Consensus-Based Distributed Recursive Least-Squares Estimation using Ad Hoc Wireless Sensor Networks†

Ioannis D. Schizas, Gonzalo Mateos and Georgios B. Giannakis (contact author)
Dept. of ECE, University of Minnesota
200 Union Str. SE, Minneapolis, MN 55455, USA
Emails: {schizas,mate0058,georgios}@ece.umn.edu

***Abstract*— Recursive least-squares (RLS) schemes are of paramount importance for online estimation and tracking of signals, especially when the state and/or data model are unknown. Here, a distributed RLS-like algorithm is developed that can operate in ad hoc wireless sensor networks (WSNs). The novel algorithm is obtained by writing the weighted squared-error cost associated with an RLS algorithm in a separable form and applying the alternating-direction method of multipliers to minimize it in a distributed fashion. This distributed adaptive scheme can be applied in general WSNs that are challenged by communication noise and do not necessarily possess a Hamiltonian cycle. Relative to competing alternatives, the novel algorithm offers more efficient communications. Numerical examples indicate that the proposed scheme is resilient to communication noise, while it performs efficient tracking of time-varying processes.**

I. Introduction

With the advent of WSNs, distributed estimation and tracking of signals based on sensor observations has drawn a lot of interest recently. This task becomes even more challenging in ad hoc WSNs where power and bandwidth constraints motivate single-hop communications only between neighboring sensors. With these constraints in mind batch least-squares distributed estimation has been considered in [7], [10].

In many applications though sensors need to perform estimation in a constantly changing environment without having available any state and/or data models. This led to the development of distributed adaptive estimation schemes. Specifically, the distributed incremental (I-) RLS approach in [6] allows the online incorporation of new information while performing least-squares estimation; see also [5]. However, I-RLS is operational as long as the WSN communication graph possesses a Hamiltonian cycle [4]. Avoiding the need for such a cycle and further exploiting all the available communication links in a WSN the diffusion RLS scheme was proposed in [2]. The resultant algorithm however, requires the exchange of regressors and observation data among neighboring sensors at every time instant rendering the approach costly from a communication perspective, as well as less robust in the presence of communication noise.

† Work in this paper was supported by the USDoD ARO Grant No. W911NF-05-1-0283; and also through collaborative participation in the C&N Consortium sponsored by the U. S. ARL under the CTA Program, Cooperative Agreement DAAD19-01-2-0011. The U. S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

Different from [6], [2], here we derive a distributed RLS adaptive algorithm which: i) can be applied to general WSNs that are challenged by communication noise and do not necessarily possess a Hamiltonian cycle; ii) efficiently tracks fast time-varying processes; iii) can easily handle sensor failures; and, iv) incurs communication cost which is linear with respect to the dimensionality of the signal we wish to estimate, while it exhibits higher convergence rates with respect to existing approaches.

After stating the problem in Section II, we proceed to reformulate the error cost associated with the exponentially weighted least-squares estimator (EWLSE) as the optimal solution of a separable constrained convex minimization problem (Section III). Then, the alternating-direction method of multipliers is utilized in order to minimize the separable EW-LS cost in a distributed fashion. This way local adaptive recursions, that allow the online incorporation of sensor data, are derived and constitute the distributed RLS algorithm (Section III-A). After elaborating on the distributed operation of the novel algorithm and providing some remarks, numerical simulations in Section IV expose the tracking capabilities and convergence properties of the distributed RLS scheme and compare them with existing alternatives.

II. Problem Statement

Consider a WSN with J sensors, where single-hop communications are allowed so that the j-th sensor communicates only with nodes j' in its neighborhood $\mathcal{N}_j \subseteq [1, J]$ having cardinality $|\mathcal{N}_j|$. Sensor links are assumed to be symmetric, and the WSN is modelled as an undirected graph whose vertices are the sensors and its edges represent the available links; see Fig. 1. Similar to [7], [10] and [2] we assume that the communication graph is connected. The WSN is deployed in order to estimate a $p \times 1$ parameter vector $\mathbf{s}_o$. Every sensor, say the j-th, at time instant t ($t = 0, 1, 2, \ldots$ denotes discrete time) acquires a $p \times 1$ regressor vector $\mathbf{h}_j(t)$ and a scalar observation $x_j(t)$ for $j = 1, \ldots, J$. A pertinent approach in this setup is to form the exponentially weighted least-squares estimator (EW-LSE) given at time instant t as [3]

$$\hat{\mathbf{s}}_{\text{ewls}}(t) := \arg\min_{\mathbf{s}} \sum_{\tau=0}^{t} \sum_{j=1}^{J} \lambda^{t-\tau} \|x_j(\tau) - \mathbf{h}_j^T(\tau)\mathbf{s}\|^2 + \lambda^t \mathbf{s}^T \mathbf{\Phi}_o \mathbf{s} \tag{1}$$

where $\lambda \in (0, 1]$ implements forgetting factor and $\mathbf{\Phi}_o$ is a positive definite matrix used for regularization. Notice that (1)

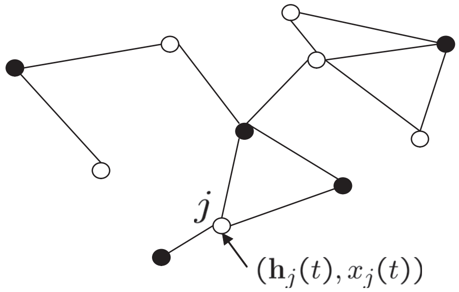


Fig. 1. An ad-hoc wireless sensor network.

provides the EW-LSE $\hat{\mathbf{s}}_{\text{ewls}}(t)$ for $\mathbf{s}_o$, given the observations and regressors $\{x_j(\tau), \mathbf{h}_j(\tau)\}_{j=1}^J$ within the time interval $\tau \in [0, t]$.

If the sensor observations $\mathbf{x}(t) := [x_1(t), \ldots, x_J(t)]^T$ and the regressor vectors in $\mathbf{H}(t) := [\mathbf{h}_1(t) \ldots \mathbf{h}_J(t)]^T$ were available at a central location, then $\hat{\mathbf{s}}_{\text{ewls}}(t)$ could be obtained using the RLS algorithm (centralized RLS) [3]. However, both the observations in $\mathbf{x}(t)$ and the regressors in $\mathbf{H}(t)$ are scattered across the WSN. One approach would be to have each sensor transmit $x_j(t), \mathbf{h}_j(t)$ to a fusion center (FC) and then directly apply RLS. This approach approach however is not only communication costly, but also is prone to FC failures. The goal of this paper is to minimize the cost in (1) in a distributed fashion and derive local recursions that enable each sensor to estimate $\mathbf{s}_o$ by exchanging messages with its immediate (single-hop) neighbors only.

III. Distributed Recursive Least-Squares Scheme

Our approach is to rewrite the minimization problem in (1) in an equivalent form that is amenable to distributed implementation. Then, we utilize the alternating-direction method of multipliers [1, pg. 253-261],[7], to split the original problem in (1) into simpler subtasks that can be implemented in parallel. Since summands in (1) are coupled through $\mathbf{s}$ it is not straightforward to decompose this minimization problem. To this end, we define the auxiliary variable $\mathbf{s}_j$ to represent the local estimate at sensor j, and consider the constrained minimization problem

$$\begin{aligned} \{\hat{\mathbf{s}}_j(t)\} := \arg\min \textstyle\sum_{\tau=0}^{t} \sum_{j=1}^{J} \lambda^{t-\tau} \|x_j(\tau) - \mathbf{h}_j^T(\tau)\mathbf{s}_j\|^2 \\ + J^{-1}\lambda^t \textstyle\sum_{j=1}^{J} \mathbf{s}_j^T \mathbf{\Phi}_o \mathbf{s}_j \qquad (2) \\ \text{s. to } \ \epsilon_j \mathbf{s}_j = \epsilon_j \bar{\mathbf{s}}_b, \ b \in \mathcal{B}, j \in \mathcal{N}_b \end{aligned}$$

where $\mathcal{B} \subseteq [1, J]$ is a subset of 'bridge' sensors maintaining local vectors $\bar{\mathbf{s}}_b$ which are utilized to impose consensus among local estimates $\mathbf{s}_j$ across all sensors. If $\mathcal{B} \equiv [1, J]$, then from the connectivity of the WSN it follows that (1) and (2) are equivalent in the sense that $\hat{\mathbf{s}}_j(t) = \hat{\mathbf{s}}_{\text{ewls}}(t)$. In fact we can impose a milder requirement on $\mathcal{B}$ and ensure that (1) and (2) are equivalent.

Specifically, the bridge sensor set is chosen to satisfy the following conditions: (i) $\forall\, j \in [1, J]$ there exists at least one $b \in \mathcal{B}$ such that $b \in \mathcal{N}_j$ (the bridge neighbors of sensor j will be denoted by $\mathcal{B}_j := \mathcal{N}_j \cap \mathcal{B}$); and, (ii) for j_1 and $j_2 \in \mathcal{N}_{j_1}$ there exists $b \in \mathcal{B}$ such that $b \in \mathcal{N}_{j_1} \cap \mathcal{N}_{j_2}$. Selecting $\mathcal{B}$ in that way ensures that $\hat{\mathbf{s}}_j(t) = \hat{\mathbf{s}}_{\text{ewls}}(t)$ (details in [8]). Further, such a bridge sensor set can be determined in a distributed fashion using e.g., the scheme in [9]. A possible selection for $\mathcal{B}$ (not unique) in the WSN given in Fig. 1, is the set formed by the black nodes. In fact the bridge-sensor set $\mathcal{B}$ trades-off communication cost for robustness to sensor failures; i.e., increasing the number of bridge sensors improves robustness to sensor failures but also increases the information exchange between sensors. Regarding the positive constants ϵ_j, they do not cause any effect whatsoever on the constraints in (2). Actually, they are used to adjust the convergence characteristics of the distributed (D-) RLS algorithm.

A. Algorithmic Construction

In this subsection, we will show how to solve (2) in a distributed fashion using the alternating-direction method of multipliers. Interestingly, this procedure will yield a distributed adaptive estimation algorithm whereby recursive updates per sensor allow estimation of $\mathbf{s}_o$, or, even tracking of a time-varying process $\mathbf{s}_o(t)$.

Let $\{\mathbf{v}_j^b\}_{j\in[1,J]}^{b\in\mathcal{B}_j}$ denote the Lagrange multipliers associated with the constraints $\mathbf{s}_j = \bar{\mathbf{s}}_b$, and be updated at the j-th sensor. Consider now the augmented Lagrangian function of the minimization problem in (2) at time instant $t+1$, namely

$$\begin{aligned} \mathcal{L}_a(\boldsymbol{s}, \bar{\boldsymbol{s}}_b, \boldsymbol{v}; t+1) = \sum_{\tau=0}^{t+1}\sum_{j=1}^{J} \lambda^{t+1-\tau} \|x_j(\tau) - \mathbf{h}_j^T(\tau)\mathbf{s}_j\|^2 \\ + J^{-1}\lambda^{t+1} \textstyle\sum_{j=1}^{J} \mathbf{s}_j^T \mathbf{\Phi}_o \mathbf{s}_j + \displaystyle\sum_{b\in\mathcal{B}}\sum_{j\in\mathcal{N}_b} (\mathbf{v}_j^b)^T \epsilon_j (\mathbf{s}_j - \bar{\mathbf{s}}_b) \\ + \sum_{b\in\mathcal{B}}\sum_{j\in\mathcal{N}_b} \frac{c_j \epsilon_j^2}{2} \|\mathbf{s}_j - \bar{\mathbf{s}}_b\|_2^2 \qquad (3) \end{aligned}$$

where $\boldsymbol{s} := \{\mathbf{s}_j\}_{j=1}^J$, $\bar{\boldsymbol{s}}_b := \{\bar{\mathbf{s}}_b\}_{b\in\mathcal{B}}$, $\boldsymbol{v} := \{\mathbf{v}_j^b\}_{j\in[1,J]}^{b\in\mathcal{B}_j}$, and $c_j > 0$ are penalty coefficients corresponding to the constraint $\epsilon_j \mathbf{s}_j = \epsilon_j \bar{\mathbf{s}}_b$, $\forall\, b \in \mathcal{B}$ and $j \in [1, J]$.

We tackle (2) using the alternating-direction method of multipliers [1], by: s1) minimizing the augmented Lagrangian function using block coordinate descent; and, s2) updating the Lagrange multipliers associated with the equality constraints in (2). To this end, let $k = 0, 1, \ldots$ denote the iteration index for the recursive algorithm to be constructed in order to minimize (2). In order to apply the alternating-direction method of multipliers suppose that during time iteration k the multipliers $\mathbf{v}_j^b(t+1; k)$ and local estimates $\mathbf{s}_j(t+1; k)$ are available at sensor j. Then, (s1) involves the following two subtasks:

s1-a) After setting $\{\mathbf{v}_j^b = \mathbf{v}_j^b(t+1; k)\}_{j\in[1,J]}^{b\in\mathcal{B}_j}$ and $\{\mathbf{s}_b = \mathbf{s}_b(t+1; k)\}_{b\in\mathcal{B}}$ in (3), minimize the augmented Lagrangian in (3) with respect to $\mathbf{s}_j$ and determine $\mathbf{s}_j(t+1; k+1)$ for $j = 1, \ldots, J$.

s1-b) After setting $\mathbf{v}_j^b = \mathbf{v}_j^b(t+1; k)$ and $\mathbf{s}_j = \mathbf{s}_j(t+1; k+1)$ in (3), minimize the augmented Lagrangian in (3) with respect

to $\bar{\mathbf{s}}_b$ and determine $\bar{\mathbf{s}}_b(t+1;k+1)$ for $b \in \mathcal{B}$.
Then, step (s2) involves updating the Lagrange multipliers using a gradient ascent-like recursion.

Interestingly, after applying steps (s1)-(s2) we obtain a set of recursions that involve communications only between single-hop neighboring sensors. Thus, the recursions used to develop the distributed (D-) RLS algorithm are summarized as

$$\mathbf{v}_j^b(t+1;k) = \mathbf{v}_j^b(t+1;k-1) + \epsilon_j c_j(\mathbf{s}_j(t+1;k) - \bar{\mathbf{s}}_b(t+1;k)) \quad (4)$$

$$\mathbf{s}_j(t+1;k+1) = \mathbf{\Phi}_j^{-1}(t+1)\boldsymbol{\psi}_j(t+1) - \frac{\epsilon_j}{2}\mathbf{\Phi}_j^{-1}(t+1) \times (\sum_{b\in\mathcal{B}_j} \mathbf{v}_j^b(t+1;k) - \epsilon_j c_j \sum_{b\in\mathcal{B}_j} \bar{\mathbf{s}}_b(t+1;k)) \quad (5)$$

$$\bar{\mathbf{s}}_b(t+1;k+1) = \sum_{j\in\mathcal{N}_b} \frac{1}{\sum_{\beta\in\mathcal{N}_b} c_\beta}(\epsilon_j^{-1}\mathbf{v}_j^b(t+1;k) + c_j\mathbf{s}_j(t+1;k+1)), \quad (6)$$

where $b \in \mathcal{B}_j$ in (4) and $b \in \mathcal{B}$ in (6), while

$$\mathbf{\Phi}_j(t+1) := \sum_{\tau=0}^{t+1} \lambda^{t+1-\tau}\mathbf{h}_j(\tau)\mathbf{h}_j^T(\tau) + \lambda^{t+1}\mathbf{\Phi}_o + \frac{\epsilon_j^2 c_j|\mathcal{B}_j|}{2}\mathbf{I}_p$$

$$\boldsymbol{\psi}_j(t+1) := \sum_{\tau=0}^{t+1} \lambda^{t+1-\tau}\mathbf{h}_j(\tau)x_j(\tau) \ . \quad (7)$$

Notice that $\boldsymbol{\psi}_j(t+1) = \lambda\boldsymbol{\psi}_j(t) + \mathbf{h}_j(t+1)x_j(t+1)$, while for $\lambda = 1$, matrix $\mathbf{\Phi}_j^{-1}(t+1)$ can be obtained recursively with a complexity of $\mathcal{O}(p^2)$ from $\mathbf{\Phi}_j^{-1}(t)$ using the matrix inversion lemma. Note that the first term in $\mathbf{s}_j(t+1)$, namely $\mathbf{\Phi}_j(t+1)^{-1}\boldsymbol{\psi}_j(t+1)$, is a regularized version of the local EW-LS estimator at sensor j at time instant $t+1$. The regularization comes from the term $\frac{\epsilon_j^2 c_j|\mathcal{B}_j|}{2}$ in $\mathbf{\Phi}_j(t+1)$. The second term in (5) is responsible for fusing information from the neighborhood of sensor j, refining in that way the estimate provided from $\mathbf{\Phi}_j^{-1}(t+1)\boldsymbol{\psi}_j(t+1)$.

Recursions (4)-(6) constitute the D-RLS algorithm whereby all the sensors $j \in [1,J]$ keep track of their local estimate $\mathbf{s}_j(t+1;k)$ and the Lagrange multipliers $\mathbf{v}_j^b(t+1;k)$ for $b \in \mathcal{B}_j$. The sensors that belong to subset $\mathcal{B}$ keep also track of the consensus enforcing variables $\bar{\mathbf{s}}_b(t+1;k)$. Note that at each time instant t, the EW-LSE cost in (2) changes. The recursions provided in (4)-(6) provide a way to minimize (2) in a distributed fashion, and we have shown that (details in [8])

Proposition 1: *For arbitrarily initialized* $\{\mathbf{v}_j^b(t+1;0)\}_{j=[1,J]}^{b\in\mathcal{B}_j}$, $\mathbf{s}_j(t+1;0)$ *and* $\bar{\mathbf{s}}_b(t+1;0)$*, and as* $k \to \infty$ *the local estimates* $\mathbf{s}_j(t+1;k)$ *reach consensus; i.e.,*

$$\lim_{k\to\infty} \mathbf{s}_j(t+1;k) = \hat{\mathbf{s}}_j(t+1) = \hat{\mathbf{s}}_{ewls}(t+1), \quad \textit{for } j \in [1,J].$$

Thus, the D-RLS recursions are able to determine the EW-LS estimator at each time instant t as long as the number of 'consensus' recursions k goes to infinity. For a time-invariant setup applying many consensus iterations, i.e. $k \gg 1$ would not be a problem, though this is not the case when the WSN has to track a time-varying process $\mathbf{s}_o(t)$. Thus, in order to make D-RLS appropriate for time-critical applications we can apply one 'consensus' iteration per time instant t (in [8] we also consider and analyze D-RLS versions with more 'consensus' iterations). In that case $t = k$ and the recursions in (4)-(6) can be simplified to (proof in [8])

$$\mathbf{v}_j^b(t) = \mathbf{v}_j^b(t-1) + \epsilon_j c_j(\mathbf{s}_j(t) - \bar{\mathbf{s}}_b(t)) \quad (8)$$

$$\mathbf{s}_j(t+1) = \mathbf{\Phi}_j^{-1}(t+1)\boldsymbol{\psi}_j(t+1) - \frac{\epsilon_j}{2}\mathbf{\Phi}_j^{-1}(t+1)(\sum_{b\in\mathcal{B}_j}\mathbf{v}_j^b(t) - \epsilon_j c_j \sum_{b\in\mathcal{B}_j}\bar{\mathbf{s}}_b(t)) \quad (9)$$

$$\bar{\mathbf{s}}_b(t+1) = \sum_{j\in\mathcal{N}_b} \frac{1}{\sum_{\beta\in\mathcal{N}_b} c_\beta}(\epsilon_j^{-1}\mathbf{v}_j^b(t) + c_j\mathbf{s}_j(t+1)). \quad (10)$$

During time instant $t+1$ sensor j receives the consensus variable $\bar{\mathbf{s}}_b(t)$ from its bridge neighbors within $\mathcal{B}_j$, and updates its Lagrange multipliers $\{\mathbf{v}_j^b(t)\}_{b\in\mathcal{B}_j}$ using (8), which are used next to compute $\mathbf{s}_j(t+1)$ through (9). After completing this iteration step, sensor j transmits to each of its bridge neighbors $b \in \mathcal{B}_j$ the vector $\epsilon_j^{-1}\mathbf{v}_j^b(t) + c_j\mathbf{s}_j(t+1)$. Subsequently, each sensor $b \in \mathcal{B}$ receives the vectors $\epsilon_j^{-1}\mathbf{v}_j^b(t) + c_j\mathbf{s}_j(t+1)$ from all its neighbors $j \in \mathcal{N}_b$ and proceeds to compute $\bar{\mathbf{s}}_b(t+1)$ using (10). This completes the $(t+1)$-st iteration and all the sensors in $\mathcal{B}$ proceed to transmit $\bar{\mathbf{s}}_b(t+1)$ to all their neighbors $j \in \mathcal{N}_b$ starting the $(t+2)-$nd iteration. The algorithm is tabulated as Algorithm 1.

Note that D-RLS allows the online incorporation of new observation data at each time instant via $\boldsymbol{\psi}_j(t+1)$ and $\mathbf{\Phi}_j(t+1)$ in (9). In that way D-RLS has the potential to efficiently track time-varying processes. This will also be confirmed by the simulations of Section IV. On the other hand, the I-RLS in [6] allows each sensor to process new data only after J time slots from the time instant it acquired them. Thus, the operation of I-RLS is limited to applications with slow varying parameters or fast sensor communications. Further, operation of the I-RLS requires determination of a Hamiltonian cycle of the WSN which is an NP-complete problem [4]. In case of sensor failures redetermining a Hamiltonian cycle might be extremely difficult if not impossible. Also connectivity of the WSN is not sufficient to guarantee existence of a Hamiltonian cycle.

The communication cost for D-RLS is $\mathcal{O}(p)$ per iteration, while the one for I-RLS is $\mathcal{O}(p^2)$. A low communication cost I-RLS is also proposed in [6] with a communication cost of $\mathcal{O}(p)$ per iteration, though the tracking challenges related to I-RLS still remain. The communication cost associated with diffusion RLS in [2] is also $O(p)$. However, we should point out that sensors not only have to exchange their local estimates with all their single-hop neighbors, but also exchange their regressor vectors and observation data. Recall that in D-RLS each sensor j does not have to transmit its regressors and observations, while it exchanges information only with a subset of its neighborhood $\mathcal{N}_j$, namely the bridge neighbors in $\mathcal{B}_j$, where $|\mathcal{B}_j| < |\mathcal{N}_j|$. This implies that D-RLS incurs smaller communication cost than [2]. Further, as will be seen

Algorithm 1 D-RLS ($t = k$)

Initialize $\{\mathbf{s}_j(0)\}_{j=1}^{J}$, $\{\bar{\mathbf{s}}_b(0)\}_{b\in\mathcal{B}}$ and $\{\mathbf{v}_j^b(0)\}_{j\in[1,J]}^{b\in\mathcal{B}_j}$ at any value.

for $t = 0, 1, \ldots$ **do**

  Every bridge sensor $b \in \mathcal{B}$: transmits $\bar{\mathbf{s}}_b(t)$ to its neighbors in $\mathcal{N}_b$

  All $j \in [1, J]$: update $\mathbf{v}_j^b(t)$ using (8).

  All $j \in [1, J]$: update $\mathbf{s}_j(t+1)$ using (9).

  All $j \in [1, J]$: transmit $\epsilon_j^{-1}\mathbf{v}_j^b(t) + c_j\mathbf{s}_j(t+1)$ to each $b \in \mathcal{B}_j$

  Bridge sensors $b \in \mathcal{B}$: compute $\bar{\mathbf{s}}_b(t+1)$ through (10).

end for

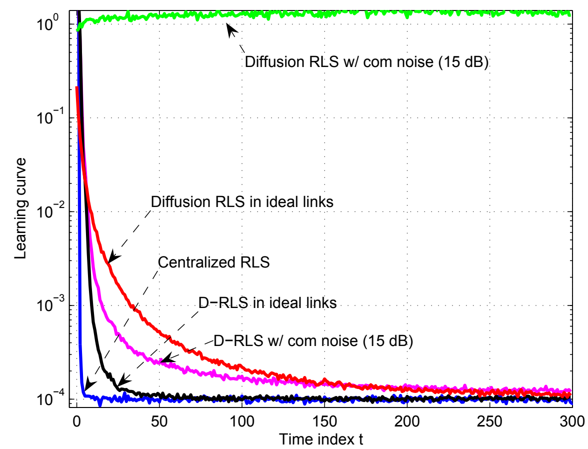
in Section IV D-RLS exhibits robustness in the presence of reception (communication) noise, whereas the distributed scheme in [2] may not provide accurate local estimates.

Remark 1: In case of a bridge sensor failure, D-RLS incurs performance loss, but remains operational after the neighbors of the failed bridge sensor modify their local recursions accordingly. Specifically, if bridge sensor $b' \in \mathcal{B}$ fails, then some of the nodes in $\mathcal{N}_{b'}$ can be converted to bridges as needed, in order for the new bridge sensor set, call it $\mathcal{B}'$, to satisfy the properties of $\mathcal{B}$. This conversion can be accommodated using the algorithm in [9]. Then, all sensors in $\mathcal{N}_{b'}$ can modify their local recursions (8)-(10) by adding the corresponding terms associated with the new bridges in $\mathcal{N}_{b'}$, and removing the ones corresponding to b'.

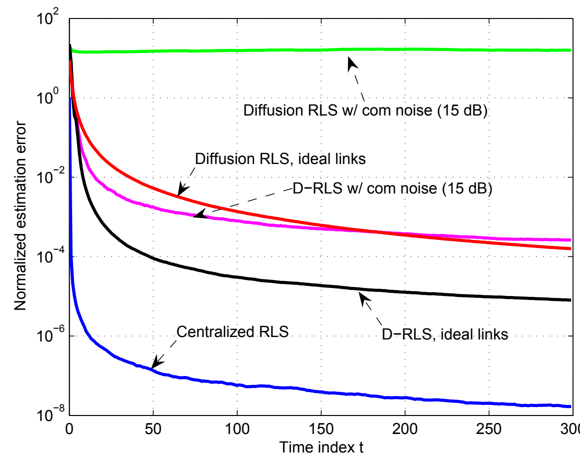
IV. Numerical Examples

Next, we test the convergence properties of the D-RLS algorithm in an ad hoc WSN and perform some comparisons with the schemes in [2] and [6]. The WSN comprises $J = 30$ sensors. We adopt a linear data model of the form $x_j(t) = \mathbf{h}_j(t)\mathbf{s}_o + \mathbf{n}_j(t)$, with $p = 4$, $\sigma_{n_j}^2 = 10^{-4}$and with the entries of $\mathbf{h}_j(t)$ uniformly distributed in $[-0.5, 0.5]$ for $j = 1, \ldots, J$. We set $\lambda = 1$, $\epsilon_j c_j = 8$ and $\epsilon_j^2 c_j = 0.05$ for all $j = 1, \ldots, 30$. Fig. 2 (a) depicts the normalized mean-square error $E_{\mathrm{l}}(t) = J^{-1}\sum_{j=1}^{J}\hat{E}[(x_j(t) - \mathbf{h}_j^T(t)\mathbf{s}_j(t))^2]$ (learning curve) versus time index t, whereas Fig. 2 (b) shows the normalized estimation error $E_{\mathrm{e}}(t) = J^{-1}\sum_{j=1}^{J}\hat{E}[\|\mathbf{s}_j(t) - \mathbf{s}_o\|^2]$, with $\hat{E}[\cdot]$ indicating mean approximation using Monte Carlo simulations. We plot the learning curves and estimation error for D-RLS, diffusion RLS in [2] and the centralized RLS assuming i) ideal links; and ii) reception noise at all sensors. Clearly, the centralized RLS, where all the information is assumed available at a central location, benchmarks both D-RLS and diffusion RLS. Next, note that the learning curve corresponding to D-RLS converges to the observation noise variance $\sigma_{n_j}^2 = 10^{-4}$ much faster than the one corresponding to the diffusion RLS. Note also that the learning curve of D-RLS is very close to the centralized RLS when sensor links are ideal. Also, Figs. 2 (a)-(b) depict that in the presence of communication noise, diffusion RLS clearly cannot provide an accurate estimate of $\mathbf{s}_o$, whereas D-RLS still is able to estimate $\mathbf{s}_o$ with some penalty though with a higher estimation error than the one achieved under ideal sensor links.

Fig. 3 depicts the tracking capability of D-RLS, assuming that $\mathbf{s}_o(t)$ obeys a first-order AR process; i.e., $\mathbf{s}_o(t) = 0.9\mathbf{s}_o(t-1) + \mathbf{v}(t)$, where $\mathbf{v}(t)$ is zero-mean white Gaussian with variance 0.1. We plot the local estimate $[\mathbf{s}_2(t)]_1$ (first entry of $\mathbf{s}_2(t)$) for both D-RLS and I-RLS, where $\lambda = 0.5$, along with $[\mathbf{s}_o(t)]_1$. Recall that in I-RLS each sensor produces an estimate once per $J = 30$ time slots. Clearly, the D-RLS outperforms the I-RLS in terms of tracking.



(a)



(b)

Fig. 2. *(a)* Normalized mean-square error $E_{\mathrm{l}}(t)$ vs. time index t; *(b)* Normalized estimation error $E_{\mathrm{e}}(t)$ vs. time index t.

V. Conclusions

We developed a distributed RLS algorithm for estimating and tracking signals using observation data collected online by sensors deployed in an ad hoc manner. Our approach involves reformulating in a separable constrained form the exponentially weighted squared-error cost whose minimization leads to the standard RLS algorithm, and then minimizing

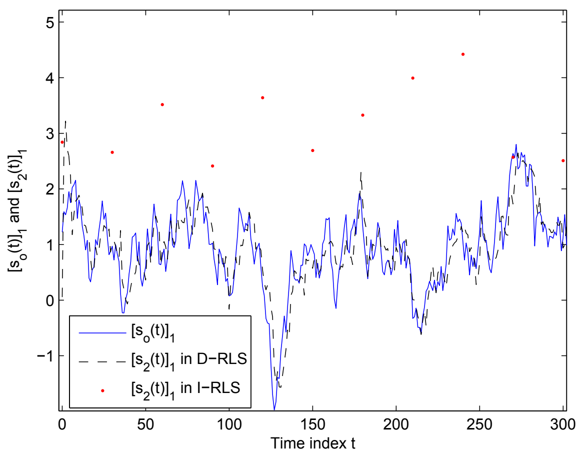


Fig. 3. Tracking with D-RLS and I-RLS.

the new cost in a distributed fashion across sensors via the alternating-direction method of multipliers. The novel D-RLS algorithm can be applied to general WSNs and performs efficient estimation and tracking even in the presence of communication noise. Numerical examples demonstrate the convergence and tracking advantages of D-RLS over existing alternatives.

Currently, we are pursuing stability and performance analysis of the D-RLS algorithm for general setups using stochastic averaging techniques. The goal is to show that the local estimates $\mathbf{s}_j(t)$ in D-RLS enjoy mean-square sense consistency whenever the same holds true for the standard RLS scheme.[1]

REFERENCES

[1] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Athena Scientific, Second Edition, 1999.

[2] F. S. Cattivelli, C. G. Lopes, A. H. Sayed, "A Diffusion RLS Scheme for Distributed Estimation Over Adaptive Networks," *Proc. of Sig. Proc. Advances in Wir. Com.*, Helsinki, Finland, June 17- 20, 2007.

[3] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, Fourth Edition, 2002.

[4] C. H. Papadimitriou, *Computational Complexity*, Addison-Wesley, 1994.

[5] A. H. Sayed and C. G. Lopes, "Adaptive estimation algorithms over distributed networks," *Proc. 21st IEICE Sig. Proc. Sympos.*, Kyoto, November 2006.

[6] A. H. Sayed and C. G. Lopes, "Distributed Recursive Least-Squares Strategies Over Adaptive Networks," *Proc. 40th Asilomar Conference on Signals, Systems and Computers*, pp. 233-237, Pacific Grove, CA, October 2006.

[7] I. D. Schizas, A. Ribeiro and G. B. Giannakis, "Consensus in Ad Hoc WSNs with Noisy Links - Part I: Distributed Estimation of Deterministic Signals", *IEEE Transactions on Signal Processing*, Dec. 2007.

[8] I. D. Schizas, G. Mateos and G. B. Giannakis, "Distributed Recursive Least-Squares for Consensus-Based Adaptive In-Network Processing", *IEEE Transactions on Signal Processing*, submitted 2008.

[9] J. Wu and H. Li, "A Dominating-Set-Based Routing Scheme in Ad Hoc Wireless Networks," *Telecommunication Systems Journal*, Vol. 3, pp. 63–84, Sep. 2001.

[10] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," *Proc. of the 4th Intl. Symposium on Info. Proc. in Sensor Networks*, pp. 63–70, Berkeley, CA, 2005.

[1]The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies of the Army Research Laboratory or the U. S. Government.