

Sparse Time-Frequency decomposition for multiple signals with same frequencies

Thomas Y. Hou,^{*} Zuoqiang Shi[†]

July 8, 2015

Abstract

In this paper, we consider multiple signals sharing 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 simultaneously sparsity approximation and fast Fourier transform, some efficient algorithms is developed. 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 test this method. The performances of this method are very promising.

1 Introduction

Data is an important bridge connecting the human being and the natural world. Many important information of the real world is encoded in the data. In many applications, frequencies of the signal are usually very useful to reveal the underlying physical mechanism. Therefore, in the past several decades, many researchers devoted to find efficient time frequency analysis methods and many powerful methods have been developed, including the windowed Fourier transform [17], the wavelet transform [5, 17], the Wigner-Ville distribution [9], etc. Recently, an adaptive time frequency analysis method, the Empirical Mode Decomposition (EMD) method [15, 20] was developed which provides an efficient adaptive method to extract frequency information from complicate multiscale data. Inspired by EMD, recently several new time frequency analysis methods have been proposed see e.g. the synchrosqueezed wavelet transform [6], the Empirical wavelet transform [8], the variational mode decomposition [10].

In the last few years, inspired by the EMD method and compressive sensing [3, 2, 7], Hou and Shi proposed a data-driven time-frequency analysis method based on the sparsest

^{*}Applied and Comput. Math, MC 9-94, Caltech, Pasadena, CA 91125. *Email:* hou@cms.caltech.edu.

[†]Mathematical Sciences Center, Tsinghua University, Beijing, China, 100084. *Email:* zqshi@math.tsinghua.edu.cn.

time-frequency representation [12]. In this method, the signal is assumed to fit following model:

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

where $a_k(t)$, $\theta_k(t)$ are smooth functions, $\theta'_k(t) > 0$, $k = 1, \dots, M$. We assume that $a_k(t)$ and θ'_k are "less oscillatory" than $\cos \theta_k(t)$. After some scaling, we can always make the signal lies in the time interval $[0, 1]$. So, here we assume the time span of the interval is $[0, 1]$. We borrow the terminology in EMD method and also call $a_k(t) \cos \theta_k(t)$ as the Intrinsic Mode Functions (IMFs) [15]. This model is also known as Adaptive Harmonic model (AHM) and is widely used in the time-frequency analysis literatures [1, 6].

The AHM model seems to be natural and simple. In this model, the main difficulty is how to compute the decomposition. In the AHM model, the number of degrees of freedom is much larger than the given signal. This makes the possible decomposition is not unique. One essential problem is how to set up a criterion to pick up the "best" decomposition and this criterion should be practical in computation. Inspired by the compressive sensing, we proposed to decompose the signal by looking for the sparsest decomposition. And the sparsest decomposition is obtained by solving a nonlinear optimization problem formulated as following:

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

where \mathcal{D} is the dictionary consist of all IMFs which makes the decomposition satisfies the AHM model (see [12] for precise definition of the dictionary).

Two kinds of algorithms were proposed to solve the optimization problem (2). The first one is based on matching pursuit [16, 12] and the other one is based basis pursuit [4, 13]. The convergence of the algorithm based on matching pursuit was also analyzed under the assumption of certain scale separation property [14].

In the previous works, we consider the case that only one signal is given. However, in many application, we can obtain many different signals and these signals have same frequency structure. For instance, in the monitoring of buildings, usually many sensor are put 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 same instantaneous frequencies which associate with the natural frequencies of the building. If we analyze the signals from different sensors individually, the structure that these signals share the same instantaneous frequencies is wasted. Intuitively, by taking advantage of this structure, we may get more robust and efficient methods. In this paper, we will propose several methods by exploiting this special frequency structure.

In this paper, we consider multiple signals sharing same instantaneous frequencies. Naturally, we modify the adaptive harmonic model (1) to deal with this kind of signals.

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

where M is the number of signals, f^j is the j th signal. For each $j = 1, \dots, M$ fixed, it is the AHM model (1), i.e. $a_k^j(t)$, $\theta_k(t)$ are smooth functions, $\theta_k'(t) > 0$, $k = 1, \dots, K$. 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 same phase functions θ_k . We call (3) Multiple Adaptive Harmonic model (MAHM).

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

$$\begin{aligned} & \min_{(a_k^j, \theta_k)_{k=1, \dots, K}^{j=1, \dots, M}} && K && (4) \\ \text{Subject to: } & f^j(t) = \sum_{k=1}^K a_k^j(t) \cos \theta_k(t), && a_k^j(t) \cos \theta_k(t) \in \mathcal{D}, && k = 1, \dots, K, j = 1, \dots, M. \end{aligned}$$

The dictionary \mathcal{D} we used in this paper is same as that in [12]. It can be written as

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

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

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

where $L_\theta = \lfloor \frac{\theta(1) - \theta(0)}{2\pi} \rfloor$, $\lfloor \cdot \rfloor$ is the largest integer less than (\cdot) , and $\lambda \leq 1/2$ is a parameter to control 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 above optimization problem (4). First, we give a generic algorithm based on matching pursuit and nonlinear least squares in Section 2. This algorithm can be accelerated by fast Fourier transform (FFT) if the signal is periodic. This will be reported in Section 3. For general nonperiodic signals, in Section 4, we also develop an efficient algorithm based on group sparsity and the algorithm in Section 2. This algorithm can be also generalized to deal with the signals with outliers or missing samples (Section 5). In Section 6, we present several numerical results including both synthetic and real signal to demonstrate the performance of our method. At the end, some conclusion remarks are made in Section 7.

2 Method based on matching pursuit

Following the same idea as that in [12], we proposed a method based on matching pursuit to get the sparsest decomposition, see Algorithm 1.

Algorithm 1 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:

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

Subject to: $\theta_k' \geq 0$, $a_k^j(t) \in V(\theta_k)$, $j = 1, \dots, M$.

4: Update the residual

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

5: Set $k = k + 1$.

6: **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 also very similar as that in [12]. The only difference is the update of the phase function. In our model, different signals share the same phase function. So in the algorithm, we only update one common phase function by taking some average among different signals.

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

Algorithm 2 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: $\theta_k, a_k^j, b_k^j, j = 1, \dots, M$.

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

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

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

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

3: Calculate averaged updation of frequency:

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

where $\Gamma_k^{j,n+1} = (a_k^{j,n+1})^2 + (b_k^{j,n+1})^2$ and $(\cdot)'$ denote the derivative of (\cdot) with respect to t .

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:

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

5: **end while**

3 Periodic signals

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).

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

where $\lambda \leq 1/2$ 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 this case, we developed an efficient algorithm based fast Fourier transform to solve the least-squares problem (8) in [12]. To make this paper self-contain, we also include the algorithm here.

Suppose that the signal r_{k-1} is measured over a uniform grid $t_j = j/N$, $j = 0, \dots, N-1$. Here, we assume that the sample points are fine enough such that r_{k-1} can be interpolated to any grid with small error. Let $\bar{\theta} = \frac{\theta - \theta(0)}{\theta(1) - \theta(0)}$ 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 $\widehat{r}_{\theta_k^n}$:

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

where $\theta_{k, j}^n$, $j = 0, \dots, N-1$ 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

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

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

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 $b_{\theta_k^n}$:

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

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

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

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

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 $\{t_i\}_{i=1}^N$:

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

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

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 $V_p(\theta, \lambda)$:

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

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

Remark 3.1. *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.*

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 [12].

4 nonperiodic signals

In most of the applications, the signals are not periodic. To apply the FFT-based algorithm in previous section, we have to do 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 θ coordinate. Then, one natural way to do periodic extension is to look for the sparsest representation of the signal over the over-complete Fourier basis defined in θ coordinate.

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

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

\mathbf{f} is the sample of the signal at $t_j, j = 1, \dots, N_s$, N_s is the number of sample 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. $\Phi_\theta \in \mathbb{C}^{N_s \times N_b}$ is a matrix consisting of basis functions, N_b is the number of Fourier modes. $\Phi_\theta(j, k) = e^{ik\theta(t_j)/\bar{L}_\theta}$, $j = 1, \dots, N_s$, $k = -N_b/2 + 1, \dots, N_b/2$. \bar{L}_θ is a positive integer determined by the length of the period of the extended signal which will be discussed later.

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

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

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

- 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 updation of frequency:

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

where $\Gamma_k^{j,n+1} = (a_k^{j,n+1})^2 + (b_k^{j,n+1})^2$ and $(\cdot)'$ denote the derivative of (\cdot) with respect to t .

- 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**
-

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. To take advantage of this structure, inspired by the methods for simultaneous sparsity approximation [18, 19], we propose to solve following optimization problem to get the periodic extension of M signals simultaneously.

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

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

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

The problem left is how 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 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 cubic spline interpolation) to get the signals over the uniformly distributed sample points. Now, we define a matrix, $\bar{\Phi}_\theta \in \mathbb{C}^{N_b \times N_b}$, consisting of complete Fourier basis,

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

$\bar{\Phi}_\theta(j, k)$ represents the element of $\bar{\Phi}_\theta$ at j th row and k th column. Using the property of the Fourier basis, we know that $\bar{\Phi}_\theta$ is an orthonormal matrix which will give us lots of convenience in deriving the fast algorithm.

For the convenience of notation, denote Ω be the set of index of \mathbf{F} ,

$$\Omega = \{(j, k) : j = 1, \dots, N_b, (j-1)\Delta\theta \leq \theta(1) - \theta(0), k = 1, \dots, N_b\}. \quad (24)$$

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,

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

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

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

$$\min_{\mathbf{X} \in \mathbb{R}^{N_b}} \|\mathbf{X}\|_{2,1} + \frac{\mu}{2} \|\bar{\Phi}_\theta \cdot \mathbf{X} - \bar{\mathbf{F}}\|_2^2 \quad (26)$$

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

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

- $(\mathbf{X}^{k+1}, \bar{\mathbf{Y}}^{k+1}) = \arg \min_{\mathbf{X}, \bar{\mathbf{Y}} \in \mathbb{R}^{N_b \times M}} \|\mathbf{X}\|_{2,1} + \frac{\mu}{2} \|\mathbf{Y}\|_2^2 + \frac{\gamma}{2} \|\bar{\mathbf{Y}} - \bar{\Phi}_\theta \cdot \mathbf{X} + \bar{\mathbf{F}} + \mathbf{Q}^k / \gamma\|_2^2$
- $\mathbf{Q}^{k+1} = \mathbf{Q}^k + \gamma \cdot (\bar{\mathbf{Y}}^{k+1} - \bar{\Phi}_\theta \cdot \mathbf{X}^{k+1} + \bar{\mathbf{F}})$

Here $\gamma > 0$ is a parameter. In above ALM iteration, the main computational load is to solve the optimization problem in the first step. Notice 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 like that in the split Bregman iteration [11]. Using this idea, we get following iterative algorithm to solve (26).

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

- $\mathbf{X}^{k+1} = \arg \min_{\mathbf{X} \in \mathbb{R}^{N_b \times M}} \|\mathbf{X}\|_{2,1} + \frac{\gamma}{2} \|\bar{\mathbf{Y}}^k - \bar{\Phi}_\theta \cdot \mathbf{X} + \bar{\mathbf{F}} + \mathbf{Q}^k / \gamma\|_2^2$
- $\bar{\mathbf{Y}}^{k+1} = \arg \min_{\bar{\mathbf{Y}} \in \mathbb{R}^{N_b \times M}} \frac{\mu}{2} \|\mathbf{Y}\|_2^2 + \frac{\gamma}{2} \|\bar{\mathbf{Y}} - \bar{\Phi}_\theta \cdot \mathbf{X}^{k+1} + \bar{\mathbf{F}} + \mathbf{Q}^k / \gamma\|_2^2$
- $\mathbf{Q}^{k+1} = \mathbf{Q}^k + \gamma \cdot (\bar{\mathbf{Y}}^{k+1} - \bar{\Phi}_\theta \cdot \mathbf{X}^{k+1} + \bar{\mathbf{F}})$

The good news is that the optimization problems in the first and second step can be solved explicitly and we can use fast Fourier transform to accelerate the computation. First, let us see the first optimization problem. Using the fact that $\bar{\Phi}_\theta$ is an orthonormal matrix, we have

$$\begin{aligned} \mathbf{X}^{k+1} &= \arg \min_{\mathbf{X} \in \mathbb{R}^{N_b \times M}} \|\mathbf{X}\|_{2,1} + \frac{\gamma}{2} \|\mathbf{X} - \bar{\Phi}_\theta^* \cdot (\bar{\mathbf{Y}}^k + \bar{\mathbf{F}} + \mathbf{Q}^k / \gamma)\|_2^2 \\ &= \mathcal{S}_\gamma \left(\bar{\Phi}_\theta^* \cdot (\bar{\mathbf{Y}}^k + \bar{\mathbf{F}} + \mathbf{Q}^k / \gamma) \right) \end{aligned} \quad (27)$$

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

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

If \mathbf{V} is a matrix and \mathbf{v}_j is its j th row, then $\mathcal{S}_\gamma(\mathbf{V})$ is a matrix of the same size as \mathbf{V} and the j th row is defined to be $\mathcal{S}_\gamma(\mathbf{v}_j)$. Notice that $\bar{\Phi}_\theta$ is the matrix consisting of Fourier basis in θ coordinate, so the matrix-matrix multiplication can be evaluated by applying 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) which is the larger the better. Now, let μ goes to infinity, then we can solve

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

Summarizing 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 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** $\|\mathbf{Q}^k - \mathbf{Q}^{k-1}\|_2 > \epsilon_0$ **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 means apply Fourier transform on each column.
 - 3: $\bar{\mathbf{Y}}^{k+1}(j, k) = \begin{cases} (\bar{\Phi}_\theta \cdot \mathbf{X}^{k+1} - \bar{\mathbf{F}} - \mathbf{Q}^k/\gamma)(j, k), & (j, k) \in \Omega, \\ 0, & \text{otherwise.} \end{cases}$
 - 4: $\mathbf{Q}^{k+1} = \mathbf{Q}^k + \gamma \cdot (\bar{\mathbf{Y}}^{k+1} - \bar{\Phi}_\theta \cdot \mathbf{X}^{k+1} + \bar{\mathbf{F}})$
 - 5: **end while**
 - 6: $\mathbf{X} = \mathbf{X}^{k+1}$.
-

algorithm to deal with nonperiodic data. Before giving the complete algorithm, there is still one issue we need to address. In the derivation of Algorithm 4, we assume that the sample points are uniformly distributed in θ coordinate with given θ . For general signal which may not be uniformly sampled in θ coordinate, we need to do interpolation before applying Algorithm 4.

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

Now, we have an algorithm for nonperiodic signal 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 formulated in the following way,

$$\min_{\substack{\mathbf{X} \in \mathbb{C}^{N_b \times M} \\ \mathbf{Z} \in \mathbb{R}^{N_s \times M}}} \|\mathbf{X}\|_{2,1} + \|\mathbf{Z}\|_1, \quad \text{subject to: } \Phi_\theta \cdot \mathbf{X} + \mathbf{Z} = \mathbf{F}. \quad (31)$$

where Φ_θ and \mathbf{F} are same as those in (21)

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

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

Algorithm 5 (Algorithm for nonperiodic signal)

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

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

- 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 averaged update of frequency:

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

where $\Gamma_k^{j,n+1} = (a_k^{j,n+1})^2 + (b_k^{j,n+1})^2$ and $(\cdot)'$ denote the derivative of (\cdot) with respect to t .

- 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** $\|\mathbf{Q}^k - \mathbf{Q}^{k-1}\|_2 > \epsilon_0$ **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 means apply Fourier transform on each column.
- 3: $\bar{\mathbf{Y}}^{k+1}(j, k) = \begin{cases} (\bar{\Phi}_\theta \cdot \mathbf{X}^{k+1} + \bar{\mathbf{Z}}^k - \bar{\mathbf{F}} - \mathbf{Q}^k/\gamma)(j, k), & (j, k) \in \Omega, \\ 0, & \text{otherwise.} \end{cases}$
- 4: $\bar{\mathbf{Z}}^{k+1} = \mathcal{T}_{\gamma^{-1}}(\bar{\mathbf{Y}}^{k+1} + \bar{\mathbf{F}} + \mathbf{Q}^k/\gamma - \bar{\Phi}_\theta^* \cdot \mathbf{X})$, where \mathcal{T}_γ is a shrinkage operator.
For any $x \in \mathbb{R}$,

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

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

- 5: $\mathbf{Q}^{k+1} = \mathbf{Q}^k + \gamma \cdot (\bar{\mathbf{Z}}^{k+1} - \bar{\Phi}_\theta \cdot \mathbf{X}^{k+1} + \bar{\mathbf{F}})$
 - 6: **end while**
 - 7: $\mathbf{X} = \mathbf{X}^{k+1}$, $\mathbf{Z} = \bar{\mathbf{Z}}^{k+1}|_\Omega$, Ω is defined in (24).
-

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: Synthetic signal with white noise

In this example, the signals are generated as following:

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

where $X^j(t)$, $j = 1, \dots, 10$ 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 that the noise is so large such that we can not 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, since the noise is too large, the frequencies are totally wrong. However, if we use the structure that 10 signals have same instantaneous frequency, the frequency recovered is much better.

Example 2: Cable tension estimate¹

¹The 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.

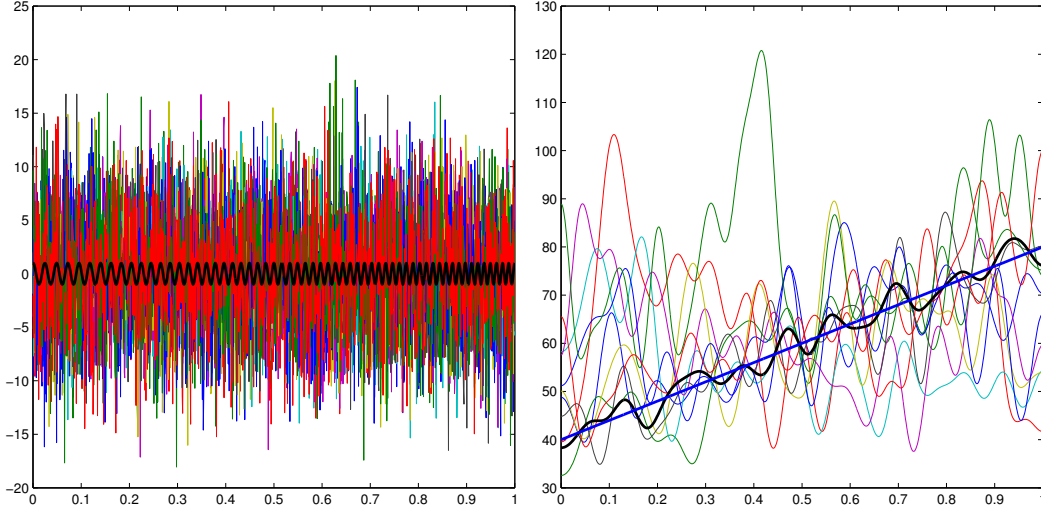


Figure 1: Left: 10 measurements generated by $f(t) + 5X(t)$. Bold black: signal without noise; thin lines: 10 measurements with larger noise; Right: instantaneous frequency recovered from signal with 10 different measurements with larger noise. Bold blue: exact frequency; bold black: recovered frequency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

This example is a real problem about the tension estimate of the cables in bridge. Before demonstrating the numerical results, we give some backgrounds first.

For large span bridges, such as cable-stayed bridge and suspension bridge, the cables are a crucial element for 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, estimation of the time-varying cable tension forces is important for the maintenance and safety assessment of cable-based bridges.

One often 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, cable tension force, F , can be calculated by

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

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

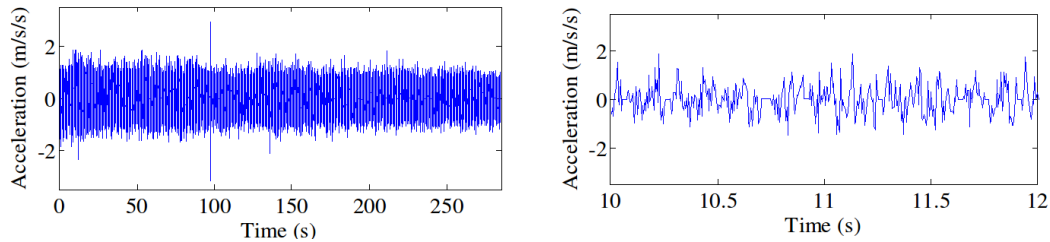


Figure 2: Left: vibration signal of the cable; Right: zoom in of the signal to demonstrate the missing samples (flat segments).

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 demonstrated very clearly in the zoom in picture in Fig. 2. The tension force estimated by our method is given in Fig. 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 match with the measured force. If we use 1-5 modes together to estimate the tension force, the result is much better, Fig. 4.

7 Conclusion

In this paper, we consider multiple signals and these signals share the same instantaneous frequencies. This kind of signals emerge in many scientific and engineering problems. By exploiting the structure of the common frequencies, we develop some algorithms to find the sparsest time-frequency decomposition. These algorithms can be accelerated by FFT, so they are very efficient. The algorithms are also very robust to noise, since the information of multiple signals are used simultaneously.

Acknowledgments. This work was supported by NSF FRG Grant DMS-1159138, DMS-1318377, an AFOSR MURI Grant FA9550-09-1-0613 and a DOE grant DE-FG02-06ER25727. The research of Dr. Z. Shi was supported by a NSFC Grant 11201257.

References

- [1] C. K. Chui and H. N. Mhaskar, Signal decomposition and analysis via extraction of frequencies, *Appl. Comput. Harmon. Anal.*, 2015, <http://dx.doi.org/10.1016/j.acha.2015.01.003>
- [2] E. Candès and T. Tao, Near optimal signal recovery from random projections: Universal encoding strategies?, *IEEE Trans. on Information Theory*, **52(12)**, pp. 5406-5425, 2006.

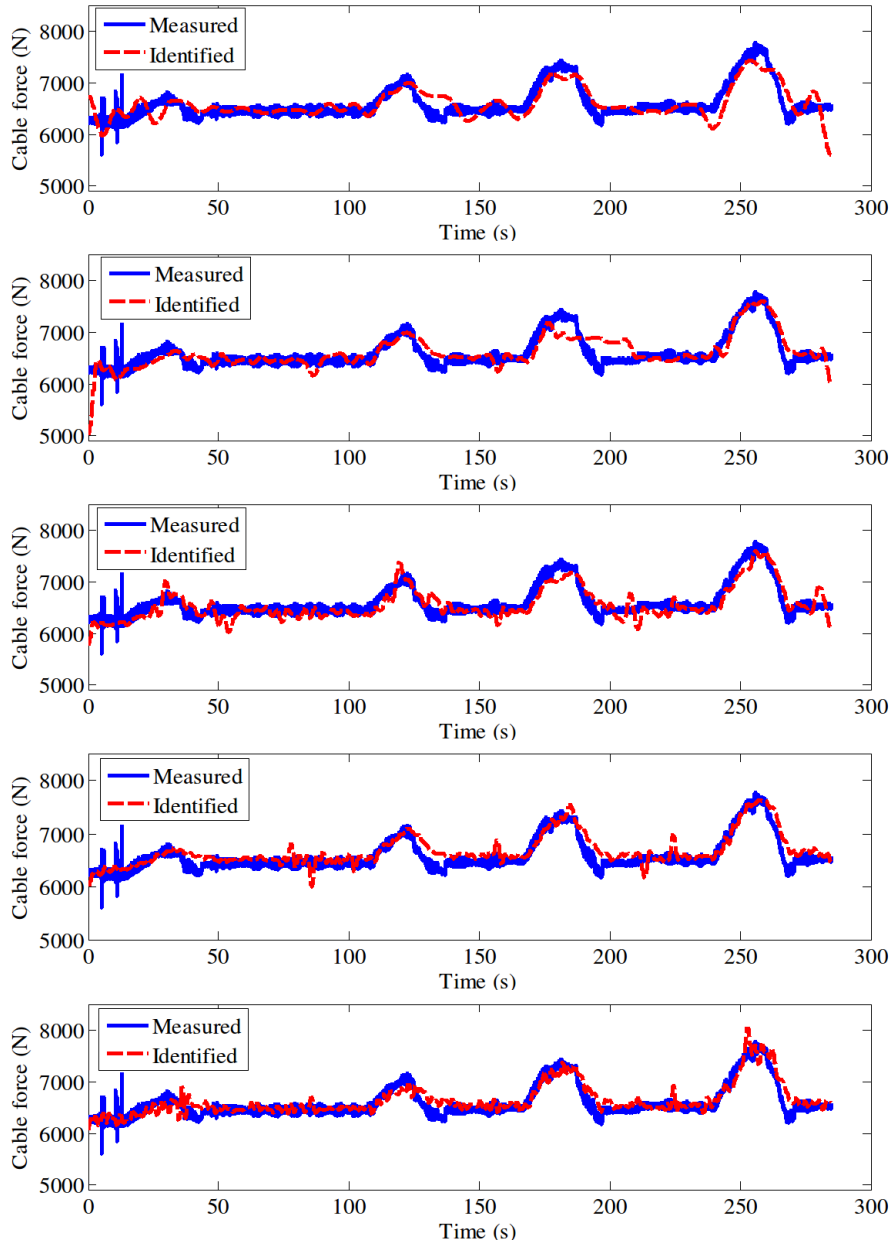


Figure 3: Cable force estimated by single natural mode 1-5 from up to bottom.

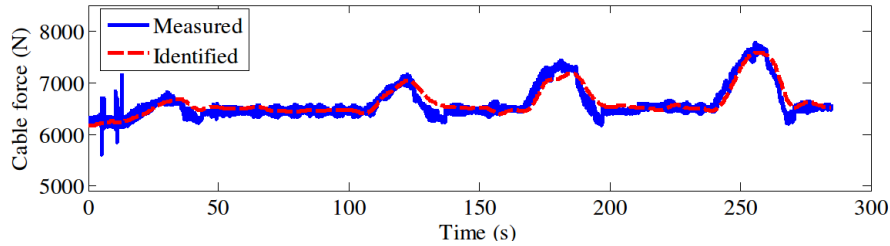


Figure 4: Cable force estimated by using 1-5 natural modes together.

- [3] E. Candes, J. Romberg, and T. Tao, Robust uncertainty principles: Exact signal recovery from highly incomplete frequency information, *IEEE Trans. Inform. Theory*, **52**, pp. 489-509, 2006.
- [4] S. Chen, D. Donoho and M. Saunders, Atomic decomposition by basis pursuit, *SIAM J. Sci. Comput.*, **20**, pp. 33-61, 1998.
- [5] I. Daubechies, Ten Lectures on Wavelets, CBMS-NSF Regional Conference Series on Applied Mathematics, Vol. 61, SIAM Publications, 1992.
- [6] I. Daubechies, J. Lu and H. Wu, Synchrosqueezed wavelet transforms: an empirical mode decomposition-like tool, *Appl. Comp. Harmonic Anal.*, **30** (2011), pp. 243-261.
- [7] D. L. Donoho, Compressed sensing, *IEEE Trans. Inform. Theory*, **52**, pp. 1289-1306, 2006.
- [8] K. Dragomiretskiy and D. Zosso, Variational Mode Decomposition, *IEEE Trans. Signal Process*, **62**, pp. 531-544, 2014.
- [9] P. Flandrin, Time-Frequency/Time-Scale Analysis, Academic Press, San Diego, CA, 1999.
- [10] J. Gilles, Empirical Wavelet Transform, *IEEE Trans. Signal Process*, **61**, pp. 3999-4010, 2013.
- [11] Tom Goldstein and Stanley Osher, The Split Bregman Method for L_1 -Regularized Problems, *SIAM J. Imaging Sci.*, **2**, pp. 323-343, 2009.
- [12] T. Y. Hou and Z. Shi, Data-Driven Time-Frequency Analysis, *Appl. Comput. Harmon. Anal.*, **35**, pp. 284-308, 2013.
- [13] T. Y. Hou and Z. Shi, Sparse Time-Frequency decomposition by dictionary learning, arXiv cs.IT 1311.1163, 2014.
- [14] T. Y. Hou, Z. Shi and P. Tavallali, Convergence of a Data-Driven Time-Frequency Analysis Method, *Appl. Comput. Harmon. Anal.*, **37**, pp. 235-270, 2014.

- [15] 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. A*, **454** (1998), pp. 903-995.
- [16] S. Mallat and Z. Zhang, Matching pursuit with time-frequency dictionaries, *IEEE Trans. Signal Process*, **41**, pp. 3397-3415, 1993.
- [17] S. Mallat, A wavelet tour of signal processing: the Sparse way, Academic Press, 2009.
- [18] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, Algorithms for simultaneous sparse approximation. Part I: Greedy pursuit, *Signal Processing*, special issue "Sparse approximations in signal and image processing", **86**, pp. 572-588, 2006.
- [19] J. A. Tropp, Algorithms for simultaneous sparse approximation. Part II: Convex relaxation, *Signal Processing*, special issue "Sparse approximations in signal and image processing", **86**, pp. 589-602, 2006.
- [20] Z. Wu and N. E. Huang, Ensemble Empirical Mode Decomposition: a noise-assisted data analysis method, *Advances in Adaptive Data Analysis*, **1**, pp. 1-41, 2009.