

# Distributed Reference-Free Fault Detection Method for Autonomous Wireless Sensor Networks

Chun Lo, *Student Member, IEEE*, Jerome P. Lynch, *Member, IEEE*, and Mingyan Liu, *Senior Member, IEEE*

**Abstract**—Compact and low-cost sensors used in wireless sensor networks are vulnerable to deterioration and failure. As the number and scale of sensor deployments grow, the failure of sensors becomes an increasingly paramount issue. This paper presents a distributed, reference-free fault detection algorithm that is based on local pair-wise verification between sensors monitoring the same physical system. Specifically, a linear relationship is shown to exist between the outputs of a pair of sensors measuring the same system. Using this relationship, faulty sensors may be detected within subsystems of the global system. Moreover, faulty sensors suffering from sparse spikes in their measurements can be identified with spike magnitudes and times accurately estimated. An appealing feature of the proposed method is that the need for reference sensors and complete knowledge of the system input are not required. Due to the pair-wise nature of the proposed algorithm, it can also be performed in a completely decentralized fashion. This ensures the method can be scaled to large sensor networks and lead to significant energy savings derived from reduced wireless communication compared to centralized approaches.

**Index Terms**—Fault detection, in-network data processing, wireless sensor networks (WSNs).

## I. INTRODUCTION

RECENT advances in communication technology and embedded systems has resulted in wireless sensors that are smaller in size, consume less power and achieve higher transmission rates. By adopting wireless sensor networks (WSNs), the cost of monitoring systems, such as structural [2], [3] and environmental monitoring systems [4], [5], are greatly reduced due to the eradication of expensive wiring. The small form factor of wireless sensors also opens up new applications that require mobility or portability, such as animal [6] and vehicle [7] tracking. However, small form factors and low costs also render wireless sensors more susceptible to faults and failures. To ensure wireless sensor networks are

reliable over long periods of service, *automated* detection and identification of sensor faults must be an integral part of their design and operation, especially for applications where the monitoring system is unattended after deployment. With most wireless sensors limited in both energy and processing capacity [8], it is highly desirable to design *energy efficient* and *low-complexity* fault detection algorithms that can be embedded directly into the wireless sensors for in-network execution.

Over the past decade, sensor fault and failure detection has been extensively studied by the control system field. A sensor failure is defined as when a sensor is in an irrecoverable state of inoperation (e.g., not turning on or not responding). In contrast, a sensor fault is defined as when the sensor is outputting measurements but the measurements are intermittently or permanently corrupted. Various forms of sensor faults exist including excessive noise, mean drifts, random spikes and nonlinearity in the sensor transduction mechanism.

Most of the prior studies on sensor fault detection are based on concepts associated with system failure detection dating back to the 1980s [9]–[11]. The most popular approach is to make use of analytical redundancies in the system. The main concept behind this approach is that when multiple sensors are attached to the same physical system, they observe common system dynamics resulting in correlation between the observations made by these sensors. Such correlations can be utilized to evaluate the behavior of a sensor of unknown fault status. Any discrepancy detected may be treated as a potential sensor fault feature and subject to further analysis. Many notable studies have explored analytical redundancy to determine sensor faults [12]–[18]. For example, Li *et al.* [14] presented a reference-based sensor fault detection method which requires the existence of a number of fault-free sensors defined *a priori* as reference sensors. The method represents the output of any uncertain sensor (i.e., a sensor with unknown fault status) as a function of the output of the reference sensors based on the model of the system. Fault detection is based on discrepancies between the predicted and measured output of the uncertain sensor. This method requires the number of reference sensors to be equal to or larger than the number of uncertain sensors. Alternatively, Da and Lin [15] proposed a sensor fault detection algorithm using a bank of Kalman filters. Abnormal sensor behavior is detected by comparing the system state estimated by a subset of sensors with that estimated by all of the sensors under the assumption that these two states should agree with each other when the system is fault-free. This method is more efficient in detecting soft sensor faults (i.e., those that occur gradually) but can generally be applied

Manuscript received November 9, 2012; revised January 2, 2013; accepted January 17, 2013. Date of publication February 1, 2013; date of current version April 10, 2013. This work was supported in part by the National Science Foundation under Grant CCF-0910765, the Office of Naval Research under Contract N00014-09-C0103 and Contract N00014-12-1-0178, and the U.S. Department of Commerce, the National Institute of Standards and Technology Technology Innovation Program under Cooperative Agreement 70NANB9H9008. A preliminary version of this work has appeared in the 4th International Symposium on Resilient Control Systems [1]. The associate editor coordinating the review of this paper and approving it for publication was Dr. Ashish Pandharipande.

The authors are with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: chunlo@umich.edu; jerlynch@umich.edu; mingyan@umich.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/JSEN.2013.2244881

to sudden and intermittent faults. Ricquebourg *et al.* [17] proposed a Markov chain modeling approach to sensor fault detection. The method captures the sensor dynamics by a Markov chain under a transferable belief framework. Once the model is established, any sensor whose observation disagrees with the Markov chain transition is further analyzed using pre-defined decision rules. A fault detection filter design method based on the state space model was proposed by Dai *et al.* [18] to increase the sensitivity of the discrepancy estimation to the faults. The filter detects faults existing in the system instead of individual sensors by observing the output discrepancy of the entire system. When the noise disturbance is band-limited, and the disturbance power spectral density is known, the study showed that the filter zeros can be placed near the disturbance frequency to attenuate the noise disturbance, i.e., increase the significance of the fault signal in the discrepancy observation.

These prior studies all require knowledge of the system model and/or the presence of reference sensors. However, an accurate system model may be difficult to obtain in practice and ensuring the existence of (especially a large number of) reference sensors can be challenging, if at all possible. Furthermore, the aforementioned algorithms are centralized and require a base station where data from all of the sensor nodes has been aggregated *prior to* the execution of the detection algorithm. While such an approach is appropriate for wired monitoring systems, it is potentially ill-suited for WSNs where communication of raw data to a centralized base station can consume scarce energy. There have been some efforts aimed at attempting to detect sensor faults in a decentralized manner. For example, Chen *et al.* [19] proposed an iterative and distributive detection algorithm. This method requires the network to be dense so that sensors have highly correlated outputs. Each sensor is evaluated by its neighbors who collectively generate the “tendency state” of the sensor via a voting process. Afterward, each sensor’s “tendency state” is compared with its neighbors’ to decide if the sensor is faulty or not. Although the detection process occurs locally, each sensor is required to compare its output with its neighbors several times; at least 5 neighbors are recommended to ensure the method has a high level of accuracy. This drives the communication cost unnecessarily high and increases the processing time.

In this paper, a novel fault detection method which utilizes system redundancy but *without* requiring knowledge of a physics-based system model or the existence of reference sensors is proposed. The performance of the algorithm will not be affected even if all of the sensors in the network are faulty. The method is especially well suited for resource constrained WSNs because the detection algorithm is run locally by each wireless sensor node resulting in reduced communication demand compared to existing centralized methods. Under certain conditions, knowledge of system inputs is not necessary for the detection algorithm to work. The method is capable of detecting general faults within arbitrary pairs of sensors (i.e., subsystems). However, the method is further specialized to *identify* spike faults from other type of faults with the aim of detecting and quantifying (e.g., location, magnitude) spikes so that they could be removed during post-processing. This is a very appealing feature of the proposed

algorithm that the aforementioned studies lack, as spike faults are common in sensors. One of the main causes of this type of sensor fault is loose electrical contacts inside a sensor node or from poor shielding in an active electro-magnetic environment. In addition, the noise process inherent to a sensor (if excessive can be regarded as a fault) is also characterized by the sensor fault detection methodology proposed. This illustrates the versatility of the fault detection methodology.

The remainder of this paper is organized as follows. In Section II, the fault detection problem is formally stated. This section establishes a linear relationship between pairs of sensors from a linear time invariant (LTI) model to lay foundation for the fault detection method introduced in Section III. In Section III, an analytical redundancy-based sensor fault detection method using pairs of sensors is presented. The method presented is quite general and is not limited to any particular type of sensor fault but the application of the method to the detection of spike faults is offered. The performance of the fault detection method is evaluated by simulation and presented in detail in Section IV. Finally, the paper concludes in Section V with a summary of the study’s key findings and a discussion of future research efforts.

## II. PAIR-WISE LINEAR RELATIONSHIPS IN LINEAR SYSTEMS: FOUNDATION FOR FAULT DETECTION

This section lays a theoretical foundation for the sensor fault detection methodology to be described in Section III. Consider a set of wireless sensors attached to a time-invariant physical system. Since sensor responses all depend on the common physical system, a linear relationship exists between the system outputs measured by these sensors. This relationship can then be exploited to evaluate the “correctness” of the sensor measurements. Specifically, sensors can pair up and check whether their outputs are consistent with this linear relationship; inconsistencies can then be used to determine whether one or both of the sensors may be faulty. This pair-wise comparison can be performed between any pair of sensors and only the result of the comparison needs to be conveyed to the base station or a central processing node in the WSN. For the purpose of conserving energy in the WSN, sensors are generally grouped within close proximity of each other.

The structure of the proposed algorithm is illustrated in Fig. 1. The algorithm can be separated into a training (also called “model parameter identification”) phase and a detection phase. During the training phase, each sensor node learns the relationship between itself and each of its neighbors. For example, in Fig. 1(a), sensor 2 broadcasts its measurement data to neighbouring sensors 1 and 3. After the data is received, sensors 1 and 3 calculate the relationship between their outputs and sensor 2’s output [Fig. 1(b)]. This model parameter identification processes is performed by each sensor one after another and the results are stored locally, as shown in Fig. 1(c). During the detection phase, the network is partitioned into pairs of neighboring sensors; this can be done centrally or in a distributed fashion. Each pair of sensors then performs a comparison according to their trained relationship. Fig. 1(d) shows a network partitioned into 3 pairs: {1, 2}, {2, 3} and {4, 5}. For

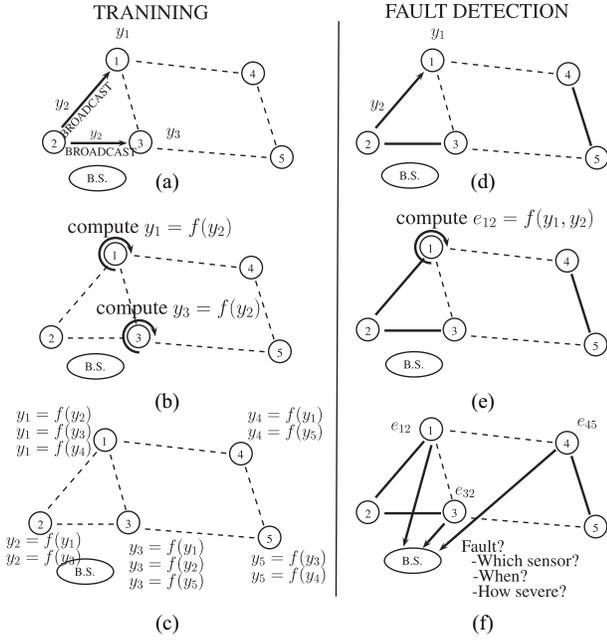


Fig. 1. During training: (a) each sensor broadcasts its output, (b) linear relationship between sensor pairs is calculated, and (c) finally pair-wise linear relationships of the network are constructed. For fault detection: (d) base station divides the sensor network into pairs, (e) each pair performs the fault detection method, and (f) each pair sends their results,  $e$ , back to the base station (B.S.).

example, consider sensor pair  $\{1, 2\}$ . Sensor 2 first transmits its output to its partner, sensor 1. Sensor 1 then checks whether its measured output agrees with the output predicted by the previously trained relationship [Fig. 1(e)]. Finally, each sensor pair will report its results to the base station [Fig. 1(f)]. In this architecture, fault detection is executed locally and only the diagnostic results need to be sent to the base station, thereby drastically reducing the communication cost in the multi-hop communication network. With computation and communication requirements distributed over the entire network, the WSN is more scalable to larger node counts while consuming less energy. The relationship between sensor outputs,  $y_p = f(y_q)$  is now derived. To simplify the discussion, the relationship is established between two sensors but the derivation can be generalized to sets of sensors. Consider a physical system that is represented mathematically by the following *deterministic* state space model

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k), \end{aligned} \quad (1)$$

where  $\mathbf{x}(k) \in \mathbb{R}^n$  is the state vector of the system,  $\mathbf{u}(k) \in \mathbb{R}^l$  is the input vector, and  $\mathbf{y}(k) \in \mathbb{R}^m$  is the output vector of the sensors. Furthermore,  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is the state transition matrix which defines the transition of system states,  $\mathbf{B} \in \mathbb{R}^{n \times l}$  is the input matrix which represents the relationship between the input and the system state,  $\mathbf{C} \in \mathbb{R}^{m \times n}$  is the output matrix, and  $\mathbf{D} \in \mathbb{R}^{m \times l}$  is the feed-through matrix. In the remainder of this paper, time-invariant is assumed. The system is also assumed to be stable, i.e., the output of the system,  $\mathbf{y}$ , is bounded when the system input,  $\mathbf{u}$ , is bounded.

Taking the Z-transform of (1), the discrete-time frequency domain representation is derived:

$$\begin{aligned} z\mathbf{X}(z) &= \mathbf{A}\mathbf{X}(z) + \mathbf{B}\mathbf{U}(z) \\ \mathbf{Y}(z) &= \mathbf{C}\mathbf{X}(z) + \mathbf{D}\mathbf{U}(z). \end{aligned} \quad (2)$$

Eliminating  $\mathbf{X}(z)$  in the second equation of (2) using the first equation of (2), resulting in the following expression:

$$\mathbf{Y}(z) = (\mathbf{C}(z\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D})\mathbf{U}(z). \quad (3)$$

For the  $p^{\text{th}}$  sensor, the individual observation model is

$$Y_p(z) = (\mathbf{C}_p(z\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}_p)\mathbf{U}(z) \quad (4)$$

where  $\mathbf{C}_p$  and  $\mathbf{D}_p$  are the  $p^{\text{th}}$  rows of the matrices  $\mathbf{C}$  and  $\mathbf{D}$ , respectively. As a result, the transfer function  $H_{pq}(z)$  between the outputs of sensor  $p$  and  $q$  is:

$$H_{pq}(z) = \frac{Y_p(z)}{Y_q(z)} = \frac{(\mathbf{C}_p(z\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}_p)\mathbf{U}(z)}{(\mathbf{C}_q(z\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}_q)\mathbf{U}(z)}. \quad (5)$$

The above expression shows that there exists a linear relationship between any pair of sensors. If the expression  $(\mathbf{C}_p(z\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}_p) \in \mathbb{R}^{1 \times l}$  is expressed as a polynomial function of  $z$ , then the  $i^{\text{th}}$  element of the vector can be expressed as  $\sum_{j=0}^n \alpha_{ij} z^j$ , where  $n$  is the rank of  $(z\mathbf{I} - \mathbf{A})^{-1}$  and the coefficients  $\alpha_{ij}$ ,  $i = 1, \dots, l$ ,  $j = 0, \dots, n$ , are determined by the various state space matrices (i.e.,  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$  and  $\mathbf{D}$ ). Similarly, the  $i^{\text{th}}$  element of  $(\mathbf{C}_q(z\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}_q)$  can be expressed as  $\sum_{j=0}^n \beta_{ij} z^j$ . Let  $\mathbf{U}^T(z) = [U_1, \dots, U_l]$ , and the transfer function (5) can be written as:

$$\frac{Y_p(z)}{Y_q(z)} = \frac{\sum_{j=0}^n \left( \sum_{i=1}^l U_i \alpha_{ij} \right) z^j}{\sum_{j=0}^n \left( \sum_{i=1}^l U_i \beta_{ij} \right) z^j}. \quad (6)$$

Notice that this relationship depends on the system input  $\mathbf{U}$ . The input driving the system dynamics may be known or can be measured by sensors; in other cases, it can be difficult to obtain. However, if the excitation of the system can be aggregated as a single source (i.e., as a scalar time-history function), then  $l = 1$  and  $\mathbf{U}$  in the numerator and denominator cancel each other. Thus the dependence on the system input in (5) is eliminated. Hence, a linear relationship uniquely defined by the physics of the system and the sensors measurements can be obtained:

$$\frac{Y_p(z)}{Y_q(z)} = \frac{\alpha_n z^n + \alpha_{n-1} z^{n-1} + \dots + \alpha_1 z + \alpha_0}{\beta_n z^n + \beta_{n-1} z^{n-1} + \dots + \beta_1 z + \beta_0}. \quad (7)$$

Reduction of the system input to a single source is quite common in many engineering systems. For example, mechanical systems excited by ambient, white noise processes and civil engineering structures exposed to base motion (i.e., earthquakes), would all be modeled by a single excitation source. In order to simplify the discussion, the scalar system input case is considered in the remainder of the paper. Note however that the following derivation and discussion remain valid under multi-input systems if they are known (or measurable) and  $\alpha_j$  (and  $\beta_j$ ) is replaced with  $\sum_{i=1}^l U_i \alpha_{ij}$  (and  $\sum_{i=1}^l U_i \beta_{ij}$ ) in (7).

Both  $y_p$  and  $y_q$  correspond to system outputs. However, if  $y_p$  is viewed as the output while  $y_q$  is viewed as the *input* of another, unspecified system, then (7) is essentially the Z-domain representation of an autoregressive with exogenous input (ARX) time-series model [20]. ARX( $n$ ,  $m$ ) is a linear time series model with  $n$  output terms (autoregressive terms) and  $m$  (exogenous) input terms. It is widely used to model various types of systems and natural phenomena. The definition of an ARX model is :

$$\sum_{i=0}^m \alpha_i y_q(k-i) = \sum_{i=0}^n \beta_i y_p(k-i). \quad (8)$$

The above derivation shows that the relationship between the outputs of two sensors is precisely captured by the ARX model, which is defined by of coefficients  $\alpha_i$  and  $\beta_i$ . This ARX model ( $\alpha_i$  and  $\beta_i$ ) can be acquired by first storing the output pairs,  $y_p$  and  $y_q$  over a certain period of time when sensors work under normal conditions and then  $\alpha_i$  and  $\beta_i$  are calculated from the stored data through standard least square calculations [20], [21] or through the iterative Burg's method [22]. Even if the historical data are corrupted by (zero mean) Gaussian noise, these training methods are able to extract accurate model coefficients. This is because when the size of the historical data is large enough, least square calculation or Burg's method is able to eliminate the variance of the noise in the data. Therefore, the ARX model training is insensitive to noise existing in the training data. After the training, only the ARX model coefficients  $\alpha_i$  and  $\beta_i$  need to be stored for use in future fault detection. The length of the time history needed is equal to the dimension of the state  $x$  of the original state space model in (1). Consequently, the number of coefficients  $n$  of the ARX model should be equal to or larger than the size of the state dimension.

In terms of computational complexity on wireless sensors, the training of ARX coefficients requires  $\mathcal{O}(v^2N)$  operations for the least square method and  $\mathcal{O}(vN)$  operations for Burg's method as stated in [23] and [24], respectively, where  $v$  is the number of ARX coefficients and  $N$  is the number of training data samples. The ARX training with Burg's method has been implemented in field deployed WSNs by Lynch, *et al.* [25], where 8.351 seconds of execution time and 3.031J power consumption are reported for training with  $v = 30$  and  $N = 4000$  by a Power PC embedded 32-bit microprocessor (Motorola MPC555). This is feasible in WSNs as the need for ARX model training is usually infrequent. The detection process of the proposed algorithm requires  $\mathcal{O}(vN')$  operations as indicated in the next section (where  $N'$  is the number of data points to be detected). The detection process is expected to be significantly faster than the training process.

The ARX model representation of the relationship between sensor pairs is extremely valuable and will be exploited fully. While (7) provides a closed-form analytical expression for the relationship between sensor pairs, it would require an accurate representation of the system in the form of a state space model (i.e., knowledge of  $A$ ,  $B$ ,  $C$ , and  $D$ ). In contrast, the equivalent ARX model in (8) can be determined after the network has been deployed only using sensor outputs  $y_p$  and  $y_q$ . After an ARX model has been determined between two sensors, the

model is stored locally in each wireless sensor in the form of the model coefficients,  $\alpha_i$  and  $\beta_i$ .

### III. SPIKE ERROR DETECTION USING ARX MODELS

In the previous section, it was shown that an ARX model can accurately represent the linear relationship between sensor outputs when the number of coefficients is equal to or larger than the dimension of the system state,  $n$ . In this section, the actual sensor fault detection methodology based on ARX models is presented. A spike fault is a voltage spike (or impulse) superimposed on the sensor measurement. Spikes typically occur randomly in time and can be constant or of varying magnitude. Here, it is assumed that each sensor can potentially suffer from a spike fault and that there are no reference (i.e., known faultless) sensors at the time of execution. The spike error could occur randomly at any time and on any sensor. It is assumed that the duration of a spike error is short and the occurrence of these spike errors is sparse (i.e., the probability that the spikes occur consecutively is low). The precise definition of sparsity to be used herein is given by: the spikes occur independently at each time instance with low probability (5% or less)

As shown in the previous section, using the ARX model allows the outputs of two sensors,  $y_1$  and  $y_2$  to be related through the use of coefficients  $\alpha_i$  and  $\beta_j$ :

$$\sum_{i=0}^m \alpha_i y_1(k-i) = \sum_{j=0}^n \beta_j y_2(k-j). \quad (9)$$

To simplify the expression, the coefficient values are normalized by letting  $a_i = \frac{\alpha_i}{\alpha_0}$  and  $b_j = \frac{\beta_j}{\beta_0}$  for  $i = 1, \dots, m$ ,  $j = 0, \dots, n$ . Moreover, as in the rest of the paper, the model is assumed to have the same number of coefficients for  $y_1$  and  $y_2$  (i.e.,  $m = n$ ). Now,  $y_1$  can be represented as a function of past outputs and the current and past outputs of  $y_2$ :

$$y_1(k) = \sum_{i=1}^n -a_i y_1(k-i) + \sum_{j=0}^n b_j y_2(k-j). \quad (10)$$

The relationship of sensor outputs as provided by (10) will serve as the basis for determining if a sensor is faulty. Specifically, two error functions are defined. The first error function denoted by  $e_1(k)$ , is the difference in sensor 1's actual output  $\tilde{y}_1(k)$  and ideal output without the fault  $y_1(k)$ , i.e.,  $e_1(k) = \tilde{y}_1(k) - y_1(k)$ ; the error function  $e_2(k)$  is similarly defined. The second error function is the *cross-error* function denoted by  $e_{12}(k)$ . This is the difference between the observed output from sensor 1,  $\tilde{y}_1(k)$ , and the *estimated* output of sensor 1 (based on use of the ARX model using time-history data from sensor 1 and 2),  $\hat{y}_1(k)$ :

$$\hat{y}_1(k) = \sum_{i=1}^n -a_i \tilde{y}_1(k-i) + \sum_{i=0}^n b_i \tilde{y}_2(k-i). \quad (11)$$

The estimated output  $\hat{y}_1(k)$  is stable due to the fact that, unlike a classical observer, there is no feedback of the estimation error. Moreover,  $\hat{y}_1(k)$  is a one-step estimator with inputs solely based on the sensor measurements of a stable system.

The cross-error function can then be stated as:

$$e_{12}(k) = \tilde{y}_1(k) - \hat{y}_1(k) \quad (12)$$

$$e_{12}(k) = (y_1(k) + e_1(k)) - \left( \sum_{i=1}^n -a_i(y_1(k-i) + e_1(k-i)) + \sum_{i=0}^n b_i(y_2(k-i) + e_2(k-i)) \right) \quad (13)$$

$$e_{12}(k) = e_1(k) + \sum_{i=1}^n a_i e_1(k-i) - \sum_{i=0}^n b_i e_2(k-i). \quad (14)$$

It should be noted that (14) can be rewritten in a compact form using vector notation:

$$e_{12}(k) = \mathbf{a}^T \mathbf{e}_1(k) - \mathbf{b}^T \mathbf{e}_2(k)$$

where  $\mathbf{a}^T = [1, a_1, a_2, \dots, a_n]$ ,  $\mathbf{b}^T = [b_0, b_1, \dots, b_n]$ , and  $\mathbf{e}_1^T(k) = [e_1(k), e_1(k-1), \dots, e_1(k-n)]$ , with  $\mathbf{e}_2^T(k)$  similarly defined.

The cross-error function consists of a weighted summation of errors from a pair of sensors over a period of time. In general, the cross-error function gives zero values when there is no faults within the two sensors in a designated pair (i.e., they agree with each other) and gives non-zero values when any kind of faults ( $\mathbf{e} \neq \mathbf{0}$ ) exist. However, it is difficult to determine which sensor is faulty and what type of fault is present based solely on the cross-error function. In particular, if  $\mathbf{e}_1$  and  $\mathbf{e}_2$  represent the sensor measurement noise which is i.i.d. Gaussian distributed,  $e_{12}$  provides information of the noise characteristic of the sensors. For instance, if the Gaussian distribution (mean, variance) of the measurement noise of sensor 1 and 2 are  $(\mu_1, \sigma_1^2)$  and  $(\mu_2, \sigma_2^2)$  respectively, the distribution of  $e_{12}$  is Gaussian with mean equals to  $\sum_{i=0}^n (a_i \mu_1 - b_i \mu_2)$  and variance equals to  $\sum_{i=0}^n (a_i^2 \sigma_1^2 + b_i^2 \sigma_2^2)$ . If the error is due to a spike (with amplitude  $d$ ) in one of the sensors, say sensor 1, at time  $k-i$ , then  $\mathbf{e}_1$  will be a perfect impulse function with zero entries except for the component at  $k-i$  of magnitude  $d$  [Fig. 2(a)]. When sensor 2 is fault-free,  $e_{12}$  will be equal to  $a_i d$  at component  $k-i$  according to (14). As a result, a spike in sensor 1's output produces a cross-error function  $e_{12}$ , proportional to the ARX coefficient vector,  $\mathbf{a}$  [Fig. 2(b)]. Similarly, if a spike occurred in sensor 2 and no spike error occurred in sensor 1, the cross-error  $e_{12}$  will be proportional to the ARX coefficients  $\mathbf{b}$  [Fig. 2(c)]. When both sensors have spike errors, the cross-error function  $e_{12}$  will be equal to the sum of ARX coefficients  $\mathbf{a}$  and  $\mathbf{b}$  with appropriate proportionality [Fig. 2(d)]. This insight provides a method for identifying spikes in the cross-error function and to classify the sensor fault type (i.e., no faults, sensor 1 faulty, sensor 2 faulty and both sensors faulty). In addition, this method allows the fault to be identified in time. The detection performance depends on the baseline ARX coefficient vectors  $\mathbf{a}$  and  $\mathbf{b}$  which can be divided into 2 cases.

**Case1:**  $\mathbf{a} \neq c\mathbf{b}$  where  $c \in \mathbb{R}$ . For case 1, coefficient  $\mathbf{a}$  is not proportional to  $\mathbf{b}$ . As the cross-error function of a spike signal carries the characteristics of the ARX coefficients, detecting spike errors is similar to detecting a target signal

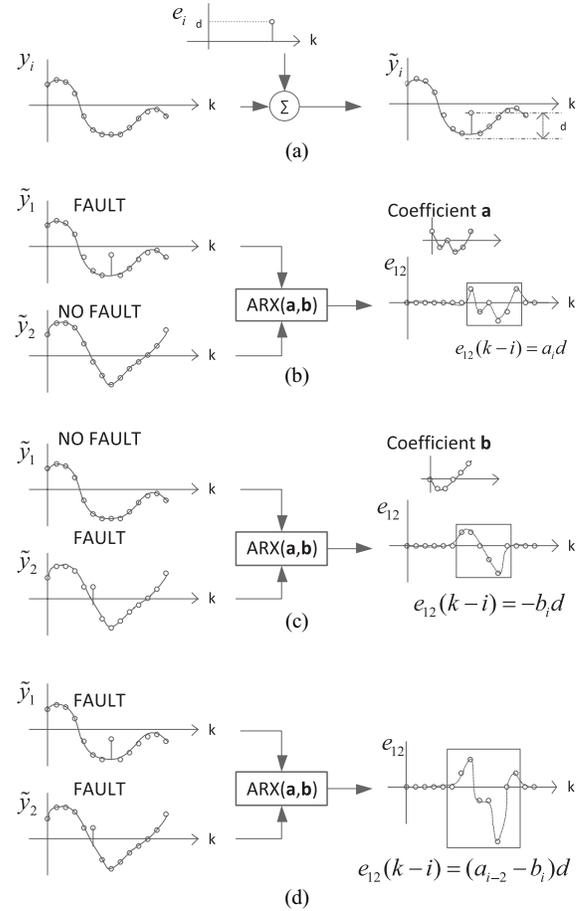


Fig. 2. Description of cross-error functions due to spike faults. (a) Superposition of a spike fault on a sensor output, (b) sensor 1 exhibits a spike fault, (c) sensor 2 exhibits a spike fault, and (d) both sensor 1 and 2 exhibit spike faults.

with a known waveform but with unknown amplitude and delay. In the telecommunication field, one popular method to detect signals under a binary hypothesis is to use matched filtering [26]. It convolves the received unknown signal with a filter which is the same as the target signal. When the target signal arrives, it will “match” with the filter and yield a high output. To identify spike errors, the cross-error function can be passed through two matched filters which have coefficients equal to  $\mathbf{a}$  and  $\mathbf{b}$ , respectively. The spikes can have either positive or negative magnitude; this information is unknown to the system. As a result, only the absolute value of a matched filter's output are considered.

Consider the cross-error function  $e_{12}(k)$  with measurement noise,  $\xi$ , which is additive from both sensors' noise processes. Assume there is a spike error on sensor 1 at time  $j$  ( $e_1(k) = 0$  for  $k \neq j$ ) and a spike error on sensor 2 at time  $l$  ( $e_2(k) = 0$  for  $k \neq l$ ), the cross-error function (14) becomes:

$$e_{12}(k) = \sum_{i=0}^n a_i e_1(k-i) - \sum_{i=0}^n b_i e_2(k-i) + \xi(k) \quad (15)$$

$$e_{12}(k) = a_{k-j} e_1(j) - b_{k-l} e_2(l) + \xi(k). \quad (16)$$

If we treat the ARX coefficients  $\mathbf{a}$  and  $\mathbf{b}$  as the matching filter, then the output of the matched filters, defined as matched error

function  $e_{12}^a$  and  $e_{12}^b$  respectively, at time  $k$  are:

$$e_{12}^a(k) = \left| \sum_{i=-\infty}^{\infty} a_{i-k} e_{12}(i) \right| \quad (17)$$

$$e_{12}^b(k) = \left| \sum_{i=-\infty}^{\infty} b_{i-k} e_{12}(i) \right| \quad (18)$$

where  $a_i = 0, b_i = 0$  when  $i < 0$  or  $i > n$  (i.e., outside the range of the filter).

Replace  $e_{12}(i)$  with (16) in (17):

$$e_{12}^a(k) = \left| \sum_{i=-\infty}^{\infty} a_{i-k} (a_{i-j} e_1(j) - b_{i-l} e_2(l) + \zeta(i)) \right| \quad (19)$$

$$e_{12}^b(k) = \left| \sum_{i=-\infty}^{\infty} (a_{i-k} a_{i-j} e_1(j) - a_{i-k} b_{i-l} e_2(l) + a_{i-k} \zeta(i)) \right|. \quad (20)$$

Equation (19) is dominated by the first two terms if the spike errors  $(e_1, e_2)$  have much larger magnitude than the measurement noise  $\zeta$ . Also, the first term is maximized when  $k = j$  and this maximum value is always bigger than the second term if  $\|a\|^2 \approx \|b\|^2$  and  $e_1(j) \approx e_2(l)$ . This means that if a spike error appears in sensor 1, it will be enlarged by matched filter  $a$  to result in a large value in  $e_{12}^a$  and suppressed by matched filter  $b$  to result in a small value in  $e_{12}^b$ . The reverse holds true for a spike error in sensor 2. As a result, the fault detection algorithm can detect peaks in the function and discriminate the corresponding spike error in sensor 1 and sensor 2, respectively. Moreover, the matched error function can locate exactly when the spike error occurred.

The following shows the computation complexity of the detection process. Let  $\nu$  be the number of ARX coefficients and  $N'$  be the number of data to be detected. From (11) and (12), the complexity for calculating the cross-error function for  $N'$  data is  $\mathcal{O}(\nu N')$ . Similarly, the complexity for calculation the matched filter outputs [(17) and (18)] for  $N'$  data is  $\mathcal{O}(\nu N')$ . Therefore, the overall complexity for the detection process is  $\mathcal{O}(\nu N')$ .

**Case2:**  $a \approx cb$  where  $c \in \mathbb{R}$ . When  $a$  and  $b$  can be related as  $a \approx cb$  for some constant  $c \in \mathbb{R}$ , the transfer function of  $\frac{Y_1}{Y_2}$  will be close to  $c$ . This means that the outputs of sensor 1 and sensor 2 are highly correlated to each other. For these systems, the proposed fault detection algorithm can detect a spike fault which has occurred in the sensor pair, but is not able to decide which sensor the spike fault belongs to. Intuitively, this is because the characteristic waveforms of the spike error in the cross-error function corresponding to sensor 1 and sensor 2 will have the same shape. Hence, the separating ability of the matched filter is lost. Although the magnitude and sign of the waveforms (which depend on the magnitude,  $d$ , and sign of the spike errors) are different and thus have different matched filter outputs, these outputs are not useful in identifying a faulty sensor since the sign and magnitude of the spike error is not known *a priori*. However, the detection algorithm does

at least know a spike error has occurred because the cross-error function still carries the characteristics of the ARX coefficients and is thus not equal to zero. In fact, the output of sensors being highly correlated is equivalent to having hardware redundancy. If a pair of sensors gives contradicting outputs, the algorithm is not able to tell which sensor is abnormal (without knowing which sensor is normal).

For the case  $a \approx cb$ , it is possible that the method fails to tell there is a spike error within the sensor pair when both sensors have spikes at the same time with appropriate magnitude. For example, if the system output at sensor 1 is proportional to the system output at sensor 2,  $y_1(k) = cy_2(k)$ , and the spike errors are  $e_1(k) = m$  and  $e_2(k) = cm$ , then the cross-error function  $e_{12}(k) = a^T e_1 - b^T e_2 = 0$ . Therefore, both matched filters will give zero results, which means no error is detected. In other words, if both sensors have faults such that their faulty outputs agree with each other, the method is unable to detect either error solely by evaluating the two sensor outputs.

Although there are limitations to the proposed detection algorithm, it should be mentioned that these limitations are neither common nor important cases. Having highly correlated sensors deployed in the same system is in general not cost effective because the output of one sensor is just a scale of the other. Also, the case where the algorithm is completely ineffective (i.e., when both sensors have spike errors at the same time instance with appropriate magnitude) has a very low probability of occurring because spike errors due to loose electrical contacts or electro-magnetic wave occur randomly.

Once spike errors are detected, the detection algorithm also provides a method to correct these errors. Recall that the coefficient  $a_0$  is normalized to 1. The cross-error function, therefore, is directly representing the magnitude of the spike error if signals used in prediction  $\hat{y}(k)$  are not corrupted by other spike errors and the sensor does not have background noise. Moreover, the matched error functions,  $e_{12}^a$  and  $e_{12}^b$ , reveal the position of the spike errors. As a result, the spike error can be eliminated easily.

#### IV. SIMULATION AND RESULTS

This section verifies the performance of the proposed ARX-based sensor fault detection method. These simulations will explore the accuracy of the methodology with respect to: the magnitude of the spike fault, the frequency of spike fault occurrence, and the type of sensor measurement.

##### A. Simulation Settings

A generic lumped mass dynamical system will be adopted to simulate a physical system such as a bridge, vehicle, etc. Fig. 3 presents a five degree-of-freedom (DOF) lumped mass dynamical system whose degrees-of-freedom are denoted as  $x_1(k)$  through  $x_5(k)$ . The masses,  $m_i$ , are connected via discrete springs and viscous dampers with spring constants,  $k_i$ , and damping coefficients,  $c_i$ , respectively. An external force,  $u_i(k)$ , is applied to each mass. The model parameters used in [14] are adopted in this paper. Each mass is set to be 1kg with each spring constant set to  $10kN/m$ .

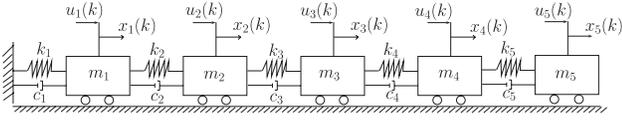


Fig. 3. Five DOF spring-mass-damper system for methodology validation.

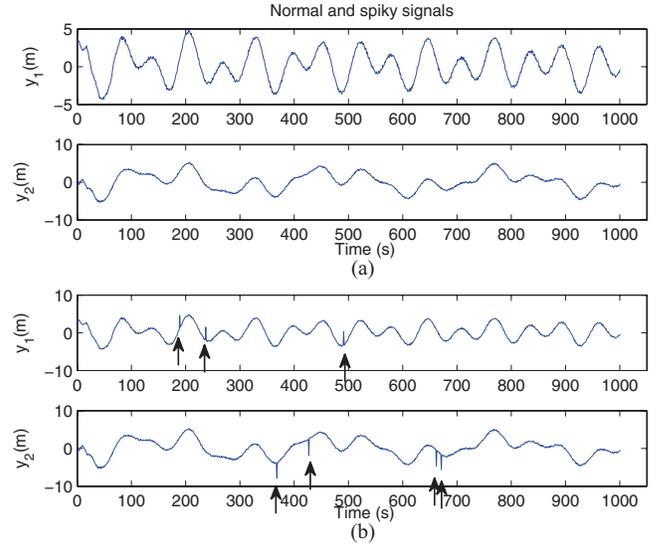
 TABLE I  
SIMULATION EXCITATIONS

Excitation	Training	Testing
Single-tone harmonic $\bar{\omega} = [10 \ 40] \text{rad/sec}$ $ U_{\max}  = [10 \ 13] \text{N}$	Solid line (Red)	Square
Double-tone harmonic $\bar{\omega}_1 = [10 \ 40] \text{rad/sec}$ $\bar{\omega}_2 = [100 \ 150] \text{rad/sec}$ $ U_{\max}  = [10 \ 13] \text{N}$	Dotted line (Blue)	Star
White Gaussian signal $\text{variance}(\sigma^2) = 100 \text{ N}$	Dash-dot line (Black)	Triangle

Similarly, each viscous damper has its damping coefficient set to 10.5 N sec/m. The natural frequencies of the dynamical system are: 4.52, 13.22, 20.84, 26.78, and 30.54 Hz. Each mode of vibration response is under-damped with damping ratios of 1.5%, 4.4%, 6.8%, 8.8%, and 10.1% for mode 1 (4.53 Hz) through 5 (30.54 Hz), respectively. The system is observed using either displacement sensors (i.e.,  $y_i = x_i$ ) or accelerometers (i.e.,  $y_i = \ddot{x}_i$ ) at each degree-of-freedom. The system is excited by three sources of excitation. For the first excitation, a harmonic load is applied identically to each degree-of-freedom defined by a single frequency ( $\bar{\omega}$ : uniformly distributed between 10 and 40 rad/sec) and with a random amplitude ( $|U_{\max}|$ : uniformly distributed between 10 and 13 N) and offset. For the second excitation, the external load is again harmonic but with two major frequencies ( $\bar{\omega}_1$ : uniformly distributed between 10 and 40 rad/sec and  $\bar{\omega}_2$ : uniformly distributed between 100 and 150 rad/sec). The amplitude associated with both tones is also random ( $|U_{\max}|$ : uniformly distributed between 10 and 13 N). The third excitation is a white noise source with a variance of 100 N identically applied to each degree-of-freedom. Table I summarizes the excitations used; all three are used for training ARX models and for spike fault detection validation. For each single simulation, the parameters of the excitation are randomly chosen and fixed through out that simulation. It should be noted that the excitation, even of the same type, are different (i.e., generated separately) for training and testing simulations.

For the training of the ARX pair-wise time series models, all three excitation types are utilized with a unique ARX model found for each excitation and measurement type. A validation analysis is done to determine the optimal number of coefficients [21]. Here, the model order is set to 25  $a_i$  coefficients and 25  $b_i$  coefficients. As previously mentioned, the coefficient  $a_0$  is set to 1.

In the simulations, the system is excited by the external excitation and the system response is measured at each degree-of-freedom,  $y_i$ . Random Gaussian noise is added to all measured responses to simulate a low level of sensor measurement noise. To emulate sensor spike faults, spikes are introduced to the


 Fig. 4. Response of  $m_1$  and  $m_2$  under a two tone harmonic excitation. (a) Fault-free displacement time history and (b) same displacement time histories with spike faults (40% of the maximum response amplitude) randomly introduced.

measurements of degree-of-freedom one and two with random sign and magnitude. For example, consider the fault free sensor response at  $m_1$  and  $m_2$  under the two-tone excitation as shown in Fig. 4(a). An example of the two time histories with simulated spike faults with the actual spikes denoted is shown in Fig. 4(b). Each time history in Fig. 4 has 4% noise (with respect to the signal variance) introduced. To determine the accuracy of the proposed sensor fault detection algorithm, the detection rate is used as a metric. Detection rate is the percentage of spikes to be correctly identified by the algorithm. This is equivalent to the percentage of true-positives; related metric would be the percentage of false-positives.

For illustrative purposes, consider the measured displacement of mass  $m_1$  and mass  $m_2$  denoted in Fig. 4(b). Using the ARX pair-wise model between  $y_1$  and  $y_2$ , the output at  $m_1$  is predicted by the output at  $m_2$ . The difference in the predicted and measured output,  $e_{12}$ , is plotted in Fig. 5(a). As can be seen, the spike faults in both outputs is creating non-trivial elevations in  $e_{12}$  in the vicinity of the actual faults. However, which sensor in the pair is experiencing the faults cannot be determined by the cross-error function alone. Rather, the use of the matched filters is needed to determine which sensor has the fault and where in time the faults are located. Use of (17) and (18) are used to determine  $e_{12}^a$  and  $e_{12}^b$  in Fig. 5(b) and (c), respectively. As can be seen, the convolved error function reveals when the spike faults occur. To identify the spikes, a threshold level is defined. Any disturbance that is larger than the threshold will be declared as a spike error of the corresponding sensor. For example,  $e_{12}^a$  exceeding a defined threshold corresponds to faults in  $y_1$  while exceedance of a threshold in  $e_{12}^b$  corresponds to faults in  $y_2$ . For all the simulations presented, the threshold is determined by the following steps. First, the cross-error function,  $e_{ij}$ , is calculated by the trained ARX model for a pair of normal (fault-free) sensors. The cross-error function is not expected to be equal to zero because there exists observation noise in the

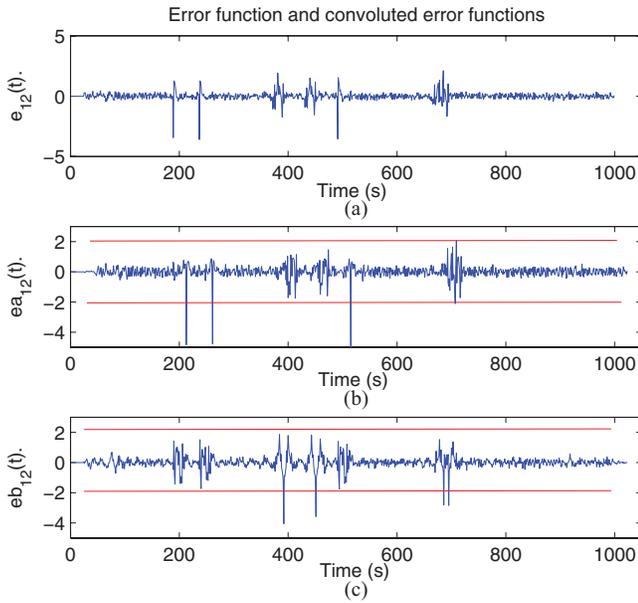


Fig. 5. (a) Cross-error function between sensor 1 and 2 corresponding to outputs presented in Fig. 4(b), (b) error function convoluted with coefficients  $a$ , and (c) error function convoluted with coefficients  $b$ .

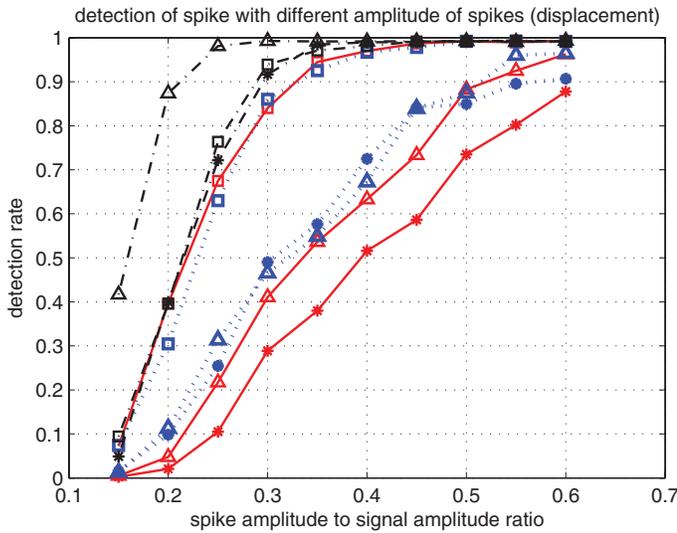


Fig. 6. Detection rate of spike faults versus different spike amplitudes when measuring displacement.

sensors and the ARX model will also have some prediction error. Afterward, the cross-error function is passed through the two matched filter and the variance of the  $e_{ij}^a$  and  $e_{ij}^b$  calculated. The threshold is set to be 6 times the standard deviation of the convoluted error function of the fault-free sensors such that the false alarm rate caused by the sensor measurement noise is almost zero.

### B. Simulation Results Under Various Scenarios

Simulations are carried out to evaluate the detection rate of spike errors versus different spike error amplitudes when measuring the acceleration and displacement response of the spring-mass-damper structure. These simulations considered the three different combinations of system excitations for both training and testing (validation). The legends of the different

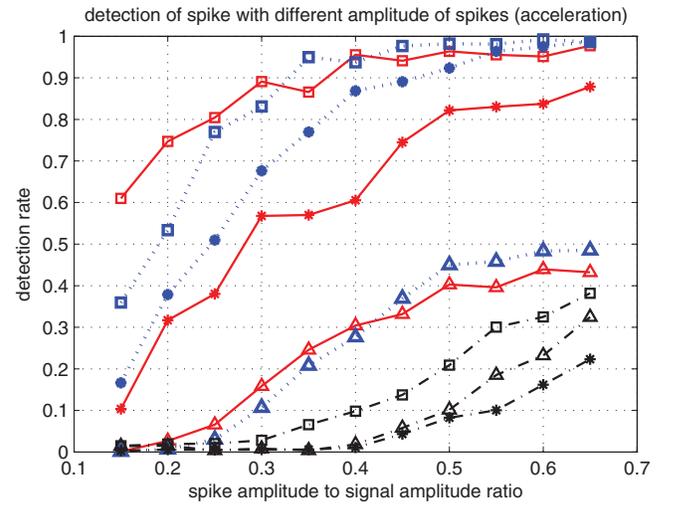


Fig. 7. Detection rate of spike faults versus different spike amplitudes when measuring acceleration.

excitation combinations for Figs. 6–10 are summarized in Table I. For example, if the system response to the single-tone sinusoidal excitation is used during training and white Gaussian noise excitation is used during performance testing, a red solid curve with triangle markers is used for the detection curve. In the simulations, the sensor noise is set to 10% of the sensor output variance. As can be seen for both displacement (Fig. 6) and acceleration (Fig. 7) outputs, the detection accuracy of the proposed method increases with the amplitude of the spikes. This is an obvious finding because smaller amplitude spike faults are more likely to be obscured by the sensor noise and thus more difficult to identify. Fig. 6 presents the detection rate when the sensors measure the displacement of the masses in the spring-mass-damper system. The results show that the detection algorithm performed well in all system input combinations. For large spike faults such as those whose amplitudes were 60% or greater than the signal amplitude, the sensor fault detection rate of the algorithm was high (>90%) regardless of the excitation used to train ARX models or when determining the sensor faults. However, for smaller spike amplitudes, the method accuracy exhibits some dependency on the nature of the excitation used to train the ARX models. In general, ARX models trained from white noise excitations provided the best baseline models. Especially when used to determine sensor spikes from similarly broadband excited time-history outputs, even small spikes (e.g., spikes white amplitudes only 25% of the signal amplitude) can be detected with detection rates in exceedance of 98%. Even for systems excited by harmonic loads, sensor faults with spike to signal amplitudes of 0.3 have detection rates of 90% or greater. In the absence of white noise excitations, the more broadband an excitation is, the better suited it is for the training of the ARX relationships between sensor pairs. For example, Fig. 6 shows that the ARX models created using the response of the system to the double tone harmonic excitation were more effective compared to those created using the single tone harmonic response.

When measuring displacement, the ARX models fall into Case 1 as discussed in Section III (i.e.,  $a \neq cb$  where

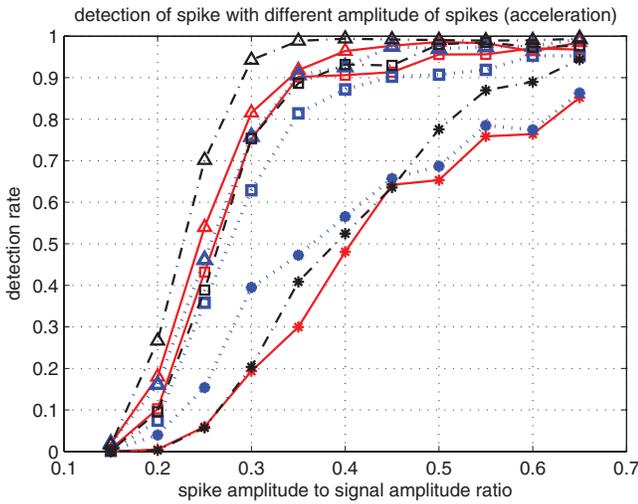


Fig. 8. Detection rate of spike faults versus different spike amplitudes when measuring acceleration. The viscous damping constant is decreased from 10.5–0.6 N sec/m.

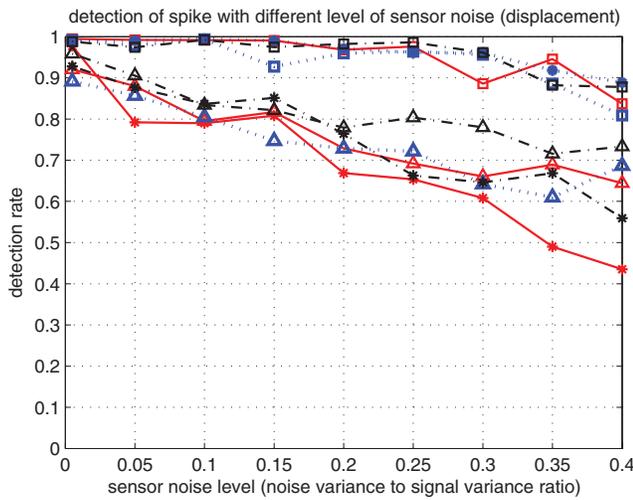


Fig. 9. Detection rate of spike faults versus different levels of sensor observation noise when measuring displacement. Spike is fixed at 30% of the maximum peak-to-peak amplitude of the sensor measurement.

$c \in \mathbb{R}$ ). In contrast, when measuring acceleration of the system, the system outputs are more correlated. This fact is confirmed when investigating the ARX model coefficients of relationships trained between various system outputs. With  $a \approx cb$ , the detection accuracy of the method decreases. For example, Fig. 7 presents the detection rate when the sensors measure the acceleration of the system. The performance of the sensor fault detection method in this setting is similar to the one measuring displacement except that the combinations in which Gaussian white noise is used as a training signal appear to perform worse with detection rates significantly lower than the other input combinations. When the system is excited by Gaussian white noise, its acceleration response exhibits the greatest correlation resulting in the lowest detection accuracy. While the detection algorithm is able to detect that faults exist in the sensor pairs, it is not able to classify which sensor the fault belongs to resulting in low detection rates.

To verify the deteriorated performance presented in Fig. 7 is due to strong correlation in the measured system response, the

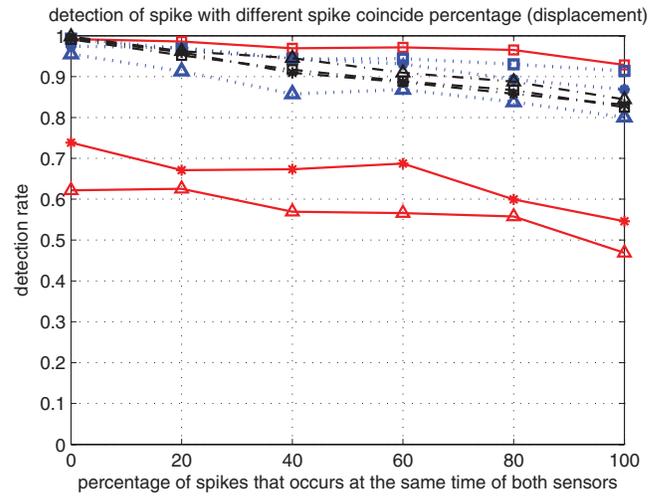


Fig. 10. Detection rate of spike faults versus different percentage of coincide spike error on both sensors.

viscous damping coefficient of the spring-mass-damper system is reduced from  $c_i = 10.5$  Nsec/m to  $c_i = 0.6$  N. All the other model parameters remain the same (i.e.,  $m_i = 1$  kg and  $k_i = 10$  kN/m). ARX models fitted to the new acceleration data shows  $a \neq cb$  resulting in a Case 1 system. The detection rate results are shown in Fig. 8, with the detection rate similar to that when using displacement outputs (Fig. 6).

Next, the accuracy of the sensor fault detection algorithm is quantified for noisy sensor measurements. Here, the original system is used (i.e.,  $c_i = 10.5$  N sec/m). Fig. 9 shows the detection rate of spike errors versus different levels of sensor observation noise while measuring the displacements of the system degrees-of-freedom. The magnitude of the spike errors are fixed at 30% of the maximum peak-to-peak amplitude of the sensor outputs. As can be seen, the detection accuracy deteriorates when the sensor noise exceeds 20% in all cases. When noise increases, the response of the spike error in the matched filter output will be increasingly dominated by noise. As a result, it is harder to set a good threshold for detection since the threshold must rise to be above the noise level.

Finally, the performance of the fault detection algorithm is investigated under scenarios of sensor faults occurring at the same time on both sensors in a pair. During this simulation, system displacements are measured with sensor noise fixed at 10% of the sensor output variance. Fig. 10 investigates the situation when spike faults happen at the same time on both sensors. The sign and magnitude of spikes are random in this simulation. The plot shows the detection rate as a function of the percentage of spike errors that happen at the same time on both sensors. Although the spike errors of different sensors have different characteristic waveforms and match to different matched filters, the coincident occurrence of spike errors still affects the detection performance. Due to faults occurring at the same time, the characteristic waveform on the cross-error function can be partially canceled out by the other spike with appropriate sign. As a result, the detection accuracy decreases (slowly) with the rate of coincidence. However, it should be noted that the overall performance of the algorithm is still high ( $>0.8$  detection rate for most cases).

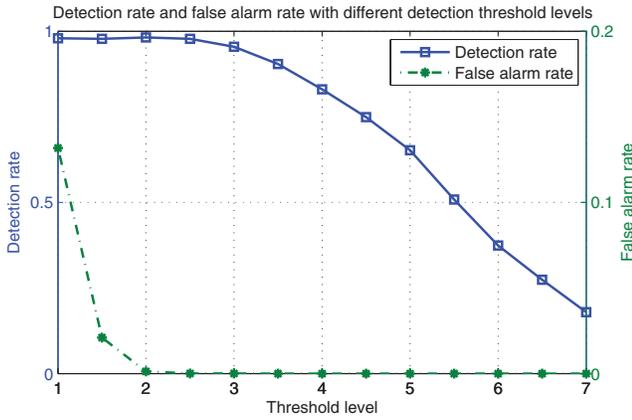


Fig. 11. Tradeoff between the sensor fault detection method's detection rate and false alarm rate with different threshold levels.

In all of the simulations presented, the threshold level is set to prevent false alarms. Hence, most of the errors in the detection algorithm are misses. In fact, higher detection rates can be achieved if false alarms can be tolerated by changing the threshold level. Fig. 11 illustrates the trade off between the detection rate and false alarm rate with different thresholds. The magnitude of the spike errors are fixed at 30% of the maximum peak-to-peak amplitude of the sensor outputs and the sensor noise is set to 10% of the sensor output variance. The rest of the simulation settings are the same as the simulation shown in Fig. 6. The threshold level is the multiplier of the variance of the convoluted error function of the fault-free sensors. As can be seen, lowering the threshold level can increase the detection rate while the false alarm rate also increases rapidly.

The rest of this section shows the performance of the proposed method on sensor data from a field-deployed WSN. A previous study was conducted focused on reducing the cost and installation complexity of monitoring systems on ships; wireless sensors were proposed by Swartz, *et al.* [27] as an alternative to traditional wired sensors. A monitoring system consisting of 20 wireless sensors were installed on a U.S. Navy ship in 2008. Moreover, a traditional wired hull monitoring system was installed in the ship alongside the WSN. During sea trials, some of the wireless sensors suffered from spike errors and excessive noise. Consider two wired sensors measuring hull strain shown in Fig. 12; these sensors were from the wired monitoring system and are generally fault-free (i.e., no spikes, etc.) Two wireless sensors in the WSN were collocated with these wired sensors and had spike errors and noise. These faulty sensors are used to illustrate the sensor fault detection algorithm in this paper. An ARX model of order  $n = 30$  is trained using 60 seconds (6000 points) of the spike free signals from the two wired sensors. With the trained ARX coefficients, 270 seconds (27000 points) of strain signals from the wireless sensors with spike errors are examined by the proposed method. Part of the signals (3000 points) of the two wireless sensors are shown in Fig. 13(a) (sensor s1) and (c) (sensor s2). The spikes detected by naked eye are marked by squares in Fig. 13(a) and (c). The output of the matched filters is shown in Fig. 13(b) and (d), and the automatically detected spikes are marked in stars. As can be

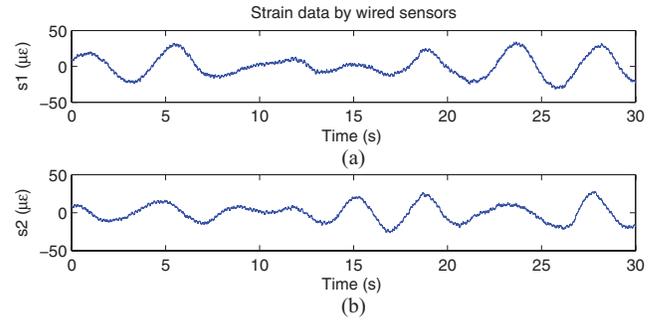


Fig. 12. Strain signals from wired sensors.

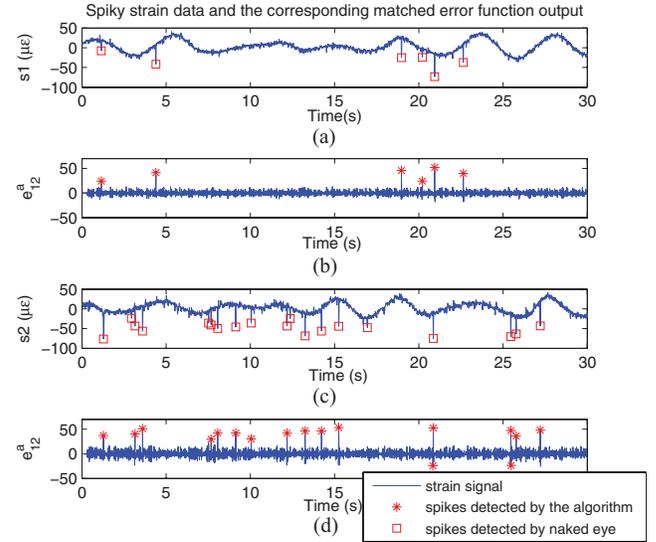


Fig. 13. (a) and (c) Strain signals from wireless sensors with spike errors and (b) and (d) the corresponding output of the matched error function.

seen, 21 out of 25 (84%) spikes were detected with 2 false alarms. For the examination of 27000 data points, 83.3% of spikes were detected (194 out of 233) and the false alarm rate is 0.04%. This level of accuracy is regarded as impressively high for actual field deployed wireless sensors with moderate amounts of noise.

## V. CONCLUSION

An ARX-based spike fault detection method which does not require the system-input information or the *a priori* establishment of reference sensors is proposed for LTI physical systems. The method is based on pair-wise relationships of sensors, and these relationships are learned online when the system is functioning normally. Moreover, the proposed method is able to identify all of the faulty sensors. Single-excitation systems are considered in this paper but the same approach can be used for multiple-excitation systems if the excitation inputs are known. Because the detection is done on a pairwise basis, it is well suited for WSNs in which power and communication resources are limited. In the simulations conducted, the detection accuracy exhibited dependency on the magnitude of the spikes, the sensor observation noise, the ARX coefficients and the threshold set for spike detection after the matched filter. The algorithm gives good performance; it only loses part of its effectiveness under situations where a

pair of sensors is highly correlated. However, such situations are either unlikely in practice or can be avoided when pairing sensors for execution of the proposed method.

Further research will be done to develop a method to separate other types of faults based on ARX coefficients. Moreover, the system itself was assumed healthy in the analysis. If the system fails gradually, then the system becomes time variant. In such cases, the ARX models could be updated periodically if it is known that the sensors are functioning properly (i.e., not faulty). If the trained system model deviates from normal, it means faults may have occurred in the system. Given a system fault, the detection algorithm will indicate some sensors have errors but it will know if these errors are due to spikes or not. Because the matched filters were designated to pick up spikes, other errors will simply appear as noise, or small fluctuations. However, further analysis is needed to determine whether the error is from a sensor fault or from a damaged system.

## REFERENCES

- [1] C. Lo, J. Lynch, and M. Liu, "Reference-free detection of spike faults in wireless sensor networks," in *Proc. 4th Int. Sympos. Resilient Control Syst.*, Aug. 2011, pp. 148–153.
- [2] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon, "Health monitoring of civil infrastructures using wireless sensor networks," in *Proc. 6th Int. Sympos. Sensor Netw. Inform.*, Apr. 2007, pp. 254–263.
- [3] J. Lynch, "An overview of wireless structural health monitoring for civil structures," *Philosoph. Trans. Royal Soc. A, Math., Phys. Eng. Sci.*, vol. 365, no. 1851, pp. 345–372, 2007.
- [4] T. Wark, P. Corke, P. Sikka, L. Klingbeil, Y. Guo, C. Crossman, P. Valencia, D. Swain, and G. Bishop-Hurley, "Transforming agriculture through pervasive wireless sensor networks," *IEEE Pervas. Comput.*, vol. 6, no. 2, pp. 50–57, Apr.–Jun. 2007.
- [5] J. Polastre, R. Szewczyk, A. Mainwaring, D. Culler, and J. Anderson, "Analysis of wireless sensor networks for habitat monitoring," in *Wireless Sensor Networks*, C. S. Raghavendra, K. M. Sivalingam, and T. Znati, Eds. New York, USA: Springer-Verlag, 2004, pp. 399–423.
- [6] P. Sikka, P. Corke, and L. Overs, "Wireless sensor devices for animal tracking and control," in *Proc. IEEE 29th Annu. Int. Conf. Local Comput. Netw.*, Nov. 2004, pp. 446–454.
- [7] J. Kim, J. Lynch, J. Lee, and C. Lee, "Truck-based mobile wireless sensor networks for the experimental observation of vehicle–bridge interaction," *Smart Mater. Struct.*, vol. 20, no. 6, p. 065009, May 2011.
- [8] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Commun. ACM*, vol. 43, no. 5, pp. 51–58, May 2000.
- [9] E. Chow and A. Willsky, "Analytical redundancy and the design of robust failure detection systems," *IEEE Trans. Automat. Control*, vol. 29, no. 7, pp. 603–614, Jul. 1984.
- [10] P. Frank, "Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy: A survey and some new results," *Automatica*, vol. 26, no. 3, pp. 459–474, 1990.
- [11] R. Isermann, "Supervision, fault-detection and fault-diagnosis methods—an introduction," *Control Eng. Pract.*, vol. 5, no. 5, pp. 639–652, 1997.
- [12] T. Menke and P. Maybeck, "Sensor/actuator failure detection in the Vista F-16 by multiple model adaptive estimation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 31, no. 4, pp. 1218–1229, Oct. 1995.
- [13] N. Piercy, "Sensor failure estimators for detection filters," *IEEE Trans. Automat. Control*, vol. 37, no. 10, pp. 1553–1558, Oct. 1992.
- [14] Z. Li, B. Koh, and S. Nagarajaiah, "Detecting sensor failure via decoupled error function and inverse input–output model," *J. Eng. Mech.*, vol. 133, no. 11, pp. 1222–1228, 2007.
- [15] R. Da and C. Lin, "Sensor failure detection with a bank of kalman filters," in *Proc. Amer. Control Conf.*, vol. 2. 1995, pp. 1122–1126.
- [16] T. Kobayashi and D. Simon, "Application of a bank of kalman filters for aircraft engine fault diagnostics," DTIC, Fort Belvoir, VA, Tech. Rep., 2003.
- [17] V. Riquebourg, D. Menga, M. Delafosse, B. Marhic, L. Delahoche, and A. Jolly-Desodt, "Sensor failure detection within the tbm framework: A markov chain approach," in *Proc. Inform. Process. Manag. Uncertain.*, vol. 8. 1991, p. 323.
- [18] X. Dai, Z. Gao, T. Breikin, and H. Wang, "Zero assignment for robust  $H_2/H_\infty$  fault detection filter design," *IEEE Trans. Signal Process.*, vol. 57, no. 4, pp. 1363–1372, Apr. 2009.
- [19] J. Chen, S. Kher, and A. Somani, "Distributed fault detection of wireless sensor networks," in *Proc. Workshop Dependabil. Issues Wirel. Hoc Netw. Sensor Netw.*, Sep. 2006, pp. 65–71.
- [20] L. Lennart, "System identification: Theory for the user," in *PTR Prentice Hall*, NJ, USA: Upper Saddle River, 1999.
- [21] C. Bishop, *Pattern Recognition and Machine Learning*. New York, USA: Springer-Verlag, 2006.
- [22] J. Burg, "Maximum entropy spectral analysis," in *Proc. 37th Annu. Int. Meeting*, 1967, pp. 1–18.
- [23] L. Trefethen and D. Bau III, *Numerical Linear Algebra*. Philadelphia, PA, USA: SIAM, 1997.
- [24] N. Andersen, "On the calculation of filter coefficients for maximum entropy spectral analysis," *Geophysics*, vol. 39, no. 1, pp. 69–72, 1974.
- [25] J. Lynch, A. Sundararajan, K. Law, A. Kiremidjian, and E. Carryer, "Embedding damage detection algorithms in a wireless sensing unit for operational power efficiency," *Smart Mater. Struct.*, vol. 13, no. 4, p. 800, 2004.
- [26] J. G. Proakis, *Digital Communications 5th edition*. New York, USA: McGraw-Hill, 2007.
- [27] R. Swartz, A. Zimmerman, J. Lynch, J. Rosario, T. Brady, L. Salvino, and K. Law, "Hybrid wireless hull monitoring system for naval combat vessels," *Struct. Infrastruct. Eng.*, vol. 8, no. 7, pp. 621–638, 2012.



**Chun Lo** (S'12) received the Bachelor's degree in electrical and computer engineering from the Hong Kong University of Science and Technology, Hong Kong, in 2008. He is currently pursuing the Ph.D. degree at the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, USA.

His current research interests include fault-detection algorithm and resource allocation in distributed systems, with emphasis on wireless sensor networks.



**Jerome P. Lynch** (M'04) received the Graduate degree from Stanford University, Stanford, CA, USA, where he received the M.S. and Ph.D. degrees in civil and environmental engineering in 1998 and 2002, respectively, and the M.S. degree in electrical engineering in 2003.

He is currently an Associate Professor of civil and environmental engineering with the University of Michigan, Ann Arbor, MI, USA, where he is a Faculty Member with the Department of Electrical Engineering and Computer Science. His current

research interests include wireless structural monitoring, feedback control, and damage detection algorithms.

Dr. Lynch was a recipient of the 2005 Office of Naval Research Young Investigator Award and the 2009 Presidential Early Career Award for Scientists.



**Mingyan Liu** (M'00–SM'11) received the Ph.D. degree in electrical engineering from the University of Maryland, College Park, MD, USA, in 2000.

She is currently a Professor with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, USA, where she joined in 2000. Her current research interests include optimal resource allocation, performance modeling and analysis, and energy-efficient design of wireless, mobile ad hoc, and sensor networks.

Dr. Liu was a recipient of the 2002 NSF CAREER Award, the University of Michigan Elizabeth C. Crosby Research Award in 2003, and the 2010 EECS Department Outstanding Achievement Award. She was on the Editorial Board of the IEEE/ACM TRANSACTIONS ON NETWORKING and the ACM TRANSACTIONS ON SENSOR NETWORKS.