Hard Thresholding Based LMS Algorithm for Adaptive Graph Signal Recovery

Rachita Mohanty and Rajib Lochan Das
Communication and Signal Processing Area
Dhirubhai Ambani University, Gandhinagar, Gujarat, India
{202421026, rajib_das}@dau.ac.in

***Abstract*—This paper proposes a sparsity-aware adaptive algorithm for graph signal recovery by integrating a hard-thresholding mechanism into the Graph Least Mean Squares (GLMS) framework. The proposed Hard Thresholding Graph LMS (HT-GLMS) algorithm exploits spectral sparsity while adaptively estimating bandlimited graph signals from incomplete and noisy observations. At each iteration, a thresholding step suppresses insignificant spectral coefficients while retaining dominant components, leading to improved convergence speed and reduced steady-state error. Simulation results on synthetic graph structures demonstrate that HT-GLMS consistently outperforms conventional GLMS algorithms in terms of normalized mean-square deviation (NMSD). Performance analysis under different threshold values reveals an optimal threshold range closely aligned with the true bandwidth of the graph signal. Furthermore, tracking experiments under abrupt signal changes show that HT-GLMS achieves faster reconvergence and lower post-change error, demonstrating strong adaptability in dynamic graph environments. These results establish HT-GLMS as an effective and computationally efficient approach for adaptive sparse graph signal estimation.**

***Index Terms*—Graph signal processing, adaptive filtering, graph LMS, hard thresholding, sparse recovery, tracking performance.**

I. Introduction

Graph Signal Processing (GSP) has emerged as a powerful framework for analyzing data defined over irregular structures such as sensor networks, transportation systems, and social networks [1]–[3]. In many practical scenarios, graph signals are only partially observed or contaminated with noise, necessitating robust recovery algorithms that can exploit the underlying spectral and structural properties of the graph.

Adaptive methods, particularly those derived from the Least Mean Squares (LMS) algorithm, have been widely investigated for their simplicity, low computational cost, and suitability for real-time graph environments [4]–[6], [13]. The Graph Least Mean Squares (GLMS) algorithm extends the conventional LMS to the graph spectral domain by iteratively updating the graph Fourier coefficients using local information [5]. While GLMS effectively reconstructs smooth or band-limited graph signals, it does not explicitly exploit the inherent sparsity that often exists in the graph spectral domain. In many graph-based applications, only a small subset of spectral coefficients carries meaningful information, while the rest are negligible. Since GLMS applies uniform adaptation across all coefficients, it fails to leverage this spectral sparsity, leading to inefficient updates, slower convergence, and suboptimal steady-state accuracy, particularly under noisy or under-sampled measurement conditions [6], [13].

To mitigate these challenges, sparsity-aware extensions of LMS have been proposed, including the Proportionate Graph LMS (PGLMS) and threshold-based variants [7]–[10], [14], [15]. The PGLMS algorithm improves convergence speed by assigning coefficient-dependent step sizes, emphasizing larger-magnitude coefficients that dominate the signal representation. Although PGLMS achieves faster adaptation for sparse signals, it exhibits several limitations: (i) it requires careful parameter tuning to maintain stability, (ii) it can amplify noise in weak coefficients, (iii) it performs poorly when the signal sparsity pattern changes dynamically, and (iv) it does not significantly improve the steady-state error compared to conventional LMS or GLMS. Recent works have extended proportionate adaptation mechanisms to more general frameworks, including iterative reweighting and fast-LMS variants [7], [14], [16].

Among sparsity-enforcing techniques, Hard Thresholding (HT) offers a computationally efficient approach for promoting sparse representations in adaptive estimation. From a compressed sensing perspective, the hard-thresholding operation can be interpreted as a projection of the estimated coefficient vector onto a sparse subspace by retaining only the most significant components. After each adaptation step, coefficients with magnitudes below a predefined threshold are set to zero, thereby enforcing sparsity and suppressing noise-driven fluctuations [11], [12]. Motivated by these properties, this work introduces a Hard Thresholding Graph LMS (HT-GLMS) algorithm for adaptive recovery of bandlimited graph signals.

II. Review of Graph Adaptive Filtering

A. Problem Formulation

Consider an undirected weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$ with $N = |\mathcal{V}|$ nodes, edge set $\mathcal{E}$, and adjacency matrix $\mathbf{W} \in \mathbb{R}^{N\times N}$.

The combinatorial graph Laplacian (L) is defined as

$$\mathbf{L} = \mathbf{D}_g - \mathbf{W}, \tag{1}$$

where $\mathbf{D}_g$ denotes the *degree matrix* with diagonal entries

$$(\mathbf{D}_g)_{ii} = \sum_j W_{ij}.$$

The eigen-decomposition of $\mathbf{L}$ is given by

$$\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top, \tag{2}$$

where $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_N]$ contains orthonormal graph Fourier basis vectors and $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_N)$ contains the Laplacian eigenvalues.

A graph signal $\mathbf{x}_0 \in \mathbb{R}^N$ is said to be F-bandlimited if

$$\mathbf{x}_0 = \mathbf{U}\mathbf{s}_0, \tag{3}$$

where, the spectral coefficient vector $\mathbf{s}_0 \in \mathbb{R}^N$ is F-bandlimited, i.e., only the first F coefficients are nonzero while the remaining $N - F$ coefficients are zero.

While s_0 is estimated from its noisy version, it is expected that $\hat{s}_0$ will contain only F number of nonzero elements & thus $\hat{s}_0$ will be considered a sparse signal and can be exploited to achieve better performance, as discussed below.

Considering additive noise $\mathbf{v}$, the noisy observation $\mathbf{y}_n$ can be written as

$$\mathbf{y}_n = \mathbf{x}_0 + \mathbf{v}. \tag{4}$$

Since observing all nodes may be impractical, only a subset of nodes is sampled, where M out of N nodes are selected for measurement. Let $\mathbf{B} \in \mathbb{R}^{N \times N}$ denote a diagonal sampling matrix defined as

$$B_{ii} = \begin{cases} 1, & \text{if node } i \text{ is sampled,} \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

The observation $\mathbf{y}$ is obtained by sampling the noisy signal:

$$\mathbf{y} = \mathbf{B}\mathbf{y}_n. \tag{6}$$

Substituting $\mathbf{y}_n = \mathbf{x}_0 + \mathbf{v}$, we get

$$\mathbf{y} = \mathbf{B}(\mathbf{x}_0 + \mathbf{v}). \tag{7}$$

Using $\mathbf{x}_0 = \mathbf{U}\mathbf{s}_0$, we get

$$\mathbf{y} = \mathbf{B}\mathbf{U}\mathbf{s}_0 + \mathbf{B}\mathbf{v}. \tag{8}$$

Now, define the sampled noise as

$$\mathbf{v}' = \mathbf{B}\mathbf{v}. \tag{9}$$

Thus, we obtain

$$\mathbf{y} = \mathbf{B}\mathbf{U}\mathbf{s}_0 + \mathbf{v}'. \tag{10}$$

Let us consider

$$\mathbf{A} = \mathbf{B}\mathbf{U}. \tag{11}$$

Thus, we may write

$$\mathbf{y} = \mathbf{A}\mathbf{s}_0 + \mathbf{v}'. \tag{12}$$

Our objective is to estimate the spectral coefficients $\mathbf{s}_0$ and reconstruct the graph signal as

$$\hat{\mathbf{x}} = \mathbf{U}\hat{\mathbf{s}}, \tag{13}$$

based on sampled and noisy observations of the underlying graph signal.

The Graph Least Mean Squares (GLMS) algorithm is one of the most popular methods for adaptive signal estimation. In the following section, the GLMS algorithm is briefly discussed.

B. Graph Least Mean Squares (GLMS) [5]

To estimate the spectral coefficients, the instantaneous quadratic cost function is defined as

$$J(n) = \frac{1}{2}\|\mathbf{y}(n) - \mathbf{A}\hat{\mathbf{s}}(n)\|_2^2. \tag{14}$$

The gradient of the cost function with respect to $\hat{\mathbf{s}}(n)$ is given by

$$\nabla J(n) = \frac{\partial J(n)}{\partial \hat{\mathbf{s}}(n)} = -\mathbf{A}^\top \mathbf{e}(n) \tag{15}$$

where the error signal is given by

$$\mathbf{e}(n) = \mathbf{y}(n) - \mathbf{A}\hat{\mathbf{s}}(n). \tag{16}$$

Using a stochastic gradient descent strategy, the GLMS update becomes

$$\hat{\mathbf{s}}(n+1) = \hat{\mathbf{s}}(n) + \mu \mathbf{A}^\top \mathbf{e}(n), \tag{17}$$

where $\mu > 0$ denotes the step size.

For mean–square stability, the step size must satisfy

$$0 < \mu < \frac{2}{\lambda_{\max}(\mathbf{A}^\top \mathbf{A})}.$$

Although GLMS provides reliable estimation of spectral coefficients, it does not explicitly exploit sparsity in the graph spectral domain. To address this limitation, a hard-thresholding mechanism is incorporated into the GLMS framework.

III. Proposed Graph Hard-Thresholding LMS (HT-GLMS) Algorithms

The hard-thresholding strategy is incorporated into the GLMS framework to enforce sparsity in the graph spectral domain.

In this work, two sparsity-promoting approaches are considered: (i) a fixed-sparsity method based on retaining a predefined number of coefficients, and (ii) a magnitude-based method using a threshold value.

For both approaches, the hard-thresholding operator is applied to the intermediate GLMS update, as given below.

$$\hat{\mathbf{s}}^+(n+1) = \hat{\mathbf{s}}(n) + \mu \mathbf{A}^\top \mathbf{e}(n). \tag{18}$$

A. Proposed Algorithm I: HT-GLMS$_K$

In this formulation, the spectral sparsity level K is assumed to be known a priori. Sparsity is enforced by retaining only the K largest-magnitude spectral coefficients after each update. The hard-thresholding operator $H_K(\cdot)$ is defined as

$$[H_K(\mathbf{x})]_i = \begin{cases} x_i, & \text{if } |x_i| \text{ is among the } K \text{ largest,} \\ 0, & \text{otherwise.} \end{cases} \tag{19}$$

The update rule of the proposed algorithm is thus given by

$$\hat{\mathbf{s}}(n+1) = H_K\big(\hat{\mathbf{s}}^+(n+1)\big), \tag{20}$$

and the reconstructed graph signal is

$$\hat{\mathbf{x}}(n) = \mathbf{U}\hat{\mathbf{s}}(n). \tag{21}$$

B. Proposed Algorithm II: HT-GLMS$_\tau$

To eliminate the need for prior knowledge of K, a magnitude-based threshold τ is employed. Instead of fixing the number of retained coefficients, this approach selects coefficients based on their magnitudes.

The update rule is given by

$$\hat{s}_i(n+1) = \begin{cases} \hat{s}_i^{+}(n+1), & \text{if } \left|\hat{s}_i^{+}(n+1)\right| \geq \tau, \\ 0, & \text{otherwise.} \end{cases} \tag{22}$$

The reconstructed graph signal is thus

$$\hat{\mathbf{x}}(n) = \mathbf{U}\hat{\mathbf{s}}(n). \tag{23}$$

IV. Results and Discussion

In all simulations, the step size is set to $\mu = 0.01$, and the algorithms are executed for $T = 2000$ iterations. The reported results correspond to averages over 100 independent Monte-Carlo runs.

A. Synthetic Graph Topology

Fig. 1 illustrates the synthetic graph topology used in the experiments. An undirected random graph with $N = 50$ nodes is generated to represent the underlying network structure for the graph signal processing experiments. The topology provides a connected and moderately dense graph, enabling reliable signal propagation across nodes. All edges are assigned unit weights.

Since observing all graph nodes may be impractical, only $M = 30$ nodes are sampled in the simulations, resulting in a reduced set of measurements.

The graph Laplacian $\mathbf{L}$ is constructed according to (1), and its eigenvectors form the Graph Fourier basis $\mathbf{U}$.

The simulated graph signal is generated as

$$\mathbf{x}_0 = \mathbf{U}\mathbf{s}_0, \tag{24}$$

where $\mathbf{s}_0 \in \mathbb{R}^N$ denotes the spectral coefficient vector. The graph signal is assumed to be bandlimited to the first $F = 16$ graph frequencies. The spectral vector $\mathbf{s}_0$ is generated randomly such that only the first F components are nonzero, while the remaining coefficients are set to zero. Among these F components, only $K \leq 16$ coefficients carry dominant energy, resulting in a spectrally sparse signal within the bandlimited subspace. This setting enables a fair comparison between GLMS, HT-GLMS$_K$, and HT-GLMS$_\tau$.

Since estimation is performed in the spectral coefficient domain, the performance is evaluated using the Normalized Mean Square Deviation (NMSD) defined as

$$\text{NMSD}(n) = 10\log_{10}\left(\frac{\|\mathbf{s}_0 - \hat{\mathbf{s}}(n)\|_2^2}{\|\mathbf{s}_0\|_2^2}\right). \tag{25}$$

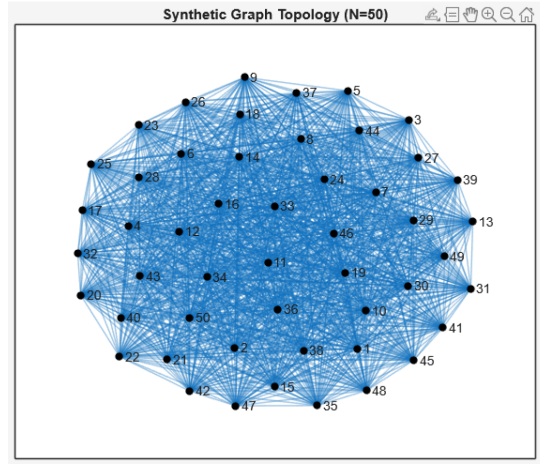

Fig. 1: Synthetic Graph Topology ($N = 50$).

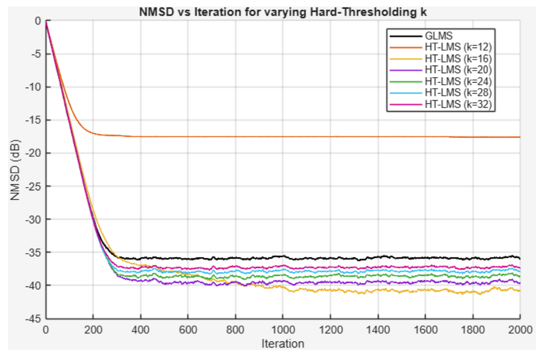

Fig. 2: NMSD vs Iteration for different sparsity levels K.

B. Performance of HT-GLMS$_K$ for different Sparsity Levels(K)

Fig. 2 illustrates the steady-state NMSD performance of the proposed HT-GLMS$_K$ algorithm for varying sparsity levels K. The results indicate a clear optimum at $K \approx 16$, where the minimum steady-state NMSD of approximately -41 dB is achieved. For K below this value, excessive sparsity induces bias due to the removal of informative spectral components. Conversely, increasing K beyond the effective sparsity level introduces noise-dominated coefficients, leading to increased steady-state variance and slight performance degradation.In comparison to the conventional GLMS, which converges to approximately -36 dB, the proposed HT-GLMS$_K$ algorithm achieves a steady-state improvement of nearly 5 dB.

C. Performance of HT-GLMS$_\tau$ for Different Magnitude Thresholds (τ)

Fig. 3 shows the NMSD versus iteration for different magnitude threshold values. All HT-GLMS variants converge faster and achieve lower steady-state NMSD compared to conventional GLMS (approximately -36 dB). Among the tested values, the threshold $\tau = 0.0015$ provides the best steady-state performance, reaching nearly -41 dB. Smaller thresholds (e.g., 0.0005) offer limited improvement due to weak

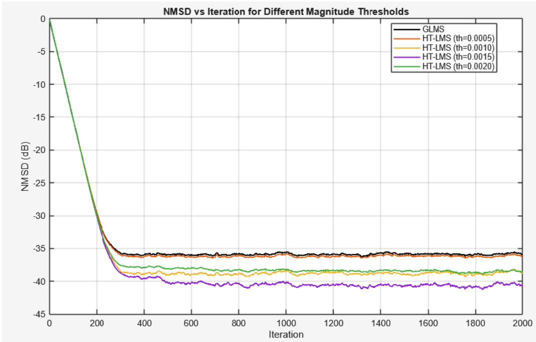

Fig. 3: NMSD vs Iteration for different magnitude thresholds.

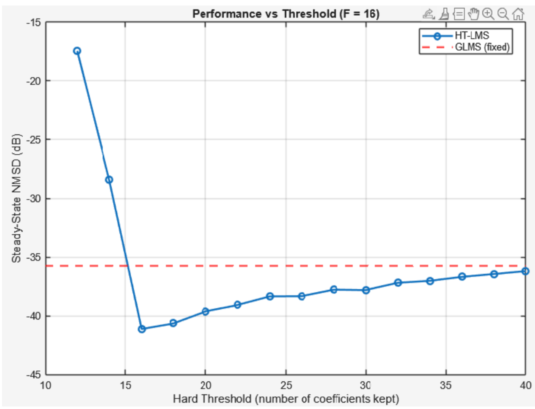

Fig. 4: Sensitivity of Steady-State NMSD to the Sparsity Level K.

sparsity enforcement, whereas larger thresholds (e.g., 0.0020) slightly degrade performance because of excessive suppression of informative coefficients.

D. Steady-State NMSD vs. Sparsity Level K

Fig. 4 presents the steady-state NMSD as a function of the sparsity level K for $F = 16$. This plot shows that increasing the sparsity level K beyond 16 results in only slight variations in the NMSD, indicating that the algorithm is relatively less sensitive to larger values of K. The best performance is achieved at $K = 16$, where the steady-state NMSD reaches approximately -41 dB. For example, when $K = 20$, the steady-state NMSD is around -40 dB, showing only a small degradation compared to the optimal case. This behavior indicates that moderate overestimation of K does not significantly affect the performance.

From a sensitivity point of view, the algorithm is highly sensitive to smaller values of K. When K is reduced below 16, the steady-state NMSD degrades rapidly due to the removal of active spectral components during the thresholding step, leading to significant performance loss.

Therefore, in practice, the sparsity level K should not be chosen smaller than the true signal bandwidth to avoid degradation in reconstruction performance.

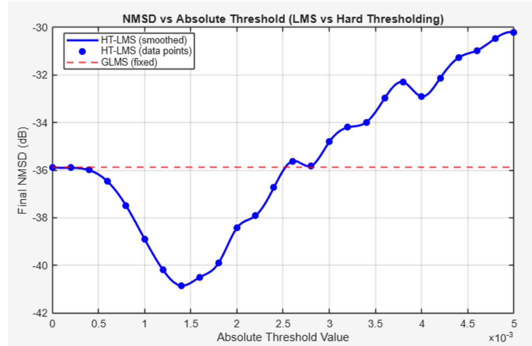

Fig. 5: Sensitivity of Steady-State NMSD to the Absolute Threshold τ.

E. Steady-State NMSD vs. Absolute Threshold τ.

Fig. 5 illustrates the steady-state NMSD as a function of the absolute threshold value τ. It can be observed that the algorithm performance varies significantly with the choice of the threshold parameter. The best performance is achieved at $\tau \approx 1.5 \times 10^{-3}$, where the steady-state NMSD reaches approximately -41 dB.

When the threshold value τ is too small, the sparsity enforcement becomes weak and many insignificant spectral coefficients remain in the estimate, resulting in a slightly higher NMSD. From a sensitivity perspective, the algorithm becomes less effective in suppressing noise-dominated coefficients when the threshold is very small.

Conversely, when the threshold value increases beyond the optimal point, the NMSD gradually degrades. For example, at $\tau \approx 4.0 \times 10^{-3}$, the steady-state NMSD is approximately -33 dB. This degradation occurs because larger threshold values suppress not only insignificant coefficients but also some active spectral components during the hard-thresholding step.

These observations indicate that the algorithm is sensitive to the choice of the threshold parameter, and selecting an appropriate value of τ is essential for achieving optimal recovery performance.

F. Tracking Performance Under Different Sparsity Levels and Thresholds

To evaluate the tracking capability of the proposed method, the true spectral coefficient vector is abruptly changed from s_0 to s_1 at iteration $n = 1000$ while preserving the sparsity level. The new coefficient vector s_1 is generated by randomly modifying the dominant spectral components, simulating a sudden variation in the underlying graph signal.

Fig. 6 shows the tracking performance of the algorithm for different sparsity levels K. The abrupt change produces a spike in the NMSD due to the mismatch between the estimated and true coefficients. After the change, the algorithms gradually reconverge to the new solution. From the figure, it can be observed that the algorithms require approximately 300 iterations after the change point to reconverge and reach a stable steady-state region. Among the tested values, $K = 16$

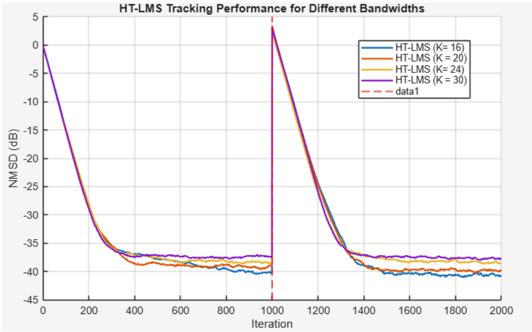

Fig. 6: Tracking Performance Under Different Sparsity Levels K.

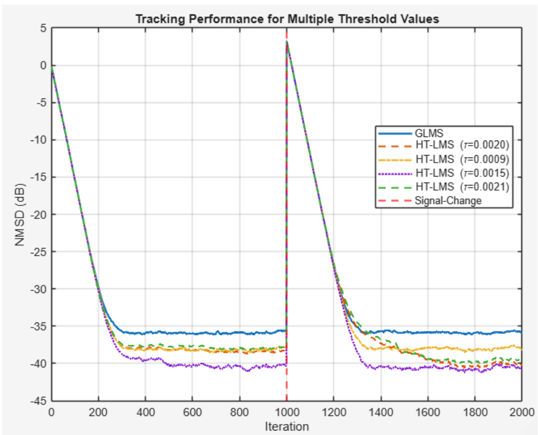

Fig. 7: Tracking performance of HT-GLMS$_\tau$ for different threshold values.

achieves the best steady-state performance, while larger values such as $K = 20$ and $K = 30$ result in slightly higher NMSD.

Fig. 7 presents the tracking performance of HT-GLMS$_\tau$ for different threshold values. Similar behavior is observed, where a spike occurs at the change point followed by reconvergence. Again, the algorithm reconverges within approximately 300 iterations after the abrupt change. The best performance is obtained for $\tau = 0.0015$, which achieves the lowest steady-state NMSD. These results confirm that appropriate selection of the sparsity level and threshold parameter improves both the steady-state accuracy and the tracking capability of the proposed algorithm.

G. Discussion

The results confirm that the integration of hard thresholding into GLMS improves steady-state accuracy and tracking behavior.

Compared to GLMS, HT-GLMS provides the following:

- Approximately 5 dB improvement in steady-state NMSD,
- Faster recovery after abrupt spectral changes,
- Improved robustness against noise-dominated coefficients.

Although GLMS has slightly lower computational overhead, HT-GLMS achieves superior performance when spectral sparsity exists within the bandlimited subspace.

V. Conclusion

This paper presented the Hard Thresholding Graph LMS (HT-GLMS) algorithm for sparse graph signal estimation. By incorporating a hard-thresholding step into the conventional GLMS framework, the proposed method effectively exploits spectral sparsity in bandlimited graph signals. Simulation results on a synthetic 50-node graph demonstrate that HT-GLMS achieves higher steady-state accuracy and faster convergence than conventional GLMS. The thresholding mechanism suppresses noise-dominated coefficients while preserving the dominant spectral components, thereby improving the estimation performance with only a minor additional computational cost.

Tracking experiments further show that the proposed algorithms adapt effectively to abrupt changes in the underlying graph signal, achieving faster reconvergence and improved post-change estimation accuracy. These results highlight the robustness and effectiveness of the HT-GLMS approach for adaptive graph signal recovery in dynamic environments.

Future work could be directed toward adaptive threshold selection strategies, the exploration of alternative sparsity-promoting mechanisms such as soft-thresholding, and the development of distributed implementations for large-scale and time-varying graph networks.

References

[1] D. I. Shuman *et al.*, "The emerging field of signal processing on graphs," *IEEE Signal Processing Magazine*, 2013.

[2] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs," *IEEE Transactions on Signal Processing*, 2013.

[3] A. Ortega, P. Frossard, J. Kovacevic, J. F. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[4] S. Chen *et al.*, "Discrete signal processing on graphs: Sampling theory," *IEEE Trans. Signal Process.*, 2015.

[5] P. Di Lorenzo, S. Barbarossa, and P. Banelli, "Adaptive least mean squares estimation of graph signals," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 3, pp. 555–568, 2018.

[6] R. Nassif and C. Richard, "A graph diffusion LMS strategy for adaptive graph signal processing," in *Proc. ICASSP*, 2017.

[7] R. Torkamani, H. Zayyani, and M. Korki, "Proportionate adaptive graph signal recovery," *IEEE Trans. Signal Process.*, 2023.

[8] Y. Chen, Y. Gu, and A. O. Hero, "Sparse LMS for system identification," in *Proc. ICASSP*, 2009.

[9] L. Flokas and P. Maragos, "Hard threshold least mean squares algorithm," *arXiv preprint arXiv:1608.01128*, 2016.

[10] H.-S. Lee *et al.*, "Sparse diffusion least mean-square algorithm with hard thresholding over networks," *Mathematics*, 2023.

[11] S. Foucart, "Hard thresholding pursuit: An algorithm for compressive sensing," *SIAM Journal on Numerical Analysis*, 2011.

[12] S. Theodoridis, "Sparsity-aware adaptive learning: A set-theoretic view," *Signal Processing*, 2015.

[13] W. Wang, J. Zhang, M. Liu, and X. Jin, "Robust adaptive estimation of graph signals based on LMS strategies," *Symmetry*, 2022.

[14] C.-H. Lee, B. D. Rao, and H. Garudadri, "Proportionate adaptive filtering algorithms derived using an iterative reweighting framework," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 171–186, 2020.

[15] Z. Qin, J. Tao, L. Yang, and M. Jiang, "Proportionate recursive maximum correntropy criterion adaptive filtering algorithms and their performance analysis," *arXiv preprint arXiv:2210.12311*, 2022.

[16] K.-L. Chen, C.-H. Lee, B. D. Rao, and H. Garudadri, "A generalized proportionate-type normalized subband adaptive filter," in *Proc. Asilomar Conf. Signals, Systems, and Computers*, pp. 749–753, 2019.