# Fast Graph Filters for Decentralized Subspace Projection


**Daniel Romero**, Siavash Mollaebrahim, Baltasar Beferull-Lozano, and Cesar Asensio-Marco

Dept. of Information and Communication Technology, University of Agder, Grimstad, Norway

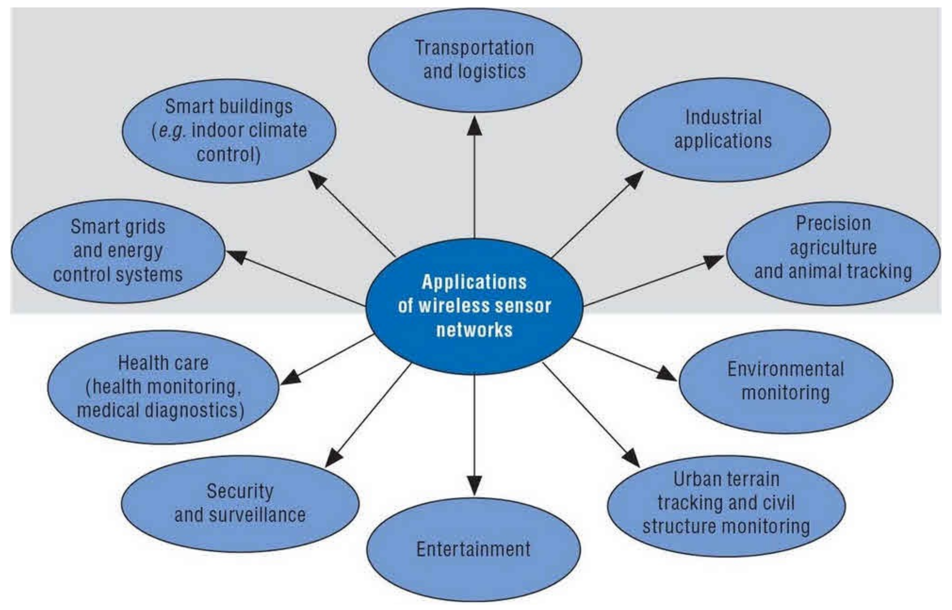


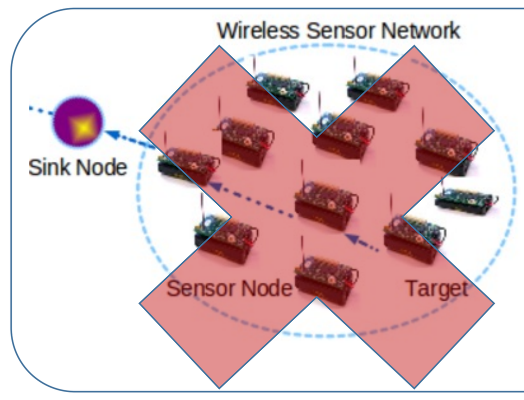
Published in IEEE Transactions on Signal Processing

UNIVERSITETET I AGDER — icassp 2022 Singapore


## Decentralized Inference in Sensor Networks

Wireless sensor networks enable diverse inference tasks in a great number of applications



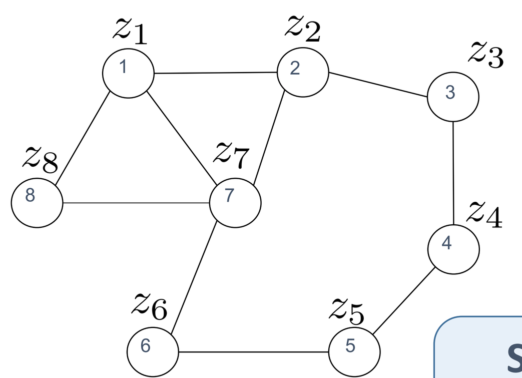




**Centralized Algorithms**

- Large energy consumption
- Single point of failure (sink node/central processor)
- Communication bottlenecks

Motivate → Decentralized Algorithms

## Goal: Decentralized Subspace Projection




Observed (graph) signal $\boldsymbol{z} = [z_1, \ldots\ldots, z_N]^T$

$$\boldsymbol{z} = \boldsymbol{\xi} + \boldsymbol{v}$$

Signal of interest $\boldsymbol{\xi} = \boldsymbol{U}_{\parallel}\boldsymbol{\alpha}$,

Noise $\boldsymbol{v} \in \mathbb{R}^N$.

$\boldsymbol{U}_{\parallel} \in \mathbb{R}^{N\times r}, r < N$

Orthonormal columns w.l.o.g.

**Subspace Projection**

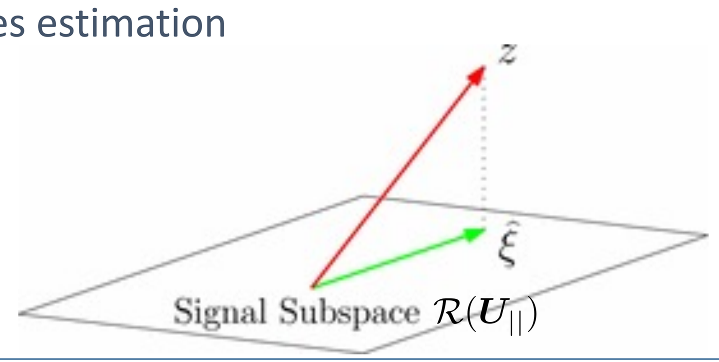
Given $\boldsymbol{z}$ and $\boldsymbol{U}_{\parallel}$, find $\boldsymbol{\xi}$.

**Example of subspace projection:** Least-squares estimation

$$\hat{\boldsymbol{\xi}} := [\hat{\xi}_1, \ldots, \hat{\xi}_N]^T = \boldsymbol{P}_{\boldsymbol{U}_{\parallel}}\boldsymbol{z}$$

$$\boldsymbol{P}_{\boldsymbol{U}_{\parallel}} = \boldsymbol{U}_{\parallel}\boldsymbol{U}_{\parallel}^T$$

Projector onto $\mathcal{R}(\boldsymbol{U}_{\parallel})$




## Contribution Relative to Prior Art

- **Maximum convergence rate**
  [Barbarossa et al. '09] [Insausti et al. '12]
  - ➢No memory
  - ➢Asymptotic convergence

Edge weights are designed

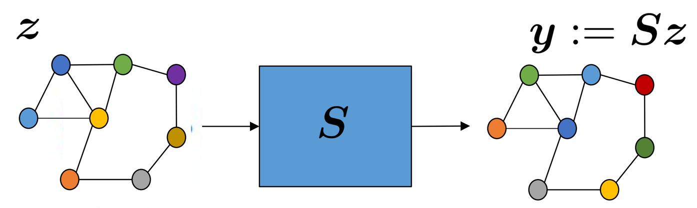
- **Distributed subspace projection via graph filters**
  [Sandryhaila et al. '14] [Safavi et al. '15]
  [Segarra et al. '17]
  - ➢Memory
  - ➢Convergence in finite number of local exchanges

Edge weights must be given

Contribution:
Weights for convergence in a nearly **minimal number of local exchanges**.

## Decentralized Projections via Graph Filters

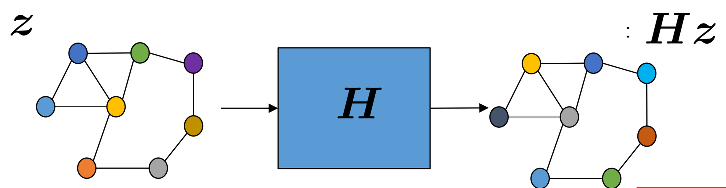
- Graph weight/shift matrix: matrix $\boldsymbol{S} \in \mathbb{R}^{N\times N}$ such that $(\boldsymbol{S})_{n,n'} = 0$ if $(n,n') \notin \mathcal{E}$ (Edge set)




$$(\boldsymbol{y})_n = \sum_{n'=1}^{N} (\boldsymbol{S})_{n,n'}(\boldsymbol{z})_{n'} = \sum_{n':(n,n')\in\mathcal{E}} (\boldsymbol{S})_{n,n'}(\boldsymbol{z})_{n'}$$

Admits decentralized computation

- **Graph filter** $\boldsymbol{H} := \sum_{l=0}^{L-1} c_l \boldsymbol{S}^l$ of order $L \leq N$




$\boldsymbol{z} \to \boldsymbol{S} \to \boldsymbol{S}\boldsymbol{z} \to \boldsymbol{S} \to \boldsymbol{S}^2\boldsymbol{z} \ldots \to \boldsymbol{S} \to \boldsymbol{S}^{L-1}\boldsymbol{z}$

Decentralized computation via L-1 neighbor exchanges

Subspace Projection: $\boldsymbol{y}_{SP} = \boldsymbol{P}_{\boldsymbol{U}_{\parallel}}\boldsymbol{z}$

Graph Filter: $\boldsymbol{y}_{GF} = \sum_{l=0}^{L-1} c_l \boldsymbol{S}^l \boldsymbol{z}$

Decentralized computation of projections via L-1 neighbor data exchanges if

$$\boldsymbol{y}_{SP} = \boldsymbol{y}_{GF} \quad \forall \boldsymbol{z}$$

## Problem Formulation

| Given | Find | Such that |
|---|---|---|
| • Subspace matrix $\boldsymbol{U}_{\parallel}$<br>• Edge set $\mathcal{E}$ | • Filter order $L$<br>• Shift matrix $\boldsymbol{S}$<br>• Filter coefficients<br>$\mathbf{c} := [c_0, c_1, \ldots, c_{L-1}]^\top$ | $\boldsymbol{y}_{SP} = \boldsymbol{y}_{GF} \quad \forall \boldsymbol{z}$ |

## Feasible Shifts → $\boldsymbol{U}_{\parallel}\boldsymbol{U}_{\parallel}^\top = \sum_{l=0}^{L-1} c_l \boldsymbol{S}^l$

- **Step 1:** decompose $\boldsymbol{S}$

$\mathcal{R}(\boldsymbol{S}_{\parallel}) = \mathcal{R}(\boldsymbol{U}_{\parallel})$, $\mathcal{R}(\boldsymbol{S}_{\perp}) \perp \mathcal{R}(\boldsymbol{U}_{\parallel})$, $\boldsymbol{F}_{\perp} = \boldsymbol{V}_{\perp}\boldsymbol{\Lambda}_{\perp}\boldsymbol{V}_{\perp}^\top$, $\boldsymbol{F}_{\parallel} = \boldsymbol{V}_{\parallel}\boldsymbol{\Lambda}_{\parallel}\boldsymbol{V}_{\parallel}^\top$

$$\boldsymbol{S} = \boldsymbol{S}_{\parallel} + \boldsymbol{S}_{\perp} = \boldsymbol{U}_{\parallel}\boldsymbol{F}_{\parallel}\boldsymbol{U}_{\parallel}^\top + \boldsymbol{U}_{\perp}\boldsymbol{F}_{\perp}\boldsymbol{U}_{\perp}^\top = [\boldsymbol{U}_{\parallel}\boldsymbol{V}_{\parallel}, \boldsymbol{U}_{\perp}\boldsymbol{V}_{\perp}]\begin{bmatrix}\boldsymbol{\Lambda}_{\parallel} & \\ & \boldsymbol{\Lambda}_{\perp}\end{bmatrix}\begin{bmatrix}\boldsymbol{V}_{\parallel}^\top\boldsymbol{U}_{\parallel}^\top \\ \boldsymbol{V}_{\perp}^\top\boldsymbol{U}_{\perp}^\top\end{bmatrix}$$

- **Step 2**: rewrite

$$\boldsymbol{U}_{\parallel}\boldsymbol{U}_{\parallel}^\top = [\boldsymbol{U}_{\parallel}\boldsymbol{V}_{\parallel}, \boldsymbol{U}_{\perp}\boldsymbol{V}_{\perp}]\begin{bmatrix}\boldsymbol{I} & \\ & \boldsymbol{0}\end{bmatrix}\begin{bmatrix}\boldsymbol{V}_{\parallel}^\top\boldsymbol{U}_{\parallel}^\top \\ \boldsymbol{V}_{\perp}^\top\boldsymbol{U}_{\perp}^\top\end{bmatrix}$$

- **Combine**

$$[\boldsymbol{U}_{\parallel}\boldsymbol{V}_{\parallel}, \boldsymbol{U}_{\perp}\boldsymbol{V}_{\perp}]\begin{bmatrix}\boldsymbol{I} & \\ & \boldsymbol{0}\end{bmatrix}\begin{bmatrix}\boldsymbol{V}_{\parallel}^\top\boldsymbol{U}_{\parallel}^\top \\ \boldsymbol{V}_{\perp}^\top\boldsymbol{U}_{\perp}^\top\end{bmatrix} = [\boldsymbol{U}_{\parallel}\boldsymbol{V}_{\parallel}, \boldsymbol{U}_{\perp}\boldsymbol{V}_{\perp}]\sum_{l=0}^{L-1} c_l \begin{bmatrix}\boldsymbol{\Lambda}_{\parallel}^l & \\ & \boldsymbol{\Lambda}_{\perp}^l\end{bmatrix}\begin{bmatrix}\boldsymbol{V}_{\parallel}^\top\boldsymbol{U}_{\parallel}^\top \\ \boldsymbol{V}_{\perp}^\top\boldsymbol{U}_{\perp}^\top\end{bmatrix}$$

$\boldsymbol{\Lambda}_{\parallel} = \text{diag}\{\lambda_1, \ldots, \lambda_r\}$

$\boldsymbol{\Lambda}_{\perp} = \text{diag}\{\lambda_{r+1}, \ldots, \lambda_N\}$

$$\Rightarrow \begin{bmatrix}\mathbf{1}_r \\ \mathbf{0}_{N-r}\end{bmatrix} = \begin{bmatrix}1 & \lambda_1 & \ldots & \lambda_1^{L-1} \\ 1 & \lambda_2 & \ldots & \lambda_2^{L-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \lambda_N & \ldots & \lambda_N^{L-1}\end{bmatrix}\begin{bmatrix}c_0 \\ c_1 \\ \vdots \\ c_{L-1}\end{bmatrix}$$

## Minimizing the Filter Order

$$\begin{bmatrix}\mathbf{1}_r \\ \mathbf{0}_{N-r}\end{bmatrix} = \begin{bmatrix}1 & \lambda_1 & \ldots & \lambda_1^{L-1} \\ 1 & \lambda_2 & \ldots & \lambda_2^{L-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \lambda_N & \ldots & \lambda_N^{L-1}\end{bmatrix}\begin{bmatrix}c_0 \\ c_1 \\ \vdots \\ c_{L-1}\end{bmatrix}$$

→ Minimum $L$ equals number of distinct eigenvalues $\{\lambda_n\}_{n=1}^N$ !!!

Non-convex

**Proposed criterion surrogates no. of different eigenvalues:**

$\|\boldsymbol{F}_{\parallel} \otimes \boldsymbol{I} - \boldsymbol{I} \otimes \boldsymbol{F}_{\parallel}\|_\star$ $\quad$ $\|\boldsymbol{F}_{\perp} \otimes \boldsymbol{I} - \boldsymbol{I} \otimes \boldsymbol{F}_{\perp}\|_\star$

$$\underset{\boldsymbol{F}_{\parallel},\boldsymbol{F}_{\perp},\boldsymbol{S}_{\parallel},\boldsymbol{S}_{\perp}}{\text{minimize}} \sum_{n=1}^{r}\sum_{n'=1}^{r}|\lambda_n - \lambda_{n'}| + \sum_{n=r+1}^{N}\sum_{n'=r+1}^{N}|\lambda_n - \lambda_{n'}|$$

$$\text{s.t.} \quad (\boldsymbol{S}_{\parallel} + \boldsymbol{S}_{\perp})_{n,n'} = 0 \;\text{ if }\; (n,n') \notin \mathcal{E}, \quad n,n' = 1, \ldots, N$$

$$\boldsymbol{F}_{\parallel} = \boldsymbol{F}_{\parallel}^\top, \; \boldsymbol{F}_{\perp} = \boldsymbol{F}_{\perp}^\top$$

$$\boldsymbol{S}_{\parallel} = \boldsymbol{U}_{\parallel}\boldsymbol{F}_{\parallel}\boldsymbol{U}_{\parallel}^\top, \; \boldsymbol{S}_{\perp} = \boldsymbol{U}_{\perp}\boldsymbol{F}_{\perp}\boldsymbol{U}_{\perp}^\top$$

$$\frac{\text{tr}(\boldsymbol{F}_{\parallel})}{r} = 1, \quad \frac{\text{tr}(\boldsymbol{F}_{\perp})}{N-r} \leq 1 - \epsilon$$

Promotes **sparsity** in the differences of eigenvalues

These two constraints can also be relaxed

**Convex!!!**

## Simulation Experiments

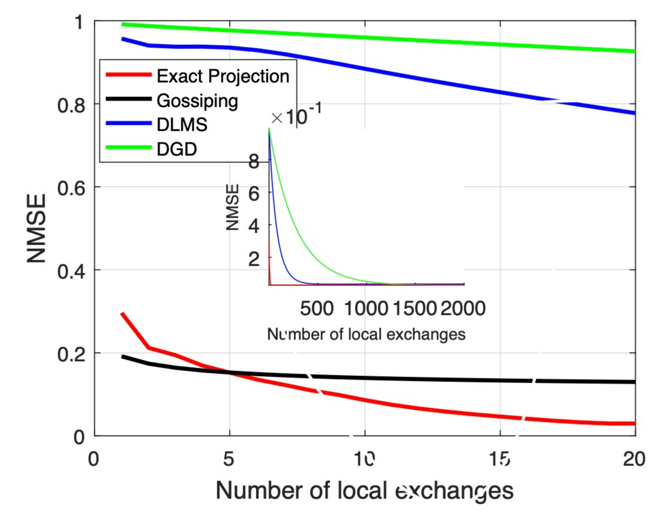



Fig. 4: NMSE as a function of the number of communications performed per node ($N = 20, r = 3, \beta = 5$, Erdős–Rényi graph with $p_{\text{miss}} = 0.6$, $\rho = 0.1$, $I_{\max} = 1000$, $\eta_{\perp} = 0.9, \eta_{\parallel} = 0.1, \epsilon = 0.1$).

- $\boldsymbol{z}$ drawn from a zero mean, unit variance Gaussian distribution.
- $\boldsymbol{U}_{\parallel}$ generated by applying Gram–Schmidt to an $N \times r$ matrix with independent and uniformly distributed (0,1) entries.
- Peformance metric

$$\text{NMSE}(\boldsymbol{H}_l) := \mathbb{E}\left[\|\boldsymbol{\xi} - \boldsymbol{H}_l\boldsymbol{z}\|_2^2\right] / \mathbb{E}\left[\|\boldsymbol{\xi}\|_2^2\right]$$

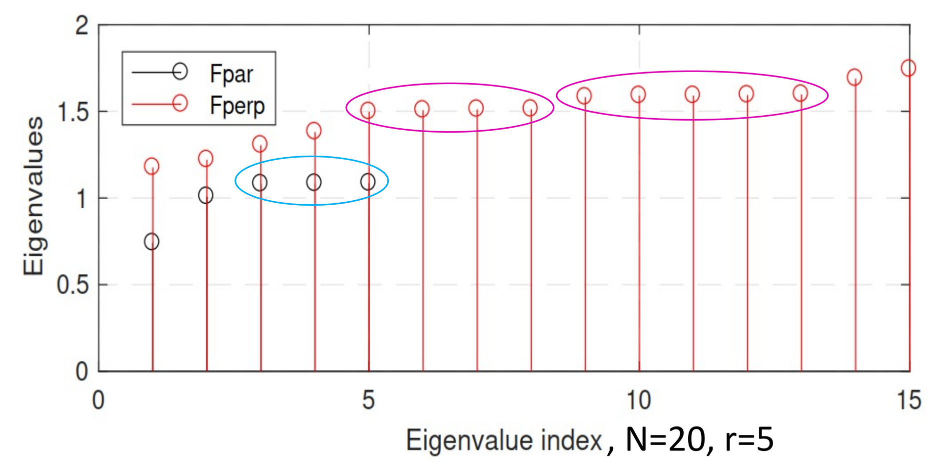
- Compared with:


[Gossiping] Barbarossa, G. Scutari, T. Battisti, "Distributed signal subspace projection algorithms with maximum convergence rate for sensor networks with topological constraints", ICASSP 2009.

[Rank-1] S. Segarra, A. G. Marques, and A. Ribeiro, "Optimal graph-filter design and applications to distributed linear network operators," IEEE Trans. Signal Process. 2017.

[DLMS] G. Mateos, I. D. Schizas, A. Ribeiro, and G. B. Giannakis, "Performance analysis of the consensus-based distributed LMS algorithm," EURASIP J. Advances Signal Process., Nov. 2009.

[DGD] L. Shi, L. Zhao, W. Song, G. Kamath, Y. Wu, and X. Liu, "Distributed least-squares iterative methods in large-scale networks: A survey," ZTE Commun., vol. 3, pp. 37–45, Aug. 2017.


## Contributions

Framework for subspace projection with graph filters

Novel convex relaxation approach to minimize the filter order

Approximate filters when the topology does not allow an exact filter

Theoretical analysis of the feasibility of a projection with a given topology

ADMM solvers