Advances in Data Science and Adaptive Analysis
Vol. 9, No. 4 (2017) 1750010 (17 pages)

DOI: 10.1142/S2424922X17500103

World Scientific
www.worldscientific.com

Sparse Time-Frequency Decomposition for Multiple Signals with Same Frequencies

Thomas Y. Hou

Applied and Computational Mathematics, MC 9-94
Caltech, Pasadena, California 91125, USA
hou@cms.caltech.edu

Zuoqiang Shi*

Yau Mathematical Sciences Center
Tsinghua University, Beijing 100084, P. R. China
zqshi@tsinghua.edu.cn

Received 20 August 2017
Accepted 15 September 2017
Published 3 November 2017

In this paper, we consider multiple signals sharing the same instantaneous frequencies. This kind of data is very common in scientific and engineering problems. To take advantage of this special structure, we modify our data-driven time-frequency analysis by updating the instantaneous frequencies simultaneously. Moreover, based on the simultaneous sparsity approximation and the Fast Fourier Transform, we develop several efficient algorithms to solve this problem. Since the information of multiple signals is used, this method is very robust to the perturbation of noise and it is applicable to the general nonperiodic signals even with missing samples or outliers. Several synthetic and real signals are used to demonstrate the robustness of this method. The performances of this method seems quite promising.

Keywords: Sparse time-frequency decomposition; instantaneous frequency; multiple measurements; simultaneous sparsity approximation.

1. Introduction

Data provide a lot of important information for us to understand the world around us. In many applications, the frequencies of a signal are usually very useful to reveal the underlying physical mechanism. Therefore, in the past several decades, a lot of effort has been devoted to find efficient time frequency analysis methods and powerful methods. These numerical methods include the windowed Fourier transform [Mallat (2009)], the wavelet transform [Daubechies (1992); Mallat (2009)], the Wigner–Ville distribution [Flandrin (1999)], etc. In particular, the Empirical Mode

*Corresponding author.

Decomposition (EMD) method developed by Professor Huang *et al.* [1998] and Wu and Huang [2009] provides an efficient and fully adaptive method to extract frequency information from complicate multiscale data. The EMD method has found many applications in many scientific and engineering disciplines and has inspired a number of subsequent developments, see, e.g. the synchrosqueezed wavelet transform [Daubechies *et al.* (2011)], the Empirical wavelet transform [Dragomiretskiy and Zosso (2014)], and the variational mode decomposition [Gilles (2013)].

Inspired by the EMD method and compressive sensing [Candes *et al.* (2006); Candès and Tao (2006); Donoho (2006)], Hou and Shi proposed a data-driven time-frequency analysis method based on the sparsest time-frequency representation [Hou and Shi (2013); Hou *et al.* (2014)]. In this method, we decompose signals that consist of a finite number of intrinsic mode functions in the following form:

$$f(t) = \sum_{k=1}^{K} a_k(t) \cos \theta_k(t), \quad t \in [0, 1], \tag{1}$$

where $a_k(t), \theta_k(t)$ are smooth functions, $\theta_k'(t) > 0, k = 1, \ldots, M$. We assume that $a_k(t)$ and θ_k' are "less oscillatory" than $\cos \theta_k(t)$. We say that a signal $f(t)$ is less oscillatory than $g(t)$ if $\|\frac{d}{dt} f(t)\|_{L^2} \leq \lambda \|\frac{d}{dt} g(t)\|_{L^2}$ for some positive constant $\lambda < 1/2$. After applying a change of variable in time, we can always map the time domain of this signal into the time interval $[0, 1]$. Without loss of generality, we may assume that the time span of the interval is $[0, 1]$. We borrow the terminology in the EMD method and call $a_k(t) \cos \theta_k(t)$ as the Intrinsic Mode Functions (IMFs) Huang *et al.* [1998]. This model is also known as the Adaptive Harmonic Model (AHM) and has been widely used in the time-frequency analysis literature [Chui and Mhaskar (2016); Daubechies *et al.* (2011)].

How to compute the decomposition in the AHM model is a major challenge. A major difficulty is due to the fact that there are a large number of degrees of freedom in a signal. As a result, the decomposition of a signal using the AMH model is typically not unique. One essential problem is how to set up a criterion to pick up the "best" decomposition and this criterion should be easy to compute in practice. Inspired by the compressive sensing, we proposed to decompose a multiscale signal by looking for the sparsest decomposition in Hou and Shi [2013]. And the sparsest decomposition is obtained by solving a nonlinear optimization problem formulated as follows:

$$\min_{(a_k)_{1\leq k\leq K}, (\theta_k)_{1\leq k\leq M}} K, \quad \text{Subject to:} \quad f = \sum_{k=1}^{K} a_k \cos \theta_k, \quad a_k \cos \theta_k \in \mathcal{D}, \tag{2}$$

where $\mathcal{D}$ is the dictionary consisting of all IMFs. By choosing the dictionary this way, we automatically impose the AHM model in our decomposition (see Hou and Shi [2013] for precise definition of the dictionary).

We proposed two different methods to solve the optimization problem (2). The first one is based on matching pursuit [Hou and Shi (2013); Mallat and Zhang (1993)] and the other one is based on basis pursuit [Chen *et al.* (1998); Hou and

Shi (2016)]. The convergence of the algorithm based on matching pursuit has been established under the certain scale separation assumption [Hou *et al.* (2014)].

In our previous works, we focussed on the case when only one signal is given. However, in many applications, we can obtain many different signals and these signals have the same frequency structure. For example, when monitoring the safety of a building, people usually put many sensors in different locations of the same building to measure the vibration. From these sensors, many measurements of the vibration are obtained. Since the signals from different sensors measure the same building, they should have the same instantaneous frequencies associated with the natural frequencies of the building. If we analyze the signals from different sensors individually, we would not take full advantage of the property that these signals share the same instantaneous frequencies. By analyzing these different signals simultaneously, we can take full advantage of this property to obtain more robust and efficient decomposition methods.

The main purpose of this paper is to introduce several novel decomposition methods that simultaneously decompose different signals that share the same frequency structure. To begin with, we assume that we are given multiple signals that share the same instantaneous frequencies. To take advantage of this special property, we first modify the adaptive harmonic model (1) to deal with this kind of signals.

$$f^j(t) = \sum_{k=1}^{K} a_k^j(t) \cos\theta_k(t), \quad t \in [0,1], \quad j = 1, \ldots, M, \tag{3}$$

where M is the number of signals, f^j is the jth signal. For each $j = 1, \ldots, M$ fixed, it is the AHM model (1), i.e. $a_k^j(t), \theta_k(t)$ are smooth functions, and we assume that the instantaneous frequencies are positive, i.e. $\theta_k'(t) > 0, k = 1, \ldots, K$. As before, we assume that $a_k^j(t)$ and θ_k' are "less oscillatory" than $\cos\theta_k(t)$. The main feature of this model is that different signals f^j have the same phase functions θ_k. We call (3) the Multiple Adaptive Harmonic Model (MAHM).

Inspired by our previous work for a single signal, we also look for the sparsest decomposition that satisfies the MAHM model (3) by solving the following optimization problem:

$$\min_{(a_k^j, \theta_k)_{k=1,\ldots,K}^{j=1,\ldots,M}} \quad K \tag{4}$$

$$\text{Subject to:} \quad f^j(t) = \sum_{k=1}^{K} a_k^j(t) \cos\theta_k(t), \quad a_k^j(t)\cos\theta_k(t) \in \mathcal{D}.$$

The dictionary $\mathcal{D}$ that we will use in this paper is the same as that in Hou and Shi [2013]. It can be written as

$$\mathcal{D} = \{a\cos\theta : a \in V(\theta,\lambda), \theta' \in V(\theta,\lambda), \text{ and } \theta'(t) \geq 0, \ \forall\, t \in [0,1]\}. \tag{5}$$

Here, $V(\theta,\lambda)$ is the collection of all the functions that are "less oscillatory" than $\cos\theta(t)$. In general, it is most effective to construct $V(\theta,\lambda)$ as an over-complete

Fourier basis given below:

$$V(\theta,\lambda)=\text{span}\left\{1,\left(\cos\left(\frac{k\theta}{2L_\theta}\right)\right)_{1\le k\le 2\lambda L_\theta},\left(\sin\left(\frac{k\theta}{2L_\theta}\right)\right)_{1\le k\le 2\lambda L_\theta}\right\},\qquad(6)$$

where $L_\theta=\lfloor\frac{\theta(1)-\theta(0)}{2\pi}\rfloor$, $\lfloor x\rfloor$ is the largest integer less than x, and $\lambda\le 1/2$ is a parameter that controls the smoothness of $V(\theta,\lambda)$. In our computations, we typically choose $\lambda=1/2$.

In the rest of the paper, we will introduce several algorithms to approximately solve the above optimization problem (4). First, we give a generic algorithm based on matching pursuit and nonlinear least squares in Sec. 2. This algorithm can be accelerated by the Fast Fourier transform (FFT) if the signal is periodic. This will be reported in Sec. 3. In Sec. 4, we develop an efficient algorithm for general nonperiodic signals based on the group sparsity and the algorithm introduced in Sec. 2. In Sec. 5, we generalize this algorithm to deal with signals that have outliers or missing samples. Finally, we present several numerical results including both synthetic and real signal to demonstrate the performance of our method in Sec. 6. Some concluding remarks are made in Sec. 7.

2. Method Based on Matching Pursuit

In this section, we introduce a method based on matching pursuit to get the sparsest decomposition. This is inspired by a similar algorithm that was proposed in Hou and Shi [2013]. This decomposition method is described in Algorithm 1 given below.

Algorithm 1. A decomposition method based on matching pursuit

Input: Signals $f^j(t)$, $j=1,\dots,M$.

Output: Phase functions and the corresponding envelopes: θ_k, a_k^j, $k=1,\dots,K, j=1,\dots,M$.

1: Set $r_1^j=f^j(t), j=1,\dots,M$ and $k=1$.

2: **while** $\max_j(\|r_k^j\|_{l^2})>\epsilon_0$ **do**

3: Solve the following nonlinear least-square problem:

$$\min_{a_k^j,\theta_k}\quad\sum_{j=1}^M\|r_k^j-a_k^j\cos\theta_k\|_{l^2}^2\qquad(7)$$

$$\text{Subject to:}\quad\theta_k'\ge 0,\quad a_k^j(t)\in V(\theta_k),\ \ j=1,\dots,M.$$

4: Update the residual

$$r_{k+1}^j=r_k^j-a_k^j(t)\cos\theta_k,\ \ j=1,\dots,M.$$

5: Set $k=k+1$.

6: **end while**

Algorithm 2. A Gauss–Newton iteration to solve the nonlinear least-squares problem

Input: Initial guess of the phase function $\theta_k^0 = \theta_0$.

Output: Phase functions and the corresponding envelopes: $\theta_k, a_k^j, \quad j = 1, \ldots, M$.

1: **while** $\|\theta_k^{n+1} - \theta_k^n\|_{l^2} > \epsilon_0$ **do**

2: Solve the following least-square problem for each signal r_{k-1}^j, $j = 1, \ldots, M$:

$$\min_{a_k^{j,n+1}, b_k^{j,n+1}} \|r_{k-1}^j - a_k^{j,n+1}(t)\cos\theta_k^n(t) - b_k^{j,n+1}(t)\sin\theta_k^n(t)\|_{l^2}^2 \tag{8}$$

$$\text{Subject to} \quad a_k^{j,n+1}(t),\ b_k^{j,n+1}(t) \in V(\theta_k^n).$$

3: Calculate the averaged update of the instantaneous frequencies:

$$\Delta\omega_j = \frac{a_k^{j,n+1}\frac{\mathrm{d}}{\mathrm{d}t}(b_k^{j,n+1}) - b_k^{j,n+1}\frac{\mathrm{d}}{\mathrm{d}t}(a_k^{j,n+1})}{(a_k^{j,n+1})^2 + (b_k^{j,n+1})^2}, \quad \Delta\omega = \frac{\sum_{j=1}^M \Delta\omega_j \Gamma_k^{j,n+1}}{\sum_{j=1}^M \Gamma_k^{j,n+1}}, \tag{9}$$

where $\Gamma_k^{j,n+1} = (a_k^{j,n+1})^2 + (b_k^{j,n+1})^2$.

4: Update θ_k^n

$$\Delta\theta = \int_0^t \Delta\omega(s)ds, \quad \theta_k^{n+1} = \theta_k^n - \beta\Delta\theta,$$

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

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

5: **end while**

To solve the nonlinear least-squares problem (7) in Algorithm 1, we use the well-known Gauss–Newton type iteration. This algorithm is very similar to the one used in Hou and Shi [2013]. The only difference is the update of the phase function. In this new model, different signals share the same phase function. As a result, we only update one common phase function by taking some average among different signals.

By integrating Algorithm 1 with 2, we can obtain a complete algorithm to compute the sparsest decomposition. The key step and also the most expensive step is to solve the least-squares problem (8). It does not need too much time to solve (8) for one time, however, we have to solve this least-squares problem many times to get the final decomposition. This makes the above algorithm not very practical in real world applications. In the next two sections, we will propose some acceleration algorithms to speed up the above algorithm.

3. Periodic Signals

We first consider a special case when the signals are periodic in time. If the signals are periodic, we can use a standard Fourier basis to construct the $V(\theta, \lambda)$ space

instead of the over-complete Fourier basis given in (6).

$$V_p(\theta,\lambda)=\text{span}\left\{1,\left(\cos\left(\frac{k\theta}{L_\theta}\right)\right)_{1\le k\le \lambda L_\theta},\left(\sin\left(\frac{k\theta}{L_\theta}\right)\right)_{1\le k\le \lambda L_\theta}\right\},\tag{10}$$

where $\lambda \le 1/2$ is a parameter to control the smoothness of functions in $V_p(\theta,\lambda)$ and $L_\theta=(\theta(1)-\theta(0))/2\pi$ is a positive integer.

In Hou and Shi [2013], we already developed an efficient algorithm based on the FFT to solve the least-squares problem (8) for periodic signals. For the sake of completeness, we include this algorithm below.

Suppose that the signal r_{k-1} is measured over a uniform grid $t_j=j/N,\ j=0,\ldots,N-1$. Here, we assume that the sample points are fine enough such that r_{k-1} can be interpolated to any grid with a small error. Let $\overline{\theta}=\frac{\theta-\theta(0)}{\theta(1)-\theta(0)}$ be the normalized phase function and $L_\theta=\frac{\theta(1)-\theta(0)}{2\pi}$, which is an integer.

The FFT-based algorithm to approximately solve (8) 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 $\widehat{r}_{\theta_k^n}$:

$$r_{\theta_k^n,\,j}=\text{Interpolate}(r_{k-1},\theta_{k,j}^n),\tag{11}$$

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

$$\widehat{r}_{\theta_k^n}(\omega)=\frac{1}{N}\sum_{j=1}^{N} r_{\theta,j}e^{-i2\pi\omega\overline{\theta}_{k,\,j}^n},\quad \omega=-N/2+1,\ldots,N/2,\tag{12}$$

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

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 $b_{\theta_k^n}$:

$$a_{\theta_k^n}=\mathcal{F}^{-1}[(\widehat{r}_{\theta_k^n}(\omega+L_{\theta_k^n})+\widehat{r}_{\theta_k^n}(\omega-L_{\theta_k^n}))\cdot\chi_\lambda(\omega/L_{\theta_k^n})],\tag{13}$$

$$b_{\theta_k^n}=\mathcal{F}^{-1}[i\cdot(\widehat{r}_{\theta_k^n}(\omega+L_{\theta_k^n})-\widehat{r}_{\theta_k^n}(\omega-L_{\theta_k^n}))\cdot\chi_\lambda(\omega/L_{\theta_k^n})],\tag{14}$$

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

$$\mathcal{F}^{-1}(\widehat{r}_{\theta_k^n})=\frac{1}{N}\sum_{\omega=-N/2+1}^{N/2}\widehat{r}_{\theta_k^n}e^{i2\pi\omega\overline{\theta}_{k,j}^n},\quad j=0,\ldots,N-1.\tag{15}$$

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

$$a(t_i)=\text{Interpolate }(a_\theta,t_i),\quad i=0,\ldots,N-1,\tag{16}$$

$$b(t_i)=\text{Interpolate }(b_\theta,t_i),\quad i=0,\ldots,N-1.\tag{17}$$

The low-pass filter $\chi_\lambda(\omega)$ in the second step is determined by the choice of $V_p(\theta,\lambda)$. In this paper, we choose the following low-pass filter $\chi_\lambda(\omega)$ to define $V_p(\theta,\lambda)$:

$$\chi_\lambda(\omega)=\begin{cases}1+\cos(\pi\omega/\lambda), & -\lambda<\omega<\lambda\\ 0, & \text{otherwise.}\end{cases} \tag{18}$$

By incorporating the FFT-based solver in Algorithm 2, we get an FFT-based iterative algorithm, which is summarized in Algorithm 3.

Remark 1. We remark that the projection operator $P_{V_p(\theta;\eta)}$ is in fact a low-pass filter in the θ-space. For nonperiodic data, we apply a mirror extension to $\Delta\theta$ before we apply the low-pass filter.

Algorithm 3. (FFT-based algorithm to solve the nonlinear least-squares problem)

Input: Initial guess of the phase function $\theta_k^0=\theta_0,\quad \eta=0$.

Output: Phase functions and the corresponding envelopes: $\theta_k, a_k^j,\quad j=1,\ldots,M$.

1: **while** $\eta<\lambda$ **do**

2: **while** $\|\theta_k^{n+1}-\theta_k^n\|_{l^2}>\epsilon_0$ **do**

3: 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 $\widehat{r}_{\theta_k^n}$.

4: 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 $b_{\theta_k^n}$.

5: Interpolate a_θ and $b_{\theta_k^n}$ back to the uniform mesh of t.

6: Calculate averaged update of the instantaneous frequencies:

$$\Delta\omega_j=\frac{a_k^{j,n+1}\frac{\mathrm{d}}{\mathrm{d}t}(b_k^{j,n+1})-b_k^{j,n+1}\frac{\mathrm{d}}{\mathrm{d}t}(a_k^{j,n+1})}{(a_k^{j,n+1})^2+(b_k^{j,n+1})^2},\quad \Delta\omega=\frac{\sum_{j=1}^M\Delta\omega_j\Gamma_k^{j,n+1}}{\sum_{j=1}^M\Gamma_k^{j,n+1}}, \tag{19}$$

where $\Gamma_k^{j,n+1}=(a_k^{j,n+1})^2+(b_k^{j,n+1})^2$.

7: Update θ_k^n

$$\Delta\theta'=P_{V(\theta;\eta)}(\Delta\omega),\quad \Delta\theta(t)=\int_0^t\Delta\theta'(s)ds,\quad \theta_k^{n+1}=\theta_k^n-\beta\Delta\theta,$$

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

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

and $P_{V_p(\theta;\eta)}$ is the projection operator to the space $V_p(\theta;\eta)$.

8: **end while**

9: $\eta=\eta+\Delta\eta$

10: **end while**

The Gauss–Newton type iteration is sensitive to the initial guess. In order to abate the dependence on the initial guess, we gradually increase the value of η to improve the approximation to the phase function so that it converges to the correct value. The detailed explanation can be found in Hou and Shi [2013].

4. Nonperiodic Signals

As we know, the signals that we deal with in practice are in general not periodic. To apply the FFT-based algorithm described in the previous section, we have to perform a periodic extension for general nonperiodic signals. In this paper, the signals are assumed to satisfy the MAHM model (3). One consequence of the MAHM model is that the phase function can be used as a coordinate and the signals are sparse over the Fourier basis in the θ-coordinate. Then, one natural way to perform the periodic extension is to look for the sparsest representation of a signal over an over-complete Fourier basis defined in the θ-coordinate.

More precisely, the extension is obtained by solving an l_1 minimization problem.

$$\min_{\mathbf{x}\in\mathbb{R}^{N_b}} \|\mathbf{x}\|_1, \quad \text{subject to} \quad \mathbf{\Phi}_\theta \cdot \mathbf{x} = \mathbf{f}, \tag{20}$$

where $\mathbf{f}$ is the sample of the signal at $t_j, j = 1, \ldots, N_s$, and N_s is the number of sample points. The grid points, t_j, may not be uniformly distributed, however, we assume that the sample points are fine enough such that $\mathbf{f}$ can be interpolated over any other grids without loss of accuracy. Here $\mathbf{\Phi}_\theta \in \mathbb{C}^{N_s\times N_b}$ is a matrix consisting of basis functions, N_b is the number of Fourier modes, $\mathbf{\Phi}_\theta(j,k) = e^{ik\theta(t_j)/\bar{L}_\theta}, j = 1, \ldots, N_s, k = -N_b/2+1, \ldots, N_b/2$, and $\bar{L}_\theta$ is a positive integer determined by the length of the period of the extended signal, which will be discussed later.

To get the periodic extension of M signals, one way is to solve (20) M times independently. However, these M signals are not independent. They share the same phase functions. Inspired by the methods that perform simultaneous sparsity approximations [Tropp (2006); Tropp *et al.* (2006)], we propose to solve following optimization problem to get the periodic extension of M signals simultaneously.

$$\min_{\mathbf{X}\in\mathbb{R}^{N_b\times M}} \|\mathbf{X}\|_{2,1}, \quad \text{subject to} \quad \mathbf{\Phi}_\theta \cdot \mathbf{X} = \mathbf{F}, \tag{21}$$

where $\mathbf{X}$ is a $N_b \times M$ matrix, $\mathbf{X} = (x_{j,k})_{j=1,\ldots,N_b,k=1,\ldots M}$, $\mathbf{F} = [\mathbf{f}^1, \ldots, \mathbf{f}^M]$ is a $N_s \times M$ matrix and

$$\|\mathbf{X}\|_{2,1} = \sum_{j=1}^{N_b} \left(\sum_{k=1}^{M} |x_{j,k}|^2 \right)^{1/2}. \tag{22}$$

The remaining task is to solve the optimization problem (21) efficiently. Since the sample points are fine enough, we can assume that the sample points are uniformly distributed in the θ-coordinate, i.e. $\theta(t_j) = (j-1)\Delta\theta$ and $\Delta\theta = 2\pi\bar{L}_\theta/N_b$. Otherwise, we could use some interpolation method (for instance the cubic spline interpolation) to get the signals over the uniformly distributed sample points. Now,

we define a matrix, $\bar{\mathbf{\Phi}}_\theta \in \mathbb{C}^{N_b \times N_b}$, consisting of the complete Fourier basis,

$$\bar{\mathbf{\Phi}}_\theta(j,k) = \frac{1}{\sqrt{N_b}} e^{ik(j-1)\Delta\theta/\bar{L}_\theta}, \quad j = 1, \ldots, N_b, \;\; k = 1, \ldots, N_b, \tag{23}$$

where $\bar{\mathbf{\Phi}}_\theta(j,k)$ represents the element of $\bar{\mathbf{\Phi}}_\theta$ at jth row and kth column. Using the property of the Fourier basis, we know that $\bar{\mathbf{\Phi}}_\theta$ is an orthonormal matrix, which is a very desirable property for us to derive a fast algorithm.

We denote Ω as the index set of $\mathbf{F}$,

$$\Omega = \{(j,k) : j = 1, \ldots, N_b, (j-1)\Delta\theta \le \theta(1) - \theta(0), k = 1, \ldots, N_b\}. \tag{24}$$

Then, for any $(j,k) \in \Omega$, $\mathbf{F}(j,k)$ is a sample point. We also define an extension of $\mathbf{F}$ by zero padding,

$$\bar{\mathbf{F}}(j,k) = \begin{cases} \mathbf{F}(j,k), & (j,k) \in \Omega, \\ 0, & \text{otherwise.} \end{cases} \tag{25}$$

where $j, k = 1, \ldots, N_b$.

First, we remove the constraint in (21) by introducing a penalty term,

$$\min_{\mathbf{x} \in \mathbb{R}^{N_b}} \|\mathbf{X}\|_{2,1} + \frac{\mu}{2} \|\mathbf{\Phi}_\theta \cdot \mathbf{X} - \mathbf{F}\|_{2,2}^2. \tag{26}$$

Here, $\mu > 0$ is a parameter of penalty. We will let μ go to infinity later. Then the solution of (26) will converge to the solution of (21).

Let $\bar{\mathbf{Y}} = \bar{\mathbf{\Phi}}_\theta \cdot \mathbf{X} - \bar{\mathbf{F}}$, and $\mathbf{Y} = \bar{\mathbf{Y}}|_\Omega = \mathbf{\Phi}_\theta \cdot \mathbf{X} - \mathbf{F}$. Using the Augmented Lagrange Multiplier method (ALM) to solve the unconstrained optimization problem (26), we obtain the following algorithm. Let $\mathbf{Q}^0 = \mathbf{0}$, and repeat the following two steps until the algorithm converges.

- $(\mathbf{X}^{k+1}, \bar{\mathbf{Y}}^{k+1}) = \arg\min_{\mathbf{X}, \bar{\mathbf{Y}} \in \mathbb{R}^{N_b \times M}} \|\mathbf{X}\|_{2,1} + \frac{\mu}{2}\|\mathbf{Y}\|_{2,2}^2 + \frac{\gamma}{2}\|\bar{\mathbf{Y}} - \bar{\mathbf{\Phi}}_\theta \cdot \mathbf{X} + \bar{\mathbf{F}} + \mathbf{Q}^k/\gamma\|_{2,2}^2$.
- $\mathbf{Q}^{k+1} = \mathbf{Q}^k + \gamma \cdot (\bar{\mathbf{Y}}^{k+1} - \bar{\mathbf{\Phi}}_\theta \cdot \mathbf{X}^{k+1} + \bar{\mathbf{F}})$.

Here, $\gamma > 0$ is a parameter. In the above ALM iteration, the main computational load is to solve the optimization problem in the first step. Note that the objective functional depends on two terms, $\mathbf{X}$ and $\bar{\mathbf{Y}}$. It is natural to minimize the functional alternately by optimizing one of these two terms when the other one is fixed, which is similar in spirit to the split Bregman iteration [Goldstein and Osher (2009)]. Using this idea, we get the following iterative algorithm to solve (26).

Let $\mathbf{Q}^0 = 0$, $\bar{\mathbf{Y}}^0 = 0$ and repeat

- $\mathbf{X}^{k+1} = \arg\min_{\mathbf{X} \in \mathbb{R}^{N_b \times M}} \|\mathbf{X}\|_{2,1} + \frac{\gamma}{2}\|\bar{\mathbf{Y}}^k - \bar{\mathbf{\Phi}}_\theta \cdot \mathbf{X} + \bar{\mathbf{F}} + \mathbf{Q}^k/\gamma\|_{2,2}^2$,
- $\bar{\mathbf{Y}}^{k+1} = \arg\min_{\bar{\mathbf{Y}} \in \mathbb{R}^{N_b \times M}} \frac{\mu}{2}\|\mathbf{Y}\|_{2,2}^2 + \frac{\gamma}{2}\|\bar{\mathbf{Y}} - \bar{\mathbf{\Phi}}_\theta \cdot \mathbf{X}^{k+1} + \bar{\mathbf{F}} + \mathbf{Q}^k/\gamma\|_{2,2}^2$,
- $\mathbf{Q}^{k+1} = \mathbf{Q}^k + \gamma \cdot (\bar{\mathbf{Y}}^{k+1} - \bar{\mathbf{\Phi}}_\theta \cdot \mathbf{X}^{k+1} + \bar{\mathbf{F}})$.

The good news is that the optimization problems in the first and the second step can be solved explicitly and we can use the FFT to accelerate the computation. To see this, we consider the first optimization problem. Using the fact that $\bar{\mathbf{\Phi}}_\theta$ is an orthonormal matrix, we have

$$\begin{aligned}\mathbf{X}^{k+1} &= \arg\min_{\mathbf{X}\in\mathbb{R}^{N_b\times M}} \|\mathbf{X}\|_{2,1} + \frac{\gamma}{2}\|\mathbf{X} - \bar{\mathbf{\Phi}}_\theta^* \cdot (\bar{\mathbf{Y}}^k + \bar{\mathbf{F}} + \mathbf{Q}^k/\gamma)\|_{2,2}^2 \\ &= \mathcal{S}_\gamma(\bar{\mathbf{\Phi}}_\theta^* \cdot (\bar{\mathbf{Y}}^k + \bar{\mathbf{F}} + \mathbf{Q}^k/\gamma)), \end{aligned} \tag{27}$$

where $\bar{\mathbf{\Phi}}_\theta^*$ is the conjugate transpose of $\bar{\mathbf{\Phi}}_\theta$. Here $\mathcal{S}_\gamma$ is a shrinkage operator, which is defined as follows. For any $\mathbf{v}\in\mathbb{C}^M$

$$\mathcal{S}_\gamma(\mathbf{v}) = \begin{cases} \dfrac{\|\mathbf{v}\|_2 - \gamma}{\|\mathbf{v}\|_2}\,\mathbf{v}, & \|\mathbf{v}\|_2 > \gamma, \\ 0, & \|\mathbf{v}\|_2 \le \gamma. \end{cases} \tag{28}$$

If $\mathbf{V}$ is a matrix and $\mathbf{v}_j$ is its jth row, then $S_\gamma(V)$ is a matrix of the same size as $\mathbf{V}$ and the jth row is defined as $S_\gamma(\mathbf{v}_j)$. Notice that $\bar{\mathbf{\Phi}}_\theta$ is the matrix consisting of the Fourier basis in the θ-coordinate, so the matrix–matrix multiplication can be evaluated by applying the discrete Fourier transform of each column of the second matrix, which can be accelerated by FFT.

In the second optimization problem, recall that $\mu > 0$ is a penalty parameter in (26). The larger this parameter is, the better the performance will be. By letting μ go to infinity, we can solve

$$\bar{\mathbf{Y}}^{k+1}(j,k) = \begin{cases} (\bar{\mathbf{\Phi}}_\theta \cdot \mathbf{X}^{k+1} - \bar{\mathbf{F}} - \mathbf{Q}^k/\gamma)(j,k), & (j,k)\in\Omega, \\ 0, & \text{otherwise.} \end{cases} \tag{29}$$

Summarizing the above derivation, we get a fast algorithm to solve (21), see Algorithm 4. By combining Algorithm 4 and the algorithm in previous section, we can get a complete algorithm to deal with nonperiodic data. Before giving the complete algorithm, there is still one issue we need to address. In the derivation

Algorithm 4. (Fourier extension by group sparsity)

Input: $\mathbf{Q}^0 = 0, \bar{\mathbf{Y}}^0 = 0$.

Output: Fourier coefficients on over-complete Fourier basis $\mathbf{X}$.

1: **while** $\|\mathbf{Q}^k - \mathbf{Q}^{k-1}\|_{2,2} > \epsilon_0$ **do**

2: $\mathbf{X}^{k+1} = \mathcal{S}_{\gamma^{-1}}(\mathcal{F}_c(\bar{\mathbf{Y}}^k + \bar{\mathbf{F}} + \mathbf{Q}^k/\gamma))$, where $\mathcal{F}_c$ is an operator that applies the Fourier transform to each column.

3: $\bar{\mathbf{Y}}^{k+1}(j,k) = \begin{cases} (\bar{\mathbf{\Phi}}_\theta \cdot \mathbf{X}^{k+1} - \bar{\mathbf{F}} - \mathbf{Q}^k/\gamma)(j,k), & (j,k)\in\Omega, \\ 0, & \text{otherwise.} \end{cases}$

4: $\mathbf{Q}^{k+1} = \mathbf{Q}^k + \gamma\cdot(\bar{\mathbf{Y}}^{k+1} - \bar{\mathbf{\Phi}}_\theta \cdot \mathbf{X}^{k+1} + \bar{\mathbf{F}})$

5: **end while**

6: $\mathbf{X} = \mathbf{X}^{k+1}$.

of Algorithm 4, we assume that the sample points are uniformly distributed in the θ-coordinate with a given θ. For a general signal that may not be uniformly sampled in the θ-coordinate, we need to perform interpolation before applying Algorithm 4.

In order to perform interpolation, we first need to provide the interpolation points that are uniformly distributed in the θ-coordinate. In this paper, the interpolation points are chosen as $\theta(0) + (j-1)\Delta\theta, j = 1, \ldots, N_b$, $\Delta\theta = 2\pi\bar{L}_\theta/N_b$. We set $\bar{L}_\theta = 2L_\theta$ and $L_\theta = \lfloor \frac{\theta(1)-\theta(0)}{2\pi} \rfloor$. This choice corresponds to the two-fold over-complete Fourier basis used in (6). Here, $N_b = 2N_s$ and N_s is the number of sample points of the original signal. The original signals are interpolated over the interpolation points by using the cubic spline interpolation.

Algorithm 5. (Algorithm for a nonperiodic signal)

Input: Initial guess of the phase function $\theta_k^0 = \theta_0, \quad \eta = 0$.

Output: Phase functions and the corresponding envelopes: $\theta_k, a_k^j, \quad j = 1, \ldots, M$.

1: **while** $\eta < \lambda$ **do**

2: **while** $\|\theta_k^{n+1} - \theta_k^n\|_{l^2} > \epsilon_0$ **do**

3: Interpolate r_{k-1} to a uniform mesh in the θ_k^n-coordinate to get $r_{\theta_k^n}$.

4: Using Algorithm 4 to get the Fourier coefficients of $r_{\theta_k^n}$ on the over-complete Fourier basis.

5: 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 $b_{\theta_k^n}$.

6: Interpolate a_θ and $b_{\theta_k^n}$ back to the uniform mesh of t.

7: Calculate the averaged update of frequency:

$$\Delta\omega_j = \frac{a_k^{j,n+1}\frac{\mathrm{d}}{\mathrm{d}t}(b_k^{j,n+1}) - b_k^{j,n+1}\frac{\mathrm{d}}{\mathrm{d}t}(a_k^{j,n+1})}{(a_k^{j,n+1})^2 + (b_k^{j,n+1})^2}, \quad \Delta\omega = \frac{\sum_{j=1}^M \Delta\omega_j \Gamma_k^{j,n+1}}{\sum_{j=1}^M \Gamma_k^{j,n+1}}, \tag{30}$$

where $\Gamma_k^{j,n+1} = (a_k^{j,n+1})^2 + (b_k^{j,n+1})^2$.

8: Update θ_k^n

$$\Delta\theta' = P_{V(\theta;\eta)}(\Delta\omega), \quad \Delta\theta(t) = \int_0^t \Delta\theta'(s)ds, \quad \theta_k^{n+1} = \theta_k^n - \beta\Delta\theta,$$

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

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

and $P_{V_p(\theta;\eta)}$ is the projection operator to the space $V_p(\theta;\eta)$.

9: **end while**

10: $\eta = \eta + \Delta\eta$.

11: **end while**

Algorithm 6. (Algorithm for signals with outliers)

Input: $\mathbf{Q}^0 = 0, \bar{\mathbf{Y}}^0 = 0$.

Output: Fourier coefficients on over-complete Fourier basis $\mathbf{X}$, estimate of outliers $\mathbf{Z}$.

1: **while** $\|\mathbf{Q}^k - \mathbf{Q}^{k-1}\|_{2,2} > \epsilon_0$ **do**

2: $\mathbf{X}^{k+1} = \mathcal{S}_{\gamma^{-1}}(\mathcal{F}_c(\bar{\mathbf{Y}}^k - \bar{\mathbf{Z}}^k + \bar{\mathbf{F}} + \mathbf{Q}^k/\gamma))$, where $\mathcal{F}_c$ is an operator that applies the Fourier transform to each column.

3: $\bar{\mathbf{Y}}^{k+1}(j,k) = \begin{cases} (\bar{\mathbf{\Phi}}_\theta \cdot \mathbf{X}^{k+1} + \bar{\mathbf{Z}}^k - \bar{\mathbf{F}} - \mathbf{Q}^k/\gamma)(j,k), & (j,k) \in \Omega, \\ 0, & \text{otherwise.} \end{cases}$

4: $\bar{\mathbf{Z}}^{k+1} = \mathcal{T}_{\gamma^{-1}}(\bar{\mathbf{Y}}^{k+1} + \bar{\mathbf{F}} + \mathbf{Q}^k/\gamma - \bar{\mathbf{\Phi}}_\theta^* \cdot \mathbf{X})$, where $\mathcal{T}_\gamma$ is a shrinkage operator. For any $x \in \mathbb{R}$,

$$\mathcal{T}_\gamma(x) = \begin{cases} \dfrac{|x| - \gamma}{|x|}\, x, & |x| > \gamma, \\ 0, & |x| \le \gamma. \end{cases}$$

For any matrix $\mathbf{A} = (a_{jk})$, $\mathcal{T}_\gamma(\mathbf{A}) = (\mathcal{T}_\gamma(a_{jk}))$.

5: $\mathbf{Q}^{k+1} = \mathbf{Q}^k + \gamma \cdot (\bar{\mathbf{Z}}^{k+1} - \bar{\mathbf{\Phi}}_\theta \cdot \mathbf{X}^{k+1} + \bar{\mathbf{F}})$

6: **end while**

7: $\mathbf{X} = \mathbf{X}^{k+1}$, $\mathbf{Z} = \bar{\mathbf{Z}}^{k+1}|_\Omega$, Ω is defined in (24).

Now, we have an algorithm for nonperiodic signals, which is summarized in Algorithm 5.

5. Signals with Outliers or Missing Samples

To deal with the signal with outliers, we need to enlarge the dictionary to include all the impulses. In this case, the optimization problem is reformulated as follows:

$$\min_{\substack{\mathbf{X} \in \mathbb{C}^{N_b \times M} \\ \mathbf{Z} \in \mathbb{R}^{N_s \times M}}} \|\mathbf{X}\|_{2,1} + \|\mathbf{Z}\|_{1,1}, \quad \text{subject to:} \quad \mathbf{\Phi}_\theta \cdot \mathbf{X} + \mathbf{Z} = \mathbf{F}, \tag{31}$$

where $\mathbf{\Phi}_\theta$ and $\mathbf{F}$ are the same as those in (21).

Following the derivation similar to that in Sec. 4, we obtain Algorithm 6 to solve the above optimization problem (31). By using the same interpolation procedure as in Sec. 4, we can integrate Algorithm 6 with the algorithm in Sec. 3 to get the method to deal with signals with outliers.

For the signals with missing samples, we first assign the value of the signal as the average of the signal at the locations where the sample is missing and then treat the missing sample as the outliers.

6. Numerical Results

In this section, we use two numerical examples, one is synthetic and one is real data, to demonstrate the performance of the method proposed in this paper. The

first example is a simple synthetic signal, which is used to test the robustness of the method to the perturbation of white noise.

Example 1. A synthetic signal with white noise

In this example, the signals are generated as follows:

$$f(t) = \cos(40\pi(t+1)^2), \quad f^j(t) = f(t) + 5X^j(t), \quad j = 1, \ldots, 10, \ t \in [0,1], \quad (32)$$

where $X^j(t), j = 1, \ldots, 10$ are independent white noise with standard deviation $\sigma^2 = 1$. The number of sample points is 512 and the sample points are uniformly distributed over $[0, 1]$.

The original signals are plotted in the left figure of Fig. 1. As we can see, the noise is so large that we cannot see any pattern of the original clean signal. The recovered instantaneous frequency is shown in the right figure of Fig. 1. If we recover the frequency from each signal separately, the frequencies are totally wrong due to the large amplitude of the noise. However, if we use the special structure that these 10 signals have the same instantaneous frequency, the frequency that we recover is much better.

Example 2. Cable tension estimate[a]

This example is a real world application in which one would like to estimate the tension of the cables in a cable bridge. Before demonstrating the numerical results, we give some backgrounds first.

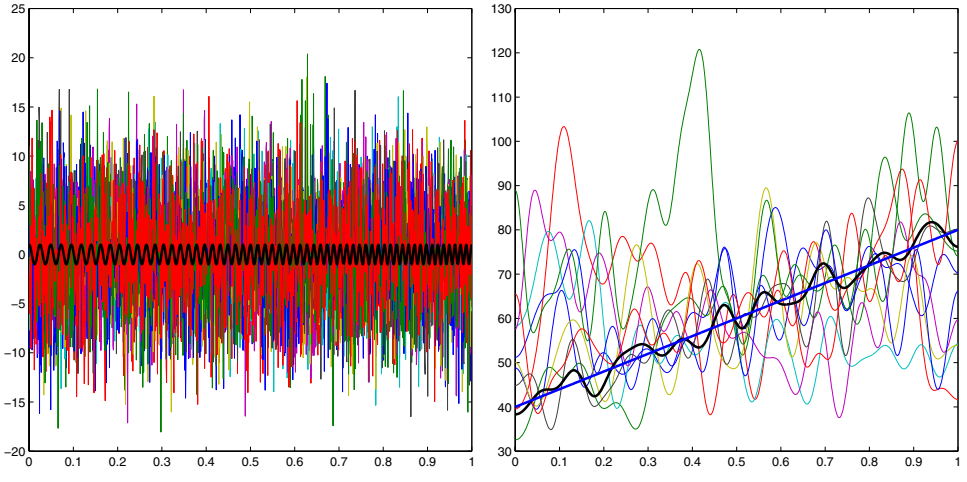

Fig. 1. (Color online) Left: 10 measurements generated by $f(t) + 5X(t)$. Bold black: the signal without noise; thin lines: 10 measurements with larger noise; Right: the instantaneous frequency recovered from 10 different measurements with larger noise. Bold blue: the exact frequency; bold black: the recovered frequency from 10 signals; the thin lines: the frequency obtained from 10 signals separately.

[a]The authors would like to thank Prof. Yuequan Bao of the School of Civil Engineering of Harbin Institute of Technology for the experimental data and for permission to use it in this paper.

For a large span bridge, such as a cable-stayed bridge and suspension bridge, the collection of the cables is a crucial element for the overall structural safety. Due to the moving vehicles and other environmental effects, the cable tension forces vary over time. This variation in cable tension forces may cause fatigue damage. Therefore, the estimation of the time-varying cable tension forces is important for the maintenance and safety assessment of cable-based bridges.

One commonly used method to estimate the cable tension force is based on the instantaneous frequency estimate of the cable vibration signal. According to the flat taut string theory that neglects both sag-extensibility and bending stiffness, the cable tension force, F, can be calculated by

$$F(t) = 4mL^2 \left(\frac{\omega_n(t)}{2\pi n} \right)^2, \tag{33}$$

where $\omega_n(t)$ is the time-varying nth natural frequency in $radius/s$ and m, L are mass density and length of the cable.

Also from the flat taut string theory, an important and useful feature of the vibrations of the cable is that the natural frequencies of the higher modes are integer multiples of the fundamental frequency, that is $\omega_n(t) = n\omega_1(t)$. This feature implies that we can combine the information of different modes together to recover the instantaneous frequency. Then the method developed in this paper for multiple signals applies after some minor modifications. In step 5 of Algorithm 5, we can extract several modes based on the relation $\omega_n(t) = n\omega_1(t)$. More specifically, the multiple modes are extracted as follows

$$a^j_{\theta^n_k} = \mathcal{F}^{-1}[(\widehat{r}_{\theta^n_k}(\omega + L_{\theta^n_k}) + \widehat{r}_{\theta^n_k}(\omega - L_{\theta^n_k})) \cdot \chi_\lambda(j \cdot \omega / L_{\theta^n_k})], \quad j = 1, \ldots, M$$

$$b^j_{\theta^n_k} = \mathcal{F}^{-1}[i \cdot (\widehat{r}_{\theta^n_k}(\omega + L_{\theta^n_k}) - \widehat{r}_{\theta^n_k}(\omega - L_{\theta^n_k})) \cdot \chi_\lambda(j \cdot \omega / L_{\theta^n_k})]. \quad j = 1, \ldots, M,$$

where M is the number of models that we use. The only difference between the above extraction and the original one (13)–(14) is that an integer j is multiplied in the cutoff function χ_λ. The integer j corresponds to a different mode since we have the relation $\omega_n(t) = n\omega_1(t)$.

Accordingly, the calculation of the instantaneous frequency requires a minor modification. In the calculation of $\Delta\omega_j$, we need to divide it by integer j to get the

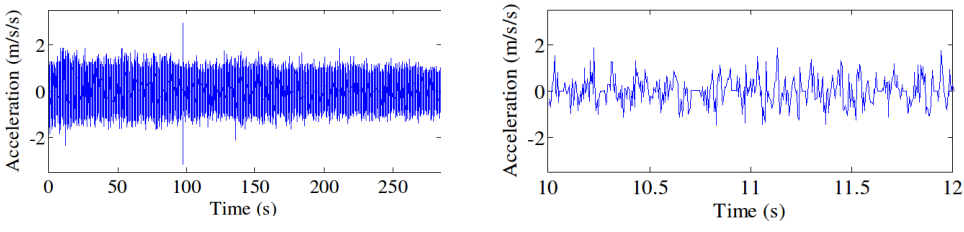

Fig. 2. Left: vibration signal of the cable; Right: zoom in of the signal to demonstrate the missing samples (flat segments).

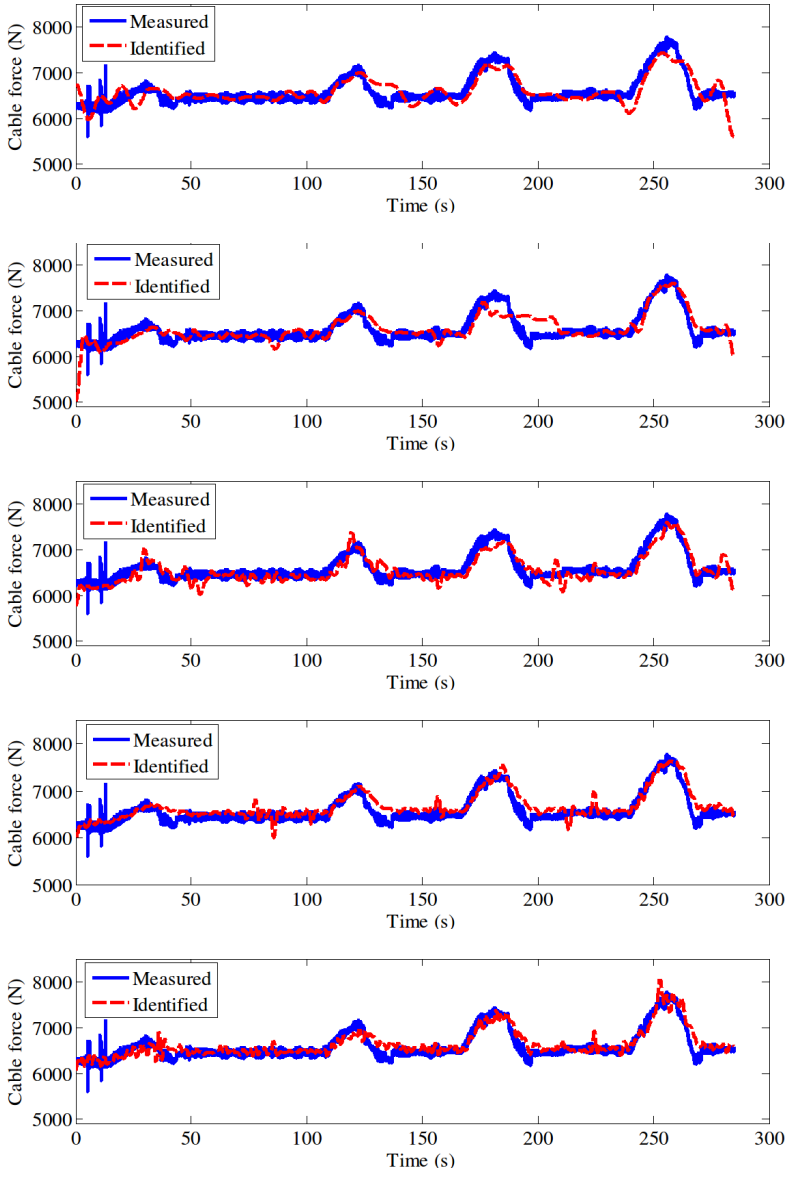

Fig. 3. The cable force estimated by a single natural mode 1–5 from the top to the bottom.

correct fundamental frequency.

$$\Delta\omega_j = \frac{a_k^{j,n+1}\frac{\mathrm{d}}{\mathrm{d}t}(b_k^{j,n+1}) - b_k^{j,n+1}\frac{\mathrm{d}}{\mathrm{d}t}(a_k^{j,n+1})}{j\cdot((a_k^{j,n+1})^2 + (b_k^{j,n+1})^2)}, \quad \Delta\omega = \frac{\sum_{j=1}^{M}\Delta\omega_j\Gamma_k^{j,n+1}}{\sum_{j=1}^{M}\Gamma_k^{j,n+1}}.$$

The original experimental signal is given in Fig. 2. Obviously, the signal has some outliers. In the original signal, about 10% of the samples were lost, which is

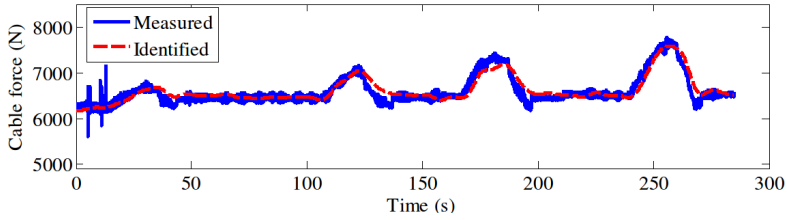

Fig. 4. The cable force estimated by using 1–5 natural modes simultaneously.

demonstrated very clearly in the zoom-in picture in Fig. 2. The tension force estimated by our method is given in Figs. 3 and 4. The tension forces obtained from different modes are shown in Fig. 3. If only one mode is used, the estimation of the force is not very accurate. There are many oscillations although the overall picture matches roughly the measured force. Figure 4 shows the result obtained using 1–5 modes together. As we can see, the estimated force fits the measured force much better.

7. Conclusion

In this paper, we study how to extract instantaneous frequencies from multiple signals that share the same instantaneous frequencies. This kind of signals arises in many scientific and engineering problems. In some applications, although we have only one signal, different components have a fixed relation in frequencies, as in the example of tension force estimate that we considered in this paper. By exploiting the special property that these signals share the same instantaneous frequencies, we formulated this problem as a l^1 optimization problem and developed several algorithms to find the sparsest time-frequency decomposition. The results that we obtained using our algorithms are much better than extracting the instantaneous frequencies one by one. We further proposed FFT-based algorithms to accelerate the convergence of these algorithms. As a result, the FFT-based algorithms are very efficient. By using the group sparsity algorithm, our algorithms can also handle the nonperiodic signals and incomplete signals such as signals with outlier or missing data. In both synthetic and real data tests, we have demonstrated that the algorithms are very efficient and robust.

Acknowledgments

This work was supported in part by NSF Grants DMS-1318377 and DMS-1613861. The research of Dr. Z. Shi was supported by NSFC Grants 11371220 and 11671005.

References

Candes, E., Romberg, J. and Tao, T. (2006). Robust uncertainty principles: Exact signal recovery from highly incomplete frequency information. *IEEE Trans. Inf. Theory*, **52**: 489–509.

Candès, E. and Tao, T. (2006). Near optimal signal recovery from random projections: Universal encoding strategies? *IEEE Trans. Inf. Theory*, **52**: 5406–5425.

Chen, S., Donoho, D. and Saunders, M. (1998). Atomic decomposition by basis pursuit. *SIAM J. Sci. Comput.*, **20**: 33–61.

Chui, C. K. and Mhaskar, H. N. (2016). Signal decomposition and analysis via extraction of frequencies. *Appl. Comput. Harmon. Anal.*, **40**: 97–136.

Daubechies, I. (1992). *Ten Lectures on Wavelets*, CBMS-NSF Regional Conference Series on Applied Mathematics, Vol. 61. SIAM Publications,

Daubechies, I., Lu, J. and Wu, H. (2011). Synchrosqueezed wavelet transforms: An empirical mode decomposition-like tool. *Appl. Comput. Harmon. Anal.*, **30**: 243–261.

Donoho, D. L. (2006). Compressed sensing. *IEEE Trans. Inf. Theory*, **52**: 1289–1306.

Dragomiretskiy, K. and Zosso, D. (2014). Varitational Mode Decomposition. *IEEE Trans. Signal Process*, **62**: 531–544.

Flandrin, P. (1999). *Time-Frequency/Time-Scale Analysis*. Academic Press, San Diego, CA.

Gilles, J. (2013). Empirical Wavelet Transform, *IEEE Trans. Signal Process*, **61**: 3999–4010.

Goldstein, T. and Osher, S. (2009). The Split Bregman method for L_1-regularized problems. *SIAM J. Imaging Sci.*, **2**: 323–343.

Hou, T. Y. and Shi, Z. (2013). Data-driven time-frequency analysis. *Appl. Comput. Harmon. Anal.*, **35**: 284–308.

Hou, T. Y. and Shi, Z. (2016). Sparse time-frequency decomposition by dictionary adaptation, *Philoso. Trans. A* **374**: 20150192.

Hou, T. Y., Shi, Z. and Tavallali, P. (2014). Convergence of a data-driven time-frequency analysis method. *Appl. Comput. Harmon. Anal.*, **37**: 235–270.

Huang, N. E. *et al.* (1998). The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proc. R. Soc. Lond. A*, **454**: 903–995.

Mallat, S. (2009). *A Wavelet Tour of Signal Processing: The Sparse Way*. Academic Press, US.

Mallat, S. and Zhang, Z. (1993). Matching pursuit with time-frequency dictionaries. *IEEE Trans. Signal Process.*, **41**: 3397–3415.

Tropp, J. A. (2006). Algorithms for simultaneous sparse approximation. Part II: Convex relaxation. *Signal Process.*, **86**: 589–602.

Tropp, J. A., Gilbert, A. C. and Strauss, M. J. (2006). Algorithms for simultaneous sparse approximation. Part I: Greedy pursuit. *Signal Process.*, **86**: 572–588.

Wu, Z. and Huang, N. E. (2009). Ensemble empirical mode decomposition: A noise-assisted data analysis method. *Adv. Adapt. Data Anal.*, **1**: 1–41.