

Distributed Decision-Making in Wireless Sensor Networks for Online Structural Health Monitoring

Qing Ling, Zhi Tian, and Yue Li

Abstract: In a wireless sensor network (WSN) setting, this paper presents a distributed decision-making framework and illustrates its application in an online structural health monitoring (SHM) system. The objective is to recover a damage severity vector, which identifies, localizes, and quantifies damages in a structure, via distributive and collaborative decision-making among wireless sensors. Observing the fact that damages are generally scarce in a structure, this paper develops a nonlinear 0-norm minimization formulation to recover the sparse damage severity vector, then relaxes it to a linear and distributively tractable one. An optimal algorithm based on the alternating direction method of multipliers (ADMM) and a heuristic distributed linear programming (DLP) algorithm are proposed to estimate the damage severity vector distributively. By limiting sensors to exchange information among neighboring sensors, the distributed decision-making algorithms reduce communication costs, thus alleviate the channel interference and prolong the network lifetime. Simulation results in monitoring a steel frame structure prove the effectiveness of the proposed algorithms.

Index Terms: Distributed decision-making, structural health monitoring (SHM), wireless sensor networks (WSNs).

I. INTRODUCTION

Recent development in micro-fabrication and wireless communication technologies enables the application of wireless sensor networks (WSNs), which consist of spatially distributed sensors with embedded sensing, computation, and wireless communication capabilities [1]. Accompanied with the unprecedented data collection opportunities, new challenges also emerge due to two main constraints in network resources: Communication bandwidth and battery power. In a large-scale centralized network, extensive communication between wireless sensors and the fusion center results in strong interference, as well as constitutes the main source of energy consumption.

To meet the rigid bandwidth and energy constraints, distributed in-network processing has attracted interest in a wide range of areas, such as classification [2], estimation [3], consensus [4], and learning [5]. Without using any fusion center, decisions are made in a collaborative manner via local information exchange among one-hop neighboring sensors.

This paper considers the application of distributed decision-making in an online structural health monitoring (SHM) system. SHM refers to the process of damage detection for civil,

aerospace and mechanical engineering systems [6]. Here the damage is defined as changes to the material or geometric properties of these systems due to either internal factors such as aging, or exterior forces such as natural disasters. Through acquisition and interpretation of critical structural response data, an online SHM system periodically assesses the health condition of a structure at three main levels: 1) Identification of anomalies and damages in a structure, 2) localization of damage, and 3) quantification of damage severity [7].

Contrary to previous work which needs frequent and extensive multi-hop data exchange between sensors and the fusion center [8], [9], the proposed online SHM system emphasizes distributed decision-making at limited communication costs. By applying the auto-regressive and auto-regressive with exogenous inputs (AR-ARX) method as the embedded damage detection approach [10], sensors independently calculate corresponding damage-sensitive coefficients in each monitoring period. A vector of damage severity coefficients, which identifies, localizes, and quantifies damages, is distributively solved via information exchange among neighboring sensors. Such distributed processing is repeated in every sampling period for online monitoring during normal operations. Upon detecting damages, sensors can either trigger alarms autonomously or transmit the refined damage information to a central console to help repair the structure.

The main challenges in designing the online SHM system are twofold: 1) *How can we formulate the problem as a distributively tractable one?* 2) *how can we solve the problem in an energy-efficient way?* This paper contributes on both aspects, including problem formulation and algorithm design:

- 1) Taking an optimization approach, we propose a series of optimization formulations for estimating damage over a large field. Considering damage as a rare phenomenon in a structure, the damage severity vector only has a small number of non-zero elements. Recognition of this important sparsity property allows us to formulate a 0-norm minimization problem (**P0**), which recovers the sparse damage severity vector from the distributed damage-sensitive coefficients. Motivated by recent advances in compressed sampling and sparse signal recovery [11], we relax the nonlinear 0-norm minimization formulation to a linear 1-norm minimization problem (**P1**) for computational tractability. Further, to enable distributed algorithm design in the presence of the coupling effect among the decision variables of all sensors, we introduce an approximated formulation (**Pa**) which decouples sensors that are not neighbors to each other.
- 2) We derive efficient iterative algorithms to recover the damage severity vector in a distributed manner. The alternating direction method of multipliers (ADMM) is adopted to pro-

Manuscript received December 21, 2008.

Q. Ling and Z. Tian are with the Department of Electrical and Computer Engineering, Michigan Technological University, Houghton, Michigan, 49931, email: {qling, ztian}@mtu.edu.

Y. Li is with the Department of Civil and Environmental Engineering, Michigan Technological University, Houghton, Michigan, 49931, email: yueli@mtu.edu.

vide the optimal solution to **Pa**. However, ADMM generally converges slowly and needs to exchange Lagrangian multipliers among neighboring sensors. To alleviate the communication burden, we further propose a heuristic distributed linear programming (DLP) algorithm, which has faster convergence rate and obviates the need of exchanging extra information besides intermediate decision variables.

The methodology developed here for distributed detection, message passing, and decision-making, also applies to other monitoring applications of energy-constrained large-scale sensor networks where the phenomena of interest are sparse.

This paper is organized as follows. Section II surveys related work. The AR-ARX method is described in Section III. Section IV discusses the optimization formulations and proposes two distributed algorithms. Simulation results are provided in Section V to verify the effectiveness of the distributed SHM algorithms. Section VI summarizes the paper.

II. RELATED WORK

In WSNs, there are four categories of infrastructures according to the way of data processing and information transmitting:

- 1) *Centralized infrastructure*: Sensors send back raw measurement data to a fusion center for data processing.
- 2) *Node-level distributed infrastructure*: Sensors send back refined data, which is extracted from the raw data, to a fusion center for data processing.
- 3) *Hierarchical infrastructure*: Sensors are divided into several clusters. Within a cluster, sensors send information to a cluster head for data processing. Cluster heads may exchange information with each other, make decisions collaboratively, and send decisions to a central console.
- 4) *Network-wide distributed infrastructure*: Each sensor exchanges information only with its neighboring sensors, and makes decision autonomously. The final decisions are then send to a central console.

In centralized and node-level distributed infrastructures, sensors need to communicate with the fusion center frequently. However, in a large-scale WSN, one-hop communication between sensors and the fusion center is generally impossible. In the hierarchical infrastructure, each sensor is limited to exchange information with its cluster head, but it is difficult to predefine clusters in a large-scale WSN. Furthermore, failure of a cluster head leads to the malfunction of the whole cluster. Hence to improve the scalability and robustness of the network, it is preferred to apply a network-wide distributed infrastructure, in which each sensor exchanges information among its neighboring sensors and makes decision autonomously.

This paper is among the efforts toward the broad objective of distributed in-network processing [2]–[5]. Though problem formulations are different, the common design principle is to accomplish an otherwise centralized task in a distributive way and to improve the scalability, robustness, and lifetime of a network.

A well-studied distributed in-network processing task is consensus averaging [4], [12]. Sensors dynamically exchange current estimates with one-hop neighbors and update their local es-

timates, until the whole network reaches consensus on an averaged scalar. A more complicated task is to distributively optimize an objective function, which is common in estimation and learning. In [5] and [13], separable objective functions are optimized based on the decentralized incremental sub-gradient approach. An estimation problem is formulated to be with a separable objective function and a set of consensus constraints in [14]. By iteratively updating local estimates, the network reaches a consensus which minimizes the estimation error.

This paper considers the problem of estimating a vector for the distributed network, though each sensor only needs to know its own decision scalar eventually. The problem is formulated as a linear program, in which the objective function is separable and each constraint decouples the relationship among non-neighboring sensors. We adopt primal-dual methods, which have been proved to be powerful tools for distributed inference [15], [16], as optimal distributed solutions. However, the primal-dual methods generally require nontrivial rounds of iterations to reach convergence, thus results in significant communication overhead. Furthermore, sensors need to exchange extra information, such as Lagrangian multipliers, in addition to exchanging intermediate decision variables. By investigating the specific problem structure, this paper also propose a heuristic distributed algorithm, which has faster convergence rate and exchanges fewer information.

Our work on distributed decision-making not only contributes to the general WSN literature as mentioned above, but also represents a new approach to the interdisciplinary problem of SHM using WSNs. In this paper we use the AR-ARX method as the embedded damage identification approach. The AR-ARX method classifies the damaged pattern via comparing the statistics between baseline measurements and current measurements [10]. It is important to note that the proposed distributed online SHM framework can be combined with other damage identification tools. For detailed review of damage identification methods, readers are referred to [7].

Online SHM based on WSNs has emerged in recent years as a promising technique to monitor structural health conditions. Wireless sensors are equipped with sensing units to measure structural responses, communication units to transmit information without using expensive coaxial wires, and computation units to process raw data and make decisions. Extensive survey of SHM based on WSNs can be found in [17].

A hierarchical network infrastructure is considered in [8], in which sensors collect raw data; while cluster heads exchange damage information, make final decisions, and transmit to a central console. In [9], sensors apply the AR-ARX method in a node-level distributed manner and transmit refined data to a fusion center. However, study of online SHM based on the network-wide distributed infrastructure is still in its beginning stage. In this paper, we focus on distributed decision-making for online SHM, aiming at improving scalability and robustness of a WSN under the constraints of bandwidth and energy.

III. AR-ARX METHOD

The AR-ARX method is a statistic pattern recognition approach which is composed of a modeling stage where the struc-

ture is known to be undamaged, and a decision-making stage where the damage state is unknown [10]. The basic idea is to classify damaged patterns via comparing the statistics between baseline measurements and current measurements.

A. AR and ARX Models

Let $z = \{z_k\}$ be the time-series response of a structure at a specific sensor location. Assuming the response to be stationary, an AR process model is used to fit the discrete measurement data:

$$z_k = \sum_{i=1}^p b_i z_{k-i} + r_k^z. \quad (1)$$

The response of the structure at sample time k , as denoted by z_k , is a function of p previous observations of the system response, plus a residual error term r_k^z . Weights on the previous observations of z_{k-i} are denoted by coefficients $\{b_i\}$.

It is assumed that the residual error of the AR model r_k^z is influenced by the unknown excitation. As a result, a second time-series model, an autoregressive with exogenous inputs (ARX) model, is chosen to model the relationship between the residual errors and the measured response of the system:

$$z_k = \sum_{i=1}^a \alpha_i z_{k-i} + \sum_{j=0}^b \beta_j r_{k-j}^z + \epsilon_k^z. \quad (2)$$

Coefficients on past measurements and the residual errors of the AR model are $\{\alpha_i\}$ and $\{\beta_i\}$, respectively. The residual of the ARX model ϵ_k^z is a damage sensitive feature being used to identify the existence of damage in the structure.

In the modeling stage, i.e., the structure is known to be undamaged, the AR and ARX models are constructed under various ambient vibration levels. The coefficients of models, i.e., $\{b_i\}$, $\{\alpha_i\}$, and $\{\beta_i\}$, and the standard variances of the residuals, i.e., $\{\sigma^2(\epsilon_k^z)\}$, are stored in the database of each sensor, denoted as $\{b_i^{DB}\}$, $\{\alpha_i^{DB}\}$, $\{\beta_i^{DB}\}$, and $\{\sigma^2(\epsilon_k^{DB})\}$.

B. Statistical Pattern Recognition

In the decision-making stage, an AR model is fitted based on the response z_k of the structure in an unknown state (damaged or undamaged). The coefficients of the fitted AR model are compared to the database of AR-ARX model pairs previously calculated for the undamaged structure. A match is determined by minimizing the Euclidian distance D of the newly derived AR model and the database AR model coefficients, b_i^z and b_i^{DB} , respectively. The Euclidian distance D is defined as:

$$D = \sum_{i=1}^p (b_i^{DB} - b_i^z)^2. \quad (3)$$

If no structural damage is experienced and the operational conditions of the two models are close to one another, the selected AR model from the database will closely approximate the measured response. If a damage has been sustained by the structure, even the closest AR model of the database will not approximate the measured structural response well.

The measured response z_k of the structure in the unknown state, and the residual errors r_k^z of the fitted AR model, are substituted into the database ARX model to determine the residual error ϵ_k^z of the ARX model:

$$z_k = \sum_{i=1}^a \alpha_i^{DB} z_{k-i} + \sum_{j=0}^b \beta_j^{DB} r_{k-j}^z + \epsilon_k^z. \quad (4)$$

The residual errors $\{\epsilon_k^z\}$ of the ARX model are the damage sensitive feature.

Here we briefly discuss the statistics of the residual errors. Define the ratio of the variance of the residual errors to that in the database as:

$$y = \frac{\sigma^2(\epsilon_k^z)}{\sigma^2(\epsilon_k^{DB})}. \quad (5)$$

Here $\sigma^2(\epsilon_k^z)$ and $\sigma^2(\epsilon_k^{DB})$ are variances of the residual errors. Suppose the models are accurate and the ambient excitations are Gaussian random variables with zero means. Hence the residual errors are also Gaussian variables with zero means. If the structure is undamaged, i.e., the system models remain the same, and the noise level keeps invariant, then y follows F-distribution, and the degree-of-freedom is equal to the length of the measured data. If the structure is damaged, i.e., coefficients of the system models change, variance of the residual errors increases because of model mismatch. From this angle of view, y is a *damage-sensitive coefficient*.

C. Damage-Sensitive Coefficient and Severity Coefficient

Assume a large-scale WSN is deployed in a sensing area. In the modeling stage, each sensor builds up a database for the measured response signals. In the decision-making stage, sensors periodically perform monitoring tasks, firstly collecting batches of data and secondly generating damage-sensitive coefficients. Based on the damage-sensitive coefficients, the damages are identified, localized, and quantified.

It has been proved in [18] that a damage-sensitive coefficient of one sensor is composed of accumulated damage influences of all monitoring points and a random noise. Consider sensor i which has a damage-sensitive coefficient y_i as defined in (5). The coefficient y_i follows F-distribution if the structure is undamaged. Otherwise a damage-related term is appended. Therefore, y_i is the summation of a damage-related term s_i and a random variable e_i with F-distribution:

$$y_i = s_i + e_i. \quad (6)$$

Here s_i is decided by the cumulative effects of the damaged points. The random variable e_i is under F-distribution with degree-of-freedom (n, n) , where n is nearly the number of sampling points. When n is large, the probability density function of e_i is almost symmetric with respect to $e_i = 1$.

Suppose the large network is densely deployed such that one damaged point is collocated with one sensor point. Therefore, letting \mathcal{L} be the set of L sensors, without loss of generality, s_i can be written as:

$$s_i = \sum_{j \in \mathcal{L}} c_j f_{ji}. \quad (7)$$

Here c_j is positive and represents the severity of the damage point collocated with sensor j , and is defined as *severity coefficient* of point j . The normalized basis function f_{ji} represents the effect of damage in point j on point i . The model in (7) is general enough to describe a wide range of structural damage behavior, provided that the parameters $\{f_{ij}\}$ can be acquired via, say, statistical learning and/or domain knowledge. Since the focus of this paper is on information processing using sensor networks, we investigate regular-shaped structures as an illustrative example for clear exposition on the algorithm design. In this case, one critical observation based on simulation validation is that, the influence of one damage decreases as the distance from the damage increases. Generally speaking, this kind of influence can be described by an *isotropic Gaussian-shaped* basis function:

$$f_{ji} = e^{-d_{ji}^2/\sigma_j^2}. \quad (8)$$

Here d_{ji} is the distance between j and i and σ_j is the known impact coefficient of j . The isotropic Gaussian-shaped basis function assumes isotropic influence of a damage point, which is adopted for analytical convenience but still able to approximate most practical cases in regular-shaped structures.

IV. DISTRIBUTED SHM ALGORITHMS

In this section, we formulate the SHM task as a nonlinear 0-norm minimization problem and then relax it to a 1-norm linear program. The centralized formulation is further slightly modified to make distributed processing possible. An optimal algorithm based on the ADMM and a heuristic DLP algorithm are proposed to estimate damage severity coefficients in a distributed way.

A. Problem Formulation

Combining (6)–(8), the objective of the SHM task is to recover $\{c_i\}$ from noise-polluted measurements $\{y_i\}$ and known $\{d_{ji}\}$ and $\{\sigma_i\}$. Note that for two neighboring sensors i and j , d_{ji} can be obtained in initializing the network. Therefore the basis function value f_{ji} is known by (8) in advance. Our novel idea hinges on a key observation that damages are generally scarce in a structure from an engineering perspective. Hence, the severity vector $\mathbf{c}=[c_1, \dots, c_L]^T$, which is a concatenation of the severity coefficients, is sparse. Sparsity of \mathbf{c} can be measured by 0-norm $\|\mathbf{c}\|_0$ which counts the number of nonzeros of \mathbf{c} . Capitalizing on the sparse nature of \mathbf{c} , we propose a 0-norm minimization formulation **P0**:

$$\begin{aligned} \min \quad & \|\mathbf{c}\|_0 \\ \text{s.t.} \quad & |y_i - \sum_{j \in \mathcal{L}} f_{ji}c_j - 1| \leq \theta, \forall i \in \mathcal{L}, \\ & c_i \geq 0, \forall i \in \mathcal{L}. \end{aligned} \quad (9)$$

We predefine a common threshold θ and impose a constraint $|e_i - 1| = |y_i - \sum_{j \in \mathcal{L}} f_{ji}c_j - 1| \leq \theta$ on each i since e_i is with mean value 1. Therefore, the resulting random noises are constrained to an interval. The role of the linear constraints is similar as that of the commonly used quadratic constraints in dealing with random noises. However, the linear constraint terms may

lead to a simple heuristic distributed linear programming solution as we will discuss in this section.

The problem **P0** has linear constraints and a nonlinear objective function. Motivated by the recent progress on compressive sensing [11], we relax it to a 1-norm minimization formulation, denoted as **P1**:

$$\begin{aligned} \min \quad & \|\mathbf{c}\|_1 \\ \text{s.t.} \quad & |y_i - \sum_{j \in \mathcal{L}} f_{ji}c_j - 1| \leq \theta, \forall i \in \mathcal{L}, \\ & c_i \geq 0, \forall i \in \mathcal{L}. \end{aligned} \quad (10)$$

Here **P1** is a standard linear program, which can be easily solved in a centralized way. As demonstrated extensively in the compressive sensing literature [11], **P0** and **P1** may have the same solution when \mathbf{c} is sparse enough, and the matrix composed of the coefficients $\{f_{ji}\}$ is well conditioned.

However, to satisfy the constraint $|y_i - \sum_{j \in \mathcal{L}} f_{ji}c_j - 1| \leq \theta$ in **P1** for any j , all decision variables $\{c_j, j \in \mathcal{L}\}$ should be known to i . To make distributed decision-making possible, we further introduce an approximated formulation **Pa** as follows:

$$\begin{aligned} \min \quad & \|\mathbf{c}\|_1 \\ \text{s.t.} \quad & |y_i - f_{ii}c_i - \sum_{j \in \mathcal{N}_i} f_{ji}c_j - 1| \leq \theta, \forall i \in \mathcal{L}, \\ & c_i \geq 0, \forall i \in \mathcal{L}. \end{aligned} \quad (11)$$

Here \mathcal{N}_i denotes the set of one-hop neighbors of sensor i . Note that the basis function is normalized such that $f_{ii} = 1$. Furthermore, the transmission ranges of sensors are homogeneous, such that if i is a one-hop neighbor of j then j is also a one-hop neighbor of i . In this way, we are able to limit the information exchange to among one-hop neighbors, as illustrated in the following distributed algorithms. Intuitively, the performance gap between **P1** and **Pa** is small if given i , f_{ji} is small for any $j \notin \mathcal{N}_i$ and $j \neq i$. Fortunately this is generally the case in practice. Considering the isotropic Gaussian-shaped basis function, if d_{ji} is large enough such that j and i are not one-hop neighbor of each other, then f_{ji} is small.

B. Alternating Direction Method of Multipliers

Let us rewrite (11) using slack variables $\{s_{1i}\}$ and $\{s_{2i}\}$ to form the following equivalent linear program:

$$\begin{aligned} \min \quad & \sum_{i=1}^L c_i \\ \text{s.t.} \quad & f_{ii}c_i + \sum_{j \in \mathcal{N}_i} f_{ji}c_j - s_{1i} + \theta + 1 - y_i = 0, \forall i \in \mathcal{L}, \\ & f_{ii}c_i + \sum_{j \in \mathcal{N}_i} f_{ji}c_j + s_{2i} - \theta + 1 - y_i = 0, \forall i \in \mathcal{L}, \\ & c_i \geq 0, s_{1i} \geq 0, s_{2i} \geq 0, \forall i \in \mathcal{L}. \end{aligned} \quad (12)$$

Equation (12) has an optimal iterative solution based on the ADMM [15]. For any $i \in \mathcal{L}$, slack variables s_{1i} and s_{2i} , Lagrangian multipliers γ_{1i} and γ_{2i} , λ_{1i} and λ_{2i} , and decision vari-

able c_i are updated for time $t + 1$ as follows:

$$\begin{aligned} s_{1i}(t+1) &= \frac{m_i}{d} [\lambda_{1i}(t)]^+, \\ s_{2i}(t+1) &= \frac{m_i}{d} [\lambda_{2i}(t)]^-, \end{aligned} \quad (13)$$

$$\begin{aligned} \gamma_{1i}(t+1) &= \frac{1}{m_i} \left(\sum_{j \in \mathcal{N}_i \cup i} f_{ji} c_j(t) - (y_i - 1 - \theta) \right), \\ \gamma_{2i}(t+1) &= \frac{1}{m_i} \left(\sum_{j \in \mathcal{N}_i \cup i} f_{ji} c_j(t) - (y_i - 1 + \theta) \right), \end{aligned} \quad (14)$$

$$\begin{aligned} \lambda_{1i}(t+1) &= \lambda_{1i}(t) + \frac{d}{m_i} (\gamma_{1i}(t+1) - s_{1i}(t+1)), \\ \lambda_{2i}(t+1) &= \lambda_{2i}(t) + \frac{d}{m_i} (\gamma_{2i}(t+1) - s_{2i}(t+1)), \end{aligned} \quad (15)$$

$$\begin{aligned} g_i(t+1) &= \sum_{j \in \mathcal{N}_i \cup i} (-df_{ij} e_{1j}(t+1) - f_{ij} \lambda_{1j}(t+1) \\ &\quad - df_{ij} e_{2j}(t+1) - f_{ij} \lambda_{2j}(t+1) + 2df_{ij}^2 c_i(t)) - 1, \end{aligned} \quad (16)$$

$$\begin{aligned} h_i(t+1) &= \sum_{j \in \mathcal{N}_i \cup i} df_{ij}^2, \\ c_i(t+1) &= [g_i(t+1)/h_i(t+1)]^+. \end{aligned} \quad (17)$$

Here $[\cdot]^+$ denotes the projection to $\max\{\cdot, 0\}$; $[\cdot]^-$ denotes the projection to $\min\{\cdot, 0\}$; m_i denotes the number of neighbors of sensor i plus 1, namely, the number of sensors which are inside the communication range of sensor i ; d is a constant positive coefficient.

According to (13)–(16), each sensor only need to know the decision variables and Lagrangian multipliers of its one-hop neighboring sensors to update its own slack variables, Lagrangian multipliers, and decision variable. In summary, we have the following distributed ADMM solution to \mathbf{Pa} :

Algorithm 1: ADMM

- Step 1: Each sensor i holds a predefined common threshold θ , a predefined constant d , and a calculated damage-sensitive coefficient y_i . Then sensor i calculates $\{f_{ji}, j \in \mathcal{N}_i\}$ by estimating its distances from neighboring sensors. Sensor i also calculates m_i by counting the number of its one-hop neighbors. The initial decision variable, slack variables, and Lagrangian multipliers are all set as 0.
- Step 2: In iteration $t + 1$, each sensor i broadcasts its current decision variable $c_i(t)$, Lagrangian multipliers $\lambda_{1i}(t)$ and $\lambda_{2i}(t)$, $\gamma_{1i}(t)$, and $\gamma_{2i}(t)$ to its one neighboring sensors $j \in \mathcal{N}_i$.
- Step 3: Each sensor i updates its slack variables $s_{1i}(t+1)$ and $s_{2i}(t+1)$ according to (13), its multipliers $\lambda_{1i}(t+1)$ and $\lambda_{2i}(t+1)$ according to (14), its multipliers $\gamma_{1i}(t+1)$ and $\gamma_{2i}(t+1)$ according to (15), and its decision variable $c_i(t+1)$ according to (16).
- Step 4: Repeat Step 2 and Step 3 until reaching convergence.
-

C. Distributed Linear Programming

ADMM is an optimal solution to \mathbf{Pa} . However, the communication load of ADMM may be high. Firstly, the convergence rate

of ADMM is generally slow; thus a large amount of information exchange is needed to guarantee convergence. Secondly, in each iteration, sensors exchange not only decision variables but also Lagrangian multipliers. Considering that communication cost is the main source of energy consumption, the following question arises: *Can we design a more energy-efficient algorithm for a distributed network?*

This paper proposes a distributed linear programming (DLP) algorithm to solve \mathbf{Pa} in a more energy-efficient way. An inspiration for low-cost distributed processing comes from [19], where Tseng proposed a distributed optimization framework for linear programs satisfying a certain diagonal dominance conditions. In the distributed optimization framework, an objective function is divided into uncoupled terms, and constraints are assigned to different processors. This distributed processing framework stimulates us to assign the objective function and constraints in \mathbf{Pa} to individual sensors as:

$$\begin{aligned} \min c_i, \\ \text{s.t. } |y_i - f_{ii} c_i - \sum_{j \in \mathcal{N}_i} f_{ji} c_j - 1| \leq \theta, c_i \geq 0. \end{aligned} \quad (18)$$

Solution to (17) is $c_i = [y_i - 1 - \theta - \sum_{j \in \mathcal{N}_i} f_{ji} c_j]^+ / f_{ii}$ or null if $y_i - 1 + \theta - \sum_{j \in \mathcal{N}_i} f_{ji} c_j < 0$. Therefore, we have the following distributed DLP algorithm to \mathbf{Pa} :

Algorithm 2: DLP

- Step 1: Each sensor i holds a predefined common threshold θ , and a calculated damage-sensitive coefficient y_i . The initial decision variable is set as 0.
- Step 2: In iteration $t + 1$, each sensor i broadcasts its current decision variable $c_i(t)$ to its one neighboring sensors $j \in \mathcal{N}_i$.
- Step 3: Each sensor i updates its decision variable by $c_i(t+1) = [y_i - 1 - \theta - \sum_{j \in \mathcal{N}_i} f_{ji} c_j(t)]^+ / f_{ii}$.
- Step 4: Repeat Step 2 and Step 3 until reaching convergence.
-

The simplicity of DLP over ADMM is evident. Furthermore, the convergence rate of DLP is faster than that of ADMM, and sensors do not need to exchange extra information other than current decision variables.

D. Application-Related Issues

In this subsection, we will discuss three application-related issues, including selection of basis function, asynchronous optimization and network robustness.

- 1) *Selection of basis function:* Selection of basis function plays a critical role in the online SHM system. Intrinsicly, the basis function of one point is decided by the structure itself. Since direct experiments on the structure is impractical, an alternative approach is to simulate the influence of damages with software in advance. However, simulation and pre-programming with respect to individual sensors are time-consuming for a large network. Hence we rather consider a general type of basis function, such as the isotropic Gaussian shape in (8), which has been proved to be a good approximation of practical cases. The control factor σ_i is

identical for any i and can be obtained from simple simulation.

Selection of the basis function, combined with selection of the communication range of sensors, is also related to the performance gap between **P1** and **Pa**. If the control factor σ_i is large while the communication range is small, influence of i is underestimated in relaxing **P1** to **Pa**; thus the performance gap is no longer neglectable. Hence, it is essential for sensors to communicate with an adequate range in order to ensure modeling accuracy.

Let us consider the consequences of model mismatch, namely, if σ_i is different from its real value. If point i is damaged and σ_i is chosen to be larger than its real value, then a correctly decided c_i may lead to infeasibility of **Pa**, since $\{c_j, j \in \mathcal{N}_i\}$ possibly have to be negative to meet the threshold constraints. Here the simple updating rule of DLP shows to be robust because Step 3 in Algorithm 2 still updates decision variables even when the **Pa** is infeasible.

If point i is damaged and σ_i is chosen to be smaller than its real value, then the neighboring sensors may report damages to match the damage-related coefficients. This directly results in false-positive alarms. An extreme setting is $\sigma_i = 0$, which ignores the interrelationship between neighboring points and leads to a trivial solution, namely, reporting the damage for point i if $y_i - 1 \geq \theta$.

2) *Asynchronous optimization*: In our online monitoring scheme, decision making is carried out in every sampling period based on the damage-sensitive coefficients $\{y_i\}$ collected during this period. Hence, coarse synchronization is needed to ensure that each sensor i contributes its data y_i within the right sampling period. However, in the decision-making period, though the network is assumed to be synchronized for each iteration, the ADMM and DLP algorithms can be implemented asynchronously [15]. In Step 2 of Algorithms 1 and 2, sensors can no longer broadcast synchronously. On the contrary, sensors hold random timers, wake up randomly, and sends requests to neighboring sensors. Upon receiving requests, sensors wake up and start to broadcast. Then sensors turn to sleep mode to save energy.

The advantages of the asynchronous scheme are three-fold. Firstly, sensors only need to synchronize for sampling and decision-making periods, other than for each iteration of the optimization process. Hence the burden of fine synchronization can be avoided. Secondly, interference of synchronized broadcasting is diminished via randomly asynchronous broadcasting; therefore the probability of packet loss is reduced. Thirdly, switching between random wake-up mode and sleep mode helps the network to save energy, thus prolongs the network lifetime.

3) *Network robustness*: Finally we discuss the advantage of robustness brought by the distributed network infrastructure. In the centralized network, failure of one sensor not only results in the loss of corresponding measurement, but also brings difficulty for routing. Whereas in the distributed network, simple one-hop communication is in place of end-to-end communication between sensors and the fusion center; hence the network robustness is improved.

Furthermore, the distributed algorithms are robust to

packet loss and communication errors. Packet loss of intermediate decision variables and Lagrangian multipliers means the absence of one iteration step. According to our discussion of asynchronous optimization, it does not affect the convergence of the algorithms. Similarly, communication errors can be treated as perturbation of the iterations, and do not obstruct the convergence of the algorithm as long as the optimization formulation keeps unaltered.

V. SIMULATION RESULTS

In this section, we provide extensive simulation results to illustrate the effectiveness of the proposed distributed online SHM algorithms based on a two-dimensional structure model.

A. General Settings

We consider a steel frame structure with 12 stories and 9 bays, simplified as a two-dimensional model, as in Fig. 1. A grid network of 120 sensors is deployed in the joint points. The width of a bay is 24 feet and the height of a floor is 14 feet. Ambient Gaussian white noises are imposed to the foundation. Response of the structure is analyzed by the finite element program OpenSees [20]. Damage patterns are introduced by reducing the structural stiffness of one or several columns.

In each monitoring period of both modeling and decision-making stages, 1000 acceleration output measurements are sampled to generate AR and ARX models. The order of the AR model is set as $p = 30$ in (1) and the orders of the ARX model are set as $a = 5$ and $b = 5$ in (2). After a monitoring period of the modeling stage, each sensor stores the model information in its database, as described in Section III-A. During a monitoring period of the decision-making stage, each sensor identifies AR and ARX models and calculates the damage-sensitive coefficient according to Section III-C. Then sensors collaboratively estimate the severity vector based on the distributed algorithms. A sensor reports to the central console upon detecting a damage, namely, when the corresponding severity coefficient is larger than 0.

Here we compare the performance of four algorithms:

- 1) *Centralized*: The ADMM algorithm in which sensors have infinite communication range. Hence it returns the optimal solution to the centralized formulation **P1**. In the basis function, $\sigma_i = 14$ for all i .
- 2) *ADMM*: The ADMM algorithm in which each sensor has a communication range slightly larger than 24 feet. That is, each sensor is able to communicate with 4 neighboring sensors. The ADMM algorithm is optimal to the distributed formulation **Pa**. In the basis function, $\sigma_i = 14$ for all i .
- 3) *DLP*: The DLP algorithm in which each sensor also has a communication range slightly larger than 24 feet. In the basis function, $\sigma_i = 14$ for all i .
- 4) *Threshold*: The DLP algorithm in which each sensor also has a communication range slightly larger than 24 feet. In the basis function, $\sigma_i = 0$ for all i . This solution reports damage for point i if $y_i - 1 \geq \theta$.

Throughout the simulations the threshold is set as $\theta = 0.04$.

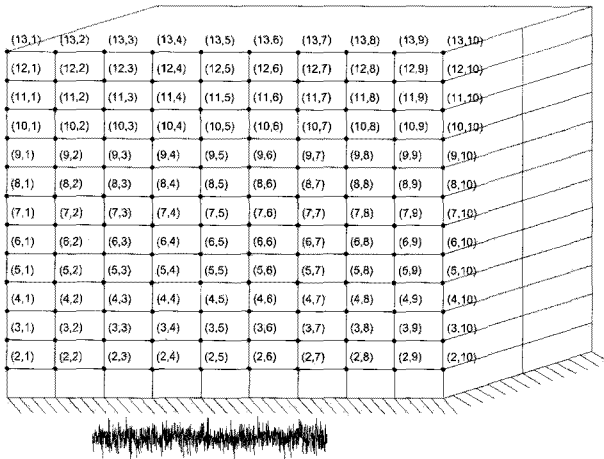


Fig. 1. Two-dimensional model of a steel frame structure with 12 stories and 9 bays. Gaussian random white noises are imposed to the foundation to simulate ambient vibrations.

To describe the performance of the algorithms, we adopt the following criteria: Convergence rate (which decides energy-efficiency of the network), false-negative rate (neglecting a damage when it occurs), and false-positive rate (reporting a damage when it does not exist).

B. Single-Damage Pattern

Let us firstly consider a single-damage pattern, in which we reduce 57% stiffness for the column between sensors (6,5) (located in 6th floor, 5th bay) and (7,5) (located in 7th floor, 5th bay). The spatial distribution of the damage-sensitive coefficients is shown in Fig. 2. The centralized, ADMM, and DLP solutions all converge to a similar decision vector and report damages in correct points, as depicted in Fig. 3. Performance gap between the centralized and ADMM solutions are trivial, since the distributed formulation \mathbf{Pa} is a good approximation of the centralized formulation \mathbf{PI} .

Comparing the convergence rate, the ADMM solution converges after 40 iterations while the DLP solution converges after 4 iterations. Furthermore, the ADMM solution requires more information exchange per iteration. For the ADMM solution, each sensor needs to broadcast one decision variable and four Lagrangian multipliers in each iteration; while for the DLP solution, each sensor only needs to broadcast one decision variable in each iteration. Hence the DLP solution drastically improves the energy-efficiency of the network.

Secondly, we still consider the single-damage pattern discussed above with 4 more data sets to study false alarms of the three distributed algorithms. Damage identification results are shown in Table 1. The optimal ADMM solution to \mathbf{Pa} is sometimes infeasible because of model mismatch. The threshold algorithm tends to generate false-positive alarms as it ignores interrelationship between neighboring points. The DLP algorithm is robust to model mismatch and reduces false-alarms by introducing the interrelationship.

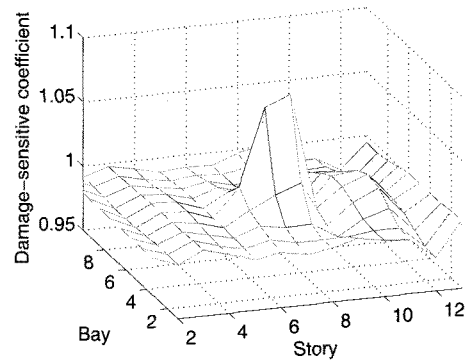


Fig. 2. Spatial distribution of the damage-sensitive coefficients after introducing damage to the column between sensors (6,5) and (7,5).

Table 1. Performance of the three distributed algorithms for the single-damage pattern.

Data set	Algorithm	Feasibility	False-negative	False-positive
1	ADMM	Yes	0	0
	DLP	Yes	0	0
	Threshold	Yes	0	0
2	ADMM	No	–	–
	DLP	Yes	0	0
	Threshold	Yes	0	1
3	ADMM	No	–	–
	DLP	Yes	1	0
	Threshold	Yes	1	1
4	ADMM	Yes	0	0
	DLP	Yes	0	0
	Threshold	Yes	0	1

C. Multiple-Damage Pattern

In the multiple-damage pattern, more than one columns are damaged. Consider damage of the column beneath the sensor (1,1) and damage of the column between sensors (6,5) and (7,5), both with stiffness reduction as 57%. Table 2 provides simulation results on 4 sets of data, in which each data set contains 1000 sampling points. The ADMM solution greatly suffers from the infeasibility problem, while the threshold algorithm generates many false-positive alarms. On the contrary, the DLP algorithm both tackles the infeasibility problem and reduces the false-positive alarms.

D. Damage Severity

Now we discuss the ability of the online SHM algorithms, ADMM and DLP, to evaluate damage severity. By setting stiffness reduction as different levels, relative severity coefficients are shown in Fig. 4. For each sensor point, the severity coefficient increases at the stiffness reduction increases, therefore the proposed online SHM algorithms are able to not only localize damage position but also quantify damage severity. It should be noted that the severity coefficient of point (6,5) is always larger than that of point (7,5). This common phenomenon in the AR-ARX method indicates that damage in a column has larger effect on the upper part of a structure than that on the lower part.

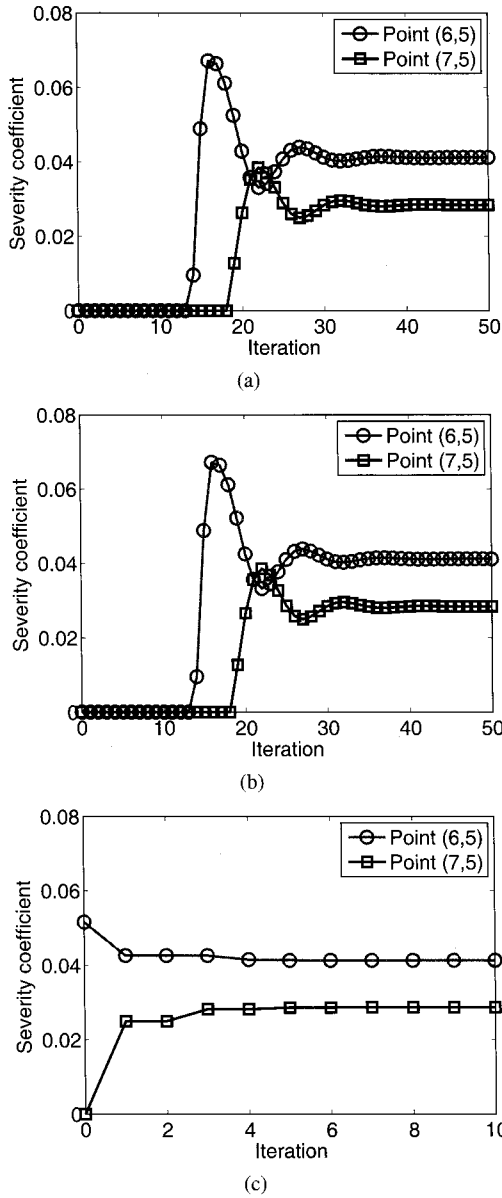


Fig. 3. Convergence properties of the three algorithms: (a) Centralized, (b) ADMM, and (c) DLP.

VI. CONCLUSIONS

In this paper, we discuss the distributed decision-making problem in a large WSN, and focus on its application in on-line SHM. Observing the fact that damages are generally scarce in a structure, this paper develops a nonlinear 0-norm minimization formulation to recover the sparse damage severity vector. Motivated by the current progress on compressive sensing, we relax the nonlinear program to a 1-norm convex program, and further an approximated linear program which is distributively tractable. Two algorithms, an optimal algorithm based on ADMM and a heuristic DLP algorithm, are proposed to make decisions in a distributive and collaborative way. The distributed in-network processing scheme limits information exchange to among one-hop neighboring sensors, hence improves energy-efficiency and network robustness.

Under the context of in-network processing for WSNs, one

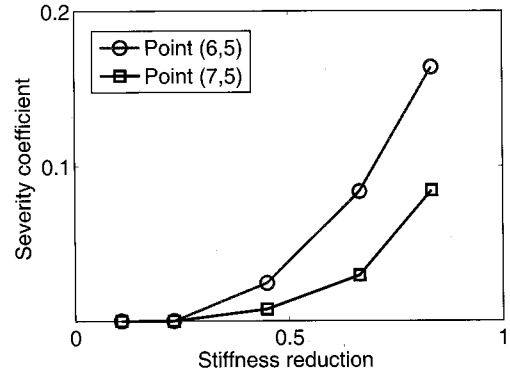


Fig. 4. Sensitivity of the ADMM and DLP algorithms to stiffness reduction.

Table 2. Performance of the three distributed algorithms for the multiple-damage pattern.

Data set	Algorithm	Feasibility	False-negative	False-positive
1	ADMM	No	-	-
	DLP	Yes	0	2
	Threshold	Yes	0	15
2	ADMM	No	-	-
	DLP	Yes	0	1
	Threshold	Yes	0	4
3	ADMM	No	-	-
	DLP	Yes	1	1
	Threshold	Yes	1	9
4	ADMM	No	-	-
	DLP	Yes	1	1
	Threshold	Yes	1	1

of our future work is to consider larger sets of basis functions. Herein choosing basis functions for better description of the sensing field will be a challenging but interesting topic.

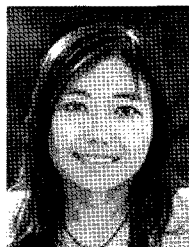
REFERENCES

- [1] I. Akyildiz, W. Su, Y. Sankarasubramanian, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, pp. 102-114, 2002.
- [2] J. Kotecha, V. Ramachandran, and A. Sayeed, "Distributed multitarget classification in wireless sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 23, pp. 703-713, 2005.
- [3] J. Xiao, A. Ribeiro, Z. Luo, and G. Giannakis, "Distributed compression-estimation using wireless sensor networks," *IEEE Signal Process. Mag.*, vol. 23, pp. 27-41, 2006.
- [4] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Inf. Theory*, vol. 52, pp. 2508-2530, 2006.
- [5] J. Predd, S. Kulkarni, and H. Poor, "Distributed learning in wireless sensor networks," *IEEE Signal Process. Mag.*, vol. 23, pp. 56-69, 2006.
- [6] C. Farrar, T. Duffey, S. Doebbling, and D. Nix, "A statistical pattern recognition paradigm for vibration-based structural health monitoring," in *Proc. Int. Workshop on Structural Health Monitoring*, 1999, pp. 1-10.
- [7] S. Doebbling, C. Farrar, and M. Prime, "A summary review of vibration-based damage identification methods," *The Shock and Vibration Digest*, vol. 30, pp. 91-105, 1998.
- [8] Y. Gao, B. Spencer, and M. Ruiz-Sandoval, "Distributed computing strategy for structural health monitoring," *Structural Control and Health Monitoring*, vol. 13, pp. 488-507, 2006.
- [9] J. Lynch, "Embedding damage detection algorithms in a wireless sensing unit for operational power efficiency," *Smart Materials and Structures*, vol. 13, pp. 800-810, 2004.
- [10] H. Sohn and C. Farrar, "Damage diagnosis using time series analysis of vibration signals," *Smart Materials and Structures*, vol. 10, pp. 445-451, 2001.

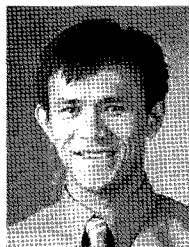
- [11] D. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, pp. 1289–1306, 2006.
- [12] F. Fagnani and S. Zampieri, "Randomized consensus algorithms over large scale networks," *IEEE J. Sel. Areas Commun.*, vol. 26, pp. 634–649, 2008.
- [13] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," in *Proc. Int. Symposium on Information Processing in Sensor Networks*, 2004, pp. 20–27.
- [14] I. Schizas, A. Ribeiro, and G. Giannakis, "Consensus in ad hoc WSNs with noisy links. Part I: distributed estimation of deterministic signals," *IEEE Trans. Signal Process.*, vol. 56, pp. 350–364, 2008.
- [15] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, 2nd Ed., Athena Scientific, 1997.
- [16] D. Bertsekas, *Numerical Optimization*, 2nd Ed., Athena Scientific, 1999.
- [17] J. Lynch, "An overview of wireless structural health monitoring for civil structures," *Philosophical Transactions of the Royal Society A*, vol. 365, pp. 345–372, 2007.
- [18] Q. Ling, Z. Tian, Y. Yin, and Y. Li, "Localized structural health monitoring in wireless sensor networks," *IEEE Sensors J.*, to appear.
- [19] P. Tseng, "Distributed computation for linear programming problems satisfying a certain diagonal dominance condition," *Mathematics of Operations Research*, vol. 15, pp. 33–48, 1990.
- [20] Pacific Earthquake Engineering Research Center, "OpenSees: Open system for earthquake engineering simulation," 1999.



Qing Ling received the B.S. degree in automation and the Ph.D. degree in control theory and control engineering from the University of Science and Technology of China, Hefei, Anhui, China, in 2001 and 2006, respectively. Since 2006, he has been with the Department of Electrical and Computer Engineering, Michigan Technological University, Houghton, MI, USA, as a post-doctoral research fellow. His current research focuses on distributed optimization and in-network signal processing. He received the President Scholarship of the Chinese Academy of Sciences in 2006.



Zhi Tian received the B.E. degree in Electrical Engineering (Automation) from the University of Science and Technology of China, Hefei, Anhui, China, in 1994, the M. S. and Ph.D. degrees from George Mason University, Fairfax, VA, USA, in 1998 and 2000 respectively. Since August 2000, she has been with the department of Electrical and Computer Engineering, Michigan Technological University, where she is currently an Associate Professor. Her current research focuses on signal processing for wireless communications, particularly on ultra-wideband systems, cognitive radios and distributed sensor networking. She serves as Associate Editor for *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, and was an Associate Editor for *IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS*. She is the recipient of a 2003 CAREER award from US NSF.



Yue Li joined the Department of Civil and Environmental Engineering, Michigan Technological University, as Donald and Rose Ann Tomasini Assistant Professor of Structural Engineering in August 2005. He earned his Ph.D. degree in Civil Engineering, with an emphasis in Structural Engineering, from Georgia Institute of Technology, Atlanta, GA, in August 2005. His research interests include structural reliability analysis, probabilistic design, natural and man-made hazard mitigation, structural load modeling and combinations of loads, structural monitoring and condition assessment, bridge engineering, performance-based engineering, earthquake engineering, wind engineering, and wood engineering. He received Michigan Tech Research Excellent Fund Award in 2008.