Distributed Estimation with Sparsely Accessible Information

1st Mahdi Shamsi
Electrical Engineering Department
Sharif University of Technology
Tehran, Iran
mahdi.shamsi@alum.sharif.edu

2nd Farokh Marvasti
Electrical Engineering Department
Sharif University of Technology
Tehran, Iran
marvasti@sharif.edu

Abstract—This paper addresses sparse observability constraints in Diffusion Least Mean Squares (DLMS) and proposes a framework for analyzing combination strategies. A thresholding-based algorithm is introduced to identify the sparse support vector under incomplete information. The method effectively handles sparse observations in both time and transform domains, achieving a 30–40 dB improvement in Mean Square Deviation (MSD) over conventional DLMS.

Index Terms—Distributed Estimation, DLMS, Sparse Mask, Signal recovery over network.

I. Introduction

Distributed algorithms are widely used in research fields like signal reconstruction on graphs [1], [2], multi-agent reinforcement learning [3], [4], and the Diffusion Least Mean Square (DLMS) algorithm [5]–[9], due to their robustness, reliability, and fast convergence.

Many studies assume nodes have full access to the target vector, allowing independent estimation. Our approach uniquely addresses masked measurements, an area largely unexplored in previous studies. Prior works typically model censored data using step functions that restrict measurements to nonnegative values [10], [11], which can be adjusted by rotating or negating the regressor vector when zero measurements occur. Unlike partial diffusion, where masked measurements are intentionally used to reduce communication or computation [12], [13], our method treats them as inherent environmental constraints.

This paper focuses on scenarios where nodes have limited visibility of the target and must collaboratively estimate it. We propose a framework inspired by signal flow analysis [14] and introduce a thresholding-based algorithm to recover missing components, leveraging prior work on sparsity-based recovery [15], [16].

Our proposed algorithm enables efficient information sharing and target support extraction while controlling data flow and estimation combinations. Simulations show it improves Mean Square Deviation (MSD) by 30–40 dB over conventional DLMS, achieving near full-observability performance at a given observability ratio.

Contributions of this paper include:

- Introducing the problem of partial observability,
- Mathematical analysis for the optimal combination,
- Introducing a method for estimation diffusion without prior support knowledge.

Section II illustrates the PO scenario, while Sections III and IV discuss optimal combination weighting strategies. Simulation results are presented in Section VI, and the paper concludes in the final section.

II. LMS Diffusion with Partial Observations

The primary algorithm used in this study is DLMS, known for its strong performance [17]. At node i, the local estimation follows three steps:

$$g(\vec{\omega}_i(t), \Theta_i(t)) = \begin{cases} 1. & s_i(t) = \vec{\mathrm{a}}_{i,t}^{\dagger} M_i \vec{\omega}^{\mathrm{opt}} + \nu_i(t) \\ 2. & err_i(t) = s_i(t) - \vec{\mathrm{a}}_{i,t}^{\dagger} \vec{\omega}_i(t-1) \\ 3. & \vec{\omega}_i(t) = \vec{\omega}_i(t-1) + \mu . err_i(t) \vec{\mathrm{a}}_{i,t} \end{cases},$$

where $\Theta_i(t)$ includes the measurement vector $\vec{\mathrm{a}}_{i,t}$ and adaptation rate μ, while $\nu_i(t)$ represents measurement noise. The transpose operator is denoted by $\dagger$.

In the combination step, DLMS typically assumes full access to $\vec{\omega}^{\mathrm{opt}}$, but real-world scenarios often involve partially observable targets, where nodes rely on limited measurements. This observability is modeled using the mask operator $M_i = \mathcal{T}^{\dagger} D_i \mathcal{T}$, where the diagonal elements of D_i indicate component observability in the transform domain. Fig. 1 illustrates this pattern.

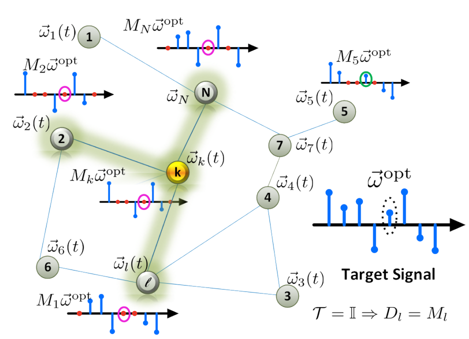


Fig. 1: Illustration of masked data access and cooperative estimation in a network (time domain masking: $\mathcal{T} = \mathbb{I}$).

Given the estimation of node i as $\vec{\omega}_i = M_i \vec{\omega}^{\mathrm{opt}} + \vec{e}_i$, where $\vec{e}_i$ represents the estimation error at an appropriate time t (omitted for brevity), the goal is to combine the estimations of its neighboring nodes (nodes in $\aleph_i$) to minimize the error. The resulting estimation is $\vec{\psi}_i = \sum_{k\in\aleph_i} G_{k,i}\vec{\omega}_k$, where $G_{k,i}$ is the weighting matrix indicating the impact of $\vec{\omega}_k$ on the i^{th} node. The error is defined as:

$$J(i) \triangleq \mathbb{E}\{\|\vec{\psi}_i - \vec{\omega}^{\text{opt}}\|_2^2\}, \quad (1)$$

and the objective is to determine the optimal weights that minimize this error.

Throughout this proof procedure, two assumptions are taken into consideration (independence assumption):
1- The estimation error, $\vec{e}_i$, adheres to the orthogonality principle, with zero mean components and variance σ_i^2.
2- Each target component l satisfies $\mathbb{E}\{(\omega_l^{\text{opt}})^2\} = \sigma_0^2(l)$.

Given the diagonal structure of $M_k = \mathcal{T}^\dagger \text{diag}(\vec{d}_k)\mathcal{T}$, the weighting matrices, $G_{k,i}$, are also assumed to be of the form $G_{k,i} = \mathcal{T}^\dagger \text{diag}(\vec{g}_{k,i})\mathcal{T}$, with the l^{th} component ($g_{k,i}(l)$) determining the impact of the corresponding component of $\vec{\omega}_k$ in the transform domain ($\mathcal{T}$)[1]. As a result, the error can be expressed as:

$$\begin{aligned} J_i &= \mathbb{E}\{\| \sum_{k\in\aleph_i} \mathcal{T}^\dagger \text{diag}(\vec{g}_{k,i})\mathcal{T}.(M_k\vec{\omega}^{\text{opt}} + \vec{e}_k) - \vec{\omega}^{\text{opt}}\|_2^2\} \\ &= \mathbb{E}\{\|[(\sum_{k\in\aleph_i} \text{diag}(\vec{g}_{k,i})D_k) - \mathbb{I}]\vec{\Omega}^{\text{opt}}\|_2^2 + \|\text{diag}(\vec{g}_{k,i})\vec{E}_k\|_2^2\}, \end{aligned}$$

where, $\vec{\Omega}^{\text{opt}}$ and $\vec{E}_k$ represent transformed versions of $\vec{\omega}^{\text{opt}}$ and $\vec{e}_k$, respectively.

Consequently, the error can be segmented into two distinct components: estimation error ($J^{\text{est}}(i)$) and combination error ($J^{\text{comb}}(i)$), which can be expressed as follows:

$$J_i^{\text{est}} \triangleq \sum_{j=1}^{L}\sum_{k\in\aleph_i} (g_{k,i}(j))^2\, \mathbb{E}\{(E_k(j))^2\} = \sum_{j=1}^{L}\sum_{k\in\aleph_i} (g_{k,i}(j))^2\sigma_k^2,$$

$$\begin{aligned} J_i^{\text{comb}} &\triangleq \sum_{j=1}^{L}(\sum_{k\in\aleph_i} g_{k,i}(j)d_k(j) - 1)^2.\,\mathbb{E}\{(\Omega_o(j))^2\} \\ &= \sum_{j=1}^{L}(\sum_{k\in\aleph_i} g_{k,i}(j)d_k(j) - 1)^2\sigma_0^2(j). \end{aligned}$$

It is evident that the minimization problem can be broken down into L separate subproblems, each of which can be independently resolved.

III. Optimal Combination Weighting

Consider a component j. If $d_k(j) = 0$, node $k \in \aleph_i$ does not contribute in the combination step, preventing error reduction. If $d_k(j) \neq 0$, the error increases by $(g_{k,i}(j))^2\sigma_k^2$, making it reasonable to set $g_{k,i}(j) = 0$. When no neighboring nodes estimate the l^{th} component, the error equals σ_0^2.

For brevity, we redefine $d_k(j)$, $g_{k,i}(j)$, and $\sigma_0(j)$ as d_k, c_k, and σ_0, respectively, and assume $k \in [1, N]$. The cost function is then:

$$J = (\sum_{k=1}^{N} c_k d_k - 1)^2.\sigma_0^2 + \sum_{k=1}^{N} c_k^2\sigma_k^2. \quad (2)$$

By taking the derivative of the cost function J with respect to c_l, and equating it to zero, we obtain:

$$\frac{\partial J}{\partial c_l} = 2d_k(\sum_{k=1}^{N} c_k d_k - 1).\sigma_0^2 + 2c_l\sigma_l^2$$
$$\xrightarrow{\frac{\partial J}{\partial c_l}=0} c_l = \frac{1 - \sum_{k\neq l} c_k d_k}{d_l^2 + \lambda_l^{-2}} \times d_l,$$

where $\lambda_l = \frac{\sigma_0}{\sigma_l}$ represents the SNR for the given index l. Notably, if $d_l = 0$, then $c_l = 0$. In the special case where $\sigma_k = \sigma$ and $d_k \in \{0, 1\}$, all coefficients take a uniform value $c_k = c = \frac{1}{m+\lambda^{-2}}$, where $\lambda^2 = \frac{\sigma_0^2}{\sigma^2}$ and m is the number of nonzero d_k values.

The optimal weights can be obtained by solving for the column vector $\vec{c}$, where m non-zero d_i's are arranged from 1 to m, using the following matrix equation:

$$\begin{bmatrix} d_1^2 + \lambda_1^{-2} & d_1.d_2 & \cdots & d_1.d_m \\ d_2.d_1 & d_2^2 + \lambda_2^{-2} & \cdots & d_2.d_m \\ \vdots & \vdots & \ddots & \vdots \\ d_m.d_1 & d_m.d_2 & \cdots & d_m^2 + \lambda_m^{-2} \end{bmatrix} \vec{c} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{bmatrix}$$

In order to solve the problem, the coefficient matrix is decomposed as the sum of a diagonal matrix (Λ^{-2}) and a rank-1 matrix ($B = \vec{d}.\vec{d}^\dagger$), defined as follows:

$$\Lambda \triangleq \text{diag}([\lambda_1, \ldots, \lambda_m]);\ \vec{d}^\dagger \triangleq \begin{bmatrix} d_1 & d_2 & \cdots & d_m \end{bmatrix}.$$

This transforms the problem to the form $(\Lambda^{-2} + B)\vec{c} = \vec{d}$. The Sherman-Morrison formula can then be used to compute the inverse of $(\Lambda^{-2} + B)$ as follows:

$$(\Lambda^{-2} + B)^{-1} = \Lambda^2 - \frac{\Lambda^2\vec{d}.\vec{d}^\dagger\Lambda^2}{1 + \vec{d}^\dagger\Lambda^2\vec{d}}.$$

This yields a solution for the vector $\vec{c}$:

$$\begin{aligned} \vec{c} &= \Big(\Lambda^2 - \frac{\Lambda^2\vec{d}.\vec{d}^\dagger\Lambda^2}{1 + \vec{d}^\dagger\Lambda^2\vec{d}}\Big)\vec{d} = \Lambda^2\vec{d} \times \Big(1 - \frac{\vec{d}^\dagger\Lambda^2\vec{d}}{1 + \vec{d}^\dagger\Lambda^2\vec{d}}\Big) \\ &= \frac{1}{1 + \vec{d}^\dagger\Lambda^2\vec{d}}\Lambda^2\vec{d}, \end{aligned} \quad (3)$$

with each element computed as[2]:

$$c_l = \frac{d_l\lambda_l^2}{1 + \sum_{k=1}^{m} d_k^2\lambda_k^2}, \quad (4)$$

where l is the index of a typical node.

This outcome highlights how each node's observation of target signal components and its estimation error affect the combination step. When all components have equal

[1] For clarity, we use the following notation throughout the manuscript: $\mathcal{T}$ represents the transform matrix, t denotes the time index, T refers to the simulation time, and $\mathbb{T}$ represents the thresholding level. Each of these terms is defined accordingly in the manuscript.

[2] We will refer to this group of coefficients as Least Squares Solution (LSS) in case of comparison.

observability ($d_k = d_0$), the following equation holds: $c_l = (d_0\lambda_l^2)/(1+d_0^2\sum_{k=1}^{m}\lambda_k^2)$. An increase in relative SNR raises the relative weight among neighboring nodes and necessitates attenuation of the observation.

In fully observable cases, $G_{k,i}$ reduces to a scalar weight $g_{k,i}$, simplifying the objective function to:

$$\begin{aligned} J(i) &= \mathbb{E}\{\|\sum_{k\in\aleph_i} g_{k,i}(M_i\vec{\omega}^{\text{opt}}+\vec{e}_i)-\vec{\omega}^{\text{opt}}\|_2^2\} \\ &= \mathbb{E}\{\|\sum_{k\in\aleph_i} g_{k,i}\vec{E}_i\|_2^2\} + \mathbb{E}\{\|(\sum_{k\in\aleph_i} g_{k,i}D_k-\mathbb{I})\vec{\Omega}^{\text{opt}}\|_2^2\}. \end{aligned}$$

Thus, the objective function can be expressed as:

$$J(i) = \sum_{k\in\aleph_i} g_{k,i}L\sigma_k^2 + \sum_{j=1}^{L}(\sum_{k\in\aleph_i} g_{k,i}d_k(j)-1)^2\sigma_0^2.$$

To simplify the notation, the index i can be eliminated, resulting in a linear system of equations:

$$(\tilde{\Lambda}+\sum_{j=1}^{L}\vec{d}_j\vec{d}_j^{\,\dagger})\vec{g} = \sum_{j=1}^{L}\vec{d}_j,$$

where $\tilde{\Lambda} \triangleq L.\text{diag}([\lambda_1^{-2},\ldots,\lambda_m^{-2}])$, $\vec{d}_j \triangleq [d_1(j),\ldots,d_m(j)]^\dagger$, and $\vec{g} \triangleq [g_{1,i},\ldots,g_{m,i}]^\dagger$. By using the Sherman-Morrison formula, the solution $S=(\tilde{\Lambda}+\sum_{j=1}^{L}\vec{d}_j\vec{d}_j^{\,\dagger})^{-1}$ can be recursively found as:

$$S_r = S_{r-1} + \frac{S_{r-1}\vec{d}_k.\vec{d}_k^{\,\dagger}S_{r-1}}{1+\vec{d}_k^{\,\dagger}S_{r-1}\vec{d}_k};\ S_0=\tilde{\Lambda}^{-1}, S=S_L. \quad (5)$$

The matrix $\tilde{D}=\sum_{j=1}^{L}\vec{d}_j\vec{d}_j^{\,\dagger}$ encapsulates the cooperation between nodes x and y as neighbors of node i, with elements $\tilde{D}_{x,y}$ representing their interaction via the inner product $\langle\tilde{d}_x,\tilde{d}_y\rangle$. Here, $\tilde{d}_a = [d_a(1),d_a(2),\ldots,d_a(L)]^\dagger$ for $a\in\aleph_i$. Additionally, each g_l is influenced by its aggregate contribution $\sum_{j=1}^{L}d_l(j)$ across the combined vector components.

IV. Unbiased Estimation

Regarding the combination weights derived in (3), the resulting combination is biased, with the bias given by:

$$\begin{aligned} \text{bias} &= \mathbb{E}\left\{\sum_{k\in\aleph_i}\mathcal{T}^\dagger\text{diag}(\vec{g}_{k,i})\mathcal{T}\cdot(M_k\vec{\omega}^{\text{opt}}+\vec{e}_k)-\vec{\omega}^{\text{opt}}\right\} \\ &= \mathcal{T}^\dagger\text{diag}([b_1,\ldots,b_L])\mathcal{T}\vec{\omega}^{\text{opt}}. \end{aligned} \quad (6)$$

By following a similar procedure as in (3), we consider a generic component j, yielding:

$$b_j \triangleq \sum_{k\in\aleph_i} g_{k,i}(j)d_k(j)-1 = \frac{1}{1+\sum_{l\in\aleph_i}[d_l(j)\lambda_l(j)]^2}. \quad (7)$$

To achieve an unbiased combination, the optimization in (1) is constrained by enforcing the unbiasedness condition, $\mathbb{E}\{\vec{\psi}_i\}=\vec{\omega}^{\text{opt}}$, which leads to the following problem:

$$\min J(i) \quad \text{s.t.} \quad \mathbb{E}\{\vec{\psi}_i\}=\vec{\omega}^{\text{opt}}. \quad (8)$$

For a generic component j, the optimization is formulated using a Lagrange multiplier by constraining (2) with $\sum_{k=1}^{N}c_kd_k=1$:

$$R(\vec{c}) = \sum_{k=1}^{N}c_k^2\sigma_k^2 - \gamma\left(\sum_{k=1}^{N}c_kd_k-1\right). \quad (9)$$

Taking the derivative of the cost function R with respect to c_l and setting it to zero gives:

$$\frac{\partial R(\vec{c})}{\partial c_l} = 2c_l\sigma_l^2-\gamma d_l \implies c_l=\frac{\gamma d_l}{2\sigma_l^2}.$$

Applying the constraint results in:

$$\sum_{k=1}^{N}\frac{\gamma d_k}{2\sigma_k^2}d_k = 1 \implies \gamma=\frac{2}{\sum_{k=1}^{N}\frac{d_k^2}{\sigma_k^2}}, \quad (10)$$

leading to the final unbiased coefficients[3]:

$$c_l = \frac{d_l}{\sigma_l^2\sum_{k=1}^{N}\frac{d_k^2}{\sigma_k^2}}. \quad (11)$$

Substituting (11) into (2), the combination error is expressed as:

$$R(\vec{c}) = \sum_{l=1}^{N}\left(\frac{d_l}{\sigma_l^2\sum_{k=1}^{N}\frac{d_k^2}{\sigma_k^2}}\right)^2\sigma_l^2 = \frac{1}{\sum_{k=1}^{N}\frac{d_k^2}{\sigma_k^2}}. \quad (12)$$

Under identical performance and statistically similar conditions for all nodes $l\in\aleph_i$, we assume $\sigma=\sigma_l$, simplifying the coefficients to $c_l=\frac{d_l}{\sum_{k=1}^{N}d_k^2}$. For a binary mask representing node l, this reduces to $c_l=\frac{d_l}{m}$, where m is the number of nonzero d_k values, effectively averaging over available estimations.

With a locally converging estimator, σ_l^2 tends to decrease over time, leading the least squares solution in (4) to eventually align with (11). In this case, for each SNR term, we have $\lambda_l\gg1$, implying:

$$\begin{aligned} c_l^{\text{LSS}} &= \frac{d_l\lambda_l^2}{1+\sum_{k=1}^{m}d_k^2\lambda_k^2} \approx \frac{d_l\lambda_l^2}{\sum_{k=1}^{m}d_k^2\lambda_k^2} \\ &= \frac{d_l}{\sigma_l^2\sum_{k=1}^{N}\frac{d_k^2}{\sigma_k^2}} = c_l^{\text{unbiased}}. \end{aligned} \quad (13)$$

V. No Information on Observability

In practical settings, observability vector information is often unavailable, making support and values unknown. Assuming trivial attenuation ($D_i(j)\in\{0,1\}$), knowing the support vector is crucial. To address this, we use a thresholding approach inspired by sparse signal recovery. As each estimator approaches its local target $M_i\vec{\omega}^{\text{opt}}$, we estimate support using component magnitudes in the transform domain. Defining the threshold level as $\beta(t)=\beta_1e^{-t/\tau}+\beta_0$, components with magnitudes above $\beta(t)$ are considered observed values of $\vec{\omega}^{\text{opt}}$, enabling information diffusion across the network.

Each node shares its estimated components in 3 steps:

[3] For clarity, this set of coefficients will be referred to as the unbiased case for comparison purposes.

Algorithm 1 Threshold-based Diffusion LMS.

for $t = 1, \dots, T$ do
  1- Adaptation, $\vec{\Omega}_i = g(\vec{\omega}_i, \Theta_i)$
  2- Thresholding, $\tilde{D}_i = \Xi(\vec{\Omega}_i, \beta)$
  3- Combination, $\vec{\psi}_i, \vec{\omega}_i = h(\vec{\psi}_j, \tilde{\vec{\Omega}}_j, \tilde{D}_j | j \in \aleph_i)$
end for

The function $h(\cdot)$ consists of three operations:

$$\begin{cases} \text{Locally update: } \vec{\psi}_i \leftarrow D_i((1-\eta)\vec{\psi}_i + \eta\vec{\Omega}_i) + (\mathbb{I} - D_i)\vec{\psi}_i \\ \text{Diffuse: } \vec{\psi}_i \leftarrow \sum_{j \in \aleph_i} D_{\text{comb},j} \vec{\psi}_j \\ \text{Combine: } \vec{\omega}_i \leftarrow \mathcal{T}^{\dagger}[D_i((1-\alpha)\vec{\psi}_i + \alpha\vec{\Omega}_i) + (\mathbb{I} - D_i)\vec{\Omega}_i] \end{cases}$$

Here, $D_{\text{comb},j}$ defines a combination strategy, averaging active components across neighboring nodes when observability power and noise variance are uniform.

The parameters α and η control information flow: α regulates inflow from the network into a node's estimation and η manages outflow from each node into the network.

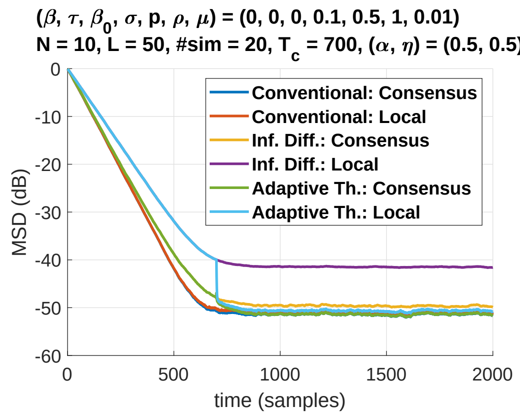
VI. Simulation Results and Discussion

In this section, MSD serves as the performance measure for simulation output, reported at both node ("local") and network ("consensus") levels. The local target for node i is $M_i\vec{\omega}^{\text{opt}}$, while the network target is $\vec{\omega}^{\text{opt}}$ for all nodes.

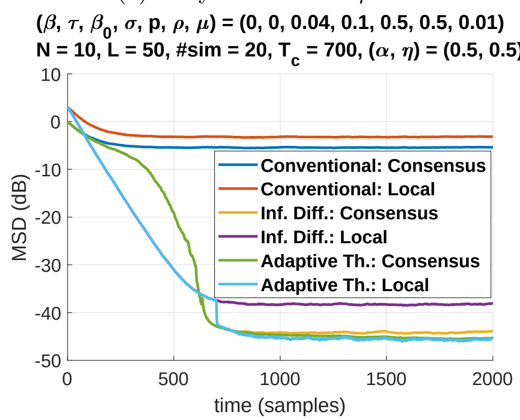
For fair comparisons, local MSD is normalized by $\frac{L}{\sum_j D_i(j)}$, where L is the length of $\vec{\omega}^{\text{opt}}$ and $D_i(j)$ follows probability ρ. The combination strategy is averaging. Each link among N nodes is generated with probability p, and changes in η and α follow step functions at $t = T_c < T$. Measurements $\vec{a}_{i,t}$ are normally distributed, with noise $\nu_i \sim \mathcal{N}(0, \sigma^2)$. Results are averaged over multiple simulations.

We examine two partial observation scenarios: time-domain observation ($\mathcal{T} = \mathbb{I}$, Fig. 2) and DCT-domain observation (Fig. 3). The results show that tuning α and η regulates network-wide information flow (Section V), ensuring robustness under various noise conditions. Fig. 2 highlights the shortcomings of conventional combination strategies, where ignoring observability disrupts consensus and local convergence (Section II, Eq. 5). Our thresholding method ("Inf. Diff.") enhances diffusion without relying on local data, while the adaptive approach ("Adaptive Th.") selectively improves estimation.

Additionally, Fig. 3 demonstrates the proposed method's effectiveness in handling DCT-domain partial observations across noise levels. For low noise ($\sigma = 0.1$), parameters $(\beta_0, \alpha, \eta, \mu)$ were set to $(700, 0.5, 0.5, 0.04)$, and for very low noise ($\sigma = 0.01$), they were adjusted to $(1000, 0.7, 0.4, 0.01)$.



(a) Fully obseravable: $\rho = 1$.



(b) Time PO: $\rho = 0.5$.

Fig. 2: The DLMS with Time-Domain PO.

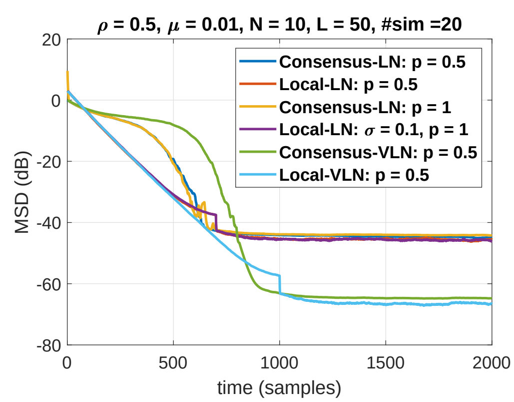


Fig. 3: The DLMS with DCT-Domain PO.

VII. Conclusion

We introduced a framework for analyzing combination strategies in Diffusion LMS with partial observations, including unknown support scenarios. The optimal weighting was derived, converging to the unbiased constrained solution. We proposed a thresholding-based algorithm to enhance information flow and estimation accuracy without prior mask knowledge. Simulations confirmed its robustness under various noise conditions, effectively handling partial observability in time and transform domains. The results highlight its advantages over conventional DLMS and its potential for practical applications.

References

[1] Mengke Lian, Zhenyuan Guo, Xiaoxuan Wang, Shiping Wen, and Tingwen Huang, "Distributed algorithms for linear equations over general directed networks," IEEE Transactions on Neural Networks and Learning Systems, pp. 1–9, 2024.

[2] Jhon A. Castro-Correa, Jhony H. Giraldo, Mohsen Badiey, and Fragkiskos D. Malliaros, "Gegenbauer graph neural networks for time-varying signal reconstruction," IEEE Transactions on Neural Networks and Learning Systems, vol. 35, no. 9, pp. 11734–11745, 2024.

[3] Guangzheng Hu, Yuanheng Zhu, Dongbin Zhao, Mengchen Zhao, and Jianye Hao, "Event-triggered communication network with limited-bandwidth constraint for multi-agent reinforcement learning," IEEE Transactions on Neural Networks and Learning Systems, vol. 34, no. 8, pp. 3966–3978, 2023.

[4] Rafael Pina, Varuna De Silva, Joosep Hook, and Ahmet Kondoz, "Residual q-networks for value function factorizing in multi-agent reinforcement learning," IEEE Transactions on Neural Networks and Learning Systems, vol. 35, no. 2, pp. 1534–1544, 2024.

[5] Jie Chen, Cédric Richard, and Ali H Sayed, "Diffusion lms over multitask networks," IEEE Transactions on Signal Processing, vol. 63, no. 11, pp. 2733–2748, 2015.

[6] Eweda Eweda, Jose CM Bermudez, and Neil J Bershad, "Analysis of a diffusion lms algorithm with probing delays for cyclostationary white gaussian and non-gaussian inputs," Signal Processing, p. 109428, 2024.

[7] Han-Sol Lee, Changgyun Jin, Chanwoo Shin, and Seong-Eun Kim, "Sparse diffusion least mean-square algorithm with hard thresholding over networks," Mathematics, vol. 11, no. 22, pp. 4638, 2023.

[8] Sheng Zhang and Hing Cheung So, "Diffusion average-estimate bias-compensated lms algorithms over adaptive networks using noisy measurements," IEEE Transactions on Signal Processing, vol. 68, pp. 4643–4655, 2020.

[9] Allan E. Feitosa, Vítor H. Nascimento, and Cássio G. Lopes, "Favor the tortoise over the hare: An efficient detection algorithm for cooperative networks," IEEE Transactions on Signal Processing, vol. 72, pp. 3153–3170, 2024.

[10] Zhaoting Liu, Chunguang Li, and Yiguang Liu, "Distributed censored regression over networks," IEEE Transactions on Signal Processing, vol. 63, no. 20, pp. 5437–5449, 2015.

[11] Zhaoting Liu and Chunguang Li, "Censored regression with noisy input," IEEE Transactions on Signal Processing, vol. 63, no. 19, pp. 5071–5082, 2015.

[12] Reza Arablouei, Stefan Werner, Yih-Fang Huang, and Kutluyıl Doğançay, "Distributed least mean-square estimation with partial diffusion," IEEE Transactions on Signal Processing, vol. 62, no. 2, pp. 472–484, 2013.

[13] Vahid Vahidpour, Amir Rastegarnia, Azam Khalili, Wael M Bazzi, and Saeid Sanei, "Analysis of partial diffusion lms for adaptive estimation over networks with noisy links," IEEE Transactions on Network Science and Engineering, vol. 5, no. 2, pp. 101–112, 2017.

[14] Mahdi Shamsi, Alireza Moslemi Haghighi, Nasim Bagheri, and Farrokh Marvasti, "A flexible approach to interference cancellation in distributed sensor networks," IEEE Communications Letters, vol. 25, no. 6, pp. 1853–1856, 2021.

[15] Mahdi Shamsi, Mahmoud Ghandi, and Farokh Marvasti, "A nonlinear acceleration method for iterative algorithms," Signal Processing, vol. 168, pp. 107346, 2020.

[16] Farokh Marvasti, Arash Amini, Farzan Haddadi, Mahdi Soltanolkotabi, Babak Hossein Khalaj, Akram Aldroubi, Saeid Sanei, and Janathon Chambers, "A unified approach to sparse signal processing," EURASIP journal on advances in signal processing, vol. 2012, no. 1, pp. 1–45, 2012.

[17] Fei Hua, Roula Nassif, Cédric Richard, Haiyan Wang, and Ali H Sayed, "Diffusion lms with communication delays: Stability and performance analysis," IEEE Signal Processing Letters, vol. 27, pp. 730–734, 2020.