



Distributed spatial filtering and feature fusion



Alexander Bertrand, Cem Ates Musluoglu and Charles Hovine

Download slides: <https://tinyurl.com/EUSIPCO23>



European Research Council
Established by the European Commission

Outline

I- Spatial filtering and distributed signal processing

II- Towards a generic meta-algorithm for distributed data-driven spatial filtering

- II.A- Preliminaries
- II.B- The DASF framework

III- Fully-connected DASF algorithm (FC-DASF)

- III.A- Algorithm derivation
- III.B- Technical properties

IV- Topology-independent DASF (TI-DASF)

V- The DASF toolbox

VI- Extensions

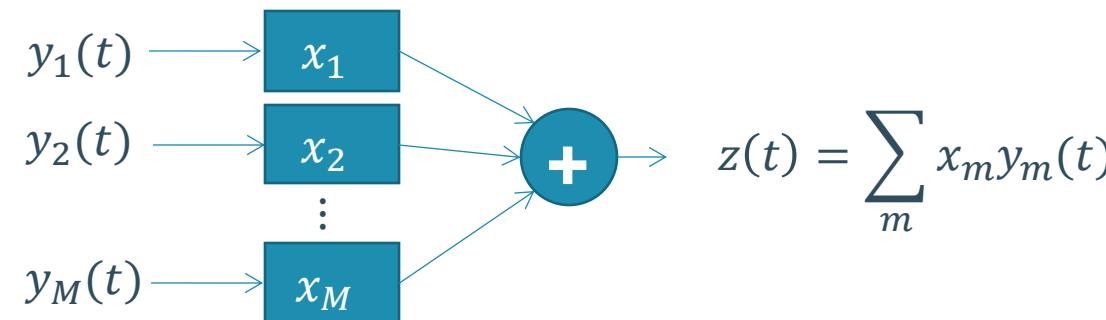
I- Spatial filtering and distributed signal processing

Preliminaries

Spatial Filtering

- **Spatial filter:** linearly combine channels of a multi-channel signal $\mathbf{y}(t) = [y_1(t) \ y_2(t) \ \dots \ y_M(t)]^T$

- **MISO:** $z(t) = \mathbf{x}^T \mathbf{y}(t)$ with $\mathbf{x} \in R^M \rightarrow M$ input channels, 1 output channel



- **MIMO:** $\mathbf{z}(t) = \mathbf{X}^T \mathbf{y}(t)$ with $\mathbf{X} \in R^{M \times Q} \rightarrow M$ input channels, Q output channels

Data-Driven Spatial Filtering: generic representation (*)

- **Data-driven:** optimize X as a function of the sensor data (statistics)

$$\begin{aligned} & \underset{X}{\text{minimize}} \quad f(X^T \mathbf{y}(t)) \\ & \text{subject to } g_j(X^T \mathbf{y}(t)) = 0, \\ & \qquad \qquad h_j(X^T \mathbf{y}(t)) \leq 0 \end{aligned}$$

→ We optimize X , but are generally (more) interested in $\mathbf{z}(t) = X^T \mathbf{y}(t)$.

Example: minimum mean squared error (MMSE),
a.k.a. multi-channel Wiener filter (MWF)

$$\min \mathbb{E}[\|\mathbf{d}(t) - X^T \mathbf{y}(t)\|_F^2]$$

Example: max-SNR filtering

$$\max \frac{\mathbb{E}[\|X^T \mathbf{y}(t)\|_F^2]}{\mathbb{E}[\|X^T \mathbf{n}(t)\|_F^2]}$$

Example: linearly constrained minimum variance (LCMV) beamformer

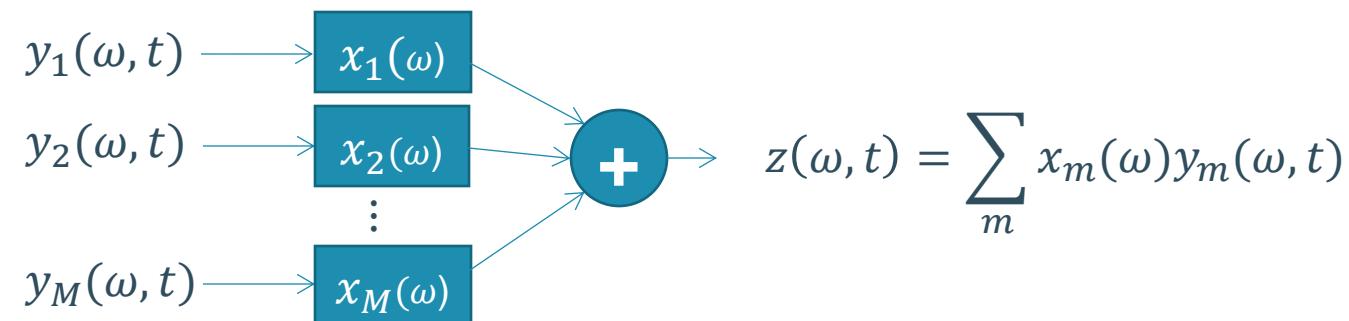
$$\begin{aligned} & \min \mathbb{E}[\|X^T \mathbf{y}(t)\|_F^2] \\ & \text{s.t. } X^T B = H \end{aligned}$$

Example: principal component analysis (PCA)

$$\begin{aligned} & \max \mathbb{E}[\|X^T \mathbf{y}(t)\|_F^2] \\ & \text{s.t. } X^T X = I \end{aligned}$$

Data-driven Spatial Filtering

- Typical assumption: $y(t)$ is ergodic and (short-term) stationary
- If statistics of $y(t)$ change (slowly) over time → **adaptive** spatial filter
- Extensions to convolutive spatio-temporal filter via time domain or Short-Time Fourier Transform domain
(abstracted here)



Wireless Sensor Networks

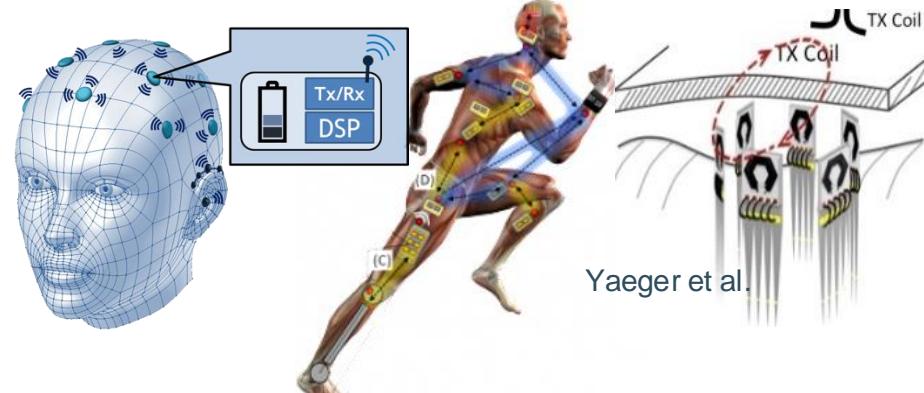
Wireless Sensor Network:

A network of inter-connected devices equipped with **sensing, computing and communication** capabilities.

Microphone networks (a.k.a. acoustic sensor networks)



Physiological sensor networks & body-area networks



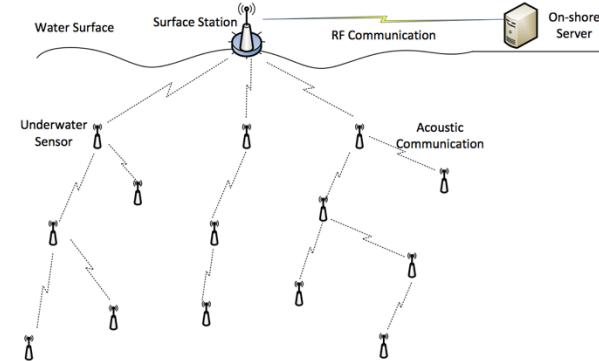
And more...

(radar, environmental monitoring, ...)



Camera networks

Under-water sensing



structural health monitoring



Smart homes & Ambient assisted living

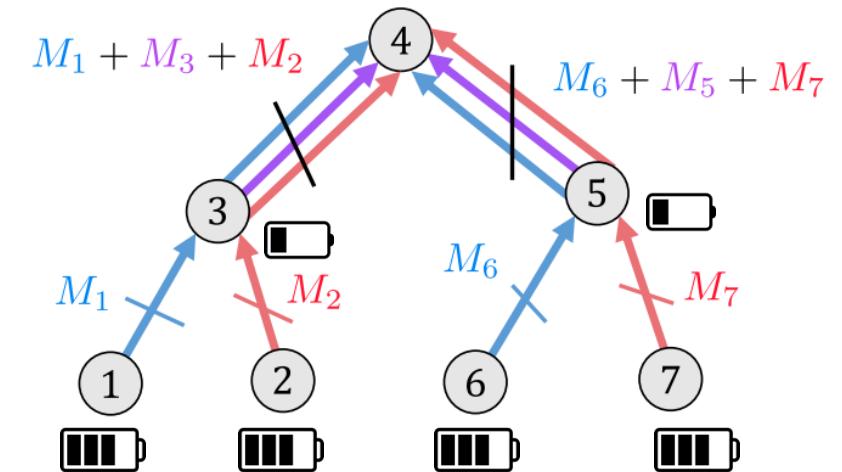


Distributed Signal Processing

Motivation

Data centralization in a fusion center:

- Simple, use of off-the-shelf SP algorithms
- Wireless transmission of raw sensor data (impossible?)
- Large computational requirement at the fusion center
- Single point of failure



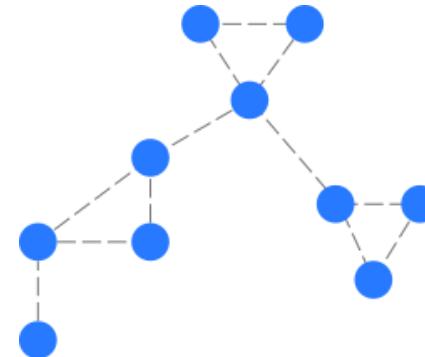
Distributed Signal Processing

Motivation

Distributed processing:

No Fusion Center – All nodes equal

- Scalable / compressive data aggregation
- Computational burden is shared
- Robust against node failure
- SP algorithm design is tedious

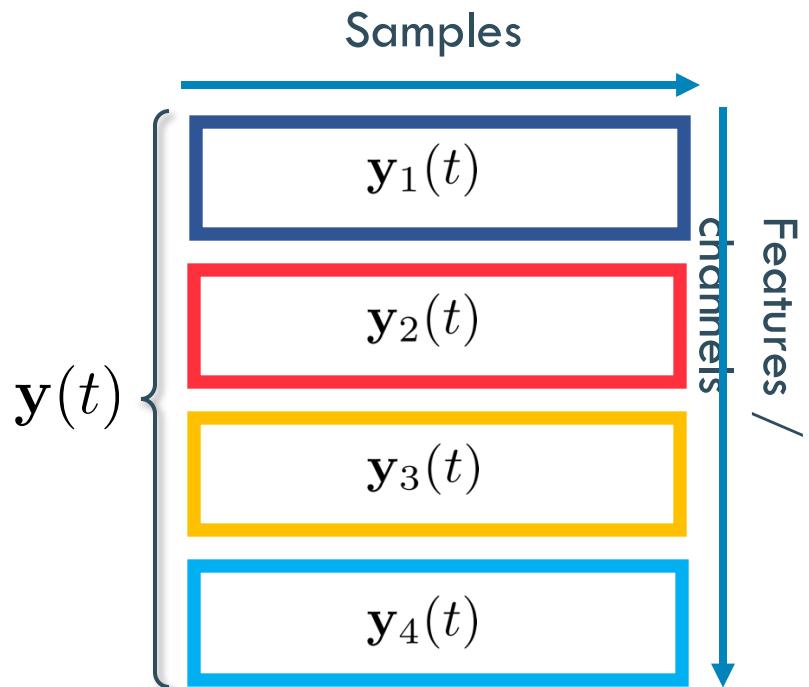


Distributed Signal Processing

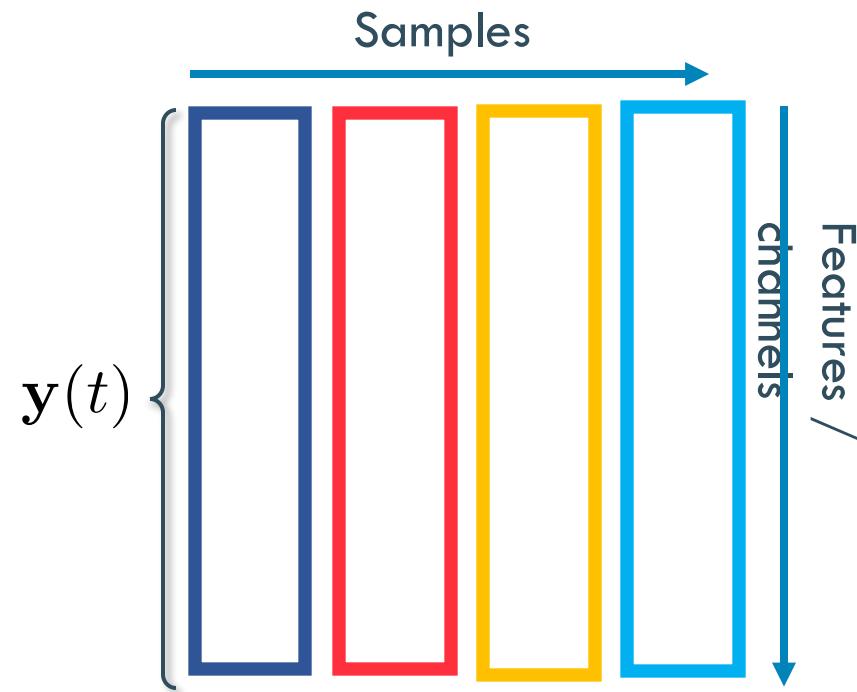
Problem Taxonomy

Feature-split

(spatial filtering: features = channels)



Sample-split



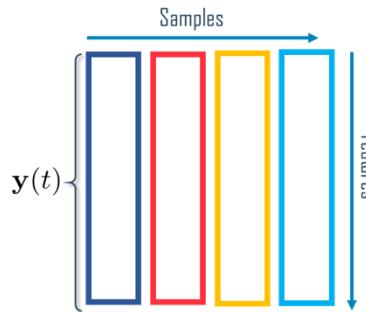
Distributed Signal Processing

Problem Taxonomy

- **Sample-split** typically leads to a **node-separable objective** with a **joint variable X**

$$\min_X \sum_k f_k(X)$$

!! f_k only depends on data from node k !!



- = focus of most distributed SP ‘workhorse’ algorithms
(consensus, diffusion, ADMM, PDMM, gossip, ...)

- Typical strategy:
$$\min_X \sum_k f_k(X_k)$$

s.t. $X_i = X_j \forall i, j$



Only communicate intermediate estimates
 X_k^i (usually no sensor data)

Distributed Signal Processing

Main challenge

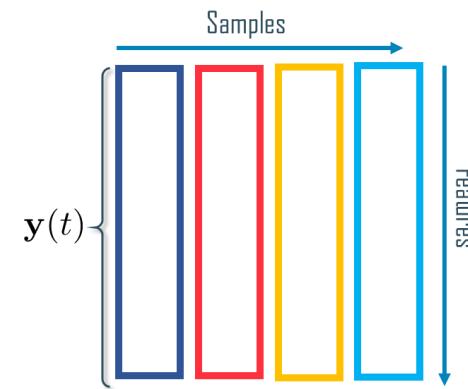
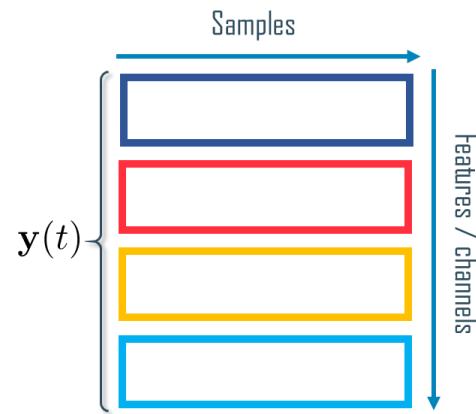
Objective is *not* node-separable

$$\min_X f(X^T \mathbf{y}(t))$$



node-separable objective

$$\min_X \sum_k f_k(X)$$



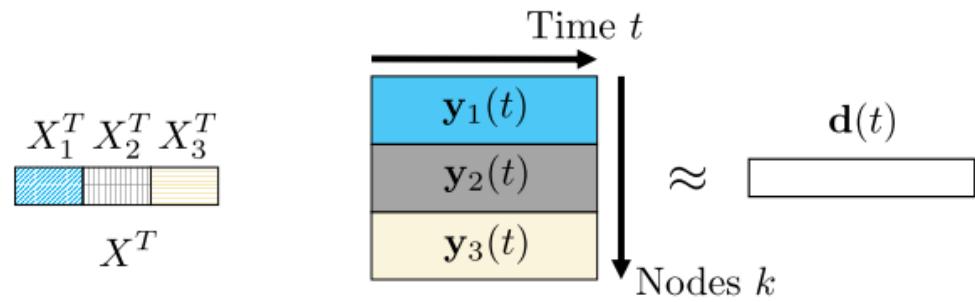
Distributed Signal Processing

Main challenge

Node-separable argument

MMSE example

$$\sum_t \|\mathbf{d}(t) - \sum_k X_k^T \mathbf{y}_k(t)\|^2$$

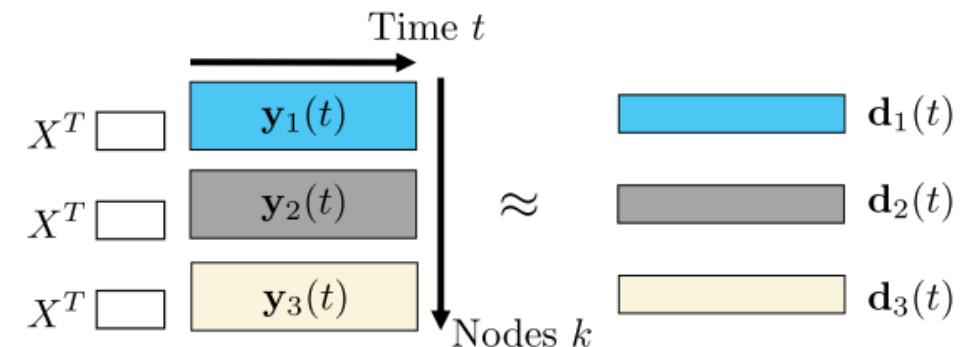


Interest is (usually) in output signal $X^T \mathbf{y}(t)$

Node-separable objective

MMSE example

$$\sum_k \sum_t \|\mathbf{d}_k(t) - X^T \mathbf{y}_k(t)\|^2$$



Interest is (usually) in estimated parameter \hat{X}

Distributed Signal Processing

Main challenge

Feature / channel split: how to deal with **inter-node signal covariance?**

$$E [\|X^T \mathbf{y}(t)\|_F^2] = \text{tr}(X^T E [\mathbf{y}(t) \mathbf{y}^T(t)] X) = \text{tr}(X^T R_{\mathbf{y}\mathbf{y}} X)$$

$$E[\mathbf{y}\mathbf{y}^T] = R_{\mathbf{y}\mathbf{y}} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & R_{14} \\ R_{21} & R_{22} & R_{23} & R_{24} \\ R_{31} & R_{32} & R_{33} & R_{34} \\ R_{41} & R_{42} & R_{43} & R_{44} \end{bmatrix}$$

Inter-node covariance

Node-specific covariance

Distributed Signal Processing

Main challenge

Feature-split

(spatial filtering: features = channels)

$$R_{yy} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & R_{14} \\ R_{21} & R_{22} & R_{23} & R_{24} \\ R_{31} & R_{32} & R_{33} & R_{34} \\ R_{41} & R_{42} & R_{43} & R_{44} \end{bmatrix}$$

None of the nodes observes the network-wide covariance matrix R_{yy}

Sample-split

$$\begin{bmatrix} S_{11} & S_{12} & S_{13} & S_{14} \\ S_{21} & S_{22} & S_{23} & S_{24} \\ S_{31} & S_{32} & S_{33} & S_{34} \\ S_{41} & S_{42} & S_{43} & S_{44} \end{bmatrix}$$



$$\begin{bmatrix} S_{11} & S_{12} & S_{13} & S_{14} \\ S_{21} & S_{22} & S_{23} & S_{24} \\ S_{31} & S_{32} & S_{33} & S_{34} \\ S_{41} & S_{42} & S_{43} & S_{44} \end{bmatrix}$$

$$\begin{bmatrix} S_{11} & S_{12} & S_{13} & S_{14} \\ S_{21} & S_{22} & S_{23} & S_{24} \\ S_{31} & S_{32} & S_{33} & S_{34} \\ S_{41} & S_{42} & S_{43} & S_{44} \end{bmatrix}$$

- Each node observes the **full covariance matrix** R_{yy}
- Aggregated sample covariance can be obtained with an in-network sum

Distributed Spatial filtering → *ad hoc* treatment in literature

IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING, VOL. 5, NO. 4, AUGUST 2011

Distributed Principal Subspace Estimation in Wireless Sensor Networks

Lin Li, *Student Member, IEEE*, Anna Scaglione, *Fellow, IEEE*, and Jonathan H. Manton, *Senior Member, IEEE*

Distributed Delay and Sum Beamformer for Speech Enhancement via Randomized Gossip

Yuan Zeng and Richard C. Hendriks

IEEE/ACM TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING, VOL. 22, NO. 1, JANUARY 2014

260 IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 61, NO. 13, JULY 1, 2013

Distributed LCMV Beamforming in a Wireless Sensor Network With Single-Channel Per-Node Signal Transmission

Alexander Bertrand, *Member, IEEE*, and Marc Moonen, *Fellow, IEEE*

Abstract—Linearly constrained minimum variance (LCMV) beamforming is a popular spatial filtering technique for signal estimation or signal enhancement in many different fields. We consider distributed LCMV (D-LCMV) beamforming in wireless sensor networks (WSNs) with either a fully connected or a tree topology. In both cases, we propose a distributed algorithm for its multiple sensor signals to a single-channel signal of which observations are then transmitted to other nodes. We envision an adaptive-to-explicit implementation where each node adapts its local LCMV beamformer coefficients based on the received sensor signal statistics, as well as to changes in the statistics of the wirelessly received signals. Although the per-node signal transmission and computational power is greatly reduced compared to a centralized implementation, the task of each node is to generate the centralized LCMV beamformer output as if it had access to all sensor signals in the entire network, without an explicit computation of the network-wide sensor signal covariance matrix. We provide theoretical results for convergence and optimality of the D-LCMV beamformer. These theoretical results are validated by means of Monte Carlo simulations, which demonstrate the performance of the D-LCMV beamformer.

Index Terms—Distributed beamforming, distributed signal estimation, LCMV beamforming, signal enhancement, wireless sensor networks (WSNs).

I. INTRODUCTION

A. Distributed Signal Estimation in Wireless Sensor Networks

A wireless sensor network (WSN) consists of a collection of A sensor nodes that are connected with each other through wireless links. Each node is equipped with one or more sensors and has computing capabilities for local signal processing. The sensor nodes collect observations of a physical phenomena, e.g., temperature, humidity, etc., and perform local signal processing tasks, e.g., localization, detection or estimation of certain signals or parameters. Some approaches require a so-called fusion center (e.g., [1]–[16]) that gathers all the sensor signals, whereas other algorithms are distributed such that all processing happens inside the network (e.g., [17]–[23]). The latter is usually preferred, especially when it is scalable in terms of communication bandwidth and processing power.

Most of the WSN literature focuses on distributed *parameter estimation* (DEPE), where a parameter vector with fixed dimensions is estimated from the collected data using distributed processing techniques, e.g., consensus-based methods [24]–[26]. However, in this paper we focus on distributed *signal estimation* (DSE) or signal enhancement, which relies on in-network signal fusion based on spatial filtering or beamforming techniques [4], [5], [14]–[27]. Rather than performing an iterative estimation of each individual sample of the desired signal, DSE algorithms iteratively improve the in-network fusion rules in a time-recursive fashion. DSE algorithms typically operate at higher data rates (compared to DEPE algorithms) and often require specific assumptions on the network, such as, e.g., fully connected, or tree topologies to avoid feeding in the signal fusion paths [11], [15], [16], [19], [20], [25]. In such topology-constrained networks, the main benefit of a distributed implementation lies in the in-network fusion/compression of the collected sensor data, i.e., nodes exchange only single-channel (scalar) signal observations instead of multi-channel (vector) signal observations. Even in small-scale networks, this can yield an important reduction in communication bandwidth, in particular for data-intensive tasks such as, e.g., audio processing [20]–[22], [24], [27].

In a fully connected network, all signals are gathered in a fusion center; traditional (centralized) beamforming techniques can be used to obtain an enhanced signal, based on the estimated covariance between all possible sensor signal pairs. Distributed (in-network) beamforming, on the other hand, is a more challenging problem since each node only has access to a subset of the sensor signals, such that the full covariance matrix cannot be estimated directly. A training phase could be used to construct this matrix, but this approach heavily affects adaptability and flexibility, even in slowly varying scenarios. Therefore, distributed beamforming is often based on suboptimal heuristics to maintain this adaptability and flexibility, and its performance then often heavily depends on the chosen network hierarchy or topology [17]–[19].

Manuscript received December 21, 2012; revised April 10, 2013; accepted April 12, 2013. Date of publication April 23, 2013; date of current version June 10, 2013. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Yimin D. Zhang. This work of A. Bertrand was supported by a Postdoctoral Fellowship of the Research Foundation—Flanders (FWO). This work was carried out at the EURECOM Laboratory of KU Leuven, in the framework of the European Project OptiC (Optimizing Communications in Engineering) and FP7/2009/02/OPTIC, Concerted Research Action GOA-MaNet, the Belgian Programme on Interuniversity Attraction Poles initiated by the Belgian National Fund for Scientific Research (FNRS), “Dynamic systems, control and optimization,” 2007–2011, Research Project IMinds, and Research Project FWO nr. 060008/007 “Signal processing and network management for cognitive sensor networks.” The corresponding author is assumed by the authors.

The authors are with the EURECOM Department of Electrical Engineering, France, and with the Institute of Mathematics and the Future Health Department, KU Leuven, Belgium (e-mail: alexander.bertrand@eurecom.fr; marc.moonen@cs.kuleuven.be).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TSP.2013.2359486

1053-587X/13/\$31.00 © 2013 IEEE
Authorized licensed use limited to: KU Leuven Libraries. Downloaded on August 04, 2023 at 12:32:35 UTC from IEEE Xplore. Restrictions apply.

IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING, VOL. 5, NO. 4, AUGUST 2011

Distributed Principal Subspace Estimation in Wireless Sensor Networks

Lin Li, *Student Member, IEEE*, Anna Scaglione, *Fellow, IEEE*, and Jonathan H. Manton, *Senior Member, IEEE*

Distributed Delay and Sum Beamformer for Speech Enhancement via Randomized Gossip

Yuan Zeng and Richard C. Hendriks

IEEE/ACM TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING, VOL. 22, NO. 1, JANUARY 2014

260 IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 61, NO. 13, JULY 1, 2013

Distributed LCMV Beamforming in a Wireless Sensor Network With Single-Channel Per-Node Signal Transmission

Alexander Bertrand, *Member, IEEE*, and Marc Moonen, *Fellow, IEEE*

Abstract—Linearly constrained minimum variance (LCMV) beamforming is a popular spatial filtering technique for signal estimation or signal enhancement in many different fields. We consider distributed LCMV (D-LCMV) beamforming in wireless sensor networks (WSNs) with either a fully connected or a tree topology. In both cases, we propose a distributed algorithm for its multiple sensor signals to a single-channel signal of which observations are then transmitted to other nodes. We envision an adaptive-to-explicit implementation where each node adapts its local LCMV beamformer coefficients based on the received sensor signal statistics, as well as to changes in the statistics of the wirelessly received signals. Although the per-node signal transmission and computational power is greatly reduced compared to a centralized implementation, the task of each node is to generate the centralized LCMV beamformer output as if it had access to all sensor signals in the entire network, without an explicit computation of the network-wide sensor signal covariance matrix. We provide theoretical results for convergence and optimality of the D-LCMV beamformer. These theoretical results are validated by means of Monte Carlo simulations, which demonstrate the performance of the D-LCMV beamformer.

Index Terms—Distributed beamforming, distributed signal estimation, LCMV beamforming, signal enhancement, wireless sensor networks (WSNs).

I. INTRODUCTION

A. Distributed Signal Estimation in Wireless Sensor Networks

A wireless sensor network (WSN) consists of a collection of A sensor nodes that are connected with each other through wireless links. Each node is equipped with one or more sensors and has computing capabilities for local signal processing. The sensor nodes collect observations of a physical phenomena, e.g., temperature, humidity, etc., and perform local signal processing tasks, e.g., localization, detection or estimation of certain signals or parameters. Some approaches require a so-called fusion center (e.g., [1]–[16]) that gathers all the sensor signals, whereas other algorithms are distributed such that all processing happens inside the network (e.g., [17]–[23]). The latter is usually preferred, especially when it is scalable in terms of communication bandwidth and processing power.

Most of the WSN literature focuses on distributed *parameter estimation* (DEPE), where a parameter vector with fixed dimensions is estimated from the collected data using distributed processing techniques, e.g., consensus-based methods [24]–[26]. However, in this paper we focus on distributed *signal estimation* (DSE) or signal enhancement, which relies on in-network signal fusion based on spatial filtering or beamforming techniques [4], [5], [14]–[27]. Rather than performing an iterative estimation of each individual sample of the desired signal, DSE algorithms iteratively improve the in-network fusion rules in a time-recursive fashion. DSE algorithms typically operate at higher data rates (compared to DEPE algorithms) and often require specific assumptions on the network, such as, e.g., fully connected, or tree topologies to avoid feeding in the signal fusion paths [11], [15], [16], [19], [20], [25]. In such topology-constrained networks, the main benefit of a distributed implementation lies in the in-network fusion/compression of the collected sensor data, i.e., nodes exchange only single-channel (scalar) signal observations instead of multi-channel (vector) signal observations. Even in small-scale networks, this can yield an important reduction in communication bandwidth, in particular for data-intensive tasks such as, e.g., audio processing [20]–[22], [24], [27].

In a fully connected network, all signals are gathered in a fusion center; traditional (centralized) beamforming techniques can be used to obtain an enhanced signal, based on the estimated covariance between all possible sensor signal pairs. Distributed (in-network) beamforming, on the other hand, is a more challenging problem since each node only has access to a subset of the sensor signals, such that the full covariance matrix cannot be estimated directly. A training phase could be used to construct this matrix, but this approach heavily affects adaptability and flexibility, even in slowly varying scenarios. Therefore, distributed beamforming is often based on suboptimal heuristics to maintain this adaptability and flexibility, and its performance then often heavily depends on the chosen network hierarchy or topology [17]–[19].

Manuscript received December 21, 2012; revised April 10, 2013; accepted April 12, 2013. Date of publication April 23, 2013; date of current version June 10, 2013. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Yimin D. Zhang. This work of A. Bertrand was supported by a Postdoctoral Fellowship of the Research Foundation—Flanders (FWO). This work was carried out at the EURECOM Laboratory of KU Leuven, in the framework of the European Project OptiC (Optimizing Communications in Engineering) and FP7/2009/02/OPTIC, Concerted Research Action GOA-MaNet, the Belgian Programme on Interuniversity Attraction Poles initiated by the Belgian National Fund for Scientific Research (FNRS), “Dynamic systems, control and optimization,” 2007–2011, Research Project IMinds, and Research Project FWO nr. 060008/007 “Signal processing and network management for cognitive sensor networks.” The corresponding author is assumed by the authors.

The authors are with the EURECOM Department of Electrical Engineering, France, and with the Institute of Mathematics and the Future Health Department, KU Leuven, Belgium (e-mail: alexander.bertrand@eurecom.fr; marc.moonen@cs.kuleuven.be).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TSP.2013.2359486

1053-587X/13/\$31.00 © 2013 IEEE
Authorized licensed use limited to: KU Leuven Libraries. Downloaded on August 04, 2023 at 12:32:35 UTC from IEEE Xplore. Restrictions apply.

IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 69, 2021

3653

Distributed Adaptive Trace Ratio Optimization in Wireless Sensor Networks

Cem Ates Musluoglu and Alexander Bertrand , Senior Member, IEEE

IEEE SIGNAL PROCESSING LETTERS, VOL. 26, NO. 5, MAY 2019

675

Distributed Rate-Constrained LCMV Beamforming

Jie Zhang , Andreas I. Koutouvelis , Richard Heusdens , and Richard C. Hendriks

Abstract—In this letter, we propose a decentralized framework for rate-constrained linearly constrained minimum variance (LCMV) beamforming in wireless acoustic sensor networks. To save the energy usage within the network, we propose to minimize the transmission cost and put a constraint on the noise reduction

method block-diagonalizes the noise/noisy correlation matrix using linear equality constraints, leading to an efficient distributed implementation for the LCMV beamformer. However, this method does not take into account the quantization noise introduced during the communication between the devices. Nor to account, our proposed cost matrix is based on minima of the noise was introduced with respect to the communication.

© 2019 IEEE. This work was supported by the National Research Foundation of South Africa (NRF) under Grant 112533. The authors would like to thank the anonymous reviewers for their valuable comments and suggestions.

Manuscript received March 17, 2017; accepted May 30, 2015. Date of publication June 9, 2017; date of current version June 13, 2015. The associate editor coordinating the review of this manuscript and approved it for publication was Prof. Sergio Theodoridis. This work was carried out at the ESTEC Laboratory of KU Leuven, in the frame of KU Leuven Research Council BOF-STG-14-006. © 2019 IEEE. This work was supported by the European Union's Horizon 2020 research and innovation program under grant agreement No. 645222 (DCCAS), by the Belgian Federal Science Policy Office IAP/P7/23 (BESTCOM, 2012–2017), project Models Mediated IT, Research Projects Projects FWO/6.003.14.001, and by the European Commission for temporary communication (FWO/6.003.14.001). Details of other grants are given in the Acknowledgments. This paper is based on parts of the thesis [1].

A. Bertrand was with Department of Electrical Engineering (ESAT), Studies Center for Dynamical Systems, Signal Processing and Data Analytics, KU Leuven, Belgium. He is also with MINDS Medical IT (e-mail: alexander.bertrand@esat.kuleuven.be).

M. Ates was with Department of Electrical Engineering (ESAT), Studies Center for Dynamical Systems, Signal Processing and Data Analytics, KU Leuven, 3000 Leuven, Belgium (e-mail: maresm@esat.kuleuven.be).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/LSP.2015.2441729

Manuscript received March 17, 2017; accepted May 30, 2015. Date of publication June 9, 2017; date of current version June 13, 2015. The associate editor coordinating the review of this manuscript and approved it for publication was Prof. Sergio Theodoridis. This work was carried out at the ESTEC Laboratory of KU Leuven, in the frame of KU Leuven Research Council BOF-STG-14-006. © 2019 IEEE. This work was supported by the European Union's Horizon 2020 research and innovation program under grant agreement No. 645222 (DCCAS), by the Belgian Federal Science Policy Office IAP/P7/23 (BESTCOM, 2012–2017), project Models Mediated IT, Research Projects Projects FWO/6.003.14.001, and by the European Commission for temporary communication (FWO/6.003.14.001). Details of other grants are given in the Acknowledgments. This paper is based on parts of the thesis [1].

A. Bertrand was with Department of Electrical Engineering (ESAT), Studies Center for Dynamical Systems, Signal Processing and Data Analytics, KU Leuven, Belgium (e-mail: alexander.bertrand@esat.kuleuven.be).

M. Ates was with Department of Electrical Engineering (ESAT), Studies Center for Dynamical Systems, Signal Processing and Data Analytics, KU Leuven, 3000 Leuven, Belgium (e-mail: maresm@esat.kuleuven.be).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/LSP.2015.2441729

the problem in the intended manner logging sent signal from the file was from our [21] to [2] general performance compared

to solve the distributed put highly

transformer

we proposed

introduced on perfor-

pared

ated user

des, where de

... Mi, mis

one target

denote the ring nodes

denoted by

In the usy STFT

given by

(1)

istic trans-

the eith mi-

get some re-

ments, the

50 ms) and

only reflect

the speech

$\mathcal{N}_s(u, l)$

(2)

Hovine

IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 63, NO. 18, SEPTEMBER 15, 2015

Distributed Canonical Correlation Analysis in Wireless Sensor Networks With Application to Distributed Blind Source Separation

Alexander Bertrand, Member, IEEE, and Marc Moonen, Fellow, IEEE

I. INTRODUCTION

CANONICAL correlation analysis (CCA) is a widely-used data analysis tool, in particular for finding the maximum two distinct sets of data or signals [1], [2]. Basically, it looks for directions in the data with maximal cross-correlation, i.e., it computes optimal linear combinations of the signals in both sets such that the resulting signals are maximally correlated. It is strongly related to partial least squares (PLS) [3], which search for directions with maximal covariance instead of maximal correlation. Both techniques can be viewed as extensions of principal component analysis (PCA) to two data sets. Future extensions of CCA to more than two data sets (multi-set ICA or MCCA), have also been proposed [4], but these are beyond the scope of this paper.

The CCA concept has become a widely-used signal processing (SP) tool, in particular in biomedical signal processing [5]–[13], but also in array processing [14], radar anti-jamming [15], speaker identification [16], SIMO and MIMO equalization [17]–[18], and many other applications [19]. It is noted that due to its multi-set framework, CCA is often used for multi-modal signal analysis [17]–[19], [12], [16].

One important SP application of CCA is blind source separation (BSS) [5], [20]–[23]. This CCA-based BSS approach assumes that the hidden sources have a different autocorrelation structure, which is a valid assumption in many applications. For example, CCA-based BSS has been used to remove ocular artifacts [24] in electroencephalography (EEG) data. The technique fits in the family of second-order statistics (SOS) BSS techniques [24]–[26], and has a significantly lower computational complexity compared to higher-order statistics BSS techniques, such as independent component analysis [27] and fast implementations thereof [20].

In this paper, we consider the use of CCA to analyze data that is slowly-varying in wireless sensor networks (WSNs) possibly in a (slowly-varying) dynamic scenario where the CCA directions may change over time (e.g., for adaptive BSS). Computing the network-wide CCA requires two matrix inversions and a generalized eigenvalue decomposition involving three network-wide correlation matrices that together capture the correlation between every possible sensor signal pair in the WSN. In principle, this would require to forward all the raw sensor signal observations through the network and compute three network-wide correlation matrices can then be estimated. However, this usually results in a high bit rate over the wireless links, and it requires a high routing efficiency (in the case of partially-connected networks), and an FC with a large computational power.

Manuscript received June 17, 2014; revised January 15, 2015; accepted May 30, 2015. Date of publication June 9, 2017; date of current version June 13, 2015. The associate editor coordinating the review of this manuscript and approved it for publication was Prof. Sergio Theodoridis. This work was carried out at the ESTEC Laboratory of KU Leuven, in the frame of KU Leuven Research Council BOF-STG-14-006. © 2019 IEEE. This work was supported by the European Union's Horizon 2020 research and innovation program under grant agreement No. 645222 (DCCAS), by the Belgian Federal Science Policy Office IAP/P7/23 (BESTCOM, 2012–2017), project Models Mediated IT, Research Projects Projects FW0/6.003.14.001, and by the European Commission for temporary communication (FWO/6.003.14.001). Details of other grants are given in the Acknowledgments. This paper is based on parts of the thesis [1].

A. Bertrand was with Department of Electrical Engineering (ESAT), Studies Center for Dynamical Systems, Signal Processing and Data Analytics, KU Leuven, 3000 Leuven, Belgium (e-mail: alexander.bertrand@esat.kuleuven.be).

M. Moonen was with Department of Electrical Engineering (ESAT), Studies Center for Dynamical Systems, Signal Processing and Data Analytics, KU Leuven, 3000 Leuven, Belgium (e-mail: maresm@esat.kuleuven.be).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/LSP.2015.2441729

Manuscript received June 17, 2014; revised January 15, 2015; accepted May 30, 2015. Date of publication June 9, 2017; date of current version June 13, 2015. The associate editor coordinating the review of this manuscript and approved it for publication was Prof. Sergio Theodoridis. This work was carried out at the ESTEC Laboratory of KU Leuven, in the frame of KU Leuven Research Council BOF-STG-14-006. © 2019 IEEE. This work was supported by the European Union's Horizon 2020 research and innovation program under grant agreement No. 645222 (DCCAS), by the Belgian Federal Science Policy Office IAP/P7/23 (BESTCOM, 2012–2017), project Models Mediated IT, Research Projects Projects FW0/6.003.14.001, and by the European Commission for temporary communication (FWO/6.003.14.001). Details of other grants are given in the Acknowledgments. This paper is based on parts of the thesis [1].

A. Bertrand was with Department of Electrical Engineering (ESAT), Studies Center for Dynamical Systems, Signal Processing and Data Analytics, KU Leuven, 3000 Leuven, Belgium (e-mail: alexander.bertrand@esat.kuleuven.be).

M. Moonen was with Department of Electrical Engineering (ESAT), Studies Center for Dynamical Systems, Signal Processing and Data Analytics, KU Leuven, 3000 Leuven, Belgium (e-mail: maresm@esat.kuleuven.be).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/LSP.2015.2441729

Manuscript received June 17, 2014; revised January 15, 2015; accepted May 30, 2015. Date of publication June 9, 2017; date of current version June 13, 2015. The associate editor coordinating the review of this manuscript and approved it for publication was Prof. Sergio Theodoridis. This work was carried out at the ESTEC Laboratory of KU Leuven, in the frame of KU Leuven Research Council BOF-STG-14-006. © 2019 IEEE. This work was supported by the European Union's Horizon 2020 research and innovation program under grant agreement No. 645222 (DCCAS), by the Belgian Federal Science Policy Office IAP/P7/23 (BESTCOM, 2012–2017), project Models Mediated IT, Research Projects Projects FW0/6.003.14.001, and by the European Commission for temporary communication (FWO/6.003.14.001). Details of other grants are given in the Acknowledgments. This paper is based on parts of the thesis [1].

A. Bertrand was with Department of Electrical Engineering (ESAT), Studies Center for Dynamical Systems, Signal Processing and Data Analytics, KU Leuven, 3000 Leuven, Belgium (e-mail: alexander.bertrand@esat.kuleuven.be).

M. Moonen was with Department of Electrical Engineering (ESAT), Studies Center for Dynamical Systems, Signal Processing and Data Analytics, KU Leuven, 3000 Leuven, Belgium (e-mail: maresm@esat.kuleuven.be).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/LSP.2015.2441729

Distributed Spatial filtering

Common (ad hoc) strategies

- #1: Use centralized algorithm (e.g. stochastic gradient descent) and compute each network-wide inner product via some type of **consensus iteration**

Example: Oja's algorithm for PCA

$$\mathbf{x}^{i+1} = \mathbf{x}^i + \mu \left(\beta^i \mathbf{y}(i) - (\beta^i)^2 \mathbf{x}^i \right) \text{ with } \beta^i = \mathbf{y}(i)^T \mathbf{x}^i = \sum_k y_k(i) x_k^i$$



Oja's iterative procedure
(outer loop iteration)



Consensus iteration for in-network
averaging / summation (nested iteration)

Distributed Spatial filtering

Common (ad hoc) strategies

#1: Use centralized algorithm (e.g. stochastic gradient descent) and compute each network-wide inner product via some type of **consensus iteration**

1) **communication-intensive**:

nested iterations (2 time scales)

Multiple transmissions for each *single* collected sample

2) **does not scale** well with network size

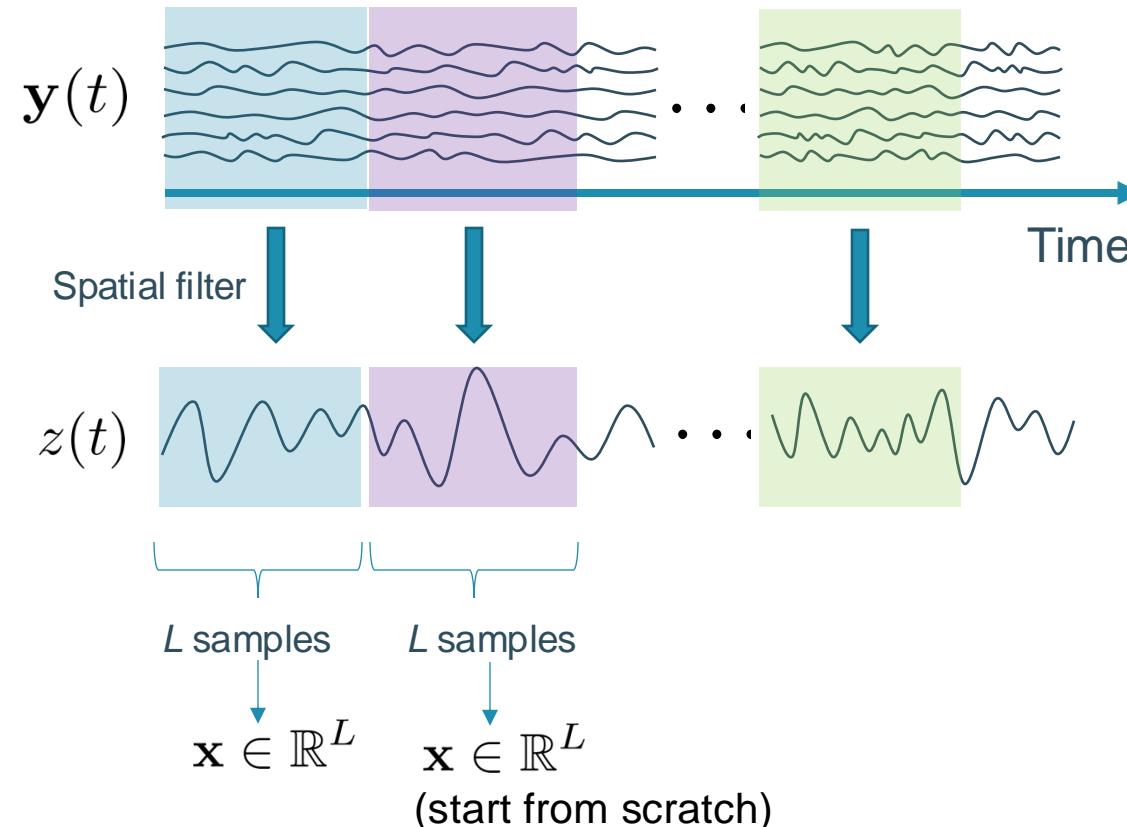
3) strategy not always applicable (depends on chosen centralized algo)

4) often **per-sample transmissions** ($i = t$) → inefficient compared to sample-block transmissions

Distributed Spatial filtering

Common (ad hoc) strategies

- #2: Treat block of samples of the filter output signal as **consensus variable x** and run a workhorse distributed algo (consensus, diffusion, ADMM, ...) on each separate block



Distributed Spatial filtering

Common (ad hoc) strategies

#2: Treat block of samples of the filter output signal as **consensus variable x** and run a workhorse distributed algo (consensus, diffusion, ADMM, ...) on each separate block

1) **communication-intensive :**

- Solving 1 block of L samples = multiple iterations, each one requiring transmission of L-dimensional vector
- Start from scratch for the next block of L samples

2) **does not scale** well with network size

3) strategy not always applicable (depends on optimization problem)

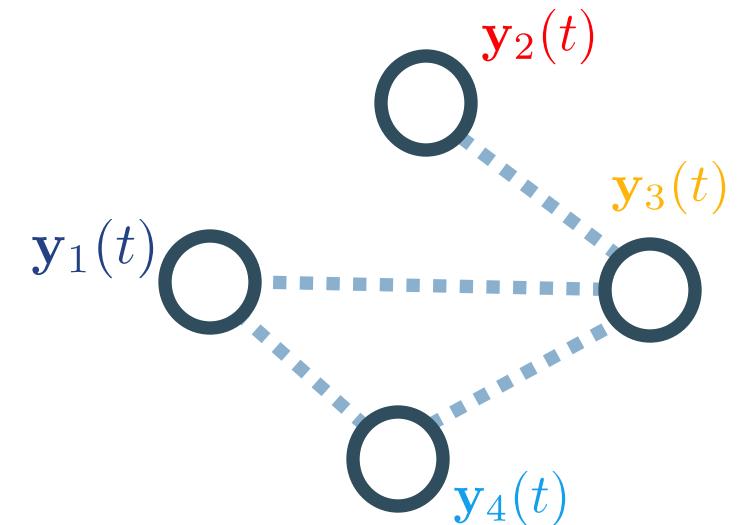
Distributed Spatial filtering

Common (ad hoc) strategies

#3: Make (unrealistic) assumptions on covariance matrix, e.g.

- $y(t)$ or noise is spatially white
- R_{yy} has same structure as the communication network

$$R_{yy} = \begin{bmatrix} R_{11} & 0 & R_{13} & R_{14} \\ 0 & R_{22} & R_{23} & 0 \\ R_{31} & R_{32} & R_{33} & R_{34} \\ R_{41} & 0 & R_{43} & R_{44} \end{bmatrix}$$



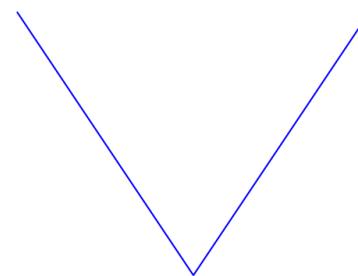
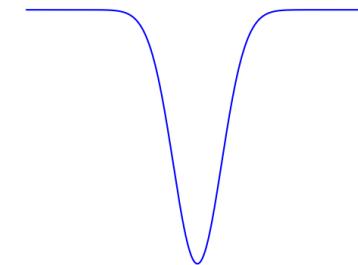
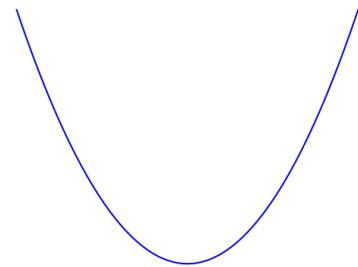
Distributed Signal Processing

Problem Taxonomy

Convex

Smooth

Non-smooth



Non-convex

Note: many spatial filtering
problems are non-convex
(PCA, max-SNR, CCA, ...)

Distributed Signal Processing

Algorithms Overview

Algorithm	Exchanged data	Data split	Non-convex problems	Non-smooth problems	Adaptive
Consensus	Opt. variables	Samples	depends	depends	
Diffusion	Opt. variables	Samples			
ADMM or PDMM	Opt. variables	Samples (or Features/channels)			



Problem for spatial filtering
→ requires split in features/channels

Distributed Signal Processing

Algorithms Overview

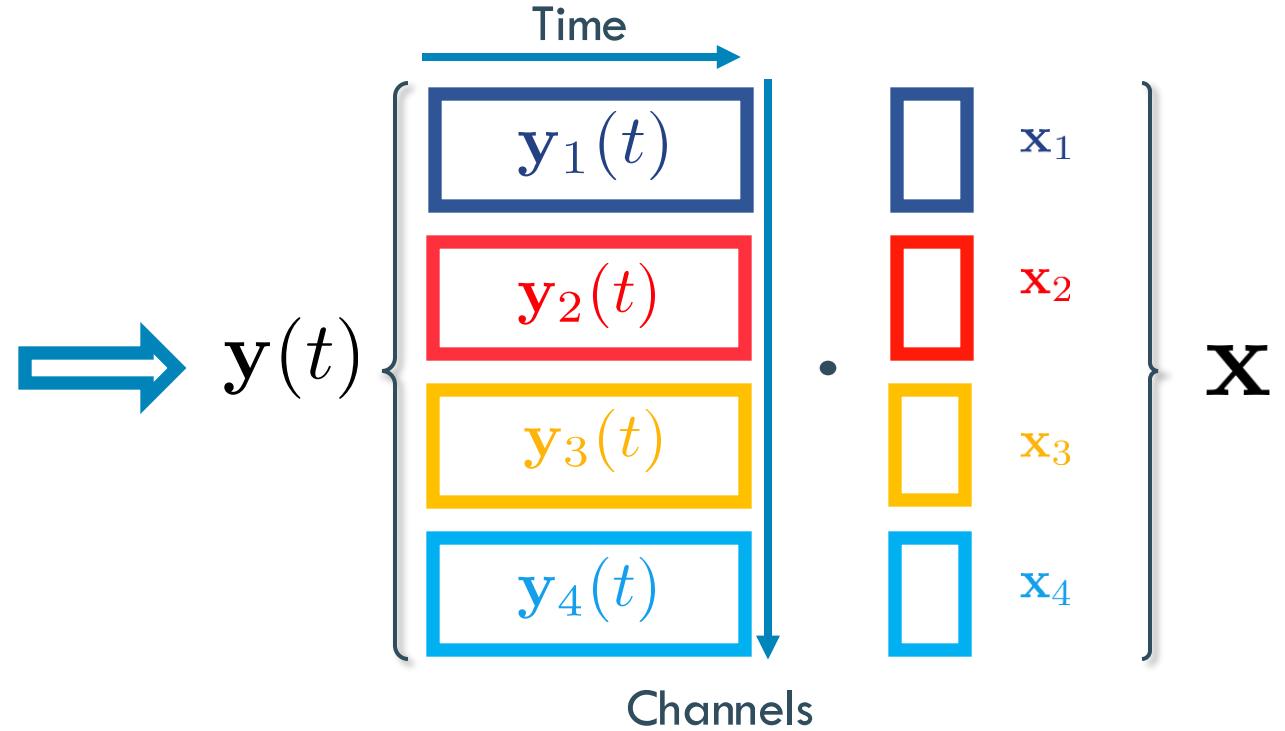
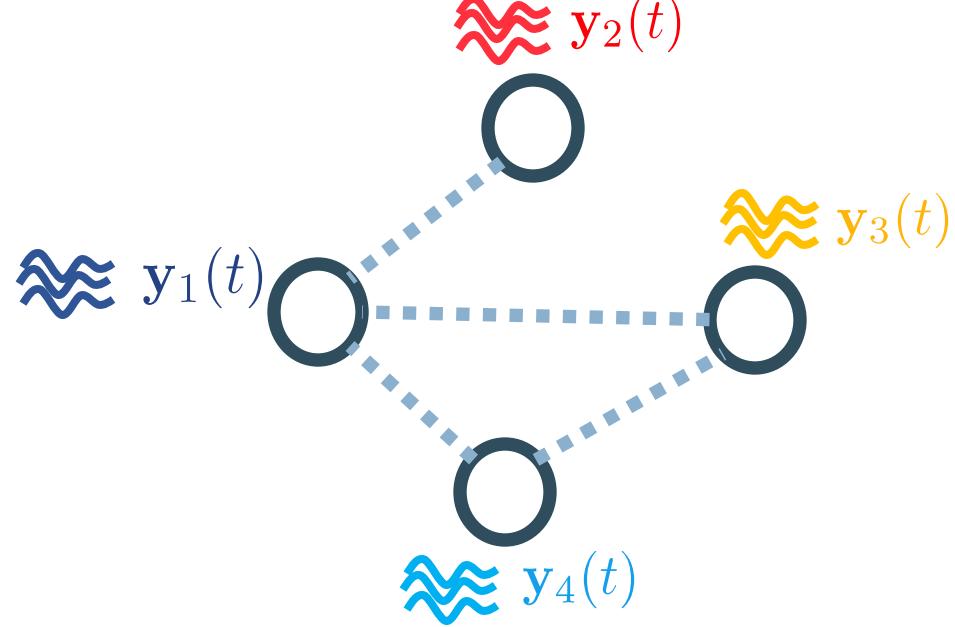
Algorithm	Exchanged data	Data split	Non-convex problems	Non-smooth problems	Adaptive
Consensus	Opt. variables	Samples	depends	depends	
Diffusion	Opt. variables	Samples			
ADMM or PDMM	Opt. variables	Samples (or Features/channels)			
DASF	Signals	Features/channels			

Spoiler Alert !

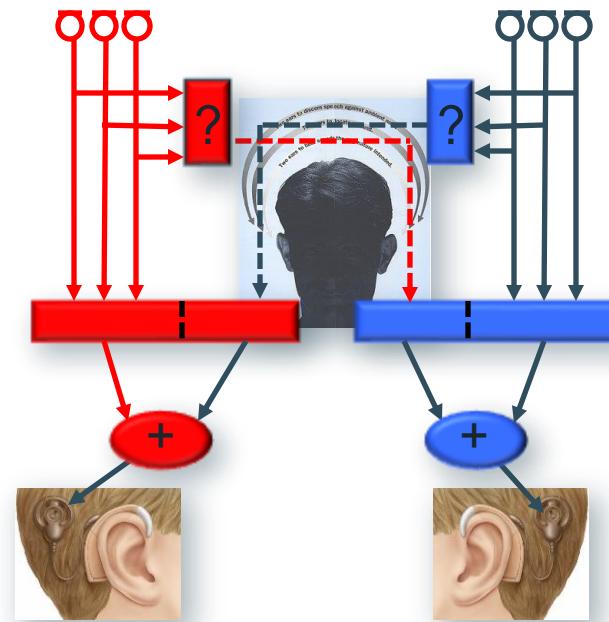
II- Towards a generic meta-algorithm for distributed data-driven spatial filtering

II.A- Preliminaries

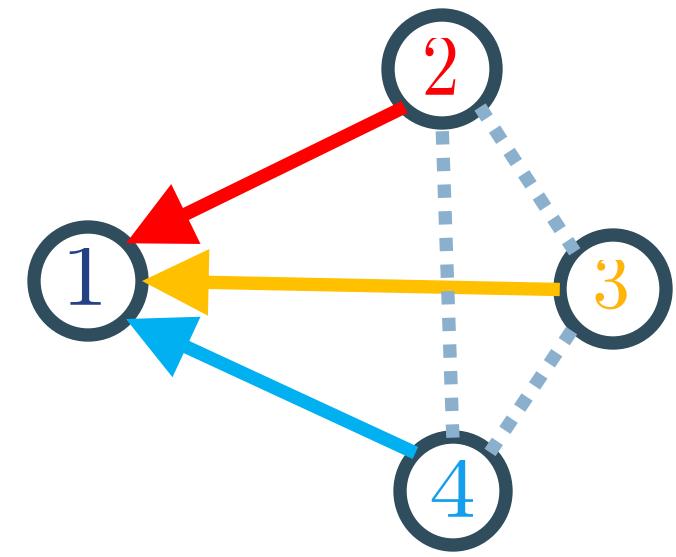
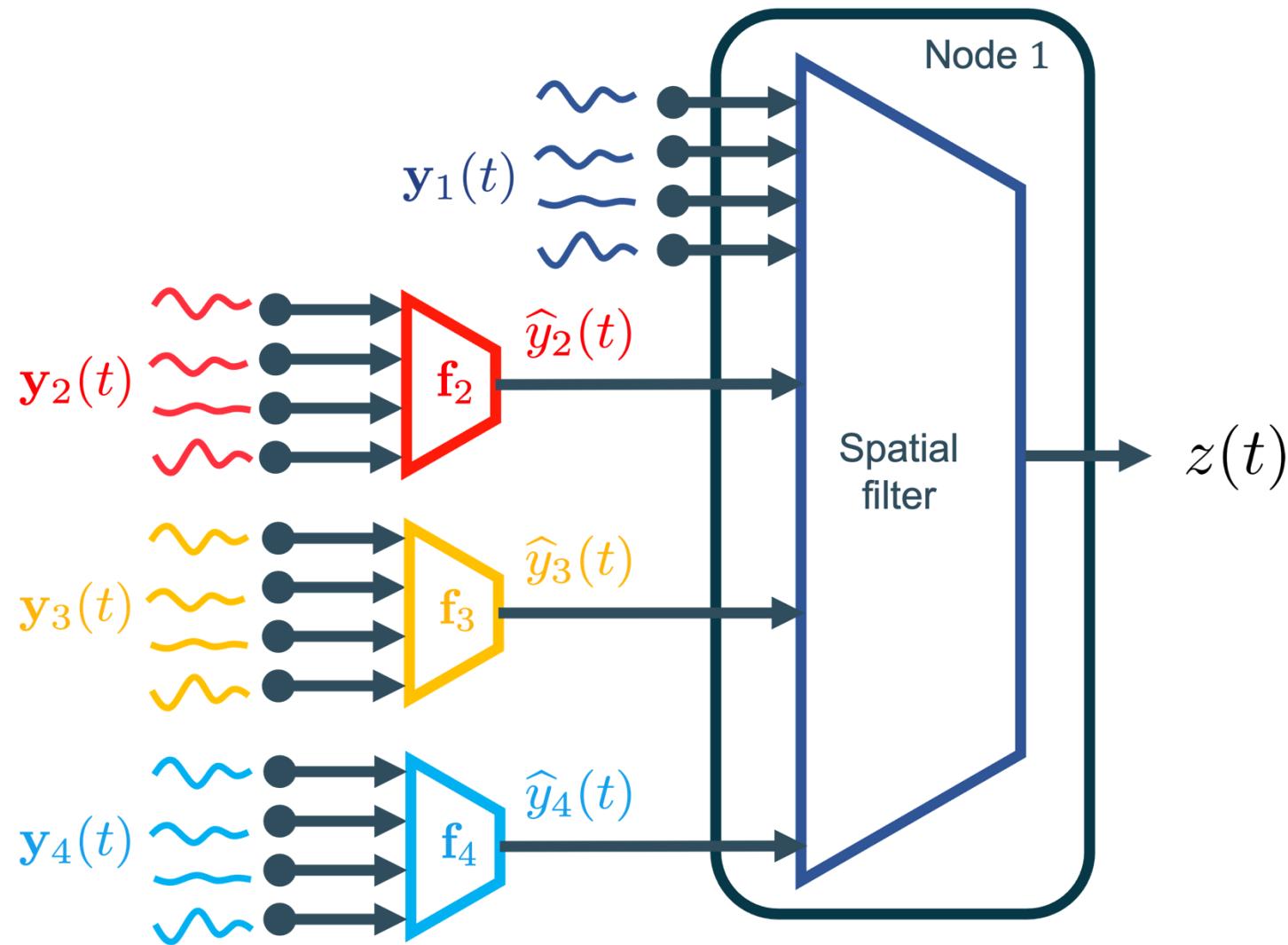
Distributed spatial filtering setting



Data flow (2 node example)



Data flow (Fully-connected network)

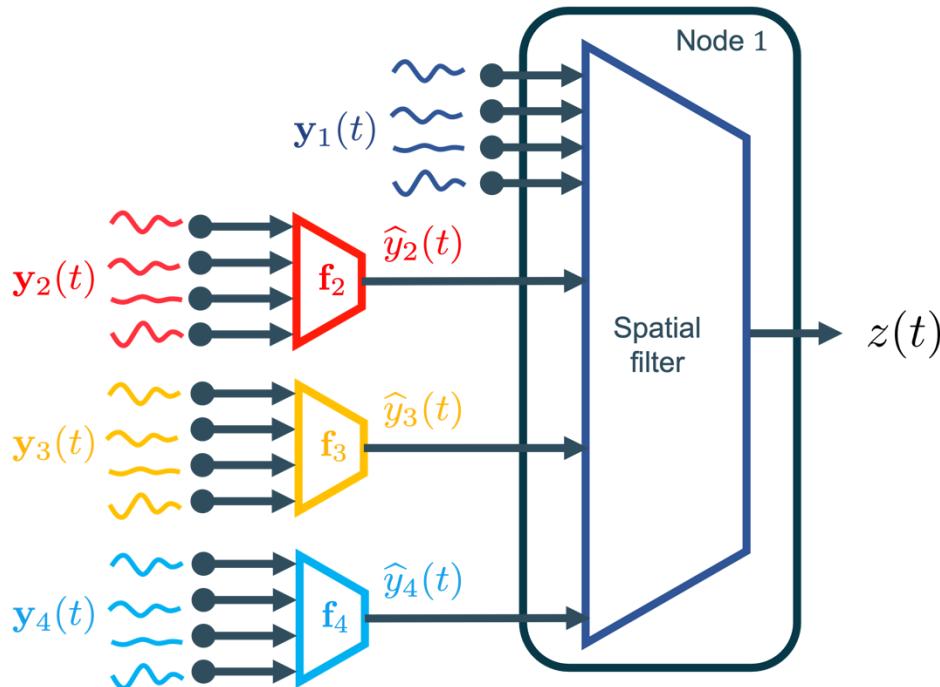


Remember this notation!

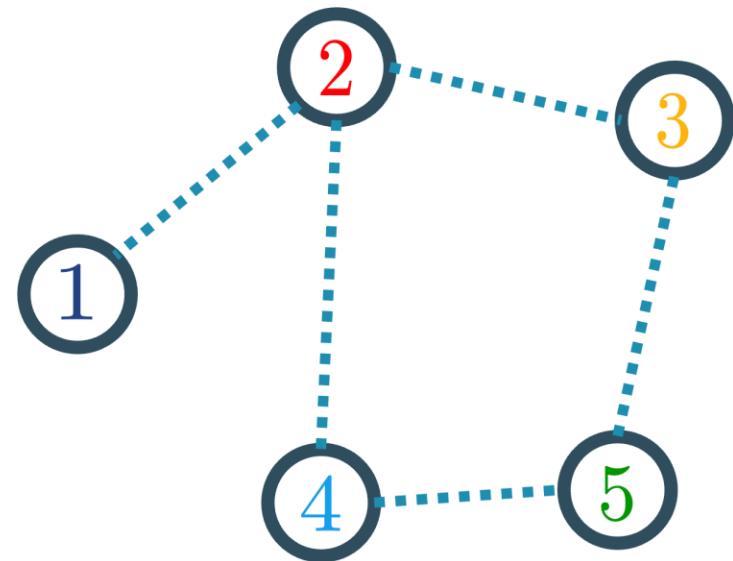


y_k = **original** (raw) data

\hat{y}_k = **fused** data



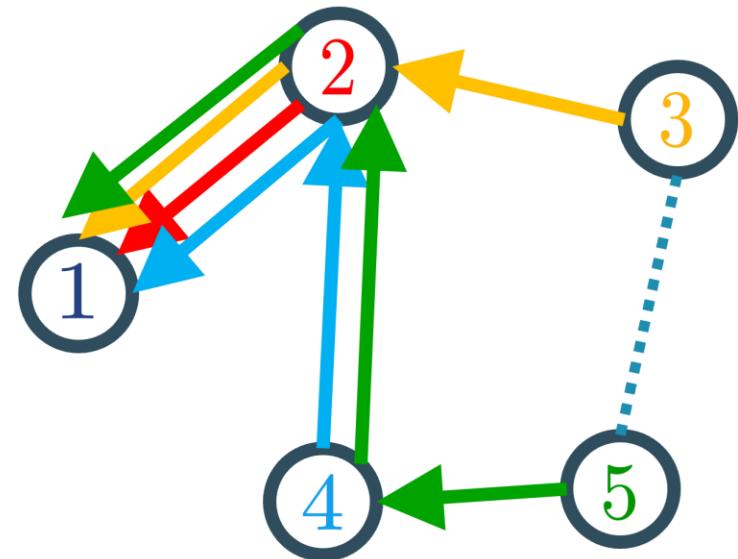
Data flow (General)



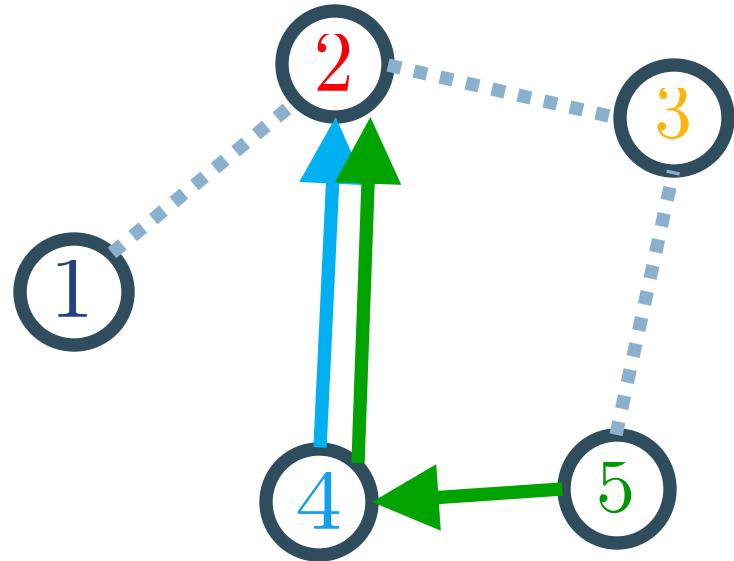
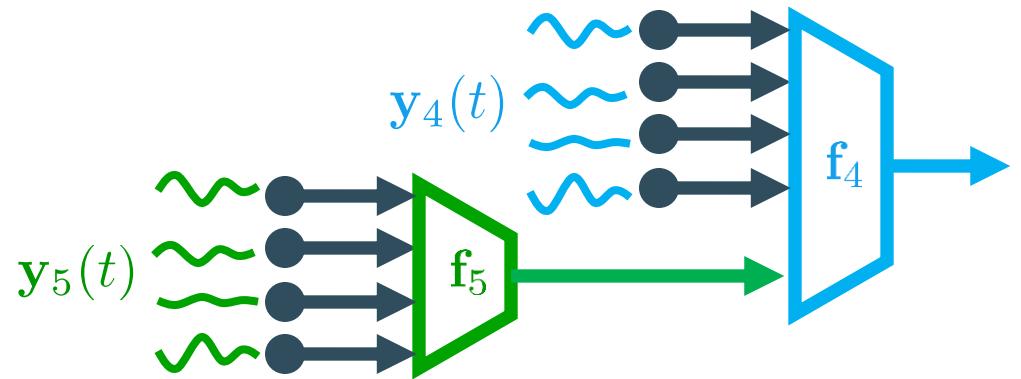
Data flow (General)

Avoid relaying data !

(not scalable)

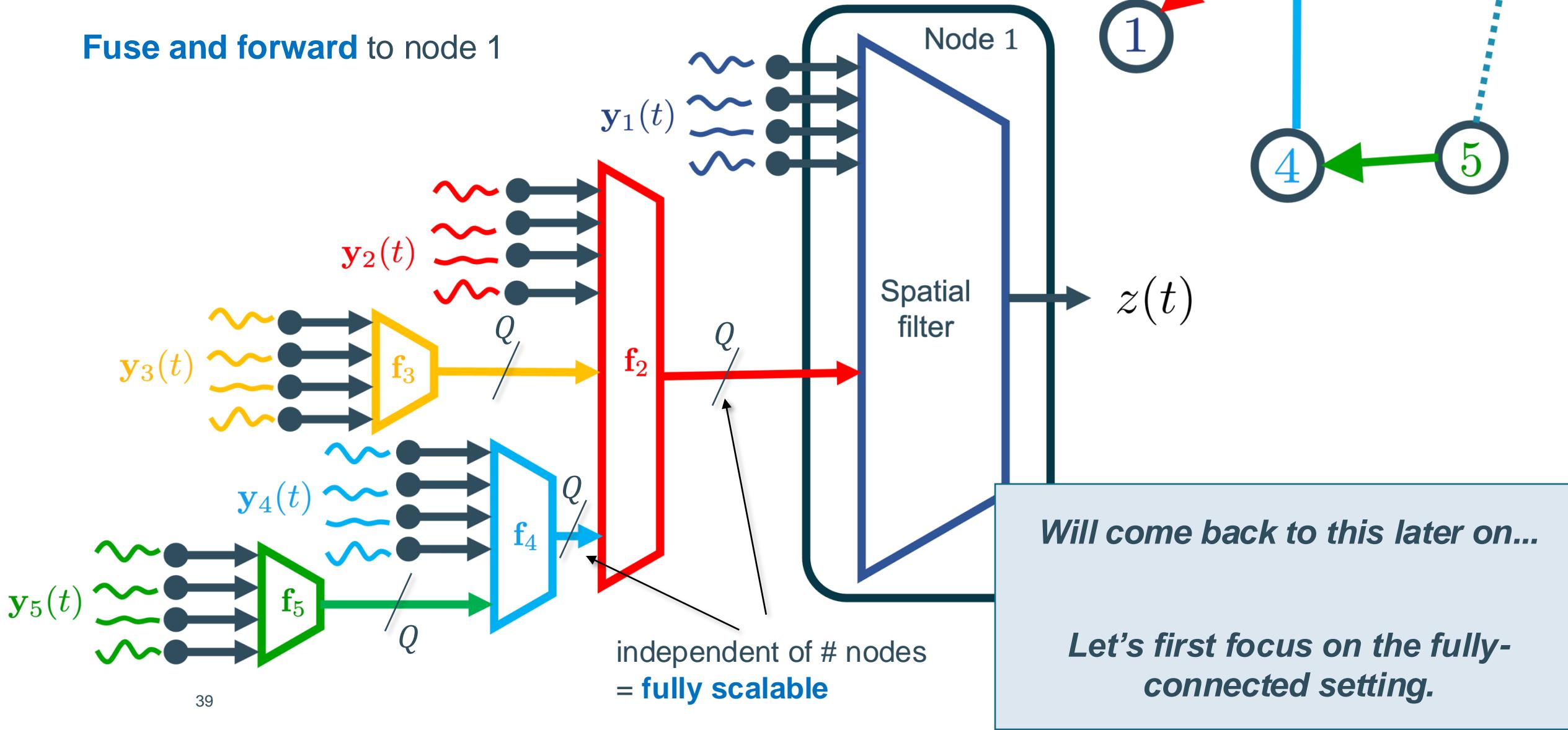


Data flow (General)

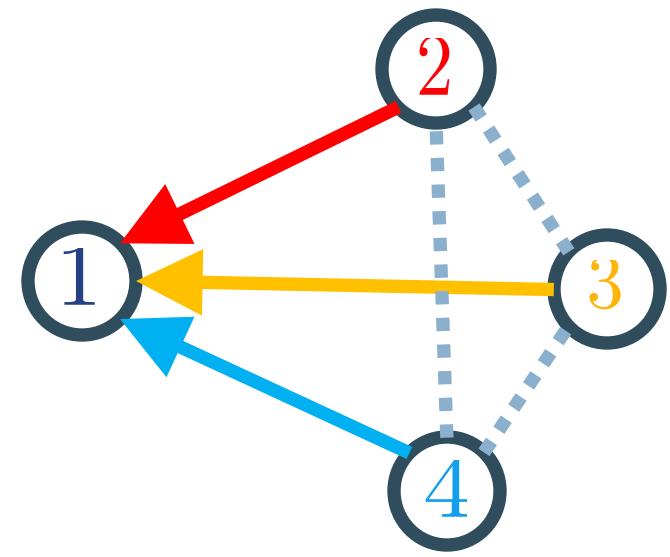
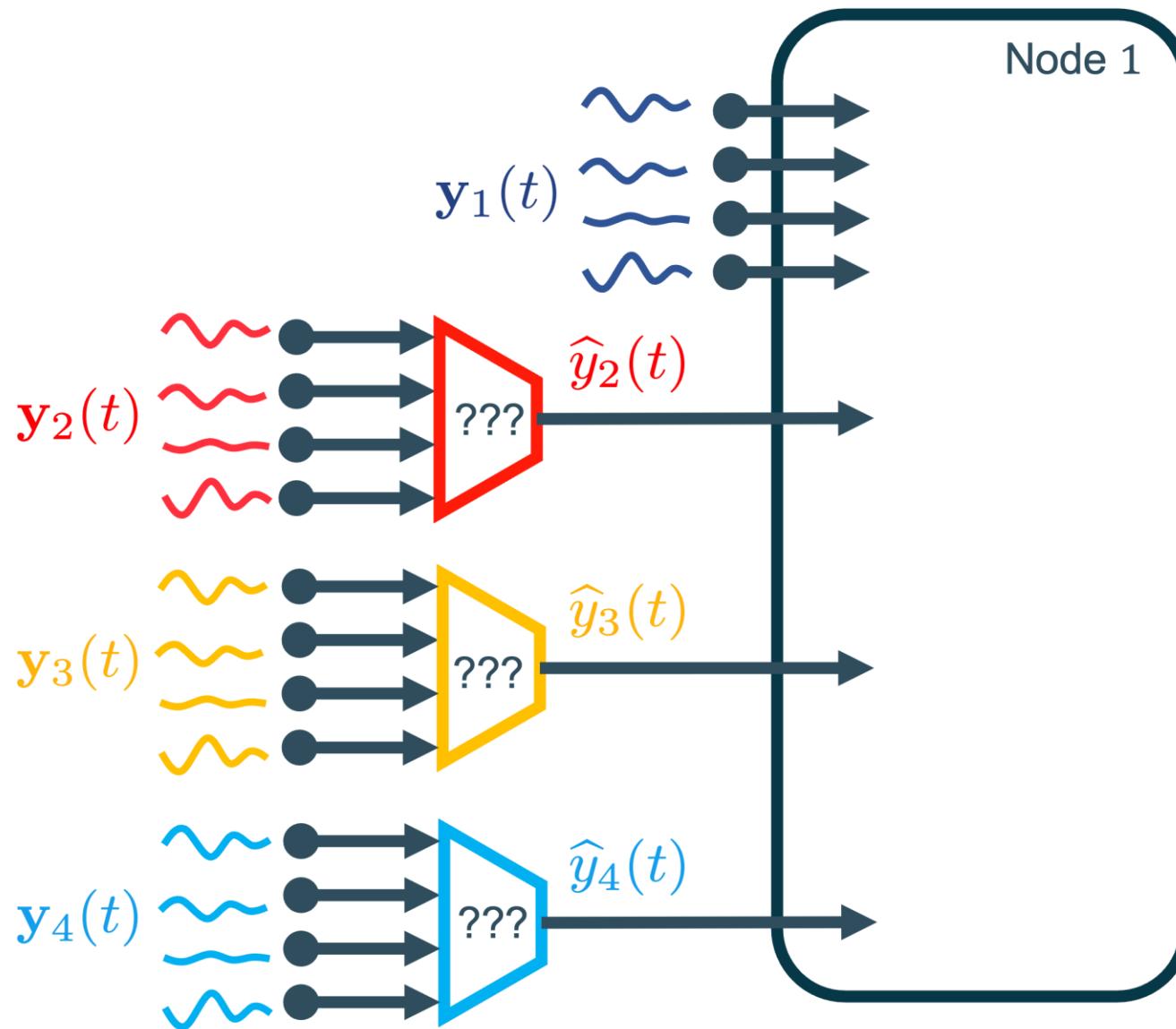


Data flow (General)

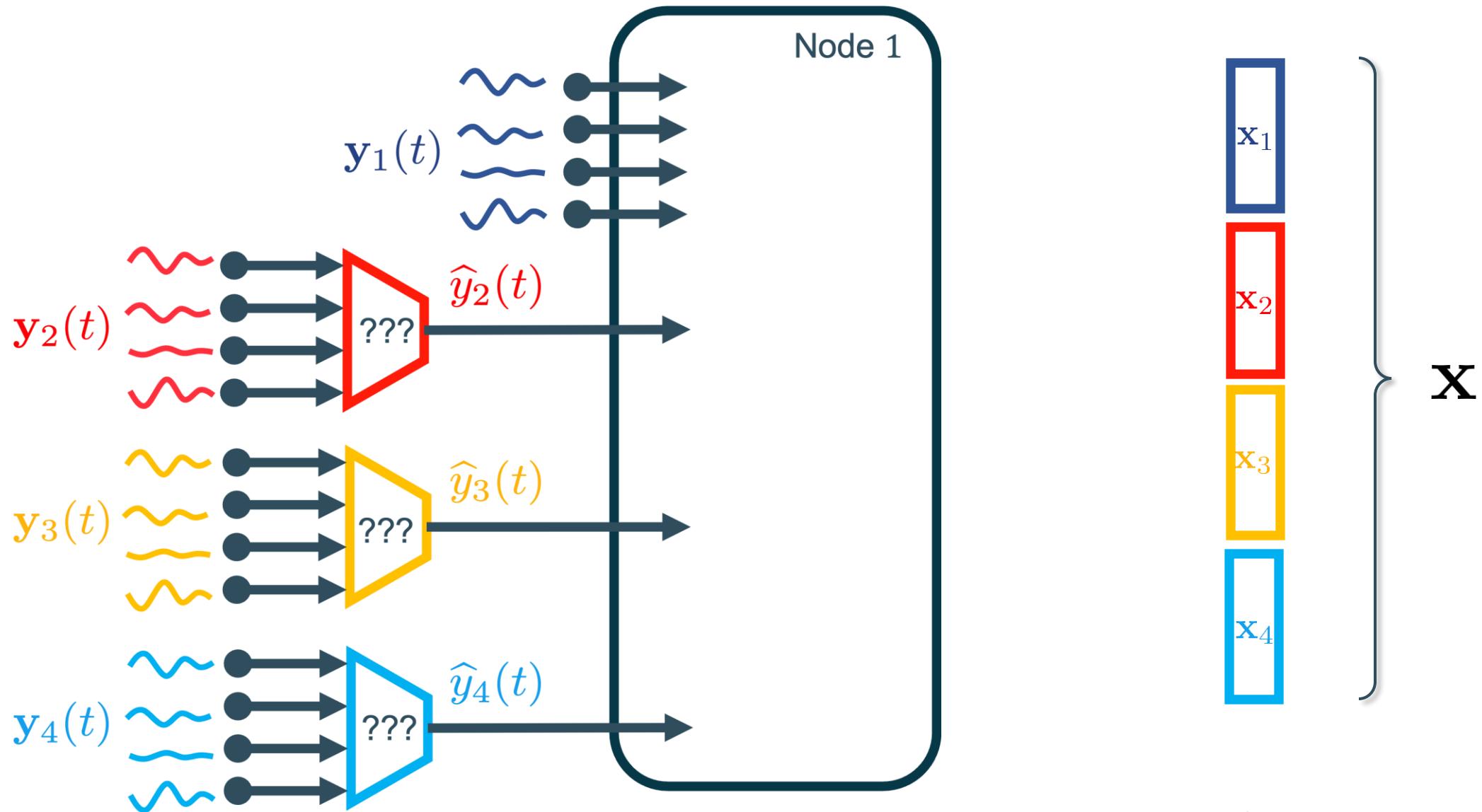
Fuse and forward to node 1



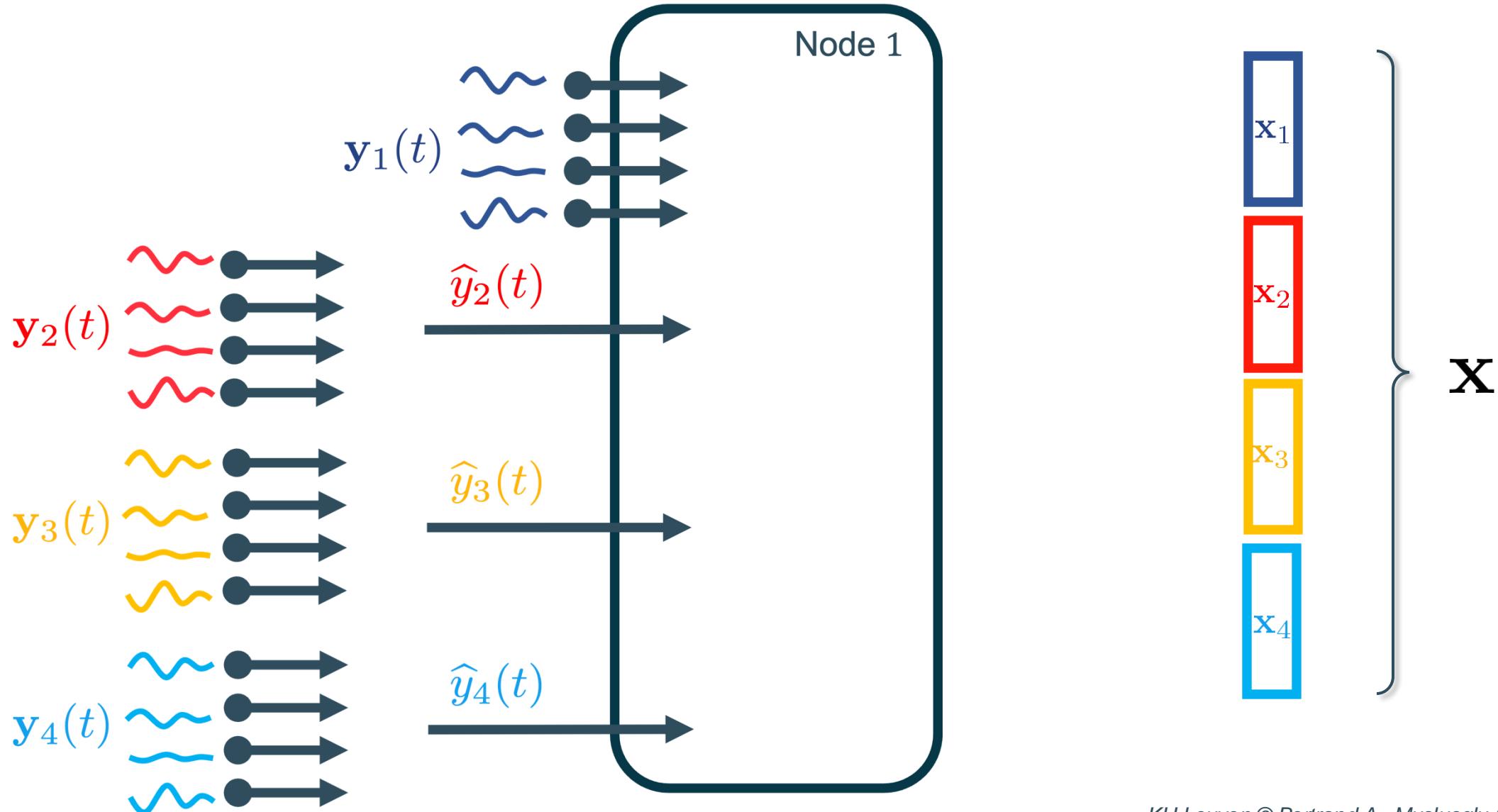
Data flow (Fully-connected network)



Data flow (Fully-connected network)



Data flow (Fully-connected network)



First attempt: alternating optimization / block-coordinate descent / non-linear Gauss-Seidel

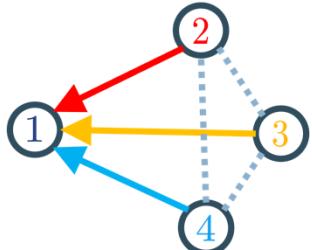
MMSE Example

- Centralized problem:

$$\min_{\mathbf{x}} \mathbb{E} \left[|d(t) - \mathbf{x}^T \mathbf{y}(t)|^2 \right] = \min_{\mathbf{x}} \mathbb{E} \left[|d(t) - \sum_k \mathbf{x}_k^T \mathbf{y}_k(t)|^2 \right]$$

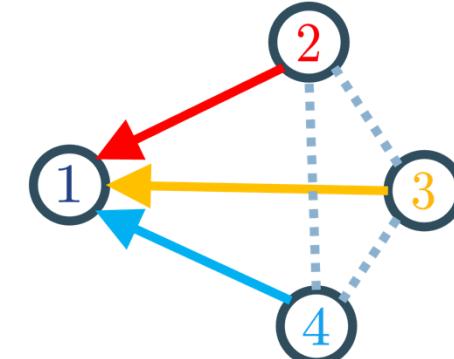
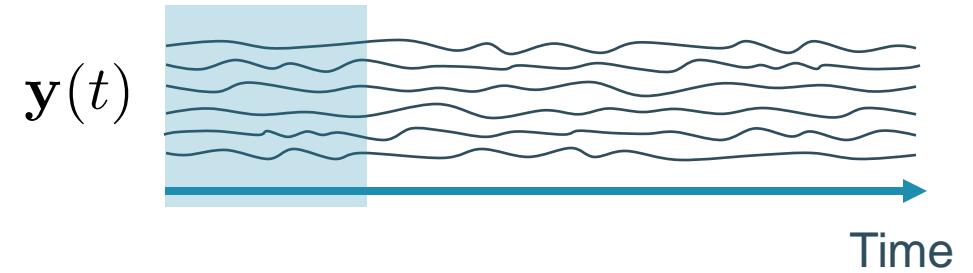
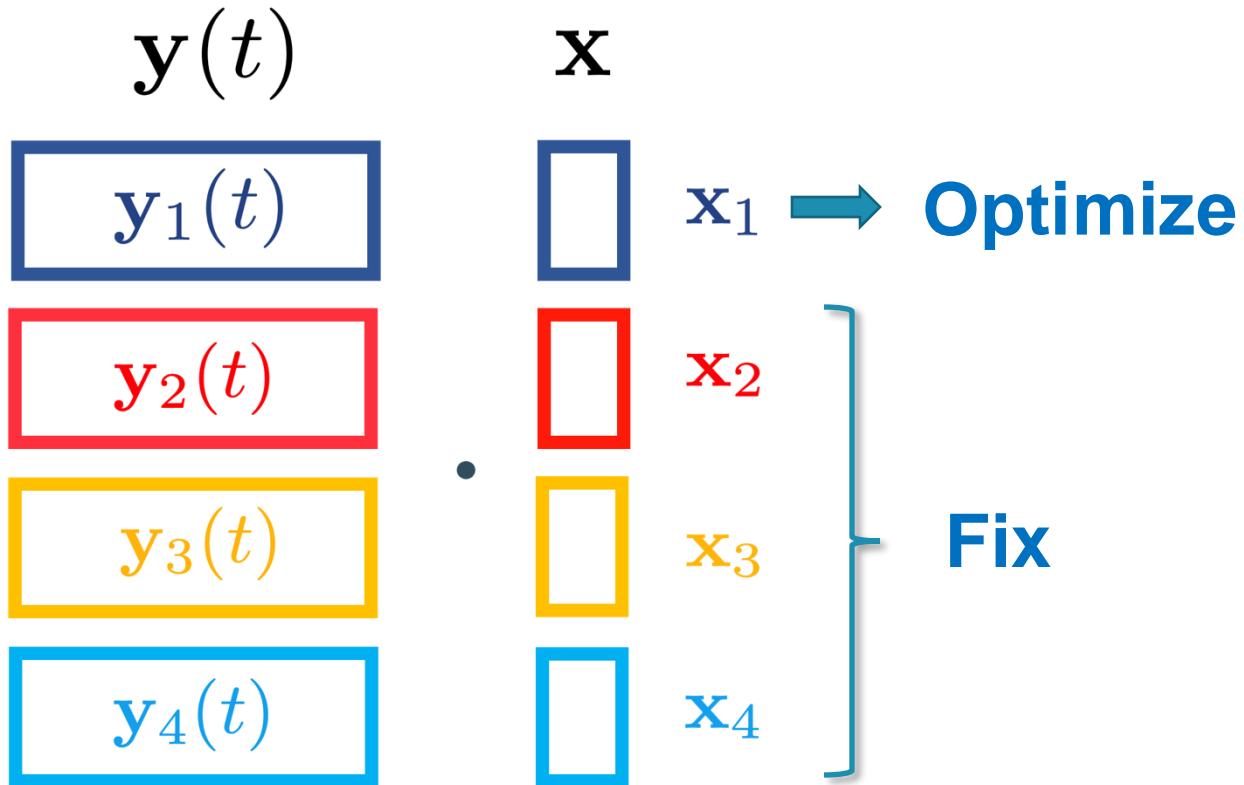
- Local MMSE problem at node 1:

$$\min_{\mathbf{x}_1} \mathbb{E} \left[|d(t) - \mathbf{x}_1^T \mathbf{y}_1(t) - \sum_{k \neq 1} \hat{y}_k(t)|^2 \right]$$



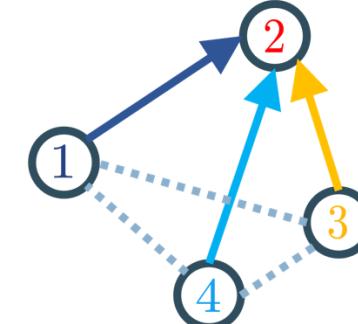
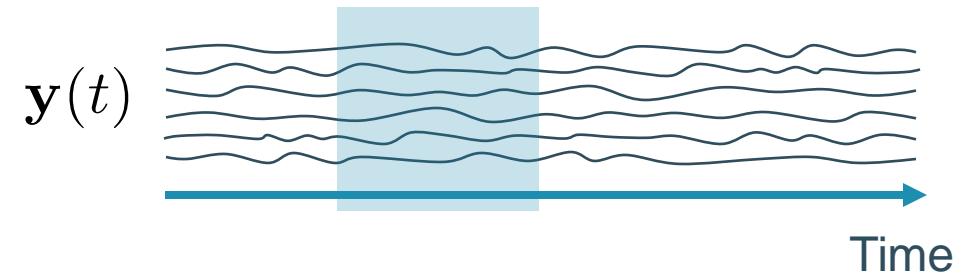
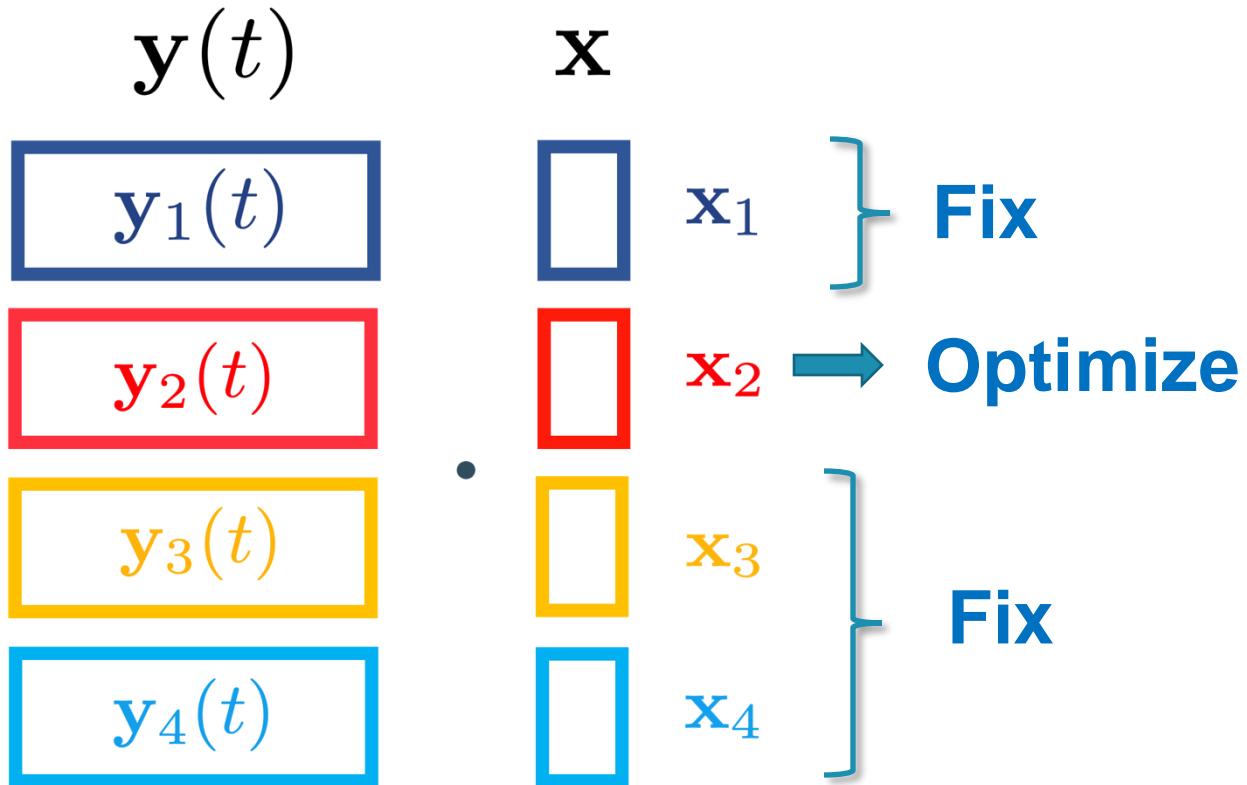
First attempt: alternating optimization / block-coordinate descent / non-linear Gauss-Seidel

- Update in alternating fashion
- Spread iterations over different sample blocks



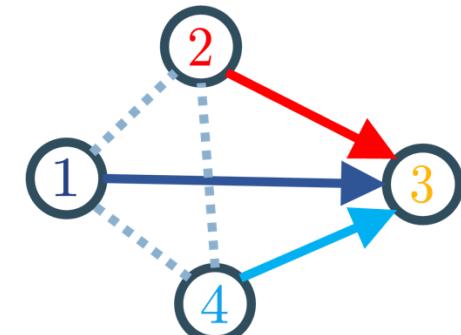
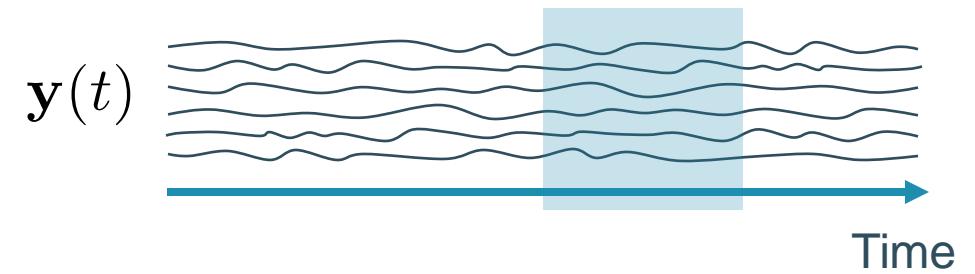
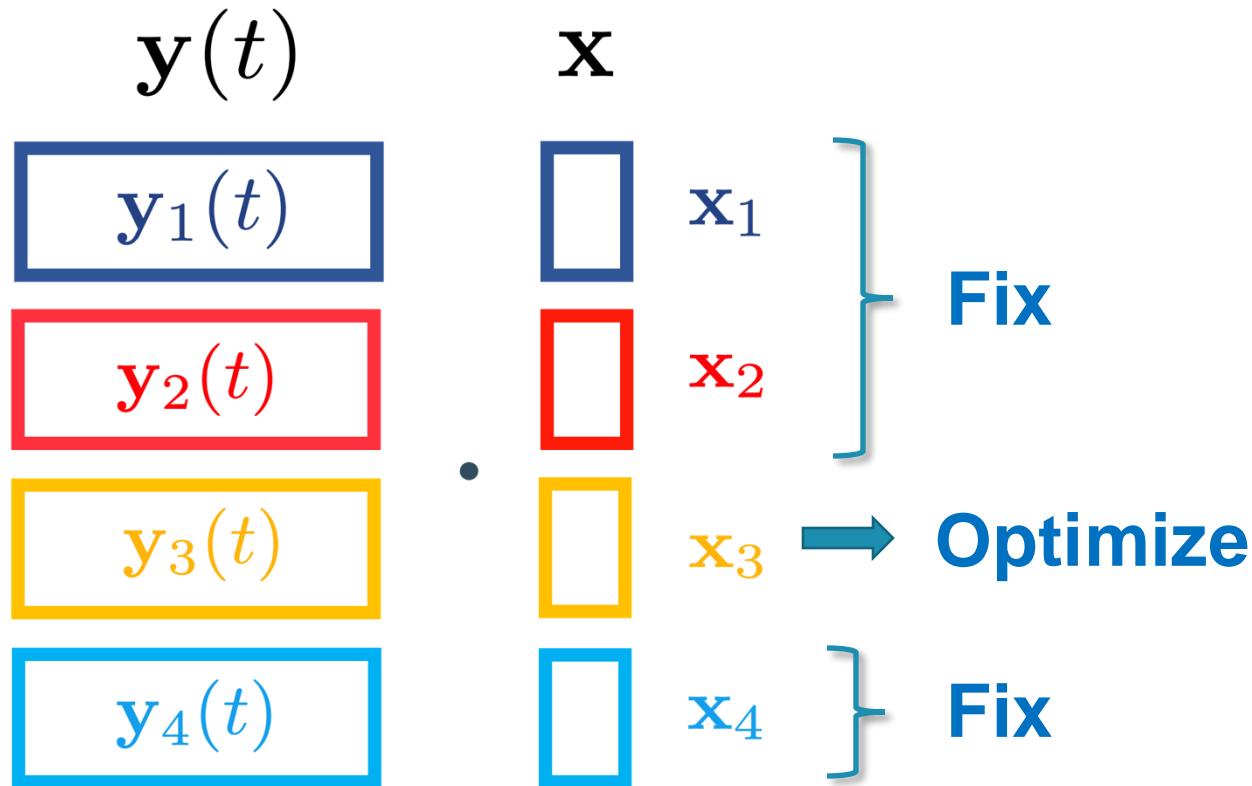
First attempt: alternating optimization / block-coordinate descent / non-linear Gauss-Seidel

- Update in alternating fashion
- Spread iterations over different sample blocks



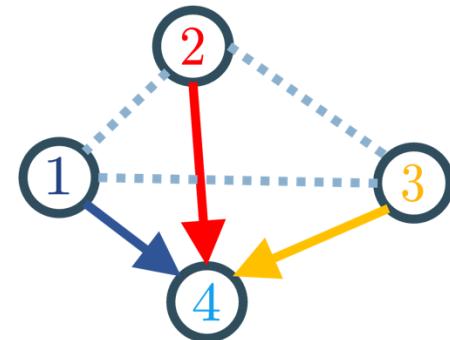
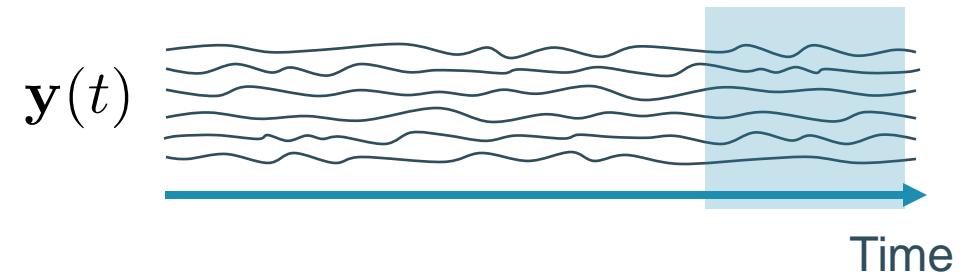
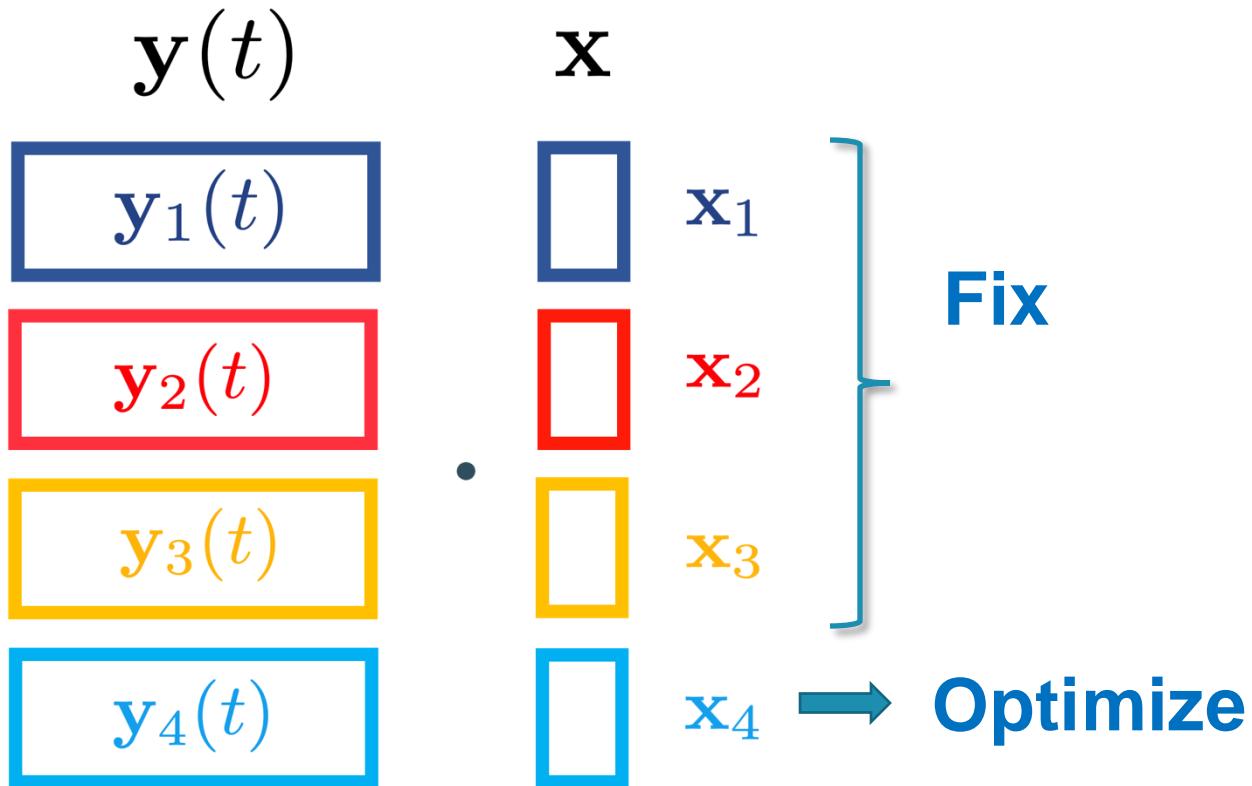
First attempt: alternating optimization / block-coordinate descent / non-linear Gauss-Seidel

- Update in alternating fashion
- Spread iterations over different sample blocks



First attempt: alternating optimization / block-coordinate descent / non-linear Gauss-Seidel

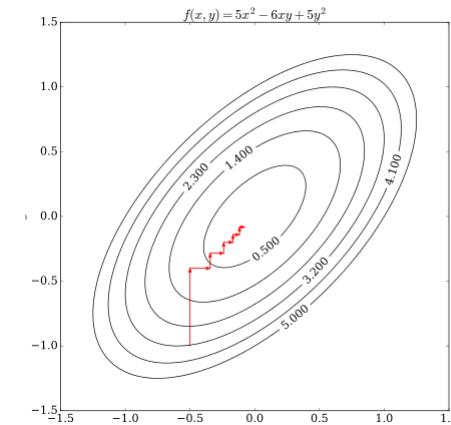
- Update in alternating fashion
- Spread iterations over different sample blocks



First attempt:
alternating optimization / block-coordinate descent / non-linear Gauss-Seidel

Two problems with this approach:

- 1) Fixing all x_k 's except one
 - **inefficient** in moving through optimization landscape
 - effect worsens for large # nodes



- 2) Local sub-problems generally have a **different structure** than the original problem, and can be hard(er) to solve (see next slide)

First attempt: alternating optimization / block-coordinate descent / non-linear Gauss-Seidel

PCA Example

$$\begin{aligned} \max_{\mathbf{x}} \mathbb{E}[|\mathbf{x}^T \mathbf{y}(t)|^2] \\ \text{subject to } \mathbf{x}^T \mathbf{x} = 1 \end{aligned} \quad \longrightarrow \quad \text{Solution is an eigenvalue problem: } R_{\mathbf{y}\mathbf{y}} \mathbf{x} = \lambda \mathbf{x}$$

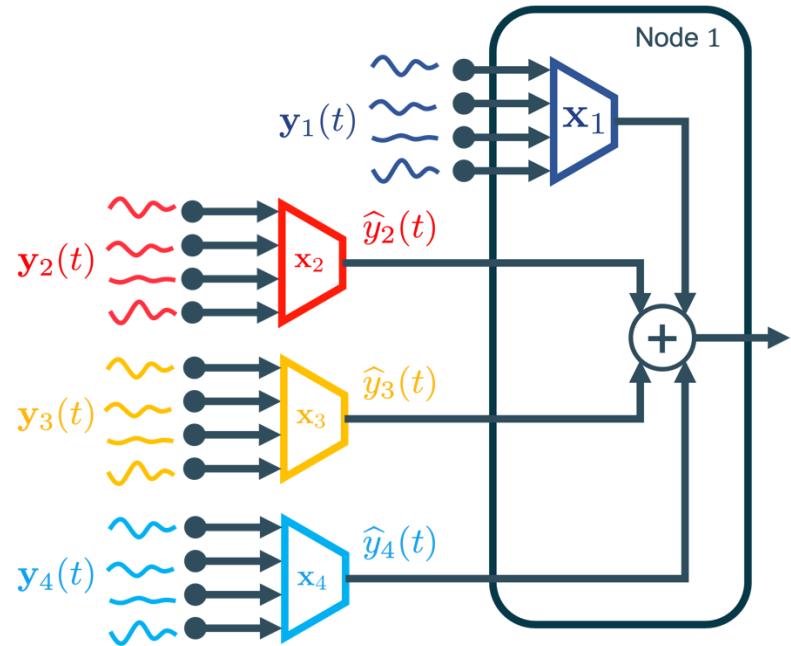
- All x_k are fixed except x_1 $\max_{\mathbf{x}_1} \mathbb{E}[|\mathbf{x}_1^T \mathbf{y}_1(t) + \sum_{k \neq 1} \hat{y}_k(t)|^2]$

$$\text{subject to } \mathbf{x}_1^T \mathbf{x}_1 = 1 - \sum_{k \neq 1} \mathbf{x}_k^T \mathbf{x}_k$$

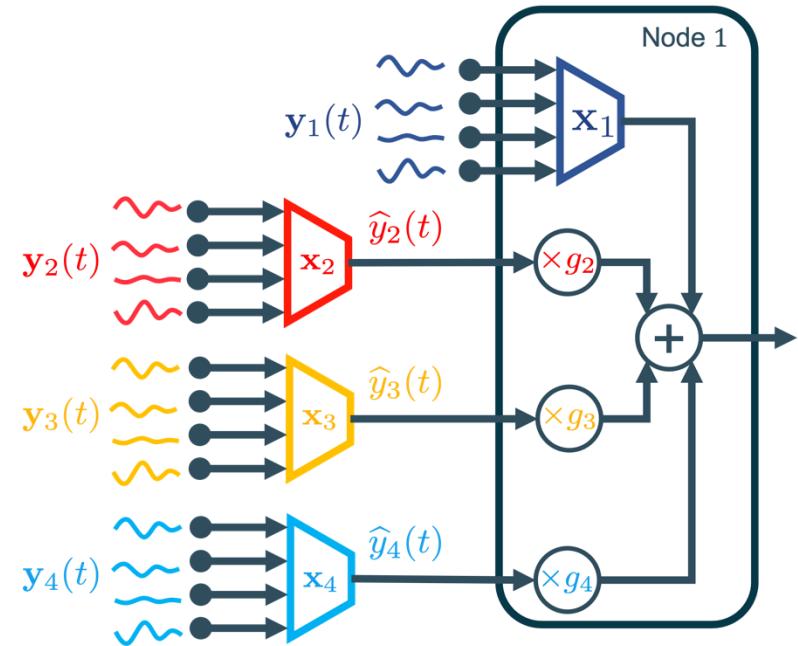
- Solution **not** an eigenvalue problem anymore

$$R_{\mathbf{y}_1 \mathbf{y}_1} \mathbf{x}_1 = \lambda \mathbf{x}_1 - \mathbb{E} \left[\mathbf{y}_1(t) \sum_{k \neq 1} \hat{y}_k(t) \right]$$

Second attempt...



Additional degrees
of freedom



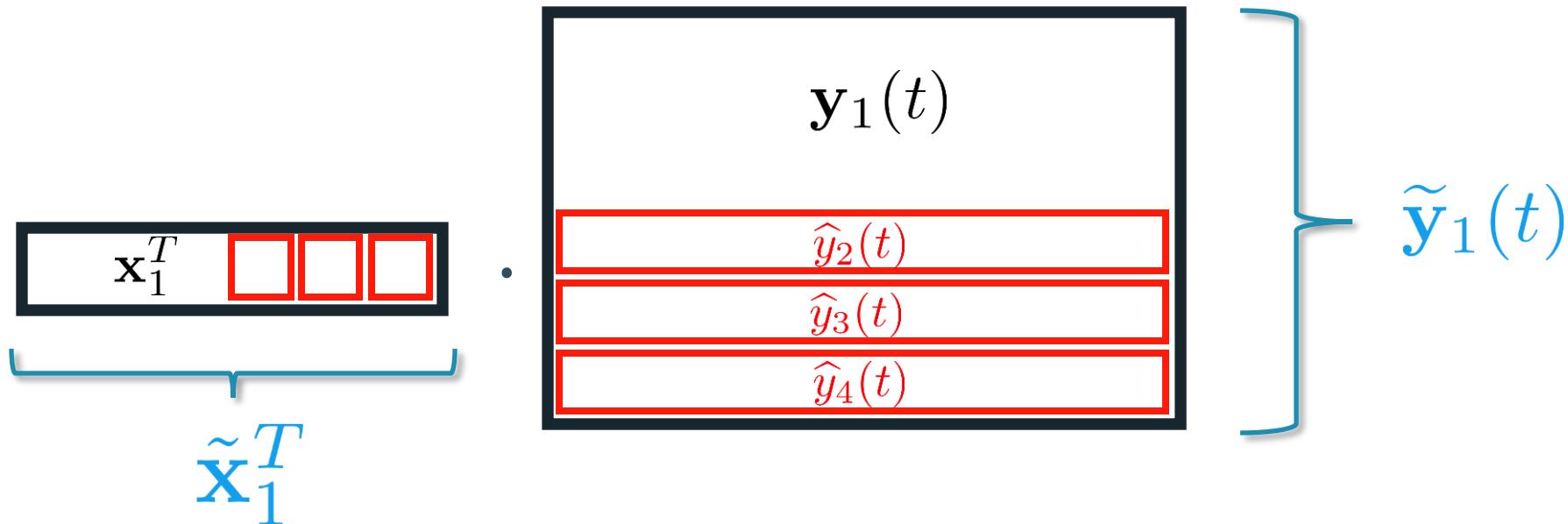
Remember this notation!



θ_k = **original** (raw) data / variable

$\hat{\theta}_k$ = **fused** data / variable

$\tilde{\theta}_k$ = **stack** local θ with $\hat{\theta}$ from neighboring node(s)



Revisiting MMSE example

Local problem at node 1

$$\min_{\tilde{\mathbf{x}}_1} \mathbb{E}[|d(t) - \tilde{\mathbf{x}}_1^T \tilde{\mathbf{y}}_1(t)|^2]$$

$$\Rightarrow \tilde{\mathbf{x}}_1^* = R_{\tilde{\mathbf{y}}_1 \tilde{\mathbf{y}}_1}^{-1} \mathbb{E}[\tilde{\mathbf{y}}_1(t)d(t)]$$

Original (**centralized**) problem:

$$\min_{\mathbf{x}} \mathbb{E}[|d(t) - \mathbf{x}^T \mathbf{y}(t)|^2]$$

$$\Rightarrow \mathbf{x}^* = R_{\mathbf{y} \mathbf{y}}^{-1} \mathbb{E}[\mathbf{y}(t)d(t)]$$



Local problem has **same structure** as original centralized problem, with **smaller dimensions**

Formalization:

II.B- The DASF(*) framework

(*) DASF= *distributed adaptive signal fusion*

(or if you want: distributed adaptive spatial filtering)

The scope of DASF

Example

LS / MMSE: Signal estimation

$$\underset{\mathbf{x}}{\text{minimize}} \ f(\mathbf{x}^T \mathbf{y}(t))$$

$$\underset{\mathbf{x}}{\text{minimize}} \mathbb{E}[| | d(t) - \mathbf{x}^T \mathbf{y}(t) | |]^2$$

The optimization variable x always appears as an inner product with a signal $y(t)$

The scope of DASF

MIMO spatial filters
(Q output channels)

$$\underset{X}{\text{minimize}} \ f(X^T \mathbf{y}(t))$$

$$X = \begin{bmatrix} \mathbf{x}(1) & \dots & \mathbf{x}(Q) \end{bmatrix}$$

Example
LS / MMSE: Signal estimation

$$\underset{X}{\text{minimize}} \mathbb{E}[||\mathbf{d}(t) - X^T \mathbf{y}(t)||]^2$$

The optimization variable X always appears as an inner product with a signal $y(t)$

The scope of DASF

Constraints

$$\underset{X}{\text{minimize}} \ f(X^T \mathbf{y}(t))$$

$$\text{subject to } g_j(X^T \mathbf{y}(t)) = 0,$$

$$h_j(X^T \mathbf{y}(t)) \leq 0$$

The optimization variable X always appears as an inner product with a signal $\mathbf{y}(t)$

The scope of DASF

Multiple signal sets

$$\underset{X}{\text{minimize}} \ f(X^T \mathbf{y}(t), X^T \mathbf{v}(t))$$

subject to $g_j(X^T \mathbf{y}(t), X^T \mathbf{v}(t)) = 0,$
 $h_j(X^T \mathbf{y}(t), X^T \mathbf{v}(t)) \leq 0$

Example

Max-SNR/GEVD: Dimensionality reduction

$$\underset{X}{\text{maximize}} \mathbb{E}[||X^T \mathbf{y}(t)||]^2$$

subject to $\mathbb{E}[X^T \mathbf{n}(t) \mathbf{n}(t)^T X] = I$

The optimization variable X always appears as an inner product with a signal $\mathbf{y}(t), \mathbf{v}(t), \dots$

The scope of DASF

Multiple variables

$$\underset{(X,W)}{\text{minimize}} \ f(X^T \mathbf{y}(t), W^T \mathbf{v}(t))$$

$$\text{subject to } g_j(X^T \mathbf{y}(t), W^T \mathbf{v}(t)) = 0, \\ h_j(X^T \mathbf{y}(t), W^T \mathbf{v}(t)) \leq 0$$

All optimization variables X, W, \dots always appear as an inner product with a signal $\mathbf{y}(t), \mathbf{v}(t), \dots$

Example

CCA: Correlation between data sets

$$\underset{(X,W)}{\text{maximize}} \ \mathbb{E}[\text{tr}(X^T \mathbf{y}(t) \mathbf{v}^T(t) W)]$$

$$\text{subject to } \mathbb{E}[X^T \mathbf{y}(t) \mathbf{y}(t)^T X] = I$$

$$\mathbb{E}[W^T \mathbf{v}(t) \mathbf{v}(t)^T W] = I$$

The scope of DASF

Deterministic matrices

$$\underset{X}{\text{minimize}} \ f(X^T \mathbf{y}(t), X^T B, X^T C)$$

$$\begin{aligned} & \text{subject to } g_j(X^T \mathbf{y}(t), X^T B, X^T C) = 0, \\ & h_j(X^T \mathbf{y}(t), X^T B, X^T C) \leq 0 \end{aligned}$$

Example

(Robust) minimum variance beamforming

$$\underset{X}{\text{minimize}} \mathbb{E}[||X^T \mathbf{y}(t)||]^2$$

$$\begin{aligned} & \text{subject to } X^T B = H, \\ & \text{tr}(X^T X) \leq \alpha^2 \end{aligned}$$

$$X^T X = (X^T \cdot I)(X^T \cdot I)^T = (X^T \cdot C)(X^T \cdot C)^T$$

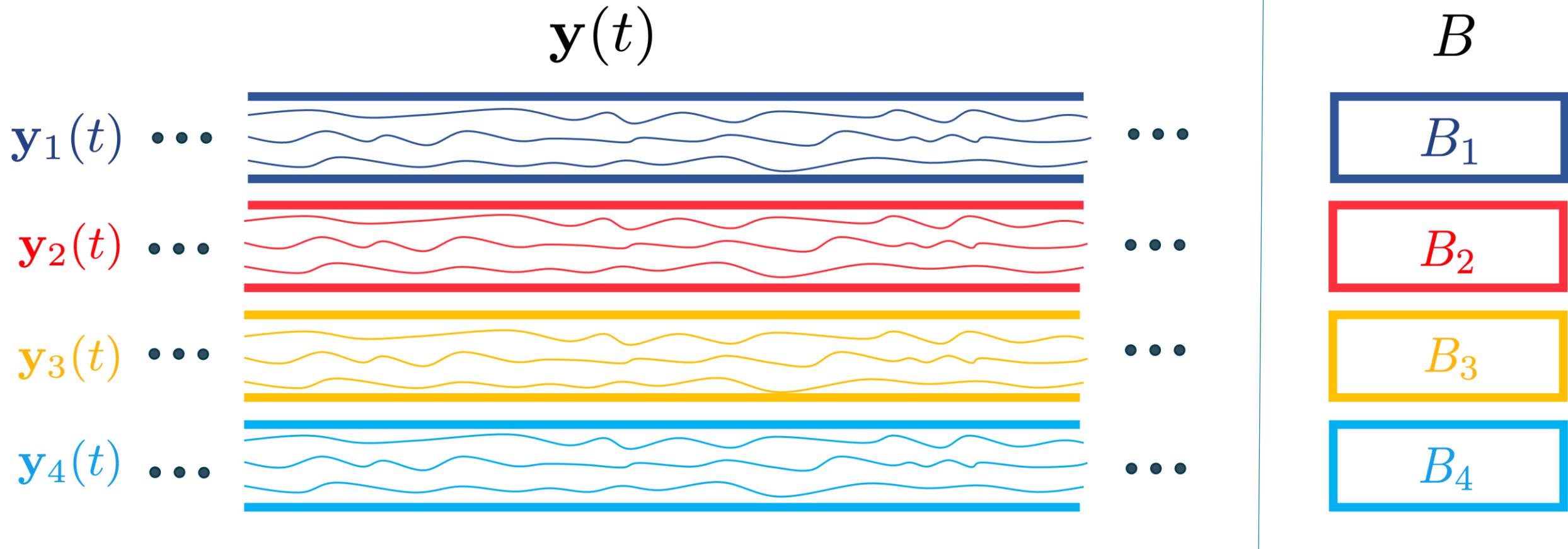
All optimization variables X, W, \dots always appear as an inner product with either:

a signal $\mathbf{y}(t), \mathbf{v}(t), \dots$

... or a deterministic matrix B, C, \dots

The scope of DASF

Deterministic matrices treated similarly as stochastic signal y but are fixed, i.e., not time-dependent



The scope of DASF

$$\underset{X^{(a)}, \forall a}{\text{minimize}} \ f \left(X^{(a)T} \mathbf{y}^{(b)}(t), X^{(a)T} B^{(c)}, \dots \right), \quad \forall a, b, c$$

$$\text{subject to } h_j \left(X^{(a)T} \mathbf{y}^{(b)}(t), X^{(a)T} B^{(c)}, \dots \right) \leq 0 \quad \forall j \in \mathcal{J}_I,$$

$$h_j \left(X^{(a)T} \mathbf{y}^{(b)}(t), X^{(a)T} B^{(c)}, \dots \right) = 0 \quad \forall j \in \mathcal{J}_E.$$

$a \in \mathcal{A}$: Set of all optimization variables

$b \in \mathcal{B}$: Set of all signals

$c \in \mathcal{C}$: Set of all deterministic matrices that interact as in inner product with $X^{(a)}$ s

\mathcal{J}_I : Set of all inequality constraints

\mathcal{J}_E : Set of all equality constraints

All optimization variables $X^{(a)}$ always appear as an inner product with either:

a signal $\mathbf{y}^{(b)}(t)$

... or a deterministic matrix $B^{(c)}$

... let's keep it simple today

$$\underset{X}{\text{minimize}} \ f(X^T \mathbf{y}(t), X^T B)$$

$$\begin{aligned} \text{subject to } g_j(X^T \mathbf{y}(t), X^T B) &= 0, & \forall j \in \mathcal{J}_I, \\ h_j(X^T \mathbf{y}(t), X^T B) &\leq 0 & \forall j \in \mathcal{J}_E. \end{aligned}$$

X always appears as an inner product with :

signal $y(t)$

... or deterministic matrix B

A trick to remember



If X appears alone or in an inner product with itself, e.g.,

$$\underset{X}{\text{minimize}} \ \|X\|_F$$



(with $B = I$)

$$\underset{X}{\text{minimize}} \ \|X^T B\|_F$$

$$\begin{aligned} & \underset{X}{\text{minimize}} \ f(X^T \mathbf{y}(t), X^T B) \\ & \text{subject to } g_j(X^T \mathbf{y}(t), X^T B) = 0, \\ & \quad h_j(X^T \mathbf{y}(t), X^T B) \leq 0 \end{aligned}$$

Functions of sums

Loss and constraints only depend on X through $X^T \mathbf{y}(t)$ and possibly on $X^T B$

$$f(\begin{matrix} X \\ \textcolor{blue}{\boxed{}} \\ \textcolor{red}{\boxed{}} \\ \textcolor{yellow}{\boxed{}} \\ \textcolor{cyan}{\boxed{}} \end{matrix} \cdot \begin{matrix} \mathbf{y}(t) \\ \textcolor{red}{\boxed{}} \\ \textcolor{yellow}{\boxed{}} \\ \textcolor{cyan}{\boxed{}} \end{matrix}) = f\left(\sum_k X_k^T \mathbf{y}_k(t)\right)$$

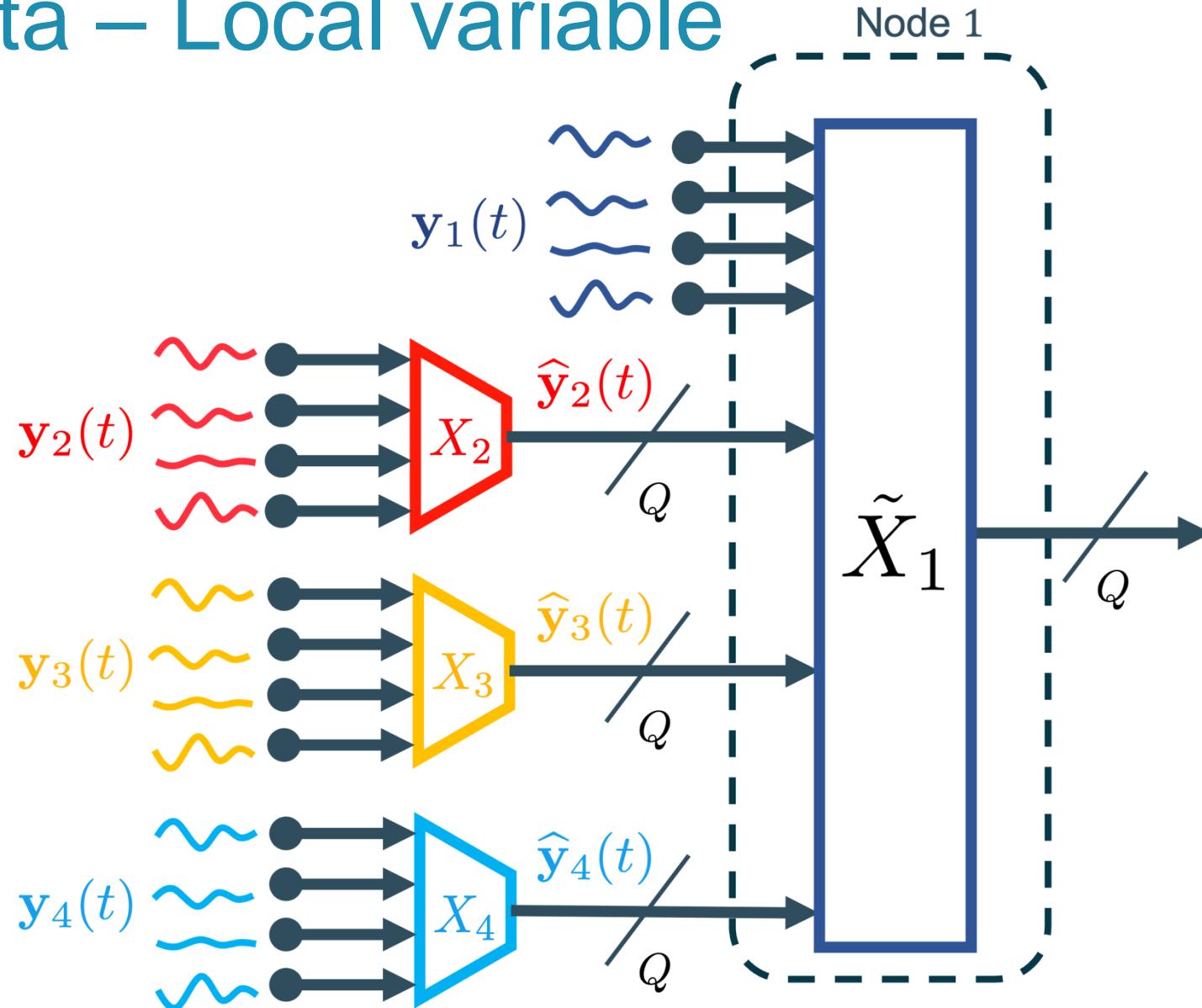
Functions of sums

Loss and constraints only depend on X through $X^T y(t)$ and possibly on $X^T B$

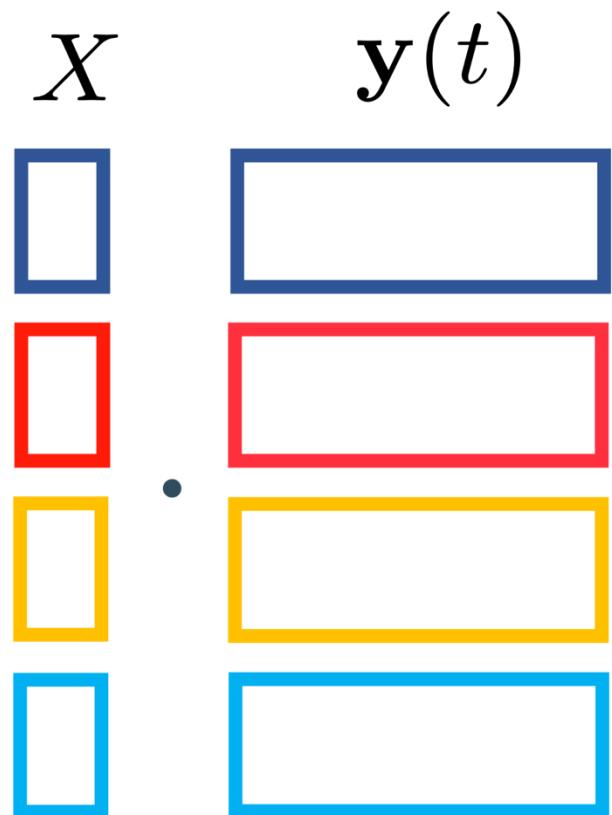
$$f(\begin{array}{c} X \\ \cdot \\ B \end{array}) = f(\sum_k X_k^T B_k)$$

The diagram illustrates the expression $\sum_k X_k^T B_k$. It shows two sets of boxes, X and B , each containing four colored boxes (blue, red, yellow, blue). A dot between the two sets indicates element-wise multiplication. Below the sets, a brace groups them together and is labeled $X^T B$.

Local data – Local variable



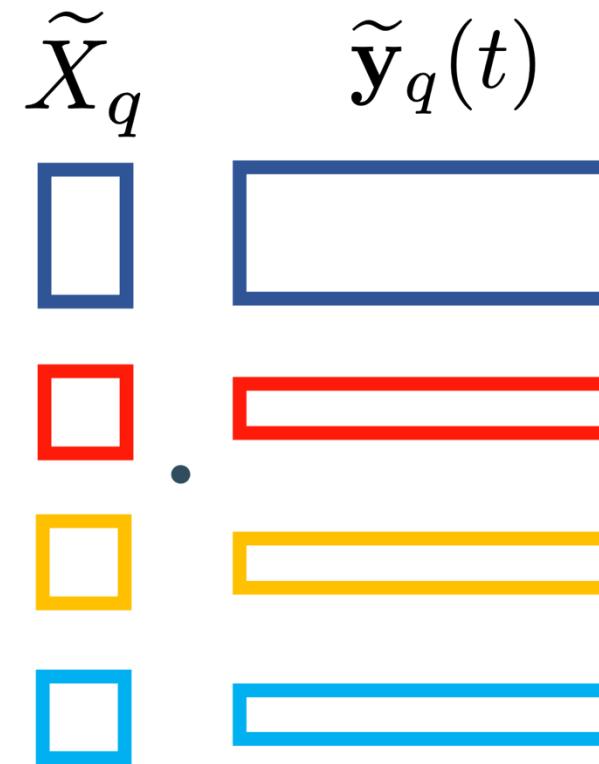
Local data – Local variable



$$X^T \mathbf{y}(t)$$



At node q



$$\tilde{X}_q^T \tilde{\mathbf{y}}_q(t)$$

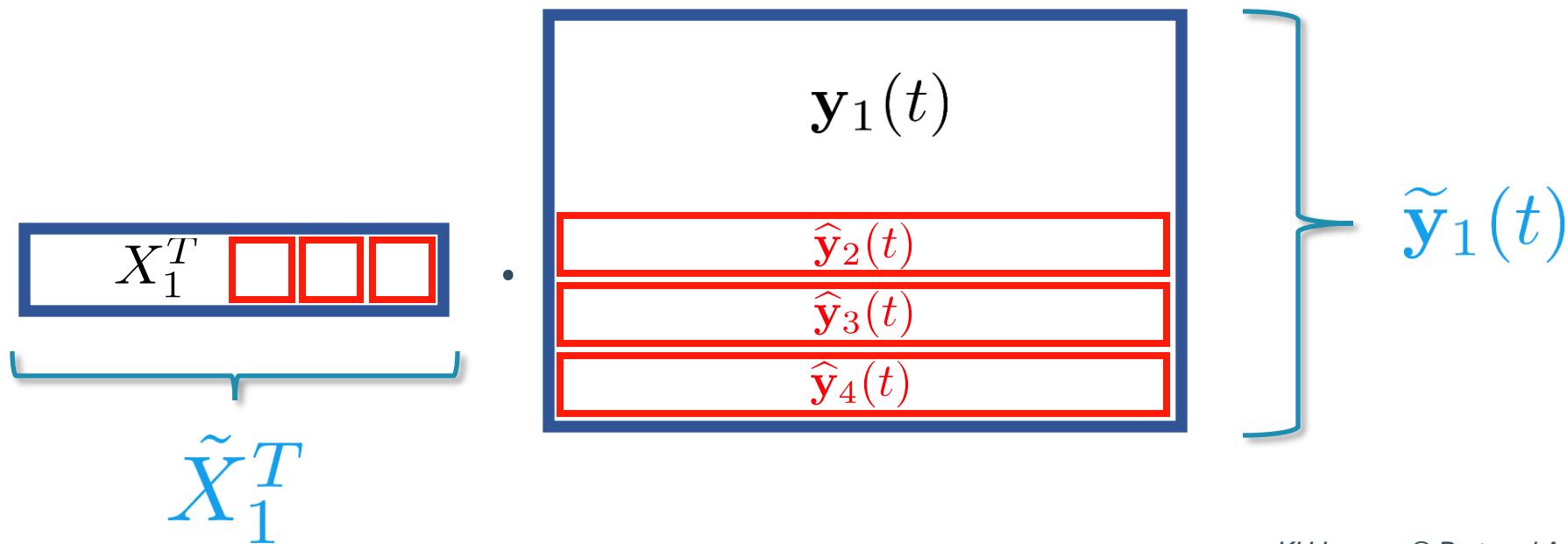
Remember this notation!



θ = **original** (raw) data / variable

$\hat{\theta}_k$ = **fused** data / variable

$\tilde{\theta}_k$ = **stack** local θ with $\hat{\theta}_k$ from neighboring node(s)

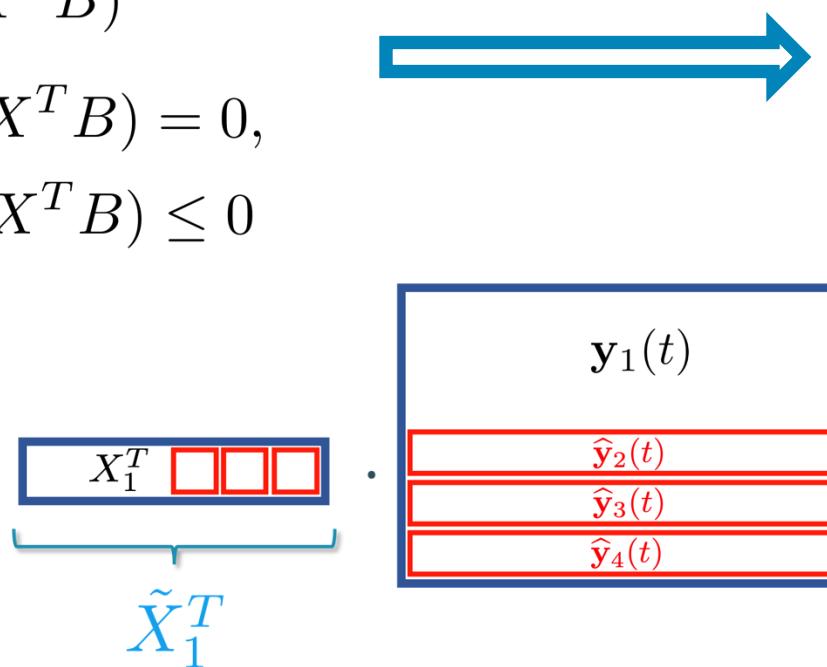


Local problems ~ Centralized problems

Original (**centralized**) problem:

$$\underset{X}{\text{minimize}} \ f(X^T \mathbf{y}(t), X^T B)$$

$$\text{subject to } g_j(X^T \mathbf{y}(t), X^T B) = 0, \\ h_j(X^T \mathbf{y}(t), X^T B) \leq 0$$



Local problem at node k

$$\underset{\tilde{X}_k}{\text{minimize}} \ f(\tilde{X}_k^T \tilde{\mathbf{y}}_k(t), \tilde{X}_k^T \tilde{B}_k)$$

$$\text{subject to } g_j(\tilde{X}_k^T \tilde{\mathbf{y}}_k(t), \tilde{X}_k^T \tilde{B}_k) = 0, \\ h_j(\tilde{X}_k^T \tilde{\mathbf{y}}_k(t), \tilde{X}_k^T \tilde{B}_k) \leq 0$$

$$\left. \begin{array}{c} \mathbf{y}_1(t) \\ \hat{\mathbf{y}}_2(t) \\ \hat{\mathbf{y}}_3(t) \\ \hat{\mathbf{y}}_4(t) \end{array} \right\} \tilde{\mathbf{y}}_1(t)$$

'Plug and play': Can use the same solver as for original (centralized) problem

Local problems ~ Centralized problems

	Global	Local
MMSE:	$\underset{X}{\text{minimize}} \mathbb{E}[\mathbf{d} - X^T \mathbf{y} ^2]$	$\underset{\tilde{X}_k}{\text{minimize}} \mathbb{E}[\mathbf{d} - \tilde{X}_k^T \tilde{\mathbf{y}}_k ^2]$
LCMV:	$\underset{X}{\text{minimize}} \mathbb{E}[X^T \mathbf{y} ^2]$ subject to $X^T B = H$	$\underset{\tilde{X}_k}{\text{minimize}} \mathbb{E}[\tilde{X}_k^T \tilde{\mathbf{y}}_k ^2]$ subject to $\tilde{X}_k^T \tilde{B}_k = H$
CCA:	$\underset{X, W}{\text{maximize}} \mathbb{E}[\text{tr}(X^T \mathbf{y} \mathbf{v}^T W)]$ subject to $\mathbb{E}[X^T \mathbf{y} \mathbf{y}^T X] = I_Q$ $\mathbb{E}[W^T \mathbf{v} \mathbf{v}^T W] = I_Q$	$\underset{\tilde{X}_k, \tilde{W}_k}{\text{maximize}} \mathbb{E}[\text{tr}(\tilde{X}_k^T \tilde{\mathbf{y}}_k \tilde{\mathbf{v}}_k^T \tilde{W}_k)]$ subject to $\mathbb{E}[\tilde{X}_k^T \tilde{\mathbf{y}}_k \tilde{\mathbf{y}}_k^T \tilde{X}_k] = I_Q$ $\mathbb{E}[\tilde{W}_k^T \tilde{\mathbf{v}}_k \tilde{\mathbf{v}}_k^T \tilde{W}_k] = I_Q$

Local problems ~ Centralized problems

	Global	Local
PCA:	$\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} \mathbb{E}[\mathbf{x}^T \mathbf{y}(t) ^2] \\ & \text{subject to } \mathbf{x}^T \mathbf{x} = 1 \end{aligned}$ <p>(with $B = I$)  Need to rewrite to fit DASF framework</p> $\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} \mathbb{E}[\mathbf{x}^T \mathbf{y}(t) ^2] \\ & \text{subject to } (\mathbf{x}^T B) (B^T \mathbf{x}) = 1 \end{aligned}$ <p> $R_{\mathbf{y}\mathbf{y}} \mathbf{x}^* = \lambda(BB^T) \mathbf{x}^*$ generalized eigenvalue problem</p>	$\begin{aligned} & \underset{\tilde{\mathbf{x}}_k}{\text{minimize}} \mathbb{E}[\tilde{\mathbf{x}}_k^T \tilde{\mathbf{y}}_k(t) ^2] \\ & \text{subject to } (\tilde{\mathbf{x}}_k^T \tilde{B}_k) (\tilde{B}_k^T \tilde{\mathbf{x}}_k) = 1 \end{aligned}$ <p> $R_{\tilde{\mathbf{y}}_k \tilde{\mathbf{y}}_k} \tilde{\mathbf{x}}_k^* = \lambda(\tilde{B}_k \tilde{B}_k^T) \tilde{\mathbf{x}}_k^*$ generalized eigenvalue problem</p>

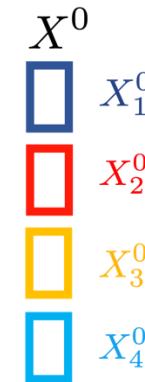
III- The fully-connected DASF algorithm

(FC-DASF)

III.A- Algorithm derivation

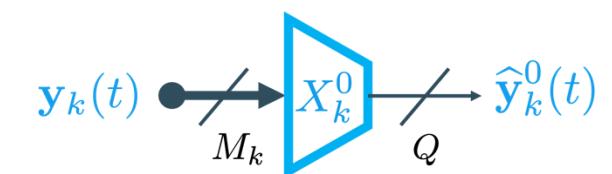
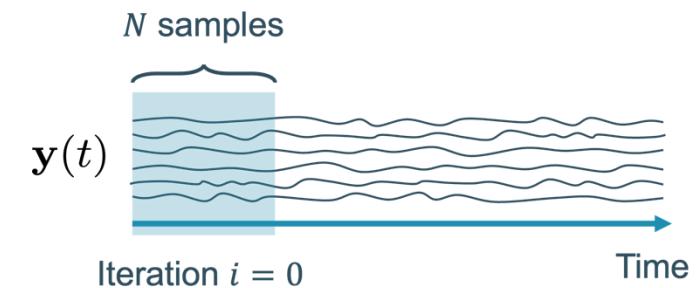
The FC-DASF algorithm

0) Initialization: X^i initialized randomly for $i = 0$



1) Sensor data collection & compression:

- Each node k collects N samples of \mathbf{y}_k
- Each node k compresses its N samples : $\hat{\mathbf{y}}_k^0 = X_k^{0T} \mathbf{y}_k$
- ... and does the same for B_k (if applicable): $\hat{B}_k^0 = X_k^{0T} B_k$

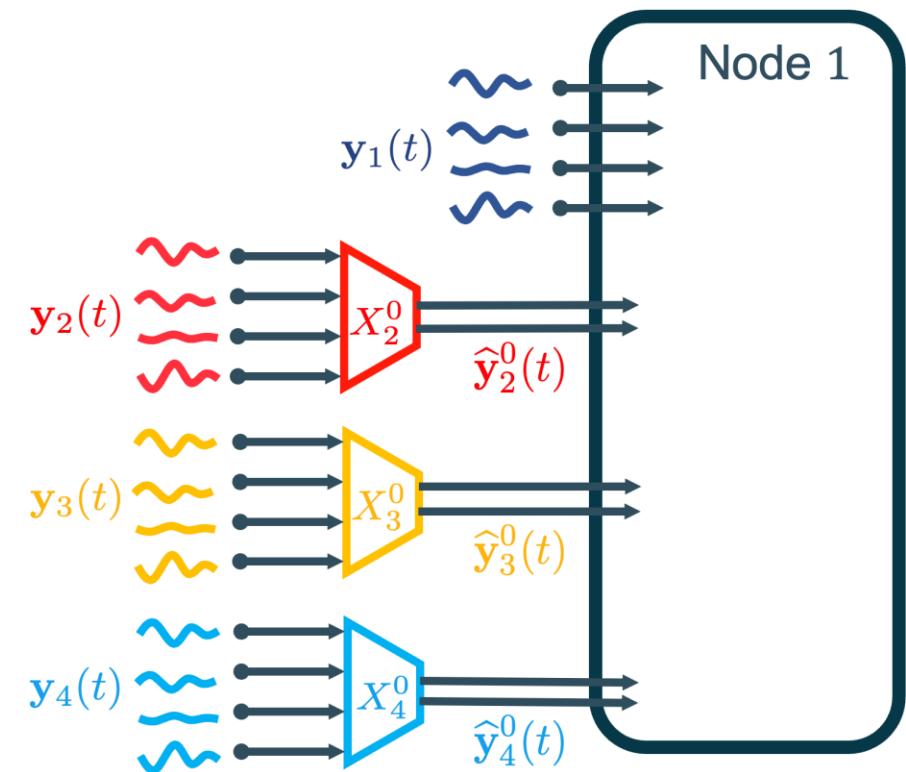


Note: compression factor is M_k/Q **with Q the #columns of X**, i.e., the number of desired outputs in the spatial filter (**often Q=1**)

The FC-DASF algorithm

2) Transmit compressed data to updating node

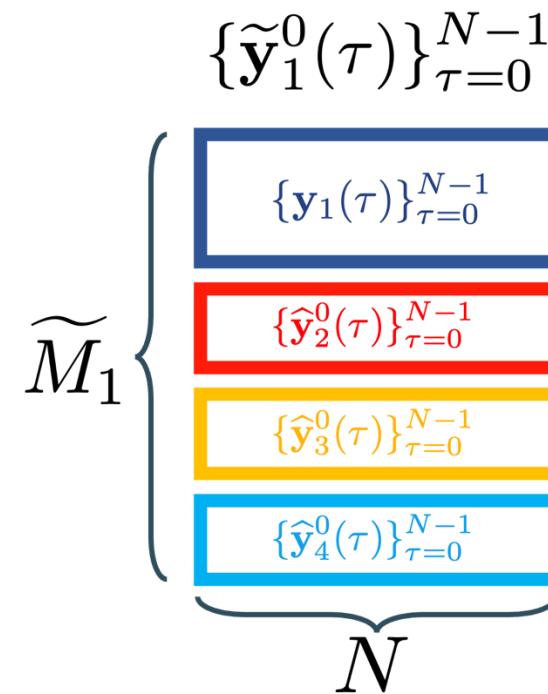
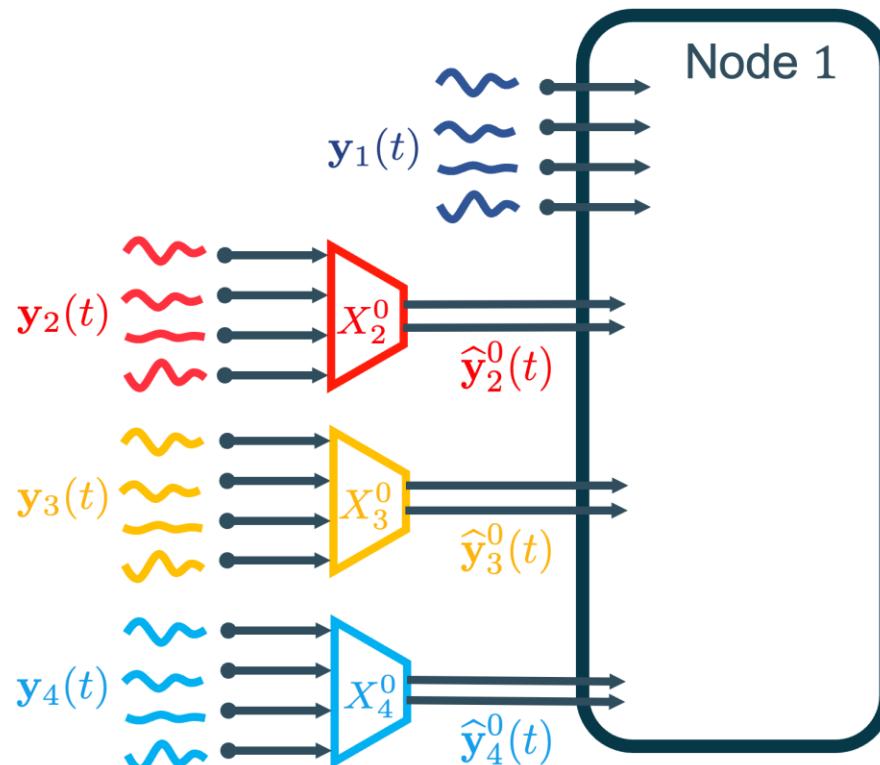
- Select new updating node q (in this example $q = 1$)
- Nodes $k \neq q$ transmit N samples of \hat{y}_k^0 to node q
(+ \hat{B}_k^0 if applicable, negligible communication cost)



The FC-DASF algorithm

3) Solve local problem at updating node $q = 1$

- Node $q = 1$ has access to N samples $\{\tilde{y}_q^0(\tau)\}_{\tau=0}^{N-1}$ and to \tilde{B}_q^0



The FC-DASF algorithm

4) Solve local problem at updating node $q = 1$

- Compute \tilde{X}_1^1 as the solution of

$$\tilde{X}_1^1 \triangleq \underset{\tilde{X}_1}{\operatorname{argmin}} f(\tilde{X}_1^T \tilde{\mathbf{y}}_1^0(t), \tilde{X}_1^T \tilde{B}_1^0)$$

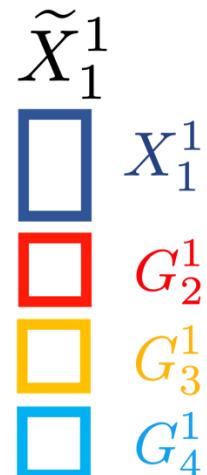
subject to $g_j(\tilde{X}_1^T \tilde{\mathbf{y}}_1^0(t), \tilde{X}_1^T \tilde{B}_1^0) = 0,$

$$h_j(\tilde{X}_1^T \tilde{\mathbf{y}}_1^0(t), \tilde{X}_1^T \tilde{B}_1^0) \leq 0$$

- Iteration index i
- Updating node index q

'Plug and play': Can use the same solver as for original (centralized) problem

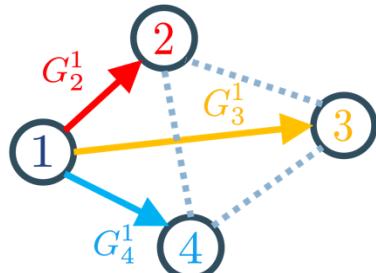
- Partition \tilde{X}_1^1



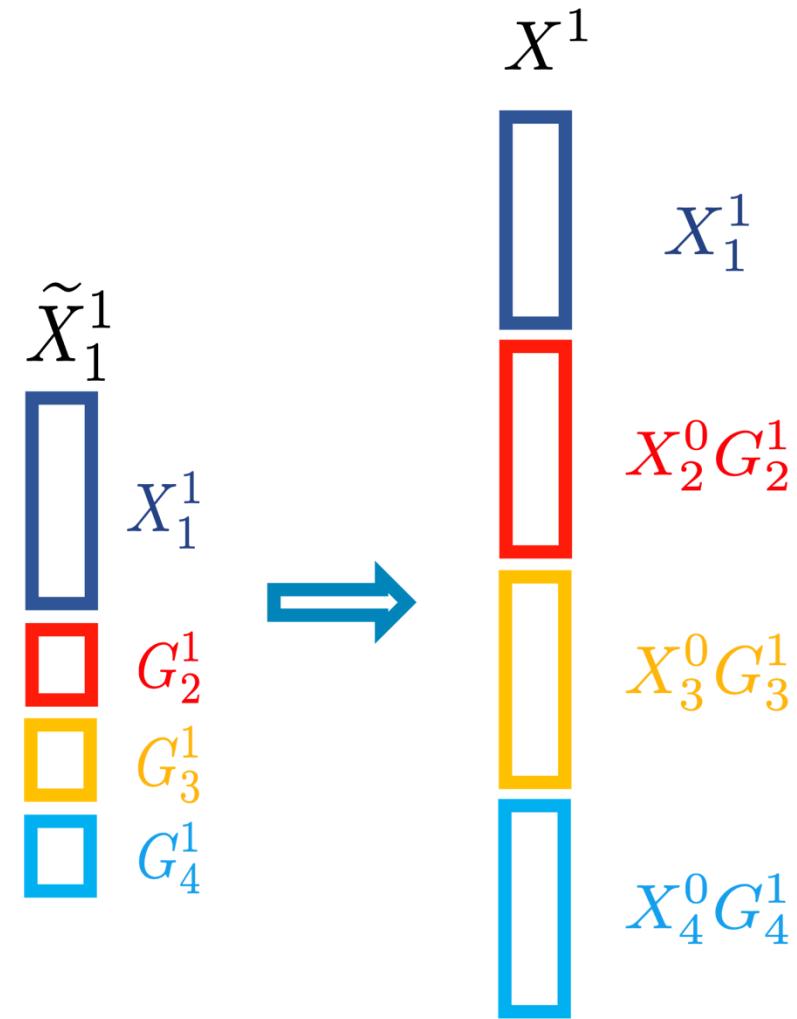
The FC-DASF algorithm

5) Update all nodes

- Node $q = 1$ transmits $Q \times Q$ matrices G_k^1



- X^1 is the new estimate of X at iteration $i = 1$



The FC-DASF algorithm

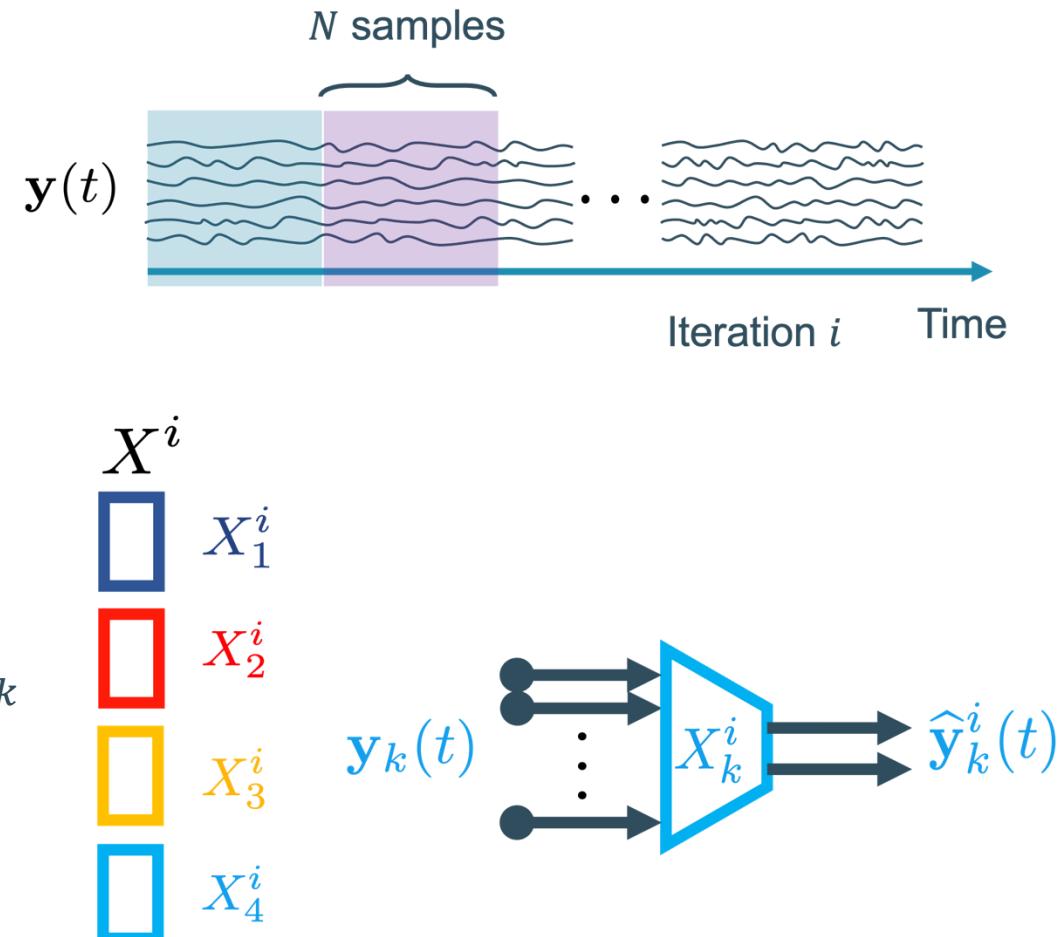
6) If desired: compute output of spatial filter for current block of N samples

- At updating node $q = 1$: $\left\{ \mathbf{z}(\tau) = (\tilde{\mathbf{X}}_q^1)^T \tilde{\mathbf{y}}_q^0(\tau) \right\}_{\tau=0}^{N-1}$
- If needed: transmit $\{\mathbf{z}(\tau)\}_{\tau=0}^{N-1}$ from node q to any node that is a data sink

The FC-DASF algorithm

1) Sensor data collection & compression:

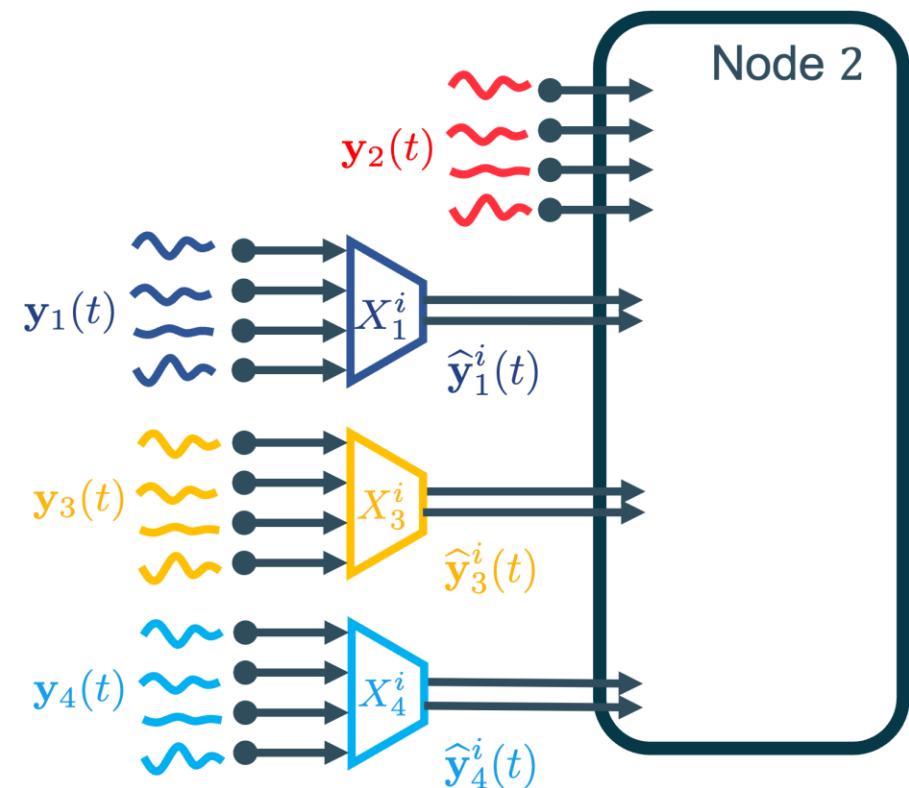
- Each node k collects N new samples of \mathbf{y}_k
- Each node k compresses its N samples : $\hat{\mathbf{y}}_k^i = \mathbf{X}_k^{iT} \mathbf{y}_k$
- ... and does the same for B_k (if applicable): $\hat{B}_k^i = \mathbf{X}_k^{iT} B_k$



The FC-DASF algorithm

2) Transmit compressed data to updating node

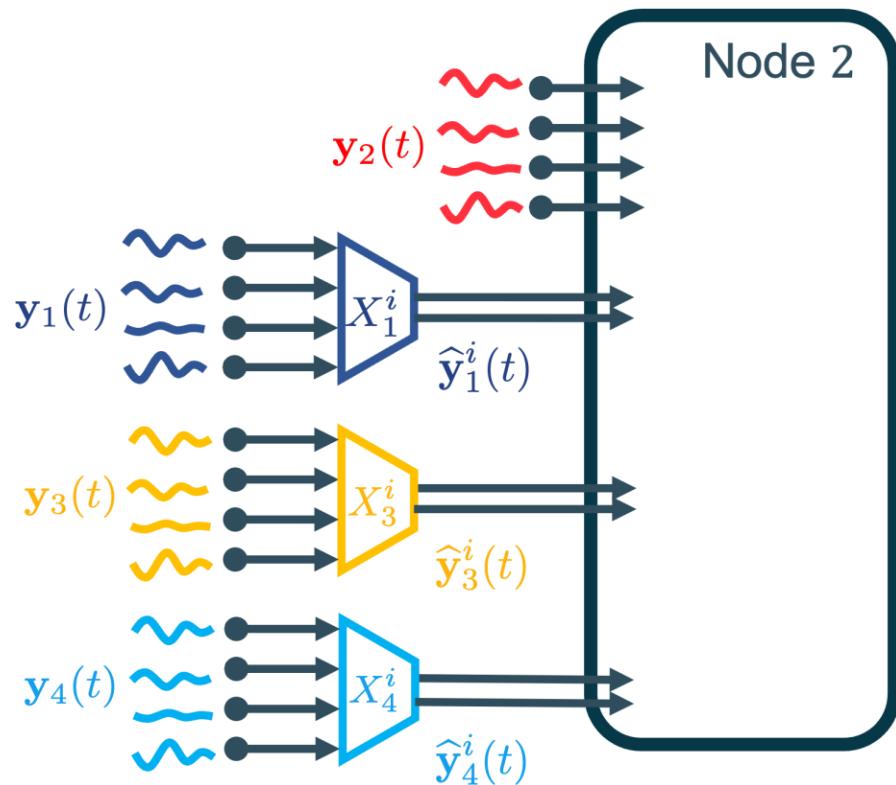
- Select new updating node q (in this example $q = 2$)
- Nodes $k \neq q$ transmit N samples of \hat{y}_k^i to node q
(+ \hat{B}_k^i if applicable)



The FC-DASF algorithm

3) Solve local problem at updating node $q = 2$

- Node $q = 2$ has access to N samples $\{\tilde{y}_q^i(\tau)\}_{\tau=iN}^{iN+N-1}$ and to \tilde{B}_q^i



$$\widetilde{M}_2 \left\{ \begin{array}{l} \{\tilde{y}_2^i(\tau)\}_{\tau=iN}^{iN+N-1} \\ \{\hat{y}_1^i(\tau)\}_{\tau=iN}^{iN+N-1} \\ \{\hat{y}_3^i(\tau)\}_{\tau=iN}^{iN+N-1} \\ \{\hat{y}_4^i(\tau)\}_{\tau=iN}^{iN+N-1} \end{array} \right.$$

$$\widetilde{M}_2 = M_2 + Q(K - 1)$$

The FC-DASF algorithm

4) Solve local problem at updating node $q = 2$

- Compute \tilde{X}_q^{i+1} as the solution of

$$\tilde{X}_q^{i+1} \triangleq \underset{\tilde{X}_q}{\operatorname{argmin}} f(\tilde{X}_q^T \tilde{\mathbf{y}}_q^i(t), \tilde{X}_q^T \tilde{B}_q^i)$$

subject to $g_j(\tilde{X}_q^T \tilde{\mathbf{y}}_q^i(t), \tilde{X}_q^T \tilde{B}_q^i) = 0,$

$$h_j(\tilde{X}_q^T \tilde{\mathbf{y}}_q^i(t), \tilde{X}_q^T \tilde{B}_q^i) \leq 0$$

● Iteration index i
● Updating node index q

'Plug and play': Can use the same solver as for original (centralized) problem

- Partition \tilde{X}_2^{i+1}

$$\tilde{X}_2^{i+1}$$

$$\square X_2^{i+1}$$

$$\square G_1^{i+1}$$

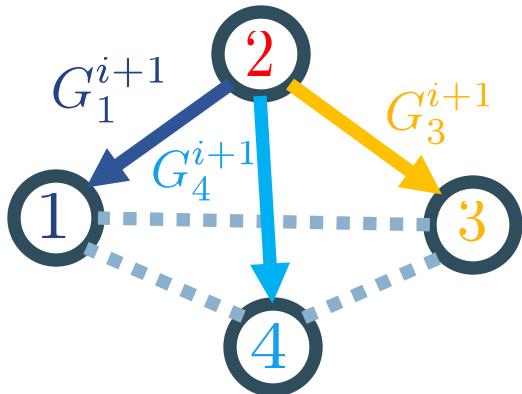
$$\square G_3^{i+1}$$

$$\square G_4^{i+1}$$

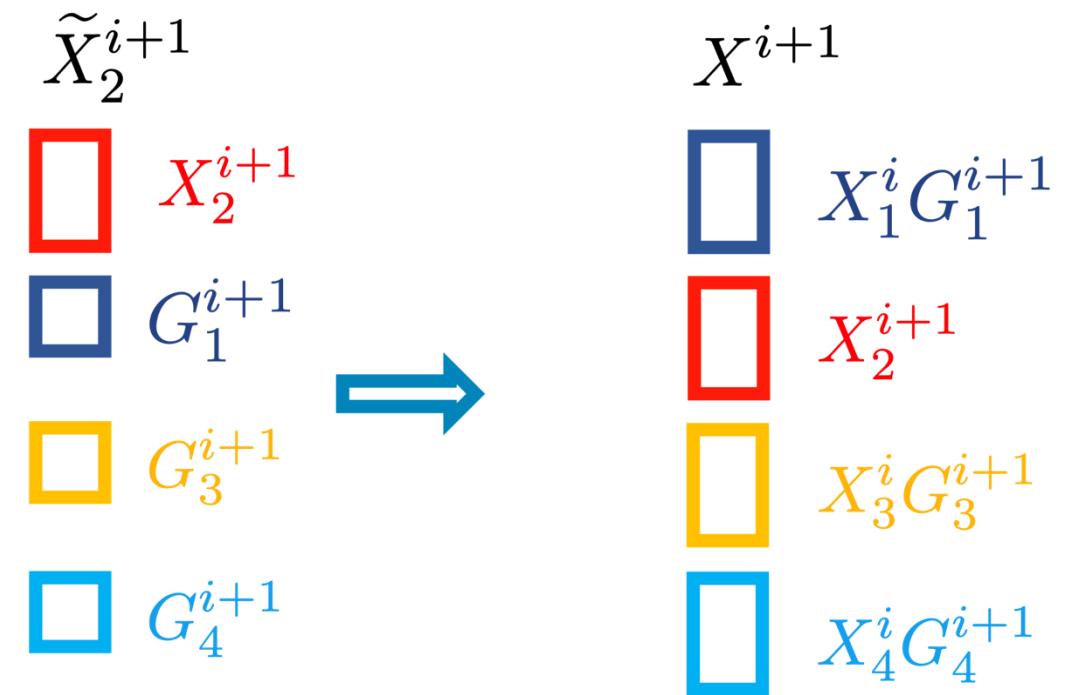
The FC-DASF algorithm

5) Update all nodes

- Node $q = 2$ transmits $Q \times Q$ matrices G_k^{i+1}



- X^{i+1} is the new estimate of X at iteration $i + 1$



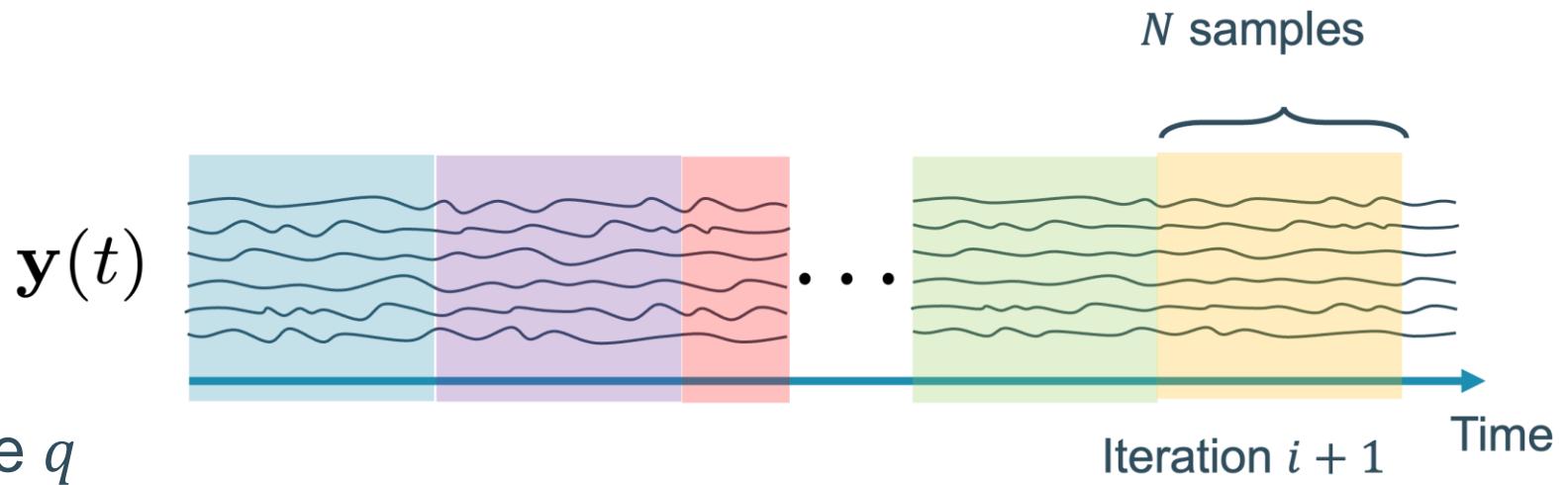
The FC-DASF algorithm

6) If desired: compute output of spatial filter for current block of N samples

- At updating node $q = 2$: $\{\mathbf{z}(\tau) = (\tilde{X}_q^{i+1})^T \tilde{\mathbf{y}}_q^i(\tau)\}_{\tau=iN}^{iN+N-1}$
- If needed: transmit $\{\mathbf{z}(\tau)\}_{\tau=iN}^{iN+N-1}$ from node q to any node that is a data sink

The FC-DASF algorithm

- New samples



- Select new updating node q

- Repeat procedure

Summary of FC-DASF

Algorithm 1: Fully-Connected Distributed Adaptive Signal Fusion (FC-DASF) Algorithm

Initialization

Initialize X^0 , $i \leftarrow 0$.

repeat

Select updating node q

Choose the updating node as $q \leftarrow (i \bmod K) + 1$.

Collect N new samples,
compress and transmit

1) Every node k collects a new batch of N samples of \mathbf{y}_k ,
compresses these to N samples of $\hat{\mathbf{y}}_k^i$ using $\hat{\mathbf{y}}_k^i(t) = X_k^{iT} \mathbf{y}_k(t)$ and
transmits them to node q .

\hat{B}_k^i is computed as $X_k^{iT} B_k$ and transmitted to node q .

at Node q do

Solve local problem at
node q

2a) Compute \tilde{X}_q^{i+1} as the solution of the local problem. If the
solution is not unique, select the solution which minimizes
 $\|\tilde{X}_q^{i+1} - \tilde{X}_q^i\|_F$ with $\tilde{X}_q^i = [X_q^{iT}, I_Q, \dots, I_Q]^T$.

2b) Partition X_q^{i+1} as $[X_q^T, G_1^T, \dots, G_{q-1}^T, G_{q+1}^T, \dots, G_K^T]^T$.

2c) Transmit G_k^{i+1} to node k for every $k \neq q$.

end

Disseminate G 's

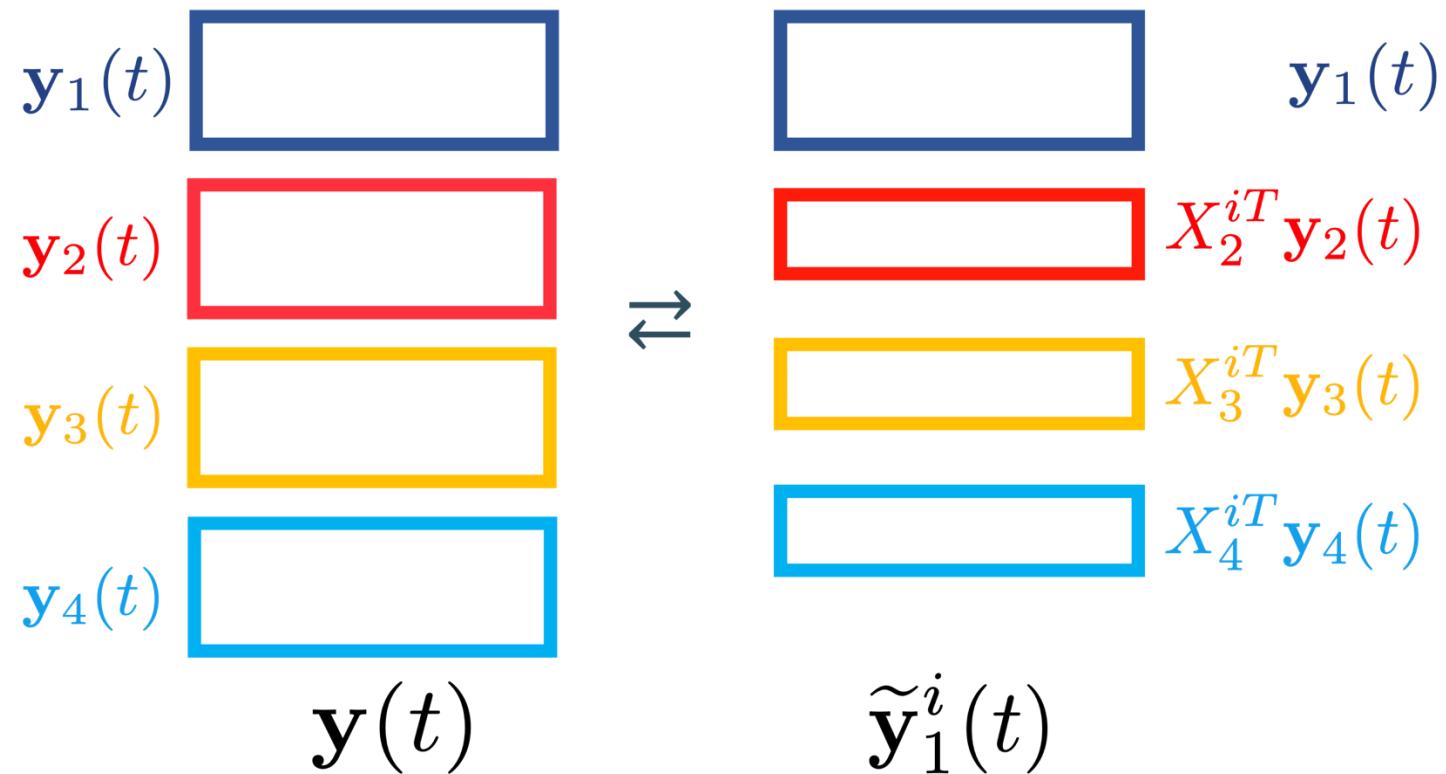
3) Every node updates $X_k^{i+1} = \begin{cases} X_q^{i+1}, & \text{if } q = k, \\ X_k^i G_k^{i+1}, & \text{if } q \neq k. \end{cases}$

$i \leftarrow i + 1$

III.B- Technical properties

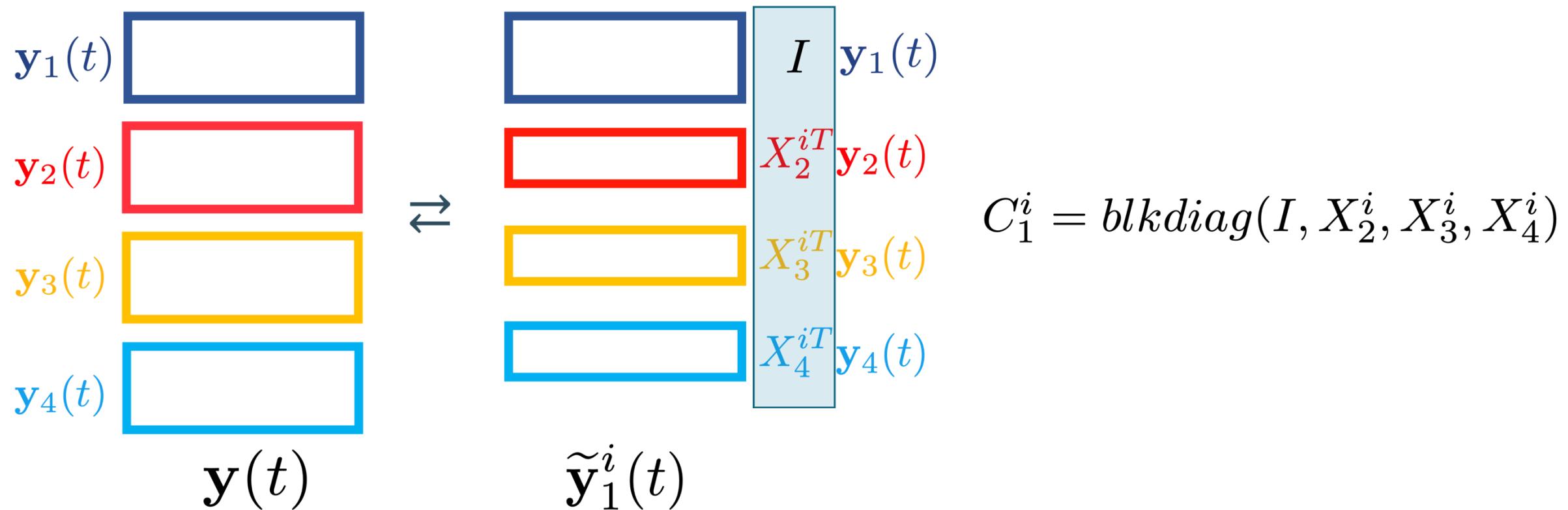
Linear relationship

$$\forall i, \forall q, \exists C_q^i \in R^{M \times \widetilde{M}_q} : C_q^{iT} \mathbf{y}(t) = \tilde{\mathbf{y}}_q^i(t)$$



Linear relationship

$$\forall i, \forall q, \exists C_q^i \in R^{M \times \widetilde{M}_q} : C_q^{iT} \mathbf{y}(t) = \tilde{\mathbf{y}}_q^i(t)$$



Linear relationship

Parameterization of variable X via C_q^i : $X = C_q^i \tilde{X}_q$

Local problem at node q :

$$\underset{\tilde{X}_q}{\text{minimize}} \quad f(\tilde{X}_q^T \tilde{\mathbf{y}}_q(t), \tilde{X}_q^T \tilde{B}_q) \quad \Leftrightarrow$$

$$\text{subject to } \tilde{X}_q \in \tilde{\mathcal{S}}_q^i$$

$$\underset{\tilde{X}_q}{\text{minimize}} \quad f(X^T \mathbf{y}(t), X^T B)$$

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

$$X = C_q^i \tilde{X}_q$$

$\tilde{\mathcal{S}}_q^i$: Constraint set of the local problem

\mathcal{S} : Constraint set of the global problem

Feasible points

- It can be shown that:

$$\tilde{X}_q \in \tilde{\mathcal{S}}_q^i \iff X = C_q^i \tilde{X}_q \in \mathcal{S}$$

- Since $\tilde{X}_q^{i+1} \in \tilde{\mathcal{S}}_q^i$ (by design):

$$X^i \in \mathcal{S}, \forall i > 0$$

i.e., all points generated by DASF are feasible points of the centralized problem

Monotonous decrease in objective

Introducing short notation: $F(X) = f(X^T \mathbf{y}(t), X^T B)$

$(F(X^i))_i$ is a monotonously decreasing converging sequence

Assumptions

- Objective and constraint functions are **smooth**
(we will briefly cover the non-smooth setting later)
- The problems are **well-posed**
i.e., a small change in the parameters of the problems results in a small change in the optimal solution (~ no rank deficient covariance matrices etc.)
- Solutions of the centralized problem are **KKT points**,
i.e. so-called linearly independent constraint qualifications (**LICQ**) are satisfied
- $\mathcal{S} \cap \{X : F(X) \leq F(X^0)\}$ is **compact**

Fixed points of DASF

- Under some technical conditions (akin to LICQ in optimization literature):

Any fixed point \bar{X} of DASF is a stationary point of the centralized problem

fixed point = invariant under DASF update, i.e., $X^{i+1} = X^i = \bar{X}$ for any updating node q

- These conditions can be shown to be satisfied ('with high probability') if the number of constraints J is upper bounded as

$$J \leq KQ^2$$

K : Number of nodes in the network
 Q : Number of columns of X

Examples

Stiefel manifold constraints: $X^T X = I_Q$

$$X \text{ is } M \times Q \quad \rightarrow J = \frac{Q(Q + 1)}{2}$$

$J \leq KQ^2$ always satisfied

Linear constraints: $X^T B = H$

$$B \text{ is } M \times L \quad \rightarrow J = QL$$

$$J \leq KQ^2 \Leftrightarrow L \leq KQ$$

Example: $Q = 2, K = 5 \Rightarrow L \leq 10$

Convergence of FC-DASF

Under previous assumptions and conditions:

$(x^i)_i$ converges to a stationary point \bar{X} if the number of ‘achievable’ solutions is finite

- # ‘achievable’ solutions here depends on problem and solver:
 - **MMSE** is strongly convex, so only 1 achievable solution by definition
 - **PCA**:
$$\min_X \text{tr}(X^T R_{yy} X) \quad \rightarrow \text{Infinitely many solutions (if } \bar{X} \text{ is a solution, then also } \bar{X}U \text{ with } U \text{ orth.)}$$
$$\text{s. t. } X^T X = I \quad \rightarrow \text{Eigenvalue solver extracts the } \bar{X} \text{ containing the eigenvectors (unique up to sign ambiguities), so # ‘achievable’ solutions by the solver is finite}$$
- In case of infinite solution set → add a deterministic selection mechanism that leads to a finite set of achievable solutions (e.g. minimum norm, greedy selection, etc.)

Convergence of FC-DASF

Minima are the only **stable** fixed points of DASF

→ Follows from monotonic decrease of the objective

If all minima are global minima:

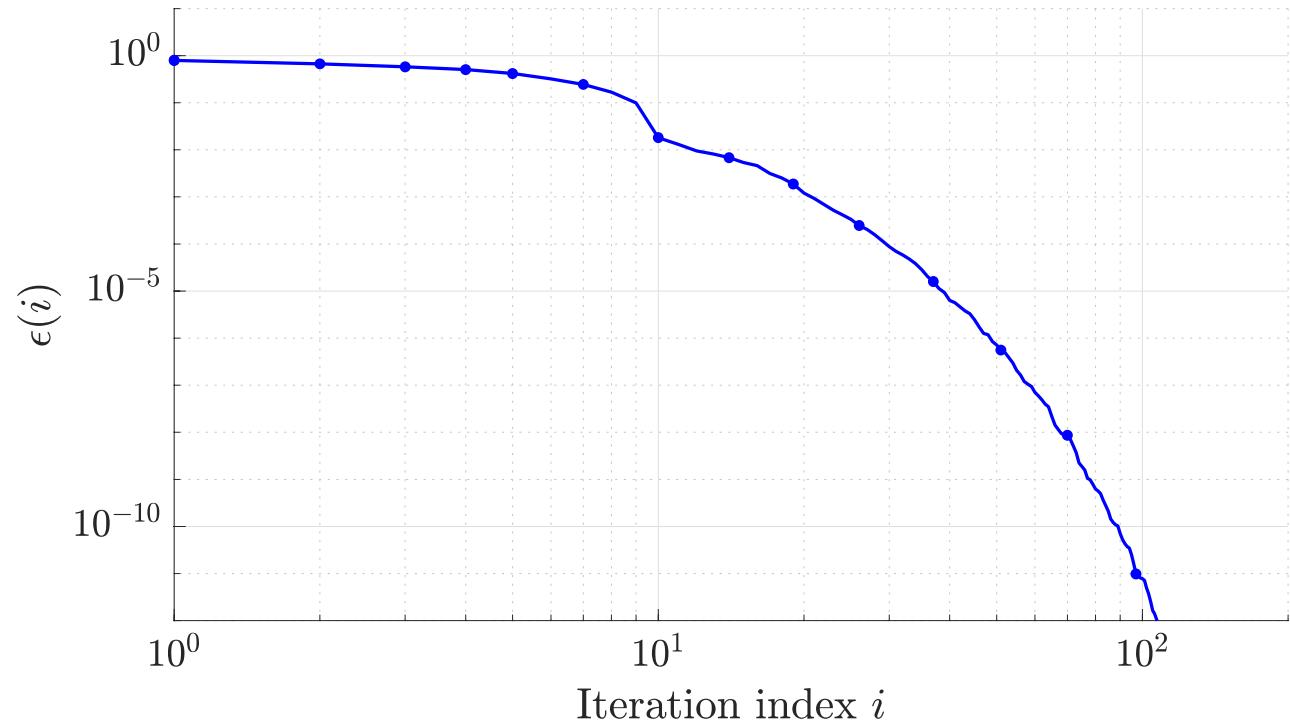
- **convergence to global solution** in practice (due to instability of all other stationary points)
- holds for many spatial filtering problems (e.g. PCA, CCA, generalized Rayleigh, convex problems, etc.)

Simulations

Trace ratio optimization (TRO)
(stationary setting)

$$\begin{aligned} & \underset{X}{\text{maximize}} \frac{\mathbb{E}[||X^T \mathbf{y}(t)||^2]}{\mathbb{E}[||X^T \mathbf{v}(t)||^2]} \\ & \text{subject to } X^T X = I_Q \end{aligned}$$

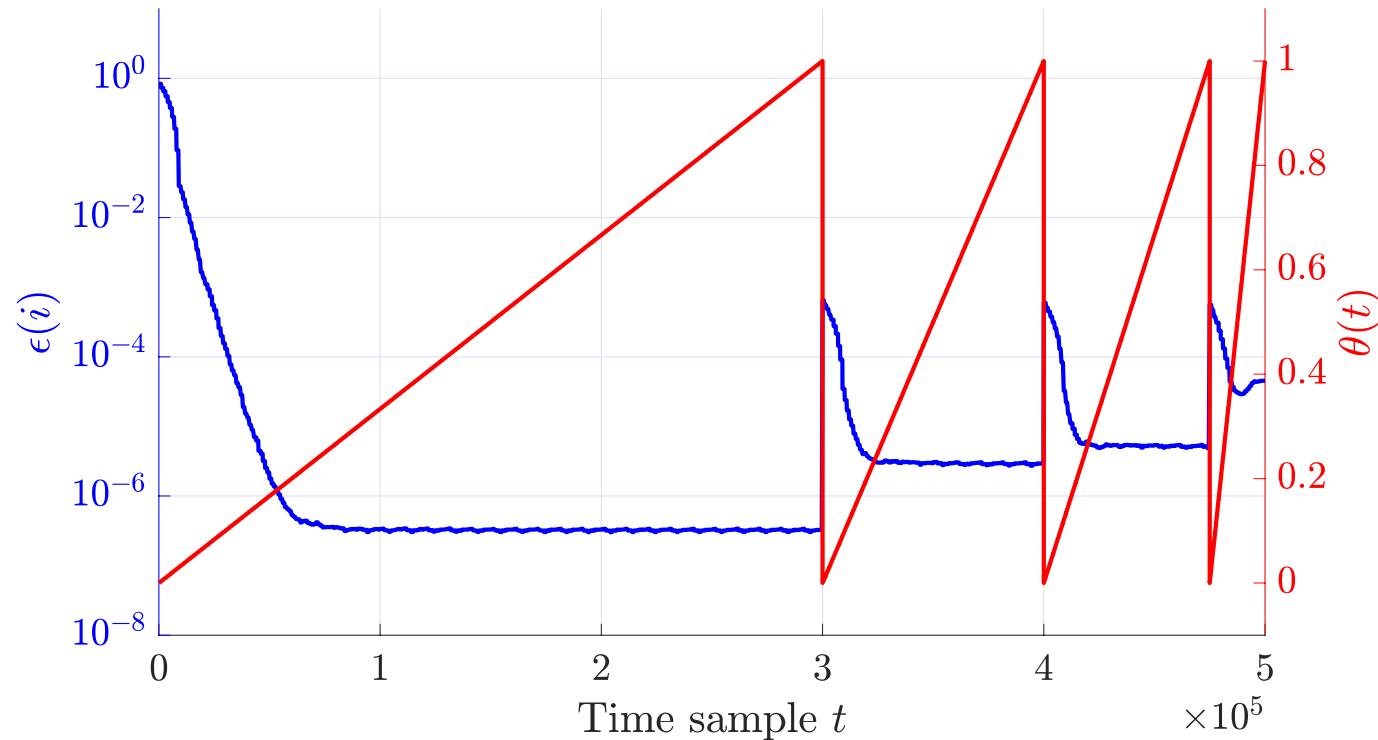
$$\epsilon(i) = \text{median} \left(\frac{||X^i - X^*||_F^2}{||X^*||_F^2} \right)$$



Simulations

TRO (adaptive setting)

$\theta(t)$ is a time-varying parameter in the generation of the sensor data

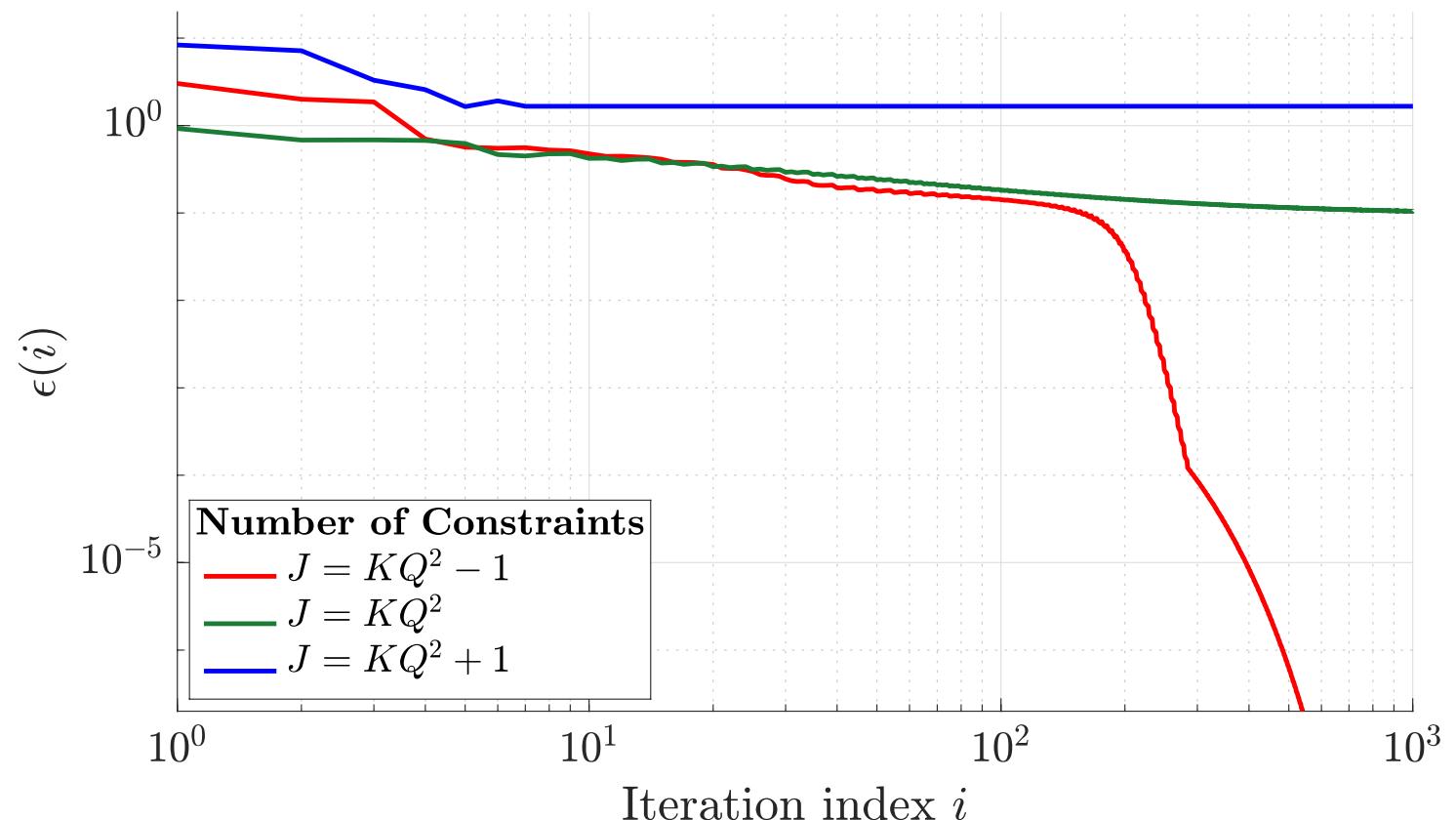


Convergence vs number of constraints

LCMV problem (stationary setting)

$$\underset{X}{\text{minimize}} \mathbb{E}[||X^T \mathbf{y}(t)||^2]$$

$$\text{subject to } X^T B = H$$



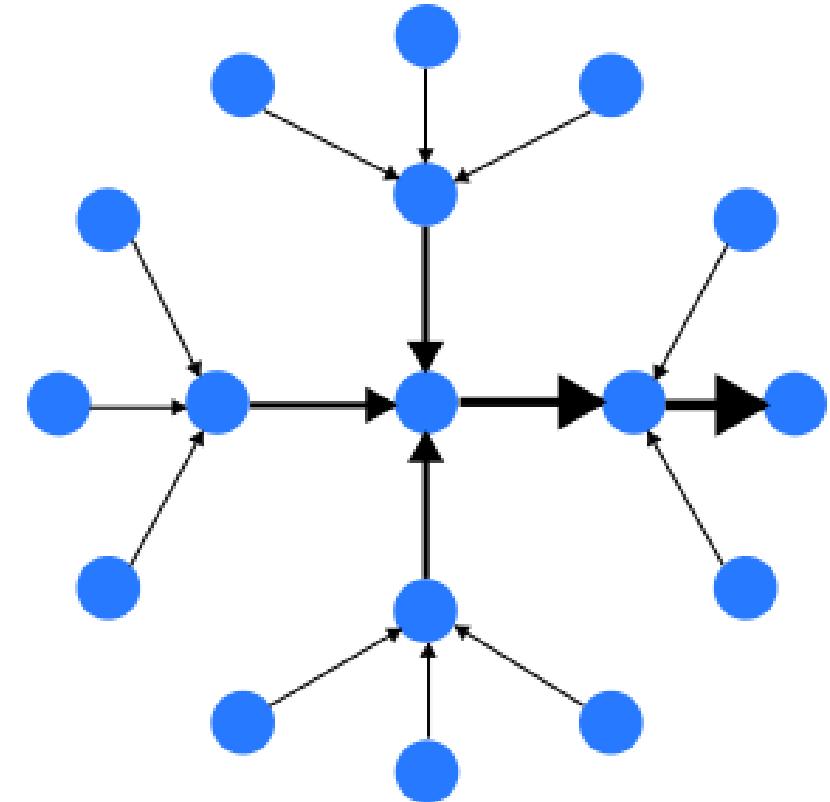
IV- Topology-independent DASF (TI-DASF)

Extending DASF to Arbitrary Topologies

Naïve Approach: Data Relaying

What if we emulate an FC topology by relaying the data?

- Requires an additional networking layer
 - Worst-case bandwidth (Line topology): $\mathcal{O}(K)$
 - Scales poorly with the number of nodes
 - Should ideally be independent of the network's size

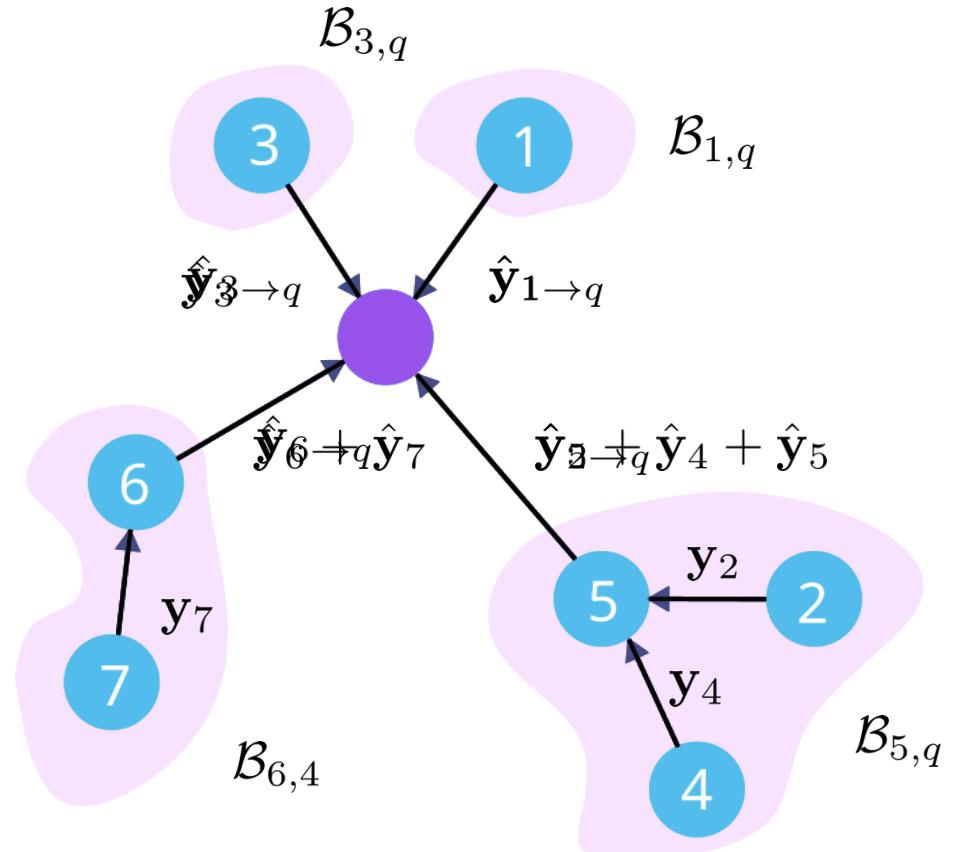


Tree-Topologies: Fuse and Forward

1. **Group** the nodes according to the branch $\mathcal{B}_{n,q}$ they belong to
2. **Recursively sum** (a new batch of) the data within each branch

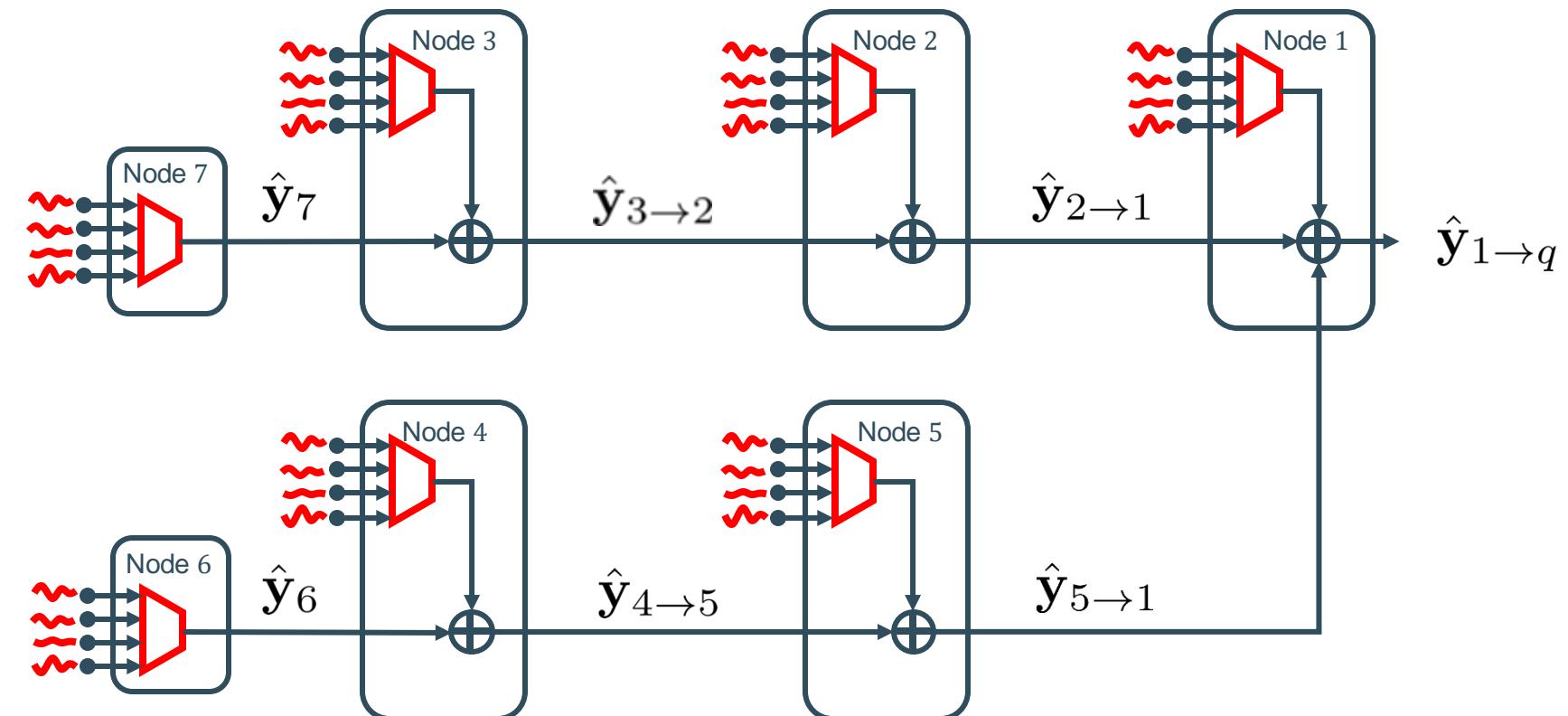
$$\hat{\mathbf{y}}_{n \rightarrow q} = \sum_{k \in \mathcal{B}_{n,q}} \hat{\mathbf{y}}_k$$

3. **Collect the sums** at the updating node



Tree topology: fuse and forward

Fuse and forward towards updating node q



It is as if the branch consisted
of a single node with filter

$$\begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \\ X_7 \end{bmatrix}$$

and data

$$\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \mathbf{y}_3 \\ \mathbf{y}_4 \\ \mathbf{y}_5 \\ \mathbf{y}_6 \\ \mathbf{y}_7 \end{bmatrix}$$

With compressed data

$$\hat{\mathbf{y}}_{1 \rightarrow q}$$

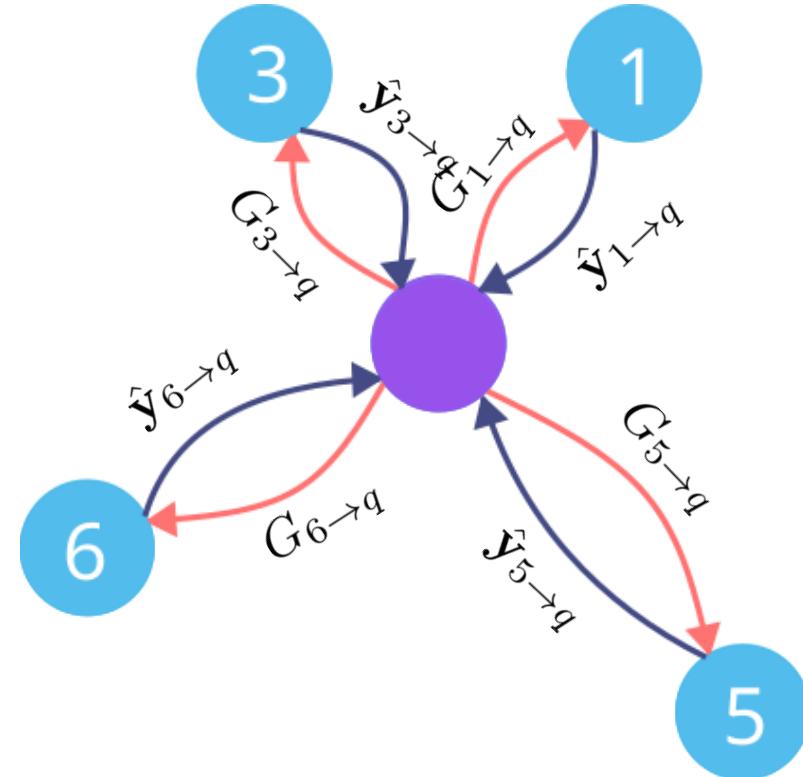
Tree Topology: Updating Node POV

- The updating node has 4 neighbors, sending it 4 signals.
- As if this was a FC network, it constructs

$$\tilde{\mathbf{y}}_q = \begin{bmatrix} \mathbf{y}_q \\ \hat{\mathbf{y}}_{1 \rightarrow q} \\ \hat{\mathbf{y}}_{3 \rightarrow q} \\ \hat{\mathbf{y}}_{5 \rightarrow q} \\ \hat{\mathbf{y}}_{6 \rightarrow q} \end{bmatrix}$$

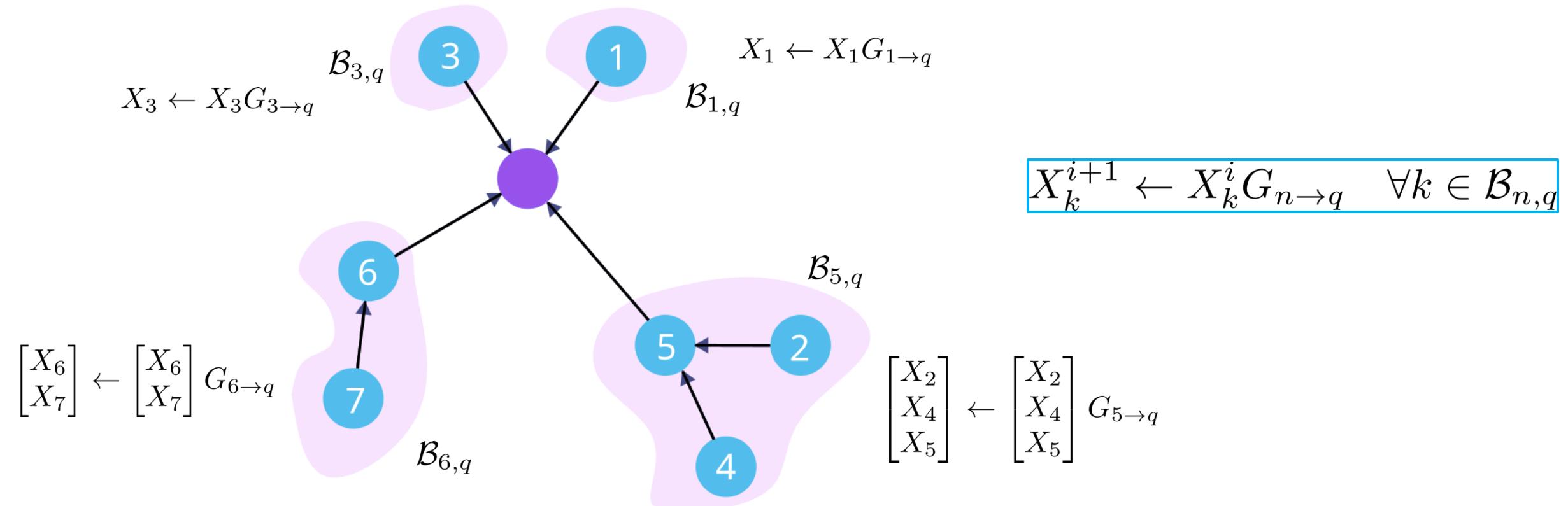
- And solve the usual local problem, obtaining

$$\tilde{\mathbf{X}}_q = \begin{bmatrix} X_q \\ G_{1 \rightarrow q} \\ G_{3 \rightarrow q} \\ G_{5 \rightarrow q} \\ G_{6 \rightarrow q} \end{bmatrix}$$



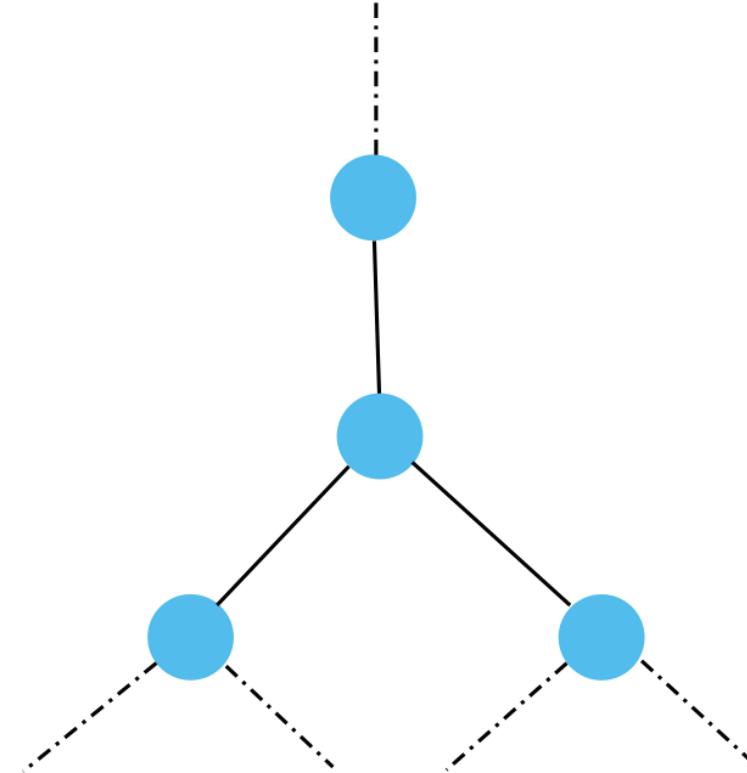
Tree Topology: Updating Node POV

- The update is performed in each branch, as if it consisted of a single node i.e.



Tree-Topology: Non-Updating Node POV

- Any node k fuses and sends data to remaining neighbors once it has received data from $|\mathcal{N}_k| - 1$ neighbors
→ data flow arises naturally
- Updates filter as soon as an update matrix is received
- Forwards the update to other neighbors



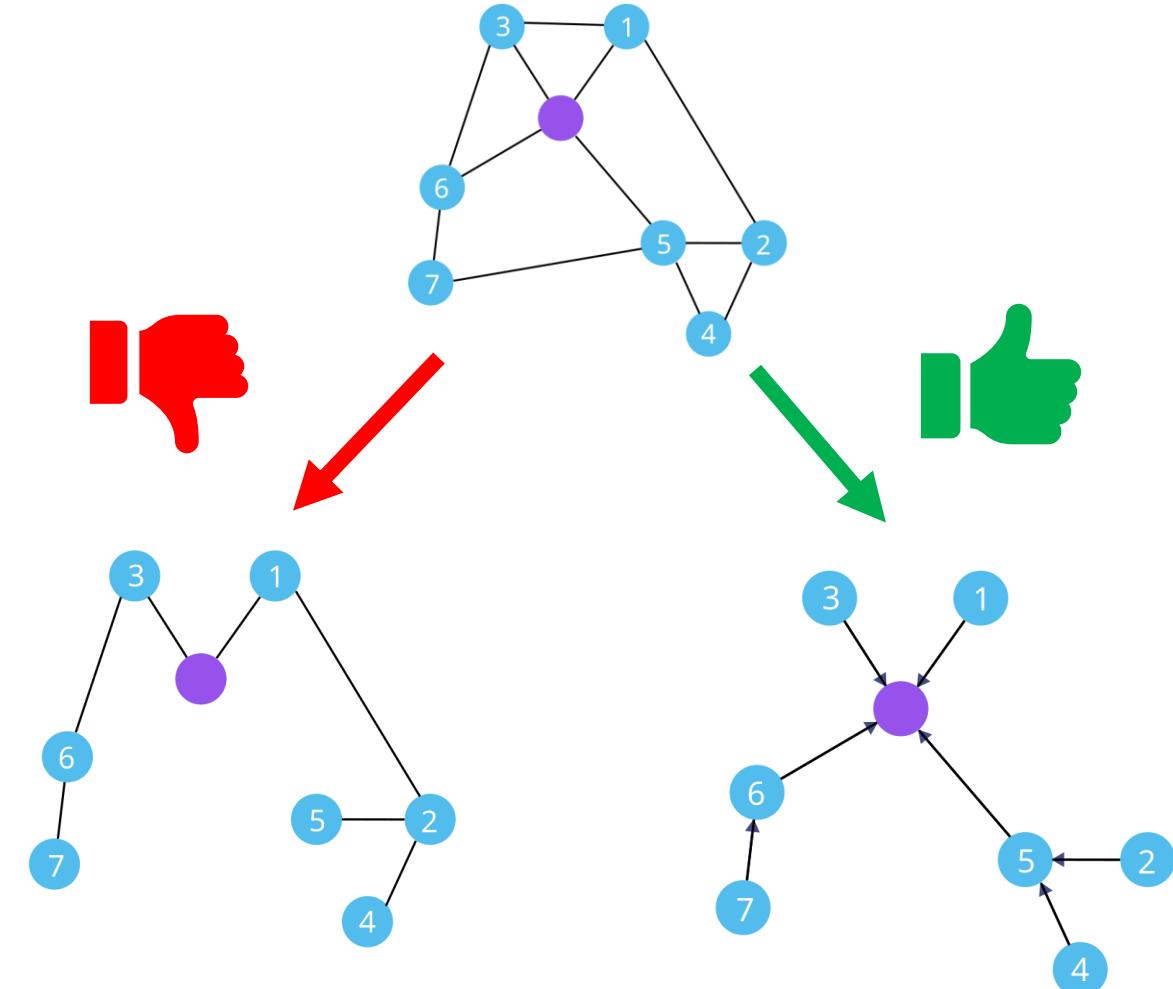
Tree topology: fuse and forward

A node is unaware of the existence of other nodes in the network (except its neighbors)

- Each node transmits Q-channel signals (*once per block of N samples*)
- Communication load at each node only depends on # neighbors
 - **Independent of network size**
 - **Fully scalable**

What about arbitrary topologies?

- Prune to spanning tree at each iteration
- **Keep as many neighbors of the updating node as possible**
→ maximize the degrees of freedom in \tilde{X}_q
- Pruning can be reused across cycles



Building a Spanning Tree

1. The updating node sends an **ADOPT** to its neighbors
2. Upon receiving an **ADOPT** the first time, the parent link is registered
3. Once they have a **parent**, the nodes send an **ADOPT** to their children

- Tree **expands** from the updating node
- Each node has a **single** parent (first **ADOPT** received)
- **Asynchronous**
- **Decentralized**
- Per-node message complexity $\mathcal{O}(|\mathcal{N}_k|)$
- Time complexity $\mathcal{O}(D)$

```
procedure buildtree ( $q$ )
  for  $l \in \mathcal{K}$  do
    Initialize set of children of node  $l$ :  $\mathcal{L}_l \leftarrow \{\}$ 
    Initialize number of acknowledgment messages
    received:  $r_l \leftarrow 0$ 
  At node  $l \in \mathcal{N}_q$ 
    Set parent to node  $q$ :  $p_l \leftarrow q$ 
  At node  $l \neq q$ 
    while  $p_l$  is not set do
      Wait for message (ADOPT,  $m$ )
      Set parent to node  $m$ :  $p_l \leftarrow m$ 
      Send (OK,  $l$ ) to  $p_l$  as an acknowledgment
    Send (ADOPT,  $l$ ) to  $\mathcal{N}_l \setminus \{p_l\}$ 
    while  $r_l < |\mathcal{N}_l|$  do
      Wait for message ( $a, m$ )
      if  $a = \text{ADOPT}$  then
        Send (NOK,  $\cdot$ ) to node  $m$  to indicate that it
        already has a parent
      else
        if  $a = \text{OK}$  then
          Add  $m$  to set of children of  $l$ :
           $\mathcal{L}_l \leftarrow \mathcal{L}_l + \{m\}$ 
       $r_l \leftarrow r_l + 1$ 
```

Per-Node Bandwidth and Complexity

	Centralized	Fully-Connected	Arbitrary
TX Bandwidth	$\mathcal{O}(K)$	$\mathcal{O}(Q)$	$\mathcal{O}(Q)$
RX Bandwidth	$\mathcal{O}(K)$	$\mathcal{O}(KQ)$	$\mathcal{O}(\mathcal{N}_k Q)$
Problem dimension	$\sum_k^K M_k$	$(K - 1)Q^2 + M_k Q$	$ \mathcal{N}_k Q^2 + M_k Q$

Independent of
network size K

Convergence

- Same convergence properties under LICQ-like conditions
- These LICQ-like conditions result in a new bound on the maximum # constraints

$$J \leq \min \left(\frac{Q^2}{K - 1} \sum_{k \in \mathcal{K}} |\mathcal{N}_k|, (1 + \min_{k \in \mathcal{K}} |\mathcal{N}_k|) Q^2 \right)$$

K : Number of nodes in the network

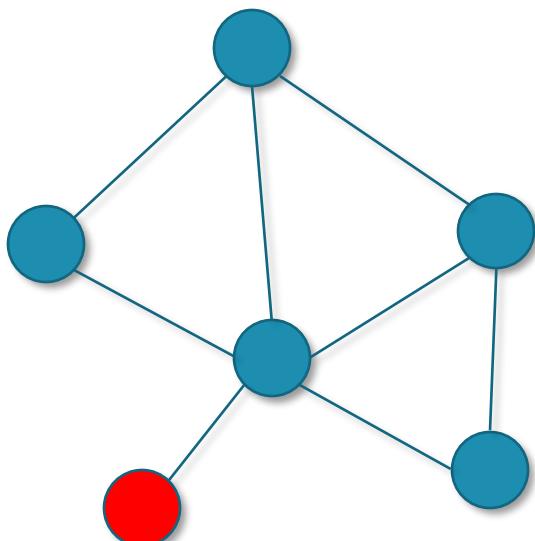
Q : Number of columns of X

\mathcal{N}_k : Neighbors of node k (excl. node k itself)

Convergence

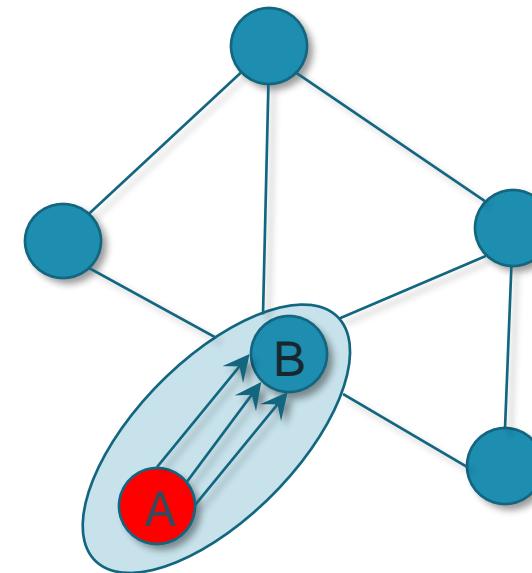
One single node with few neighbors can have a large impact

$$J \leq \min \left(\frac{Q^2}{K-1} \sum_{k \in \mathcal{K}} |\mathcal{N}_k|, (1 + \min_{k \in \mathcal{K}} |\mathcal{N}_k|) Q^2 \right)$$



If more constraints needed
→ merge nodes

i.e., A sends raw data to B, who treats this as own sensor data.



Examples

Stiefel manifold constraints: $X^T X = I_Q$

$$X \text{ is } M \times Q \quad \rightarrow J = \frac{Q(Q + 1)}{2}$$

always satisfied

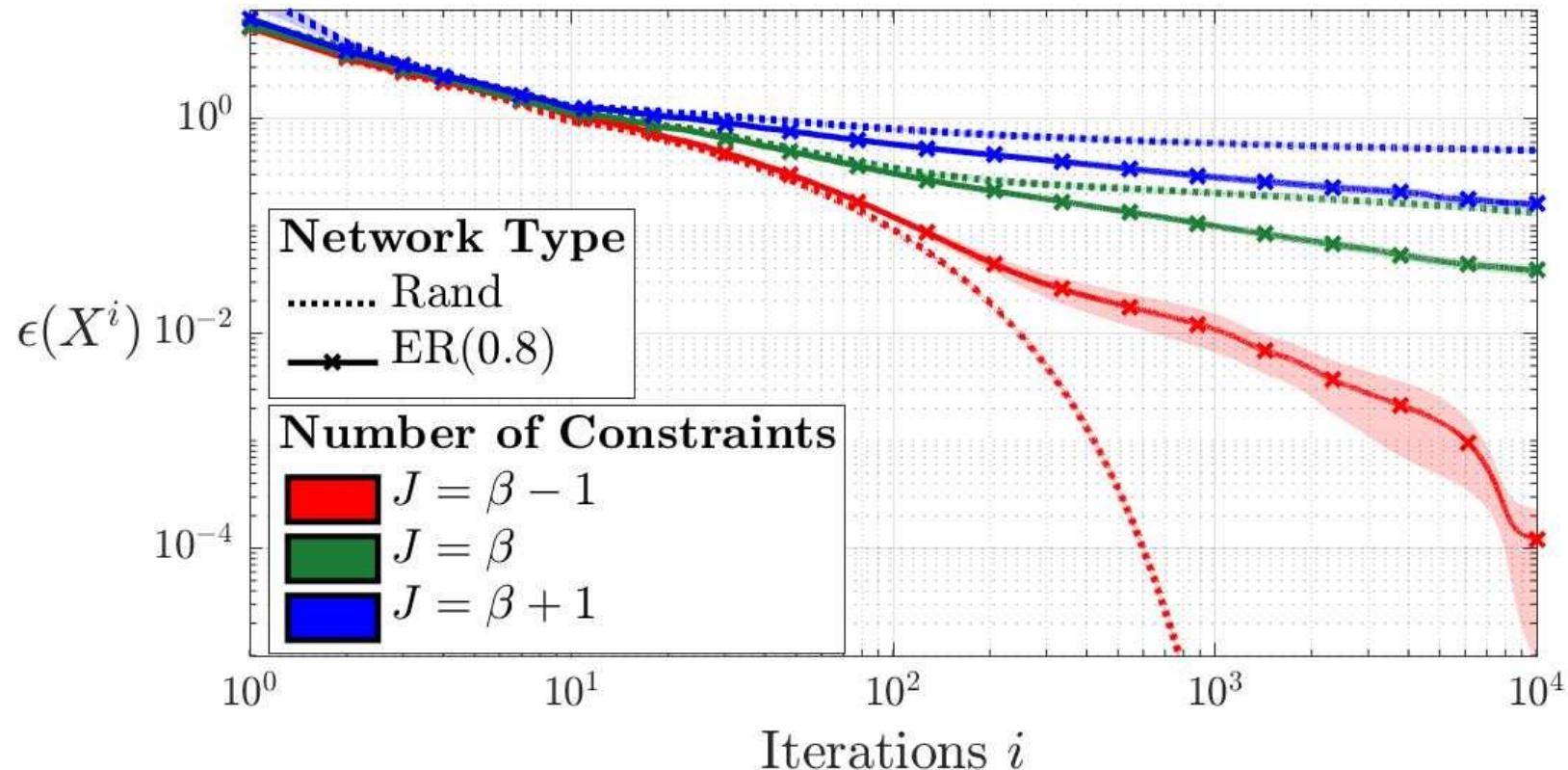
Linear constraints: $X^T B = H$

$$B \text{ is } M \times L \quad \rightarrow J = QL$$

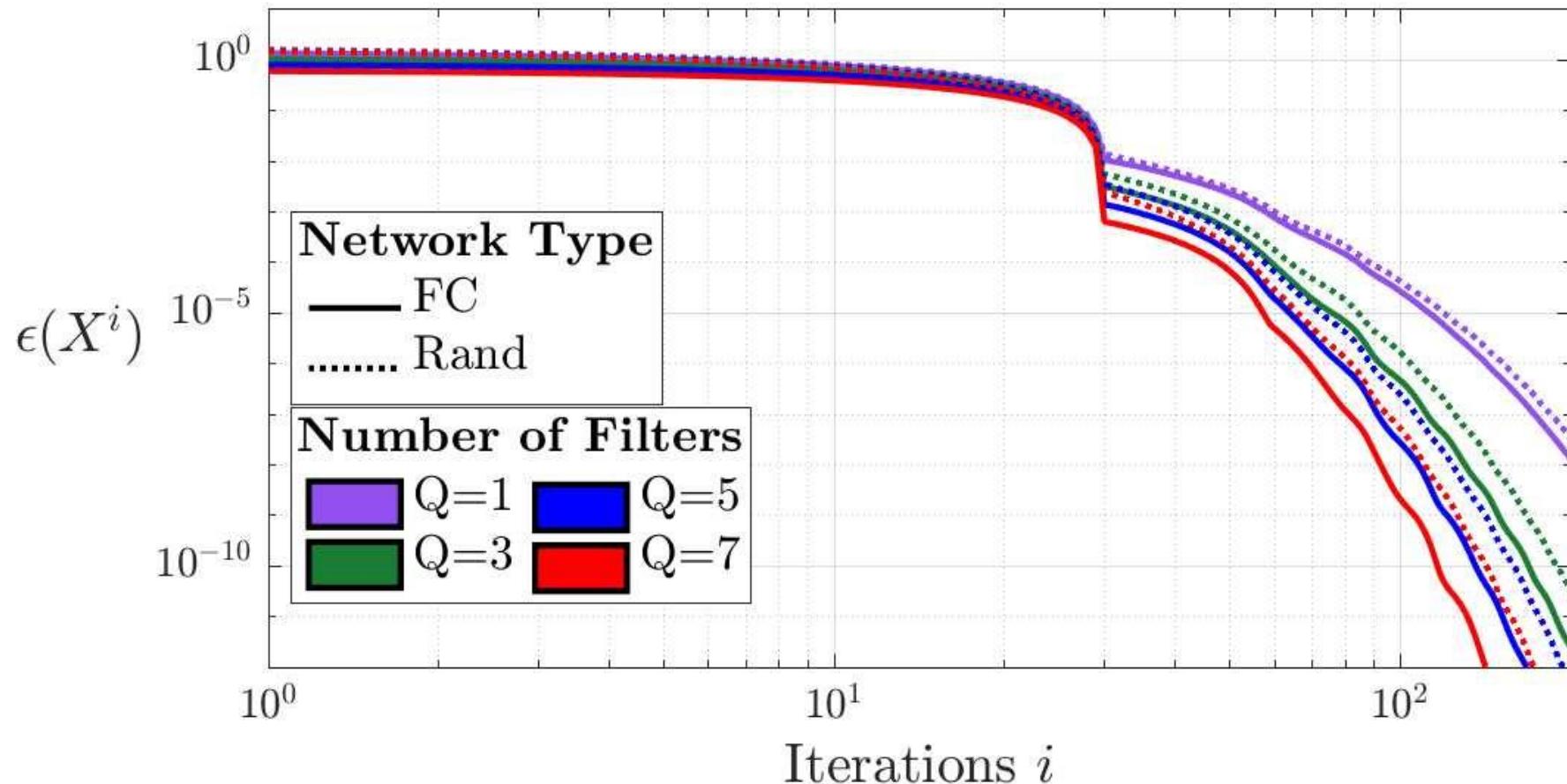
$$L \leq (1 + \min |\mathcal{N}_q|)Q$$

$$L \leq \frac{Q}{K - 1} \sum |\mathcal{N}_q|$$

Convergence Impact of Constraints Bounds



Convergence Impact of Bandwidth



V- The DASF toolbox

DASF Toolbox

Purpose:

- Experiment with the DASF algorithm on your favorite spatial filter

Entry-Point:

A function expecting the following inputs:

- Function containing your **solver** for the centralized spatial filter design (i.e., function)
- Input signals
- Network topology
- Sequence of updating nodes
- Number of channels and output filters

} (*)

(*) can also be automatically generated

Several examples available:

- PCA/GEVD
- MMSE
- TRO
- QCQP
- SCQP
- LCMV
- CCA
- **Easily add your own!**



VI- Extensions

Non-smooth problems

- Convergence and optimality has also been shown for **non-smooth problems**

$$\underset{X}{\text{minimize}} \ f(X^T \mathbf{y}(t)) + g(X^T \boldsymbol{\Gamma})$$

$$\text{subject to } l_j(X_k^T \mathbf{y}_k(t)) = 0,$$

$$h_j(X_k^T \mathbf{y}_k(t)) \leq 0$$

- **Non-smooth term** must be **node-separable**, i.e. $g(X^T \boldsymbol{\Gamma}) = \sum_k g(X_k^T \boldsymbol{\Gamma}_k)$
- Limitations: (satisfied for l_1 , l_2 , $l_{1,2}$, Frobenius norm, ...)
- **constraints** must also be **node-separable**
 - In arbitrary networks, constraints can depend **on B only**, not $\mathbf{y}(t)$

Non-smooth problems

Examples

- Block-sparse generalized CCA (SUMCORR variant):

$$\min_X \mathbb{E}[\text{tr}(X^T \mathbf{y}(t) \mathbf{y}(t)^T X)] + \sum_k \|X_k\|_{2,1}$$

$$\text{s.t. } \mathbb{E}[X_k^T \mathbf{y}_k(t) \mathbf{y}_k(t)^T X_k] = I_Q \quad \forall k \in \mathcal{K}$$

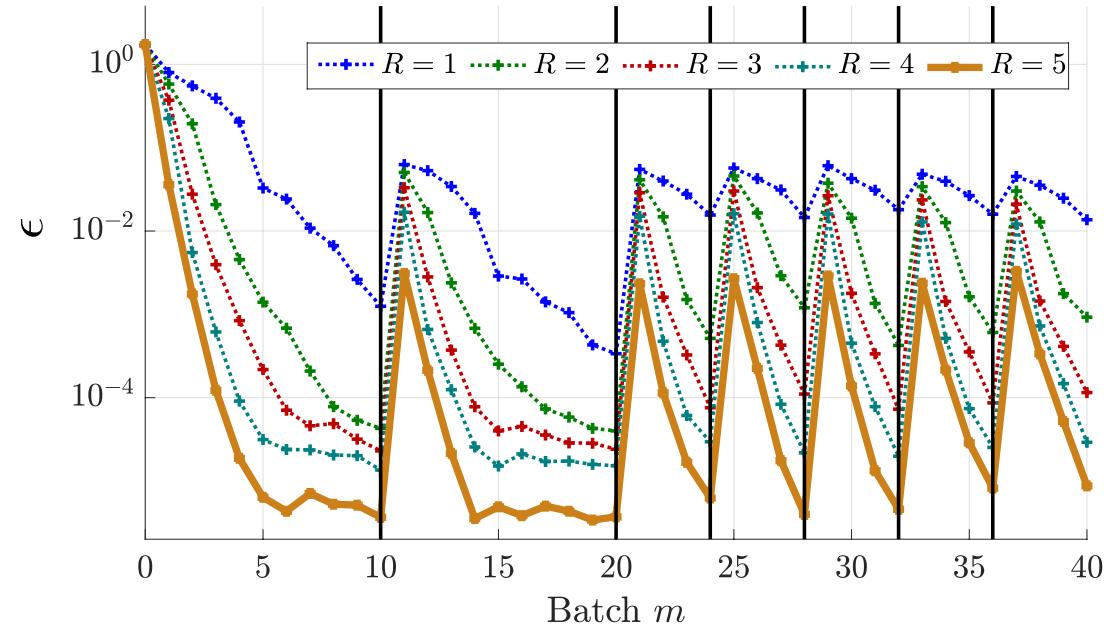
- Block-sparse regularized MMSE / MWF

$$\min_X \mathbb{E}[\|X^T \mathbf{y}(t) - \mathbf{d}(t)\|_F^2] + \sum_k \|X_k\|_F$$

$$\text{s.t. } \mathbb{E}[\|X_k^T \mathbf{y}_k(t)\|_F^2] \leq P_k \quad \forall k \in \mathcal{K}$$

Tracking improvements

- Re-use data from previous iterations
- Efficient method allows an R times faster convergence with an L -fold increase in bandwidth with $R \gg L$
(typically, $R = K$, $L = 2$)



Parallel updates

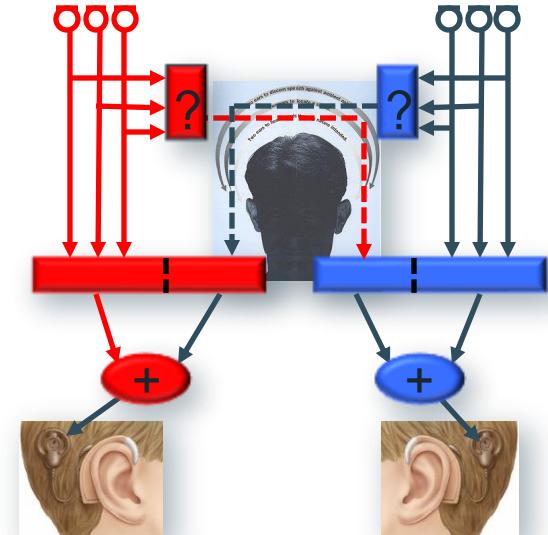
- Nodes (or subsets of nodes) can update simultaneously
- Requires relaxation on the updates: $X^{i+1} \leftarrow \alpha X^i + (1 - \alpha)X^{i+1}$
- No analytical convergence results (yet)

Node-specific problems

Problem	Objective to minimize at node k	Constraints
MMSE	$\min \mathbb{E}[\ \mathbf{d}_k(t) - X(k)^T \mathbf{y}(t)\ ^2]$	—
LCMV	$\min \mathbb{E}[\ X(k)^T \mathbf{y}(t)\ ^2]$	$X(k)^T B = H_k$

- Each node wishes to solve a **node-specific** problem

$$\begin{aligned} & \underset{X(k) \in \mathbb{R}^{M \times Q}}{\text{minimize}} && f_k(X(k)^T \mathbf{y}(t), X(k)^T B) \\ & \text{subject to} && X(k) \in \mathcal{S}_k \end{aligned}$$



- Works if the node-specific solutions are the same up to a full rank transformation(*):

$$\forall (k, l), \exists D_{k,l} : X(k)^* = X(l)^* \cdot D_{k,l}$$

(*) this happens in, e.g., speech denoising, where each node uses a different reference microphone to estimate the same speech signal [Doclo et al. 2009]

Fractional programming

Problem	Objective to minimize	Constraints
TRO	$-\frac{\mathbb{E}[X^T \mathbf{y}(t) ^2]}{\mathbb{E}[X^T \mathbf{v}(t) ^2]}$	$X^T X = I_Q$
RTLS	$\frac{\mathbb{E}[\mathbf{x}^T \mathbf{y}(t) - d(t) ^2]}{1 + \mathbf{x} ^2}$	$ \mathbf{x}^T L ^2 \leq \delta^2$

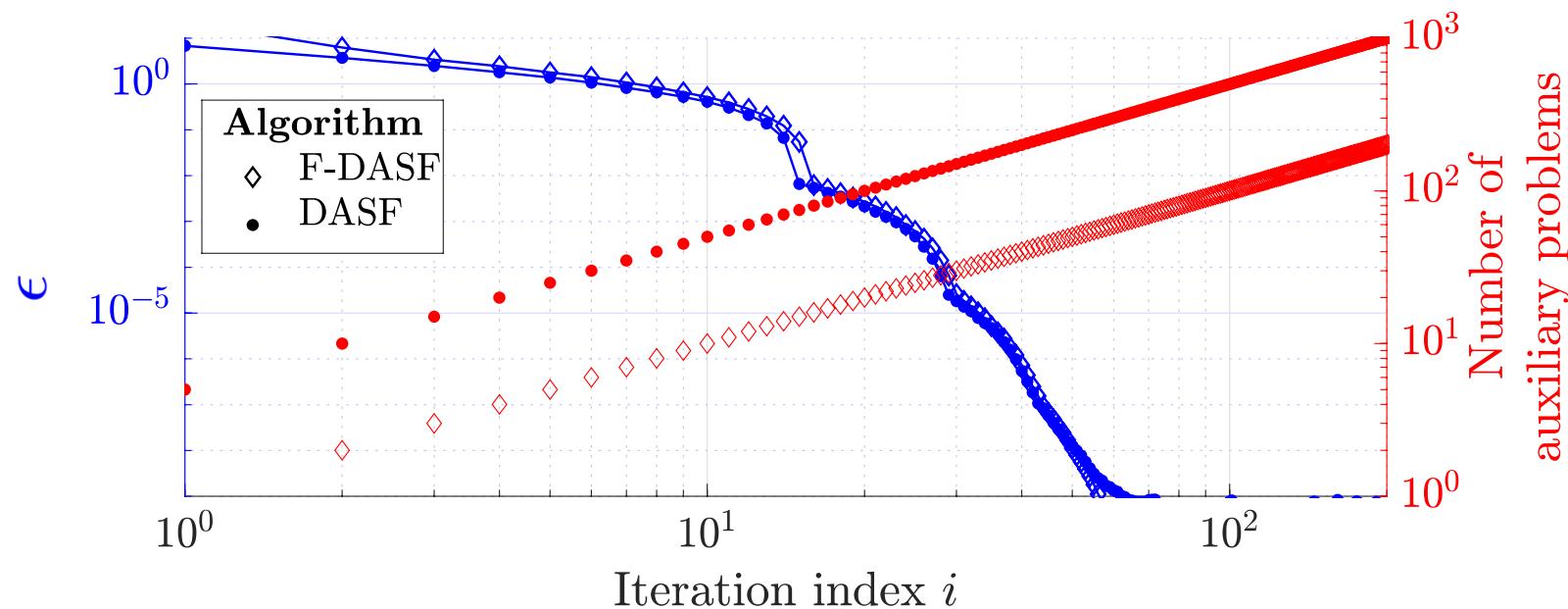
- Special version of DASF for fractional programs:

$$\min_X \frac{f_1(X)}{f_2(X)}, \text{ s.t., } X \in \mathcal{S}$$

- Fractional programs are solved with an iterative procedure (Dinkelbach)
 - large computational cost at each node
- Main idea: interleave steps of DASF with iterative solver
 - with some modifications in the proof, can still prove convergence

Fractional programming

- Convergence rate similar to DASF, with significantly less computations



VII- Conclusion

Conclusions

- Generic algorithmic framework for **distributed & adaptive** spatial filtering
- Covers large number of optimization problems including PCA, GEVD, CCA, MMSE, LCMV, TRO, MAXVAR, SUMCORR, sparsity norms, etc.
- **Convergence guarantees** under mild assumptions generally satisfied in practice
- Various extensions

Bibliography: key papers

[Musluoglu et al. 2023a] C. A. Musluoglu and A. Bertrand, "A Unified Algorithmic Framework for Distributed Adaptive Signal and Feature Fusion Problems—Part I: Algorithm Derivation," in *IEEE Transactions on Signal Processing*, vol. 71, pp. 1863-1878, 2023

[Musluoglu et al. 2023b] C. A. Musluoglu, C. Hovine and A. Bertrand, "A Unified Algorithmic Framework for Distributed Adaptive Signal and Feature Fusion Problems — Part II: Convergence Properties," in *IEEE Transactions on Signal Processing*, vol. 71, pp. 1879-1894, 2023

Bibliography (I)

- [Doclo et al. 2009]** S. Doclo, M. Moonen, T. Van den Bogaert and J. Wouters, "Reduced-Bandwidth and Distributed MWF-Based Noise Reduction Algorithms for Binaural Hearing Aids," in *IEEE Trans. Audio, Speech, and Language Processing*, vol. 17, no. 1, pp. 38-51, 2009.
- [Musluoglu et al. 2023a]** C. A. Musluoglu and A. Bertrand, "A Unified Algorithmic Framework for Distributed Adaptive Signal and Feature Fusion Problems—Part I: Algorithm Derivation," in *IEEE Trans. Signal Processing*, vol. 71, pp. 1863-1878, 2023
- [Musluoglu et al. 2023b]** C. A. Musluoglu, C. Hovine and A. Bertrand, "A Unified Algorithmic Framework for Distributed Adaptive Signal and Feature Fusion Problems — Part II: Convergence Properties," in *IEEE Trans. Signal Processing*, vol. 71, pp. 1879-1894, 2023
- [Bertrand et al. 2011]** A. Bertrand, and M. Moonen, "Distributed LCMV beamforming in wireless sensor networks with node-specific desired signals." *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2011.
- [Zeng et al. 2013]** Y. Zeng and Richard C. Hendriks. "Distributed delay and sum beamformer for speech enhancement via randomized gossip." *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 1 22.1, pp. 260-273, 2013.

Bibliography (II)

- [Li et al. 2011]** L. Li, A. Scaglione, and J. H. Manton. "Distributed principal subspace estimation in wireless sensor networks." *IEEE Journal of Selected Topics in Signal Processing* vol. 5, no.4, pp725-738, 2011.
- [Mota et al. 2011]** J.F.C Mota, et al. "Distributed basis pursuit." *IEEE Transactions on Signal Processing* vol. 60, pp. 1942-1956, 2011.
- [Markovich et al. 2015]** S. Markovich-Golan, et al. "Optimal distributed minimum-variance beamforming approaches for speech enhancement in wireless acoustic sensor networks." *Signal Processing*, vol.107, pp. 4-20, 2015.
- [Fu et al. 2016]** X. Fu, et al. "Efficient and distributed algorithms for large-scale generalized canonical correlations analysis." *2016 IEEE 16th international conference on data mining (ICDM)*. IEEE, 2016.
- [Bertrand et al. 2018]** A. Bertrand, and M. Moonen. "Distributed canonical correlation analysis in wireless sensor networks with application to distributed blind source separation." *IEEE Transactions on Signal Processing* vol. 63, no.18, pp. 4800-4813, 2018.
- [Zhang et al. 2019]** J. Zhang, et al. "Distributed rate-constrained LCMV beamforming." *IEEE Signal Processing Letters*, vol. 26, no. 5, pp. 675-679, 2019.
- [Musluoglu et al. 2021]** C.A. Musluoglu,, and A. Bertrand. "Distributed adaptive trace ratio optimization in wireless sensor networks." *IEEE Transactions on Signal Processing*, vol. 69 pp. 3653-3670, 2021.