Penalty-Based Multitask Estimation with Non-Local Linear Equality Constraints

Fei Hua*†‡, Roula Nassif‡, Cédric Richard‡, Haiyan Wang*†
†School of Marine Science and Technology, Northwestern Polytechnical University (NPU), Xi'an 710072, China
*Key Laboratory of Ocean Acoustics and Sensing (NPU), Ministry of Industry and Technology, Xi'an 710072, China
‡ Laboratoire Lagrange, Université Côte d'Azur, OCA, CNRS, Nice 06108, France
Email: {fei.hua, roula.nassif}@oca.eu cedric.richard@unice.fr hywang@nwpu.edu.cn

***Abstract*—We consider distributed estimation problems over multitask networks where the parameter vectors at distinct agents are coupled via a set of linear equality constraints. Unlike previous existing works, the current work assumes that each constraint involves agents that are not necessarily one-hop neighbors. At each time instant, we assume that each agent has access to the instantaneous estimates of its one-hop neighbors and to the past estimates of its multi-hop neighbors through a multi-hop relay protocol. A distributed penalty-based algorithm is then derived and its performance analyses in the mean and in the mean-square-error sense are provided. Simulation results show the effectiveness of the strategy and validate the theoretical models.**

I. Introduction

Distributed estimation is used in a wide range of applications including communication [1], spectrum sensing [2], distributed localization [3], and power system monitoring [4]. Several useful distributed solutions, such as incremental strategies [5], diffusion strategies [6]–[11], and consensus strategies [12], [13] have been proposed in the literature to address single-task problems where all agents in the network collaborate to estimate a common parameter vector from noisy measurements. Among them, diffusion strategies are advantageous in terms of stability range, robustness, and performance [8]–[10].

In many applications, however, it happens that the agents in the network have to infer multiple parameter vectors simultaneously. Networks of this type are referred to as multitask networks [14], [15]. Multitask diffusion strategies were derived by exploiting prior information on the relationships among tasks. For example, appropriate regularization terms can be used to promote similarities between the tasks [14], [16], [17]. In [18], a diffusion-based algorithm is proposed to solve node-specific estimation problems where each node consists of a set of local parameters and a set of network global parameters. In [19], [20], the parameter space is decomposed into two orthogonal subspaces. The relations among tasks are modeled by assuming that they all share one of the subspaces. In some applications, such as the network flow problem [21], the basis pursuit problem [22], and the interference management problem [23], the parameter vectors may be related via a set of linear equality constraints. Distributed projection-based [24] and penalty-based [25] estimation algorithms were proposed to solve multitask estimation problems where each agent is interested in estimating its own parameter vector and where the parameter vectors at neighboring agents are related according to a set of linear equality constraints. In the current work, we consider a more general multitask scenario where each equality constraint involves agents that are not necessarily one-hop neighbors.

The work of F. Hua was partly supported by China Scholarship Council and NSFC grant 61471298. The work of R. Nassif and C. Richard was supported in part by ANR and DGA grant ANR-13-ASTR-0300 (ODISSEE project). The work of H. Wang was partly supported by NSFC under grants 61571365 and 61671386.

Let N denote the number of agents in the network and P the total number of constraints. We are interested in devising a distributed adaptive solution to solve the following optimization problem:

$$\underset{\boldsymbol{w}_1,\ldots,\boldsymbol{w}_N}{\text{minimize}} \quad J^{\text{glob}}(\boldsymbol{w}_1,\ldots,\boldsymbol{w}_N) \triangleq \sum\nolimits_{k=1}^{N} J_k(\boldsymbol{w}_k), \tag{1a}$$

$$\text{subject to} \quad \sum\nolimits_{\ell\in\mathcal{I}_p} \boldsymbol{D}_{p\ell}\boldsymbol{w}_\ell + \boldsymbol{b}_p = 0, \quad p=1,\ldots,P \tag{1b}$$

Each agent k seeks to estimate its parameter vector $\boldsymbol{w}_k \in \mathbb{R}^{M_k\times 1}$, and has knowledge of its local cost $J_k(\cdot)$ and the set of linear equality constraints that it is involved in. Each constraint is indexed by p, and defined by the $L_p \times M_\ell$ matrices $\boldsymbol{D}_{p\ell}$, the $L_p \times 1$ vector $\boldsymbol{b}_p$, and the set $\mathcal{I}_p$ of agent indices involved in the p-th constraint. The previous works [24], [25] assumed that $\mathcal{I}_p \subseteq \mathcal{N}_k$ for all $k \in \mathcal{I}_p$ with $\mathcal{N}_k$ denoting the one-hop neighborhood of agent k that consists of all agents that are connected to k by an edge. In this paper, we relax this assumption by considering scenarios where the constraints involve agents that are not necessarily one-hop neighbors. In order to derive a distributed solution relying solely on local interactions between neighbors, we shall employ multi-hop relay protocols to enable non-neighboring agents to share their estimates in order to satisfy their constraints. A penalty-based distributed estimation algorithm is derived and its stochastic behavior in the mean and in the mean-square-error sense is analyzed. Simulations are conducted to illustrate the effectiveness of the proposed algorithm and to validate the theoretical models.

Notations: All vectors are column vectors. The all-one vector of length N is denoted by $\mathbb{1}_N$ and the identity matrix of size N is denoted by $\boldsymbol{I}_N$. The (k,ℓ)-th block of a block matrix is denoted by $[\cdot]_{k,\ell}$. The operator $\text{col}\{\cdot\}$ stacks its vector entries on top of each other. The symbol $\otimes$ refers to the Kronecker product. The symbol $\text{vec}(\cdot)$ denotes the vectorization operator that stacks the columns of a matrix on top of each other. The symbol $\mathcal{N}_k^{(h)}$ refers to the h-hop neighborhood of agent k, that is, $\ell \in \mathcal{N}_k^{(h)}$ means that the smallest number of hops from agent k to agent ℓ is equal to h.

II. Problem Formulation and Penalty-Based Solution

Consider a strongly connected network of N agents. At each time instant i, each agent k has access to a zero-mean scalar observation $d_k(i)$, and to a zero-mean regression vector $\boldsymbol{x}_k(i) \in \mathbb{R}^{M_k\times 1}$ with positive definite covariance matrix $\boldsymbol{R}_{x,k} = \mathbb{E}\{\boldsymbol{x}_k(i)\boldsymbol{x}_k^\top(i)\}$. The observations $\{d_k(i), \boldsymbol{x}_k(i)\}$ are assumed to satisfy a linear regression model:

$$d_k(i) = \boldsymbol{x}_k^\top(i)\boldsymbol{w}_k^o + z_k(i), \tag{2}$$

where $\boldsymbol{w}_k^o$ is an $M_k \times 1$ unknown parameter vector to be estimated by agent k, and $z_k(i)$ is a zero-mean noise with variance $\sigma_{z,k}^2$ assumed to be spatially and temporally independent. In order to estimate $\boldsymbol{w}_k^o$, we associate with agent k the mean-square-error cost which is strongly convex and second-order differentiable :

$$J_k(\boldsymbol{w}_k) = \mathbb{E}|d_k(i) - \boldsymbol{x}_k^\top(i)\boldsymbol{w}_k|^2. \tag{3}$$

Let us collect the parameter vectors $\boldsymbol{w}_k$ and $\boldsymbol{w}_k^o$ from across the network into the following vectors of length $M = \sum_{k=1}^N M_k$:

$$\boldsymbol{w} \triangleq \text{col}\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_N\}, \quad \boldsymbol{w}^o \triangleq \text{col}\{\boldsymbol{w}_1^o, \ldots, \boldsymbol{w}_N^o\}.$$

Problem (1) can be written equivalently as:

$$\begin{aligned} \underset{\boldsymbol{w}}{\text{minimize}} \quad & \sum\nolimits_{k=1}^N \mathbb{E}|d_k(i) - \boldsymbol{x}_k^\top(i)\boldsymbol{w}_k|^2, && \text{(4a)} \\ \text{subject to} \quad & \boldsymbol{\mathcal{D}}\boldsymbol{w} + \boldsymbol{b} = \boldsymbol{0}. && \text{(4b)} \end{aligned}$$

where $\boldsymbol{\mathcal{D}}$ is a $P \times N$ block matrix with block entries $\boldsymbol{D}_{p\ell}$ and $\boldsymbol{b}$ is a $P \times 1$ block column vector with block entries $\boldsymbol{b}_p$. We shall assume that $P < N$ and that $\boldsymbol{\mathcal{D}}$ is full row rank, so that (4b) has at least one solution.

The positive-definite quadratic problem (4) has a unique global minimum given by:

$$\boldsymbol{w}^\star = \boldsymbol{w}^o - \boldsymbol{\mathcal{R}}_x^{-1}\boldsymbol{\mathcal{D}}^\top(\boldsymbol{\mathcal{D}}\boldsymbol{\mathcal{R}}_x^{-1}\boldsymbol{\mathcal{D}}^\top)^{-1}(\boldsymbol{\mathcal{D}}\boldsymbol{w}^o + \boldsymbol{b}), \tag{5}$$

where $\boldsymbol{\mathcal{R}}_x \triangleq \text{diag}\{\boldsymbol{R}_{x,1}, \ldots, \boldsymbol{R}_{x,N}\}$.

Instead of using (5), we are interested in an adaptive distributed solution that is able to learn from streaming data and that relies on local interactions between neighboring agents. Penalty methods offer a simple way for tackling constrained optimization problems. These methods consist of approximating the constrained problem (4) into an unconstrained one by adding to the objective function a penalty term that penalizes any violation of the constraints:

$$\underset{\boldsymbol{w}}{\text{minimize}} \quad J_\eta^{\text{glob}}(\boldsymbol{w}) \triangleq \sum\nolimits_{k=1}^N J_k(\boldsymbol{w}_k) + \eta\|\boldsymbol{\mathcal{D}}\boldsymbol{w} + \boldsymbol{b}\|^2, \tag{6}$$

with $\eta > 0$ a scalar parameter that controls the relative importance of adhering to the constraints. Increasing η improves the approximation (6) in quality [26]–[29], i.e., $\boldsymbol{w}^o(\eta)$ gets closer to $\boldsymbol{w}^\star$. The above problem is strongly convex for any η and its closed form solution parameterized by η is given by:

$$\boldsymbol{w}^o(\eta) = (\boldsymbol{\mathcal{R}}_x + \eta\boldsymbol{\mathcal{D}}^\top\boldsymbol{\mathcal{D}})^{-1}(\boldsymbol{\mathcal{R}}_x\boldsymbol{w}^o - \eta\boldsymbol{\mathcal{D}}^\top\boldsymbol{b}). \tag{7}$$

Applying a steepest-descent iteration to minimize the cost in (6) with respect to $\boldsymbol{w}_k$ and starting from an initial condition $\boldsymbol{w}_k(0)$, we obtain the following algorithm at node k:

$$\begin{aligned} \boldsymbol{w}_k(i+1) = \boldsymbol{w}_k(i) - \mu\Big[&\boldsymbol{R}_{x,k}\boldsymbol{w}_k(i) - \boldsymbol{r}_{dx,k} + \\ &\eta\sum_{p\in\mathcal{J}_k}\boldsymbol{D}_{pk}^\top\Big(\sum_{\ell\in\mathcal{I}_p}\boldsymbol{D}_{p\ell}\boldsymbol{w}_\ell(i) + \boldsymbol{b}_p\Big)\Big], \end{aligned} \tag{8}$$

where $\boldsymbol{r}_{dx,k} \triangleq \mathbb{E}\{d_k(i)\boldsymbol{x}_k(i)\}$, $\mathcal{J}_k$ denotes the set of constraint indices involving agent k, i.e., $\mathcal{J}_k \triangleq \{p|k \in \mathcal{I}_p\}$. In order to evaluate $\sum_{\ell\in\mathcal{I}_p}\boldsymbol{D}_{p\ell}\boldsymbol{w}_\ell(i) + \boldsymbol{b}_p$ in (8), agent k needs the estimates $\boldsymbol{w}_\ell(i)$ from all agents $\ell \in \mathcal{I}_p$. These agents are not necessarily in the one-hop neighborhood of k, and need to employ multi-hop relay protocols to share their own estimate. We shall assume that the route from agent $\ell \in \mathcal{I}_p$ to agent k with the smallest number of relays or hops, which is often the most energy-efficient route [30], is known. Instead of using $\boldsymbol{w}_\ell(i)$ since it may not be available, agent k will use past estimate $\boldsymbol{w}_\ell(i-j)$ of agent ℓ where the delay j depends on the smallest number of hops from agent ℓ to agent k, denoted by $h_{\ell k}$. In this work, we shall assume that $j = h_{\ell k} - 1$. With this multi-hop relay protocol, at each time instant i, agent k has access to $\boldsymbol{w}_\ell(i+1-h_{\ell k})$.

Usually, the second-order moments $\boldsymbol{R}_{x,k}$ and $\boldsymbol{r}_{dx,k}$ in (8) are not available beforehand. We replace them by the instantaneous approximations [31]:

$$\boldsymbol{R}_{x,k} \approx \boldsymbol{x}_k(i)\boldsymbol{x}_k^\top(i), \quad \boldsymbol{r}_{dx,k} \approx d_k(i)\boldsymbol{x}_k(i), \tag{9}$$

Replacing $\boldsymbol{w}_\ell(i)$ in (8) by $\boldsymbol{w}_\ell(i+1-h_{\ell k})$, and splitting the update iteration into two incremental steps by introducing the intermediate estimate $\boldsymbol{\phi}_k(i+1)$, we obtain the following adaptive algorithm at agent k:

$$\boldsymbol{\phi}_k(i+1) = \boldsymbol{w}_k(i) + \mu\boldsymbol{x}_k(i)\Big(d_k(i) - \boldsymbol{x}_k^\top(i)\boldsymbol{w}_k(i)\Big), \tag{10a}$$

$$\begin{aligned} \boldsymbol{w}_k(i+1) = &\boldsymbol{\phi}_k(i+1) \\ &- \mu\eta\sum_{p\in\mathcal{J}_k}\boldsymbol{D}_{pk}^\top\Big(\sum_{\ell\in\mathcal{I}_p}\boldsymbol{D}_{p\ell}\boldsymbol{\phi}_\ell(i+2-h_{\ell k}) + \boldsymbol{b}_p\Big), \end{aligned} \tag{10b}$$

where in the second step (10b), we replaced $\boldsymbol{w}_\ell(i+1-h_{\ell k})$ by the intermediate estimate $\boldsymbol{\phi}_\ell(i+2-h_{\ell k}))$ which is a better estimate for the solution at agent ℓ. We set $\boldsymbol{\phi}_\ell(i+2-h_{\ell k}) = \boldsymbol{0}$ if $i+2-h_{\ell k} < 0$, and $h_{kk} = 1$. In the first step (10a), which is the adaptation step, node k uses its own data to update its estimate $\boldsymbol{w}_k(i)$ to an intermediate estimate $\boldsymbol{\phi}_k(i+1)$. In the second step (10b), which corresponds to the penalization step, node k collects the intermediate estimates $\boldsymbol{\phi}_\ell(i+2-h_{\ell k})$ from nodes $\ell \in \mathcal{I}_p$ for all $p \in \mathcal{J}_k$.

III. Stochastic Behavior Analysis

In this section, we study the mean and the mean-square-error behavior of algorithm (10) with respect to $\boldsymbol{w}^o(\eta)$ (7) and $\boldsymbol{w}^\star$ (5). Let $\widetilde{\boldsymbol{w}}_k(i) \triangleq \boldsymbol{w}_k^o(\eta) - \boldsymbol{w}_k(i)$ and $\widetilde{\boldsymbol{\phi}}_k(i) \triangleq \boldsymbol{w}_k^o(\eta) - \boldsymbol{\phi}_k(i)$ denote the error vectors at agent k. Taking into account the delayed information emerging from the multi-hop protocol, we introduce the extended network error vectors:

$$\begin{aligned} \widetilde{\boldsymbol{w}}_e(i) \triangleq \text{col}\big\{&\widetilde{\boldsymbol{w}}_1(i), \ldots, \widetilde{\boldsymbol{w}}_N(i), \widetilde{\boldsymbol{\phi}}_1(i), \ldots, \widetilde{\boldsymbol{\phi}}_N(i), \ldots, \\ &\widetilde{\boldsymbol{\phi}}_1(i-H+2), \ldots, \widetilde{\boldsymbol{\phi}}_N(i-H+2)\big\}, \end{aligned} \tag{11}$$

$$\begin{aligned} \widetilde{\boldsymbol{\phi}}_e(i) \triangleq \text{col}\big\{&\widetilde{\boldsymbol{\phi}}_1(i), \ldots, \widetilde{\boldsymbol{\phi}}_N(i), \widetilde{\boldsymbol{\phi}}_1(i-1), \ldots, \widetilde{\boldsymbol{\phi}}_N(i-1), \\ &\ldots, \widetilde{\boldsymbol{\phi}}_1(i-H+1), \ldots, \widetilde{\boldsymbol{\phi}}_N(i-H+1)\big\}, \end{aligned} \tag{12}$$

where $H = \max_{k,\ell}\{h_{\ell k}\}$. We define $\boldsymbol{w}^\delta \triangleq \boldsymbol{w}^o(\eta) - \boldsymbol{w}^o$, $\boldsymbol{w}_e^\delta \triangleq \mathbb{1}_H \otimes \boldsymbol{w}^\delta$ and $\boldsymbol{w}^{\delta\prime} \triangleq \boldsymbol{w}^\star - \boldsymbol{w}^o$, $\boldsymbol{w}_e^{\delta\prime} \triangleq \mathbb{1}_H \otimes \boldsymbol{w}^{\delta\prime}$. Before proceeding, we introduce the following assumption on the regressors.

Assumption 1. *The regressors $\boldsymbol{x}_k(i)$ arise from a zero-mean random process that is temporally white and spatially independent.*

This assumption is commonly used in the adaptive filtering literature [31]. It helps to simplify the derivations without constraining the conclusions.

A. Mean error behavior analysis

Using recursion (10) and data model (2), the $H \times 1$ block vector $\widetilde{\boldsymbol{w}}_e(i)$ can be written as:

$$\widetilde{\boldsymbol{w}}_e(i+1) = \boldsymbol{\mathcal{B}}(i)\widetilde{\boldsymbol{w}}_e(i) - \mu\boldsymbol{g}(i) + \mu\boldsymbol{r}(i) + \mu\eta\boldsymbol{f}, \tag{13}$$

where

$$\boldsymbol{\mathcal{B}}(i) \triangleq \boldsymbol{\mathcal{H}}\left[\boldsymbol{I}_{MH} - \mu\boldsymbol{\mathcal{R}}_{x,e}(i)\right], \tag{14}$$

$$\boldsymbol{g}(i) \triangleq \boldsymbol{\mathcal{H}}\boldsymbol{p}_{zx,e}(i), \tag{15}$$

$$\boldsymbol{r}(i) \triangleq \boldsymbol{\mathcal{H}}\boldsymbol{\mathcal{R}}_{x,e}(i)\boldsymbol{w}_e^\delta, \tag{16}$$

$$\boldsymbol{f} \triangleq \text{col}\Big\{\boldsymbol{\mathcal{D}}^\top(\boldsymbol{\mathcal{D}}\boldsymbol{w}^o(\eta) + \boldsymbol{b}), \boldsymbol{0}_{M\times 1}, \ldots, \boldsymbol{0}_{M\times 1}\Big\}, \tag{17}$$

with $\boldsymbol{\mathcal{H}}$ an $H \times H$ block matrix, $\boldsymbol{\mathcal{R}}_{x,e}(i)$ an $H \times H$ block diagonal matrix, and $\boldsymbol{p}_{zx,e}(i)$ an $H \times 1$ block vector:

$$\boldsymbol{\mathcal{H}} \triangleq [\boldsymbol{\mathcal{I}} - \mu\eta\, \boldsymbol{\mathcal{C}}_{\mathcal{D},e}], \tag{18}$$

$$\boldsymbol{\mathcal{R}}_{x,e}(i) \triangleq \text{diag}\{\boldsymbol{\mathcal{R}}_x(i), \boldsymbol{0}_{M\times M}, \ldots, \boldsymbol{0}_{M\times M}\}, \tag{19}$$

$$\boldsymbol{p}_{zx,e}(i) \triangleq \text{col}\{\boldsymbol{p}_{zx}(i), \boldsymbol{0}_{M\times 1}, \ldots, \boldsymbol{0}_{M\times 1}\}, \tag{20}$$

where

$$\boldsymbol{\mathcal{I}} = \begin{bmatrix} \boldsymbol{I}_M & \boldsymbol{0}_{M\times M(H-1)} \\ \boldsymbol{I}_{M(H-1)} & \boldsymbol{0}_{M(H-1)\times M} \end{bmatrix}, \tag{21}$$

$$\boldsymbol{\mathcal{C}}_{\mathcal{D},e} = = \left[\begin{array}{ccccc} \boldsymbol{C}_{D,1} & \boldsymbol{C}_{D,2} & \cdots & \boldsymbol{C}_{D,H} \\ \hline \multicolumn{4}{c}{\boldsymbol{0}_{M(H-1)\times MH}} \end{array}\right], \tag{22}$$

$$\boldsymbol{\mathcal{R}}_x(i) = \text{diag}\Big\{\boldsymbol{x}_k(i)\boldsymbol{x}_k^{\top}(i)\Big\}_{k=1}^{N}, \tag{23}$$

$$\boldsymbol{p}_{zx}(i) = \text{col}\Big\{\boldsymbol{x}_k(i)z_k(i)\Big\}_{k=1}^{N}. \tag{24}$$

and $\boldsymbol{C}_{D,h}$ is an $N \times N$ block matrix with (k,l)-th block given by:

$$[\boldsymbol{C}_{D,h}]_{k,l} = \begin{cases} \sum_{p\in\mathcal{J}_k} \boldsymbol{D}_{pk}^{\top}\boldsymbol{D}_{p\ell} & \text{if} \quad \ell \in \mathcal{I}_p \cap \mathcal{N}_k^{(h)}, \\ \boldsymbol{0} & \text{otherwise.} \end{cases} \tag{25}$$

Note that $\sum_{h=1}^{H} \boldsymbol{C}_{D,h} = \boldsymbol{\mathcal{D}}^{\top}\boldsymbol{\mathcal{D}}$.

Taking the expectation of both sides of (13) and using the fact that $\mathbb{E}\{\boldsymbol{g}(i)\} = \boldsymbol{0}$, we obtain:

$$\mathbb{E}\widetilde{\boldsymbol{w}}_e(i+1) = \boldsymbol{\mathcal{B}}\mathbb{E}\widetilde{\boldsymbol{w}}_e(i) + \mu\boldsymbol{r} + \mu\eta\boldsymbol{f}, \tag{26}$$

where

$$\boldsymbol{\mathcal{B}} \triangleq \boldsymbol{\mathcal{H}}\,[\boldsymbol{I}_{MH} - \mu\boldsymbol{\mathcal{R}}_{x,e}], \tag{27}$$

$$\boldsymbol{r} \triangleq \boldsymbol{\mathcal{H}}\boldsymbol{\mathcal{R}}_{x,e}\boldsymbol{w}_e^{\delta}, \tag{28}$$

$$\boldsymbol{\mathcal{R}}_{x,e} \triangleq \text{diag}\{\boldsymbol{\mathcal{R}}_x, \boldsymbol{0}_{M\times M}, \ldots, \boldsymbol{0}_{M\times M}\}. \tag{29}$$

The mean error vector $\mathbb{E}\{\widetilde{\boldsymbol{w}}_e(i)\}$ converges as $i \to \infty$ if the matrix $\boldsymbol{\mathcal{B}}$ is stable, i.e., the spectral radius of $\boldsymbol{\mathcal{B}}$ is less than 1. In this case, the asymptotic mean bias is given by:

$$\mathbb{E}\widetilde{\boldsymbol{w}}_e(\infty) = \lim_{i\to\infty} \mathbb{E}\widetilde{\boldsymbol{w}}_e(i) = \mu(\boldsymbol{I}_M - \boldsymbol{\mathcal{B}})^{-1}(\boldsymbol{r} + \eta\boldsymbol{f}). \tag{30}$$

Using similar definition with (11), we find that the extended mean error recursion $\mathbb{E}\widetilde{\boldsymbol{w}}_e'(i)$ with respect to $\boldsymbol{w}^{\star}$evolves according to:

$$\mathbb{E}\widetilde{\boldsymbol{w}}_e'(i+1) = \boldsymbol{\mathcal{B}}\mathbb{E}\widetilde{\boldsymbol{w}}_e'(i) + \mu\boldsymbol{r}_s. \tag{31}$$

where $\boldsymbol{r}_s \triangleq \boldsymbol{\mathcal{H}}\boldsymbol{\mathcal{R}}_{x,e}\boldsymbol{w}_e^{\delta\prime}$. When $\boldsymbol{\mathcal{B}}$ is stable, we obtain:

$$\mathbb{E}\widetilde{\boldsymbol{w}}_e'(\infty) = \lim_{i\to\infty} \mathbb{E}\widetilde{\boldsymbol{w}}_e'(i) = \mu(\boldsymbol{I}_{MH} - \boldsymbol{\mathcal{B}})^{-1}\boldsymbol{r}_s. \tag{32}$$

We observe that, when $\boldsymbol{w}^o$ satisfies the constraints, we have $\boldsymbol{w}^o = \boldsymbol{w}^o(\eta) = \boldsymbol{w}^{\star}$, $\boldsymbol{r} = \boldsymbol{0}$, $\boldsymbol{f} = \boldsymbol{0}$, and $\boldsymbol{r}_s = \boldsymbol{0}$. In this case, the asymptotic mean biases $\mathbb{E}\widetilde{\boldsymbol{w}}_e(\infty)$ and $\mathbb{E}\widetilde{\boldsymbol{w}}_e'(\infty)$ reduce to zero.

B. Mean-square-error behavior analysis

We shall evaluate the weighted variance $\mathbb{E}\{\|\widetilde{\boldsymbol{w}}_e(i+1)\|_{\boldsymbol{\Sigma}}^2\}$ where $\boldsymbol{\Sigma}$ is a positive semi-definite matrix that we are free to choose. Let $\boldsymbol{\sigma} \triangleq \text{vec}(\boldsymbol{\Sigma})$. In the following, we use the alternative notation $\|\boldsymbol{w}\|_{\boldsymbol{\sigma}}^2$ to refer to the same weighted squared norm $\|\boldsymbol{w}\|_{\boldsymbol{\Sigma}}^2$. Following the same line of reasoning as in [7], [10] for single-task diffusion strategies, and extending the arguments to our multitask scenario, we find:

$$\mathbb{E}\{\|\widetilde{\boldsymbol{w}}_e(i+1)\|_{\boldsymbol{\sigma}}^2\} = \mathbb{E}\{\|\widetilde{\boldsymbol{w}}_e(i)\|_{\boldsymbol{\mathcal{F}}\boldsymbol{\sigma}}^2\} + [\text{vec}(\boldsymbol{\mathcal{Y}}(i))]^{\top}\boldsymbol{\sigma}, \tag{33}$$

where $\boldsymbol{\mathcal{F}}$ is the $(MH)^2 \times (MH)^2$ matrix given by:

$$\boldsymbol{\mathcal{F}} \triangleq \mathbb{E}\{\boldsymbol{\mathcal{B}}^{\top}(i) \otimes \boldsymbol{\mathcal{B}}^{\top}(i)\}. \tag{34}$$

It is sufficient in this work to consider the case of sufficiently small step-sizes where the influence of terms involving higher-order powers of μ can be ignored [10], [31]. In this case, $\boldsymbol{\mathcal{F}}$ can be approximated by $\boldsymbol{\mathcal{F}} \approx \boldsymbol{\mathcal{B}}^{\top} \otimes \boldsymbol{\mathcal{B}}^{\top}$. Under this approximation, the stability of $\boldsymbol{\mathcal{F}}$ is ensured if the matrix $\boldsymbol{\mathcal{B}}$ is stable.

The matrix $\boldsymbol{\mathcal{Y}}(i)$ in (33) is given by:

$$\begin{aligned}\boldsymbol{\mathcal{Y}}(i) \triangleq &\mu^2\boldsymbol{\mathcal{G}}^{\top} + \mu^2\boldsymbol{\mathcal{Q}}^{\top} + 2\mu\boldsymbol{\mathcal{P}}^{\top}(i) + \mu^2\eta^2\boldsymbol{f}\boldsymbol{f}^{\top} + \\ &2\mu\eta\boldsymbol{f}\mathbb{E}\widetilde{\boldsymbol{w}}_e^{\top}(i)\boldsymbol{\mathcal{B}}^{\top} + 2\mu^2\eta\boldsymbol{f}\boldsymbol{r}^{\top},\end{aligned} \tag{35}$$

where $\boldsymbol{\mathcal{G}}$ and $\boldsymbol{\mathcal{Q}}$ are $H \times H$ block matrices given by:

$$\begin{aligned}\boldsymbol{\mathcal{G}} &\triangleq \mathbb{E}\{\boldsymbol{g}(i)\boldsymbol{g}^{\top}(i)\} \\ &= \boldsymbol{\mathcal{H}}\text{diag}\{\boldsymbol{\mathcal{T}}, \boldsymbol{0}_{M\times M}, \ldots, \boldsymbol{0}_{M\times M}\}\,\boldsymbol{\mathcal{H}}^{\top},\end{aligned} \tag{36}$$

$$\begin{aligned}\boldsymbol{\mathcal{Q}} &\triangleq \mathbb{E}\{\boldsymbol{r}(i)\boldsymbol{r}^{\top}(i)\} \\ &= \boldsymbol{\mathcal{H}}\text{diag}\{\boldsymbol{\mathcal{T}}_1, \boldsymbol{0}_{M\times M}, \ldots, \boldsymbol{0}_{M\times M}\}\,\boldsymbol{\mathcal{H}}^{\top},\end{aligned} \tag{37}$$

with $\boldsymbol{\mathcal{T}} \triangleq \text{diag}\{\sigma_{z,k}^2\boldsymbol{R}_{x,k}\}_{k=1}^{N}$, $\boldsymbol{\mathcal{T}}_1 \triangleq \mathbb{E}\{\boldsymbol{\mathcal{R}}_x(i)\boldsymbol{\mathcal{K}}_1\boldsymbol{\mathcal{R}}_x(i)\}$, and $\boldsymbol{\mathcal{K}}_1 \triangleq \boldsymbol{w}^{\delta}(\boldsymbol{w}^{\delta})^{\top}$. The evaluation of $\boldsymbol{\mathcal{T}}_1$ depends on higher order moments of the regressors. In the following, we shall evaluate $\boldsymbol{\mathcal{T}}_1$ when the regressors are zero-mean real Gaussian. For any square matrix $\boldsymbol{A}$ and zero-mean Gaussian regressors, we have [32]:

$$\begin{aligned}\mathbb{E}\{\boldsymbol{x}_k(i)\boldsymbol{x}_k^{\top}(i)\boldsymbol{A}\boldsymbol{x}_{\ell}(i)\boldsymbol{x}_{\ell}^{\top}(i)\} = &\boldsymbol{R}_{x,k}\boldsymbol{A}\boldsymbol{R}_{x,\ell} + \\ &\delta_{k,\ell}(\boldsymbol{R}_{x,k}\boldsymbol{A}^{\top}\boldsymbol{R}_{x,k} + \boldsymbol{R}_{x,k}\text{tr}(\boldsymbol{R}_{x,k}\boldsymbol{A})).\end{aligned} \tag{38}$$

Using (38), it can be verified that $\boldsymbol{\mathcal{T}}_1$ can be expressed as:

$$\begin{aligned}\boldsymbol{\mathcal{T}}_1 = \boldsymbol{\mathcal{R}}_x\boldsymbol{\mathcal{K}}_1\boldsymbol{\mathcal{R}}_x + \sum_{k=1}^{N}\Big(&\mathcal{S}_k(\boldsymbol{I}_N \otimes \boldsymbol{R}_{x,k})\boldsymbol{\mathcal{K}}_1^{\top}(\boldsymbol{I}_N \otimes \boldsymbol{R}_{x,k})\mathcal{S}_k + \\ &\mathcal{S}_k(\boldsymbol{I}_N \otimes \boldsymbol{R}_{x,k})\boldsymbol{\mathcal{Z}}_k\mathcal{S}_k\Big),\end{aligned} \tag{39}$$

where $\mathcal{S}_k \triangleq \text{diag}(\boldsymbol{e}_k^{\top}) \otimes \boldsymbol{I}_{M_k}$ is an $N \times N$ block diagonal matrix and $\boldsymbol{e}_k$ is the column vector with a unit entry at position k and zeros elsewhere. $\boldsymbol{\mathcal{Z}}_k$ is an $N \times N$ block matrix with the (m,n)-th block given by:

$$[\boldsymbol{\mathcal{Z}}_k]_{m,n} = [\text{vec}(\boldsymbol{R}_{x,k})]^{\top}\text{vec}([\boldsymbol{\mathcal{K}}_1]_{m,n})I_{M_k}. \tag{40}$$

The $H \times H$ block matrix $\boldsymbol{\mathcal{P}}(i)$ is given by:

$$\begin{aligned}\boldsymbol{\mathcal{P}}(i) &\triangleq \mathbb{E}\{\boldsymbol{\mathcal{B}}(i)\widetilde{\boldsymbol{w}}_e(i)\boldsymbol{r}^{\top}(i)\} \\ &= \boldsymbol{\mathcal{H}}\text{diag}\{\boldsymbol{\mathcal{T}}_2, \boldsymbol{0}_{M\times M}, \ldots, \boldsymbol{0}_{M\times M}\}\,\boldsymbol{\mathcal{H}}^{\top}\end{aligned} \tag{41}$$

with

$$\boldsymbol{\mathcal{T}}_2(i) \triangleq \mathbb{E}\{\boldsymbol{\mathcal{R}}_x(i)\boldsymbol{\mathcal{K}}_2(i)(I_M - \mu\boldsymbol{\mathcal{R}}_x(i))\}, \tag{42}$$

$$\boldsymbol{\mathcal{K}}_2(i) \triangleq \mathbb{E}\{\widetilde{\boldsymbol{w}}(i)\}(\boldsymbol{w}^{\delta})^{\top}. \tag{43}$$

Note that, following similar arguments as the one for $\boldsymbol{\mathcal{Q}}$, $\boldsymbol{\mathcal{P}}(i)$ can be evaluated for zero-mean real Gaussian regressors.

Starting from the initial condition $\widetilde{\boldsymbol{w}}_e(0)$ in (33) and comparing the expressions of $\mathbb{E}\{\|\widetilde{\boldsymbol{w}}_e(i+1)\|_{\boldsymbol{\sigma}}^2\}$ and $\mathbb{E}\{\|\widetilde{\boldsymbol{w}}_e(i)\|_{\boldsymbol{\sigma}}^2\}$, we obtain:

$$\begin{aligned}\mathbb{E}\{\|\widetilde{\boldsymbol{w}}_e(i+1)\|_{\boldsymbol{\sigma}}^2\} = &\mathbb{E}\{\|\widetilde{\boldsymbol{w}}_e(i)\|_{\boldsymbol{\sigma}}^2\} + \\ &\big[\text{vec}(\mathbb{E}\{\widetilde{\boldsymbol{w}}_e(0)\widetilde{\boldsymbol{w}}_e^{\top}(0)\})\big]^{\top}(\boldsymbol{\mathcal{F}} - I_{(MH)^2})\boldsymbol{\mathcal{F}}^i\boldsymbol{\sigma} + \\ &[\text{vec}(\boldsymbol{\mathcal{Y}}(i))]^{\top}\boldsymbol{\sigma} + \boldsymbol{\Gamma}(i)\boldsymbol{\sigma},\end{aligned} \tag{44}$$

where $\boldsymbol{\Gamma}(i)$ is an $(MH)^2 \times 1$ vector that evolves according to:

$$\boldsymbol{\Gamma}(i+1) = [\text{vec}(\boldsymbol{\mathcal{Y}}(i))]^{\top}(\boldsymbol{\mathcal{F}} - \boldsymbol{I}_{(MH)^2}) + \boldsymbol{\Gamma}(i), \tag{45}$$

with $\boldsymbol{\Gamma}(0) = \boldsymbol{0}_{(MH)^2\times 1}$.

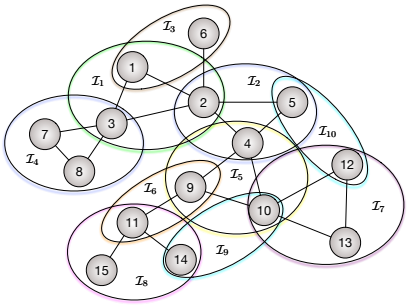

Fig. 1: Multitask MSE network with constraints.

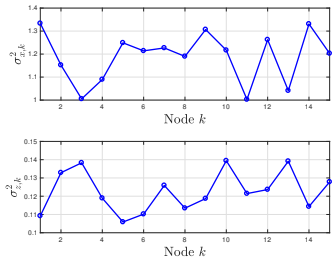

Fig. 2: Regressors and noise variances.

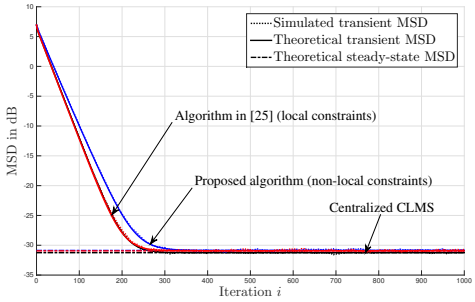

Fig. 3: MSD comparison (perfect model scenario).

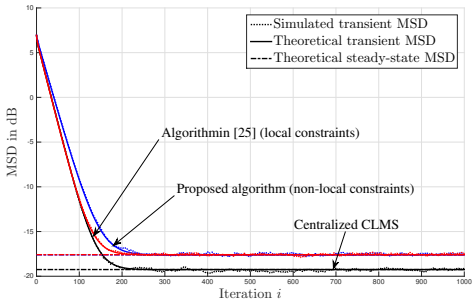

Fig. 4: MSD comparison (imperfect model scenario).

Recursion (33) converges to a steady-state value if the matrix $\boldsymbol{\mathcal{F}}$ is stable and the mean error vector $\mathbb{E}\{\widetilde{\boldsymbol{w}}_e(i)\}$ converges. In this case, we obtain from (33):

$$\lim_{i\to\infty} \mathbb{E}\{\|\widetilde{\boldsymbol{w}}_e(i)\|^2_{(\boldsymbol{I}_{(MH)^2}-\boldsymbol{\mathcal{F}})\boldsymbol{\sigma}}\} = [\text{vec}(\boldsymbol{\mathcal{Y}}(\infty))]^\top \boldsymbol{\sigma}, \tag{46}$$

where $\boldsymbol{\mathcal{Y}}(\infty)$ can be obtained from (30) and (35). Define the network mean-square-deviation (MSD) as: $\zeta^\star = \lim_{i\to\infty} \frac{1}{N}\mathbb{E}\{\|\widetilde{\boldsymbol{w}}(i)\|^2\}$. In this case, the steady-state network MSD can be written as:

$$\zeta^\star = \lim_{i\to\infty} \frac{1}{N}\mathbb{E}\{\|\widetilde{\boldsymbol{w}}_e(i)\|^2_{\boldsymbol{\sigma}_{ss}}\} \tag{47}$$

with $\boldsymbol{\sigma}_{ss} = \text{vec}\left(\text{diag}\{\boldsymbol{I}_M, \boldsymbol{0}_{M\times M}, \ldots, \boldsymbol{0}_{M\times M}\}\right)$, then the network MSD can be obtained from (46) by selecting $\boldsymbol{\sigma}$ that satisfies:

$$(\boldsymbol{I}_{(MH)^2} - \boldsymbol{\mathcal{F}})\boldsymbol{\sigma} = \frac{1}{N}\boldsymbol{\sigma}_{ss}. \tag{48}$$

Replacing $\boldsymbol{\sigma}$ in (46) by $\frac{1}{N}(\boldsymbol{I}_{(MH)^2} - \boldsymbol{\mathcal{F}})^{-1}\boldsymbol{\sigma}_{ss}$ we have:

$$\zeta^\star = \frac{1}{N}[\text{vec}(\boldsymbol{\mathcal{Y}}(\infty))]^\top (\boldsymbol{I}_{(MH)^2} - \boldsymbol{\mathcal{F}})^{-1}\boldsymbol{\sigma}_{ss}. \tag{49}$$

Finally, the transient and steady-state behavior of $\mathbb{E}\{\|\widetilde{\boldsymbol{w}}'_e(i)\|^2_{\boldsymbol{\sigma}}\}$ can be derived from the following relation:

$$\mathbb{E}\{\|\widetilde{\boldsymbol{w}}'_e(i)\|^2_{\boldsymbol{\sigma}}\} = \mathbb{E}\{\|\widetilde{\boldsymbol{w}}_e(i)\|^2_{\boldsymbol{\sigma}}\} + 2\mathbb{E}\{\widetilde{\boldsymbol{w}}_e(i)\}\boldsymbol{\Sigma}\boldsymbol{w}_{d,e} + \|\boldsymbol{w}_{d,e}\|^2_{\boldsymbol{\Sigma}}$$

with $\boldsymbol{w}_{d,e} = \mathbb{1}_H \otimes (\boldsymbol{w}^\star - \boldsymbol{w}^o(\eta))$.

IV. Simulations

In this section, we provide experimental results to illustrate the convergence of algorithm (10) and to validate our theoretical models. We considered a network of 15 nodes with the topology and the constraints shown in Fig. 1. We randomly sampled 10 linear constraints of the form $\sum_{\ell\in\mathcal{I}_p} d_{p\ell}\boldsymbol{w}_\ell = b_p \cdot \mathbb{1}_2$, where the coefficients $d_{p\ell}$ and b_p were randomly chosen from $\{-2,-1,1,2\}$. The regression vectors $\boldsymbol{x}_k(i)$ were zero-mean Gaussian with covariance matrix $\boldsymbol{R}_{x,k} = \sigma^2_{x,k}\boldsymbol{I}_2$. The noises $z_k(i)$ were zero-mean i.i.d. Gaussian random variables independent of any other signal with variances $\sigma^2_{z,k}$. The variances $\sigma^2_{x,k}$ and $\sigma^2_{z,k}$ used in the simulations are shown in Fig. 2. The results were averaged over 200 Monte-Carlo runs.

We considered two scenarios: i) the parameter vector $\boldsymbol{w}^o = \boldsymbol{w}_o$ where $\boldsymbol{w}_o$ satisfies the constraints, i.e. $\boldsymbol{w}^o = \boldsymbol{w}^\star$(Fig.3); ii) $\boldsymbol{w}^o$ does not satisfy the constraints, specifically, we perturbed it as $\boldsymbol{w}^o = \boldsymbol{w}_o + \boldsymbol{u}$ where $\boldsymbol{u} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ (Fig.4). The step-size μ was set to 0.02. We compared our algorithm (10) with the centralized CLMS algorithm [24], [33]. Furthermore, for comparison purposes, we assumed additional links connecting nodes 1 to 6, 5 to 12, 10 to 14, and 14 to 15. In this case, the constraints are local and the algorithm derived in [25] can be applied. We set $\mu = 0.018$ for the centralized CLMS algorithm and for algorithm [25] (with additional links) so that their steady-state MSD match. Observe that the simulation results match well the theoretical models. Furthermore, our algorithm performs well in the mean-square-error compared to the centralized solution. Finally, observe that, as expected, the delays emerging from the multi-hop protocols required in non-local constraints scenarios will lead to a slower convergence rate.

V. Conclusion

We proposed a distributed multitask algorithm for estimating multiple parameter vectors that are coupled through non-local linear equality constraints. Based on the penalty method, we solved the original constrained problem by approximating it into an unconstrained one. A multi-hop relay protocol was employed in order to deal with the non-local constraints and to devise a distributed algorithm. The stochastic behavior of the algorithm in the mean and in the mean-square-error sense was studied. Simulation results were conducted to show the effectiveness of the proposed method and to validate our theoretical performance analysis.

REFERENCES

[1] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo, "Distributed detection and estimation in wireless sensor networks," in *Academic Press Library in Signal Processing*, S. Theodoridis and R. Chellappa, Eds. Academic Press, Elsevier, 2013, vol. 2, pp. 329–408.

[2] Y. Zhang, W. P. Tay, K. H. Li, and D. Gaïti, "Distributed boundary estimation for spectrum sensing in cognitive radio networks," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 11, pp. 1961–1973, 2014.

[3] F. Meyer, O. Hlinka, H. Wymeersch, E. Riegler, and F. Hlawatsch, "Distributed localization and tracking of mobile networks including noncooperative objects," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 1, pp. 57–71, 2016.

[4] L. Xie, D.-H. Choi, S. Kar, and H. V. Poor, "Fully distributed state estimation for wide-area monitoring systems," *IEEE Transactions on Smart Grid*, vol. 3, no. 3, pp. 1154–1169, 2012.

[5] D. P. Bertsekas, "A new class of incremental gradient methods for least squares problems," *SIAM Journal on Optimization*, vol. 7, no. 4, pp. 913–926, 1997.

[6] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1035–1048, 2010.

[7] A. H. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z. J. Towfic, "Diffusion strategies for adaptation and learning over networks: an examination of distributed strategies and network behavior," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 155–171, 2013.

[8] S.-Y. Tu and A. H. Sayed, "Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks," *IEEE Transactions on Signal Processing*, vol. 60, no. 12, pp. 6217–6234, 2012.

[9] Z. J. Towfic and A. H. Sayed, "Stability and performance limits of adaptive primal-dual networks," *IEEE Transactions on Signal Processing*, vol. 63, no. 11, pp. 2888–2903, 2015.

[10] A. H. Sayed, "Diffusion adaptation over networks," in *Academic Press Library in Signal Processing*, S. Theodoridis and R. Chellappa, Eds. Academic Press, Elsevier, 2013, vol. 3, pp. 322–454.

[11] ——, "Adaptation, learning, and optimization over networks," *Foundations and Trends in Machine Learning*, vol. 7, no. 4-5, pp. 311–801, 2014.

[12] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proceedings of the 4th international symposium on Information processing in sensor networks*. IEEE, 2005, p. 9.

[13] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[14] J. Chen, C. Richard, and A. H. Sayed, "Multitask diffusion adaptation over networks," *IEEE Transactions on Signal Processing*, vol. 62, no. 16, pp. 4129–4144, 2014.

[15] ——, "Diffusion LMS over multitask networks," *IEEE Transactions on Signal Processing*, vol. 63, no. 11, pp. 2733–2748, 2015.

[16] Y. Wang, W. P. Tay, and W. Hu, "A multitask diffusion strategy with optimized inter-cluster cooperation," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 3, pp. 504–517, 2017.

[17] R. Nassif, C. Richard, A. Ferrari, and A. H. Sayed, "Proximal multitask learning over networks with sparsity-inducing coregularization," *IEEE Transactions on Signal Processing*, vol. 64, no. 23, pp. 6329–6344, 2016.

[18] J. Plata-Chaves, N. Bogdanović, and K. Berberidis, "Distributed diffusion-based LMS for node-specific adaptive parameter estimation," *IEEE Transactions on Signal Processing*, vol. 63, no. 13, pp. 3448–3460, 2015.

[19] J. Chen, C. Richard, A. O. Hero, and A. H. Sayed, "Diffusion LMS for multitask problems with overlapping hypothesis subspaces," in *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2014.

[20] J. Chen, C. Richard, and A. H. Sayed, "Multitask diffusion adaptation over networks with common latent representations," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 3, pp. 563–579, 2017.

[21] D. P. Bertsekas, *Network optimization: continuous and discrete models*. Athena Scientific, 1998.

[22] J. F. Mota, J. M. Xavier, P. M. Aguiar, and M. Puschel, "Distributed basis pursuit," *IEEE Transactions on Signal Processing*, vol. 60, no. 4, pp. 1942–1956, 2012.

[23] C. Shen, T.-H. Chang, K.-Y. Wang, Z. Qiu, and C.-Y. Chi, "Distributed robust multicell coordinated beamforming with imperfect CSI: An ADMM approach," *IEEE Transactions on signal processing*, vol. 60, no. 6, pp. 2988–3003, 2012.

[24] R. Nassif, C. Richard, A. Ferrari, and A. H. Sayed, "Diffusion LMS for multitask problems with local linear equality constraints," *IEEE Transactions on Signal Processing*, to appear. Also available as arXiv:1610.02943.

[25] F. Hua, R. Nassif, C. Richard, H. Wang, and J. Huang, "Penalty-based multitask distributed adaptation over networks with constraints," submitted to 2017 Asilomar Conference on Signals, Systems and Computers.

[26] B. T. Polyak, *Introduction to Optimization*. Optimization Software, New York, 1987.

[27] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear programming: theory and algorithms*. John Wiley & Sons, 2013.

[28] D. G. Luenberger and Y. Ye, *Linear and nonlinear programming*. Springer, 2015, vol. 228.

[29] Z. J. Towfic and A. H. Sayed, "Adaptive penalty-based distributed stochastic convex optimization," *IEEE Transactions on Signal Processing*, vol. 62, no. 15, pp. 3924–3938, 2014.

[30] W. Dargie and C. Poellabauer, *Fundamentals of wireless sensor networks: theory and practice*. John Wiley & Sons, 2010.

[31] A. H. Sayed, *Adaptive filters*. John Wiley & Sons, 2011.

[32] K. B. Petersen and M. S. Pedersen, "The matrix cookbook. version: November 15, 2012," 2012.

[33] O. L. Frost, "An algorithm for linearly constrained adaptive array processing," *Proceedings of the IEEE*, vol. 60, no. 8, pp. 926–935, 1972.