Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Enhanced Gradient Tracking Algorithms for Distributed Quadratic Optimization via Sparse Gain Design

Guido Carnevale * **Michelangelo Bin** ** **Ivano Notarnicola** *
Lorenzo Marconi * **Giuseppe Notarstefano** *

* *Dept. of Electrical, Electronic and Information Engineering, Alma Mater Studiorum Università di Bologna, Bologna, Italy {name.lastname}@unibo.it.*
** *Dept. of Electrical and Electronic Engineering, Imperial College London, London, UK m.bin@imperial.ac.uk*

Abstract: In this paper we propose a new control-oriented design technique to enhance the algorithmic performance of the distributed gradient tracking algorithm. We focus on a scenario in which agents in a network aim to cooperatively minimize the sum of convex, quadratic cost functions depending on a common decision variable. By leveraging a recent system-theoretical reinterpretation of the considered algorithmic framework as a closed-loop linear dynamical system, the proposed approach generalizes the diagonal gain structure associated to the existing gradient tracking algorithms. Specifically, we look for closed-loop gain matrices that satisfy the sparsity constraints imposed by the network topology, without however being necessarily diagonal, as in existing gradient tracking schemes. We propose a novel procedure to compute stabilizing sparse gain matrices by solving a set of nonlinear matrix inequalities, based on the solution of a sequence of approximate linear versions of such inequalities. Numerical simulations are presented showing the enhanced performance of the proposed design compared to existing gradient tracking algorithms.

1. INTRODUCTION

Optimization algorithms are iterative procedures updating a decision variable with the aim of minimizing a given cost function. Many of these procedures can be seen as dynamical systems incorporating some "feedback" actions and, thus, can be conveniently studied in the context of control and system theory. Pioneering works approaching optimization from a system theoretical perspective are Wang and Elia (2010, 2011). More recently, Lessard et al. (2016) investigated the design of centralized optimization algorithms by robust control arguments, while Hu and Lessard (2017) provided different interpretations for several optimization algorithms in terms of loop-shaping, PID controllers, lag compensators, and other well-known control techniques. In this paper we focus on optimization problems that are *distributed*, namely that are addressed by a network of agents with a peer-to-peer structure, i.e., without any centralized unit that knows all the data and takes all the decisions. We mention existing approaches to sparse gain design for dynamical systems because distributed solutions are associated to sparse system matrices. In Lin et al. (2013), a Lyapunov-based technique for optimal sparse state-feedback design is proposed leveraging the Alternating Direction Method of Multipliers (ADMM). A similar problem is considered in Lin et al. (2011) in which, instead an augmented Lagrangian approach is employed while in Fardad and Jovanović (2014) the authors used sequential convex programming to accomplish sparse design. In Babazadeh and Nobakhti (2016), an algorithm for a sparse gain synthesis based on ADMM and regularization is proposed. In Lamperski and Lessard (2015) a unified design strategy for decentralized linear quadratic state-feedback controllers is proposed to account also for delays. Very recently, a strategy based on the so-called Projection Lemma is proposed in Ferrante et al. (2019) for the design of structured stabilizers for linear systems.

We consider a specific kind of distributed optimization algorithms. It is proposed by the early works Nedić and Ozdaglar (2009) and Nedić et al. (2010), in which the gradient method has been combined with consensus averaging in order to design a distributed optimization algorithm. This method has been extended by introducing a "tracking action" based on the dynamic average consensus (see Zhu and Martínez (2010), Kia et al. (2019)) in order to let the agents obtain a local estimate of the centralized gradient of the whole cost function. This distributed algorithm is called *gradient tracking*, and it has been proposed in several variants in Varagnolo et al. (2015), Di Lorenzo and Scutari (2016), Nedić et al. (2017), Qu and Li (2017), Xu et al. (2017), Xi et al. (2017), Xin and Khan (2018), Scutari and Sun (2019).

In this paper, we propose a variation of the gradient tracking algorithm by introducing more general local descent directions. We build on a reinterpretation given in Bin

⋆ This result is part of a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 638992 - OPT4SMART).

et al. (2019) of the considered optimization algorithm as a control problem, in which the control objective consists in steering the local agent estimates toward the consensual optimum, and the optimization algorithm plays the role of an output-feedback regulator trying to fulfill such goal. The choice of the control gains is thus approached as a stabilization problem whose solution must *(i)* satisfy the sparsity constraints imposed by the network topology preserving, in this way, the distributed implementation of the algorithm, and *(ii)* guarantee the sought stability and tracking properties of the closed-loop system. The combination of these requirements gives rise to a set of nonlinear matrix inequalities that are in general challenging to solve. We thus propose a novel procedure providing a solution to the problem, which is based on the recursive solution of a sequence of approximate versions of the original inequalities. The contributions of the paper can be summarized as follows: (i) we further develop the system-theoretical interpretation of distributed optimization algorithms of Bin et al. (2019), by showing that more general descent directions are possible, and even natural, when the problem is seen under a control-theoretical perspective; (ii) we propose a new iterative procedure to solve the resulting stabilization problem; (iii) we show through simulations that this novel approach turns out to be less conservative in terms of convergence rate than the standard gradient tracking. As shown in the numerical study, the proposed design approach yields considerable performance enhancements with respect to the state-of-art algorithms. Moreover, the pursued approach gives new insight on the usage of control theoretical methods in the study of distributed optimization algorithms.

The work is organized as follows. In Section 2 we introduce the distributed optimization framework. In Section 3 we develop our novel approach for sparse gain design. Finally, in Section 4 simulation results comparing our approach with the existing gradient tracking scheme are presented.

Notation. We consider discrete-time systems of the form $x(t+1) = \phi(x(t))$. Time arguments are omitted when clear from the context and, for compactness, x^+ is used in place of $x(t+1)$. A square matrix is said to be Schur if all its eigenvalues lie inside the open unit disc. Given a square matrix $F \in \mathbb{R}^{d\times d}$, a set $\mathcal{V} \subset \mathbb{R}^d$ is said to be F-invariant if for all $v \in \mathcal{V}$ it holds $Fv \in \mathcal{V}$. We denote by I_d the $d \times d$ identity matrix and by 0_d the $d \times d$ matrix of zeros. The column vector of d ones is denoted by 1_d. Moreover, we define $\mathbf{1} := 1_N \otimes I_d$, in which $\otimes$ denotes the Kronecker product. For $x \in \mathbb{R}^{n_1}$ and $z \in \mathbb{R}^{n_2}$, we denote by $\mathrm{col}(x, z) \in \mathbb{R}^{n_1+n_2}$ their column concatenation. For a finite set S, we denote by $|S|$ its cardinality. Given a matrix $M \in \mathbb{R}^{n\times n}$, we denote by $\|M\|_2$ its 2-norm.

2. DISTRIBUTED OPTIMIZATION

We consider the following optimization problem

$$\min_{\theta\in\mathbb{R}^d} \sum_{i=1}^{N} f_i(\theta), \tag{1}$$

in which, for each $i \in \{1, 2, \ldots, N\}$, $f_i : \mathbb{R}^d \to \mathbb{R}$ has the following quadratic form

$$f_i(\theta) = \frac{1}{2}(\theta - \Gamma_i\theta_0)^\top C_i(\theta - \Gamma_i\theta_0), \tag{2}$$

with $C_i \in \mathbb{R}^{d\times d}$ symmetric and positive definite, $\Gamma_i \in \mathbb{R}^{d\times p}$, and $\theta_0 \in \mathbb{R}^p$. It is easy to show that (1) admits a unique optimal solution $\theta^\star \in \mathbb{R}^p$ given by

$$\theta^\star = \Sigma\theta_0, \tag{3}$$

with

$$\Sigma := \left(\sum_{i=1}^{N} C_i\right)^{-1} \sum_{i=1}^{N} C_i\Gamma_i.$$

We consider solution algorithms to solve (1) that are *distributed* in the following sense. We are given a network of N agents, each one storing and updating a *local* estimate $x_i \in \mathbb{R}^d$ of the optimal solution $\theta^\star$ of (1). The agents have access only to partial information of the data of (1), given by the "output"

$$y_i := \nabla f_i(x_i) = C_i x_i + Q_i\theta_0, \tag{4}$$

in which $Q_i := -C_i\Gamma_i$.

We model communication among the agents in the network by means of a fixed, undirected and connected graph $\mathcal{G} = (\{1, 2, \ldots, N\}, \mathcal{E})$. A pair (i, j) belongs to $\mathcal{E}$ if the agents i and j can exchange information. We denote by $\mathcal{N}_i$ the set of neighbors of agent i. We assume that the agents can exchange their local estimates x_i and outputs y_i with their neighbors. Our goal is to design a *local update law* for each agent ensuring that, asymptotically, each local estimate converges to the optimum (3). More precisely, the problem reads as follows.

Problem 2.1. Find an update law for x_i, depending only on the local available information given by the quantities (x_j, y_j) for all $j \in \mathcal{N}_i$, such that

$$\lim_{t\to\infty} x_i(t) = \Sigma\theta_0 = \theta^\star. \qquad \triangle$$

We approach Problem 2.1 by using an iterative distributed optimization method based on the gradient tracking algorithm (see the introduction for relevant related literature), described by the following pair of updates executed by each agent i of the network

$$x_i^+ = \sum_{j\in\mathcal{N}_i} a_{ij}x_j - \gamma s_i \tag{5a}$$

$$s_i^+ = \sum_{j\in\mathcal{N}_i} \tilde{a}_{ij}s_j + \nabla f_i(x_i^+) - \nabla f_i(x_i), \tag{5b}$$

in which $\gamma > 0$ is a control parameter called "stepsize", a_{ij} and $\tilde{a}_{ij}$ are entries of the row stochastic matrix $A \in \mathbb{R}^{N\times N}$ and of the column stochastic matrix $\tilde{A} \in \mathbb{R}^{N\times N}$ respectively, and the variable $s = \mathrm{col}(s_1, s_2, \ldots, s_N) \in \mathbb{R}^{Nd}$ is an auxiliary variable whose role is to asymptotically provide an estimate of (namely track) the centralized gradient of the cost function i.e., $\sum_{i=1}^{N} \nabla f_i(x_i)$. For this reason, the variable s is typically referred to as the "gradient tracker". It can be proved that, under strong convexity of the local functions f_i and Lipschitz continuity of their gradients (properties fulfilled by (2)), for each initial condition $x_i(0)$ and with the tracker initialized as $s_i(0) = \nabla f_i(x_i(0))$ for all $i \in \{1, 2, \ldots, N\}$, for sufficiently small values of stepsize γ, the sequence $\{(x_1(t), x_2(t), \ldots, x_N(t))\}_{t\geq 0}$ generated by (5a) converges to the optimum $\theta^\star$. In order to eliminate the non causal term $\nabla f_i(x_i^+)$ from (5b), we introduce the following change of variables

$$s_i \mapsto z_i := s_i - \nabla f_i(x_i),$$

which leads to

$$x_i^+ = \sum_{j\in\mathcal{N}_i} a_{ij}x_j - \gamma(z_i + \nabla f_i(x_i)) \tag{6a}$$

$$z_i^+ = \sum_{j\in\mathcal{N}_i} \tilde{a}_{ij}z_j + \sum_{j\in\mathcal{N}_i} \tilde{a}_{ij}\nabla f_j(x_j) - \nabla f_i(x_i). \tag{6b}$$

By letting $x := \mathrm{col}(x_1, x_2, \dots, x_N)$, $z := \mathrm{col}(z_1, z_2, \dots, z_N)$ and $y := (y_1, y_2, \dots, y_N)$, we rewrite (6) as

$$\begin{aligned} x^+ &= \mathbf{A}x - \gamma z - \gamma y \\ z^+ &= \tilde{\mathbf{A}}z + (\tilde{\mathbf{A}} - I)y \\ y &= \mathrm{col}(\nabla f_1(x_1), \nabla f_2(x_2), \dots, \nabla f_N(x_N)) \end{aligned} \tag{7}$$

in which $\mathbf{A} := A \otimes \mathbf{1}$ and $\tilde{\mathbf{A}} := A \otimes \mathbf{1}$. Exploiting the linear structure of the gradients (cf. (4)), system (7) can be equivalently written as

$$\begin{bmatrix} x \\ z \end{bmatrix}^+ = F_\gamma \begin{bmatrix} x \\ z \end{bmatrix} + G_\gamma \theta_0, \tag{8}$$

in which

$$F_\gamma := \begin{bmatrix} \mathbf{A} - \gamma C & -\gamma I \\ (\tilde{\mathbf{A}} - I)C & \tilde{\mathbf{A}} \end{bmatrix}, \quad G_\gamma := \begin{bmatrix} -\gamma Q \\ (\tilde{\mathbf{A}} - I)Q \end{bmatrix},$$

where

$$C := \begin{bmatrix} C_1 & & & \\ & C_2 & & \\ & & \ddots & \\ & & & C_N \end{bmatrix}, \quad Q := \begin{bmatrix} Q_1 \\ Q_2 \\ \vdots \\ Q_N \end{bmatrix}.$$

If $\gamma = 0$, the state matrix F_γ has an eigenvalue at 1 with multiplicity $2d$ and, therefore, it is not Schur. For γ positive and sufficiently small, instead, the eigenvalues of F_γ move inside the unit circle, and F_γ becomes Schur, see Bin et al. (2019). In this sense, the term $-\gamma(z + y)$ can be interpreted as a *stabilizing output-feedback control action*, conferring asymptotic stability on (8) with $\theta_0 = 0$. The general idea behind this paper is that we may substitute the "control gain" $-\gamma I$ with a general matrix $K \in \mathbb{R}^{Nd\times Nd}$, in this way obtaining the following variation of gradient tracking algorithm

$$\begin{bmatrix} x \\ z \end{bmatrix}^+ = F_K \begin{bmatrix} x \\ z \end{bmatrix} + G_K \theta_0, \tag{9}$$

in which

$$F_K := \begin{bmatrix} \mathbf{A} + KC & K \\ (\tilde{\mathbf{A}} - I)C & \tilde{\mathbf{A}} \end{bmatrix}, \quad G_K := \begin{bmatrix} KQ \\ (\tilde{\mathbf{A}} - I)Q \end{bmatrix}.$$

By following Bin et al. (2019), we introduce some definitions to formally establish sufficient conditions to solve Problem 2.1. Define $\mathbf{n} := 2Nd$ and, with $n_\nu \leq \mathbf{n}$, consider an n_ν-dimensional subspace $\mathcal{V}$ of $\mathbb{R}^{\mathbf{n}}$, and let $T \in \mathbb{R}^{\mathbf{n}\times\mathbf{n}}$ be an orthonormal matrix of the form $T = [T_1, T_2]$, with $T_1 \in \mathbb{R}^{\mathbf{n}\times n_\nu}$ and $T_2 \in \mathbb{R}^{\mathbf{n}\times(\mathbf{n}-n_\nu)}$ satisfying

$$\mathrm{Im}(T_1) = \mathcal{V} \quad , \quad \mathrm{Im}(T_2) = \mathcal{V}^\perp.$$

Then, $\mathcal{V}$ is F-invariant if and only if

$$T^\top F T = \begin{bmatrix} F_I & F_J \\ 0 & F_E \end{bmatrix},$$

for some $F_I \in \mathbb{R}^{n_\nu\times n_\nu}$, $F_J \in R^{n_\nu\times(\mathbf{n}-n_\nu)}$ and $F_E \in \mathbb{R}^{(\mathbf{n}-n_\nu)\times(\mathbf{n}-n_\nu)}$.

Definition 2.2. The subspace $\mathcal{V}$ is said to be:

- internally stable if F_I is Schur;
- externally anti-stable if F_E has no eigenvalues inside the open unit disc. △

Let $\mathcal{O}$ be an affine subspace of $\mathbb{R}^{\mathbf{n}}$ of the form

$$\mathcal{O} := \mathcal{V} + U\theta_0, \tag{10}$$

for some linear subspace $\mathcal{V}$ of $\mathbb{R}^{\mathbf{n}}$ of dimension n_ν and for some matrix $U \in \mathbb{R}^{(\mathbf{n}-n_\nu)\times p}$ satisfying $\mathrm{Im}(U) \subset \mathcal{V}^\perp$.

Definition 2.3. Consider system (9). A set $\mathcal{O}$ of the form (10) is said to be an admissible initialization set if $\mathcal{V}$ is F_K-invariant and externally anti-stable. △

With the above definitions at reach, the results of Bin et al. (2019) are summarized within the following theorem.

Theorem 2.4. Consider system (9) and suppose that $(x(0), z(0)) \in \mathcal{O}$, in which $\mathcal{O}$ is an admissible initialization set of the form (10). If

- $\mathcal{V}$ is internally stable;
- $U = T_2(T_2^\top T_2)^{-1}T_2^\top \Pi$, with $\Pi = \mathrm{col}(\mathbf{1}\Sigma, -C\mathbf{1}\Sigma - Q)$,

then it holds

$$\lim_{t\to\infty} x(t) = \mathbf{1}\Sigma\theta_0 = \mathbf{1}\theta^\star,$$

namely all the estimates $x_1, \dots, x_N$ asymptotically converge to the optimal solution of (1), hence solving Problem 2.1. △

Theorem 2.4 suggests that the matrix F_K in (9) needs to be designed to possess an internally stable subspace that we can use to define an admissible initialization set. We underline, however, that not every subspace fits our purposes. In fact, if the matrix U in (10) is not zero, then the admissible initialization of the algorithm would depend on the unknown variable θ_0 and, as such, it would not be implementable. In the following, we construct a matrix K ensuring that the corresponding matrix F_K possesses an invariant subspace $\mathcal{V}$ with the desired properties, whose corresponding matrix U is zero. The search of such K is approached as a stabilization problem. Consider the transformation matrix $T := [T_1, T_2]$ with

$$T_1 := \begin{bmatrix} I & 0 \\ 0 & R \end{bmatrix}, \quad T_2 := \tfrac{1}{\sqrt{N}} \begin{bmatrix} 0 \\ \mathbf{1} \end{bmatrix},$$

in which $R \in \mathbb{R}^{Nd\times N(d-1)}$ is such that $RR^\top = I$ and $R^\top \mathbf{1} = 0$. Then, it holds $T^{-1} = T^\top$, and T transforms F_K to

$$T^\top F_K T = \begin{bmatrix} F_{K_I} & F_{K_J} \\ 0 & F_{K_E} \end{bmatrix},$$

in which

$$\begin{aligned} F_{K_I} &:= \begin{bmatrix} \mathbf{A} + KC & KR \\ R^T(\tilde{\mathbf{A}} - I)C & R^T\tilde{\mathbf{A}}R \end{bmatrix} \in \mathbb{R}^{(\mathbf{n}-d)\times(\mathbf{n}-d)}, \\ F_{K_J} &:= \tfrac{1}{\sqrt{N}} \begin{bmatrix} K\mathbf{1} \\ R^\top\tilde{\mathbf{A}}\mathbf{1} \end{bmatrix} \in \mathbb{R}^{(\mathbf{n}-d)\times d}, \\ F_{K_E} &:= I \in \mathbb{R}^{d\times d}. \end{aligned} \tag{11}$$

The structure of the matrix $T^\top F_K T$ implies that there exists an $(\mathbf{n} - d)$-dimensional subspace $\mathcal{V}$ that is F_K-invariant. This subspace is given by vectors $w \in \mathbb{R}^{\mathbf{n}}$ such that $Tw = \mathrm{col}(\tilde{w}_1, 0)$, with $\tilde{w}_1 \in \mathbb{R}^{\mathbf{n}-d}$. Equivalently we can say that $\mathcal{V} = \{\mathrm{col}(x, z) \in \mathbb{R}^{\mathbf{n}} | z := (z_1; z_2; \dots; z_N) \in \mathbb{R}^{Nd}, \sum_{i=1}^N z_i = 0\}$. We point out that, using the definition in Theorem 2.4, the choice for T_2 implies that $U = 0$. We stress that, according to (10), having $U = 0$ ensures that the algorithm can be properly initialized without relying on any unknown quantity (in this case $x(0)$ is arbitrary, while $z(0)$ is only constrained to have a zero mean). We also remark that this is consistent with (and actually slightly milder than) the usual initialization of gradient tracking algorithms (see Bin et al. (2019)). It remains to show that K can be chosen to guarantee that $\mathcal{V}$ is also internally stable, in this way completing the design in view of Theorem 2.4. According to Definition 2.2, we have that $\mathcal{V}$ is internally stable if and only if F_{K_I} is Schur. Moreover, the matrix F_{K_I} can be further decomposed as

$$F_{K_I} = F_{K_{I0}} + B_{I0}KH,$$

in which

$$F_{K_{I0}} := \begin{bmatrix} \mathbf{A} & 0 \\ R^\top(\tilde{\mathbf{A}} - I)C & R^\top \tilde{\mathbf{A}} R \end{bmatrix}, \quad B_{I0} := \begin{bmatrix} I \\ 0 \end{bmatrix},$$

with $H := [C, R]$. The design of a matrix K such that F_{K_I} is Schur, on the other hand, can be cast as the stabilization of the following linear system

$$x_I^+ = F_{K_{I0}} x_I + B_{I0} u_0, \tag{12a}$$

$$u_0 = KHx_I. \tag{12b}$$

From (12), it can be seen that the matrix F_{K_I} is the closed-loop matrix obtained by choosing a feedback control law. This shows that the design of the gradient tracking algorithm can be posed as a feedback stabilization problem. Notice that, however, in order to preserve the distributed nature of the optimization algorithm, we need to add an additional sparsity requirement on the gain, which will be discussed in the following section.

3. SPARSE GAIN DESIGN

In this section we present our algorithmic strategy to design a sparse gain K for (9) in order to solve Problem 2.1.

3.1 LMI Approach to Feedback Design

In this section, we derive a Linear Matrix Inequality (LMI) to obtain a stabilizing gain K for system (12). The approach relies on a discrete-time version of the Lyapunov-based approach presented in Boyd et al. (1994) for continuous-time systems. We consider the closed-loop system obtained by substituting the feedback control (12b) in (12a)

$$x_I^+ = (F_{K_{I0}} + B_{I0}KH)x_I. \tag{13}$$

For the sake of readability, from now on, we drop the subscripts in (13) and write

$$x^+ = (F + BKH)x. \tag{14}$$

The linear time-invariant system (14) is asymptotically stable if and only if there exist $Q = Q^\top \in \mathbb{R}^{n_\nu \times n_\nu}$ and $K \in \mathbb{R}^{Nd \times Nd}$ satisfying

$$\begin{cases} Q > 0 \\ Q - (F + BKH)^\top Q(F + BKH) > 0. \end{cases} \tag{15}$$

Notice that (15) is not linear in the unknown (Q, K). However, it can be equivalently written as

$$\begin{cases} Q > 0 \\ Q - (Q(F + BKH))^\top Q^{-1}(Q(F + BKH)) > 0. \end{cases} \tag{16}$$

Using the Schur complement lemma (cf. Boyd et al. (1994)), we can write (16) as

$$\begin{bmatrix} Q & Q(F + BKH) \\ (Q(F + BKH))^\top & Q \end{bmatrix} > 0 \tag{17}$$

that is still meant to be solved in the unknowns Q and K.

Let $P := Q^{-1}$. Since Q is symmetric, then also P is symmetric. By pre- and post-multiplying (17) by the following symmetric and positive definite matrix

$$\begin{bmatrix} P & 0 \\ 0 & P \end{bmatrix},$$

we obtain the equivalent inequality

$$\begin{bmatrix} P & (F + BKH)P \\ P(F + BKH)^\top & P \end{bmatrix} > 0, \tag{18}$$

which is still not linear, because of the product between the unknowns P and K. We thus introduce a further matrix $L \in \mathbb{R}^{Nd \times 2Nd}$ defined as $L := KHP$, and rewrite (18) as

$$\begin{cases} \begin{bmatrix} P & FP + BL \\ PF^\top + L^\top B^\top & P \end{bmatrix} > 0 & \text{(19a)} \\ L - KHP = 0. & \text{(19b)} \end{cases}$$

Although (19) is linear in both the unknowns P and L, it is still not sufficient to provide a *distributed* solution, since the feedback control law (12b) would not be implementable by a network of agents. In fact, the resulting matrix K obtained from (19b) need not be sparse (and typically it will not), as no *sparsity constraints* are imposed in (19). In the next subsection we show how sparsity constraints in K can be included in the solution of (19), and we develop an algorithmic procedure to solve the resulting problem.

3.2 Encoding Sparsity of the Gain Matrix

In this subsection we add a set of constraints imposing a sparsity pattern to the gain K in order to match the network topology. Formally, $K \in \mathbb{R}^{Nd \times Nd}$ must be such that its (i, j)-th is zero whenever $(i, j) \notin \mathcal{E}$.

For each pair $(i, j) \in \mathcal{E}$, let $M^{ij} \in \mathbb{R}^{Nd \times Nd}$ be the matrix having zeros everywhere except for the (i, j)-entry which is equal to 1. Then, a matrix K satisfying the sparsity constraint of the network can be expressed as a linear combination of the matrices $M^{ij} \otimes I_d$, i.e.,

$$K = \sum_{(i,j) \in \mathcal{E}} k_{ij}(M^{ij} \otimes I_d), \tag{20}$$

with $k_{ij} \in \mathbb{R}$ for all $(i, j) \in \mathcal{E}$. For notational convenience, we let $\boldsymbol{k} \in \mathbb{R}^{|\mathcal{E}|}$ collect all the coefficients k_{ij} in a single vector with an arbitrary ordering of the edges. The expansion (20) can be used to encode in (19) the desired sparsity constraints. In particular, by substituting (20) in (19b), we obtain the following constraints

$$\begin{cases} \begin{bmatrix} P & FP + BL \\ PF^\top + L^\top B^\top & P \end{bmatrix} > 0 & \text{(21a)} \\ L - \displaystyle\sum_{(i,j) \in \mathcal{E}} k_{ij} M^{ij} HP = 0, & \text{(21b)} \end{cases}$$

in the unknowns P, L and $\boldsymbol{k}$.

We stress that including the sparsity constraints (20) makes (21) a nonlinear problem because of the product between P and $\boldsymbol{k}$ in (21b). Unfortunately, no general procedure exists to solve nonlinear problems of this form. Therefore, in the following we propose an iterative procedure to tackle such nonlinear problem.

In the proposed procedure, at each iteration τ, an *approximate* version of (21) is solved, in which the decision variable P in the equality constraint (21b) is substituted with a fixed value, denoted by $\widehat{P}_\tau$, which coincides with the solution found in the previous iteration. In this way, at each iteration τ, we obtain a *linear* matrix inequality in the variables P, L and $\boldsymbol{k}$, given by

$$\begin{cases} \begin{bmatrix} P & FP+BL \\ PF^\top + L^\top B^\top & P \end{bmatrix} > 0 & \text{(22a)} \\ L - \sum_{(i,j)\in\mathcal{E}} k_{ij} M^{ij} H \widehat{P}_\tau = 0, & \text{(22b)} \end{cases}$$

which can be efficiently solved using numerical routines. Once a solution $(P_\tau, L_\tau, \boldsymbol{k}_\tau)$ to (22) is obtained, the matrix P_τ serves as new value for $\widehat{P}_{\tau+1}$ in the next iteration $\tau+1$. The procedure starts with an arbitrary initialization $\widehat{P}_0$ and is repeated until some convergence criterion is satisfied, e.g., until $\|P_{\tau+1} - P_\tau\|_2$ falls below a given threshold $\epsilon > 0$. Notice that the described procedure has no theoretical convergence guarantees. However, in Section 4 we show the effectiveness of the proposed scheme (22) for the design of sparse feedback through simulations.

Remark 3.1. We underline that (22) is a feasibility problem. Then, once its constraints are fulfilled, we can also include an optimality criterion in its selection. For this reason, in our numerical experiments we also add a cost function in order to optimize the convergence rate of the resulting optimization algorithm. △

The following Algorithm 1 summarizes the described iterative procedure.

Algorithm 1 Iterative Procedure for Sparse Gain Design

given tolerance $\epsilon > 0$
initialize $\widehat{P}_0$
for $\tau = 0, 1, 2, \ldots$ **do**
  obtain $(P_\tau, L_\tau, \boldsymbol{k}_\tau)$ as a solution to (22)
  if $\|P_\tau - \widehat{P}_\tau\| < \epsilon$ **then**
    set $\boldsymbol{k}^\star = \boldsymbol{k}_\tau$
    break
  else
    $\widehat{P}_{\tau+1} = P_\tau$
  end if
end for
retrieve the sparse gain $K^\star = \sum_{(i,j)\in\mathcal{E}} k^\star_{ij} M^{ij}$.

4. SIMULATIONS

In this section we propose a numerical study to show the effectiveness of the proposed design strategy. In particular, we compare the convergence behavior of the gradient tracking including the sparse (possibly non-diagonal) gain K (cf. (9)) with its basic version with diagonal gains (cf. (8)).

As mentioned in Remark 3.1, we include in the solution of each problem (22) the minimization of the objective

$$\|F + BKH\|_2 + \beta\|P - \widehat{P}_\tau\|_2,$$

in which $\beta > 0$ represents a trade-off parameter in the following sense. Minimizing the term $\|F + BKH\|$ reflects in maximizing the convergence rate of the resulting distributed optimization algorithm. Indeed, $\|F+BKH\|_2$ is directly related to the maximum singular value of the closed-loop matrix $F + BKH$. The term $\|P - \widehat{P}_\tau\|_2$ is, instead, a "regularization" introduced to foster the convergence of the iterative procedure in Algorithm 1. The design parameter β can be thus chosen to privilege one of the two terms as desired.

We term "Basic GT" the standard gradient tracking with diagonal gains, while we term "Rev. GT" the algorithm that implements the sparse gain K designed using our procedure described in Section 3. In order to to choose the stepsize γ in the Basic GT, we resort to Algorithm 1. Indeed, the case of diagonal K is a special case obtained by imposing a graph structure with only self-loops.

In the following, we present simulations obtained for different numbers of agents N and for different "graph density" $d_A := |\mathcal{E}|/N^2$. We set $\widehat{P}_0 = I$ and $\beta = 1$. In the figures below we plot the norm of the difference between the mean vector and the optimal solution of (1), namely

$$\|e(t)\|_2 := \left\| \theta^\star - \frac{1}{N}\sum_{i=1}^N x_i(t) \right\|_2 . \quad (23)$$

In Figure 1, three networks with respectively 5, 10 and 15 agents are considered for comparison. In all the cases we consider $d = 2$ and $d_A = 0.7$. Figure 1 shows that in

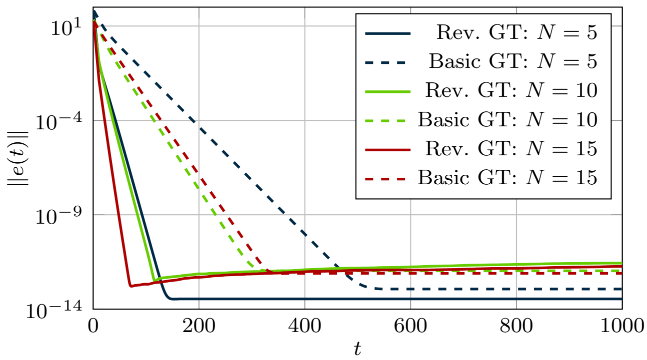

Fig. 1. Evolution of error $e(t)$ for different values of N.

all cases the Rev. GT has a faster convergence rate than the Basic GT. The converge rate enhances as the number of agents increases. This behavior can be explained by noticing that a larger number of agents implies a "larger space" in which the sparse matrix K is searched.

In Figure 2, four networks with density d_A equal to 0.3, 0.6, 0.75, and 0.9 are considered. In all the cases we consider $d = 2$ and $N = 10$. Figure 2 shows that only in one case the Basic GT is faster than the Rev. GT. Specifically, it occurs when graph density is really low, namely $d_A = 0.3$. This confirms our interpretation that the Basic GT is actually obtained by limiting the gain structure choice to diagonal matrices. We point out that, although the Basic GT is faster, both gains are associated to the same cost value

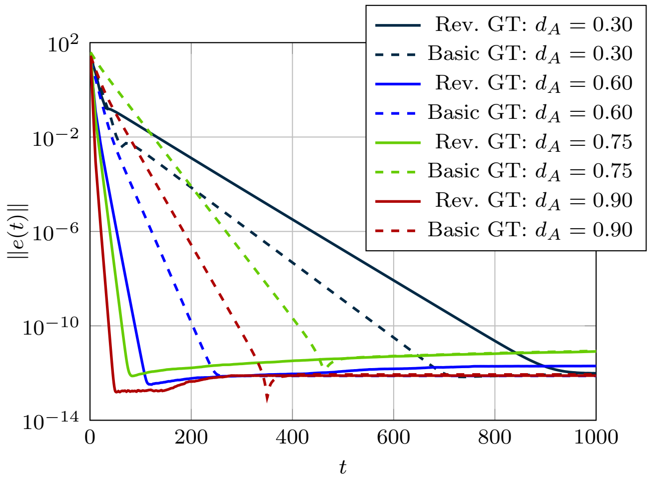

Fig. 2. Evolution of error $e(t)$ for different values of d_A.

$\|F + BKH\|_2 + \beta\|P - \widehat{P}_\tau\|_2$. This is reasonable since the cost function does not encode directly an information related to the convergence rate of the closed-loop system. Moreover, also the regularization term plays a role in the optimization procedure. Consistently to the previous case, we observe that the algorithm performance enhances as the graph density d_A increases.

5. CONCLUSIONS

In this paper we have proposed a novel method to enhance the gradient tracking algorithm for distributed quadratic optimization. As shown in simulations, the proposed procedure leads to a considerable increase of the convergence rate. This improvement is due to a more general gain structure, obtained by approaching the problem from a system theoretical point of view. In particular, we have investigated the possibility to use a more general structure for the gain matrix, instead of the diagonal one, without the violation of the sparsity constraints. We have proposed a novel procedure to impose those constraints by solving a set of nonlinear matrix inequalities, based on the solution of a sequence of approximate linear versions of such inequalities.

REFERENCES

Babazadeh, M. and Nobakhti, A. (2016). Sparsity promotion in state feedback controller design. *IEEE Transactions on Automatic Control*, 62(8), 4066–4072.

Bin, M., Notarnicola, I., Notarstefano, G., and Marconi, L. (2019). A system theoretical perspective to gradient-tracking algorithms for distributed quadratic optimization. In *IEEE Conference on Decision and Control (CDC)*, 2994–2999.

Boyd, S., El Ghaoui, L., Feron, E., and Balakrishnan, V. (1994). *Linear matrix inequalities in system and control theory*, volume 15. Siam.

Di Lorenzo, P. and Scutari, G. (2016). Next: In-network nonconvex optimization. *IEEE Transactions on Signal and Information Processing over Networks*, 2(2), 120–136.

Fardad, M. and Jovanović, M.R. (2014). On the design of optimal structured and sparse feedback gains via sequential convex programming. In *IEEE American Control Conference (ACC)*, 2426–2431.

Ferrante, F., Dabbene, F., and Ravazzi, C. (2019). On the design of structured stabilizers for LTI systems. *IEEE Control Systems Letters*, 4(2), 289–294.

Hu, B. and Lessard, L. (2017). Control interpretations for first-order optimization methods. In *IEEE American Control Conference (ACC)*, 3114–3119.

Kia, S.S., Van Scoy, B., Cortes, J., Freeman, R.A., Lynch, K.M., and Martinez, S. (2019). Tutorial on dynamic average consensus: The problem, its applications, and the algorithms. *IEEE Control Systems Magazine*, 39(3), 40–72.

Lamperski, A. and Lessard, L. (2015). Optimal decentralized state-feedback control with sparsity and delays. *Automatica*, 58, 143–151.

Lessard, L., Recht, B., and Packard, A. (2016). Analysis and design of optimization algorithms via integral quadratic constraints. *SIAM Journal on Optimization*, 26(1), 57–95.

Lin, F., Fardad, M., and Jovanović, M.R. (2011). Augmented Lagrangian approach to design of structured optimal state feedback gains. *IEEE Transactions on Automatic Control*, 56(12), 2923–2929.

Lin, F., Fardad, M., and Jovanović, M.R. (2013). Design of optimal sparse feedback gains via the alternating direction method of multipliers. *IEEE Transactions on Automatic Control*, 58(9), 2426–2431.

Nedić, A., Olshevsky, A., and Shi, W. (2017). Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM Journal on Optimization*, 27(4), 2597–2633.

Nedić, A. and Ozdaglar, A. (2009). Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1), 48–61.

Nedić, A., Ozdaglar, A., and Parrilo, P.A. (2010). Constrained consensus and optimization in multi-agent networks. *IEEE Transactions on Automatic Control*, 55(4), 922–938.

Qu, G. and Li, N. (2017). Harnessing smoothness to accelerate distributed optimization. *IEEE Transactions on Control of Network Systems*, 5(3), 1245–1260.

Scutari, G. and Sun, Y. (2019). Distributed nonconvex constrained optimization over time-varying digraphs. *Mathematical Programming*, 176(1-2), 497–544.

Varagnolo, D., Zanella, F., Cenedese, A., Pillonetto, G., and Schenato, L. (2015). Newton-Raphson consensus for distributed convex optimization. *IEEE Transactions on Automatic Control*, 61(4), 994–1009.

Wang, J. and Elia, N. (2010). Control approach to distributed optimization. In *IEEE Allerton Conf. on Communication, Control, and Computing*, 557–561.

Wang, J. and Elia, N. (2011). A control perspective for centralized and distributed convex optimization. In *IEEE Conf. on Decision and Control and European Control Conf. (CDC)-(ECC)*, 3800–3805.

Xi, C., Xin, R., and Khan, U.A. (2017). ADD-OPT: Accelerated distributed directed optimization. *IEEE Transactions on Automatic Control*, 63(5), 1329–1339.

Xin, R. and Khan, U.A. (2018). A linear algorithm for optimization over directed graphs with geometric convergence. *IEEE Control Systems Letters*, 2(3), 315–320.

Xu, J., Zhu, S., Soh, Y.C., and Xie, L. (2017). Convergence of asynchronous distributed gradient methods over stochastic networks. *IEEE Transactions on Automatic Control*, 63(2), 434–448.

Zhu, M. and Martínez, S. (2010). Discrete-time dynamic average consensus. *Automatica*, 46(2), 322–329.