Available online at www.sciencedirect.com

ScienceDirect

AASRI Procedia 5 (2013) 77 – 84

AASRI Procedia

www.elsevier.com/locate/procedia

2013 AASRI Conference on Parallel and Distributed Computing Systems

A Distributed Variable Tap-length Algorithm within Diffusion Adaptive Networks

Leilei Li[a*], Jingfeng Feng[a], Jianhui He[a], Jonathon A. Chambers[b]

[a]*Academy of Broadcasting Planning in SARFT, No. 2 FuXingMenWai Street Beijing, 10045, China*
[b] *School of Electronic, Electrical & systems Engineering, Loughborough University, Loughborough LE11 3TU, UK*

Abstract

In this work, an adaptive algorithm with variable tap-length is proposed for diffusion networks, where all nodes exchange information with their connected group without a fusion center and utilize both exchanged information and their local observation to make their individual estimations. Based on a fractional tap-length solution and the fully distributed structure of networks, we propose the algorithm to resolve the estimation problems on both tap-length and tap-weights. The proposed algorithm with adaptive tap-length structure enables the networks to reduce communication energy and network resources. The convergence properties of this algorithm are confirmed by the simulation results.

Keywords: variable tap-length, distributed algorithms, adaptive filters, diffusion networks;

1. Background

As an essential parameter in filter structure, tap-length makes an impact on the performance of an adaptive filter. The recent work [1] demonstrates that an adaptive filter with deficient tap-length would makes the

* Corresponding author. Tel.: +8610-86092072; fax: +8610-86092645 .
E-mail address: lileilei@abp.gov.cn .

doi:10.1016/j.aasri.2013.10.061

mean-square-error (MSE) deteriorate; in order to avoid this situation, we can select an adequate long tap-length at the steady-state stage, but it will cause the high calcualtion cost and the high excess mean-square-error (EMSE). Since the research [2] firstly utilized the variable tap-length to improve the performance of an adaptive algorithm, several research articles [3]-[5] on adaptive tap-length algorithms have been presented. Compared with the previous algorithms, the fractional tap-length (FT) algorithm presented in [6] achieved a robust and improved performance with a low- complexity. In addition, the works [7, 8] also provided the analysis for the FT algorithm on the relation between parameter selection and the steady-state performance.

Within the networks, the tap-length of the adaptive filter at each node is assumed as a fixed value in many applications of distributed adaptive algorithms [9, 10]. Nevertheless, when tap-length is unknown or variable it gives rise to the tap-length estimation problem for distributed networks. The incremental adaptive algorithm with the fractional tap-length solution has been proposed before in the literature [11]. This algorithm is based on incremental learning over a distributed network, where a Hamiltonian cycle is built to make all the nodes communicate its neighbor at every iteration. The network topology is therefore determined by this type of connection. In order to resolve the tap-length estimation problem within networks with fully distributed structure, we introduce the fractional tap-length solution. In this type of network, each node not only communicates with its neighbors, but also experiences the effect of the entire network. In this paper, the fractional solution is utilized in the proposed distributed variable tap-length algorithm to resolve the both tap-length and tap weights estimation problems in a distributed diffusion network. We provide performance analysis for the new algorithm and provide simulation results to confirm its convergence properties.

The remainder of this work consists of 4 parts. Section 2 provides the estimation problem which motivates us to formulate the proposed algorithm. And then, the steady-state performance of this algorithm is analyzed in Section 3. Section 4 shows that the simulation is carried out to confirm the analysis. In section 5, we draw the conclusions.

2. Motivation and Formulation

Assume a network consists of N nodes, which are distributed over some area. When two nodes can exchange information with each other in a direct way, we regard them as connected. Each node can connect to itself. All the nodes communicating with node k are included in the its neighborhood, which is defined as $\aleph_k$. The total number of neighbors of node k is denoted by n_k . At each instant i , every node k receives the time measurements $\{d_k(i), u_{k,i}\}$ of some random process $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$, which have relation to the unknown system vector w^o of size L^o as follows:

$$\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i} w^o + \boldsymbol{v}_k(i) \tag{1}$$

where $\boldsymbol{u}_{k,i}$ is a $L^o \times 1$ regressor and $\boldsymbol{v}_k(i)$ is a temporally and spatially white noise sequence, i.e.

$$E\boldsymbol{v}_k(i)\boldsymbol{v}_l(j) = \delta_{kl}\delta_{ij}\sigma_{v,k}^2 \tag{2}$$

where E is expectation operator and δ indicates the Kronecker delta.

The adaptation rules are decouple into the tap weights estimation and tap-length estimation to simplify the analysis so that tap weights and tap-length of w^o are calculated respectively. The data are collected from all nodes $k = 1,\ldots,N$ into global matrices: $\boldsymbol{d} = \mathrm{col}\{\boldsymbol{d}_1,\ldots,\boldsymbol{d}_N\}$ $(N\times 1)$, $\boldsymbol{v} = \mathrm{col}\{\boldsymbol{v}_1,\ldots,\boldsymbol{v}_N\}$ $(N\times 1)$, $\boldsymbol{U} = \mathrm{col}\{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_N\}$ $(N\times L^o)$, where col is used to make its elements in column. Therefore, we can rewrite the expression *(1)* as

$$\boldsymbol{d} = \boldsymbol{U}w^o + \boldsymbol{v} \tag{3}$$

where $R_v = E\boldsymbol{v}\boldsymbol{v}^*$ and * is complex conjugate transposition.

Since the tap-length is unknown, we assume the estimate tap-length w as $L(i)$. In order to seek the optimal tap-length L^o, we recall the linear minimum mean-square estimation problem [11]:

$$\min_w J_{L(i)}(w) \text{ and } J_{L(i)}(w) = E\left\|\boldsymbol{d} - \boldsymbol{U}_{L(i)} w_{L(i)}\right\|^2 \tag{4}$$

where $\boldsymbol{U}_{L(i)}$ is a matrix of size $N \times L(i)$. Then, the segmented cost function is defined as $J_M(w) = E\left\|\boldsymbol{d} - \boldsymbol{U}_M w_M\right\|^2$, where M is defined as $1 \le M \le L(i)$, w_M and $\boldsymbol{U}_M$ is composed of the initial M elements/column vectors of the corresponding quantity, respectively, as

$$w_M = \mathrm{col}\{w(1), \ldots, w(M)\} \qquad (M \times 1) \tag{5}$$

$$\boldsymbol{U}_M = \mathrm{col}\{\boldsymbol{u}_1(1:M), \ldots, \boldsymbol{u}_N(1:M)\} \qquad (N \times M) \tag{6}$$

We resort to the mean-square errors estimation problem and exploit its minimum difference in order to obtain the optimal tap-length L^o

$$\min\{L(i) \mid J_M(w) - J_{L(i)}(w) \le v\} \tag{7}$$

where v indicates a small positive value and is predetermined by system requirements. At time instant i, we define the segmented mean-square error and the mean-square error respectively as, $J_{L(i)-\Delta}(w_{i-1}) = e^2_{L(i)-\Delta}(w_{i-1})$ and $J_{L(i)}(w_{i-1}) = e^2_{L(i)}(w_{i-1})$, where $\Delta = L(i) - M$ is an integer to avoid the suboptimum tap-length during the learning process so that the value of Δ is generally more than one.

We recall the standard pseudo fractional tap-length implementation for a single adaptive filter as in [6]

$$L_f(i) = \left(L_f(i-1) - \alpha\right) + \beta\left(e^2_{L(i)-\Delta}(w_{i-1}) - e^2_{L(i)}(w_{i-1})\right) \tag{8}$$

$$L(i) = \begin{cases} \lfloor L_f(i) \rfloor, & \text{if} \left|L(i-1) - L_f(i)\right| > \gamma \\ L(i-1), & \text{otherwise} \end{cases} \tag{9}$$

where α is small positive value and denotes the leakage parameter utilized to avoid an undesirable large value of $L_f(i)$, β denotes the step-size parameter for updating fractional tap-length $L_f(i)$. Equation *(9)* is used to calculate the integer estimate $L(i)$, $\lfloor \cdot \rfloor$ denotes rounding the included value to the nearest integer and the threshold γ, a small integer, is generally set as one.

The distributed concept is introduced for tap-length adaptation at each node k, and we can rewrite *(8)* as

$$l_{k,f}(i-1) = \sum_{l \in \aleph_k} b_{l,k} L_{k,f}(i-1) \tag{10}$$

$$L_{k,f}(i) = \left(l_{k,f}(i-1) - \alpha_k\right) + \beta_k\left(e^2_{L_k(i-1)-\Delta_k}(w_{k,i-1}) - e^2_{L_k(i-1)}(w_{k,i-1})\right) \tag{11}$$

where α_k and β_k indicate the local parameters, $l_{k,f}(i-1)$ and $L_{k,f}(i-1)$ denote the integrated and local fractional tap-length estimates, $b_{l,k}$ is local weight coefficient for tap-length combination and the set of $b_{l,k}$ satisfies

$$\sum_{l \in \aleph_k} b_{l,k} = 1 \tag{12}$$

and $L_k(i)$ is integer estimate tap-length, given by

$$L_k(i) = \begin{cases} \lfloor L_{k,f}(i) \rfloor, & \text{if} \left| L_k(i-1) - L_{k,f}(i) \right| > \gamma_k \\ L_k(i-1), & \text{otherwise} \end{cases} \tag{13}$$

where γ_k is the local small integer and

$$e_{L_k(i)-\Delta_k}(w_{k,i-1}) = d_k(i) - u_{k,i}(1 : L_k(i) - \Delta_k)(w_{k,i-1}(1 : L_k(i) - \Delta_k)) \tag{14}$$

$$e_{L_k(i)}(w_{k,i-1}) = d_k(i) - u_{k,i}(1 : L_k(i))(w_{k,i-1}(1 : L_k(i))) \tag{15}$$

We note that equation *(10)* is a consultation step, where the integrated estimate $l_{k,f}(i)$ collects the estimates $L_{k,f}(i)$ from the neighbors of node k and combines them through the weights $b_{l,k}$. Equation *(11)* is a local tap-length adaptation step, where the local segmented error $e_{L_k(i)-\Delta_k}(w_{k,i-1})$ and local error $e_{L_k(i)}(w_{k,i-1})$ are used to update the local fractional estimate $L_{k,f}(i)$. Then, we used the distributed adaptation form described in [12, 13] to summarize the proposed algorithm below:

$$\boxed{\begin{array}{l} \psi_{k,i-1} = \sum_{l \in \aleph_k} a_{l,k} w_{l,i-1} \\ L_{\max} = \max\{size(w_{l.i-1})\} \text{and make } \psi_{k,i-1} \text{ a } L_{\max} \times 1 \text{ column vector} \\ w_{k,i} = \psi_{k,i-1} + \mu_k u_{k,i}^{*}(d_k(i) - u_{k,i}\psi_{k,i-1}) \\ l_{k,f}(i-1) = \sum_{l \in \aleph_k} b_{l,k} L_{k,f}(i-1) \\ L_{k,f}(i) = (l_{k,f}(i-1) - \alpha_k) + \beta_k\left(e^2_{L_k(i)-\Delta_k}(w_{k,i-1}) - e^2_{L_k(i)}(w_{k,i-1})\right) \\ L_k(i) = \begin{cases} \lfloor L_{k,f}(i) \rfloor, & \text{if} \left| L_k(i-1) - L_{k,f}(i) \right| > \gamma_k \\ L_k(i-1), & \text{otherwise} \end{cases} \\ \text{make } w_{k,i} \text{ a } L_k(i) \times 1 \text{ column vector} \end{array}} \tag{16}$$

where the operator $size(\cdot)$ is used calculate the length of column vector and the set of local weights $a_{l,k}$ satisfies $\Sigma a_{l,k} = 1 (l \in \aleph_k)$. The proposed algorithm introduced the combined functions for both tap-length estimation and tap-weight estimation in order to take advantage of the communications among nodes.

3. Steady-state Performance Analysis

Based on weighted spatial-temporal energy conservation arguments [12], we study the steady-state performance of the new algorithm. According to the works [6, 7], the solution of the fractional tap-length would be a fixed value in the steady-state. Therefore, we use the assumption that in the steady-state the estimate $w_{k,\infty}$ with the fixed size $\widetilde{L}$ to simplify the analysis. Firstly, we introduce the stochastic quantities:

$$\boldsymbol{d}_i = \mathrm{col}\{\boldsymbol{d}_1(i),\dots,\boldsymbol{d}_N(i)\},\ \boldsymbol{v}(i) = \mathrm{col}\{\boldsymbol{v}_1(i),\dots,\boldsymbol{v}_N(i)\},\ \boldsymbol{U}_i = \mathrm{diag}\{\boldsymbol{u}_{1,i},\dots,\boldsymbol{u}_{N,i}\} \text{ and } \boldsymbol{\psi}_i = \mathrm{col}\{\boldsymbol{\psi}_{1,i},\dots,\boldsymbol{\psi}_{N,i}\}.$$

Using *(1)*, we obtain

$$\boldsymbol{d}_i = \boldsymbol{U}_i w_c^o + \boldsymbol{v}_i \quad \text{with } w_c^o = Xw^o \text{ and } X = \mathrm{col}\{I_{\widetilde{L}},\dots,I_{\widetilde{L}}\} \text{ of size } N\widetilde{L}\times\widetilde{L} \tag{17}$$

Then, we can obtain a more compact state-space form of w_i :

$$w_i = Q\boldsymbol{\psi}_{i-1} + C\boldsymbol{U}_i^*\left(\boldsymbol{d}_i - \boldsymbol{U}_i Q\boldsymbol{\psi}_{i-1}\right) \text{ with } Q = A\otimes I_{\widetilde{L}}\ \left(N\widetilde{L}\times N\widetilde{L}\right), C = \mathrm{diag}\{\mu_1 I_{\widetilde{L}},\dots,\mu_N I_{\widetilde{L}}\}\ \left(N\widetilde{L}\times N\widetilde{L}\right) \tag{18}$$

where the $N\times N$ diffusion combination matrix A consists of the elements $\{a_{l,k}\}$. The $\otimes$ operator is used to represent Kronecker products. Then, we introduce the weight error vector

$$\widetilde{\boldsymbol{w}}_i = \boldsymbol{w}_c^o - \boldsymbol{w}_i = \left(I_{N\widetilde{L}} - C\boldsymbol{U}_i^*\boldsymbol{U}_i\right)Q\widetilde{\boldsymbol{w}}_{i-1} - C\boldsymbol{U}_i^*\boldsymbol{v}_i \tag{19}$$

When we perform weighted energy balance on both sides and take expectation with the assumption that the regression data are temporally and spatially independent, we get the following variance relation:

$$E\|\widetilde{\boldsymbol{w}}_i\|_{\Sigma}^2 = E\|\widetilde{\boldsymbol{w}}_i\|_{\Sigma'}^2 + E\boldsymbol{v}_i^*\boldsymbol{U}_i C\Sigma C\boldsymbol{U}_i^*\boldsymbol{v}_i \tag{20}$$

$$\Sigma' = Q^*\Sigma Q - Q^*\Sigma C\cdot E\left(\boldsymbol{U}_i^*\boldsymbol{U}_i\right)Q - Q^*E\left(\boldsymbol{U}_i^*\boldsymbol{U}_i\right)C\Sigma Q + Q^*E\left(\boldsymbol{U}_i^*\boldsymbol{U}_i C\Sigma C\boldsymbol{U}_i^*\boldsymbol{U}_i\right)Q \tag{21}$$

For regressors from Gaussian data, we introduce the eigendecomposition $R_u = T\Lambda T^*$ with $\Lambda = \mathrm{diag}\{\Lambda_1,\dots,\Lambda_N\}$ and define the transformed quantities:

$$\overline{\boldsymbol{w}}_i = T^*\widetilde{\boldsymbol{w}}_i,\ \overline{\boldsymbol{U}}_i = \boldsymbol{U}_i T,\ \overline{Q} = TQT^*,\ \overline{\Sigma} = T\Sigma T^* \text{ and } \overline{\Sigma}' = T\Sigma' T^*.$$

Then we can rewrite the variance relations *(20)-(21)* and transform them to

$$E\|\overline{\boldsymbol{w}}_i\|_{\overline{\Sigma}}^2 = E\|\overline{\boldsymbol{w}}_i\|_{\overline{\Sigma}'}^2 + b^T\overline{\sigma} \tag{22}$$

$$\overline{F} = \left(\overline{Q}\oplus\overline{Q}^{*T}\right)\left(I_{N^2\widetilde{L}^2} - \left(I_{N\widetilde{L}}\oplus\Lambda C\right) - \left(\Lambda C\oplus I_{N\widetilde{L}}\right) + (C\oplus C)P\right)\overline{\sigma} \tag{23}$$

where the $\oplus$ operator is used for the block Kronecker product and

$$P = \mathrm{diag}\{P_1,\dots,P_N\}, \overline{\sigma} = \mathrm{bvec}\{\overline{\Sigma}\} \tag{24}$$

with $P_l = \mathrm{diag}\{\Lambda_1 \otimes \Lambda_l, ..., \lambda_l \lambda_l^T + \tau\Lambda_l \otimes \Lambda_l, ..., \Lambda_N \otimes \Lambda_l\}$ where $\tau = 1$ for complex data and $\tau = 2$ for real data. We select the weighting matrix $\overline{\Sigma}$ and define the following spatial filtering matrices

$$S_{q,k} = \mathrm{diag}\{0_{(k-1)\widetilde{L}}, I_{\widetilde{L}}, 0_{(N-k)\widetilde{L}}\} \text{ and } S_{\lambda,k} = \mathrm{diag}\{0_{(k-1)\widetilde{L}}, \Lambda_k, 0_{(N-k)\widetilde{L}}\}$$

where 0_l denotes a $l \times l$ matrix with zero elements. We note that at each node k, the mean-square performance can be expressed as ,

$$\xi_k = E\|\widetilde{\boldsymbol{w}}_{k,\infty}\|^2 = E\|\overline{\boldsymbol{w}}_{k,\infty}\|^2 \qquad (\text{MSD}) \tag{25}$$

$$\zeta_k = E\|\widetilde{\boldsymbol{w}}_{k,\infty}\|^2_{\lambda_k} = E\|\overline{\boldsymbol{w}}_{k,\infty}\|^2_{\lambda_k} \qquad (\text{EMSE}) \tag{26}$$

We exploit the above spatial filtering matrices and obtain the local steady state mean-square performance of node k

$$\xi_k = b^T\left(I - \overline{F}\right)^{-1} \mathrm{bvec}\{S_{q,k}\} \qquad (\text{MSD}) \tag{27}$$

$$\zeta_k = b^T\left(I - \overline{F}\right)^{-1} \mathrm{bvec}\{S_{\lambda,k}\} \qquad (\text{EMSE}) \tag{28}$$

4. Simulation

The simulation results of the distributed algorithm are provided and compared with the theoretical performance. In an $N = 20$ nodes diffusion network with incremental cooperation, we seek an unknown system vector with $L^o = 10$. The regressors at node k are drawn according to the equation $\mu_k(i) = \rho_k u_k(i-1) + b_k c_k(i)$, where $\rho_k \in (0,1]$ and $b_k = \sqrt{\sigma_{u,k}^2\left(1-\rho_k^2\right)}$. A first-order autoregressive (AR) process with a pole at ρ_k is generated by this equation and c_k is a white Gaussian noise with mean $\overline{c}_k = 0$ and variance $\sigma_{c_k}^2 = 1$.

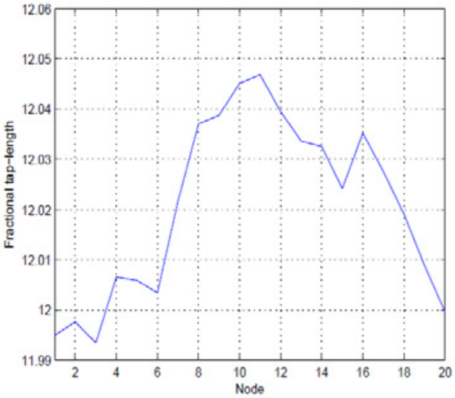

Fig. 1. Fractional tap-length versus nodes

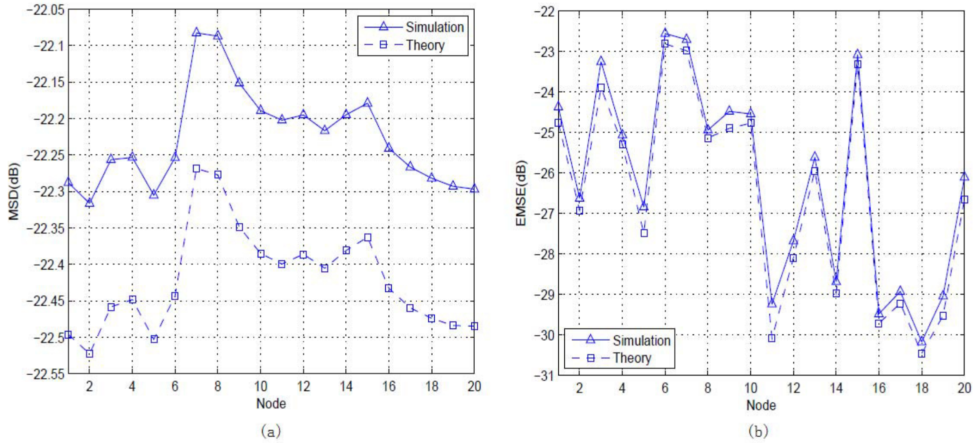

Fig. 2. (a) Steady-state MSD versus nodes (b) Steady-state EMSE versus nodes

The parameters for the algorithm *(16)* are given by: $\gamma = \gamma_k = 1$, $\alpha = \alpha_k = 0.05$, $\Delta = \Delta_k = 3$, $\beta = \beta_k = 1$. In the simulation, we set the step-size of the proposed algorithm as $\mu = \mu_k = 0.02$. The experiment is carried out by 100 independent Monte Carlo runs to obtain the simulated results. The steady-state results are averaged over the last 1000 instantaneous samples of 10000 iterations. Fig.1 illustrates the converged fractional tap-length throughout the network in the steady-state, with a difference of approximately 0.26 at 12.02. Therefore, as it is expected, in the steady-state the proposed algorithm can obtain a good estimate of the tap-length. Fig.2 shows the steady-state performance of the proposed algorithm. The simulation results provide a good match with the theoretical curves.

5. Conclusions

We presented a distributed variable tap-length algorithm in a diffusion network. Our algorithm exploits the fractional tap-length solution, estimation theory and diffusion LMS distributed adaptive algorithm. In this work, we provide the analysis on the performance of the distributed algorithm and the simulation results, which agree well with the theoretical results.

Acknowledgements

This material was based on work supported by the National Science Foundation 61001169.

Corresponding Author:

Leilei Li, lileilei@abp.gov.cn, 13611389100.

References

[1] Mayyas K. Performance analysis of the deficient length LMS adaptive algorithm. IEEE Trans. on Signal Processing 2005; vol.53, no.8:2727-2734.
[2] Pritzker Z, Feuer A. Variable length stochastic gradient algorithm. IEEE Trans. on Signal Processing 1991; vol.39, no.4:977-1001.
[3] Won YK, Park RH, Park JH and Lee BU. Variable LMS algorithm using the time constant concept. IEEE Trans. on Consumer Electron 1994; vol.40, no.3:655-661.
[4] Riera-Palou F, Norsa JM, Cruickshank DGM. Linear equalizers with dynamic and automatic length selection. Electronics Letters 2001; vol.37, no.25:1553-1554.
[5] Gong Y, Cowan CFN. Structure adaptation of linear MMSE adaptive filters. Proc. Inst. Elect. Eng.-Vision, Image, Signal Process. 2004; vol.151, no.4,:271-277.
[6] Gong Y, Cowan CFN. An LMS style variable tap-length algorithm for structure adaptation. IEEE Trans. on Signal Processing 2005; vol.53, no.7: 2400-2407.
[7] Zhang Y, Li N, Chambers J, Sayed AH. Steady state performance analysis of a variable tap-length LMS algorithm. IEEE Tran. Signal Processing 2008; vol.56, no.2:839-845.
[8] Li L, Chambers JA. A novel adaptive leakage factor scheme for enhancement of a variable tap-length learning algorithm. ICASSP 2008: 2837-2840.
[9] Lopes CG, Sayed AH. Incremental adaptive strategies over distributed networks. IEEE Trans. Signal Processing 2007; vol.55, no.8:4064-4077.
[10] Li L, Chambers JA. A new incremental affine projection based adaptive algorithm for distributed networks. Fast Communication in Signal Processing 2008; vol.88, no.10: 2599-2603.
[11] Li L, Zhang Y, Chambers JA. Variable length adaptive filtering within incremental learning algorithms for distributed networks. Asilomar Conf. on Signals, Systems and Computers 2008: 225-229.
[12] Lopes CG, Sayed AH. Diffusion least-mean squares over adaptive networks: Formulation and performance analysis. IEEE Trans. on Signal Processing 2008; vol.56, no.7:3122-3136.
[13] Cattivelli FS, Sayed AH. Diffusion LMS Strategies for distributed estimation. IEEE Trans. on Signal Processing 2010; vol.58, no.3:1035-1048.