Smart Structures and Systems, *Vol. 14, No. 1 (2014) 1-16*
DOI: http://dx.doi.org/10.12989/sss.2014.14.1.001

Spatio-temporal protocol for power-efficient acquisition wireless sensors based SHM

Nikola Bogdanovic*[1], Dimitris Ampeliotis[1], Kostas Berberidis[1],
Fabio Casciati[2] and Jorge Plata-Chaves[1]

*[1]Department of Computer Engineering and Informatics, University of Patras & C.T.I RU-8,
26500, Rio - Patra, Greece*
[2]Department of Civil Engineering and Architecture, University of Pavia,Via Ferrata 1, 27100 Pavia, Italy

(Received September 16, 2013, Revised April 25, 2014, Accepted June 30, 2014)

Abstract. In this work, we address the so-called sensor reachback problem for Wireless Sensor Networks, which consists in collecting the measurements acquired by a large number of sensor nodes into a sink node which has major computational and power capabilities. Focused on applications such as Structural Health Monitoring, we propose a cooperative communication protocol that exploits the spatio-temporal correlations of the sensor measurements in order to save energy when transmitting the information to the sink node in a non-stationary environment. In addition to cooperative communications, the protocol is based on two well-studied adaptive filtering techniques, Least Mean Squares and Recursive Least Squares, which trade off computational complexity and reduction in the number of transmissions to the sink node. Finally, experiments with real acceleration measurements, obtained from the Canton Tower in China, are included to show the effectiveness of the proposed method.

Keywords: spatio-temporal correlated data; sensor reachback; adaptive predictor; wireless sensor network; structural health monitoring

1. Introduction

Recent advances in microelectronics and wireless communications have enabled the development of low cost, low power devices that integrate sensing, processing and wireless communication capabilities. These devices, named sensor nodes, implement Wireless Sensor Networks (WSNs) which, as an alternative to the conventional wired systems, provide accurate and continuous monitoring of a phenomenon over some specific territory or structure. Typical applications of WSNs range from medical to military, and from home to industry. Within our interests, one outstanding application is Structural Health Monitoring (SHM), which consists in monitoring the behavior of civil structures, such as buildings, bridges, aircrafts and ships, during forced vibration testing or natural excitation (e.g. earthquakes, winds, live loading), as described by Lynch (2006).

∗Corresponding author, Ph. D. Student, E-mail: bogdanovic@ceid.upatras.gr

http://www.techno-press.org/?journal=sss&subpage=8 ISSN: 1738-1584 (Print), 1738-1991 (Online)

When accomplishing monitoring tasks such as SHM, one of the most fundamental issues is the so-called sensor reachback problem, which has received considerable attention (Barros *et al*. 2004). In more detail, the sensor reachback problem is related to the several difficulties appearing when transmitting the acquired sensor observations to a data-collecting node, often called sink node, which has increased processing and power consumption capabilities as compared to the sensor nodes. Firstly, the amount of data generated by the sensor nodes is immense, due to the fact that structural monitoring applications need to transfer relatively large amounts of dynamic response measurement data with sampling frequencies as high as 1000 Hz (Nagayama *et al*. 2010). Also, the number of sensor nodes may be very large. Next, the assumption that all sensors have direct, line-of-sight link to the sink does not hold in the case of these structures. Radio communication on and around structures made of concrete or steel components is usually complicated due to radio wave reflection, absorption, and other phenomena that result in poor received signal quality. Moreover, sensor nodes are frequently installed in partially- or completely- obscured areas, such as between girders. As a result, not all sensors may always have a channel to the sink of good enough quality and therefore, direct communication between each sensor node and the sink would consume all the energy stored in the batteries of the sensor nodes very quickly.

The problem of the limited energy that the sensor nodes can afford for data transmission can be alleviated by relying on recent advances in the field of cooperative communications. To reduce the amount of data required to be transmitted to the sink node, and therefore, to handle the problem associated with the massive data generated at the sensor nodes, the correlation among measurements by neighboring sensors can be leveraged (Barros and Servetto 2006). For instance, the data collected by the sensors on each span of a bridge are correlated since they are measuring the vibration of the same part of the physical structure. In addition, in some cases of bridge design, two adjacent spans are connected to a common anchorage, resulting in the data across the two spans to be correlated. Similarly, in the case of large buildings, it is natural to group the sensors of the several distinct parts of the building (e.g., floors). In all these cases, data compression approaches exploiting the correlation of the data, such as the Slepian-Wolf coding, offer the potential to greatly reduce the amount of information that needs to be transmitted (Stankovic *et al.* 2010). However, the Slepian-Wolf coding gives only information-theoretical bounds for data compression and it is quite difficult to be incorporated into a practical system.

In this work, we extend the work of Ampeliotis *et al.* (2012) and develop a communication protocol which, based on a Time Division Multiple Access (TDMA) strategy and adaptive filtering techniques such as Least Mean Squares (LMS) and Recursive Least Squares (RLS), aims at overcoming the difficulties associated with the sensor reachback problem. To do so, the protocol allows the sink node to keep an exact replica of the adaptive filters that, at each node, exploit the spatial and temporal correlations among sensor measurements to predict the current measurement from their own past measurements as well as past measurements obtained by their neighbors. Specifically, in the designed protocol each node is assigned a time slot that is divided into two sub-slots. During the first sub-slot, each sensor acquires a new measurement and computes the prediction error of its associated adaptive filter. If the prediction error is small enough (i.e., below a predefined threshold), then during the first sub-slot the considered sensor node sends the output of its filter to its neighbors, so that they can use this value as input for the prediction filters they operate. In the opposite case, i.e., when the prediction error is not that small, the node updates its filter (i.e., using an LMS or RLS update step) and sends its actual measurement to its neighbors. Afterwards, if the prediction is not accurate, since a Multiple Input Single Output (MISO) channel is known to result in energy savings as compared to the Single Input Single Output (SISO) case

(Cui *et al*. 2004), all the nodes which collaborated during the first sub-slot will form a MISO channel to simultaneously transmit the current measurement to the sink node. This way, with the aim of having an exact replica of all the filters implemented by the cooperating sensor nodes, the sink node is able to incorporate the transmitted measurement to the input of the aforementioned filters and update the filter associated with the considered sensor node.

After deriving the communication protocol, both LMS-type and RLS-type implementations of the new technique have been tested extensively via real acceleration measurements from the Canton Tower. For both kinds of implementations, it turns out that the proposed strategy may offer considerable savings in transmitted energy, especially if an appropriate selection of the cooperating sensor nodes has been undertaken. Depending on the adaptive filtering technique, LMS or RLS, it has been shown that different tradeoffs between computational complexity and savings in transmitted energy can be achieved.

The remainder of this paper is organized as follows. Section 2 is devoted to problem formulation. The proposed protocol is explained in more detail in Section 3. The results obtained by applying the protocol on real acceleration measurements from the Canton Tower in China during an earthquake are presented in Section 4. Section 5 concludes the paper and provides a discussion about possible future extensions.

2. Formulation of the problem

Let us consider a dense wireless sensor network consisting of N nodes, deployed on a civil structure that we wish to monitor. Consider also that node n $(n = 1,2 \dots, N)$ has K neighbors, in the sense that they are close enough to node n so that wireless communication with low power can be accomplished. We will denote the neighbors of node n as $k_{n,1}, k_{n,2}, \dots, k_{n,K}$. Each sensor node n, at some discrete time instant t, acquires the measurement $y_{n,t}$ which is related to an event that takes place in the area where the wireless sensor network has been deployed. As an example, $y_{n,t}$ may represent an acceleration measurement, that captures oscillations of the structure. Let us define the vectors of m past measurements of each sensor node n as

$$\boldsymbol{y}_{n,t} = [y_{n,t-1} \quad y_{n,t-2} \quad \cdots \quad y_{n,t-m}]^T \qquad n = 1,2, \dots, N \tag{1}$$

Also, let us define the stacked vectors

$$\boldsymbol{u}_{n,t} = \left[\boldsymbol{y}_{n,t}^T \quad \boldsymbol{y}_{k_{n,1},t}^T \quad \boldsymbol{y}_{k_{n,2},t}^T \cdots \quad \boldsymbol{y}_{k_{n,K},t}^T\right]^T \qquad n = 1,2, \dots, N \tag{2}$$

which represent the past m measurements of all sensor nodes in the neighborhood of node n. Consider now the correlation matrices defined as

$$\mathbf{R}_n = E[\mathbf{u}_{n,t}\mathbf{u}_{n,t}^T] \ , \qquad n = 1,2, \dots, N \tag{3}$$

Clearly, if matrices $\mathbf{R}_n$ are diagonal, the sensor measurements within all neighborhoods are correlated, neither in time nor in space. In contrast, if the matrices $\mathbf{R}_n$ are only block-diagonal with block size m, the measurements are correlated in time but spatially uncorrelated. In this work, we will focus on the general case where $\mathbf{R}_n$ are of a general form, implying that the sensor measurements are correlated both in time and in space.

Thus, we are interested in deriving a network protocol able to transmit the sensor measurements to the data-collecting node in an energy-efficient way. Such a protocol should take advantage of the aforementioned correlations, in order to reduce the number of transmissions toward the sink. Furthermore, the protocol should provide accuracy guarantees for the received data.

3. A TDMA based cooperative protocol

3.1 Predictors and correlation of measurements

As mentioned in the previous section, we are interested in deriving an energy-efficient protocol for the transmission of the measurements to the data-collecting node. To this end, if we were able to reduce the number of information bits that need to be transmitted, this would have a considerable effect on the energy spent by the data-gathering process. Such a reduction in the number of information bits that need to be transmitted can be accomplished if we take advantage of the correlations among the measurements. In particular, if we are able to identify and send only the "new" information that lies in the measurements, then significant energy savings would emerge. A way for identifying such "new information" employs the notion of signal predictors.

The nature of the observed phenomenon makes the measurements $y_{n,t}$ predictable, at least to some extent. In particular, if the data-collecting node had knowledge of previous measurements acquired by sensor n (and possibly previous measurements of other nodes in the vicinity of node n) , then it could compute an estimate of $y_{n,t}$. This estimate, of course, corresponds to information already known to the data-collecting node. In principle, we can distinguish between two different types of prediction functions, namely, (a) one that does not change with time, which implies that the correlation mechanism is constant or stationary, and (b) a time-varying prediction function, implying that the statistics of the signals measured by the nodes of the network have a dynamic behavior. Assuming the process to be stationary, the prediction function can be realized as a linear filter with coefficients obtained by minimizing the mean-squared error between the measurements $y_{n,t}$ and their predicted values.

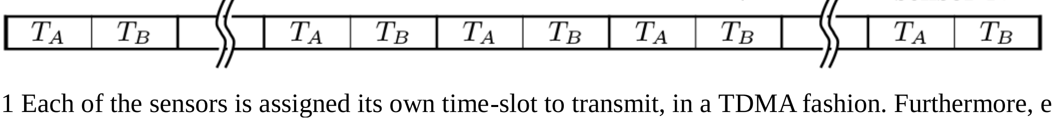


Fig. 1 Each of the sensors is assigned its own time-slot to transmit, in a TDMA fashion. Furthermore, each time-slot is divided into two sub-slots. During the first sub-slot of duration T_A, each sensor n transmits to its k_n neighbors. During the second sub-slot of duration T_B, node n and its neighbors transmit to the sink node in a cooperative fashion

However, in most real world applications the observation processes are non-stationary since their statistical characteristics are changing in time. As a result, the optimal coefficients of the predictor are changing in time as well. In order to track these changes, a practical approach is to iteratively estimate them by updating previous filter coefficients as it is done in adaptive filters (Sayed 2008). Such an approach offers the additional benefit that the data-collecting node does not need to know the statistics of the underlying process. Rather these statistics are in effect estimated by the adaptive filter.

3.2 A simple cooperative TDMA protocol

As already mentioned in the introduction, another approach for reducing the energy required to transmit data relies on the concept of cooperative communications. In particular, in cooperative communications, a number of accurately synchronized nodes transmit data concurrently so that the system resembles a transmitter with multiple antennas. During the previous phase, the nodes have agreed upon the data that will be sent. In effect, benefits similar to Multiple Input Multiple Output (MIMO) systems can be achieved (Cui *et al*. 2004), hence the terms virtual MIMO or distributed MIMO are often used alternatively to denote cooperative communication systems.

For illustration purposes, let us consider now a straightforward cooperative communication protocol for the problem at hand, in which correlation among the measurements acquired by the nodes of the network is not taken into account. According to this protocol, each sensor node is assigned its own time-slot in order to transmit information, in a Time Division Multiple Access (TDMA) fashion. Cooperative communications can be incorporated into this protocol, by dividing each time-slot into two sub-slots as depicted in Fig. 1. During the first sub-slot of duration T_A, each sensor n transmits its estimated (or observed) value to its K neighbors. During the second sub-slot of duration T_B, node n and its neighbors transmit to the sink node in a cooperative fashion. In such a scenario, both the Amplify and Forward (AF) as well as the Decode and Forward (DF) methods (Hong *et al*. 2007) can be adopted.

3.3 Cooperative TDMA exploiting correlation

Consider now an extension of the aforementioned protocol, where the correlation of the measurements is taken into account. Since the measurements may be correlated both in time and in space, the idea of using past measurements acquired by node n as well as past measurements from nearby sensor nodes in order to predict new measurements of node n seems well justified. This fact can be used to save a noticeable percentage of the transmissions to the sink node, in the case where the sink node can itself predict the required measurements within some predefined accuracy. Thus, let each sensor node n keep a time varying prediction filter $\mathbf{f}_{n,t}$ as well as a data vector

$$\widetilde{\boldsymbol{u}}_{n,t} = \left[\widetilde{\boldsymbol{y}}_{n,t}^T \quad \widetilde{\boldsymbol{y}}_{k_{n,1},t}^T \quad \widetilde{\boldsymbol{y}}_{k_{n,2},t}^T \cdots \quad \widetilde{\boldsymbol{y}}_{k_{n,K},t}^T\right]^T \tag{4}$$

so that the output of the filter, defined as

$$\hat{y}_{n,t} = \mathbf{f}_{n,t}^T \cdot \widetilde{\boldsymbol{u}}_{n,t} \tag{5}$$

is an approximation of the actual measurement $y_{n,t}$ obtained by sensor n at time t. In particular, $\hat{y}_{n,t}$ is a prediction of the actual measurement $y_{n,t}$. In the above expressions, we have used the vectors

$$\tilde{\boldsymbol{y}}_{n,t} = [\tilde{y}_{n,t-1} \quad \tilde{y}_{n,t-2} \quad \cdots \quad \tilde{y}_{n,t-m}]^T \\ n = 1,2,\dots,N \tag{6}$$

to represent approximate versions of the past m measurements obtained by sensor n. Thus, vectors $\tilde{\boldsymbol{u}}_{n,t}$ and $\mathbf{f}_{n,t}$ have dimensions $m \cdot K \times 1$. Let us now define a binary variable $b_{n,t}$ according to the prediction error, as

$$b_{n,t} = \begin{cases} 0 & \text{if} \quad |\hat{y}_{n,t} - y_{n,t}| \leq e \\ 1 & \text{if} \quad |\hat{y}_{n,t} - y_{n,t}| > e \end{cases} \tag{7}$$

where e denotes a small positive constant. The approximate measurements $\tilde{y}_{n,t}$ are defined as,

$$\tilde{y}_{n,t} = \begin{cases} \hat{y}_{n,t} & \text{if} \quad b_{n,t} = 0 \\ y_{n,t} & \text{if} \quad b_{n,t} = 1 \end{cases} \tag{8}$$

Based on the above definitions, the protocol of each sensor node n can be seen in Table 1. At a time instant t, each sensor acquires its new measurement $y_{n,t}$ and starts a synchronized loop to track the N time-slots that will follow. As seen from Table 1, node n is active in two cases: (a) when the current slot index s is equal to n, and (b) when the current slot index s is equal to the index of any of its neighbors. In case (a), the node computes the output of its prediction filter and compares it to the actual measurement $y_{n,t}$. Thus, it computes the binary variable $b_{n,t}$ that determines whether the prediction was accurate or not. In the case where the prediction was not accurate, the prediction filter is updated using an adaptive algorithm. Table 1 summarizes the steps followed in order to perform the update of the filter, for the cases of the LMS and the RLS update algorithms. As a general rule, the LMS algorithm should be used when reduced computational complexity is required. On the other hand, one should opt for the RLS algorithm in the case where the statistics of the measurements change abruptly with time, given that the computational complexity requirements can be met. Also, as it will be shown in the experiments' section, RLS performs better when adaptation stalls and restarts very often during operations, as it is the case with the suggested technique. Regardless of the algorithm used for the update, $y_{n,t}$ is used as a desired response signal. Then, the sensor node n computes $\tilde{y}_{n,t}$, which is either the output of the prediction filter (accurate prediction) or the actual measurement (inaccurate prediction). Thus, sensor n updates its input vector $\tilde{\mathbf{u}}_{n,t+1}$ and sends $\tilde{y}_{n,t}$ and $b_{n,t}$ to its neighbors. Finally, $\tilde{y}_{n,t}$ is sent to the sink node only if the prediction was inaccurate, otherwise the sink node is able to compute $\tilde{y}_{n,t}$ using a prediction filter. In case (b), i.e. when a neighbor of n is active, node n listens for the transmitted values $\tilde{y}_{s,t}$ and $b_{s,t}$. It then updates its input vector $\tilde{\mathbf{u}}_{n,t+1}$ with the received value $\tilde{y}_{s,t}$ and, in the sequel, helps its neighbor transmit to the sink by relaying $\tilde{y}_{s,t}$ if $b_{s,t}$ was 1.

The protocol followed by the sink node is depicted in Table 2. At each time instant, the sink node also executes a loop so as to track the N time-slots, in a synchronized fashion. For the first T_A seconds of each slot, the sink node is inactive because sensor-to-sensor communication takes place. At the following T_B seconds however, the sink node is receiving the measurement $\tilde{y}_{s,t}$ of the node assigned to the current slot. Of course, in the case where the prediction at node s was accurate, such a message will not be transmitted. Thus, the sink node must implement a procedure to detect such "empty" messages. The result of the detection process is a binary variable $\hat{b}_{s,t}$ which will be equal to $b_{s,t}$ in the case where the detection is correct. In the sequel, the sink node is able to compute $\tilde{y}_{s,t}^{(S)}$, (that is, a copy of $\tilde{y}_{s,t}$ at the sink) either as the output of a local prediction filter, i.e.

$$\tilde{y}_{s,t}^{(S)} = \mathbf{f}_{s,t}^{(S)^T} \cdot \tilde{\boldsymbol{u}}_{s,t}^{(S)} , \tag{9}$$

in the case where $\hat{b}_{s,t} = 0$ (accurate prediction) or by setting it equal to the received measurement $\tilde{y}_{s,t}$ (inaccurate prediction). In the case of inaccurate prediction, the sink node must use the same adaptive algorithm as the sensor s to update its local prediction filter for sensor s, so that the two filters are equal (of course, if all channels are error free). Finally, the sink node must update the input vectors of all the prediction filters affected by $\tilde{y}_{s,t}$, that is the prediction filter for node s and the local prediction filters of all its neighbors.

It can be verified by the above description of the proposed data collection protocol, that in the case where all channels are error-free, the reconstructed sequences $\tilde{y}_{n,t}^{(S)}$ at the sink node satisfy the distortion criterion

$$\max_{n,t} | \tilde{y}_{n,t}^{(S)} - y_{n,t} | \leq e . \tag{10}$$

In fact, the maximum allowed distortion parameter e offers a trade-off between accurate reconstruction of the measurements by the sink node, and the number of transmissions required. Also, some other factors, such as the degree to which the measured signals can be predicted and the specific characteristics of the adaptive algorithm used to update the coefficients of the prediction filters, may influence the performance of the proposed protocol.

3.4 Cooperative neighborhood selection

Firstly, let us analyze the merits and drawbacks of having cooperation among the sensor nodes. For a given node n, cooperation with K neighbors actually requires K additional transmissions to these neighbors at each time instant. Although the energy cost of the additional inter-node transmissions can be low due to their proximity, one should also take into account the channel quality between the cooperating nodes which may introduce additional distortion to the data being sent.

On the other hand, the gains can overcome the cooperation costs in case that the number of transmissions toward the sink is reduced due to the exploitation of high spatial correlation among the measurements in the cooperating neighborhood. Certainly, the cooperation gains are not the same for all the nodes. In fact, the relation between the values of temporal correlation among the measurements of node n on one side, and the values of their spatial correlation with the measurements of the cooperating nodes should determine how beneficial the cooperation may be. Furthermore, an additional benefit can be obtained once the transmissions toward the sink are required. As previously explained, the cooperating nodes may simultaneously transmit to the sink node; thus forming MISO channel and improving energy-efficiency.

Table 1 The protocol executed by sensor node *n*

```
Initialize $\mathbf{f}_{n,0}$, $\tilde{\boldsymbol{u}}_{n,0}$ and $e$
Initialize $\mu$            (If LMS update is used)
Initialize $\lambda, \mathbf{P}_{\mathbf{n},-\mathbf{1}} = \delta^{-1}\mathbf{I}$ (If RLS update is used)
For $t = 0$ to $+\infty$
  Acquire the measurement $y_{n,t}$
  For $s = 1$ to $N$
    If $s = n$ then
      $\hat{y}_{n,t} = \mathbf{f}_{n,t}^{T}\tilde{\boldsymbol{u}}_{n,t}$
      $b_{n,t} = \begin{cases} 0 & \text{if } |\hat{y}_{n,t} - y_{n,t}| \le e \\ 1 & \text{if } |\hat{y}_{n,t} - y_{n,t}| > e \end{cases}$
      $\tilde{y}_{n,t} = \begin{cases} \hat{y}_{n,t} & \text{if } b_{n,t} = 0 \\ y_{n,t} & \text{if } b_{n,t} = 1 \end{cases}$
      If $b_{n,t} = 1$
        $\mathbf{f}_{n,t+1} = \mathbf{f}_{n,t} + \mu\,(y_{n,t} - \hat{y}_{n,t})\,\tilde{\boldsymbol{u}}_{n,t}$      (LMS Update)
                                   OR
        $\mathbf{k}_{\mathbf{n},\mathbf{t}} = \frac{\lambda^{-1}\mathbf{P}_{\mathrm{n,t-1}}\tilde{\boldsymbol{u}}_{n,t}}{1+\lambda^{-1}\tilde{\mathbf{u}}_{\mathbf{n},\mathbf{t}}^{\mathbf{T}}\mathbf{P}_{n,t-1}\tilde{\mathbf{u}}_{\mathbf{n},\mathbf{t}}}$
        $\xi_{n,t} = y_{n,t} - \hat{y}_{n,t}$                                     (RLS Update)
        $\mathbf{f}_{n,t+1} = \mathbf{f}_{n,t} + \mathbf{k}_{\mathbf{n},\mathbf{t}}\xi_{n,t}$
        $\mathbf{P}_{\mathbf{n},\mathbf{t}} = \lambda^{-1}\mathbf{P}_{\mathbf{n},\mathbf{t}-\mathbf{1}} - \lambda^{-1}\mathbf{k}_{\mathbf{n},\mathbf{t}}\tilde{\mathbf{u}}_{\mathbf{n},\mathbf{t}}^{\mathbf{T}}\mathbf{P}_{n,t-1}$
      End
      Update $\tilde{\boldsymbol{u}}_{n,t+1}$ using $\tilde{y}_{n,t}$
      Send $\tilde{y}_{n,t}$ and $b_{n,t}$ to the neighbors ($T_A$ sub-slot)
      If $b_{n,t} = 1$
        Send $\tilde{y}_{n,t}$ to the sink ($T_B$ sub-slot)
      End
    Elseif $s \in \{k_{n,1}, k_{n,2}, \dots, k_{n,k}\}$
      Listen for $\tilde{y}_{s,t}$ and $b_{s,t}$ ($T_A$ sub-slot)
      Update $\tilde{\boldsymbol{u}}_{n,t+1}$ using $\tilde{y}_{s,t}$
      If $b_{s,t} = 1$
        Send $\tilde{y}_{s,t}$ to the sink ($T_B$ sub-slot)
      End
    Else
      Sleep ($T_A + T_B$ seconds)
    End
  End
End
```

Table 2 The protocol executed by the sink node

```
Initialize $\mathbf{f}_{s,0}^{(S)}$, $\widetilde{\boldsymbol{u}}_{s,0}^{(S)}$  $(s = 1,2,\dots,N)$
Initialize $\mu$                (If  LMS update is used)
Initialize $\lambda, \mathbf{P}_{\mathbf{s},-\mathbf{1}}$   $(s = 1,2,\dots,N)$   (If  RLS update is used)
For $t = 0$ to $+\infty$
  For $s = 1$ to $N$
    Sleep ($T_A$ seconds)
    Listen for $\tilde{y}_{s,t}$ ($T_B$ sub-slot)
    $\hat{b}_{s,t} = \begin{cases} 0 & \text{if } \tilde{y}_{s,t} \text{ was not detected} \\ 1 & \text{if } \tilde{y}_{s,t} \text{ was detected} \end{cases}$
    If $\hat{b}_{s,t} = 0$
      $\tilde{y}_{s,t}^{(S)} = \mathbf{f}_{s,t}^{(S)^T} \widetilde{\boldsymbol{u}}_{s,t}^{(S)}$
      $\mathbf{f}_{\mathbf{s},\mathbf{t}+\mathbf{1}}^{(S)} = \mathbf{f}_{\mathbf{s},\mathbf{t}}^{(S)}$
    Else
      $\tilde{y}_{s,t}^{(S)} = \tilde{y}_{s,t}$
      $\mathbf{f}_{\mathbf{s},\mathbf{t}+\mathbf{1}}^{(S)} = \mathbf{f}_{\mathbf{s},\mathbf{t}}^{(S)} + \mu\,(\tilde{y}_{s,t} - \mathbf{f}_{s,t}^{(S)^T} \widetilde{\boldsymbol{u}}_{s,t}^{(S)})\widetilde{\boldsymbol{u}}_{s,t}^{(S)}$        (LMS Update)
                                    OR
      $\mathbf{k}_{\mathbf{s},\mathbf{t}} = \frac{\lambda^{-1}\mathbf{P}_{\mathrm{s,t-1}}\widetilde{\boldsymbol{u}}_{s,t}^{(S)}}{1+\lambda^{-1}\widetilde{\boldsymbol{u}}_{s,t}^{(S)^T}\mathbf{P}_{s,t-1}\widetilde{\boldsymbol{u}}_{s,t}^{(S)}}$
      $\xi_{s,t} = \tilde{y}_{s,t} - \mathbf{f}_{s,t}^{(S)^T} \widetilde{\boldsymbol{u}}_{s,t}^{(S)}$        (RLS Update)
      $\mathbf{f}_{\mathbf{s},\mathbf{t}+\mathbf{1}}^{(S)} = \mathbf{f}_{\mathbf{s},\mathbf{t}}^{(S)} + \mathbf{k}_{\mathbf{s},\mathbf{t}}\xi_{s,t}$
      $\mathbf{P}_{\mathbf{s},\mathbf{t}} = \lambda^{-1}\mathbf{P}_{\mathbf{s},\mathbf{t}-\mathbf{1}} - \lambda^{-1}\mathbf{k}_{\mathbf{s},\mathbf{t}}\widetilde{\boldsymbol{u}}_{s,t}^{(S)^T}\mathbf{P}_{s,t-1}$
    End
    Update $\widetilde{\boldsymbol{u}}_{s,t+1}^{(S)}$ using $\tilde{y}_{s,t}^{(S)}$
    For $i$=1 to $K$
      Update $\widetilde{\boldsymbol{u}}_{k_{s,i},t+1}$ using $\tilde{y}_{s,t}^{(S)}$
    End
  End
End
```

Not surprisingly, in the simulation section it turns out that choosing the suitable cooperating neighborhood, in terms of its size and the actual nodes involved, plays a significant role in enhancing the performance of the protocol. Therefore, the optimization of the cooperating neighborhood requires (a) the knowledge of all channels among the nodes (including the sink) and (b) the knowledge of the auto- and cross- correlation functions of all nodes. Regarding the former issue, in a practical system, all the involved channels may be estimated during a training period in which all nodes participate. Initially, all the nodes would send the training sequence to the sink and all other nodes. Afterwards, the nodes would also transmit to the sink the sequences that are received from all other nodes. Consequently, the sink could estimate all the involved channels in a centralized manner. However, here we focus on the criterion (b). Hence, in Section 4.2., we show a simple neighborhood selection procedure based on the correlation functions of all the nodes.

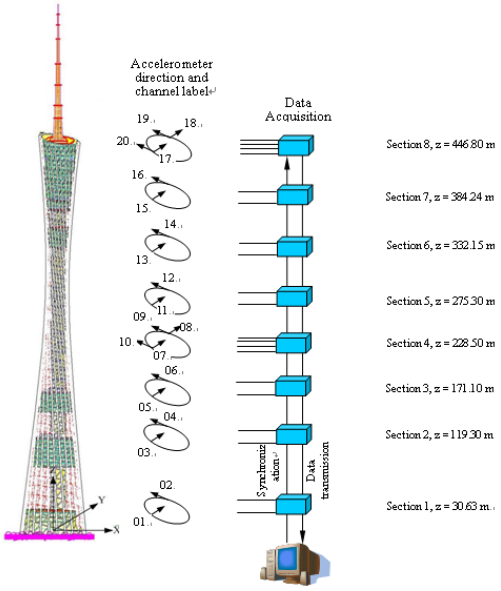


Fig. 2 The distribution of accelerometers along the tower

4. Numerical results

In order to demonstrate the effectiveness of the proposed algorithms, we have performed extensive experiments with real data. Although the dataset has not been designed for our protocol, the experiments show certain performance benefits of cooperation among the nodes. In particular, the acceleration measurements from the Canton Tower obtained during an earthquake have been used in order to present these cooperation gains. Toward this aim, we examine the number of transmissions toward the sink as a function of the maximum allowed absolute distortion, i.e., the value of the parameter e.

4.1 The Canton tower monitoring system

The Canton Tower (the Guangzhou TV and Sightseeing Tower) was constructed in 2010 in Guangzhou, China. It has already attracted the interest of several researchers (Casciati *et al*. 2009). It is a super-tall structure with a height of 610 m. On the top level of the tower at height of 454 m an antennary mast is mounted with 164 m height (see Fig. 2).

The tower is a tube-in-tube structure; the outer tube is made of steel and the inner one is a reinforced concrete tube. The two tubes are linked together by 36 floors and 4 levels of connection girders. The underground part of the tower is 10 m height and consists of 2 floors with plan dimensions of 167 m by 176 m. The outer tube is shaped by concrete-filled-tube (CFT) columns, spaced in an oval shape, inclined vertically, and connected by hollow steel rings and braces. The oval shape dimensions varies from 60 m by 80 m at the underground level (altitude of -10 m) to their minimum values of 20.65 m by 27.5 m at the altitude of 280 m, and then they increase again to 40.5 m by 54 m at the top level of the tube (altitude of 450 m). The oval shape of the top level is rotated 45 degrees horizontally relative to that of the bottom level. The top level plan is also inclined 15.5 degrees to the horizontal plane. The inner tube shape is an oval with constant dimensions along its height (14 m by 17 m), and its centroid is not that of the outer tube. The thickness of the tube varies from 1m at the bottom to 0.4 m at the top (Ni *et al*. 2009).

The tower was instrumented with an SHM system comprised of 600 sensors. The system was designed and implemented by the Hong Kong Polytechnic University for both in-construction and in-service real-time monitoring of the tower (Ni *et al*. 2008). Note that Fig. 2 indicates the locations of accelerometers along the tower height as well as the axes being measured. The dynamical response of the tower to an earthquake was recorded by 17 sensors measuring two different axes. The measured acceleration data sequences obtained from several sensors are demonstrated in Fig. 3 for six minutes of response during an earthquake. The sampling frequency of the signal was 50 Hz.

4.2 Numerical results

In this subsection, we analyze the gains that may be achieved by applying the two schemes described in Section 3.3, for different cooperation scenarios. Due to the fact that the LMS-based algorithm has been examined in the conference companion of this paper (Ampeliotis *et al*. 2012), here we mostly focus on the performance of the RLS-based protocol. It has been concluded that for a given distortion, the number of required transmissions from a certain node toward the sink varies with reference to (a) the number of the filter coefficients of a sensor node, (b) the size of cooperating neighborhood and (c) which node(s) are selected for cooperation. To perform a fair comparison among different cooperation scenarios, we assume the same filter lengths. For instance, for the filter length of 18, a non-cooperative node exploits its 18 past measurements. On the contrary, by cooperating with one neighbor, a node exploits 9 of its own past measurements and 9 of the neighbor’s.

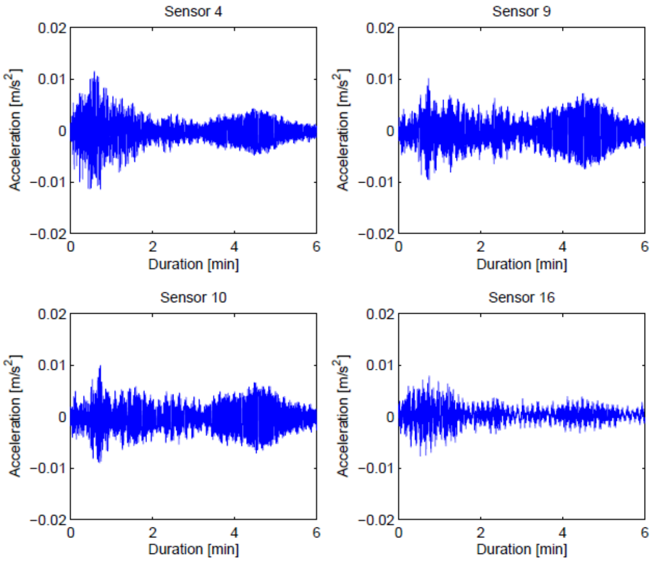
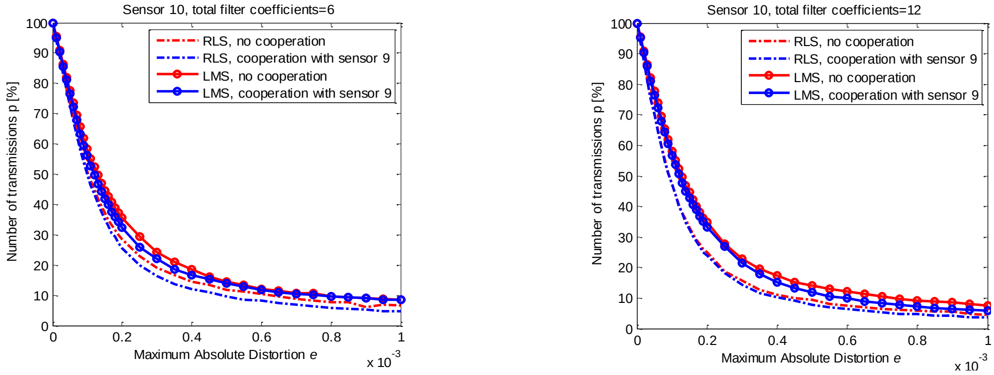


Fig. 3 The measured acceleration data sequences



(a) Number of filter coefficients=6 (b) Number of filter coefficients=12

Fig. 4 Performance comparison between the LMS-based and RLS-based schemes for sensor 10

In Fig. 4(a), we compare the LMS-based and RLS-based schemes for sensor node 10, which is located in the middle of the tower, for a filter length equal to 6. In both schemes, we analyze a cooperative scenario, i.e., cooperation with sensor 9, and non-cooperative, where node 10 relies only on its own measurements. It can be seen that for both protocols there is a benefit due to cooperation. Not surprisingly, for correlated input signals, the RLS-type implementation generally performs better due to the faster convergence speed and better ability to adapt in a fast time-varying environment. Also, RLS performs better when adaptation stalls and restarts very often during operations. Of course, such performance gains come at an increased computational complexity over the LMS-type implementation. In Fig. 4(b), for the same simulation setting we use a greater filter length, and the cooperation gain seems to be smaller (yet still existing).

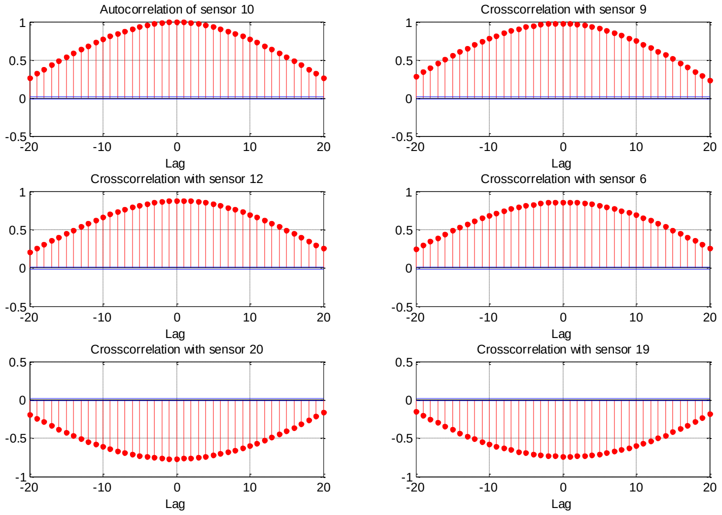
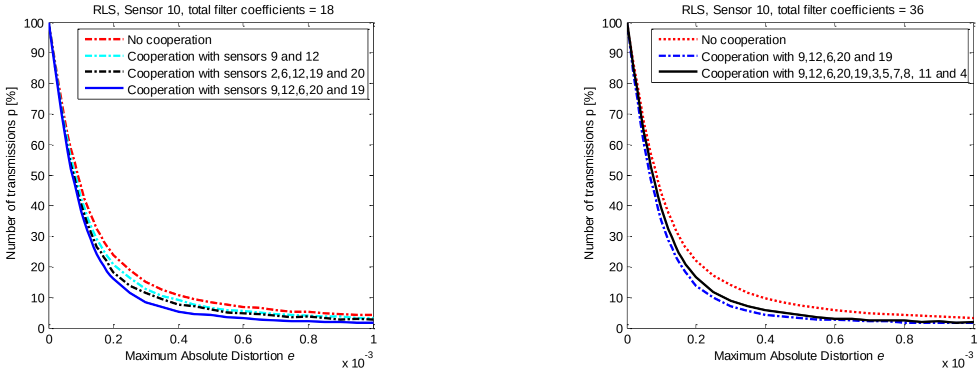


Fig. 5 The averaged performance for 6 nodes

Table 3 The crosscorrelation coefficients between the data of sensor 10 and other sensors

Correlation coefficient	Sensor number
0.9796	9
0.8756	12
0.8559	6
0.7688	20
0.7387	19
0.6286	4
0.4347	11
0.4199	2
0.3869	16
0.1520	17



(a) Number of filter coefficients=18

(b) Number of filter coefficients=36

Fig. 6 Performance of the RLS-based scheme for different cooperating neighborhoods for sensor 10

In the following, we focus on the RLS-based scheme. Let us analyze how the performance changes as the cooperating neighborhood size grows, for greater filter lengths. The results for sensor 10 demonstrate that, in general, the performance can be improved by increasing the number of cooperating nodes. However, in order to maximize the gains, one should carefully select suitable cooperating neighbors; see Fig. 6(a).

A simple, yet effective, way to determine a suitable cooperating neighborhood is to analyze the correlation coefficients for each node at the zero-th lag. Note that in this setting, we do not take into account the criterion of channel quality among the nodes. Thus, after ordering the absolute values of correlation coefficients, for a neighborhood size of 6, one should just select the best 5 nodes ordered by this criterion. In Table 3, we order 10 nodes according to their relevance to node

10. For the signals considered in these experiments, this simple approach provides good results due to the fact that the crosscorrelation functions are wide enough, so the zero-th lag correlation coefficients gives enough information even for longer filter lengths. The autocorrelation function of node 10 and its crosscorrelation with several sensors are plotted in Fig. 5. Note that the cooperation gain for each sensor is dependent on the relation between the values of its auto- and cross- correlations at the different lags. In case that its autocorrelation at the limit lags, defined by the filter length, is greater than the crosscorrelation close to the zero-th lag, then for this node the cooperation will not be useful. On the other hand, when the crosscorrelation have greater values, then the neighbors add new information, so the predictor learns better the process. Furthermore, observe that the performance of the protocol may deteriorate by randomly adding nodes into the previously selected neighborhood. In Fig. 6(b), we illustrate this by plotting the curve obtained for a neighborhood consisting of 12 nodes (including node 10 itself). In fact, in addition to the group of 6 nodes performing well (solid line), we added 6 other nodes which seemed to be less relevant to sensor 10. Actually, five of these less relevant nodes measure a different axis than node 10 (see Fig. 2). Due to the adaptive nature of the protocol, they actually reduced the cooperation gains with respect to the scenario with properly selected cooperation neighborhood.

Finally, in Fig. 7 we plot the averaged performance of 6 sensors where in the cooperative case all of them cooperate with their best 5 neighbors. Although not all of them experience cooperation benefits to the same extent, there is an average performance improvement as compared to the non-cooperative case.

To conclude, the power consumption of a sensor node may be reduced by leveraging spatio-temporal correlations among sensor nodes. To this end, a crucial issue to be considered is the selection of an optimal cooperating neighborhood in terms of both its size and the nodes involved.

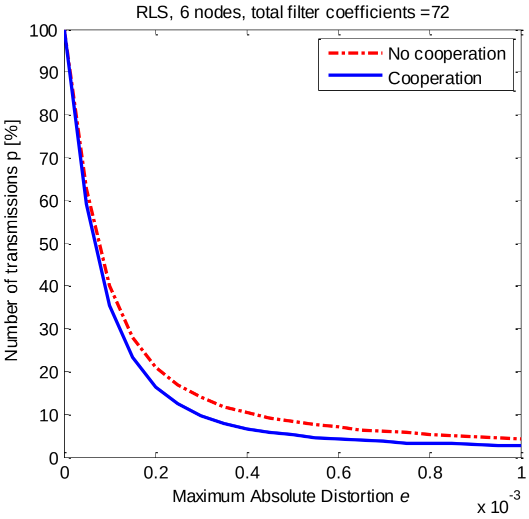


Fig. 7 The averaged performance for 6 nodes

5. Conclusions

The proposed spatio-temporal data gathering protocol reduces the power consumption by reducing the number of measurements transmitted to a sink node, within some prescribed distortion. Additionally, the protocol leverages the idea of cooperative communication in order to reduce the required transmission power. The experiments with real acceleration measurements demonstrate its efficiency and indicate savings in transmitted energy.

As it was presented in the simulation section, one of the major factors that influence the performance of the proposed protocol is the determination of the cooperating neighborhoods used for prediction. Thus, the development of a method to select such neighborhoods in an optimal manner would be highly desirable. Furthermore, a dynamic version of such an algorithm, able to modify these coalitions in an on-line fashion would also be very important.

Acknowledgments

The authors would like to acknowledge the European Commission for funding SmartEN (Grant No. 238726) under the Marie Curie ITN FP7 program, as the research work presented here is supported by this program. The opinions expressed in this paper do not necessarily reflect those of the sponsors.

References

Ampeliotis, D., Bogdanovic, N., Berberidis, K., Casciati, F. and AlSaleh, R. (2012), “Power-efficient wireless sensor reachback for SHM”, *Proceedings of the 6th International Conference of Association for Bridge Maintenance and Safety (IABMAS 2012)*, Lake Como, Italy.

Barros, J., Peraki, C. and Servetto, S. (2004). “Efficient network architectures for sensor reachback”, *Proceedings of the 13th International Zurich Seminar on Communications (IZS 2004)*, Zurich, Switzerland.

Barros, J. and Servetto, S. (2006). “Network information flow with correlated sources”, *IEEE T. Inform. Theory*, **52**(1), 155 -170.

Casciati, F., Alsaleh, R. and Fuggini, C. (2009). “Gps-based shm of a tall building: torsional effects”, *Proceedings of the 7th International Workshop on Structural Health Monitoring (IWSHM 2009)*, Stanford, CA, USA.

Cui, S., Goldsmith, R. and Bahai, A. (2004), “Energy efficiency of MIMO and cooperative MIMO techniques in sensor networks”, *IEEE J. Sel. Areas Comm.*, **22**(6), 1089- 1098.

Hong, Y.W., Huang, W.J., Chiu, F.H. and Kuo, C.C. (2007), *Cooperative communications in resource -constrained wireless networks*, IEEE Signal Processing Magazine, March.

Lynch, J.P. (2006), “A summary review of wireless sensors and sensor networks for structural health monitoring”, *Shock Vib. Dig.*, **38**(2), 91-128.

Nagayama, T., Moinzadeh, P., Mechitov, K., Ushita, M., Makihata, N., Ieiri, M., Agha, G., Spencer, J.B.F., Fujino, Y. and Seo, J. (2010), “Reliable multi-hop communication for structural health monitoring”, *Smart Struct. Syst.*, **6**(5), 481-503.

Ni, Y.Q. *et al.* (2008), *A benchmark problem for the structural health monitoring of highrise slender structures*, http://www.cse.polyu.edu.hk/benchmark/index.htm.

Ni, Y.Q., Xia, Y., Liao, W.Y. and Ko, J.M. (2009), “Technology innovation in developing the structural health monitoring system for guangzhou new tv tower”, *Struct. Control Health Monit.*, **16**(1), 73-98.

Sayed, A.H. (2008), *Adaptive filters*, John Wiley & Sons, Hoboken, NJ, USA.

Stankovic, V., Stankovic, L. and Cheng, S. (2010), "Distributed source coding: Theory and applications", *Proceedings of the 18th European Signal Processing Conference, (EUSIPCO 2010)*, Aalborg, Denmark.

FC