The Gradient Tracking Is a Distributed Integral Action

Published Version:
The Gradient Tracking Is a Distributed Integral Action / Notarnicola, Ivano; Bin, Michelangelo; Marconi, Lorenzo; Notarstefano, Giuseppe. - In: IEEE TRANSACTIONS ON AUTOMATIC CONTROL. - ISSN 0018-9286. - STAMPA. - 68:12(2023), pp. 10050719.7911-10050719.7918. [10.1109/tac.2023.3248487]

Availability:
This version is available at: https://hdl.handle.net/11585/959315 since: 2024-02-19

Published:
DOI: http://doi.org/10.1109/tac.2023.3248487

(Article begins on next page)

I. Notarnicola, M. Bin, L. Marconi and G. Notarstefano, "The Gradient Tracking Is a Distributed Integral Action," in *IEEE Transactions on Automatic Control*, vol. 68, no. 12, pp. 7911-7918, Dec. 2023

The final published version is available online at:

https://doi.org/10.1109/TAC.2023.3248487

The Gradient Tracking Is a Distributed Integral Action

Ivano Notarnicola, *Member, IEEE*, Michelangelo Bin, *Member, IEEE*,
Lorenzo Marconi, *Fellow, IEEE*, Giuseppe Notarstefano, *Member, IEEE*

***Abstract*— We revisit the recent Gradient Tracking algorithm for distributed consensus optimization from a control theoretic viewpoint. We show that the algorithm can be constructed by solving a servomechanism control problem stemming from the distributed implementation of a centralized gradient method. Moreover, we show that, if expressed in proper coordinates, the Gradient Tracking embeds an integral action fed by a signal related to the consensus error. Finally, we provide an alternative convergence analysis based on Lyapunov arguments that also proves exponential asymptotic stability of the optimal equilibrium.**

***Index Terms*— Control for Optimization, Gradient Tracking, Integral Control, Distributed optimization**

I. Introduction

Since the dawn of control theory, integral actions have been fundamental components of control schemes [1] (the PID being an eminent example), as they are able to generate steady-state feedforward controls not computable in advance that automatically balance unknown steady-state residual terms. In this article, we revisit the recently proposed distributed *Gradient Tracking* algorithm for consensus optimization problems under a new control-theoretic perspective. We find that it fundamentally consists in a distributed integral action compensating for the incomplete information sharing due to network constraints.

More specifically, we build the Gradient Tracking algorithm starting from a naive fully distributed gradient descent policy, unavoidably characterized by a non-zero steady-state error due to the lack of full connectivity. Then, we correct this policy by adding a (distributed) servomechanism aimed at automatically compensating such steady-state error. In this way we are quite naturally led to the Gradient Tracking algorithm in the form proposed in [2], which has the advantage of a measurement-free initialization more robust to possible uncertainty in the measurements. Next, we show that, by means of a suitable change of coordinates related to the normal form, the algorithm explicitly exhibits the expression of a distributed integral action. Finally, under customary Lipschitz-continuity and strong convexity assumptions, we provide an alternative convergence proof that is shorter and simpler[1] than existing ones [3]–[12]. Moreover, in addition to exponential convergence, we also prove Lyapunov stability of the optimal equilibrium.

Overall, our findings confirm that a control-informed approach to (distributed) optimization carries considerable advantages in terms of insight and tools for analysis, in this way aligning with a recent trend concerning the applications of control to optimization.

Related Literature: Distributed consensus optimization over networks traces back to the early works [13]–[15], in which the gradient method has been combined with consensus averaging in order to design a distributed (sub)gradient algorithm. This method has been later extended by the Gradient Tracking algorithm analyzed in several variants, see, e.g., [3]–[12]. In [16], [17] accelerated versions based on the heavy-ball method and the Nesterov scheme have been proposed. An extension to stochastic optimization is proposed in [18].

From an optimization viewpoint, the Gradient Tracking algorithm is thought as embedding a *tracking mechanism* that, based on *dynamic average consensus* [19], aims to track the gradient of the global cost function by means of neighboring communications only. Such "centralized gradient" estimate is then used in the local gradient-based updates of the agents. From a control viewpoint, instead, the Gradient Tracking rather resembles an output-feedback controller, with the *tracking mechanism*[2] that is reminiscent of an observer. Indeed, we emphasize that the estimate of the centralized gradient is correct only at the optimal steady state, and not during transitory.

Pioneering works approaching distributed optimization algorithms from a control theoretic perspective are [20], [21]. In [22], distributed optimization methods are developed based on continuous-time proportional-integral controllers. More recently, geometric control tools have been used in [23]. In [24] a passivity-based perspective is adopted to analyze distributed optimization algorithms with delays. A distributed saddle-flow is proposed in [25], where the convergence analysis follows an *output regulation* approach. Recently, a contraction analysis for primal-dual dynamics has been investigated

I. Notarnicola, L. Marconi and G. Notarstefano are with the Department of Electrical, Electronic and Information Engineering, Alma Mater Studiorum - Università di Bologna, Bologna, Italy. The corresponding author is I. Notarnicola ivano.notarnicola@unibo.it.

M. Bin is with the Department of Electrical and Electronic Engineering, Imperial College London, London, UK.

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).

[1] In particular, existing proofs are typically based on optimization-informed approaches involving long lists of inequalities to show convergence properties of some specific sequences. Other works rely on Lyapunov-like arguments or control tools (as e.g., the small gain theorem) but do not fully exploit a system theoretic perspective as the one we pursue in this paper.

[2] We stress that, throughout the paper, the term "tracking" is not intended in the canonical sense of control theory, e.g., in the context of reference tracking. Yet, it is only used in reference to the "tracking mechanism" embedded in the Gradient Tracking algorithm.

in [26], while a convergence analysis approach based on semidefinite programming is given in [27]. In [28], primal-dual algorithms are interpreted as proportional-integral controllers. In [29] a general, canonical form for distributed optimization algorithms based on gradients is proposed, while in [30] the same methodology is extended to time-varying graphs. Finally, building on [2], in [31] Gradient Tracking algorithms for quadratic programs with sparse (possibly non-diagonal) gains are investigated.

Organization: In Section II, we introduce the framework of distributed consensus optimization and the Gradient Tracking algorithm. In Section III, we derive the Gradient Tracking by solving a control problem aimed at finding a distributed implementation of a vanilla centralized gradient descent algorithm. Here, we recover the Gradient Tracking in the canonical coordinates of [2]. Moreover, by means of a suitable change of coordinates, we show that the Gradient Tracking algorithm embeds an integral action. Finally, in Section IV we provide a proof of exponential stability of the optimal equilibrium.

II. The Gradient Tracking Algorithm

A. Distributed Consensus Optimization

In a *consensus optimization problem*, $N \in \mathbb{N}$ agents in a network cooperatively seek a solution to the problem

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

in which, for each $i = 1, \dots, N$, the function $f_i : \mathbb{R}^d \to \mathbb{R}$ is known to agent i only. Hence, the function $\sum_{i=1}^N f_i$ to be minimized, to which we shall refer as the *global cost function*, is not individually accessible by the agents. We consider Problem (1) under the following assumptions.

Assumption 2.1: For all $i = 1, \dots, N$, the function f_i has Lipschitz continuous gradient.

Assumption 2.2: The global cost function $\sum_{i=1}^N f_i$ is strongly convex.

Under Assumption 2.2, Problem (1) admits a unique global minimum, which we denote by[3] $\mathrm{x}_\star \in \mathbb{R}^d$. In the following, we use the compact notation $\nabla \mathbf{F}(x) := (\nabla f_1(x_1), \dots, \nabla f_N(x_N))$, for all $x = (x_1, \dots, x_N) \in \mathbb{R}^{Nd}$. We underline that, by optimality, $(\mathbf{1} \otimes I_d)^\top \nabla \mathbf{F}(\mathbf{1} \otimes \mathrm{x}_\star) = 0$ with $\mathbf{1} := (1, \dots, 1) \in \mathbb{R}^N$.

Agents communicate according to a directed and strongly connected communication network formally represented by a graph $\mathcal{G} = (\{1, \dots, N\}, \mathcal{E})$ with edge set $\mathcal{E} \subseteq \{1, \dots, N\}^2$. We denote by $\mathcal{N}_i := \{j \in \{1, \dots, N\} \mid (i,j) \in \mathcal{E}\}$ the neighborhood of Agent i. The communication is synchronous. As customary in consensus-based algorithms, we assume that agents are equipped with properly defined weights that are used to combine neighboring states. Specifically, we consider two weight matrices $R \in \mathbb{R}^{N \times N}$ and $C \in \mathbb{R}^{N \times N}$ with positive entries, i.e., $r_{ij} \geq 0$ and $c_{ij} \geq 0$, matching the communication graph $\mathcal{G}$ in the sense that $r_{ij} = c_{ij} = 0 \iff (i,j) \notin \mathcal{E}$. Moreover, we assume that R is row stochastic and C is column stochastic, implying

$$R\mathbf{1} = \mathbf{1}, \quad \pi_R^\top R = \pi_R^\top, \quad C\pi_C = \pi_C, \quad \mathbf{1}^\top C = \mathbf{1}^\top \tag{2a}$$

for some $\pi_R \in \mathbb{R}^N$ and $\pi_C \in \mathbb{R}^N$ with positive entries. In view of the Perron-Frobenius theorem [32], we can choose π_R and π_C so that

$$\mathbf{1}^\top \pi_R = 1, \qquad \mathbf{1}^\top \pi_C = 1, \qquad \pi_R^\top \pi_C > 0, \tag{2b}$$

all of which shall be assumed in the remainder of the paper. We stress that R and C need not be equal nor symmetric.

B. The Gradient Tracking Algorithm

The Gradient Tracking distributed algorithm combines a gradient-based method and two averaging mechanisms aimed, respectively, at aggregating the neighboring estimates and at reconstructing ("tracking") the current value of the gradient $\sum_{i=1}^N \nabla f_i(x_{i,k})$ of the global cost function, inaccessible to the agents due to the network constraints. More specifically, each agent i maintains a local state variable $x_{i,k} \in \mathbb{R}^d$ (with $k \in \mathbb{N}$ denoting the discrete time index) representing an estimate of the optimal solution $\mathrm{x}_\star$ of Problem (1), and an auxiliary variable $s_{i,k} \in \mathbb{R}^d$ representing the aforementioned estimate of the gradient of the global cost function. At each $k \in \mathbb{N}$, these variables are updated as

$$x_{i,k+1} = \sum_{j \in \mathcal{N}_i} r_{ij} x_{j,k} - \gamma s_{i,k} \tag{3a}$$

$$s_{i,k+1} = \sum_{j \in \mathcal{N}_i} c_{ij} s_{j,k} + \nabla f_i(x_{i,k+1}) - \nabla f_i(x_{i,k}), \tag{3b}$$

in which, for all $i = 1, \dots, N$, $x_{i,0} \in \mathbb{R}^d$ is arbitrary and

$$s_{i,0} = \nabla f_i(x_{i,0}). \tag{4}$$

Moreover, r_{ij} and c_{ij} are the (i,j)-th entries of R and C introduced in Section II-A (cf. (2a)), while $\gamma > 0$ is a design parameter, the *stepsize*, that must be chosen small enough to ensure convergence, see, e.g., [3]–[12].

Due to the properties of R, Equation (3a) can be seen as a forced consensus update law, where each agent tries to pull its estimate towards the direction defined by its local estimate $s_{i,k}$ of the global cost function's gradient. Equation (3b), instead, implements a decentralized tracking algorithm for the gradient of the global cost function, where the averaging term aggregating the neighboring estimates $s_{j,k}$ employs column-stochastic weights c_{ij}. Necessity of column stochasticity has been already investigated in the context of push-pull gradient methods, see, e.g., [11] and reference therein, and it will be further discussed in Section III.

For the estimates produced by the Gradient Tracking algorithm (15), the following convergence result holds.

Theorem 2.3: Let Assumptions 2.1 and 2.2 hold. Then, every solution[4] $((x_i)_{i=1,\dots,N}, (s_i)_{i=1,\dots,N})$ of System (3) initialized according to (4) satisfies

$$\lim_{k \to \infty} |x_{i,k} - \mathrm{x}_\star| = 0$$

[3] Throughout the paper, we use the roman x when referring to the optimization variable of problem (1), and the italic x when referring to the agents' estimates.

[4] Throughout the paper, by solution we shall always tacitly mean a maximal solution defined on the whole time domain $\mathbb{N}$.

$$\lim_{k\to\infty} |s_{i,k}| = 0 \tag{5}$$

exponentially, for all $i = 1, \ldots, N$.

Several proofs of Theorem 2.3, as well as of similar results concerning variations of (3), can be found in the literature. See, for instance, [3]–[12]. In addition, by leveraging the control-theoretic reinterpretation of the Gradient Tracking given in forthcoming Section III, we provide in Section IV an alternative analysis based on Lyapunov arguments, which also establishes asymptotic stability.

III. The Gradient Tracking as a Control Problem

In this section, we show that the Gradient Tracking algorithm can be obtained in a rather natural way by solving a control problem where the "plant" is a vanilla distributed implementation of the centralized gradient descent method, and the "controller" is a servomechanism aimed at compensating, at the steady state, for the lack of information due to the network constraints. This approach highlights that the Gradient Tracking algorithm embeds a distributed integral action processing an error signal closely related to the consensus error.

Without loss of generality, in the remainder of the article we focus on the scalar version of Problem (1) obtained with $d = 1$. This simplifies the computations without limiting the validity of the results. Indeed, all presented results apply also to non-scalar optimization problems with $d > 1$ by properly introducing Kronecker products.

A. From Centralized to Decentralized

We start from the standard gradient descent method applied to Problem (1), reading as[5]

$$\mathrm{x}_{k+1} = \mathrm{x}_k - \gamma \tfrac{1}{N}\mathbf{1}^\top \nabla \mathbf{F}(\mathbf{1}\mathrm{x}_k), \tag{6}$$

with $\gamma > 0$ the stepsize and x_0 arbitrary. Under Assumptions 2.1 and 2.2, every sequence generated by (6) converges at a linear rate (exponentially) to the optimal solution $\mathrm{x}_\star$, see [33]. The algorithm (6) requires the knowledge of the gradient of the global cost function, and hence must be executed by a central unit. As a first step toward a distributed implementation of (6), we consider an algorithm made of N identical copies of (6), each one executed by a different agent. To this end, we define $x_k := \mathbf{1}\mathrm{x}_k$, stacking N copies of x_k, and we write the following update laws derived from (6)

$$x_{k+1} = x_k - \gamma \tfrac{1}{N}\mathbf{1}\mathbf{1}^\top \nabla \mathbf{F}(x_k), \tag{7}$$

initialized so as $x_0 \in \mathrm{span}(\mathbf{1})$. Every sequence $(x_k)_{k\in\mathbb{N}}$ generated by (7) satisfies $x_k \in \mathrm{span}(\mathbf{1})$ for all $k \in \mathbb{N}$, and converges to $\mathbf{1}\mathrm{x}_\star$.

Notice that the initialization $x_0 \in \mathrm{span}(\mathbf{1})$ in (7) is necessary for convergence. Nevertheless, such a constraint can be relaxed by considering the following modification

$$x_{k+1} = Rx_k - \gamma \tfrac{1}{N}\mathbf{1}\mathbf{1}^\top \nabla \mathbf{F}(x_k) \tag{8}$$

where R is the row stochastic matrix introduced in Section II-A. Row stochasticity of R implies that the components of x_k orthogonal to $\mathrm{span}(\mathbf{1})$ are filtered out, ensuring convergence from an arbitrary initial condition.

While (8) is composed of N replicas, each one implemented locally on a single agent, it is still not acceptable since each replica needs access to all the gradients ∇f_i, $i = 1, \ldots, N$. Indeed, the matrix $\frac{1}{N}\mathbf{1}\mathbf{1}^\top$ in (8) can be seen as the weighted adjacency matrix of a complete graph. Towards a fully-distributed implementation, we now focus on how to relax such an all-to-all communication, with the aim of obtaining a scheme in which each agent i only performs local computations and only employs neighboring information. We start by considering a general update law of the following form

$$x_{k+1} = Rx_k - \gamma\nabla\mathbf{F}(x_k) + u_k, \tag{9}$$

in which $x_0 \in \mathbb{R}^N$ is arbitrary, R is as in (8), the term $-\gamma\nabla\mathbf{F}(x_k) = -\gamma\,\mathrm{col}(\nabla f_1(x_{1,k}), \ldots, \nabla f_N(x_{N,k}))$ is a locally computable term providing, for each agent i, a control action equal to the local gradient $\nabla f_i(x_{i,k})$, and $u_k := (u_{1,k}, \ldots, u_{N,k})$ is a further control input to be designed. Clearly, taking R as a doubly stochastic matrix and $u = 0$ is a feasible choice implementing the naive distributed gradient method proposed, e.g., in [14]. However, with this choice the resulting estimates converge to the optimum $\mathrm{x}_\star$ only if vanishing stepsizes are adopted, at the expense of a highly degraded convergence and lack of alertness.

In general, convergence to the optimum $\mathrm{x}_\star$ is obtained if the generated sequence converges to an equilibrium $x_{\mathrm{ss}} \in \mathbb{R}^N$ fulfilling, at once, the following two properties:

P1. Consensus: $x_{\mathrm{ss}} \in \mathrm{span}(\mathbf{1})$.

P2. Optimality: $\mathbf{1}^\top\nabla\mathbf{F}(x_{\mathrm{ss}}) = 0$.

We thus focus on the choice u in (9) to achieve both **P1** and **P2**. Motivated by the algorithm (8), we add and subtract $\gamma\frac{1}{N}\mathbf{1}\mathbf{1}^\top\nabla\mathbf{F}(x_k)$ in (9) to obtain

$$\underbrace{x_{k+1} = Rx_k - \gamma\tfrac{1}{N}\mathbf{1}\mathbf{1}^\top\nabla\mathbf{F}(x_k)}_{\text{Algorithm (8)}} + \underbrace{u_k + \gamma(\tfrac{1}{N}\mathbf{1}\mathbf{1}^\top - I)\nabla\mathbf{F}(x_k)}_{\text{Mismatch}}. \tag{10}$$

This reformulation highlights that the "control input" u must be chosen to compensate for the mismatch between the (non-implementable, but correct) centralized descent term $-\gamma\frac{1}{N}\mathbf{1}\mathbf{1}^\top\nabla\mathbf{F}(x_k)$ and the locally computable (but incorrect) gradient term $-\gamma\nabla\mathbf{F}(x_k)$. We underline that this compensation property of u is not necessary at all times k, but only at steady state. Indeed, u must be designed such that the following hold.

R1. Existence of an Optimal Equilibrium: There exists a state-input equilibrium pair $(x_{\mathrm{ss}}, u_{\mathrm{ss}}) \in \mathbb{R}^{2N}$ for (9), such that u_{ss} satisfies $u_{\mathrm{ss}} := -\gamma(\frac{1}{N}\mathbf{1}\mathbf{1}^\top - I)\nabla\mathbf{F}(x_{\mathrm{ss}})$.

R2. Attractiveness of the Equilibrium: All solutions of System (9) converge to the equilibrium x_{ss}.

We underline that, in view of (10), Item **R1** implies that x_{ss} is an equilibrium of the centralized algorithm (8). Hence, x_{ss} is optimal in the sense that it satisfies **P1** and **P2**. We approach the problem of designing u ensuring **R1** and **R2** as a canonical set-point control problem [34]. In particular, we seek a *feedback* expression for u. Namely, we require u to be generated as the output of a distributed dynamical system with

[5]The scaling factor $\frac{1}{N}$ in (6) is ineffective, but it is instrumental for the successive derivations.

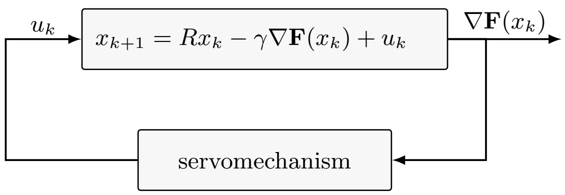

Fig. 1. Block diagram of (13).

state $z = (z_1, \dots, z_N) \in \mathbb{R}^N$, input $\nabla\mathbf{F}(x)$, and described by equations of the form[6]

$$\begin{aligned} z_{k+1} &= H_1 z_k + H_2 \nabla\mathbf{F}(x_k), \quad z_0 \in Z_0, \\ u_k &= K_1 z_k + K_2 \nabla\mathbf{F}(x_k), \end{aligned} \tag{11}$$

for some $Z_0 \subset \mathbb{R}^N$ and some matrices H_1, H_2, K_1, K_2 of suitable dimensions that match the network constraints.

In the following subsection, we pick a particular realization of the controller (11) and show that **R1** holds. Attractiveness of the equilibrium x_{ss} (Item **R2**) is instead the subject of the next Section IV. Eventually, we show that the obtained closed-loop system is indeed a representation in different coordinates of the Gradient Tracking (3). Finally, we show that, in certain coordinates, it can be seen as the interconnection between a controlled system and an integral action.

B. Design of (11) and Compensation

We first notice that we can take $K_2 = 0$ in (11) without loss of generality, since (9) already has a decentralized gradient term. Consequently, we can also pick $K_1 = I$. Indeed, since K_1 must be invertible, this is always true up to a change of coordinates. This leads to $u_k = z_k$ for all $k \in \mathbb{N}$. Second, in order to guarantee that every equilibrium (x_{ss}, z_{ss}) of the closed-loop system for which **P1** and **P2** hold satisfies $z_{ss} = u_{ss} = -\gamma(\frac{1}{N}\mathbf{1}\mathbf{1}^\top - I)\nabla\mathbf{F}(x_{ss}) = \gamma\nabla\mathbf{F}(x_{ss})$, we must ensure that $I - H_1 = \frac{1}{\gamma}H_2$. This suggests the choice $H_2 = -\gamma(H_1 - I)$. Finally, we notice that the previous condition $z_{ss} = \gamma\nabla\mathbf{F}(x_{ss})$ also implies $\mathbf{1}^\top z_{ss} = 0$. As better clarified below, without knowing z_{ss} a priori this can be achieved within the limits of the previous choices by taking H_1 column stochastic and

$$Z_0 := \{z_0 \in \mathbb{R}^N \mid \mathbf{1}^\top z_0 = 0\}. \tag{12}$$

Hence, without loss of generality, we can take $H_1 = C$, with C the column stochastic matrix introduced in Section II-A.

In summary, the previous reasoning suggests choosing $K_1 = I$, $K_2 = 0$, $H_1 = C$, $H_2 = -\gamma(C - I)$, and (12) as initialization set. The resulting (feedback) interconnection of (9) and (11) reads as

$$x_{k+1} = Rx_k - \gamma\nabla\mathbf{F}(x_k) + z_k \tag{13a}$$

$$z_{k+1} = Cz_k - \gamma(C - I)\nabla\mathbf{F}(x_k), \tag{13b}$$

with x_0 arbitrary and $z_0 \in Z_0$, initialized so as $\mathbf{1}^\top z_0 = 0$. Notice that the system (13) is distributed, in that each update law only uses local information. A block diagram of (13) is depicted in Figure 1.

[6] The state z has N components, one for each agent. As it is shown later in Section III-B, this turns out to be sufficient in the considered single-variable case. Nevertheless, different choices are possible.

We now study the reachable equilibria of (13). First, notice that, in view of (13b), every equilibrium (x_{ss}, z_{ss}) of (13) satisfies

$$\begin{aligned} 0 &= (C - I)z_{ss} - \gamma(C - I)\nabla\mathbf{F}(x_{ss}) \\ &\implies z_{ss} - \gamma\nabla\mathbf{F}(x_{ss}) \in \operatorname{span}(\pi_C). \end{aligned}$$

Moreover, (13a) implies

$$\begin{aligned} (I - R)x_{ss} &= -\gamma\nabla\mathbf{F}(x_{ss}) + z_{ss} \\ &\implies \pi_R^\top\big(z_{ss} - \gamma\nabla\mathbf{F}(x_{ss})\big) = 0. \end{aligned}$$

Since $\pi_R^\top \pi_C > 0$ (see (2)), the previous relations imply

$$z_{ss} = \gamma\nabla\mathbf{F}(x_{ss}). \tag{14}$$

This yields two consequences. First, from (13a) we obtain $x_{ss} \in \ker(I - R) = \operatorname{span}(\mathbf{1})$, which is **P1**. Second, we notice that every solution originating from Z_0 satisfies $\mathbf{1}^\top z_k = 0$ for all $k \in \mathbb{N}$. Hence, every equilibrium reachable from (12) necessarily satisfies $\mathbf{1}^\top z_{ss} = 0$. Then, (14) implies $\mathbf{1}^\top\nabla\mathbf{F}(x_{ss}) = \mathbf{1}^\top z_{ss} = 0$, which is **P2**. Hence, we have shown that every equilibrium of (13) is a consensual stationary point of Problem (1). We stress that Assumptions 2.1 and 2.2 play no role in establishing such property.

In view of Assumption 2.2, the latter condition actually implies that $x_{ss} = \mathbf{1}\mathrm{x}_\star$ with $\mathrm{x}_\star$ being the unique optimal solution of Problem (1). Therefore, we conclude that there is a unique equilibrium reachable from Z_0 given by

$$(x_{ss}, z_{ss}) = \big(\mathbf{1}\mathrm{x}_\star, \gamma\nabla\mathbf{F}(\mathbf{1}\mathrm{x}_\star)\big).$$

Thus, we have shown that the equilibrium reachable by the solutions of (13) with z originating in the set Z_0 is necessarily optimal, in the sense that all estimates are equal to the same optimal solution $\mathrm{x}_\star$ of Problem (1). The previous analysis results from the additional steady-state constraints introduced by the servomechanism (13b). Finally, we notice that, in view of (14), the steady-state control action u_{ss} satisfies

$$u_{ss} = z_{ss} = \gamma\nabla\mathbf{F}(x_{ss}) = -\gamma\left(\tfrac{1}{N}\mathbf{1}\mathbf{1}^\top - I\right)\nabla\mathbf{F}(x_{ss}),$$

where we exploited the optimality of x_{ss} which implies $\mathbf{1}^\top\nabla\mathbf{F}(x_{ss}) = 0$. This shows that u_{ss} perfectly balances out the mismatch term of (10). Hence, **R1** holds.

Remark 3.1: As a special case, when both R and C are doubly stochastic, the previous steady-state constraints become

$$z_{ss} - \gamma\nabla\mathbf{F}(x_{ss}) \in \operatorname{span}(\mathbf{1}), \quad 0 = \mathbf{1}^\top(\nabla\mathbf{F}(x_{ss}) + z_{ss}),$$

which, in turn, imply that z_{ss} has no component along in the consensual direction $\mathbf{1}$. We also emphasize that the first condition does not imply that $\nabla\mathbf{F}(x) \in \operatorname{span}(\mathbf{1})$.

An alternative characterization in terms of transfer functions of (13) is possible setting $\gamma\nabla\mathbf{F}$ as input and x as output, see, e.g., [29]. For undirected networks, this approach shows a zero-pole cancellation in 1 and the presence of $N - 1$ zeros in 1 (see also (20)).

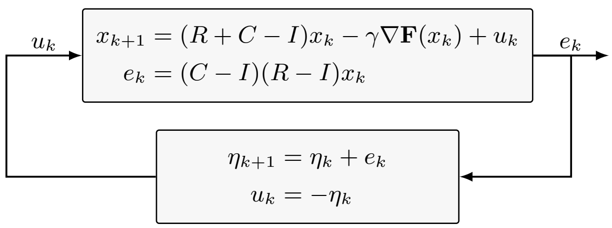

Fig. 2. Block diagram of (19) highlighting the integral action embedded in the Gradient Tracking algorithm.

C. On the Equivalence Between (3) and (13)

We now show that the obtained system (13) with the initialization (12) is equivalent to the Gradient Tracking algorithm in its original form (3). For, by following [2], we consider the change of coordinates

$$z \mapsto s := \nabla \mathbf{F}(x) - \gamma^{-1} z.$$

In these new coordinates, (13) reads as

$$x_{k+1} = Rx_k + \gamma s_k \tag{15a}$$
$$s_{k+1} = Cs_k + \nabla \mathbf{F}(x_{k+1}) - \nabla \mathbf{F}(x_k), \tag{15b}$$

which is precisely (3) written in compact form.

Moreover, in the new coordinates, the initialization constraint induced by the choice (12) reads as

$$0 = \mathbf{1}^\top (\nabla \mathbf{F}(x_0) - s_0) = \sum_{i=1}^{N} \big(s_{i,0} - \nabla f_i(x_{i,0})\big), \tag{16}$$

which shows that the initialization constraint (4) is a possible choice for which (16) holds.

The coordinates (x, z), originally proposed in [2], will be referred to as the "canonical coordinates". A noteworthy difference with the original coordinates (x, s) is that the former is expressed in a usual causal form (all the quantities on the right hand-side depend on the present time k). Moreover, the initialization (12) does not require any initial measurement ($z_{i,0} = 0$ for all $i = 1, \dots, N$ is a feasible, fully decentralized initialization), hence conferring on the algorithm robustness with respect to possible uncertainties in the computations of ∇f_i. This robustness property does not hold for the Gradient Tracking (3) in the original coordinates (x, s), since the initialization (4) requires the computation of $\nabla f_i(x_{i,0})$, and any error or uncertainty in it causes a persistent bias in the estimates.

D. The Gradient Tracking Embeds an Integral Action

Before moving to the stability analysis carried out in Section IV, we show that the Gradient Tracking algorithm embeds an *integral action* processing the *error signal*

$$e := (C - I)(R - I)x, \tag{17}$$

which, in turn, is closely related to the consensus error $(R - I)x$. This makes an interesting connection with output regulation theory [34] and with the concept of internal model [34], [35].

With (13b) in mind, change coordinates according to

$$z \mapsto \eta := (C - I)x - z. \tag{18}$$

In the new coordinates, (13b) (and, thus, the Gradient Tracking (3)) reads as follows

$$x_{k+1} = (R + C - I)x_k - \gamma \nabla \mathbf{F}(x_k) - \eta_k \tag{19a}$$
$$\eta_{k+1} = \eta_k + \underbrace{(C - I)(R - I)x_k}_{e_k}. \tag{19b}$$

The subsystem (19b) is an integrator processing the regulation error (17). In control-theoretic terms, System (19) can be interpreted as a closed-loop system involving a "plant" (9) controlled by a *proportional-integral (PI)* control policy $u = (C - I)x - \eta$, where $(C - I)x$ is the proportional term and $-\eta$ the integral one. A graphical representation is given in Figure 2.

Furthermore, we underline that, by denoting

$$\tilde{y} := x - \mathbf{1}\mathrm{x}_\star, \quad \tilde{u} := -\gamma \nabla \mathbf{F}(\tilde{y} + \mathbf{1}\mathrm{x}_\star) + (C - I)(\tilde{y} + \mathbf{1}\mathrm{x}_\star),$$

and changing coordinates from x to $\tilde{y}$, we obtain

$$\begin{aligned} \tilde{y}_{k+1} &= R\tilde{y}_k - \eta_k + \tilde{u}_k \\ \eta_{k+1} &= \eta_k + (C - I)(R - I)\tilde{y}_k, \end{aligned} \tag{20}$$

which, seen as a system with input $\tilde{u}$ and output $\tilde{y}$, is in "normal form" [34, Chapter 9]. Moreover, its *zero dynamics* is precisely the integrator

$$\eta_{k+1} = \eta_k,$$

so as (20) is not minimum-phase. Finally, we observe that the value of η for which $\tilde{y} = 0$ is

$$\eta^\circ := -\nabla \mathbf{F}(\mathbf{1}\mathrm{x}_\star) + (C - I)\mathbf{1}\mathrm{x}_\star.$$

In turn, this underlines that the optimality constraint **P2** comes from the condition $\mathbf{1}^\top \eta^\circ = 0$ that, in view of (18), and as explained in the previous section, comes from the initialization $\mathbf{1}^\top z_0 = 0$ (or, also, from the original condition (4)).

We conclude this section by pointing out that the algorithm expressed in normal form is not implementable in a distributed way since the matrix $(C - I)(R - I)$ involves two-hop communication. Hence, the tracker subsystem of the Gradient Tracking algorithm, either in original (15b) or in canonical (13b) coordinates, can be interpreted as a distributed implementation of a PI controller. In this respect, we stress that the aforementioned integral action is associated to the eigenvalue in 1 of the z dynamics (13b), and not of the x dynamics (13a), which may or may not have an eigenvalue in 1 as well.

IV. Stability Analysis

In this section, we consider System (13), representing the Gradient Tracking algorithm in canonical coordinates, and we study stability and attractiveness of the unique equilibrium $(x_{\mathrm{ss}}, z_{\mathrm{ss}}) := (\mathbf{1}\mathrm{x}_\star, \gamma \nabla \mathbf{F}(\mathbf{1}\mathrm{x}_\star))$ reachable from Z_0 as introduced in Section III-B. In particular, we prove that such equilibrium is attractive, in fact exponentially asymptotically stable, provided that $z_0 \in Z_0$ with Z_0 defined as in (12). We also provide an estimate of the (linear) convergence rate. In doing so, we take advantage of the control-theoretic framework presented in the previous sections and provide a Lyapunov-based analysis.

As already mentioned, we assume for simplicity that $d = 1$. However, we remark that this can be done without loss of generality, since all the arguments used in the proof can be directly applied, component-wise, to the multidimensional case where $d > 1$, the only difference being that *(i)* R and C must be replaced by $R \otimes I_d$ and $C \otimes I_d$ ($\otimes$ denotes the Kronecker product), and *(ii)* the dimension of the matrices introduced in the following must be consistently adapted.

A. Main Result

Consider the "error" coordinates

$$\begin{bmatrix} x_k \\ z_k \end{bmatrix} \longmapsto \begin{bmatrix} \tilde{x}_k \\ \tilde{z}_k \end{bmatrix} := \begin{bmatrix} x_k - \mathbf{1}\mathrm{x}_\star \\ z_k - \gamma \nabla \mathbf{F}(\mathbf{1}\mathrm{x}_\star) \end{bmatrix},$$

and further change coordinates as $\tilde{z}_k \longmapsto (S^\top \tilde{z}_k, \mathbf{1}^\top \tilde{z}_k / N)$ with $S \in \mathbb{R}^{N\times(N-1)}$ a matrix whose columns form an orthonormal basis for the orthogonal complement of $\operatorname{span}(\mathbf{1})$. Thus, S satisfies

$$S^\top \mathbf{1} = 0, \quad S^\top S = I_{N-1}, \quad I_N = SS^\top + \tfrac{1}{N}\mathbf{1}\mathbf{1}^\top. \tag{21}$$

As C in (13b) is column stochastic, and since optimality of $\mathrm{x}_\star$ implies $\mathbf{1}^\top \nabla \mathbf{F}(\mathbf{1}\mathrm{x}_\star) = 0$, it follows that $\mathbf{1}^\top \tilde{z}_{k+1} = \mathbf{1}^\top \tilde{z}_k = \mathbf{1}^\top \tilde{z}_0 = \mathbf{1}^\top z_0$ for all $k \in \mathbb{N}$. Therefore, attractiveness of the optimal equilibrium $(x_{\rm ss}, z_{\rm ss})$ for (13) with $z_0 \in Z_0$ is implied by global attractiveness of the origin of the "reduced" subsystem $\xi_k := (\tilde{x}_k, S^\top \tilde{z}_k)$, whose dynamics reads

$$\begin{aligned} \xi_{k+1} &= \underbrace{\begin{bmatrix} R & S \\ 0 & S^\top C S \end{bmatrix}}_{F} \xi_k - \gamma \underbrace{\begin{bmatrix} I_N \\ S^\top (C - I) \end{bmatrix}}_{G} u_k \\ u_k &= \nabla \mathbf{F}\Big(\underbrace{\begin{bmatrix} I_N & 0_{N\times(N-1)} \end{bmatrix}}_{H} \xi_k + \mathbf{1}\mathrm{x}_\star \Big) - \nabla \mathbf{F}(\mathbf{1}\mathrm{x}_\star). \end{aligned} \tag{22}$$

Therefore, from now on we focus on (22) only.

We first introduce some quantities instrumental for the technical results that follow. Let $V \in \mathbb{R}^{(2N-1)\times(2N-1)}$ be such that $FV = VJ$, where $J := \operatorname{diag}(1, J_2)$ is a Jordan form of F. Notice that J_2 is a Schur matrix. As $(\mathbf{1}_N, 0)$ is an eigenvector of F associated with the simple eigenvalue 1, we can take without loss of generality

$$V = \begin{bmatrix} \mathbf{1} & V_{21} \\ 0 & V_{22} \end{bmatrix}, \qquad V^{-1} = \begin{bmatrix} \pi_R^\top & \beta^\top \\ K_{21} & K_{22} \end{bmatrix}, \tag{23}$$

for some $\beta \in R^{N-1}$, some $K_{21} \in \mathbb{R}^{(N-1)\times N}$ satisfying $K_{21}\mathbf{1} = 0$, and with $\pi_R \in \mathbb{R}^N$ as in (2).

Let $\Pi := \operatorname{diag}(1, \Pi_2)$, where $\Pi_2 = \Pi_2^\top > 0$ is the unique solution[7] of $J_2^\top \Pi_2 J_2 - \Pi_2 = -V_{21}^\top SS^\top V_{21} - V_{22}^\top V_{22}$. Then, $\Pi = \Pi^\top > 0$ satisfies $J^\top \Pi J - \Pi = -V^\top \operatorname{diag}(SS^\top, I_{N-1})V$. Therefore, the matrix $P := V^{-\top}\Pi V^{-1}$ satisfies $P = P^\top > 0$ and solves

$$F^\top P F - P = -\begin{bmatrix} SS^\top & 0 \\ 0 & I_{N-1} \end{bmatrix}. \tag{24}$$

[7]Existence and uniqueness follow by the fact that J_2 is Schur and, being V_{22} invertible in view of (23), $-V_{21}^\top SS^\top V_{21} - V_{22}^\top V_{22}$ is symmetric and negative definite.

Lemma 4.1: Let F and G be as in (22) and P as in (24). Then, with $\theta := \pi_C^\top \pi_R > 0$,

$$G^\top P F \begin{bmatrix} \frac{1}{N}\mathbf{1}\mathbf{1}^\top \\ 0 \end{bmatrix} = \theta H \begin{bmatrix} \frac{1}{N}\mathbf{1}\mathbf{1}^\top \\ 0 \end{bmatrix} = \frac{\theta}{N}\mathbf{1}\mathbf{1}^\top. \tag{25}$$

The proof of Lemma 4.1 is postponed to Section IV-C.

Remark 4.2: An interesting interpretation of Lemma 4.1 is related to passivity theory. Specifically, Equation (25) can be seen as the input-output constraint imposed by the Kalman-Yakubovich-Popov conditions [36, Lemma 3]. It implies that System (22) with output matrix θH, when restricted to $\operatorname{Im}\operatorname{col}\left(\frac{1}{N}\mathbf{1}\mathbf{1}^\top, 0\right)$, behaves as a passive system.

Finally, define the quadratic Lyapunov candidate $\ell(\xi) := \xi^\top P \xi$. Then, the following theorem establishes the main result of the section.

Theorem 4.3: Suppose that Assumptions 2.1 and 2.2 hold. Then, there exist $\gamma^\star, c > 0$ such that, for every solution ξ of (22) obtained with $\gamma \in (0, \gamma^\star)$, the following holds

$$\ell(\xi_{k+1}) - \ell(\xi_k) \leq -\frac{1}{2}\left(\gamma c \left|\tfrac{1}{N}\mathbf{1}^\top \tilde{x}_k\right|^2 + \left|S^\top \tilde{x}_k\right|^2 + \left|S^\top \tilde{z}_k\right|^2\right) \tag{26}$$

for all $k \in \mathbb{N}$.

The proof of Theorem 4.3 is postponed to Section IV-B, while hereafter we underline two important consequences. First, we observe that Theorem 4.3 implies exponential convergence of (13) to the optimal equilibrium $(\mathbf{1}\mathrm{x}_\star, \gamma\nabla\mathbf{F}(\mathbf{1}\mathrm{x}_\star))$ provided that the initialization $z_0 \in Z_0$ holds. Indeed, since $|\xi_k|^2 = |\tilde{x}_k|^2 + |S^\top \tilde{z}_k|^2$ and $|\tilde{x}_k|^2 = |\mathbf{1}\mathbf{1}^\top \tilde{x}_k/N + SS^\top \tilde{x}_k|^2 = |\mathbf{1}^\top \tilde{x}_k|^2/N + |S^\top \tilde{x}_k|^2$, then

$$\begin{aligned} \gamma c|\mathbf{1}^\top \tilde{x}_k/N|^2 + |S^\top \tilde{x}_k|^2 + |S^\top \tilde{z}_k|^2 &\geq \min\{1, \gamma c/N\}|\xi_k|^2 \\ &\geq \frac{\min\{1, \gamma c/N\}}{\max \sigma(P)} \ell(\xi_k) \end{aligned}$$

with $\sigma(P)$ denoting the spectrum of P. Hence, (26) and positive definiteness of ℓ imply

$$\ell(\xi_{k+1}) \leq \mu^2 \ell(\xi_k)$$

with $\mu^2 := \max\{0, 1 - \min\{1, \gamma c/N\}/(2\max\sigma(P))\}$. Thus, by induction, and by using $|\tilde{x}_k| \leq |\xi_k|$, we obtain

$$|\tilde{x}_k| \leq \alpha\mu^k |\xi_0|, \quad \forall k \in \mathbb{N} \tag{27}$$

with $\alpha := \sqrt{\max\sigma(P)/\min\sigma(P)}$. Notice that (27) also represents the explicit expression for the linear convergence rate of the Gradient Tracking algorithm.

Second, we observe that, in view of the initialization $z_0 \in Z_0$, and since ξ leaves out the component $\mathbf{1}^\top \tilde{z}$, the equilibrium $(x_{\rm ss}, z_{\rm ss}) = (\mathbf{1}\mathrm{x}_\star, \gamma\nabla\mathbf{F}(\mathbf{1}\mathrm{x}_\star))$ is not Lyapunov stable in the ordinary sense for (13). Nevertheless, (26) implies that it is stable "modulo $z_0 \in Z_0$" in the following sense: for every $\epsilon > 0$, there exists $\delta_\epsilon > 0$, such that every solution of (13) with $z_0 \in Z_0$ and $|x_0 - x_{\rm ss}| + |z_0 - z_{\rm ss}| \leq \delta_\epsilon$ satisfies $|x_k - x_{\rm ss}| + |z_k - z_{\rm ss}| \leq \epsilon$ for all $k \in \mathbb{N}$.

Finally, we remark that, since the variable $\mathbf{1}^\top \tilde{z}$ is marginally stable, then the Gradient Tracking is not robust with respect to additive perturbations with nonzero mean value acting on the z dynamics (13b). A counterexample showing this fragility is reported in [37, Sec. V].

B. Proof of Theorem 4.3

For every solution ξ of (22), and every $k \in \mathbb{N}$, the increment $\Delta_k := \ell(\xi_{k+1}) - \ell(\xi_k)$ satisfies

$$\begin{aligned}\Delta_k = \xi_k^\top (F^\top P F - P)\xi_k \quad &(28a)\\ - 2\gamma u_k^\top G^\top P F \xi_k \quad &(28b)\\ + \gamma^2 u_k^\top G^\top P G u_k. \quad &(28c)\end{aligned}$$

In light of (21), we can decompose ξ as

$$\xi = \begin{bmatrix} \frac{1}{N}\mathbf{1}\mathbf{1}^\top \tilde{x} + SS^\top \tilde{x} \\ S^\top \tilde{z}\end{bmatrix} = \underbrace{\begin{bmatrix} \frac{1}{N}\mathbf{1}\mathbf{1}^\top \tilde{x} \\ 0\end{bmatrix}}_{\xi^{\shortparallel}} + \underbrace{\begin{bmatrix} SS^\top \tilde{x} \\ S^\top \tilde{z}\end{bmatrix}}_{\xi^\perp}, \quad (29)$$

where the term $\xi^{\shortparallel}$ represents the "average component" of ξ, while $\xi^\perp$ the deviation from it.

In view of (24), and since $S^\top \mathbf{1} = 0$ by construction, the quadratic term (28a) satisfies

$$\begin{aligned}&\xi_k^\top (F^\top P F - P)\xi_k \\ &= \begin{bmatrix} SS^\top \tilde{x}_k \\ S^\top \tilde{z}_k\end{bmatrix}^\top \begin{bmatrix} -SS^\top & 0 \\ 0 & -I\end{bmatrix}\begin{bmatrix} SS^\top \tilde{x}_k \\ S^\top \tilde{z}_k\end{bmatrix} \\ &+ \left(\begin{bmatrix} \frac{1}{N}\mathbf{1}\mathbf{1}^\top \tilde{x}_k \\ 0\end{bmatrix} + 2\begin{bmatrix} SS^\top \tilde{x}_k \\ S^\top \tilde{z}_k\end{bmatrix}\right)^\top \underbrace{\begin{bmatrix} -SS^\top & 0 \\ 0 & -I\end{bmatrix}\begin{bmatrix} \frac{1}{N}\mathbf{1}\mathbf{1}^\top \tilde{x}_k \\ 0\end{bmatrix}}_{0} \\ &= -\left|S^\top \tilde{x}_k\right|^2 - \left|S^\top \tilde{z}_k\right|^2. \end{aligned} \quad (30)$$

We now focus on the second term (28b). First, we add $\pm \nabla \mathbf{F}\left(H\xi_k^{\shortparallel} + \mathbf{1}\mathrm{x}_\star\right)$ in the definition of u_k in (22), obtaining

$$\begin{aligned}u_k = \nabla \mathbf{F}\left(H\xi_k^{\shortparallel} + \mathbf{1}\mathrm{x}_\star\right) - \nabla \mathbf{F}(\mathbf{1}\mathrm{x}_\star) \\ + \nabla \mathbf{F}(H\xi_k + \mathbf{1}\mathrm{x}_\star) - \nabla \mathbf{F}\left(H\xi_k^{\shortparallel} + \mathbf{1}\mathrm{x}_\star\right).\end{aligned}$$

Under Assumption 2.2, we have

$$\begin{aligned}&\left(\nabla \mathbf{F}\left(H\xi_k^{\shortparallel} + \mathbf{1}\mathrm{x}_\star\right) - \nabla \mathbf{F}(\mathbf{1}\mathrm{x}_\star)\right)^\top H\xi_k^{\shortparallel} \\ &= \left(\sum_{i=1}^N \nabla f_i\left(\tfrac{1}{N}\mathbf{1}^\top \tilde{x}_k + \mathrm{x}_\star\right)\right)^\top \left(\tfrac{1}{N}\mathbf{1}^\top \tilde{x}_k\right) \\ &\geq \frac{c_0}{N^2}\left|\mathbf{1}^\top \tilde{x}_k\right|^2, \end{aligned} \quad (31)$$

for some c_0 independent of γ, and for all $k \in \mathbb{N}$. Moreover, under Assumption 2.1 the following hold for all $k \in \mathbb{N}$

$$\begin{aligned}\left|\nabla \mathbf{F}\left(H\xi_k^{\shortparallel} + \mathbf{1}\mathrm{x}_\star\right) - \nabla \mathbf{F}(\mathbf{1}\mathrm{x}_\star)\right| \leq c_1 |\xi_k^{\shortparallel}| \\ \left|\nabla \mathbf{F}(H\xi_k + \mathbf{1}\mathrm{x}_\star) - \nabla \mathbf{F}\left(H\xi_k^{\shortparallel} + \mathbf{1}\mathrm{x}_\star\right)\right| \leq c_1 |\xi_k^\perp|,\end{aligned} \quad (32)$$

with c_1 being the Lipschitz constant of $\nabla \mathbf{F}$. Therefore, by using Lemma 4.1 and the inequalities (31)-(32), we can bound the term (28b) as follows

$$\begin{aligned}-2\gamma u_k^\top G^\top P F \xi_k &= -2\gamma u_k^\top G^\top P F \xi_k^{\shortparallel} - 2\gamma u_k^\top G^\top P F \xi_k^\perp \\ &\leq -2\gamma \frac{\theta}{N} u_k^\top H \xi_k^{\shortparallel} + 2\gamma \left|G^\top P F\right| |u_k| \left|\xi_k^\perp\right| \\ &\leq -\gamma \frac{2\theta c_0}{N^2}\left|\mathbf{1}^\top \tilde{x}_k\right|^2 + \gamma \frac{2|H| c_1}{N} |\xi_k^\perp||\xi_k^{\shortparallel}| \\ &\quad + 2\gamma \left|G^\top P F\right| |u_k| \left|\xi_k^\perp\right| \\ &\leq -\gamma c_2 \left|\tfrac{1}{N}\mathbf{1}^\top \tilde{x}_k\right|^2 + \gamma c_3 |\xi_k^\perp||\xi_k^{\shortparallel}| + \gamma c_4 \left|\xi_k^\perp\right|^2 \end{aligned} \quad (33)$$

with $c_2 := 2\theta c_0$, $c_3 := 2\theta |H| c_1/N + 2\left|G^\top P F\right| c_1$, and $c_4 := 2\left|G^\top P F\right| c_1$.

Finally, under Assumption 2.1, the last term (28c) can be bounded as follows

$$\gamma^2 u_k^\top G^\top P G u_k \leq \gamma^2 \left|G^\top P G\right| |u_k|^2 \leq \gamma^2 c_5 (|\xi_k^\perp|^2 + |\xi_k^{\shortparallel}|^2), \quad (34)$$

with $c_5 := \left|G^\top P G\right| c_1^2$.

In view of (30), (33) and (34), we can upper bound the increment Δ_k in (28) as

$$\begin{aligned}\Delta_k &\leq -\left|S^\top \tilde{x}_k\right|^2 - \left|S^\top \tilde{z}_k\right|^2 - \gamma c_2 \left|\tfrac{1}{N}\mathbf{1}^\top \tilde{x}_k\right|^2 \\ &\quad + \gamma c_3 |\xi_k^\perp||\xi_k^{\shortparallel}| + \gamma c_4 \left|\xi_k^\perp\right|^2 + \gamma^2 c_5 (|\xi_k^\perp|^2 + |\xi_k^{\shortparallel}|^2) \\ &\leq \left(\gamma \frac{c_2}{4} + \gamma^2 c_5 - \gamma c_2\right)\left|\tfrac{1}{N}\mathbf{1}^\top \tilde{x}_k\right|^2 \\ &\quad + \left(\gamma \frac{c_3^2}{c_2} + \gamma^2 c_5 + \gamma c_4 - 1\right)\left(\left|S^\top \tilde{x}_k\right|^2 + \left|S^\top \tilde{z}_k\right|^2\right),\end{aligned}$$

where, in the last inequality, we have used $|\xi_k^\perp|^2 = \left|S^\top \tilde{x}_k\right|^2 + \left|S^\top \tilde{z}_k\right|^2$ and the Young's inequality

$$2\gamma c_3 \left|\xi_k^{\shortparallel}\right| \left|\xi_k^\perp\right| \leq \gamma c_3 \left(\epsilon \left|\xi_k^{\shortparallel}\right|^2 + \epsilon^{-1}\left|\xi_k^\perp\right|^2\right)$$

with $\epsilon = c_2/(2c_3)$.

Therefore, for all $\gamma \in (0, \gamma^\star)$, where

$$\gamma^\star := \min\left\{\frac{c_2}{4(c_3^2 + c_4 c_2)}, \frac{1}{2\sqrt{c_5}}, \frac{c_2}{4c_5}\right\}$$

we get (26) for all $k \in \mathbb{N}$, with $c := c_2$. □

C. Proof of Lemma 4.1

Since $\pi_R^\top \mathbf{1} = 1$ and $K_{21}\mathbf{1} = 0$, we have

$$\begin{aligned}G^\top P F \begin{bmatrix} \frac{1}{N}\mathbf{1}\mathbf{1}^\top \\ 0\end{bmatrix} &= G^\top V^{-\top} \Pi J V^{-1} \begin{bmatrix} \frac{1}{N}\mathbf{1}\mathbf{1}^\top \\ 0\end{bmatrix} \\ &= G^\top V^{-\top} \begin{bmatrix} 1 & 0 \\ 0 & \Pi_2 J_2\end{bmatrix}\begin{bmatrix} \frac{1}{N}\mathbf{1}^\top \\ 0\end{bmatrix} \\ &= \begin{bmatrix} I & (C - I)^\top S\end{bmatrix}\begin{bmatrix} \pi_R \\ \beta\end{bmatrix}\frac{1}{N}\mathbf{1}^\top \\ &= \left(\pi_R + (C - I)^\top S\beta\right)\frac{1}{N}\mathbf{1}^\top. \end{aligned} \quad (35)$$

From (23), we get

$$\begin{aligned}F^\top V^{-\top} &= \begin{bmatrix} R^\top & 0 \\ S^\top & S^\top C^\top S\end{bmatrix}\begin{bmatrix} \pi_R & K_{21}^\top \\ \beta & K_{22}^\top\end{bmatrix} \\ &= \begin{bmatrix} \star & \star \\ S^\top(\pi_R + C^\top S\beta) & \star\end{bmatrix}\end{aligned}$$

where $\star$ is a placeholder for terms whose exact expression is not relevant. On the other hand, it also holds

$$F^\top V^{-\top} = (V^{-1}F)^\top = (JV^{-1})^\top = \begin{bmatrix} \pi_R & \star \\ \star & \star\end{bmatrix}.$$

In view of the properties of S (see (21)), equating both previous expressions for $F^\top V^{-\top}$ yields

$$\begin{aligned}&S^\top(\pi_R + C^\top S\beta) = \beta = S^\top S\beta \\ &\implies S^\top(\pi_R + (C - I)^\top S\beta) = 0 \\ &\implies \pi_R + (C - I)^\top S\beta \in \ker S^\top\end{aligned}$$

$$\implies \exists\, \theta \in \mathbb{R} \ \text{ s.t. } \ \pi_R + (C - I)^\top S\beta = \theta \mathbf{1}.$$

From the latter implication and (35), we obtain (25). The claim of the lemma then follows from (2b), since

$$\theta = \pi_C^\top(\theta\mathbf{1}) = \pi_C^\top(\pi_R + (C - I)^\top S\beta) = \pi_C^\top \pi_R.$$

V. Conclusions

We revisited the Gradient Tracking algorithm under the lens of control theory. We found that the design of the algorithm can be equivalently posed as a control problem, whose solution consists in a distributed integral action applied to an error signal related to the consensus error. With this novel interpretation at reach, we provided an alternative convergence proof for the Gradient Tracking, also establishing Lyapunov stability of the equilibrium corresponding to the optimal solution of the consensus optimization problem.

References

[1] S. Bennett, "A brief history of automatic control," *IEEE Control Syst. Mag.*, vol. 16, no. 3, pp. 17–25, 1996.

[2] M. Bin, I. Notarnicola, L. Marconi, and G. Notarstefano, "A system theoretical perspective to gradient-tracking algorithms for distributed quadratic optimization," in *IEEE 58th Conference on Decision and Control (CDC)*, 2019, pp. 2994–2999.

[3] D. Varagnolo, F. Zanella, A. Cenedese, G. Pillonetto, and L. Schenato, "Newton-Raphson consensus for distributed convex optimization," *IEEE Trans. Autom. Control*, vol. 61, no. 4, pp. 994–1009, 2016.

[4] P. Di Lorenzo and G. Scutari, "Next: In-network nonconvex optimization," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 2, pp. 120–136, 2016.

[5] A. Nedić, A. Olshevsky, and W. Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," *SIAM J. Optim.*, vol. 27, no. 4, pp. 2597–2633, 2017.

[6] G. Qu and N. Li, "Harnessing smoothness to accelerate distributed optimization," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 3, pp. 1245–1260, 2018.

[7] J. Xu, S. Zhu, Y. C. Soh, and L. Xie, "Convergence of asynchronous distributed gradient methods over stochastic networks," *IEEE Trans. Autom. Control*, vol. 63, no. 2, pp. 434–448, 2018.

[8] C. Xi, R. Xin, and U. A. Khan, "ADD-OPT: Accelerated distributed directed optimization," *IEEE Trans. Autom. Control*, vol. 63, no. 5, pp. 1329–1339, 2018.

[9] R. Xin and U. A. Khan, "A linear algorithm for optimization over directed graphs with geometric convergence," *IEEE Contr. Syst. Lett.*, vol. 2, no. 3, pp. 315–320, 2018.

[10] G. Scutari and Y. Sun, "Distributed nonconvex constrained optimization over time-varying digraphs," *Math. Program.*, vol. 176, no. 1-2, pp. 497–544, 2019.

[11] S. Pu, W. Shi, J. Xu, and A. Nedic, "Push-pull gradient methods for distributed optimization in networks," *IEEE Trans. Autom. Control*, vol. 66, no. 1, pp. 1–16, 2021.

[12] Y. Tian, Y. Sun, and G. Scutari, "Achieving linear convergence in distributed asynchronous multiagent optimization," *IEEE Trans. Autom. Control*, vol. 65, no. 12, pp. 5264–5279, 2020.

[13] B. Johansson, T. Keviczky, M. Johansson, and K. H. Johansson, "Subgradient methods and consensus algorithms for solving convex optimization problems," in *IEEE 47th Conference on Decision and Control (CDC)*, 2008, pp. 4185–4190.

[14] A. Nedić and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. Autom. Control*, vol. 54, no. 1, pp. 48–61, 2009.

[15] A. Nedić, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Trans. Autom. Control*, vol. 55, no. 4, pp. 922–938, 2010.

[16] R. Xin and U. A. Khan, "Distributed heavy-ball: A generalization and acceleration of first-order methods with gradient tracking," *IEEE Trans. Autom. Control*, vol. 65, no. 6, pp. 2627–2633, 2020.

[17] G. Qu and N. Li, "Accelerated distributed Nesterov gradient descent," *IEEE Trans. Autom. Control*, vol. 65, no. 6, pp. 2566–2581, 2020.

[18] S. Pu and A. Nedić, "Distributed stochastic gradient tracking methods," *Mathematical Programming*, vol. 187, no. 1, pp. 409–457, 2021.

[19] S. S. Kia, B. Van Scoy, J. Cortés, R. A. Freeman, K. M. Lynch, and S. Martínez, "Tutorial on dynamic average consensus: The problem, its applications, and the algorithms," *IEEE Control Syst. Mag.*, vol. 39, no. 3, pp. 40–72, 2019.

[20] J. Wang and N. Elia, "Control approach to distributed optimization," in *IEEE Allerton Conference on Communication, Control, and Computing (Allerton)*, 2010, pp. 557–561.

[21] ——, "A control perspective for centralized and distributed convex optimization," in *IEEE 50th Conference on Decision and Control and European Control Conference (CDC)-(ECC)*, 2011, pp. 3800–3805.

[22] G. Droge, H. Kawashima, and M. B. Egerstedt, "Continuous-time proportional-integral distributed optimisation for networked systems," *Journal of Control and Decision*, vol. 1, no. 3, pp. 191–213, 2014.

[23] C. Ebenbauer, S. Michalowsky, V. Grushkovskaya, and B. Gharesifard, "Distributed optimization over directed graphs with the help of Lie brackets," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 15 343–15 348, 2017.

[24] T. Hatanaka, N. Chopra, T. Ishizaki, and N. Li, "Passivity-based distributed optimization with communication delays using PI consensus algorithm," *IEEE Trans. Autom. Control*, vol. 63, no. 12, pp. 4421–4428, 2018.

[25] C.-Y. Chang, M. Colombino, J. Cortés, and E. Dall'Anese, "Saddle-flow dynamics for distributed feedback-based optimization," *IEEE Contr. Syst. Lett.*, vol. 3, no. 4, pp. 948–953, 2019.

[26] P. Cisneros-Velarde, S. Jafarpour, and F. Bullo, "Distributed and time-varying primal-dual dynamics via contraction analysis," *IEEE Trans. Autom. Control*, 2021.

[27] A. Sundararajan, B. Hu, and L. Lessard, "Robust convergence analysis of distributed optimization algorithms," in *IEEE Allerton Conference on Communication, Control, and Computing*, 2017, pp. 1206–1212.

[28] D. Jakovetić, D. Bajović, J. Xavier, and J. M. Moura, "Primal–dual methods for large-scale and distributed convex optimization and data analytics," *Proceedings of the IEEE*, vol. 108, no. 11, pp. 1923–1938, 2020.

[29] A. Sundararajan, B. Van Scoy, and L. Lessard, "A canonical form for first-order distributed optimization algorithms," in *IEEE American Control Conference (ACC)*, 2019, pp. 4075–4080.

[30] ——, "Analysis and design of first-order distributed optimization algorithms over time-varying graphs," *IEEE Trans. Control Netw. Syst.*, vol. 7, no. 4, pp. 1597–1608, 2020.

[31] G. Carnevale, M. Bin, I. Notarnicola, L. Marconi, and G. Notarstefano, "Enhanced gradient tracking algorithms for distributed quadratic optimization via sparse gain design," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 2696–2701, 2020.

[32] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge University Press, 2012.

[33] D. P. Bertsekas, *Nonlinear programming*. Athena Scientific, 1999.

[34] A. Isidori, *Lectures in Feedback Design for Multivariable Systems*. Springer, 2017.

[35] M. Bin, J. Huang, A. Isidori, L. Marconi, M. Mischiati, and E. Sontag, "Internal Models in Control, Bioengineering, and Neuroscience," *Annual Rev. Control Robot. Auton. Syst.*, vol. 5, no. 1, 2022.

[36] L. Hitz and B. Anderson, "Discrete positive-real functions and their application to system stability," in *Proceedings of the Institution of Electrical Engineers*, vol. 116. IET, 1969, pp. 153–155.

[37] M. Bin, I. Notarnicola, and T. Parisini, "Stability, linear convergence, and robustness of the Wang-Elia algorithm for distributed consensus optimization," 2022. [Online]. Available: https://arxiv.org/abs/2204.10030