A Computationally Efficient Algorithm for Distributed Adaptive Signal Fusion Based on Fractional Programs

Cem Ates Musluoglu, Alexander Bertrand {cemates.musluoglu, alexander.bertrand}@esat.kuleuven.be

1. Fractional Programs

$$\mathbb{P}: \underset{X}{\text{minimize}}\ r(X) = \frac{f_1(X)}{f_2(X)}$$
$$\text{subject to } X \in \mathcal{S}$$

Dinkelbach's procedure

$$X^{i+1} \leftarrow \underset{X\in\mathcal{S}}{\text{argmin}}\ f_1(X) - \rho^i \cdot f_2(X)$$
$$\rho^{i+1} \leftarrow r(X^{i+1}) = f_1(X^{i+1})/f_2(X^{i+1})$$
$*$

- Iteratively solve auxiliary problems
- Convergence to a solution X^* of $\mathbb{P}$ under mild conditions

2. Fractional Programs for Spatial Filtering...

Trace ratio optimization (TRO):

$$\underset{X}{\text{minimize}}\ -\frac{\mathbb{E}[||X^T\mathbf{y}(t)||^2]}{\mathbb{E}[||X^T\mathbf{v}(t)||^2]}$$
$$\text{subject to } X^TX = I$$

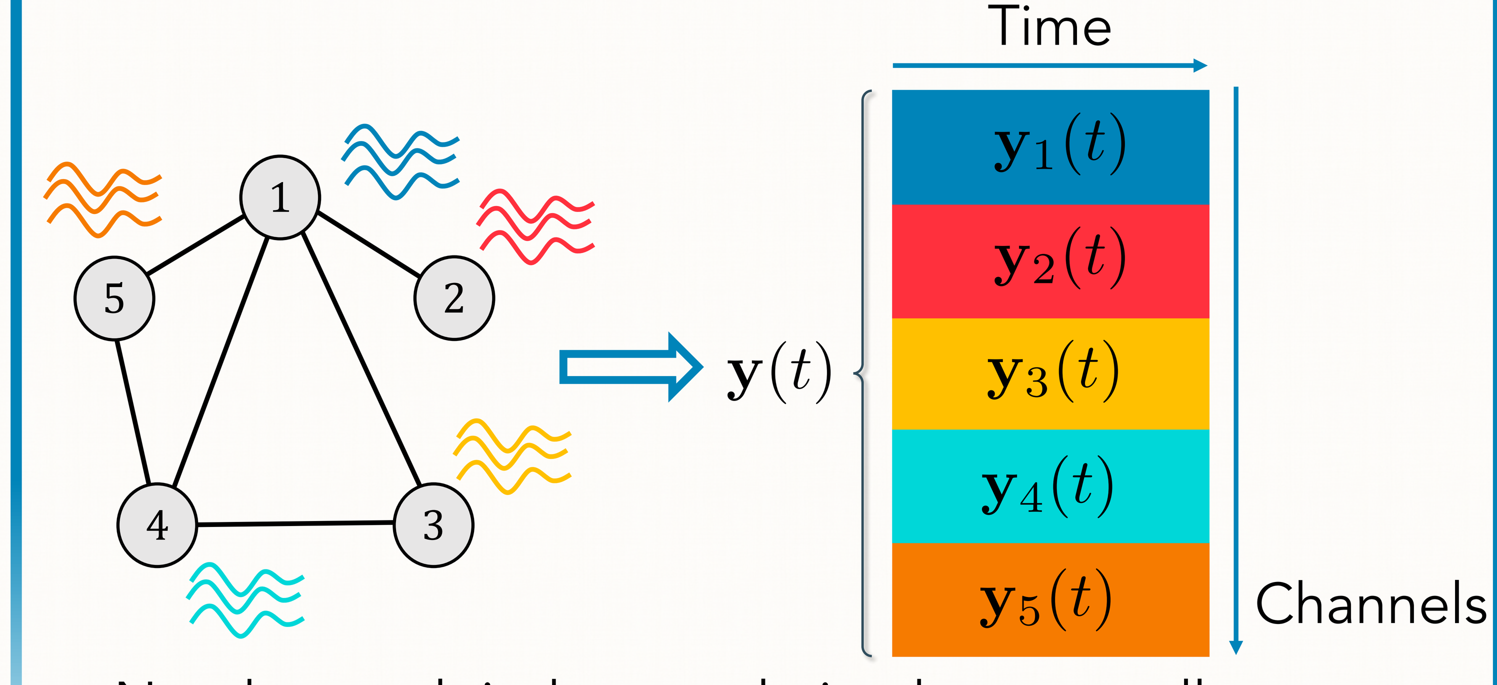
Regularized total least squares (RTLS):

$$\underset{\mathbf{x}\in\mathbb{R}^M}{\text{minimize}}\ \frac{\mathbb{E}[|\mathbf{x}^T\mathbf{y}(t)-d(t)|^2]}{1+\mathbf{x}^T\mathbf{x}}$$
$$\text{subject to } ||\mathbf{x}^TL||^2 \leq 1,$$

General form of spatial filtering problems with fractional objective:

$$\mathbb{P}: \underset{X}{\text{minimize}}\ \frac{f_1(X)}{f_2(X)} = \frac{F_1(X^T\mathbf{y}(t))}{F_2(X^T\mathbf{y}(t))}$$
$$\text{subject to } X \in \mathcal{S}$$

... in a Distributed Setting



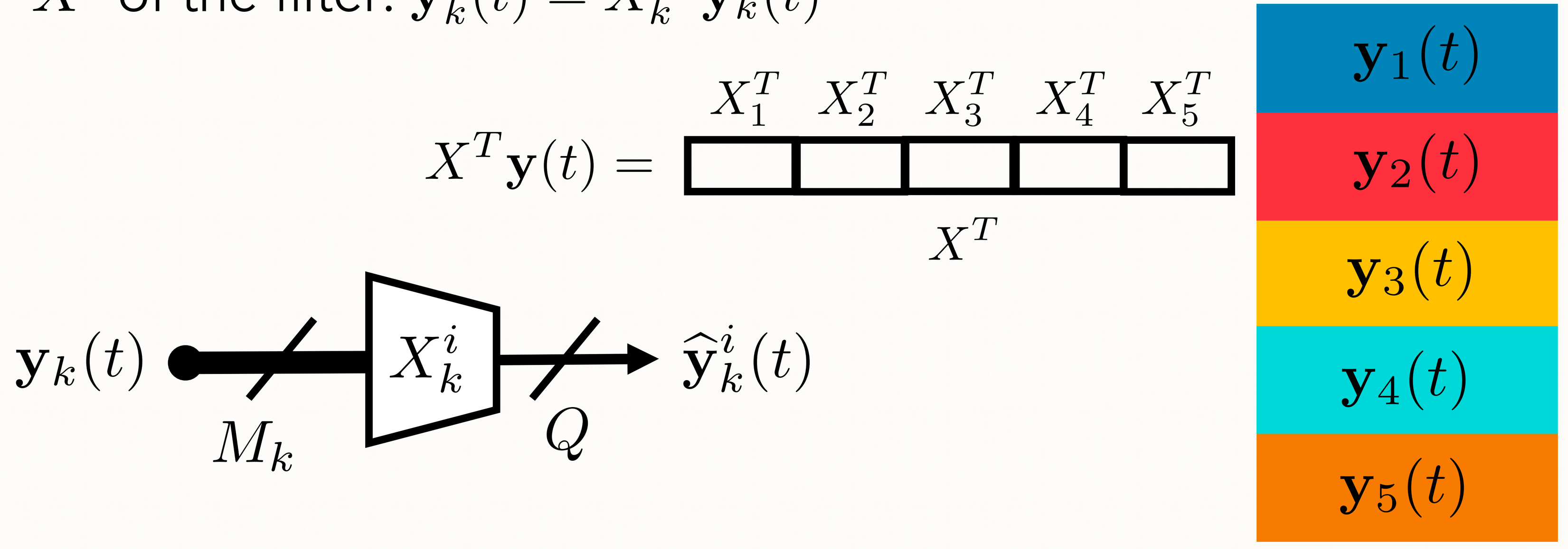
Need to exploit the correlation between all channels of $\mathbf{y}$ but centralization is too costly

Goal

Solve spatial filtering problems with fractional objective without centralizing the data

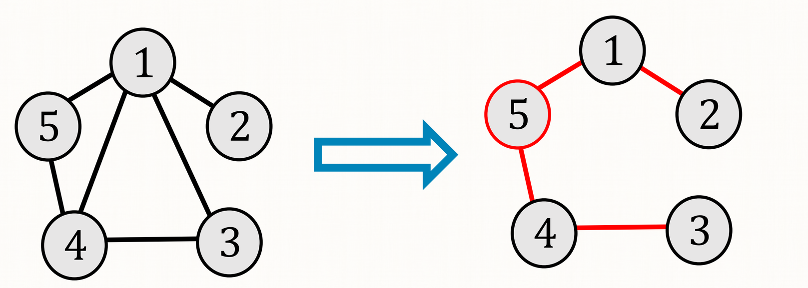
3. Distributed Data Flow

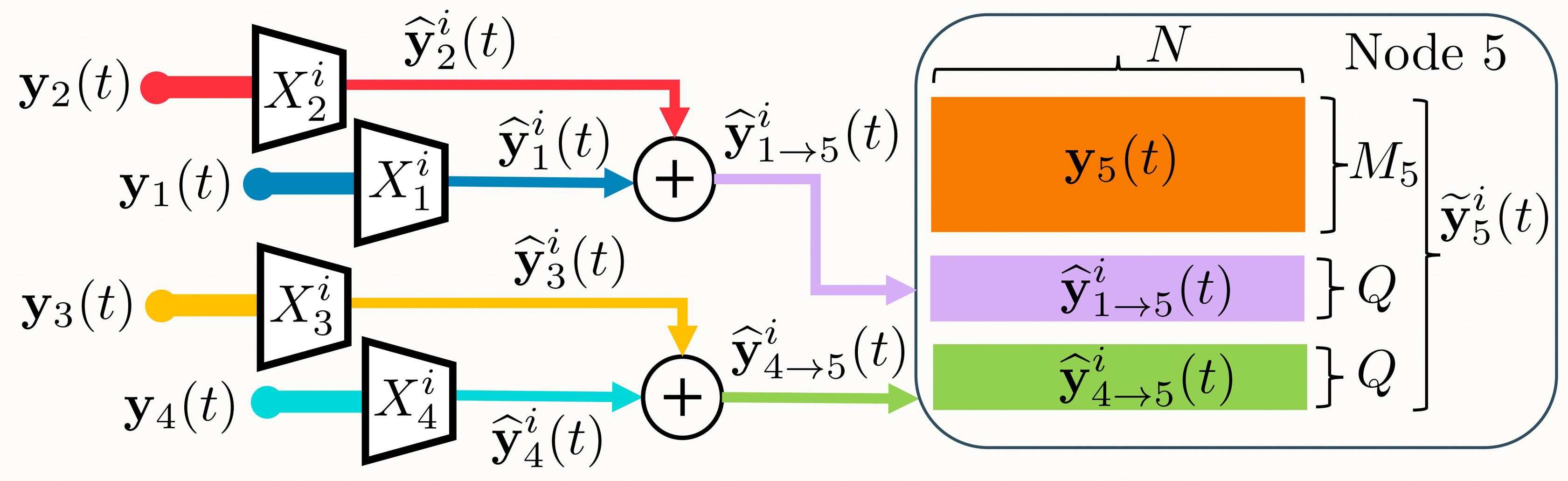
- Compress signals measured at nodes k using current estimate X^i of the filter: $\widehat{\mathbf{y}}_k^i(t) = X_k^{iT}\mathbf{y}_k(t)$



- Send N samples of compressed signals towards the updating node q after pruning the graph:

Example with $q=5$:





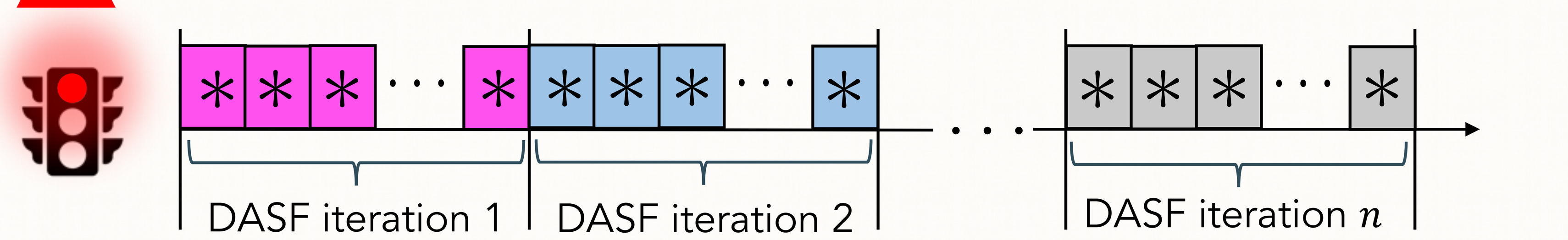
4. Distributed Adaptive Signal Fusion

- DASF algorithm[1,2]: At node q build and solve local version of the network-wide problem using $\widetilde{\mathbf{y}}_q^i$ instead of $\mathbf{y}$

$$\boxed{\widetilde{X}_q^{i+1} \leftarrow \underset{\widetilde{X}_q\in\widetilde{\mathcal{S}}_q}{\text{argmin}}\ \frac{F_1(\widetilde{X}_q^T\widetilde{\mathbf{y}}_q^i(t))}{F_2(\widetilde{X}_q^T\widetilde{\mathbf{y}}_q^i(t))}}$$

- Repeat process for other updating nodes and new signal samples

DASF is computationally expensive for fractional programs

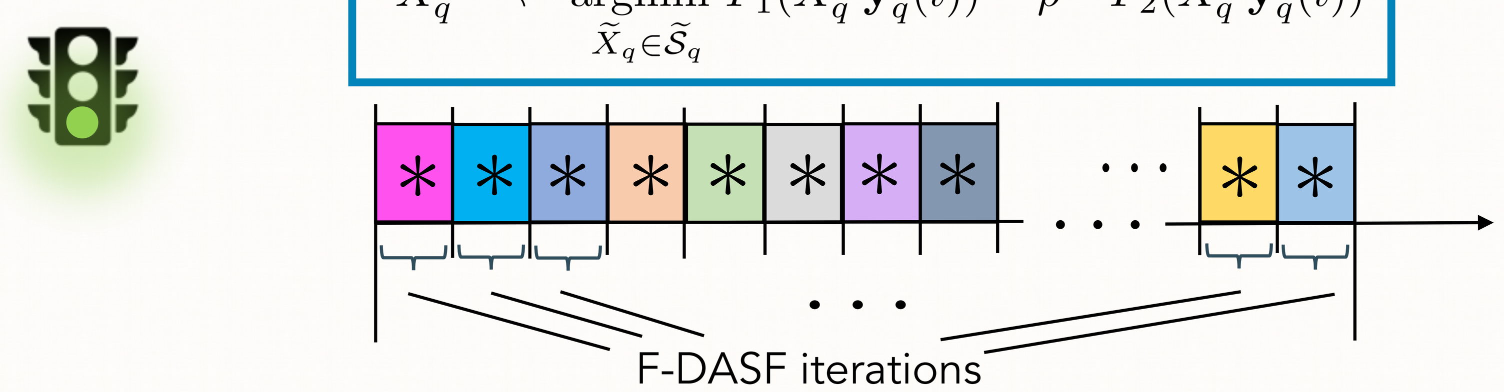


$*$: Dinkelbach iteration

Proposed method: Fractional-DASF

- **Interleave steps** of Dinkelbach and DASF
- Apply **only one step of Dinkelbach** at each updating node:

$$\boxed{\widetilde{X}_q^{i+1} \leftarrow \underset{\widetilde{X}_q\in\widetilde{\mathcal{S}}_q}{\text{argmin}}\ F_1(\widetilde{X}_q^T\widetilde{\mathbf{y}}_q^i(t)) - \rho^i \cdot F_2(\widetilde{X}_q^T\widetilde{\mathbf{y}}_q^i(t))}$$

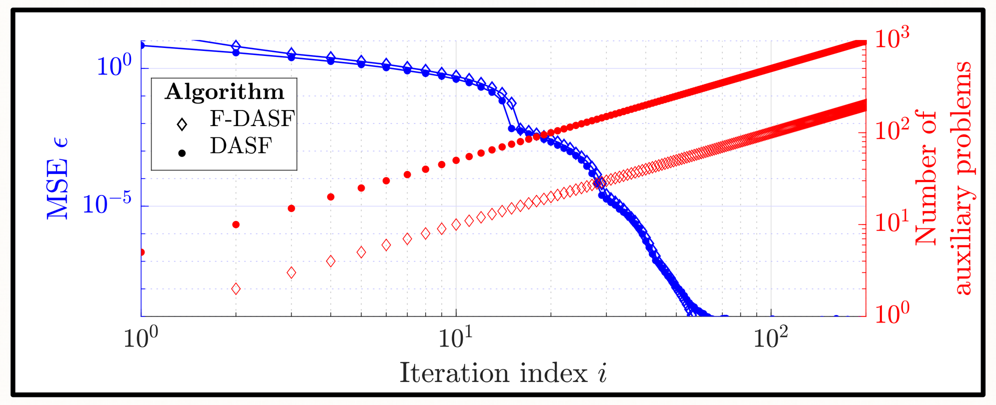


5. Results

- F-DASF also converges to a solution X^* of $\mathbb{P}$ with the same convergence rate as the DASF algorithm
- F-DASF requires significantly less computational power at each node

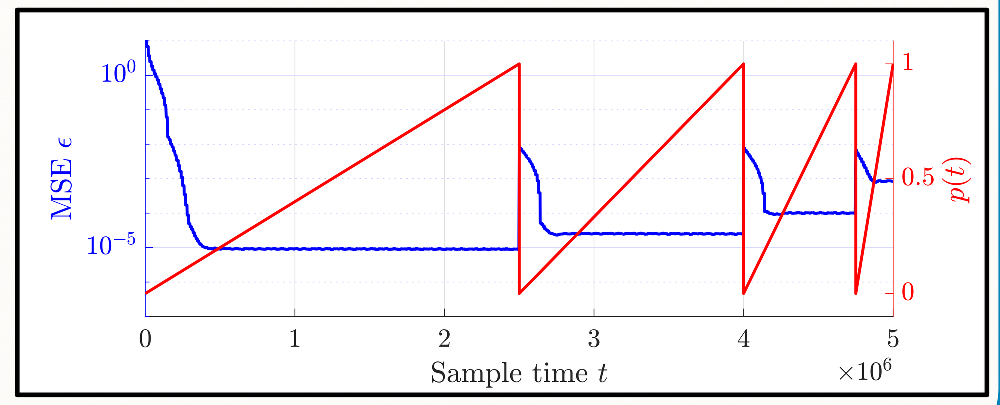
Stationary signal setting

$5\times$ less computations compared to DASF for the RTLS problem



Adaptive setting

- Statistics of $\mathbf{y}$ change with p
- Slow changes tracked, abrupt ones corrected



[1] C. A. Musluoglu, and A. Bertrand, "A unified algorithmic framework for distributed adaptive signal and feature fusion problems—Part I: Algorithm derivation," *IEEE Transactions on Signal Processing*, 2023. DOI: https://doi.org/10.1109/TSP.2023.3275272

[2] C. A. Musluoglu, C. Hovine, and A. Bertrand, "A Unified Algorithmic Framework for Distributed Adaptive Signal and Feature Fusion Problems--Part II: Convergence Properties," *IEEE Transactions on Signal Processing*, 2023. DOI: https://doi.org/10.1109/TSP.2023.3275273