Distributed Blind Adaptive Algorithms Based on Constant Modulus for Wireless Sensor Networks

Reza Abdolee
Electrical and Computer Engineering
McGill University
3480 University Street Montreal, PQ, Canada H3A 2A7
Email: reza.abdolee@mail.mcgill.ca

Benoit Champagne
Electrical and Computer Engineering
McGill University
3480 University Street Montreal, PQ, Canada H3A 2A7
Email: benoit.champagne@mcgill.ca

***Abstract*—In this paper, we propose and study the distributed blind adaptive algorithms for wireless sensor network applications. Specifically, we derive distributed forms of the blind least mean square (LMS) and recursive least square (RLS) algorithms based on the constant modulus (CM) criterion. We assume that the inter-sensor communication is single-hop with Hamiltonian cycle to save the power and communication resources. The distributed blind adaptive algorithm runs in the network with the collaboration of nodes in time and space to estimate the parameters of an unknown system or a physical phenomenon. Simulation results demonstrate the effectiveness of the proposed algorithms, and show their superior performance over the corresponding non-cooperative adaptive algorithms.**

Keywords: Distributed adaptive algorithms, Wireless sensor networks, Incremental network topology, Constant modulus criterion

I. Introduction

Decentralized signal processing offers significant advantages over its centralized counterpart [1]. In a centralized approach, in order to reach a consensus on the underlying signal parameters of interest, each sensor must communicate with the fusion center. This causes network congestion and result in a waste of communication resources, such as power and bandwidth. More importantly, any malfunction in the fusion center may cause a network breakdown. By developing robust decentralized signal processing algorithms, we can distribute the computation between the local nodes, reduce the amount of communications overhead in the exchange of information, and remove the dependence of the network on the fusion center. Within the above framework of cooperative, *in-network* distributed processing, there has been much interest lately in the study of new distributed *adaptive* algorithms for the solution of parameter estimation problems in which the underlying signal statistics are unknown or time-varying. Clearly, adaptivity can help the network to track variations in the desired signal parameter over time as new measurements become available. More importantly, as a result of distributed adaptive processing, a sensor network becomes robust against changes in the network environment, network topology and node failure.

Recently, there have been some advances in distributed adaptive signal processing for sensor network applications. In [2], [3] and [4], distributed adaptive LMS and RLS algorithms are proposed for parameter estimation in networks with incremental or diffusion topology. These techniques are developed based on ideal (i.e. distorsionless) inter-sensor channel for the exchange of information in the distributed cooperation. In [5] and [6] the authors have proposed distributed LMS and RLS algorithms, respectively, for non-ideal inter-sensor wireless channels by incorporating additive noise.

These algorithms which were initially developed for parameter estimation, can be applied more generally to obtain distributed solutions to various problems of adaptive filtering. When used in this way, these algorithms are classified as non-blind, or training-based, since they require a reference signal to drive the adaptation process. In practice, the use of a reference signal might entail significant costs (especially reduced bandwidth efficiency) and in many cases, it is physically infeasible. Therefore, developing *blind* distributed adaptive algorithms is somehow indispensable and will be the next logical step in the research trend. Generally, the use of blind adaptation is possible in scenarios where there exists side information about the transmitted signal, also called signal restoration properties.

In this work, we develop new adaptive algorithms for distributed blind equalization that use the constant energy envelope property of the received signals. Specifically, we focus on a basic signal model in which each sensor has access to a filtered copy of a constant envelope signal contaminated by additive noise. We assume that the unknown filtering applied to the desired signal is identical for each sensor, up to an independent phase shift. We derive distributed forms of the blind LMS and RLS algorithms which allow the sensors to cooperate over wireless to identify the common adaptive equalizer weights needed for envelope restoration. To save power and bandwidth, the new distributed algorithms use an incremental approach for inter-sensor communications, i.e. single-hop Hamiltonian cycle. The effectiveness of the proposed algorithms is demonstrated by simulations, which show a significant performance gain in signal restoration compared to the non-cooperative algorithms.

II. System model and problem formulation

The system model under consideration is shown in block diagram form in Fig. 1. We consider a sub-network of N neighboring sensors (nodes) geographically distributed over

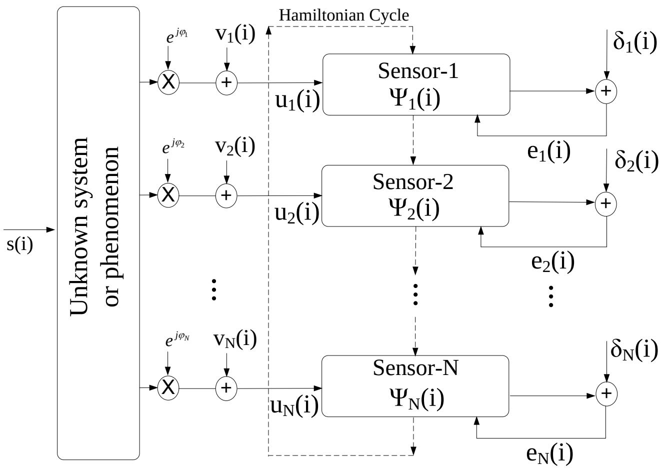

Fig. 1: System model for distributed blind adaptive equalization

an area where a physical phenomenon of interest is being monitored. Each sensor measures the distorted signal coming from the output of an unknown system, modeled as a linear, (possibly) time-varying filter with constant envelope source signal $s(i)$ as input. We assume that the unknown filtering applied to the desired signal is identical for each sensor, but that the measurements are made in the presence of independent phase shift and additive measurement noise at each sensor. Specifically, the measured signal by sensor k at discrete-time i, denoted $u_k(i)$, bears the following relation with the system's parameters and input $s(i)$

$$u_k(i) = \sum_{l=0}^{L} e^{j\phi_k} \beta(i,l) s(i-l) + v_k(i) \tag{1}$$

where $\beta(i,l)$, $l = 0, \ldots, L$ denote the impulse response coefficients of the unknown system for lag l at time i, L is the assumed system order, $v_k(i)$ is an additive noise component at the kth sensor while ϕ_k represents the phase shift of the measured signal by the kth sensor. These unknown phase shifts, which are assumed to remain constant over the integration time of the adaptation process, are modeled as independent and identically (i.i.d.) random variables uniformly distributed between $[0, 2\pi]$. The additive noise terms $\{v_k(i)\}$ are modeled as i.i.d. white noise sequence, with each sample having a complex circular symmetric Gaussian distribution, i.e. $v_k(i) \sim \mathcal{C}(0, \sigma_k^2)$ where σ_k^2 denotes the measurement noise power at the kth sensor. The above system model formulation is suitable for adaptive system modeling, system identification and channel equalization.

Because of the distortion induced by the unknown system and the additive noise, the measured signal $u_k(i)$ at the kth sensor will generally not exhibit the constant modulus property of the input. The problem of interest here is to devise a blind adaptive equalizer, in the form of a time-varying finite impulse response (FIR) filter with the global coefficient vector $\mathbf{w}(i) = [w(i,0), w(i,1), \ldots, w(i, M-1)]^T \in \mathbb{C}^{M\times 1}$ where M denotes the filter length, that can be used at each sensor to restore the constant modulus property in its measurement $u_k(i)$. Assuming slow time-variations in the unknown system and adaptive process, we can represent them in terms of their corresponding time-varying system functions $B_i(z) = \sum_{l=0}^{L} \beta(i,l) z^{-l}$ and $W_i(z) = \sum_{l=0}^{M-1} w(i,l) z^{-l}$, respectively, where z denotes the unit delay operator. To perform the desired equalization task adequately, the adaptive solution should ideally satisfy the condition $W_i(z) = 1/B_i(z)$.

In practice, because of measurement noise and lag in the adaptive process, this condition can only be approximately satisfied. In a traditional, i.e. non-cooperative approach, each sensor would run its own copy of a standard blind adaptive algorithm for constant modulus restoration, such as the LMS-CMA [7] or the RLS-CMA [8]. However, this approach does not exploit available means of communication between the sensors and is therefore sub-optimal. In this paper, we seek a distributed solution to the above blind adaptive equalization problem in which each sensor maintains and locally update its own copy of the adaptive equalizer weights (that can be used to filter its measurement signal), but cooperates through exchange of information over wireless links in seeking a globally optimal solution (i.e. across the set of N sensors).

Let $\boldsymbol{\psi}_k(i) \in \mathbb{C}^{M\times 1}$ denote local adaptive equalizer weight vector of sensor k at time i. To save power and bandwidth, we assume an incremental approach for inter-sensor communications, i.e. single-hop pre-defined Hamiltonian cycle, as shown by the dashed line in Fig. 1. At each step in this cycle, repeated once per iteration over the adaptation time index i, the kth sensor recursively updates its weight vector, i.e. $\boldsymbol{\psi}_k(i-1) \rightarrow \boldsymbol{\psi}_k(i)$, by making use of the updated weight vector $\boldsymbol{\psi}_{k-1}(i)$ from its predecessor in the cycle, and then communicates the result of this update to its successor. The choice and definition of the sequence of sensors visited in a cycle is based on link and availability considerations that fall outside the scope of this work. Here, the wireless channels used in inter-sensor communication are perfect (noise-free and distortionless), but generalization in the style of [5] and [6] can be envisaged. In the following sections, we develop the proposed distributed blind adaptive LMS-CMA and RLS-CMA.

III. DISTRIBUTED LMS-CMA

The new distributed algorithms for blind adaptation will be derived by breaking down the centralized CM-based optimization problem into a set of local optimization problems, in which the only coupling is through the exchange of a node's updated weight vector to its successor in the Hamiltonian cycle. This approach will be first applied to derive a distributed LMS-CMA in this section, and then extended to derive RLS-CMA in the next section.

We begin by considering a centralized LMS formulation for the CM-based adaptation in a sensor network. With reference to Fig. 1, the output of the equalizer at node k at time instant i is given by:

$$y_k(i) = \mathbf{u}_k(i) \boldsymbol{\psi}_k(i) \tag{2}$$

where, $\mathbf{u}_k(i) = [u_k(i), u_k(i-1), \ldots, u_k(i-M+1)]$ is the local data vector at node k. By collecting the local data vectors in the central processor, we form a global data matrix $U(i) \triangleq [\mathbf{u}_1(i)^T, \mathbf{u}_2(i)^T, \ldots, \mathbf{u}_N(i)^T]^T$ for further processing. In the expanded form, the latter can be written as:

$$U(i) \triangleq \begin{bmatrix} u_1(i) & u_1(i-1) & \ldots & u_1(i-M+1) \\ u_2(i) & u_2(i-1) & \ldots & u_2(i-M+1) \\ \vdots & \vdots & \vdots & \vdots \\ u_N(i) & u_N(i-1) & \ldots & u_N(i-M+1) \end{bmatrix} \quad (3)$$

For the CM criterion, the global cost function at the central processor is formulated as:

$$J(\mathbf{w}) = E[\|\boldsymbol{\delta} - |\mathbf{y}(i)|^p\|^q], \quad (4)$$

where p and q are positive real numbers, $\mathbf{y}(i) \triangleq [y_1(i), y_2(i), \ldots, y_N(i)]^T$ and $\boldsymbol{\delta} \triangleq [\delta_1, \delta_2 \ldots \delta_N]^T$. The kth entry of $\boldsymbol{\delta}$, δ_k, is a positive real number that represent the desired constant modulus value to be restored at the kth node. For the sake of generality, we keep the subscript k in our derivation, although we shall later assume $\delta_k = 1$ for $k = 1, \ldots, N$ when presenting simulation results in Section V. The use of the parameters p and q allows additional flexibility in the problem solution (see e.g. [8]). The traditional mean square error (MSE)-based CM cost function corresponds to the choice $p = 1$ and $q = 2$. Here, the use of $q = 2$ (i.e. MSE-based CM) is favored, since as we explain below, it enables a partial decomposition of the cost function as a sum of simple terms, which in turn is amenable to distributed adaptive processing.

The global equalizer's coefficients, denoted by $\mathbf{w}_o \in \mathbb{C}^{M\times 1}$, can be found by minimizing the above cost function; i.e.,

$$\mathbf{w}_o = \min_{\mathbf{w}\in\mathbb{C}^M} J(\mathbf{w}) = \min_{\mathbf{w}\in\mathbb{C}^M} E[\|\boldsymbol{\delta} - |U(i)\mathbf{w}|^p\|^q] \quad (5)$$

In (4), the use of absolute value in $|U(i)\mathbf{w}|^p$ must be interpreted element-wise, i.e. $|U(i)\mathbf{w}|^p = [|\mathbf{u}_1(i)\mathbf{w}|^p, |\mathbf{u}_2(i)\mathbf{w}|^p, \ldots, |\mathbf{u}_N(i)\mathbf{w}|^p]^T$. By expanding the squared Euclidean norm of (4) when $q = 2$, we can write:

$$\begin{aligned} J(\mathbf{w}) &= E[\|\boldsymbol{\delta} - |U(i)\mathbf{w}|^p\|^2] \\ &= \sum_{k=1}^{N} J_k(\mathbf{w}) \end{aligned}$$

where $J_k(\mathbf{w}) = E\|\delta - |\mathbf{u}_k(i)\mathbf{w}|^p\|^2$ can be interpreted as the the local objective function at node k. In a centralized scheme, the steepest descent iterative solution to the above optimization can be expressed based on the partial derivative of the local objective functions as:

$$\begin{aligned} \mathbf{w}(i) &= \mathbf{w}(i-1) - \mu\nabla J(\mathbf{w}(i-1)) \\ &= \mathbf{w}(i-1) - \mu\sum_{k=1}^{N} \nabla J_k(\mathbf{w}(i-1)) \end{aligned} \quad (6)$$

where $0 < \mu \le 1$ is the step size of the steepest descent iteration. After calculating the partial derivative, we obtain:

$$\mathbf{w}(i) = \mathbf{w}(i-1) + \mu\sum_{k=1}^{N} E[\mathbf{u}_k^H y_k(i)\,|y_k(i)|^{p-2}(\delta_k - |y_k(i)|^p)] \quad (7)$$

Proceeding as in [2], the steepest descent update formula in (7) can be implemented in a distributed manner by cooperation of the local nodes as given in the algorithm below :

$$\begin{aligned} &\boldsymbol{\psi}_0(i) \leftarrow \mathbf{w}(i-1) \\ &\boldsymbol{\psi}_k(i) = \boldsymbol{\psi}_{k-1}(i) + \mu E[\mathbf{u}_k^H y_k(i)\,|y_k(i)|^{p-2}(\delta_k - |y_k(i)|^p)], \\ &\quad k = 1, 2 \ldots N \\ &\mathbf{w}(i) \leftarrow \boldsymbol{\psi}_N(i) \end{aligned} \quad (8)$$

In the distributed steepest descent algorithm (8), we need to perform N iterations over the spatial dimension k, i.e. node $k = 1$ to node $k = N$ in a predefined cycle. During the ith such cycle, node k uses the updated estimate received from its predecessor in the cycle, i.e. $\boldsymbol{\psi}_{k-1}(i)$, to update its current estimate $\boldsymbol{\psi}_k(i)$, which is then transmitted to its successor. That is, update 7 is realized via a sequence of N single hop wireless information exchange between adjacent nodes in the cycle. The final distributed LMS-CMA can now be obtained by approximating the gradient in (8) with its stochastic version by using instantaneous data at the time instant i. For compactness in presentation, we introduce the error function at node k, defined as $e_k(i) = y_k(i)\,|y_k(i)|^{p-2}(\delta_k - |y_k(i)|^p)$. The results are summarized in algorithm 1, which is somewhat similar in structure to the non-blind distributed LMS algorithm developed in [2]. The effectiveness of the algorithm above will be demonstrated through numerical simulations in Section V. In the next Section, we derive an incremental distributed RLS-CMA by using a similar approach.

Algorithm 1 Distributed LMS-CMA

$\boldsymbol{\psi}_0(i) \leftarrow \mathbf{w}(i-1)$
for $k = 1 : N$ **do**
  $y_k(i) = \mathbf{u}_k(i)\boldsymbol{\psi}_k(i)$
  $e_k(i) = y_k(i)\,|y_k(i)|^{p-2}(\delta_k - |y_k(i)|^p)$
  $\boldsymbol{\psi}_k(i) = \boldsymbol{\psi}_{k-1}(i) + \mu\mathbf{u}_k^H e_k(i)$
end for
$\mathbf{w}(i) \leftarrow \boldsymbol{\psi}_N(i)$

IV. DISTRIBUTED RLS-CMA

The general form of the weighted least squares (WLS) cost function for CM signal restoration at a central processor can be expressed as:

$$J(\mathbf{w}, i) = \sum_{l=0}^{i} \lambda^{i-l}\|\boldsymbol{\delta} - |U(l)\mathbf{w}|^p\|^2 \quad (9)$$

where $0 < \lambda \le 1$ is the forgetting factor, $U(l)$ is the data matrix given in (3), and $\mathbf{w}$ is the global equalizer's weights. This cost function provides a weighted sum of the modulus errors at the different nodes, from time $l = 0$ to current time $l = i$, with past errors weighted by λ^{i-l}. In this work, based on the value of the parameter p, we derive two different versions of the distributed RLS-CMA. In the first case, we set $p = 1$ and then develop a first version of the distributed RLS-CMA without making any assumption about the signal environment; whereas for the second version, p can take any arbitrary real positive value, but in this case, we need to assume that the signal environment is slowly varying or stationary.

A. *Distributed RLS-CMA for* $p = 1$

In this case, the global objective function in (9) can be written based on the local data as:

$$J(\mathbf{w}, i) = \sum_{k=1}^{N}\sum_{l=0}^{i} \lambda^{i-l} |\delta_k - |\,\mathbf{u}_k(l)\mathbf{w}|\,|^2 \tag{10}$$

The stationary point of this cost function can be found by computing its partial derivative and equating it to zero, which yields:

$$\sum_{k,l} \lambda^{i-l}\mathbf{u}_k^H(l)\mathbf{u}_k(l)\mathbf{w} = \sum_{k,l} \lambda^{i-l}\alpha_k(l)\mathbf{u}_k^H(l)\delta_k \tag{11}$$

where the summation $\sum_{k,l}$ is over the range $1 \le k \le N$ and $0 \le l \le l$, and we have introduced

$$\alpha_k(i) = \frac{y_k(i)}{|y_k(i))|}. \tag{12}$$

Equivalently, (11) can be expressed in matrix form as

$$R(i)\mathbf{w} = \mathbf{r}(i) \tag{13}$$

where we define

$$R(i) = \sum_{l=0}^{i} \lambda^{i-l} U^H(l)U(l), \qquad \mathbf{r}(i) = \sum_{l=0}^{i} \lambda^{i-l} U_\alpha^H(l)\boldsymbol{\delta}(l), \tag{14}$$

and $U_\alpha^H(i) = [\alpha_1(i)\mathbf{u}_1^H(i), \alpha_2(i)\mathbf{u}_2^H(i), \ldots, \alpha_N(i)\mathbf{u}_N^H(i)]$. Therefore, the WLS solution at current time i can be computed as $\mathbf{w}(i) = R^{-1}(i)\mathbf{r}(i)$. Alternatively, the optimal weight vector in (13) can be recursively updated by means of the following relation:

$$\mathbf{w}(i) = \mathbf{w}(i-1) + R^{-1}(i)[U_\alpha^H(i)\boldsymbol{\delta} - U^H(i)U(i)\mathbf{w}(i-1)] \tag{15}$$

As a first step towards the derivation of a distributed RLS solution for the CM problem, we focus on the efficient updating of the required inverse correlation matrix $R^{-1}(i)$ in (15). Indeed, to avoid costly matrix inversion, we can compute $R^{-1}(i)$ recursively, in a distributed and incremental manner as explained below. Using the definition of the sample correlation matrix in [9], we know that $R(i) = \lambda R(i-1) + U^H(i)U(i)$. Equivalently, by expanding the product $U^H(i)U(i)$ in terms of the local sensor observations, we obtain:

$$R(i) = \lambda R(i-1) + \sum_{k=1}^{N} \mathbf{u}_k^H(i)\mathbf{u}_k(i) \tag{16}$$

Equation (16) can be iteratively updated in time and space using only the local data at sensor k, by proceeding as follows:

$$\begin{aligned} &R_0(i) \leftarrow \lambda R(i-1) \\ &R_k(i) = R_{k-1}(i) + \mathbf{u}_k^H \mathbf{u}_k(i), \\ &\quad k = 1, 2 \ldots N \\ &R(i) \leftarrow R_N(i) \end{aligned} \tag{17}$$

We note that there is no physical sensor corresponding to the index $k = 0$; the latter is introduced only for convenience in joining both ends of the incremental cycle of spatial updates over index k as time is incremented from $i-1$ to i, i.e., $R_0(i) \equiv \lambda R_N(i-1)$.

Note that each update in (17) only invoke a rank one additive term. Therefore, the inverse of the sampled correlation matrix given in (17) can be computed locally according to Woodbury's identity. As a result, the inverse of the global correlation matrix can be calculated in distributed fashion by using the set of local data as:

$$\begin{aligned} &R_0(i)^{-1} \leftarrow \lambda^{-1} R^{-1}(i-1) \\ &R_k^{-1}(i) = R_{k-1}^{-1}(i) - \frac{R_{k-1}^{-1}(i)\mathbf{u}_k^H(i)\mathbf{u}_k(i)R_{k-1}^{-1}(i)}{1+\mathbf{u}_k(i)R_{k-1}(i)\mathbf{u}_k^H(i)}, \\ &\quad k = 1, 2 \ldots N \\ &R^{-1}(i) \leftarrow R_N^{-1}(i) \end{aligned} \tag{18}$$

We note that to update its local estimate of the inverse correlation matrix with this approach, sensor k only makes use of its local observation vector $\mathbf{u}_k(i)$ along with the inverse correlation matrix estimate of its predecessor in the incremental cycle, i.e. $R_{k-1}^{-1}(i)$.

The recursive formula given in (15) can also be updated in a distributed manner based on the local data at the kth sensor, for $k = 1, \ldots, N$. Indeed, by expanding the term $\{U_\alpha^H(i)\boldsymbol{\delta} - U^H(i)U(i)\mathbf{w}(i-1)\}$, we can obtain the following recursion formula:

$$\mathbf{w}(i) = \mathbf{w}(i-1) + R^{-1}(i)\sum_{k=1}^{N} \mathbf{u}_k^H(i)e_k(i) \tag{19}$$

where we define the modulus error at node k as

$$e_k(i) = \alpha_k(i)\delta_k - \mathbf{u}_k(i)\mathbf{w}(i-1). \tag{20}$$

Equation (19) can be implemented in a distributed manner as:

$$\begin{aligned} &\boldsymbol{\psi}_0(i) \leftarrow \mathbf{w}(i-1) \\ &e_k(i) = \alpha_k(i)\delta_k - \mathbf{u}_k(i)\mathbf{w}(i-1), \\ &\boldsymbol{\psi}_k(i) = \boldsymbol{\psi}_{k-1}(i) + R^{-1}(i)\mathbf{u}_k^H(i)e_k(i), \\ &\quad k = 1, 2 \ldots N \\ &\mathbf{w}(i) \leftarrow \boldsymbol{\psi}_N(i) \end{aligned} \tag{21}$$

Finally, we can arrive at a fully distributed, incremental algorithm by substituting $\mathbf{w}(i-1)$ and $R^{-1}(i)$ with $\boldsymbol{\psi}_{k-1}(i)$ and $R_k^{-1}(i)$, respectively, with the latter quantity being updated in a distributed manner as in (18). Similar to [10], the substitution of $\mathbf{w}(i-1)$ by $\boldsymbol{\psi}_{k-1}(i)$ leads to better adaptive performance, whereas substituting $R^{-1}(i)$ with the local update $R_k^{-1}(i)$ causes some performance degradation. By applying these modifications, we obtain the first version of the distributed RLS-CMA given in Algorithm 2. In this algorithm, during the ith cycle time, sensor $k-1$ forwards its updated local weight vector estimate $\boldsymbol{\psi}_{k-1}(i)$ and inverse correlation matrix estimate $R_{k-1}^{-1}(i)$ to sensor k where the corresponding estimates are updated using only the local observation $\mathbf{u}_k(i)$.

Algorithm 2 Distributed adaptive RLS-CMA when $p=1$

$\boldsymbol{\psi}_0(i) \leftarrow \mathbf{w}(i-1); R_0^{-1}(i) \leftarrow \lambda^{-1}R^{-1}(i-1)$
for $k=1:N$ **do**
  $R_k^{-1}(i) = R_{k-1}^{-1}(i) - \frac{R_{k-1}^{-1}(i))\mathbf{u}_k^H(i)\mathbf{u}_k(i)R_{k-1}^{-1}(i)}{1+\mathbf{u}_k(i)R_{k-1}(i)\mathbf{u}_k^H(i)}$
  $e_k(i) = \alpha_k(i)\delta_k - \mathbf{u}_k(i)\boldsymbol{\psi}_{k-1}(i)$
  $\boldsymbol{\psi}_k(i) = \boldsymbol{\psi}_{k-1}(i) + R_k^{-1}(i)\mathbf{u}_k^H(i)e_k(i)$
end for
$\mathbf{w}(i) \leftarrow \boldsymbol{\psi}_N(i); R^{-1}(i) \leftarrow R_N^{-1}(i)$

B. Distributed RLS-CMA for general value of p

In this case, the global objective function (9) takes the following form:

$$J(\mathbf{w},i) = \sum_{k=1}^{N} J_k(\mathbf{w},i) \tag{22}$$

where

$$J_k(\mathbf{w},i) = \sum_{l=0}^{i} \lambda^{i-l}|\delta_k - |\mathbf{u}_k(l)\mathbf{w}|^p|^2, \tag{23}$$

is the local cost function at node k. Here, each local cost function can be transformed into the conventional RLS cost function by applying the suggested technique in [8]. According to this letter, if we assume the signal environment is stationary or slowly varying, then the difference between $\mathbf{u}_k(i)\mathbf{w}(i-1)$ and $\mathbf{u}_k(i)\mathbf{w}(i)$ is negligible. Hence, the local cost function in (23) can be rearranged as:

$$\begin{aligned} J_k(\mathbf{w},i) \;\; &= \;\; \sum_{l=0}^{i} \lambda^{i-l}|\delta_k - |\mathbf{u}_k(l)\mathbf{w}(l-1)|^{p-2} \\ &\times \;\; \mathbf{w}^H(l-1)\mathbf{u}_k^H(l)\mathbf{u}_k(l)\mathbf{w}|^2 \end{aligned} \tag{24}$$

This can be expressed more compactly as:

$$J_k(\mathbf{w},i) = \sum_{l=0}^{i} \lambda^{i-l}|\delta_k - \mathbf{z}_k(l)\mathbf{w}|^2 \tag{25}$$

where we define

$$\mathbf{z}_k(l) = |\mathbf{u}_k(l)\mathbf{w}(l-1)|^{p-2}\mathbf{w}^H(l-1)\mathbf{u}_k^H(l)\mathbf{u}_k(l) \tag{26}$$

As a result of this approximation, the global cost function takes the following form:

$$J(\mathbf{w},i) = \sum_{l=0}^{i} \lambda^{i-l}\|\boldsymbol{\delta} - |Z(l)\mathbf{w}|^p\|^2 \tag{27}$$

where $Z(i) = [\mathbf{z}_1^T(i), \mathbf{z}_2^T(i), \ldots, \mathbf{z}_N^T(i)]^T$ is the modified data matrix. Computing the partial derivative of (27), and equating it to zero yields:

$$\sum_{l=0}^{i} \lambda^{i-l}Z^H(l)Z(l)\mathbf{w} = \sum_{l=0}^{i} \lambda^{i-l}Z^H(l)\boldsymbol{\delta} \tag{28}$$

The solution of (28) can be given as $\mathbf{w}(i) = R_z^{-1}(i)\mathbf{r}_z(i)$, where $R_z(i) = \sum_{l=0}^{i} \lambda^{i-l}Z^H(l)Z(l)$ and $\mathbf{r}_z(i) = \sum_{l=0}^{i} \lambda^{i-l}Z^H(i)\boldsymbol{\delta}$. In the same way as we have shown in IV-A, the optimal weights $\mathbf{w}(i)$ can be updated by the recursive formula given below:

$$\mathbf{w}(i) = \mathbf{w}(i-1) + R_z^{-1}(i)Z^H(i)[\boldsymbol{\delta} - Z(i)\mathbf{w}(i-1)] \tag{29}$$

By following the same procedure as in IV-A, this calculation can be performed in a distributed mean as follow:

$$\begin{aligned} &\boldsymbol{\psi}_0(i) \leftarrow \mathbf{w}(i-1) \\ &\mathbf{z}_k(i) = |\mathbf{u}_k(i)\mathbf{w}(i-1)|^{p-2}\mathbf{w}^H(i-1)\mathbf{u}_k^H(i)\mathbf{u}_k(i) \\ &e_k(i) = \delta_k - \mathbf{z}_k(i)\mathbf{w}(i-1) \\ &\boldsymbol{\psi}_k(i) = \boldsymbol{\psi}_{k-1}(i) + R_z^{-1}(i)\mathbf{z}_k^H(i)e_k(i) \\ &\quad k = 1,2\ldots N \\ &\mathbf{w}(i) \leftarrow \boldsymbol{\psi}_N(i) \end{aligned} \tag{30}$$

Again, the global autocorrelation matrix $R_z^{-1}(i)$ can be updated based on the local data, say:

$$\begin{aligned} &R_0(i)^{-1} \leftarrow \lambda^{-1}R_z^{-1}(i-1) \\ &R_k^{-1}(i) = R_{k-1}^{-1}(i) - \frac{R_{k-1}^{-1}(i)\mathbf{z}_k^H(i)\mathbf{z}_k(i)R_{k-1}^{-1}(i)}{1+\mathbf{z}_k(i)R_{k-1}(i)\mathbf{z}_k^H(i)}, \\ &\quad k = 1,2\ldots N \\ &R_z^{-1}(i) \leftarrow R_N^{-1}(i) \end{aligned} \tag{31}$$

Finally, in the recursion part of (30), we can substitute $\mathbf{w}(i-1)$ and $R_z^{-1}(i)$ with $\boldsymbol{\psi}_{k-1}(i)$ and $R_k^{-1}(i)$, respectively, to attain the second version of distributed RLS-CMA, which is summarized in Algorithm 3.

Algorithm 3 Distributed adaptive RLS-CMA, p general

$\boldsymbol{\psi}_0(i) \leftarrow \mathbf{w}(i-1); R_0^{-1}(i) \leftarrow \lambda^{-1}R_z^{-1}(i-1)$
for $k=1:N$ **do**
  $\mathbf{z}_k(i) = |\mathbf{u}_k(i)\boldsymbol{\psi}_{k-1}(i)|^{p-2}\boldsymbol{\psi}_{k-1}^H(i)\mathbf{u}_k^H(i)\mathbf{u}_k(i)$
  $R_k^{-1}(i) = R_{k-1}^{-1}(i) - \frac{R_{k-1}^{-1}(i)\mathbf{z}_k^H(i)\mathbf{z}_k(i)R_{k-1}^{-1}(i)}{1+\mathbf{z}_k(i)R_{k-1}(i)\mathbf{z}_k^H(i)}$
  $e_k(i) = \delta_k - \mathbf{z}_k(i)\boldsymbol{\psi}_{k-1}(i)$
  $\boldsymbol{\psi}_k(i) = \boldsymbol{\psi}_k(i) + R_k^{-1}(i)\mathbf{z}_k^H(i)e_k(i)$
end for
$\mathbf{w}(i) \leftarrow \boldsymbol{\psi}_N(i); R_z^{-1}(i) \leftarrow R_N^{-1}(i)$

V. Simulation Results

In our simulations, we use the system model described in section II. In particular, we consider a quadrature amplitude modulation (QAM) communication framework with independent source signal samples $s(i)$ uniformly distributed over a unit magnitude QAM constellation [11] ;accordingly, the value of δ_k is set to 1 for $k=1,\ldots,N$. The unknown system in

Fig.1 is modeled as a time-invariant FIR filter with length $M = 10$ and randomly generated parameter vector $\boldsymbol{\beta}$, where each entry is derived from an i.i.d. complex circular Gaussian distribution with zero-mean and variance one. We consider a network of $N = 5$ distributed sensors with identical value of the signal-to-noise ratio (SNR) set to 20dB.

In our simulations, we compare the proposed distributed versions of the LMS-CMA and RLS-CMA to their non-distributed counterparts, i.e. in which the sensor nodes individually attempt to process their inputs without benefiting from any exchange of information with other nodes in the network. The performance of the developed algorithms is evaluated based on mean square error (MSE) criterion. Both distributed and non-distributed LMS-CMA run with equal step size of $\mu = 0.0001$. In the RLS-based algorithms, the forgetting factor is set to $\lambda = 0.96$, and $R_0^{-1}(0) = \beta I$ with the parameter $\beta = 0.01$. Both the LMS and RLS-based algorithms are initialized with the weight vector $\mathbf{w}(-1) = [1, 0, \ldots, 0]$.

The results shown in Fig. 2 and 3 are drawn over 500 independent runs, with different system parameter $\boldsymbol{\beta}$ selected as above for each run. The graphs in Fig. 2 indicate that, following an initial period of rapid learning, the distributed LMS-CMA and RLS-CMA achieve their steady-state level of residual error faster than the non-distributed LMS-CMA and RLS-CMA, respectively. Moreover, as a result of the spatial diversity introduced by the local nodes, the distributed LMS-CMA and RLS-CMA offer better steady state performance (i.e. lower residual error) as compared to the non-distributed algorithms. Note that for the Fig. 2, the values of p and q are set to two. Finally, the effect of the choice of the parameter p is illustrated in Fig. 3, where we observe that for this particular scenario, the best performance is obtained with $p = 1.5$.

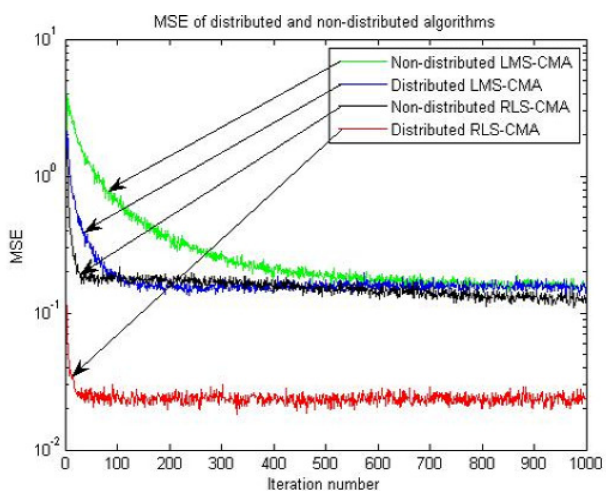

Fig. 2: MSE of distributed and non-distributed adaptive blind algorithms

VI. Conclusion

In this paper, we develop distributed LMS-CMA and RLS-CMA for wireless sensor network applications. In our model, the developed blind algorithm runs in the network in a distributed and adaptive manner over the joint time and space domains to estimate and track the parameters of an unknown underlying system. Simulation results demonstrate the effectiveness of the proposed algorithms, and show their superior performance over the corresponding non-distributed adaptive algorithms.

In this work, we have used a system identification example to develop the proposed algorithms. However, the estimation scenario under consideration can be generalized to more complex situations by modifying the underlying system model and making changes to the adaptive process running on individual nodes; this avenue is currently under investigation.

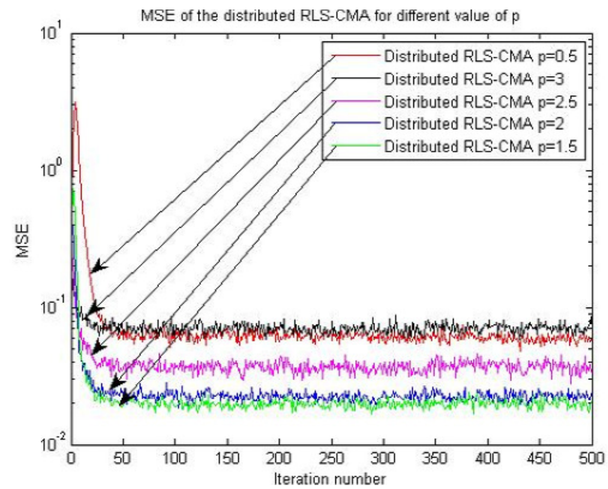

Fig. 3: MSE of the distributed RLS-CMA for different value of p

References

[1] M. Rabbat and R. Nowak, “Distributed optimization in sensor networks,” in *Int. Symp. on Inf. Proc. in Sensor Networks*, 2004, pp. 20–27.

[2] A. Sayed and C. Lopes, “Adaptive processing over distributed networks,” *IEICE Trans. Fund. of Elect. and Comm. Comp. Sci.*, vol. 90, no. 8, pp. 1504–1510, 2007.

[3] C. Lopes and A. Sayed, “Diffusion least-mean squares over adaptive networks,” in *Proc. IEEE Int. Conf. on Acoust., Speech, Signal Process.*, vol. 3, 2007, pp. 917–920.

[4] F. Cattivelli, C. Lopes, and A. Sayed, “Diffusion recursive least-squares for distributed estimation over adaptive networks,” *IEEE Trans. Signal Process.*, vol. 56, no. 5, pp. 1865–1877, 2008.

[5] I. Schizas, G. Mateos, and G. Giannakis, “Distributed LMS for consensus-based in-network adaptive processing,” *IEEE Trans. Signal Process.*, vol. 57, no. 6, pp. 2365–2382, 2009.

[6] G. Mateos, I. Schizas, and G. Giannakis, “Distributed recursive least-squares for consensus-based in-network adaptive estimation,” *IEEE Trans. Signal Process.*, vol. 57, no. 11, pp. 4583–4588, 2009.

[7] R. Johnson, P. Schniter, T. Endres, J. Behm, D. Brown, and R. Casas, “Blind equalization using the constant modulus criterion: A review,” *Proceedings of the IEEE*, vol. 86, no. 10, pp. 1927–1950, 1998.

[8] Y. Chen, T. Le-Ngoc, B. Champagne, and C. Xu, “Recursive least squares constant modulus algorithm for blind adaptive array,” *IEEE Trans. Signal Process.*, vol. 52, no. 5, pp. 1452–1456, 2004.

[9] S. Haykin, *Adaptive filter theory*. Prentice-Hall Information Systems Science, 1996.

[10] A. Sayed and C. Lopes, “Distributed processing over adaptive networks,” in *Proc. IEEE Int. Symp. on Signal Proc. & and Its App.*, 2007, pp. 1–3.

[11] J. Proakis and M. Salehi, *Digital communications*. McGraw-hill New York, 1995.