KU LEUVEN

STADIUS
Center for Dynamical Systems, Signal Processing and Data Analytics

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

ICASSP 2023
4 - 10 JUNE, RHODES ISLAND, GREECE

Cem Ates Musluoglu and Alexander Bertrand

Presentation by Cem Ates Musluoglu

cemates.musluoglu@esat.kuleuven.be

European Research Council
Established by the European Commission

Fractional programs

$$\underset{X}{\text{minimize}}\; r(X) = \frac{f_1(X)}{f_2(X)}$$

$$\text{subject to } X \in \mathcal{S}$$

Dinkelbach's procedure

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

- Select ρ^0 arbitrarily

$$X^{i+1} \leftarrow \underset{X \in \mathcal{S}}{\operatorname{argmin}}\ f_1(X) - \rho^i f_2(X).$$

$$\rho^{i+1} \leftarrow r(X^{i+1}) = f_1(X^{i+1})/f_2(X^{i+1}).$$

- Repeat these steps by incrementing i: Dinkelbach's iterative procedure

- Convergence to ρ^* and X^* under mild conditions (e.g., compact constraint set)

Fractional programs in spatial filtering

Goal: Exploit spatial coherence between channels of $\boldsymbol{y}$ by optimally combining them using a linear filter X applied to $\boldsymbol{y}$: $X^T\boldsymbol{y}$

- General form:

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

- Trace ratio optimization

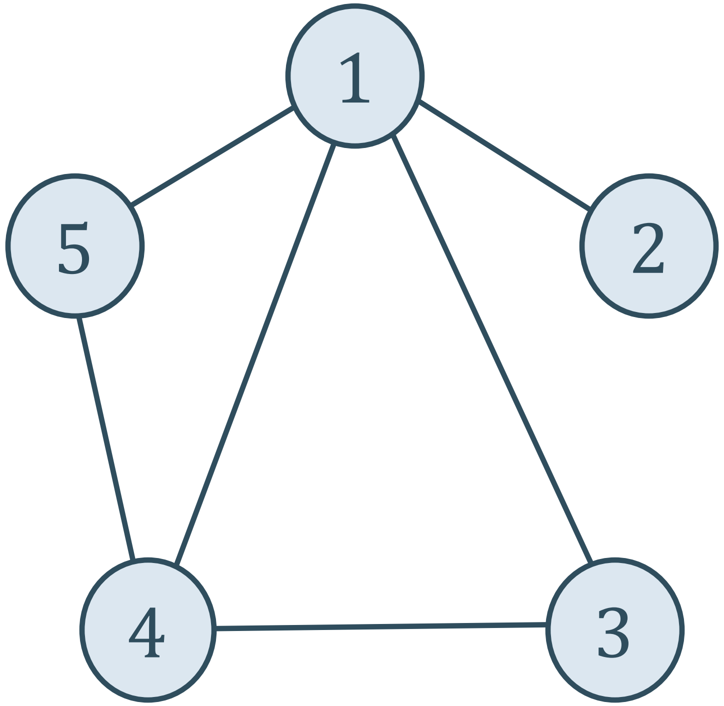
$$\underset{X}{\text{minimize}} \ -\frac{\mathbb{E}[||X^T\mathbf{y}(t)||^2]}{\mathbb{E}[||X^T\mathbf{v}(t)||^2]}$$
$$\text{subject to } X^TX = I$$

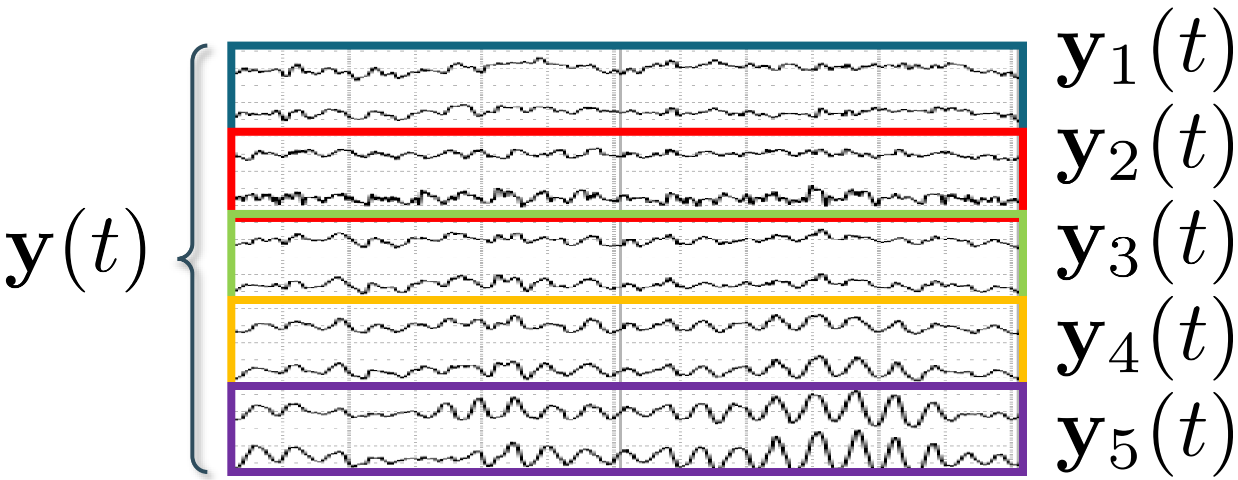
- Regularized total least squares

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

Distributed spatial filtering setting

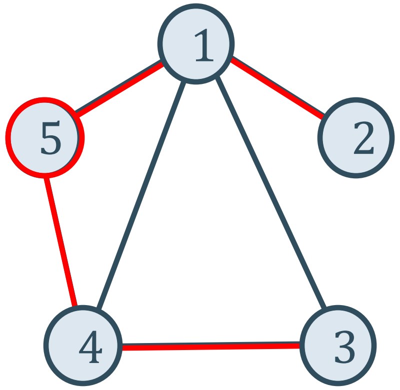
Each node k is a local sensor array measuring <u>multi-channel</u> signal $\boldsymbol{y}_k(t)$



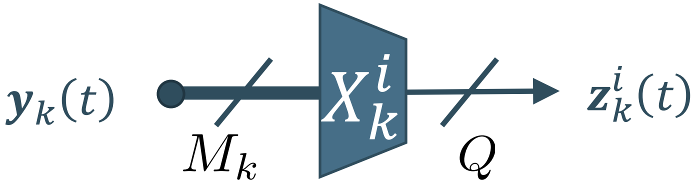


Solve fractional program without data centralization

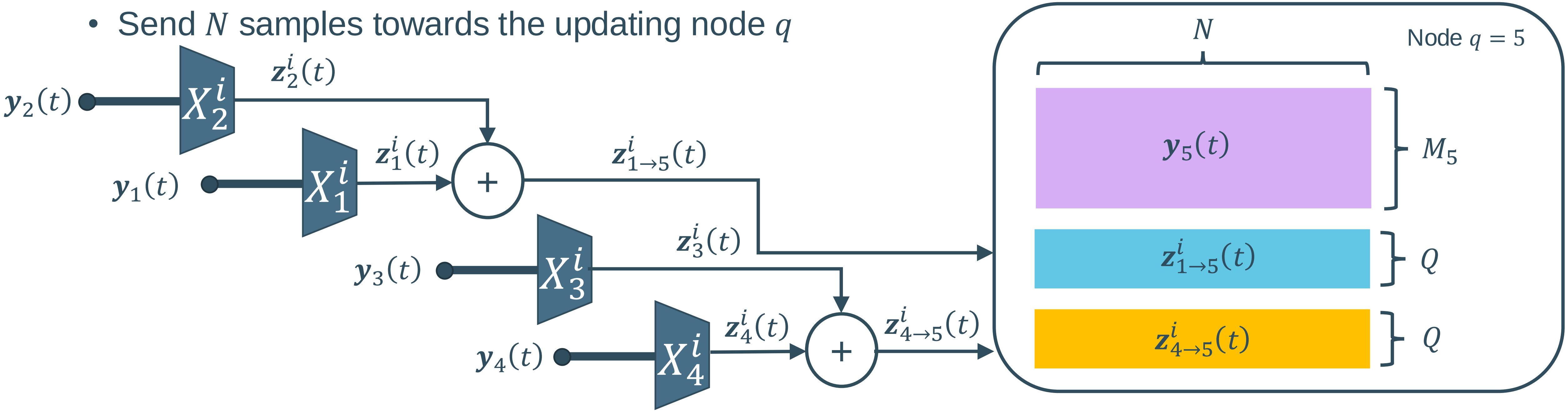
The DASF framework [1] (1/2)



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



- Send N samples towards the updating node q



[1] C. A. Musluoglu and A. Bertrand, "A unified algorithmic framework for distributed adaptive signal and feature fusion problems — part I: Algorithm derivation," Submitted for publication - preprint available at https://arxiv.org/abs/2208.08867

The DASF framework [1] (2/2)

- At node q, build a **compressed version of the network-wide problem** using the available local data and **solve it** to obtain new estimate X^{i+1}.

$$\begin{bmatrix} \boldsymbol{y}_5(t) \\ \boldsymbol{z}^i_{1\to5}(t) \\ \boldsymbol{z}^i_{4\to5}(t) \end{bmatrix} = \widetilde{\boldsymbol{y}}_5(t)$$

Local problem

$$\begin{aligned} \underset{\widetilde{X}_q}{\text{minimize}} \quad & \frac{F_1(\widetilde{X}_q^T \widetilde{\mathbf{y}}_q(t))}{F_2(\widetilde{X}_q^T \widetilde{\mathbf{y}}_q(t))} \\ \text{subject to} \quad & \widetilde{X}_q \in \widetilde{\mathcal{S}}_q \end{aligned}$$

Network-wide problem

$$\begin{aligned} \underset{X}{\text{minimize}} \quad & \frac{F_1(X^T \mathbf{y}(t))}{F_2(X^T \mathbf{y}(t))} \\ \text{subject to} \quad & X \in \mathcal{S} \end{aligned}$$

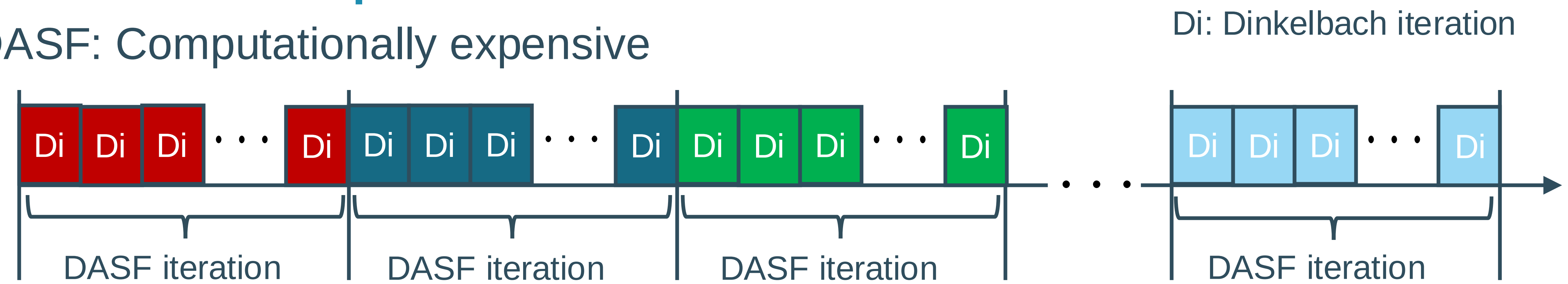
- Repeat for other nodes with a new batch of samples.

- Convergence to X^*, the global solution of the centralized problem (for any network topology) [1].

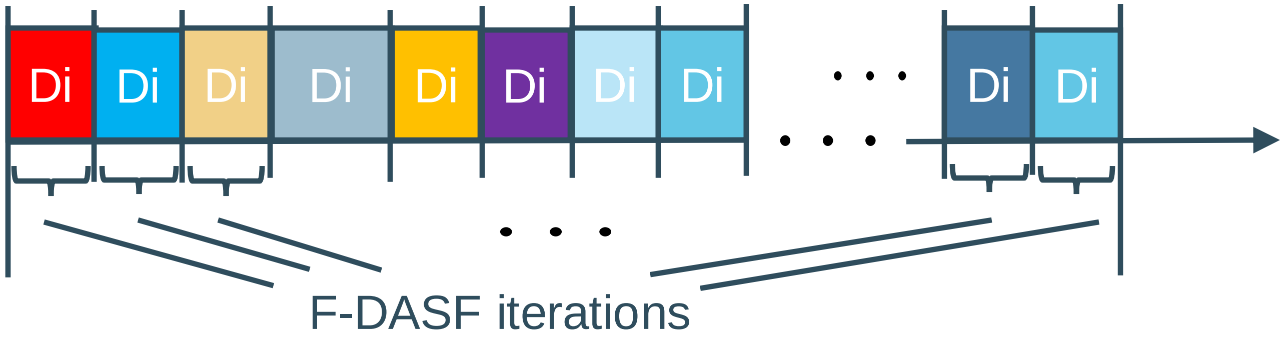
[1] C. A. Musluoglu and A. Bertrand, “A unified algorithmic framework for distributed adaptive signal and feature fusion problems — part I: Algorithm derivation,” Submitted for publication - preprint available at https://arxiv.org/abs/2208.08867

F-DASF: Proposed method

- DASF: Computationally expensive



- Proposed solution: Interleave steps of DASF and Dinkelbach
- **F-DASF: Partially solve** the local fractional program: **Apply only one step of Dinkelbach** at each updating node



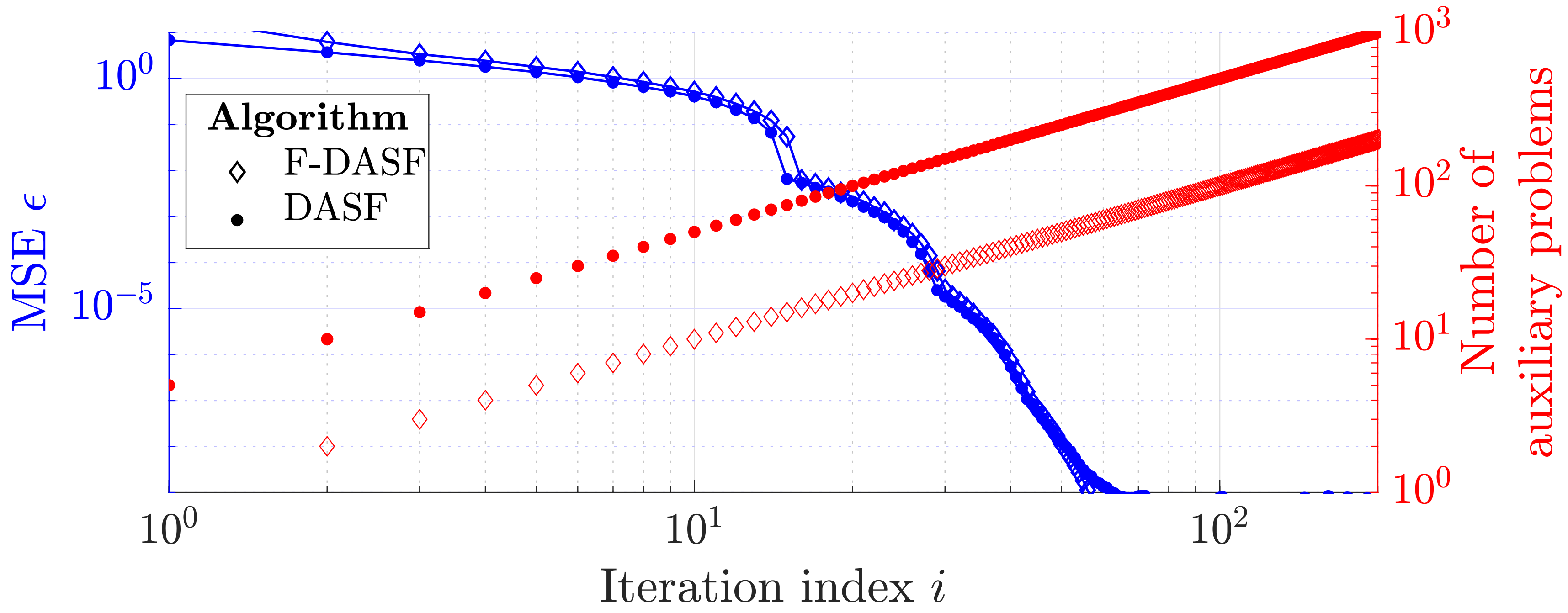
Results – Stationary setting

Regularized total least squares:

$$\min_{\mathbf{x}\in\mathbb{R}^M} \frac{\mathbb{E}[|\mathbf{x}^T\mathbf{y}(t) - d(t)|^2]}{1+\mathbf{x}^T\mathbf{x}} = \frac{\mathbf{x}^T R_{\mathbf{yy}}\mathbf{x} - 2\mathbf{x}^T\mathbf{r}_{\mathbf{y}d} + r_{dd}}{1+\mathbf{x}^T\mathbf{x}}$$
$$\text{s. t. } ||\mathbf{x}^T L||^2 \leq 1,$$

$$\epsilon(\mathbf{x}^i) = \frac{||\mathbf{x}^i - \mathbf{x}^*||^2}{||\mathbf{x}^*||^2}$$

Similar convergence rate for ~5 times less computation



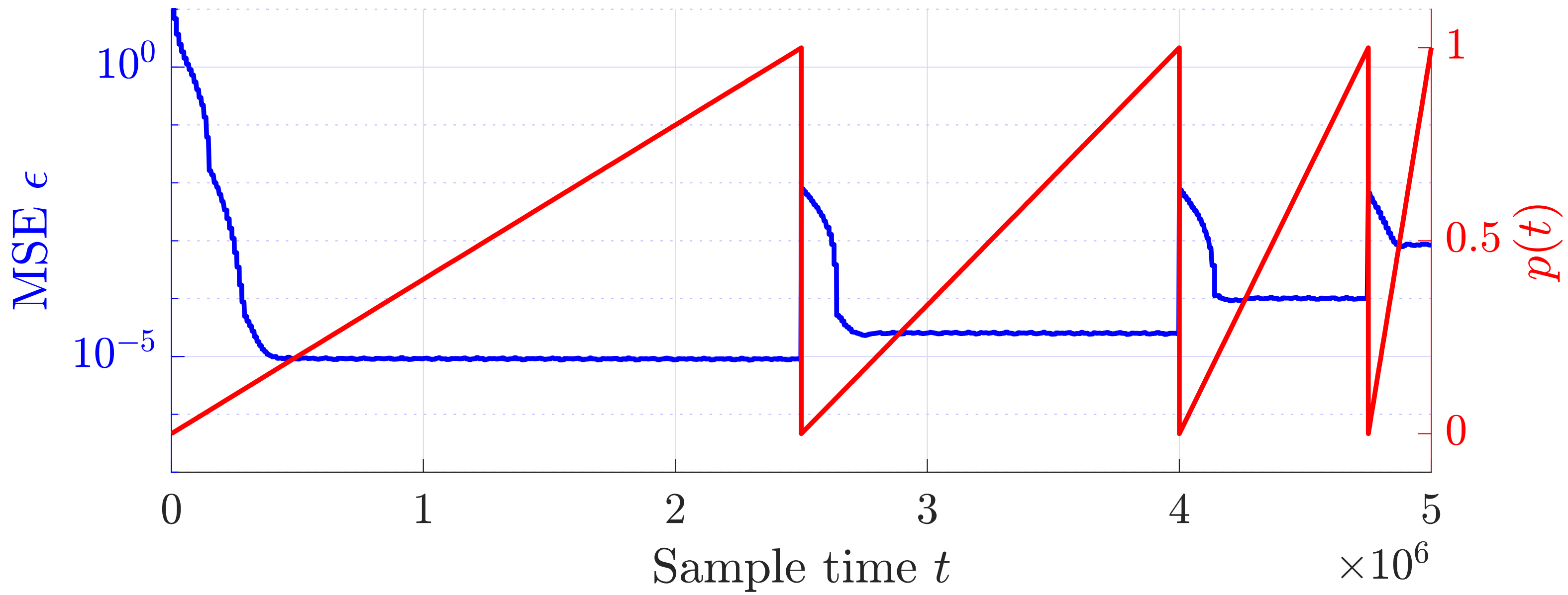
Results – Adaptive setting

Regularized total least squares:

$$\min_{\mathbf{x}\in\mathbb{R}^M} \frac{\mathbb{E}[|\mathbf{x}^T\mathbf{y}(t) - d(t)|^2]}{1+\mathbf{x}^T\mathbf{x}}$$
$$\text{s. t. } ||\mathbf{x}^T L||^2 \leq 1,$$

$$\epsilon(i) = \frac{||\mathbf{x}^i - \mathbf{x}^{*i}||^2}{||\mathbf{x}^{*i}||^2}$$

Able to track changes in statistics of signals (statistics of $\boldsymbol{y}$ dependent on p)



Conclusion

- For fractional programs, the original DASF algorithm is computationally expensive

- F-DASF applies a single step of Dinkelbach's procedure at each node, and achieves similar convergence rates as DASF

Future work:

- Detailed convergence analysis

Thank you