Adaptive Distributed Incremental Networks Based on the Selective Partial Update Affine Projection Algorithm

Ali-Reza Danaee and Mohammad Shams Esfand Abadi
Faculty of Electrical and Computer Engineering, Shahid Rajaee Teacher Training University, Tehran, Iran
Emails: ar.danaee@srttu.edu, mshams@srttu.edu

***Abstract*—This paper presents the problem of distributed estimation in an adaptive incremental network based on the selective partial update affine projection algorithm (SPU-APA). In dSPU-APA, the weight coefficients are partially updated at each node during the adaptation. The distributed SPU-APA (dSPU-APA) has low computational complexity feature and close convergence speed to ordinary dAPA. The good performance of dSPU-APA is demonstrated by several computer simulations in distributed network.**

***Index Terms*—Affine projection algorithm, adaptive distributed estimation, incremental network, selective partial update.**

I. Introduction

In contrast to classical centralized methods, distributed processing uses local computations at each node and communications among neighboring nodes to solve problems over the entire network which is a practical use among a wide range of applications. Because of data statistical variations and network topology changes, the processing should be adaptive and requires two time scales in distributed networks. During the initial period of time, each node makes a special estimation and then through consensus iterations, the nodes incorporate further estimations to attain the desired estimate [1], [2]. In incremental learning algorithms over a distributed network, each node cooperates only with one adjacent node to utilize the spatial dimension, whilst doing local computations in the time dimension [3]. Within the last decade, different incremental adaptive strategies such as distributed least mean squares (dLMS), distributed normalized LMS (dNLMS), and distributed affine projection algorithm (dAPA) were introduced over distributed networks [3], [4], [5]. Unfortunately, in comparison with dLMS and dNLMS, the computational complexity of dAPA is high.

In some of applications such as adaptive distributed networks, a high computational load is needed to achieve an acceptable performance. Therefore the computational complexity is an important problem in these applications. Several single adaptive filter algorithms such as adaptive filter algorithms with selective partial updates have been proposed to reduce the computational complexity. These algorithms update only a subset of filter coefficients at each time iteration. The selective partial update normalized least mean square (SPU-NLMS) [6], [7] is important examples of this family of adaptive filter algorithms. The SPU approach was also successfully extended to APA in [6].

In this paper we apply the approach of SPU-APA to adaptive distributed network and the distributed SPU-APA (dSPU-APA) is established. This algorithm has good convergence speed, low steady-state error, and low computationally complexity features. This paper is organized as follows. In Section II, the dSPU-APA is introduced. Finally, before concluding the paper, the good performance of dSPU-APA is demonstrated.

Throughout the paper, the following notations are adopted:

$(.)^T$	Transpose of a vector or a matrix.
$Tr(\cdot)$	Trace of a matrix.
$(.)^{-1}$	Inverse of the matrix

II. The dSPU-APA Weight Update Equation

Due to special features of selective partial update affine projection algorithms such as high convergence speed, low computational complexity and low steady state mean square error, we apply SPU-APA for a distributed estimation in an incremental network. The weight vector update equation for the dAPA in an incremental network over J nodes is introduced as

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \sum_{k=1}^{J} \mu_k \mathbf{X}_k(n)(\mathbf{X}_k^T(n)\mathbf{X}_k(n))^{-1}\mathbf{e}_k(n), \quad (1)$$

where

$$\mathbf{e}_k(n) = \mathbf{d}_k(n) - \mathbf{X}_k^T(n)\mathbf{h}_{k-1}(n), \quad (2)$$

is the error vector, and μ_k is the step-size of each node. In (2), $\mathbf{X}_k(n) = [\mathbf{x}_k(n), \mathbf{x}_k(n-1), \ldots, \mathbf{x}_k(n-(P-1))]$ is the $M \times P$ block data signal matrix and $\mathbf{d}_k(n) = [d_k(n), d_k(n-1), \ldots, d_k(n-(P-1))]^T$ is the $P \times 1$ data vector where $\mathbf{x}_k(n) = [x_k(n), x_k(n-1), \ldots, x_k(n-M+1)]^T$. We assume the linear data model between the unknown system vector $\mathbf{h}_t$, $\mathbf{d}_k(n)$, and $\mathbf{X}_k(n)$ as $\mathbf{d}_k(n) = \mathbf{X}_k^T(n)\mathbf{h}_t + \mathbf{v}_k(n)$, where $\mathbf{v}_k(n)$ is a temporally and spatially independent noise sequence with variance $\sigma_{v,k}^2$ at each node.

The selective partial update approach in single adaptive filter was successfully extended to APA in [6]. Following the same approach in [6], the weight vector update equation for dSPU-APA is introduced as

$$\mathbf{w}_F(n) = \mathbf{w}_F(n-1) + \sum_{k=1}^{J} \mu_k \mathbf{X}_{k,F}(n)(\mathbf{X}_{k,F}^T(n)\mathbf{X}_{k,F}(n))^{-1}\mathbf{e}_k(n), \quad (3)$$

where $F = \{j_1, j_2, \ldots, j_N\}$ denote the indexes of the N blocks out of K blocks that should be updated at every adaptation, and

$$\mathbf{X}_{k,F}(n) = [\mathbf{X}^T_{k,j_1}(n), \mathbf{X}^T_{k,j_2}(n), \ldots, \mathbf{X}^T_{k,j_N}(n)]^T, \quad (4)$$

is the $NL \times P$ matrix and

$$\mathbf{X}_{k,i}(n) = [\mathbf{x}_{k,i}(n), \mathbf{x}_{k,i}(n-1), \ldots, \mathbf{x}_{k,i}(n-(P-1))] \quad (5)$$

is the $L \times P$ matrix. The indexes of F are obtained by the following procedure:

1) Compute the following values for $1 \leq i \leq K$

$$\mathrm{Tr}(\mathbf{X}^T_{k,i}(n)\mathbf{X}_{k,i}(n)) \quad (6)$$

2) The indexes of F are correspond to N largest values of (6).

In dSPU-APA, the weight coefficients are partially updated at every iteration for each node. The dSPU partial rank algorithm (dSPU-PRA) can be established when the adaptation of the weight coefficients is performed only once every P iterations. Equation (3) can be represented in the form of full update equation as

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \sum_{k=1}^{J} \mu_k \mathbf{A}_k(n)\mathbf{X}_k(n)(\mathbf{X}^T_k(n)\mathbf{A}_k(n)\mathbf{X}_k(n))^{-1}\mathbf{e}_k(n), \quad (7)$$

where the $\mathbf{A}_k(n)$ matrix is the $M \times M$ diagonal matrix with the $\mathbf{1}$ and $\mathbf{0}$ blocks each of length L on the diagonal and the positions of 1's on the diagonal determine which coefficients should be updated in each iteration. The positions of $\mathbf{1}$ blocks (N blocks and $N \leq K$) on the diagonal of $\mathbf{A}_k(n)$ matrix for each iteration in dSPU-APA are determined by the indexes of F.

Fig. 1 shows the learning algorithm in an incremental network for dSPU-APA. Based on the local data ($\mathbf{d}_k(n)$, $\mathbf{X}_k(n)$), and $\mathbf{h}_{k-1}(n)$ from its previous node $(k-1)$ in the cycle, the update for local estimation is performed at each time instant n. Finally, the local estimation $\mathbf{h}_J(n)$ is used as the global estimation for $\mathbf{w}(n)$. This value is used as the initial local estimation $\mathbf{h}_1(n+1)$ for the next time instant $n+1$. The pseudo-code implementation of dSPU-APA has been described in Table I.

TABLE I
PSEUDO-CODE IMPLEMENTATION OF DAPAS

For each time instant $n \geq 0$ repeat:
$k = 1, \ldots, J$
$\mathbf{h}_0(n) = \mathbf{w}(n-1)$
$\mathbf{h}_k(n) = \mathbf{h}_{k-1}(n) + \mu_k \mathbf{A}_k(n)\mathbf{X}_k(n)(\mathbf{X}^T_k(n)\mathbf{A}_k(n)\mathbf{X}_k(n))^{-1}\mathbf{e}_k(n)$
end
$\mathbf{w}(n) = \mathbf{h}_J(n)$

III. SIMULATION RESULTS

We justified the performance of dSPU-APA in an incremental network with $J = 20$ nodes in a system identification set-up. The impulse response of the random unknown system has $M = 16$ taps. The correlated input signal, $x_k(n)$, at each node are generated by passing a white Gaussian noise process with unit variance through a first order autoregressive model filter with the z-transfer function $\frac{\sqrt{1-\alpha_k^2}}{1-\alpha_k z^{-1}}$ and $\alpha_k \in [0, .5)$. The noise sequence of each node is a white Gaussian process with variance $\sigma^2_{v,k} \in (0, 0.1]$. Fig. 2 shows the node profiles of $\sigma^2_{v,k}$ and α_k. As we can see, node 10 has the highest correlated input. Therefore, we present the results for this node. In all simulations, the simulated MSE learning curves are obtained by ensemble averaging over 200 independent trials. Also, the steady-state MSE values are obtained by averaging over 1000 steady-state samples from 100 independent realizations for each value of μ for a given algorithm. The step-sizes were chosen to get approximately the same steady-state MSE in all learning curves.

Figs. 3, and 4 show the learning curves of MSE and MSD for dSPU-APA in node 10. The number of blocks (K) was set to 4. This figure describes that by increasing the parameter N, the performance of dSPU-APA will be close to dAPA especially for $N = 3$. Furthermore, the computational complexity of dSPU-APA is lower than dAPA due to SPU approach.

Figs. 5-7 show the simulated steady-state MSE, EMSE, and MSD values of dSPU-APA for each node with $\mu_k = 0.5$. The curves have been obtained for various values of N. The steady-state errors for dSPU-APA with high values of N is lower than dSPU-APA with low values of N. For $N = 3$, the steady-state error values are very close to dAPA algorithm. Furthermore, the computational complexity of dSPU-APA is lower than dAPA.

Fig. 8 shows the simulated steady-state MSE values of dSPU-APA as function of the step-size for node 10. Various values for N have been chosen. The step-sizes change in the range [0.008, 0.5] which are in the stability bound of dSPU-APA. By increasing the parameter N, the steady-state MSE values of dSPU-APA are close to dAPA algorithm.

IV. SUMMARY AND CONCLUSION

In this paper we applied selective partial update affine projection adaptive algorithms to the distributed estimation in an incremental network. Based on this, dSPU-APA was established. Simulation results show that dSPU-APA has good convergence speed, low steady-state MSE, EMSE, and MSD, and low computational complexity features.

REFERENCES

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, pp. 102–114, 2002.

[2] D. Li, K. D. Wong, Y. H. Hu, and A. M. Sayeed, "Detection, classification, and tracking of targets," *IEEE Signal Processing Magazine*, vol. 19, pp. 17–29, 2002.

[3] C. G. Lopes and A. H. Sayed, "Incremental adaptive strategies over distributed networks," *IEEE Trans. Signal Processing*, vol. 55, pp. 4064–4077, 2007.

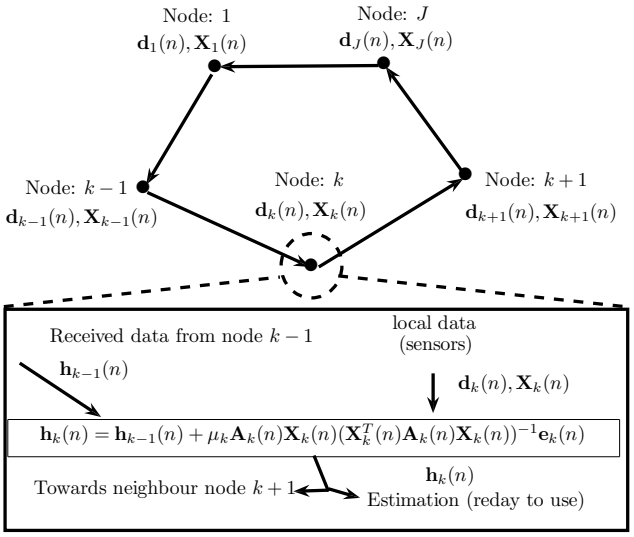

Fig. 1. Data processing of the family of dAPAs in an incremental network.

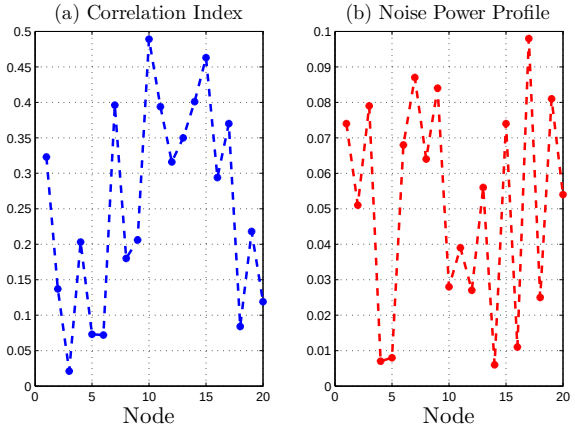

Fig. 2. Node profile: (a) Correlation index for the Gaussian data Network (b) Noise power for the Gaussian data network.

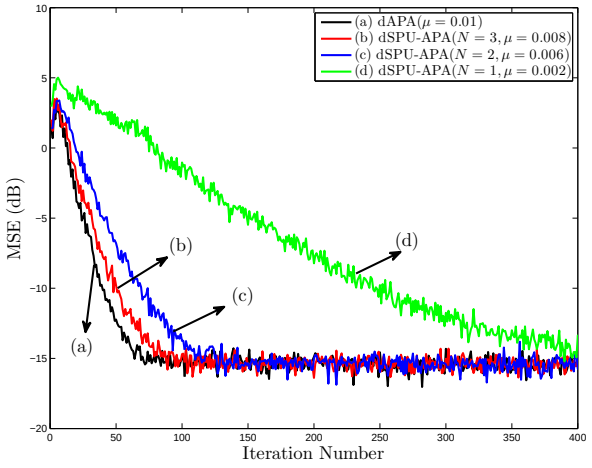

Fig. 3. Simulated learning MSE curves of dAPA and dSPU-APA with $N = 1, 2, 3$ for node 10.

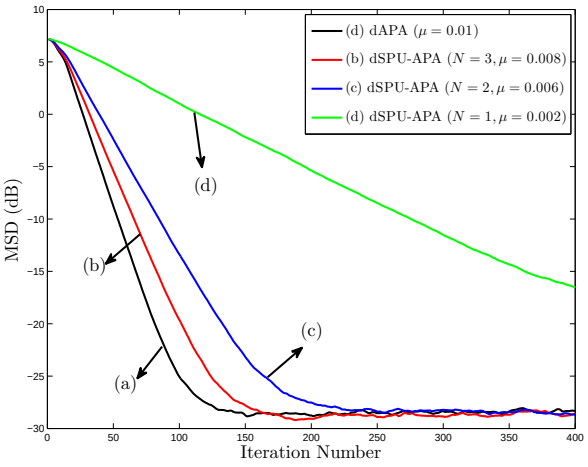

Fig. 4. Simulated learning MSD curves of dAPA and dSPU-APA with $N = 1, 2, 3$ for node 10.

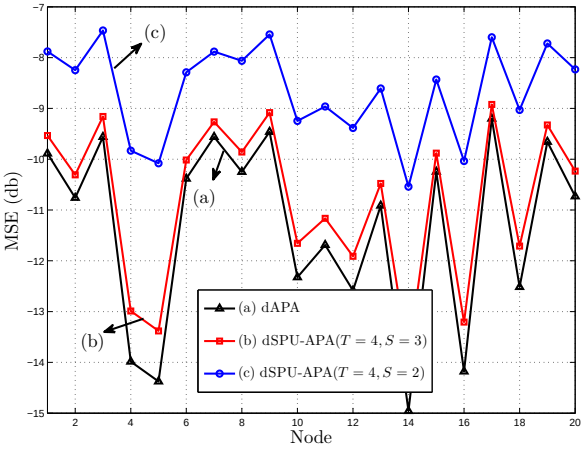

Fig. 5. Steady-state MSE of all nodes for dAPA and dSPU-APA using $\mu_k = 0.5$ ($N = 2, 3$).

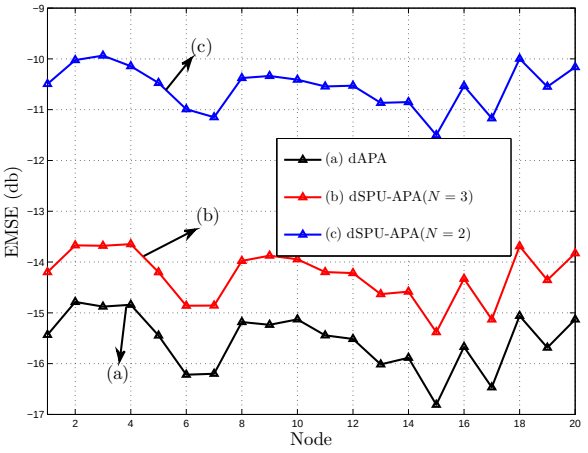

Fig. 6. Steady-state EMSE of all nodes for dAPA and dSPU-APA using $\mu_k = 0.5$ ($N = 2, 3$).

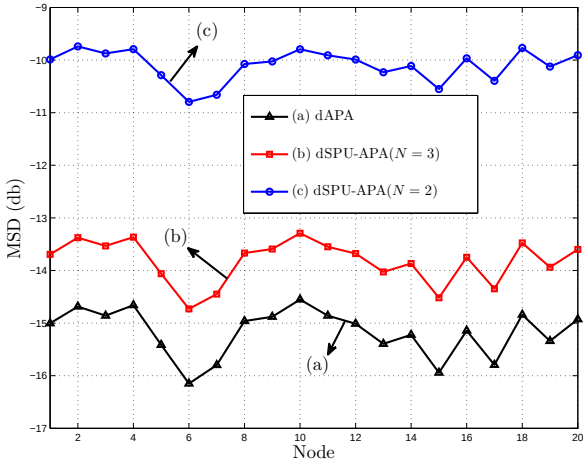

Fig. 7. Steady-state MSD of all nodes for dAPA and dSPU-APA using $\mu_k = 0.5$ ($N = 2, 3$).

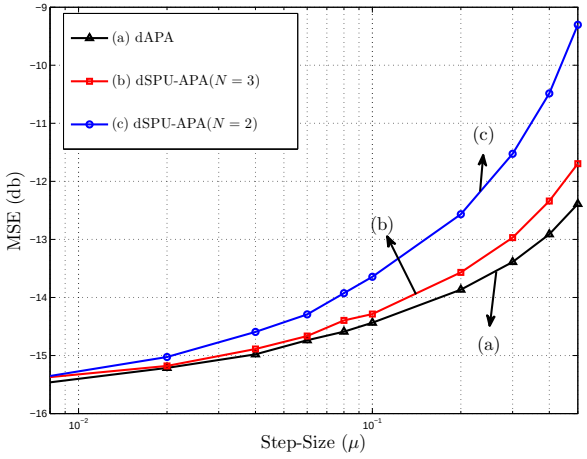

Fig. 8. Simulated steady-state MSE curves of dAPA and dSPU-APA at node 10 as a function of the step-size ($N = 1, 2, 3$).

[4] L. Li, J. A. Chambers, C. G. Lopes, and A. H. Sayed, "Distributed estimation over an adaptive incremental network based on the affine projection algorithm," *IEEE Trans. Signal Processing*, vol. 58, pp. 151–164, 2010.

[5] X. Zhao and A. H. Sayed, "Performance limits for distributed estimation over LMS adaptive networks," *IEEE Trans. Speech., Audio Processing*, vol. 60, pp. 5107–5124, 2012.

[6] K. Doğançay and O. Tanrıkulu, "Adaptive filtering algorithms with selective partial updates," *IEEE Trans. Circuits, Syst. II: Analog and Digital Signal Processing*, vol. 48, no. 8, pp. 762–769, Aug. 2001.

[7] S. Werner, M. L. R. de Campos, and P. S. R. Diniz, "Partial-update NLMS algorithms with data-selective updating," *IEEE Trans. Signal Processing*, vol. 52, no. 4, pp. 938–948, Apr. 2004.