2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

On-line Modal Parameter Monitoring of Bridges Exploiting Multi-core Capacity by Recursive Stochastic Subspace Identification Method

Ping Lin, Nanxiong Zhang and Bin Ni

***Abstract*— On-line modal parameter monitoring of structures like bridges has compelling practical significance. This paper describes an example of applying recursive stochastic subspace identification methods to the monitoring of bridge modal parameters. The usual recursive stochastic subspace identification algorithm tailored to modal analysis is first presented in details. Then the algorithm is adapted to exploit the power of the multi-core capacity of today's computers so as to improve the speed and effectiveness of the original algorithm. The results of the experiments based on the real data of Donghai Bridge have confirmed the above positive effects of the modified recursive stochastic subspace identification algorithm.**

I. INTRODUCTION

The past years witnessed several severe bridge collapse accidents around the world. For instance, the one happening at Minneapolis in August 2007 was one of such disastrous and heartbroken events. Unfortunate accidents like this should and could be avoided in the future, which implies engineers an urgent task of building better structure health monitoring systems that can better monitor critical structures like bridges. In particular, real-time monitoring of the modal parameters of bridges under working condition is a highly demanded capability for a bridge health monitoring system.

Modal analysis is a standard practice in structure health monitoring system. For structures like bridges, a typical type of methods to do modal analysis is categorized as OMA (operational modal analysis), which means that no artificial stimulus are applied and only the responses signals under ambient stimulus are measured. So this type of method is also called output-only method. By OMA methods, the modal parameters of a structure can be obtained from the measured acceleration signals induced by the external loads acting as unknown stimulus to the structure. Stochastic subspace identification (SSI) is a popular method among all the OMA methods. It is widely adopted to do off-line modal analysis for various structures. In terms of monitoring, however, on-line monitoring, which is at least as fast as the data streaming-in rate, is more desired.

However, direct application of SSI algorithm to real-time monitoring the modal parameters of bridges is facing the difficulty of overwhelming computation intensity. Consequentially, having the ability to eliminate the redundancy in using SSI to do on-line monitoring, recursive type of SSI algorithm (RSSI) has been introduced and successfully applied to several applications. See for instance [1] [2] for more details. Directly applying the RSSI algorithm to bridges, however, still confronts some challenges. First, the computation intensity is still high if all modes of interest are to be tracked. Second, it will be difficult to identify the modes with relatively small "amplitude".

Ping Lin, Nanxiong Zhang and Bin Ni are all with the DSP group of National Instruments Shanghai branch, No.43, 1387 Zhangdong Road, Shanghai, China ping.lin@ni.com; pinglin@gmail.com

In this paper, we are going to introduce a natural solution to meet these challenges. Also, the results obtained by applying our method to real data from Donghai bridge monitoring system will be presented and analyzed. Last, some discussions and suggestions will be made for future advances.

II. FROM SSI TO RSSI

In order to establish the RSSI algorithm, let's first look at the corresponding SSI algorithm (see [3] for more details).

A. SSI

Consider the following linear system:

$$x_{k+1} = Ax_k + w_k \ , \quad (1)$$

$$y_k = Cx_k + v_k \ , \quad (2)$$

where w_k and v_k are zero mean, white vector sequences with covariance matrix:

$$\mathbf{E}[\begin{pmatrix} w_p \\ v_p \end{pmatrix}(w_q^T v_q^T)] = \begin{pmatrix} Q & S \\ S^T & R \end{pmatrix}\delta_{pq} \ . \quad (3)$$

Given enough measurements of the output $y_k \in \mathbb{R}^l$, we would like to determine the system matrices $A \in \mathbb{R}^{n\times n}$, $C \in \mathbb{R}^{l\times n}$ up to within a similarity transformation and the second order statistics of the noises $w_k \in \mathbb{R}^n$ and $v_k \in \mathbb{R}^l$, Q, R and S.

A key notation in SSI algorithm is the following block Hankel matrix.

Choose integers i, j such that $i > n$ and $j >> i$ and define the output block Hankel matrix as:

$$Y_{0|2i-1} \triangleq \left(\begin{array}{ccccc} y_0 & y_1 & y_2 & \cdots & y_{j-1} \\ y_1 & y_2 & y_3 & \cdots & y_j \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ y_{i-1} & y_i & y_{i+1} & \cdots & y_{i+j-2} \\ \hline y_i & y_{i+1} & y_{i+2} & \cdots & y_{i+j-1} \\ y_{i+1} & y_{i+2} & y_{i+3} & \cdots & y_{i+j} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ y_{2i-1} & y_{2i} & y_{2i+1} & \cdots & y_{2i+j-2} \end{array}\right)$$

$$\triangleq \begin{pmatrix} Y_{0|i-1} \\ Y_{i|2i-1} \end{pmatrix} \triangleq \begin{pmatrix} Y_p \\ Y_f \end{pmatrix} \ ,$$

where the superscript "p" is for "past" and "f " for "future".

978-1-4244-2079-7/08/$25.00 ©2008 AACC.

First we do the LQ decomposition of the block Hankel matrix $Y_{0|2i-1}$:

$$\begin{pmatrix} Y_{0|i-1} \\ Y_{i|i} \\ Y_{i+1|2i-1} \end{pmatrix} = \begin{pmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \\ L_{31} & L_{32} & L_{33} \end{pmatrix} \begin{pmatrix} Q_1^T \\ Q_2^T \\ Q_3^T \end{pmatrix} .$$

Then the orthogonal projection of Y_f onto Y_p can be expressed as:

$$\mathcal{O}_i \triangleq Y_f/Y_p = \begin{pmatrix} L_{21} \\ L_{31} \end{pmatrix} Q_1^T$$

Similarly, the orthogonal projection of Y_f^- onto Y_p^+ as:

$$\mathcal{O}_{i-1} \triangleq Y_f^-/Y_p^+ = Y_{i+1|2i-1}/\begin{pmatrix} Y_{0|i-1} \\ Y_{i|i} \end{pmatrix} = (L_{31} L_{32}) \begin{pmatrix} Q_1^T \\ Q_2^T \end{pmatrix} ,$$

with the meanings of superscripts "+" and "-" self-revealing.

It has been proved in [3] that:

$$\mathcal{O}_i = \Gamma_i \hat{X}_i \ ,$$

where Γ_i is the extended observability matrix of original linear system and $\hat{X}_i$ the Kalman filter state sequence. Specifically, $\Gamma_i = (C^T \ (CA)^T \ (CA^2)^T \ \cdots \ (CA^{i-1})^T)^T$ and $\hat{X}_i = (\hat{x}_i \quad \hat{x}_{i+1} \quad \cdots \quad \hat{x}_{i+j-2} \quad \hat{x}_{i+j-1})$

Similarly,

$$\mathcal{O}_{i-1} = \Gamma_{i-1} \hat{X}_{i+1} \ ,$$

where $\Gamma_{i-1} \triangleq \underline{\Gamma_i}$ referring to Γ_i without the last l rows.

The Kalman filter state sequence $\hat{X}_i$ can be obtained by doing an SVD of the orthogonal projection $\mathcal{O}_i$ and (A, C) can then be obtained by a least square procedure using the Kalman filter state sequence. The second order statistics of the noise vectors are obtained from the residue of this least square problem. If needed, the Kalman gain can be further computed from the obtained matrices by solving the associated Raccati equation. Since in our context of modal analysis, only the matrices (A,C) are needed, we will skip the steps after obtaining matrices (A,C) hereafter.

Suppose

$$\begin{pmatrix} L_{21} \\ L_{31} \end{pmatrix} = (U_1 \ U_2) \begin{pmatrix} S_1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix} = U_1 S_1 V_1^T \ ,$$

choose

$$\Gamma_i = U_1 S_1^{1/2} \ ,$$

then

$$\hat{X}_i = \Gamma_i^{\dagger} \mathcal{O}_i \ ;$$

further

$$\hat{X}_{i+1} = \Gamma_{i-1}^{\dagger} \mathcal{O}_{i-1}.$$

(A, C) can be obtained from the following least square problem:

$$\begin{pmatrix} \hat{X}_{i+1} \\ Y_{i|i} \end{pmatrix} = \begin{pmatrix} A \\ C \end{pmatrix} \hat{X}_i + \begin{pmatrix} \rho_w \\ \rho_v \end{pmatrix} .$$

Summarize the above procedure in a compact form as follows:

1) Selections of i and n;

2) Construction of the block Hankel matrix $Y_{0|2i-1}$;

3) LQ decomposition of $Y_{0|2i-1}$:

$$\begin{pmatrix} Y_{0|i-1} \\ Y_{i|i} \\ Y_{i+1|2i-1} \end{pmatrix} = \begin{pmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \\ L_{31} & L_{32} & L_{33} \end{pmatrix} \begin{pmatrix} Q_1^T \\ Q_2^T \\ Q_3^T \end{pmatrix} ;$$

4) SVD of $\begin{pmatrix} L_{21} \\ L_{31} \end{pmatrix}$:

$$\begin{pmatrix} L_{21} \\ L_{31} \end{pmatrix} = (U_1 \ U_2) \begin{pmatrix} S_1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix} \ ;$$

5) Estimation of (A, C):

$$\begin{pmatrix} \hat{A} \\ \hat{C} \end{pmatrix} = \begin{pmatrix} (\underline{U_1} S_1^{1/2})^{\dagger} L_{31} \\ L_{21} \end{pmatrix} V_1 S_1^{-1/2}. \tag{4}$$

One point worth mentioning is that in the above procedure, there is no need to actually compute the orthogonal matrices $Q's$; only the lower triangle matrix L is needed.

B. RSSI

Having the SSI basis, the RSSI algorithm can be constructed by recursifying the core decompositions in the SSI algorithm: LQ and SVD, in sequence.

First, with s data samples available, we employ the SSI algorithm to initially obtain the L factor in the LQ decomposition and the U_1 and S_1 components in the SVD, from which (A, C) can be computed. Then, suppose, a new data sample comes in; we need to produce the updated L and U_1 and S_1, so that a new set of (A, C) can be obtained, which includes the information contained in the new data sample. This means we need to have operations taking into account the new data sample: LQ update and SVD update.

More specifically, removing the first block element of the last column of the output block Hankel matrix $Y_{0|2i-1}$ and appending the new data sample at the end of this column, we form a new column and then place it after $Y_{0|2i-1}$, that is the $[\ Y_{0|2i-1} \mid y^{s+1}\ ]$, where the superscript $s+1$ indicates this column vector contains the samples up to the $(s+1)$th sample. It is easy to show that LQ decomposition of $[\ Y_{0|2i-1} \mid y^{s+1}\ ]$, when only the L factor being sought, can be done by Givens rotations on

$$\left(\begin{array}{cc|c} L_1^s & & y_p^{s+1} \\ L_2^s & L_3^s & y_f^{s+1} \end{array}\right), \tag{5}$$

where

$$Y_{0|2i-1} = \begin{pmatrix} Y_p \\ Y_f \end{pmatrix} = \begin{pmatrix} L_1^s & \\ L_2^s & L_3^s \end{pmatrix} \begin{pmatrix} (Q_1^s)^T \\ (Q_2^s)^T \end{pmatrix} \ ,$$

and

$$y^{s+1} = \begin{pmatrix} y_p^{s+1} \\ y_f^{s+1} \end{pmatrix} \ .$$

The Givens rotations can be viewed as of two steps. First, it produces the following intermediate result:

$$\left(\begin{array}{cc|c} L_1^{s+1} & & 0 \\ L_2^{s+1} & L_3^s & z_f^{s+1} \end{array}\right),$$

where

$$L_2^{s+1}(L_2^{s+1})^T = L_2^s(L_2^s)^T + y_f^{s+1}(y_f^{s+1})^T - z_f^{s+1}(z_f^{s+1})^T \ . \tag{6}$$

Then, it continues to annihilate z_f^{s+1} and thus ends up with an updated version of the lower triangle matrix:

$$\begin{pmatrix} L_1^{s+1} & \\ L_2^{s+1} & L_3^{s+1} \end{pmatrix} .$$

As have been shown in the SSI algorithm, it is the matrix L_2 that needs to be decomposed by the SVD. That is,

$$L_2 = U_1 S_1 V_1^T \ .$$

So (6) is the basis to do this update. Notice, since

$$L_2 L_2^T = U_1 S_1^2 U_1^T \ ,$$

we should strictly speaking be talking about (symmetric) eigenvalue decomposition and its update. So the update is as follows: having

$$L_2^s(L_2^s)^T = U_1^s(S_1^s)^2(U_1^s)^T \ ,$$

seek U_1^{s+1} and S_1^{s+1} such that

$$L_2^{s+1}(L_2^{s+1})^T = U_1^{s+1}(S_1^{s+1})^2(U_1^{s+1})^T \ ,$$

given (6).

This is a standard problem in numerical linear algebra. Equation (6) is a rank-2 update of a symmetric matrix; its eigenvalue and eigenvector matrices can be updated from the old eigen-pair using the newly obtained vectors y_f^{s+1} and z_f^{s+1}. A convenient algorithm called FAST is used in this paper [4].

Having U_1^{s+1} and S_1^{s+1}, the system matrices (A, C) can be obtained through (4). To facilitate the eigenvalue update, however, a little modification is needed to apply to this formula. From

$$L_2 = U_1 S_1 V_1^T \ ,$$

we obtain

$$V_1 = L_2^T U_1 S_1^{-1} \ ;$$

plug this into (4), we obtain

$$\begin{pmatrix} \hat{A}^{s+1} \\ \hat{C}^{s+1} \end{pmatrix} = \begin{pmatrix} 0 & (\underline{U_1}^{s+1}(S_1^{s+1})^{1/2})^\dagger \\ I & 0 \end{pmatrix} L_2^{s+1}(L_2^{s+1})^T U_1^{s+1}(S_1^{s+1})^{-3/2} \ , \tag{7}$$

which is a ready formula for the computation of the updated matrices (A, C).

Notice that in (5), forgetting factor can be added to deemphasize the history data so as to make the algorithm adapt to changes in the system's dynamic properties. This results in the following modified version of matrix (5):

$$\left(\begin{array}{cc|c} \eta L_1^s & & y_p^{s+1} \\ \eta L_2^s & \eta L_3^s & y_f^{s+1} \end{array}\right) ,$$

where $\eta \in (0, 1)$ is the forgetting factor.

III. APPLYING RSSI TO BRIDGE MODAL PARAMETER MONITORING

In this section, we first show how the general RSSI algorithm is connected to modal parameters of a structure and then use Donghai Bridge as an example to show the challenges encountered when the RSSI algorithm is directly applied to monitor the modal parameters of Donghai Bridge. Finally, our solution to overcome these challenges is presented.

A. Modal Parameter Calculations

When applying SSI algorithm to bridge modal analysis, we only need to go a little further after obtaining (A, C). Standard procedure relates the system matrices (A, C) with the modal parameters as follows [5]: eigen-decompose A as $A = \Phi \Lambda \Phi^{-1}$, where $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n)$ and $\Phi = [\phi_1 \ \phi_2 \ \cdots \ \phi_n]$ are the eigenvalue matrix and corresponding eigenvector matrix of A, respectively. The modal parameters, that is, the resonance frequencies f_i, damping ratios ζ_i and mode shape matrix V are found from

$$f_i = \frac{\arg \lambda}{2\pi\tau} \ , \tag{8}$$

$$\zeta_i = \frac{\ln|\lambda|}{\sqrt{(\ln|\lambda|)^2 + (\arg\lambda)^2}} \ , \tag{9}$$

$$V = C\Phi \ , \tag{10}$$

where τ is the sampling period.

By combining the above formulas with the RSSI algorithm, the modal parameters can also be updated each time a new measurement is made. Resonance frequency vs. time curves, for instance, can be thus obtained. If the RSSI algorithm is fast enough, this updating procedure runs in real-time.

B. Challenges

Donghai Bridge has been selected as the application example in this paper. As China's first sea-crossing bridge, Donghai Bridge, a cable-stay bridge stretching across the East China Sea has a full length of 32.50 km, a 25.32 km portion of which is above water. The main navigation span, which is of 420 m, has navigation capacity of 5000 t, navigation height of 40 m. Obviously, the monitoring system for Donghai Bridge is large-scale with a variety of quantities to be monitored and transmitted.

When the RSSI algorithm is applied to Donghai Bridge to do on-line monitoring, the acceleration signals from tens of accelerometers mainly residing around the main navigation span are used to identify the corresponding resonance frequencies. The well-synchronized acceleration data from these accelerometers are timestamped and then transmitted over network to the monitoring center. This multi-channel signal is then on-line decimated to a reasonable rate for accommodating the targeting resonance frequency range since the upper bound of the frequency range of interest for modal parameter monitoring of bridges is relatively low (usually under several to 1 Hertz). After this, the signal is fed into an RSSI algorithm to produce the frequency-time curves, representing resonance frequency monitoring of the bridge. The overall schematic gram is shown in Fig. 1.

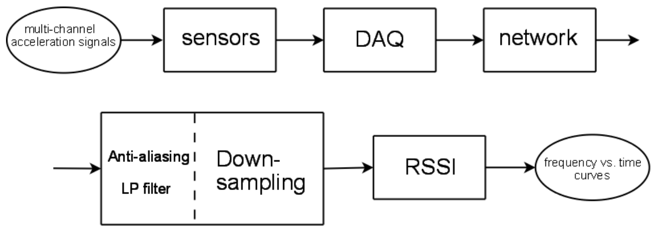

Fig. 1. on-line modal frequency monitoring schematic diagram

The results, however, reveal several limitations of this approach. First, suppose we would like to monitor frequencies under 2 Hz; then a certain number of states must be maintained in order to possibly see all the significant resonance frequencies in this range. This will enforce a lower bound of the computations that must be finished in the interval between two adjacent monitoring samples. Experiments showed that this is too much for the RSSI algorithm to run in real-time on a mainstream desktop of today. Second, this approach can fail to detect some resonance frequencies that can be otherwise easily found by some other experimental methods. This remains true when the order of the system increases to some extent. Of course, increasing the order further may reveal these "hidden" frequencies, but the price would be the presence of a lot of spurious frequencies usually to the extent that messes up the frequency-time curve graph.

C. Solution

To overcome these difficulties, we need to adjust a little the original schematic design of the whole process by inserting a step right after decimation to bandpass filter the signals into several sub-bands that is a partition of the frequency range of interest. Then in each sub-band, an RSSI algorithm is employed. This modification has two effects: first, it naturally multi-threads the algorithm so that the multi-core capacity of the today's computer can be utilized directly; second, deliberate choices of sub-bands can better expose the "hidden" frequencies without excessively increasing the system order. This modified scheme is shown in Fig. 2. The decimated signal is simultaneously filtered into several properly chosen sub frequency bands, in each of which an RSSI algorithm is applied to the filtered signal. After an initialization stage of the RSSI algorithm, the identified resonance frequencies are produced over time, forming frequency-time curves. Moreover, the speed of computation is managed to be fast enough. Thus, the resonance frequencies of the bridge are successfully being tracked in real-time. We call "mRSSI" the RSSI algorithm after this modification with "m" for "multiple". The details of these above mentioned experiments are presented in the next section.

IV. RESULTS AND DISCUSSIONS

The experiments in the following are all done based on the actual data from Donghai Bridge monitoring system. In particular, acceleration signals from 22 accelerometers

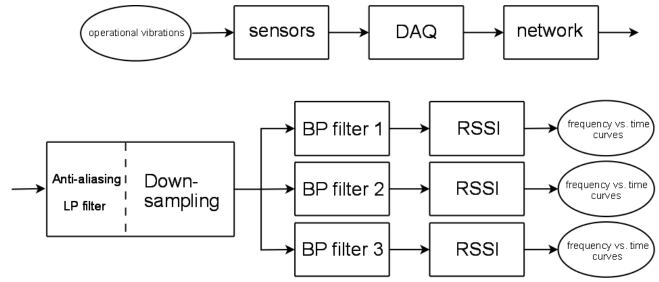

Fig. 2. Modified on-line modal frequency monitoring schematic diagram

are chosen to form the output vector y in the RSSI/mRSSI algorithms. These accelerometers are all located around the main navigation span of the bridge and they are all placed to measure the accelerations of the bridge in the vertical direction. The sampling rate is 50 Hz and the decimation factor is chosen to be 16 in all the experiments below. The raw data is of a duration of 10 minutes. So, the number of samples after decimation by a factor of 16 would be 1875 and the upper bound of the frequency range being monitored would be 1.5625 Hz, which is in accordance with the primary frequency range of interest for bridge modal analysis. The computer used in these experiments is a laptop with an Intel Core Duo 2 T7200 2G CPU.

A. Results

1) Validity: The results from the RSSI algorithm and mRSSI algorithm need to be validated with existing modal analysis results, in particular, results from existing off-line methods that are either in time-domain or frequency domain. For this purpose, both results from the on-line methods and the off-line methods need to be obtained using the same set of data set. The experiments to obtain the results from the RSSI algorithm and the mRSSI algorithm can be conveniently chosen to be the same as the two sets of experiments in the section following immediately. Thus, the results are shown in Fig. 3 and Fig. 4.

Table I lists all of the off-line results for the first several modes, in which a consistency within the off-line methods is first seen. Referring to the aforementioned two figures, an agreement with high fidelity between the on-line methods and the off-line methods is further established. Notice that "Tester 1" and "Tester 2" columns in Table I represent two separate runs of experiments using the peak-picking method. Also, there is a "Theoretical" column in Table I, which results from theoretical calculations. This can sometimes act as a guideline for the experimental modal analysis.

2) Speed: This set of experiments compares the speed of the RSSI algorithm and that of the mRSSI algorithm running on the same data set and with the same number of resonance frequencies being monitored. Specifically, there are 7 resonance frequencies being tracked in each experiment. In order to minimize the computations in the RSSI algorithm, the parameter i, who is the major factor to decide the amount of computations, is chosen to be the minimum when the 7 resonance frequencies are tracked with acceptable quality.

TABLE I
REFERRED RESONANCE FREQUENCIES OBTAINED BY DIFFERENT METHODS

Ordinal Number	Resonance Frequency (Hz)			
	SSI	Tester 1	Tester 2	Theoretical
1	0.367	0.368	0.3675	0.361
2		0.434		0.436
3	0.506	0.507	0.5096	0.505
4	0.635	0.637	0.6419	0.606
5	0.779	0.767	0.7791	0.897
6	1.037	1.040	1.0498	0.957
7	1.262	1.232	1.2305	1.042
8	1.152	1.127	1.1172	1.199

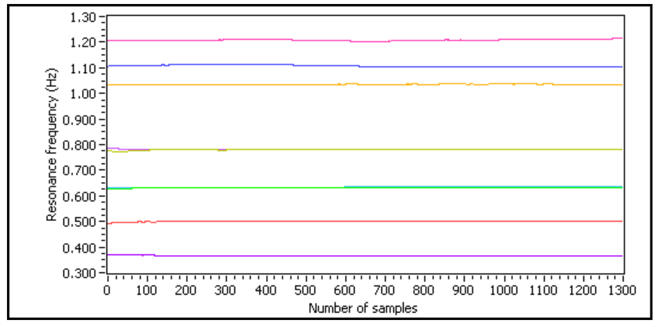

Fig. 4. Resonance frequency monitoring results by mRSSI

The (i, n) pair for the RSSI algorithm in this experiment is chosen to be $(20, 18)$. For the mRSSI algorithm, three sub-bands are chosen to cover the frequency range (0, 1.5 Hz), which are (0, 0.55 Hz), (0.55 Hz, 0.85 Hz) and (0.85 Hz, 1.5 Hz). In each band, an RSSI algorithm is employed to do frequency monitoring in that band. The (i, n) pairs for all of the three sub-bands are chosen to identically be $(12, 6)$.

The frequency-time curves (in the form of "frequency" vs. "number of samples" curves) for each case are shown in Fig. 3 and Fig. 4, respectively. (Notice that Fig. 4 is obtained by gluing together the results from each sub-band; the same approach is applied to the following experiments by the mRSSI algorithm.) The x-axis of both figures represents the number of monitoring samples (after decimation) up to that instance. From these figures, we see that the 7 monitored frequencies are effectively identical for both cases after a stage of stabilization. The time consumed per monitoring sample, however, differs greatly. The RSSI algorithm costs about 60% longer time than that of the mRSSI one $\left(\frac{334500 \text{ ms}}{838 \text{ samples}} : \frac{323594 \text{ ms}}{1298 \text{ samples}} = 1.60 : 1\right)$. This shows that the mRSSI algorithm utilizes better the power of the multi-core. A consequence of this is that when using the RSSI algorithm directly, the time consumed per monitoring sample is $334500/838$ ms $= 399.16$ ms as above, but when using mRSSI algorithm, the time consumed per monitoring sample is $323594/1298$ ms $= 249.30$ ms. The actual monitoring sample interval in real-time is $16/50$ s $= 0.32$ s $= 320$ ms. This clearly indicates that the RSSI algorithm does not run in real-time in this case, but the mRSSI one does!

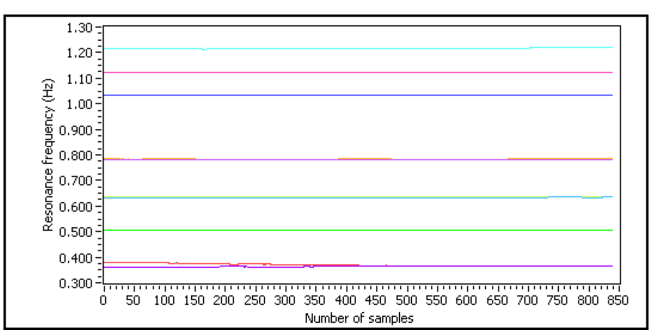

Fig. 3. Resonance frequency monitoring results by RSSI

3) "Hidden" frequencies: Some modal modes can be readily found by some common off-line experimental methods, for instance, the aforementioned frequency domain methods. When the RSSI algorithm is applied directly, however, these modes are not easily identified. The reason might be that the energies of these modes are much lower than those of the ones that are able to be identified by the RSSI algorithm. The mRSSI algorithm, when applied properly, is ideal to mitigate this issue. The following experiment shows this clearly.

In this experiment, a sub-band (0.4 Hz, 0.46 Hz) has been chosen deliberately for the single mode at about 0.434 Hz. Other sub-bands are selected accordingly. Overall, the sub-bands are (0, 0.6 Hz), (0.4 Hz, 0.46 Hz), (0.6 Hz, 1 Hz) and (1 Hz, 1.5 Hz). The (i, n) pairs in each band are chosen to be $(10, 6)$, $(10, 2)$, $(10, 6)$ and $(10, 6)$, respectively. Fig. 5 is the monitoring result, which shows there are 8 resonance frequencies being tracked, the 0.434 Hz one plus the same set of 7 frequencies in the preceding experiment. The time spent in total is 301352 ms with 1390 samples being monitored, which results in the time consumed per sample being 301352/1390 ms $= 216.8$ ms < 320 ms, which ensures the whole monitoring process is in real-time.

If instead, the RSSI algorithm is used to the whole frequency range of interest, that is, (0, 1.5 Hz), the mode at 0.434 Hz is not seen until the order of the system is chosen so high that there present too many spurious modes blurring the whole frequency-time curve graph. In fact, the (i, n) pair are chosen to be $(30, 30)$. With such a high order, the computation burden for real-time monitoring is already much higher than what an average desktop can afford. Fig. 6 is the resulting frequency-time graph in this case. There are 11 instead of 8 frequencies being monitored from this figure. The extra 3 frequencies, however, are not identified by the aforementioned off-line frequency methods. The total time spent is 367295 ms with 470 samples being monitored. So, the time consumed per sample is 781.5 ms, which is much greater than the real-time sampling interval 320 ms.

The above results demonstrate that the mRSSI algorithm is much more effective and efficient than the RSSI algorithm in terms of identifying these "hidden" resonance frequencies.

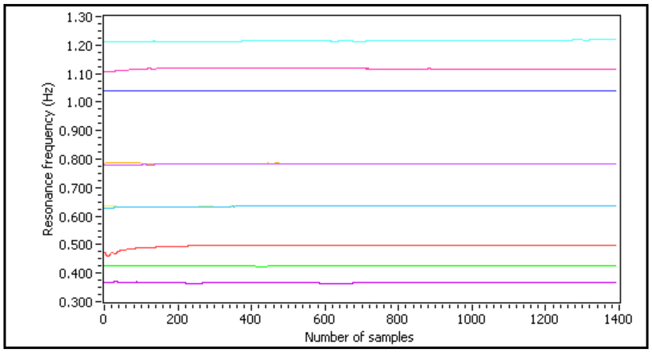

Fig. 5. "Hidden" resonance frequency monitoring by mRSSI

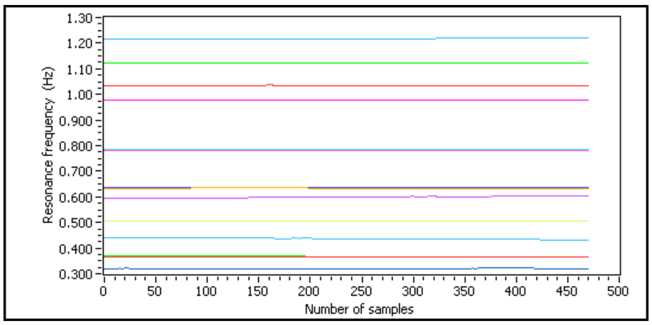

Fig. 6. "Hidden" resonance frequency monitoring by RSSI

B. Discussions

In the above experiments, the number of resonance frequencies and the "hidden" frequency are assumed to be quantities *a priori*, which actually must be obtained by some means when a monitoring task is to be performed. Off-line methods, for example the aforementioned SSI and frequency domain method, as well as theoretical modal analysis methods, can be employed to obtain information such as the number of resonance frequencies in a particular range, the frequencies distribution and the frequencies that are possibly difficult to identify. A stabilization diagram can also be constructed to facilitate the choice of the system order. The information of the system order can be used in both the RSSI algorithm and the mRSSI algorithm to set the number of n. In accordance with [3], it is a good practice to set i no fewer than n. The general rule is that the ratio between i and n should be increased as the measured signal gets more noisy. The frequency distribution information is especially helpful when the frequency range of interest is divided into several sub-bands in the mRSSI algorithm. Granted, all of these settings, especially the pair (i, n), should be finally tuned according to the actual resulting frequency-time curves produced by these on-line monitoring methods.

In this paper, we have only shown that the resonance frequencies are monitored over time. According to the monitoring purposes at hand, some other quantities, however, may be better indicators. For instance, the residue associated with the least square problem where (A, C) are derived is a common choice in damage detection. Damping ratios are also the choices in certain applications. The mode shapes can sometimes be very useful as well. The bottom line is that system matrices (A, C) are tracked in real-time so that any other quantities that are derived from them can be further calculated and thus monitored. The specific form of the indicator is of course directly dependent on the monitoring purposes. In summary, on the basis of (A, C), a lot of related indicators can be formulated and thus tracked on-line.

V. CONCLUSIONS AND FUTURE WORK

Bridge modal parameters, in particular, resonance frequencies are monitored on-line. The RSSI algorithm is discussed in details and it is further adapted to embrace the power of multi-core CPUs. The method is applied to real data from Donghai Bridge and the results are shown and discussed, which demonstrates the feasibility and effectiveness of the method proposed in this paper. The proposed mRSSI algorithm has the advantage of fully utilizing multi-core power on multi-core machines; also it can better detect resonance frequencies that are not so easily detected by the usual RSSI algorithm.

There is a family of derived applications based on items derived from the modal parameters of structures, for instance, load estimation, vibration level estimation and fatigue estimation according to [6]. Among these applications, some have the potential and necessity to be further turned on-line by using our on-line modal parameter methods. An example of this is the tension monitoring of the cables on a cable-stay bridge. Indeed, the tension can be directly derived from the modal parameters and thus, in turn, monitored. This shows the potential big impacts on the application world of the RSSI algorithm and/or mRSSI algorithm, which provides attractive future research topics. Besides, how to make less heuristic the process of choosing a proper configuration in these algorithms is also interesting and important.

VI. ACKNOWLEDGMENTS

The authors gratefully acknowledge the support from National Instruments (Ltd.), Shanghai branch and the comments from the reviewers.

REFERENCES

[1] De Cock K., Mercre G., De Moor B., Recursive subspace identification for in-flight modal analysis of airplanes, *in Proc. of the International Conference on Noise and Vibration Engineering (ISMA 2006)*, Leuven, Belgium, Sept. 2006, pp. 1563-1577.

[2] Goethals I., Mevel L., Benveniste A., De Moor B. Recursive output-only subspace identification for in-flight flutter monitoring, *in Proc. of the 22nd International Modal Analysis Conference (IMAC-XXII)*, Dearborn, Michigan, Jan. 2004.

[3] Van Overschee P., De Moor B., *Subspace Identification for Linear Systems: Theory, Implementation*, Applications, Kluwer Academic Publishers, 1996.

[4] Edward C. Real, Donald W. Tufts and James W. Cooley, Two Algorithms for Fast Approximate Subspace Tracking, *IEEE Transactions on Signal Processing*, Vol.47, No.7, July 1999, pp. 1936-1945.

[5] M. Basseville, L. Mevelb and M. Goursatc, Statistical model-based damage detection and localization: subspace-based residuals and damage-to-noise sensitivity ratios, *Journal of Sound and Vibration*, Vol. 275, Issue 3-5, 23 August 2004, pp. 769-794.

[6] R. Brincker, C. Ventura, P. Andersen, Why Output Only Modal Analysis is a Desirable Tool for a Wide Range of Practical Applications, *Proceedings of the International Modal Analysis Conference (IMAC)*, February 3-6, 2003, Kissimmee, Florida, USA., 2003.