Sparse Mask Retrieval for Distributed Estimation in Diffusion LMS

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 explores a thresholding-based algorithm for Diffusion LMS (DLMS) under limited observability. We analyze estimator convergence in mean and energy, deriving an optimal thresholding strategy. The method effectively handles sparse observations in both time and transform domains. Simulations validate our error analysis and highlight the benefits of a cooperative approach, showing a 10–15 dB improvement in Mean Square Deviation (MSD).

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

I. Introduction

Diffusion Least Mean Square (DLMS) enhances distributed estimation by enabling nodes to collaborate through information exchange and iterative refinement [1]–[5]. Extensions, such as Bayesian-learning-based DLMS, improve adaptability in nonstationary and noisy environments [6].

This paper examines scenarios where nodes have limited target visibility and must cooperatively estimate signals. Inspired by [7], a thresholding-based approach was proposed for support vector identification to enhance estimation under incomplete information [8]. Prior work has explored sparse estimation, partial diffusion strategies, and robust weighting to mitigate communication overhead and noise interference [9]–[12].

Further, compressive diffusion, frequency-domain methods, and adaptive censoring improve estimation efficiency under missing data and censored measurements [13]–[18]. Graph signal processing extends these techniques to structured data, addressing partial observability via smoothness and topology learning [19]–[21].

Our approach tackles error behavior of a network of estimators with masked measurements, an unexplored topic in the previous study [8]. We analyze estimator convergence and demonstrate how estimation error variance can be predicted based on the estimator's parameters and measurement noise variance. This provides an efficient method for anticipating error behavior and adjusting the threshold level to recover each node's masks. Simulation results validate our analysis, confirming the accuracy of our approach in predicting the mean square deviation (MSD) behavior of distributed estimators.

The structure of the article is as follows:
- Section II investigates the DLMS algorithm with partial observations.
- Section III analyzes the convergence behavior in terms of mean and energy for each estimator.
- Section IV discusses the requirements for thresholding and mask retrieval.
- Section V presents an approach to determine the optimal thresholding level by calculating the error probability of mask estimation.
- Section VI provides a detailed evaluation of the proposed approach and demonstrates its performance in two combination scenarios with sparse observations in time and transform domains.
- Finally, Section VII concludes the article.

II. The DLMS Algorith with Masked Measurements

The primary algorithm in this study is DLMS with masked measurements [8]. At node i, local estimation follows three steps:

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

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

In the combination step, DLMS typically assumes full access to $\vec{\omega}^{\text{opt}}$, but real-world scenarios involve partial observability, modeled using the mask operator $M_i = \mathcal{T}^{\dagger} D_i \mathcal{T}$, where diagonal elements of D_i indicate observability in the transform domain ($\mathcal{T}$). Fig. 1 illustrates this pattern.

For node i, the estimation is given by $\vec{\omega}_i = M_i\vec{\omega}^{\text{opt}} + \vec{e}_i$, where $\vec{e}_i$ is the estimation error. The goal is to combine estimates from neighboring nodes $\aleph_i$:

$$\vec{\psi}_i = \sum_{k\in\aleph_i} G_{k,i}\vec{\omega}_k, \tag{1}$$

to minimize the error, where $G_{k,i}$ is the weighting matrix. The estimation error is defined as:

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

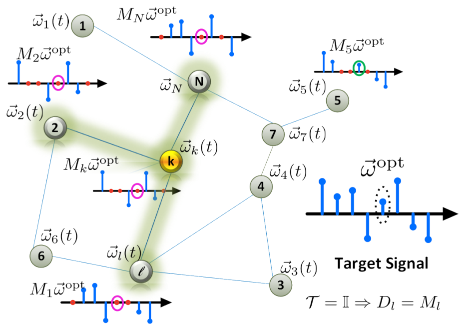

Fig. 1: Masked data access and cooperative estimation in a network (time-domain masking: $\mathcal{T} = \mathbb{I}$).

and the objective is to find optimal weights that minimize it. For each neighboring node $l \in \aleph_i$, the Least Squares Solution (LSS) converges to the unbiased solution [8]:

$$c_l^{\text{LSS}} = \frac{d_l \lambda_l^2}{1 + \sum_{k=1}^{m} d_k^2 \lambda_k^2} \approx \frac{d_l}{\sigma_l^2 \sum_{k=1}^{N} \frac{d_k^2}{\sigma_k^2}} = c_l^{\text{unbiased}}, \quad (3)$$

where d_l is the masking component, λ_l is the estimation SNR, and σ_l^2 is the estimation error variance.

It should be noted that, in practice, observability information is often unknown, requiring support identification. Assuming binary attenuation ($D_i(j) \in \{0, 1\}$), thresholding methods can retrieve the masks.

III. Error Behavior: Convergence Analysis

To justify the credibility of the thresholding-based approach, we analyze the convergence of each local estimator and investigate its behavior in both transient and steady-state modes.

We begin by studying the mean convergence of the estimators' algorithm. For simplicity, the node index is omitted. The measured data are assumed to follow a regressor model:

$$s(t) = \vec{\mathrm{a}}_t^\dagger M \vec{\omega}^{\text{opt}} + \nu(t). \quad (4)$$

We define the estimation error as $\vec{e}(t) \triangleq \vec{\omega}(t) - M\vec{\omega}^{\text{opt}}$. Accordingly, the estimation error can be calculated as:

$$\begin{aligned} &\vec{\omega}(t) = \vec{\omega}(t-1) + \mu(s(t) - \vec{\mathrm{a}}_t^\dagger \vec{\omega}(t-1))\vec{\mathrm{a}}_t \\ \xRightarrow{-M\vec{\omega}^{\text{opt}}} \quad &\vec{e}(t) = \vec{e}(t-1) - \mu[\vec{\mathrm{a}}_t^\dagger \vec{e}(t-1) - \nu(t)]\vec{\mathrm{a}}_t. \end{aligned} \quad (5)$$

Taking the expectation of both sides, we define $\vec{\epsilon}(t) \triangleq \mathbb{E}\{\vec{e}(t)\}$ and let $R_a = \mathbb{E}\{\vec{\mathrm{a}}_t \vec{\mathrm{a}}_t^\dagger\}$. This leads to:

$$\begin{aligned} \vec{\epsilon}(t) &= \mathbb{E}\{\vec{e}(t-1)\} - \mu\, \mathbb{E}\{\vec{\mathrm{a}}_t[\vec{\mathrm{a}}_t^\dagger \vec{e}(t-1) - \nu(t)]\} \\ \vec{\epsilon}(t) &= \vec{\epsilon}(t-1) - \mu\, \mathbb{E}\{\vec{\mathrm{a}}_t \vec{\mathrm{a}}_t^\dagger\}\, \mathbb{E}\{\vec{e}(t-1)\} - \mathbb{E}\{\vec{\mathrm{a}}_t\}\, \mathbb{E}\{\nu(t)\} \\ &= \vec{\epsilon}(t-1) - \mu R_a \vec{\epsilon}(t-1) = (I - \mu R_a)^t \vec{\epsilon}(0). \end{aligned} \quad (6)$$

Typically, by finding the maximum eigenvalue of R_a, denoted by $\max \operatorname{eig}(R_a)$, the condition $\mu < \frac{2}{\max \operatorname{eig}(R_a)}$ ensures the convergence of (6) as $t \to +\infty$.

It is worth mentioning that, as (6) guarantees convergence in the mean for unmasked components, it also confirms that the masked ones converge to zero, ensuring they do not contribute randomness. By defining the expected error in the transform domain $\vec{\varepsilon}(t) \triangleq \mathcal{T}\vec{\epsilon}(t)$, we can verify the behavior by replacing the mask with its transform domain counterpart $M = \mathcal{T} D \mathcal{T}^\dagger$:

$$\vec{\varepsilon}(t) = \mathcal{T}\vec{\epsilon}(t) = \mathcal{T}\, \mathbb{E}\{\vec{\omega}(t) - \mathcal{T} D \mathcal{T}^\dagger \vec{\omega}^{\text{opt}}\} = \mathbb{E}\{\vec{\Omega}(t) - D\vec{\Omega}^{\text{opt}}\}.$$

To facilitate further analysis, we focus on normally distributed regression vectors, with each component independently and identically distributed (i.i.d.) from a zero-mean normal distribution with variance σ_a^2. Thus, we have $R_a = \sigma_a^2 I$, which results in:

$$\vec{\epsilon}(t) = (1 - \mu\sigma_a^2)^t \vec{\epsilon}(0) \xRightarrow{\mathcal{T} \times \cdots} \vec{\varepsilon}(t) = (1 - \mu\sigma_a^2)^t \vec{\varepsilon}(0). \quad (7)$$

The condition for convergence in the mean becomes:

$$|1 - \mu\sigma_a^2| < 1 \quad \Rightarrow \quad \mu < \frac{2}{\sigma_a^2}. \quad (8)$$

On the other hand, we are also interested in the energy of the estimation error. Hence, it might be beneficial to find the expected energy of $\vec{\epsilon}(t)$ which also follows the Parseval's relation with its transform domain version:

$$\xi(t) \triangleq \mathbb{E}\{\|\vec{e}(t)\|^2\} = \mathbb{E}\{\|\vec{\varepsilon}(t)\|^2\}. \quad (9)$$

Then, multiplying each side of (5) by itself and taking the expectation results in:

$$\begin{aligned} \xi(t) &= \mathbb{E}\{\vec{e}(t)^\dagger \vec{e}(t)\} \\ &= \mathbb{E}\{\big(\vec{e}(t-1) - \mu[\vec{\mathrm{a}}_t^\dagger \vec{e}(t-1) - \nu(t)]\vec{\mathrm{a}}_t\big)^\dagger \\ &\qquad \big(\vec{e}(t-1) - \mu[\vec{\mathrm{a}}_t^\dagger \vec{e}(t-1) - \nu(t)]\vec{\mathrm{a}}_t\big)\}. \end{aligned} \quad (10)$$

Considering two terms, the measurement noise ($\nu(t)$) and estimation error ($\vec{e}(t)$), we face three components contributing to $\xi(t)$:

1) Cross-Effect:

$$\begin{aligned} \text{Cross term} &= \mathbb{E}\{\mu\nu(t)(\vec{e}(t-1) - \mu\vec{\mathrm{a}}_t^\dagger \vec{e}(t-1)\vec{\mathrm{a}}_t)^\dagger \vec{\mathrm{a}}_t\} \\ &= \mu\, \mathbb{E}\{\nu(t)\}\, \mathbb{E}\{(\vec{e}(t-1) - \mu\vec{\mathrm{a}}_t^\dagger \vec{e}(t-1)\vec{\mathrm{a}}_t)^\dagger \vec{\mathrm{a}}_t\} = 0. \end{aligned} \quad (11)$$

2) Measurement Noise Effect:

$$\begin{aligned} &\text{Measurement noise term} = \mathbb{E}\{\mu^2 \nu(t)\vec{\mathrm{a}}_t^\dagger \nu(t)\vec{\mathrm{a}}_t\} \\ &= \mu^2\, \mathbb{E}\{\nu(t)^2\}\, \mathbb{E}\{\|\vec{\mathrm{a}}_t\|^2\} = \mu^2\sigma_\nu^2 L \sigma_a^2 = L\mu^2\sigma_\nu^2\sigma_a^2. \end{aligned} \quad (12)$$

3) Estimation Error Effect:

$$\begin{aligned} \text{Est. err.} &= \mathbb{E}\{\|\vec{e}(t-1) - \mu\vec{\mathrm{a}}_t\vec{\mathrm{a}}_t^\dagger \vec{e}(t-1)\|^2\} \\ &= \mathbb{E}\{\|\vec{e}(t-1)\|^2\} + \mu^2\, \mathbb{E}\{\|\vec{\mathrm{a}}_t\vec{\mathrm{a}}_t^\dagger \vec{e}(t-1)\|^2\} \\ &\quad - 2\mu\, \mathbb{E}\{\vec{e}(t-1)^\dagger \vec{\mathrm{a}}_t\vec{\mathrm{a}}_t^\dagger \vec{e}(t-1)\}. \end{aligned} \quad (13)$$

By definition, $\xi(t-1) = \mathbb{E}\{\|\vec{e}(t-1)\|^2\}$. By applying the trace operator, $\operatorname{Tr}(\cdot)$, and using $\operatorname{Tr}(B.A) = \operatorname{Tr}(A.B)$, the remaining terms simplify as follows:

$$\begin{aligned} \mathbb{E}\{\vec{e}(t-1)^\dagger \vec{\mathrm{a}}_t\vec{\mathrm{a}}_t^\dagger \vec{e}(t-1)\} &= \operatorname{Tr}(\mathbb{E}\{\vec{e}(t-1)\vec{e}(t-1)^\dagger\}\, \mathbb{E}\{\vec{\mathrm{a}}_t\vec{\mathrm{a}}_t^\dagger\}) \\ &= \sigma_a^2\, \mathbb{E}\{\|\vec{e}(t-1)\|^2\} = \sigma_a^2 \xi(t-1). \end{aligned}$$

Similarly, for $\mathbb{E}\{\|\vec{a}_t\vec{a}_t^\dagger\vec{a}_t\vec{a}_t^\dagger\|\}$:

$$\mathbb{E}\{\|\vec{a}_t\vec{a}_t^\dagger\vec{e}(t-1)\|^2\} = (2+L)\sigma_a^4\xi(t-1). \tag{14}$$

Thus, (10) becomes:

$$\xi(t) = L\mu^2\sigma_\nu^2\sigma_a^2 + \big(1 - 2\mu\sigma_a^2 + (2+L)\mu^2\sigma_a^4\big)\xi(t-1).$$

Defining $\kappa_1 \triangleq L\mu^2\sigma_\nu^2\sigma_a^2$ and $\kappa_2 \triangleq 1 - 2\mu\sigma_a^2 + (2+L)\mu^2\sigma_a^4$, we write:

$$\begin{aligned}\xi(t) &= \kappa_1 + \kappa_2\xi(t-1) = \kappa_1\sum_{k=0}^{t-1}\kappa_2^k + \kappa_2^t\xi(0)\\ &= \kappa_1\frac{1-\kappa_2^t}{1-\kappa_2} + \kappa_2^t\xi(0). \end{aligned}\tag{15}$$

If $|\kappa_2| < 1$, convergence is guaranteed. Moreover, the steady-state error level is determined as:

$$\lim_{t\to+\infty}\xi(t) = \frac{\kappa_1}{1-\kappa_2} = \frac{L\mu^2\sigma_\nu^2}{2-(2+L)\mu\sigma_a^2}. \tag{16}$$

IV. Thresholding Mask Retrieval

Having an estimate of σ_ν^2, one can design a threshold level to prioritize the signal component over noise elements picked up. As (3) suggests, one requires complete knowledge of the SNR level to attain optimal LSS weighting, which is quite impractical, as discussed earlier.

To relax the optimality constraint, we propose ignoring the target components that are relatively small compared to the estimation noise level in (16).

Consider the auto-correlation matrix of $\vec{\varepsilon}(t)$ as follows:

$$\begin{aligned}R_e(t) &= \mathbb{E}\{\vec{e}(t)\vec{e}(t)^\dagger\}\\ &= \mathbb{E}\big\{\big(\vec{e}(t-1) - \mu\big[\vec{a}_t^\dagger\vec{e}(t-1) - \nu(t)\big]\vec{a}_t\big)\\ &\qquad\big(\vec{e}(t-1) - \mu\big[\vec{a}_t^\dagger\vec{e}(t-1) - \nu(t)\big]\vec{a}_t\big)^\dagger\big\},\end{aligned}\tag{17}$$

which can be rewritten as a recursive relation as:

$$\begin{aligned}R_e(t) &= R_e(t-1) - 2\mu\sigma_a^2R_e(t-1) + \mu^2\big[\sigma_a^4\dots\\ &\quad\big(\mathrm{Tr}(R_e(t-1))\mathbb{I} + 2R_e(t-1)\big) + \sigma_\nu^2\sigma_a^2\mathbb{I}\big].\end{aligned}\tag{18}$$

In order to solve this system of equations, we break it into two parts.

1) Off-Diagonal Components: Considering the off-diagonal components of $R_e(t)$, we can write:

$$R_e(t) = (1 - 2\mu\sigma_a^2 + 2\mu^2\sigma_a^4)^tR_e(0). \tag{19}$$

Based on the convergence criteria $0 < \kappa_2 < 1$, and $0 < 1 - 2\mu\sigma_a^2$, the condition $0 < 1 - 2\mu\sigma_a^2 + 2\mu^2\sigma_a^4 < 1$ is guaranteed, and (19) will converge to zero for large t.

It is also worth mentioning that $R_e(0)$ is determined by the initialization procedure and depends on the target signal as follows:

$$\begin{aligned}R_e(0) &= \mathbb{E}\{\vec{e}(0)\vec{e}(0)^\dagger\} = \mathbb{E}\{(\vec{\omega}(0) - \vec{\omega}^{\mathrm{opt}})(\vec{\omega}(0) - \vec{\omega}^{\mathrm{opt}})^\dagger\}\\ &= R_{\omega(0)} + R_{\omega_{\mathrm{opt}}} - \mathbb{E}\{\vec{\omega}(0)\}\,\mathbb{E}\{\vec{\omega}^{\mathrm{opt}}\}^\dagger - \mathbb{E}\{\vec{\omega}^{\mathrm{opt}}\}\,\mathbb{E}\{\vec{\omega}(0)\}^\dagger.\end{aligned}$$

It is common to have a zero initialization or a zero-mean i.i.d. random initialization, which results in:

$$R_{\omega(0)} = \sigma_{\omega(0)}^2\mathbb{I} \Rightarrow R_e(0) = \sigma_{\omega(0)}^2\mathbb{I} + R_{\omega_{\mathrm{opt}}}. \tag{20}$$

On the other hand, considering a target signal with uncorrelated components, $R_e(0)$ would be a diagonal matrix, implying that for off-diagonal components, $R_e(t) = 0$.

It should be noted that when the off-diagonal components are zero, the independence assumption is valid.

2) Diagonal Components: For diagonal components, we define $\vec{r}_e(t) \triangleq \mathrm{diag}(R_e(t))$. It should be noted that $\mathrm{Tr}(R_e(t)) = \mathbb{1}_L^\dagger\vec{r}_e(t)$, which means:

$$\begin{aligned}\vec{r}_e(t) &= \big[(1 - 2\mu\sigma_a^2 + 2\mu^2\sigma_a^4)\mathbb{I} + \mu^2\sigma_a^4\mathbb{1}_{L\times L}\big]^t\vec{r}_e(0)\\ &+\sigma_\nu^2\sigma_a^2\sum_{k=0}^{t-1}\big[(1 - 2\mu\sigma_a^2 + 2\mu^2\sigma_a^4)\mathbb{I} + \mu^2\sigma_a^4\mathbb{1}_{L\times L}\big]^k\vec{\mathbb{1}}_L.\end{aligned}\tag{21}$$

In order to simplify (21), it can be shown:

$$(\delta_2\mathbb{I} + \delta_1\mathbb{1}_{L\times L})^n = \frac{(\delta_2 + L\delta_1)^n - \delta_2^n}{L}\mathbb{1}_{L\times L} + \delta_2^n\mathbb{I}. \tag{22}$$

By defining $\delta_2 \triangleq 1 - 2\mu\sigma_a^2 + 2\mu^2\sigma_a^4$ and $\delta_1 \triangleq \mu^2\sigma_a^4$, we can see $\delta_2 + L\delta_1 = \kappa_2$. Thus, we have

$$\begin{aligned}\vec{r}_e(t) &= \Big(\frac{(\delta_2 + L\delta_1)^t - \delta_2^t}{L}\mathbb{1}_{L\times L} + \delta_2^t\mathbb{I}\Big)\vec{r}_e(0)\\ &+ \sigma_\nu^2\sigma_a^2\sum_{k=0}^{t-1}\Big(\frac{(\delta_2 + L\delta_1)^k - \delta_2^k}{L}\mathbb{1}_{L\times L} + \delta_2^k\mathbb{I}\Big)\vec{\mathbb{1}}_L.\end{aligned}\tag{23}$$

It is straightforward to show that:

$$\vec{r}_e(t) = \frac{1}{L}\left(\kappa_2^t\mathbb{1}_{L\times L}\vec{r}_e(0) + \kappa_1\frac{1-\kappa_2^t}{1-\kappa_2}\vec{\mathbb{1}}_L\right). \tag{24}$$

By considering cases with $t \gg 1$, we have $\delta_2^t \to 0$ and $\delta_1^t \to 0$. Thus, the error floor can be achieved as follows:

$$\lim_{t\to+\infty}\vec{r}_e(t) = \frac{\kappa_1}{L(1-\kappa_2)}\vec{\mathbb{1}}_L, \tag{25}$$

which shows the expected error floor of each estimation component.

By the matrix representation, for $t \gg 1$ we have:

$$R_e(t) = \frac{\kappa_1}{1-\kappa_2}\mathbb{I}. \tag{26}$$

As a result, by noting $\xi(t) = \vec{\mathbb{1}}_L^\dagger\vec{r}_e(t)$ we have:

$$\xi(t) = \kappa_2^t\xi(0) + \kappa_1\frac{1-\kappa_2^t}{1-\kappa_2}, \tag{27}$$

which is the same result as in (15).

V. Mask Estimation Error

Considering the estimation error $\vec{e}(t)$, we have a multivariate Gaussian distribution with a mean vector $\vec{\epsilon}(t)$ from (6) and a covariance matrix $R_e(t) - \vec{\epsilon}(t)\vec{\epsilon}(t)^\dagger$, as given in (26). Based on the description following (20), we can write:

$$R_e(0) = \sigma_{\omega(0)}^2 I + \vec{\omega}^{\mathrm{opt}}(\vec{\omega}^{\mathrm{opt}})^\dagger, \quad \vec{\epsilon}(0) = \vec{\omega}^{\mathrm{opt}},$$

which results in:

$$\begin{aligned}\mathrm{mean} &= (I - \mu\sigma_a^2)^t\vec{\epsilon}(0) = (I - \mu\sigma_a^2)^t\vec{\omega}^{\mathrm{opt}},\\ \mathrm{cov} &= \frac{1}{L}\left(\kappa_2^t\left(L\sigma_{\omega(0)}^2 + \|\vec{\omega}^{\mathrm{opt}}\|^2\right) + \kappa_1\frac{1-\kappa_2^t}{1-\kappa_2}\right)I + [\delta_2^t\\ &\quad - (I - \mu\sigma_a^2)^{2t}]\vec{\omega}^{\mathrm{opt}}(\vec{\omega}^{\mathrm{opt}})^\dagger - \delta_2^t\,\mathrm{diag}\left[\vec{\omega}^{\mathrm{opt}}(\vec{\omega}^{\mathrm{opt}})^\dagger\right].\end{aligned}$$

For $t \gg 1$, we can write:

$$\text{mean} = 0, \quad \text{cov} = \frac{\kappa_1}{1-\kappa_2} I. \tag{28}$$

This implies that each component is i.i.d. zero-mean normally distributed with a variance of $\frac{\kappa_1}{1-\kappa_2}$.

Let x denote the l-th component of an arbitrary node i and $\mathbb{T}$ be the thresholding level. We investigate two cases: false positive (FP), which refers to mistakenly detecting a masked x as an unmasked component, and false negative (FN), which refers to ignoring an unmasked component as a masked one:

$$\text{FP} = \text{Prob}(|x| > \mathbb{T}), \quad x \sim \mathcal{N}(0, \sigma_{\text{est}}^2),$$
$$\text{FN} = \text{Prob}(|x| < \mathbb{T}), \quad x \sim \mathcal{N}(\omega_l^{\text{opt}}, \sigma_{\text{est}}^2),$$

where $\sigma_{\text{est}}^2 \triangleq \frac{\kappa_1}{1-\kappa_2}$. These result in:

$$\text{FP} = 2Q\left(\frac{\mathbb{T}}{\sigma_{\text{est}}}\right), \ \text{FN} = Q\left(-\frac{\mathbb{T}+\omega_l^{\text{opt}}}{\sigma_{\text{est}}}\right) - Q\left(\frac{\mathbb{T}-\omega_l^{\text{opt}}}{\sigma_{\text{est}}}\right). \tag{29}$$

If each component is masked with a probability of $1-\rho$, then the total error can be expressed as:

$$\text{Total Error} = \rho \cdot \text{FN} + (1-\rho) \cdot \text{FP}. \tag{30}$$

To minimize the total error, one can take its derivative with respect to $\mathbb{T}$, set it to zero, and use numerical methods to solve it for the specific scenario at hand.

VI. Simulation

Fig. 2 shows that error estimation from (15) aligns with observations. Using the correction coefficient:

$$\frac{L}{\sum_j D_i(j)} \cdot \frac{\|M_i(\vec{\omega}_i - \vec{\omega}^{\text{opt}})\|^2}{\|M_i \vec{\omega}^{\text{opt}}\|^2}, \tag{31}$$

the local error is evaluated as:

$$\frac{\|\vec{\omega}_i - M_i \vec{\omega}^{\text{opt}}\|^2}{\|M_i \vec{\omega}^{\text{opt}}\|^2}, \tag{32}$$

Combining strategies improve consensus performance. Three methods were tested:

1) "avg": Averaging over neighboring nodes.
2) "opt": Optimal LSS weighting from (3).
3) "unbiased": Optimal unbiased weighting from (3).

As predicted by (3), the LSS weighting converges to an unbiased estimation in high SNR scenarios.

In Fig. 3, it is demonstrated that (29) and (30) (denoted as "theory" in the figure) effectively predict the probability of incorrect estimation of mask components. At index $t = 700$, we attempted to estimate the unknown mask, and the error probability as a function of the thresholding level $\mathbb{T}$ is reported.

The results indicate that the theoretical and practical probabilities of estimation error are closely aligned. It is noteworthy that an optimal threshold value $\mathbb{T}$ exists at the intersection of $\rho \cdot FP$ and $(1-\rho) \cdot FN$, owing to their monotonic behavior.

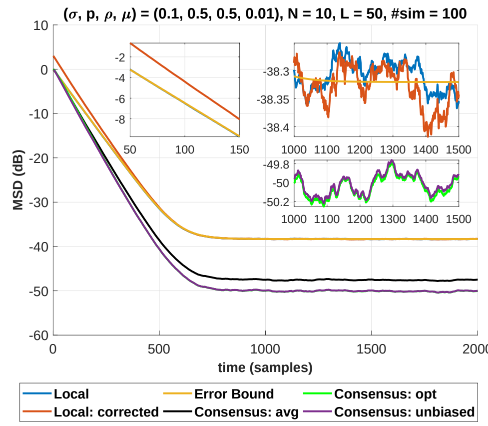

Fig. 2: Error Bound: $\rho = 0.5$.

Additionally, it can be observed that for lower values of $\mathbb{T}$, the false positive (FP) rate plays a dominant role, whereas for higher values of $\mathbb{T}$, the false negative (FN) rate becomes more significant.

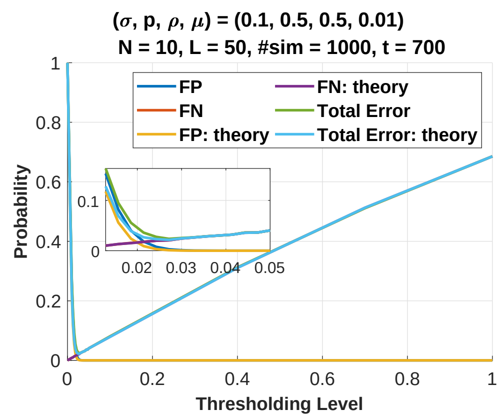

Fig. 3: Mask Estimation.

VII. Conclusion

We analyzed the error behavior of distributed estimators with masked measurements and examined a thresholding-based algorithm that improves information flow and enhances node estimation through shared data. Our analysis demonstrated how this approach enables accurate estimation without prior knowledge of the mask. Simulations validated its effectiveness in predicting DLMS convergence under sparse and partial information access.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] Fuyi Huang, Sheng Zhang, and Wei Xing Zheng, “Bayesian-learning-based diffusion least mean square algorithms over networks,” IEEE Transactions on Neural Networks and Learning Systems, 2023.

[7] 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.

[8] Mahdi Shamsi and Farokh Marvasti, “Distributed estimation with partially accessible information: An imat approach to lms diffusion,” arXiv preprint arXiv:2310.09970, 2023.

[9] 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.

[10] 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.

[11] Sung-Hyuk Yim, Han-Sol Lee, and Woo-Jin Song, “A proportionate diffusion lms algorithm for sparse distributed estimation,” IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 62, no. 10, pp. 992–996, 2015.

[12] Hadi Zayyani and Amirhossein Javaheri, “A robust generalized proportionate diffusion lms algorithm for distributed estimation,” IEEE transactions on circuits and systems II: Express Briefs, vol. 68, no. 4, pp. 1552–1556, 2020.

[13] Muhammed O Sayin and Suleyman Serdar Kozat, “Compressive diffusion strategies over distributed networks for reduced communication load,” IEEE Transactions on Signal Processing, vol. 62, no. 20, pp. 5308–5323, 2014.

[14] Yishu Peng, Sheng Zhang, Hongyang Chen, Zhengchun Zhou, and Xiaohu Tang, “Frequency-domain diffusion bias-compensated adaptation with periodic communication,” IEEE Transactions on Signal and Information Processing over Networks, 2023.

[15] Anil Jadhav, Dhanya Pramod, and Krishnan Ramanathan, “Comparison of performance of data imputation methods for numeric dataset,” Applied Artificial Intelligence, vol. 33, no. 10, pp. 913–933, 2019.

[16] 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.

[17] Mohammad Reza Gholami, Magnus Jansson, Erik G Ström, and Ali H Sayed, “Diffusion estimation over cooperative multi-agent networks with missing data,” IEEE Transactions on Signal and Information Processing over Networks, vol. 2, no. 3, pp. 276–289, 2016.

[18] Daniel G Tiglea, Renato Candido, Luis A Azpicueta-Ruiz, and Magno TM Silva, “Reducing the communication and computational cost of random fourier features kernel lms in diffusion networks,” in ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2023, pp. 1–5.

[19] Zhaogeng Liu, Feng Ji, Jielong Yang, Xiaofeng Cao, Muhan Zhang, Hechang Chen, and Yi Chang, “Refining euclidean obfuscatory nodes helps: A joint-space graph learning method for graph neural networks,” IEEE Transactions on Neural Networks and Learning Systems, vol. 35, no. 9, pp. 11720–11733, 2024.

[20] Geert Leus, Antonio G Marques, José MF Moura, Antonio Ortega, and David I Shuman, “Graph signal processing: History, development, impact, and outlook,” IEEE Signal Processing Magazine, vol. 40, no. 4, pp. 49–60, 2023.

[21] Hoi-To Wai, Yonina C Eldar, Asuman E Ozdaglar, and Anna Scaglione, “Community inference from partially observed graph signals: Algorithms and analysis,” IEEE Transactions on Signal Processing, vol. 70, pp. 2136–2151, 2022.