DISTRIBUTED RLS ESTIMATION FOR COOPERATIVE SENSING IN SMALL CELL NETWORKS

Stefania Sardellitti, Sergio Barbarossa

Dept. of Information Engineering, Electronics and Telecommunications, Sapienza Univ. of Rome
Via Eudossiana 18, 00184, Rome, Italy
E-mail: Stefania.Sardellitti@uniroma1.it, sergio@infocom.uniroma1.it

ABSTRACT

Online adaptive algorithms have been largely applied for recursive estimation and tracking of sparse signals. In this paper we propose a distributed recursive least squares (RLS) algorithm incorporating an l_1-norm regularization with time-varying regularization coefficient that enables a recursive distributed solution with no losses with respect to the centralized solution. The method is especially useful in cooperative sensing when the parameters to be estimated are structurally sparse and time-varying. As well known, the l_1-norm is useful to recover sparsity, but it also introduces a non negligible bias. To tackle this issue, we further apply a garotte correction to our distributed mechanism that strongly reduces the bias. Numerical results are included to validate the estimation and tracking capabilities of the proposed algorithm.

Index Terms— Distributed adaptive algorithms, collaborative sensing, small cell networks

1. INTRODUCTION

The deployment of small cell networks has the potential to increase the spectral efficiency of modern cellular systems thanks to a strong spatial reuse of radio resources. However, the massive and possibly unplanned deployment requires a special care in handling interference. This requires some sort of cognition of the small cell base stations, also denoted as small cell eNode B (SC-eNB) or Home enhanced Node B (HeNB), in the LTE terminology. The first feature of any cognitive approach requires sensing the spectrum before allocating radio resources. On the other hand, local sensing can be impaired by shadowing effects that can deteriorate their capabilities. For this reason, cooperative sensing techniques are been studied quite extensively [1]. A characteristic of the observed signal that can be advantageously exploited to improve the estimation accuracy is the sparsity of the parameter vectors to be estimated. Sparsity may manifest in the spectral domain or in the space domain, for example. The least absolute shrinkage and selection operator (lasso) was proposed in [2] as an effective way to recover sparsity. Distributed sensing algorithms incorporating the lasso constraint were proposed in [3] and [4], where the authors used the alternating-direction method of multiplied (ADMM) [5] to enable a parallel solution amenable for decentralized networks. A further characteristic of the interference is the intrinsic time-varying nature of nodes' activity. This motivates the use of recursive methods as more suitable to track time-varying phenomena. For example, in [6] the authors proposed a distributed RLS algorithm for cooperative estimation, minimizing an exponentially-weighted least-squares cost using ADMM, but without sparsity constraints. An online adaptive estimation merging recursive least square with an l_1-norm constraint was then introduced in [7], valid for a single observation node. In this paper, we propose a *collaborative* RLS algorithm including an l_1-norm with a time-varying penalization coefficient that enables a *distributed and recursive* solution. Our formulation is also amenable to find a closed form solution in each step of the algorithm, which is useful for real-time implementation. Furthermore, to reduce the undesired bias of the lasso operator, we correct the solution by incorporating a *garotte* thresholding function [8]. Numerical results are included to validate the proposed approach and test its tracking capabilities.

This work was funded by the European Community 7th Framework Programme Project ICT-TROPIC, under grant nr. 318784.

2. MODEL STATEMENT

Let us consider a network composed of N nodes, where the observation $x_i(l)$ collected by node i, at time l, follows a linear model

$$x_i(l) = \boldsymbol{a}_i^T(l)\boldsymbol{\theta} + n_i(l), \; i = 1, \ldots, N \tag{1}$$

where $\boldsymbol{\theta}$ is the L-size column vector to be estimated, $\boldsymbol{a}_i(l)$ is a known time-varying regression vector of size L and $n_i(l)$ is the observation noise assumed Gaussian, with $n_i(l) \sim \mathcal{N}(0, \sigma_{n,i}^2)$. The vector $\boldsymbol{\theta}$ could represent the power transmitted by a set of L nodes at a given frequency f_0. In such a case, the m-th entry $a_{im}(l)$ of $\boldsymbol{a}_i(l)$ would be the channel gain from transmitter m to receiver i, at time l and frequency

f_0. The goal is to estimate $\boldsymbol{\theta}$ starting from the observations $x_i(l)$ in a distributed way, i.e. without requiring the presence of a sink node that collect all the measurements.

The vector $\boldsymbol{\theta}$ is generally sparse, because the transmitters are not active all the time, and time-varying. Our goal is to estimate and track slow variations of $\boldsymbol{\theta}$ over time, possibly exploiting its sparsity to improve the estimation accuracy. Sparsity can be enforced for example by assuming a Bayesian approach incorporating some prior knowledge of $\boldsymbol{\theta}$. For example, $\boldsymbol{\theta}$ could be modeled as the outcome of a random vector having a Laplacian prior pdf, i.e.

$$p_\Theta(\boldsymbol{\theta}) = \left(\frac{\mu}{2}\right)^L \exp(-\mu\|\boldsymbol{\theta}\|_1) \quad (2)$$

with $\mu > 0$, having denoted with $\|\boldsymbol{\theta}\|_1$ the l_1 norm of $\boldsymbol{\theta}$. Following a Bayesian approach, the estimated vector can be found as the maximum a posteriori estimator, i.e., as the solution of the following problem

$$\max_{\boldsymbol{\theta}} p_{X/\Theta}(\boldsymbol{x}/\boldsymbol{\theta}) p_\Theta(\boldsymbol{\theta}) \quad (3)$$

where $p_{X/\Theta}(\boldsymbol{x}/\boldsymbol{\theta})$ is the pdf of the observation vector collected by all the nodes $\boldsymbol{x} = [x_1, \ldots, x_N]$ conditioned to $\boldsymbol{\theta}$. Under the Gaussian assumption about the noise, taking the log of the function in (3), the estimate $\boldsymbol{\theta}$, at time n, must be the vector that minimizes the following function

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} \sum_{l=0}^{n} [x_i(l) - \boldsymbol{a}_i^T(l)\,\boldsymbol{\theta}]^2 \frac{1}{2\sigma_{n,i}^2} + \mu\|\boldsymbol{\theta}\|_1. \quad (4)$$

Alternatively, (4) can be seen as a least mean square approach with an l_1-norm penalty to enforce sparsity. Since we are interested in tracking slow time-variations of $\boldsymbol{\theta}$, we reformulate the problem as a recursive least-squares (RLS) approach [9], i.e. as the minimization of the following weighted sum

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} \sum_{l=0}^{n} \beta^{n-l} [x_i(l) - \boldsymbol{a}_i^T(l)\,\boldsymbol{\theta}]^2 \frac{1}{2\sigma_{n,i}^2} + \mu\|\boldsymbol{\theta}\|_1 \quad (5)$$

where $\beta \in (0, 1]$ is a forgetting factor.

In general, the solution of the problem in (5) requires a centralized approach because the problem cannot be decoupled. Nevertheless, a distributed solution can be found by introducing the local estimates $\boldsymbol{\theta}_i$, for each node, and then add the constraint that all local estimates must be equal to a common, unknown, instrumental variable $\boldsymbol{z}$ to force all the nodes to converge to the same value [5]. The resulting constrained problem can be formulated as the minimization of the so called augmented Lagrangian

$$\begin{aligned} \min_{\boldsymbol{\theta}_i} \quad & \sum_{i=1}^{N} \sum_{l=0}^{n} \beta^{n-l} [x_i(l) - \boldsymbol{a}_i^T(l)\,\boldsymbol{\theta}]^2 \frac{1}{2\sigma_{n,i}^2} + \\ & \frac{\rho_n}{2} \sum_{i=1}^{N} \|\boldsymbol{\theta}_i - \boldsymbol{z}\|^2 + \mu\|\boldsymbol{z}\|_1 \\ \text{s.t.} \quad & \boldsymbol{\theta}_i = \boldsymbol{z} \end{aligned} \quad (6)$$

where the additional term $\frac{\rho_n}{2} \sum_{i=1}^{N} \|\boldsymbol{\theta}_i - \boldsymbol{z}\|^2$, with $\rho_n > 0$, does not alter the solution of the original problem, but it offers the following benefits: i) it makes the objective function differentiable under milder conditions than the original Lagrangian and ii) it ensures strict convexity of the objective function. Notice that we allow the coefficient ρ_n of the additional term to be possibly time-varying, for reasons that will be clarified later on in the search for a recursive solution. The N problems in (6) are amenable for a distributed solution. In our setting, the objective function in (6) is strongly convex and then this problem admits a unique solution that can be found in a distributed way using the *alternating direction method of multipliers (ADMM)* [5], similarly to approach used in [10]. Using the ADMM approach, the algorithm proceeds through the following recursive updates:

$$\begin{aligned} \boldsymbol{\theta}_i[k+1, n] =& \arg\min_{\boldsymbol{\theta}_i} \Bigg\{ \sum_{i=1}^{N} \sum_{l=0}^{n} \beta^{n-l} [x_i(l) - \boldsymbol{a}_i^T(l)\boldsymbol{\theta}]^2 \frac{1}{2\sigma_{n,i}^2} \\ & + \boldsymbol{\lambda}_i^T[k, n](\boldsymbol{\theta}_i - \boldsymbol{z}[k, n]) + \tfrac{\rho_n}{2}\|\boldsymbol{\theta}_i - \boldsymbol{z}[k, n]\|_2^2 \Bigg\}, \\ \boldsymbol{z}[k+1, n] =& \arg\min_{\boldsymbol{z}} \Bigg\{ \mu\|\boldsymbol{z}\|_1 + \frac{\rho_n}{2} \sum_{i=1}^{N} \|\boldsymbol{\theta}_i[k+1, n] - \boldsymbol{z}\|^2 \\ & - \textstyle\sum_{i=1}^{N} \boldsymbol{\lambda}_i[k, n]^T \boldsymbol{z} \Bigg\}, \\ \boldsymbol{\lambda}_i[k+1, n] =& \boldsymbol{\lambda}_i[k, n] + \rho_n \left(\boldsymbol{\theta}_i[k+1, n] - \boldsymbol{z}[k+1, n]\right), \end{aligned} \quad (7)$$

where $\boldsymbol{\theta}_i[k, n]$ denotes the estimate of $\boldsymbol{\theta}_i$ at step k and time index n, while λ_i is the Lagrange multiplier vector associated to the equality constraint $\boldsymbol{\theta}_i = \boldsymbol{z}$. The first and second steps in (7) can be expressed in closed form. In particular, defining the vector threshold function $\boldsymbol{t}_\mu(\boldsymbol{x})$ as the component-wise thresholding function $t_\mu(x)$ applied to each element of vector $\boldsymbol{x}$, with

$$t_\mu(x) = \begin{cases} x - \mu, & x > \mu \\ 0, & -\mu \le x \le \mu \\ x + \mu, & x < -\mu \end{cases} \quad (8)$$

the algorithm can be written as

$$\begin{aligned} \boldsymbol{\theta}_i[k+1, n] =& \left(\sum_{l=0}^{n} \beta^{n-l} \boldsymbol{a}_i(l) \boldsymbol{a}_i^T(l)/\sigma_{n,i}^2 + \rho_n \boldsymbol{I} \right)^{-1} \cdot \\ & \cdot \left(\sum_{l=0}^{n} \beta^{n-l} \boldsymbol{a}_i(l) x_i(l)/\sigma_{n,i}^2 - \boldsymbol{\lambda}_i[k, n] + \rho_n \boldsymbol{z}[k, n] \right) \\ \boldsymbol{z}[k+1, n] =& \frac{1}{\rho_n N}\, t_\mu \left(N\overline{\boldsymbol{\lambda}[k, n]} + \rho_n N \overline{\boldsymbol{\theta}[k+1, n]} \right) \\ \boldsymbol{\lambda}_i[k+1, n] =& \boldsymbol{\lambda}_i[k, n] + \rho_n \left(\boldsymbol{\theta}_i[k+1, n] - \boldsymbol{z}[k+1, n]\right). \end{aligned}$$

where the notation $\overline{\boldsymbol{x}}$ means average over all the nodes, i.e. $\overline{\boldsymbol{x}} := \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{x}_i$. From this formulation we can notice that the first and third steps can be run in parallel, while the second step requires the computation of an average value which can be obtained by running a distributed consensus algorithm [11]. The only assumption necessary to guarantee the convergence of the consensus algorithm is that the graph representing the links among the N cooperative nodes is connected.

Our next goal now is to find a recursive solution. For simplicity of notation, we assume $\sigma_{n,i}^2 = 1$. Let us define the $L \times L$ matrix

$$\boldsymbol{\Phi}_i(n) := \sum_{l=0}^{n} \beta^{n-l} \boldsymbol{a}_i(l)\boldsymbol{a}_i^T(l) + \rho_n \boldsymbol{I} \quad (9)$$

and the L-dimensional vector

$$\boldsymbol{v}_i(n) = \sum_{l=0}^{n} \beta^{n-l} \boldsymbol{a}_i(l) x_i(l) \quad (10)$$

where we set $\rho_n = c\beta^n$ with $c > 0$. Under this assumption, the following recursive updating rules can be obtained:

$$\boldsymbol{\Phi}_i(n+1) = \beta\boldsymbol{\Phi}_i(n) + \boldsymbol{a}_i(n+1)\boldsymbol{a}_i^T(n+1) \quad (11)$$

$$\boldsymbol{v}_i(n+1) = \beta\boldsymbol{v}_i(n) + \boldsymbol{a}_i(n+1)x_i(n+1) \quad (12)$$

so that $\boldsymbol{\Phi}_i^{-1}(n+1)$ can be computed recursively with a complexity $\mathcal{O}(L^2)$ by using the matrix inversion lemma as

$$\boldsymbol{\Phi}_i^{-1}(n+1) = \frac{1}{\beta}\left[\boldsymbol{\Phi}_i^{-1}(n) - \frac{\boldsymbol{\Phi}_i^{-1}(n)\boldsymbol{a}_i(n+1)\boldsymbol{a}_i^T(n+1)\boldsymbol{\Phi}_i^{-1}(n)}{\beta + \boldsymbol{a}_i^T(n+1)\boldsymbol{\Phi}_i^{-1}(n)\boldsymbol{a}_i(n+1)}\right]. \quad (13)$$

Note that the per-sensor computational complexity has been significatively reduced since the matrix inversion for each iteration has been avoided. This recursive formulation has been made possible by the choice $\rho_n = c\beta^n$. However, this implies that asymptotically, as n goes to infinity, the problem in (6) looses the strict convexity property because the regularization term tends to disappear. To overcome this problem, we propose a block formulation that works as follows. If we divide the time intervals in blocks of length N_s, we can write the running time index n as $n = KN_s + p$, where $K = 0, 1, \ldots,$ denotes the block index, whereas $p = 0, \ldots, N_s - 1$ is the index within the block. As a consequence, the first term in the right side of (9) can be written as

$$\begin{aligned}
&\sum_{l=0}^{N_sK+p} \beta^{N_sK+p-l}\boldsymbol{a}_i(l)\boldsymbol{a}_i^T(l) \\
&= \sum_{l=0}^{N_sK-1} \beta^{N_sK+p-l}\boldsymbol{a}_i(l)\boldsymbol{a}_i^T(l) + \sum_{l=KN_s}^{KN_s+p} \beta^{KN_s+p-l}\boldsymbol{a}_i(l)\boldsymbol{a}_i^T(l) \\
&= \sum_{k'=0}^{K-1} \beta^{(K-k')N_s} \sum_{q=0}^{N_s-1} \beta^{p-q}\boldsymbol{a}_i(k'N_s+q)\boldsymbol{a}_i^T(k'N_s+q) + \\
&\sum_{m=0}^{p} \beta^{p-m}\boldsymbol{a}_i(m+KN_s)\boldsymbol{a}_i^T(m+KN_s).
\end{aligned}$$

Since $0 < \beta < 1$, we can choose N_s so that $\beta^{(K-k')N_s} \approx 0$, for $k' < K$. With this choice, the first term in this last expression can be neglected. As a consequence, the matrix

Table I: LADMM ALGORITHM

1: Set $n = 0, k = 0$, and initialize $\boldsymbol{\lambda}_i[0,0], \forall i$, and $\boldsymbol{z}[0,0]$ randomly;
2: if $n = 0$ or $\mathrm{mod}(n, N_s) = 0$
  then $t = 0, \rho = c\beta^t$
    $\boldsymbol{\Phi}_i^{-1}(n) = (\boldsymbol{a}_i(n)\boldsymbol{a}_i^T(n) + c\boldsymbol{I})^{-1}$
    $\boldsymbol{v}_i(n) = \boldsymbol{a}_i(n)x_i(n), t = t + 1$
  else $\rho = c\beta^t$
    compute $\boldsymbol{v}_i(n)$ and $\boldsymbol{\Phi}_i^{-1}(n)$ $\forall i$ as in (12) and (13)
    $t = t + 1$
  end
3: Repeat until convergence over index k
  $\boldsymbol{\theta}_i[k+1,n] = \boldsymbol{\Phi}_i^{-1}(n)\left(\boldsymbol{v}_i(n) - \boldsymbol{\lambda}_i[k,n] + \rho\boldsymbol{z}[k,n]\right)$
  Run consensus over $\boldsymbol{\theta}_i[k+1,n]$ and $\boldsymbol{\lambda}_i[k,n]$ to get $\overline{\boldsymbol{\theta}[k+1,n]}$ and $\overline{\boldsymbol{\lambda}[k,n]}$ until convergence;
  $\boldsymbol{z}[k+1,n] = \frac{1}{\rho N} t_\mu\left(N\overline{\boldsymbol{\lambda}[k,n]} + \rho N\overline{\boldsymbol{\theta}[k+1,n]}\right)$
  $\boldsymbol{\lambda}_i[k+1,n] = \boldsymbol{\lambda}_i[k,n] + \rho\left(\boldsymbol{\theta}_i[k+1,n] - \overline{\boldsymbol{\theta}[k+1,n]}\right)$
  Set $k = k + 1$, if the convergence criterion is satisfied, set $n = n + 1$ and go to step 2, otherwise go to step 3.

$\boldsymbol{\Phi}_i(p + KN_s)$, for $K > 1$, can be approximated as

$$\begin{aligned}
&\boldsymbol{\Phi}_i(p+KN_s) \\
&= \sum_{m=0}^{p} \beta^{p-m}\boldsymbol{a}_i(m+KN_s)\boldsymbol{a}_i^T(m+KN_s) + c\beta^p\boldsymbol{I} \\
&= \beta\boldsymbol{\Phi}_i(p-1+KN_s) + \boldsymbol{a}_i(p+KN_s)\boldsymbol{a}_i^T(p+KN_s).
\end{aligned}$$

Using this updating rule, we can exploit the matrix inversion lemma to compute $\boldsymbol{\Phi}_i^{-1}(p + KN_s)$ for $0 \le p \le N_s - 1$ and $K = 0, 1, \ldots$ iteratively, as in (13). In this way, the inversion operation is only needed at the beginning of each block. In summary, the RLS algorithm with lasso penalty, which we call Lasso ADMM (LADMM), is reported in Table I.

Since the lasso constraint is known for introducing a bias in the estimate, we introduce the non-negative garotte estimator, as in [8]. Inspired from the approach in [8], we replace the nonlinear function $\boldsymbol{t}_\mu(\boldsymbol{x})$ in the LADMM algorithm with the garotte thresholding function $\boldsymbol{t}_{\mathrm{g}}(\boldsymbol{x})$ defined as a vector whose entries are derived applying the threshold $t_{\mathrm{g}}(x) = (1 - \mu^2/x^2)x$ if $|x| > \mu$ and $t_{\mathrm{g}}(x) = 0$ for $-\mu \le x \le \mu$. We call this second algorithm Garotte ADMM (GADMM).

3. NUMERICAL RESULTS

To test the convergence of the proposed algorithm, we consider as an example a network of $N = 20$ cooperative nodes. The vector $\boldsymbol{\theta}$ represents the activity of $L = 4$ base stations on a given frequency. We assume that the regression vector follows a Gaussian distribution $\boldsymbol{a}(l) \sim (\boldsymbol{0}_L, \boldsymbol{I}_L)$, with noise variance $\sigma_{n,i}^2 = 10^{-1}, \forall i$. The entries of vector $\boldsymbol{\theta}$ switch from zero (no activity) to a non-zero value at some unknown time instant, to mimic the time-varying nature of radio nodes' activity. In Fig. 1 we report two entries of the vector $\boldsymbol{\theta}$ versus time, along with their estimates obtained by using the LADMM algorithm described in Table I, for two values of the forgetting factor: $\beta = 0.6$ (upper subplots) and $\beta = 0.9$ (lower subplots). We set the parameters as $\mu = 10, c = 100$, $N_s = 10$. We can notice that, at the beginning there is a strong improvement of estimation accuracy resulting from the cooperative approach. We can also observe how the RLS estimates are able to track the abrupt variations of the estimated parameters. The periodic fluctuations appearing in the estimated values are mainly due to the block structure of the proposed RLS.

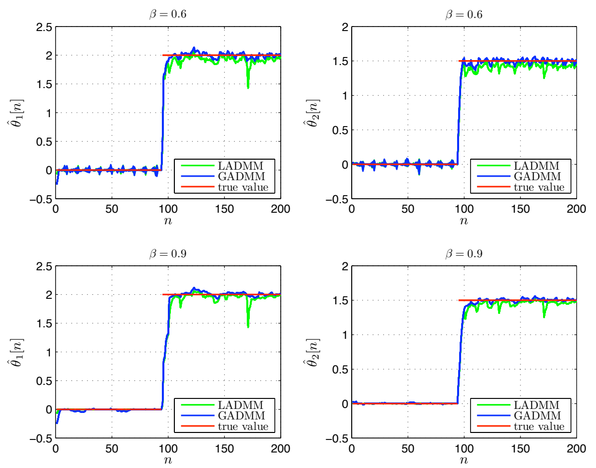

Fig. 1. Two entries of the parameter $\boldsymbol{\theta}$ versus the number of current observations n for two values of $\beta, c = 100, N_s = 10$.

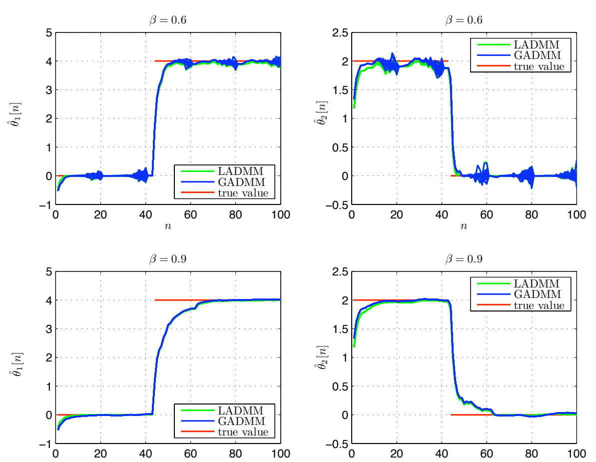

Fig. 2. Two entries of the parameter $\boldsymbol{\theta}$ versus the number of current observations n for two values of $\beta, c = 700, N_s = 20$.

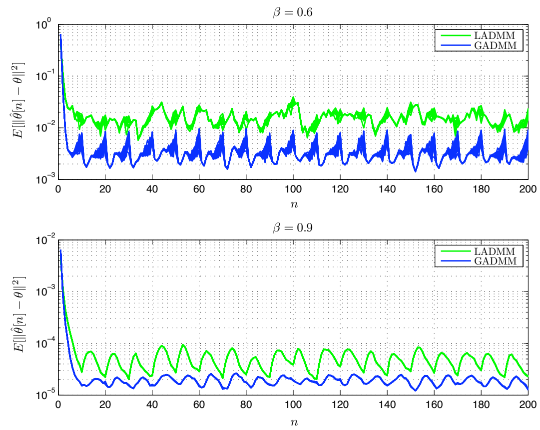

Fig. 3. Average mean square estimation error versus the number of current observations n for a time-invariant $\boldsymbol{\theta}$.

In Fig. 1, we compare the LADMM with the garotte ADMM algorithm (GADMM). As expected, the garotte correction yields a smaller bias. To better evaluate the impact of the RLS filter memory on the system performance, we have plotted in Fig. 2 the estimates of two entries of the vector $\boldsymbol{\theta}$ for $c = 700$ and $N_s = 20$. We can notice that, as β increases, the accuracy of the estimate improves but the algorithm becomes less robust to sudden time variations of the base stations' activity. Finally, in Fig. 3 we report the average estimation error $E[\|\hat{\boldsymbol{\theta}}[n] - \boldsymbol{\theta}\|^2]$ versus the time index n for a time-invariant $\boldsymbol{\theta}$, obtained by averaging the results over 100 independent realizations, setting $\beta = 0.6$ (upper subplot), $\beta = 0.9$ (lower subplot) with $c = 100$, $N_s = 10$. We can note that again the GADMM performs better than the LADMM approach and, as expected, the impact of the periodical updating rule on ρ_n is more evident for lower β values.

4. CONCLUSIONS

In this paper we have proposed a distributed RLS algorithm with l_1-norm constraint useful for cooperative sensing of slowly time-varying phenomena. The use of the alternating direction method of multipliers with a time-varying quadratic constraint has been instrumental to find out a recursive distributed solution. A garotte correction has then been introduced to improve the estimation accuracy on the non-zero entries of the estimated vector. The method incorporates an intermediate consensus step and, in its present form, it assumes a double time-scale approach: For each iteration of the distributed algorithm, we need to run a consensus algorithm. However, adopting methods already appeared in the literature, the problem can also be reformulated in a single time-scale framework [6], where at each iteration only a single consensus update is run, without waiting for the full convergence.

5. REFERENCES

[1] E. Axell, G. Leus, E. G. Larsson, H. V. Poor, "Spectrum sensing for cognitive radio : State-of-the-art and recent advances," *IEEE Signal Process. Mag.*, pp. 101–116, May 2012.

[2] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *J. Royal. Statist. Soc.*, vol. 58, no. 1, pp. 267–288, 1996.

[3] J. A. Bazerque and G. B. Giannakis, "Distributed spectrum sensing for cognitive radio networks by exploiting sparsity," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1847–1862, March 2010.

[4] G. Mateos, J. A. Bazerque and G. B. Giannakis, "Distributed sparse linear regression," *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5262–5276, October 2010.

[5] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Belmont, MA: Athena Scientific, 1989.

[6] G. Mateos and G. B. Giannakis, "Distributed recursive least-squares: Stability and performance analysis," *IEEE Trans. Signal Process.*, vol. 60, no. 7, pp. 3740–3754, July 2012.

[7] D. Angelosante, J. A. Bazerque and G. B. Giannakis, "Online adaptive estimation of sparse signal: where RLS meets the l_1-norm," *IEEE Trans. Signal Process.*, vol. 58, no. 7, pp. 3436–3447, July 2010.

[8] M. Yuan, Yi Lin, "On the non-negative garrotte estimator," *J. Royal Statist. Soc.*, vol. 69, pp. 143–161, 2007.

[9] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory,* Prentice Hall, 1993.

[10] 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 Trans. Signal Process.*, vol. 56, pp. 350–364, January 2008.

[11] R. Olfati-Saber, J. A. Fax, R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. of the IEEE*, vol. 95, pp. 215-233, January 2007.