ECHO STATE WIRELESS SENSOR NETWORKS

Dmitriy Shutin and Gernot Kubin

Signal Processing and Speech Communication Laboratory
Technical University Graz, Inffeldgasse 12
A-8010 Graz, Austria

ABSTRACT

This paper addresses the question of temporal learning in spatially distributed wireless sensor networks (WSN). We propose to fuse WSNs with the Echo States Network learning concepts to infer the spatio-temporal dynamics of the data collaboratively measured by sensors. We prove that a WSN topology described by a bidirected graph is strongly connected, which is a sufficient and necessary condition for implementing in-network distributed learning. For strongly connected networks we develop a systematic method to satisfy the conditions resulting in echo states in sensor networks. The effectiveness of the learning approach is demonstrated with several controlled model experiments.

1. INTRODUCTION

In recent years the advances in electronics and digital communications have made wireless sensor networks (WSN) a very promising tool for efficiently solving large-scale decision and information-processing tasks [1, 2]. WSNs allow to collect and process spatially distributed and time-varying data. Due to energy constraints and often limited communication capabilities, the operation of WSNs relies on distributed processing, when the operation of the whole network is achieved through the synergy of individual sensors, able to sense, compute, and communicate data.

The goal of sensor networks is to collect information about the environment they are sensing and to infer properties or patterns governing the observed data and describing their information content. When the statistics or models of the observed phenomenon can be assumed as known, the task of inference in WSNs boils down to the problem of decentralized estimation or detection (see [3] and references therein). However, when little or no *a priori* information about the measured process is available, or when the observed data is sparse, it is often more efficient to construct the model from the data itself (see [4] and references therein). The second scenario does not depend so much on prior assumptions, but rather on learning-theoretic approaches, i.e., pattern recognition, nonparametric regression, neural networks, etc. Current learning approaches in WSNs address applications of classical "non-temporal" statistical learning setups, when data acquired by each sensor is assumed to be i.i.d. samples drawn from some fixed but unknown distribution. However, to the best of our knowledge, learning in WSNs for processes which have memory both in time and in space has not been addressed in the literature. The study of the latter forms the main focus of this work.

In this contribution we consider the problem of spatio-temporal learning with sensor networks that is posed as follows: spatially distributed sensors acquire collectively a set of data samples $\{\boldsymbol{u}[n], \boldsymbol{y}[n]\} \subset \mathcal{U} \times \mathcal{Y}$ during the time interval $n = 0, \ldots N-1$. Measurements $\boldsymbol{u}[n] \in \mathbb{R}^L$, and $\boldsymbol{y}[n] \in \mathbb{R}^L$ represent the snapshots of some spatial scalar field measured by L sensors at the time n. Our goal is the inference of the unknown functional relationship

$$\boldsymbol{y}[n] = g(\ldots, \boldsymbol{y}[n-k], \ldots, \boldsymbol{u}[n-l], \ldots).$$

We will also assume that g exists and that it has memory both in time and space. This assumption is what makes this inference a spatio-temporal learning task. To solve this task we propose to extend a class of recurrent neural networks, known as Echo State Networks (ESN) [5], for spatio-temporal learning applications in WSNs– an Echo State Wireless Sensor Network (ES-WSN). Learning in ESNs is achieved by creating a network of interconnected neurons called a reservoir. The output of the reservoir, created by exiting the network with the input $\boldsymbol{u}[n]$, produces a basis for transforming network inputs into the so-called echo states. The latter are then (usually) linearly combined to obtain the desired network output. The estimation of the optimal combiner coefficients constitutes the corresponding ESN training stage.

In WSNs, especially *ad hoc* WSNs [6], the network topology is usually random and unknown prior to deployment. Sensors have to self-organize and self-configure to form a network by establishing connections with their neighbors. The topology of such networks is reminiscent of the ESN's reservoir, and, if the number of sensors is large, such a reservoir creates a sufficiently rich basis for mapping the input data into echo states. By equipping the sensors with computational functionalities we can make sensors perform neuron-like computation, which is essentially an application of a static nonlinear map to the weighted sum of inputs and states of other sensors. Note that this can be seen as the distributed compressed sensing which is currently an active research area in the WSN community [7]. Quite naturally distributed sensor measurements can be used as training examples for learning. This task can be solved either centrally at the fusion center (i.e., at the data accumulation and processing center), or distributively (in-network computations) by finding an optimal regressor collaboratively. However, such collaborative processing is only feasible when the topology of the WSN satisfies specific constraints, which will be discussed later in the text.

In what follows we outline the main concepts of the proposed ES-WSNs. In Section 2 we define the used terminology and notations. Section 3 discusses how to satisfy the conditions for the existence of echo states. There we also propose a distributed initialization scheme. Finally, in Section 4 we demonstrate the learning performance of ES-WSN on model-generated synthetic data.

2. STRUCTURE OF ES-WSN

In the rest of the paper we will generally abstract from the networking aspects and physical link properties of the network, assuming that the sensors are able to exchange real-valued messages. We will also assume that, for the purpose of learning, the sensors are already organized into a network, and each sensor "knows" its neighbors. Furthermore, we will assume that the communication links between the sensors are reciprocal, i.e., that data can flow in both ways between any two connected sensors.

Let a WSN consist of L sensors $\mathcal{S}_l$, $l = 0, \ldots, L-1$. Interconnections between the sensors induce a network topology, which can be represented by a bidirected graph (shown in Fig. 1), where the graph vertices represent sensors, and

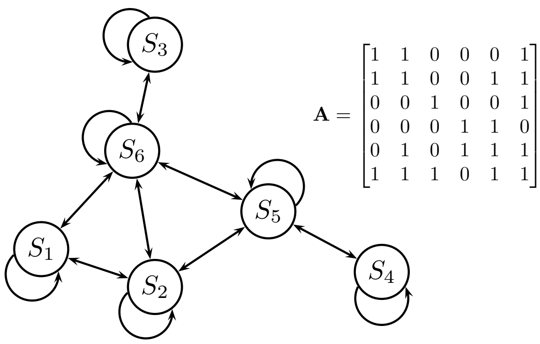

Fig. 1. A possible configuration of a sensor network and the corresponding adjacency matrix.

its edges correspond to the existing communication links. The graph topology is represented by an adjacency matrix $\boldsymbol{A}$, with elements $a_{lk} = 1$, when sensors $\mathcal{S}_k$ and $\mathcal{S}_l$ are connected, and $a_{lk} = 0$ otherwise. As a neighborhood $\mathcal{N}_l$ of

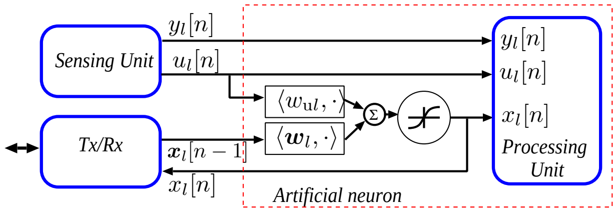

Fig. 2. A model of a sensor neuron.

the sensor $\mathcal{S}_l$ we will denote the subset of all sensors in the WSN that are connected to $\mathcal{S}_l$, i.e.,

$$\mathcal{N}_l = \{\mathcal{S}_k |\text{ such that } \quad a_{lk} = 1, k = 0, \ldots, L-1\}$$

Note that the neighborhood $\mathcal{N}_l$ can also include sensor $\mathcal{S}_l$ as well.

In order to combine WSNs and ESNs, we impose neuron-like processing tasks on sensors (Fig. 2). As we see, in order to produce a state $x_l[n]$ each sensor $\mathcal{S}_l$ applies a static nonlinear map to a weighted sum of the states $\boldsymbol{x}_l[n-1]$ received from $\mathcal{S}_k \in \mathcal{N}_l$, and a weighted measurement $u_l[n]$. Observe that such input transformation can also be seen as a distributed (due to the recursive nature of the network) data compression stage. Formally, the response of each sensor is represented by the following state update equation

$$x_l[n] = f(w_{\text{ul}}^T u_l[n] + \boldsymbol{w}_l^T \boldsymbol{x}[n-1]) \qquad (1)$$

where w_{ul} is the weighting coefficient for the input data $u_l[n]$. Coefficients $\boldsymbol{w}_l = [w_{l0}, \ldots, w_{l(L-1)}]^T$ are the weights of the internal network connections, with elements defined as

$$w_{lk} = \begin{cases} 0 & a_{lk} = 0 \\ \tilde{w}_{lk} & a_{lk} = 1, \end{cases} \quad k = 0, \ldots, L-1. \qquad (2)$$

Both $\tilde{w}_{lk}$ and w_{ul} are randomly generated at each sensor during the network initialization. Typically in ESNs this is done by drawing samples from a zero-mean Gaussian distribution.

The collective response of the network is governed by the "measurement" equation

$$\boldsymbol{y}[n] = \boldsymbol{W}_{\text{o}} \begin{bmatrix} \boldsymbol{x}[n] \\ \boldsymbol{u}[n] \end{bmatrix}. \qquad (3)$$

Learning in ESNs consists in finding the optimal weighting matrix $\boldsymbol{W}_o$ that minimizes the distance between the desired response $\boldsymbol{y}[n]$ and the network output, while keeping all other network weights fixed. Since the matrix $\boldsymbol{W}_o$ enters (3) linearly, the optimal solution can be found by solving a system of linear equations. This clearly alleviates a significant computational burden from the resource-limited sensors due to the existence of numerous efficient (as well as distributed) techniques for solving linear systems of equations [8, 9].

Now let us consider some technical aspects arising when extending the echo state principle onto wireless sensor networks.

3. LEARNING METHODS FOR ES-WSN

In order to ensure proper functioning of the ES-WSN it is crucial to make sure that the internal network weights $\boldsymbol{w}_l$ satisfy the echo state property. In [5] Jaeger defines sufficient and necessary conditions for the network to have echo states assuming the activation function $f(\cdot)$ in (1) is Lipschitz continuous. Let $\sigma_{\max}(\boldsymbol{W})$ denote the largest singular value of the matrix $\boldsymbol{W} = [\boldsymbol{w}_0, \ldots, \boldsymbol{w}_{L-1}]^T$, and let $\mu(\boldsymbol{W})$ denote its spectral radius. Then, $\sigma_{\max}(\boldsymbol{W}) < 1$ is a sufficient, and $\mu(\boldsymbol{W}) < 1$ is a necessary condition for the network to have echo states (see [5] for the proof).

In classical ESNs both the sufficient and necessary conditions can be easily satisfied by randomly generating entries of the matrix $\boldsymbol{W}$ and then applying a global scaling constant α such that $\sigma_{\max}(\alpha\boldsymbol{W}) < 1$. In the case of WSNs such global scaling is impossible to implement without a centralized processor, which is usually not available in WSNs. Instead, we have to generate the weights $\boldsymbol{w}_l$ locally, i.e., independently at each sensor, and then implement some in-network processing to ensure that $\sigma_{\max}(\boldsymbol{W}) < 1$. In what follows we propose a simple distributed initialization procedure that achieves this goal.

3.1. Mathematical preliminaries

First, let as recall that due to the reciprocity of the links between the sensors the adjacency matrix is symmetric, i.e, $\boldsymbol{A} = \boldsymbol{A}^T$. Let us now restate several properties from matrix theory that we will exploit later in the text.

Definition 1. *: A matrix $A \in \mathbb{C}^{L\times L}$ is said to be reducible if there is an $L \times L$ permutation matrix $\boldsymbol{P}$ such that*

$$\boldsymbol{P}^T\boldsymbol{A}\boldsymbol{P} = \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{0} & \boldsymbol{A}_{22} \end{bmatrix}$$

where $\boldsymbol{A}_{11}$ and $\boldsymbol{A}_{22}$ are square matrices of order less than L. A WSN whose topology is described by a reducible adjacency matrix $\boldsymbol{A}$ is called a reducible WSN.

The next two definitions illustrate why irreducibility is an important property in WSNs, in particular in learning applications.

Definition 2. *A graph is said to be strongly connected if for each pair of nodes $\mathcal{S}_k$, $\mathcal{S}_l$, $k \neq l$, there is a directed path consisting of m directed lines that connect $\mathcal{S}_k$ and $\mathcal{S}_l$*

Lemma 1. *If the graph adjacency matrix $\boldsymbol{A}$ is irreducible, then the network is strongly connected [10].*

In distributed learning applications the results of local data processing, e.g., the local error gradient [11], have to be made available to the whole network. Violation of the strong connectivity leads to some sensors being cut off from the rest of the network. It is easy to conclude that strong connectivity is a sufficient and necessary condition that guarantees that the results of local computations can reach any other sensor in the network. Thus, for the in-network learning the corresponding WSN topology must be strongly connected.

Now, we are ready to formulate a proposition that is crucial for ES-WSN initialization as well as for general in-network learning in WSNs.

Proposion 1. *For a WSN with reciprocal communication links between the sensors the following is true:*

1. *The corresponding adjacency matrix $\boldsymbol{A}$ is either irreducible, or $\boldsymbol{P}^T\boldsymbol{A}\boldsymbol{P}$ is block-diagonal (i.e., $\boldsymbol{A}_{12} = 0$) with irreducible submatrices $\boldsymbol{A}_{jj}$ corresponding to separate disjoint WSNs,*

2. *If for a reducible network the corresponding diagonal submatrices $\boldsymbol{W}_{jj}$ are independently initialized such that $\sigma_{\max}(\boldsymbol{W}_{jj}) < 1$, then the total spectral norm is also bounded as $\sigma_{\max}(\boldsymbol{W}) < 1$*

The proof of this result can be found in Appendix A. From this result we can immediately draw an important conclusion:

Corollary 1.1. *For a single irreducible WSN it is possible to find a closed directed loop $\mathcal{L}$ through the network such that starting at some sensor $\mathcal{S}_0$ we can form a directed path*

$$\mathcal{L} = \mathcal{S}_0 \rightarrow \mathcal{S}_k \rightarrow \ldots \mathcal{S}_l \rightarrow \mathcal{S}_0.$$

This result follows directly from the strong connectivity of the WSN with reciprocal links. Note that the loop $\mathcal{L}$ is not necessarily a Hamiltonian loop through the graph. As we will show, forming a loop through the network is required for the distributed ES-WSN initialization. Observe, that in many other in-network learning algorithms, e.g., [11, 9, 12], it is also required to form a network loop. Its existence was usually assumed, but the conditions that guarantee this existence were not previously specified.

3.2. Distributed normalization of the internal connection matrix $\boldsymbol{W}$

Let us now pose the initialization problem in ES-WSNs. After having randomly and independently generated coefficients $\boldsymbol{w}_l$ at each sensor, our goal is to find distributively a scaling constant α such that $\sigma_{\max}(\alpha\boldsymbol{W}) < 1$.

A simple way to satisfy this bound without the exact computation of the largest singular value of $\boldsymbol{W}$ is to use its upper bound [13]

$$\sigma^2_{\max}(W) \le \max_l \gamma_l \qquad (4)$$

where

$$\gamma_l = \sum_k |w_{lk}| c_k, \quad \text{and} \quad c_k = \sum_l |w_{lk}|.$$

Note that (4) is tighter than the well known Schur bound [14] $\sigma^2_{\max}(W) \le \|\boldsymbol{W}\|_1 \|\boldsymbol{W}\|_\infty$. We will now show how to compute $\max_l \gamma_l$ using only local communications, i.e., communications between the connected sensors only.

Let as assume that the adjacency matrix describing the topology of the WSN is irreducible. Then, there exists a loop $\mathcal{L}$ that visits all sensors at least once. Let as also assume that this loop is known.

A closer look at (4) reveals that in order to compute the bound $\beta = \max_l\{\gamma_l\}$, each sensor $\mathcal{S}_l$, in addition to the lth row $\boldsymbol{w}_l^T$ of $\boldsymbol{W}$, also needs to know the corresponding column $\boldsymbol{w}_l'$ (see Fig. 3). The coefficients $\boldsymbol{w}_l'$ determine how the output $x_l[n]$ is weighted by the sensors in $\mathcal{N}_l$. Clearly, in case of reciprocal communication links this will require only local exchange of weights. This requires exactly one loop $\mathcal{L}$ through the network, during which the sensors sequentially transmit their weights to the corresponding neighbors. At the end of the loop each sensor $\mathcal{S}_l$ will know the lth row and lth column from the matrix $\boldsymbol{W}$.

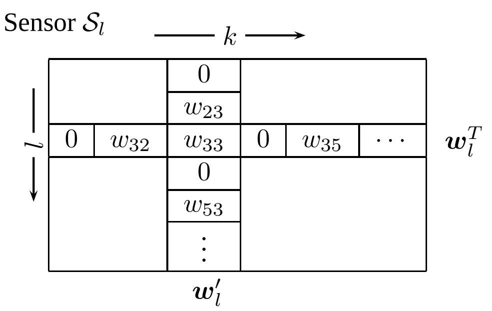

Fig. 3. Structure of the internal connection weights at the sensor $\mathcal{S}_l$.

The next step involves computing $c_l = \|\boldsymbol{w}_l'\|_1$ at each sensor and distributing the results to the neighbors, which again requires local transmissions only. Once done, each sensor can compute $\gamma_l = \sum_k |w_{lk}| c_k$ and compare his result to that received from the previous sensor in the loop. The largest number is sent forward to the next sensor along the path. Thus, after the loop is complete, the last sensor in the loop knows the bound β. During the final loop the bound β is used to normalize the coefficients $\boldsymbol{w}_l$ and is transmitted further along the loop. Thus, four loops are necessary to normalize the coefficients without direct computation of $\sigma_{\max}(\boldsymbol{W})$. We summarize these steps in the Algorithm 1.

Table 1 Computation of $\max_l \sum_k |w_{lk}| c_k$

% — share w_{lk} between the neighbors —%
for each $\mathcal{S}_l \in \mathcal{L}$
  for each $\mathcal{S}_k \in \mathcal{N}_l$
    $w'_{kl} = w_{lk}$
  end
end
% — computing c_l for each sensor —%
for each $\mathcal{S}_l \in \mathcal{L}$
  for each $\mathcal{S}_k \in \mathcal{N}_l$
    $c_{kl} = ||\boldsymbol{w}_l'||_1$
  end
end
% — computing the bound β —%
$\beta_{-1} = -\infty$
for each $\mathcal{S}_l \in \mathcal{L}$
  $\gamma_l = \sum_k |w_{lk}| c_{lk}$
  $\beta_l = \max\{\gamma_l, \beta_{l-1}\}$
end
% — normalizing the weights $\boldsymbol{w}_l$—%
for each $\mathcal{S}_l \in \mathcal{L}$
  $\beta_l = \beta_{l-1}$
  $\boldsymbol{w}_l = \alpha \boldsymbol{w}_l / \beta_l, \quad \alpha < 1$
end

3.3. Learning algorithm

Let us now discuss how to implement the actual learning procedure. Sensors communicate a message (packet) by broadcasting to all neighboring sensors, or to a single sensor along the loop. Note that sensors in the network can also operate as relays by simply passing the data further along to the given loop $\mathcal{L}$. Clearly, a single loop corresponds to one time step n in (1) and (3) for all $l = 0, \dots, L-1$.

We will now consider learning strategies for two types of WSNs: networks with a fusion center (FC), and networks using collaborative in-network learning.

3.3.1. FC-based learning

In this regime learning is quite straight-forward – each sensor transmits a message $\{l, x_l[n], u_l[n], y_l[n]\}$ to the FC. This can be done either directly, when there is a direct connection to the FC, or relayed through the network, which would require an appropriate network protocol. Then the task of the FC is to find the coefficients $\boldsymbol{W}_o$ in (3) by solving

$$\hat{\boldsymbol{W}}_o = \underset{\boldsymbol{W}_o}{\operatorname{argmin}} \left\| \boldsymbol{y}[n] - \boldsymbol{W}_o \begin{bmatrix} \boldsymbol{x}[n] \\ \boldsymbol{u}[n] \end{bmatrix} \right\|_2^2. \qquad (5)$$

If the data arrives at the fusion center sequentially (with the period between data snapshots corresponding to the length of a single loop), one can make use of the classical LMS or RLS algorithms [15] to find the optimal weights. Clearly, it is also possible to accumulate the data over time and then solve (5) simultaneously for all time instances. In general this approach will require at least $O(L^2)$ loops in order to estimate L^2 coefficients of the matrix $\boldsymbol{W}_o$. Observe that FC-based learning can still be used when the network is reducible, i.e., consists of several disjoint WSNs, since learning takes place in the fusion center. However, the disjoint networks can be connected through the FC with appropriate network protocols.

3.3.2. Distributed in-network learning

In this regime the sensors have to solve (5) distributively. The major difference to FC-based learning is that optimal weights become available to all sensors as the network is trained. This clearly allows each sensor to take actions that are globally optimal (with respect to the WSN), and opens a way to designing self-aware sensor networks.

The formal basis for such computations is readily provided by many distributed optimization algorithms [9, 8]. The optimal regressor $\boldsymbol{W}_o$ is found by defining a loopy path through the sensor network such that every sensor is visited at least once. Along this path the sensors exchange messages consisting of a current estimate of $\boldsymbol{W}_o$, and update it using the local data $\{u_l[n], x_l[n], y_l[n]\}$ only. A single update of $\boldsymbol{W}_o$ is achieved after the loop is complete and all measurements have been incorporated in the update. Note also that unlike the case of FC-based learning, in-network based learning is only meaningful for irreducible networks.

4. SIMULATION RESULTS

Let us now consider a model example that illustrates learning in a sensor network. For simplicity let us assume that the sensors are randomly scattered in a 2D plane. The adjacency matrix of the network is sparse, with $\approx 15\%$ of nonzero entries.

The synthetic data used to test the learning performance of the network was generated using the following multivariate ARMA model:

$$\boldsymbol{s}[n] = \sum_{q=1}^{3} \tanh(\boldsymbol{A}_q \boldsymbol{s}[n-q]) + \boldsymbol{B}_0 \boldsymbol{\epsilon}[n]. \qquad (6)$$

Parameters $\boldsymbol{A}_q \in \mathbb{R}^{L\times L}$ and $\boldsymbol{B}_0 \in \mathbb{R}^{L\times L}$ in (6) are randomly generated matrices. These matrices are normalized such that $\|\boldsymbol{A}_q\| = 1.5$, and $\|\boldsymbol{B}_0\| = 0.1$. Signals $\boldsymbol{s}[n] = [s_1[n], \ldots, s_L[n]]^T$ and $\boldsymbol{\epsilon}[n] = [\epsilon_1[n], \ldots, \epsilon_L[n]]^T$ are input and output signals, respectively. Signal $\boldsymbol{\epsilon}[n]$ is generated by drawing i.i.d. samples from a multivariate zero-mean normal distribution with the covariance matrix $\boldsymbol{I}$.

The internal network weights $\boldsymbol{W}$ are drawn from the standard zero-mean Gaussian distribution and normalized

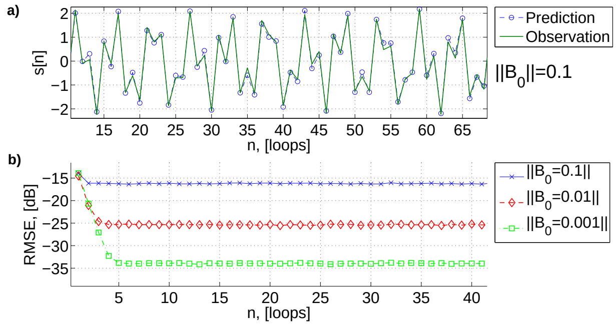

Fig. 4. a)Original and predicted signal waveform for $\|\boldsymbol{B}_0\| = 0.1$. b) Root-mean squared prediction error for different norms of the matrix $\|\boldsymbol{B}_0\|$..

as proposed in Algorithm 1 such that $\sigma_{\max}(\boldsymbol{W}) = 0.95$. The input weight $w_{\mathrm{ul},l}$ is drawn from a zero-mean Gaussian distribution. During the training stage it is assumed that each sensor "measures" a pair of values $u_l[n] = s_l[n-1]$, and $y_l[n] = s_l[n]$, i.e., the learning objective is a one-step-ahead predictor.

After the training stage, which consists in solving (5) using pseudo inverse, we generate the test data anew using (6) with a different random noise sequence $\boldsymbol{\epsilon}[n]$, and compare predictions of the trained network to the true output $\boldsymbol{s}[n]$. We assess the learning performance in terms of the test RMSE, averaged over 100 realizations of $\boldsymbol{s}[n]$. Results for different choices of $\|\boldsymbol{B}_0\|$ and $L = 50$ sensors are shown in Fig. 4. As we see, after the initial transient, the learning algorithm successfully predicts the observed signal $\boldsymbol{s}[n]$. Clearly, the learning performance deteriorates as the variance of driving noise term $\boldsymbol{B}_0\boldsymbol{\epsilon}[n]$ increases.

5. CONCLUSIONS

In this contribution we propose and discuss Echo State Wireless Sensor Networks for solving spatio-temporal learning tasks. We use the topology of the WSN as an echo state reservoir for representing the spatio-temporal dynamics of the measured data using echo states. The learning problem can then be solved by finding the optimal regression of echo states onto the desired network targets measured by sensors. This typically linear problem can be efficiently solved at a fusion center or distributively by sensors. We showed that for in-network learning it is essential to have a strongly connected WSN. This allows to form a loop through the network, which in turn is essential for propagating (and collecting) the local information. Assuming the WSN is represented by a single graph, reciprocity of the wireless links between the sensors is a sufficient condition for the network to be strongly connected. For strongly connected networks we developed a novel distributed initialization algorithm that ensures existence of echo states in WSNs. The learning method was demonstrated on a synthetic nonlinear AR model of the first order. The obtained results show that the ES-WSN can successfully learn the spatio-temporal dynamics of the observed data.

A. PROOF OF PROPOSITION 1

We first prove the part 1) of the proposition. If the matrix $\boldsymbol{A}$ is reducible then there is an $L \times L$ permutation matrix $\boldsymbol{P}$ such that

$$\boldsymbol{P}^T \boldsymbol{A} \boldsymbol{P} = \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix}$$

where $\boldsymbol{A}_{21} = \boldsymbol{0}$, and $\boldsymbol{A}_{11}$ and $\boldsymbol{A}_{22}$ are square matrices of order less than L. Under the condition of reciprocity of the sensor links, the matrix $\boldsymbol{P}^T \boldsymbol{A} \boldsymbol{P}$ must also be symmetric. Indeed, the permutation is equivalent to a simple relabeling of sensors in the network, which does not destroy the reciprocity property. Thus $\boldsymbol{P}^T \boldsymbol{A} \boldsymbol{P}$ is also symmetric, which means that $\boldsymbol{A}_{12}^T = \boldsymbol{A}_{21} = \boldsymbol{0}$. By induction the same argumentation can be applied to submatrices $\boldsymbol{A}_{11}$ and $\boldsymbol{A}_{22}$ – they are either irreducible or block diagonal. As a consequence, for a network represented by a single graph it is sufficient to have reciprocal communication links to ensure irreducibility. To prove part 2) of the proposition let us assume that $\boldsymbol{A}_{11}$ and $\boldsymbol{A}_{22}$ are irreducible, i.e., the corresponding subnetworks are strongly connected. Then, within each subnetwork we can form a closed loop and apply the proposed normalization procedure, such that $\sigma_{\max}(\boldsymbol{W}_{11}) = \alpha_1 < 1$ and $\sigma_{\max}(\boldsymbol{W}_{22}) = \alpha_2 < 1$. Then it follows that $\sigma_{\max}(\boldsymbol{P}^T \boldsymbol{W} \boldsymbol{P}) = \sigma_{\max}(\boldsymbol{W})$ and

$$\sigma_{\max}(\boldsymbol{W}) = \max \Big(\sigma_{\max}(\boldsymbol{W}_{11}), \sigma_{\max}(\boldsymbol{W}_{22}) \Big) < 1, \quad (7)$$

which proves the result.

B. REFERENCES

[1] A. Swami, Q. Zhao, Y.-W. Hong, and L. Tong, Eds., *Wireless Sensor Networks: Signal Processing and Communications*, Wiley Interscience, 2007.

[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks,” 2002.

[3] B. Chen, L. Tong, and P.K. Varshney, “Channel-aware distributed detection in wireless sensor networks,” *IEEE Signal Proc. Magazine*, vol. 23, no. 4, pp. 16–26, 2006.

[4] J.B. Predd, S.B. Kulkarni, and H.V. Poor, “Distributed learning in wireless sensor networks,” *IEEE Signal Proc. Magazine*, vol. 23, no. 4, pp. 56–69, 2006.

[5] H. Jaeger, “The echo state approach to analyzing and training recurrent neural networks,” Tech. Rep. Tech.Rep. 148, German National Research Center for Information Technology, 2001.

[6] Carlos De M. Cordeiro and Dharma P. Agrawal, *Ad Hoc & Sensor Networks: Theory And Applications*, World Scientific Publishing Company, 2006.

[7] Jin-Jun Xiao, A. Ribeiro, Zhi-Quan Luo, and G.B. Giannakis, “Distributed compression-estimation using wireless sensor networks,” *IEEE Signal Proc. Magazine*, vol. 23, no. 4, pp. 27–41, 2006.

[8] I.D. Schizas, G. Mateos, and G.B. Giannakis, “Consensus-Based Distributed Recursive Least-Squares Estimation using Ad Hoc Wireless Sensor Networks,” in *Conference Record of the Forty-First Asilomar Conference on Signals, Systems and Computers ACSSC 2007*, 2007, pp. 386–390.

[9] Ali H. Sayed and Cassio G. Lopes, “Distributed Recursive Least-Squares Strategies Over Adaptive Networks,” in *Proc. 40th Asilomar Conf. on Sign., Sys. and Comp.*, Cassio G. Lopes, Ed., 2006, pp. 233–237.

[10] Peter Lancaster and Miron Tismenetsky, *The Theory of Matrices*, Academic Press, 1985.

[11] Michael Rabbat and Robert Nowak, “Distributed optimization in sensor networks,” in *Proc. of the 3d Int. Symposium on Information Proc. in Sensor Networks*, New York, NY, USA, 2004, pp. 20–27, ACM.

[12] J. B. Predd, S. R. Kulkarni, and H. V. Poor, “Regression in sensor networks: Training distributively with alternating projections,” in *Proc. of SPIE on Advanced Sig. Proc. Alg., Archit., and Implem.*, F. T. Luk, Ed., Aug. 2005, vol. 5910, pp. 42–56.

[13] Vladimir Nikiforov, “Revisiting Schur’s bound on the largest singular value,” 2007, Published online http://arxiv.org/abs/math/0702722.

[14] I. Schur, “Bemerkungen zur Theorie der beschränkten Bilinearformen mit unendlich vielen Veränderlichen,” *J. reine angew. Math.*, vol. 140, pp. 1.28, 1911.

[15] Todd K. Moon and Wynn C. Stirling, *Mathematical Methods and Algorithms for Signal Processing*, Prentice-Hall, 2000.