2021 60th IEEE Conference on Decision and Control (CDC)
December 13-15, 2021. Austin, Texas

Linear Convergence of Distributed Mirror Descent with Integral Feedback for Strongly Convex Problems

Youbang Sun and Shahin Shahrampour, *Senior Member, IEEE*

2021 60th IEEE Conference on Decision and Control (CDC) | 978-1-6654-3659-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/CDC45484.2021.9682809

Abstract— **Distributed optimization often requires finding the minimum of a global objective function written as a sum of local functions. A group of agents work collectively to minimize the global function. We study a continuous-time decentralized mirror descent algorithm that uses purely local gradient information to converge to the global optimal solution. The algorithm enforces consensus among agents using the idea of integral feedback. Recently, the asymptotic convergence of this algorithm was studied for when the global function is strongly convex but local functions are convex. Using control theoretical tools, in this work, we prove (theoretically) that the algorithm indeed achieves local exponential convergence. We also provide a numerical experiment on a real data-set as a validation of the convergence speed of our algorithm.**

I. Introduction

Distributed gradient-based optimization is well-studied in the literature. Generally, the problem is to find the optimal solution for a global objective function that is a sum of local cost functions assigned to various agents. Each agent only has limited knowledge of the global problem, and the agents must work collectively to reach consensus around the optimum for the global objective function. Distributed optimization has applications in distributed resource allocation [1], distributed sensor localization [2], distributed cooperative control [3], social learning [4], and beyond.

Naturally, one of the most fundamental questions in distributed optimization is that whether a distributed algorithm is able to match the performance of its centralized counterpart. The basic idea of gradient descent with local averaging has proven to be a simple yet powerful approach. The seminal work of [5] is a prominent point in case, which shows this approach converges for convex problems using a *diminishing step-size* sequence, which decreases the influence of local gradients and allows all agents to reach consensus. However, as soon as assumptions like *smoothness* and/or *strong convexity* come into play, a diminishing step-size may no longer be optimal in centralized optimization, thereby being a sub-optimal choice for decentralized algorithms as well.

A number of works proposed *gradient tracking*, that uses an additional term to ensure consensus with non-decreasing step-sizes. This line of work includes EXTRA [6] and DEXTRA [7], where we can observe decentralized performances on par with their respective centralized problems. In continuous-time distributed optimization, another approach, termed *integral feedback*, has been used in the literature in a similar spirit. The integral feedback introduces another variable to account for differences between agents and helps the network reach consensus. Examples of recent works adopting this approach include [8]–[11].

However, most of the recent works in distributed gradient-based optimization have focused on gradient descent. Although effective, gradient descent sometimes cannot yield desirable results by not exploiting the geometry of the problem. Mirror descent [12], on the other hand, is widely used in large-scale optimization problems. Mirror descent replaces the Euclidean distance in gradient descent with *Bregman divergence* as the regularizer, and it can be viewed as a more general version of gradient descent. For some of high-dimensional optimization problems, mirror descent can provide significantly faster convergence rates compared to gradient descent [13].

Motivated by the generality of mirror descent, in this work we focus on distributed mirror descent (DMD). Most of prior work on DMD is in discrete time (see e.g., [14]–[18]). With the exception of [16], the works above either use diminishing step-size sequence or multi-communications per round in order to reach consensus. For the same reasons mentioned for gradient descent, a diminishing step-size would not be optimal for strongly convex problems, resulting in slower convergence compared to centralized methods. In this work, we study continuous-time DMD with integral feedback, recently proposed in [19]. The authors focused on a setup where the global objective is strongly convex but the local functions are convex, and they provided asymptotic convergence analysis. In the current work, we use dynamical systems tools (Lyapunov's indirect method) to prove the *local exponential* convergence of DMD with integral feedback, which provides a *theoretical* analysis on the linear convergence observed (empirically) in [19]. We also test our algorithm on a real data-set to show that the proposed algorithm indeed converges exponentially fast (or linearly in log-scale).

We remark that DMD in continuous time has also been studied prior to this work, mostly by focusing on reduction of noise variance in stochastic optimization [20], [21]. [22] also motivates mirror descent using RLC circuits and utilize derivative and integration in the algorithm. The distinction between [22] and the current work includes different assumptions on the objective functions, which yields different convergence results.

II. Problem Formulation

Notation: We let $[n]$ denote the set $\{1, 2, 3, \ldots, n\}$ for

Y. Sun and S. Shahrampour are with the Department of Mechanical and Industrial Engineering at Northeastern University, Boston, MA 02115, USA. {sun.youb;s.shahrampour}@northeastern.edu.

any integer n. $x^\top$ (and $A^\top$) denotes transpose of vector x (and matrix A), respectively. I_d represents identity matrix of size $d \times d$. We let $\mathbb{1}_d$ denote d-dimensional vector of all ones. $\langle x, y\rangle$ denotes the standard inner product between x and y and $\|x\| = \sqrt{\langle x, x\rangle}$ is the Euclidean norm of vector x. $A \otimes B$ represents the Kronecker product of matrices A and B. The i-th element of the vector x is denoted by $[x]_i$, and the ij-th element of the matrix A is denoted by $[A]_{ij}$. We let $det(A)$ denote the determinant of matrix A and use $\text{col}\{v_1, \ldots, v_n\}$ to denote the vector that stacks all vectors v_i for $i \in [n]$. We use $\text{diag}\{a_1, \ldots, a_n\}$ to represent an $n \times n$ diagonal matrix that has the scalar a_i in its i-th diagonal element. We use $Re[\cdot]$ to denote the real part of a complex number. We use 0 to represent the null vector and the null matrix when it is clear from the context.

A. Distributed Optimization

Distributed convex optimization consists of minimizing an objective function $F : \mathbb{R}^d \to \mathbb{R}$. F is written as a sum of local cost functions, denoted by $f_i : \mathbb{R}^d \to \mathbb{R}$ for $i \in [n]$, and the cost function f_i is associated with agent i. The minimization task is as follows

$$\underset{x \in \mathbb{R}^d}{\text{minimize}} \qquad F(x) = \sum_{i=1}^{n} f_i(x). \tag{1}$$

In a distributed optimization setup, agents only have the information about their associated local functions, and the network of agents relies on communication between agents in order to find the solution to the global task presented in (1). We now introduce some assumptions on the local and global functions.

Assumption 1: For any agent $i \in [n]$ in the network, we assume that the local cost function $f_i : \mathbb{R}^d \to \mathbb{R}$ is convex and differentiable.

From this assumption, we can immediately get that the global function F is also convex and differentiable, but we impose an additional assumption on the global cost function as follows.

Assumption 2: The global function $F : \mathbb{R}^d \to \mathbb{R}$ is strongly convex. There exists a unique minimizer for F and the optimal value denoted by $F^\star = F(x^\star)$ exists. The gradients of local functions $\nabla f_i(x)$ are locally continuously differentiable around $x^\star$.

B. Network Settings

The agents form a network, modeled by a simple undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the agents are denoted by nodes $\mathcal{V} = [n]$ and the connection between two agents i and j is captured by the edge $\{i, j\} \in \mathcal{E}$. The neighborhood of agent i is denoted by $\mathcal{N}_i \triangleq \{j \in \mathcal{V} : \{i, j\} \in \mathcal{E}\}$. The agents work collectively to find the optimum of the global cost function, which is the sum of all local cost functions.

Assumption 3: The graph $\mathcal{G}$ is connected, i.e., there exists a path between any two distinct agents $i, j \in \mathcal{V}$. The graph Laplacian is denoted by $\mathcal{L} \in \mathbb{R}^{n \times n}$.

The connectivity assumption implies that $\mathcal{L}$ has a unique null eigenvalue. That is, $\mathcal{L}\mathbb{1}_n = 0$, and $\mathbb{1}_n$ is the only direction (eigenvector) recovering the zero eigenvalue.

C. Mirror Descent

We now provide a brief introduction of centralized mirror descent algorithm and explain the transition from discrete mirror descent (as mentioned in [12]) to a continuous-time setup. Later, in Section II-D we derive the distributed mirror descent updates in continuous time.

In gradient descent method, each iterate can be seen as an optimization problem on a simplified model, constructed by a first order approximation of a function plus a Euclidean regularizer. Mirror descent replaces the Euclidean regularizer with *Bregman divergence*. Bregman divergence is defined with respect to a distance generating function (DGF) $\phi : \mathbb{R}^d \to \mathbb{R}$, as follows

$$\mathcal{D}_\phi(x, x') \triangleq \phi(x) - \phi(x') - \langle \nabla\phi(x'), x - x'\rangle. \tag{2}$$

In discrete time, the mirror descent algorithm with learning rate η is written as

$$\begin{aligned} x^{(k+1)} = \underset{x \in \mathbb{R}^d}{\text{argmin}} \Big\{ & F(x^{(k)}) + \eta \nabla F(x^{(k)})^\top (x - x^{(k)}) \\ & + \mathcal{D}_\phi(x, x^{(k)}) \Big\}. \end{aligned} \tag{3}$$

Bregman divergence is regarded as a more general version of the Euclidean regularizer. When using the Euclidean distance as the Bregman divergence (i.e., $\mathcal{D}_\phi(x, x^{(k)}) = \frac{1}{2}\|x - x^{(k)}\|^2$) we recover gradient descent. Hence, mirror descent is seen as a more general version of gradient descent.

Assumption 4: The distance generating function ϕ is closed, differentiable and μ_ϕ-strongly convex.

Assumption 5: The Hessian of distance generating function, $\nabla^2\phi$, is locally continuously differentiable around the neighborhood of $x^\star$ (the minimizer of F).

The two assumptions above on ϕ are satisfied by some of the commonly used Bregman divergences, such as $\phi(x) = \frac{1}{2}\|x\|^2$, DGF of the Euclidean distance, and the negative entropy function $\phi(x) = \sum_{j=1}^{d} [x]_j \log([x]_j)$, DGF of the Kullback–Leibler divergence.

Now, we introduce an equivalent form of the update above for more convenient analysis. This equivalent form is based on the *convex conjugate* (also known as *Fenchel dual*) of function ϕ, which is denoted by $\phi^\star$ and defined as follows

$$\phi^\star(z) \triangleq \sup_{x \in \mathbb{R}^d} \{\langle x, z\rangle - \phi(x)\}.$$

From the definition, we can derive the the subsequent relationship,

$$z = \nabla\phi(x) \Longleftrightarrow x = \nabla\phi^\star(z).$$

This means $\nabla\phi^\star$ will map the range of z back to $\mathbb{R}^d$. Assumption 4 on the DGF ϕ guarantees the μ_ϕ^{-1}-smoothness property on $\phi^\star$ (see e.g., [23]). Using the definition of $\phi^\star$, the update (3) can be rewritten in the following equivalent form

$$\begin{aligned} z^{(k+1)} &= z^{(k)} - \eta \nabla F(x^{(k)}) \\ x^{(k+1)} &= \nabla\phi^\star(z^{(k+1)}). \end{aligned} \tag{4}$$

Then, the continuous-time update can be obtained by setting η infinitesimally small as follows

$$\begin{aligned} \dot{z} &= -\nabla F(x), \\ x &= \nabla \phi^\star(z), \\ x(0) = x_0, z(0) &= z_0 \text{ with } x_0 = \nabla \phi^\star(z_0), \end{aligned} \tag{5}$$

This setup was studied in [24].

D. Distributed Mirror Descent with Integral Feedback

In this section, we introduce the distributed algorithm for mirror descent shown in (5). Our end goal is to have all agents converge to the global optimum in (1) and reach consensus. Motivated by [8], [9], we use *integral feedback* to get

$$\begin{aligned} \dot{z_i} &= -\nabla f_i(x_i) + \sum_{j\in\mathcal{N}_i}(x_j - x_i) + \int_0^t \sum_{j\in\mathcal{N}_i}(x_j - x_i) \\ x_i &= \nabla \phi^\star(z_i), \end{aligned} \tag{6}$$

with $x_i(0) = x_{i0},\ z_i(0) = z_{i0},$ and $x_{i0} = \nabla\phi^\star(z_{i0})$.

The algorithm only utilizes gradient information of the local costs. The first equation updates the dual variable z_i using gradient information, a consensus term, and the integral feedback. Then, the second equation updates the primal variable by mirroring the dual variable back with function $\phi^\star$. For convenience, we stack vectors from all agents and define the following notation,

$$\begin{aligned} \mathbf{L} &\triangleq \mathcal{L} \otimes I_d \\ \mathbf{x} &\triangleq \text{col}\{x_1, x_2, \dots, x_n\} \\ \mathbf{z} &\triangleq \text{col}\{z_1, z_2, \dots, z_n\}, \\ \nabla\phi^\star(\mathbf{z}) &\triangleq \text{col}\{\nabla\phi^\star(z_1), \nabla\phi^\star(z_2), \dots, \nabla\phi^\star(z_n)\} \\ \nabla f(\mathbf{x}) &\triangleq \text{col}\{\nabla f_1(x_1), \nabla f_2(x_2), \dots, \nabla f_n(x_n)\}. \end{aligned} \tag{7}$$

Additionally, we introduce a variable $\mathbf{y}$ to replace the integral. Then, the dynamical system (6) can be written using the newly defined notations,

$$\begin{aligned} \dot{\mathbf{z}} &= -(\nabla f(\mathbf{x}) + \mathbf{L}\mathbf{x} + \mathbf{y}), \\ \dot{\mathbf{y}} &= \mathbf{L}\mathbf{x}, \\ \mathbf{x} &= \nabla\phi^\star(\mathbf{z}), \end{aligned} \tag{8}$$

where $\mathbf{y} \in \mathbb{R}^{nd}$ and $\mathbf{y}(0) = \mathbb{0}$.

III. Main Results

In this section, we provide the convergence results of (8). In particular, we prove that under our assumptions, all agents in the network will converge exponentially fast to the global minimum of F in (1). In a previous work, the authors showed that under a subset of assumptions, the algorithm will asymptotically converge to the global optimum (without providing the rate).

Theorem 1: [[19]] Given Assumptions 1-4, for any starting point $x_i(0) = x_{i0}, z_i(0) = z_{i0}$ with $x_{i0} = \nabla\phi^\star(z_{i0})$, the distributed mirror descent algorithm with integral feedback proposed in (6) will converge to the global optimum asymptotically, i.e., $\lim_{t\to\infty} x_i(t) = x^\star$ for any $i \in [n]$.

The proof of this theorem can be found in [19], where it is also shown that agents reach consensus at the global optimal point, which is the unique equilibrium of the dynamical system (8). The equilibrium point for $\mathbf{x}, \mathbf{y}, \mathbf{z}$ is denoted by

$$\mathbf{x}^\star = \mathbb{1}_n \otimes x^\star, \quad \mathbf{y}^\star = -\nabla f(\mathbf{x}^\star),$$

$$\mathbf{z}^\star = \mathbb{1}_n \otimes z^\star = \mathbb{1}_n \otimes \nabla\phi(x^\star).$$

A. Coordinate Transformation

We use the change of variables in [19] for further analysis. Let $\mathbf{S} = \mathbf{L}^{\frac{1}{2}}$, and recall that $\mathbf{L} = \mathcal{L} \otimes I_d$ is a symmetric positive semi-definite matrix. We then introduce a new variable $\mathbf{w}(t) = \mathbf{S}\int_0^t \mathbf{x}(\tau)d\tau$. From (8) it is easy to show that $\mathbf{y} = \mathbf{S}\mathbf{w}$. We then center the variables by moving the system's equilibrium to the origin as follows

$$\tilde{\mathbf{x}} \triangleq \mathbf{x} - \mathbf{x}^\star, \quad \tilde{\mathbf{y}} \triangleq \mathbf{y} - \mathbf{y}^\star, \quad \tilde{\mathbf{w}} \triangleq \mathbf{w} - \mathbf{w}^\star, \quad \tilde{\mathbf{z}} \triangleq \mathbf{z} - \mathbf{z}^\star. \tag{9}$$

The first two equations in (8) can be rewritten as

$$\begin{aligned} \dot{\tilde{\mathbf{z}}} &= -(\nabla f(\tilde{\mathbf{x}} + \mathbf{x}^\star) - \nabla f(\mathbf{x}^\star)) - \mathbf{L}\tilde{\mathbf{x}} - \mathbf{S}\tilde{\mathbf{w}}, \\ \dot{\tilde{\mathbf{w}}} &= \mathbf{S}\tilde{\mathbf{x}}, \end{aligned} \tag{10}$$

Next, we perform a dimension reduction on variable $\tilde{\mathbf{w}}$. Define $r \triangleq \frac{1}{\sqrt{n}}\mathbb{1}_n$ and let $\mathcal{L} = Q\Lambda Q^\top$, where $\Lambda = diag\{0, \lambda_1, ..., \lambda_{n-1}\}$. From Assumption 3 it is clear that r is the first column of Q. We then define $R \in \mathbb{R}^{n\times(n-1)}$ such that $Q = [r, R]$. The following relationships follow subsequently

$$\begin{aligned} r^\top R = 0, \quad R^\top R = I_{n-1}, \quad RR^\top = I_n - \frac{1}{n}\mathbb{1}_n\mathbb{1}_n^\top, \\ r^\top \mathcal{L} r = 0, \quad R^\top \mathcal{L} R \succ 0. \end{aligned} \tag{11}$$

Now, let

$$\mathbf{r} \triangleq r \otimes I_d, \quad \mathbf{R} \triangleq R \otimes I_d, \quad \mathbf{Q} \triangleq Q \otimes I_d, \tag{12}$$

and define new vectors by the following transformations from $\tilde{\mathbf{w}}$,

$$\mathbf{W} \triangleq \mathbf{Q}^\top \tilde{\mathbf{w}} = \begin{bmatrix} \mathbf{r}^\top \\ \mathbf{R}^\top \end{bmatrix} \tilde{\mathbf{w}} = \begin{bmatrix} \mathbf{r}^\top \tilde{\mathbf{w}} \\ \mathbf{R}^\top \tilde{\mathbf{w}} \end{bmatrix} = \begin{bmatrix} \mathbf{W}_1 \\ \mathbf{W}_2 \end{bmatrix}.$$

Note that for $\mathbf{W}_1$, from (10) we can derive that

$$\dot{\mathbf{W}}_1 = \mathbf{r}^\top \dot{\tilde{\mathbf{w}}} = \mathbf{r}^\top \mathbf{S}\tilde{\mathbf{x}} = 0, \quad \mathbf{W}_1(0) = -\mathbf{r}^\top \mathbf{w}^\star = 0.$$

Therefore $\mathbf{W}_1 \equiv 0$ for all time t, and the system variable can be represented by $\mathbf{W}_2$ only. $\tilde{\mathbf{w}} = \begin{bmatrix} \mathbf{r} & \mathbf{R} \end{bmatrix} \mathbf{W} = \mathbf{r}\mathbf{W}_1 + \mathbf{R}\mathbf{W}_2 = \mathbf{R}\mathbf{W}_2$.

Furthermore, we replace variable $\mathbf{z}$ with $\mathbf{x}$. Since $\mathbf{z} = \nabla\phi(\mathbf{x})$, we have

$$\dot{\tilde{\mathbf{z}}} = \frac{d}{dt}(\mathbf{z} - \mathbf{z}^\star) = \nabla^2\phi(\mathbf{x})\dot{\tilde{\mathbf{x}}}.$$

Assumption 4 implies that $\nabla^2\phi(\mathbf{x})$ is positive definite and therefore invertible. Now, we can rewrite the system in (10) using only variables $\tilde{\mathbf{x}}$ and $\mathbf{W}_2$ as follows

$$\begin{aligned}\dot{\tilde{\mathbf{x}}} &= -\nabla^2\phi(\tilde{\mathbf{x}}+\mathbf{x}^\star)^{-1}(\nabla f(\tilde{\mathbf{x}}+\mathbf{x}^\star) - \nabla f(\mathbf{x}^\star) \\ &\qquad + \mathbf{L}\tilde{\mathbf{x}} + \mathbf{SRW}_2), \qquad (13)\\ \dot{\mathbf{W}}_2 &= \mathbf{R}^\top\mathbf{S}\tilde{\mathbf{x}},\end{aligned}$$

Thus, the (exponential) stability of (8) can be analyzed using the (exponential) stability of (13).

B. Exponential Convergence

With the system transformation in place, we can discuss the convergence and stability of distributed mirror descent (with integral feedback) in the following theorem.

Theorem 2: (Main Result) Given Assumptions 1-5, the origin is a locally exponentially stable equilibrium of (8) and (13).

Proof: If we linearize the system (13) at the origin, using the notation $\nabla^2\phi(\tilde{\mathbf{x}}+\mathbf{x}^\star)^{-1}|_{\tilde{\mathbf{x}}=0} = \mathbf{D}$, $\nabla^2 f(\tilde{\mathbf{x}}+\mathbf{x}^\star)|_{\tilde{\mathbf{x}}=0} = \mathbf{H}$, the linearized version of (13) is

$$\begin{bmatrix}\dot{\tilde{\mathbf{x}}}\\ \dot{\mathbf{W}}_2\end{bmatrix} = -\mathbf{M}\begin{bmatrix}\tilde{\mathbf{x}}\\ \mathbf{W}_2\end{bmatrix}, \qquad (14)$$
$$\text{where} \quad \mathbf{M} \triangleq \begin{bmatrix}\mathbf{D}(\mathbf{H}+\mathbf{L}) & \mathbf{DSR}\\ -\mathbf{R}^\top\mathbf{S} & 0\end{bmatrix}.$$

We denote by $\lambda_1, ..., \lambda_{(2n-1)d}$ the eigenvalues of the linearized system matrix $\mathbf{M}$ in (14). Based on Lemma 4, $Re[\lambda_i] > 0$ for all eigenvalues. Lemma 4 and its proof are provided later in the paper. Now, from Theorem 3.2 in [25], since $Re[\lambda_i] > 0$, the equilibrium of system (13), as well as the equilibrium of system (8) given by Theorem 1, are both locally exponentially stable. This means there exists $\delta > 0$ such that for any $\|col\{\mathbf{x},\mathbf{y},\mathbf{z}\} - col\{\mathbf{x}^\star,\mathbf{y}^\star,\mathbf{z}^\star\}\| \leq \delta$, the system state variables converge to the equilibrium (global optimal solution) exponentially fast. ■

Recall from Theorem 1 that the system (8) also exhibits global asymptotic convergence to the equilibrium. Then, for any starting point for $col\{\mathbf{x},\mathbf{y},\mathbf{z}\}$, the state variables can converge to a neighborhood of radius δ of equilibrium in a finite time $T(\delta)$. Combined with the exponential convergence rate within the ball, this means that (after a finite time), the system exhibits exponential convergence to the global optimal solution.

We now provide the following two lemmas used in the proof of Theorem 2.

Lemma 3: Given Assumptions 1-3, the matrix $(\mathbf{H}+\mathbf{L})$ is positive definite.

Proof: First, $(\mathbf{H}+\mathbf{L})$ is symmetric since both $\mathbf{H}$ and $\mathbf{L}$ are symmetric. For any non-zero vector $v \in \mathbb{R}^{nd}$, from Assumptions 1 and 3, we know that

$$v^\top\mathbf{H}v \geq 0, v^\top\mathbf{L}v \geq 0 \Rightarrow v^\top(\mathbf{H}+\mathbf{L})v = v^\top\mathbf{H}v + v^\top\mathbf{L}v \geq 0.$$

Furthermore, Since $\mathbb{1}_n$ is the unique eigenvector of $\mathcal{L}$ recovering the null eigenvalue, when $v^\top\mathbf{L}v = 0$, v must satisfy $v = \mathbb{1}_n \otimes u$ for some u. Then, Assumption 2 ensures $v^\top\mathbf{H}v = (\mathbb{1}_n \otimes u)^\top\mathbf{H}(\mathbb{1}_n \otimes u) = u^\top\nabla^2 F(x^\star)u > 0$. This shows that the symmetric matrix $\mathbf{H}+\mathbf{L}$ is positive definite. ■

Lemma 4: Given Assumptions 1-5, $Re[\lambda_i] > 0$ for all eigenvalues $\lambda_1, ..., \lambda_{(2n-1)d}$ of $\mathbf{M} = \begin{bmatrix}\mathbf{D}(\mathbf{H}+\mathbf{L}) & \mathbf{DSR}\\ -\mathbf{R}^\top\mathbf{S} & 0\end{bmatrix}$.

Proof: For any $i \in [(2n-1)d]$, λ_i must be a solution to $det(\mathbf{M} - \lambda I_{(2n-1)d}) = 0$. First, let us rule out the possibility of having $\lambda_i = 0$.

$$\begin{aligned}det(\mathbf{M}) &= det\left(\begin{bmatrix}\mathbf{D}(\mathbf{H}+\mathbf{L}) & \mathbf{DSR}\\ -\mathbf{R}^\top\mathbf{S} & 0\end{bmatrix}\right)\\ &= det\left(\begin{bmatrix}\mathbf{D}(\mathbf{H}+\mathbf{L}) & 0\\ -\mathbf{R}^\top\mathbf{S} & I_{(n-1)d}\end{bmatrix}\right.\\ &\qquad \left.\begin{bmatrix}I_{nd} & (\mathbf{D}(\mathbf{H}+\mathbf{L}))^{-1}\mathbf{DSR}\\ 0 & \mathbf{R}^\top\mathbf{S}(\mathbf{D}(\mathbf{H}+\mathbf{L}))^{-1}\mathbf{DSR}\end{bmatrix}\right)\\ &= det(\mathbf{D}(\mathbf{H}+\mathbf{L}))det(\mathbf{R}^\top\mathbf{S}(\mathbf{D}(\mathbf{H}+\mathbf{L}))^{-1}\mathbf{DSR})\\ &= det(\mathbf{D})det(\mathbf{H}+\mathbf{L})det(\mathbf{R}^\top\mathbf{S}(\mathbf{H}+\mathbf{L})^{-1}\mathbf{SR}).\end{aligned} \qquad (15)$$

Since $\mathbf{R}^\top\mathbf{SSR} = (R^\top\mathcal{L}R)\otimes I_d \succ 0$, the null space of $\mathbf{SR}$ is 0. Then, $\mathbf{R}^\top\mathbf{S}(\mathbf{H}+\mathbf{L})^{-1}\mathbf{SR}$ is positive definite since $\mathbf{H}+\mathbf{L}$ is positive definite (Lemma 3). As a result, this confirms that $det(\mathbf{M}) > 0$, implying $\lambda_i \neq 0$ for all $i \in [(2n-1)d]$.

The next step is to look at the characteristic polynomial of $\mathbf{M}$, where we have

$$\begin{aligned}0 &= det(\mathbf{M} - \lambda I_{(2n-1)d})\\ &= det\left(\begin{bmatrix}\mathbf{D}(\mathbf{H}+\mathbf{L}) - \lambda I_{nd} & \mathbf{DSR}\\ -\mathbf{R}^\top\mathbf{S} & -\lambda I_{(n-1)d}\end{bmatrix}\right)\\ &= det(\mathbf{D}(\mathbf{H}+\mathbf{L}) - \lambda I_{nd} - \mathbf{DSR}(\lambda I_{(n-1)d})^{-1}\mathbf{R}^\top\mathbf{S})\\ &\qquad\qquad det(-\lambda I_{(n-1)d})\\ &= det(\mathbf{D})det((\mathbf{H}+\mathbf{L}) - \lambda\mathbf{D}^{-1} - \frac{1}{\lambda}\mathbf{SRR}^\top\mathbf{S})\\ &= det((\mathbf{H}+\mathbf{L}) - \lambda\mathbf{D}^{-1} - \frac{1}{\lambda}\mathbf{S}(I_{nd} - \mathbf{rr}^\top)\mathbf{S})\\ &= det((\mathbf{H}+\mathbf{L}) - \lambda\mathbf{D}^{-1} - \frac{1}{\lambda}\mathbf{L}).\end{aligned} \qquad (16)$$

Observe that $(\mathbf{H}+\mathbf{L}) - \lambda\mathbf{D}^{-1} - \frac{1}{\lambda}\mathbf{L}$ is a symmetric matrix, and $det((\mathbf{H}+\mathbf{L}) - \lambda\mathbf{D}^{-1} - \frac{1}{\lambda}\mathbf{L}) = 0$ implies that there exists a non-zero vector $v \in \mathbb{R}^{nd}$ for any solution of λ such that

$$v^\top((\mathbf{H}+\mathbf{L}) - \lambda\mathbf{D}^{-1} - \frac{1}{\lambda}\mathbf{L})v = 0.$$

Since $\mathbf{L}$ is positive semi-definite, $\mathbf{D}^{-1}$ and $(\mathbf{H}+\mathbf{L})$ are positive definite, $v^\top\mathbf{L}v \geq 0, v^\top\mathbf{D}^{-1}v > 0, v^\top(\mathbf{L}+\mathbf{H})v > 0$. When $v^\top\mathbf{L}v = 0$,

$$\lambda = \frac{v^\top(\mathbf{L}+\mathbf{H})v}{v^\top\mathbf{D}^{-1}v} > 0,$$

when $v^\top\mathbf{L}v > 0$,

$$\lambda = \frac{v^\top(\mathbf{L}+\mathbf{H})v \pm \sqrt{(v^\top(\mathbf{L}+\mathbf{H})v)^2 - 4v^\top\mathbf{L}vv^\top\mathbf{D}^{-1}v}}{2v^\top\mathbf{D}^{-1}v},$$

certifying that $Re[\lambda] > 0$ in both cases. ■

IV. Numerical Simulation

In this section, we use a real data-set to show the linear convergence of the training loss in a regression problem. We will investigate the performance of distributed mirror descent with and without integral feedback. We utilize Euler's discretization scheme on algorithm (8). The resulting discrete-time algorithm for distributed mirror descent with integral feedback is provided below.

$$\begin{aligned} {z_i}^{(k+1)} &= {z_i}^{(k)} - \Bigg(\nabla f_i({x_i}^{(k)}) + {y_i}^{(k)} \\ &\quad + \sum_{j\in\mathcal{N}_i} ({x_i}^{(k)} - {x_j}^{(k)})\Bigg)\Delta t, \\ {y_i}^{(k+1)} &= {y_i}^{(k)} + \sum_{j\in\mathcal{N}_i} ({x_i}^{(k)} - {x_j}^{(k)})\Delta t, \\ {x_i}^{(k+1)} &= \nabla\phi^\star({z_i}^{(k+1)}). \end{aligned} \tag{17}$$

Details of this discretization is omitted in this manuscript and has been provided in [19].

Distance Generating Function for MD: We use the *Negative Entropy* as our distance generation function ϕ, namely,

$$\phi(x) = \sum_{j=1}^{d} [x]_j \log([x]_j) \Longrightarrow [z]_i = [\nabla\phi(x)]_i = 1+\log([x]_i).$$

Based on Section II-C, the corresponding convex conjugate function $\phi^\star$ can be written below,

$$\phi^\star(z) = \sum_{j=1}^{d} e^{[z]_j-1} \Longrightarrow [x]_i = [\nabla\phi^\star(z)]_i = e^{[z]_i-1}.$$

The reason for our choice of DGF is that Kullback–Leibler divergence is one the most commonly used Bregman divergences other than Euclidean distance, which simply reduces the method to distributed gradient descent with integral feedback as in [9].

Network Structure: We consider a 3×3 grid network, which results in a 9-agent network. The connectivity degree for each agent is between $2-4$. This network satisfies Assumption 3.

Data Set and Model: We use the *Wine Quality Data Set* in UCI ML repository [26]. This is a regression data-set with 11 continuous input variables. Each agent is assigned 400 data instances with no overlap. For agent $i \in [n]$, we denote the input data and output data as $A_i \in \mathbb{R}^{400\times11}$ and $b_i \in \mathbb{R}^{400}$, respectively. The model is a linear regression where the loss function is defined as the quadratic error loss, $f_i(x) = \frac{1}{2}\left\|A_i x - b_i\right\|^2$. We can verify that this setup satisfies Assumptions 1 and 2. The global objective function $F(x) = \sum_{i\in[n]} \frac{1}{2}\left\|A_i x - b_i\right\|^2 = \frac{1}{2}\left\|\mathbf{A}x - \mathbf{b}\right\|^2$, where $\mathbf{A}, \mathbf{b}$ are the stacked version of A_i, b_i, respectively. Moreover, we can calculate the closed form solution of the global problem, $x^\star = \mathbf{A}^\dagger\mathbf{b}$, where $\mathbf{A}^\dagger$ denotes the pseudo-inverse of $\mathbf{A}$.

Note that the selected model is not necessarily optimal for test prediction accuracy, and the aim of this numerical simulation is to show the ability of our proposed algorithm to converge exponentially fast to the optimal loss on a given data set. Finding a better model to fit this data set is not the main focus of this work.

Performance: We provide the trajectory of our proposed algorithm and also a comparison between our work and prior works [14], [17] on distributed mirror descent without integral feedback. In particular, we once run the algorithm without integral feedback using diminishing step-size $\frac{1}{\sqrt{k}}$ to ensure consensus, and once using a constant step-size in optimization, which is unable to reach optimal solution.

The plot of $F(x_1^{(k)}) - F(x^\star)$ is shown in Fig. 2, representing the convergence speed of the three algorithms. We can see that our proposed algorithm converges faster than diminishing step-size setup, while the constant step-size setup without integral feedback fails to converge. We plot $\log(F(x_1^{(k)}) - F(x^\star))$ in Fig. 2 to further display the exponential convergence (i.e., linear in log-scale) speed of our proposed method.

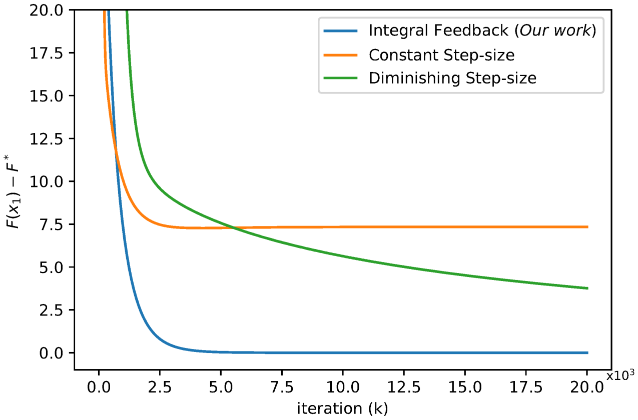

Fig. 1: The trajectory of difference between $F(x)$ and optimal $F(x^\star)$ evaluated at agent 1.

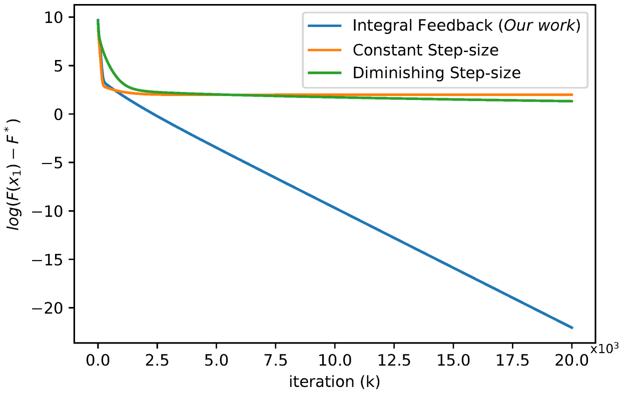

Fig. 2: The trajectory of log-distance between $F(x)$ and optimal $F(x^\star)$ evaluated at agent 1.

V. Conclusion

In this paper, we studied the distributed optimization problem, where a network of agents work together to find the optimal solution for a global objective function. We studied a distributed mirror descent algorithm that benefits from the idea of integral feedback. We established that the convergence rate of our algorithm is exponential (locally), which shows the advantage of adopting integral feedback for strongly convex problems. Our claim is supported by empirical results on a real data-set.

Though our work provides exponential convergence rate for strongly convex distributed optimization, more analysis is needed to generalize this work to other network settings, such as dynamic networks and networks with delays. Another interesting direction includes the theoretical analysis of the discretized version of this algorithm, shown in (17). These are open questions left for future works.

Acknowledgments

We gratefully acknowledge the support of NSF ECCS-2136206 and NSF ECCS-1933878 Award. We also thank Ashish Cherukuri, who brought to our attention a recent reference.

References

[1] A. Chavez, A. Moukas, and P. Maes, "Challenger: A multi-agent system for distributed resource allocation," in *Proceedings of the first international conference on Autonomous agents*, 1997, pp. 323–331.

[2] U. A. Khan, S. Kar, and J. M. Moura, "Distributed sensor localization in random environments using minimal number of anchor nodes," *IEEE Transactions on Signal Processing*, vol. 57, no. 5, pp. 2000–2016, 2009.

[3] Z. Qu, *Cooperative control of dynamical systems: applications to autonomous vehicles*. Springer Science & Business Media, 2009.

[4] S. Shahrampour, A. Rakhlin, and A. Jadbabaie, "Distributed detection: Finite-time analysis and impact of network topology," *IEEE Transactions on Automatic Control*, vol. 61, no. 11, pp. 3256–3268, 2015.

[5] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

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

[7] C. Xi and U. A. Khan, "Dextra: A fast algorithm for optimization over directed graphs," *IEEE Transactions on Automatic Control*, vol. 62, no. 10, pp. 4980–4993, 2017.

[8] B. Gharesifard and J. Cortés, "Distributed continuous-time convex optimization on weight-balanced digraphs," *IEEE Transactions on Automatic Control*, vol. 59, no. 3, pp. 781–786, 2013.

[9] S. S. Kia, J. Cortés, and S. Martínez, "Distributed convex optimization via continuous-time coordination algorithms with discrete-time communication," *Automatica*, vol. 55, pp. 254–264, 2015.

[10] X. Zeng, P. Yi, and Y. Hong, "Distributed continuous-time algorithm for constrained convex optimizations via nonsmooth analysis approach," *IEEE Transactions on Automatic Control*, vol. 62, no. 10, pp. 5227–5233, 2017.

[11] S. Yang, Q. Liu, and J. Wang, "A multi-agent system with a proportional-integral protocol for distributed constrained optimization," *IEEE Transactions on Automatic Control*, vol. 62, no. 7, pp. 3461–3467, 2016.

[12] A. S. Nemirovsky and D. B. Yudin, "Problem complexity and method efficiency in optimization." 1983.

[13] A. Ben-Tal, T. Margalit, and A. Nemirovski, "The ordered subsets mirror descent optimization method with applications to tomography," *SIAM Journal on Optimization*, vol. 12, no. 1, pp. 79–108, 2001.

[14] S. Shahrampour and A. Jadbabaie, "Distributed online optimization in dynamic environments using mirror descent," *IEEE Transactions on Automatic Control*, vol. 63, no. 3, pp. 714–725, 2018.

[15] D. Yuan, Y. Hong, D. W. Ho, and G. Jiang, "Optimal distributed stochastic mirror descent for strongly convex optimization," *Automatica*, vol. 90, pp. 196–203, 2018.

[16] M. Rabbat, "Multi-agent mirror descent for decentralized stochastic optimization," in *IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 2015, pp. 517–520.

[17] J. Li, G. Chen, Z. Dong, and Z. Wu, "Distributed mirror descent method for multi-agent optimization with delay," *Neurocomputing*, vol. 177, pp. 643–650, 2016.

[18] T. T. Doan, S. Bose, D. H. Nguyen, and C. L. Beck, "Convergence of the iterates in mirror descent methods," *IEEE Control Systems Letters*, vol. 3, no. 1, pp. 114–119, 2019.

[19] Y. Sun and S. Shahrampour, "Distributed mirror descent with integral feedback: Asymptotic convergence analysis of continuous-time dynamics," *IEEE Control Systems Letters*, vol. 5, no. 5, pp. 1507–1512, 2020.

[20] A. Borovykh, N. Kantas, P. Parpas, and G. A. Pavliotis, "To interact or not? the convergence properties of interacting stochastic mirror descent," in *International Conference on Machine Learning (ICML) Workshop on 'Beyond First order methods in ML Systems*, 2020.

[21] M. Raginsky and J. Bouvrie, "Continuous-time stochastic mirror descent on a network: Variance reduction, consensus, convergence," in *IEEE Conference on Decision and Control (CDC)*, 2012, pp. 6793–6800.

[22] Y. Yu and B. Açıkmeşe, "RLC circuits-based distributed mirror descent method," *IEEE Control Systems Letters*, vol. 4, no. 3, pp. 548–553, 2020.

[23] J.-B. Hiriart-Urruty and C. Lemaréchal, *Fundamentals of convex analysis*. Springer Science & Business Media, 2012.

[24] W. Krichene, A. Bayen, and P. L. Bartlett, "Accelerated mirror descent in continuous and discrete time," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2015, pp. 2845–2853.

[25] H. K. Khalil, *Nonlinear control*. Pearson Higher Ed, 2014.

[26] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis, "Modeling wine preferences by data mining from physicochemical properties," *Decision Support Systems*, vol. 47, no. 4, pp. 547–553, 2009.