Adaptive Partial Update Algorithm Over Wireless Sensor Networks

Rupali Dewda[1], Megha Gupta[2], Vijay Barfa[3]
[1, 2] *Electronics Department, Rishiraj Institute of Technology, Indore, M.P. (India)*
[3]*Reliance Jio Infocomm Ltd., Indore*

***Abstract-* Adaptive partial update algorithm is developed based on incremental method. The proposed algorithm apply in real time changing environment. The proposed algorithm responds to linear estimation with nodes in co – operative manner and less number of computation. The algorithm has powerful advantages is that it require less number of coefficient and reduced computational and communication complexity in wireless sensor network. It is efficient because it have power of solving distributed estimation and optimization by learning mechanism. In wireless sensor networks there are various application that involve phenomenon in which space parameter are varying like surveillance, environment monitoring, battle field, precision agriculture and medical application. In this paper three algorithm sequential partial update, stochastic partial update and max – partial update are compared in terms of mean square error (MSE). Performance characteristic and complexity analysis of each algorithm are compared with MATLAB simulation.**

***Index Terms-* Incremental Networks, Max – partial update, sequential partial update, stochastic partial update, Mean square error.**

I. INTRODUCTION

A wireless sensor network (WSNs) composed of an array of sensor nodes, i.e. tiny embedded devices which are distributed in the geographical area. The adaptive distributed strategy which depends on the incremental mode of co-operation between different nodes, these nodes perform local computation and share the result with the immediate nodes [1]. The resulting algorithm is distributed, co-operative and able to react to the real time changing environment. These nodes interchangeably called agents. Adaptive filters play an important role in the fields related to digital signal processing and communication, such as noise cancellation, system identification, beamforming, channel equalization [2]. The LMS algorithm is widely used because of its low computational complexity and simplicity in implementation.

A well-known approach to controlling computational complexity is applying partial update (PU) method for adaptive filters. A partial update adaptive filter reduces computational complexity by updating part of the coefficient vector instead of updating the entire vector or by updating part of the time. Moreover, the partial update adaptive filters may converge faster than the full-update filters and achieve lower steady-state MSE in particular applications [3].

An important objective of adaptive signal processing is to ascertain the unknown and possibly time-varying signal statistics in conjunction with system estimating [2]. This paper presents a fundamental principle of adaptive signal processing and motivates partial-update adaptive signal processing as a low complexity implementation option in the face of resource constraints. In the context of adaptive system identification, partial coefficient updating is proposed as an attractive approach to complexity reduction [7]. This paper presents the potential benefits of partial-update adaptive signal processing in addition to allowing compliance with the existing resource constraints.

Distributed processing deals with the extract information from data collected at nodes that are distributed over a geographic area. For example, each node in a network could collect noisy observations related to a certain parameter or the phenomenon of interest. The nodes would then interact with their neighbors in a certain manner, as dictated by the network topology, in order to arrive at an estimate of the parameter or phenomenon of interest. The objective is to arrive at an estimate that is as accurate as the one that would be obtained if each node had access to the information across the entire network [3 - 4].

In adaptive signal processing, there are two different approaches that can be implemented to perform signal processing tasks over sensor network are centralizes and distributed techniques. In a centralized approach, nodes send their measured data to a central unit known as a fusion center for further processing and storage. In this approach transmitting the measured data to the fusion center may cause network congestion and results in waste of communication resources and power. In addition the fusion center requires relatively high computation power to process large amount of accumulating information. In distributed approach measured data are locally exchanged and processed within the network [4], [9]. In distributed approach, the network computational load is divided between nodes and no centralized infrastructure is needed. The single or multi-hop data transmission also reduces the network energy consumption because the power loss of wireless transmission increases linearly with respect to the propagation distance. These advantages give support the use of distributed signal processing for various applications in sensor networks.

Over the past few years, there has been covering a large area of research on distributed signal processing, as it supports the promise of overcoming the issue of bandwidth scarcity and limited energy budget in dense sensor networks. Distributed adaptive signal processing is going forth as a central enabling technology to sustain the carrying out of flexible co-operative learning and data processing strategy. The nodes which are distributed geographically have sensed, computing and communications capabilities. Distributed adaptive algorithm are useful for the solution of optimization problems and parameter estimation over the nets, where the signal statics are time-varying and unknown. The adaptive filter helps the network to track variations of the desired signal parameter and result of distributed adaptive processing a sensor network becomes robust against changes in the environmental condition and network topology [2].

In this paper, we study and develop distributed adaptive techniques of incremental type, for monitoring time varying physical phenomenon in sensor network under real- world constraints changes in environmental condition.

II. DISTRIBUTED LMS ALGORITHM

Here, we design and implement distributed algorithm that enable a network of nodes to function as an adaptive entity. Adaptive filter has the ability to respond in real time to its data and to variations in the statistical properties of this data, the same can be extended to the network domain. Specifically, the purpose of this paper is as follows:

1. Based on extensive work on distributed optimization. We can produce a family of incremental adaptive algorithm for distributed estimation.
2. The aim is to apply the developed incremental algorithm to meet the addressed adaptive network structures composed of an interconnected circle of nodes that are capable to respond to data in real time and to track variances in the statistical attributes of the data as follows:
   i) When an agent (nodes) receives a new piece of information from the environment, then the agents update its local estimate of the parameter of interest with this new information
   ii) The agents then shares the local estimates of the parameter with its immediate neighbors in a process that allows the information to flow to the other agents in the network.
   iii) To analyze the performance of the resulting interconnected network of nodes. This job is challenging since an adaptive network has comprised a "scheme of organizations" that processes data cooperatively in both time and space. Different nodes will converge to different mean-square-error (MSE) levels, reflecting the statistical diversity of the data and the different noise levels.

Finally, we considered all the factors that tend to design of an incremental adaptive algorithm over ring topologies and we derive closed form expressions for its mean-square performance in this paper.

III. INCREMENTAL LMS ALGORITHM FOR DISTRIBUTED ESTIMATION

For distributed optimization problem there have been extensive work for an incremental solution [4], [8] Consider a network with P nodes as shown in figure. 1

Let $\{d_k(i), v_{k,i}\}$, k = 1, 2, 3... P be the data available for a particular node k at a time instant i from the environment. At the time index i, the sensor at node k collects a measurement $d_k(i)$, where i denotes the discrete time index and k indicates the node index, and assuming an autoregressive (AR) model is adopted to represent these measurements as follows:

$$d_k(i) = \sum_{m=1}^{M} \beta_m \, d_k(i-m) + n_k(i) \quad (1)$$

Where $n_k(i)$ is additive zero – mean noise
Coefficients $\{\beta_m\}$ are the parameter of the underlying model.
Define parameter w^0 which is the desired optimum solution for the network which is
M×1 parameter vector

$$w^0 = col\{\beta_1, \beta_2, \ldots \beta_M\} \quad (2)$$

and regressor vector

$$u_{k,i} = d_k(i-1)\ d_k(i-2) \ldots d_k(i-m)] \quad (3)$$

then (1) at each node k can be given as

$$d_k^{(i)} = u_{k,i} w^0 + n_k(i) \quad (4)$$

Here, the objective is to estimate the model parameter vector w^0 from the measurement d_k (i) and $u_{k,i}$ over the network. Thus, in order to find the M × 1 vector w^0 , we formulate the linear space – time LMS estimation problem as

$$min_w \text{ J (w) and J (w)} = E||\boldsymbol{d} - \boldsymbol{U}w||^2 \quad (5)$$

Where $\{d_k$ (i), $u_{k,i}\}$ are realization of $\{\boldsymbol{d}_k, \boldsymbol{u}_k\}$. Thus optimum minimum mean – square error (MMSE) solution w^0 is calculated, for which the normal equation (5) is satisfied

$$R_{du} = R_u w^0 \quad (6)$$

Where R_u = E**U*U** and R_{du} = E**U*d**

When nodes in the network have access to data in order to get advantage of node cooperation, we can present a distributed network with incremental learning, where at least one cyclic path can be shown across the network. In this type of network, information should be transferred from one node to its immediate node in a cyclic manner to return to the initial node (see Fig. 1)

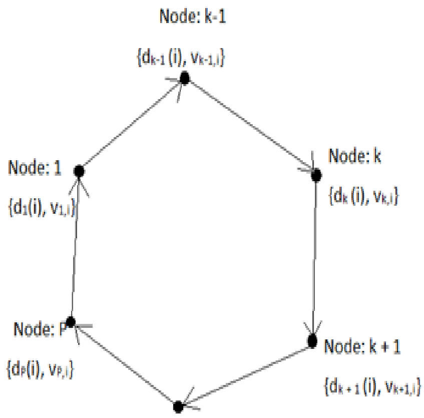

Fig. 1 Distributed network accessing data with P nodes

The incremental LMS solution for distributed network can given by [4]

$$\gamma_0^{(i)} = w_{i-1} \quad (7)$$

$$\gamma_k^{(i)} = \gamma_{k-1}^{(i)} + \mu_k v_{k,j}^* (d_k(i) - v_{k,i}\gamma_{k-1}^{(i)}) \quad (8)$$

K = 1, 2… P

$$w_i = \gamma_k^{(i)} \quad (9)$$

Where
$\gamma_k^{(i)}$= local estimate at node *k* at time *i*
μ_k = step size parameter at node *k*
w_i =estimate of w_i at node *k*
$\gamma_{k-1}^{(i)}$ = local estimate of immediate node *k* -1
$v_{k,i}$ = input at node *k* at i^{th} iteration.
$v_{k,i}^*$ is the hermitian of $v_{k,i}$ the above mentioned algorithm uses local data realizations d_k(i), $v_{k,i}$ and $\gamma_{k-1}^{(i)}$ weight estimate of immediate node. This incremental procedure purely relies on the local data estimation and gives truly distributed solution. At each iteration, each node uses local data realization $\{d_k(i)\ v_{k,i}\}$ and weight estimate received from its immediate nodes to perform following three tasks:

1) Evaluate a local error quantity
$$e_k(i) = d_k(i) - v_{k,i}\gamma_{k-1}^{(i)} \quad (10)$$
2) update its weight estimate:
$$\gamma_k^{(i)} = \gamma_{k-1}^{(i)} + \mu_k v_{k,i} e_k(i) \quad (11)$$
3) Pass the updated weight estimate to its neighbor node.

The distributed incremental adaptive algorithm implementation generally has better steady state performance and convergence rate. This paper shows the simulation results and performance analysis of the incremental LMS algorithm for distributed estimation.

IV. PARTIAL UPDATE INCREMENTAL SOLUTION

Partial update adaptive strategies are developed to reduce computational complexity. Even though the incremental adaptive solutions reduce the amount of communication to a considerable amount, the number of calculations of each iteration is equal to the LMS. In some application of adaptive filters have a large number of coefficients so updating the entire coefficient vector is expensive in term of memory, the number of hardware multiplier required for computation and power consumption. The more number of hardware multiplier implies more power. Here we propose an incremental partial update strategy which reduces computational complexity for considerable amount [3], [7], [8].

The incremental, partial update strategies reduce both communication complexity and computational complexity to a significant amount.

These techniques are following advantages:

- Less bandwidth is sufficient for the communication, i.e. Communication complexity is reduced as nodes in the network communicates with the immediate neighbors only.
- Computational complexity is brought down as number of hardware multipliers required is less compared to the usual LMS techniques.
- Low-energy sources reduces physical complexity through this technique and the need for the central processor is eliminated through the Incremental techniques in which nodes have local computational capabilities.

In this paper, we review some of incremental, partial update LMS techniques like Incremental-sequential partial update, incremental-stochastic partial update and incremental-Max-partial update algorithm.

A. SEQUENTIAL PARTIAL UPDATE LMS ALGORITHM

The sequential partial update method updates a subset of the adaptive filter coefficients so as to reduce the computational complexity associated with the adaptation process at each iteration for every node in the network. In this sense sequential partial update results in decimation of the adaptive filter coefficient vector. Coefficient subset to be updated are selected in a deterministic manner [5], [7].

The update equation is given by

$$\gamma_k^{(i)} = \gamma_{k-1}^{(i)} + \mu_k I_{N,k}^{(i)} e_k^{(i)} v_{k,i}^* \qquad (12)$$

Where $$e_k^{(i)} = d_k(i) - v_{k,i}\gamma_{k-1}^{(i)} \qquad (13)$$

$$I_{N,k}^{(i)} = \begin{bmatrix} b1(i) & 0 & 0 & . & . & 0 \\ 0 & b2(i) & 0 & . & . & 0 \\ . & . & b3(i) & . & . & . \\ . & . & . & b4(i) & . & . \\ . & . & . & . & . & . \\ 0 & . & . & . & . & bM(i) \end{bmatrix}$$

$$\sum_{j=1}^{M} b_j(i) = N \,, \; b_j(i) \in \{0,1\}$$

$I_{N,k}^{(i)}$ is the coefficient selection matrix to select a subset of N coefficient out of M total coefficient at node k at i^{th} iteration.

Let the coefficient index set be Q = {1, 2, 3…M} i.e. there are M coefficient totally out of which N coefficient are to be updated. Then Q is divided into S number of subset $L_1, L_{2,},..., L_s$ with each subset having N coefficient where S = c_N^M. Let R = M/N be an integer then R coefficient subsets are arranged in periodic sequences with respective coefficient selection matrix $I_{N,k}^{(i)}$

$$I_{N,k}^{(i)} = \begin{bmatrix} b1(i) & 0 & 0 & . & . & 0 \\ 0 & b2(i) & 0 & . & . & 0 \\ . & . & b3(i) & . & . & . \\ . & . & . & b4(i) & . & . \\ . & . & . & . & . & . \\ 0 & . & . & . & . & bM(i) \end{bmatrix}$$

b_j(i) = 1 if j ∈ $J_{(i\, mode\, R)}$+ 1 and zero otherwise.

For a given M and N, $I_{N,k}^{(i)}$ is not unique. Updating N out of M coefficient reduces the complexity of the adaptation process by a factor R.

B. STOCHASTIC PARTIAL UPDATE LMS ALGORITHM

Stochastic partial update algorithm improves the performance of the network over the sequential partial update algorithm with same amount of computational complexity reduction. In this method coefficient subset to be updated are chosen randomly instead of deterministic fashion as in the sequential partial update algorithm [3],[6].

The update equation is given by

$$\gamma_k^{(i)} = \gamma_{k-1}^{(i)} + \mu_k I_{N,k}^{(i)} e_k^{(i)} v_{k,i}^* \qquad (14)$$

The coefficient selection matrix is given by

$$I_{N,k}^{(i)} = \begin{bmatrix} b1(i) & 0 & 0 & . & . & 0 \\ 0 & b2(i) & 0 & . & . & 0 \\ . & . & b3(i) & . & . & . \\ . & . & . & b4(i) & . & . \\ . & . & . & . & . & . \\ 0 & . & . & . & . & bM(i) \end{bmatrix}$$

$b_j(i) = 1 \; if \; j \in J_{m(i)}$ and zero otherwise.

Where m(i) is an independent random process with probability mass function

P_r(m(i) = c) = π_c , c = 1… R

$\sum_{c=1}^{R} \pi_c = 1$

The computational complexity of stochastic algorithm (STPU) is same as that of the SEPU and slower than incremental LMS algorithm by a factor R because of the decimation of the adaptive filter coefficient.

C. MAX - PARTIAL UPDATE LMS ALGORITHM

In max-partial update algorithm at each iteration largest magnitude vector entries are updated.

This is a data dependent partial update technique which is based on finding N largest magnitude entries from M total coefficient [7].

The update equation is given by

$$\gamma_k^{(i)} = \gamma_{k-1}^{(i)} + \mu_k I_{N,k}^{(i)} e_k^{(i)} v_{k,i}^* \qquad (15)$$

Where $e_k^{(i)} = d_k(i) - v_{k,i}\gamma_{k-1}^{(i)}$ (16)

The coefficient selection matrix $I_{N,k}^{(i)}$ is given by

$$I_{N,k}^{(i)} = \begin{bmatrix} b1(i) & 0 & 0 & . & . & 0 \\ 0 & b2(i) & 0 & . & . & 0 \\ . & . & b3(i) & . & . & . \\ . & . & . & b4(i) & . & . \\ . & . & . & . & . & . \\ 0 & . & . & . & . & bM(i) \end{bmatrix}$$

$b_j(i) = 1 \; If \; |v(i-j+1)| \in max_{1\le l\le M}(|v(i-j+1)|, N)$

$b_j(i) = 0 \; otherwis$

Here $max_j(w_j, N)$ indicates the set of N maxima of w_j.

The Max partial update is similar to the sequential partial update in decimating the coefficient update vector, only the magnitude of the update vector entries to be ranked before updating instead of deterministic fashion in sequential update method. This coefficient selection scheme determines the convergence of the adaptive filters.

This reduces complexity by factor R = M/N

V. SIMULATION

In this part we compare the result of each technique. Number of nodes in network P = 20. The regressor vector or data vector $v_{,,i}$ is $1\times M$ and collects the data as follows.

$$v_{k,i} = col\{v_k(i), v_k(i-1), \ldots \ldots v_k(i-M+1)\} \qquad (17)$$

In this network every node k depends on local statistics and influenced by immediate neighbors. 300 independent experiments were performed and averaged. In all experiment step size parameter should be small as possible and constant. The curve are generated for 100 iterations. The mean square error is taken as the performance unit. Mean square error (MSE) gives how far the local estimate from the optimum weight w^0 . The performances of proposed algorithm are compared with that of incremental algorithm.

Parameter setting

Ring topology is considered for all proposed algorithm as shown in figure 1

Table: Proposed incremental algorithms MSE results

% of coefficient update	Number of coefficients M	Step size μ	Updated coefficient N	Simulation MSE Result for Sequential Partial Update Incremental LMS	Simulation MSE Result for Stochastic Partial Update Incremental LMS	Simulation MSE Result for Max Partial Update Incremental LMS
70	10	0.03	7	0.1203	0.1389	0.0593
50	10	0.03	5	0.2005	0.1910	0.1014
30	10	0.03	3	0.2255	0.2025	0.1710

Above MSE result are compared with incremental algorithm in which all the coefficient are updated whose MSE is 0.0078

The simulation results for performance estimation are compared with incremental LMS in which all the coefficient is updated at each iteration. From simulation result and figure, we say that max – partial update outperform sequential partial update and stochastic partial update in performance. The stochastic PU technique gives better performance over sequential PU for same computational complexity. But sequential PU converges with faster convergence rate compared to others to the algorithm. Stochastic PU convergence at a faster rate compared to max partial update.

The advantage of the proposed algorithm over incremental algorithm is achieved at the cost of degradation in performance.

From the observing simulation result, it is obvious-

1. Because the incremental mode of communication is considered every node k is influenced by its immediate neighbors.
2. Mean – square error depends on number of coefficient updated.
3. It is more sensitive to local statistics.

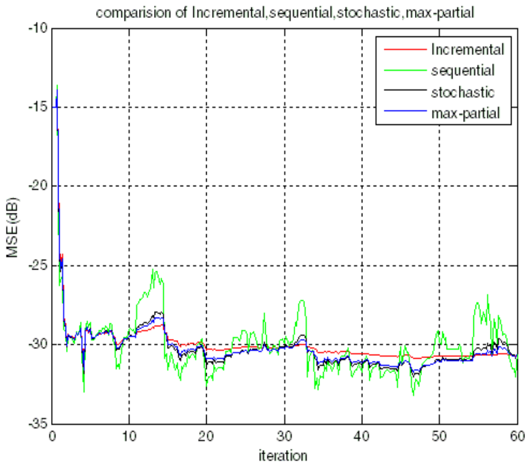

Figure 2 Comparison of each technique with incremental LMS for 70% coefficient update

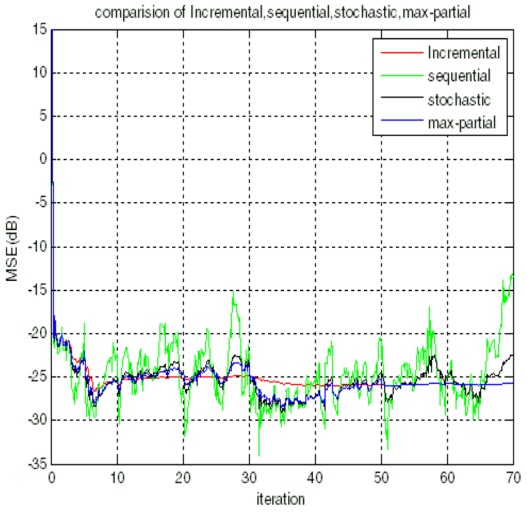

Fig. 3 Comparison of each techniques with incremental LMS for 50 % Coefficient update

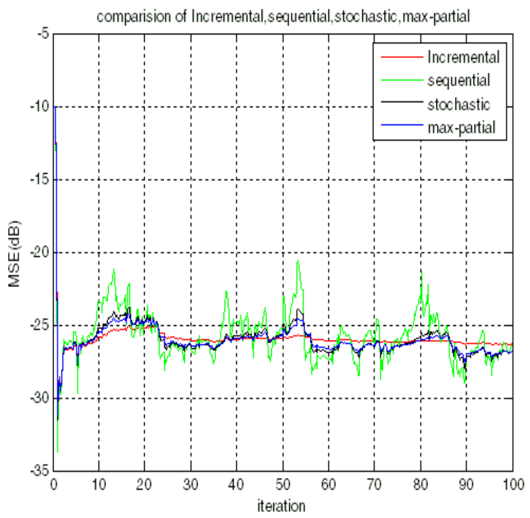

Fig.4 Comparison of each techniques with incremental LMS for 30 % Coefficient update

VI. CONCLUSION

It is clear from the result and analysis that sequential and stochastic partial update algorithm reduced the computational complexity but stochastic partial update algorithm gives better performance compared to sequential partial update algorithm. Max – partial update algorithm converge faster and has congruous steady state performances and reduce computational complexity as the same amount as other two algorithms. So with little worse in the performance the computational complexity can reduce to a considerable amount. This reduces power consumption and suitable for low energy budget, i.e. low energy sources.

REFERENCES

[1]. Bei Xie, Tamal Bose, "Partial update algorithm for adapting filtering"

[2]. R. Abdolee, B. Champagne, "Diffusion LMS strategies in sensor networks with noisy input data application', submitted to IEEE/ ACM Trans. On Networking.

[3]. Cassio G.Lopes and Ali H. Sayed. "Incremental adaptive strategies over distributed Networks", IEEE Trans. On signal processing, Vol. 55, No 8, August 2007.

[4]. Leilei Li, Jonathon A. Chambers and Cassio G. Lopes. "Distributed estimation over an adaptive incremental network based on the affine projection algorithm", IEEE Trans. On signal processing, Vol. 58, No. 1, January 2010.

[5]. Mahesh. Godavarti and Alfred O. Hero, "Sequential partial update LMS algorithm" IEEE Trans. on signal processing, October 2001

[6]. Mahesh. Godavarti and Alfred O.Hero, "Stochastic partial update LMS algorithm" IEEE Trans. On signal processing, October 2001.

[7]. Kutluyil Dogancay," partial-update adaptive signal processing Design, analysis and implementation"

[8]. Mahesh. Godavarti and Alfred O. Hero, "Partial Update LMS algorithm", journal of latex class files, vol.1, No.11, November 2002

[9]. M.G Rabbat and R. D. Nowak "quantized incremental algorithms for distributed optimization", IEEE J.Sel. areas Commun., vol.23,no.4.pp798- 808,Apr.2005