Appl. Comput. Harmon. Anal. 35 (2013) 284–308

Contents lists available at SciVerse ScienceDirect

Applied and Computational Harmonic Analysis

www.elsevier.com/locate/acha

Data-driven time–frequency analysis

Thomas Y. Hou [a], Zuoqiang Shi [b,*]

[a] Applied and Comput. Math., Caltech, Pasadena, CA 91125, United States
[b] Mathematical Sciences Center, Tsinghua University, Beijing 100084, China

ARTICLE INFO

Article history:
Received 25 February 2012
Revised 18 August 2012
Accepted 18 October 2012
Available online 24 October 2012
Communicated by Stephane G. Mallat

Keywords:
Time–frequency analysis
Instantaneous frequency
Sparse decomposition
Matching pursuit

ABSTRACT

In this paper, we introduce a new adaptive data analysis method to study trend and instantaneous frequency of nonlinear and nonstationary data. This method is inspired by the Empirical Mode Decomposition method (EMD) and the recently developed compressed (compressive) sensing theory. The main idea is to look for the sparsest representation of multiscale data within the largest possible dictionary consisting of intrinsic mode functions of the form $\{a(t)\cos(\theta(t))\}$, where $a \in V(\theta)$, $V(\theta)$ consists of the functions smoother than $\cos(\theta(t))$ and $\theta' \geqslant 0$. This problem can be formulated as a nonlinear l^0 optimization problem. In order to solve this optimization problem, we propose a nonlinear matching pursuit method by generalizing the classical matching pursuit for the l^0 optimization problem. One important advantage of this nonlinear matching pursuit method is it can be implemented very efficiently and is very stable to noise. Further, we provide an error analysis of our nonlinear matching pursuit method under certain scale separation assumptions. Extensive numerical examples will be given to demonstrate the robustness of our method and comparison will be made with the state-of-the-art methods. We also apply our method to study data without scale separation, and data with incomplete or under-sampled data.

1. Introduction

1.1. A brief review of time–frequency analysis and instantaneous frequency

Developing a truly adaptive data analysis method is important for our understanding of many natural phenomena. Traditional data analysis methods, such as the Fourier transform, use pre-determined basis. They provide an effective tool to process linear and stationary data. However, there are still some limitations in applying these methods to analyze nonlinear and nonstationary data. Time–frequency analysis has been developed to overcome the limitations of the traditional techniques by representing a signal with a joint function of both time and frequency. The recent advances of wavelet analysis have opened a new path for time–frequency analysis. A significant breakthrough of wavelet analysis is the use of multiscales to characterize signals. This technique has led to the development of several wavelet-based time–frequency analysis techniques [19,8,23].

Another important approach in the time–frequency analysis is to study instantaneous frequency of a signal. Some of the pioneering work in this area was due to Van der Pol [36] and Gabor [14], who introduced the so-called Analytic Signal (AS) method that uses the Hilbert transform to determine instantaneous frequency of a signal. This AS method is one of the most popular ways to define instantaneous frequency. Until very recently, this method works mostly for monocomponent signals

* Corresponding author.
E-mail addresses: hou@cms.caltech.edu (T.Y. Hou), zqshi@math.tsinghua.edu.cn (Z. Shi).

1063-5203/$ – see front matter
http://dx.doi.org/10.1016/j.acha.2012.10.001

in which the number of zero-crossings is equal to the number of local extrema [1]. There were other attempts to define instantaneous frequency such as the zero-crossing method [32,33,24] and the Wigner–Ville distribution method [1,21,30,11, 20,28]. However most of these methods are rather restrictive. The zero-crossing method cannot apply to the signal with multiple components and is sensitive to noise. The methods based on the Wigner–Ville distribution suffer from the basic limitation introduced by the Wigner–Ville distribution, i.e. the interference between different components. These methods extract the instantaneous frequency only but do not give the corresponding components directly. Some post processing is required if one would like to extract these components.

More substantial progress has been made only recently with the introduction of the EMD method [18]. The EMD method provides an effective tool to decompose a signal into a collection of Intrinsic Mode Functions (IMFs) that allow well-behaved Hilbert transforms for computation of physically meaningful time–frequency representation. We remark that the Hilbert spectral representation based on the wavelet projection has been carried in [27].

There has been some recent progress in developing a mathematical framework for an EMD like method using synchrosqueezed wavelet transforms by Daubechies, Lu and Wu [9]. This seems to be a promising approach. We have performed some preliminary numerical experiments to compare the synchrosqueezed wavelet approach with our method. We find that the two methods give complementary results. The synchrosqueezed wavelet method does not extract IMFs directly. Some post processing is required to obtain IMFs.

1.2. Adaptive time–frequency analysis via a nonlinear optimization

Inspired by the EMD method and the recently developed compressed (compressive) sensing theory, we propose a data-driven time–frequency analysis method. There are two important ingredients of this method. The first one is that the basis that is used to decompose the data is derived from the data rather than determined *a priori*. This explains the name "data-driven" in our method. The second ingredient is to look for the sparsest decomposition of the signal among the largest possible dictionary consisting of intrinsic mode functions. The adoption of this data-driven basis and the search for the sparsest decomposition over this highly redundant basis make our time–frequency analysis method fully adaptive to the signal. As we are going to demonstrate later, our method can reveal some hidden physical information of the signal, such as trend and instantaneous frequency.

Our data-driven time–frequency analysis method is motivated by the observation that many multiscale data have a sparse representation over certain multiscale basis. This basis is unknown *a priori* and is adapted to the data. Finding such nonlinear multiscale basis is an essential ingredient of our method. In some sense, our problem is more difficult than the compressed (compressive) sensing problem in which the basis is assumed to be known *a priori*. In our method, we reformulate the problem as a nonlinear optimization and find the basis and the decomposition simultaneously by looking for the sparsest decomposition among all the possible decompositions.

Our nonlinear optimization problem is formulated as follows:

$$\begin{aligned} &\underset{(a_k)_{1\leqslant k\leqslant M},\,(\theta_k)_{1\leqslant k\leqslant M}}{\text{Minimize}} \quad M,\\ &\text{subject to:} \quad \begin{cases} f=\sum_{k=1}^{M} a_k\cos\theta_k,\\ a_k\cos\theta_k\in\mathcal{D}, \end{cases} \end{aligned} \tag{1}$$

where $\mathcal{D}$ is the dictionary we use to decompose the signal which will be defined later in the paper. When the signal is polluted by noise, the equality in the above constraint is relaxed to be an inequality depending on the noise level.

This optimization problem can be viewed as a nonlinear version of the l^0 minimization problem and is known to be very challenging to solve. Inspired by the compressed (compressive) sensing theory [13,2,5,10], we propose an l^1-regularized nonlinear matching pursuit method to solve this nonlinear optimization problem.

1.3. Nonlinear matching pursuit

Our nonlinear matching pursuit follows an idea similar to that of matching pursuit [22]. In each step, we solve an l^1 regularized nonlinear least square problem to get the decomposition:

$$\begin{aligned} &\min_{a,\theta} \quad \gamma\|\hat{a}\|_{l^1}+\|f-a\cos\theta\|_{l^2}^2,\\ &\text{subject to:} \quad a\cos\theta\in\mathcal{D}, \end{aligned}$$

where $\gamma>0$ is a regularization parameter and $\hat{a}$ is the representation of a in an overcomplete Fourier basis which will be detailed later. Denote by r the residual after subtracting $a\cos\theta$ from f, i.e. $r=f-a\cos\theta$. We can then treat r as a new signal to extract the remaining components.

There are two important advantages of this nonlinear matching pursuit approach. The first one is that this method is very stable to noise perturbation. The second one is that it can be implemented very efficiently. For periodic data, the resulting method can be solved approximately by the Fast Fourier Transform, and the complexity of our algorithm is of order $O(N\log N)$ where N is the number of data sample points that we use to represent the signal. The low computational

cost and the robustness to noise perturbation make this method very effective in many applications. Moreover, for data that satisfy certain scale separation conditions, we prove that our method recovers the IMFs and their instantaneous frequencies accurately.

We perform extensive numerical experiments to test the robustness and the accuracy of our data-driven time–frequency analysis method for both synthetic data and some real data. Our results show that the nonlinear matching pursuit can indeed decompose a multiscale signal into a sparse collection of intrinsic mode functions. We also compare our method with the original EMD method. For data without noise, we find that our method gives results comparable to those obtained by the EMD method. Moreover, for noisy data, our method seems to provide better estimation of the instantaneous frequency and IMFs than EMD and recently developed EEMD method [37].

1.4. End effect and under-sampled data

A common difficulty in many data analysis methods is the relatively large error produced near the boundary of the data set. For the EMD method, this source of error is referred to as the "end effect", which is primarily caused by the use of cubic spline interpolation in constructing the envelopes and the median of the signal [37]. Our data-driven time–frequency analysis method seems to be less sensitive to this end effect, especially when the data satisfy certain scale separation property.

We also demonstrate that our data-driven time–frequency analysis method can be applied to recover the original signal with missing data in certain interval. The recovered signal as well as their instantaneous frequency seems to have reasonably good accuracy. We also apply our method to decompose under-sampled data. The result is quite encouraging even if the under-sampled data are polluted by noise.

1.5. Organization of this paper

The remaining of the paper is organized as follows. In Section 2, we give a brief review of some existing data analysis methods such as matching pursuit, basis pursuit and the EMD method. We also review an earlier version of our adaptive data analysis method based on the TV^3 norm. We introduce our data-driven time–frequency analysis method in Section 3. A simplified version of our method is introduced for periodic data. In Section 4, we present some numerical experiments to demonstrate the performance of our method. We also study incomplete or under-sampled data. In Section 5, we generalize our method to analyze data with poor scale separation property. We present some error analysis of our method in Section 6. Some conclusions are made in Section 7.

2. Brief review of the existing sparse decomposition methods

A considerable focus of activities in recent signal processing literature has been the development of the sparse signal representations over a redundant dictionary. Among these methods, matching pursuit [22] and basis pursuit [6] have attracted a lot of attention in recent years due to the development of compressed (compressive) sensing. All these methods consist of two parts: a dictionary to decompose the signal and a decomposition method to select the sparsest decomposition.

2.1. Dictionaries

A dictionary is a collection of parameterized waveforms $\mathcal{D} = \{\phi_\gamma\}_{\gamma\in\Gamma}$. Many dictionaries have been proposed in the literature. Here we review a few of them that have been used widely.

2.1.1. A Fourier dictionary

A Fourier dictionary is a collection of sinusoidal waveforms with $\gamma = (\omega, \nu)$, where $\omega \in [0, 2\pi]$ is an angular frequency variable and $\nu \in \{0, 1\}$ indicates the phase type: sine or cosine. More specifically, the waveforms consist of the following two families,

$$\phi_{\omega,0}(t) = \cos(\omega t), \qquad \phi_{\omega,1}(t) = \sin(\omega t), \quad \forall t \in \mathbb{R}. \tag{2}$$

For a standard Fourier dictionary, ω runs through the set of all cosines with Fourier frequencies $\omega_k = 2k\pi/n$, $k = 0, 1, \ldots, n/2$, and all sines with Fourier frequencies $\omega_k = 2k\pi/n$, $k = 1, \ldots, n/2 - 1$, where n is the number of sample points. We can also obtain an overcomplete Fourier dictionary by sampling the frequencies more finely. Let $l > 1$. We may choose $\omega_k = 2k\pi/(ln)$, $k = 0, 1, \ldots, ln/2$ for cosines and $\omega_k = 2k\pi/(ln)$, $k = 1, \ldots, ln/2 - 1$ for sines. This is an l-fold overcomplete system. In our algorithm to decompose non-periodic data, we will use this kind of overcomplete Fourier dictionary.

2.1.2. A wavelet dictionary

A wavelet dictionary is a collection of translations and dilations of the basic mother wavelet ψ, together with translations of the scaling function φ defined below:

$$\phi_{a,b,0}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right), \qquad \phi_{a,b,1}(t) = \frac{1}{\sqrt{a}}\varphi\left(\frac{t-b}{a}\right), \quad \forall t \in \mathbb{R}. \tag{3}$$

In this dictionary, the index $\gamma = (a, b, \nu)$, where $a \in (0, \infty)$ is a scale variable, $b \in \mathbb{Z}$ indicates location and $\nu \in \{0, 1\}$ indicates gender. For a standard wavelet dictionary, we let a, b run through the discrete collection of mother wavelets with dyadic scales $a_j = 2^j/n$, $j = j_0, \ldots, \log_2(n) - 1$, and locations that are integer multiples of the scale $b_{j,k} = ka_j$, $k = 0, \ldots, 2^j - 1$, and the collection of scaling functions at the coarse scale j_0. This dictionary consists of n waveforms, which form an orthonormal basis. As in the Fourier dictionary, an overcomplete wavelet dictionary can be obtained by sampling the locations more finely.

2.1.3. A time–frequency dictionary

A typical time–frequency dictionary is the Gabor dictionary due to Gabor [14]. In this dictionary, we take $\gamma = (\omega, \tau, \theta, \delta)$, where $\omega \in [0, \pi)$ is frequency, τ is a location, θ is a phase, and δ is the duration. We define the waveform as follows:

$$\phi_\gamma(t) = \exp\left(-\frac{(t-\tau)^2}{\delta^2}\right)\cos\big(\omega(t-\tau) + \theta\big), \quad \forall t \in \mathbb{R}. \tag{4}$$

Such waveforms consist of frequencies near ω and essentially vanish far away from τ.

2.1.4. An EMD dictionary

We can also define a dictionary via the EMD method. In the EMD method [18], the dictionary is the collection of all Intrinsic Mode Functions (IMFs), which are functions defined descriptively by enforcing the following two conditions:

1. The number of the extrema and the number of the zero crossings of the function must be equal or differ at most by one;
2. At any point of the function, the average of the upper envelope and the lower envelope defined by the local extrema should be zero (symmetric with respect to zero).

In some sense, this dictionary is the largest one among the dictionaries listed here. Roughly speaking, any oscillatory sinusoidal wave multiplied by a smooth envelope function satisfies the definition of IMFs. In fact, many commonly used dictionaries are included in this dictionary. For example, all the elements of a Fourier dictionary defined in (2) (standard or overcomplete) are IMFs. The elements in the Gabor dictionary given above, which is generated by applying a Gaussian envelope on a sinusoidal wave, are IMFs. Some wavelets, such as the Morlet wavelet, also satisfy the conditions of IMFs.

Inspired by the EMD method, we will use a variant of the EMD dictionary to construct a sparse decomposition of a signal via nonlinear optimization.

2.2. Decomposition methods

In this subsection, we review a few decomposition methods that can be used to give a sparse decomposition of a signal by exploiting the intrinsic sparsity structure of the signal. In recent years, there have been a lot of research activities in looking for the sparest representation of a signal over a redundant dictionary [22,6,10,4,5], i.e. looking for a decomposition of a signal f over a given dictionary $\mathcal{D} = \{\phi_\gamma\}_{\gamma \in \Gamma}$ as

$$f = \sum_{k=1}^{M} \alpha_{\gamma_k}\phi_{\gamma_k} + R^M, \tag{5}$$

with the smallest M, where R^M is the residual. Whether or not a signal can be decomposed into a sparse decomposition depends on the choice of the dictionary that we use to decompose the signal. In general, a more redundant dictionary tends to give better adaptivity, which implies better sparsity of the decomposition. However, if the dictionary is not a basis, the decomposition is not unique. We need to give a criterion to select the "best" decomposition among all the possible choices.

2.2.1. Matching pursuit

In [22], Mallat and Zhang introduced a general decomposition method called matching pursuit that exploits the sparsity of a signal. Starting from an initial approximation $\mathbf{s}^0 = 0$ and a residual $\mathbf{r}^0 = s$, matching pursuit builds up a sequence of sparse approximations step by step. At stage k, the method identifies an atom that best matches the residual and then adds it to the current approximation, so that $s^k = s^{k-1} + \alpha_k\phi_{\gamma k}$, where $\alpha_k = \langle r^{k-1}, \phi_{\gamma k}\rangle$ and $r_k = \mathbf{s} - \mathbf{s}^k$. After m steps, one has a representation of the form (5), with residual $R^m = r^m$. A similar algorithm was proposed for Gabor dictionaries by S. Qian and D. Chen [31].

An intrinsic feature of this algorithm is that when stopped after a few steps, it yields an approximate sparse representation using only a few atoms. When the dictionary is orthogonal, the method works perfectly. If the dictionary is not orthogonal, the situation is less clear. To improve the performance of matching pursuit, many other algorithms are proposed, such as the Orthogonal Matching Pursuit (OMP) which is analyzed by J. Tropp and A. Gilbert [35], compressive

sampling matching pursuit by D. Needell and J. Tropp [25], Regularized Orthogonal Matching Pursuit (ROMP) by D. Needell and R. Vershynin [26] etc.

2.2.2. Basis pursuit

Another important class of decomposition methods is basis pursuit, which was introduced by S. Chen, D. Donoho and M. Saunders [6]. First, we reformulate the decomposition problem in the following way. Suppose we have a discrete dictionary of p waveforms and we collect all these waveforms as columns of an n by p matrix $\mathbf{\Phi}$. The decomposition problem (5) can be reformulated as:

$$\mathbf{s} = \mathbf{\Phi}\alpha, \tag{6}$$

where $\alpha = (\alpha_\gamma)$ is the vector of coefficients in (5).

The basic idea of basis pursuit is to find a sparse representation of the signal whose coefficients have a minimal l^1 norm, i.e. the decomposition is obtained by solving the problem

$$\min \|\alpha\|_{l^1}, \quad \text{subject to } \mathbf{\Phi}\alpha = \mathbf{s}. \tag{7}$$

Recently, basis pursuit has received a lot of attention. An important property of basis pursuit is that it can recover the exact solution of the original l^0 minimization problem under some sparsity condition on the data [4,10]. There has been extensive research to obtain a sparse representation by basis pursuit in a variety of applications. An essential component of basis pursuit is to solve the l^1 minimization problem. The computational cost of solving this l^1 minimization is more expensive than that of the least-square problem in matching pursuit, although several powerful algorithms have been introduced to speed up the l^1 minimization problem, such as the split Bregman method [15], proximal algorithms [7,3] and so on. We remark that the split Bregman method is also known as the augmented Lagrangian technique and can be derived from Douglas–Rachford algorithm [34] or PPXA+ [29].

2.2.3. The EMD decomposition via a sifting process

The EMD method decomposes a signal to its IMFs sequentially. The basic idea behind this approach is the removal of the local median from a signal by using a sifting process. Specifically, for a given signal, $f(t)$, one tries to decompose it as a sum of the local median $m(t)$ and an IMF. A cubic spline polynomial is used to interpolate all the local maxima (minima) to obtain an upper (lower) envelope. By averaging the upper and lower envelopes, one obtains an approximate median for $m(t)$. One then decides whether or not to accept the obtained $m(t)$ as our local median depending on whether $f(t) - m(t)$ gives an acceptable IMF that satisfies the two conditions that are specified in the definition of an EMD dictionary. If $f(t) - m(t)$ does not satisfy these conditions, one can treat $f(t) - m(t)$ as a new signal and construct a new candidate for the IMF by using the same procedure described above. This sifting process continues until we obtain a satisfactory IMF, which we denote as $f_n(t)$. Now we can treat $f(t) - f_n(t)$ as a new signal, and apply the same procedure to generate the second IMF, $f_{n-1}(t)$. This procedure continues until $f_0(t)$ is either monotone or contains at most one extremum. For more details of the sifting process, we refer to [18].

2.2.4. Decomposition based on a nonlinear TV^3 minimization

Inspired by the EMD method, we proposed a decomposition method based on a nonlinear TV^3 minimization in our previous paper [16]. Here TV^3 is the total variation of the third order derivative of a function, defined as $TV^3(f) = \int_a^b |f^{(4)}(t)|\,dt$. We use a TV^3 norm because the l^1 norm or the total variation norm is not strong enough to enforce the regularity of the median or the envelope.

In our approach, we decompose a signal $f(t)$ into its local median a_0 and an IMF $a_1\cos\theta(t)$ by solving the following nonlinear optimization problem:

$$\begin{aligned} (P) \quad & \min_{a_0,a_1,\theta} \quad TV^3(a_0) + TV^3(a_1), \\ & \text{subject to:} \quad \begin{cases} a_0 + a_1\cos\theta = f, \\ \theta'(t) \geqslant 0, \quad \forall t \in \mathbb{R}. \end{cases} \end{aligned} \tag{8}$$

To solve this nonlinear optimization problem, we proposed the following Newton type of iterative method:

Initialization: $\theta^0 = \theta_0$.
Main iteration:
Step 1: Update a_0^n, a_1^n, b_1^n by solving the following linear optimization problem:

$$\begin{aligned} & \left(a_0^n, a_1^n, b_1^n\right) \in \operatorname*{Argmin}_{a_0,a_1,b_1} \quad TV^3\left(a_0^n\right) + TV^3\left(a_1^n\right) + TV^3\left(b_1^n\right), \\ & \text{subject to:} \quad a_0^n + a_1^n\cos\theta^{n-1} + b_1^n\sin\theta^{n-1} = f. \end{aligned} \tag{9}$$

Step 2: Update the phase function θ:

$$\theta^n = \theta^{n-1} - \mu \arctan\left(\frac{b_1^n}{a_1^n}\right), \tag{10}$$

where $\mu \in [0,1]$ is chosen to enforce that θ^n is an increasing function:

$$\mu = \max\left\{\alpha \in [0,1]:\ \frac{d}{dt}\left(\theta_k^{n-1} - \alpha \arctan\left(\frac{b_1^n}{a_1^n}\right)\right) \geqslant 0\right\}. \tag{11}$$

Step 3: If $\|\theta^n - \theta^{n-1}\|_2 \leqslant \epsilon_0$, stop. Otherwise, go to Step 1.

In [16], we performed a number of numerical experiments and compared the results with those obtained by the EMD (or EEMD) method. Our results show that this method shares many important properties with the original EMD method. Moreover, its performance does not depend on numerical parameters such as the number of sifting or the stop criterion, which seem to have a major effect on the original EMD method. In many cases, this algorithm converges to the right decomposition in a few steps. However, we have not been able to prove that its convergence is guaranteed.

There are two limitations of this approach. The first one is that the computational cost to solve the TV^3 minimization problem is relatively high, even if we use the split Bregman method of Goldstein and Osher [15]. The second one is that this method is more sensitive to noise perturbation, although a nonlinear filter was introduced to alleviate this difficulty. In comparison, the nonlinear matching pursuit method we introduce in this paper is very stable to noise perturbation and has a relatively low computational cost.

3. A sparse time–frequency decomposition method based on nonlinear matching pursuit

Our data-driven time–frequency analysis method is based on finding the sparsest decomposition of a signal by solving a nonlinear optimization problem. First, we need to construct a large dictionary that can be used to obtain a sparse decomposition of the signal. In principle, the larger the dictionary is, the more adaptive (or sparser) the decomposition is.

3.1. Dictionary

In our method, the dictionary is chosen to be:

$$\mathcal{D} = \left\{a\cos\theta:\ a, \theta' \text{ is smoother than } \cos\theta,\ \forall t \in \mathbb{R},\ \theta'(t) \geqslant 0\right\}. \tag{12}$$

Let $V(\theta,\lambda)$ be the collection of all the functions that are smoother than $\cos\theta(t)$. In general, it is most effective to construct $V(\theta,\lambda)$ as an overcomplete Fourier basis given below:

$$V(\theta,\lambda) = \operatorname{span}\left\{1, \left(\cos\left(\frac{k\theta}{2L_\theta}\right)\right)_{1\leqslant k\leqslant 2\lambda L_\theta}, \left(\sin\left(\frac{k\theta}{2L_\theta}\right)\right)_{1\leqslant k\leqslant 2\lambda L_\theta}\right\}, \tag{13}$$

where $L_\theta = \lfloor \frac{\theta(1)-\theta(0)}{2\pi} \rfloor$, $\lfloor \mu \rfloor$ is the largest integer less than μ, and $\lambda \leqslant 1/2$ is a parameter to control the smoothness of $V(\theta,\lambda)$. In our computations, we typically choose $\lambda = 1/2$. The dictionary $\mathcal{D}$ can be written as

$$\mathcal{D} = \left\{a\cos\theta:\ a \in V(\theta,\lambda),\ \theta' \in V(\theta,\lambda),\ \text{and } \theta'(t) \geqslant 0,\ \forall t \in \mathbb{R}\right\}. \tag{14}$$

In some sense, the dictionary $\mathcal{D}$ defined above can be considered as a collection of IMFs. This property makes our method as adaptive as EMD. We also call an element of $\mathcal{D}$ as an IMF. Since the dictionary $\mathcal{D}$ is highly redundant, the decomposition over this dictionary is not unique. We need a criterion to select the "best" one among all possible decompositions. We assume that the data we consider have an intrinsic sparse structure in the time–frequency plane in some nonlinear and nonstationary basis. However, we do not know this basis *a priori* and we need to derive (or learn) this basis from the data. Based on this consideration, we adopt sparsity as our criterion to choose the best decomposition. This criterion yields the following nonlinear optimization problem:

$$P: \quad \underset{(a_k)_{1\leqslant k\leqslant M},(\theta_k)_{1\leqslant k\leqslant M}}{\text{Minimize}} \quad M,$$
$$\text{subject to:} \quad \begin{cases} f = \sum_{k=1}^M a_k\cos\theta_k, \\ a_k\cos\theta_k \in \mathcal{D}, \quad k = 1,\dots,M, \end{cases} \tag{15}$$

$$P_\delta: \quad \underset{(a_k)_{1\leqslant k\leqslant M},(\theta_k)_{1\leqslant k\leqslant M}}{\text{Minimize}} \quad M,$$
$$\text{subject to:} \quad \begin{cases} \|f - \sum_{k=1}^M a_k\cos\theta_k\|_{l^2} \leqslant \delta, \\ a_k\cos\theta_k \in \mathcal{D}, \quad k = 1,\dots,M, \end{cases} \tag{16}$$

if the signal has noise with noise level δ.

After this optimization problem is solved, we get a very clear time–frequency representation $\omega_k(t) = \theta_k'(t)$ with the amplitude $a_k(t)$.

3.2. Nonlinear matching pursuit

The above optimization problem can be seen as a nonlinear l^0 minimization problem. Thanks to the recent developments of compressed (compressive) sensing, two types of methods have been developed to study this problem. Specifically, matching pursuit has been shown to be a powerful method to solve the l^0 minimization problem while basis pursuit has provided an effective alternative by relaxing the original l^0 minimization problem to a convex l^1 minimization problem. Since the dictionary we adopt here has infinitely many elements, the l^1 norm of the coefficient vector consisting of infinitely many elements may not be well defined. Moreover, it is impossible to evaluate the l^1 norm of a vector with infinitely many elements in real computations. Therefore, it is not so straightforward to generalize basis pursuit directly. On the other hand, the idea of matching pursuit can be generalized.

Applying the idea of matching pursuit to our problem directly, we need to solve a nonlinear least square problem to get the decomposition:

$$\begin{aligned} &\min_{a,\theta} \quad \|f - a\cos\theta\|_{l^2}^2, \\ &\text{subject to:} \quad a\cos\theta \in \mathcal{D}. \end{aligned}$$

For non-periodic data, since we use the overcomplete Fourier basis to construct $V(\theta,\lambda)$, the above nonlinear least square problem may be ill-conditioned or even underdetermined (the number of sample points is less than the number of bases of $V(\theta,\lambda)$). Moreover, in this case, the simple least square method would introduce severe interference among different IMFs. In order to stabilize the above optimization problem and remove the interference, we add an l^1 term to regularize the nonlinear least square problem. This would give us the following algorithm based on the l^1 regularized nonlinear least square:

- $r_0 = f$, $k = 1$.

Step 1: Solve the following l^1-regularized nonlinear least-square problem (P_2):

$$\begin{aligned} P_2: \quad (a_k,\theta_k) \in \operatorname*{Argmin}_{a,\theta} \quad & \gamma\|\hat{a}\|_{l^1} + \|r_{k-1} - a\cos\theta\|_{l^2}^2, \\ \text{subject to:} \quad & a \in V(\theta,\lambda),\ \theta' \geqslant 0,\ \forall t \in \mathbb{R}, \end{aligned} \tag{17}$$

where $\gamma > 0$ is a regularization parameter and $\hat{a}$ is the representation of a in the overcomplete Fourier basis previously detailed in (13).

Step 2: Update the residual

$$r_k = f - \sum_{j=1}^{k} a_j \cos\theta_j. \tag{18}$$

Step 3: If $\|r_k\|_{l^2} < \epsilon_0$, stop. Otherwise, set $k = k+1$ and go to Step 1.

If signals are periodic, we can use the standard Fourier basis to construct $V(\theta,\lambda)$ instead of the overcomplete Fourier basis. The l^1 regularization term is not needed (i.e. we can set $\gamma = 0$) since the standard Fourier basis are orthogonal to each other. In the next section, we will use this property to further simplify above algorithm for periodic signals.

3.2.1. An l^1 regularized nonlinear least square solver

One of the main difficulties in solving our l^1 regularized nonlinear least square problem is that the objective functional is nonconvex since the basis is not known *a priori*. We need to find the basis and the decomposition simultaneously. In the following, we propose a Gauss–Newton type method to solve the l^1 regularized nonlinear least square problem. To alleviate the difficulty of generating a good initial guess for our iterative method, we gradually enlarge the space $V(\theta,\eta)$ to update θ' by increasing η during the iterations. Note that $V(\theta,\eta=0)$ for the first iteration consists only constants while $V(\theta,\eta=\lambda)$ during the final iteration is the space in which our nonlinear optimization is defined. Throughout our computations, we choose $\lambda = 1/2$ and the increment $\Delta\eta = \lambda/20$.

- $\theta_k^0 = \theta_0$, $\eta = 0$.

Step 1: Solve the following l^1 regularized least-square problem:

$$P_{2,l_2}: \quad \left(a_k^{n+1}, b_k^{n+1}\right) \in \operatorname*{Argmin}_{a,b} \quad \gamma\left(\|\hat{a}\|_{l^1} + \|\hat{b}\|_{l^1}\right) + \left\| r_{k-1} - a\cos\theta_k^n - b\sin\theta_k^n \right\|_{l^2}^2,$$
$$\text{subject to:} \quad a \in V\left(\theta_k^n, \lambda\right),\ b \in V\left(\theta_k^n, \lambda\right),$$

where $\hat{a}, \hat{b}$ are the representations of a, b in the $V(\theta_k^n, \lambda)$ space.

Step 2: Update θ_k^n:

$$\Delta\theta' = P_{V(\theta^n;\eta)}\left(\frac{d}{dt}\left(\arctan\left(\frac{b_k^{n+1}}{a_k^{n+1}}\right)\right)\right), \qquad \Delta\theta = \int_0^t \Delta\theta'(s)\,ds, \qquad \theta_k^{n+1} = \theta_k^n - \beta\Delta\theta, \tag{19}$$

where $\beta \in [0, 1]$ is chosen to make sure that θ_k^{n+1} is monotonically increasing:

$$\beta = \max\left\{\alpha \in [0,1]: \frac{d}{dt}\left(\theta_k^n - \alpha\Delta\theta\right) \geqslant 0\right\}, \tag{20}$$

where $P_{V(\theta_k^n;\eta)}$ is the projection operator to the space $V(\theta_k^n;\eta)$ and $V(\theta_k^n;\eta)$ is the space defined in (13).

Step 3: If $\|\theta_k^{n+1} - \theta_k^n\|_2 > \epsilon_0$, set $n = n+1$ and go to Step 1. Otherwise, go to Step 4.

Step 4: If $\eta \geqslant \lambda$, stop. Otherwise, set $\eta = \eta + \Delta\eta$ and go to Step 1. λ is the parameter we choose in (13).

3.2.2. Some implementation details

In the first step of the above algorithm, we solve an l^1 regularized least square problem. The l^1 regularization tends to stabilize the least square problem using an overcomplete Fourier basis. It also favors a sparse decomposition of the data.

In the second step to update the phase function, the time derivative is calculated in the following way:

$$\frac{d}{dt}\left(\arctan\left(\frac{b_k^{n+1}}{a_k^{n+1}}\right)\right) = \frac{a_k^{n+1}(b_k^{n+1})' - b_k^{n+1}(a_k^{n+1})'}{(a_k^{n+1})^2 + (b_k^{n+1})^2}, \tag{21}$$

where the derivatives are approximated by a central difference.

In the above formula (21), when we calculate the change of the instantaneous frequency at the points where the denominator $(a_k^{n+1})^2 + (b_k^{n+1})^2$ is small, the error may be amplified and our algorithm may become unstable. To overcome this difficulty, we modify our computation in the degenerate region where the denominator is smaller than certain predetermined threshold α. More precisely, we interpolate the values of $\Delta\theta$ from the non-degenerate region to obtain the value of $\Delta\theta$ in the degenerate region. In our computations, we set the threshold α to be 0.1.

It is known that a Gauss–Newton type iteration is sensitive to the initial guess. In general, it is very hard to find a good initial guess when the signal is polluted by noise. In order to abate the dependence on the initial guess in the above iteration, we gradually increase the value of η to improve the approximation to the phase function so that it converges to the correct value. When η is small, $\Delta\theta'$ is confined to a small space. In this small space, the objective functional has fewer extrema. The iteration may find an good approximation for $\Delta\theta'$. By gradually increasing η, we enlarge the space for $\Delta\theta'$ which allows for the small scale structures of $\Delta\theta'$ to develop gradually during the iterations. In our computations, this process converges even with a very rough initial guess.

The initial guess of θ can be also generated by other time–frequency analysis methods, such as the synchrosqueezed wavelet transforms [9]. In the following numerical examples, we obtain our initial guess using a simple approach based on the Fourier transform. More precisely, by estimating the wavenumber by which the high frequency components are centered around, we can obtain a reasonably good initial guess for θ. The initial guess for θ obtained in this way is a linear function. As we will see in the following numerical examples, even with these relatively rough initial guesses for θ, our algorithm still converges to the right answer with accuracy determined by the noise level.

3.3. A fast algorithm for periodic data

In the iterative algorithm given in previous section, we need to solve an l^1 regularized least square problem in each step. This is the most expensive part of the algorithm especially when the number of the data points is large. In this subsection, we introduce a method based on the Fast Fourier Transform (FFT) for periodic data.

One big advantage of our algorithm for periodic data is that we can use a standard Fourier basis to construct the $V(\theta, \lambda)$ space in the following way instead of the overcomplete Fourier basis given in (13).

$$V_p(\theta,\lambda) = \operatorname{span}\left\{1, \left(\cos\left(\frac{k\theta}{L_\theta}\right)\right)_{1\leqslant k\leqslant \lambda L_\theta}, \left(\sin\left(\frac{k\theta}{L_\theta}\right)\right)_{1\leqslant k\leqslant \lambda L_\theta}\right\}, \tag{22}$$

where $\lambda \leqslant 1/2$ is a parameter to control the smoothness of functions in $V_p(\theta,\lambda)$ and $L_\theta = (\theta(T) - \theta(0))/2\pi$ is a positive integer. Here we use a subscript p to denote this space is for the periodic signal.

Since the standard Fourier basis is an orthogonal basis, the l^1 regularized term is not necessary in our nonlinear optimization. As a result, the nonlinear least-square problem that we need to solve in the iterative algorithm is reduced to the following optimization problem (by setting $\gamma = 0$):

$$\begin{aligned} &\min_{a,b} \quad \left\| r_k - a\cos\theta_k^n - b\sin\theta_k^n \right\|_{l^2}^2, \\ &\text{subject to} \quad a, b \in V_p\left(\theta_k^n, \lambda\right). \end{aligned} \tag{23}$$

Notice that in the iterative process, the derivative of the phase function θ_k^n is always monotonically increasing. Thus, we can use θ_k^n as a new coordinate. In this new coordinate, $\cos\theta_k^n$, $\sin\theta_k^n$ and the bases of $V_p(\theta_k^n, \lambda)$ are simple Fourier modes, then the least-square problem can be solved by using the Fast Fourier Transform.

3.3.1. A FFT-based algorithm

Suppose that the signal r_{k-1} is measured over a uniform grid $t_j = j/N$, $j = 0, \ldots, N-1$ and it is well resolved which means that r can be interpolated to any grid with small error. Let $\bar{\theta}_k^n = \frac{\theta_k^n - \theta_k^n(0)}{\theta_k^n(T) - \theta_k^n(0)}$ be the normalized phase function and $L_{\theta_k^n} = \frac{\theta_k^n(T) - \theta_k^n(0)}{2\pi}$ which is an integer.

Our FFT-based algorithm to approximately solve (23) is given below:

Step 1: Interpolate r_{k-1} from $\{t_i\}_{i=1}^N$ in the physical space to a uniform mesh in the θ_k^n-coordinate to get $r_{\theta_k^n}$ and compute the Fourier transform $\hat{r}_{\theta_k^n}$:

$$r_{\theta_k^n, j} = \text{Interpolate}\left(r_{k-1}, \theta_{k,j}^n\right), \tag{24}$$

where $\theta_{k,j}^n$, $j = 0, \ldots, N-1$ are uniformly distributed in the θ_k^n-coordinate, i.e. $\theta_{k,j}^n = 2\pi L_{\theta_k^n} j/N$. And the Fourier transform of $r_{\theta_k^n}$ is given as follows

$$\hat{r}_{\theta_k^n}(\omega) = \frac{1}{N}\sum_{j=1}^{N} r_{\theta_k^n, j} e^{-i2\pi\omega\bar{\theta}_{k,j}^n}, \quad \omega = -N/2+1, \ldots, N/2, \tag{25}$$

where $\bar{\theta}_{k,j}^n = \frac{\theta_{k,j}^n - \theta_{k,0}^n}{2\pi L_{\theta_k^n}}$.

Step 2: Apply a cutoff function to the Fourier transform of $r_{\theta_k^n}$ to compute a and b on the mesh of the θ_k^n-coordinate, denoted by $a_{\theta_k^n}$ and $b_{\theta_k^n}$:

$$a_{\theta_k^n} = \mathcal{F}^{-1}\left[\left(\hat{r}_{\theta_k^n}(\omega + L_{\theta_k^n}) + \hat{r}_{\theta_k^n}(\omega - L_{\theta_k^n})\right) \cdot \chi_\lambda(\omega/L_{\theta_k^n})\right], \tag{26}$$

$$b_{\theta_k^n} = \mathcal{F}^{-1}\left[i \cdot \left(\hat{r}_{\theta_k^n}(\omega + L_{\theta_k^n}) - \hat{r}_{\theta_k^n}(\omega - L_{\theta_k^n})\right) \cdot \chi_\lambda(\omega/L_{\theta_k^n})\right]. \tag{27}$$

$\mathcal{F}^{-1}$ is the inverse Fourier transform defined in the θ_k^n coordinate:

$$\mathcal{F}^{-1}\left(\hat{r}_{\theta_k^n}\right) = \frac{1}{N}\sum_{\omega=-N/2+1}^{N/2} \hat{r}_{\theta_k^n} e^{i2\pi\omega\bar{\theta}_{k,j}^n}, \quad j = 0, \ldots, N-1. \tag{28}$$

Step 3: Interpolate $a_{\theta_k^n}$ and $b_{\theta_k^n}$ from the uniform mesh $\{\theta_{k,j}^n\}_{j=1}^N$ in the θ_k^n-coordinate back to the physical grid points $\{t_i\}_{i=1}^N$:

$$a(t_i) = \text{Interpolate}(a_{\theta_k^n}, t_i), \quad i = 0, \ldots, N-1, \tag{29}$$

$$b(t_i) = \text{Interpolate}(b_{\theta_k^n}, t_i), \quad i = 0, \ldots, N-1. \tag{30}$$

The low-pass filter $\chi_\lambda(\omega)$ in the second step is determined by the choice of $V_p(\theta, \lambda)$. The definition of $V_p(\theta, \lambda)$ in (22) implies that $\chi_\lambda(\omega)$ a stair function given as following:

$$\chi_\lambda(\omega) = \begin{cases} 1, & -\lambda < \omega < \lambda, \\ 0, & \text{otherwise.} \end{cases} \tag{31}$$

In the theoretical analysis in the subsequent section, we will see that the stair function is not a good choice as a filter. We can define a different space for $V_p(\theta, \lambda)$ by choosing an appropriate $\chi_\lambda(\omega)$. This opens up many choices for $V_p(\theta, \lambda)$. In this paper, we choose the following low-pass filter $\chi_\lambda(\omega)$ to define $V_p(\theta, \lambda)$:

$$\chi_\lambda(\omega) = \begin{cases} 1 + \cos(\pi\omega/\lambda), & -\lambda < \omega < \lambda, \\ 0, & \text{otherwise.} \end{cases} \tag{32}$$

3.3.2. A FFT-based nonlinear least square solver

By incorporating the FFT-based solver in our iterative algorithm, we get the following FFT-based iterative algorithm:

- $\theta_k^0 = \theta_0$, $\eta = 0$.

Step 1: Interpolate r_{k-1} to a uniform mesh in the θ_k^n-coordinate to get $r_{\theta_k^n}$ and compute the Fourier transform $\hat{r}_{\theta_k^n}$.
Step 2: Apply a cutoff function to the Fourier transform of $r_{\theta_k^n}$ to compute a and b on the mesh of the θ_k^n-coordinate, denoted by $a_{\theta_k^n}$ and $b_{\theta_k^n}$.
Step 3: Interpolate $a_{\theta_k^n}$ and $b_{\theta_k^n}$ back to the uniform mesh of t.
Step 4: Update θ^n in the t-coordinate:

$$\Delta\theta' = P_{V(\theta;\,\eta)}\left(\frac{d}{dt}\left(\arctan\left(\frac{b_k^{n+1}}{a_k^{n+1}}\right)\right)\right), \qquad \Delta\theta(t) = \int_0^t \Delta\theta'(s)\,ds, \qquad \theta_k^{n+1} = \theta_k^n - \beta\Delta\theta, \tag{33}$$

where $\beta \in [0, 1]$ is chosen to make sure that θ_k^{n+1} is monotonically increasing:

$$\beta = \max\left\{\alpha \in [0,1]:\ \frac{d}{dt}\left(\theta_k^n - \alpha\Delta\theta\right) \geqslant 0\right\}, \tag{34}$$

and $P_{V_p(\theta;\,\eta)}$ is the projection operator to the space $V_p(\theta;\,\eta)$.
Step 5: If $\|\theta_k^{n+1} - \theta_k^n\|_2 < \epsilon_0$, go to Step 6. Otherwise, set $n = n + 1$ and go to Step 1.
Step 6: If $\eta \geqslant \lambda$, stop. Otherwise, set $\eta = \eta + \Delta\eta$ and go to Step 1.

We remark that the projection operator $P_{V_p(\theta;\,\eta)}$ is in fact a low-pass filter in the θ-space. For non-periodic data, we apply a mirror extension to $\Delta\theta$ before we apply the low-pass filter.

In the second step of the above FFT-based iteration method, we can see that it has a close connection to the wavelet transform. In some sense, this step is equivalent to employing the continuous wavelet transform with a fixed scale in the θ-coordinate. The low-pass filter χ_λ plays a role similar to the scale function in the multiresolution analysis. Inspired by this observation, the space $V(\theta, \lambda)$ can be constructed based on multiresolution analysis. Let ψ be the wavelet and ϕ be the corresponding scaling function. An alternative construction of the space $V(\theta, \lambda)$ is given as follows:

$$V(\theta, \lambda) = \operatorname{span}\left\{\phi\left(\lambda\theta(t) - n\right),\ n \in \mathbb{Z}\right\}. \tag{35}$$

A typical choice of λ is to set $\lambda = 1/\omega$ where ω is the central frequency of the wavelet ψ.

Remark 3.1. We have recently proved that the algorithm proposed in this subsection will converge if the signal has certain scale separation property [17]. More specifically, let

$$f(t) = a_0(t) + a_1(t)\cos\theta^*(t), \quad t \in [0, T]. \tag{36}$$

If the Fourier spectra of a_0 and a_1 as a function of θ^* are separable from the spectrum of $a_1\cos\theta^*$ in the θ^*-space, then we can prove that our algorithm will converge to the *exact decomposition* of the original data. We remark that the above requirement is weaker than requiring the separation of spectra in the physical space t since transforming the data into the θ^*-space makes it more adaptive to the signal.

If the above requirement is not satisfied, we can still prove that our algorithm would converge to a decomposition with an error controlled by the degree of separation between the spectra of a_0, a_1 and that of $a_1\cos\theta^*$ in θ^*-space.

4. Numerical results

In this section, we will perform extensive numerical studies to demonstrate the effectiveness of our nonlinear matching pursuit method. First we will present numerical results for the FFT-based algorithms for periodic data or data with a good scale separation property (see Section 6 for the definition of the scale separation property). In the second subsection, we will present numerical results for the l^1 regularized nonlinear matching pursuit which gives reasonably accurate decompositions for non-periodic data and even for under-sampled data or data with missing information in some physical domain.

4.1. Numerical results for the FFT-based algorithms

In this subsection, we present a number of numerical experiments to demonstrate the accuracy and robustness of our FFT-based algorithms. We also compare the performance of our method with that of EMD or EEMD. A main focus of our numerical study is the robustness of the decomposition to signals that are polluted by a significant level of noise. When the signal is free of noise, we observe that the performance of our method is comparable to that of the EMD/EEMD

method. However, when the signal is polluted by noise with a significant noise-to-signal ratio, our nonlinear matching pursuit method tends to give better performance than that of the EMD/EEMD method.

Throughout this section, we denote $X(t)$ as white noise with zero mean and variance $\sigma^2 = 1$. The Signal-to-Noise Ratio (SNR, measured in dB) is defined by

$$\mathrm{SNR[dB]} = 10\log_{10}\left(\frac{\operatorname{var} f}{\sigma^2}\right). \tag{37}$$

We will apply our method to several different signals with increasing level of difficulty.

Example 1. The first example we consider is a simple nonstationary signal consisting of a single IMF, which is given below

$$f(t) = \cos\bigl(60\pi t + 15\sin(2\pi t)\bigr). \tag{38}$$

In Fig. 1, we plot the original signal on the left column and the instantaneous frequency on the right column. The curve with the red color corresponds to the exact instantaneous frequency and the one with the blue color corresponds to the one obtained by our method. The top row corresponds to the original signal without noise, $f(t)$. The middle row corresponds to the same signal with a moderate noise level ($f(t) + X(t)$, SNR $= -3.01$ dB) and the bottom row corresponds to the signal with large noise ($f(t) + 3X(t)$, SNR $= -12.55$ dB). In the case when no noise is added, the instantaneous frequency obtained by our method is almost indistinguishable from the exact one, see the first row of Fig. 1. When the signal has noise, our method can still extract the instantaneous frequency and corresponding IMF with reasonable accuracy, see Fig. 1 and Fig. 2.

In Fig. 2, we compare the IMFs extracted by our method with those obtained by the EMD/EEMD method. For the signal without noise, we use the EMD method to decompose the signal. For the signal with noise, we use the EEMD method to decompose the signal. In the EEMD approach, the number of ensembles is chosen to be 200 and the standard deviation of the added noise is 0.2. In each ensemble, the number of sifting is set to 8. Even though the signal has only a single IMF, the EEMD method still produces several IMFs. Among different components of IMFs that are produced by the EEMD method, we select the one that is closest to the exact IMF in l^2 norm and display it in Fig. 2.

When the signal does not have noise, both our method and the EMD method produce qualitatively the same result for this simple signal, see the first row of Fig. 2. When noise is added, the situation is quite different. The IMFs extracted by our method still have reasonable accuracy. However, the IMF decomposed by EEMD fails to capture the phase of the exact IMF in some region. As a consequence, the accuracy of the instantaneous frequency obtained by the EEMD method is very poor (not shown here).

Example 2. Now, we consider a signal that consists of three IMFs.

$$\begin{aligned} f(t) = {} & \frac{1}{1.5 + \cos(2\pi t)}\cos\bigl(60\pi t + 15\sin(2\pi t)\bigr) + \frac{1}{1.5 + \sin(2\pi t)}\cos\bigl(160\pi t + \sin(16\pi t)\bigr) \\ & + \bigl(2 + \cos(8\pi t)\bigr)\cos\bigl(140\pi (t+1)^2\bigr). \end{aligned} \tag{39}$$

In Fig. 3, we study the accuracy of the instantaneous frequencies obtained by our method with the exact instantaneous frequencies. The upper row corresponds to the signal without noise. As we can see, it is hard to tell any hidden structure from this signal even without noise. Our method recovers the three components of the instantaneous frequencies (blue) that match the exact instantaneous frequencies (red) extremely well. They are almost indistinguishable from each other. In the case when noise is added to the original signal, the polluted signal looks really complicated and one cannot recognize any hidden pattern from this polluted signal. It is quite amazing that our method could still recover the three components of the instantaneous frequencies with accuracy comparable to the noise level, see the bottom row of Fig. 3.

Example 3 *(Length-of-Day data).* Next, we apply our method to the Length-of-Day data, see Fig. 4. The data we adopt here was produced by Gross [12], covering the period from 20 January 1962 to 6 January 2001, for a total of 14,232 days (approximate 39 years). In our previous paper [16], we also studied this data set. Due to the high computational cost associated with the l^1 minimization, we cannot decompose the entire data set. Instead, we decompose a segment of the data that contains 700 consecutive days. Thanks to the low computational cost of the FFT-based nonlinear matching pursuit method, we can now study the entire data set without any compromise.

Fig. 5 displays the first 5 IMFs extracted by the FFT-based method. These IMFs are sorted by their frequencies from high to low. We note that the results obtained by our method do not suffer from the mode mixing phenomenon that is present in the EMD decomposition. Moreover, each component is enforced to be an IMF by the construction of our dictionary. Thus, there is no need to do shifting or post-processing as was done in the EMD or the EEMD method. And the IMFs we obtain match qualitatively those obtained by EEMD with post-processing [37].

It is interesting to note that the each IMF that we obtain has a clear physical interpretation. For example, the period of C_1 is around 14 days, corresponding to the semi-monthly tides. The period of C_2 is about 28 days, corresponding to the monthly tides. Similarly, the period of C_4 is about half a year, corresponding to the semi-annual cycle and C_5 corresponds to the annual cycle.

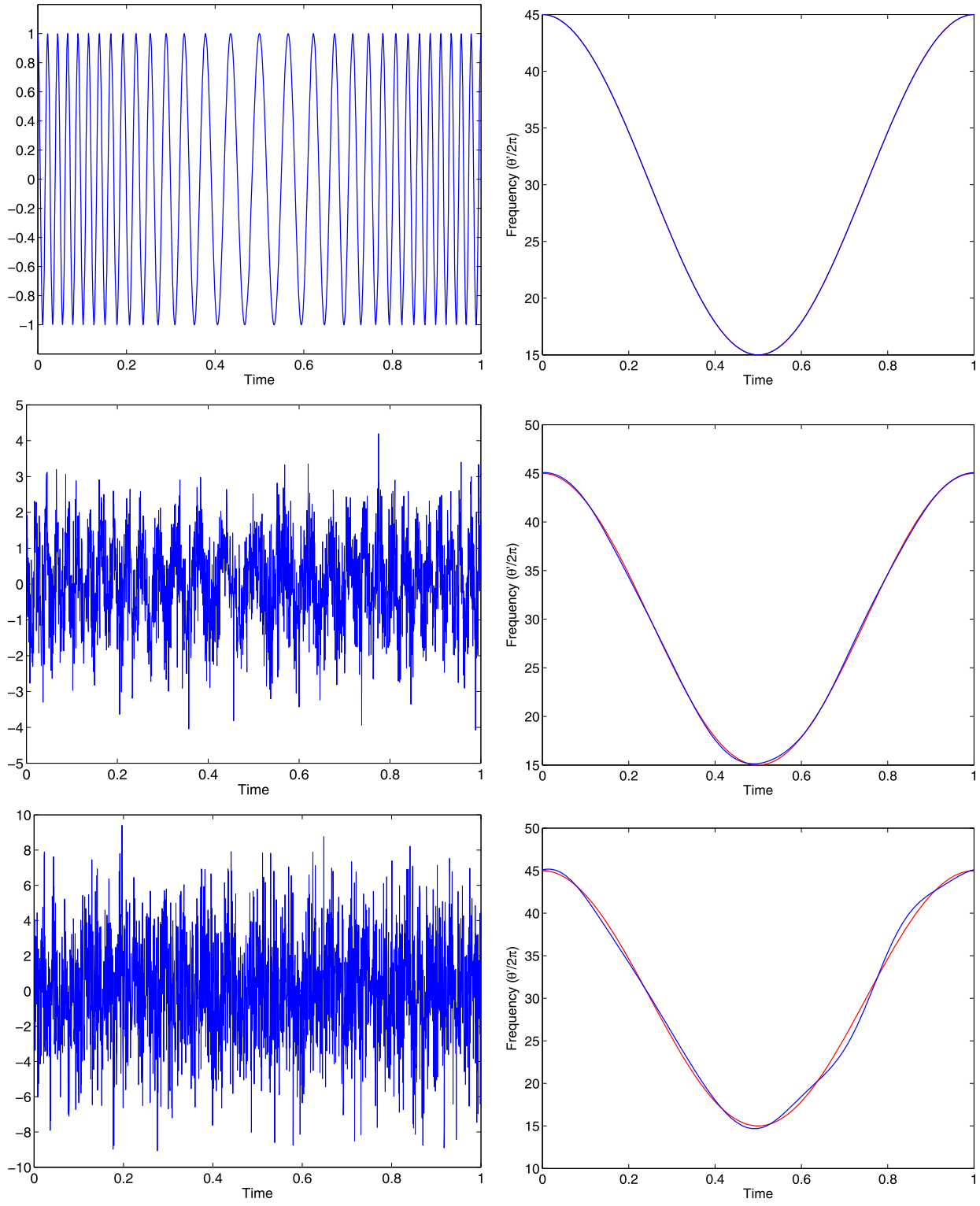

Fig. 1. Top row: left: the original signal defined by (38) without noise; right: instantaneous frequencies; red: exact frequency; blue: numerical results. Middle row: the same as the top row except white noise $X(t)$ is added to the original signal, the corresponding SNR is -3.01 dB. Bottom row: white noise $3X(t)$ is added to the original signal, the corresponding SNR is -12.55 dB. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.1.1. Comparison of CPU time

In this subsection, we compare the computational cost of three methods, the SynchroSqueezing Transform (SST), the EMD/EEMD method, and our Nonlinear Matching Pursuit method (NMP). In Table 1, we give the CPU time of these three methods. The data we use are the same as those given in Examples 1, 2 and 3. For the data given in Examples 1 and 2, the number of sample points is 2048.

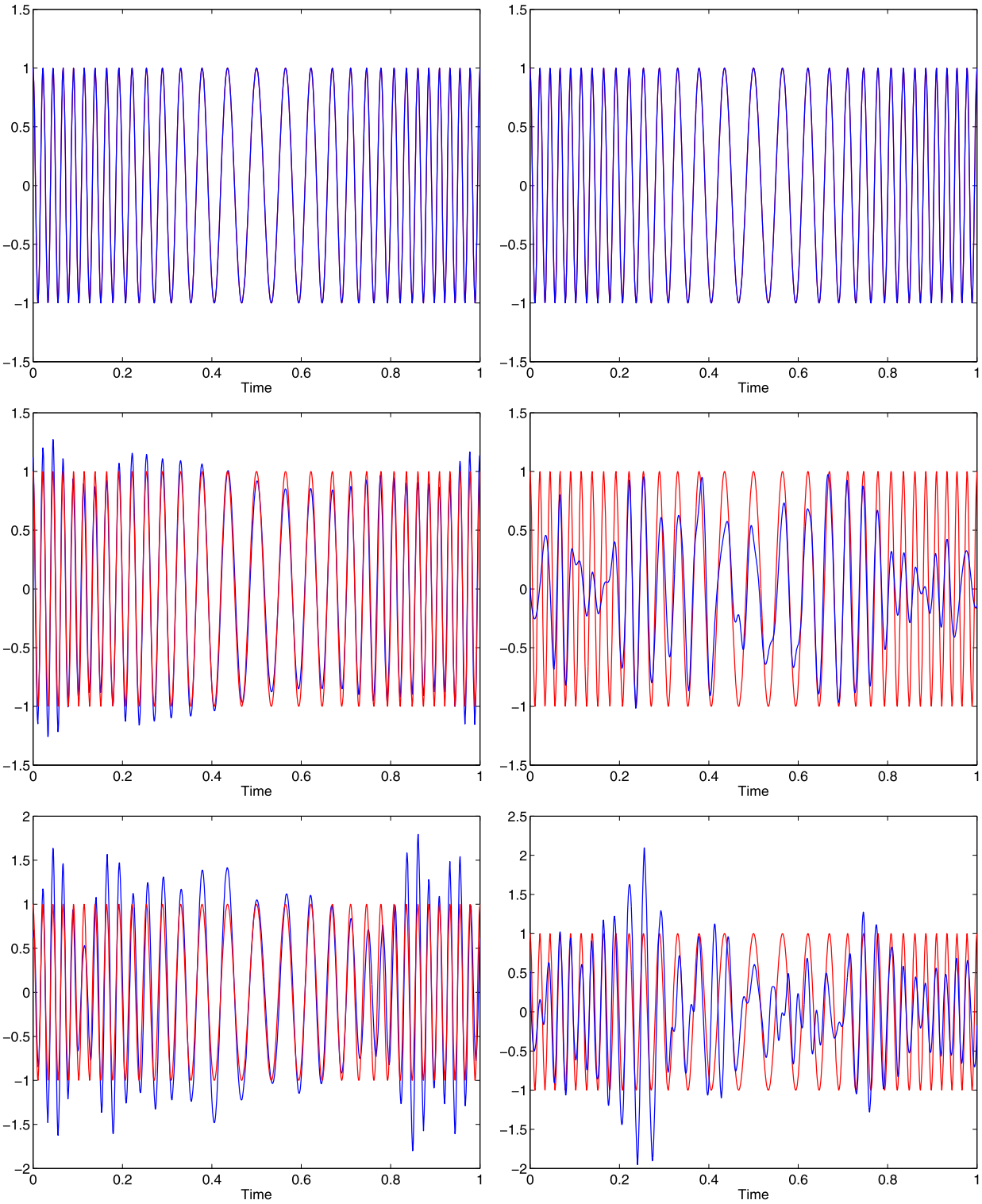

Fig. 2. The IMFs extracted by our method and EMD/EEMD method. Left column: IMFs extracted by our method; right column: IMFs obtained by EMD/EEMD method. Top row: IMFs from $f(t)$; middle row: IMFs from $f(t)+X(t)$; bottom row: IMFs from $f(t)+3X(t)$. $f(t)$ is defined in (38).

For data without noise, our method is faster than SST but slower than EMD. On the other hand, for data with noise, our method is still faster than SST and is much faster than the EEMD method. We remark that our method can give both IMFs and corresponding instantaneous frequencies directly, but one can only get instantaneous frequencies using SST and IMFs using the EMD method. If we are interested in the IMFs in the SST method (or instantaneous frequencies in the EMD method), some post processing method is required to extract the corresponding information. This will give some extra computational cost. This extra computational cost is not accounted for in our comparison given in Table 1. If we consider the overall cost, our method would give more superior performance than that of SST or EMD/EEMD.

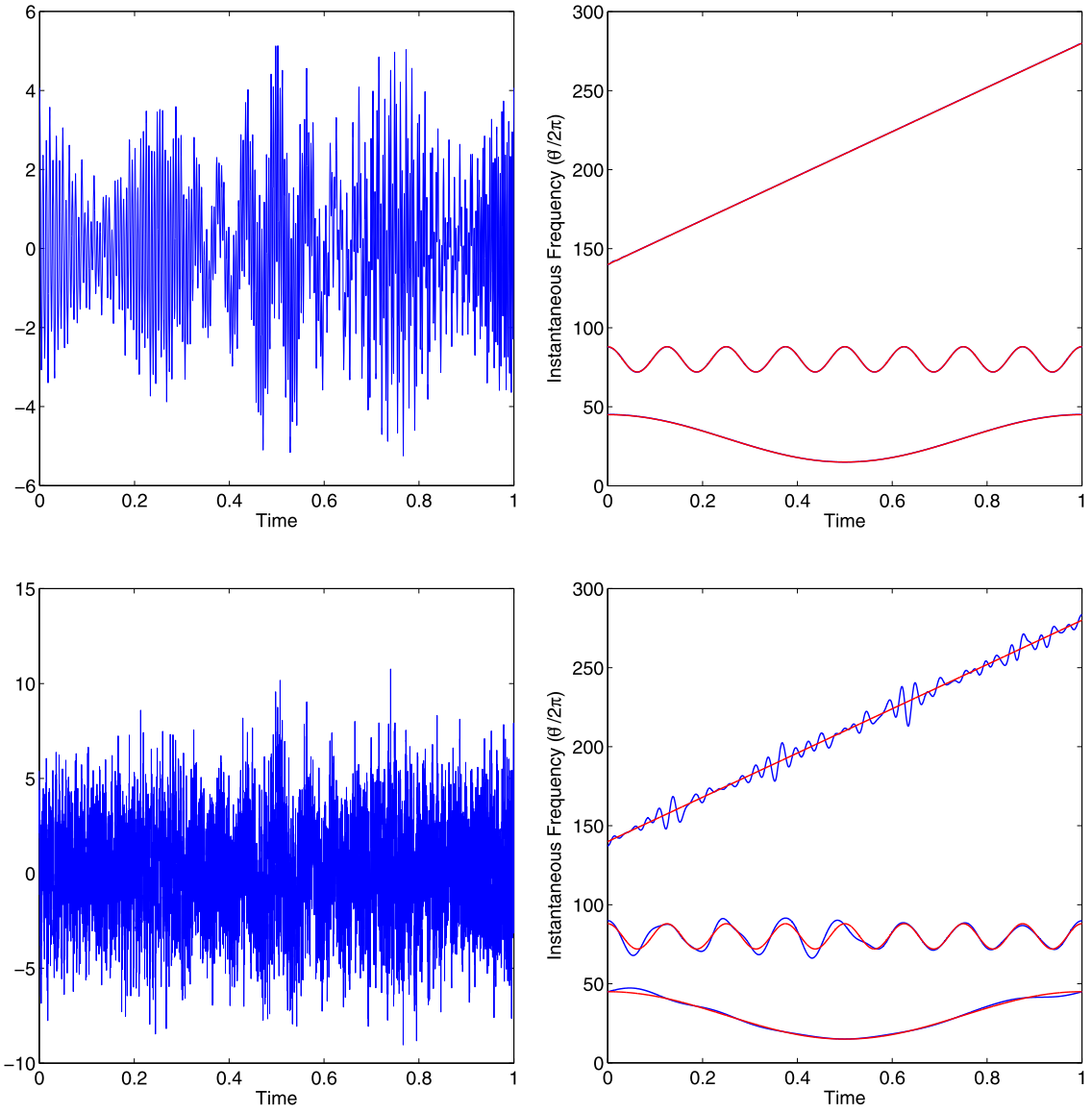

Fig. 3. Upper row: left: the signal defined in (39) without noise; right: instantaneous frequencies; red: exact frequencies; blue: numerical results. Lower row: the same as the upper row except that white noise $2X(t)$ was added to the original signal, the corresponding SNR is -0.8 dB. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

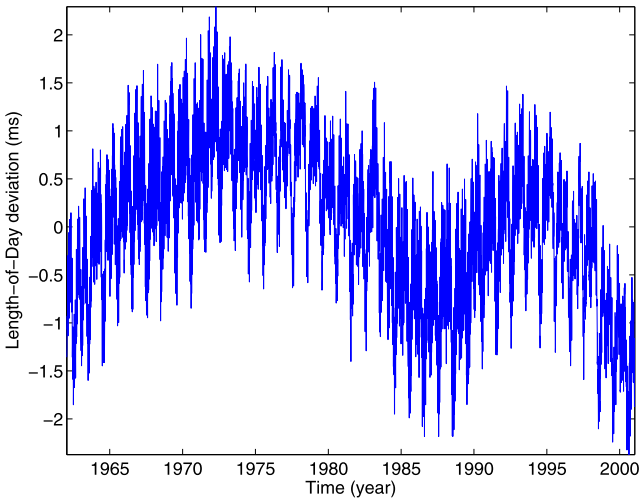

Fig. 4. The daily Length-of-Day data from January 20, 1962 to January 6, 2001.

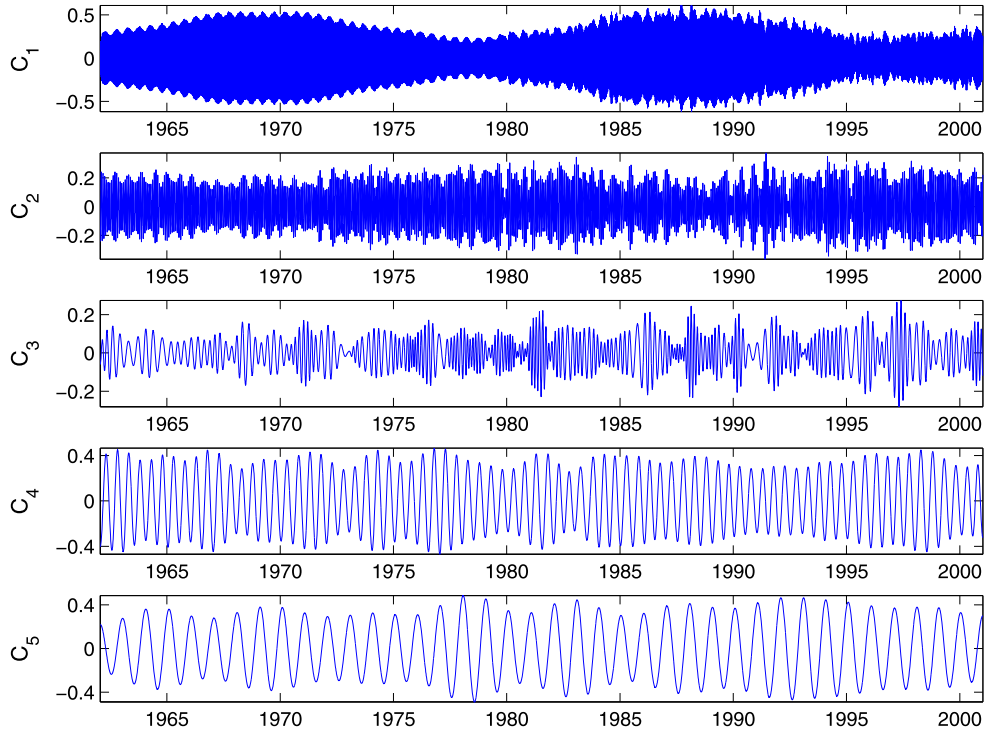

Fig. 5. The first 5 IMFs with highest frequencies given by our FFT-based method.

Table 1
Running time of SST, NMP and EMD/EEMD in seconds. EEMD instead of EMD is used for data with noise and the number of ensembles is equal to 800.

	SST	NMP	EMD/EEMD
Ex1 without noise	1.45	0.22	0.19
Ex1 with noise ($3X$)	1.49	0.31	264
Ex2 without noise	1.53	0.59	0.25
Ex2 with noise ($2X$)	1.47	0.79	248
LOD	11.59	5.94	1024

The computational cost of our method also depends on the initial guess. If the algorithm starts from a poor initial guess, it may require more iterations. Then the computational cost would increase. Moreover, for data with poor scale separation, different initial guesses may lead to different decompositions. In order to abate the effect of the initial guess, we gradually relax the constraint on the phase function θ in our algorithm. Numerical studies show that this procedure works well for signals with good scale separation. But for signals with poor scale separation, looking for a good initial guess is still an important issue that we are currently investigating.

4.2. Numerical results for the l^1 regularized nonlinear matching pursuit

The nonlinear matching pursuit method based on the Fast Fourier Transform works well only for periodic data. For non-periodic data or data with poor scale separation, the results obtained by this method tend to produce some oscillations near the boundary. This so-called "end effect" is also present in the EMD method and other data analysis methods. In our method, the "end effect" comes from the use of the Fourier transform in the algorithm. To remove this "end effect" error, we need to use the l^1 regularized nonlinear matching pursuit described in Section 3.2 with $V(\theta, \lambda)$ being the overcomplete Fourier basis defined in (13).

4.2.1. Numerical results for non-periodic data

In this subsection, we perform a numerical experiment to test the effectiveness of our l^1 regularized nonlinear matching pursuit for non-periodic data. We first consider the following data in our experiment:

$$\theta_1 = 20\pi(t+1)^2 + 1, \qquad \theta_2 = 161.4\pi t + 4(1-t)^2 \sin(16\pi t),$$
$$f(t) = \frac{1}{1.5 + \sin(1.5\pi t)} + (2t+1)\cos\theta_1 + (2-t)^2 \cos\theta_2. \tag{40}$$

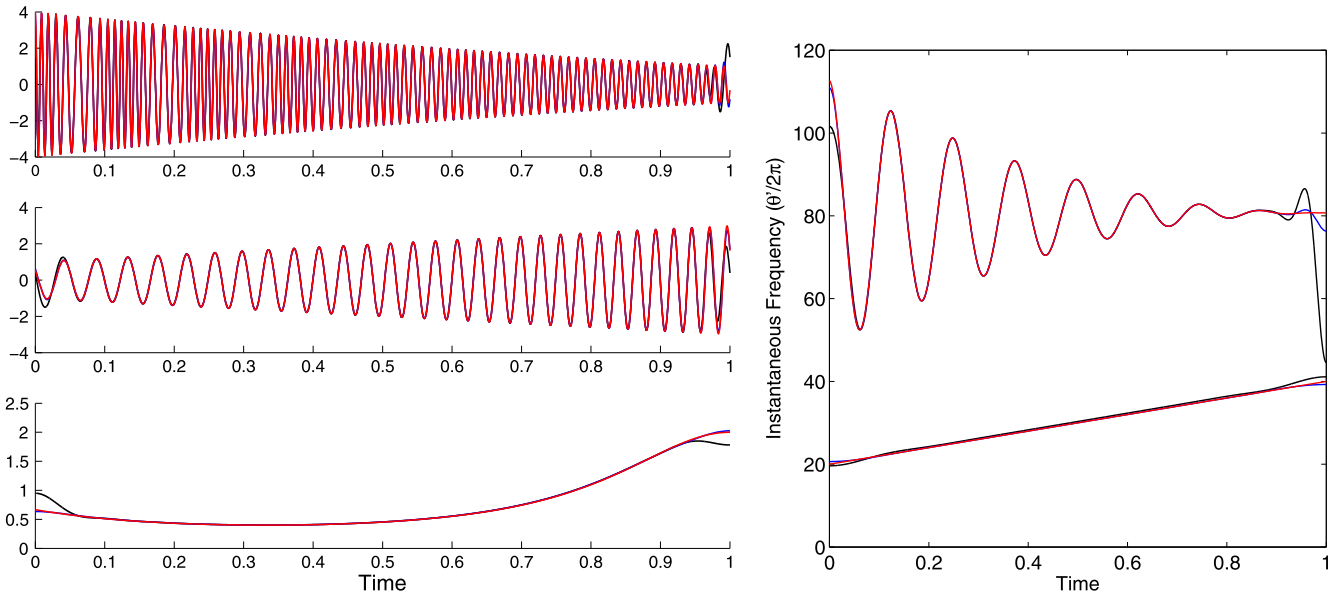

Fig. 6. IMF (left) and instantaneous frequency (right) of the signal in (40) obtained from different methods. Red: exact; blue: l^1 regularized nonlinear matching pursuit; black: FFT-based algorithm. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

In this numerical example, the parameter γ is chosen to be 1. From Fig. 6, we observe that the l^1 regularized nonlinear matching pursuit seems to produce considerably smaller error near the boundary for this non-periodic signal.

This example shows that using the l^1 regularized nonlinear matching pursuit, we can handle non-periodic data reasonably well. On the other hand, the computational cost is considerably higher than that of the FFT-based algorithm due to the extra cost of solving a l^1 regularized least square problem in each iteration. There are various methods to speed up the l^1 optimization. A more efficient l^1 solver could reduce the computational cost significantly. One can also use a hybrid approach by applying the FFT-based algorithm in the majority of the interior domain and using the l^1 minimization only near the boundary of the signal. This would lead to considerable saving.

4.2.2. Numerical results for data with incomplete or sparse samples

Another advantage of the l^1 regularized nonlinear matching pursuit given in Section 3.2 is that it can handle the incomplete data and the data with sparse samples. For these kinds of data, FFT-based algorithm does not work, since we cannot do the Fourier transform. Thus, we need to employ the original algorithm proposed in Section 3.2. In this section, we will use a few examples to illustrate the power of our method to deal with the incomplete data.

The signals considered in this section are all periodic. So in the computations, we use the standard Fourier basis instead of the overcomplete Fourier basis. Although the signal is periodic, we cannot set the regularization parameter $\gamma = 0$, since the sample points cannot well resolve the signal. If $\gamma = 0$, the resulting least square problem may become underdetermined. In the computations of this section, γ is chosen to be 0.1.

The first example is an incomplete signal given by (41).

$$\theta(t) = 120\pi t + 10\cos(4\pi t), \qquad a(t) = 2 + \cos(2\pi t), \qquad f(t) = a(t)\cos\theta(t), \quad t \in [0, 0.4] \cup [0.6, 1]. \tag{41}$$

For this signal, we have only eighty percent of the original data and miss twenty percent of the data in the gap interval $[0.4, 0.6]$. In Fig. 7, we plot the recovered signal in the gap interval $[0.4, 0.6]$ (see the middle panel). The initial guess we use is $\theta_0 = 120\pi t$. The recovered signal matches the original signal almost perfectly in the gap interval. The recovered instantaneous frequency also matches the exact instantaneous frequency with high accuracy.

In Fig. 8, we perform the same numerical experiment by enlarging the interval of missing data from $(0.4, 0.6)$ to $(0.3, 0.7)$, i.e. we miss forty percent of the data. Even for this more challenging example, our method still gives quite reasonable reconstruction of the original data in the region of missing data. The recovered instantaneous frequency still approximates the exact instantaneous frequency with reasonable accuracy, especially away from the region of missing data.

Finally, we consider an example with insufficient samples. The signal is generated as follows:

$$\theta(t_i) = 120\pi t_i + 10\cos(2\pi t_i), \qquad a(t_i) = 2 + \cos(2\pi t_i), \qquad f(t_i) = a(t_i)\cos\theta(t_i), \quad t_i \in [0, 1], \tag{42}$$

and $i = 1, 2, \ldots, N$. The location t_i is chosen randomly in $[0, 1]$. In this example, the number of samples is 64. This means that we have about one sample point within one period of the signal on average.

Fig. 9 gives the results obtained by our method. In the case of insufficient samples without noise, the recovered signal and the original signal are almost indistinguishable.

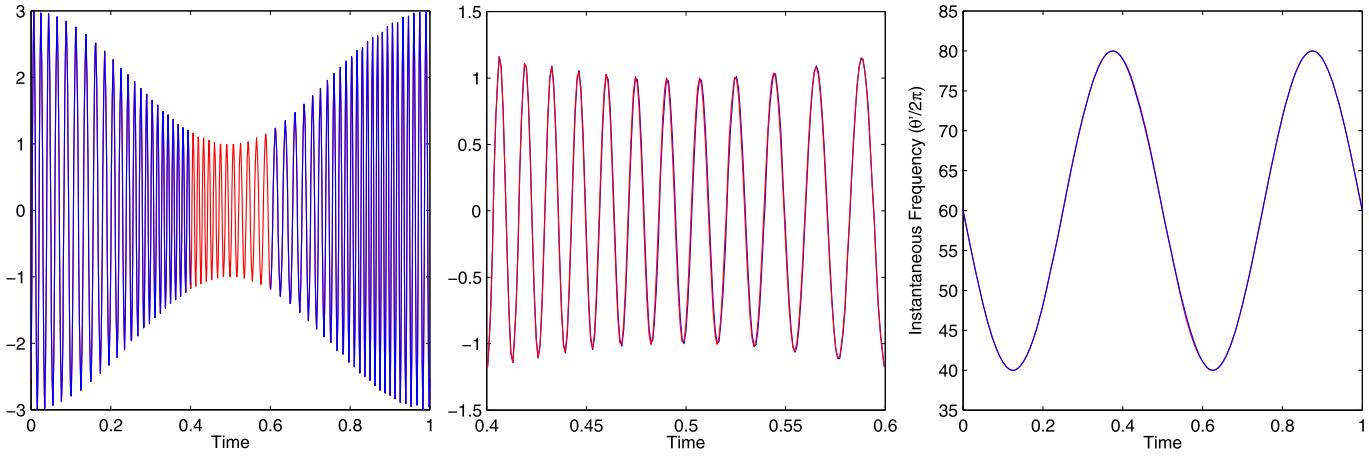

Fig. 7. Left: blue: the original incomplete data, the gap is $(0.4, 0.6)$; red: the missing data recovered by our method. Middle: the recovered missing data, red: exact; blue: numerical. Right: the instantaneous frequencies, red: exact; blue: numerical. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

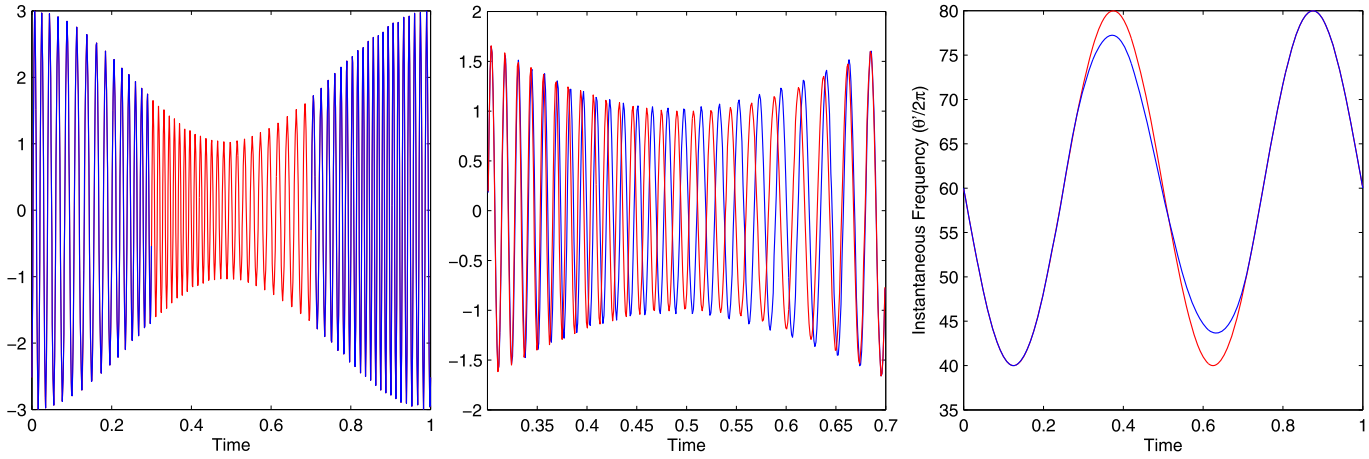

Fig. 8. Left: blue: the original incomplete data, the gap is $(0.3, 0.7)$; red: the missing data recovered by our method. Middle: the recovered missing data, red: exact; blue: numerical. Right: the instantaneous frequencies, red: exact; blue: numerical. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

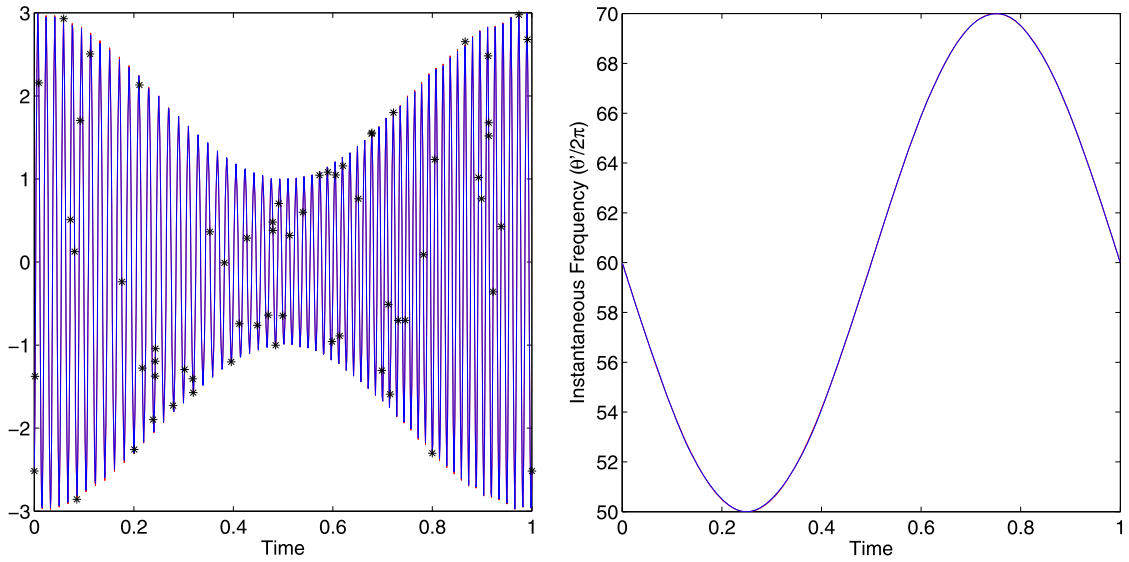

Fig. 9. Left: original samples, red: exact; blue: recovered; '$*$' represent the sample points. Right: instantaneous frequency, red: exact; blue: numerical. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

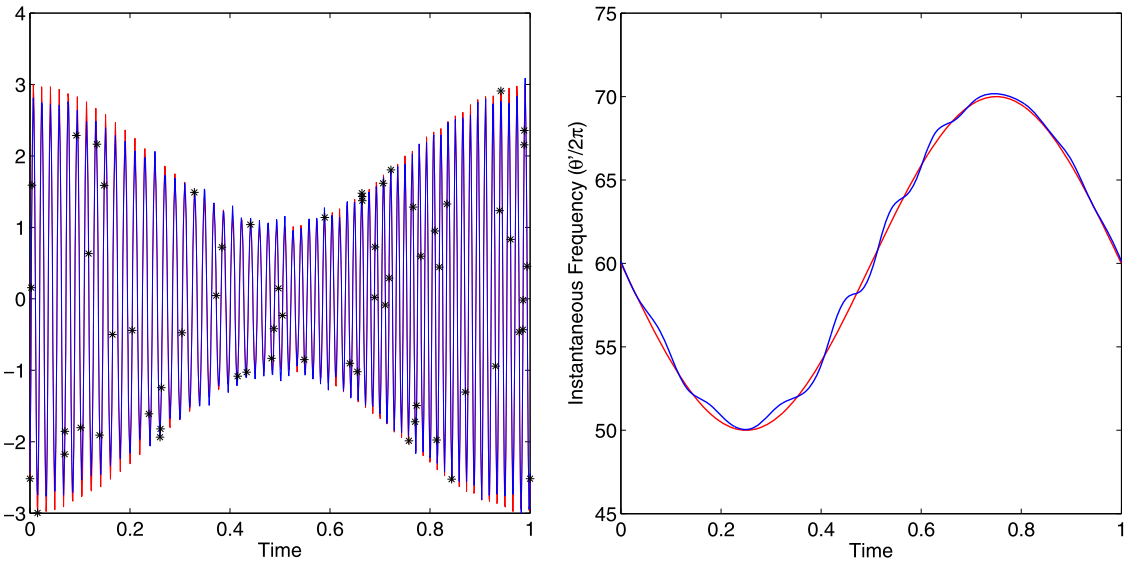

Fig. 10. Left: original samples, red: exact; blue: recovered from the noised data, $f(t_i) + 0.2X(t_i)$; '$*$' represent the sample points. Right: instantaneous frequencies, red: exact; blue: numerical. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

We now add Gaussian noise to the original samples and apply our method to this noisy data. The result is given in Fig. 10. In this case, the noise $0.2X(t)$ is added to the original signal $f(t)$ given in (42). We can see that both the recovered signal and the instantaneous frequency still have reasonable accuracy. This shows that our method is stable with respect to noise perturbation even for sparse under-sampled data.

In the above examples, we both use $120\pi t$ as the initial guess. For some real data with sparse samples, it is challenging to generate a reasonable initial guess since most of the traditional time–frequency analysis methods fail to produce a good result.

5. Generalizations for the l^1 regularized nonlinear matching pursuit

In the previous sections, we propose an iterative algorithm based on l^1 regularized nonlinear matching pursuit and show that for data with a good scale-separation property, this algorithm can give an accurate decomposition. In this section, we will generalize this method to deal with the signal with poor scale separation. More precisely, we consider the signal given as follows:

$$f = \sum_{k=1}^{M} a_k \cos\theta_k, \quad a_k \cos\theta_k \in \mathcal{D}, \tag{43}$$

where $\mathcal{D}$ is defined in (14). But now we do not require that the instantaneous frequencies θ_k' are well separated. As a result, f does not satisfy the scale-separation condition defined in Section 6.

It is well known that for data consisting of components with interfering frequencies, matching pursuit with a Gabor dictionary may not give a sparse decomposition [23]. Since our method is based on matching pursuit, it is not surprising that it may not be able to generate the sparsest decomposition either.

To illustrate, we consider the following signal consisting of two IMFs whose instantaneous frequencies intersect each other. The signal is generated by the analytical formula given below.

$$f(t) = \cos\bigl(20\pi t + 40\pi t^2 + \sin(2\pi t)\bigr) + \cos(40\pi t). \tag{44}$$

Fig. 11 plots the instantaneous frequencies and IMFs recovered by the nonlinear matching pursuit given in the previous section. Near the point of intersection, both the instantaneous frequencies and IMFs produce noticeable errors. The good news is that the instantaneous frequency recovered by our method is still in phase with the exact one. Furthermore, the accuracy is quite reasonable in the region far away from the point of intersection. This shows that our method has a temporal locality property, which is important in many physical applications.

To further improve the accuracy of our decomposition when there are a number of instantaneous frequencies that are not well separated, we need to decompose these components simultaneously since these IMFs have strong correlation. Assume that we can learn from our l^1 regularized nonlinear matching pursuit that there are M components of IMFs whose instantaneous frequencies are not well separated, we modify our decomposition method to find them by solving the following optimization problem:

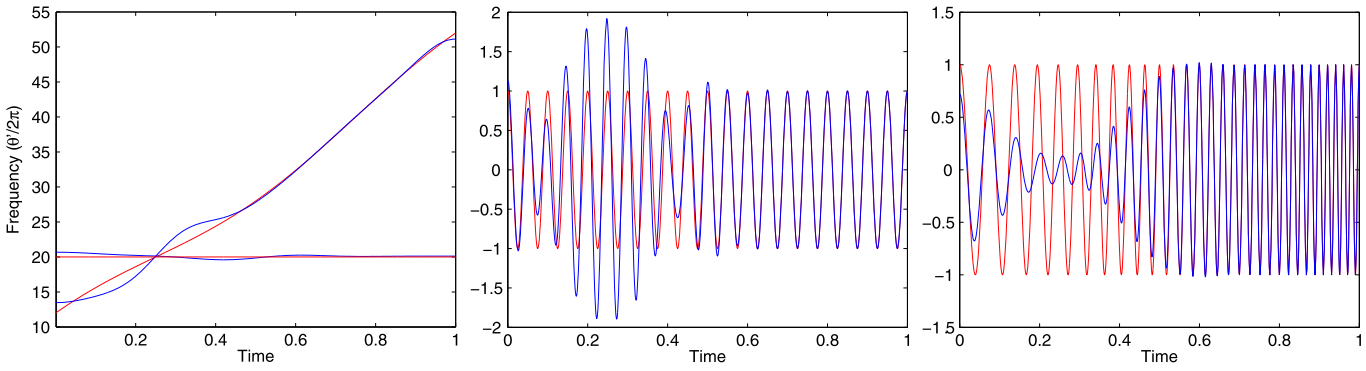

Fig. 11. Left: instantaneous frequencies; red: exact frequencies; blue: numerical results. Middle and right: IMFs extracted by the previous nonlinear matching pursuit. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$\min_{a_k,\theta_k}\left\| f-\sum_{k=1}^{M} a_k\cos\theta_k\right\|_{l^2}^2,\quad \text{such that } a_k\cos\theta_k\in\mathcal{D}, \tag{45}$$

where $\hat{a}_k$ is the representation of a_k in $V(\theta_k,\lambda)$ space. In order to stabilize above nonlinear least square problem, here we also need to add some l^1 terms, then we get following l^1 regularized nonlinear least square problem:

$$\min_{a_k,\theta_k}\left(\gamma\sum_{k=1}^{M}\|\hat{a}_k\|_{l^1}+\left\| f-\sum_{k=1}^{M} a_k\cos\theta_k\right\|_{l^2}^2\right),\quad \text{such that } a_k\cos\theta_k\in\mathcal{D}, \tag{46}$$

where $\gamma>0$ is a regularized parameter and $\hat{a}_k$ is the representation of a_k in $V(\theta_k,\lambda)$ space. Even though this signal is periodic, we cannot drop the l^1 terms since the different components may have strong correlation.

Based on the l^1 regularized nonlinear matching pursuit that we introduced in the previous sections, we propose following iterative method to solve the above optimization problem.

Initialize: $n=0$, $\eta=0$.

Step 1: Solve the following $l^1 1$ regularized least-square problem:

$$\begin{aligned}&\left(a_k^{n+1},b_k^{n+1}\right)\in\operatorname*{Argmin}_{a_k,b_k}\left(\gamma\sum_{k=1}^{M}\left(\|\hat{a}_k\|_{l^1}+\|\hat{b}_k\|_{l^1}\right)+\left\| f-\sum_{k=1}^{M}\left(a_k\cos\theta_k^n+b_k\sin\theta_k^n\right)\right\|_{l^2}^2\right),\\ &\text{subject to:}\quad a_k\in V\left(\theta_k^n,\lambda\right),\ b\in V\left(\theta_k^n,\lambda\right),\end{aligned} \tag{47}$$

where $\hat{a}_k$, $\hat{b}_k$ are the representations of a_k, b_k in the $V(\theta_k^n)$ space.

Step 2: Update θ_k^n:

$$\Delta\theta_k'=P_{V(\theta_k^n;\,\eta)}\left(\frac{d}{dt}\left(\arctan\left(\frac{b_k^{n+1}}{a_k^{n+1}}\right)\right)\right),\qquad \Delta\theta_k=\int_0^t\Delta\theta_k'(s)\,ds,\qquad \theta_k^{n+1}=\theta_k^n-\beta_k\Delta\theta_k, \tag{48}$$

where $\beta_k\in[0,1]$ is chosen to make sure that θ_k^{n+1} is monotonically increasing:

$$\beta_k=\max\left\{\alpha\in[0,1]:\ \frac{d}{dt}\left(\theta_k^n-\alpha\Delta\theta_k\right)\geqslant 0\right\}, \tag{49}$$

and $P_{V(\theta_k^n;\,\eta)}$ is the projection operator to the space $V(\theta_k^n;\eta)$.

Step 3: If $\sum_{k=1}^{M}\|\theta_k^{n+1}-\theta_k^n\|_2>\epsilon_0$, set $n=n+1$ and go to Step 1. Otherwise, go to Step 4.

Step 4: If $\eta\geqslant\lambda$, stop. Otherwise, set $\eta=\eta+\Delta\eta$ and go to Step 1.

Here we give an example to demonstrate that this new method has the capability to deal with the signal with poor scale separation. Fig. 12 gives the results obtained by our new method for the signal given in (44). We can see that both the instantaneous frequencies and IMFs match the exact ones pretty well. These results are much better than those given in Fig. 11.

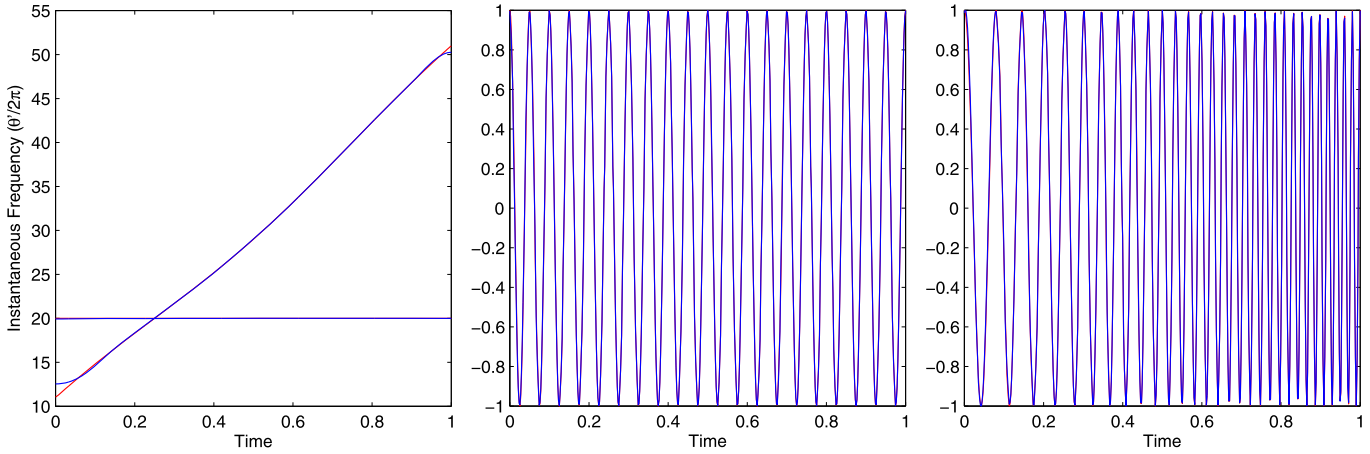

Fig. 12. Left: instantaneous frequencies. Middle (first component) and right (second component): IMFs obtained by extracting two IMFs together. Red: exact results; blue: numerical results. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

6. Error analysis for the data with scale separation

In this section, we perform some error analysis for our nonlinear matching pursuit method. We consider data defined in the entire physical space $\mathbb{R}$. In this setting, we can also employ the FFT-based algorithm developed in Section 3.3 for periodic data to decompose these signals. To guarantee uniqueness of the decomposition, we need to impose certain scale separation property for the data that we try to decompose. Before we state our result, we first define what we mean by scale separation for a given signal.

Definition 6.1 *(Scale separation).* One function $f(t) = a(t)\cos\theta(t)$ is said to satisfy a scale-separation property with a separation factor $\epsilon > 0$, if $a(t)$ and $\theta(t)$ satisfy the following conditions:

$$\begin{aligned} &a(t) \in C^1(\mathbb{R}), \qquad \theta \in C^2(\mathbb{R}),\\ &\inf_{t\in\mathbb{R}} \theta'(t) > 0, \qquad M = \sup_{t\in\mathbb{R}} \left|\theta''(t)\right| < \infty,\\ &\left|\frac{a'(t)}{\theta'(t)}\right|, \left|\frac{\theta''(t)}{(\theta'(t))^2}\right| \leqslant \epsilon, \quad \forall t \in \mathbb{R}. \end{aligned}$$

Definition 6.2 *(Well-separated signal).* A signal $f : \mathbb{R} \to \mathbb{R}$ is said to be well-separated with separation factor ϵ and frequency ratio $d > 1$ if it can be written as

$$f(t) = \sum_{k=1}^{K} a_k(t)\cos\theta_k(t)$$

where all $f_k(t) = a_k(t)\cos\theta_k(t)$ satisfies the scale-separation property with separation factor ϵ, and their phase function θ_k satisfies

$$\theta_k'(t) \geqslant d\theta_{k-1}'(t), \quad \forall t \in \mathbb{R}. \tag{50}$$

Theorem 6.1. *Let $f(t)$ be a function satisfying the scale-separation property with separation factor ϵ and frequency ratio d as defined in Definition 6.2. Choose a low-pass filter ϕ such that its Fourier transform $\hat{\phi}$ has support in $[-\Delta, \Delta]$ with $\Delta < \frac{d-1}{d+1/2}$ and $\hat{\phi}(k) = 1$, $\forall k \in [-\Delta/2, \Delta/2]$. If in the nth step, the approximate phase function $\theta_{k_0}^n$ satisfies the following condition:*

$$\left|\frac{(\theta_{k_0}^n)'(t)}{\theta_{k_0}'(t)} - 1\right| < \frac{\Delta}{2}, \qquad \left|\frac{(\theta_{k_0}^n)''(t)}{((\theta_{k_0}^n)'(t))^2}\right| \leqslant \epsilon, \tag{51}$$

then the error in the $(n+1)$st step in the FFT-based algorithm is of order ϵ, i.e.

$$\left|\theta_{k_0}^{n+1}(t) - \theta_{k_0}(t)\right| = O(\epsilon). \tag{52}$$

In order to prove the above theorem, we need the following lemma:

Lemma 6.1. *For any $a(t) \in C^1(\mathbb{R})$, $\theta \in C^2(\mathbb{R})$, we have*

$$\left| \int a(\tau) e^{-i\theta(\tau)} \phi(\tau - t)\, d\tau - a(t) e^{-i\theta(t)} \hat{\phi}\big(\theta'(t)\big) \right| \leqslant \sup|a'(t)| I_1 + \frac{1}{2}|a(t)| \sup|\theta''(t)| I_2, \tag{53}$$

where $I_n = \int |t^n \phi(t)|\, dt$.

Proof. The proof follows from the following direct calculations:

$$\begin{aligned}
&\left| \int a(\tau) e^{-i\theta(\tau)} \phi(\tau - t)\, d\tau - a(t) e^{-i\theta(t)} \hat{\phi}\big(\theta'(t)\big) \right| \\
&= \left| \int \big(a(\tau) - a(t)\big) e^{i\theta(\tau)} \phi(\tau - t)\, d\tau + a(t) \int \big(e^{-i\theta(\tau)} - e^{-i(\theta(t) - \theta'(t)(\tau - t))}\big) \phi(\tau - t)\, d\tau \right| \\
&= \left| \int \big(a(\tau) - a(t)\big) e^{i\theta(\tau)} \phi(\tau - t)\, d\tau + a(t) \int \big(e^{-i(\theta(\tau) - \theta(t) - \theta'(t)(\tau - t))} - 1\big) e^{-i(\theta(t) - \theta'(t)(\tau - t))} \phi(\tau - t)\, d\tau \right| \\
&\leqslant \sup|a'(t)| \int |\tau \phi(\tau)|\, d\tau + |a(t)| \left| \int \big(e^{-\frac{1}{2} i\theta''(s(\tau))(\tau - t)^2} - 1\big) e^{-i(\theta(t) + \theta'(t)(\tau - t))} \phi(\tau - t)\, d\tau \right| \\
&\leqslant \sup|a'(t)| \int |\tau \phi(\tau)|\, d\tau + |a(t)| \int \left| \frac{1}{2} \theta''\big(s(\tau)\big)(\tau - t)^2 \phi(\tau - t) \right| d\tau \\
&\leqslant \sup|a'(t)| \int |\tau \phi(\tau)|\, d\tau + \frac{1}{2}|a(t)| \sup|\theta''(t)| \int |\tau^2 \phi(\tau)|\, d\tau \\
&= \sup|a'(t)| I_1 + \frac{1}{2}|a(t)| \sup|\theta''(t)| I_2. \qquad \square
\end{aligned} \tag{54}$$

Remark 6.1. We remark that since we typically deal with data of finite support and extend them periodically to the whole domain, the estimates for I_1 and I_2 in the above lemma are effectively taken only in the finite support of the data.

Corollary 6.1. *If the Fourier transform of the low-pass filter ϕ is symmetric, i.e. $\hat{\phi}(k) = \hat{\phi}(-k)$, then we have*

$$\left| \int a(\tau) \cos\big(\theta(\tau)\big) \phi(\tau - t)\, d\tau - a(t) \cos\theta(t) \hat{\phi}\big(\theta'(t)\big) \right| \leqslant \sup|a'(t)| I_1 + \frac{1}{2}|a(t)| \sup|\theta''(t)| I_2, \tag{55}$$

$$\left| \int a(\tau) \sin\big(\theta(\tau)\big) \phi(\tau - t)\, d\tau - a(t) \sin\theta(t) \hat{\phi}\big(\theta'(t)\big) \right| \leqslant \sup|a'(t)| I_1 + \frac{1}{2}|a(t)| \sup|\theta''(t)| I_2. \tag{56}$$

Now we can prove Theorem 6.1.

Proof of Theorem 6.1. In order to simplify the notation, we denote $\bar{\theta} = \theta^n_{k_0}$ and use $\overline{(\cdot)}$ to represent the mapping from t to $\bar{\theta}$, i.e. $\overline{f}(\bar{\theta}) = f(t)$, $\forall f$.

According to our FFT-based algorithm in Section 3.3, we update $\theta^{n+1}_{k_0}$ as follows:

$$\theta^{n+1}_{k_0} = \bar{\theta} - \arctan\left(\frac{b(t)}{a(t)}\right), \quad a(t) = A\big(\bar{\theta}(t)\big),\ b(t) = B\big(\bar{\theta}(t)\big), \tag{57}$$

where

$$A(\gamma) = 2 \int \overline{f}(\bar{\theta}) \cos(\bar{\theta}) \phi(\bar{\theta} - \gamma)\, d\bar{\theta}, \qquad B(\gamma) = 2 \int \overline{f}(\bar{\theta}) \sin(\bar{\theta}) \phi(\bar{\theta} - \gamma)\, d\bar{\theta}. \tag{58}$$

We first estimate $A(\gamma)$ as follows:

$$A(\gamma) = 2 \int \overline{f}(\bar{\theta}) \cos(\bar{\theta}) \phi(\bar{\theta} - \gamma)\, d\bar{\theta} = 2 \sum_{k=1}^{n} \int \bar{a}_k(\bar{\theta}) \cos\theta_k(t) \cos(\bar{\theta}) \phi(\bar{\theta} - \gamma)\, d\bar{\theta}.$$

For $k \neq k_0$, we have

$$2 \int \bar{a}_k(\bar{\theta}) \cos\theta_k(t) \cos(\bar{\theta}) \phi(\bar{\theta} - \gamma)\, d\bar{\theta} = \int \bar{a}_k(\bar{\theta}) \big(\cos\big(\theta_k(t) + \bar{\theta}\big) + \cos\big(\theta_k(t) - \bar{\theta}\big)\big) \phi(\bar{\theta} - \gamma)\, d\bar{\theta}. \tag{59}$$

Since

$$\left|\frac{d\bar{a}_k(\bar{\theta})}{d\bar{\theta}}\right| = \left|\frac{a_k'(t)}{\bar{\theta}'(t)}\right| \leqslant \epsilon \left|\frac{\theta_k'(t)}{\bar{\theta}'(t)}\right|, \tag{60}$$

we obtain

$$\left|\frac{d^2}{d\bar{\theta}^2}\big(\theta_k(t) \pm \bar{\theta}\big)\right| = \left|\frac{\theta_k''(t)\bar{\theta}'(t) - \theta_k'(t)\bar{\theta}''(t)}{(\bar{\theta}'(t))^3}\right| \leqslant \epsilon \left|\frac{(\theta_k'(t))^2 - \theta_k'(t)\bar{\theta}'(t)}{(\bar{\theta}'(t))^2}\right|, \quad \text{for } k \neq k_0. \tag{61}$$

Now we apply Corollary 6.1 for $k \neq k_0$ to obtain

$$\begin{aligned} &2\int \bar{a}_k(\bar{\theta})\cos\theta_k(t)\cos(\bar{\theta})\phi(\bar{\theta}-\phi)\,d\bar{\theta} \\ &\quad = \bar{a}_k(\bar{\theta})\cos\big(\theta_k(t)+\bar{\theta}\big)\hat{\phi}\left(\frac{\theta_k'(t)}{\bar{\theta}'(t)}+1\right) + \bar{a}_k(\bar{\theta})\cos\big(\theta_k(t)-\bar{\theta}\big)\hat{\phi}\left(\frac{\theta_k'(t)}{\bar{\theta}'(t)}-1\right) + O(\epsilon). \end{aligned} \tag{62}$$

Using the condition $\Delta < \frac{d-1}{d+1/2}$, $\theta_k'(t) > d\theta_{k-1}'(t)$ and $|\frac{\bar{\theta}'(t)}{\theta_{k_0}'(t)} - 1| < \frac{\Delta}{2}$, we get

$$\frac{\theta_k'(t)}{\bar{\theta}'(t)} - 1 = \frac{\theta_k'(t)}{\theta_{k_0}(t)}\frac{\theta_{k_0}'(t)}{\bar{\theta}'(t)} - 1 > d(1-\Delta/2) - 1 > \Delta, \quad \text{if } k > k_0, \tag{63}$$

$$\frac{\theta_k'(t)}{\bar{\theta}'(t)} - 1 = \frac{\theta_k'(t)}{\theta_{k_0}(t)}\frac{\theta_{k_0}'(t)}{\bar{\theta}'(t)} - 1 < (1+\Delta/2)/d - 1 < -\Delta, \quad \text{if } k < k_0, \tag{64}$$

$$\frac{\theta_k'(t)}{\bar{\theta}'(t)} + 1 > 1 > \Delta. \tag{65}$$

Since the support of $\hat{\phi}$ is within $[-\Delta, \Delta]$, we have

$$2\int \bar{a}_k(\bar{\theta})\cos\theta_k(t)\cos(\bar{\theta})\phi(\bar{\theta}-\phi)\,d\bar{\theta} = O(\epsilon). \tag{66}$$

For $k = k_0$, we proceed as follows

$$2\int \bar{a}_k(\bar{\theta})\cos\theta_k(t)\cos(\bar{\theta})\phi(\bar{\theta}-\phi)\,d\bar{\theta} = \int \bar{a}_{k_0}(\bar{\theta})\big(\cos\big(\theta_{k_0}(t)+\bar{\theta}\big) + \cos\big(\theta_{k_0}(t)-\bar{\theta}\big)\big)\phi(\bar{\theta}-\phi)\,d\bar{\theta}. \tag{67}$$

Similarly, by using the assumption

$$\left|\frac{d\bar{a}_{k_0}(\bar{\theta})}{d\bar{\theta}}\right| = \left|\frac{a_{k_0}'(t)}{\bar{\theta}'(t)}\right| \leqslant \epsilon \left|\frac{\theta_{k_0}'(t)}{\bar{\theta}'(t)}\right|, \tag{68}$$

we obtain the following estimates:

$$\left|\frac{d}{d\bar{\theta}}\big(\theta_{k_0}(t)+\bar{\theta}\big)\right| = \left|\frac{\theta_{k_0}'(t)}{\bar{\theta}'(t)}+1\right| > 1 > \Delta, \tag{69}$$

$$\left|\frac{d}{d\bar{\theta}}\big(\theta_{k_0}(t)-\bar{\theta}\big)\right| = \left|\frac{\theta_{k_0}'(t)}{\bar{\theta}'(t)}-1\right| < \frac{\Delta}{2}, \tag{70}$$

$$\left|\frac{d^2}{d\bar{\theta}^2}\big(\theta_{k_0}(t)\pm\bar{\theta}\big)\right| = \left|\frac{\theta_{k_0}''(t)\bar{\theta}'(t) - \theta_{k_0}'(t)\bar{\theta}''(t)}{(\bar{\theta}'(t))^3}\right| \leqslant \epsilon \left|\frac{(\theta_{k_0}'(t))^2 - \theta_{k_0}'(t)\bar{\theta}'(t)}{(\bar{\theta}'(t))^2}\right|. \tag{71}$$

By applying Corollary 6.1 again, we get

$$\begin{aligned} &2\int \bar{a}_k(\bar{\theta})\cos\theta_k(t)\cos(\bar{\theta})\phi(\bar{\theta}-\phi)\,d\bar{\theta} \\ &\quad = \bar{a}_{k_0}(\bar{\theta})\cos\big(\theta_{k_0}(t)+\bar{\theta}\big)\hat{\phi}\left(\frac{\theta_{k_0}'(t)}{\bar{\theta}'(t)}+1\right) + \bar{a}_{k_0}(\bar{\theta})\cos\big(\theta_{k_0}(t)-\bar{\theta}\big)\hat{\phi}\left(\frac{\theta_{k_0}'(t)}{\bar{\theta}'(t)}-1\right) + O(\epsilon) \\ &\quad = \bar{a}_{k_0}(\bar{\theta})\cos\big(\theta_{k_0}(t)-\bar{\theta}\big) + O(\epsilon). \end{aligned} \tag{72}$$

Finally, we get the following estimate for $a(t)$,

$$a(t) = A\big(\bar{\theta}(t)\big) = a_{k_0}(t)\cos\big(\theta_{k_0}(t)-\bar{\theta}\big) + O(\epsilon). \tag{73}$$

For $b(t)$, we can obtain a similar estimate

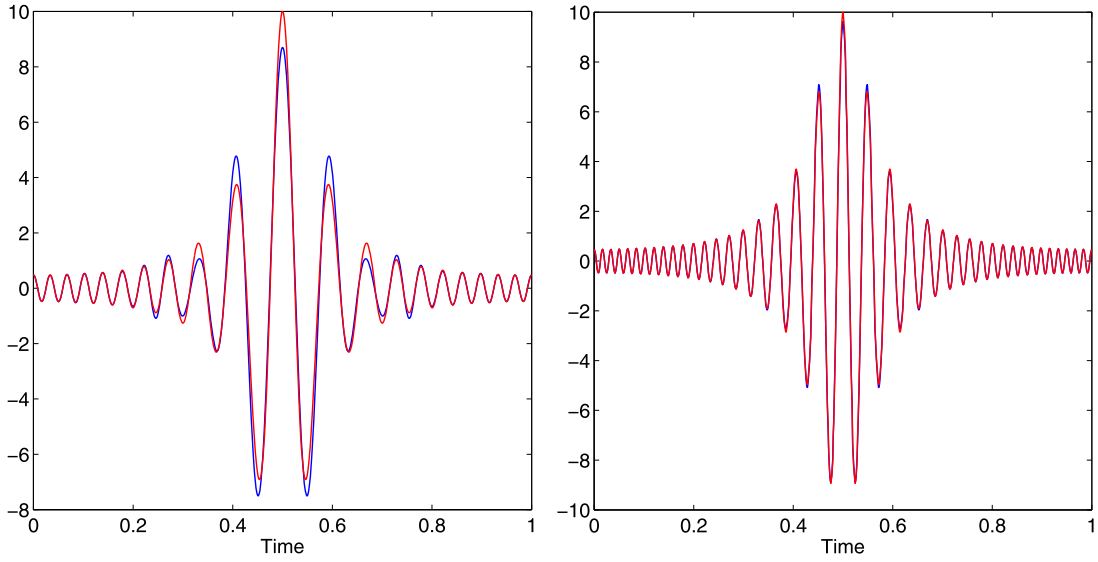

Fig. 13. IMFs with different factors of scale separation. Left: poor scale separation; right: good scale separation.

$$b(t) = B\big(\bar{\theta}(t)\big) = a_{k_0}(t)\sin\big(\theta_{k_0}(t) - \bar{\theta}\big) + O(\epsilon). \tag{74}$$

Thus, we have

$$\Delta\theta = \arctan\left(\frac{B(t)}{A(t)}\right) = \theta_{k_0}(t) - \bar{\theta} + O(\epsilon), \tag{75}$$

which implies that

$$\big|\theta_{k_0}^{n+1}(t) - \theta_{k_0}(t)\big| = O(\epsilon). \tag{76}$$

This completes the proof. □

Remark 6.2. Under the same assumption, we can prove that the error of the instantaneous frequency is also of order ϵ, i.e.

$$\big|\big(\theta_{k_0}^{n+1}\big)'(t) - \theta_{k_0}'(t)\big| = O(\epsilon). \tag{77}$$

The argument is almost the same as the above proof, except that the calculation is a little more involved.

From the above theorem, we can see that the accuracy of our method depends on the factor of scale separation. This is also consistent with our numerical results. In the following numerical example, we compare the IMFs obtained by our method for two different signals. One has poor scale separation, the other one has better scale separation. The signals are given by (78). The signal f_2 has a better scale separation property than f_1 since its instantaneous frequency is twice of that of f_1. As shown in Fig. 13, the error we obtain for f_2 is considerably smaller than that of f_1.

$$\begin{aligned} &a_0(t) = a_1(t) = \frac{1}{1.1 + \cos(2\pi t)}, \qquad \theta = 10\sin(2\pi t) + 40\pi t, \\ &f_1(t) = a_0(t) + a_1(t)\cos\theta(t), \qquad f_2(t) = a_0(t) + a_1(t)\cos\big(2\theta(t)\big). \end{aligned} \tag{78}$$

We would like to point out that the error estimate given by Theorem 6.1 is highly over-estimated. In Fig. 13, we can see that even for the signal with a poor scale separation property, the instantaneous frequency we obtain is still reasonably accurate, although the corresponding scale separation factor $\epsilon \approx 1.4$ is quite big according to Definition 6.1.

In the estimate of Lemma 6.1, instead of taking the supreme over $\mathbb{R}$, we can take supreme over a finite interval, since we can choose a low-pass filter ϕ that decays exponentially fast. Then, we can get a more local estimate:

$$\left|\int a(\tau)e^{i\theta(\tau)}\phi(\tau - t)\,d\tau - a(t)e^{i\theta(t)}\hat{\phi}\big(\theta'(t)\big)\right| \leqslant \sup_{t\in S_\phi}\big|a'(t)\big|I_1 + \frac{1}{2}\big|a(t)\big|\sup_{t\in S_\phi}\big|\theta''(t)\big|I_2, \tag{79}$$

where $S_\phi = \{t \in \mathbb{R}\colon\ |\phi(t)| > \epsilon\}$.

This seems to suggest that for the signal that does not have a good scale separation property in some region, its influence on the accuracy of the decomposition is limited to that region. This is the temporal locality property we have mentioned in Section 5.1. This property can be also seen in Fig. 13. In this example, the scales are not well separated in the center of the

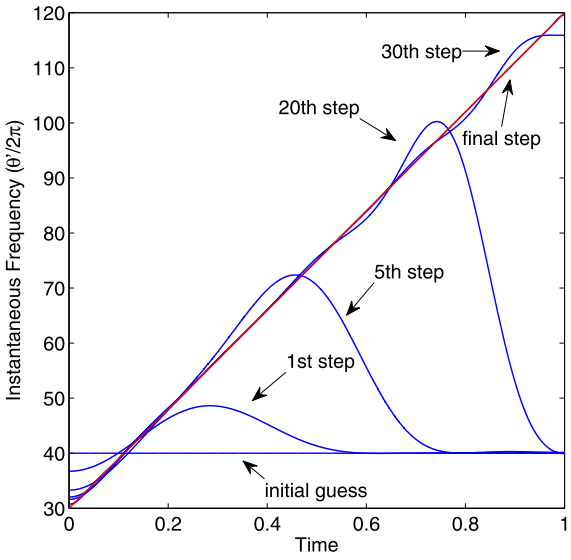

Fig. 14. Instantaneous frequency at different steps.

interval. However, the scales are better separated near the two ends. We can see that the error near the boundary is much smaller than that in the center.

From this analysis, we can also see that the low-pass filter with smooth Fourier spectrum (such as the cosine function given by (32)) would perform better than that with discontinuous spectrum (such as the stair function given by (31)) in terms of maintaining the temporal locality property of the decomposition. The low-pass filter with discontinuous spectrum decays much slower in the time domain due to the Gibbs phenomena. This is why we use the cosine low-pass filter instead of the stair one.

Theorem 6.1 tells us that if we have a good initial guess, then there is no need to do iterations. But in most cases, we have only a rough initial guess, and the condition in Theorem 6.1 may not be satisfied. In this case, the iterative procedure in our algorithm improves the result gradually as the number of iterations increases. Our method will generate a good approximation after a number of iterations. In Fig. 14, we show how the iteration can improve the approximation of the instantaneous frequency for a simple chirp signal: $f(t) = \cos(10\pi(3t+1)^2)$.

In this example, the initial guess for the instantaneous frequency is a constant, $\theta_0 = 80\pi t$. Near the intersection of $\theta_0' = 80\pi$ and the exact instantaneous frequency, the initial guess is relatively good. After one step, we can get a better approximation in this local region. This new estimate gives us a better guess for the instantaneous frequency in a slightly larger interval that contains the good interval given by the initial guess. Then in the next step, we can get a good approximation in an even larger interval. Gradually, we can get an accurate approximation of the instantaneous frequency in the whole interval. Fig. 14 plots the approximate instantaneous frequency in different steps. As the number of iterations increases, the region in which we have a good approximation becomes larger and larger. Finally, the iterative algorithm produces an accurate instantaneous frequency in the entire domain.

7. Conclusion

In this paper, we introduce a new data-driven time–frequency analysis method based on the nonlinear matching pursuit. The adaptivity of our decomposition is obtained by looking for the sparsest representation of signals in the time–frequency domain from a largest possible dictionary that consists of all possible candidates for Intrinsic Mode Functions (IMFs). Solving this nonlinear optimization problem is in general very difficult. We propose a nonlinear matching pursuit method to solve this nonlinear optimization problem by generalizing matching pursuit for the l^0 optimization problem. One important advantage of this nonlinear matching pursuit method is it can be implemented very efficiently. Further, this approach is very stable to noise. For data with good scale separation property, our method gives an accurate decomposition up to the boundary.

We have also carried out some theoretical study for the nonlinear optimization method proposed in this paper. In the case when the signal satisfies certain scale separation conditions, we show that our iterative algorithm gives an approximate decomposition with the accuracy determined by the scale separation factor of the signal.

There are some remaining issues to be studied in the future, such as data with poor scale separation property, the so-called 'end effect' of data, and data with incomplete or sparse samples and so on. We have addressed these issues to some extent in this paper, but much more work need to be done to resolve these challenging issues.

Another important problem is to decompose data with intra-wave frequency modulation. This type of data is known to be very challenging. Traditional data analysis methods tends to introduce many harmonics and could not capture the intrinsic features of these data. Direct application of the methods proposed in this paper does not yield satisfactory results either

since these data do not have a good scale separation property. Recently we have made some progress in decomposing type of data. Specifically, we need to enlarge our dictionary by incorporating additional elements consisting of periodic functions of different shapes. We then modify our decomposition method to look for the appropriate shape function that gives rise to the sparsest decomposition of the signal. Our numerical preliminary study on both synthetic and real data suggest this algorithm works well for a class of signals whose dominant component is generated by a single shape function. This method is also stable under noise perturbation. Since this method is quite different from the ones presented in this paper, we will report this work in a subsequent paper.

Another direction is to generalize this adaptive data analysis method to high dimensional data. In some physical applications such as propagation of nonlinear ocean waves, each waveform has a dominating propagation direction. In this case, our method has a natural generalization by adopting a multi-dimensional phase function.

Acknowledgments

This work was in part supported by the AFOSR MURI grant FA9550-09-1-0613. We would like to thank Professors Norden E. Huang and Zhaohua Wu for many stimulating discussions on EMD/EEMD and topics related to the research presented here. We would also like to thank Professors Ingrid Daubechies, Stanley Osher, and Zuowei Shen for their interest in this work and for a number of valuable discussions. Prof. Hou would like to express his gratitude to the National Central University (NCU) for their support and hospitality during his visits to NCU in the past two years. We also like to thank the two anonymous reviewers for their constructive comments and suggestions which help to improve the quality of this paper.

References

[1] B. Boashash, Time–Frequency Signal Analysis: Methods and Applications, Longman–Cheshire/John Wiley Halsted Press, Melbourne/New York, 1992.
[2] A.M. Bruckstein, D.L. Donoho, M. Elad, From sparse solutions of systems of equations to sparse modeling of signals and images, SIAM Rev. 51 (2009) 34–81.
[3] F.R. Bach, R. Jenatton, J. Mairal, G. Obozinski, Optimization with sparsity-inducing penalties, Found. Trends Mach. Learn. 4 (1) (2012) 1–106.
[4] E. Candès, T. Tao, Near optimal signal recovery from random projections: Universal encoding strategies?, IEEE Trans. Inform. Theory 52 (12) (2006) 5406–5425.
[5] E. Candes, J. Romberg, T. Tao, Robust uncertainty principles: Exact signal recovery from highly incomplete frequency information, IEEE Trans. Inform. Theory 52 (2006) 489–509.
[6] S. Chen, D. Donoho, M. Saunders, Atomic decomposition by basis pursuit, SIAM J. Sci. Comput. 20 (1998) 33–61.
[7] P.L. Combettes, J.-C. Pesquet, Proximal splitting methods in signal processing, in: H.H. Bauschke, R. Burachik, P.L. Combettes, V. Elser, D.R. Luke, H. Wolkowicz (Eds.), Fixed-Point Algorithms for Inverse Problems in Science and Engineering, Springer-Verlag, New York, 2010, pp. 185–212.
[8] I. Daubechies, Ten Lectures on Wavelets, CBMS–NSF Regional Conf. Ser. in Appl. Math., vol. 61, SIAM Publications, 1992.
[9] I. Daubechies, J. Lu, H. Wu, Synchrosqueezed wavelet transforms: an empirical mode decomposition-like tool, Appl. Comput. Harmon. Anal. 30 (2011) 243–261.
[10] D.L. Donoho, Compressed sensing, IEEE Trans. Inform. Theory 52 (2006) 1289–1306.
[11] P. Flandrin, Time–Frequency/Time–Scale Analysis, Academic Press, San Diego, CA, 1999.
[12] R.S. Gross, Combinations of Earth orientation measurements: SPACE2000, COMB2000, and POLE2000, JPL Publication 01-2, Jet Propulsion Laboratory, Pasadena, CA, 2001.
[13] R. Gribonval, M. Nielsen, Sparse representations in unions of bases, IEEE Trans. Inform. Theory 49 (12) (2003) 3320–3325.
[14] D. Gabor, Theory of communication, J. IEE 93 (1946) 426–457.
[15] Tom Goldstein, Stanley Osher, The split Bregman method for L_1-regularized problems, SIAM J. Imaging Sci. 2 (2009) 323–343.
[16] T.Y. Hou, Z. Shi, Adaptive data analysis via sparse time–frequency representation, Adv. Adapt. Data Anal. 3 (2011) 1–28.
[17] T.Y. Hou, Z. Shi, P. Tavallali, On the stability and convergence of a data-driven time-frequency decomposition method, preprint.
[18] N.E. Huang, et al., The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis, Proc. R. Soc. Lond. Ser. A 454 (1998) 903–995.
[19] D.L. Jomes, T.W. Parks, A high resolution data-adaptive time–frequency representation, IEEE Trans. Acoust. Speech Signal Process. 38 (1990) 2127–2135.
[20] P.J. Loughlin, B. Tracer, On the amplitude- and frequency-modulation decomposition of signals, J. Acoust. Soc. Am. 100 (1996) 1594–1601.
[21] B.C. Lovell, R.C. Williamson, B. Boashash, The relationship between instantaneous frequency and time–frequency representations, IEEE Trans. Signal Process. 41 (1993) 1458–1461.
[22] S. Mallat, Z. Zhang, Matching pursuit with time–frequency dictionaries, IEEE Trans. Signal Process. 41 (1993) 3397–3415.
[23] S. Mallat, A Wavelet Tour of Signal Processing: The Sparse Way, Academic Press, 2009.
[24] W.K. Meville, Wave modulation and breakdown, J. Fluid Mech. 128 (1983) 489–506.
[25] D. Needell, J. Tropp, CoSaMP: Iterative signal recovery from noisy samples, Appl. Comput. Harmon. Anal. 26 (2008) 301–321.
[26] D. Needell, R. Vershynin, Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit, Found. Comput. Math. 9 (3) (2009) 317–334.
[27] S. Olhede, A.T. Walden, The Hilbert spectrum via wavelet projections, Proc. R. Soc. Lond. Ser. A 460 (2004) 955–975.
[28] B. Picinbono, On instantaneous amplitude and phase signals, IEEE Trans. Signal Process. 45 (1997) 552–560.
[29] J.C. Pesquet, N. Pustelnik, A parallel inertial proximal optimization method, Pac. J. Optim. 8 (2) (2012) 273–305.
[30] S. Qian, D. Chen, Joint Time–Frequency Analysis: Methods and Applications, Prentice Hall, 1996.
[31] S. Qian, D. Chen, Signal representation using adaptive normalized Gaussian functions, Signal Process. 36 (1994) 1–11.
[32] S.O. Rice, Mathematical analysis of random noise, Bell Syst. Tech. J. 23 (1944) 282–310.
[33] J. Shekel, Instantaneous frequency, Proc. IRE 41 (1953) 548.
[34] S. Setzer, G. Steidl, T. Teuber, Deblurring Poissonian images by split Bregman techniques, J. Vis. Commun. Image Represent. 21 (3) (2010) 193–199.
[35] J. Tropp, A. Gilbert, Signal recovery from random measurements via orthogonal matching pursuit, IEEE Trans. Inform. Theory 53 (2007) 4655–4666.
[36] B. Van der Pol, The fundamental principles of frequency modulation, Proc. IEE 93 (1946) 153–158.
[37] Z. Wu, N.E. Huang, Ensemble empirical mode decomposition: a noise-assisted data analysis method, Adv. Adapt. Data Anal. 1 (2009) 1–41.