Distributed Sensing for High Quality Structural Health Monitoring using Wireless Sensor Networks

Xuefeng Liu, Jiannong Cao
Dept. of Computing
The Hong Kong Polytechnic University
{csxfliu,csjcao}@comp.polyu.edu.hk

Wen-Zhan Song
Dept. of Computer Science
Georgia State University
wsong@gsu.edu

Shaojie Tang
Dept. of Computer Science
Illinois Institute of Technology
tangshaojie@gmail.com

***Abstract*—In recent years, using wireless sensor networks (WSNs) for structural health monitoring (SHM) has attracted increasing attention. Traditional centralized SHM algorithms developed by civil engineers can achieve the highest damage detection quality since they have the raw data from all the sensor nodes. However, directly implementing these algorithms in a typical WSN is impractical considering the large amount of data transmissions and extensive computations required. Correspondingly, many SHM algorithms have been tailored for WSNs to become distributed and less complicated. However, the modified algorithms usually cannot achieve the same damage detection quality of the original centralized counterparts. In this paper, we select a classical SHM algorithm: the eigen-system realization algorithm (ERA), and propose a distributed version for WSNs. In this approach, the required computations in the ERA are updated incrementally along a path constructed from the deployed sensor nodes. This distributed version is able to achieve the same quality of the original ERA using much smaller wireless transmissions and computations. The efficacy of the proposed approach is demonstrated through both simulation and experiment.**

I. Introduction

Recent years have witnessed a booming interests of using wireless sensor networks (WSNs) for structural health monitoring (SHM) because of the advantage of low-cost and ease of deployment. In a typical WSN-based SHM system, sensors are deployed on different locations of a structure to collect the structure's responses under ambient or forced excitation. These data are then transmitted wirelessly to a server where one or more SHM algorithms are implemented to extract damage-sensitive vibration characteristics. By examining these characteristics, damage can be detected and located [1].

SHM is generally implemented in a round-by-round manner. In each round, to accurately capture the vibration response of civil infrastructure, deployed sensors need to acquire data at a 'high enough' sampling rate for a 'long enough' period of time [2]. The number of data collected at each node in a single round hence can reach the level of tens of thousands. Due to the large amount of data generated and the generally limited wireless bandwidth and energy supply in WSNs, it is difficult, if not impossible for wireless nodes to continuously transmit the raw data to the designated recipient.

To address this problem, in-network processing has always been regarded as the first choice. Through in-network processing, only important information, which uses much fewer bits than the raw data, needs to be transmitted. In-network processing is therefore able to save energy and wireless bandwidth without sacrificing system's performance.

Although there exist many types of in-network processing techniques in WSNs such as data compression [3] etc., we are more interested in implementing SHM algorithms originally proposed by civil engineers because from which, damage information can be directly obtained. Some SHM algorithms are intrinsically distributed and allow each node to process data independently without sharing information with others. Implementing these algorithms in a WSN is relatively simple and the existing work can be found in [4] and [5].

However, most SHM algorithms are centralized and require the raw data from all deployed nodes. Embedding these algorithms within a WSN is much more difficult. An example of embedding centralized SHM algorithms within a WSN be found in [6], where a multi-level damage localization strategy is proposed. In [7] and [8], some SHM algorithms such as the ERA[9] and the FDD[10] are made distributed through the idea of 'divide and conquer'. Other existing works in this area can be found in [11] and [12].

However, one problem that generally exists in WSN-tailored SHM algorithms is that the damage detection capability of these algorithms cannot be guaranteed to be comparable with the centralized counterpart. Due to this drawback, we found that our civil collaborators prefer to use wireless sensor nodes simply as 'data collectors' despite its potential capability to self-detect damage. Another difficulty of embedding SHM algorithms with a WSN is associated with the computational resources they require. Many SHM algorithms, particularly for the centralized ones, require complicated signal processing techniques such as matrix computation and system identification [9] and hence cannot be directly embeded in the generally resource-constrained wireless sensor nodes.

In this paper, we proposed a distributed version of a SHM algorithm: the eigensystem realization algorithm (ERA) [9]. The ERA is a classical algorithm and has been widely used by civil engineers to detect structural damage. The distributed ERA is able to achieve the same accuracy of the centralized counterpart using much less wireless transmissions as well as computational resources. We also show that the advantage of this distributed version becomes more significant in a sparse network with large number of sensor nodes. This property is

1052-8725/12 $26.00 © 2012 IEEE
DOI 10.1109/RTSS.2012.60

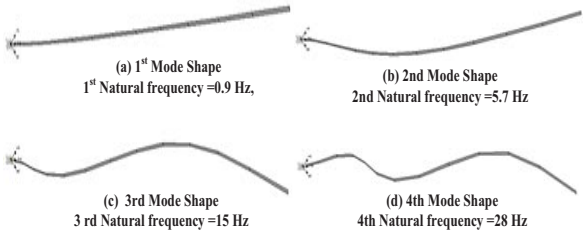


Fig. 1: The first four modal parameters of a cantilever beam

very intriguing since it fits well the real situation when WSNs are used in large civil infrastructures.

II. Modal parameters and the ERA

In this section, we introduce some fundamentals of the ERA.

A. Modal Parameters

According to the vibration theory, every structure has tendency to oscillate with much larger amplitude at some frequencies than others. These frequencies are called natural frequencies. When a structure is vibrating under one of its natural frequencies, the corresponding vibration pattern it exhibits is called the mode shape for this natural frequency. For example, the natural frequencies and the corresponding mode shapes of a n-degrees-of-freedom structure are denoted respectively as:

$$\Omega = \{\omega_1, \omega_2, ..., \omega_n\}, \mathbf{\Psi} = [\mathbf{\Psi_1}, \mathbf{\Psi_2}, \cdots, \mathbf{\Psi_n}] \quad (1)$$

where ω_i and $\mathbf{\Psi_i} = [\phi_i^1, \phi_i^2, \cdots, \phi_i^n]^T$ $(i = 1, ..., n)$ are the i^{th} natural frequency and the corresponding mode shape, respectively. $\phi_i^k (k = 1, 2, ..., n)$ is the value of $\mathbf{\Psi_i}$ at the k^{th} degree of freedom. For convenience, ω_i and $\mathbf{\Psi_i}$ are also called **modal parameters** corresponding to the i^{th} **mode** of a structure. As an example, Fig. 1 shows the first few natural frequencies and mode shapes of a typical cantilevered beam.

Modal parameters ω_i and $\mathbf{\Psi}_i$ are internal characteristics of structure. By examining the changes in the identified modal parameters, possible damage can be detected and further located. Modal parameters are identified based on the acceleration time histories measured by deployed sensor nodes.

B. The ERA

In this section, we briefly introduce how he modal parameters are estimated in the ERA.

Assume a total of m sensor nodes are deployed on a structure and the collected data are denoted as $\mathbf{y}(k) = [y^1(k), y^2(k), ..., y^m(k)]'$ $(k = 1, ..., N_{ori})$ where $y^i(k)$ is the data sampled by the i^{th} sensor at k^{th} time step and N_{ori} is the total number of data points collected in each node. To obtain modal parameters, the ERA first identifies, from measured responses $\mathbf{y}(k)$, a series of parameters $\mathbf{Y}(k)$ called **Markov parameters**. The Markov parameters $\mathbf{Y}(k)$ are defined as the cross-correlation function (CCF) of the measurement $\mathbf{y}$ and a reference signal y^{ref}:

$$\mathbf{Y}(k) = [CCF_{y^1 y^{ref}}(k), \cdots, CCF_{y^m y^{ref}}(k)]^T \quad (2)$$

where $CCF_{y^i y^{ref}}$ is the CCF between y^i and y^{ref}. Generally speaking, measured signal from any sensor can be selected as y^{ref}. To accurately estimate $CCF_{y^i y^{ref}}$, we first use the Welch's method [13] to calculate the cross spectral density (CSD) between y^i and y^{ref}, and then implement inverse Fourier transform $(ifft)$ on the CSD to obtain the CCF.

In the Welch's method, to calculate the CSD of two signals x and y, x and y are first divided into n_d number of overlapping segments. The CSD of x and y, denoted as G_{xy} is then calculated as

$$G_{xy}(\omega) = \frac{1}{n_d \cdot N} \sum_{i=1}^{n_d} X_i^*(\omega) \cdot Y_i(\omega) \quad (3)$$

where $X_i(\omega)$ and $Y_i(\omega)$ are the Fourier transforms of the i^{th} segment of x and y, and '*' denotes the complex conjugate. N is the number of data points in each segment of x (or y) as well as the obtained $G_{xy}(\omega)$. N is generally taken as 1024 or 2048 to give reasonable results. n_d practically ranges from 10 to 20. Note that if 50% overlap of two consecutive segments is employed in the CSD estimation, x and y need to contain at least $N(\frac{n_d}{2} + 0.5)$ data points. For convenience, unless specified, we set $N = 2048$, $n_d = 20$ and the overlap is 50% in this paper. Correspondingly, the number of raw data to be sampled at each sensor node should be at least $\mathbf{21,504}$.

Having obtained the CSD of y^{ref} with each signal in $\mathbf{y}$, the Markov parameters $\mathbf{Y}(k)$ is then calculated as the CCF of the CSD. Note that only first half of the obtained CCF are used as the Markov parameters (i.e. $k = 1, ..., N/2$).

With the obtained $\mathbf{Y}(1)$,$\mathbf{Y}(2)$,$\cdots$,$\mathbf{Y}(N/2)$, the ERA begins by forming a Hankel matrix using these Markov parameters. Moreover, instead in the original form of Hankel matrix, the ERA algorithm allows us to use a modified one shown below:

$$\mathbf{H}(k-1) = \begin{bmatrix} \mathbf{Y}(k) & \mathbf{Y}(k+\alpha) & \cdot\cdot & \mathbf{Y}(k+\beta\alpha-\alpha) \\ \mathbf{Y}(k+1) & \mathbf{Y}(k+\alpha+1) & \cdot\cdot & \cdot\cdot \\ \vdots & & & \\ \mathbf{Y}(k+\alpha-1) & \mathbf{Y}(k+2\alpha-1) & \cdot\cdot & \mathbf{Y}(k+\beta\alpha-1) \end{bmatrix} \quad (4)$$

Note that we only need to calculate two Hankel matrices $\mathbf{H}(0)$ and $\mathbf{H}(1)$ $(\mathbf{H}(0), \mathbf{H}(1) \in \mathbb{R}^{\alpha m \times \beta})$ to identify modal parameters. α and β in Eq. 4 correspond to the number of block rows and columns in the Hankel matrix. In this paper, $\alpha = 10$ and $\beta = 100$, which are large enough to accurately identify the modal parameters of important modes [9].

Perform the singular value decomposition (SVD) on $\mathbf{H}(0)$:

$$\mathbf{H}(0) \stackrel{svd}{=} \mathbf{U}\mathbf{S}\mathbf{V}^T \quad (5)$$

where $\mathbf{U} \in \mathbb{R}^{\alpha m \times \alpha m}$ and $\mathbf{V}^T \in \mathbb{R}^{\beta \times \beta}$ are matrices of left and right eigenvectors of $\mathbf{H}(0)$, respectively, and $\mathbf{S} \in \mathbb{R}^{\alpha m \times \beta}$ is the diagonal matrix of singular values. Both $\mathbf{U}$ and $\mathbf{V}$ are unitary matrices which means that

$$\mathbf{U}^T\mathbf{U} = \mathbf{U}\mathbf{U}^T = \mathbf{I}_{\alpha m} \quad (6)$$

and

$$\mathbf{V}^T\mathbf{V} = \mathbf{V}\mathbf{V}^T = \mathbf{I}_\beta \tag{7}$$

According to the vibration theory [14], for a structure whose vibration is dominated by the first n modes, the rank of $\mathbf{H}(0)$ is only $2n$ and the singular values in $\mathbf{S}$ has $2n$ non-zero values:

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}_{2n} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{8}$$

where $\mathbf{S}_{2n} = diag(d_1, \cdots, d_{2n})$, $d_1 \geq \cdots \geq d_{2n} > 0$ are the singular values of $\mathbf{H}(0)$. Correspondingly, $\mathbf{H}(0)$ can be re-expressed as:

$$\mathbf{H}(0) = \mathbf{U}_{2n}\mathbf{S}_{2n}\mathbf{V}_{2n}^T \tag{9}$$

where $\mathbf{U}_{2n} \in \mathbb{R}^{\alpha m \times 2n}$ and $\mathbf{V}_{2n} \in \mathbb{R}^{\beta \times 2n}$ are the first $2n$ columns of the matrices $\mathbf{U}$ and $\mathbf{V}$, respectively. Notice that although $\mathbf{U}$ and $\mathbf{V}$ are not unitary, they still contain orthogonal columns:

$$\mathbf{U}_{2n}^T\mathbf{U}_{2n} = \mathbf{V}_{2n}^T\mathbf{V}_{2n} = \mathbf{I}_{2n} \tag{10}$$

Using Eq. 9 and the Hankel matrix $\mathbf{H}(1)$ constructed from Eq. 4, two matrices $\mathbf{A}$ and $\mathbf{\Gamma}$ are found [9]:

$$\mathbf{A} = \mathbf{S}_{2n}^{-1/2}\mathbf{U}_{2n}^T \tag{11}$$

$$\mathbf{H}(1)\mathbf{V}_{2n}\mathbf{S}_{2n}^{-1/2}, \mathbf{\Gamma} = [\mathbf{I}_{2n}, \mathbf{0}]\mathbf{U}_{2n}\mathbf{S}_{2n}^{-1/2} \tag{12}$$

Performing the eigen decomposition on $\mathbf{A}$:

$$\mathbf{A} = \mathbf{\Phi}\Lambda\mathbf{\Phi}^{-1} \tag{13}$$

where $\mathbf{\Phi} \in \mathbb{R}^{2n \times 2n}$, and $\Lambda = diag(\lambda_1, \lambda_2, ..., \lambda_n)$. The natural frequency ω_i can be determined as

$$\omega_i = \sqrt{[Imag(s_i)]^2 + [Real(s_i)]^2} \tag{14}$$

where $s_i = \frac{ln(\lambda_i)}{\Delta t}$, Δt is the sampling period when the raw data is collected, $Imag(.)$ and $Real(.)$ are the imaginary and real part a complex number, respectively. The mode shape $\mathbf{\Psi}$ at the measured degrees of freedom, denoted as $\hat{\mathbf{\Psi}}$ can be calculated as

$$\hat{\mathbf{\Psi}} = \mathbf{\Gamma}\mathbf{\Phi} \tag{15}$$

The whole procedure of ERA can be summarized as two stages: the Markov parameters are identified first and based on which, the modal parameters are then identified.

III. The Challenges and State of Art

If we were to embed the ERA in resource-limited WSNs, at least two riddles need to be resolved. First, the ERA is centralized and requires measured raw data from all the sensors to be delivered to a central station. This is not suitable for WSNs because the amount of required data transmissions is large. We need to find a way to make it distributed so as to save wireless transmissions. Second, since the ERA requires many intensive computations such as the SVD, we need to reduce the computational load of these computations if they are to be implemented in resource-limited wireless sensor nodes.

Recently, the ERA has been tailored for WSNs and become computationally less inexpensive and/or require less wireless transmissions. For example, in [8], a cluster-based ERA has been proposed to identify modal parameters which uses the idea similar to the 'divide and conquer'. In this cluster-based approach, deployed nodes are divided into a number of single-hop clusters. Local modal parameters within each cluster are identified first and then are 'stitched' together to obtain the modal parameters for the whole structure.

However, we should notice that the accuracy of modal parameters obtained using this approach cannot be guaranteed to be comparable with the centralized one. This problem also occurs in many WSN-tailored SHM algorithms which use the idea of 'divide and conquer' approach to identify modal parameters [15] in a distributed way. This is because each cluster only uses its local information, which can result in the ill-conditioned problem which reduces the accuracy of identified local modal parameters [9], and this inaccuracy cannot be rectified via the 'stitching' process afterwards.

IV. The Distributed ERA

In this section, we will introduce the design of the distributed ERA. The goal is to *achieve the same or similar accuracy of the centralized counterpart but using much less computational resources and wireless transmissions.* For clarity, we use a step-by-step manner and one improvement is described in each of the following five subsections.

A. Using broadcast instead of aggregation

In the traditional ERA, measured raw data from all the nodes are first streamed to a sink, where the CCF and the Markov parameters are calculated. However, from Eq. 2, it can be seen that Markov parameters represent the relationship between signal pairs, and hence they do not have to be calculated in the sink. Instead, the reference node can broadcast its data to the whole network. Using the received data, each node can calculate its Markov parameters. The Markov parameters of all nodes are further transmitted to the sink, where they form into Hankel matrices to estimate the modal parameters. The data flow in this scheme is shown in Fig. 2.

Compared with the traditional ERA, this modification along already saves wireless transmissions significantly compared with the centralized ERA. For example, to obtain a total of $N/2$ Markov parameters for each of the m sensor nodes, The total data to be transmitted in the centralized scheme is:

$$E_1 = N/2(n_d + 1)\sum_{i=1}^{m} Dep^i \tag{16}$$

where Dep^i is the depth of the i^{th} node in the shortest path tree (SPT); while in the modified scheme, the required data transfer is only

$$E_2 = E_{bc} + N/2\sum_{i=1}^{m} Dep^i \tag{17}$$

, where E_{bc} is the total data to be transmitted when broadcasting the reference data. Considering in our case, $n_d = 20$

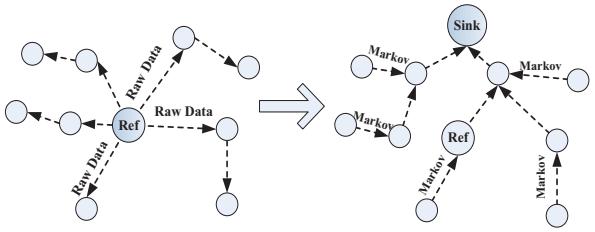


Fig. 2: Data flow of this approach. Left: Ref broadcasts its data, Right: Sensor nodes transmit Markov parameters

and E_{bc} is a relative small value compared with the first term in Eq. 17, we have $E_2 << E_1$.

However, this scheme still has some drawbacks. First, the sink should be in charge of implementing the SVD of $\mathbf{H}(0)$ as in the centralized approach. Considering the expensive associated computations required, implementing the SVD of $\mathbf{H}(0)$ including a large number of nodes generally needs a PC. Second, in each round of damage detection, $N/2$ Markov parameters need to be transmitted from each node. This still a time and energy consuming task. These two problems will be addressed in the following sections.

B. Calculating the SVD of $\mathbf{H}(0)$ *incrementally*

In this section, we mainly focus on how the most expensive part in the ERA, the SVD of $\mathbf{H}(0)$, can be calculated in a more efficient way.

The traditional SVD is implemented based on the Hankel matrix constructed using all the sensor nodes. For a $\mathbf{H}(0) \in \mathbb{R}^{\alpha m \times \beta}$ matrix of with rank n, the time complexity and space complexity are $O((\alpha m)^2\beta + \beta^2 \cdot \alpha m)$ and $O(\alpha m \cdot \beta)$, respectively [16]. To reduce the computational resource required, the SVD of $\mathbf{H}(0)$ should be implemented in an incremental manner. Incremental SVD means that, the SVD of a 'small-sized' Hankel matrix $\mathbf{H}(0)$ is calculated first which only involves data from a few sensor nodes. Then the data from remaining ones are incorporated incrementally into $\mathbf{H}(0)$ and each time the data from a new sensor is added, the SVD of the updated $\mathbf{H}(0)$ is obtained using only the previous SVD result and the newly added data.

Some incremental SVD updating methods can be found in [17][18]. These methods can solve the folloing problem: given the old matrix $\mathbf{A} \in \mathbb{R}^{p \times q}$ with rank n $(p, q >> n)$, assume we already have $\{\mathbf{U}, \mathbf{S}, \mathbf{V}\}$, which are the SVD results of $\mathbf{A}$. Let $\mathbf{T} \in \mathbb{R}^{r \times q}$ be the new matrix that will be added at the bottom of $\mathbf{A}$. The incremental SVD is able to obtain $\{\mathbf{U}', \mathbf{S}', \mathbf{V}'\}$, which are the SVD results of the new matrix $\begin{pmatrix}\mathbf{A}\\ \mathbf{T}\end{pmatrix}$, using only $\{\mathbf{U}, \mathbf{S}, \mathbf{V}\}$ and $\mathbf{T}$. It should be noted that after updating, $\mathbf{S}'$ is still a diagonal matrix as $\mathbf{S}$, and both $\mathbf{U}'$ and $\mathbf{V}'$ are still unitary as were the case for $\mathbf{U}$ and $\mathbf{V}$: $\mathbf{U}'^T\mathbf{U}' = \mathbf{V}'^T\mathbf{V}' = \mathbf{I}$. According to [18], for a $\mathbf{H}(0) \in \mathbb{R}^{\alpha m \times \beta}$ matrix of with rank n, the incremental method has time complexity $O(\alpha m \cdot \beta \cdot n)$ and space complexity $O((\alpha m + \beta)n)$- much better than traditional SVD.

Unfortunately, these method cannot be directly used here to update the SVD of Hankel matrix $\mathbf{H}(0)$ since *the newly added Markov parameters from a sensor node are not appended at the bottom of the* $\mathbf{H}(0)$ *but at the bottom of each* ***block row*** *of* $\mathbf{H}(0)$ (see the definition of Hankel matrix Eq. 4). Modification is therefore needed.

Let us suppose that initially, we have constructed $\mathbf{H}^m(0)$, the Hankel matrix $\mathbf{H}(0)$ including m nodes:

$$\mathbf{H}^m(0) = \begin{bmatrix} \mathbf{Y}(1) & \mathbf{Y}(\alpha+1) & \cdots & \mathbf{Y}(\beta\alpha-\alpha+1) \\ \mathbf{Y}(2) & \mathbf{Y}(\alpha+2) & \cdots & \mathbf{Y}(\beta\alpha-\alpha+2) \\ \vdots & & & \\ \mathbf{Y}(\alpha) & \mathbf{Y}(2\alpha) & \cdots & \mathbf{Y}(\beta\alpha) \end{bmatrix} \quad (18)$$

where $\mathbf{Y}(k) = [Y^1(k), Y^2(k), \cdots, Y^m(k)]^T$ and $Y^i(k)$ is the k^{th} Markov parameter of the i^{th} node.

Still assume the structure under test is dominated by its first n modes. The SVD of $\mathbf{H}^m(0)$ is implemented and we obtain three matrices $\mathbf{U}_{2n}$, $\mathbf{S}_{2n}$, and $\mathbf{V}_{2n}$ and they satisfy

$$\mathbf{H}^m(0) = \mathbf{U}_{2n}\mathbf{S}_{2n}\mathbf{V}_{2n}^T \quad (19)$$

Assume now we have Markov parameters from another sensor node $\{Y^{m+1}(1), Y^{m+1}(2), ..., Y^{m+1}(N/2)\}$. Then the Hankel matrix $\mathbf{H}^{m+1}(0)$ which corresponds to all these $m+1$ sensor nodes can be represented as:

$$\mathbf{H}^{m+1}(0) = \begin{bmatrix} \mathbf{Y}(1) & \mathbf{Y}(\alpha+1) & \cdot\cdot & \mathbf{Y}(\beta\alpha-\alpha+1) \\ Y^{m+1}(1) & Y^{m+1}(\alpha+1) & \cdot\cdot & \cdot\cdot \\ \vdots & & & \\ \mathbf{Y}(\alpha) & \mathbf{Y}(2\alpha) & \cdot\cdot & \mathbf{Y}(\beta\alpha) \\ Y^{m+1}(\alpha) & Y^{m+1}(2\alpha) & \cdot\cdot & Y^{m+1}(\beta\alpha) \end{bmatrix} \quad (20)$$

To use the incremental SVD updating in [18], *we need perform a series of row switching procedures on* $\mathbf{H}^{m+1}(0)$*, through which the Markov parameters of the newly added sensor node have been put at the end of the matrix.* These row switching procedures can be represented as a transformation matrix $\tilde{\mathbf{T}}_s \in \mathbb{R}^{\alpha(m+1) \times \alpha(m+1)}$ which satisfies:

$$\tilde{\mathbf{T}}_s\mathbf{H}^{m+1}(0) = \begin{bmatrix} & \mathbf{H}^m(0) & & \\ Y^{m+1}(1) & Y^{m+1}(\alpha+1) & \cdots & \cdots \\ \vdots & & & \\ Y^{m+1}(\alpha) & Y^{m+1}(2\alpha) & \cdots & \cdots \end{bmatrix} \quad (21)$$

Let the SVD results for $\tilde{\mathbf{T}}_s\mathbf{H}^{m+1}(0)$ be denoted as $\tilde{\mathbf{U}}_{2n}$, $\tilde{\mathbf{S}}_{2n}$ and $\tilde{\mathbf{V}}_{2n}$. $\{\tilde{\mathbf{U}}_{2n}, \tilde{\mathbf{S}}_{2n}, \tilde{\mathbf{V}}_{2n}\}$ satisfy

$$\tilde{\mathbf{T}}_s\mathbf{H}^{m+1}(0) \overset{svd}{=} \tilde{\mathbf{U}}_{2n}\tilde{\mathbf{S}}_{2n}\tilde{\mathbf{V}}_{2n}^T \quad (22)$$

and they can be calculated using the SVD of $\mathbf{H}^m(0)$ and the newly obtained Markov parameters $\{Y^{m+1}(1), Y^{m+1}(2), ..., Y^{m+1}(N/2)\}$. Multiply $\tilde{\mathbf{T}}_s^{-1}$ at both sides of Eq. 22, we can obtain three matrices $\mathbf{U}'_{2n}$, $\mathbf{S}'_{2n}$ and $\mathbf{U}'_{2n}$, which satisfy the following equation:

$$\mathbf{H}^{m+1}(0) = \tilde{\mathbf{T}}_s^{-1}\tilde{\mathbf{U}}_{2n}\tilde{\mathbf{S}}_{2n}\tilde{\mathbf{V}}_{2n}^T \quad (23)$$

Define $\mathbf{U}'_{2n} = \tilde{\mathbf{T}}_s^{-1}\tilde{\mathbf{U}}_{2n}$, $\mathbf{S}'_{2n} = \tilde{\mathbf{S}}_{2n}$,$\mathbf{V}'_{2n} = \tilde{\mathbf{V}}_{2n}$, we have

$$\mathbf{H}^{m+1}(0) = \mathbf{U}'_{2n}\mathbf{S}'_{2n}\mathbf{V}'^{T}_{2n} \tag{24}$$

Lemma 1. *$\mathbf{U}'_{2n}$, $\mathbf{S}'_{2n}$ and $\mathbf{V}_{2n}$ in the Eq. 24 are the SVD of $\mathbf{H}^{m+1}(0)$: $\mathbf{S}'_{2n}$ is a diagonal matrix, and $\mathbf{V}'^{T}_{2n}\mathbf{V}'_{2n} = \mathbf{U}'^{T}_{2n}\mathbf{U}'_{2n} = \mathbf{I}_{2n}$*

Proof: According to incremental SVD algorithm in [18], $\mathbf{S}'_{2n} = \tilde{\mathbf{S}}_{2n}$ is a diagonal matrix and $\mathbf{V}'_{2n} = \tilde{\mathbf{V}}_{2n}$ is composed of orthogonal columns. To prove that all the columns in $\mathbf{U}'_{2n}$ are orthogonal with each other, we first define the primary matrix switching transformation, $\mathbf{T}_{ij}$, when it multiplies with a matrix $\mathbf{A}$, switches all the matrix elements of $\mathbf{A}$ in row i with their counterparts in row j. $\mathbf{T}_{ij}$ is represented as:

$$\mathbf{T}_{ij} = \begin{array}{c} \\ {\scriptstyle i^{th}row}\rightarrow \\ \\ {\scriptstyle j^{th}row}\rightarrow \\ \\ \end{array} \begin{array}{c} \begin{array}{ccccccc} & & {\scriptstyle i^{th}col} & & & {\scriptstyle j^{th}col} & \\ & & \downarrow & & & \downarrow & \end{array} \\ \begin{pmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 0 & \cdots & 1 & & \\ & & & \ddots & & & \\ & & 1 & \cdots & 0 & & \\ & & & & & \ddots & \\ & & & & & & 1 \end{pmatrix} \end{array} \tag{25}$$

By comparing Eq. 20 with Eq. 21, it can be seen that converting $\mathbf{H}^{m+1}(0)$ to $\tilde{\mathbf{T}}_s\mathbf{H}^{m+1}(0)$ can be realized using a series of simple $\mathbf{T}_{ij}$ transformations. This indicates that

$$\tilde{\mathbf{T}}_s = \prod_{k=1}^{z}\mathbf{T}_{a_k b_k} \tag{26}$$

According to [16], the inverse of $\mathbf{T}_{ij}$ is itself: $\mathbf{T}_{ij}^{-1} = \mathbf{T}_{ij}$, therefore, $\tilde{\mathbf{T}}_s^{-1} = \prod_{k=z}^{1}\mathbf{T}_{a_k b_k} = \tilde{\mathbf{T}}_s^{T}$, we have:

$$\mathbf{U}'^{T}_{2n}\mathbf{U}'_{2n} = (\tilde{\mathbf{T}}_s^{-1}\tilde{\mathbf{U}}_{2n})^{T}(\tilde{\mathbf{T}}_s^{-1}\tilde{\mathbf{U}}_{2n}) = \mathbf{I}_{2n} \tag{27}$$

Therefore, we have proved that $\mathbf{U}'_{2n}$,$\mathbf{S}'_{2n}$, and $\mathbf{V}'_{2n}$ shown in the Eq. 23 are the SVD of $\mathbf{H}^{m+1}(0)$. ■

By comparing Eq. 21 and Eq. 22, it can be seen that $\tilde{\mathbf{T}}_s$ can expressed as

$$\tilde{\mathbf{T}}_s = \begin{array}{c} \\ {\scriptstyle (m+1)^{th}_{row}}\rightarrow \\ \\ {\scriptstyle (2m+1)^{th}_{row}}\rightarrow \\ \\ {\scriptstyle (m\alpha+1)^{th}_{row}}\rightarrow \\ {\scriptstyle (m\alpha+2)^{th}_{row}}\rightarrow \\ \\ {\scriptstyle (m\alpha+\alpha)^{th}_{row}}\rightarrow \end{array} \begin{pmatrix} 1 & & & & & & & & & \\ & \ddots & & & & & & & & \\ & & 1 & & & & & & & \\ & & & 0 & 1 & & & & & \\ & & & & & \ddots & & & & \\ & & & & & & 1 & & & \\ & & & & & & & 0 & 1 & \\ & & \vdots & & & & & & & \ddots \\ & & & 1 & & & & & & \\ & & & & & & & 1 & & \\ & & & & & & \vdots & & & \\ & & & & & & & & & 1 \end{pmatrix} \tag{28}$$

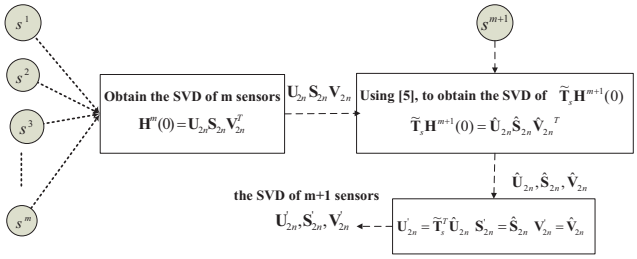


Fig. 3: Incrementally updating the SVD of $\mathbf{H}(0)$

Also because $\tilde{\mathbf{T}}_s^{-1} = \tilde{\mathbf{T}}_s^{T}$, we can use $\mathbf{U}'_{2n} = \tilde{\mathbf{T}}_s^{T}\tilde{\mathbf{U}}_{2n}$ to replace $\mathbf{U}'_{2n} = \tilde{\mathbf{T}}_s^{-1}\tilde{\mathbf{U}}_{2n}$ when calculating $\mathbf{U}'_{2n}$. An even better approach can be utilized in which we do not need to find out the mathematical expression of $\tilde{\mathbf{T}}_s^{-1}$. Since $\tilde{\mathbf{T}}_s$ moves the Markov parameters of the newly added sensor node from each block row of $\mathbf{H}^{m+1}(0)$ to the last α rows, we only need to do the reverse operations on $\tilde{\mathbf{U}}_{2n}$ to obtain $\tilde{\mathbf{T}}_s^{-1}\tilde{\mathbf{U}}_{2n}$. The above procedures are summarized in Fig. 3.

There is one requirement about the number of nodes that need to be included in the initial Hankel matrix $\mathbf{H}(0)$ before updating can be started. The incremental SVD updating method in [18] requires that the rank of initial matrix $\mathbf{A}$ implementing SVD should be no less than the final matrix after the new data $\mathbf{T}$ is added: i.e. $rank(\mathbf{A}) = rank(\begin{pmatrix}\mathbf{A}\\ \mathbf{T}\end{pmatrix})$. As was mentioned in section II, if the vibration of a structure is dominated by its first n modes, the order of the final Hankel matrix $\mathbf{H}(0)$ is $2n$. Considering the fact that the vibration of most large civil infrastructures is dominated by only the first few modes, $n = 5$ is used in this paper. Therefore, the rank of initial $\mathbf{H}(0)$ should be larger than 10. Since $\alpha = 10$ and $\beta = 100$ are adopted when constructing Hankel matrix in this paper, we require that the number of sensor nodes included in the initial $\mathbf{H}(0)$ should be larger than 4. This makes the size of the initial $\mathbf{H}(0)$ be 40-by-100, leaving enough margin to ensure the rank of initial $\mathbf{H}(0)$ meet the requirement.

We summarize the procedures when the incremental SVD updating scheme is adopted in WSNs:

1) A reference node broadcasts its data to the network;
2) All the other nodes receive the data, calculate their Markov parameters and transmit them to the sink;
3) After the sink receives the Markov parameters from 4 nodes, it constructs an initial $\mathbf{H}(0)$ and calculates its SVD. The SVD will be updated as in Fig. 3 when the Markov parameters from new nodes are received.
4) Having received the Markov parameters from all the nodes, the sink constructs $\mathbf{H}(1)$ and using the latest $\mathbf{H}(0)$ to identify modal parameters.

C. Transmitting SVD Results Instead of Markov Parameters

Incremental SVD updating, although can decrease the computational load, is essentially still a centralized approach since it is implemented at a sink node. In this section, we will introduce an approach to decrease the wireless transmissions associated with the delivery of Markov parameters.

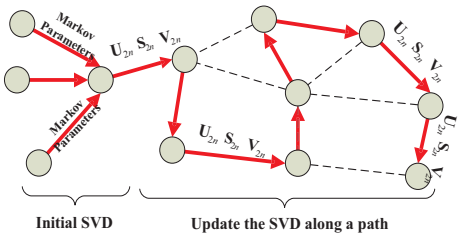


Fig. 4: Incrementally and distributedly updating the SVD

The key still lies in the SVD. Assume the $N/2$-length Markov parameters from each of the m nodes, when they on the way to the sink, will go through a certain node. Correspondingly, the total number of data to be forwarded by this node to its parent, denoted as E_{T0}, is calculated as:

$$E_{T0} = \frac{mN}{2} \tag{29}$$

On the other hand, if this node constructs a Hankel matrix using these Markov parameters, implements the SVD, and then transmits the SVD results $\{\mathbf{U}_{2n}, \mathbf{S}_{2n}, \mathbf{V}_{2n}\}$, the data size to be transmitted, denoted as E_{T1}, would be

$$E_{T1} = \alpha m \cdot 2n + (2n)^2 + 2n \cdot \beta \tag{30}$$

where the three terms correspond to the data amount of $\mathbf{U}_{2n}$, $\mathbf{S}_{2n}$, and $\mathbf{V}_{2n}$, respectively. Considering $N = 2048$, $n = 5$, $\alpha = 10$ and $\beta = 100$, it can be further approximated as:

$$E_{T1} \approx \frac{mN}{20} + P \tag{31}$$

where $P = (2n)^2 + 2n\beta = 1100$ is a constant value. Comparing Eq. 29 with Eq. 31, we can see *that transmitting Markov parameters in the form of SVD is more efficient than in their original form even when m is small, and this advantage is more obvious with the increase of m.*

Inspired by this observation, we design a scheme as shown in Fig. 4 to update the SVD of $\mathbf{H}(0)$ incrementally and distributedly. Initially, a number of nodes ($m \geq 3$ in this paper) transmit their identified Markov parameters to a designated node. This node then constructs, from the received and its own Markov parameters, a Hankel matrix $\mathbf{H}(0)$ and implements the SVD. The SVD results $\{\mathbf{U}_{2n}, \mathbf{S}_{2n}, \mathbf{V}_{2n}\}$ are then transmitted along a path which visits each of the remaining nodes at least once. On the path, when $\{\mathbf{U}_{2n}, \mathbf{S}_{2n}, \mathbf{V}_{2n}\}$ reach one en-route node, they will be updated by incorporating the its Markov parameters using the method proposed in the previous section.

It should be noticed that, once $\{\mathbf{U}_{2n}, \mathbf{S}_{2n}, \mathbf{V}_{2n}\}$ meet a node on the path and are updated by incorporating its Markov parameters, the size of $\{\mathbf{U}_{2n}, \mathbf{S}_{2n}, \mathbf{V}_{2n}\}$ will be increased by $\frac{N}{20}$. Therefore, the path along which the $\{\mathbf{U}_{2n}, \mathbf{S}_{2n}, \mathbf{V}_{2n}\}$ are updated should be carefully designed to minimize the total number of data to be transmitted. Ideally, if exists, the Hamiltonian path as was shown in Fig. 4 should be selected which visits each node exactly once.

If the Hamiltonian path does not exist, we use the solution to the following problem: *Given a connected graph $G(V, E)$, how to find a path which includes each vertex in V at least once and the length of the path is minimized.* We first build a minimum spanning tree (MST), then perform a depth-first tree traversal. The obtained traversal path has 2-approximation to our problem: Assume A be the optimal traversal path for visiting each node at least once, then clearly $|MST| < |A|$. While our depth-first tree traversal has length at most $2|MST| < 2|A|$.

To obtain the modal parameters, we still need another Hankel matrix $\mathbf{H}(1)$. To find $\mathbf{H}(1)$, we first re-construct $\mathbf{H}(0)$ using the final updated $\{\mathbf{U}_{2n}, \mathbf{S}_{2n}, \mathbf{V}_{2n}\}$ by Eq. 9. It can be observed from Eq. 4 that $\mathbf{H}(0)$ already contains all the Markov parameters in $\mathbf{H}(1)$ except one term $\mathbf{Y}(\beta\alpha + 1)$. Therefore, each node in the network then transmits its $Y^i(\beta\alpha+1)$ to the end point of the path and it will identify modal parameters of the structure accordingly. Considering the number of transmission associated with $\mathbf{Y}(\beta\alpha + 1)$ is small, they are omitted in the remaining of the paper.

D. Transmitting the SVD Along the Path of MCDS Instead of the Path of All Nodes

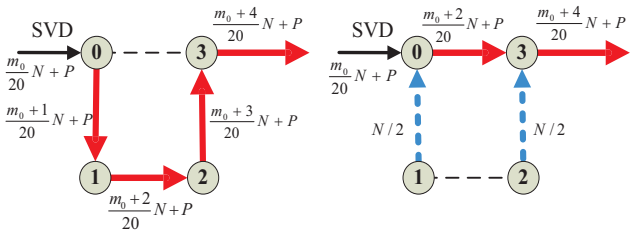
In this section, we first show some drawbacks of the approach proposed in the previous section. In an ideal condition where the Hamiltonian path exists in a network consisting m sensor nodes, the total number of data to be transmitted is

$$E_T = \frac{3}{2}N + \sum_{k=4}^{m-1}(\frac{N}{20}k + P) = \frac{N}{40}m^2 + (P - \frac{N}{40})m + \frac{6}{5}N - 4P \tag{32}$$

where the first and second terms at the left side of the equation correspond to the number of transmissions when the initial Hankel matrix is constructed and when $\{\mathbf{U}_{2n}, \mathbf{S}_{2n}, \mathbf{V}_{2n}\}$ travel along the Hamiltonian path, respectively.

It can be observed that the overall number of wireless transmissions will be quadratically increased with the number of nodes. This shows that updating the SVD along a long path can incur large number of wireless transmissions. Another drawback of this scheme is that updating the SVD is *sequential* and a node on the route must wait for the SVD results from its previous node. Correspondingly, the delay can be larger than the centralized approach where the sink node can immediately start updating once receives info from a few nodes.

If we can find a shorter path than the Hamiltonian to update the SVD, the required wireless transmissions may be decreased. We use an example to show that this 'shorter' path does exist. Assume node 0 shown in Fig. 5a receives the SVD result of m_0 nodes, with the corresponding data amount to be $\frac{m_0 N}{20} + P$ according to Eq. 31. To include the Markov parameters from nodes $0, 1, 2, 3$, we can either update the SVD along the Hamiltonian path shown in Fig. 5a with the total of data transfer to be



(a) (b)

Fig. 5: The amount of data transmitted when (a) the SVD are updated along the Hamiltonian path, and when (b) the Markov parameters of nodes $1, 2$ are transmitted to $0, 3$, respectively.

$$E_{T1} = \sum_{k=1}^{4}(\frac{(m_0+k)N}{20} + P) \tag{33}$$

or we find a direct path (i.e. $0 \rightarrow 3$) shown in Fig. 5b to update SVD while nodes 1 and 2 will transmit their Markov parameters to the corresponding nodes on the path. The total amount of data in the latter scheme, denoted as E_{T2}, is

$$E_{T2} = 2 \cdot \frac{N}{2} + \sum_{k=2,4}(\frac{(m_0+k)N}{20} + P) \tag{34}$$

It can be easily proved that $E_{T2} < E_{T1}$ for any $m_0 > 0$, and as the number of nodes increases, the advantage of this scheme becomes significant.

The above observation also applies to general conditions. Assume a sensor node s^0 has received the updated $\{\mathbf{U}_{2n}, \mathbf{S}_{2n}, \mathbf{V}_{2n}\}$ including all the m_0 en-route nodes so far. Also assume there are still m_r sensor nodes, including s^0, are to be included. If the SVD is updated through the Hamiltonian path among these m_r sensor nodes, the total number of data to be transmitted, starting from s^0, will be

$$E_{T1} = \sum_{k=1}^{m_r-1}[\frac{(m_0+k)N}{20} + P] \tag{35}$$

On the other hand, we can use another scheme to update the SVD. First, we construct a path which satisfies that condition that all the m_r sensor nodes are either on the path or have neighbors on the path. For convenience, we call the nodes on the path the 'backbone' nodes and the rest the 'leaf' nodes. After the path is established, the leaf nodes will transmit their data to their corresponding backbone neighbors. As before, $\{\mathbf{U}_{2n}, \mathbf{S}_{2n}, \mathbf{V}_{2n}\}$ are updated when they travel along the path but when they meet one backbone node, they will incorporate the Markov parameters of both this node and all of its leaf nodes. The amount of data transmitted in this scheme will be:

$$E_{T2} = \frac{N}{2}(m_r - \bar{m}_r) + \sum_{i=1,2,.}^{\bar{m}_r-1}[\frac{(m_0+k_i)N}{20} + P] \tag{36}$$

where $\bar{m}_r$ is the number of backbone nodes ($\bar{m}_r \leq m_r$), $k_i (i = 1, ..., \bar{m}_r - 1)$ is total number of nodes whose Markov parameters have been incorporated when updating the SVD at the i^{th} backbone node. $k_1, k_2, ..., k_{\bar{m}_r-1}$ are selected from $\{0, 1, ..., m_r - 1\}$ and they satisfy $k_1 < k_2 <, ..., < k_{\bar{m}_r-1}$.

Moreover, comparing the total data to be transmitted in these schemes, we have:

$$E_{T1} - E_{T2} =$$
$$= (m_r - \bar{m}_r)(\frac{m_0 N}{20} + P - \frac{N}{2}) + \frac{N}{20}\sum_{k=1\sim m_r-1}^{k\neq k_1,...,k_{\bar{m}_r-1}} k \tag{37}$$

The first term in the right side of Eq. 37 is always positive even when the computational cost is not considered (in this condition, $P = 1100 > N/2 = 1024$). Therefore, we have $E_{T1} > E_{T2}$. Also can be seen is that the difference between E_{T1} and E_{T2} becomes significant as m_0 increases.

In this new scheme, given the topology of a WSN, there may exist infinite number of paths which satisfy the requirement that 'all the nodes are either on this path or have neighbors on the path'. It can be proved that statistically, the longer path will have more wireless communications during the update of SVD, but the detailed proof is omitted for brevity.

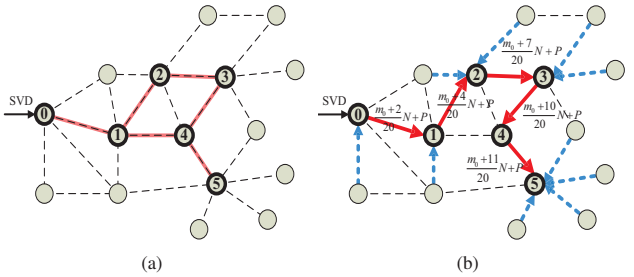
The task then becomes how to find out the shortest path which satisfies the requirement. It can be found that this requirement is very similar to the minimum connected dominating set (MCDS) problem. Given a graph $G(V, E)$, nodes in its MCDS are connected and every node in V either belongs to its MCDS or is adjacent to a nodes in MCDS. Moreover, the number of nodes in the MCDS is kept minimum.

Therefore, we can use the MCDS to find out the path. Initially, a sensor s^0 has the SVD of a Hankel matrix including a few sensor nodes. Based on the network topology of all the remaining nodes, including s^0, the MCDS is found. Then a path which includes all the nodes in the MCDS is found. If exists, the Hamiltonian path should be chosen. Otherwise, we construct a MST of the MCDS, and then perform a depth-first tree traversal to find the the path. It should be noted that if a leaf node has more than one backbone neighbors in the MCDS, it should choose the one nearest to path end to forward its data. Since the size of the SVD will accumulate every time a sensor node is incorporated, delay the incorporation of a sensor node will always be preferable in terms of the wireless transmissions. This approach is illustrated in Fig. 6, where we assume node 0 already has an initial SVD. The rationale behind is that with the help of the MCDS, *the SVD which previously must visit all the nodes in the whole network now only need to visit nodes in the MCDS. Correspondingly, the path is significantly shortened and the number of data to be transmitted is therefore decreased.*

V. Performance Evaluation

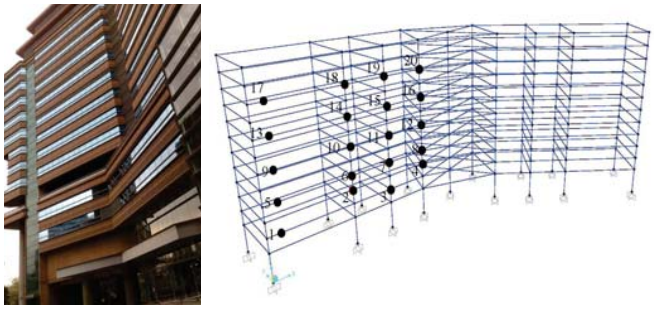
A. Simulation

First, we use simulation data to demonstrate the advantage of distributed ERA in terms of wireless communication. We



(a) (b)

Fig. 6: (a) The network topology and the graph of MCDS (b) Updating SVD along one Hamiltonian path in the MCDS



(a) (b)

Fig. 9: The LSK building and measurement locations (a) The LSK building (b) 20 measurement locations

consider three different schemes of implementing the ERA: (1) the traditional centralized ERA where all the raw data are transmitted to the sink through the SPT (2) the modification described in section IV-A where the Markov parameters are transmitted to the sink and (3) our proposed approach where the SVD is updated along the path in the MCDS.

Two scenarios are created to evaluate their performance in different network density and network size. Assume m sensor nodes with the same communication range R are randomly deployed on an square area. We first fix the number of nodes $m = 100$ but gradually decrease the network density by decreasing R from 40 to 10. The area in this scenario is fixed to be 100×100. Then we maintain the network density at a certain level but gradually increase the network size by increasing m from 100 to 500. Note that to maintain the network density, when increasing m, the size of the deployment area also needs to be increased. In both scenarios, a total of 500 simulations are performed and the average value of each scheme is calculated.

The results of the first scenario is illustrated in Fig. 7. For a more clear comparison, the amount of data transmitted in these three schemes are compared in two sub-figures in which Fig. 7a and 7b compare the total data transmitted in the first two schemes and the last two schemes, respectively. First, it is easy to understand that the amount of data transmitted for ERA is increasing with the decrease of R. (Note that these two figures show reverse x-axis). Fig. 7a clearly shows that broadcasting the reference data and then transmitting the calculated Markov parameters saves significant wireless transmissions compared with the centralized one. While Fig. 7b illustrates that compared with transmitting the Markov parameters, using the MCDS to update the SVD can further reduce wireless transmissions. Moreover, we can see from Fig. 7b that the more sparse the network, the advantage of this scheme in wireless communication becomes more significant.

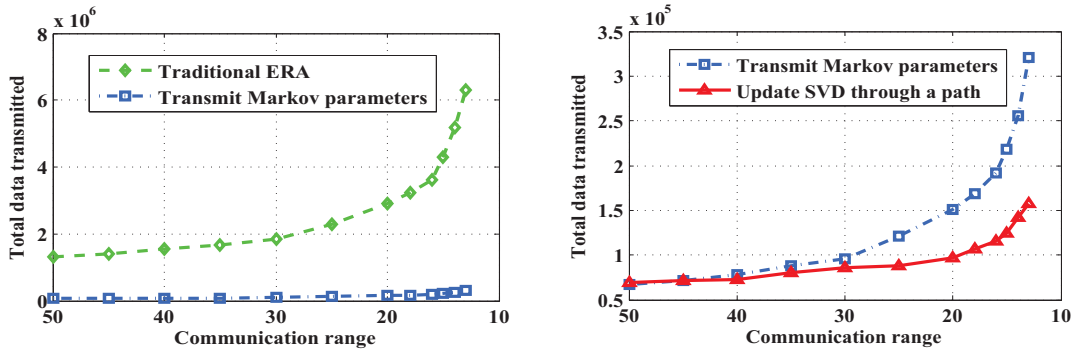
Likewise, Fig. 8 compares the data to be transmitted in different schemes under various network size. It can be seen that the data to be transmitted in the last scheme is smaller than the other two, and more number of sensor nodes the network, this advantage becomes more significant.

An important property than be seen From Fig. 7b and Fig. 8b is that the advantage of our proposed method becomes more significant in a sparse network with large number of sensor nodes. This property is very favorable for SHM since this matches the real condition when wireless sensor nodes are deployed to monitor the condition of large civil infrastructures.

B. Experiment

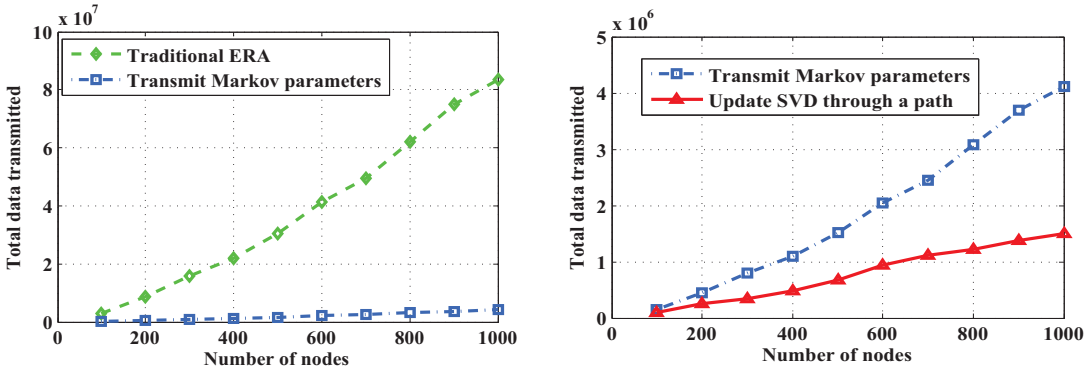
We have tested our proposed scheme through a real experiment. We deployed a number of SHM motes in the LSK Building to measure its vibration under ambient environment(see Fig. 9a). The numbering of the measured locations is shown in Fig 9b. The system setup on measurement location 17 is illustrated in Fig. 10a. At each measurement location, we use a high-sensitive external sensor named KD1300 to record the vibration (see Fig.10a) in a selected direction. Signal recorded at KD1300 is amplified and then fed into a SHM mote. For convenience, we call the SHM mote connecting to the KD1300 the sampling mote. In this experiment, wireless communication is not directly established among the sampling motes at different locations since they belong to different rooms and are not able to directly communicate well. To solve this problem, we deploy a particular SHM mote acting as local data collector near the window of each location(see the top figure of Fig. 10b). Vibrational data sampled at the sampling mote are transmitted to this collector first. These 20 collector nodes can be regarded as independent wireless sensor nodes in this WSN. In this paper, we assume the collector nodes already have the local vibration data and only consider the wireless communication among them.

During the test, we first find the network topology of the collector nodes by running the CTP on them. The topology information is then transmitted to a gateway node which is connected to a laptop computer(see the bottom figure of Fig. 10b). The laptop computer calculates the MCDS and the corresponding path. The results are then broadcast to all the collector nodes. From the received results, each collector node is aware of its status and the associated information. For a leaf node, the information contains its parent in the MCDS; for a backbone node in the MCDS, the information has all of its leaf node children, its backbone children and parent. These



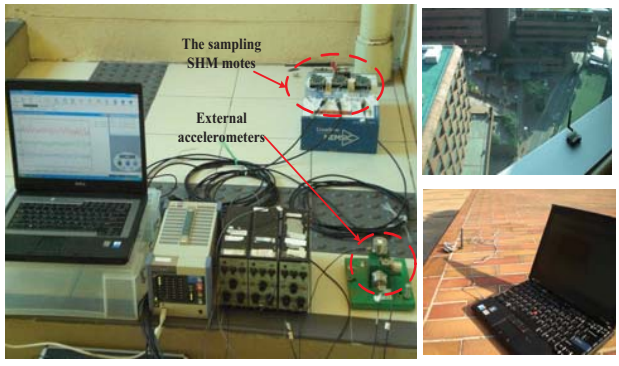
(a) (b)

Fig. 7: The required data transfer under various network density (a) the traditional ERA v.s. transmitting Markov parameters (b)Transmitting Markov parameters v.s. updating the ERA along a path in the MCDS



(a) (b)

Fig. 8: The required data transfer under various network size (a) the traditional ERA v.s. transmitting Markov parameters (b)Transmitting Markov parameters v.s. updating the ERA along a path in the MCDS



(a) (b)

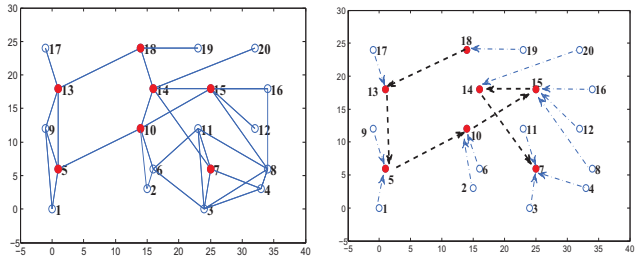
Fig. 10: Experiment setup (a) Sampling motes and sensors at a certain location(b)Top: a collector node deployed near the window; Bottom: a gateway node connected to a laptop

information will be used later when the node will decide its action. The network topology of these 20 collector nodes and the MCDS are illustrated in Fig. 11a.

All the deployed sampling motes are then synchronized using the modified flooding time synchronization protocol (FTSP) [19]. At a certain arranged global time point, all the sampling motes start sensing with sampling rate of 1024Hz. The sampling procedure lasts for 25 seconds.

After each collector node has obtained the sampled data from its corresponding sampling mote, the distributed ERA is implemented, in which the network center No. 11, is taken as the root as well as the reference node which broadcasts the raw data to the network. The updating path is shown in 11b. The distributed ERA stops when the root receives data from all its children and obtains the modal parameters. The root transmits these parameters, also through the CTP routing protocol to the gateway node.

For comparison, the Markov parameters of each sensor node are also transmitted to the gateway using the CTP. According to the topology recorded in the CTP, the total number of data



(a) (b)

Fig. 11: (a) The network topology of collector nodes and the MCDS in the network (b) the path along which the ERA is updated (arrows with black dashed lines connect backbone nodes while arrows with blue dashed-dot lines connect leaf nodes to backbone nodes).

	Our distributed ERA			Cluster-based ERA		
	Natural Frequency		Mode Shapes	Natural Frequency		Mode Shapes
Mode	f(Hz)	Diff. (10^{-4})	1-MAC (10^{-6})	f(Hz)	Diff. (10^{-2})	1-MAC (10^{-2})
1	0.7536	0.3453	1.3325	0.7512	0.32	2.623
2	0.9826	1.2438	1.7622	1.0130	3.09	0.3432
3	1.3578	0.4987	1.8991	1.3381	1.45	2.289
4	6.7283	0.7654	2.4766	6.9223	2.88	1.230
5	15.8914	1.9824	2.3987	16.10	1.31	3.343

TABLE I: The identified modal parameters and the difference with the centralized method

transmitted in the network is about $1e^5$, and in the test, it took approximately 30 minutes to finish the transmission. While on the other hand, the total data transmitted using our method is about $2.6e^4$, and it took about 25 minutes for the gateway to receive the calculated modal parameters. It should be noted that the computation time is also included. The advantage of this revised ERA is thus quite obvious.

More importantly, we will show that this modified ERA is able to give extremely accurate modal parameters as was in the traditional centralized method. Natural frequencies, and mode shapes are compared with those calculated on the laptop computer connected with the gateway using the received Markov parameters. The comparison results are shown in Table I. The error in the frequency is calculated as the difference between the estimates on the SHM Mote and the PC. The estimation error in the mode shapes are investigated in terms of the Modal Assurance Criterion (MAC)[20]. As shown in Table I, the modal parameters in the SHM Mote and those on the PC are almost identical. Also, for comparison purpose, the modal parameters identified from the clustering-based ERA proposed in [8] also shown in Table I. Our method can achieve much higher accuracy than the cluster-based method.

VI. Conclusion

In this paper, we select a classical SHM algorithm: the eigen-system realization algorithm (ERA), and propose a distributed version for WSNs. Compared with the traditional ERA which is computationally intensive and requires raw data from all the sensor nodes, the computations in the ERA is distributed this distributed version. This distributed version is able to achieve the same quality of the original ERA using much smaller wireless transmissions and computations. Through simulation and experiment, the effectiveness and efficiency of this distributed ERA in terms of required wireless transmissions, computations and accuracy are demonstrated.

VII. Acknowledgment

This work is supported by the HKRGC under GRF grant PolyU5106/11E, HK PolyU Niche Area Fund 1-BB6C, and NSF-CNS-1066391.

References

[1] C. Farrar and S. Doebling, "An overview of modal-based damage identification methods," in *Proceedings of DAMAS Conference*, 1997.

[2] S. Doebling, "Damage identification and health monitoring of structural and mechanical systems from changes in their vibration characteristics: a literature review," Los Alamos National Lab., Tech. Rep., 1996.

[3] J. Paek and Chintalapudi, "A wireless sensor network for structural health monitoring: Performance and experience," in *EmNetS-II*, 2005, pp. 30–31.

[4] J. Lynch, A. Sundararajan, and et al., "Embedding damage detection algorithms in a wireless sensing unit for operational power efficiency," *Smart Materials and Structures*, vol. 13, p. 800, 2004.

[5] G. Hackmann, F. Sun, N. Castaneda, C. Lu, and S. Dyke, "A holistic approach to decentralized structural damage localization using wireless sensor networks," in *Real-Time Systems Symposium*, 2008, pp. 35–46.

[6] G. Yan, W. Guo, S. Dyke, G. Hackmann, and C. Lu, "Experimental validation of a multi-level damage localization technique with distributed computation," *Smart Structures and Systems*, vol. 6, no. 5-6, pp. 561–578, 2010.

[7] A. Zimmerman and M. Shiraishi, "Automated modal parameter estimation by parallel processing within wireless monitoring systems," *Journal of Infrastructure Systems*, vol. 14, p. 102, 2008.

[8] X.Liu, J. Cao, and et al., "Energy efficient clustering for wsn-based structural health monitoring," in *IEEE INFOCOM*, vol. 2, 2011, pp. 1028–1037.

[9] J. Juang and R. Pappa, "Eigensystem realization algorithm for modal parameter identification and model reduction." *Journal of Guidance, Control, and Dynamics*, vol. 8, no. 5, pp. 620–627, 1985.

[10] R. Brincker, L. Zhang, and P. Andersen, "Modal identification from ambient responses using frequency domain decomposition," in *Proceedings of the 18th international modal analysis conference*, 2000, pp. 625–630.

[11] T. Fu, A. Ghosh, E. Johnson, and B. Krishnamachari, "Energy-efficient deployment strategies in structural health monitoring using wireless sensor networks," *Structural Control and Health Monitoring*, 2011.

[12] Y. Gao, B. Spencer Jr, and M. Ruiz-Sandoval, "Distributed computing strategy for structural health monitoring," *Structural control and health monitoring*, vol. 13, no. 1, pp. 488–507, 2006.

[13] F. Harris, "On the use of windows for harmonic analysis with the discrete fourier transform," *Proc. of the IEEE*, vol. 66, no. 1, pp. 51–83, 1978.

[14] D. Inman, "Engineering vibrations," 2006.

[15] T. Nagayama and B. Spencer Jr, "Structural health monitoring using smart sensors," *N.S.E.L. Report Series 001*, 2008.

[16] G. Golub and C. Van Loan, "Matrix computations (johns hopkins studies in mathematical sciences)," 1996.

[17] M. Berry, S. Dumais, and G. O'Brien, "Using linear algebra for intelligent information retrieval," *SIAM review*, pp. 573–595, 1995.

[18] H. Zha and H. Simon, "On updating problems in latent semantic indexing," *SIAM Journal on Scientific Computing*, vol. 21, p. 782, 1999.

[19] M. Maróti, B. Kusy, G. Simon, and Á. Lédeczi, "The flooding time synchronization protocol," in *Proc. of the 2nd international conference on ENSS*, 2004, pp. 39–49.

[20] R. Allemang and D. Brown, "A correlation coefficient for modal vector analysis," in *Proceedings of the 1st international modal analysis conference*, vol. 1, 1982, pp. 110–116.