Received 19 August 2021; revised 22 December 2021; accepted 2 January 2022. Date of publication 11 January 2022; date of current version 28 March 2022. The review of this article was arranged by Associate Editor C. Richard.

Digital Object Identifier 10.1109/OJSP.2022.3141965

Optimal Diffusion Learning Over Networks—Part II: Multitask Algorithms

RICARDO MERCHED (Senior Member, IEEE)

Department of Electronics and Computer Engineering, Universidade Federal do Rio de Janeiro, Rio de Janeiro 21941-901, Brazil

This work was supported in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior–Brasil (CAPES)–Finance Code 001.

ABSTRACT In Part I of this presentation, we have formulated single and multitask quadratic optimization problems, where agents are subject to quadratic, smoothing constraints over a graph. We have focused particularly on single task designs, whereby node uncertainties and their strength relative to un-regularized cost are tackled altogether by means of an adaptive penalty function. In this sequel, we readdress the multitask problem and propose new distributed implementations for their corresponding exact *leaky*-RLS solutions. We motivate a network formulation from a standalone viewpoint by capitalizing on the fact that 1) for regressors having uncorrelated entries, the performance of an efficient $\mathcal{O}(M^2)$ conjugate-gradient (CG) realization of the leaky LS solution is identical to the one of an RLS filter; 2) a CG implementation does not require inversion of the underlying sample covariance matrix. Simple arguments yield an extended network-CG algorithm that relies on node-level recursions employing distinct step-sizes. Unlike the exponentially-weighted RLS algorithm, which tapers off regularization over time, a persistent penalty strength conforms with the very purpose of the equivalent network trust-region problem, while granting a well-conditioned solution. The approach further yields another family of single-task algorithms in terms of network linearly constrained solutions, which can be contrasted to the ones proposed in Part I. In particular, the exact linearly-constrained network LMS implementation proposed here outperforms the adaptive relative-variance NLMS, under much lower computational requirements.

INDEX TERMS Adaptation, combination weights, diffusion networks, fusion, least-squares, multitask, sparsity.

I. INTRODUCTION

Regularization plays a central part in general parameter estimation, serving a variety of purposes in the realm of adaptation and learning. The benefits range from resolving non-uniqueness in rank-defficient risk and empirical cost formulations, to countering model over-fitting and ill-conditioning inherent to arbitrary data streaming. Ultimately, regularization conveys a priori information on the unknown parameters, in a way that the optimization task be restricted solely to a space of meaningful solutions [1].

Due to their mathematical tractability, quadratically regularizers to quadratic costs have been vastly studied within the adaptive filtering community, both in the stochastic and deterministic settings [1]. In general, a quadratic penalty function applied to a least-squares (LS) cost leads to a *leaky* type solution, whose form deviates from the un-regularized one in proportion to the penalty strength, say, η. For example, in the stochastic sense, a gradient-descent approach leads to the so-called *leaky*-LMS algorithm [3], while in the deterministic scenario, the regularized solution can be expressed via an analogous, exponentially-weighted *leaky*-RLS (LWRLS) algorithm [4], [5]. These formulations come, nevertheless, with a few well known drawbacks. First, for arbitrary ridge strengths, the value of the least attainable cost is always larger than the minimum value of the corresponding un-regularized function. Second, the resulting leaky-RLS recursions are of cubic complexity, and therefore impractical for real applications. The standard WRLS algorithm circumvents this problem by allowing the ridge factor to be of a special time-varying form, i.e., $\eta_i = \lambda^{i+1}$, in terms of a forgetting factor λ. This is what renders the WRLS algorithm of $\mathcal{O}(M^2)$ complexity per iteration, which is reasonable for modern applications with powerful DSP capabilities [6]. Unfortunately, this particular choice of η_i has a decaying strength, i.e., regularization dies out over time. Such assumption can be highly undesirable in situations where data streaming halts, becomes ill-conditioned, or simply because we are interested in a constrained solution, not attainable by the WRLS recursions. In this latter respect, the optimal ridge strength is obtained as the solution of a secular equation [7], as the result of the Karush-Kuhn-Tucker (KKT) multiplier method applied to the original constrained problem.

A. RELATED WORK

There have been some relevant works addressing the choice of the ridge factor and the implementation of persistently regularized RLS iterations, yet for different purposes. One of the earliest techniques concerning uniqueness and conditioning of the Ricatti variable was proposed in [8], a method referred to as *dithering*. Since the solution to the regularized stochastic problem can be interpreted as the addition of uncorrelated noise signal to the input [see (10) and (51) further ahead], the idea is to add a low power white noise directly to the input sequence, so that the sample covariance matrix becomes positive definite in the MSE sense. Although useful in its own right, the process leads to degraded performance compared to the leaky solution in the LMS sense, considering i.i.d. signals [9]. Other methods towards better conditioning which preserve $\mathcal{O}(M^2)$ complexity have been pursued indirectly in [10], [11], and more recently in [12], where the positive-definiteness of the sample covariance is assured on average. In this sense, the leaky-RLS algorithm aforementioned provides a numerically stable solution compared to the WRLS recursions with a fixed ridge strength. This, however, consists in $\mathcal{O}(M^3)$ complexity recursions, since they still require inversion of a non-structured sample covariance matrix. This complexity issue can be tackled in practice, e.g., via a dichotomous coordinate descent (DCD) algorithm [13], as inverstigated in several instances for quadratic and non-smooth regularizers [14]. The importance of having a variable regularization has been earlier recognized in other scenarios where the excitation power drops significantly [15], and in adaptive beamforming applications [16], where the sample covariance is naturally rank defficient. Recently, the DCD mechanism has been deployed for setting a varying-strength regularizer in the context of echo cancellation [17]. In this same respect, the *reweighted LS problem* is another example where time-varying regularization has been tackled instead via conjugate gradient techniques [2], addressed, e.g., in [18], [19]. Obviously, all the issues that arise regarding regularization in autonomous scenarios also exist in a network formulation; this justifies further analysis and proper extensions, which depend on the objective one is mainly concerned with.

In the context of distributed learning, regularization enforces a desired structure on the tasks performed by agents over a graph. For example, it can reflect a measure of proximity among the target parameters in a multitask setting, and/or a desired level of *spatial* sparsity imposed on the agents across the network. In Part I of this work, we have formulated such problems in terms of an extended vector parameter $\mathcal{w} = \text{col}\{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_N\}$ reflecting the structure of a strongly-connected network of N agents, subject to local linear models. More specifically, we assume that each agent k receives streaming data $\{d_k(i), \boldsymbol{u}_{k,i}\}$, related via a linear regression model of the form $d_k(i) = \boldsymbol{u}_{k,i}\boldsymbol{w}_k^o + v_k(i)$, where i is the time index, $\boldsymbol{w}_k^o$ is an unknown local parameter of size $M \times 1$, $\boldsymbol{u}_{k,i}$ is a regression (row) vector of size $1 \times M$, and $v_k(i)$ is an additive zero-mean white noise, which is time and spatially uncorrelated with other data. A global linear model for the agents is then formed by means of the following extended definitions :

$$\mathcal{w}^o = \text{col}\{\boldsymbol{w}_1^o, \boldsymbol{w}_2^o, \ldots, \boldsymbol{w}_N^o\} \tag{1}$$

$$\boldsymbol{d}_i = \text{col}\{d_1(i), d_2(i), \ldots, d_N(i)\} \tag{2}$$

$$\mathcal{U}_i = \text{bdiag}\left\{\boldsymbol{u}_{1,i}, \boldsymbol{u}_{2,i}, \ldots, \boldsymbol{u}_{N,i}\right\} \tag{3}$$

$$\boldsymbol{v}_i = \text{col}\{v_1(i), v_2(i), \ldots, v_N(i)\} \tag{4}$$

so that

$$\boldsymbol{d}_i \;=\; \mathcal{U}_i\mathcal{w}^o + \boldsymbol{v}_i \tag{5}$$

The goal in a multitask problem is to solve

$$\min_{\mathcal{w}} \{J(\mathcal{w}) \;+\; \rho(\mathcal{w})\} \tag{6}$$

where $J(\mathcal{w})$ is some aggregate risk or empirical cost function, and $\rho(\mathcal{w})$ is a regularizer whose purpose is to enforce a desired structure on the extended quantitiy $\mathcal{w}$. For example, a quadratic $\rho(\mathcal{w})$ has been considered in [20] for analysis of diffusion adaptation via a stochastic gradient-descent algorithms, considering general costs $J(\mathcal{w})$.

B. MAIN RESULTS

In this paper, we revisit the leaky-RLS problem from the viewpoint of its equivalent constrained risk formulation, whose optimal ridge factor is well defined. We motivate the idea by first considering a standalone setting, which may include non-smooth regularizers cast into approximate re-weighted LS formulations (see, e.g., [21] and the references therein). These problems motivate analogous conjugate-gradient (CG) implementations at the network level, and become particularly convenient for the following reasons: 1) First, the network CG recursions do not require covariance matrix inversions, but only multiplications by a block-sparse matrix, induced by the network topology. This will readily suggest a distributed structure for the solution with reduced comptational requirements. Note that because normally nodes make use of independent signals, correlation is only due to cooperation; 2) second, the increasing strength of the penalty term in this scenario is suitable to a CG implementation of reduced complexity, while mantaining the desired RLS convergence speed.

The CG algorithm suggests a feasible implemetation for the exact multitask problem, and yields alternative solutions to the single task LS problem that can be contrasted to the main algorithms proposed in Part I. More specifically, the following results complement the accompanying article:

1) A network leaky-RLS solution to the multitask scenario, based on an extended CG formulation (Table 2);
2) A network linearly constrained RLS solution to the single-task scenario, based on an extended CG formulation (Table 3);
3) A network linearly constrained LMS solution to the single-task scenario, based on an extended CG formulation (Table 4);

The proposed network linearly constrained LMS algorithm outperforms the relative variance NLMS proposed in Part I, regardless of the type of signals used by the agents. We highlight that the block-sparse matrix involvolved in the extended CG algorithm can be made as simple as one having two unity elements per node, leading to one addition per node as a combination strategy. Simulations verify the accuracy of the proposed recursions.

II. MOTIVATION

We shall carry the same mathematical notation of Part I in the sequel. We start from a standalone formulation, since it will be the basis for our extensions to the cooperative scenario and is useful in its own right. Under the assumptions previously stated for the general model (5), we consider a scenario based on streaming data $\{d(i), \boldsymbol{u}_i\}$, related via the linear regression,

$$d(i) = \boldsymbol{u}_i \boldsymbol{w}^o + v(i) \tag{7}$$

Let ν be a bound on the quadratic norm of the meaninful solutions of the following constrained MSE risk:

$$\min_{\boldsymbol{w}} \ \mathbb{E}\,|d(i) - \boldsymbol{u}_i\boldsymbol{w}|^2 \qquad \text{s.t.} \quad \|\boldsymbol{w}\|^2_{\boldsymbol{\Pi}} \leq \nu \tag{8}$$

where $\boldsymbol{\Pi}$ is some weighting matrix. By virtue of the KKT multiplier method, this constrained formulation becomes equivalent to the unconstrained problem,

$$\min_{\boldsymbol{w}} \ \mathbb{E}\,|d(i) - \boldsymbol{u}_i\boldsymbol{w}|^2 \ + \ \eta^o\|\boldsymbol{w}\|^2_{\boldsymbol{\Pi}} \tag{9}$$

The so-called leaky solution of (9) is given by

$$\boldsymbol{w}^{\eta^o} = (\eta^o\boldsymbol{\Pi} + \boldsymbol{R_u})^{-1}\boldsymbol{R_{du}} \tag{10}$$

where $\boldsymbol{R}_u = \mathbb{E}\,(\boldsymbol{u}_i^\star\boldsymbol{u}_i), \boldsymbol{R}_{du} = \mathbb{E}\,[\boldsymbol{u}_i^\star d(i)]$, and η^o is the positive solution of the secular equation

$$\|\boldsymbol{w}^{\eta^o}\|^2_{\boldsymbol{\Pi}} = \nu \tag{11}$$

Now, under ergodicity arguments, and for a large data set, it is common to replace (9) by its empirical counterpart, say,

$$\min_{\boldsymbol{w}} \ \frac{1}{i}\|\boldsymbol{y}_i - \boldsymbol{H}_i\boldsymbol{w}\|^2 \qquad \text{s.t.} \quad \|\boldsymbol{w}\|^2_{\boldsymbol{\Pi}} \leq \nu \tag{12}$$

where we collect realizations up to time i into the quantities

$$\boldsymbol{H}_i = \text{col}\{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_i\} \tag{13}$$

$$\boldsymbol{y}_i = \text{col}\{d(1), d(2), \ldots, d(i)\}. \tag{14}$$

This is also equivalent to a ridge regression formulation:

$$\min_{\boldsymbol{w}} \ \frac{1}{i}\|\boldsymbol{y}_i - \boldsymbol{H}_i\boldsymbol{w}\|^2 \ + \ \eta_i'\|\boldsymbol{w}\|^2_{\boldsymbol{\Pi}} \tag{15}$$

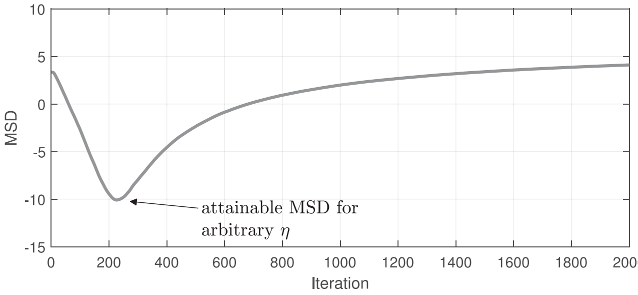


FIGURE 1. MSD of $\boldsymbol{w}_{\eta,i}$ relative to the stochastic solution.

where for arbitrary i (i.e., not large), the optimal strength η_i' must be allowed to be data dependent. Of course, we should expect that $\eta_i' \to \eta^o$ when $i \to \infty$. By defining $\eta_i \overset{\Delta}{=} i\eta_i'$, we can express (15) more compactly as

$$\min_{\boldsymbol{w}} \ \|\boldsymbol{y}_i - \boldsymbol{H}_i\boldsymbol{w}\|^2_{\boldsymbol{\Lambda}_i} \ + \ \eta_i\|\boldsymbol{w}\|^2_{\boldsymbol{\Pi}} \tag{16}$$

where we have further introduced the diagonal weighting matrix $\boldsymbol{\Lambda}_i = \text{diag}\{\lambda^{i-1}, \ldots, \lambda, 1\}$, with $0 \ll \lambda \leq 1$ denoting a forgetting factor, in order to account for possible nonstationarity in $\boldsymbol{R}_u$. The solution is given by

$$\boldsymbol{w}_{\eta_i,i} = \boldsymbol{P}_i\boldsymbol{H}_i^\star\boldsymbol{\Lambda}_i\boldsymbol{y}_i, \quad \text{with} \tag{17}$$

$$\boldsymbol{P}_i = (\eta_i\boldsymbol{\Pi} + \boldsymbol{H}_i^\star\boldsymbol{\Lambda}_i\boldsymbol{H}_i)^{-1} \tag{18}$$

where similarly to (11), the optimal ridge factor satisfies

$$\|\boldsymbol{w}_{\eta_i^o,i}\|^2_{\boldsymbol{\Pi}} = \nu. \tag{19}$$

This relation shows that we should pursue the exact strength adaptively so as to satisfy the trust-region set in (8) in the MSE sense. This is expected, since for $i \to \infty$, in a typical scenario we have $\boldsymbol{P}_i^{-1} \to \infty$, meaning that for a fixed, arbitrary η, after having learned the solution $\boldsymbol{w}_{\eta,i}$ to a certain extent, the strength of $\boldsymbol{P}_i^{-1}$ will excel η, so that the un-regularized solution will be weighted more heavily. Fig. 1 illustrates this fact via the mean-square-deviation (MSD) of $\boldsymbol{w}_{\eta,i}$ relative to $\boldsymbol{w}^{\eta^o}$ in (10), and given by

$$\text{MSD} = \mathbb{E}\,\|\boldsymbol{w}_{\eta,i} - \boldsymbol{w}_{\eta^o}\|^2 \tag{20}$$

Now, in general, the solution to (19) requires iterative methods, which may become impractical given that it must be computed at every iteration. For large i, however, the approximation (12) suggests that we can roughly set $\eta_i^o \approx \eta_i = i\eta^o$, considering that $\boldsymbol{\Lambda}_i = \boldsymbol{I}$.

The above reasoning can be used to accomodate nonsmooth regularizers as well. This becomes convenient for the identification of sparse systems, such as acoustic echo paths, or digital-TV channels, which consist of a few large coefficients within a long span of negligible samples. For example, consider the ℓ_1-norm constraint in

$$\min_{\boldsymbol{w}} \ \mathbb{E}\,|d(i) - \boldsymbol{u}_i\boldsymbol{w}|^2 \qquad \text{s.t.} \quad \|\boldsymbol{w}\|_1 \leq \nu \tag{21}$$

which leads to the equivalent regularized formulation,

$$\min_{\boldsymbol{w}} \ \mathbb{E}\,|d(i)-\boldsymbol{u}_i\boldsymbol{w}|^2 \ + \ \eta^o\|\boldsymbol{w}\|_1 \tag{22}$$

as well as to its LS counterpart:

$$\min_{\boldsymbol{w}} \ \|\boldsymbol{y}_i-\boldsymbol{H}_i\boldsymbol{w}\|^2_{\boldsymbol{\Lambda}_i} \ + \ \eta_i\|\boldsymbol{w}\|_1 \tag{23}$$

Note that we can rework (23) into a weighted LS problem as

$$\min_{\boldsymbol{w}} \ \|\boldsymbol{y}_i-\boldsymbol{H}_i\boldsymbol{w}\|^2_{\boldsymbol{\Lambda}_i} \ + \ \eta_i\|\boldsymbol{w}\|^2_{\boldsymbol{\Delta}} \tag{24}$$

with $\boldsymbol{\Delta} = \mathrm{diag}\{1/|w_0|, 1/|w_1|, \ldots 1/|w_{M-1}|\}$. Given a sufficiently small positive scalar δ, these weights are normally replaced by ones in terms of the most recent estimate, i.e.,

$$\boldsymbol{\Delta} \leftarrow \boldsymbol{\Delta}_{1,i} = \mathrm{diag}\left\{\frac{1}{|w_{0,i-1}|+\delta}, \ldots, \frac{1}{|w_{M-1,i-1}|+\delta}\right\} \tag{25}$$

leading to an *iteratively weighted LS* scheme. This corresponds to replacing the original constraint by a time-varying one, i.e., $\|\boldsymbol{w}\|^2_{\boldsymbol{\Delta}_i} \leq \nu_i$, where ν_i is an estimate for the true bound ν.

In the same vein, we can consider the ℓ_0-norm regularizer

$$\min_{\boldsymbol{w}} \ \mathbb{E}\,|d(i)-\boldsymbol{u}_i\boldsymbol{w}|^2 \qquad \text{s.t.} \quad \|\boldsymbol{w}\|_0 \leq \nu \tag{26}$$

where here ν is given by the number of non-zero entries of $\boldsymbol{w}$. Similar reasoning (see [21] and its references) allows us to interpret (26) as a weighted ℓ_1-norm problem

$$\min_{\boldsymbol{w}} \ \mathbb{E}\,|d(i)-\boldsymbol{u}_i\boldsymbol{w}|^2 \qquad \text{s.t.} \quad \|\boldsymbol{w}\|_{1,\Delta} \leq \nu \tag{27}$$

which is equivalent to a weighted version of (22), with $\boldsymbol{\Delta}$ given by (25). This in turn can be expressed as a quadratically regularized problem, viz.,

$$\min_{\boldsymbol{w}} \ \mathbb{E}\,|d(i)-\boldsymbol{u}_i\boldsymbol{w}|^2 \qquad \text{s.t.} \quad \|\boldsymbol{w}\|^2_{\Delta} \leq \nu \tag{28}$$

which, again, leads to a weighted LS formulation of the form (24) with[1]

$$\boldsymbol{\Delta} \leftarrow \boldsymbol{\Delta}_{2,i} = \mathrm{diag}\left\{\frac{1}{|w_{0,i-1}|^2+\delta}, \ldots, \frac{1}{|w_{M-1,i-1}|^2+\delta}\right\}. \tag{30}$$

A. ADAPTIVE LEAKY-RLS

Consider now the WLS problem (16) with $\eta_i\boldsymbol{\Pi} \longleftarrow \boldsymbol{\Pi}_i$. We may write (18) equivalently as

$$\boldsymbol{P}'^{-1}_i = \lambda\boldsymbol{P}'^{-1}_{i-1} \ + \ \boldsymbol{u}^\star_i\boldsymbol{u}_i \tag{31}$$

$$\boldsymbol{P}^{-1}_i = \boldsymbol{\Pi}_i + \boldsymbol{P}'^{-1}_i \tag{32}$$

In general, efficient propagation of $\boldsymbol{P}_i$ is hindered by the fact that $\boldsymbol{\Pi}_i$ is arbitrary [4], [5]. Yet, there are some useful situations where an $\mathcal{O}(M^2)$ leaky-RLS recursion can be obtained. A relevant scenario arises when the entries of the regressor are uncorrelated, or even independent. In this case, an efficient condition number dependent algorithm such as the conjugate-gradient (CG) algorithm is highly motivated.

Note that we can compute the solution to (16) iteratively in terms of

$$\boldsymbol{s}_i \ = \ \boldsymbol{H}^\star_i\boldsymbol{\Lambda}_i\boldsymbol{y}_i \ = \ \lambda\boldsymbol{s}_{i-1} \ + \ \boldsymbol{u}^\star_i d(i) \tag{33}$$

by solving

$$\boldsymbol{P}^{-1}_i\boldsymbol{w}_{\eta_i,i} = \boldsymbol{s}_i. \tag{34}$$

An $\mathcal{O}(M^2)$ solution to (34) can be attained by relying on a CG algorithm, with initial guess taken as $\boldsymbol{w}_{\eta_i,i-1}$ [2]. The CG algorithm derives from the conjugate direction (CD) method, originally applied to a quadratic cost function. The goal is to minimize a cost along a direction $\boldsymbol{p}_{n-1}$ via successive updates of the form

$$\boldsymbol{x}_n = \boldsymbol{x}_{n-1} \ + \ a_{n-1}\boldsymbol{p}_{n-1} \tag{35}$$

In the case of cost (16), the stepsize a_{n-1} is selected so as to minimize its value at $\boldsymbol{x}_n$, which has the form of a WLS solution:

$$\alpha_{n-1} = \frac{\boldsymbol{p}^\star_{n-1}\boldsymbol{P}^{-1}_i(\boldsymbol{x}_{n-1}-\boldsymbol{x}^o)}{\boldsymbol{p}^\star_{n-1}\boldsymbol{P}^{-1}_i\boldsymbol{p}_{n-1}}. \tag{36}$$

The idea motivates the computation of a set of conjugate basis vectors with respect to $\boldsymbol{P}^{-1}_i$, by relying on the representation of (16) in terms of such basis. The coefficients of this representation are obtained by enforcing conjugation between the referred directions with respect to $\boldsymbol{P}^{-1}_i$, which in turn are generated by a set of M basis vectors as subsequent gradients of (16). Table 1 lists the basic CG recursions in its original form.

The CG algorithm is further motivated by the fact that, in our context, the ridge strength provides persistent regularizarion, which happens to increase in time since $\eta_i = i\eta^o$. This leads to well conditioned recursions, while relying on low complexity iterations. That is, let $\boldsymbol{\Pi}_i = i\eta^o\boldsymbol{\Pi}$. The complexity of the CG algorithm is known to be upper bounded by $\mathcal{O}(\kappa_i M^2)$ per iteration, where κ_i is the condition number of the coefficient matrix $\boldsymbol{P}^{-1}_i$:

$$\kappa_i \ = \ \frac{i\eta^o + \sigma^2_{\max}(\boldsymbol{\Lambda}^{1/2}_i\boldsymbol{H}_i\boldsymbol{\Pi}^{-1/2})}{i\eta^o + \sigma^2_{\min}(\boldsymbol{\Lambda}^{1/2}_i\boldsymbol{H}_i\boldsymbol{\Pi}^{-1/2})} \tag{50}$$

in terms of the extreme singular values of $\boldsymbol{\Lambda}^{1/2}_i\boldsymbol{H}_i\boldsymbol{\Pi}^{-1/2}$. Hence, one can minimize the maximum number of inner iterations $N_{\max}$ based on the evolution of κ_i. In particular, we can set $N_{\max} = 1$ assuming that $\kappa_i \to 1$ quickly. Moreover, since non-smooth regularizers can be estimated according to a reweighted formulation, we may simply set $\boldsymbol{\Pi}_i$ accordingly, e.g., via (25) or (30), in the CG recursions.

B. DECOUPLED RECURSIONS

The computation of the stepsize α_n in (45) requires information from all entries of $\{\boldsymbol{r}_{i,n}, \boldsymbol{p}_{i,n}\}$, and is used in turn to

[1] Of course, mixed ℓ-norms can be considered for the trust-region problems. For example, it may take the more general form

$$\rho(\boldsymbol{w}) = \eta^o\left[(1-\beta)\|\boldsymbol{w}\|^2_{\boldsymbol{\Pi}} \ + \ \beta\|\boldsymbol{w}\|_p\right] \tag{29}$$

where p can be, e.g., the ℓ_0 or ℓ_1 norm.

TABLE 1 Leaky-Reweighted WRLS With Variable Strength η_i via CG Algorithm

```
Initialization:   $\boldsymbol{P}_0'^{-1} = 0$,   set $\eta^o$ from (11)
for $i > 0,\ n = 0$:
        $\boldsymbol{P}_i'^{-1} = \lambda \boldsymbol{P}_{i-1}'^{-1} + \boldsymbol{u}_i^\star \boldsymbol{u}_i$                (37)
        $\boldsymbol{s}_i = \lambda \boldsymbol{s}_{i-1} + \boldsymbol{u}_i^\star d(i)$                (38)
        $\boldsymbol{P}_i^{-1} = \boldsymbol{\Pi}_i + \boldsymbol{P}_i'^{-1}$                (39)
        $\boldsymbol{r}_{i,0} = \boldsymbol{s}_i - \boldsymbol{P}_i^{-1} \boldsymbol{w}_{\eta_i, i-1}$                (40)
        $\boldsymbol{x}_0 = \boldsymbol{w}_{\eta_i, i-1}$                (41)
        $\boldsymbol{p}_{i,0} = \boldsymbol{r}_{i,0}$                (42)
                                                                         (43)
    for $n = 1$ to $N_{\max}$ :
        $\boldsymbol{k}_{i,n-1} = \boldsymbol{P}_i^{-1} \boldsymbol{p}_{i,n-1}$                (44)
        $\alpha_{n-1} = \dfrac{\|\boldsymbol{r}_{i,n-1}\|^2}{\boldsymbol{p}_{i,n-1}^\star \boldsymbol{k}_{i,n-1}}$                (45)
        $\boldsymbol{x}_n = \boldsymbol{x}_{n-1} + \alpha_{n-1} \boldsymbol{p}_{i,n-1}$                (46)
        $\boldsymbol{r}_{i,n} = \boldsymbol{r}_{i,n-1} - \alpha_{n-1} \boldsymbol{k}_{i,n-1}$                (47)
        if $\|\boldsymbol{r}_{i,n}\| < \epsilon$, exit loop
        $\boldsymbol{p}_{i,n} = \boldsymbol{r}_{i,n} + \dfrac{\|\boldsymbol{r}_{i,n}\|^2}{\|\boldsymbol{r}_{i,n-1}\|^2} \boldsymbol{p}_{n-1}$                (48)
    end
                                                                         (49)
    $\boldsymbol{w}_{\eta_i, i} = \boldsymbol{x}_{N_{\max}}$
end
```

update all their entries in (46) and (47). This can be motivated from the fact that $\boldsymbol{P}_i^{-1}$ reflects correlation, either among the individual entries of $\boldsymbol{u}_i$, or through $\boldsymbol{\Pi}$ itself. Without such dependency, each bin $w_i(m)$ could make use of its own step-size, say, $\alpha_{m,n-1}$, $m = 1, \ldots, M-1$ according to individual optimal criteria. More generally, we could have posed the problem in a way that each entry of $\boldsymbol{w}_i$ is updated via $\alpha_{m,n-1}$, computed from a subset of direction vectors corresponding to the correlated entries. This can be achieved, e.g., by replacing (34) by

$$\boldsymbol{D}_m \boldsymbol{P}_i^{-1} \boldsymbol{w}_{\eta_i, i} = \boldsymbol{D}_m \boldsymbol{s}_i \tag{51}$$

where $\boldsymbol{D}_k$ is a diagonal matrix of ones and zeros, defined in terms of the elements of the m-th row of $\boldsymbol{\Pi}$ where correlation exists, denoted by $\boldsymbol{\Pi}_m$. As a result, with the identifications $\boldsymbol{P}_i^{-1} \longleftarrow \boldsymbol{D}_m \boldsymbol{P}_i^{-1}$ and $\boldsymbol{s}_i \longleftarrow \boldsymbol{D}_m \boldsymbol{s}_i$ in (40) relative to the $\alpha_{m,n-1}$, we obtain

$$\alpha_{m,n-1} = \frac{\|\boldsymbol{D}_m \boldsymbol{r}_{i,n-1}\|^2}{\boldsymbol{p}_{i,n-1}^\star \boldsymbol{D}_m \boldsymbol{k}_{i,n-1}}. \tag{52}$$

This is equivalent to subsampling the inner-products that define α_n differently for every entry of α_n:

$$\alpha_{m,n-1} = \frac{\sum_{\ell \in \mathcal{N}_m} |r_{i,n-1}(\ell)|^2}{\sum_{\ell \in \mathcal{N}_m} p_{i,n-1}^\star(\ell) k_{i,n-1}(\ell)}, \quad m = 1, \ldots, M-1. \tag{53}$$

where we denote by $r_{i,n-1}(\ell)$ the ℓ-th entry of $\boldsymbol{r}_{i,n-1}$, and similarly for $\{p_{i,n-1}(\ell), k_{i,n-1}(\ell)\}$. The set $\mathcal{N}_m$ is the set of indexes used in the computation of $\alpha_{m,n-1}$. We can proceed in the same manner regarding the computation of the ratio in (48). The above arguments can be easily adapted to the extended network formulation, as we discuss next.

III. DISTRIBUTED MULTITASK FORMULATIONS

We now return to the extended cost formulation (6) and argue similarly to the above standalone scenario. This will lead to quadratic and non-smooth regularized, distributed counterparts of the CG-based recursions of Table 1.

Assume that our objective is to promote proximity among agents, while optimizing a quadratic risk function. In Part I of our presentation, we argued that this can be motivated as the solution to the following trust-region problem:

$$\min_{\mathcal{W}} \ \mathbb{E}\,\|\boldsymbol{d}_i - \boldsymbol{\mathcal{U}}_i \mathcal{W}\|^2 \quad \text{s.t.} \quad \|\mathcal{W}\|^2_{\mathbf{B}^\star \boldsymbol{\Pi} \mathbf{B}} \leq \nu \tag{54}$$

where $\boldsymbol{d}_i$ and $\boldsymbol{\mathcal{U}}_i$ are defined in (2)–(3). This is the network analogue of (8). Here, however, the role of $\boldsymbol{\Pi}$ in (8) is replaced by $\mathbf{B}^\star \boldsymbol{\Pi} \mathbf{B}$, where $\boldsymbol{\Pi}$ is $NM \times NM$, of the form

$$\boldsymbol{\Pi} = \text{bdiag}\,\{\boldsymbol{\Pi}_1, \boldsymbol{\Pi}_2, \ldots, \boldsymbol{\Pi}_N\} \tag{55}$$

and $\mathbf{B}$ is an $NM \times NM$ combination matrix satisfying $\mathbf{B}\mathcal{W} = \mathbf{0}$, $\ell \notin \mathcal{N}_k$. For example, we can select $\mathbf{B} = \boldsymbol{\mathcal{L}} = \boldsymbol{L} \otimes \boldsymbol{I}_N$ in terms of the Laplacian rule $\boldsymbol{L}$. This problem is equivalent to

$$\min_{\mathcal{W}} \ \mathbb{E}\,\|\boldsymbol{d}_i - \boldsymbol{\mathcal{U}}_i \mathcal{W}\|^2 + \eta^o \|\mathcal{W}\|^2_{\mathbf{B}^\star \boldsymbol{\Pi} \mathbf{B}} \tag{56}$$

whose leaky solution is given by

$$\mathcal{W}^{\eta^o} = (\eta^o \mathbf{B}^\star \boldsymbol{\Pi} \mathbf{B} + \boldsymbol{R}_{\mathcal{U}})^{-1} \boldsymbol{R}_{d\mathcal{U}}. \tag{57}$$

Instantaneous gradient approximations lead to a network leaky-LMS (N-LLMS) algorithm

$$\mathcal{W}_i^{\eta^o} = (\boldsymbol{I} - \mu \eta^o \mathbf{B}^\star \boldsymbol{\Pi} \mathbf{B}) \mathcal{W}_{i-1}^{\eta^o} + \mu \boldsymbol{\mathcal{U}}_i^\star (\boldsymbol{d}_i - \boldsymbol{\mathcal{U}}_i \mathcal{W}_{i-1}^{\eta^o}) \tag{58}$$

that executes a multitask recursion. Now, define

$$\boldsymbol{H}_{k,i} = \text{col}\{\boldsymbol{u}_{k,1}, \boldsymbol{u}_{k,2}, \ldots, \boldsymbol{u}_{k,i}\} \tag{59}$$

$$\boldsymbol{y}_{k,i} = \text{col}\{d_k(1), d_k(2), \ldots, d_k(i)\} \tag{60}$$

$$\boldsymbol{y}_i = \text{col}\,\{\boldsymbol{y}_{1,i}, \boldsymbol{y}_{2,i}, \ldots, \boldsymbol{y}_{N,i}\} \tag{61}$$

$$\boldsymbol{\mathcal{H}}_i = \text{bdiag}\left\{\boldsymbol{H}_{1,i}, \boldsymbol{H}_{2,i}, \ldots, \boldsymbol{H}_{N,i}\right\} \tag{62}$$

$$\boldsymbol{\Lambda}_i = \text{bdiag}\left\{\boldsymbol{\Lambda}_{1,i}, \boldsymbol{\Lambda}_{2,i}, \ldots, \boldsymbol{\Lambda}_{N,i}\right\} \tag{63}$$

Following the same reasoning carried out in the standalone scenario, for large i, the first term in (50) can be approximated by $\|\boldsymbol{y}_i - \boldsymbol{\mathcal{H}}_i \mathcal{W}\|^2_{\boldsymbol{\Lambda}_i}/i$, so that one may solve instead

$$\min_{\mathcal{W}} \frac{1}{i} \left\|\boldsymbol{y}_i - \boldsymbol{\mathcal{H}}_i \mathcal{W}\right\|^2_{\boldsymbol{\Lambda}_i} \quad \text{s.t.} \quad \|\mathcal{W}\|^2_{\mathbf{B}^\star \boldsymbol{\Pi} \mathbf{B}} \leq \nu \tag{64}$$

or, equivalently,

$$\min_{\mathcal{W}} \left\{ \left\|\boldsymbol{y}_i - \boldsymbol{\mathcal{H}}_i \mathcal{W}\right\|^2_{\boldsymbol{\Lambda}_i} + \eta_i \|\mathcal{W}\|^2_{\mathbf{B}^\star \boldsymbol{\Pi} \mathbf{B}} \right\} \tag{65}$$

with $\nu = \|\mathcal{W}_{\eta_i', i}\|^2_{\mathbf{B}^\star \boldsymbol{\Pi} \mathbf{B}}$, and where $\eta_i \overset{\Delta}{=} i\eta_i'$. Again, this suggests that we can set $\eta_i^o \approx i\eta^o$, such the solution to (59) is given by

$$\boldsymbol{\mathcal{P}}_i = (\eta_i^o \mathbf{B}^\star \boldsymbol{\Pi} \mathbf{B} + \boldsymbol{\mathcal{H}}_i^\star \boldsymbol{\Lambda}_i \boldsymbol{\mathcal{H}}_i)^{-1} \tag{66}$$

$$\mathcal{W}_{\eta_i^o, i} = \boldsymbol{\mathcal{P}}_i \boldsymbol{\mathcal{H}}_i^\star \boldsymbol{\Lambda}_i \boldsymbol{y}_i. \tag{67}$$

For the sake of generality we can formulate ℓ_0 and ℓ_1 network problems just like in (21), (26), and (28), by resorting to their extended formulation, given some mixed-norm definition. Note that the goal in (48) is to shrink the ℓ_2-block norm of the extended vector $\mathcal{W}$ by promoting closeness among its vector entries in the quadratic sense. One can similarly promote proximity by relying on a weighted ℓ_p-block norm constraint, and more generally, starting from the already ℓ_2-norm constrained problem (50), by adding another layer of an ℓ_p-block norm constraint:

$$\min_{\mathcal{W}} \mathbb{E}\,\|\boldsymbol{d}_i - \boldsymbol{\mathcal{U}}_i\mathcal{W}\|^2 \;+\; \eta^o\|\mathbf{B}\mathcal{W}\|^2_{\mathbf{\Pi}} \;\text{ s.t. }\; \|\mathbf{Q}\mathcal{W}\|_{g,p,\mathbf{\Pi}} < \zeta \tag{68}$$

where we define the group norm $\|\mathcal{W}\|_{g,p}$ in terms of the weighted ℓ_2-norm of the vector entries $\boldsymbol{w}_k$, via the weighting matrix $\mathbf{\Pi}$. This can also be useful to promote spatial sparseness of the agents across the network in a non-cooperative mode of operation by setting $\mathbf{B} = \boldsymbol{I}$, prior to posing any cooperative formulation. After recognizing the equivalence to the unconstrained formulation, a reweighted LS solution would then follow by selecting an adaptive block diagonal weigthing matrix as

$$\mathbf{\Delta}_{2-p,i} = \begin{bmatrix} \dfrac{\mathbf{\Pi}_1}{\|\mathbf{Q}_1\mathcal{W}_{i-1}\|^{2-p}+\delta} & & \\ & \ddots & \\ & & \dfrac{\mathbf{\Pi}_M}{\|\mathbf{Q}_N\mathcal{W}_{i-1}\|^{2-p}+\delta} \end{bmatrix} \tag{69}$$

where we denote by $\mathbf{Q}_k$ the k-th $M \times NM$ block row of $\mathbf{Q}$. Similarly to (61), this results in

$$\mathcal{P}_i = (\eta_i^o\mathbf{B}^\star\mathbf{\Pi}\mathbf{B} + \tau_i^o\mathbf{\Delta}_{2-p,i} + \mathcal{H}_i^\star\mathbf{\Lambda}_i\mathcal{H}_i)^{-1} \tag{70}$$

$$\mathcal{W}_{\eta_i^o,\tau_i^o,i} = \mathcal{P}_i\mathcal{H}_i^\star\mathbf{\Lambda}_i\boldsymbol{y}_i \tag{71}$$

where τ_i^o satisfies $\|\mathcal{W}_{\eta_i^o,\tau_i^o,i}\|_{g,p} = \zeta$. Again, we can set $\tau_i^o = \tau_i = i\tau^o$, which approaches the optimal value when $i \to \infty$.

At this point, we can simply map the standalone definitions of the previous section onto the extended quantities given in (1)–(4) and (59)–(63), say, $\boldsymbol{P}_i \longrightarrow \mathcal{P}_i$, $\boldsymbol{s}_i \longrightarrow \mathcal{S}_i$, and etc. For example, without a constraint, recursions for $\boldsymbol{r}_{i,0}$ and $\boldsymbol{k}_{i,n-1}$ in Table 1 become

$$\boldsymbol{r}_{k,i,0} = \boldsymbol{s}_{k,i} - \boldsymbol{P}_{k,i}^{-1}\boldsymbol{w}_{k,i-1} - i\eta^o\sum_{\ell\in\mathcal{N}_k}\mathbf{T}_{k\ell,i}\boldsymbol{w}_{\ell,i-1} \tag{72}$$

$$\boldsymbol{k}_{k,i,n-1} = \boldsymbol{P}_{k,i}^{-1}\boldsymbol{p}_{k,i,n-1} + i\eta^o\sum_{\ell\in\mathcal{N}_k}\mathbf{T}_{k\ell,i}\boldsymbol{p}_{\ell,i,n-1} \tag{73}$$

where we have denoted by $\mathbf{T}_{k\ell,i}$ the (k,ℓ)-block element of $\mathbf{T}_i = \mathbf{B}^\star\mathbf{\Delta}_{2-p,i}\mathbf{B}$, with $\mathbf{\Delta}_{2-p,i}$ given by (63). More generally, Table 2 lists the node recursions obtained after decoupling node variables from the extended definitions, via independent stepsizes $\alpha_{k,n-1}$ similarly to (47). We refer to these recursions as diffusion CG *Reweighted Leaky-RLS* (CG-RLRLS). Observe that we can drop edges in $\mathbf{B}$ such that the product with $\mathbf{T}_i$ is simplified, and given by a single matrix multiplication with an extended quantity. That is, consider the quadratic case, and let $\mathbf{\Pi} = \boldsymbol{I}$. Note that the product of $\mathbf{B}^\star\mathbf{B}$ with any extended quantity is realized in a distributed way. For example, we can set $\mathbf{B}^\star\mathbf{B} = \boldsymbol{\mathcal{L}}$, where $\boldsymbol{\mathcal{L}}$ is the Laplacian matrix, with entries denoted by $\boldsymbol{\mathcal{L}}_{k\ell}$.

TABLE 2 Diffusion CG-Based Reweighted Leaky-RLS

```
Initialization:   $P'^{-1}_{k,0} = 0,\ \boldsymbol{w}_{k,0} = \mathbf{0}$,   set $\{\eta^o, \tau^o\}$ from (54) and (68)
for $i > 0,\ n = 0$,
for $k = 1$ to $N$,
  $P'^{-1}_{k,i} = \lambda P'^{-1}_{k,i-1} + \boldsymbol{u}^\star_{k,i}\boldsymbol{u}_{k,i}$
  $\boldsymbol{s}_{k,i} = \lambda\boldsymbol{s}_{k,i-1} + \boldsymbol{u}^\star_{k,i}d_k(i)$
  $\boldsymbol{r}_{k,i,0} = \boldsymbol{s}_{k,i} - \boldsymbol{P}^{-1}_{k,i}\boldsymbol{w}_{k,i-1} - \dfrac{i\tau^o\boldsymbol{w}_{k,i-1}}{\|\boldsymbol{w}_{k,i-1}\|^{2-p}+\delta} - i\eta^o\sum_{\ell\in\mathcal{N}_k}\boldsymbol{\mathcal{L}}_{k\ell}\boldsymbol{w}_{\ell,i-1}$
  $\boldsymbol{x}_{k,0} = \boldsymbol{w}_{k,i-1}$
  $\boldsymbol{p}_{k,i,0} = \boldsymbol{r}_{k,i,0}$
  for $n = 1$ to $N_{\max}$ :
    $\boldsymbol{k}_{k,i,n-1} = \boldsymbol{P}^{-1}_{k,i}\boldsymbol{p}_{k,i,n-1} + \dfrac{i\tau^o\boldsymbol{p}_{k,i,n-1}}{\|\boldsymbol{w}_{k,i-1}\|^{2-p}+\delta} + i\eta^o\sum_{\ell\in\mathcal{N}_k}\boldsymbol{\mathcal{L}}_{k\ell}\boldsymbol{p}_{\ell,i,n-1}$
    $\alpha_{k,n-1} = \dfrac{\sum_{\ell\in\mathcal{N}_k}\|\boldsymbol{r}_{\ell,i,n-1}\|^2}{\sum_{\ell\in\mathcal{N}_k}\boldsymbol{p}^\star_{\ell,i,n-1}\boldsymbol{k}_{\ell,i,n-1}}$
    $\boldsymbol{x}_{k,n} = \boldsymbol{x}_{k,n-1} + \alpha_{k,n-1}\boldsymbol{p}_{k,i,n-1}$
    $\boldsymbol{r}_{k,i,n} = \boldsymbol{r}_{k,i,n-1} - \alpha_{k,n-1}\boldsymbol{k}_{k,i,n-1}$
    $\boldsymbol{p}_{k,i,n} = \boldsymbol{r}_{k,i,n} + \dfrac{\|\boldsymbol{r}_{k,i,n}\|^2}{\|\boldsymbol{r}_{k,i,n-1}\|^2}\boldsymbol{p}_{k,n-1}$
  end
  $\boldsymbol{w}_{k,i} = \boldsymbol{x}_{k,N_{\max}}$
end
```

IV. SINGLE-TASK SOLUTIONS

In theory, a single-task solution can be attained from the above multitask formulation by setting the ridge strength to $\eta^o \to \infty$. In practice, however, this is not possible either by using (58) or (61), since at one point the increase of η^o will lead to numerical difficulties (see Fig. 5 in our simulations). One possible solution to this problem is to rely on a reweighted regularization as proposed in Part I of this work, where the regularization strength grows adaptively. A second approach, is to resort to the exact solutions to the constrained LMS or LS problems, by making use of the above distributed CG recursions, as we explain next.

A. CONSTRAINED CG-RLS

Recall from Part I, that in the deterministic case, we posed the exact single-task problem as

$$\min_{\mathcal{W}} \left\|\boldsymbol{y}_i - \mathcal{H}_i\mathcal{W}\right\|^2_{\mathbf{\Lambda}_i} \qquad \text{s.t.} \quad \mathbf{C}\mathcal{W} = \mathbf{0} \tag{74}$$

where $\mathbf{C}$ is a fat constraint matrix which enforces consensus among the individual agents. It is obtained by removing one redundant block row from $\mathbf{B}$. The solution, restated here for convenience, is given by

$$\mathcal{W}_i^c = \widehat{\mathcal{W}}_i - \widehat{\mathcal{P}}_i\mathbf{C}^\star(\mathbf{C}\widehat{\mathcal{P}}_i\mathbf{C}^\star)^{-1}\mathbf{C}\widehat{\mathcal{W}}_i \tag{75}$$

in terms of the local LS solutions

$$\widehat{\boldsymbol{w}}_{k,i} = \widehat{\boldsymbol{P}}_{k,i}\boldsymbol{H}^\star_{k,i}\mathbf{\Lambda}_{k,i}\boldsymbol{y}_{k,i}, \quad \text{with} \tag{76}$$

$$\widehat{\boldsymbol{P}}_{k,i} = (\epsilon\boldsymbol{I} + \boldsymbol{H}^\star_{k,i}\mathbf{\Lambda}_{k,i}\boldsymbol{H}_{k,i})^{-1}. \tag{77}$$

The form of (69) readily suggests that it can be written as

$$\mathcal{w}_i^c = \widehat{\mathcal{w}}_i - \widehat{\mathcal{P}}_i \mathbf{C}^\star \mathcal{Z}_i \tag{78}$$

where

$$\mathcal{Z}_i = \text{col}\{z_{1,i}, z_{2,i}, \ldots, z_{N,i}\} \tag{79}$$

is the solution to the linear system

$$(\mathbf{C}\widehat{\mathcal{P}}_i\mathbf{C}^\star)\mathcal{Z}_i = \mathbf{C}\widehat{\mathcal{w}}_i \tag{80}$$

and which can be obtained indirectly via CG recursions. Let $\boldsymbol{C}_{k\ell}$ denote the (k, ℓ) entry of $\mathbf{C}$. Note that assuming $N_{\max} = 1$, and using $\mathcal{Z}_{i-1}$ as the initial guess at the begining of every CG iteration, we can eliminate the recursion for the residue $\boldsymbol{r}_{k,i,n}$. The new distributed single-task algorithm is referred to as a network *constrained CG-based RLS* (C-CG-RLS), and listed in Table 3.

B. CONSTRAINED CG-LMS

The above reasoning can be similarly adopted in the solution of a linearly constrained LMS problem:

$$\min_{\mathcal{w}} \; \mathbb{E}\,\|\boldsymbol{d}_i - \boldsymbol{\mathcal{U}}_i\mathcal{w}\|^2 \qquad \text{s.t.} \quad \mathbf{C}\mathcal{w} = \mathbf{0} \tag{81}$$

which leads to the adaptive extended recursion

$$\mathcal{w}_i^u = \mathcal{w}_{i-1}^{cl} + \mu\boldsymbol{\mathcal{U}}_i^\star\boldsymbol{e}_i \tag{82}$$

$$\mathcal{w}_i^{cl} = \mathbf{A}_i\mathcal{w}_i^u \tag{83}$$

where

$$\mathbf{A}_i = \boldsymbol{I} - \mathbf{C}^\star(\mathbf{C}\mathbf{C}^\star)^{-1}\mathbf{C} \tag{84}$$

for a step-size μ, where $\boldsymbol{e}_i = \boldsymbol{d}_i - \boldsymbol{\mathcal{U}}_i\mathcal{w}_{i-1}^{cl}$, and $\mathcal{w}_i^u$ is the unconstrained LMS solution. We arrive at a distributed implementation by replacing (77) by

$$\mathcal{w}_i^{cl} = \mathcal{w}_i^u \; + \; \mathbf{C}^\star\mathcal{Z}_i \tag{85}$$

in terms of the solution to the linear system

$$\mathbf{C}\mathbf{C}^\star\mathcal{Z}_i = \mathbf{C}\mathcal{w}_i^u \tag{86}$$

via CG recursions. Table 4 lists the resulting algorithm, for $N_{\max} = 1$, and using $\mathcal{Z}_{i-1}$ as the initial guess at the begining of every CG iteration. The vector entries of $\mathcal{w}_i^{cl}$ are denoted by $\boldsymbol{w}_{k,i}$. We refer to it as a network linearly constrained CG-based LMS (C-CG-LMS).

A special case of (62) is obtained by setting $\mathbf{B} = \mathbf{Q}$ and by replacing the inequality constraint by $\|\mathbf{B}\mathcal{w}\|_{g,p,\Pi} = 0$, in which case the main unregularized risk is not altered. This formulation led to the so-called EXTRA algorithm proposed in [25], albeit derived as a gradient descent recursion from general costs, and with $\mathbf{B}$ replaced by the symmetric square-root $\mathcal{V}^{1/2}$ of the matrix $c(\boldsymbol{I}_{NM} - \mathcal{B}^*\mathcal{B}) = \mathcal{V}^{\star/2}\mathcal{V}^{1/2}$, for some scalar c, and with $\mathcal{B} = \boldsymbol{B} \otimes \boldsymbol{I}_M$, for an $N \times N$ combination matrix $\boldsymbol{B}$. As a result, for small step sizes, the algorithm derived in this manner seeks the exact constrained solution.

In contrast, the algorithms proposed here rely on exact constrained adaptive implementations, whose challenge in the extended network scenario lies instead on the realization of the involved inverses. For example, observe that $\mathbf{A}_i$ in (78) is a combination matrix whose structure extrapolates the network node distribution. That is, it can be easily verified, using the contraint $\mathbf{C}\mathcal{w} = \mathbf{0}$ along with the matrix inversion lemma, that

$$\mathbf{A}_i = \frac{1}{N}\mathbb{1}\mathbb{1}^{\mathsf{T}} \tag{87}$$

which shows that effectively, each node takes the average of estimates of *all* agents of the network, and not only locally. The social learning in this case corresponds to two consecutive combination operations necessary for the computation of $\{\boldsymbol{p}_{k,i}, \boldsymbol{k}_{k,i}\}$ in Table 4. The same two-step operation holds in the RLS case, in Table 3.

TABLE 3 **Network Linearly-Constrained CG-Based RLS**

```
Initialization:   $\widehat{\boldsymbol{P}}_{k,0} = \epsilon^{-1}\boldsymbol{I},\ \widehat{\boldsymbol{w}}_{k,0} = \mathbf{0},\ \boldsymbol{z}_{k,0} = \mathbf{0}$
for $i > 0$, $n = 0$,
for $k = 1$ to $N$,
    $\gamma_k(i) = (1 + \lambda^{-1}\boldsymbol{u}_{k,i}\widehat{\boldsymbol{P}}_{k,i-1}\boldsymbol{u}_{k,i}^\star)^{-1}$
    $\boldsymbol{g}_{k,i} = \lambda^{-1}\widehat{\boldsymbol{P}}_{k,i-1}\boldsymbol{u}_{k,i}^\star\gamma_k(i)$
    $\widehat{\boldsymbol{P}}_{k,i} = \lambda^{-1}\widehat{\boldsymbol{P}}_{k,i-1} - \boldsymbol{g}_{k,i}\boldsymbol{g}_{k,i}^\star/\gamma_k(i)$
    $\widehat{\boldsymbol{w}}_{k,i} = \widehat{\boldsymbol{w}}_{k,i-1} + \boldsymbol{g}_{k,i}[d_k(i) - \boldsymbol{u}_{k,i-1}\widehat{\boldsymbol{w}}_{k,i-1}]$
    $\boldsymbol{p}_{k,i} = \sum_{m\in\mathcal{N}_k}\boldsymbol{C}_{km}\left(\widehat{\boldsymbol{w}}_{k,i} - \widehat{\boldsymbol{P}}_{m,i}\sum_{\ell\in\mathcal{N}_m}\boldsymbol{C}_{m\ell}^\star\boldsymbol{z}_{\ell,i-1}\right)$
    $\boldsymbol{k}_{k,i} = \sum_{m\in\mathcal{N}_k}\boldsymbol{C}_{km}\widehat{\boldsymbol{P}}_{m,i}\sum_{\ell\in\mathcal{N}_m}\boldsymbol{C}_{m\ell}^\star\boldsymbol{p}_{k,i}$
    $\alpha_k = \dfrac{\sum_{\ell\in\mathcal{N}_k}\|\boldsymbol{p}_{\ell,i}\|^2}{\sum_{\ell\in\mathcal{N}_k}\boldsymbol{p}_{\ell,i}^\star\boldsymbol{k}_{\ell,i}}$
    $\boldsymbol{z}_{k,i} = \boldsymbol{z}_{k,i-1} + \alpha_k\boldsymbol{p}_{k,i}$
    $\boldsymbol{w}_{k,i}^c = \widehat{\boldsymbol{w}}_{k,i} - \widehat{\boldsymbol{P}}_{k,i}\sum_{\ell\in\mathcal{N}_k}\boldsymbol{C}_{k\ell}\boldsymbol{z}_{\ell,i}$
end
```

TABLE 4 **Network Linearly-Constrained CG-Based LMS**

```
Initialization:   $\boldsymbol{w}_{k,0} = \mathbf{0},\ \boldsymbol{z}_{k,0} = \mathbf{0}$
for $i > 0$, $n = 0$,
for $k = 1$ to $N$,
    $\boldsymbol{w}_{k,i}^u = \boldsymbol{w}_{k,i-1} + \mu\boldsymbol{u}_{k,i}^\star e_k(i)$
    $\boldsymbol{p}_{k,i} = \sum_{m\in\mathcal{N}_k}\boldsymbol{C}_{km}\left(\boldsymbol{w}_{m,i}^u - \sum_{\ell\in\mathcal{N}_m}\boldsymbol{C}_{m\ell}^\star\boldsymbol{z}_{\ell,i-1}\right)$
    $\boldsymbol{k}_{k,i} = \sum_{m\in\mathcal{N}_k}\boldsymbol{C}_{km}\sum_{\ell\in\mathcal{N}_m}\boldsymbol{C}_{m\ell}^\star\boldsymbol{p}_{k,i}$
    $\alpha_k = \dfrac{\sum_{\ell\in\mathcal{N}_k}\|\boldsymbol{p}_{\ell,i}\|^2}{\sum_{\ell\in\mathcal{N}_k}\boldsymbol{p}_{\ell,i}^\star\boldsymbol{k}_{\ell,i}}$
    $\boldsymbol{z}_{k,i} = \boldsymbol{z}_{k,i-1} + \alpha_k\boldsymbol{p}_{k,i}$
    $\boldsymbol{w}_{k,i} = \boldsymbol{w}_{k,i}^u + \sum_{\ell\in\mathcal{N}_k}\boldsymbol{C}_{k\ell}^\star\boldsymbol{z}_{\ell,i}$
end
```

V. COMPUTATIONAL COMPLEXITY

Table 5 lists the approximate order of complexity per agent in terms of the operations involved in each of the proposed algorithms in this work. For the sake of comparison, we also include the complexity of the reweighted, single-task algorithms derived in Part I.

TABLE 5 Computational Complexity of All Recursions

Distributed Algorithms	Order of Complexity/node
AAF	$\mathcal{O}(M^2)$
NLMS-RV	$\mathcal{O}(\mathcal{N}M)$
General Diffusion RLS	$\mathcal{O}(M^2)$
SAAF	$\mathcal{O}(M)$
MDF-NLMS-RV	$\mathcal{O}(\mathcal{N}M)/L$
MDF-SAAF	$\mathcal{O}(M)/L$
MDF-SAAF-US	$\mathcal{O}(M)/L$
N-LLMS	$\mathcal{O}(\mathcal{N}M)$
CG-LRLS	$\mathcal{O}(M^2)$
C-CG-RLS	$\mathcal{O}(M^2)$
C-CG-LMS	$\mathcal{O}(\mathcal{N}M)$

We highlight that the constructions considered herein imply very low complex solutions to the single-task problems when the constraint matrix is limited to its simplest form. That is, consider for example the network of Fig. 2, and the corresponding constraint with $\gamma = 1$, given by (23) in Part I of this paper. This corresponds to the $M(N-1) \times NM$ matrix $\mathbf{C} = \boldsymbol{C} \otimes \boldsymbol{I}_M$ where each row of $\boldsymbol{C}$ has at most 2 entries, either 1 ou -1. This results in very simple RLS and LMS network recursions, where only two agents are combined in $\mathcal{N}_k$ via addition. We shall see in our experiments, that depending on the level of correlation of the streaming data, the performance of the algorithm in this case can be quite similar to the one where cooperation takes place among all available agents at each neighborhood (see Fig. 5). We also bring attention to the fact that the performance of the constrained LMS algorithm (C-CG-LMS) proposed here outperforms the one of the NLMS-RV, which instead makes use of a time-varying, optimized combination matrix.

Also, the above arguments can also be extended to a more general network affine projection algorithm (APA). The APA in this context is given by

$$\boldsymbol{w}_i = \mathbf{A}_i \left[\boldsymbol{w}_{i-1} + \mu \bar{\mathcal{H}}_i^{\star} \left(\epsilon \boldsymbol{I} + \bar{\mathcal{H}}_i \mathbf{A}_i \bar{\mathcal{H}}_i^{\star} \right)^{-1} \boldsymbol{e}_i \right] \tag{88}$$

where $\bar{\mathcal{H}}_i = \text{bdiag}\{\overline{\boldsymbol{H}}_{1,i}, \overline{\boldsymbol{H}}_{2,i}, \ldots, \overline{\boldsymbol{H}}_{N,i}\}$ is in terms of the data matrix $\overline{\boldsymbol{H}}_{k,i} = \text{col}\{\boldsymbol{u}_{k,i-L+1}, \boldsymbol{u}_{k,i-L+2}, \ldots, \boldsymbol{u}_{k,i}\}$. Thus, one can envision a CG based algorithm that will deal with the innermost inverses in (82) as well, obviously, with additional computation effort. Moreover, further gains in complexity and performance can be achieved for tap-delay-line models, when implemented in the subband domain, using a reasoning similar to the one followed in Part I.

VI. SIMULATIONS

We assess the performance of the proposed recursions in multitask and single tasks settings, considering the same network of $N = 20$ agents in the experiments of Part I, illustrated in Fig. 2. The extended vector parameter $\mathcal{w}^o$ is generated as a Gaussian vector sequence, and with the same power profiles $\sigma_{u,k}^2 \boldsymbol{I}$ and $\sigma_{v,k}^2 \boldsymbol{I}$ for its $M = 10$-length entries considered before. We set the forgetting factor λ to one in all scenarios and consider adaptation initially without preconditioning.

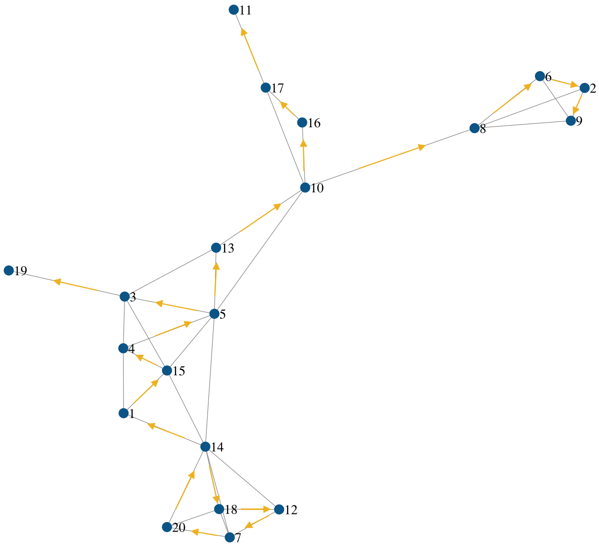


FIGURE 2. Network of $N = 20$ agents.

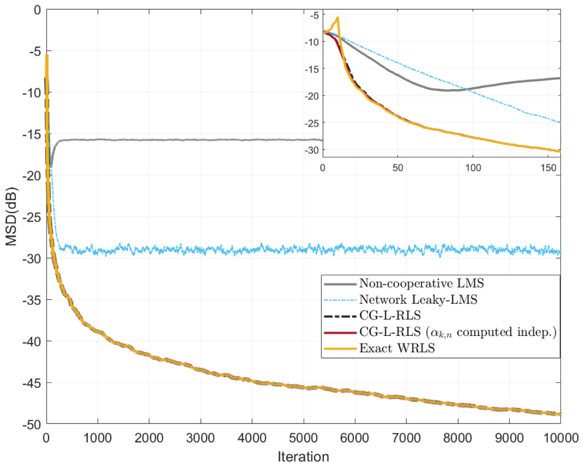


FIGURE 3. MSD for the exact LRLS and the corresponding CG recursions.

♦ ***Scenario 1 (Multitask scenario under quadratic regularization):*** We illustrate the evolution of the ensemble-average learning curves according to the network counterpart of (83),

$$\text{MSD} = \frac{1}{N} \mathbb{E} \left\| \mathcal{w}_{\eta,i} - \mathcal{w}_{\eta^o} \right\|^2 \tag{89}$$

where $\mathcal{w}_{\eta^o}$ is the solution to (48), given by (51), with $\eta^o = 10$ and white noise inputs. The performance is compared in Fig. 3 among the (i) the non-cooperative NLMS-based algorithm; (ii) the network leaky-LMS algorithm, and (iii) the network CG-L-RLS considering two forms of computation of the step-size $\alpha_{k,n}$, namely, (47) and by using the information of node k alone, in the context of the extended network formulation. The exact WLS algorithm is also illustrated for comparison, and appear on top of the latter two. We perceive a similar performance in these cases regardless of how the step-size $\alpha_{k,n}$ is computed.

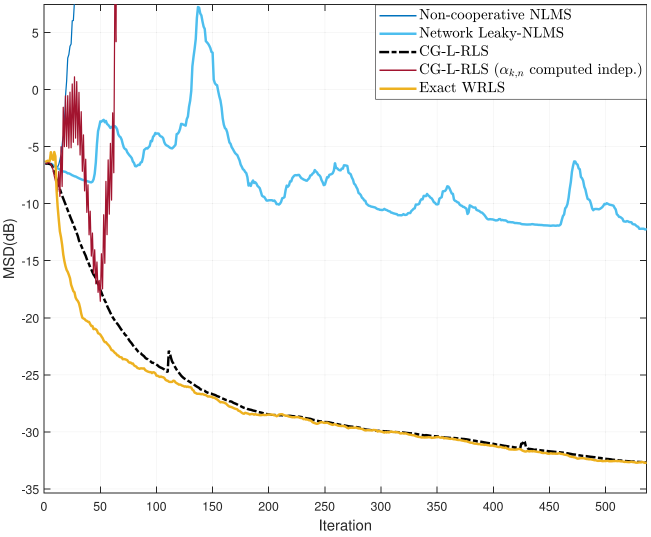


FIGURE 4. MSD for the exact LRLS and the corresponding CG recursions.

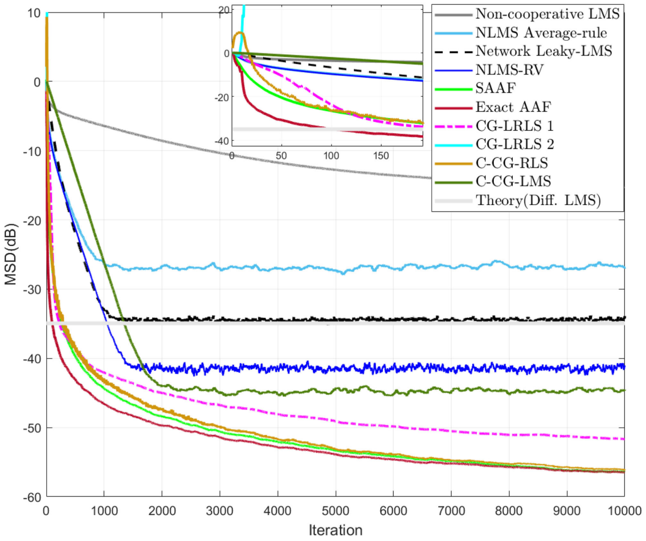


FIGURE 5. MSD for the various single-task algorithms, white noise input signals.

To highlight the importance of relying on correlated entries, we reassess the performance considering a less uniform input power profile, say, for the set of input power nodes $\sigma_u^2 \in \{-12\ \ -10.4\ \ -12.5\ \ 11\ \ 10\ \ 12.6\ \ 22.4\ \ 12.3\ \ 12.4\ \ 11.5\ \ 11.7\ \ 1.3\ \ -12.6\ \ 12.5\ \ 12.5\ \ 10.3\ \ 11.5\ \ 12.1\ \ -10.5\ \ 12.2\}$ dB. As depicted in Fig. 4, we see that while the algorithm employing the extended counterpart of (47) diverges, the proposed CG that computes $\alpha_{k,n}$ via cooperation still follows the exact L-RLS learning curve.

♦ ***Scenario 2 (Single task scenario under quadratic regularization):*** The goal in this experiment is to contrast the performance of the proposed constrained LMS and leaky-LS CG-based recursions with existing algorithms, in particular, with the reweighted regularization methods developed in Part I. We shall consider two scenarios, namely, under white noise input signals for the agents, and arbitrarily correlated signals, as specified in the first experiment of Part I, and for the same input power profile of the previous experiment.

1) WHITE INPUTS

Fig. 5 depicts the MSD evolution of all algorithms which are fairly smooth. For the leaky-type recursions, we have set the ridge strength to $\eta_o = 100$. At this level, the network leaky-LMS approximately reaches the theoretical bound of the diffusion LMS. Further increase in η_o will eventually lead to numerical difficulties. In order to attain the corresponding strength $\eta_o = \infty$, we implement the network linear constrained LMS via CG recursions of Table 4 with $\mathbf{C}$ given by (23) defined in Part I of this paper. Note that the accuracy of the proposed algorithm surpasses the one of the NLMS-RV, considering that their rate of convergence is similar during most of the initial iterations. Further improvement over these recursions is obtained via the CG leaky recursions, labeled as CG-LRLS 1 and CG-LRLS 2. to illustrate the necessity of cooperation in determining the step sizes $\alpha_{k,n}$, the latter was implemented assuming that these are computed without information from the neighbors. The zoomed picture shows divergence in this case, unlike the former, which makes use of full cooperation. Again, further increase in regularization leads to instability. The limit $\eta_o = \infty$ is reached via the C-CG-RLS algorithm, which is superior to the previous recursions. Still, the AAF algorithm developed in Part I is superior, in both exact and simplified forms (SAAF).

2) HIGHLY CORRELATED INPUTS

We consider again a rather hash scenario of highly correlated input sources, ranging from small pole first order AR processes, to music signals, specified from node $k = 1$ to $k = 20$ via the set

$$\{\mathfrak{S},\ \mathfrak{S},\ \mathfrak{S},\ \mathfrak{S}_{f_1},\ \mathfrak{S}_{f_2},\ \mathfrak{S}_{f_3},0.1,\ -0.2,\ \mathfrak{S},\ \mathfrak{S},\ \mathfrak{S}$$

$$0.3,\ 0.05,\ 0.999,\ -0.999,\ 0.1,\ -0.3,\ 0.1,\ -0.3,\ 0.999\}$$

where $\mathfrak{S}$ refers to a music segment obtained from the Matlab example file “hade” (a 9s segment from the Hallelujah Chorus of Handel’s Messiah), $\mathfrak{S}_{f_m}$ are filtered versions of it through AR filters with poles at 0.999, 0.3, and 0.5, respectively and the remaining values refer to poles r_k of AR processes generated from uniformly distributed sources. Identical music segments are used in six nodes, which renders these signals correlated in both time and space.

For comparison, we have included the multidelay adaptive filter based relative variance and simplified AAF algorithms, namely, MDF-NLMS-RV and MDF-SAAF, proposed in Part I, for models of length $M = 16$ and transmitted block sized $L = 16$. The CG-LRLS 1 can be contrasted with its exact non-distributed, leaky-RLS solution, which is followed closely by the former in the long run. The superiority of the AAF and SAAF algorithms is clear. Note, however, that in this scenario, the CG implementation of the constrained RLS algorithm

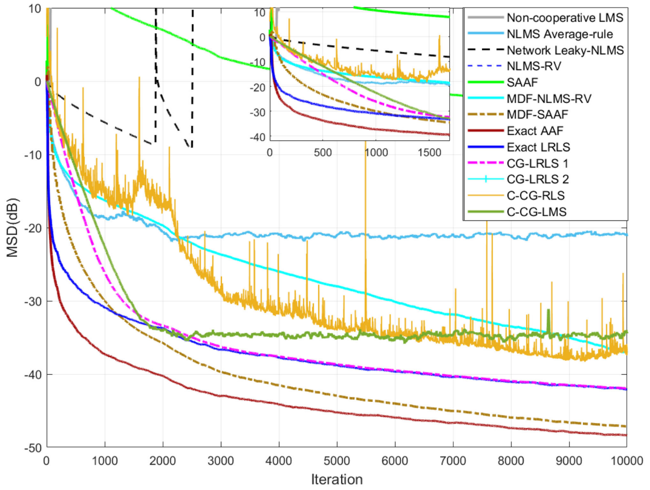


FIGURE 6. MSD for the various single-task algorithms, arbitrary correlated input signals.

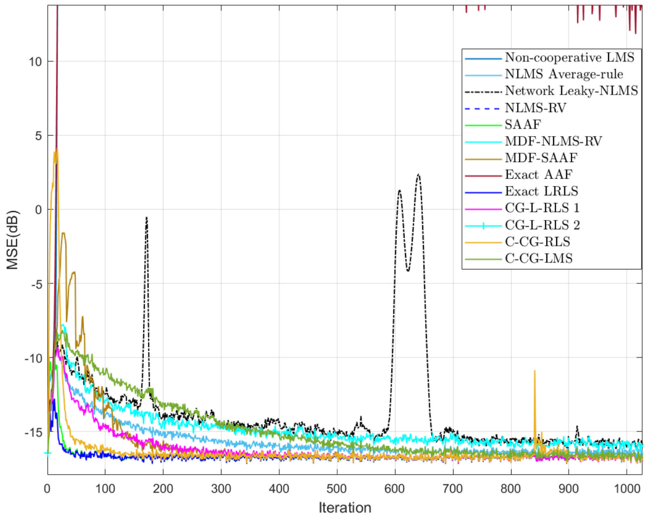


FIGURE 7. MSE for the various single-task algorithms, arbitrary correlated input signals.

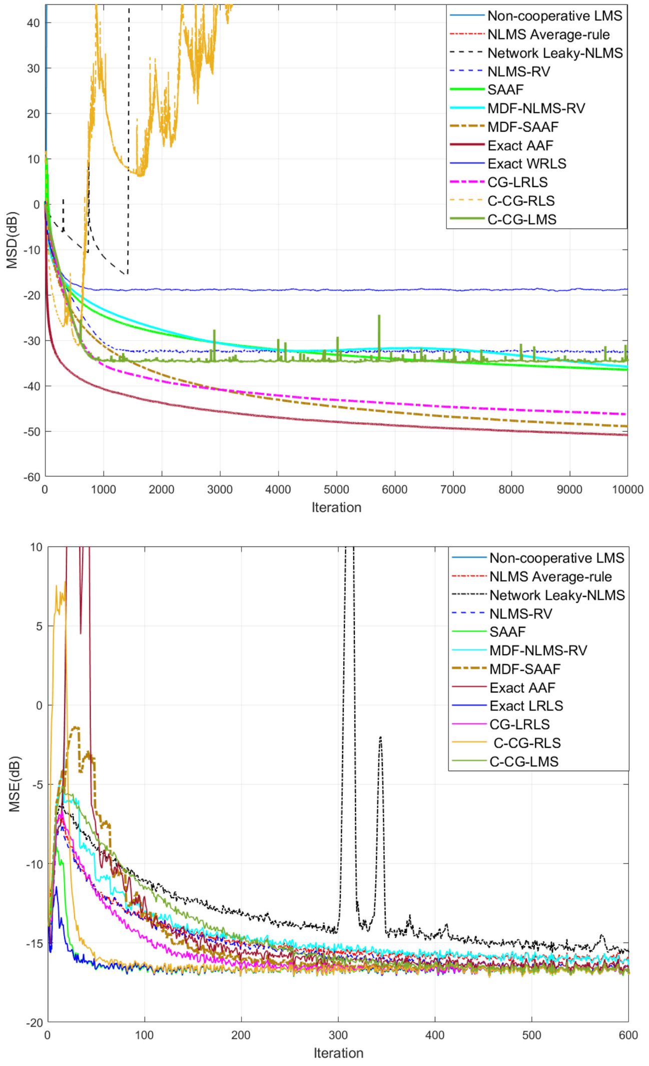


FIGURE 8. (Top) MSD and (bottom) MSE for the various single-task algorithms, considering only AR input signals.

becomes more sensitive to the various spatially correlated signals. This was not the case when independent sources are used by the agents.

3) MSE PERFORMANCE

Of course there exists a natural tradeoff between MSE and MSD performance for the various algorithms considered in this paper. We first illustrate this fact considering the above scenario in Fig. 7 considering the MSE curves corresponding to the MSD of Fig. 6. In this setting, the CG-L-RLS Algorithm 1 and the C-CG-RLS achieve the best compromise between MSE and MSD performance.

To further verify the behavior of these algorithms, we replace the music signals by AR processes, and repeat the experiment. We consider this time two distinct implementations of the constrained algorithms. Fig. 8 depicts the MSD and MSE learning curves averaged over 100 runs of every algorithm derived. We may highlight that there is a much more reasonable tradeoff between these measures for the AAF and the SAAF algorithms. Moreover, the learning curves for the CG-LRLS 2 algorithm diverge in this case.

VII. CONCLUSION

In this paper, we proposed new distributed recursions to the exact multitask problem formulated in Part I, by means of an extended network version of the CG algorithm. The approach can also be applied to single-task solutions, which are adaptively computed via a network linearly-constrained algorithms. These in turn can be compared with the reweighted formulation proposed in Part I for same objective. The resulting recursions have similar complexity and can rely on simple constraint matrices in order to enforce consensus among the agents' solutions. In particular, the exact network constrained LMS algorithm outperforms the normalized relative-variance adaptive recursions derived in Part I. This is also the case for the network constrained RLS algorithm considering well-conditioned data. For highly ill-conditioned signals, the MSD evolution of the C-CG-RLS degrades, which can be restored simply by implementing a leaky-RLS solution. The largest the ridge strength in this case, the closer to the exact AAF performance it performs. The latter turned out to exhibit the leading performance among all algorithms derived in this work. Overall, in the single-task scenario, the reweighted recursions of Part I are more robust, fast coverging and accurate in the MSD sense.

We remark that further simplification of these recursions can be attained by assuming uncorrelated node signals, similarly to the simplified recursions derived in Part I. This motivates frequency-domain, subband implementations to both multitask and single-task problems, in the form of MDF recursions. These variants will be subject of future work elsewhere.

REFERENCES

[1] A. H. Sayed, *Adaptive Filtering*. Hoboken, NJ, USA: Wiley, 2008.

[2] P. S. Chang and A. N. Willson, "Analysis of conjugate gradient algorithms for adaptive filtering," *IEEE Trans. Signal Process.*, vol. 48, no. 2, pp. 409–418, Feb. 2000, doi: 10.1109/78.823968.

[3] K. Mayyas and T. Aboulnasr, "Leaky LMS algorithm: MSE analysis for Gaussian data," *IEEE Trans. Signal Process.*, vol. 45, no. 4, pp. 927–934, Apr. 1997.

[4] E. Horita and Y. Miyanaga, "Numerically stable RLS algorithms for time-varying signals," *Electron. Commun.*, vol. 82, no. 4, pp. 26–37, Apr. 1999. [Online]. Available: https://doi.org/10.1002/(SICI)1520 6440(199904)82:4<26::AID-ECJC4)3.0.CO;2-B

[5] E. Horita, K. Sumiya, H. Urakami, and S. Mitsuishi, "A leaky RLS algorithm: Its optimality and implementation," *IEEE Trans. Signal Process.*, vol. 52, no. 10, pp. 2924–2936, Oct. 2004.

[6] T. P. Chen and Y. Chen, "Challenges and opportunities of obtaining performance from multi-core CPUs and many-core GPUs," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Taipei, Taiwan, 2009, pp. 613–616.

[7] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed. Baltimore, MD, USA: John Hopkins Univ. Press, 1989.

[8] J. Cioffi and T. Kailath, "Fast, recursive-least-squares transversal filters for adaptive filtering," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 32, no. 2, pp. 304–337, Apr. 1984.

[9] S. C. Douglas, "Performance comparison of two implementations of the leaky LMS adaptive filter," *IEEE Trans. Signal Process.*, vol. 45, no. 8, pp. 2125–2130, Aug. 1997.

[10] G. E. Bottomley and S. T. Alexander, "A novel approach for stabilizing recursive least squares filters," *IEEE Trans. Signal Process.*, vol. 39, no. 8, pp. 1770–1779, Aug. 1991.

[11] M. Salgado, G. C. Goodwin, and R. H. Middleton, "Modified least squares algorithm incorporating exponential resetting and forgetting," *Int. J. Control*, vol. 47, pp. 477–491, Apr. 1987.

[12] S. L. Gay, "Dynamically regularized fast RLS with application to echo cancellation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. Conf. Proc.*, Atlanta, GA, USA, 1996, pp. 957–960.

[13] Y. V. Zakharov, G. P. White, and J. Liu, "Low-complexity RLS algorithms using dichotomous coordinate descent iterations," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3150–3161, Jul. 2008.

[14] Y. V. Zakharov and V. H. Nascimento, "DCD-RLS adaptive filters with penalties for sparse identification," *IEEE Trans. Signal Process.*, vol. 61, no. 12, pp. 3198–3213, Jun. 2013.

[15] C. S. Ludovico and J. C. M. Bermudez, "A recursive least squares algorithm robust to low-power excitation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Montreal, QC, Canada, 2004, pp. 673–676.

[16] J. R. Guerci, "Theory and application of covariance matrix tapers for robust adaptive beamforming," *IEEE Trans. Signal Process.*, vol. 47, no. 4, pp. 977–985, Apr. 1999.

[17] C. Elisei-Iliescu, C. Stanciu, C. Paleologu, J. Benesty, C. Anghel, and S. Ciochină, "Robust variable-regularized RLS algorithms," in *Proc. Hands-Free Speech Commun. Microphone Arrays*, San Francisco, CA, USA, 2017, pp. 171–175, doi: 10.1109/HSCMA.2017.7895584.

[18] S. Ramani and J. A. Fessler, "An accelerated iterative reweighted least squares algorithm for compressed sensing MRI," in *Proc. IEEE Int. Symp. Biomed. Imag.: From Nano Macro*, 2010, pp. 257–260.

[19] M. Fornasier *et al.*, "Conjugate gradient acceleration of iteratively re-weighted least squares methods," *Comput. Optim. Appl.*, vol. 65, pp. 205–259, 2016.

[20] R. Nassif, S. Vlaski, C. Richard, and A. H. Sayed, "Learning over multitask graphs–Part I: Stability analysis," *IEEE Open J. Signal Process.*, vol. 1, pp. 28–45, 2020.

[21] Y. Chen, Y. Gu, and A. O. Hero, "Regularized least-mean-square algorithms," *arXiv:1012.5066*.

[22] A. H. Sayed, V. H. Nascimento, and S. Chandrasekaran, "Estimation and control with bounded data uncertainties," *Linear Algebra Appl.*, vol. 284, no. 1–3, pp. 259–306, 1998.

[23] R. Merched, S. Vlaski, and A. H. Sayed, "Enhanced diffusion learning over networks," in *Proc. 27th Eur. Signal Process. Conf.*, 2019, pp. 1–5.

[24] A. H. Sayed, "Adaptation, learning, and optimization over networks," *Found. Trends Mach. Learn.*, vol. 7, no. 4/5, pp. 311–801, Jul. 2014.

[25] W. Shi, Q. Ling, G. Wu, and W. Yin, "EXTRA: An exact first-order algorithm for decentralized consensus optimization," *SIAM J. Optim.*, vol. 25, no. 2, pp. 944–966, 2015.

[26] A. H. Sayed, "Adaptive networks," *Proc. IEEE*, vol. 102, no. 4, pp. 460–497, Apr. 2014.

[27] J. Tsitsiklis and M. Athans, "Convergence and asymptotic agreement in distributed decision problems," *IEEE Trans. Autom. Control*, vol. 29, no. 1, pp. 42–50, Jan. 1984.

[28] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multiagent optimization," *IEEE Trans. Autom. Control*, vol. 54, no. 1, pp. 48–61, Jan. 2009.

[29] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione, "Gossip algorithms for distributed signal processing," *Proc. IEEE*, vol. 98, no. 11, pp. 1847–1864, Nov. 2010.

[30] S. Kar and J. M. F. Moura, "Convergence rate analysis of distributed gossip (linear parameter) estimation: Fundamental limits and tradeoffs," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 4, pp. 674–690, Aug. 2011.

[31] S. Zhang, F. Zhang, H. Chen, R. Merched, and A. H. Sayed, "Frequency-domain diffusion adaptation over networks," *IEEE Trans. Signal Process.*, vol. 69, pp. 5419–5430, 2021.

[32] J. Chen, C. Richard, and A. H. Sayed, "Multitask diffusion adaptation over networks," *IEEE Trans. Signal Process.*, vol. 62, no. 16, pp. 4129–4144, Aug. 2014.

[33] J. Chen, C. Richard, and A. H. Sayed, "Multitask diffusion adaptation over networks with common latent representations," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 3, pp. 563–579, Apr. 2017.

[34] A. H. Sayed and C. Lopes, "Distributed recursive least-squares strategies over adaptive networks," in *Proc. 40th Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, CA, USA, 2006, pp. 233–237.

[35] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, "Diffusion recursive least-squares for distributed estimation over adaptive networks," *IEEE Trans. Signal Process.*, vol. 56, no. 5, pp. 1865–1877, May 2008.

[36] A. Bertrand, M. Moonen, and A. H. Sayed, "Diffusion bias-compensated RLS estimation over adaptive networks," *IEEE Trans. Signal Process.*, vol. 59, no. 11, pp. 5212–5224, Nov. 2011.

[37] Z. Liu, Y. Liu, and C. Li, "Distributed sparse recursive least-squares over networks," *IEEE Trans. Signal Process.*, vol. 62, no. 6, pp. 1386–1395, Mar. 2014.

[38] X. Cao and K. J. R. Liu, "Decentralized sparse multitask RLS over networks," *IEEE Trans. Signal Process.*, vol. 65, no. 23, pp. 6217–6232, Dec. 2017.

[39] R. Arablouei, K. Dogancay, S. Werner, and Y. Huang, "Adaptive distributed estimation based on recursive least-squares and partial diffusion," *IEEE Trans. Signal Process.*, vol. 62, no. 14, pp. 3510–3522, Jul. 2014.

[40] Y. Yu, H. Zhao, R. C. de Lamare, Y. Zakharov, and L. Lu, "Robust distributed diffusion recursive least squares algorithms with side information for adaptive networks," *IEEE Trans. Signal Process.*, vol. 67, no. 6, pp. 1566–1581, Mar. 2019.

[41] A. Rastegarnia, “Reduced-communication diffusion RLS for distributed estimation over multi-agent networks,” *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 67, no. 1, pp. 177–181, Jan. 2020.

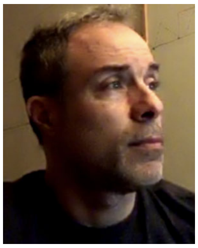
[42] P. D. Lorenzo, S. Barbarossa, and S. Sardellitti, “Distributed signal recovery based on in-network subspace projections,” in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2019, pp. 5242–5246.

[43] R. Nassif, S. Vlaski, and A. H. Sayed, “Adaptation and learning over networks under subspace constraints – Part I: Stability analysis,” *IEEE Trans. Signal Process.*, vol. 68, pp. 2948–2962, Jan. 2020.

[44] R. Nassif, S. Vlaski, C. Richard, J. Chen, and A. H. Sayed, “Multitask learning over graphs,” *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 14–25, May 2020.

[45] R. Nassif, S. Vlaski, C. Richard, and A. H. Sayed, “A regularization framework for learning over multitask graphs,” *IEEE Signal Process. Lett.*, vol. 26, no. 2, pp. 297–301, Feb. 2019.

[46] X. Cao and K. J. R. Liu, “Decentralized sparse multitask RLS over networks,” *IEEE Trans. Signal Process.*, vol. 65, no. 23, pp. 6217–6232, Dec. 2017.



RICARDO MERCHED (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in electronics engineering from the Federal University of Rio de Janeiro (UFRJ), Rio de Janeiro, Brazil, and the Ph.D. degree in electrical engineering from the University of California, Los Angeles, CA, USA, in 2001. He joined the Department of Electrical and Computer Engineering, UFRJ, in 2002. He was a Visiting Professor with the University of California Irvine, Irvine, CA, USA, and University Graduate Center, Oslo, Norway, during 2006–2007, and with École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, during 2018–2019. His current main research interests include machine learning, complex networks, and efficient DSP techniques for communications, RADAR imaging, and biomedical imaging. He was an Associate Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I, IEEE TRANSACTIONS ON SIGNAL PROCESSING LETTERS, and *EURASIP, European Journal on Advances in Signal Processing*.