DISTRIBUTED REDUCED-RANK ESTIMATION BASED ON JOINT ITERATIVE OPTIMIZATION IN SENSOR NETWORKS

Songcen Xu[*]*, Rodrigo C. de Lamare*[†] *and H. Vincent Poor*[✠]

[*]Communications Research Group, Department of Electronics, University of York, U.K.
[†]CETUC, PUC-Rio, Brazil / Dept. of Electronics, University of York, UK
[✠]Dept. of Electrical Engineering, Princeton University, Princeton, NJ 08544, USA
Emails: sx520@york.ac.uk, rcdl500@york.ac.uk, poor@princeton.edu

ABSTRACT

This paper proposes a novel distributed reduced–rank scheme and an adaptive algorithm for distributed estimation in wireless sensor networks. The proposed distributed scheme is based on a transformation that performs dimensionality reduction at each agent of the network followed by a reduced-dimension parameter vector. A distributed reduced-rank joint iterative estimation algorithm is developed, which has the ability to achieve significantly reduced communication overhead and improved performance when compared with existing techniques. Simulation results illustrate the advantages of the proposed strategy in terms of convergence rate and mean square error performance.

Index Terms— Dimensionality reduction, distributed estimation, reduced–rank methods, wireless sensor networks

1. INTRODUCTION

Distributed strategies have become fundamental for parameter estimation in wireless networks and applications such as sensor networks [1, 2] and smart grids [3, 4]. Distributed processing techniques deal with the extraction of information from data collected at nodes that are distributed over a geographic area [1]. In this context, a specific node or agent in the network collects data from its neighbors and combines them with its local information to generate an improved estimate. However, when the unknown parameter vector to be estimated has a large dimension, the network requires a large communication bandwidth between neighbor nodes to transmit their local estimate. This problem limits the application of existing algorithms in applications with large data sets as the convergence speed is dependent on the length of the parameter vector. Hence, distributed dimensionality reduction has become an important tool for distributed inference problems.

In order to perform dimensionality reduction, many algorithms have been proposed in the literature, in the context of distributed quantized Kalman Filtering [5, 6], quantized consensus algorithms [7], distributed principal subspace estimation [8], single bit strategy [9] and Krylov subspaces optimization techniques [10]. However, existing algorithms are either too costly or have unsatisfactory performance when processing a large number of parameters. As a result, trade-offs between the amount of cooperation, communication and system performance naturally exist. In this context, reduced–rank techniques are powerful tools to perform dimensionality reduction, which have been applied to DS–CDMA system [11], multi–input–multi–output (MIMO) equalization application [12], spread–spectrum systems [13] and space–time interference suppression [14]. However, limited research has been carried out on distributed reduced-rank estimation. Related approaches to reduced-rank techniques include compressive sensing-based strategies [15], which exploit sparsity to reduce the number of parameters for estimation, and attribute-distributed learning [16], which employs agents and a fusion center to meet the communication constraints.

In this paper, we propose a scheme for distributed signal processing along with a distributed reduced-rank algorithm for parameter estimation. In particular, the proposed algorithm is based on an alternating optimization strategy and is called distributed reduced-rank joint iterative optimization normalized least mean squares (DRJIO–NLMS) algorithm. In contrast to prior work on reduced-rank techniques and distributed methods, the proposed reduced-rank strategy is distributed and performs dimensionality reduction without costly decompositions at each agent. The proposed DRJIO-NLMS algorithm is flexible with regards to the amount of information that is exchanged, has low cost and high performance. The transmitted information between each neighbor involves a dimensionality reduction matrix and a reduced-dimension parameter vector. The DRJIO-NLMS algorithm can also outperform competing techniques.

This paper is organized as follows. Section 2 describes the system model. In Section 3, the distributed dimensionality reduction and adaptive processing scheme are introduced. The proposed distributed reduced-rank algorithm is illustrated in Section 4. Simulation results are provided in Section 5. Finally, we conclude the paper in Section 6.

Notation: We use boldface uppercase letters to denote matrices and boldface lowercase letters to denote vectors. We use $(\cdot)^{-1}$ to denote the inverse operator, $(\cdot)^H$ for conjugate transposition and $(\cdot)^*$ for complex conjugate.

2. SYSTEM MODEL

A distributed wireless sensor network with N nodes, which have limited processing capabilities, is considered with a partially connected topology. A diffusion protocol is employed although other strategies, such as incremental [1] and consensus-based [3] could also be used. A partially connected network means that nodes can exchange information only with their neighbors determined by the connectivity topology. In contrast, a fully connected network means that, data broadcast by a node can be captured by all other nodes in the network [17]. At every time instant i, each node k takes a scalar measurement $d_k(i)$ according to

$$d_k(i) = \boldsymbol{\omega}_0^H \boldsymbol{x}_k(i) + n_k(i), \quad i = 1, 2, \ldots, \mathrm{N}, \tag{1}$$

where $\boldsymbol{x}_k(i)$ is the $M \times 1$ input signal vector with zero mean and variance $\sigma_{x,k}^2$, $n_k(i)$ is the noise sample at each node with zero mean and variance $\sigma_{n,k}^2$. Through (1), we can see that the measurements for all nodes are related to an unknown parameter vector $\boldsymbol{\omega}_0$ with size $M \times 1$, that would be estimated by the network. Fig.1 shows an example for a diffusion–type wireless network with 20 nodes. The aim of such a network is to compute an estimate of $\boldsymbol{\omega}_0$ in a distributed fashion, which can minimize the cost function

$$J_\omega(\boldsymbol{\omega}) = \mathbb{E}[|d_k(i) - \boldsymbol{\omega}^H \boldsymbol{x}_k(i)|^2], \tag{2}$$

where $\mathbb{E}[\cdot]$ denotes the expectation operator. To solve this problem, one possible technique is the adapt–then–combine (ATC) diffusion strategy [2]

$$\begin{cases} \boldsymbol{\psi}_k(i) = \boldsymbol{\omega}_k(i-1) + \mu_k \boldsymbol{x}_k(i)\big[d_k(i) - \boldsymbol{\omega}_k^H(i-1)\boldsymbol{x}_k(i)\big]^*, \\ \boldsymbol{\omega}_k(i) = \sum\limits_{l \in \mathcal{N}_k} c_{kl} \boldsymbol{\psi}_l(i), \end{cases} \tag{3}$$

where $\mathcal{N}_k$ indicates the set of neighbors for node k, $|\mathcal{N}_k|$ denotes the cardinality of $\mathcal{N}_k$ and c_{kl} is the combination coefficient, which is calculated under the Metropolis rule

$$\begin{cases} c_{kl} = \frac{1}{max(|\mathcal{N}_k|, |\mathcal{N}_l|)}, & \text{if } k \neq l \text{ are linked} \\ c_{kl} = 0, & \text{for } k \text{ and } l \text{ not linked} \\ c_{kk} = 1 - \sum\limits_{l \in \mathcal{N}_k / k} c_{kl}, & \text{for } k = l \end{cases} \tag{4}$$

and should satisfy

$$\sum_l c_{kl} = 1, l \in \mathcal{N}_k \forall k. \tag{5}$$

With this strategy, when the dimension of the unknown parameter vector $\boldsymbol{\omega}_0$ is large, it could lead to a high communication overhead between each neighbor node and the convergence speed is reduced. In order to solve this problem and optimize the distributed processing, we incorporate at the kth node of the network distributed reduced–rank strategies based on alternating optimization techniques.

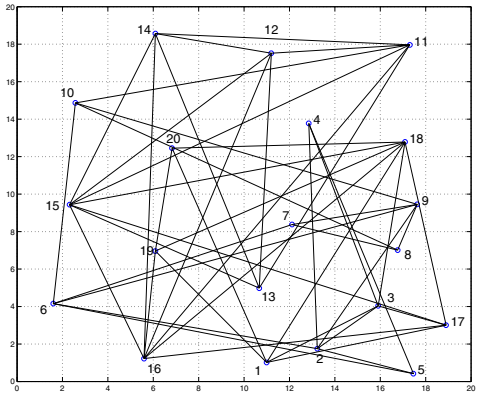

Fig. 1. Network topology with 20 nodes

3. DISTRIBUTED DIMENSIONALITY REDUCTION AND ADAPTIVE PROCESSING

The proposed distributed dimensionality reduction scheme, depicted in Fig.2, employs a transformation matrix $\boldsymbol{S}_{D_k}(i)$ to process the input signal $\boldsymbol{x}_k(i)$ with dimensions $M \times 1$ and projects it onto a lower $D \times 1$ dimensional subspace $\bar{\boldsymbol{x}}_k(i)$, where $D \ll M$. Following this procedure, a reduced–rank estimator $\bar{\boldsymbol{\omega}}_k(i)$ is computed, and the $\bar{\boldsymbol{\omega}}_k(i)$ is transmitted through each neighbor node. In particular, the transformation matrix $\boldsymbol{S}_{D_k}(i)$ and reduced–rank estimator $\bar{\boldsymbol{\omega}}_k(i)$ will be jointly optimized in the proposed scheme according to the minimum mean–squared error (MMSE) criterion.

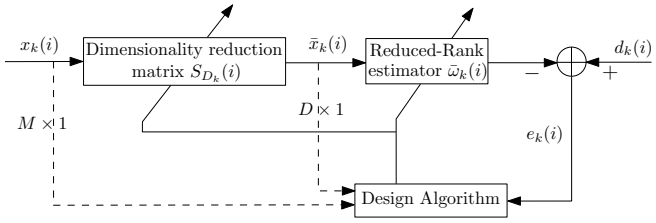

Fig. 2. Proposed dimensionality reduction scheme at each node or agent.

Specifically, we start the description of the method with an $M \times D$ matrix $\boldsymbol{S}_{D_k}(i)$, which carries out a dimensionality reduction on the input signal as given by

$$\bar{\boldsymbol{x}}_k(i) = \boldsymbol{S}_{D_k}^H(i) \boldsymbol{x}_k(i), \tag{6}$$

where, in what follows, all D–dimensional quantities are denoted with a 'bar'. The design of $\boldsymbol{S}_{D_k}(i)$ and $\bar{\boldsymbol{\omega}}_k(i)$ corresponds to the optimization problem given by

$$\left\{\boldsymbol{S}_{D_k}^{\mathrm{opt}}, \bar{\boldsymbol{\omega}}_k^{\mathrm{opt}}\right\} = \min_{\boldsymbol{S}_{D_k}(i), \bar{\boldsymbol{\omega}}_k(i)} \mathbb{E}[|d_k(i) - \bar{\boldsymbol{\omega}}_k{}^H(i) \boldsymbol{S}_{D_k}^H(i) \boldsymbol{x}_k(i)|^2] \tag{7}$$

where $\bar{\boldsymbol{\omega}}_k(i)$ is the reduced–rank estimator. By fixing $\boldsymbol{S}_{D_k}(i)$ and minimizing (7) with respect to $\bar{\boldsymbol{\omega}}_k(i)$, we have

$$\bar{\boldsymbol{\omega}}_k(i) = \bar{\boldsymbol{R}}_k^{-1}(i) \bar{\boldsymbol{p}}_k(i), \tag{8}$$

where $\bar{\boldsymbol{R}}_k(i) = \mathbb{E}[\boldsymbol{S}_{D_k}^H(i)\boldsymbol{x}_k(i)\boldsymbol{x}_k^H(i)\boldsymbol{S}_{D_k}(i)] = \mathbb{E}[\bar{\boldsymbol{x}}_k(i)\bar{\boldsymbol{x}}_k^H(i)]$ and $\bar{\boldsymbol{p}}_k(i) = \mathbb{E}[d_k^*(i)\boldsymbol{S}_{D_k}^H(i)\boldsymbol{x}_k(i)] = \mathbb{E}[d_k^*(i)\bar{\boldsymbol{x}}_k(i)]$. We then fix $\bar{\boldsymbol{\omega}}_k(i)$ and minimize (7) with respect to $\boldsymbol{S}_{D_k}(i)$, and arrive at the following expression

$$\boldsymbol{S}_{D_k}(i) = \boldsymbol{R}_k^{-1}(i)\boldsymbol{P}_{D_k}(i)\bar{\boldsymbol{R}}_{\bar{\boldsymbol{\omega}}_k}^{-1}(i), \tag{9}$$

where $\boldsymbol{R}_k(i) = \mathbb{E}[\boldsymbol{x}_k(i)\boldsymbol{x}_k^H(i)]$, $\boldsymbol{P}_{D_k}(i) = \mathbb{E}[d_k^*(i)\boldsymbol{x}_k(i)\bar{\boldsymbol{\omega}}_k^H(i)]$ and $\bar{\boldsymbol{R}}_{\bar{\boldsymbol{\omega}}_k}(i) = \mathbb{E}[\bar{\boldsymbol{\omega}}_k(i)\bar{\boldsymbol{\omega}}_k^H(i)]$. At this stage, the associated reduced–rank MMSE is described as

$$\mathbf{MMSE} = \sigma_{d_k}^2 - \bar{\boldsymbol{p}}_k^H(i)\bar{\boldsymbol{R}}_k^{-1}(i)\bar{\boldsymbol{p}}_k(i) \tag{10}$$

where $\sigma_{d_k}^2 = \mathbb{E}[|d_k(i)|^2]$. Because there is no closed-form expression for $\boldsymbol{S}_{D_k}(i)$ and $\bar{\boldsymbol{\omega}}_k(i)$ as they depend on each, we need a strategy to compute the parameters. The proposed strategy is based on an alternating optimization of $\boldsymbol{S}_{D_k}(i)$ and $\bar{\boldsymbol{\omega}}_k(i)$. In the next section, we develop a distributed reduced-rank algorithm to compute the parameters of interest.

4. PROPOSED DISTRIBUTED REDUCED-RANK ALGORITHM

In this section, we present the proposed distributed reduced–rank algorithm for distributed estimation, namely DRJIO–NLMS. Unlike prior work [8–10], the proposed algorithm does **NOT** require

- an $M \times M$ auto–correlation matrix of the input signal and an $M \times 1$ cross–correlation vector between the input signal and the desired signal used to build the Krylov subspace [10]
- Additional cost to perform eigen–decompositions [8]
- Extra adaptive processing at the local node [9]
- Costly convex optimization at the local node, which introduces extra complexity [10].

The DRJIO-NLMS algorithm is flexible, has a low cost and a very fast convergence speed. The proposed DRJIO-NLMS algorithm needs to optimize the parameters in (7) jointly. Therefore, it solves the following optimization problem in an alternating fashion:

$$\begin{aligned}\left[\boldsymbol{S}_{D_k}^{\text{opt}}, \bar{\boldsymbol{\omega}}_k^{\text{opt}}\right] = &\min_{\boldsymbol{S}_{D_k}(i), \bar{\boldsymbol{\omega}}_k^H(i)} ||\bar{\boldsymbol{\omega}}_k(i) - \bar{\boldsymbol{\omega}}_k(i-1)||^2 \\ &+ ||\boldsymbol{S}_{D_k}(i) - \boldsymbol{S}_{D_k}(i-1)||^2 \\ \text{subject to } &\bar{\boldsymbol{\omega}}_k^H(i)\boldsymbol{S}_{D_k}^H(i)\boldsymbol{x}_k(i) = d_k(i).\end{aligned} \tag{11}$$

Using the method of Lagrange multipliers and considering $\{\boldsymbol{S}_{D_k}, \bar{\boldsymbol{\omega}}_k\}$ jointly, we arrive at the following Lagrangian:

$$\begin{aligned}\mathcal{L}_k = &||\bar{\boldsymbol{\omega}}_k(i) - \bar{\boldsymbol{\omega}}_k(i-1)||^2 + ||\boldsymbol{S}_{D_k}(i) - \boldsymbol{S}_{D_k}(i-1)||^2 \\ &+ \Re[\lambda_1^*(d_k(i) - \bar{\boldsymbol{\omega}}_k^H(i)\boldsymbol{S}_{D_k}^H(i-1)\boldsymbol{x}_k(i))] \\ &+ \Re[\lambda_2^*(d_k(i) - \bar{\boldsymbol{\omega}}_k^H(i-1)\boldsymbol{S}_{D_k}^H(i)\boldsymbol{x}_k(i))],\end{aligned} \tag{12}$$

where , λ_1, λ_2 are scalar Lagrange multipliers, $||\cdot||$ denotes the Frobenius norm, and the operator $\Re[\cdot]$ retains the real part of the argument. By computing the gradient terms of (12) with respect to $\bar{\boldsymbol{\omega}}_k(i)$, $\boldsymbol{S}_{D_k}(i)$, λ_1 and λ_2, respectively, we obtain

$$\nabla_{\bar{\boldsymbol{\omega}}_k(i)}\mathcal{L} = 2\big(\bar{\boldsymbol{\omega}}_k(i) - \bar{\boldsymbol{\omega}}_k(i-1)\big) + \boldsymbol{S}_{D_k}^H(i-1)\boldsymbol{x}_k(i)\lambda_1 \tag{13}$$

$$\nabla_{\boldsymbol{S}_{D_k}(i)}\mathcal{L} = 2\big(\boldsymbol{S}_{D_k}(i) - \boldsymbol{S}_{D_k}(i-1)\big) + \boldsymbol{x}_k(i)\bar{\boldsymbol{\omega}}_k(i-1)\lambda_2 \tag{14}$$

$$\nabla_{\lambda_1}\mathcal{L} = d_k(i) - \bar{\boldsymbol{\omega}}_k^H(i)\boldsymbol{S}_{D_k}^H(i-1)\boldsymbol{x}_k(i) \tag{15}$$

$$\nabla_{\lambda_2}\mathcal{L} = d_k(i) - \bar{\boldsymbol{\omega}}_k^H(i-1)\boldsymbol{S}_{D_k}^H(i)\boldsymbol{x}_k(i). \tag{16}$$

By setting (13)–(16) to zero and solving the remaining equations, we obtain the recursions of the proposed DRJIO–NLMS algorithm described by

$$\bar{\boldsymbol{\omega}}_k(i) = \bar{\boldsymbol{\omega}}_k(i-1) + \mu(i)e_k^*(i)\bar{\boldsymbol{x}}_k(i) \tag{17}$$

$$\boldsymbol{S}_{D_k}(i) = \boldsymbol{S}_{D_k}(i-1) + \eta(i)e_k^*(i)\boldsymbol{x}_k(i)\bar{\boldsymbol{\omega}}_k^H(i-1) \tag{18}$$

where $e_k(i) = d_k(i) - \bar{\boldsymbol{\omega}}_k^H(i-1)\boldsymbol{S}_{D_k}^H(i-1)\boldsymbol{x}_k(i)$, $\mu(i) = \frac{\mu_0}{\boldsymbol{x}_k^H(i)\boldsymbol{x}_k(i)}$ and $\eta(i) = \frac{\eta_0}{\bar{\boldsymbol{\omega}}_k^H(i-1)\bar{\boldsymbol{\omega}}_k(i-1)\boldsymbol{x}_k^H(i)\boldsymbol{x}_k(i)}$ are the time–varying step sizes, while μ_0 and η_0 are the convergence factors. The recursions are computed in an alternating way with one iteration per time instant at each node.

The proposed DRJIO–NLMS algorithm includes two steps, namely adaptation step and combination step which are performed in an alternating procedure which is detailed next.

- Adaptation step

For the adaptation step, at each time instant i=1,2, . . . , I, each node k=1,2, . . . , N, starts from generating a local reduced–rank estimator through

$$\bar{\boldsymbol{\psi}}_k(i) = \bar{\boldsymbol{\omega}}_k(i-1) + \mu(i)e_k^*(i)\bar{\boldsymbol{x}}_k(i), \tag{19}$$

where $e_k(i) = d_k(i) - \bar{\boldsymbol{\omega}}_k^H(i-1)\boldsymbol{S}_{D_k}^H(i)\boldsymbol{x}_k(i)$. This local reduced–rank estimator $\bar{\boldsymbol{\psi}}_k(i)$ will be transmitted to all its neighbor nodes under the network topology structure.

Then, each node k=1,2, . . . , N, will update its dimensionality reduction matrix according to

$$\boldsymbol{S}_{D_k}(i) = \boldsymbol{S}_{D_k}(i-1) + \eta(i)e_k^*(i)\boldsymbol{x}_k(i)\bar{\boldsymbol{\omega}}_k(i-1), \tag{20}$$

and keep it locally.

- Combination step

At each time instant i=1,2, . . . , I, the combination step starts after the adaptation step finishes. Each node will combine the local reduced–rank estimators from its neighbor nodes and itself through

$$\bar{\boldsymbol{\omega}}_k(i) = \sum_{l \in \mathcal{N}_k} c_{kl}\bar{\boldsymbol{\psi}}_l(i), \tag{21}$$

to compute the reduced–rank estimator $\bar{\boldsymbol{\omega}}_k(i)$.

After the final iteration I, each node will generate the full–rank estimator $\boldsymbol{\omega}_k(I)$ from

$$\boldsymbol{\omega}_k(I) = \boldsymbol{S}_{D_k}(I)\bar{\boldsymbol{\omega}}_k(I). \tag{22}$$

In conclusion, during the distributed processing steps, only the local reduced–rank estimator $\bar{\boldsymbol{\psi}}_k(i)$ will be transmitted through the network. The proposed DRJIO–NLMS algorithm is detailed in Table.1.

Table 1. The DRJIO–NLMS Algorithm

Initialize: $\bar{\boldsymbol{\omega}}_k(0)$=0
For each time instant i=1,2, . . . , I
  For each node k=1,2, . . . , N
    $\bar{\boldsymbol{\psi}}_k(i) = \bar{\boldsymbol{\omega}}_k(i-1) + \mu(i)e_k^*(i)\bar{\boldsymbol{x}}_k(i)$
    where $e_k(i) = d_k(i) - \bar{\boldsymbol{\omega}}_k^H(i-1)\boldsymbol{S}_{D_k}^H(i)\boldsymbol{x}_k(i)$
    % $\bar{\boldsymbol{\psi}}_k(i)$ is the local reduced–rank estimator and will be
    % sent to all neighbor nodes of node k under the network
    % topology structure.
    $\boldsymbol{S}_{D_k}(i) = \boldsymbol{S}_{D_k}(i-1) + \eta(i)e_k^*(i)\boldsymbol{x}_k(i)\bar{\boldsymbol{\omega}}_k(i-1)$
    % The dimensionality reduction matrix $\boldsymbol{S}_{D_k}(i)$
    % will be updated and kept locally.
  end
  For each node k=1,2, . . . , N
    $\bar{\boldsymbol{\omega}}_k(i) = \sum_{l\in\mathcal{N}_k} c_{kl}\bar{\boldsymbol{\psi}}_l(i)$
    % The reduced–rank estimator $\bar{\boldsymbol{\omega}}_k(i)$
    % will be updated and kept locally.
  end
end
After the final iteration I
For each node k=1,2, . . . , N
  $\boldsymbol{\omega}_k(I) = \boldsymbol{S}_{D_k}(I)\bar{\boldsymbol{\omega}}_k(I)$
  where $\boldsymbol{\omega}_k(I)$ is the final full–rank estimator.
end

The computational complexity of the proposed DRJIO–NLMS algorithm is $O(DM)$. The distributed NLMS algorithm has a complexity $O(M)$, while the complexity of the distributed recursive least squares (RLS) algorithm [18] is $O(M^2)$. For both the Krylov Subspace NLMS [10] and distributed principal subspace estimation algorithms [8], the complexity reaches $O(M^3)$. Thus, the proposed DRJIO–NLMS algorithm has a much lower computational complexity, and because $D \ll M$, it is as simple as the distributed NLMS algorithm. In addition, the dimensionality reduction results in a decrease in the number of transmitted parameters from M to D which corresponds to a less stringent bandwidth requirement.

5. SIMULATION RESULTS

In this section, we compare our proposed DRJIO–NLMS algorithm with the distributed NLMS algorithm, distributed RLS algorithm [18], Krylov subspace NLMS [10] and distributed principal subspace estimation [8], based on their mean-squared error (MSE) performance. With the network topology structure outlined in Fig. 1 with $N = 20$ nodes, we consider numerical simulations under three scenarios

- Full–rank system with M=20
- Sparse system with M=20 (D valid coefficients and $M - D$ zeros coefficients)
- Full–rank system with M=60

The input signal is generated as $\boldsymbol{x}_k(i) = [x_k(i) \quad x_k(i-1) \quad \ldots \quad x_k(i-M+1)]$ and $x_k(i) = u_k(i) + \alpha_k x_k(i-1)$, where α_k is a correlation coefficient and $u_k(i)$ is a white noise process with variance $\sigma_{u,k}^2 = 1 - |\alpha_k|^2$, to ensure the variance of $\boldsymbol{x}_k(i)$ is $\sigma_{x,k}^2 = 1$. The noise samples are modeled as complex Gaussian noise with variance of $\sigma_{n,k}^2 = 0.001$. We assume that the network has perfect transmission between linked nodes.

The step size μ_0 for the distributed NLMS algorithm, Krylov subspace NLMS, distributed principal subspace estimation and DRJIO–NLMS is 0.15 and the η_0 is set to 0.5. For the distributed RLS algorithm, the forgetting factor λ is equal to 0.99 and the δ is 0.11. In Fig. 3, we compare the proposed DRJIO–NLMS with the existing strategies using the full–rank system with M=20 and D=5. The dimensionality reduction matrix $\boldsymbol{S}_{D_k}(0)$ is initialized as $[\boldsymbol{I}_D \quad \boldsymbol{0}_{D,M-D}]^T$. We observe that the proposed DRJIO–NLMS algorithm has a better performance on both the MSE level and convergence rate, which is very close to the distributed RLS algorithm. However, its complexity is an order of magnitude lower than the distributed RLS algorithm.

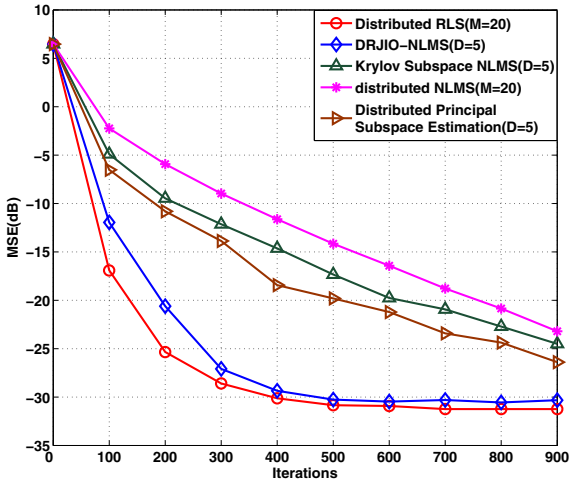

Fig. 3. Full–rank system with M=20

In a sparse system with M=20 scenario, the convergence rate for all algorithms increase, and the proposed DRJIO–NLMS algorithm still has an excellent performance as shown in Fig. 4. Specifically, the proposed DRJIO–NLMS algorithm performs very close to the distributed RLS algorithm and outperforms to other analyzed algorithms. When the full–rank system M increases to 60, Fig. 5 illustrates that, the proposed DRJIO–NLMS algorithm also shows a very fast convergence rate. For the distributed NLMS algorithm, Krylov subspace

NLMS and distributed principal subspace estimation, their convergence speed is much lower.

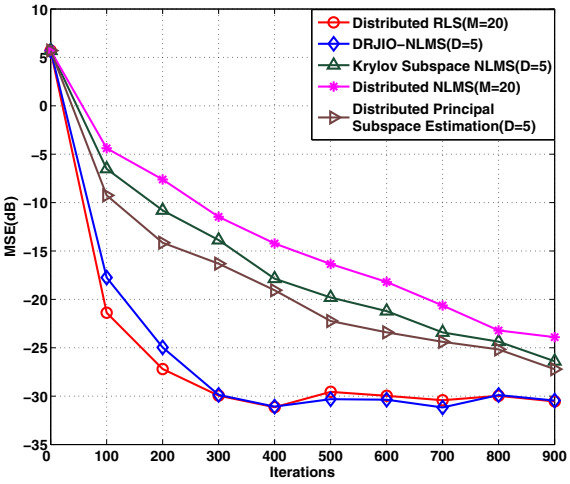

Fig. 4. Sparse system with M=20

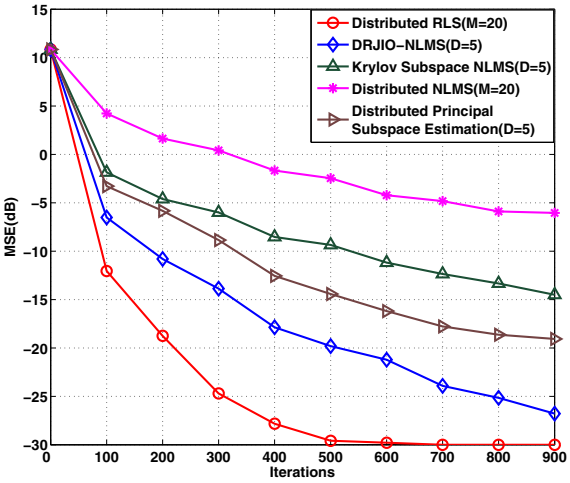

Fig. 5. Full–rank system with M=60

6. CONCLUSION

In this paper, we have proposed a novel distributed reduced–rank scheme along with an efficient algorithm for distributed estimation in wireless sensor networks. Simulation results have shown that the proposed DRJIO-NLMS algorithm has better performance and lower cost than existing algorithms in all the three scenarios considered. Furthermore, the proposed scheme requires the transmission of only D parameters instead of M, resulting in higher bandwidth efficiency than standard schemes.

REFERENCES

[1] C. G. Lopes and A. H. Sayed, "Incremental adaptive strategies over distributed networks," *IEEE Trans. Signal Process.*, vol. 48, no. 8, pp. 223–229, Aug 2007.

[2] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3122–3136, July 2008.

[3] L. Xie, D.-H. Choi, S. Kar, and H. V. Poor, "Fully distributed state estimation for wide-area monitoring systems," *IEEE Trans. Smart Grid*, vol. 3, no. 3, pp. 1154–1169, September 2012.

[4] Songcen Xu, R.C. de Lamare, and H.V. Poor, "Dynamic topology adaptation for distributed estimation in smart grids," *Proc. IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, pp. 420–423, Saint Martin, Dec 2013.

[5] E. Msechu, S. Roumeliotis, A. Ribeiro, and G. Giannakis, "Decentralized quantized kalman filtering with scalable communication cost," *IEEE Trans. Signal Process.*, vol. 56, no. 8, pp. 37273741, October 2008.

[6] J.-J. Xiao, A. Ribeiro, Z.-Q. Luo, and G. Giannakis, "Decentralized quantized kalman filtering with scalable communication cost," *IEEE Signal Process. Mag.*, vol. 23, no. 4, pp. 27–41, July 2006.

[7] S. Pereira and A. Pages-Zamora, "Distributed consensus in wireless sensor networks with quantized information exchange," *Proc. IEEE 9th Workshop Signal Process. Adv. Wireless Communication*, pp. 241–245, July 2008.

[8] Lin Li, A. Scaglione, and J.H. Manton, "Distributed principal subspace estimation in wireless sensor networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 725–738, 2011.

[9] M.O. Sayin and S.S. Kozat, "Single bit and reduced dimension diffusion strategies over distributed networks," *IEEE Signal Process. Lett.*, vol. 20, no. 10, pp. 976–979, 2013.

[10] S. Chouvardas, K. Slavakis, and S. Theodoridis, "Trading off complexity with communication costs in distributed adaptive learning via krylov subspaces for dimensionality reduction," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 2, pp. 257–273, 2013.

[11] R.C. De Lamare and R. Sampaio-Neto, "Reduced–rank adaptive filtering based on joint iterative optimization of adaptive filters," *IEEE Signal Process. Lett.*, vol. 14, no. 12, pp. 980–983, 2007.

[12] R.C. De Lamare and R. Sampaio-Neto, "Adaptive reduced-rank equalization algorithms based on alternating optimization design techniques for mimo systems," *IEEE Trans. Vehi. Techn.*, vol. 60, no. 6, pp. 2482–2494, 2011.

[13] R.C. De Lamare and R. Sampaio-Neto, "Reduced-rank space–time adaptive interference suppression with joint iterative least squares algorithms for spread-spectrum systems," *IEEE Trans. Vehi. Techn.*, vol. 59, no. 3, pp. 1217–1228, 2010.

[14] R.C. De Lamare and R. Sampaio-Neto, "Adaptive reduced-rank processing based on joint and iterative interpolation, decimation, and filtering," *IEEE Trans. Signal Process.*, vol. 57, no. 7, pp. 2503–2514, 2009.

[15] Yao Yu, A.P. Petropulu, and H.V. Poor, "Mimo radar using compressive sampling," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 1, pp. 146–163, Feb 2010.

[16] Haipeng Zheng, S.R. Kulkarni, and H.V. Poor, "Attribute-distributed learning: Models, limits, and algorithms," *IEEE Trans. Signal Process.*, vol. 59, no. 1, pp. 386–398, Jan 2011.

[17] A. Bertrand and M. Moonen, "Distributed adaptive node–specific signal estimation in fully connected sensor networks–part II: Simultaneous and asynchronous node updating," *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5292–5306, 2010.

[18] F. Cattivelli, C. G. Lopes, and A. H. Sayed, "A diffusion RLS scheme for distributed estimation over adaptive networks," *Proc. IEEE International Workshop on Signal Processing Advances for Wireless Communications*, pp. 1–5, June 2007.