster than $w_n(t)$ in the interval $[-l_n(x), l_n(x)]$. So it is highly possible that (95) holds. The intuition of the inner convergence requirements is that the filter weight should be concentrated in the middle and two consecutive filters should be close. These conditions are easy to be satisfied. What is more important is the convergence of the outer iteration, which we explore in the next subsection.

2.2.3 The Convergence Of The Outer Iteration

IF algorithm stops when the remaining signal has a sufficiently small number of extrema. So the stopping criterion is by checking the number of extrema in the remaining signal. The convergence theorem of the outer iteration we propose is based a control of the number of extrema in the remaining signal. We first present the convergence theorem and then some lemma and corollaries which we shall use to prove the theorem.

Let function $f(x), x \in \mathbb{R}$ be continuous and differentiable and $f(x)$ has a finite number of extreme points in any compact interval. So $f(x)$ has at most countable extreme points. Let $x_i, i \in \mathbb{N}$ be the extreme points of $f(x)$. Assume that $f(x)$ is strictly monotone in $[x_i, x_{i+1}], i \in \mathbb{N}$. We make use of the update step (65) as the inner iteration. Here is the main theorem in this subsection.

Theorem 4. *Let $f(x), x \in \mathbb{R}$ be a function with the described properties. In (65), if $g_n(x,y) = l_n(x)h(y)$ and*

$$\begin{aligned} c_n^{(1)}(x) + l_n'(x)c_n^{(2)}(x) > 0 \quad \textit{when} \quad f_n'(x) > 0, \\ c_n^{(1)}(x) + l_n'(x)c_n^{(2)}(x) < 0 \quad \textit{when} \quad f_n'(x) < 0, \end{aligned} \tag{96}$$

for every $n \in \mathbb{Z}$ where $c_n^{(1)}(x)$ and $c_n^{(2)}(x)$ are defined as

$$\begin{aligned} c_n^{(1)}(x) &= \int_{-L}^{L} (f_n'(x) - f_n'(g_n(x,y)+x))W(y)dy, \\ c_n^{(2)}(x) &= \int_{-L}^{L} (f_n'(x) - f_n'(g_n(x,y)+x))W(y)h(y)dy. \end{aligned} \tag{97}$$

Then the number of extreme points of $f(x) - \lim_{n\to\infty} f_n(x)$ is at most the number of extreme points of $f(x)$ if $\lim_{n\to\infty} f_n(x)$ exists.

In order to prove Theorem 4, we need following lemma and corollaries.

Lemma 1. *Let $x_0 \in [x_i, x_{i+1}]$ where x_i is a local minima and x_{i+1} is a local maxima of $f_n(x)$ and $f_n(x)$ is monotonously increasing between (x_i, x_{i+1}). In (65), if the scaling function $g_n(x,y)$ is separable, i.e. $g_n(x,y) = l_n(x)h(y)$, then $f_{n+1}(x_0) < f_{n+1}(x_0+\epsilon)$ for small $\epsilon > 0$ if and only if*

$$c_n^{(1)}(x_0) + l_n'(x_0)c_n^{(2)}(x_0) > 0, \tag{98}$$

where

$$\begin{aligned} c_n^{(1)}(x_0) &= \int_{-L}^{L} (f_n'(x_0) - f_n'(g_n(x_0,y)+x_0))W(y)dy, \\ c_n^{(2)}(x_0) &= \int_{-L}^{L} (f_n'(x_0) - f_n'(g_n(x_0,y)+x_0))W(y)h(y)dy. \end{aligned} \tag{99}$$

Proof. For any small $\epsilon > 0$, taking the first term in the Taylor expansion of

$$f_n(x_0+\epsilon) - f_n(x_0) \quad \text{and} \quad f_n(g_n(x_0+\epsilon, y) + x_0 + \epsilon) - f_n(g_n(x_0,y)+x_0), \tag{100}$$

we get

$$f_n(x_0+\epsilon) - f_n(x_0) \approx \epsilon f_n'(x_0), \tag{101}$$

and

$$\begin{aligned}
&f_n(g_n(x_0+\epsilon,y)+x_0+\epsilon)-f_n(g_n(x_0,y)+x_0)\\
=&f_n(g_n(x_0+\epsilon,y)+x_0+\epsilon)-f_n(g_n(x_0,y)+x_0+\epsilon)+f_n(g_n(x_0,y)+x_0+\epsilon)\\
&-f_n(g_n(x_0,y)+x_0)\\
\approx&f_n'(g_n(x_0,y)+x_0+\epsilon)\frac{\partial g_n(x_0,y)}{\partial x}\epsilon+f_n'(g_n(x_0,y)+x_0)\epsilon\\
\approx&f_n'(g_n(x_0),y)+x_0)\frac{\partial g_n(x_0,y)}{\partial x}\epsilon+f'(g_n(x_0,y)+x_0)\epsilon\\
=&f_n'(g_n(x_0,y)+x_0)(\frac{\partial g_n(x_0,y)}{\partial x}+1)\epsilon.
\end{aligned} \tag{102}$$

Then we get the approximation of $f_{n+1}(x_0+\epsilon)-f_{n+1}(x_0)$

$$\begin{aligned}
&f_{n+1}(x_0+\epsilon)-f_{n+1}(x_0)\\
=&f_n(x_0+\epsilon)-\int_{-L}^{L}W(y)f_n(g_n(x_0+\epsilon,y)+x_0+\epsilon)dy\\
&-(f_n(x_0)-\int_{-L}^{L}W(y)f_n(g_n(x_0,y)+x_0)dy)\\
=&f_n(x_0+\epsilon)-f_n(x_0)\\
&-\int_{-L}^{L}W(y)(f_n(g_n(x_0+\epsilon,y)+x_0+\epsilon)-f_n(g_n(x_0,y)+x_0))dy\\
\approx&\epsilon[f_n'(x_0)-\int_{-L}^{L}W(y)f_n'(g_n(x_0,y)+x_0)(\frac{\partial g_n(x_0,y)}{\partial x}+1)dy].
\end{aligned} \tag{103}$$

Since the scaling function $g_n(x_0,y)$ is separable as

$$g_n(x_0,y)=l_n(x_0)h(y), \tag{104}$$

we can even simplify the approximation of $f_{n+1}(x_0+\epsilon)-f_{n+1}(x_0)$ from (103) as

$$\begin{aligned}
&f_{n+1}(x_0+\epsilon)-f_{n+1}(x_0)\\
\approx&\epsilon[f_n'(x_0)-\int_{-L}^{L}W(y)f_n'(l_n(x_0)h(y)+x_0)(l_n'(x_0)h(y)+1)dy].
\end{aligned} \tag{105}$$

Since $h(-y)=-h(y)$ and $W(-y)=W(y)$, the product of these two functions $h(y)W(y)$ is an odd function. So we have

$$\int_{-L}^{L}W(y)h(y)dy=0. \tag{106}$$

Modify $f_n'(x_0)$ by adding a zero term to it:

$$\begin{aligned} f_n'(x_0) &= f_n'(x_0) + f'(x_0)l_n'(x_0)\int_{-L}^{L} W(y)h(y)dy \\ &= \int_{-L}^{L} f_n'(x_0)W(y)dy + \int_{-L}^{L} f_n'(x_0)l_n'(x_0)W(y)h(y)dy \\ &= \int_{-L}^{L} W(y)f_n'(x_0)(l_n'(x_0)h(y)+1)dy. \end{aligned} \tag{107}$$

Using (107), the approximation of $f_{n+1}(x_0+\epsilon) - f_{n+1}(x_0)$ in (105) can be written as

$$\begin{aligned} & f_{n+1}(x_0+\epsilon) - f_{n+1}(x_0) \\ \approx & \epsilon \int_{-L}^{L} W(y)[f_n'(x_0) - f_n'(l_n(x_0)h(y)+x_0)][l_n'(x_0)h(y)+1]dy \\ = & \epsilon \int_{-L}^{L} W(y)[f_n'(x_0) - f_n'(l_n(x_0)h(y)+x_0)]dy \\ & + \epsilon \int_{-L}^{L} W(y)[f_n'(x_0) - f_n'(l_n(x_0)h(y)+x_0)]l_n'(x_0)h(y)dy \\ = & \epsilon \int_{-L}^{L} W(y)[f_n'(x_0) - f_n'(l_n(x_0)h(y)+x_0)]dy \\ & + l'(x_0\epsilon) \int_{-L}^{L} W(y)[f_n'(x_0) - f_n'(l_n(x_0)h(y)+x_0)]h(y)dy \\ = & \epsilon[c_n^{(1)}(x_0) + l_n'(x_0)c_n^{(2)}(x_0)]. \end{aligned} \tag{108}$$

So

$$f_{n+1}(x_0+\epsilon) - f_{n+1}(x_0) > 0 \quad \text{if and only if} \quad c_n^{(1)}(x_0) + l_n'(x_0)c_n^{(2)}(x_0) > 0, \tag{109}$$

where

$$\begin{aligned} c_n^{(1)}(x_0) &= \int_{-L}^{L} W(y)[f_n'(x_0) - f_n'(l_n(x_0)h(y)+x_0)]dy \\ &= \int_{-L}^{L} W(y)[f_n'(x_0) - f_n'(g_n(x_0,y)+x_0)]dy, \\ c_n^{(2)}(x_0) &= \int_{-L}^{L} W(y)[f_n'(x_0) - f_n'(l_n(x_0)h(y)+x_0)]h(y)dy \\ &= \int_{-L}^{L} W(y)[f_n'(x_0) - f_n'(g_n(x_0,y)+x_0)]h(y)dy. \end{aligned} \tag{110}$$

□

Lemma 1 holds for a tiny monotonously increasing interval. This property can be extended to a larger interval, such as a monotonously increasing interval between a local minimal point and a local maximal point. For the strictly increasing intervals of $f_n(x)$, apply Lemma 1 on $f_n(x)$, we get following corollary.

Corollary 1. *Let x_i, x_{i+1} be two adjacent extreme points of $f_n(x)$ and $f_n(x)$ is strictly increasing on (x_i, x_{i+1}). $f_{n+1}(x)$ is also strictly increasing on (x_i, x_{i+1}) if*

$$c_n^{(1)}(x) + l_n'(x)c_n^{(2)}(x) > 0, \tag{111}$$

for any $x \in (x_i, x_{i+1})$ where

$$\begin{aligned} c_n^{(1)}(x) &= \int_{-L}^{L} (f_n'(x) - f_n'(g_n(x,y)+x))W(y)dy, \\ c_n^{(2)}(x) &= \int_{-L}^{L} (f_n'(x) - f_n'(g_n(x,y)+x))W(y)h(y)dy. \end{aligned} \tag{112}$$

Proof. $f_{n+1}(x)$ is strictly increasing on (x_i, x_{i+1}) is equivalent to $f_{n+1}(x+\epsilon) > f_{n+1}(x)$, for any $x \in (x_i, x_{i+1})$ and small $\epsilon > 0$. Since $f_n(x)$ is strictly increasing on (x_i, x_{i+1}), by Lemma 1, $f_{n+1}(x+\epsilon) > f_{n+1}(x)$ if

$$c_n^{(1)}(x) + l_n'(x)c_n^{(2)}(x) > 0. \tag{113}$$

So $f_{n+1}(x)$ is strictly increasing on (x_i, x_{i+1}) if

$$c_n^{(1)}(x) + l_n'(x)c_n^{(2)}(x) > 0, \text{ for any } x \in (x_i, x_{i+1}), \tag{114}$$

where

$$\begin{aligned} c_n^{(1)}(x) &= \int_{-L}^{L} [f_n'(x) - f_n'(g_n(x,y)+x)]W(y)dy, \\ c_n^{(2)}(x) &= \int_{-L}^{L} [f_n'(x) - f_n'(g_n(x,y)+x)]h(y)W(y)dy. \end{aligned} \tag{115}$$

□

In a similar way, apply Lemma 1 on $-f_n(x)$ for the strictly decreasing intervals of $f_n(x)$, we get following corollary.

Corollary 2. *Let x_i, x_{i+1} be two adjacent extreme points for $f_n(x)$ and $f_n(x)$ is strictly decreasing on (x_i, x_{i+1}). $f_{n+1}(x)$ is strictly decreasing on (x_i, x_{i+1}) if*

$$c_n^{(1)}(x) + l_n'(x)c_n^{(2)}(x) < 0, \tag{116}$$

for any $x \in (x_i, x_{i+1})$ where $c_n^{(1)}(x)$ and $c_n^{(2)}(x)$ are defined in the same way as in (112).

Proof. Let $h_n(x) = -f_n(x)$. So $h_n(x)$ is strictly increasing on (x_i, x_{i+1}). By Corollary(1),

$$h_{n+1}(x) := h_n(x) - \int_{-L}^{L} h_n(x + g_n(x, y))W(y)dy, \tag{117}$$

is strictly increasing on (x_i, x_{i+1}) if $d_n^{(1)}(x) + l_n'(x)d_n^{(2)}(x) > 0$ for any $x \in (x_i, x_{i+1})$ where

$$\begin{aligned} d_n^{(1)}(x) &= \int_{-L}^{L} W(y)[h_n'(x) - h_n'(x + g_n(x, y))]dy, \\ d_n^{(2)}(x) &= \int_{-L}^{L} W(y)[h_n'(x) - h_n'(x + g_n(x, y))]h(y)dy. \end{aligned} \tag{118}$$

Since

$$h_n(x) = -f_n(x), \tag{119}$$

we have that

$$h_{n+1}(x) = -f_{n+1}(x), \tag{120}$$

and

$$\begin{aligned} d_n^{(1)}(x) &= -c_n^{(1)}(x), \\ d_n^{(2)}(x) &= -c_n^{(2)}(x), \end{aligned} \tag{121}$$

where

$$\begin{aligned} c_n^{(1)}(x) &= \int_{-L}^{L} W(y)[f_n'(x) - f_n'(x + g_n(x, y))]dy, \\ c_n^{(2)}(x) &= \int_{-L}^{L} W(y)[f_n'(x) - f_n'(x + g_n(x, y))]h(y)dy. \end{aligned} \tag{122}$$

So

$$d_n^{(1)}(x) + l'(x)d_n^{(2)}(x) > 0 \tag{123}$$

is equivalent to

$$-c_n^{(1)}(x) - l'(x)c_n^{(2)}(x) > 0. \tag{124}$$

As a result, $f_{n+1}(x)$ is strictly decreasing on (x_i, x_{i+1}) if

$$c_n^{(1)}(x) + l'(x)c_n^{(2)}(x) < 0. \tag{125}$$

□

The significance of Corollary 1 and Corollary 2 lies in the number of extreme points in $f_n(x)$. Since the monotonously increasing and monotonously decreasing parts keep increasing or decreasing respectively after one step in the inner iteration, the extreme points do not change their positions from $f_n(x)$ to $f_{n+1}(x)$, i.e., x_i is a maximum point in $f_n(x)$ at the beginning of the nth step in the inner iteration; then x_i is still a maximum point after the nth step in the inner iteration. As a result, the number of extreme points does not increase from step to step in the inner iteration. Based on this analysis, we prove Theorem 4, which is the convergence theorem of the outer iteration of IF algorithm with adaptive filters.

Proof. Since

$$\begin{aligned} c_n^{(1)}(x) + l_n'(x)c_n^{(2)}(x) > 0 \quad \text{when} \quad f_n'(x) > 0, \\ c_n^{(1)}(x) + l_n'(x)c_n^{(2)}(x) < 0 \quad \text{when} \quad f_n'(x) < 0, \end{aligned} \tag{126}$$

by Corollary (1) and Corollary (2), $f_{n+1}(x)$ and $f_n(x)$ are monotonously increasing or monotonously decreasing simultaneously for each $n \in \mathbb{Z}$. Since $f_1(x) = f(x)$, f_n and $f(x)$ are monotonously increasing or monotonously decreasing simultaneously for each $n \in \mathbb{Z}$. So if $\lim_{n\to\infty} f_n(x)$ exists, $\lim_{n\to\infty} f_n(x)$ and $f(x)$ are monotonously increasing or monotonously decreasing simultaneously. Consequently, the number of extreme points of $f(x) - \lim_{n\to\infty} f_n(x)$ is no greater than the number of extreme points of $f(x)$. □

In Theorem 4, if the number of extreme points of $f(x) - \lim_{n\to\infty} f_n(x)$ is less than number of extreme points of $f(x)$, in next step of the outer iteration, the function we

start with, $f(x) - \lim_{n\to\infty} f_n(x)$, is smoother than $f(x)$. If this property keeps true for each step of the outer iteration, i.e. the number of extreme points in the remaining signal keeps decreasing, then IF algorithm with adaptive filter converges for $f(x)$.

We check two kinds of signals for which the conditions (96) are satisfied easily. Let $f(x)$ be a piecewise linear function with period T, i.e. $f(x+T) = f(x)$. In addition assume that within one period

$$f\left(\left(n+\frac{1}{2}\right)T - t\right) = f\left(\left(n+\frac{1}{2}\right)T + t\right) \quad t \in \left[0, \frac{T}{2}\right]. \tag{127}$$

Let $l(x) = L$ and $g(x,y) = y$. So we have $l'(x) = 0$. When $f'(x) > 0$

$$\begin{aligned} c_1(x) &= \int_L^L w(y)[f'(x) - f'(x+y)]dy \\ &= \int_L^L w(y)f'(x)dy - \int_L^L w(y)f'(x+y)dy \\ &= f'(x) - \int_L^L w(y)f'(x+y)dy \\ &> 0, \end{aligned} \tag{128}$$

as a result

$$c_1(x) + l'(x)c_2(x) = c_1(x) + 0 = c_1(x) > 0. \tag{129}$$

When $f'(x) < 0$,

$$\begin{aligned} c_1(x) &= \int_L^L w(y)[f'(x) - f'(x+y)]dy \\ &= \int_L^L w(y)f'(x)dy - \int_L^L w(y)f'(x+y)dy \\ &= f'(x) - \int_L^L w(y)f'(x+y)dy \\ &< 0, \end{aligned} \tag{130}$$

as a result

$$c_1(x) + l'(x)c_2(x) = c_1(x) + 0 = c_1(x) < 0. \tag{131}$$

By (129) and (131), we see that for the piecewise linearly function $f(x)$ satisfying the periodicity assumption, by applying uniform filters, the two constraints are satisfied directly.

Moreover, we consider the same constraints for sinusoidal functions. Without loss of generality, let $f(x) = sin(\frac{2\pi}{T}x)$. Since $f(x)$ is periodic with period T, we shall use IF algorithm with uniform filters. So $l'(x) = 0$ since the filter length doesn't change from point to point. Let the length of the uniform filter be $2L$. If $T = 2L$

$$\begin{aligned}
c_1(x) &= \int_{-L}^{L} W(y)[f'(x) - f'(x+y)]dy \\
&= \int_{-L}^{L} W(y)\left[\sin'\left(\frac{\pi}{L}x\right) - \sin'\left(\frac{\pi}{L}(x+y)\right)\right]dy \\
&= \frac{\pi}{L}\int_{-L}^{L} W(y)\left[\cos\left(\frac{\pi}{L}x\right) - \cos\left(\frac{\pi}{L}(x+y)\right)\right]dy \\
&= \frac{\pi}{L}\left[\cos\left(\frac{\pi}{L}x\right) - \int_{-L}^{L}\cos\left(\frac{\pi}{L}x\right)\cos\left(\frac{\pi}{L}y\right)W(y)dy\right. \\
&\qquad \left. + \int_{-L}^{L}\sin\left(\frac{\pi}{L}x\right)\sin\left(\frac{\pi}{L}y\right)W(y)dy\right] \\
&= \frac{\pi}{L}\left[\cos\left(\frac{\pi}{L}x\right) - \cos\left(\frac{\pi}{L}x\right)\int_{-L}^{L}\cos\left(\frac{\pi}{L}y\right)W(y)dy\right. \\
&\qquad \left. + \sin\left(\frac{\pi}{L}x\right)\int_{-L}^{L}\sin\left(\frac{\pi}{L}y\right)W(y)dy\right].
\end{aligned} \tag{132}$$

Since $\sin\left(\frac{\pi}{L}y\right)$ is an odd function, we have that

$$\int_{-L}^{L}\sin(\frac{\pi}{L}y)W(y)dy = 0. \tag{133}$$

Then $c_1(x)$ could be written as

$$\begin{aligned}
c_1(x) =& \frac{\pi}{L}\left[\cos\left(\frac{\pi}{L}x\right) - \cos\left(\frac{\pi}{L}x\right)\int_{-L}^{L}\cos\left(\frac{\pi}{L}y\right)W(y)dy\right] \\
&= \frac{\pi}{L}\cos\left(\frac{\pi}{L}x\right)\left[1 - \int_{-L}^{L}\cos\left(\frac{\pi}{L}y\right)W(y)dy\right].
\end{aligned} \tag{134}$$

Since

$$\left|\int_{-L}^{L}\cos\left(\frac{\pi}{L}y\right)W(y)dy\right| < \int_{-L}^{L}\left|\cos\left(\frac{\pi}{L}y\right)\right|W(y)dy < \int_{-L}^{L}W(y)dy = 1, \tag{135}$$

we get

$$-1 < \int_{-L}^{L}\cos\left(\frac{\pi}{L}y\right)W(y)dy < 1. \tag{136}$$

So

$$0 < 1 - \int_{-L}^{L} \cos\left(\frac{\pi}{L}y\right) W(y)dy < 2, \tag{137}$$

and $c_1(x)$ has the same sign of $\cos(\frac{\pi}{L}x)$. Moreover since $l'(x) = 0$ for uniform filters, we have

$$\begin{aligned} c_1(x) + l'(x)c_2(x) > 0 \quad \text{when} \quad f'(x) > 0, \\ c_1(x) + l'(x)c_2(x) < 0 \quad \text{when} \quad f'(x) < 0. \end{aligned} \tag{138}$$

If $T > 2L$, we could extend the uniform filter $W(y), y \in [-L, L]$ to be a filter $W_1(y), y \in [-\frac{T}{2}, \frac{T}{2}]$ with length T by adding two zero parts:

$$W_1(y) = \begin{cases} 0 & \text{if } -\frac{T}{2} \le y < -L, \\ W(y) & \text{if } -L \le y \le L, \\ 0 & \text{if } L < y \le \frac{T}{2}. \end{cases} \tag{139}$$

Apply the same computation as in the case $T = 2L$ with $W_1(y), y \in [-\frac{T}{2}, \frac{T}{2}]$ we get the same result that

$$\begin{aligned} c_1(x) + l'(x)c_2(x) > 0 \quad \text{when} \quad f'(x) > 0, \\ c_1(x) + l'(x)c_2(x) < 0 \quad \text{when} \quad f'(x) < 0. \end{aligned} \tag{140}$$

If $T < 2L$

$$\begin{aligned} c_1(x) &= \int_{-L}^{L} W(y)[f'(x) - f'(x+y)]dy \\ &= \int_{-L}^{L} W(y) \left[\sin'\left(\frac{2\pi}{T}x\right) - \sin'\left(\frac{2\pi}{T}(x+y)\right)\right] dy \\ &= \frac{2\pi}{T} \int_{-L}^{L} W(y) \left[\cos\left(\frac{2\pi}{T}x\right) - \cos\left(\frac{2\pi}{T}(x+y)\right)\right] dy \\ &= \frac{2\pi}{T} \left[\cos\left(\frac{2\pi}{T}x\right) - \int_{-L}^{L} \cos\left(\frac{2\pi}{T}x\right) \cos\left(\frac{2\pi}{T}y\right) W(y)dy \right. \\ &\qquad \left. + \int_{-L}^{L} \sin\left(\frac{2\pi}{T}x\right) \sin\left(\frac{2\pi}{T}y\right) W(y)dy\right] \\ &= \frac{2\pi}{T} \cos\left(\frac{2\pi}{T}x\right) \left[1 - \int_{-L}^{L} \cos\left(\frac{2\pi}{T}y\right) W(y)dy\right]. \end{aligned} \tag{141}$$

Since

$$\left[1 - \int_{-L}^{L} \cos\left(\frac{2\pi}{T}y\right) W(y)dy\right] > 0, \tag{142}$$

$c_1(x)$ has the same sign as $f'(x)$. We get that

$$\begin{aligned} c_1(x) + l'(x)c_2(x) > 0 \quad \text{when} \quad f'(x) > 0, \\ c_1(x) + l'(x)c_2(x) < 0 \quad \text{when} \quad f'(x) < 0. \end{aligned} \tag{143}$$

Based on these analysis, we get that when the signal is a sinusoidal function, applying IF algorithm with uniform filters does not change the positions of extreme points, thus does not change the number of extreme points.

We consider the direct requirement of $l'(x)$ for some small neighbourhoods near the extreme points of $f'(x)$. If the scaling function has a special form such that $g(x, y) = \frac{l(x)}{L} y$, then

$$c_1(x) + l'(x)c_2(x) > 0 \tag{144}$$

is equivalent to

$$\int_{-L}^{L} W(y) f'(x) dy - \int_{-L}^{L} W(y) f' \left(\frac{l(x)}{L} y + x \right) \left(\frac{l'(x)}{L} y + 1 \right) > 0. \tag{145}$$

We could simply get that

- $l'(x) < 0$ when x is on the left of the maxima of $f'(x)$;
- $l'(x) > 0$ when x is on the right of the maxima of $f'(x)$.

On the other hand

$$c_1(x) + l'(x)c_2(x) < 0 \tag{146}$$

is equivalent to

$$\int_{-L}^{L} W(y) f'(x) dy - \int_{-L}^{L} W(y) f' \left(\frac{l(x)}{L} y + x \right) \left(\frac{l'(x)}{L} y + 1 \right) < 0. \tag{147}$$

Similary, we could get that

- $l'(x) > 0$ when x is on the left of the minima of $f'(x)$;
- $l'(x) < 0$ when x is on the right of the minima of $f'(x)$.

2.3 Conclusions

The aim of this chapter is to study the convergence of IF algorithm. we show the convergence of the inner iteration of IF algorithm under proper conditions with both uniform filters and adaptive filters. In addition, we also show the convergence of the outer iteration of IF algorithm by controlling the number of extreme points in the remaining signal. We would like to emphasize that the conditions required in the convergence theorems can either be achieved by adjusting the filter length or be shown by checking easily after each step of the inner iteration.

CHAPTER III

ADAPTIVE LOCAL ITERATIVE FILTERING ALGORITHM

In Chapter 2, we establish the fundamental of IF algorithm by showing the convergence theorems. In this chapter, we propose Adaptive Local Iterative Filtering (ALIF) algorithm based on the convergence theorem. In Section 3.1, we set up an approach of adjusting the filter length adaptively according to the signal's local behaviour. This approach is guided by Theorem 4 and the convergence is guaranteed. In Section 3.2, we develop a local filter which is a smooth function with a compact support based on a PDE model. This local filter could be applied in ALIF algorithm. In Section 3.3, we present a special adaptivity strategy to handle signals with jumps using only one side information. In Section 3.4, we show the results of numerical experiments of ALIF algorithm on both simulated signals and real data. In Section 3.5, we present specific numerical examples which demonstrate the stability of ALIF algorithm.

3.1 Adaptive Iterative Filtering Techniques

3.1.1 The ALIF Algorithm

Adaptive Local Iterative Filtering (ALIF) algorithm is given on the next page.

In Theorem 4, if the number of extreme points of $f(x) - \lim_{n\to\infty} f_n(x)$ is less than number of extreme points of $f(x)$, in next step of the outer iteration, the function we start with, $f(x) - \lim_{n\to\infty} f_n(x)$, is smoother than $f(x)$. If this property keeps true for each step of the outer iteration, i.e. the number of extreme points in the remaining signal keeps decreasing, ALIF algorithm converges for signal $f(x)$.

Algorithm 3 IMF = ALIF(f)

IMF=$\emptyset$
while the number of extrema ≥ 2 **do**
  $f_1 = f$
  while the stopping criteria is not satisfied **do**
    compute $l_n(x)$ for f_n
    $f_{n+1} = f_n - \int_{-l_n(x)}^{l_n(x)} f(x+t)w_n(x,t)dt$
    $n = n + 1$
  end while
  IMF = IMF$\cup\{f_n\}$
  $f = f - f_n$
end while
IMF = IMF$\cup\{f\}$

3.1.2 Filter Lengths Computation

In Algorithm 3, there is a step of computing the filter length $l_n(x)$ for signal $f_n(x)$. In general, $l_n(x)$ is a positive function. For special cases where $l_n(x)$ is a positive constant function

$$l_n(x) := \left\lfloor \frac{2N}{k} \right\rfloor, \tag{148}$$

where N is the total sample points of signal $f_n(x)$ and k is the number of extreme points of $f_n(x)$. When $l_n(x)$ is a constant function, the moving average of $f_n(x)$ is degenerated to the convolution of $f_n(x)$ and $w_n(x)$ where $w_n(x)$ is a uniform filter with length $2l_n(x)$. In this section, we focus on the general situation where $l_n(x)$ is not restricted to a positive constant. For this general case, there are more than one way to compute $l_n(x)$. In [78], Wang proposed an interpolation method of the filter length. let x_i denote the position of the ith local extreme point of $f_n(x)$, then the filter length $2l_n(x_i)$ at x_i is given by

$$2l_n(x_i) = x_{i+2} - x_{i-2}. \tag{149}$$

Then the filter length $2l_n(x)$ for any other point x is given by the interpolation with the known filter length $(x_i, 2l_n(x_i))$. Similar to this approach, we propose another approach in order to utilize more local information. The half filter length $l_n(x_i)$ at

the ith local extreme point x_i is given by

$$l_n(x_i) = x_{i+1} - x_{i-1} \tag{150}$$

and the half filter length $l_n(x)$ for any other point x is given by the interpolation based on $(x_i, l_n(x_i))$. However, the computation of the filter length $l_n(x)$ in either (149) or (150) does not show any clue about convergence or relations with Theorem 4.

With the purpose of ensuring the convergence, we design adaptive techniques to compute the filter length $l_n(x)$ such that (96) holds for each step of the inner iteration based on Theorem 4. To satisfy the constraints in (96), it is necessary to adjust the filter length $l_n(x)$ according to the value of $l'_n(x)$ which satisfies (96). Since the constraints are inequalities about $l'_n(x)$, there are infinite number of choices for the value of $l'_n(x)$. As a simple choice, we first convert the two inequalities in (96) to equality constraints with right hand sides which are properly set up. Then we obtain the filter length $l_n(x)$ from point to point by the values of $l'_n(x)$ in the equality constraints.

Let the right hand sides be $t_n(x)$, then the two constraints in (96) can be written as

$$c_n^{(1)}(x) + l'_n(x)c_n^{(2)}(x) = t_n(x), \tag{151}$$

where $t_n(x)$ satisfies

$$t_n(x) \begin{cases} > 0, & \text{when} \quad f'_n(x) > 0, \\ < 0, & \text{when} \quad f'_n(x) < 0. \end{cases} \tag{152}$$

One simple selection of $t_n(x)$ is the step function as

$$t_n(x) = \begin{cases} c, & \text{when} \quad f'_n(x) > 0, \\ -c, & \text{when} \quad f'_n(x) < 0. \end{cases} \tag{153}$$

This selection works from the constraint point of view. However, there are two issues we need to consider in this selection. One is that we need to be really careful to pick

up a value for c. Large values of c lead to big values of $l'_n(x)$ and consequently cause big variation of $l_n(x)$ or even negative values in $l_n(x)$. Small values of c generate small $l'_n(x)$ and then $l_n(x)$ almost as a constant. This may work well only for some part of the signal especially when the signal is stationary. The other issue is that even if we found a good value of c for certain signal, it does not work well for all the others. Due to these considerations, a step function as (153) is not the best choice for the function $t_n(x)$.

We propose a better selection of $t_n(x)$ as

$$t_n(x) = \begin{cases} \lambda|c_n^{(1)}(x)|, & \text{when} \quad f'_n(x) > 0, \\ -\lambda|c_n^{(1)}(x)|, & \text{when} \quad f'_n(x) < 0, \end{cases} \tag{154}$$

where λ is a constant with the scale of 1. This selection is adaptive since in (154) the value of $t_n(x)$ depends on the value of $c_n^{(1)}(x)$, which reflects the local behaviour of a signal. By choosing the fixed filter $W(y), y \in [-L, L]$ with a large length, the function $h(y)$ has the value much smaller than 1. Consequently, $c_n^{(1)}(x)$ has greater absolute value than $c_n^{(2)}(x)$. With the underneath assumption that the filter length has only smooth changes, this observation plays its role in (154) by setting λ close to 1. We shall implement the selection of $t_n(x)$ as (154) later in Section 3.4.

To get the filter length $l_n(x)$ for the whole signal $f_n(x)$, we start with the filter length $l_n(x_0)$ at a particular point x_0 and extend the function $l_n(x)$ to both the left and the right sides by (151) and (154). Basically, we select the point x_0 as the local extreme point which has the minimal distance to its adjacent extreme points. Assume the right adjacent extreme point of x_0 is x_1 and the left adjacent extreme point of x_0 is x_{-1}. Then $l_n(x_0) := x_1 - x_{-1}$. Given the filter length $l_n(x)$ at point x, the filter length at point $x + dx$ is computed by

$$l_n(x + dx) = l_n(x_0) + dx \frac{t_n(x) - c_n^{(1)}(x)}{c_n^{(2)}(x)}. \tag{155}$$

So the filter length $l_n(x)$ of signal $f_n(x)$ is obtained point after point using (155). We propose to make use of this computation of the filter length in ALIF algorithm.

We point out here that the implementation of the above approach requires careful attention. Given a signal $f(x)$ if we sample it every Δx, then $x = n\Delta x$ with $n = 1, 2, \ldots, N$. To ensure the convergence of the algorithm applied to the given $f(x)$, $l'_n(x)$ has to satisfy the condition (151) at every sample point $x = n\Delta x$. We refer to the set containing the sample points as the *physical space*. In the implementation of the algorithm it is required to rescale the variable x into $y = x/\Delta x$ such that Δy, the distance between two consecutive samples points y, is equal 1. We refer to the set containing the values y as the *computational space*.

Let us consider the condition that $l'_n(x)$ should satisfy in the computational space. Let $L'_n(y)$ denote the rate of change of the filter length in the computational space. Since $l'_n(x)$ is independent of the value of Δx, at any point $x = y\Delta x$ we have that $l'_n(x) = L'_n(y)$. So (151) can be rewritten as

$$c_n^{(1)}(y\Delta x) + L'_n(y)\frac{dy}{dx}c_n^{(2)}(y\Delta x) = t_n(y\Delta x). \tag{156}$$

Let $C_n^{(1)}(y) = c_n^{(1)}(y\Delta x)$, $C_n^{(2)}(y) = c_n^{(2)}(y\Delta x)$, $T_n(y) = t_n(y\Delta x)$, and since $dy/dx = 1/\Delta x$ we get

$$C_n^{(1)}(y) + L'_n(y)\frac{1}{\Delta x}C_n^{(2)}(y) = T_n(y). \tag{157}$$

So the condition that $L'_n(y)$ should satisfy in the computational space is

$$L'_n(y) = \Delta x\frac{T_n(y) - C_n^{(1)}(y)}{C_n^{(2)}(y)}, \tag{158}$$

therefore (155) becomes

$$L_n(y+1) = L_n(y) + \frac{T_n(y) - C_n^{(1)}(y)}{C_n^{(2)}(y)}\Delta x. \tag{159}$$

ALIF algorithm is implemented based on (159).

Furthermore, in order to make sure the filter length has smooth change, we set up a threshold δ for $c_n^{(2)}(x)$. When $c_n^{(2)}(x)$ is smaller than δ, $c_n^{(2)}(x)$ is set to be δ.

3.2 Local Filters Developed From A PDE Model

To effectively handle non-linear and non-stationary signals, it is highly desirable to use filters with compact support, for the simple reason that filters with longer support may mix features that are far apart in a signal. This could be troublesome, especially for signals with transient information. However, the compact support low pass filters, such as the double average filters, used in the existing IF algorithms are not smooth enough. They may create artificial oscillations in subsequent IMFs, due to the non-smoothness. This motivates us to design filters from the solution of certain Fokker-Planck equations, because they are compactly supported, infinitely differentiable and vanishing to zero at both ends smoothly. Thus they can avoid creating any artificial oscillations due to the non-smoothness of the filters.

Let us consider the Fokker-Planck equation

$$p_t = -\alpha(h(x)p)_x + \beta(g^2(x)p)_{xx}, \quad \alpha, \beta > 0. \tag{160}$$

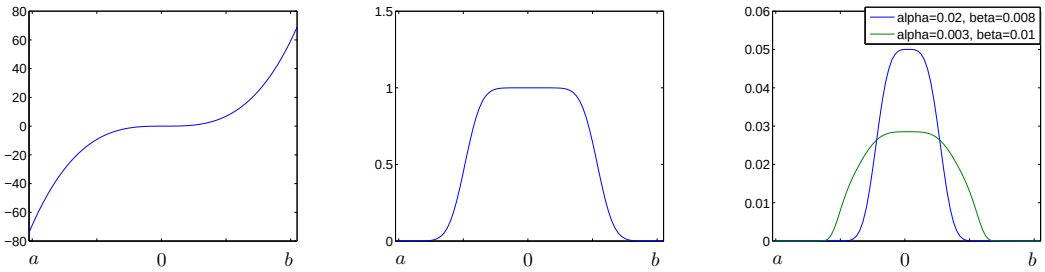
Assume $h(x)$ and $g(x)$ are smooth enough functions such that there exist $a < 0 < b$ satisfying:

- $g(a) = g(b) = 0, g(x) > 0$ for $x \in (a, b)$;
- $h(a) < 0 < h(b)$.

The $(g^2(x)p)_{xx}$ term generates diffusion effect and pulls out the density from the center of (a, b) towards a and b while the $-(h(x)p)_x$ term transports the density from a and b towards the center of the interval (a, b). When the two forces are balanced, the steady state is achieved. There exists a smooth non trivial solution $p(x)$ of the stationary problem:

$$-\alpha(h(x)p)_x + \beta(g^2(x)p)_{xx} = 0 \tag{161}$$

satisfying $p(x) \geq 0$ for $x \in (a, b)$, and $p(x) = 0$ for $x \notin (a, b)$. That means the solution is concentrated in the interval $[a, b]$ and there is no leakage outside. So $p(x)$ is a local



(a) $h(x)$ (b) $g^2(x)$ (c) $p(x)$

Figure 8: coefficient functions and steady states of (162): $h(x)$ is an odd function and here we use x^3; $g^2(x)$ is an even function and it is a smooth approximation to the step function; (c) shows two steady states for coefficients $\alpha = 0.02$, $\beta = 0.008$ and $\alpha = 0.003$, $\beta = 0.01$ respectively.

filter satisfying our requirement.

Based on this analysis, we shall use the solution to the following initial value problem as the filter in our decomposition algorithm:

$$\begin{aligned} p_t &= -\alpha(h(x)p)_x + \beta(g^2(x)p)_{xx}, \\ p(x,0) &= \delta\left(x - \frac{a+b}{2}\right), \end{aligned} \tag{162}$$

where $\delta(x)$ is the Dirac delta function . By adjusting the functions $h(x), g(x)$ as well as coefficients α, β, we can get different shapes of filters. In Figure 8, we plot two steady states for different α, β respectively for the same $h(x)$ and $g(x)$ functions.

We can see that when α is larger, the weight is concentrated more in the center; on the other hand, when β is larger, the weight is diffused more and concentrated less in the center. When designing the local filter based on the Fokker-Planck equation, we first fix the functions $f(x)$ and $g(x)$ then adjust the coefficients α and β to get to the filter shape we want.

3.2.1 Filters With Different Lengths

Another issue involved is about designing filters with different lengths but the same shape. There are two approaches available. One way is to solve the Fokker-Planck equation again with $h(x)$ and $g(x)$ scaled in x for every different a or b. Assume we

get the steady state of (162), in order to get the filter with length from $\hat{a}$ to $\hat{b}$, we solve (163) and the steady state is the filter we want.

$$\begin{aligned} p_t = &-\alpha \left(h\left(\hat{a}\frac{x-(a+b)/2}{a} + \frac{\hat{a}+\hat{b}}{2} \right) p \right)_x \\ &+ \beta \left(g^2 \left(\hat{a}\frac{x-(a+b)/2}{a} + \frac{\hat{a}+\hat{b}}{2} \right) p \right)_{xx}, \\ p(x,0) = &\ \delta\left(x - \frac{\hat{a}+\hat{b}}{2} \right). \end{aligned} \tag{163}$$

The other way is to solve the Fokker-Planck equation for a fixed a and b only once and take a special interpolation of the steady state to get the proper filter. The former approach generates the exact solution of PDE, but solving PDEs numerically is much slower than interpolating existing filters. The latter approach is easier to implement. So in our simulations, we use this special interpolation strategy to get filters with same shape for different lengths. We present this special interpolation strategy in the following paragraphs.

Assume we get the numerical solution for the steady state of (162) with $2k+1$ discrete points $x_i, i = 1, 2, ..., 2k+1$ where k is a large natural number and $[x_1, x_2, ..., x_{2k+1}]$ is a discretization of the interval $[a, b]$. The function value $p(x_i)$ is the weight in the interval $[x_i, x_{i+1})$ and the sum of weights in all the intervals equals 1. Assume for another interval $[\hat{a}, \hat{b}]$ we have a different discretization $[\hat{x}_1, \hat{x}_2, ..., \hat{x}_{2n+1}]$. There is a one-to-one linear correspondence from the interval $[\hat{a}, \hat{b}]$ to the interval $[a, b]$. By this correspondence, $[\hat{x}_1, \hat{x}_2, ..., \hat{x}_{2n+1}]$ are mapped into $[y_1, y_2, ..., y_{2n+1}]$. The values of the filter $w_j, j = 1, 2, ..., 2n+1$ in the interval $[\hat{a}, \hat{b}]$ will be the weights in the intervals $[y_j, y_{j+1}], j = 1, 2, ..., 2n+1$ respectively, where the weight in the interval $[y_j, y_{j+1}]$ is given by

$$w_{[y_j, y_{j+1}]} = \int_{y_j}^{y_{j+1}} p(x)dx. \tag{164}$$

The integral is computed by the Riemamn sum based on the discrete points x_i and the weights $p(x_i)$, $i = 1, 2, ..., 2k+1$. If y_j falls between two points x_{m_1-1}, x_{m_1} and

y_{j+1} falls between two points x_{m_2}, x_{m_2+1}. Then

$$\int_{y_j}^{y_{j+1}} p(x)dx = p(x_{m_1-1})(x_{m_1} - y_j) + \sum_{i=m_1}^{m_2-1} p(x_i) + p(x_{m_2})(y_{j+1} - x_{m_2}). \tag{165}$$

So the filter weight w_j will be

$$w_j = p(x_{m_1-1})(x_{m_1} - y_j) + \sum_{i=m_1}^{m_2-1} p(x_i) + p(x_{m_2})(y_{j+1} - x_{m_2}). \tag{166}$$

Using this special interpolation method, the shapes of filters with different lengths are the same. Moreover, the filter length could be any positive real number, it is not limited to integers.

3.3 Iterative Filtering With One Side Information

For some signals, there can be significant changes or jump in the magnitude. When computing the moving average of this type of signals, it gets severe impact from the jumps and behaves unexpectedly. It is nice that the algorithm should be able to deal with the jumps, i.e. the jumps of signals do not have huge influence on nearby positions. To get such a capacity, we make use of one side information to compute the moving average when jumps happen. The technique is based on two steps. The first step is to detect positions of jumps in a signal. Treat the left side and the right side of a jump as two signals. In the second step, for the left signal, we extend the right boundary of the signal so that the moving average can be computed; for the right one, we extend the left boundary of it. The moving average of the whole signal is the catenation of the moving average from the left side and the moving average from the right side.

To detect the positions of jumps, we compute the variation of two consecutive points. If the value of the variation is extremely larger than the average variation, a jump is regarded to exist from one point to the other. Then the signal $f(t)$ is separated into two parts by splitting at the jump: the left part $f_L(t)$ and the right part $f_R(t)$. Assume we use the uniform filters and the filter length is $2m$ and assume the position

of the jump is t_0. The left part $f_L(t)$ only provides complete data to compute the moving average upto $t_0 - m$. To compute the moving average for $[t_0 - m, t_0]$, the right boundary of $f_L(t)$ need to be extended properly. The technique we use is based on the least square fitting. For the series of points

$$(t_0 - 2m, \mathcal{L}f(t_0 - 2m)), (t_0 - 2m + 1, \mathcal{L}f(t_0 - 2m + 1)), ..., (t_0 - m, \mathcal{L}f(t_0 - m)), \quad (167)$$

we could find a linear function as the least squares solution. Compute the values of this linear function at $t = t_0 - m + 1, t_0 - m + 2, ..., t_0$ and extend the right boundary of $f_L(t)$ by these values. Similarly, the right part signal $f_R(t)$ only provides complete data to compute the moving average down to $t_0 + m$. To extend the left boundary of $f_R(t)$, we find the linear function fitting points

$$(t_0+m+1, \mathcal{L}f(t_0+m+1)), (t_0+m+2, \mathcal{L}f(t_0+m+2)), ..., (t_0+2m, \mathcal{L}f(t_0+2m)) \quad (168)$$

with least squares. Compute the values of this linear function at $t = t_0, t_0+1, ..., t_0+m$ and extend the left boundary of $f_R(t)$ by these values.

3.4 Numerical Experiments

In this section, we show the performance of ALIF algorithm given in Algorithm 3. In addition, the filter length $l_n(x)$ is computed by the adaptive techniques we propose in Section 3.1. we use the designed filter given in Figure 9 which is based on the PDE model as described in the Section 3.2, where f(x), g(x) are the functions shown in Figure 8 and the coefficients are $\alpha = 0.005$, $\beta = 0.09$. Filters with different lengths are computed using the special interpolation method introduced in Section 3.2.

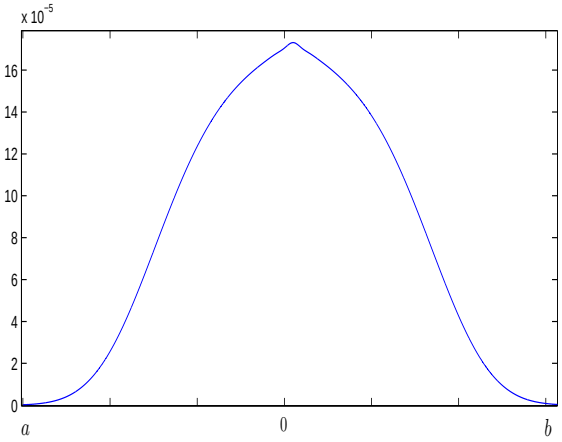


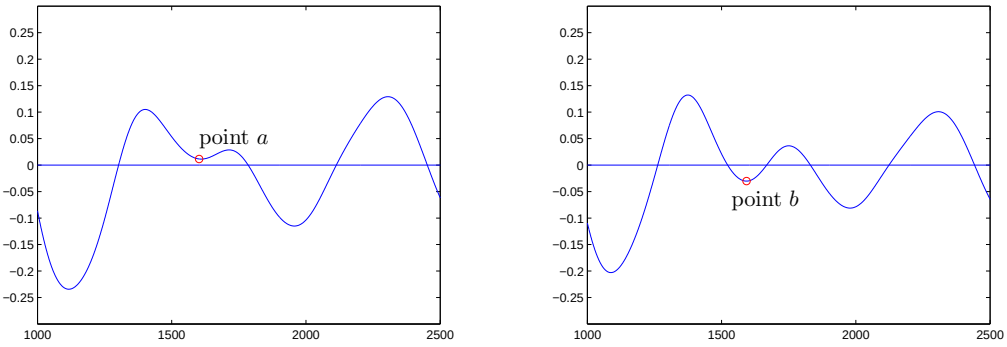
Figure 9: The local filter we use in numerical implementations.

For the stopping criterion, we can either put a limit on the maximal iterations number or stop the iteration when SD reaches certain threshold. Let $I_{k,n} = \mathcal{S}_k^n(x - I_1 - ... - I_{k-1})$ where $\mathcal{S}_k$ denotes the operator $\mathcal{S}$ to obtain the kth IMF.

$$SD := \frac{\|I_{k,n} - I_{k,n-1}\|_2}{\|I_{k,n-1}\|_2}. \tag{169}$$

It is also possible to adopt different stopping criteria for different IMFs. Consider a noisy signal, we may use looser stopping criteria for the first few IMFs to reduce the number of noise components and use stronger stopping criteria for the remaining IMFs to attract different patterns in following components. In the implementation, we use as stopping criterion $SD \leq \delta$ where δ is usually set to be 0.08.

As observed previously, a good decomposition method should capture all the finest oscillations around a moving average. That means the IMFs should satisfy at least this condition: all the local maximal values are positive and all the local minimal values are negative. Using iterative filtering method eventually we can get to this point. One example is given in Figure 10.



(a) (b)

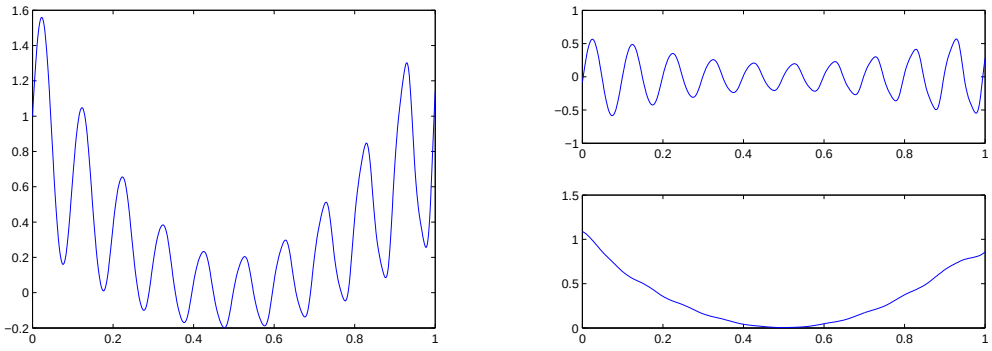
Figure 10: The effect of iterative filtering: (a) After 3 iterations the local minima point a is above 0. (b) After 5 iterations the local minima point b is below 0. Other parts do not change significantly from (a) to (b).

We test the proposed ALIF algorithm on both simulated signals and real-life data sets. Among these examples, the first three are simulated signals, the next four are real-life signals and last one is the example to show the performance of ALIF using only one side information.

Example 1 We test ALIF algorithm on a non-stationary frequency modulated signal

$$f(t) = 4(t-0.5)^2 + (2(t-0.5)^2 + 0.2)\sin\left((20\pi + 0.2\cos(40\pi t))t\right), t \in [0,1]. \quad (170)$$

From Figure 11, we see that $f(t)$ is decomposed into two components. The first is the frequency modulated signal $(2(t-0.5)^2+0.2)\sin\left((20\pi + 0.2\cos(40\pi t))t\right)$ and the second is the trend $4(t-0.5)^2$.



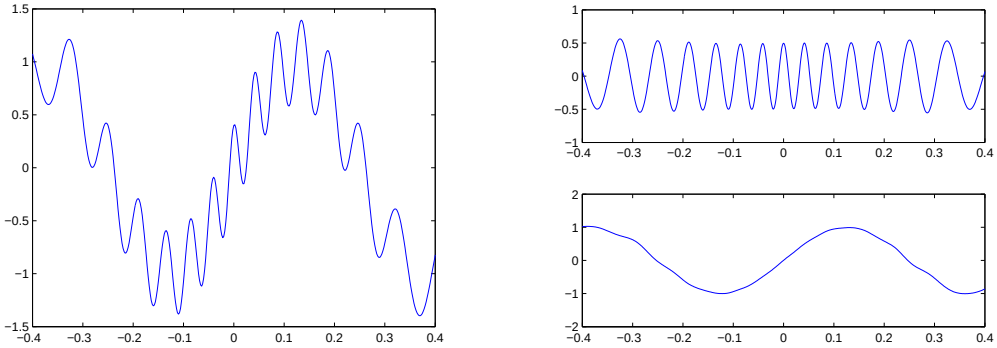
(a) (b)

Figure 11: (a) signal given in (170); (b) shows two components in the decomposition.

Example 2 We test ALIF algorithm on the highly non-stationary signal

$$f(t) = \sin(4\pi t) + 0.5\cos(50\pi|t| - 40\pi t^2), \quad t \in [-0.4, 0.4]. \tag{171}$$

As shown in Figure 12, $f(t)$ is separated into two IMFs. One has a varying instantaneous frequency and the other has a constant instantaneous frequency.



(a) (b)

Figure 12: (a) signal given in (171); (b) components obtained from the decomposition.

Example 3 In this example, we test the difference between the white noise and the chaotic signals. The chaotic signals are generated by the Lorenz system given in the

following

$$\begin{cases} \dot{x} = \theta(y - x), \\ \dot{y} = x(\rho - z) - y, \\ \dot{z} = xy - \beta z. \end{cases} \tag{172}$$

when $\rho = 28, \theta = 10$ and $\beta = 8/3$, the solution of the Lorenz system is shown in Figure 13. Each variable x, y or z is a chaotic function. We pick up the variable x for the numerical experiment and plot it in Figure 14. To test the difference in the decomposition with the white noise signal, we pick up one example $n(t) \sim (0, 5^2)$ as shown in Figure 15. The standard deviation is set to be 5 so that the white noise has a comparable magnitude with the x variable of the solution.

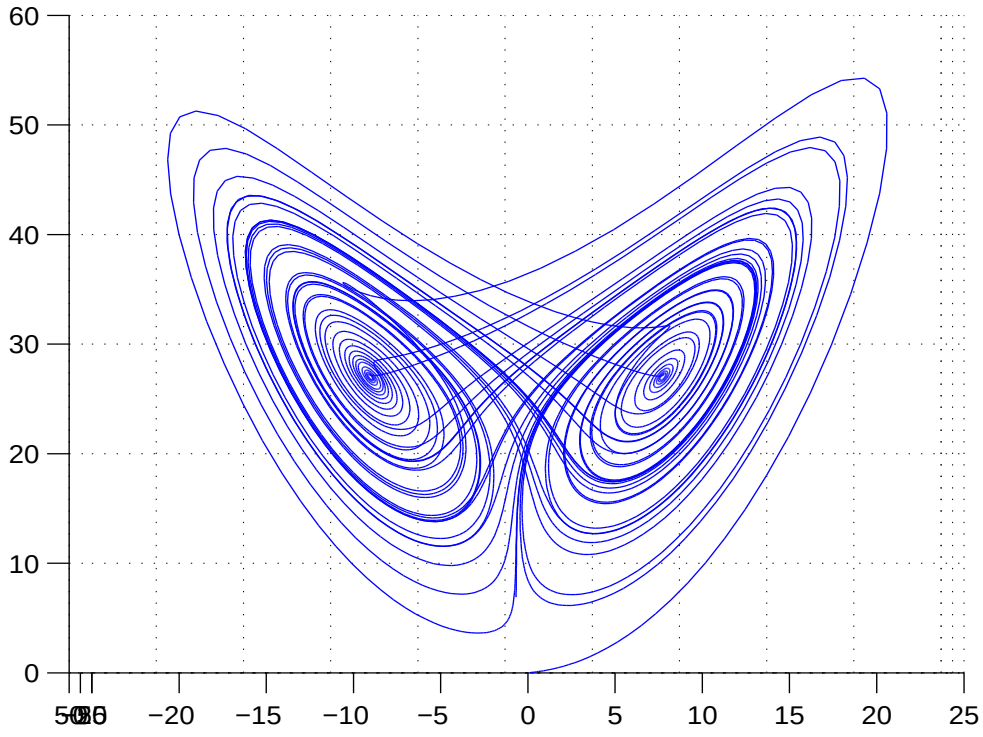


Figure 13: A plot of a solution of Lorenz system when $\rho = 28$, $\theta = 10$ and $\beta = 8/3$.

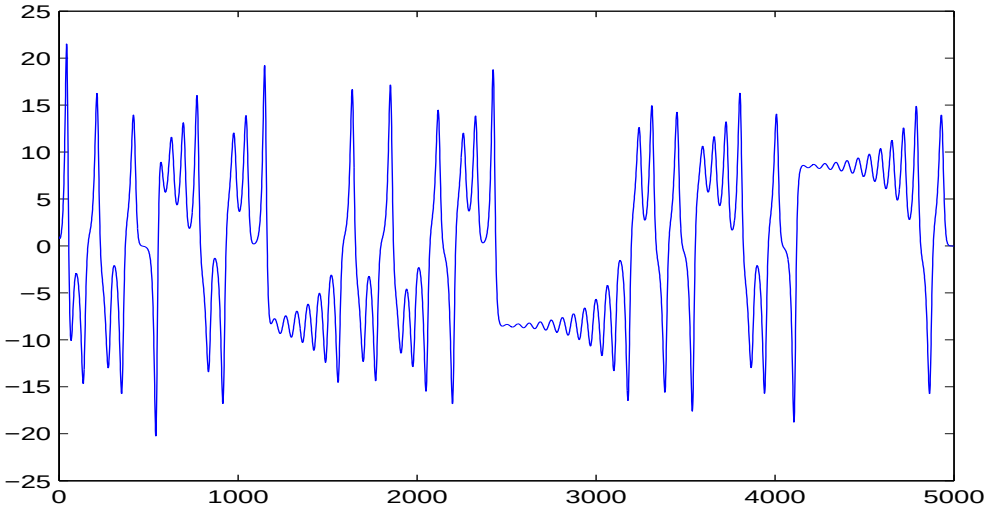


Figure 14: A plot of the x variable in a solution of Lorenz system when $\rho = 28$, $\theta = 10$ and $\beta = 8/3$.

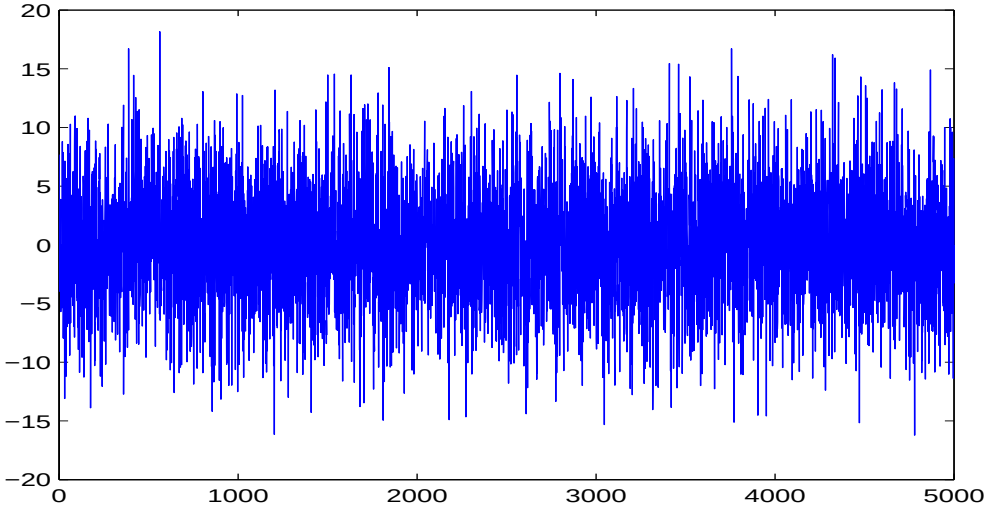


Figure 15: A white noise $n(t) \sim (0, 5^2)$.

Apply ALIF algorithm on the signal corresponding to the x variable, the decomposition of x is shown in Figure 16. Apply ALIF algorithm on the white noise signal shown in Figure 15 and plot its decomposition in Figure 17. The difference of the

components for these two signals includes that: the first few components of the white noise has a higher frequency than the first few components of the chaotic signal; the first few components of the chaotic signal, whose magnitude drops to quite small values occasionally, are non-stationary while the first few components of the white noise are relatively homogeneous. The second difference would be useful to distinguish the chaotic signals and noise signals when both have similar frequencies.

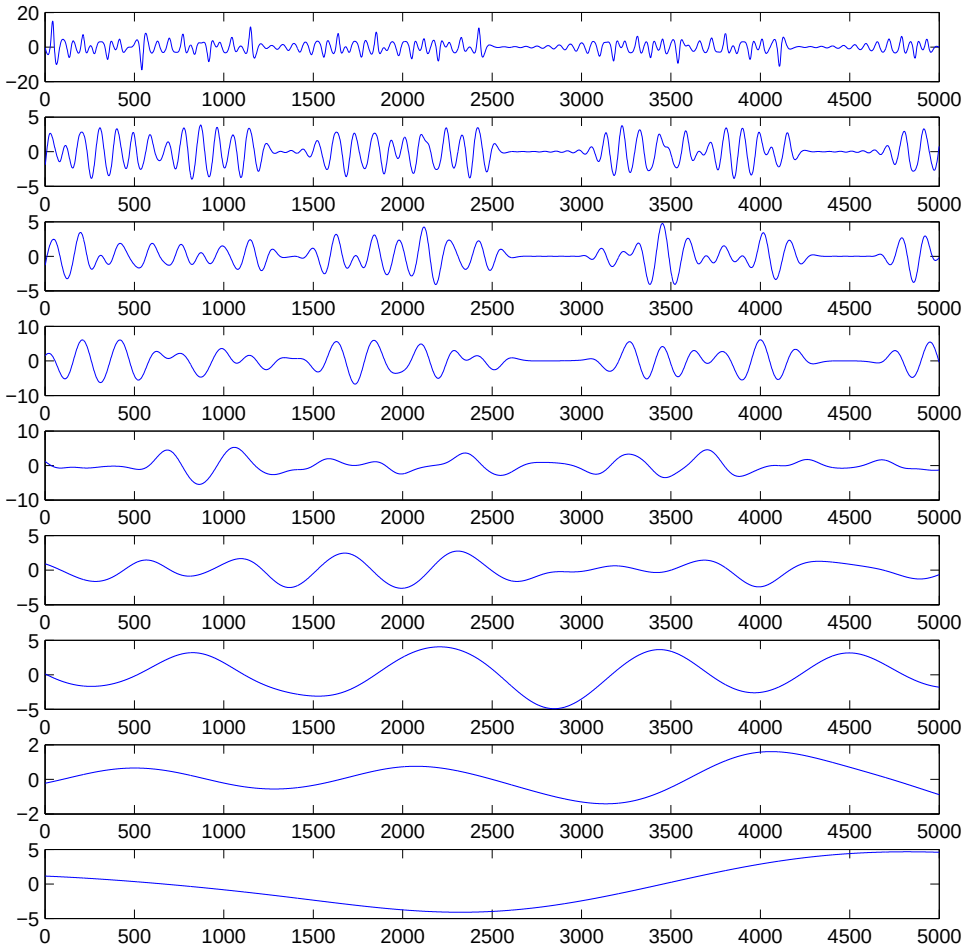


Figure 16: The decomposition of the x variable in the solution of the Lorenz system.

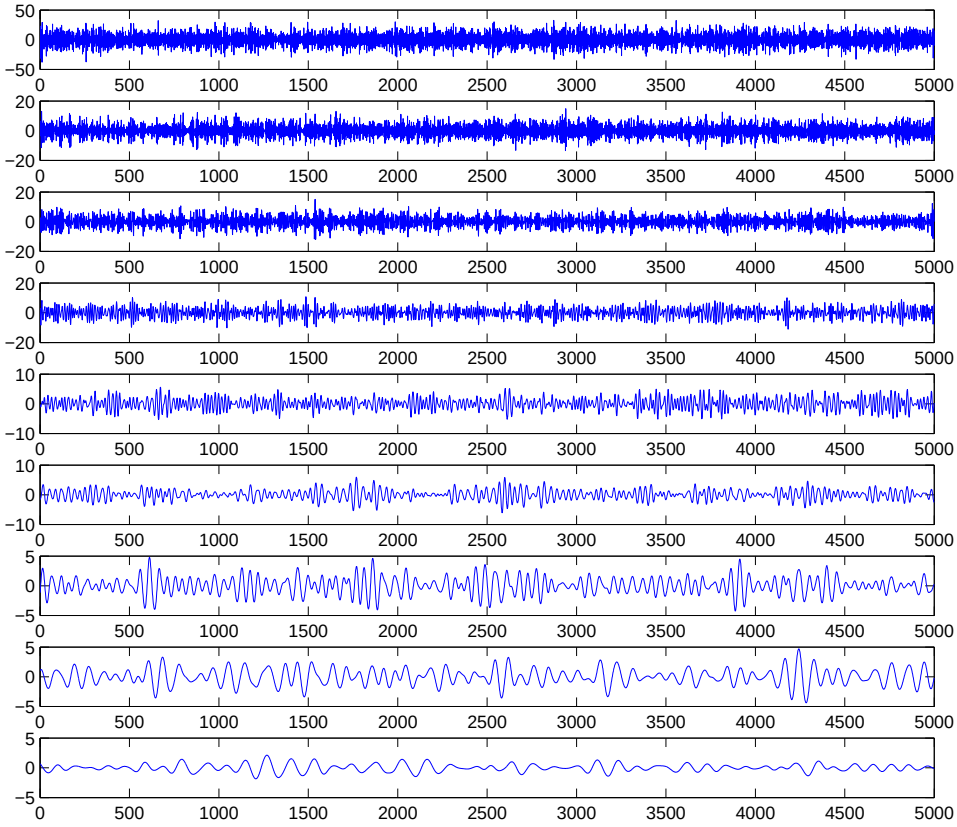
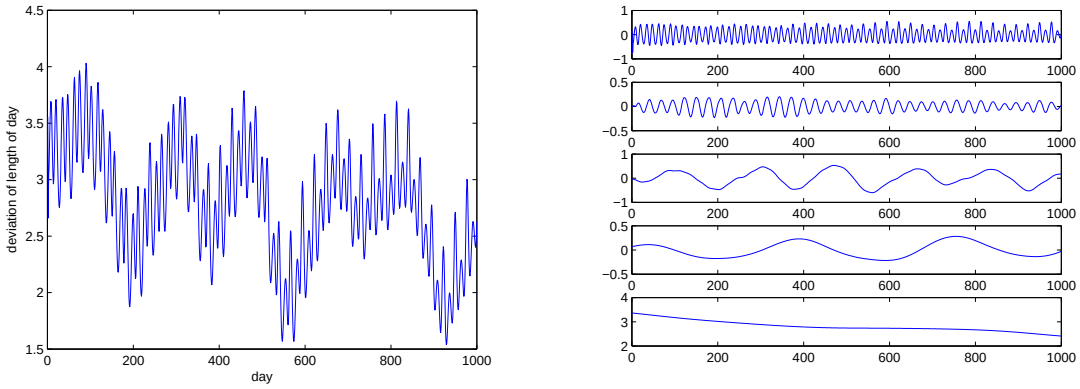


Figure 17: The decomposition of the white noise shown in Figure 15.

Example 4 We apply ALIF algorithm on some real world data. The first one is the deviation of the length of day light data for 1000 days from the year 1973 to the year 1976. This data is decomposed into 5 components as shown in Figure 18 where 4 of them are IMFs and the last one is the trend. From the four IMFs, we can see very regular patterns: the half monthly change pattern, the monthly change pattern, the half yearly change pattern as well as the yearly change pattern.



(a) (b)

Figure 18: Length of day (LOD) signal and its decomposition. (a) the LOD signal; (b) the 5 components in the decomposition.

Example 5 We apply ALIF algorithm to the monthly global ocean temperature anomalies (degrees C) from January 1880 to February 2013 where anomalies are provided as departures from the 20th century average. The data are plotted in Figure 19. In the decomposition of this data, there are several patterns corresponding to temperature rising or dropping at different scales. The trend shows that the temperature does increase for recently 100 years. However, it could be part of a regular pattern at a longer time scale.

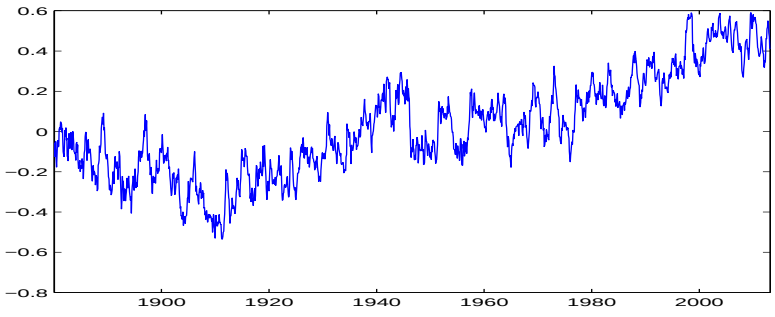


Figure 19: Monthly global ocean temperature anomalies from 1880 to 2013.

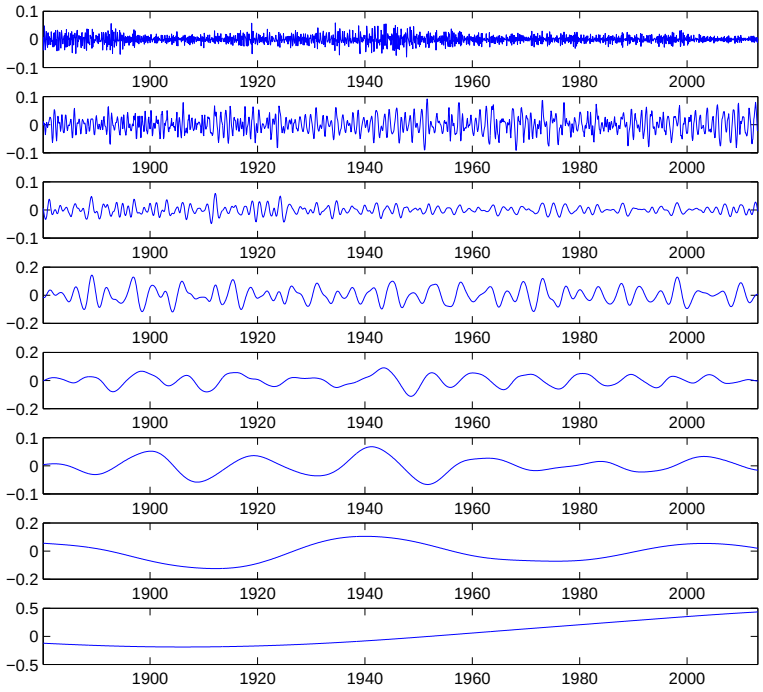
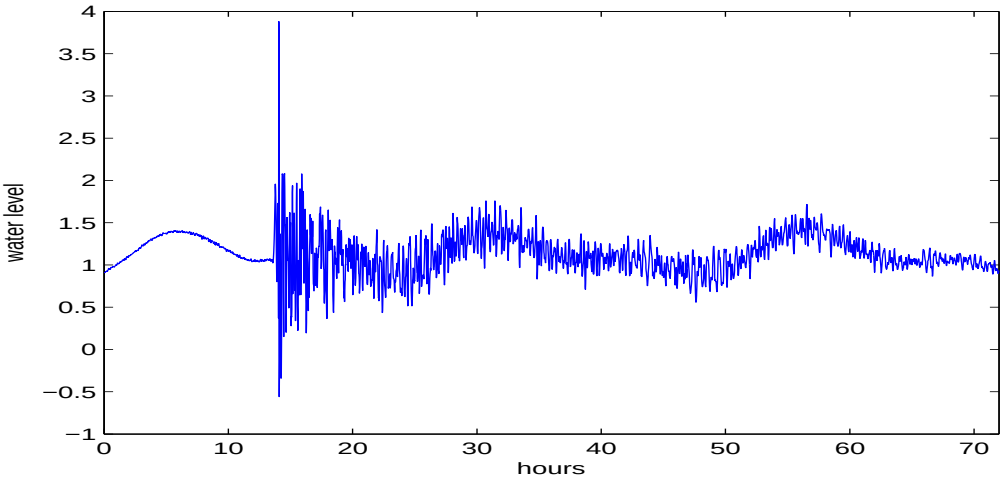
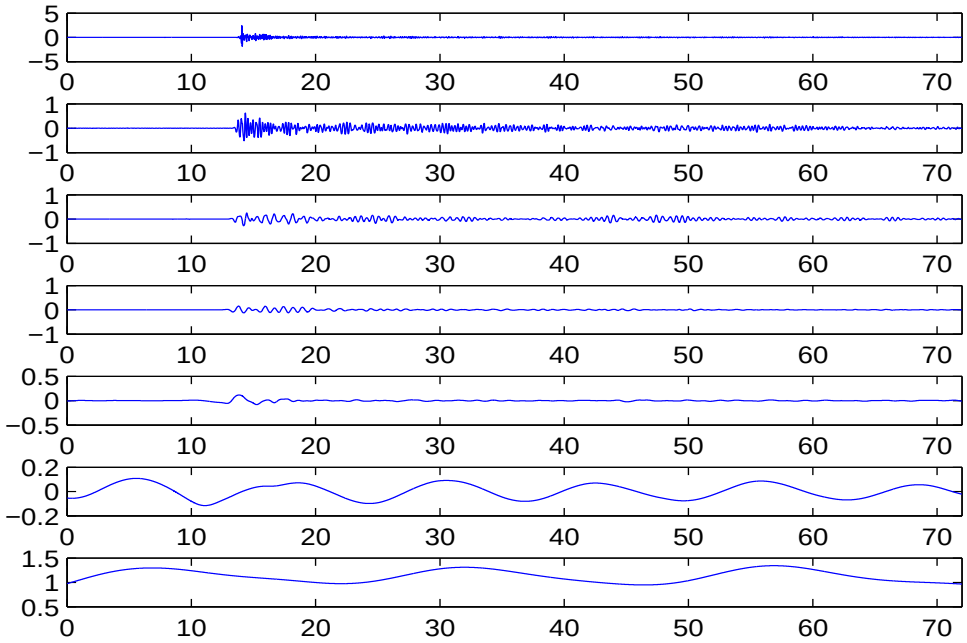


Figure 20: Decomposition of Monthly global ocean temperature anomalies.

Example 6 We apply ALIF algorithm to the water level data observed at Kawaihae, Hawaii, HI for 72 hours from March 11, 2011 to March 13, 2011 when the 2011 Tohoku earthquake and tsunami occurred. The data set of this example could be found in [1]. The data is decomposed into several components where the beginning three components is the impact of the tsunami and the last two components reveal the basic wave height. From figure 21 we see that the algorithm captures transient signals as well as the regular pattern.



(a)



(b)

Figure 21: $(a)$the given wave height signal; (b) the 7 components in the decomposition.

Example 7 We anayze the temperature of certain computer. The data is provided by Dr. Edmond Chow from Georgia Institute of Technology. The temperatures of

the CPU and the motherboard of Chow's desktop were recorded every ten minutes from 2pm Oct 3rd, 2010 to 9pm Feb 25th, 2013. The temperature data are shown in the following:

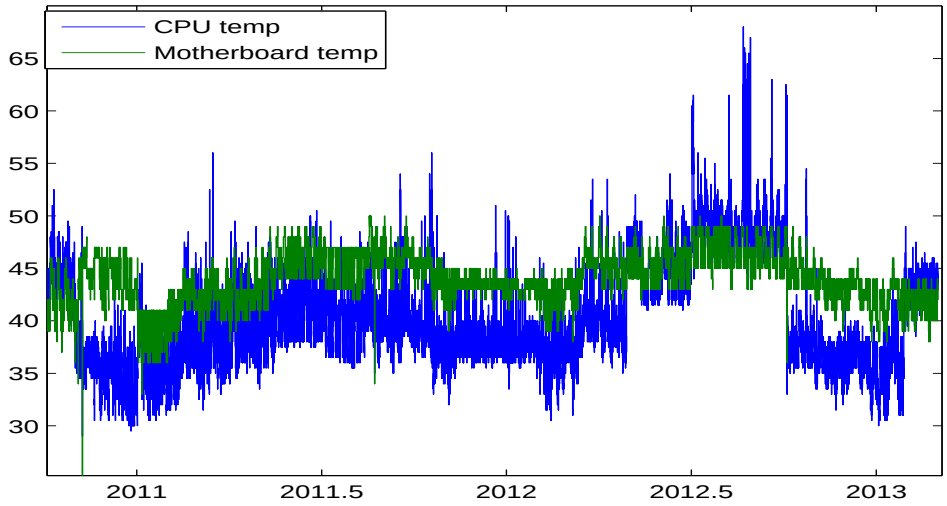


Figure 22: The CPU and motherboard temperature data of Chow's computer from Oct 2010 to Feb 2013.

Take the decomposition of both signals and we get 10 components for the CPU temperature and 12 components for the motherboard temperature.

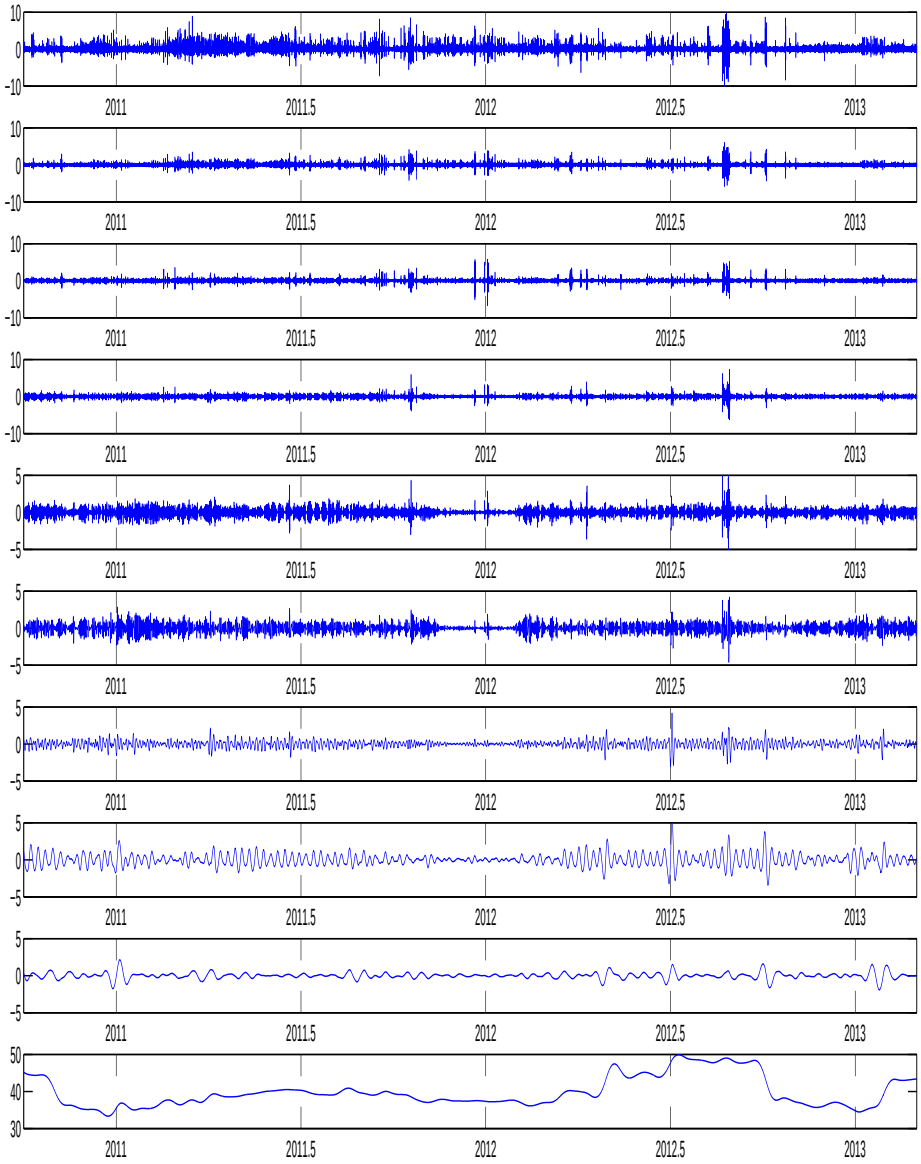


Figure 23: The decomposition result of the CPU temperature data

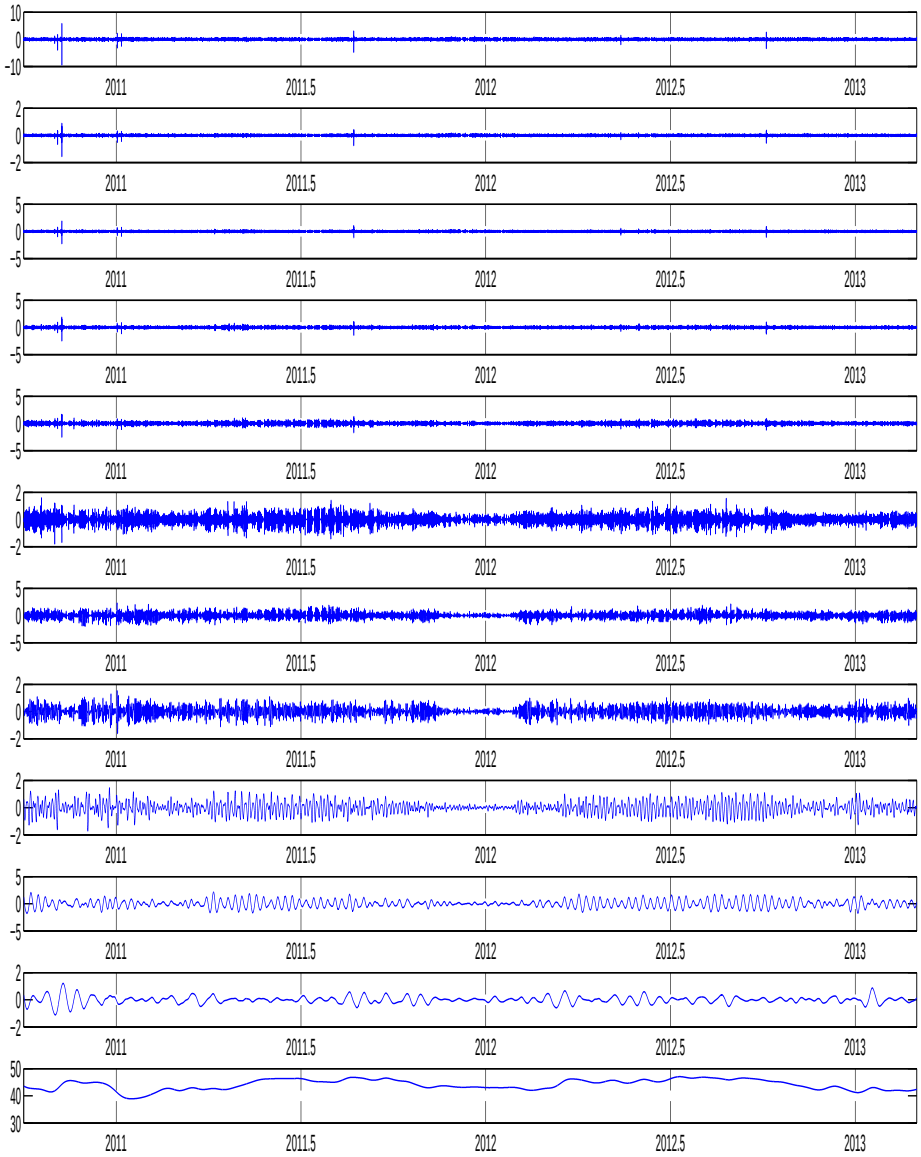


Figure 24: The decomposition result of the motherboard temperature data

We compare the last component and the sum of all the other components of both signals in Figure 25. It is clear that the CPU temperature has larger variation than the motherboard temperature. The trend of CPU is generally lower than the trend

of the motherboard. However, the CPU temperature does have higher trend than the motherboard temperature around June 2012. There might be something abnormal in the cooling system of the CPU during that period. We also derive some regular patterns based on the decomposition. In Figure 26, we plot the 6th and the 8th components of the CPU temperature for 44 consecutive days which are randomly selected. The 6th component of the CPU temperature represents a daily change pattern while the 8th component of the CPU temperature corresponds to a weekly change pattern.

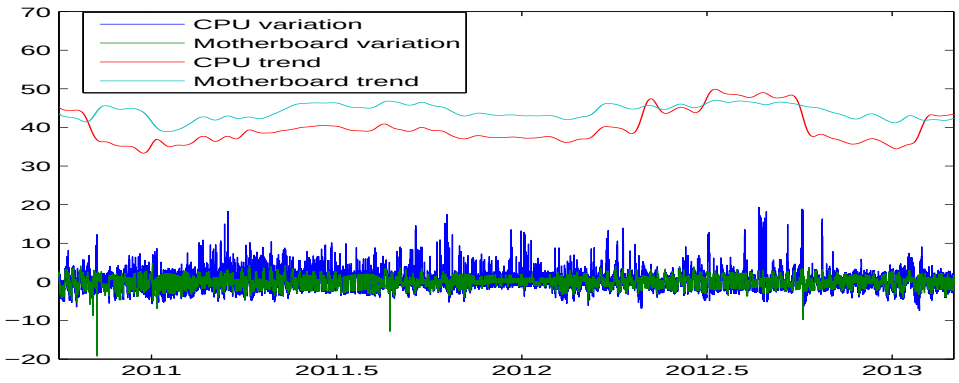


Figure 25: Comparison of the trends and the variation of two signals.

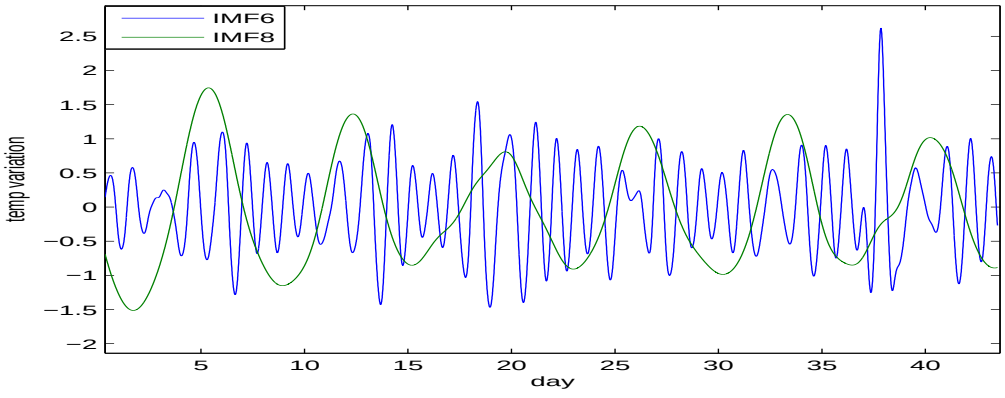


Figure 26: The 6th and 8th components of the CPU temperature. They are corresponding to a daily change pattern and a weekly change pattern respectively.

Example 8 When there are sudden changes in the amplitude of a signal, we use only one side information near this sudden change in ALIF algorithm. Consider the following signal

$$x(t) = f(t) + 0.3\sin(4\pi t) + n(t), t \in [0, 10], \tag{173}$$

where $f(t)$ is a step function

$$f(t) = \begin{cases} -0.5 & \text{if } 0 \leq t < 3.5 \text{ or } 6.75 < t \leq 10, \\ 0.5 & \text{if } 3.5 \leq t \leq 6.75, \\ 0 & \text{otherwise,} \end{cases} \tag{174}$$

where $n(t) \sim N(0, 0.1^2), t \in R$.

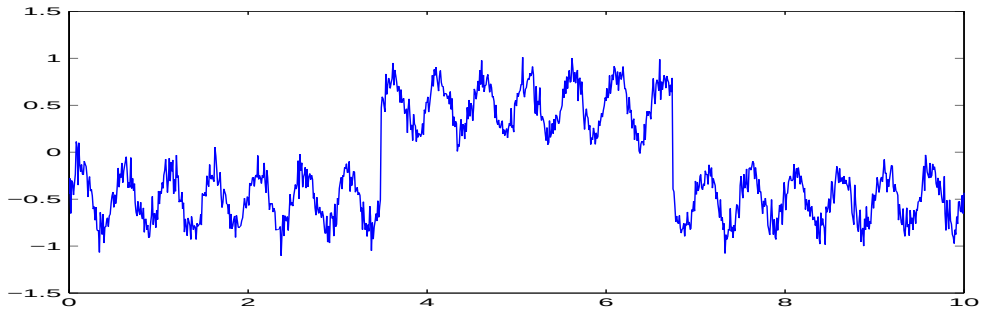


Figure 27: the signal given by (173): there are two sudden changes in the amplitude of this signal.

Using ALIF algorithm with uniform filters, we get the decomposition of $x(t)$ shown in Figure 28. The first 4 components are the reflections of the white noise $n(t)$. The 5th component is the IMF corresponding to the sinusoidal function $0.3\sin(4\pi t)$. The subsequent components are somehow related to the step function $f(t)$. The 7th and the 8th components are mainly related to the sudden changes in the magnitude of $x(t)$ since they have obvious larger amplitude at the same positions where the step function $f(t)$ jumps. The 9th components looks like a smoothed version of the step function $f(t)$. It has the same trend as the step function but loses the transient jumping behaviour. The result is like this since we use the signals behaviours from

both sides near the jumps in ALIF algorithm although the signal are severely non-stationary from one side of the jump to the other side.

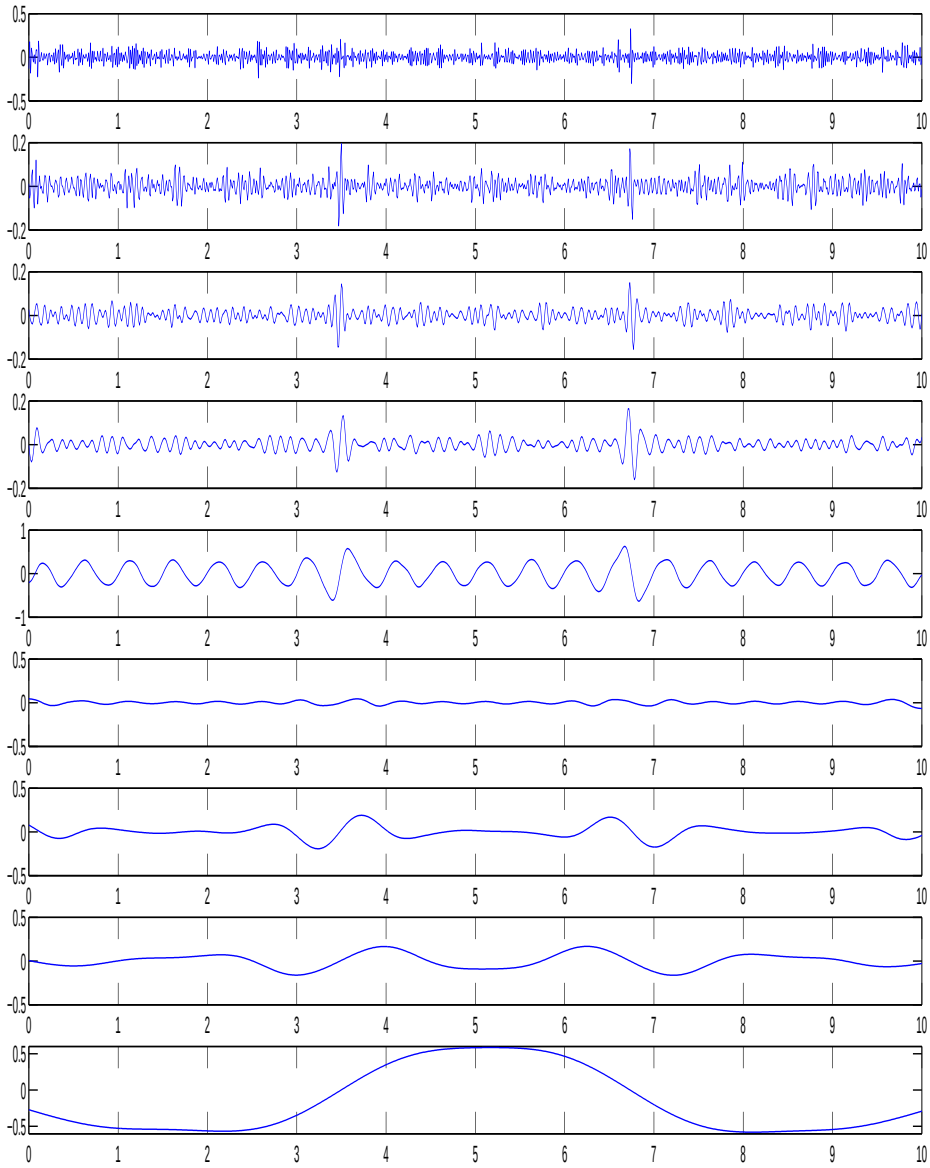


Figure 28: The decomposition of the signal (173) by IF algorithm: the last component lose the representation of the step function.

Using ALIF algorithm with only one side information, we get the decomposition of $x(t)$ shown in Figure 29. Similarly, the first four components are derived due to the impact of noise. The 5th component is corresponding to the sinusoidal function $0.3\sin(4\pi t)$. The subsequent components are related to the step function $f(t)$, where the last component is much more close to a step function compared with the last component in Figure 28.

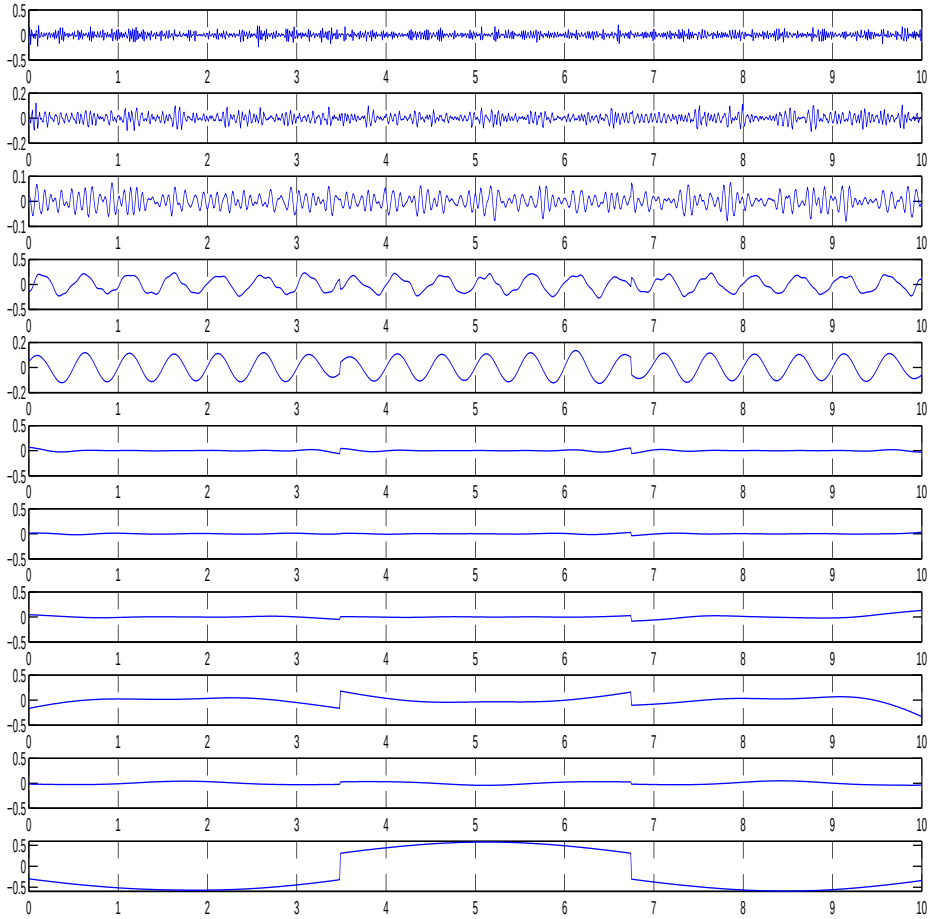


Figure 29: The decomposition of the signal (173) by IF algorithm using only one side information when there are sudden changes in the amplitude: the last component reveals the step function.

3.5 Stability

In this section, we demonstrate the stability of the proposed ALIF algorithm by two examples, one is for simulated data and the other is for real world data.

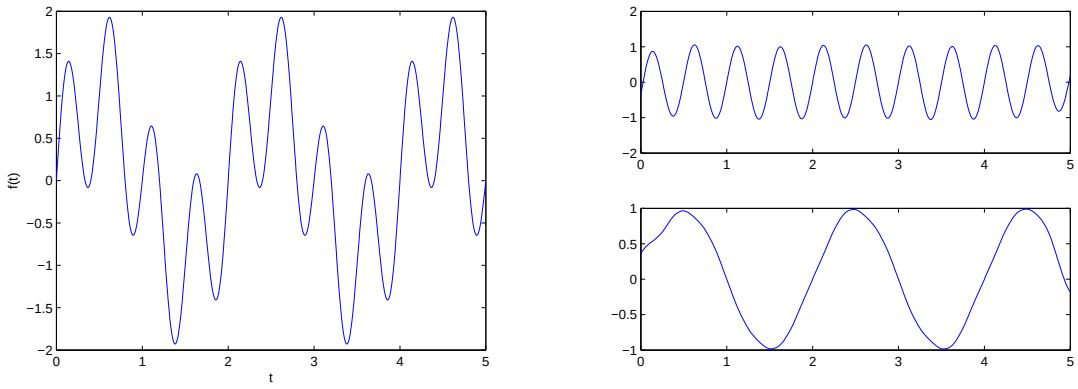
Example 1 We study a signal and its two perturbation with white noise. These signals are given as

$$f_1(t) = \sin \pi t + \sin 4\pi t, t \in [0, 5], \tag{175}$$

$$f_2(t) = \sin \pi t + \sin 4\pi t + n_2(t), t \in [0, 5], \tag{176}$$

$$f_3(t) = \sin \pi t + \sin 4\pi t + n_3(t), t \in [0, 5], \tag{177}$$

where $n_2(t)$ and $n_3(t)$ are the white noise: $n_2(t) \sim N(0, 0.01)$ and $n_3(t) \sim N(0, 1)$ for each t. We apply ALIF algorithm on $f_1(t)$, $f_2(t)$ and $f_3(t)$. From Figure 30, we see that $f_1(t)$ is separated into two IMFs, which correspond to the component $\sin 4\pi t$ and $\sin \pi t$ respectively.



(a) (b)

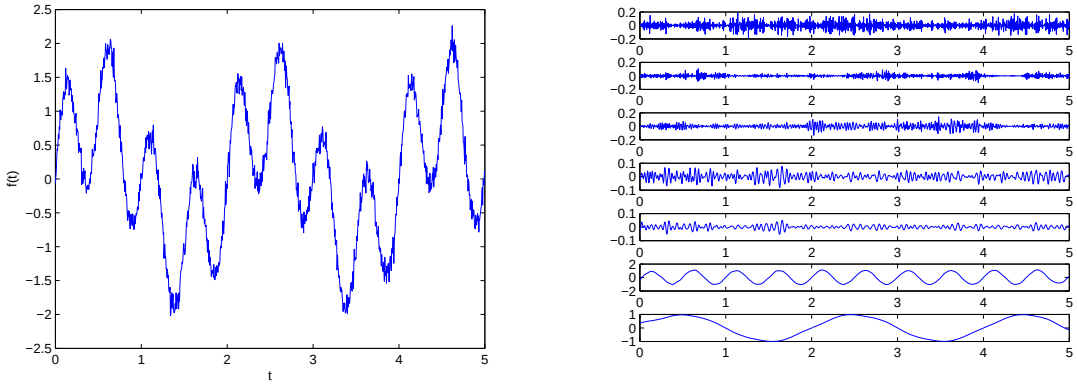
Figure 30: (a) is the signal given in (175); (b) shows all the components in the decomposition.

$f_2(t)$ is decomposed into seven IMFs as shown in Figure 31. The first few IMFs come from the impact of noise and the last two IMFs reveal the two sinusoid functions

$\sin(4\pi t)$ and $\sin(\pi t)$. $f_3(t)$ is decomposed into nine IMFs and only last seven of them are shown in Figure 32. Similar to the result of $f_2(t)$, the first few IMFs come from the impact of noise and the last two IMFs reveal the two sinusoid functions $\sin(4\pi t)$ and $\sin(\pi t)$.

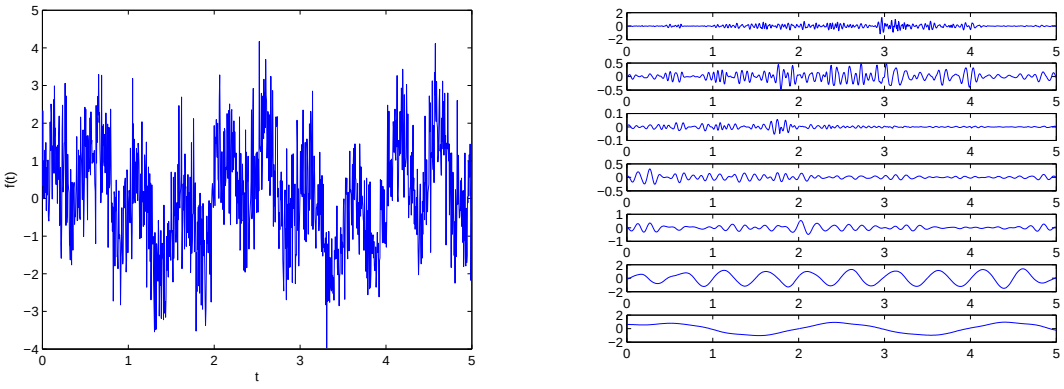
Recall that the decomposition algorithm is said to be stable if the components are consistent in the results of decomposition. In the decomposition results of $f_1(t)$, $f_2(t)$ and $f_3(t)$, there are two components which are consistent. These two components are $\sin(4\pi t)$ and $\sin(\pi t)$. The proposed ALIF algorithm can derive the intrinsic components even if the noise is pretty large like $f_3(t)$. Via this example, the proposed ALIF algorithm is shown to be stable under perturbations which are caused by adding white noise.

Another observation we get from this example is that ALIF algorithm can achieve denoising effect automatically. For signal $f_2(t)$ and $f_3(t)$, we can remove the noise simply by getting rid of the first several components and summing up the subsequent ones.



(a) (b)

Figure 31: (a) is the signal given in (176); (b) shows all the components in the decomposition.



(a) (b)

Figure 32: (a) is the signal given in (177); (b) shows the last seven among nine components in the decomposition.

Example 2 We present another example which can demonstrate the stability of ALIF algorithm. In this example, the data are from the real world. The two signals shown in Figure 33 are troposphere monthly mean temperature inferred from two research groups from Jan 1979 to Dec 2004. We see that these two signals are quite close to each other, i.e. they have almost the same increasing and decreasing patterns except the magnitudes are a little different. So we can regard one signal as a perturbation of the other.

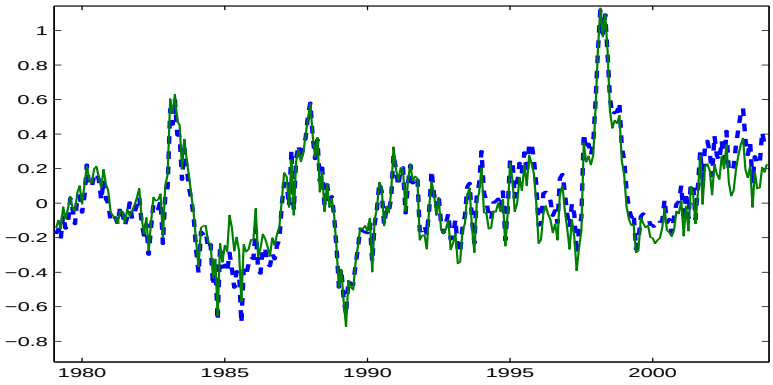


Figure 33: troposphere monthly mean temperature inferred from two research groups

Apply the proposed ALIF algorithm to both signals shown in Figure 33, we get the decompositions for each of them. The results show that ALIF algorithm generates the same number of components for these two signals. We compare the corresponding components in Figure 34. It is clear that each pair of corresponding components are close to each other. This example demonstrates that the proposed ALIF algorithm is stable for real signals under perturbations.

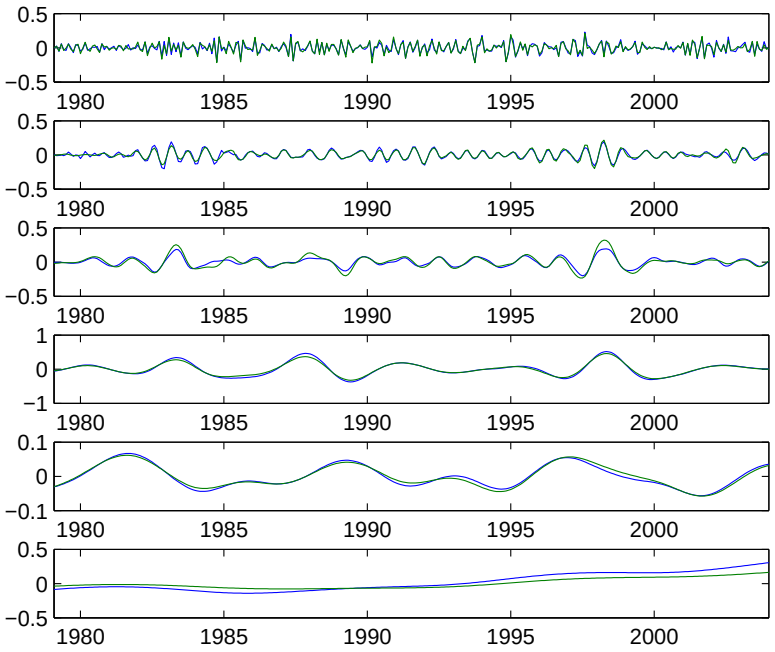


Figure 34: Decompositions of two signals shown in Figure 33: each corresponding components of two signals are small perturbations of each other.

3.6 Conclusions

In this Chapter, we propose ALIF algorithm with the purpose of developing an adaptive, local and stable iterative filtering algorithm. The adaptivity is achieved by adjusting the filter lengths accordingly based on the guidance of the convergence theorem we proved in Chapter 2. The locality is ensured by the filter we designed based on a PDE model. Although we have not established the fundamental about

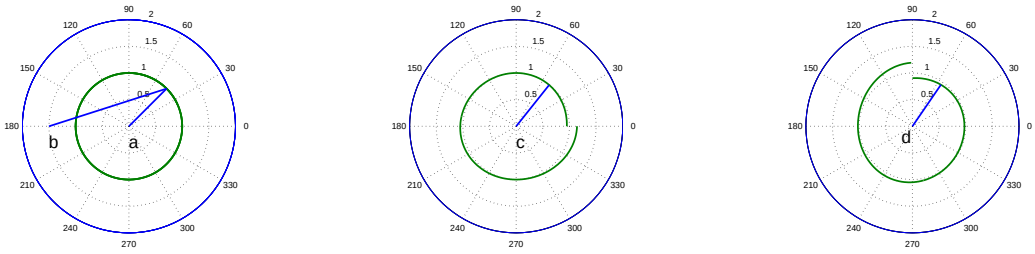
the stability, ALIF algorithm does perform stable decompositions for both simulated and real world signals.

CHAPTER IV

A NEW DEFINITION OF INSTANTANEOUS FREQUENCIES

The instantaneous frequency introduced in Chapter 1 is defined based on Hilbert transform. However, the Hilbert transform is a global operator, which violates the locality aim of the time-frequency analysis. In this section, we present a new definition of the instantaneous frequency which obeys the locality requirement.

Looking back at the two requirements of an IMF, we get that the class of IMFs contain all the sinusoidal functions, but is not limited to these functions. Small perturbations are allowed. To interpret instantaneous frequency, first we analyze a sinusoidal function, then we take the same analysis on a general function satisfying IMF requirements. Let us consider the sinusoidal function $f(t) = \sin t$, the unit circle plotted in Figure 49a (a) represent the signal and its derivative $f'(t) = \cos t$ in $f - f'$ plane. If we stand at or near the center of the unit circle like point a in Figure 35, we can see the point $(f(t), f'(t))$ rotating around where we stand. The angle keeps increasing with time, thus the instantaneous frequency, interpreted as the rate of change of the angle, is positive. However, if we stand outside of this circle, for example point b in Figure 35, we can't see the rotation of $(f(t), f'(t))$, which makes the angle increase and decrease alternately. As a result, the instantaneous frequency is no more only non-negative. Consider two signals which are the perturbation of $\sin t$, their phase angles are also the perturbations of the unit circle as shown in Figure 35. Standing at the center of the unit circle as point c or point d in Figure 35, the rotating is visible and thus the instantaneous frequency is well-behaved although it is not constant any more.



(a) (b) (c)

Figure 35: phase angles: (a) is the phase angle of $\sin t$. Its instantaneous frequency is a constant function. (b) shows the phase angle of $\sin\left((0.95 + 0.05t/\pi)t\right)$. Its instantaneous frequency is a non-constant function. (c) shows that phase angle of $(0.95 + 0.05t/\pi)\sin\left((0.95 + 0.05t/\pi)t\right)$. Its instantaneous frequency is also a non-constant function.

4.1 2D ODEs In Polar Coordinates

The phenomena we observe in Figure 35 gives a hint of 2 dimensional ordinary differential equations (ODE) in the polar coordinates, where the frequency is corresponding to one equation in the system. Consider the following linear ODE:

$$\begin{cases} \dot{x} = \cos t, \\ \dot{y} = -\sin t. \end{cases} \tag{178}$$

By changing the coordinate from (x, y) to (r, θ) by

$$x = r\cos\theta, \quad y = -r\sin\theta. \tag{179}$$

(178) can be written as

$$\begin{cases} \dot{\theta} = 1, \\ \dot{r} = 0. \end{cases} \tag{180}$$

in (180), the frequency of this ODE system is given directly as $\dot{\theta}$ in the first equation. As $\dot{\theta}$ is a constant, the frequency is a constant, which is consistent with the intuition.

Let us consider a non-linear ODE: the van der Pol oscillator given as

$$\ddot{x} + \alpha(x^2 - 1)\dot{x} + x = 0, \alpha > 0. \tag{181}$$

The standard first-order form of the van der Pol equation is

$$\begin{cases} \dot{x} = y, \\ \dot{y} = -x - \alpha(x^2 - 1)y. \end{cases} \tag{182}$$

Using polar coordinates $x = r\cos\theta$, $y = -r\sin\theta$, (182) can be written as

$$\begin{cases} \dot{r} = -\alpha(r^2\cos^2\theta - 1)r\sin^2\theta, \\ \dot{\theta} = 1 + \alpha(r^2\cos^2\theta - 1)r\sin\theta\cos\theta. \end{cases} \tag{183}$$

The equation of motion for the phase angle is the second one in (183). The frequency $\dot{\theta}$ is not a constant any more. When $\alpha << 1$, $\dot{\theta}$ is positive, which is corresponding to a rotation.

Let $f(t)$ be an IMF. It represents some pattern of oscillations. Treat it as the x coordinate of some second order ODEs, we can get the frequency naturally as the derivative of the phase angle in the polar coordinate. It is not necessary to derive the equations for such an ODE since we only use it implicitly. Instead, we first get rid of the impact of r by mapping $f(t)$ to a perturbation of the unit circle and then derive the frequency based on it. Although the perturbation is not a perfect unit circle, we can still see its rotation standing at the center of the unit circle like in Figure 35. Thus the angle $\theta(t)$ is monotonously increasing and the instantaneous frequency $w(t)$ is a positive function if the rotation is counter clockwise; the angle $\theta(t)$ is monotonously decreasing and the instantaneous frequency $w(t)$ is a negative function if the rotation is clockwise. As long as the instantaneous frequency does not change its sign, it is well behaved.

4.2 The New Definition Of Instantaneous Frequencies

Based on the analysis in Section 4.1, we propose a new definition of the instantaneous frequency for IMFs. Let $f(t)$ be a function satisfying IMF requirements, there exists an envelope function $g(t)$ of $f(t)$ such that

$$F_1(t) := f(t)/g(t) \in [-1, 1]. \tag{184}$$

Taking derivative of $f(t)$, there exists an envelope function $h(t)$ of $f'(t)$ such that

$$F_2(t) := f'(t)/h(t) \in [-1, 1]. \tag{185}$$

If we define

$$F(t) = F_1(t) + iF_2(t), \tag{186}$$

then $F(t)$ corresponds to a curve in $[-1, 1] \times [-1, 1]$ on the complex plane. $F(t)$ is a perturbation of the unit circle and we define the angle for the rotation of $F(t)$ as

$$\theta(t) = -\arctan \frac{F_2(t)}{F_1(t)}, \tag{187}$$

and the instantaneous frequency for $f(t)$ as

$$w(t) = \frac{d\theta(t)}{dt}. \tag{188}$$

.

4.3 Numerical Experiments Of Instantaneous Frequencies

The computation of the instantaneous frequency follows the new definition. We first normalize the function $f(t)$ and its derivative $f'(t)$ by their envelopes $g(t)$ and $h(t)$ respectively. The envelopes functions $g(t)$ and $h(t)$ can be simply taken as the cubic splines connecting the local extrema in $f(t)$ or $f'(t)$ respectively. Then the phase angle is computed by (187) and the instantaneous frequency is obtained by taking the derivative of the phase angle.

4.3.1 The Computation With One Side Information

The magnitude of an IMF sometimes changes significantly in a short time. Due to the impact of this kind of behaviours, the envelopes constructed for the IMF are not following the change of the IMF closely, which may cause unexpected errors in the instantaneous frequency. So it is nice that the algorithm is able to catch these sudden changes when constructing the envelopes so that they are accurate enough.

To get such a technique, we make use of only one side information based on the essentially-non-oscillatory (ENO) idea.

The technique of dealing with the sudden changes are based on two steps. The first step is to detect the change in the magnitude and the second step is to find the point for whose left hand side and right hand side should be treated separately in envelope construction. Based on the difference of consecutive extrema, we tell if there is a sudden change. Let $t_i, i \in \mathbb{N}$ be the local extreme points of the IMF $f(t)$. Check the values of

$$\left| \quad |f(t_{i+1})| - |f(t_i)| \quad \right|, \quad i \in \mathbb{N}. \tag{189}$$

If (189) is greater than a preselect threshold for some i, we realize the magnitude of $f(t)$ does have a sudden change from t_i to t_{i+1}. Assume that there is only one sudden change in the magnitude from t_i to t_{i+1}. For each point between t_i and t_{i+1}, we compute the difference of the $f(t)$ for consecutive two sample points. The sudden change is regarded to happen at the point where the left difference differs most from the right difference.

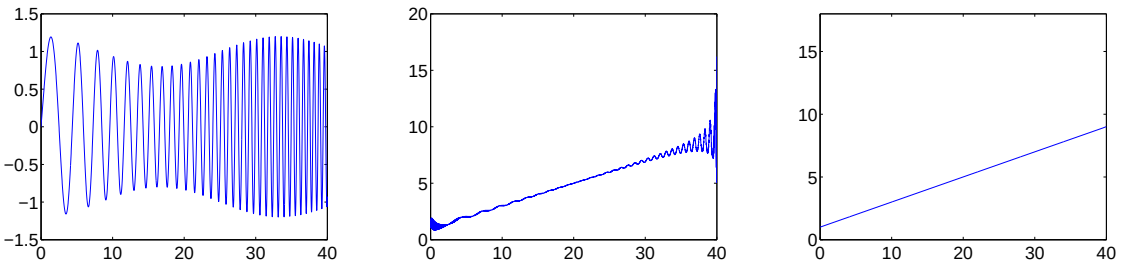
4.3.2 Numerical Examples

Let us consider the instantaneous frequencies of two examples using both the proposed local definition and the definition based on Hilbert transform. Both of these two examples are simulated signals.

Example 1 In Figure 36, the signal is given by

$$f(t) = (1 + 0.2\cos(0.06\pi t))\sin[(1 + 0.1t)t], \quad t \in [0, 40]. \tag{190}$$

The amplitude of the signal changes slowly, both frequency analysis methods show the gradual change in the instantaneous frequency as shown in figure 36. However, when there is a significant change in the amplitude of the signal, instantaneous frequencies defined by two approaches are different.



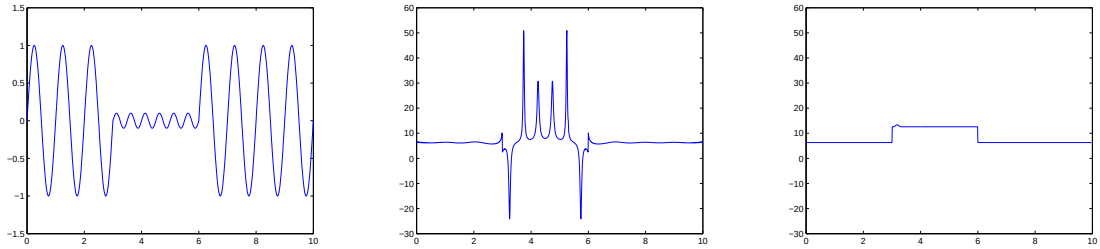
(a) (b) (c)

Figure 36: Example 1: (a) the signal defined in (190). The oscillation gradually becomes faster and the amplitude changes mildly. (b) instantaneous frequency computed using Hilbert transform. It shows the gradual change in the instantaneous frequency but has some oscillations. (c) instantaneous frequency computed using the proposed method. It also shows the gradual change in the instantaneous frequency and almost has no oscillations.

Example 2 In Figure 37 the signal is generated by

$$f(t) = \begin{cases} \sin(2\pi t) & \text{if } 0 \le t < 3 \\ 0.1\sin(4\pi t) & \text{if } 3 \le t \le 6 \\ \sin(2\pi t) & \text{if } 6 < t \le 10 \end{cases} \tag{191}$$

It has sudden changes in both the amplitude and the frequency at time 3 and 6. We expect that the time frequency analysis gives us the instantaneous frequency which is almost a step function with jumps at 3 and 6. Using the instantaneous frequency definition based on Hilbert transform, the transitory change in the amplitude affects faraway positions by Hilbert transform and this leads to strange behaviours in the instantaneous frequency. On the other hand, using the proposed method with adopting one side information, we get an instantaneous frequency which is almost a step function.



(a) (b) (c)

Figure 37: Example 2: (a) the signal defined in (191) . There are sudden changes in the amplitude. (b) instantaneous frequency computed using Hilbert transform; the instantaneous frequency is inconsistent with people's expectation. (c) instantaneous frequency computed using the proposed method. The instantaneous frequency is almost a step function as we expected.

CHAPTER V

APPLICATIONS

Besides the development in decomposition algorithms inspired by EMD, people have implemented EMD on different kinds of applications such as signal analysis, image processing, damage detection, health monitoring, climate change and so on. EMD was applied to seismic traces in [90], to ECG signals in [9] and to EMG signals in [2] for frequency attributes of those signals. Hariharan, Guan, Nunes each developed bi-dimensional EMD and performed it on image processing and texture analysis in [40], [38] and [100]. EMD was also applied to the Nile annual records, stratospheric data, atmospheric wave field, long daily ozone records and rainfall data in [109], [21],[95], [61], [145] and [112] . The decomposition results revealed certain climate cycles. EMD can also be used to detect the damages in gear systems as shown in [87], [84], [35], and [101] or to monitor health and detect disease as shown in [103], [30] and [99].

In this chapter, we perform ALIF algorithm to two different kinds of applications. One is a real world application to detect certain chemical in a plume; the other one is about the relations between the iterative filtering and ordinary differential equations.

5.1 Application On The Gas Detection

Advances in hardware provides the capability to detect possible chemical or biological substance in the air. To establish the dataset for detecting a certain substance, different experiments were designed and carried out. A a result, many hyper spectrum data are collected from these experiments. These substance may come from an unexpected explosion. They spread out into the air in a very short time and cause harmful impact on people or the environment. If these substance can be detected after the release, corresponding techniques can be taken out to prevent such noxious

spreading. So the detection algorithm plays an important role in this problem.

Recently Dimitris Manolakis in MIT Lincoln Laboratory developed a gas detection method for the hyper spectrum data. The data that the detection algorithm deals with is a hyper spectrum cubic block. For each frequency band with the hyper spectrum, the data are the records of the signals reflected from the air for a fixed region. The detection is based on the similarity of the spectrum of the collected data and the spectrum of the substance aimed to detect. The similarity is computed by

$$y = \frac{[s^T \Sigma_b^{-1}(x - \mu_b)]^2}{(s^T \Sigma_b^{-1} s) \cdot (x - \mu_b)^T \Sigma_b^{-1}, (x - \mu_b)} \tag{192}$$

where μ_b is the mean vector of the data over different spectrum frequency; Σ_b is the covariance matrix cross different frequencies and s is the standard spectrum of the substance aimed to detect. Assume certain substance exists in the plume, the spectrum of it is expected to be derived by removing the mean from the data and then normalizing via the covariance matrix. Compute the angle between the vectors representing the derived spectrum and the standard spectrum of this substance, the similarity is obtained and can be used further to infer the existence of this substance. Based on the similarity values and the experiment dataset, the performance of the detection algorithm can be evaluated by a receiver-operator-characteristic(ROC) curve described in [32]. The ROC curve is a graphical plot which shows the performance of a binary classifier. It plots the fraction of true positive out of the number of positives vs. the fraction of false positive out of the number of negatives at different thresholds. When the threshold is set such that the numbers of both true positive and false positive are zero, the plotting for this threshold is at $(0, 0)$. On the other hand, when the threshold is set such that all instances are classified as positives, the number of true positives is the same as the number of positives and the number of negative positives is the same as the negatives. So the plotting for this threshold is at $(1, 1)$. So an ROC curve must go from $(0, 0)$ to $(1, 1)$. The classifier using random guess has the ROC curve as the linear function connecting $(0, 0)$ and $(1, 1)$. The larger area under

the ROC curve, the better performance the classifier has.

Let us present one example of the detection method proposed by Dimitris Manolakis. For one dataset, the ROC curve of the detection by (192) is plotted in Figure 38. When plotting the ROC curve, the x axis is rescale by taking logarithm. The solid curve is the ROC curve for the detection method where the similarity is computed by (192). The dashed curve is the ROC curve for the detection by random guess. The dashed curve is the off-diagonal line in the same kind of plot without taking the logarithm of the x axis. It is clear that the solid curve is high above the dashed curve, which means the detection algorithm has much better performance than random guess.

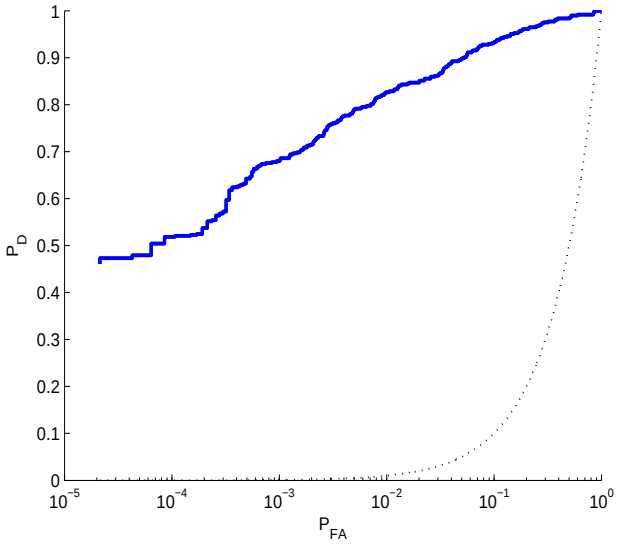


Figure 38: The ROC curve of the detection method (192).

We propose a different detection method by using ALIF algorithm. In this new detection method, we compute several similarity values and take the maximal one among them. We compute four different similarity values described in the following:

- y_1. Take the image for each frequency band in the dataset. Apply the decomposition for each horizontal vector in this image, get rid of the first IMF,

replace the vector by summing up the subsequent components. The reason of taking such a step comes from two aspects. One is that the data usually contain noise and getting rid of the first IMF can remove the noise efficiently. The other aspect is to include the spatial correlation. Since the substance region is continuous, for a position in the substance region, it is highly possible that its neighbourhood is also in the substance region. By summing up the subsequent components, each point of the substitute data is impacted by its horizontal neighbourhood. The spatial correlation is involved to a certain extent. Then y_1 is computed by (192) with x replaced.

- y_2. Process the dataset in a similar way as in y_1. Apply the decomposition for each vertical vector in the image for each frequency band. Get rid of the first IMF and sum up the remaining components. Compute y_2 by (192) where x is replaced.

- y_3. Take the image for each frequency band in the dataset. Apply the decomposition for each horizontal vector, replace it by getting rid of the first IMF; apply the decomposition again for each vertical vector, replace it by getting rid of the first IMF. Then compute y_3 by (192) with x replaced. In this detection, both the horizontal and vertical spatial correlations are involved.

- y_4. We consider removing the spectrum of the background from the dataset. Then the remaining part is the spectrum corresponding to the substance. First compute the mean of each frequency band of the dataset. This is the spectrum of the background. Second for each position in the region, there is a vector over different frequencies. Subtract the background from this vector and the difference D is derived. We make use of the decomposition method and get the components for D. Remove the trend of D and it is the derived spectrum after removing the background. Compute the cosine similarity between the derived

spectrum and the standard spectrum of the substance. This is y_4.

After compute these four different similarities, take the maximal one among them. Let

$$y = \max\{y_1, y_2, y_3, y_4\}. \tag{193}$$

Evaluate the proposed detection algorithm by the ROC curve for the same dataset we used in Figure 38. The performance of the proposed detection algorithm is plotted in Figure 39. The thin solid curve is the ROC curve of the detection method proposed by Dimitris Manolakis and the thick solid curve is the ROC curve of the new proposed detection method. It is clear the latter one has better accuracy.

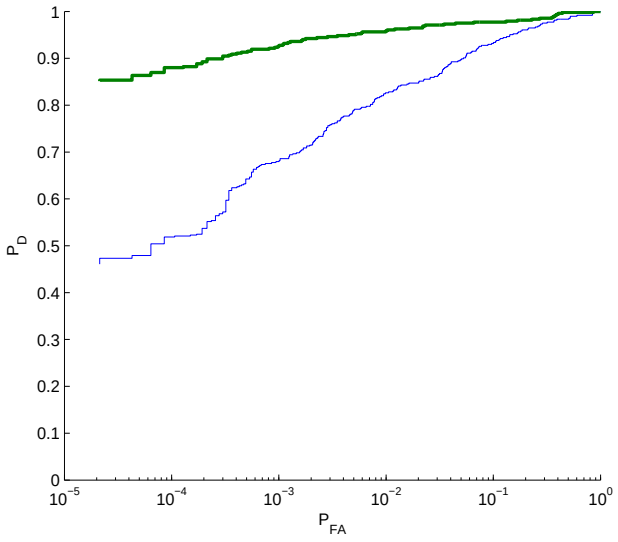


Figure 39: The ROC curve of the detection method (193).

The proposed detection method could be improved by an more accurate background. If we have the knowledge of the possible position where the substance is, we can modify the background spectrum by taking the mean only restricted to a surrounding region of these possible positions. Let y_5 be the similarity computed following this background. Then the similarity is take as

$$y = \max\{y_1, y_2, y_3, y_5\}. \tag{194}$$

Evaluate the proposed detection algorithm (194) for the same dataset. The performance of the detection methods (193) and (194) are plotted in Figure 40. The thin solid curve is the ROC curve of the detection method (193) and the thick solid curve is the ROC curve of the detection method (194). It is clear the (194) has better performance than (193). So with an accurate background, the detection algorithm is even more improved.

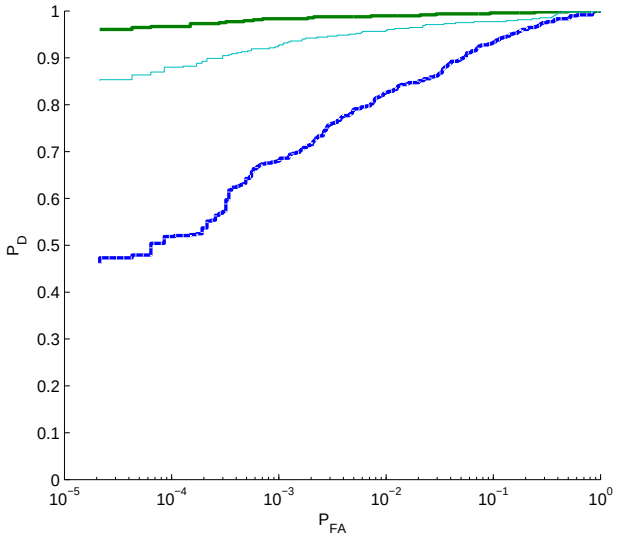


Figure 40: The ROC curve of the detection method (194).

5.2 Relations With Ordinary Differential Equations

Based on the numerical examples from Chapter 3 and the examples from Section 5.1, we can see that ALIF algorithm has good performance on decomposing signals and dealing with real world applications. In this section, we consider the reason of the good performance of ALIF algorithm.

Since ALIF algorithm is really adaptive and data dependent, each component derived from a given signal is also dependent on the signal itself. Each component is an oscillatory signal and we do not have other prior knowledge of these components. It is well known that the solutions of some second order ordinary differential equations

are oscillations. If a signal is generated from such a system, ALIF algorithm may reveal the intrinsic characters of the system.

Consider the duffing equation

$$\ddot{x} - x + x^3 = f, \tag{195}$$

with the initial condition

$$\begin{cases} x(0) = 1.1, \\ \dot{x}(0) = 0. \end{cases} \tag{196}$$

We solve this initial value problem numerically with two different f functions $f_1(t)$ and $f_2(t)$ where

$$f_1(t) = 0, \tag{197}$$

$$f_2(t) = 2\sin(25t). \tag{198}$$

When $f = f_1(t)$, the solution x_1 of the initial value problem is an single oscillation as shown in Figure 41.

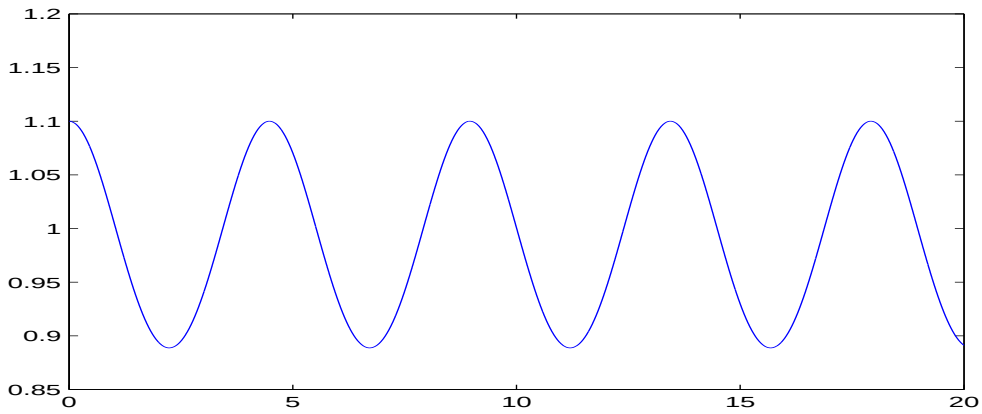


Figure 41: The solution of (195) with the initial condition (196) with $f(t) = 0$.

When $f = f_2(t)$, the solution x_2 of the initial value problem is a function where an oscillation is coupled with a faster oscillation as shown in Figure 42.

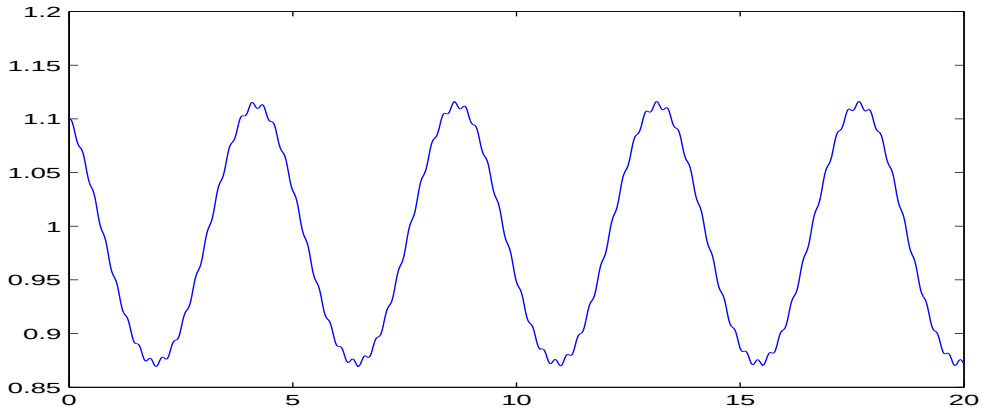


Figure 42: The solution of (195) with the initial condition (196) with $f(t) = 2\sin(25t)$.

The solution x_1 is a shift of an IMF since it is oscillatory and symmetric with respect to a constant around 1. The solution x_2 is not a shift of an IMF any more since it contains more than one oscillation modes. Take the decomposition of x_2, we get following result:

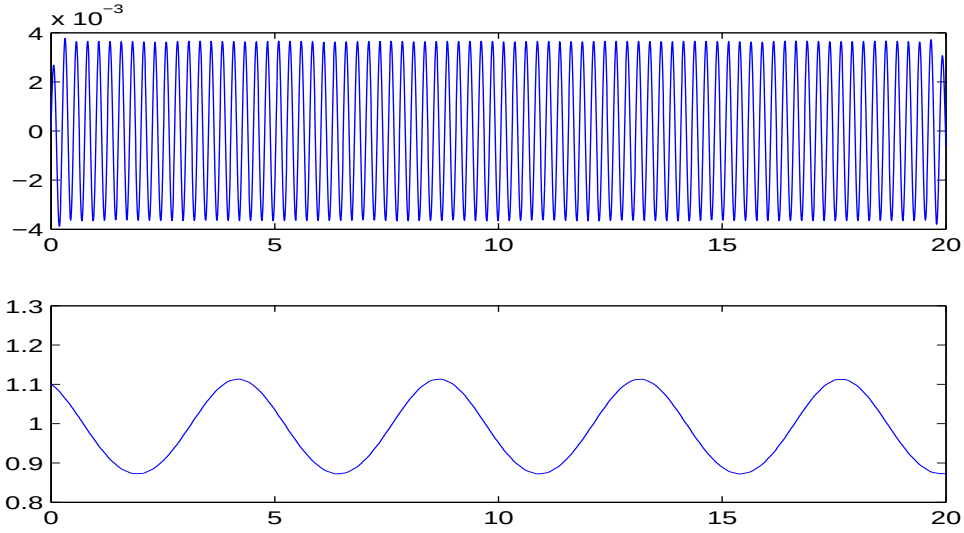
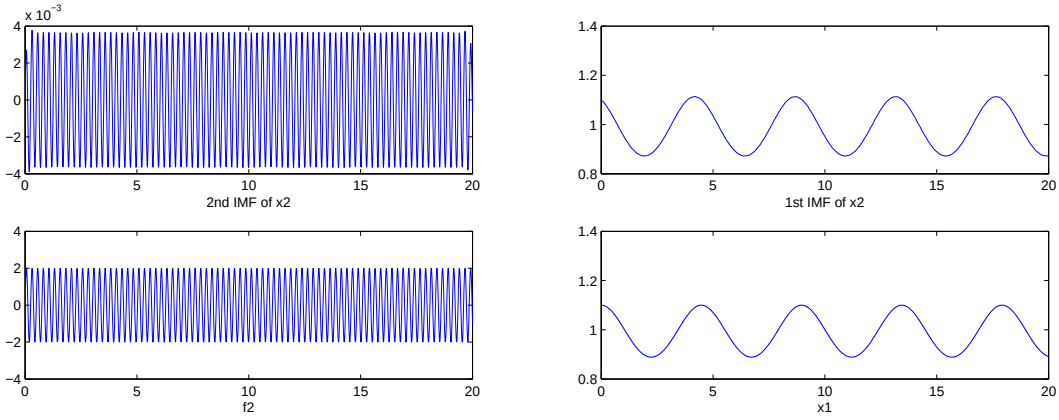


Figure 43: two components of the decomposition of $x2$.

From Figure 43, we see that ALIF algorithm generates two components for the solution x_2, one has faster oscillation and the other has slow oscillations. Both of

these two components are quite regular. Then we compare these two components with x_1 and the forcing term focusing on their oscillation modes. In Figure 44, we plot the comparison between two component of x_2, the forcing function $f_2(t)$ and the solution x_1. It is clear that the first component of x_2 is due to the impact of the nonzero forcing function $f_2(t)$ while the second component of x_2 reveals the same oscillation mode as the solution x_1.



(a) (b)

Figure 44: The comparison between the two components of x_2 and $f_2(t)$, x_1: (a) the comparison between the first component of x_2 with $f_2(t)$; (b) the comparison between the second component of x_2 with x_1.

The significance of this example is that for certain dynamical systems with periodic solutions, ALIF algorithm reveals the intrinsic oscillation modes of it. It includes the intrinsic mode of the dynamical system without any external forcing as well as the intrinsic mode of a faster periodic forcing term. In fact, not only ALIF can do this, EMD and IF algorithm both can achieve the same results. Note that the forcing function $f_2(t)$ has faster oscillation than the intrinsic mode x_1. When the frequency of the forcing function reduces, resonance will occur. We shall illustrate this point by an example of the linear ordinary differential equations. Let us consider the linear

equation

$$\ddot{x} + x = f \tag{199}$$

with the initial condition

$$x(0) = 2 \quad , \quad \dot{x} = 0. \tag{200}$$

When $f(t) = 0$, we get the solution x_0 as $2\cos t$. When f is given as a signal shown in Figure 45, the solution x for the same initial value problem is plotted in Figure 46. It is clear that the magnitude of x mainly increases twice significantly: one growth is located around from 80 to 90 and the other is right after 150.

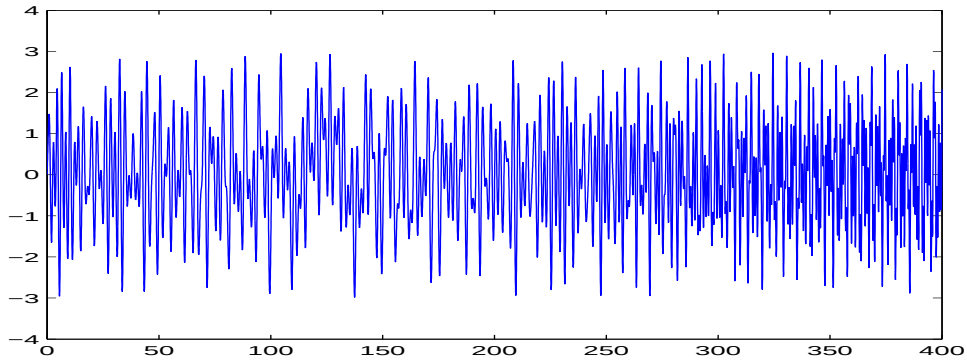


Figure 45: the forcing function in (199)

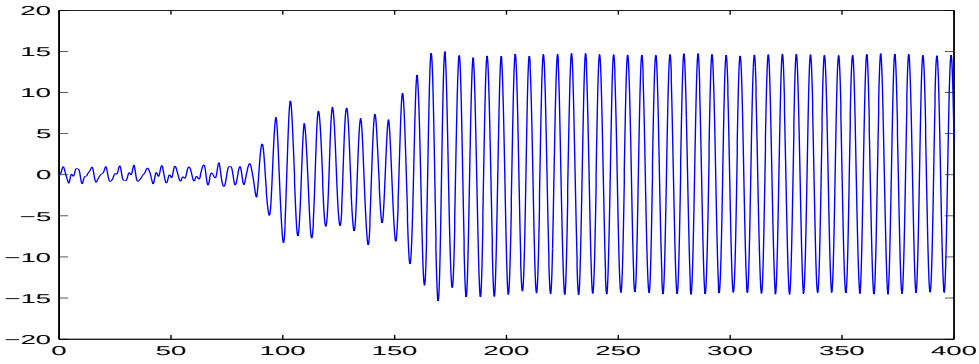
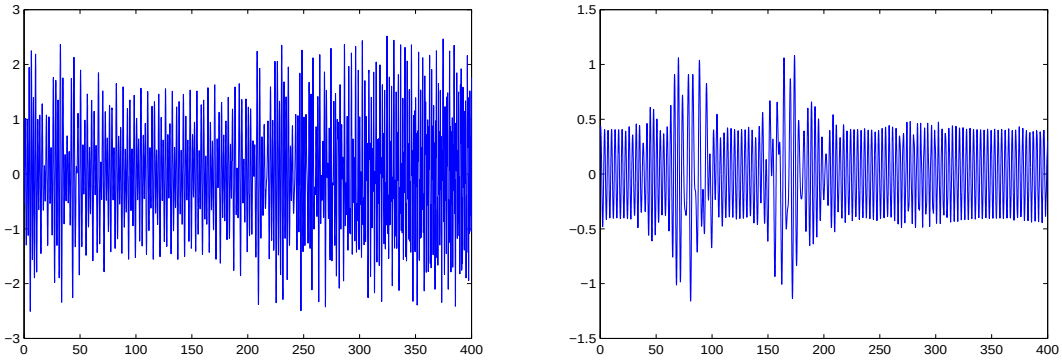


Figure 46: the solution of (199) with $f(t)$ given in Figure 45.

We take the decomposition of the function $f(t)$ and resolve the linear equation

(199) with different initial conditions. The aim is to see if the decomposition helps in understanding of the behaviour of x. Let us consider the first two components plotted in Figure 48 in the decomposition. It is clear that the first component has larger amplitude than the second component. Let $f_1(t)$ be the first component and $f_2(t)$ be the second component.



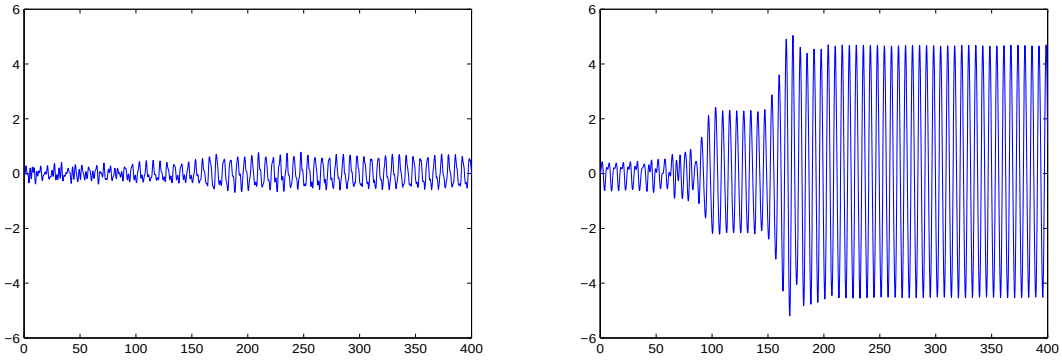
(a) the first component (b) the second component

Figure 47: The first two components in the decomposition of x.

Solve the linear equation (199) with the forcing function $f = f_1(t)$ and $f = f_2(t)$ and the initial condition given in the following

$$x(0) = 1 \quad , \quad \dot{x} = 0. \tag{201}$$

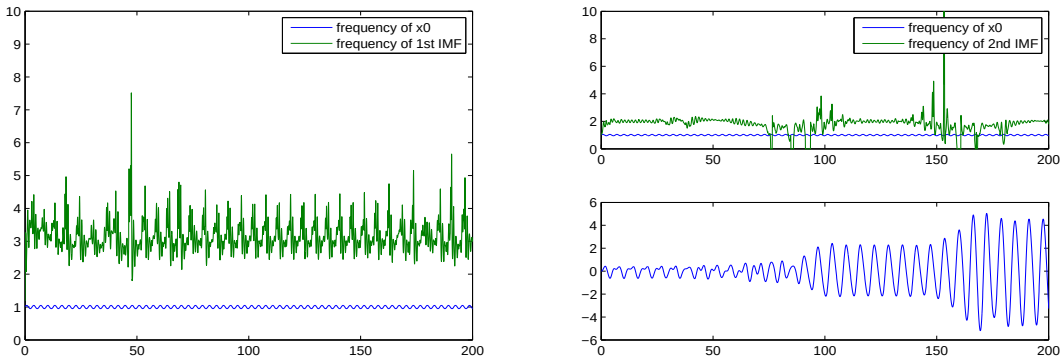
The respective solutions x_1 and x_2 are plotted in Figure 49. The amplitude of the solution $x_1(t)$ has almost no change from 0 to 400 while the amplitude of the solution $x_2(t)$ has two significant changes at the same positions where the magnitude changes in x. Although $f_1(t)$ has larger magnitude than $f_2(t)$, it does not lead to a large magnitude in the solution $x_1(t)$. So the large amplitude in the forcing function does not necessarily generate the significant changes in the amplitude of the solution.



(a) x_1 (b) x_2

Figure 48: The solutions x_1 and x_2 for (199) with $f_1(t)$ and $f_2(t)$ respectively.

We take the instantaneous frequencies of $f_1(t)$ and $f_2(t)$ into account as shown in Figure 49 where we only plot the time interval from 0 to 200. The instantaneous frequency of $f_1(t)$ is higher than the instantaneous frequency of $x_0(t)$. So their instantaneous frequencies does not intersect. On the other side, the instantaneous frequency of $f_2(t)$ intersects with the instantaneous frequency of $x_0(t)$ occasionally. What is more important, these intersections occur at the positions where the magnitude of $x_2(t)$ increases significantly. The growth in the magnitude is due to the intersections of the instantaneous frequencies.



(a) frequency of $f_1(t)$ (b) frequency of $f_2(t)$

Figure 49: The instantaneous frequency for $f_1(t)$ and $f_2(t)$ and the comparison with the instantaneous frequency of $x_0(t)$.

The intersection positions of the instantaneous frequencies indicate where the resonances occur. Based on this knowledge, the forcing function can be replaced by a combination of some of its IMFs and the solution to the linear system does not get hurt. The substitute of the forcing function is derived by summing up only the components whose instantaneous frequency has intersections with the instantaneous frequency of the intrinsic mode of this system.

CHAPTER VI

CONCLUSIONS

In this thesis, we develop Adaptive Local Iterative Filtering (ALIF) algorithms. We show the convergence of iterative filtering algorithm with both uniform filters and adaptive filters for general signals and the adaptivity of ALIF algorithm is guided by these convergence theorems. We also design a local filter based on the solution of Fokker-Planck equation and it can be used in ALIF algorithm to ensure the locality. The ALIF algorithm we propose gives neat and clean decompositions for different signals. Moreover, numerical examples show that ALIF algorithm yields stable decompositions under noise or perturbation.

With the aim of computing the instantaneous frequency locally, we also present a new definition of the instantaneous frequency based on the speed of the rotation in a perturbation of the unit circle. The examples show that this new definition has similar results with the instantaneous frequency defined based on Hilbert transform for relatively smooth signals and better performance for signals with sudden changes in the magnitude.

We apply the proposed ALIF algorithm to the gas detection application. The results show that ALIF algorithm does help on increasing the accuracy of prediction. We also analyze the solutions of some second order ODEs with the decomposition by ALIF algorithm. With these analysis, we attempt to explain the good performance of ALIF algorithm, IF algorithm or EMD. We believe that those algorithms reveal the intrinsic oscillatory patterns of some second order ODEs. Future work need to be done to certify our belief.

REFERENCES

[1] "http:oldwcatwc.arh.noaa.gov/previous.events/03-11-11_honshu/."

[2] Andrade, A., Nasuto, S., Kyberd, P., Sweeney-Reed, C., and Van Kanijn, F., "Emg signal filtering based on empirical mode decomposition," *Biomedical Signal Processing and Control*, vol. 1, no. 1, pp. 44–55, 2006.

[3] Balocchi, R., Menicucci, D., and Varanini, M., "Empirical mode decomposition to approach the problem of detecting sources from a reduced number of mixtures," in *Engineering in Medicine and Biology Society, 2003. Proceedings of the 25th Annual International Conference of the IEEE*, vol. 3, pp. 2443–2446, IEEE, 2003.

[4] Battista, B. M., Knapp, C., McGee, T., and Goebel, V., "Application of the empirical mode decomposition and hilbert-huang transform to seismic reflection data," *Geophysics*, vol. 72, no. 2, pp. H29–H37, 2007.

[5] Bekara, M. and van der Baan, M., "Random and coherent noise attenuation by empirical mode decomposition," *Geophysics*, vol. 74, no. 5, pp. V89–V98, 2009.

[6] Bhuiyan, S., Adhami, R., and Khan, J., "Fast and adaptive bidimensional empirical mode decomposition using order-statistics filter based envelope estimation," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, p. 164, 2008.

[7] Bhuiyan, S. M., Adhami, R. R., and Khan, J. F., "A novel approach of fast and adaptive bidimensional empirical mode decomposition," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pp. 1313–1316, IEEE, 2008.

[8] Bin Altaf, M., Gautama, T., Tanaka, T., and Mandic, D. P., "Rotation invariant complex empirical mode decomposition," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 3, pp. III–1009, IEEE, 2007.

[9] Blanco-Velasco, M., Weng, B., and Barner, K., "Ecg signal denoising and baseline wander correction based on the empirical mode decomposition," *Computers in Biology and Medicine*, vol. 38, no. 1, pp. 1–13, 2008.

[10] Boashash, B., "Estimating and interpreting the instantaneous frequency of a signal. i. fundamentals," *Proceedings of the IEEE*, vol. 80, no. 4, pp. 520–538, 1992.

[11] Boudraa, A.-O., Cexus, J.-C., and others, "Denoising via empirical mode decomposition," *Proc. IEEE ISCCSP*, vol. 4, 2006.

[12] Boudraa, A., Cexus, J., Salzenstein, F., and Guillon, L., "If estimation using empirical mode decomposition and nonlinear teager energy operator," in *Control, Communications and Signal Processing, 2004. First International Symposium on*, pp. 45–48, IEEE, 2004.

[13] Cai, C., Liu, W., Fu, J. S., and Lu, L., "Empirical mode decomposition of micro-doppler signature," in *Radar Conference, 2005 IEEE International*, pp. 895–899, IEEE, 2005.

[14] Chan, T. F. and Zhou, H.-M., "Eno-wavelet transforms for piecewise smooth functions," *SIAM Journal on Numerical Analysis*, vol. 40, no. 4, pp. 1369–1404, 2002.

[15] Chan, T. F. and Zhou, H.-M., "Eno-wavelet transforms and some applications," *Studies in Computational Mathematics*, vol. 10, pp. 107–133, 2003.

[16] Chan, T. F. and Zhou, H., "Optimal construction of wavelet coefficients using total variation regularization in image compression," *CAM Report, No. 00-27, Dept. of Math., UCLA*, 2000.

[17] Chen, Q., Huang, N., Riemenschneider, S., and Xu, Y., "A b-spline approach for empirical mode decompositions," *Advances in Computational Mathematics*, vol. 24, no. 1-4, pp. 171–195, 2006.

[18] Chen, Y. and Feng, M. Q., "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.

[19] Chu, P. C., Fan, C., and Huang, N., "Derivative-optimized empirical mode decomposition for the hilbert-huang transform," *Journal of Computational and Applied Mathematics*, 2013.

[20] Cohen, L., *Time-frequency analysis: theory and applications*. Prentice-Hall, Inc., 1995.

[21] Coughlin, K. and Tung, K., "11-year solar cycle in the stratosphere extracted by the empirical mode decomposition method," *Advances in space research*, vol. 34, no. 2, pp. 323–329, 2004.

[22] Daubechies, I., Lu, J., and Wu, H., "Synchrosqueezed wavelet transforms: An empirical mode decomposition-like tool," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 243–261, 2011.

[23] Deering, R. and Kaiser, J., "The use of a masking signal to improve empirical mode decomposition," in *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on*, vol. 4, pp. iv–485, Ieee, 2005.

[24] Deléchelle, E., Lemoine, J., and Niang, O., "Empirical mode decomposition: an analytical approach for sifting process," *Signal Processing Letters, IEEE*, vol. 12, no. 11, pp. 764–767, 2005.

[25] Demir, B. and Erturk, S., "Empirical mode decomposition pre-process for higher accuracy hyperspectral image classification," in *Geoscience and Remote Sensing Symposium, 2008. IGARSS 2008. IEEE International*, vol. 2, pp. II–939, IEEE, 2008.

[26] Demir, B. and Erturk, S., "Empirical mode decomposition of hyperspectral images for support vector machine classification," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 48, no. 11, pp. 4071–4084, 2010.

[27] DiPietro, R. S., Manolakis, D. G., Lockwood, R. B., Cooley, T., and Jacobson, J., "Hyperspectral matched filter with false-alarm mitigation," *Optical Engineering*, vol. 51, no. 1, pp. 016202–1, 2012.

[28] Du, R. D., Yuan, Y. B., and Chen, M., "Empirical mode decomposition application for structural seismic responses," *Applied Mechanics and Materials*, vol. 256, pp. 2096–2101, 2013.

[29] Dugatkin, D., Zhou, H., Chan, T., and Effros, M., "Lagrangian optimization of a group testing for eno wavelets algorithm," in *Proceedings to the 2002 Conference on Information Sciences and Systems, Princeton University, New Jersey*, pp. 20–22, 2002.

[30] Echeverria, J., Crowe, J., Woolfson, M., and Hayes-Gill, B., "Application of empirical mode decomposition to heart rate variability analysis," *Medical and Biological Engineering and Computing*, vol. 39, no. 4, pp. 471–479, 2001.

[31] Fan, G.-F., Qing, S., Wang, H., Hong, W.-C., and Li, H.-J., "Support vector regression model based on empirical mode decomposition and auto regression for electric load forecasting," *Energies*, vol. 6, no. 4, pp. 1887–1901, 2013.

[32] Fawcett, T., "An introduction to roc analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.

[33] Flandrin, P., Rilling, G., and Goncalves, P., "Empirical mode decomposition as a filter bank," *Signal Processing Letters, IEEE*, vol. 11, no. 2, pp. 112–114, 2004.

[34] Flandrin, P., Gonçalves, P., Rilling, G., and others, "Detrending and denoising with empirical mode decompositions," in *Proceedings of the 12th European Signal Processing Conference (EUSIPCO04)*, vol. 2, pp. 1581–1584, Citeseer, 2004.

[35] Gao, Q., Duan, C., Fan, H., and Meng, Q., "Rotating machine fault diagnosis using empirical mode decomposition," *Mechanical Systems and Signal Processing*, vol. 22, no. 5, pp. 1072–1081, 2008.

[36] Gao, Q., LI, L.-m., MENG, Q.-f., Fan, H., and LEI, Y.-g., "Trend analysis approach based on empirical mode decomposition," *Journal of Vibration and Shock*, vol. 8, p. 025, 2007.

[37] Gröchenig, K., *Foundations of time-frequency analysis*. Birkhäuser Boston, 2000.

[38] Guanlei, X., Xiaotong, W., and Xiaogang, X., "Neighborhood limited empirical mode decomposition and application in image processing," in *Image and Graphics, 2007. ICIG 2007. Fourth International Conference on*, pp. 149–154, IEEE, 2007.

[39] Guanlei, X., Xiaotong, W., and Xiaogang, X., "Time-varying frequency-shifting signal-assisted empirical mode decomposition method for am–fm signals," *Mechanical Systems and Signal Processing*, vol. 23, no. 8, pp. 2458–2469, 2009.

[40] Hariharan, H., Gribok, A., Abidi, M., and Koschan, A., "Image fusion and enhancement via empirical mode decomposition," *Journal of Pattern Recognition Research*, vol. 1, no. 1, pp. 16–31, 2008.

[41] Hariharan, H., Koschan, A., Abidi, B., Gribok, A., and Abidi, M., "Fusion of visible and infrared images using empirical mode decomposition to improve face recognition," in *Image Processing, 2006 IEEE International Conference on*, pp. 2049–2052, IEEE, 2006.

[42] He, L., Lech, M., Maddage, N. C., and Allen, N. B., "Study of empirical mode decomposition and spectral analysis for stress and emotion classification in natural speech," *Biomedical Signal Processing and Control*, vol. 6, no. 2, pp. 139–146, 2011.

[43] He, L. and Wang, H., "Spatial-variant image filtering based on bidimensional empirical mode decomposition," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 2, pp. 1196–1199, IEEE, 2006.

[44] Hong, H., Wang, X., and Tao, Z., "Local integral mean-based sifting for empirical mode decomposition," *Signal Processing Letters, IEEE*, vol. 16, no. 10, pp. 841–844, 2009.

[45] Hou, T. Y. and Shi, Z., "Data-driven time-frequency analysis," *Applied and Computational Harmonic Analysis*, 2012.

[46] Hou, T. and Shi, Z., "Adaptive data analysis via sparse time-frequency representation," *Adv. in Adap. Data Anal.*, vol. 3, no. 1, pp. 1–28, 2011.

[47] Hou, T., Yan, M., and Wu, Z., "A variant of the emd method for multi-scale data," *Adv. in Adap. Data Anal.*, vol. 1, no. 4, pp. 483–516, 2009.

[48] Hu, H. Y., Li, W. L., and Zhao, F. Q., "Fully nonparametric regression estimation based on empirical mode decomposition," *Applied Mechanics and Materials*, vol. 271, pp. 932–935, 2013.

[49] Huang, C., YANG, L., and WANG, Y., "Convergence of a convolution-filtering-based algorithm for empirical mode decomposition," *Advances in Adaptive Data Analysis*, vol. 1, no. 04, pp. 561–571, 2009.

[50] Huang, N., Shen, Z., and Long, S., "A new view of nonlinear water waves: The hilbert spectrum 1," *Annual Review of Fluid Mechanics*, vol. 31, no. 1, pp. 417–457, 1999.

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

[52] Huang, N., Wu, Z., Long, S., Arnold, K., Chen, X., and Blank, K., "On instantaneous frequency," *Adv. Adapt. Data Anal*, vol. 1, no. 2, pp. 177–229, 2009.

[53] Huang, N. E., "Computer implemented empirical mode decomposition method, apparatus and article of manufacture," Nov. 9 1999. US Patent 5,983,162.

[54] Huang, N. E., "New method for nonlinear and nonstationary time series analysis: empirical mode decomposition and hilbert spectral analysis," in *AeroSense 2000*, pp. 197–209, International Society for Optics and Photonics, 2000.

[55] Huang, N. E., "Review of empirical mode decomposition," in *Aerospace/Defense Sensing, Simulation, and Controls*, pp. 71–80, International Society for Optics and Photonics, 2001.

[56] Huang, N. E., "An adaptive data analysis method for nonlinear and nonstationary time series: the empirical mode decomposition and hilbert spectral analysis," in *Wavelet Analysis and Applications*, pp. 363–376, Springer, 2007.

[57] Huang, N. E., Wu, M.-L. C., Long, S. R., Shen, S. S., Qu, W., Gloersen, P., and Fan, K. L., "A confidence limit for the empirical mode decomposition and hilbert spectral analysis," *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 459, no. 2037, pp. 2317–2345, 2003.

[58] Huang, N. E. and Shen, S. S., *Hilbert-Huang transform and its applications*, vol. 5. World Scientific, 2005.

[59] Huang, Y.-P., Li, X.-Y., and Zhang, R.-B., "A research on local mean in empirical mode decomposition," in *Computational Science–ICCS 2007*, pp. 125–128, Springer, 2007.

[60] Huanyin, Y., Huadong, G., Chunming, H., Xinwu, L., and Changlin, W., "A sar interferogram filter based on the empirical mode decomposition method," in *Geoscience and Remote Sensing Symposium, 2001. IGARSS'01. IEEE 2001 International*, vol. 5, pp. 2061–2063, IEEE, 2001.

[61] Jánosi, I. and Müller, R., "Empirical mode decomposition and correlation properties of long daily ozone records," *Physical Review E*, vol. 71, no. 5, p. 056126, 2005.

[62] Jánosi, I. M. and Müller, R., "Empirical mode decomposition and correlation properties of long daily ozone records," *Physical Review E*, vol. 71, no. 5, p. 056126, 2005.

[63] Jin-Ping, Z. and Da-ji, H., "Mirror extending and circular spline function for empirical mode decomposition method," *Journal of Zhejiang University Science*, vol. 2, no. 3, pp. 247–252, 2001.

[64] Kijewski-Correa, T. and Kareem, A., "Performance of wavelet transform and empirical mode decomposition in extracting signals embedded in noise," *Journal of engineering mechanics*, vol. 133, no. 7, pp. 849–852, 2007.

[65] Kim, D. and Oh, H.-S., "Emd: A package for empirical mode decomposition and hilbert spectrum," *The R Journal*, vol. 1, no. 1, pp. 40–46, 2009.

[66] Kopsinis, Y. and McLaughlin, S., "Investigation and performance enhancement of the empirical mode decomposition method based on a heuristic search optimization approach," *Signal Processing, IEEE Transactions on*, vol. 56, no. 1, pp. 1–13, 2008.

[67] Kurzyna, J., Mazouffre, S., Lazurenko, A., Albarède, L., Bonhomme, G., Makowski, K., Dudeck, M., and Peradzyński, Z., "Spectral analysis of hall-effect thruster plasma oscillations based on the empirical mode decomposition," *Physics of plasmas*, vol. 12, p. 123506, 2005.

[68] Lai, R. J. and Huang, N., "Investigation of vertical and horizontal momentum transfer in the gulf of mexico using empirical mode decomposition method," *Journal of physical oceanography*, vol. 35, no. 8, pp. 1383–1402, 2005.

[69] Li, C. and Liang, M., "Time–frequency signal analysis for gearbox fault diagnosis using a generalized synchrosqueezing transform," *Mechanical Systems and Signal Processing*, vol. 26, pp. 205–217, 2012.

[70] Li, H. and Meng, G., "Detection of harmonic signals from chaotic interference by empirical mode decomposition," *Chaos, Solitons & Fractals*, vol. 30, no. 4, pp. 930–935, 2006.

[71] Li, S., Su, X., Chen, W., and Xiang, L., "Eliminating the zero spectrum in fourier transform profilometry using empirical mode decomposition," *JOSA A*, vol. 26, no. 5, pp. 1195–1201, 2009.

[72] Li, T.-Y., Zhao, Y., Nan, L., Fen, G., and Gao, H.-H., "A new method for power quality detection based on hht," in *Zhongguo Dianji Gongcheng Xuebao(Proceedings of the Chinese Society of Electrical Engineering)*, vol. 25, pp. 52–56, 2005.

[73] Li, T., Zhao, Y., and Li, N., "Apply empirical mode decomposition based hilbert transform to power system transient signal analysis [j]," *Automation of Electric Power Systems*, vol. 4, 2005.

[74] Liang, H., Bressler, S., Buffalo, E., Desimone, R., and Fries, P., "Empirical mode decomposition of field potentials from macaque v4 in visual spatial attention," *Biological cybernetics*, vol. 92, no. 6, pp. 380–392, 2005.

[75] Liang, H., Bressler, S., Desimone, R., and Fries, P., "Empirical mode decomposition: a method for analyzing neural data," *Neurocomputing*, vol. 65, pp. 801–807, 2005.

[76] Liang, H., Lin, Q.-H., and Chen, J., "Application of the empirical mode decomposition to the analysis of esophageal manometric data in gastroesophageal reflux disease," *Biomedical Engineering, IEEE Transactions on*, vol. 52, no. 10, pp. 1692–1701, 2005.

[77] Liang, H., Lin, Z., and McCallum, R., "Artifact reduction in electrogastrogram based on empirical mode decomposition method," *Medical and Biological Engineering and Computing*, vol. 38, no. 1, pp. 35–41, 2000.

[78] Lin, L., Wang, Y., and Zhou, H., "Iterative filtering as an alternative algorithm for empirical mode decomposition," *Advances in Adaptive Data Analysis*, vol. 1, no. 4, pp. 543–560, 2009.

[79] Lin, S.-L., Tung, P.-C., and Huang, N. E., "Data analysis using a combination of independent component analysis and empirical mode decomposition," *Physical Review E*, vol. 79, no. 6, p. 066705, 2009.

[80] Linderhed, A., "2d empirical mode decompositions in the spirit of image compression," in *AeroSense 2002*, pp. 1–8, International Society for Optics and Photonics, 2002.

[81] Linderhed, A., *Adaptive image compression with wavelet packets and empirical mode decomposition*. Univ., 2004.

[82] Linderhed, A., "Image compression based on empirical mode decomposition," in *Proc. of SSAB Symposium Image Analysis*, pp. 110–113, 2004.

[83] Linderhed, A., "Image empirical mode decomposition: A new tool for image processing," *Advances in Adaptive Data Analysis*, vol. 1, no. 02, pp. 265–294, 2009.

[84] Liu, B., Riemenschneider, S., and Xu, Y., "Gearbox fault diagnosis using empirical mode decomposition and hilbert spectrum," *Mechanical Systems and Signal Processing*, vol. 20, no. 3, pp. 718–734, 2006.

[85] Liu, Z., Wang, H., and Peng, S., "Texture classification through directional empirical mode decomposition," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 4, pp. 803–806, IEEE, 2004.

[86] Loughlin, P. and Tacer, B., "Comments on the interpretation of instantaneous frequency," *Signal Processing Letters, IEEE*, vol. 4, no. 5, pp. 123–125, 1997.

[87] Loutridis, S., "Damage detection in gear systems using empirical mode decomposition," *Engineering Structures*, vol. 26, no. 12, pp. 1833–1841, 2004.

[88] Lu, Z., Smith, J., Wu, Q., and Fitch, J., "Empirical mode decomposition for power quality monitoring," in *Transmission and Distribution Conference and Exhibition: Asia and Pacific, 2005 IEEE/PES*, pp. 1–5, IEEE, 2005.

[89] Maes, S., "Synchrosqueezed representation yields a new reading of the wavelet transform," in *SPIE's 1995 Symposium on OE/Aerospace Sensing and Dual Use Photonics*, pp. 532–559, International Society for Optics and Photonics, 1995.

[90] Magrin-Chagnolleau, I. and Baraniuk, R., "Empirical mode decomposition based frequency attributes," in *Proceedings of the 69th SEG Meeting*, pp. 1949–1952, 1999.

[91] Mandic, D. P. and others, "Empirical mode decomposition for trivariate signals," *Signal Processing, IEEE Transactions on*, vol. 58, no. 3, pp. 1059–1068, 2010.

[92] Mandic, D. and others, "Filter bank property of multivariate empirical mode decomposition," *Signal Processing, IEEE Transactions on*, vol. 59, no. 5, pp. 2421–2426, 2011.

[93] Manjula, M., Mishra, S., and Sarma, A., "Empirical mode decomposition with hilbert transform for classification of voltage sag causes using probabilistic neural network," *International Journal of Electrical Power & Energy Systems*, vol. 44, no. 1, pp. 597–603, 2013.

[94] Manolakis, D., Marden, D., and Shaw, G. A., "Hyperspectral image processing for automatic target detection applications," *Lincoln Laboratory Journal*, vol. 14, no. 1, pp. 79–116, 2003.

[95] McDonald, A., Baumgaertner, A., Fraser, G., George, S., Marsh, S., and others, "Empirical mode decomposition of the atmospheric wave field," in *Annales Geophysicae*, vol. 25, pp. 375–384, 2007.

[96] Meignen, S. and Perrier, V., "A new formulation for empirical mode decomposition based on constrained optimization," *Signal Processing Letters, IEEE*, vol. 14, no. 12, pp. 932–935, 2007.

[97] Mijovic, B., De Vos, M., Gligorijevic, I., Taelman, J., and Van Huffel, S., "Source separation from single-channel recordings by combining empirical-mode decomposition and independent component analysis," *Biomedical Engineering, IEEE Transactions on*, vol. 57, no. 9, pp. 2188–2196, 2010.

[98] Ming, H. X. J., "Improved algorithm of image signal denoising based on emd [j]," *Electronic Measurement Technology*, vol. 11, p. 018, 2009.

[99] Neto, E., Custaud, M., Cejka, J., Abry, P., Frutoso, J., Gharib, D., and Flandrin, P., "Assessment of cardiovascular autonomic control by the empirical mode decomposition," *Methods of information in medicine*, vol. 43, no. 1, pp. 60–65, 2004.

[100] Nunes, J., Niang, O., Bouaoune, Y., Delechelle, E., and Bunel, P., "Bidimensional empirical mode decomposition modified for texture analysis," *Image Analysis*, pp. 295–296, 2003.

[101] Parey, A., El Badaoui, M., Guillet, F., and Tandon, N., "Dynamic modelling of spur gear pair and application of empirical mode decomposition-based statistical analysis for early detection of localized tooth defect," *Journal of sound and vibration*, vol. 294, no. 3, pp. 547–561, 2006.

[102] Pines, D. and Salvino, L., "Structural health monitoring using empirical mode decomposition and the hilbert phase," *Journal of sound and vibration*, vol. 294, no. 1, pp. 97–124, 2006.

[103] Pines, D. and Salvino, L., "Health monitoring of one-dimensional structures using empirical mode decomposition and the hilbert-huang transform," in *Proceedings of SPIE*, vol. 4701, p. 127, 2002.

[104] QU, C.-s., LU, T.-z., and Tan, Y., "A modified empirical mode decomposition method with applications to signal de-noising," *Acta Automatica Sinica*, vol. 36, pp. 67–73, 2010.

[105] Rehman, N. and Mandic, D. P., "Multivariate empirical mode decomposition," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science*, vol. 466, no. 2117, pp. 1291–1302, 2010.

[106] Rilling, G., Flandrin, P., and Gonçalves, P., "Empirical mode decomposition, fractional gaussian noise and hurst exponent estimation," in *Acoustics,*

Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on, vol. 4, pp. iv–489, IEEE, 2005.

[107] Rilling, G., Flandrin, P., Gonçalvés, P., and others, "On empirical mode decomposition and its algorithms," in *IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing NSIP*, vol. 3, pp. 8–11, 2003.

[108] Rilling, G. and Flandrin, P., "On the influence of sampling on the empirical mode decomposition," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 3, pp. III–III, IEEE, 2006.

[109] Ruzmaikin, A., Feynman, J., and Yung, Y., "Is solar variability reflected in the nile river?," *Journal of geophysical research*, vol. 111, p. D21114, 2006.

[110] Sethu, V., Ambikairajah, E., and Epps, J., "Empirical mode decomposition based weighted frequency feature for speech-based emotion classification," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pp. 5017–5020, IEEE, 2008.

[111] Shi, Z. and Law, S., "Identification of linear time-varying dynamical systems using hilbert transform and empirical mode decomposition method," *Journal of applied mechanics*, vol. 74, no. 2, pp. 223–230, 2007.

[112] Sinclair, S., Pegram, G., and others, "Empirical mode decomposition in 2-d space and time: a tool for space-time rainfall analysis and nowcasting," *Hydrology and Earth System Sciences*, vol. 9, no. 3, pp. 127–137, 2005.

[113] Spanos, P., Giaralis, A., and Politis, N., "Time–frequency representation of earthquake accelerograms and inelastic structural response records using the adaptive chirplet decomposition and empirical mode decomposition," *Soil Dynamics and Earthquake Engineering*, vol. 27, no. 7, pp. 675–689, 2007.

[114] Stevenson, N., Mesbah, M., and Boashash, B., "A sampling limit for the empirical mode decomposition," in *ISSPA 2005: The 8th International Symposium on Signal Processing and its Applications, Vols 1 and 2, Proceedings*, vol. 2, pp. 647–650, IEEE, 2005.

[115] Sweeney-Reed, C. and Nasuto, S., "A novel approach to the detection of synchronisation in eeg based on empirical mode decomposition," *Journal of Computational Neuroscience*, vol. 23, no. 1, pp. 79–111, 2007.

[116] Tanaka, T. and Mandic, D. P., "Complex empirical mode decomposition," *Signal Processing Letters, IEEE*, vol. 14, no. 2, pp. 101–104, 2007.

[117] ur Rehman, N. and Mandic, D., "Empirical mode decomposition for trivariate signals," *IEEE transactions on signal processing*, vol. 58, no. 3, pp. 1059–1068, 2010.

[118] Varadarajan, N. and others, "Wind response control of building with variable stiffness tuned mass damper using empirical mode decomposition/hilbert transform," *Journal of engineering mechanics*, vol. 130, p. 451, 2004.

[119] Wang, N., Ambikairajah, E., Celler, B. G., and Lovell, N. H., "Accelerometry based classification of gait patterns using empirical mode decomposition," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pp. 617–620, IEEE, 2008.

[120] Wang, Y., Wei, G.-W., and Yang, S., "Partial differential equation transformvariational formulation and fourier analysis," *International journal for numerical methods in biomedical engineering*, vol. 27, no. 12, pp. 1996–2020, 2011.

[121] Wang, Y., Wei, G.-W., and Yang, S., "Iterative filtering decomposition based on local spectral evolution kernel," *Journal of scientific computing*, vol. 50, no. 3, pp. 629–664, 2012.

[122] Wang, Y., Wei, G.-W., and Yang, S., "Mode decomposition evolution equations," *Journal of scientific computing*, vol. 50, no. 3, pp. 495–518, 2012.

[123] Wang, Y., Wei, G.-W., and Yang, S., "Selective extraction of entangled textures via adaptive pde transform," *Journal of Biomedical Imaging*, vol. 2012, p. 3, 2012.

[124] Wang, Y., He, Z., and Zi, Y., "A comparative study on the local mean decomposition and empirical mode decomposition and their applications to rotating machinery health diagnosis," *Journal of vibration and acoustics*, vol. 132, no. 2, 2010.

[125] Wang, Y., Wang, H., and Zhang, Q., "Multi-source separation using over iterative empirical mode decomposition," in *Intelligence Computation and Evolutionary Computation*, pp. 363–368, Springer, 2013.

[126] Wei, D. and Bovik, A., "On the instantaneous frequencies of multicomponent am-fm signals," *Signal Processing Letters, IEEE*, vol. 5, no. 4, pp. 84–86, 1998.

[127] Weng, B., Blanco-Velasco, M., and Earner, K., "Ecg denoising based on the empirical mode decomposition," in *Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE*, pp. 1–4, IEEE, 2006.

[128] Weng, B., Xuan, G., Kolodzey, J., and Barner, K. E., "Empirical mode decomposition as a tool for dna sequence analysis from terahertz spectroscopy measurements," in *Genomic Signal Processing and Statistics, 2006. GENSIPS'06. IEEE International Workshop on*, pp. 63–64, IEEE, 2006.

[129] Wu, F. and Qu, L., "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*, vol. 314, no. 3, pp. 586–602, 2008.

[130] Wu, M.-C. and Hu, C.-K., "Empirical mode decomposition and synchrogram approach to cardiorespiratory synchronization," *Physical Review E*, vol. 73, no. 5, p. 051917, 2006.

[131] Wu, Z. and Huang, N., "A study of the characteristics of white noise using the empirical mode decomposition method," *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 460, no. 2046, p. 1597, 2004.

[132] Wu, Z. and Huang, N., "Statistical significance test of intrinsic mode functions," *Hilbert–Huang Transform and Its Applications*, pp. 107–127, 2005.

[133] Wu, Z. and Huang, N., "Ensemble empirical mode decomposition: A noise-assisted data analysis method," *Adv. in Adap. Data Anal.*, vol. 1, no. 1, pp. 1–41, 2009.

[134] Wu, Z., Huang, N., Long, S., and Peng, C., "On the trend, detrending, and variability of nonlinear and nonstationary time series," *Proceedings of the National Academy of Sciences*, vol. 104, no. 38, p. 14889, 2007.

[135] Wu, Z., Huang, N. E., and CHEN, X., "The multi-dimensional ensemble empirical mode decomposition method," *Advances in Adaptive Data Analysis*, vol. 1, no. 03, pp. 339–372, 2009.

[136] Xu, Z., Huang, B., and Zhang, F., "Improvement of empirical mode decomposition under low sampling rate," *Signal processing*, vol. 89, no. 11, pp. 2296–2303, 2009.

[137] Yang, J. and Lei, Y., "System identification of linear structures using hilbert transform and empirical mode decomposition," in *SPIE proceedings series*, pp. 213–219, Society of Photo-Optical Instrumentation Engineers, 2000.

[138] Yang, P., Wang, G., Bian, J., and Zhou, X., "The prediction of non-stationary climate series based on empirical mode decomposition," *Advances in Atmospheric Sciences*, vol. 27, no. 4, p. 845, 2010.

[139] Yeh, J.-R., Fan, S.-Z., and Shieh, J.-S., "Human heart beat analysis using a modified algorithm of detrended fluctuation analysis based on empirical mode decomposition," *Medical Engineering & Physics*, vol. 31, no. 1, pp. 92–100, 2009.

[140] YEH, J.-R., SHIEH, J.-S., and HUANG, N. E., "Complementary ensemble empirical mode decomposition: A novel noise enhanced data analysis method," *Advances in Adaptive Data Analysis*, vol. 2, no. 02, pp. 135–156, 2010.

[141] Zeiler, A., Faltermeier, R., Tomé, A., Keck, I., Puntonet, C., Brawanski, A., and Lang, E., "Sliding empirical mode decomposition-brain status data analysis and modeling," in *Advances in Intelligent Signal Processing and Data Mining*, pp. 311–349, Springer, 2013.

[142] Zeng, K. and He, M.-X., "A simple boundary process technique for empirical mode decomposition," in *Geoscience and Remote Sensing Symposium, 2004. IGARSS'04. Proceedings. 2004 IEEE International*, vol. 6, pp. 4258–4261, IEEE, 2004.

[143] Zhang, C., Ren, W., Mu, T., Fu, L., and Jia, C., "Empirical mode decomposition based background removal and de-noising in polarization interference imaging spectrometer," *Optics Express*, vol. 21, no. 3, pp. 2592–2605, 2013.

[144] Zhang, H. and Gai, Q., "Research on properties of empirical mode decomposition method," in *Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on*, vol. 2, pp. 10001–10004, IEEE, 2006.

[145] Zhang, X., Lai, K., and Wang, S., "A new approach for crude oil price analysis based on empirical mode decomposition," *Energy Economics*, vol. 30, no. 3, pp. 905–918, 2008.

[146] Zhang, Y., Gao, Y., Wang, L., Chen, J., and Shi, X., "The removal of wall components in doppler ultrasound signals by using the empirical mode decomposition algorithm," *Biomedical Engineering, IEEE Transactions on*, vol. 54, no. 9, pp. 1631–1642, 2007.

[147] ZHENG, T.-x. and YANG, L.-h., "Discussion and improvement on empirical mode decomposition algorithm [j]," *Acta Scientiarum Naturalium Universitatis Sunyatseni*, vol. 1, p. 000, 2007.

[148] Zhou, H.-M., "Wavelet transforms and pde techniques in image compression," 2000.

[149] Zhou, T. C. and Zhou, H., "Adaptive eno-wavelet transforms for discontinuous functions'," *CAM Report*, no. 99-21, pp. 93–100, 1999.

[150] Zhu, X., "Gravity wave characteristics in the middle atmosphere derived from the empirical mode decomposition method," tech. rep., DTIC Document, 1997.

[151] Žvokelj, M., Zupan, S., and Prebil, I., "Multivariate and multiscale monitoring of large-size low-speed bearings using ensemble empirical mode decomposition method combined with principal component analysis," *Mechanical Systems and Signal Processing*, vol. 24, no. 4, pp. 1049–1067, 2010.

VITA

Jingfang Liu was born in Zhengding, China, on 1 May 1986. After completing her work at Hebei Zhengding High School, she went to the University of Science and Technology of China in Hefei, China, where she studied electrical engineering and information science and received her Bachelor of Science in July 2008. She moved to Atlanta, Georgia in August 2008, and in August 2009 entered Georgia Institute of Technology.