A Stochastic Multi-Rate Control Framework For Modeling Distributed Optimization Algorithms

Xinwei Zhang [1] **Mingyi Hong** [1] **Sairaj Dhople** [1] **Nicola Elia** [1]

Abstract

In modern machine learning systems, distributed algorithms are deployed across applications to ensure data privacy and optimal utilization of computational resources. This work offers a fresh perspective to model, analyze, and design distributed optimization algorithms through the lens of stochastic multi-rate feedback control. We show that a substantial class of distributed algorithms—including popular Gradient Tracking for decentralized learning, and FedPD and Scaffold for federated learning—can be modeled as a certain discrete-time stochastic feedback-control system, possibly with multiple sampling rates. This key observation allows us to develop a generic framework to analyze the convergence of the entire algorithm class. It also enables one to easily add desirable features such as differential privacy guarantees, or to deal with practical settings such as partial agent participation, communication compression, and imperfect communication in algorithm design and analysis.

1. Introduction

Distributed optimization has played an important role in several traditional system-theoretic domains such as control and signal processing, and more recently, in machine learning (ML). Some contemporary applications where distributed optimization finds useful include large-scale decentralized neural network training, federated learning (FL), and multi-agent reinforcement learning. In a typical distributed optimization setting, the agents in the network jointly solve a system-level optimization problem, with the constraint that they only utilize local data, local computation, and local communication resources.

1.1. Design Considerations and Challenges

A few key design considerations for contemporary distributed algorithms are listed below:

Efficient Computation. Since local agents may contend with computational-resource and power limitations, it is desirable that they perform computation in a cost-effective manner. In practice, state-of-the-art distributed algorithms in ML applications typically utilize stochastic gradient descent (SGD) based algorithm as their local computation engine (Chang et al., 2020). So a key design consideration is to reduce the total number of data sample access, or equivalently, to improve sample efficiency.

Efficient Communication. Frequent inter-agent message exchanges can present several bottlenecks to system performance in addition to consuming power. In applications such as decentralized training (DT) and federated learning (FL), communication links may not have high enough bandwidth (Bonawitz et al., 2019; Li et al., 2020). Therefore, it is desirable that the local communication between the agents happen only when necessary, and when it happens, as little information is exchanged as possible.

Flexibility based on Practical System Considerations. Since distributed algorithms are often implemented in different environments, and they are used in applications across different domains, it is desirable that they are flexible and can take into consideration practical requirements (e.g., preserving user privacy), accommodate desired communication patterns, and allow for the possibility of agents participating occasionally (McMahan et al., 2018; Koloskova et al., 2020; Yuan et al., 2021).

Guaranteed Performance. The performance of distributed algorithms can be very different compared with their centralized counterpart, and if not designed carefully, distributed algorithms can diverge easily (Lian et al., 2017; Zhang et al., 2020). So, it is important that algorithms offer convergence guarantees at a minimum. Further, it is desirable if such guarantees can characterize the efficiency in computation and communication.

There has been remarkably high interest in distributed algorithms in recent years across applications. These algo-

[1]Department of Electric and Computer Engineering, Minnesota University, MN, United States. Correspondence to: Xinwei Zhang <zhan6234@umn.edu>, Mingyi Hong <mhong@umn.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

rithms are typically developed in an application-specific manner. They are designed, for example, to: improve communication efficiency by utilizing model compression schemes (Basu et al., 2019; Koloskova et al., 2019a); perform occasional communication (Chen et al., 2018; Sun et al., 2020); improve computational efficiency by utilizing SGD based schemes (Lian et al., 2017; Lu et al., 2019); understand the best possible communication and computation complexity (Scaman et al., 2017; Lu & De Sa, 2021); incorporate differential privacy (DP) guarantee into the system (Yuan et al., 2021); or to deal with the practical situation where even the (stochastic) gradients may not be accessible (Yuan et al., 2015; Hajinezhad et al., 2019).

Despite extensive research in distributed algorithms, several challenges persist in their synthesis and application. First, the proliferation of the algorithms indeed gives practitioners many alternatives to choose from. However, the *downside* is that there are simply too many algorithms available, so it becomes difficult to appreciate all underlying technical details and common themes linking them. Second, the current practice is that we need to design a new algorithm and develop the corresponding analysis for each particular application scenario (e.g., FL) with a specific set of requirements (e.g., communication efficiency + privacy). Given the combinatorial number of different applications and requirements, this general process readily becomes very tedious.

Therefore, we ask: Is it possible to have a generic "model" of distributed algorithms, which can abstract their important features (e.g., DP preserving mechanism, compressed communication, occasional communication) into tractable modules? If the answer is affirmative, can we design a framework that utilizes these abstract modules, unifies the analysis of (possibly a large subclass of) distributed algorithms, and subsequently facilitates the design of new ones?

A limited number of existing works have attempted to address these two questions, but the scope is still very restricted. Reference (Sundararajan et al., 2019) focuses only on the DT algorithms with linear operators on the gradients and fails to cover the FL or stochastic settings. (Koloskova et al., 2020) only considers stochastic gradient descent in FL setting, which cannot generalize to any other algorithms. Other works related to continuous-time analysis of distributed algorithms, as well as using control theory to facilitate the design and analysis, are provided in Appendix A.1

1.2. Contributions of this work

In this work, we propose to use techniques from stochastic multi-rate feedback control to "model" a class of distributed algorithms. Specifically, we design a new feedback control system to model the distributed algorithms; propose the idea of using the *multi-rate sampling technique* to model different frequencies at which communication and computation are carried out; and use stochasticity in each feedback controller to model desirable features (such as gradient compression, DP noise, and partial agent participation). Moreover, we provide a generic convergence analysis for the entire control system, and we show how the modeling procedure can facilitate algorithm analysis and design via an example based on the popular distributed gradient tracking algorithm. To our knowledge, this is the first work that attempts to develop a generic model to analyze and design stochastic distributed algorithms. Our new modeling procedure and analytical results offer the following contributions:

1) Unified Perspective. The proposed stochastic control system abstracts a number of key features of distributed algorithms into generic properties of the controllers. For example, it connects various practical system requirements and considerations (such as DP requirements, partial agent participation) with different forms of stochasticity in the controllers (e.g., additive noise, multiplicative noise). It connects major paradigms of distributed algorithms such as DT and FL, via the so-called "multi-rate" sampling technique, in which different feedback loops are sampled using different intervals. These abstractions together provide a unified view of a substantial subclass of distributed algorithms.

2) Streamlined Analysis. To analyze the basic convergence property of a stochastic algorithm, one only needs to examine a set of basic properties of the control system (e.g., certain descent property of the deterministic controller, the stochasticity arises from each controller); this can greatly reduce the effort for analysis.

3) Facilitating Algorithm Design. With the models and analysis framework ready, one can design a new algorithm and generate the corresponding theoretical performance guarantees, by first identifying the basic algorithmic components (e.g., what consensus algorithm to use, what local optimizer to use, and what features to add), translating them to the control system representation, and then applying results obtained in item 2) above.

2. Preliminaries

In this section, we introduce assumptions and notations leveraged in the remainder. First, we formally define the distributed optimization problem as minimizing a sum of smooth and possibly non-convex local loss functions on N agents (Wang & Elia, 2011):

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^{Nd_x}} \quad & f(\mathbf{x}) := \frac{1}{N}\sum_{i=1}^{N} f_i(x_i), \\ \text{s.t.} \quad & x_i = x_j,\ \forall\,(i,j)\in\mathbf{E}, \end{aligned} \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^{Nd_x}$ stacks N local variables $\mathbf{x} := [x_1; \ldots; x_N]$, $x_i \in \mathbb{R}^{d_x}$, $\forall\, i \in [N]$, where we denote the set $[N] := \{1, \ldots, N\}$, and the agents are connected by a communication graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$, which consists of a set $\mathbf{V}$ of agents indexed by $i \in [N]$, and a undirected edge set $\mathbf{E} \subset \mathbf{V} \times \mathbf{V}$. The incidence matrix $A \in \{-1, 0, 1\}^{|\mathbf{E}| \times |\mathbf{V}|}$ of graph $\mathcal{G}$ is defined as follows: if edge $(i, j) \in \mathbf{E}$ connects agent i, j with $i > j$, then $A_{(i,j),i} = 1$, $A_{(i,j),j} = -1$ and $A_{(i,j),k} = 0$, $\forall k \neq i, j$. The Laplacian matrix of the graph can be expressed as $\mathcal{L} = -A^T A$. We denote the length-n all-one vector by $\mathbb{1}_n$, averaging matrix $R := \frac{\mathbb{1}_N \mathbb{1}_N^T}{N}$, and identity matrix of dimension $N \times N$ by I. For simplicity of notation, we ignore the possible Kronecker products and vectorization when dealing with stacked vectors and matrices; for instance, we write the average of $\mathbf{x}$ as $\bar{\mathbf{x}} := \frac{\mathbb{1}_N^T}{N}\mathbf{x}$, the stacked local gradient as $\nabla f(\mathbf{x}) = [\nabla f_1(x_1); \ldots; \nabla f_N(x_N)]$, and the averaged gradient as $\nabla f(\bar{\mathbf{x}}) = \frac{1}{N}\sum_{i=1}^N \nabla f_i(\bar{\mathbf{x}})$.

We make the following blanket assumptions on (1):

A 1 (Graph connectivity) *The union of the communication graphs over time $t \in [0, \infty)$ is connected, i.e., 0 is a simple eigenvalue its Laplacian matrix, with corresponding eigenvector $\frac{\mathbb{1}_N}{\sqrt{N}}$.*

A 2 (Lipschitz gradient) *The f_i's have Lipschitz gradient with constant L_f:*

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq L_f \|x - y\|, \ \forall\, x, y \in \mathbb{R}^{d_x}, \forall\, i \in [N].$$

A 3 (Lower bounded functions) *The loss functions are lower bounded:*

$$f_i(x) \geq \underline{f}_i > -\infty, \quad \forall x \in \mathbb{R}^{d_x}, \quad \forall i \in [N],$$

$$f(\mathbf{x}) \geq f^\star \geq \sum_{i=1}^N \underline{f}_i, \quad \forall \mathbf{x} \in \mathbb{R}^{Nd_x},$$

where $f^\star$ is the infimum of $f(\mathbf{x})$.

Let us briefly comment on these assumptions. First, A1 is necessary for the problem (1) to be solved with distributed iterative methods, while allowing directed and/or not strongly connected time-varying communication graphs $\mathcal{G}(t)$. Subsequently, in Section 3, we will show that time-varying graphs can be related to many practical algorithm implementations. Second, A2 is a commonly used assumption for analyzing non-convex optimizations. We are interested in finding the (ϵ-accurate) first-order stationary points (FOSP) of the problem, which is defined as follows:

Definition 1 (FOSP, ϵ-stationary point) *The FOSP and ϵ-stationary point are defined respectively as:*

$$\nabla f(\bar{\mathbf{x}}) = 0, \quad (I - R) \cdot \mathbf{x} = 0, \tag{2a}$$

$$\|\nabla f(\bar{\mathbf{x}})\|^2 + \|(I - R) \cdot \mathbf{x}\|^2 \leq \epsilon. \tag{2b}$$

In addition, we refer to the left-hand-side (LHS) of (2b) as the stationarity gap of (1), $\|\nabla f(\bar{\mathbf{x}})\|^2$ as the convergence error, and $\|(I - R) \cdot \mathbf{x}\|^2$ as the consensus error.

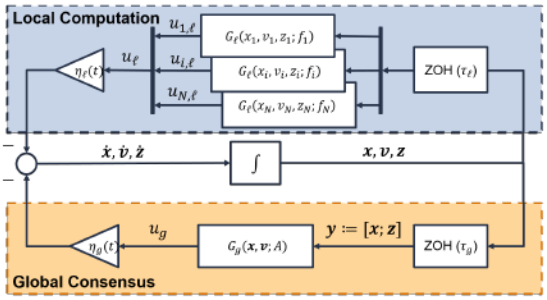


Figure 1: The multi-agent multi-rate double-loop feedback control system for solving (1).

To analyze stochastic systems, we define the expectation conditioning on all the information until time t as $\mathbb{E}_t[(\cdot)] := \mathbb{E}[(\cdot)|\text{information until } t]$, the variance as $\text{Var}_t(\cdot)$, and covariance as $\text{Cov}_t(\cdot, \cdot)$. Further, $\tilde{(\cdot)}$ denotes the stochastic version of the variables and functions.

3. System Description

In this section, we present the stochastic multi-rate feedback-control system that we propose to "model" distributed algorithms. We first develop a deterministic version of the system, discuss its properties, as well as how the system can model certain classes of (deterministic) algorithms under different sampling strategies. Then, we establish the link between different kinds of system stochasticity to desirable features of distributed algorithms.

3.1. Deterministic System

To find the FOSP of problem (1), we first develop a deterministic control system, in such a way that the system enters its stationary points if and only if one set of the state variables of the system correspond to a stationary solution of (1). First, let us define $\mathbf{x}$ as the main state variable of the system; introduce the *global consensus feedback loop* (GCFL) and *local computation feedback loop* (LCFL), where the former incorporates the dynamics from multi-agent interactions and pushes $\mathbf{x}$ to consensus, while the latter steers the system to find the stationary solution. See Figure 1 as an illustration of the system. In what follows, we introduce the different subsystems involved; note that $\eta_g(t)$ and $\eta_l(t)$ are the controller gains for the global and local controllers.

- **(GCFL).** Define a set of auxiliary state variables $\mathbf{v} := [v_1; \ldots; v_N] \in \mathbb{R}^{Nd_v}$, with $v_i \in \mathbb{R}^{d_v}$, $\forall\, i$; further define $\mathbf{y} := [\mathbf{x}; \mathbf{v}] \in \mathbb{R}^{N(d_x + d_v)}$; the time-invariant feedback controller $G_g(\cdot; A) : \mathbb{R}^{N(d_x+d_v)} \to \mathbb{R}^{N(d_x+d_v)}$ operates on $\mathbf{y}$ to ensure the agents remain coordinated, and the states $\mathbf{y}$ remain close to consensus. Finally, we denote the output at time t as $u_g(t) := G_g(\mathbf{y}(t); A)$, which can be split as $u_g(t) = [u_{g,x}(t); u_{g,v}(t)]$;

- **(LCFL).** Define another set of auxiliary state variables

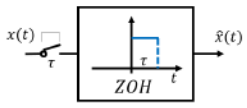


Figure 2: The zeroth-order hold (ZOH) for discretizing a continuous-time system.

$\mathbf{z} := [z_1; \ldots; z_N] \in \mathbb{R}^{Nd_z}$, with $z_i \in \mathbb{R}^{d_z}$, $\forall\, i$; define a set of time-invariant feedback controllers $G_\ell(\cdot; f_i) : \mathbb{R}^{d_x+d_v+d_z} \to \mathbb{R}^{d_x+d_v+d_z}$, one for each agent i. Further define $\mathbf{s} := [\mathbf{x}; \mathbf{v}; \mathbf{z}] \in \mathbb{R}^{N(d_x+d_v+d_z)}$. Then each agent will use LCFL to operate on its local state variables $s_i := [x_i; v_i; z_i]$, to ensure that its local system converges to a stationary solution. Finally, we denote the output at time t as $u_{i,\ell}(t) := G_\ell(s_i(t); f_i)$, which can further be split as $u_{i,\ell}(t) = [u_{i,\ell,x}(t), u_{i,\ell,v}(t), u_{i,\ell,z}(t)]$.

Throughout the paper, we use $u_{i,\ell}(t), u_g(t)$ and $G_\ell(s_i(t); f_i), G_g(\mathbf{y}(t); A)$ interchangeably.

System Discretization: The double-loop continuous-time system can be discretized by using a switch that samples the input with sample time τ, followed by a zeroth-order hold (ZOH) that keeps the signal constant between the consecutive sampling instances (Kuo, 1980); see Figure 2. More specifically, we place two ZOH units before the signal enters the two loops. This architecture offers the flexibility of choosing different sampling time for different loops resulting in three kinds of discretized systems:

- **Case I.** $\tau_g = \tau_\ell > 0$, the GCFL and LCFL are discretized with the same rate. In this case, the algorithm performs one local update followed by one step of global communication. Such an update pattern belongs to the scheme of decentralized training (DT) algorithms;
- **Case II.** $\tau_g > \tau_\ell > 0$, the local computation loop is updated more frequently. Let $\tau_g = Q \cdot \tau_\ell$, i.e., each agent performs Q steps of local computation between every two communication steps. This update strategy is related to the class of (horizontal) FL algorithms (Bonawitz et al., 2019). Further note that in the FL setting, the communication graph takes the fully connected graph as a special case;
- **Case III.** $\tau_\ell > \tau_g > 0$, the global communication loop is updated more frequently. We assume that $\tau_\ell = K \cdot \tau_g$, i.e., the agents perform K steps of communication between two local computation steps. This system is related to algorithms that aims to achieve the optimal communication complexity (Scaman et al., 2017; Sun & Hong, 2019; Rogozin et al., 2021).

Let us define $\tau := \min\{\tau_g, \tau_\ell\}$ as the minimum sampling time interval, and assume $t \bmod \tau = 0$ for the rest of the paper. We summarize the above discretization cases in Table 1 and provide some example algorithms that fit in the three cases.

We use the distributed gradient tracking (DGT) algorithm (Di Lorenzo & Scutari, 2016; Yuan et al., 2020) as an example to illustrate how to place it within the structure of the proposed system. The steps of DGT are:

$$\begin{aligned} \mathbf{x}^+ &= W\mathbf{x} - \alpha\mathbf{v}, \quad \mathbf{z}^+ = \mathbf{x}, \\ \mathbf{v}^+ &= W\mathbf{v} + (\nabla f(\mathbf{x}) - \nabla f(\mathbf{z})), \end{aligned} \tag{3}$$

where the states are initialized as $\mathbf{v}^0 = \nabla f(\mathbf{x}^0), \mathbf{z}^0 = \mathbf{x}^0$, α is the stepsize, and W is some mixing matrix. The continuous-time system corresponding to the DGT is:

$$\begin{aligned} \dot{\mathbf{x}} &= -(I - W)\mathbf{x} - \alpha\mathbf{v}, \quad \dot{\mathbf{z}} = \mathbf{x} - \mathbf{z}, \\ \dot{\mathbf{v}} &= -(I - W)\mathbf{v} + (\nabla f(\mathbf{x}) - \nabla f(\mathbf{z})), \end{aligned} \tag{4}$$

with $\tau_\ell = \tau_g = 1$. Such a discretization pattern places the above transcription in Case I. We can also extract the local and consensus controllers of the system as:

$$u_g(t) = \begin{bmatrix} (I-W) & 0 \\ 0 & (I-W) \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{v}(t) \end{bmatrix},$$

$$u_{i,\ell}(t) = \begin{bmatrix} \alpha v_i(t) \\ (\nabla f_i(x_i(t)) - \nabla f_i(z_i(t))) \\ x_i(t) - z_i(t) \end{bmatrix},$$

with $\eta_g(t) = \eta_\ell(t) = 1$. Note that using the discretization patterns in Case II and Case III, instead of Case I, leads to new variants of the DGT algorithm.

Next, let us specify a few abstract properties that the controllers need to have. These properties will later help us analyze the behavior of the entire system, and therefore, all the algorithms that it can be used to model.

PD 1 (Linear Averaging GCFL) *The controller G_g is a linear averaging operator of $\mathbf{y}$, i.e., $G_g(\mathbf{y}; A) = W_A\mathbf{y}$ for some matrix $W_A \in \mathbb{R}^{N(d_x+d_v)}$ parameterized by A, and satisfies the following properties:*

$$\begin{aligned} C_g \left\|(I-R)\cdot\mathbf{y}\right\|^2 \le \left\|W_A\mathbf{y}\right\|^2 \le \left\|(I-R)\cdot\mathbf{y}\right\|^2, \\ W_A = W_A^T,\ \langle \mathbb{1}_N, W_A\rangle = 0. \end{aligned} \tag{5}$$

PD 2 (Lipschitz Smoothness) *The local controller is Lipschitz continuous, that is:*

$$\begin{aligned} &\left\|G_\ell(s_i; f_i) - G_\ell(s_i'; f_i)\right\| \le L\left\|s_i - s_i'\right\|, \\ &\forall\, i \in [N],\ s_i, s_i' \in \mathbb{R}^{d_x+d_v+d_z}. \end{aligned}$$

PD 3 (Size of Control Signals) *For given s_i, the sizes of the control signals are upper bounded by that of the local gradients, i.e., for some positive constants C_x, C_v, C_z and $C_f = C_x^2 + C_v^2 + C_z^2$:*

$$\begin{aligned} &\left\|u_{i,\ell,x}\right\| \le C_x\left\|\nabla f_i(x_i)\right\|, \ \left\|u_{i,\ell,v}\right\| \le C_v\left\|\nabla f_i(x_i)\right\|, \\ &\left\|u_{i,\ell,z}\right\| \le C_z\left\|\nabla f_i(x_i)\right\|, \ \left\|u_{i,\ell}\right\|^2 \le C_f\left\|\nabla f_i(x_i)\right\|^2. \end{aligned}$$

These properties are easy to verify: PD1 follows A1, PD2 and PD3 can be derived from A2. Further, assume that within the sampling intervals the stepsizes are kept as constants, i.e., $\eta_g(t_1) = \eta_g(t),\ \forall t_1 \in [t, t+\tau_g)$, and $\eta_\ell(t_1) = \eta_\ell(t),\ \forall t_i \in [t, t+\tau_\ell)$,

3.2. System Stochasticity

As mentioned in the introduction, in practical ML applications, it is often preferred to use stochastic algorithms

Case	$\tau_\ell, \tau_{\mathbf{g}}$	Comm.	Comp.	Related Algorithm
I	$\tau_g = \tau_\ell > 0$	Same rate		DGD (Yuan et al., 2016), DGT (Yuan et al., 2020)
II	$\tau_g = Q\tau_\ell > 0$	Slow	Fast	FedPD (Zhang et al., 2020), Scaffold (Karimireddy et al., 2020)
III	$\tau_\ell = K\tau_g > 0$	Fast	Slow	xFilter (Sun & Hong, 2019), DSAGD (Rogozin et al., 2021)

Table 1: Summary of discretization settings, and the corresponding distributed algorithms.

rather than deterministic ones. Therefore, we consider replacing the deterministic controllers introduced previously (Fig. 1) with stochastic ones, denoted by $\tilde{G}_\ell(\cdot), \tilde{G}_g(\cdot)$. We start by providing generic discussions on how these stochastic controllers are modeled. Specific correspondence of these controllers to concrete applications will be presented in Section 5.

Additive Noise: The first form of stochastic controller has additive noise at its output. That is:

$$\tilde{u} = u + w,$$

where w is the additive noise, and in most cases we consider white noise (i.e., $\mathbb{E}[w(t)] = 0$ and $\mathrm{Cov}(w(t), w(t+h)) = 0, \forall\ h \neq 0$). Additive white noises arise in many situations, for example, in algorithms involving stochastic gradients or differential privacy.

Multiplicative Noise: The second form of stochastic controller has multiplicative noise. That is:

$$\tilde{u} = (I + \mathcal{W}) \cdot u,$$

where $\mathcal{W}$ is a random matrix. This type of stochasticity can be used to model random communication graphs, partial participation, and communication sparsification.

Mixture of Noise: The third form of stochastic controller is the combination of the previous two, involving a *mixture* of additive and multiplicative noises. This setting can be used to model complex algorithms, e.g., FL algorithms that involve both differentially private noise and agent sampling; cf. (McMahan et al., 2018).

From the above-mentioned scenarios, we can abstract the following assumptions on the stochastic controllers:

PS 1 (Expected Control Signal) *The stochastic GCFL is an unbiased estimator of its deterministic counterpart:*

$$\mathbb{E}[\tilde{G}_g(\mathbf{x}, \mathbf{v}; A)] = G_g(\mathbf{x}, \mathbf{v}; A), \forall \mathbf{x} \in \mathbb{R}^{Nd_x}, \mathbf{v} \in^{Nd_v},$$

*and (**A**) the stochastic LCFL is also unbiased, satisfying:*

$$\mathbb{E}_t[\tilde{G}_\ell(s_i; f_i)] = G_\ell(s_i; f_i),\ \forall\, i \in [N], s_i \in \mathbb{R}^{d_x+d_v+d_z},$$

*or (**B**) the stochastic LCFL is biased: there exist positive constants C_1, C_2, σ_G satisfying the following:*

$$\mathbb{E}\left[\left\langle \tilde{G}_\ell(s_i; f_i), G_\ell(s_i; f_i)\right\rangle\right] \geq C_2 \left\|G_\ell(s_i; f_i)\right\|^2 - \sigma_G^2,$$

$$\left\|\mathbb{E}[\tilde{G}_\ell(s_i; f_i)]\right\|^2 \leq C_1,\ \forall\, i \in [N], s_i \in \mathbb{R}^{d_x+d_v+d_z}.$$

Note that the controller $G_g(\mathbf{y}(t); A)$ is linear in $\mathbf{y}(t)$, thus we can guarantee it is unbiased. However, the LCFL may be nonlinear or nonconvex; consequently, PS1(A) can be difficult to satisfy. Therefore, we make a relaxed assumption PS1(B), which allows certain degrees of bias and misalignment between the deterministic controller and its stochastic counterpart. It is easy to see that PS1(A) is a special case of (B) with $C_1 = \infty, C_2 = 1, \sigma_G = 0$.

PS 2 (Bounded Variance) *There exist positive constants $B_g, B_\ell, \sigma_g, \sigma_\ell$, such that the following hold:*

$$\mathbb{E}\left[\left\|\tilde{G}_\ell(s_i; f_i) - \mathbb{E}[\tilde{G}_\ell(s_i; f_i)]\right\|^2\right]$$
$$\leq B_\ell \left\|\mathbb{E}[\tilde{G}_\ell(s_i; f_i)]\right\|^2 + \sigma_\ell^2,\ \forall\, i \in [N], s_i \in \mathbb{R}^{d_x+d_v+d_z},$$
$$\mathbb{E}\left[\left\|\tilde{G}_g(\mathbf{x}, \mathbf{v}; A) - G_g(\mathbf{x}, \mathbf{v}; A)\right\|^2\right]$$
$$\leq B_g \left\|G_g(\mathbf{x}, \mathbf{v}; A)\right\|^2 + \sigma_g^2,\ \forall \mathbf{x} \in \mathbb{R}^{Nd_x}, \mathbf{v} \in R^{Nd_v}.$$

Note that if the stochasticity in the controller is an additive white noise, then it is easy to see that $B_\ell = 0, B_g = 0$ and PS1(A) is satisfied.

PS 3 (Independence) *The stochastic noise terms in the controllers are independent, satisfying the following:*

$$\mathrm{Cov}_t\left(\tilde{G}_g(\mathbf{x}(t), \mathbf{v}(t); A), \tilde{G}_\ell(s_i(t); f_i)\right) = 0.$$

Note that we only assume independence between the consensus and local control signals at time t, while the control signals at different times can be correlated.

4. Convergence Analysis

In this section, we analyze the theoretical behavior of the stochastic system described in Section 3.2. First, we introduce an energy-like function for the system:

$$\mathcal{E}(t) := f(\bar{\mathbf{x}}(t)) - f^\star + \left\|(I - R) \cdot \mathbf{y}(t)\right\|^2. \tag{6}$$

Note that $\mathcal{E}(t) \geq 0$ for all $\mathbf{s}(t) = [\mathbf{x}(t); \mathbf{v}(t); \mathbf{z}(t)]$.

Let us begin by assuming that the deterministic system satisfies the following property.

PD 4 (Descent of Deterministic System) *The difference of the energy function of the deterministic system satisfies:*

$$\begin{aligned}\mathcal{E}(t) - \mathcal{E}(0) \leq &- \sum_{r=0}^{t/\tau-1} \gamma_1(r\tau) \cdot \left\|\nabla f(\bar{\mathbf{x}}(r\tau)\right\|^2 \\ &- \sum_{r=0}^{t/\tau-1} \gamma_2(r\tau) \cdot \left\|(I - R) \cdot \mathbf{y}(r\tau)\right\|^2,\end{aligned} \tag{7}$$

where $\gamma_1(r\tau), \gamma_2(r\tau) > 0$ are coefficients depending on the choice of $\eta_\ell, \eta_g, \tau_\ell, \tau_g$.

This property immediately implies that the algorithm converges to the FOSP of the problem, in the sense that the following holds: the convergence error and consensus error are both decreasing to zero as the LHS is lower bounded by $-\mathcal{E}(0)$. Property PD4 appears to be strong compared with Properties PD1 – PD3, since it is about the entire sequence generated by the control system. We require that the deterministic system satisfies this property because: 1) This

is in fact a standard property that a wide range of deterministic algorithms can satisfy; 2) Having this property can help us focus on investigating the effect of various kinds of stochasticity on the system performance. To see point 1) above, we note that this property has been explicitly shown in algorithms such as DGD (Zeng & Yin, 2018)[Theorem 2], DGT (Di Lorenzo & Scutari, 2016)[Theorem 3], xFilter (Sun & Hong, 2019)[Theorem 5.1], and FedDP (Zhang et al., 2020)[Theorem 1 Case I]. Of course, when designing a *new* (stochastic) algorithm, this property has to be verified for its deterministic counterpart, before we move to analyze the entire stochastic system.

Next, we move on to characterize the impact of the stochasticity in the controllers satisfying PS1 - PS3. The key challenge is to characterize the deviations of $\mathcal{E}(t)$ caused by the system stochasticity in different discretization cases.

Case I: For Case I, $\tau_g = \tau_\ell > 0$. Let us denote the states at the r^{th} sampling time instance as $(\cdot)^r := (\cdot)(r\tau_\ell)$, then the discretized system can be written as:

$$\begin{aligned}\tilde{\mathbf{x}}^{r+1} &= \tilde{\mathbf{x}}^r - \eta_\ell'^r \cdot \tilde{u}^r_{\ell,x} - \eta_g'^r \cdot \tilde{u}^r_{g,x}\\ \tilde{\mathbf{v}}^{r+1} &= \tilde{\mathbf{v}}^r - \eta_\ell'^r \cdot \tilde{u}^r_{\ell,v} - \eta_g'^r \cdot \tilde{u}^r_{g,v}\\ \tilde{\mathbf{z}}^{r+1} &= \tilde{\mathbf{z}}^r - \eta_\ell'^r \cdot \tilde{u}^r_{\ell,z},\end{aligned} \tag{8}$$

where $\eta_\ell'^r = \tau_\ell \cdot \eta_\ell(r\tau_\ell)$, $\eta_g'^r = \tau_\ell \cdot \eta_g(r\tau_\ell)$.

Then, we have the following results:

Lemma 1 *Suppose the deterministic system satisfies PD1 - PD4, and the stochastic controllers satisfy PS2 and PS3. Consider the discretization Case I with $\tau_g = \tau_\ell > 0$.*
(A) *If PS1(A) is satisfied, then we have the following:*

$$\begin{aligned}\mathbb{E}[\tilde{\mathcal{E}}^t] - \mathcal{E}^0 \le &-\sum_{r=0}^{t-1} \underbrace{(\gamma_1^r - C_{11}^r)}_{=:\gamma_1'(r)} \cdot \mathbb{E}[\|\nabla f(\tilde{\mathbf{x}}^r)\|^2]\\ &-\sum_{r=0}^{t-1} \underbrace{(\gamma_2^r - C_{12}^r)}_{=:\gamma_2'(r)} \cdot \mathbb{E}[\|(I-R)\cdot\tilde{\mathbf{y}}^r\|^2]\\ &+ C_{13}(t)\sigma_g^2 + C_{14}(t)\sigma_\ell^2,\end{aligned} \tag{9}$$

where $C_{11}^r := B_\ell \cdot (C_x^2 + C_v^2) \cdot (1 + \frac{L_f}{2N}) \cdot (\eta_\ell'^r)^2$, $C_{12}^r := C_{11}^r L_f^2 + B_g \cdot (\eta_g'^r)^2 \cdot (1 + \frac{L_f}{2N})$, $C_{13}(t) := \sum_{r=0}^{t-1} (\eta_g'^r)^2 \cdot (1 + \frac{L_f}{2N})$, $C_{14}(t) := \sum_{r=0}^{t-1} (\eta_\ell'^r)^2 \cdot (1 + \frac{L_f}{2N})$.

(B) *If PS1(B) is satisfied, then we have the following:*

$$\begin{aligned}\mathbb{E}[\tilde{\mathcal{E}}^t] - \mathcal{E}^0 \le &-\sum_{r=0}^{t-1} \underbrace{(\gamma_1^r - C_{11}'^r)}_{:=\gamma_1'(r)} \cdot \mathbb{E}[\|\nabla f(\tilde{\mathbf{x}}^r)\|^2]\\ &-\sum_{r=0}^{t-1} \underbrace{(\gamma_2^r - C_{12}'^r)}_{:=\gamma_2'(r)} \cdot \mathbb{E}[\|(I-R)\cdot\tilde{\mathbf{y}}^r\|^2]\\ &+ C_{13}(t)\sigma_g^2 + C_{14}(t)\sigma_\ell^2 + C_{15}(t)C_1 + C_{16}(t)\sigma_G^2.\end{aligned}$$

where $C_{11}'^r, C_{12}'^r, C_{15}(t), C_{16}(t)$ are positive coefficients depending on $L, L_f, C_2, C_x, C_v, B_\ell, B_g, \eta_\ell'^r, \eta_g'^r$,.

The proofs and choices of the parameters for Lemma 1(A) and (B) are provided in Appendix B.1.1 and Appendix B.1.2 due to space limits. This lemma indicates that by using stochastic controllers, the system introduces extra perturbations. Compared with (A), the result in (B) has two extra error terms which are caused by the biased stochastic local controllers. The key point is to choose $\eta_\ell'^r, \eta_g'^r$ such that $\gamma_1'(r) > 0, \gamma_2'(r) > 0$ and minimize $\{C_{1i}(t)\}_{i=3}^6$, so that the error terms accumulate slower than the rate at which the first two terms decrease. This choice depends on the specification of the deterministic algorithm. Further, we have:

Theorem 1 *Suppose the deterministic system in Case I satisfies PD1 - PD4, with stochastic controllers satisfying PS1, PS2 and PS3. The algorithm converges with:*

$$\mathbb{E}\left[\left\|\nabla f(\tilde{\bar{\mathbf{x}}}^{r_1})\right\|^2 + \|(I-R)\cdot\tilde{\mathbf{y}}^{r_1}\|^2\right] \le \frac{\mathcal{E}^0 + C_3(t)}{\sum_{r=0}^{t-1}\gamma'(r)},$$

where $\gamma'(r) := \min\{\gamma_1'(r), \gamma_2'(r)\}$, $C_3(t) = C_{13}(t)\sigma_g^2 + C_{14}(t)\sigma_\ell^2$ for PS1(A) and $C_3(t) = C_{13}(t)\sigma_g^2 + C_{14}(t)\sigma_\ell^2 + C_{15}(t)C_1 + C_{16}(t)\sigma_G^2$ for PS1(B).

For Case II and Case III, similar results can be derived. Detailed derivations are provided in Appendix B.2.

In summary, starting with a convergent deterministic system, we can replace the controllers with their stochastic versions that satisfy properties PS1-PS3. The resulting stochastic systems not only slow down by a certain factor depending on C_{i1}, C_{i2}, but also suffers form additional error terms in C_3. Let us comment on these terms:

1) Suppose that PS1(A) is satisfied and $\sigma_g = \sigma_\ell = 0$ in PS2, i.e., the variance of the controller can be fully bounded by the size of the deterministic control signal, then $C_3 = 0$. Therefore, it only requires $C_{i1} < c\gamma_1, C_{i2} < c\gamma_2$ with constant $0 \le c < 1$ for the stochastic algorithm to converge. In this case, the convergence rate of the stochastic algorithm will have the same order as the baseline deterministic algorithm.

2) If $\sigma_g, \sigma_\ell > 0$, i.e., the variance of the controller stays constant, then we need to balance between the error term and the descent terms. In this case, the convergence rate of the stochastic algorithm may slow down in order, or lose it convergence. In Section 5.3, we use the DGT algorithm to demonstrate how the parameters are specified to balance the error and the convergence rate.

5. Applications of the Framework

In this section, we demonstrate the modeling capability of the proposed control system. We first show that a few important algorithmic features can be mapped to specific types of stochastic controllers. We then combine these controllers in different ways to construct a number of popular distributed algorithms. Finally, we use the DGT algorithm as an example to illustrate how the proposed framework fa-

cilitates new algorithm design.

5.1. Mapping Features to the Stochastic Controllers

We first discuss how a number of features that are desirable to distributed algorithms can be mapped to specific stochastic controllers, which satisfy PS1-PS3.

First, we discuss a few realizations of $\tilde{G}_g(\mathbf{y}(t); A)$:

• Randomized Communication Graph (RG): Suppose the communication graph $\mathcal{G}(t)$ is randomly time-varying. This can be caused by limited bandwidth or unreliable connection, so that at time t, the agents randomly choose a subset of their neighbours to broadcast local information, and gather the information from a possibly different random subset of neighbours (Koloskova et al., 2020; Yuan et al., 2021). In this case, $\tilde{G}_g(\mathbf{y}(t); A) := \tilde{W}_A(t)\mathbf{y}(t)$, where $\tilde{W}_A(t)$ is a random matrix satisfying $\mathbb{E}[\tilde{W}_A(t)] = W_A$ and if $(i,j) \notin \mathbf{E}$, $\tilde{W}_{A,ij}(t) = 0$. An extreme is that $\tilde{W}_A$ is diagonal and no communication happens. This case satisfies PS1 and PS2.

• Partial Agent Participation (PP): Partial agent participation often arises in FL, where at each communication round, only a subset of P agents send their updates to the server (Bonawitz et al., 2019; Acar et al., 2021). PP is a more practical approach than full agent aggregation and can be viewed as a special case of randomized communication graph $\tilde{G}_g(\mathbf{y}(t); A) := \tilde{W}_A(t)\mathbf{y}(t)$, where the averaging matrix takes the following form:

$$\tilde{W}_A(t) = \frac{\mathbb{1}_N \mathbf{B}^T(t)}{\mathbb{1}_N^T \mathbf{B}(t)},\ \mathbf{B}(t) \in \{0,1\}^N,\ \mathbb{E}[\mathbf{B}(t)] = \frac{P}{N}\mathbb{1}_N,$$

where $\mathbf{B}(t)$ is a length-N random vector. In this case, it satisfies PS1 that $\mathbb{E}[\tilde{W}_A(t)] = R$ and PS2 with $\sigma_g = 0$.

• Compressed Communication (CC): A different way of resolving the communication bandwidth issue is to reduce the data transmitted as each communication round by using compression methods such as (randomized) quantization and sparsification (Tang et al., 2018; 2020). The controller can be written as:

$$\tilde{G}_g(\mathbf{y}; A) := G_g(\mathcal{W}\mathbf{y}; A),\ \mathbb{E}[\mathcal{W}] = I,$$

where $\mathcal{W}$ is a diagonal multiplicative noise matrix for compression and satisfies PS1, PS2. For example, we can set $\mathcal{W}$ as the sparsification matrix with:

$$\mathcal{W}_{jj} = \begin{cases} \frac{1}{p}, & \text{w.p. } p, \\ 0, & \text{w.p. } 1-p, \end{cases}$$

where $p < 1$ denotes the compression rate (Basu et al., 2019); or set $\mathcal{W}$ as the quantization matrix with:

$$\mathcal{W}_{jj} = \begin{cases} \frac{\lceil \mathbf{y}_j \rceil}{\mathbf{y}_j}, & \text{w.p. } \frac{\mathbf{y}_j - \lfloor \mathbf{y}_j \rfloor}{\lceil \mathbf{y}_j \rceil - \lfloor \mathbf{y}_j \rfloor}, \\ \frac{\lfloor \mathbf{y}_j \rfloor}{\mathbf{y}_j}, & \text{w.p. } \frac{\lceil \mathbf{y}_j \rceil - \mathbf{y}_j}{\lceil \mathbf{y}_j \rceil - \lfloor \mathbf{y}_j \rfloor}, \end{cases}$$

where $\lceil \cdot \rceil$, $\lfloor \cdot \rfloor$ denote the upper and lower quantization levels (Koloskova et al., 2019a) which satisfies PS1 and PS2. These methods can efficiently save the communication on structured data.

• Differential Privacy Noise: One important motivation to implement distributed systems is to guarantee user data privacy. DP is a widely used notion for measuring privacy, because it provides strong guarantees, while being easily implementable (Abadi et al., 2016). The most popular mechanism to ensure DP is called the *Gaussian mechanism*, which adds noise to the algorithm outputs (Abadi et al., 2016). In a distributed setting, this mechanism can be viewed as adding noise to the local messages before they get transmitted. To model the DP noise, the stochastic controller can be written as

$$\tilde{G}_g(\mathbf{y}; A) := W_A \cdot (\mathbf{y} + \mathbf{w}_g),$$

where $\mathbf{w}_g \sim \mathcal{N}(0, \sigma^2 I)$ with $\sigma^2 = \Omega(\frac{pt\log(\delta^{-1})}{N\epsilon^2})$ capturing the privacy noise (McMahan et al., 2018).

Next, we discuss a few realizations of $\tilde{G}_\ell(s_i(t); f_i)$:

• Clipping: Note that when implementing differentially private algorithms, local clipping operation is usually needed to bound the algorithm sensitivity, which can be written as:

$$\text{clip}(G_{\ell,i}(s_i; f_i); c) := G_{\ell,i}(s_i; f_i) \cdot \max\left\{1, \frac{c}{\|G_{\ell,i}(s_i; f_i)\|}\right\},$$

where c denotes the clipping threshold. In this case, even if $G_{\ell,i}(s_i; f_i)$ is unbiased, the non-linear clipping operation will introduce extra biased noise (Chen et al., 2020) satisfying PS1(B) with $C_1 = c$, and C_2, σ_G depending on data distribution.

• Stochastic Gradient (SG): As mentioned before, state-of-the-art ML applications often use SGD based local updates. This can be easily translated to a stochastic local controller where the stochastic gradient is estimated on sampled data:

$$\tilde{u}_{i,\ell}(t) = \nabla f_i(x_i(t)) + \underbrace{\nabla f_i(x_i(t); \xi_i(t)) - \nabla f_i(x_i(t))}_{w_i(t)},$$

where $w_i(t)$ is the additive noise; $\xi_i(t)$ is drawn uniformly from the local dataset. So $\mathbb{E}[\tilde{\nabla} f_i(x_i(t))] = \nabla f_i(x_i(t))$ which satisfies PS1, and it is common to assume that $\text{Var}(w_i(t))$ satisfies PS2 (Lian et al., 2017; Lu et al., 2019; Karimireddy et al., 2020).

• Zeroth-order Optimization (ZO): the zeroth-order optimization method have been developed in recent years in the setting that only the loss values $f_i(x_i)$ can be accessed (Yuan et al., 2015; Sahu et al., 2018; Hajinezhad et al., 2019). One can use zeroth-order method to approximate the gradient:

$$\tilde{\nabla} f_i(x_i) := \frac{f_i(x_i + \delta h) - f_i(x_i - \delta h)}{2h}\delta,$$

where δ uniformly samples from the unit sphere and h is a sufficiently small scalar. Similar to the previous case, we have:

$$\tilde{u}_{i,\ell}(t) = \nabla f_i(x_i(t)) + \underbrace{\tilde{\nabla} f_i(x_i(t)) - \nabla f_i(x_i(t))}_{w_i(t)},$$

where $w_i(t)$ is a *biased* additive noise (Yuan et al., 2015).

Note that different forms of noises can be combined to-

gether for more complex applications, e.g., in DP, we may combine DP with Clipping and SG for better performance.

5.2. Algorithm Classification

In this subsection, we discuss some popular distributed algorithms and how they fall into the proposed framework.

- We first start with DT algorithms, which belongs to Case I: DSGD (Lian et al., 2017) uses stochastic gradient as LCFL with deterministic GCFL. Its variations include (Koloskova et al., 2020) which studies random communication graph, (Koloskova et al., 2019b;a) with communication compression, and D-(DP)2SGD (Yuan et al., 2021) with differential privacy. GNSD (Lu et al., 2019) uses gradient tracking on stochastic gradient, and ZONE (Hajinezhad et al., 2019) uses zeroth-order optimization for gradient estimation.
- FL is another popular class of distributed algorithms, which uses discretization Case II. Popular algorithms include FedPD (Zhang et al., 2020) that implements the ADMM algorithm with stochastic gradient as local solver, and uses random aggregation scheme to save communication while FedDyn (Acar et al., 2021) considers partial client participation. Scaffold (Karimireddy et al., 2020) tracks local stochastic gradients to correct the update direction; DP-FedAvg (McMahan et al., 2018; Zhang et al., 2021) apply differential privacy to FedAvg; Qsparse-Local-SGD uses communication sparsification on FedAvg (Basu et al., 2019).
- Finally, we give an example algorithm trying to optimize the convergence rate dependencies via multi-step communication in Case III: DSAGD (Rogozin et al., 2021) uses stochastic gradient and multi-step averaging on random communication graphs to accelerate consensus.

We summarize the above discussions in Table 2, where we specify the discretization cases and the stochasticities in each algorithm. More detailed algorithmic correspondence are included in Appendix A.2

5.3. Algorithm Design: A Case Study

In this subsection, we take the decentralized gradient tracking (DGT) algorithm as an example to illustrate how the framework can be applied to design new algorithms for different applications. In specific, we modify the DGT algorithm to include features such as SG, RG and DP, and name the resulting algorithms as Distributed Stochastic Gradient Tracking (DSGT) (which is the same as GNSD (Lu et al., 2019)), Distributed Dynamic-graph Gradient Tracking (D^{2}GT) and Differentially Private DSGT (DP-DSGT). By verifying PD1-PD4 and PS1-PS3 for each case, we have the following informal theoretical result:

Corollary 1 (Informal) *With properly chosen stepsize, the expected stationarity gaps of DSGT, D^{2}GT, and DP-DSGT converge with rates* $\mathcal{O}(\frac{\log(t)}{\sqrt{t}})$, $\mathcal{O}(\frac{1}{t})$, *and* $\mathcal{O}(\frac{\sqrt{d_x+d_v}\log(\delta^{-1})}{N\epsilon})$ *respectively, where the expectation is taken over the iterations, and DP-DSGT satisfies the* (ϵ, δ)*-differential privacy.*

Algorithm	Discretization	Stochasticity
DSGD	Case I	SG, CC, RG
GNSD	Case I	SG
D-(DP)2SGD	Case I	SG, DP, RG
ZONE	Case I	ZO
FedPD/FedDyn	Case II	SG, RG/PP
Scaffold	Case II	SG, PP
Qsparse-Local-SGD	Case II	SG, CC
DP-FedAvg	Case II	SG, DP, PP
DSAGD	Case III	SG, RG

Table 2: Summary of the distributed stochastic algorithms, with discretization cases and stochasticity in the controller.

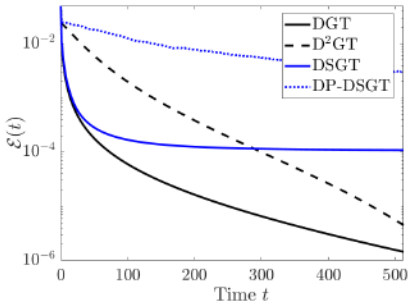


Figure 3: The convergence of the stationarity gap of DGT, D^{2}GT, GSGT and DP-DSGT.

We can see that with multiplicative noise, D^{2}GT has the fastest convergence rate, which is essentially the same order as DGT; DSGT converges slower due to the additive noise in SG, and recovers the rate obtained in (Lu et al., 2019); DP-DSGT has a constant error independent of t due to the additive noises caused by DP.

Numerical results for the algorithms on the non-convex regularized logistic regression problem (Antoniadis et al., 2011) are shown in Figure 3. In the experiment, we choose the stepsizes based on the theoretical result, i.e., η_g', η_ℓ' as constants for DGT, D^{2}GT; and $\eta_\ell'' = \mathcal{O}(1/\sqrt{\tau})$ for GNSD and DP-GNSD. It can be observed that D^{2}GT has the same convergence rate as DGT with a constant slow down, while GNSD and DP-GNSD have slower convergence rates. Due to page limitation, we refer to Appendix C for detailed discussions on the algorithm modifications, theoretical analyses and experiment settings and additional results.

6. Conclusion

In this work, we have proposed a feedback-control system to model distributed optimization algorithms from the multi-rate stochastic control perspective. We have shown that the multi-rate stochastic control system can represent a variety of distributed stochastic algorithms. Illustrative examples demonstrate how the system can help understand existing algorithms and design new algorithms.

Acknowledgements

We thank the anonymous reviewers for valuable feedback on the merit of the work, and helpful suggestions on improving the presentation. M. Hong and X. Zhang are supported in part by NSF grant CIF-1910385 and AFOSR grant 19RT0424.

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 308–318, 2016.

Acar, D. A. E., Zhao, Y., Navarro, R. M., Mattina, M., Whatmough, P. N., and Saligrama, V. Federated learning based on dynamic regularization. *arXiv preprint arXiv:2111.04263*, 2021.

Antoniadis, A., Gijbels, I., and Nikolova, M. Penalized likelihood regression for generalized linear models with non-quadratic penalties. *Annals of the Institute of Statistical Mathematics*, 63(3):585–615, 2011.

Basu, D., Data, D., Karakus, C., and Diggavi, S. Qsparse-local-sgd: Distributed sgd with quantization, sparsification and local computations. *Advances in Neural Information Processing Systems*, 32, 2019.

Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konečnỳ, J., Mazzocchi, S., McMahan, H. B., et al. Towards federated learning at scale: System design. *arXiv preprint arXiv:1902.01046*, 2019.

Chang, T.-H., Hong, M., Wai, H.-T., Zhang, X., and Lu, S. Distributed learning in the nonconvex world: From batch data to streaming and beyond. *IEEE Signal Processing Magazine*, 37(3):26–38, 2020.

Chau, N. H., Moulines, É., Rásonyi, M., Sabanis, S., and Zhang, Y. On stochastic gradient langevin dynamics with dependent data streams: The fully nonconvex case. *SIAM Journal on Mathematics of Data Science*, 3(3): 959–986, 2021.

Chen, T., Giannakis, G. B., Sun, T., and Yin, W. Lag: lazily aggregated gradient for communication-efficient distributed learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 5055–5065, 2018.

Chen, X., Wu, S. Z., and Hong, M. Understanding gradient clipping in private sgd: A geometric perspective. *Advances in Neural Information Processing Systems*, 33, 2020.

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

El Mekkaoui, K., Mesquita, D., Blomstedt, P., and Kaski, S. Federated stochastic gradient langevin dynamics. In *Uncertainty in Artificial Intelligence*, pp. 1703–1712. PMLR, 2021.

França, G., Robinson, D. P., and Vidal, R. A dynamical systems perspective on nonsmooth constrained optimization. *arXiv preprint arXiv:1808.04048*, 2018.

Hajinezhad, D., Hong, M., and Garcia, A. Zone: Zeroth-order nonconvex multiagent optimization over networks. *IEEE Transactions on Automatic Control*, 64(10):3995–4010, 2019.

Jakovetić, D. A unification and generalization of exact distributed first-order methods. *IEEE Transactions on Signal and Information Processing over Networks*, 5(1):31–46, 2018.

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S., and Suresh, A. T. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pp. 5132–5143. PMLR, 2020.

Khan, M. R. B., Jidin, R., and Pasupuleti, J. Multi-agent based distributed control architecture for microgrid energy management and optimization. *Energy Conversion and Management*, 112:288–307, 2016.

Koloskova, A., Lin, T., Stich, S. U., and Jaggi, M. Decentralized deep learning with arbitrary communication compression. In *International Conference on Learning Representations*, 2019a.

Koloskova, A., Stich, S., and Jaggi, M. Decentralized stochastic optimization and gossip algorithms with compressed communication. In *International Conference on Machine Learning*, pp. 3478–3487. PMLR, 2019b.

Koloskova, A., Loizou, N., Boreiri, S., Jaggi, M., and Stich, S. A unified theory of decentralized sgd with changing topology and local updates. In *International Conference on Machine Learning*, pp. 5381–5393. PMLR, 2020.

Kuo, B. C. Digital control systems, 1980.

Li, T., Sahu, A. K., Talwalkar, A., and Smith, V. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

Lian, X., Zhang, C., Zhang, H., Hsieh, C.-J., Zhang, W., and Liu, J. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized

parallel stochastic gradient descent. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 5336–5346, 2017.

Liu, T., Jiang, Z.-P., and Hill, D. J. *Nonlinear control of dynamic networks*. CRC Press, 2018.

Lu, S., Zhang, X., Sun, H., and Hong, M. Gnsd: A gradient-tracking based nonconvex stochastic algorithm for decentralized optimization. In *2019 IEEE Data Science Workshop (DSW)*, pp. 315–321. IEEE, 2019.

Lu, Y. and De Sa, C. Optimal complexity in decentralized training. In *International Conference on Machine Learning*, pp. 7111–7123. PMLR, 2021.

McMahan, H. B., Ramage, D., Talwar, K., and Zhang, L. Learning differentially private recurrent language models. In *International Conference on Learning Representations*, 2018.

Muehlebach, M. and Jordan, M. A dynamical systems perspective on nesterov acceleration. In *International Conference on Machine Learning*, pp. 4656–4662, 2019.

Orvieto, A. and Lucchi, A. Continuous-time models for stochastic optimization algorithms. *Advances in Neural Information Processing Systems*, 32, 2019.

Rogozin, A., Bochko, M., Dvurechensky, P., Gasnikov, A., and Lukoshkin, V. An accelerated method for decentralized distributed stochastic optimization over time-varying graphs. *arXiv preprint arXiv:2103.15598*, 2021.

Sahu, A. K., Jakovetic, D., Bajovic, D., and Kar, S. Distributed zeroth order optimization over random networks: A kiefer-wolfowitz stochastic approximation approach. In *2018 IEEE Conference on Decision and Control (CDC)*, pp. 4951–4958. IEEE, 2018.

Scaman, K., Bach, F., Bubeck, S., Lee, Y. T., and Massoulié, L. Optimal algorithms for smooth and strongly convex distributed optimization in networks. In *international conference on machine learning*, pp. 3027–3036. PMLR, 2017.

Sun, H. and Hong, M. Distributed non-convex first-order optimization and information processing: Lower complexity bounds and rate optimal algorithms. *IEEE Transactions on Signal processing*, 67(22):5912–5928, 2019.

Sun, J., Chen, T., Giannakis, G. B., Yang, Q., and Yang, Z. Lazily aggregated quantized gradient innovation for communication-efficient federated learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

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

Tang, H., Gan, S., Zhang, C., Zhang, T., and Liu, J. Communication compression for decentralized training. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 7663–7673, 2018.

Tang, Z., Shi, S., Chu, X., Wang, W., and Li, B. Communication-efficient distributed deep learning: A comprehensive survey. *arXiv preprint arXiv:2003.06307*, 2020.

Wang, J. and Elia, N. A control perspective for centralized and distributed convex optimization. In *2011 50th IEEE conference on decision and control and European control conference*, pp. 3800–3805. IEEE, 2011.

Yuan, D., Ho, D. W., and Xu, S. Zeroth-order method for distributed optimization with approximate projections. *IEEE transactions on neural networks and learning systems*, 27(2):284–294, 2015.

Yuan, K., Ling, Q., and Yin, W. On the convergence of decentralized gradient descent. *SIAM Journal on Optimization*, 26(3):1835–1854, 2016.

Yuan, K., Xu, W., and Ling, Q. Can primal methods outperform primal-dual methods in decentralized dynamic optimization? *arXiv preprint arXiv:2003.00816*, 2020.

Yuan, Y., Zou, Z., Li, D., Yan, L., Yu, D., and Duan, Z. D-(dp)2sgd: Decentralized parallel sgd with differential privacy in dynamic networks. *Wirel. Commun. Mob. Comput.*, 2021, jan 2021. ISSN 1530-8669. doi: 10.1155/2021/6679453. URL `https://doi.org/10.1155/2021/6679453`.

Zeng, J. and Yin, W. On nonconvex decentralized gradient descent. *IEEE Transactions on signal processing*, 66 (11):2834–2848, 2018.

Zhang, X., Hong, M., Dhople, S., Yin, W., and Liu, Y. Fedpd: A federated learning framework with optimal rates and adaptivity to non-iid data. *arXiv preprint arXiv:2005.11418*, 2020.

Zhang, X., Chen, X., Hong, M., Wu, Z. S., and Yi, J. Understanding clipping for federated learning: Convergence and client-level differential privacy. *arXiv preprint arXiv:2106.13673*, 2021.

A. Additional Discussions

In this section, we provide additional discussions missing in the main paper.

A.1. Related Works in Dynamic Systems

In this subsection, we provide additional discussion on existing works, which are related to using control theory, and dynamic system to analyze distributed algorithms.

Controlling the stochastic system using robust control has been a standard approach in the control theory (Liu et al., 2018). More recent works such as (Jakovetić, 2018) generalizes the small gain theorem to nonlinear control systems to analyze the system stability with stochasticity. Distributed control system has been studied for optimizing global performance in distributed energy resources applications (Khan et al., 2016). Researches have shown that centralized and decentralized deterministic optimization algorithms (Wang & Elia, 2011; França et al., 2018; Muehlebach & Jordan, 2019) can be analyzed as dynamic systems. However, these works are restricted to convex optimization with deterministic controllers in continuous time, and fail to capture the impact of the "multi-rate" discretization, thus cannot cover the FL and algorithms that performs multiple consensus steps (Scaman et al., 2017; Lu & De Sa, 2021; Rogozin et al., 2021).

From the continuous-time perspective, there are a series of related researches focus on both gradient and stochastic gradient flow algorithms. The convergence rate of the non-convex stochastic gradient flow algorithm has been studied in (Orvieto & Lucchi, 2019) as the continuous-time counterpart of stochastic gradient descent algorithm in centralized setting. Some recent works focus on analyzing the stochastic gradient Langevin dynamics (El Mekkaoui et al., 2021; Chau et al., 2021) which are closely related to the stochastic gradient descent algorithms in both centralized and distributed settings. However, they are hard to be generalized to other stochastic algorithms.

A.2. Algorithm Discussion

In this part, we provide some concrete examples on how the existing algorithms are covered by the proposed model.

First we start with the ZONE algorithm (Hajinezhad et al., 2019) in decentralized training setting:

The update steps of ZONE are:

$$\mathbf{x}^{r+1} = \mathbf{x}^r - \rho \cdot (\mathbf{v}^r + W\mathbf{x}^r) - \eta'_\ell \tilde{\nabla} f(\mathbf{x}^r)$$
$$\mathbf{v}^{r+1} = \mathbf{v}^r + W\mathbf{x}^{r+1},$$

where $W = A^T A$, $\tilde{\nabla} f(\mathbf{x}^r)$ is the stochastic zeroth-order estimation of $\nabla f(\mathbf{x}^r)$. It is easy to see that the corresponding continuous-time deterministic controllers are:

$$u_g(t) = \begin{bmatrix} \rho W & \rho I \\ -W & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{v}(t) \end{bmatrix},$$
$$u_{i,\ell}(t) = \begin{bmatrix} \nabla f(\mathbf{x}(t)) \\ 0 \end{bmatrix}.$$

ZONE corresponds to discretization Case I with $\tau_g = \tau_\ell = 1$, and has zeroth-order gradient as stochastic LCFL.

Second, we provide the mapping for FedPD (Zhang et al., 2020) and FedDyn (Acar et al., 2021) in the federated learning setting where the communication graph can be viewed as a complete graph, and $W_A = I - R$:

The update of FedPD is given by (Zhang et al., 2020):

$$\mathbf{x}^{r,q+1} = \mathbf{x}^{r,q} - \eta'_\ell \tilde{\nabla} f(\mathbf{x}^{r,q}; \xi^{r,q}) + \eta'_g \cdot (\rho \cdot (\mathbf{x}^{r,q} - \mathbf{w}^{r,q}) + \mathbf{v}^{r,0})$$
$$\mathbf{v}^{r,q+1} = \begin{cases} \mathbf{v}^{r,q} + \eta'_g \cdot (\mathbf{x}^{r,q+1} - \mathbf{w}^{r,q}), & (q+1) = Q \\ \mathbf{v}^{r,q}, & (q+1) \neq Q, \end{cases}$$
$$\mathbf{w}^{r,q+1} = \begin{cases} \frac{\eta'_g}{p} R \cdot (2\mathbf{x}^{r,q+1} - \mathbf{w}^{r,q}), & (q+1) = Q, \text{ w.p. } p \\ 2\mathbf{x}^{r,q+1} - \mathbf{w}^{r,q}, & (q+1) = Q, \text{ w.p. } 1-p \\ \mathbf{w}^{r,q}, & (q+1) \neq Q, \end{cases}$$

where $\tilde{\nabla} f(\mathbf{x}^{r,q}; \xi^{r,q})$ denotes the stochastic gradient estimated on samples $\xi^{r,q}$. Observe that $\mathbf{w}$ tracks $R\mathbf{x}$ and update with probability p, so in continuous time, we can replace $\mathbf{w}$ with $R\mathbf{x}$, and obtain the following continuous-time controllers:

$$u_g(t) = \begin{bmatrix} \rho \cdot (I - R) & \rho I \\ -(I - R) & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{v}(t) \end{bmatrix},$$
$$u_{i,\ell}(t) = \begin{bmatrix} \nabla f(\mathbf{x}(t)) \\ 0 \end{bmatrix}.$$

FedPD corresponds to discretization Case II with $\tau_g = Q\tau_\ell = 1$, and has both stochastic gradient as stochastic LCFL and random communication graph

$$\tilde{W}_A = \begin{cases} \begin{bmatrix} \rho \cdot (I - R/p) & \rho I \\ -(I - R/p) & 0 \end{bmatrix} & \text{w.p. } p, \\ \begin{bmatrix} \rho I & \rho I \\ -I & 0 \end{bmatrix} & \text{w.p. } 1-p, \end{cases}$$

in the stochastic GCFL.

The update of FedDyn is given by (Acar et al., 2021):

$$\mathbf{x}^{r,q+1} = \mathbf{x}^{r,q} - \eta'_\ell \tilde{\nabla} f(\mathbf{x}^{r,q}; \xi^{r,q}) + \eta'_g \cdot (\rho \cdot (\mathbf{x}^{r,q} - \mathbf{w}^{r,q}) + \mathbf{v}^{r,0})$$
$$\mathbf{v}^{r,q+1} = \begin{cases} \mathbf{v}^{r,q} + \eta'_g \cdot (\mathbf{x}^{r,q+1} - \mathbf{w}^{r,q}), & (q+1) = Q \\ \mathbf{v}^{r,q}, & (q+1) \neq Q, \end{cases}$$
$$\mathbf{w}^{r,q+1} = \begin{cases} \tilde{R} \cdot (2\mathbf{x}^{r,q+1} - \mathbf{w}^{r,q}), & (q+1) = Q, \\ \mathbf{w}^{r,q}, & (q+1) \neq Q, \end{cases}$$

where $\tilde{\nabla} f(\mathbf{x}^{r,q}; \xi^{r,q})$ denotes the stochastic gradient estimated on samples $\xi^{r,q}$, and $\tilde{R} := \frac{\mathbb{1}_N \mathbf{B}^T}{\mathbb{1}_N^T \mathbf{B}}$, $\mathbf{B} \in \{0,1\}^N$ is a

random vector denotes the partial participation pattern with $\mathbb{E}[\tilde{R}] = R$. Observe that $\mathbf{w}$ tracks $R\mathbf{x}$ in expectation, so in continuous time we can replace $\mathbf{w}$ with $R\mathbf{x}$, and obtain the following deterministic controllers:

$$u_g(t) = \begin{bmatrix} \rho \cdot (I - R) & \rho I \\ -(I - R) & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{v}(t) \end{bmatrix},$$
$$u_{i,\ell}(t) = \begin{bmatrix} \nabla f(\mathbf{x}(t)) \\ 0 \end{bmatrix}.$$

FedDyn corresponds to discretization Case II with $\tau_g = Q\tau_\ell = 1$, and has both stochastic gradient as stochastic LCFL and random communication graph

$$\tilde{W}_A = \begin{bmatrix} \rho \cdot (I - \tilde{R}) & \rho I \\ -(I - \tilde{R}) & 0 \end{bmatrix},$$

in the stochastic GCFL.

Lastly, we map the DSAGD algorithm (Rogozin et al., 2021) to our system:

The update of DSAGD is given by (Rogozin et al., 2021):

$$\mathbf{x}^{r,k+1} = \begin{cases} \mathbf{x}^{r,k} - \eta'_\ell \cdot (\mathbf{x}^{r,k} - \mathbf{v}^{r,k+1}), & k+1 = K \\ \mathbf{x}^{r,k}, & k+1 \neq K \end{cases},$$
$$\mathbf{v}^{r,k+1} = \tilde{W}^{r,k} \cdot (\alpha^k \mathbf{x}^{r,0} + (1 - \alpha^k) \cdot \mathbf{v}^{r,0}) - \alpha^k \beta^r \tilde{\nabla} f(\mathbf{z}^{r,0}; \xi^r)$$
$$\mathbf{z}^{r,k+1} = \begin{cases} \mathbf{z}^{r,k} - \eta'_\ell \cdot (\mathbf{x}^{r,k+1} - \mathbf{v}^{r,k+1}), & k+1 = K \\ \mathbf{z}^{r,k}, & k+1 \neq K \end{cases},$$

where $\tilde{\nabla} f(\mathbf{z}^{r,0}; \xi^r)$ denotes the stochastic gradient estimated on samples ξ^r, and $\tilde{W}^{r,k}$ are random mixing matrices. We can obtain the following deterministic controller:

$$u_g(t) = \begin{bmatrix} 0 & 0 \\ -\alpha(t) \cdot W & I - (1 - \alpha(t)) \cdot W \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{v}(t) \end{bmatrix},$$
$$u_{i,\ell}(t) = \begin{bmatrix} \mathbf{x}(t) - \mathbf{v}(t) \\ \alpha(t) \cdot \beta(t) \cdot \nabla f(\mathbf{z}(t)) \\ \mathbf{x}(t) - \mathbf{v}(t) \end{bmatrix}.$$

DSAGD corresponds to discretization Case III with $\tau_\ell = K\tau_g > 0$, and has both stochastic gradient as stochastic LCFL and random communication graph

$$\tilde{W}_A = \begin{bmatrix} 0 & 0 \\ -\alpha^k \tilde{W}^{r,k} & (I - (1 - \alpha^k) \cdot \tilde{W}^{r,k}) \end{bmatrix},$$

in the stochastic GCFL.

Algorithm connections: Interestingly, we can observe that ZONE, FedPD and FedDyn has almost the same deterministic continuous-time controllers, where the only difference is the the mixing matrix $W = R$ in FL. These three algorithms distinguish from each other by having different sampling rates and introducing different forms of stochasticities.

B. Detailed Discussions for Section 4

In this section, we provide the proof for the lemmas in Section 4. Before we start, let us introduce some useful relations:

$$\langle a, b \rangle = \frac{1}{2\alpha} \|a\|^2 + \frac{\alpha}{2} \|b\|^2 - \frac{1}{2} \left\| \frac{1}{\sqrt{\alpha}} a + \sqrt{\alpha} b \right\|^2 \leq \frac{1}{2\alpha} \|a\|^2 + \frac{\alpha}{2} \|b\|^2 \tag{10}$$
$$(I - R)^2 = I - 2R + R^2 = I - R. \tag{11}$$

B.1. Proofs for Case I

We first present the proof for Lemma 1 in Case I.

B.1.1. Case I: Lemma 1(a)

The proof for Lemma 1(A) is straightforward. We first write the difference between the consecutive energy functions as:

$$\mathbb{E}_r \left[\tilde{\mathcal{E}}^{r+1} - \tilde{\mathcal{E}}^r \right] = \underbrace{\mathbb{E}_r \left[\tilde{\mathcal{E}}^{r+1} \right] - \mathcal{E}^{r+1}}_{\Delta^{r+1}} + \underbrace{\mathcal{E}^{r+1} - \tilde{\mathcal{E}}^r}_{\text{term I}}, \tag{12}$$

where we can apply PD4 to bound the sum of term I. The main challenge is to bound Δ^{r+1}. This can be proceed by the following:

$$\begin{aligned}
\mathbb{E}_r[\tilde{\mathcal{E}}^{r+1} - \mathcal{E}^{r+1}] &= \mathbb{E}_r \left[f(\tilde{\mathbf{x}}^{r+1}) - f(\bar{\mathbf{x}}^{r+1}) \right] \\
&\quad + \mathbb{E}_r \left[\left\| (I - R) \cdot \tilde{\mathbf{y}}^{r+1} \right\|^2 - \left\| (I - R) \cdot \mathbf{y}^{r+1} \right\|^2 \right] \\
&\overset{(i)}{\leq} \mathrm{Var}_r((I - R) \cdot \tilde{\mathbf{y}}^{r+1}) + \frac{L_f}{2} \mathbb{E}_r(\left\| \tilde{\bar{\mathbf{x}}}^{r+1} - \bar{\mathbf{x}}^{r+1} \right\|^2) \\
&\quad + \mathbb{E}_r \left[\langle \nabla f(\bar{\mathbf{x}}^{r+1}), \tilde{\bar{\mathbf{x}}}^{r+1} - \bar{\mathbf{x}}^{r+1} \rangle \right] \\
&\overset{(ii)}{\leq} \left(1 + \frac{L_f}{2N} \right) \cdot \mathrm{Var}_r(\tilde{\mathbf{y}}^{r+1}) \\
&\overset{(iii)}{=} \left(1 + \frac{L_f}{2N} \right) \cdot \mathrm{Var}_r(\eta_g'^r \tilde{u}_g^r + \eta_\ell'^r \tilde{u}_{\ell,y}^r) \\
&\overset{(PS3)}{=} \left(1 + \frac{L_f}{2N} \right) \\
&\quad \times \left((\eta_g'^r)^2 \cdot \mathrm{Var}_r(\tilde{u}_g^r) + (\eta_\ell'^r)^2 \cdot \mathrm{Var}_r(\tilde{u}_{\ell,y}^r) \right) \\
&\overset{(PS2)}{\leq} \left(1 + \frac{L_f}{2N} \right) \cdot (\eta_g'^r)^2 \cdot \left(B_g \left\| u_g^r \right\|^2 + \sigma_g^2 \right) \\
&\quad + \left(1 + \frac{L_f}{2N} \right) \cdot (\eta_\ell'^r)^2 \cdot \left(B_\ell \left\| u_{\ell,y}^r \right\|^2 + \sigma_\ell^2 \right) \\
&\overset{(iv)}{\leq} \left(1 + \frac{L_f}{2N} \right) \cdot (\eta_g'^r)^2 \cdot (B_g \left\| (I - R) \cdot \tilde{\mathbf{y}}^r \right\|^2 + \sigma_g^2) \\
&\quad + \left(1 + \frac{L_f}{2N} \right) \cdot (\eta_\ell'^r)^2 \cdot \Big(B_\ell (C_x^2 + C_v^2) \\
&\quad \cdot \left(\left\| \nabla f(\tilde{\bar{\mathbf{x}}}^r) \right\|^2 + L_f^2 \left\| (I - R) \cdot \tilde{\mathbf{x}}^r \right\|^2 \right) + \sigma_\ell^2 \Big),
\end{aligned}$$

where in (i) we apply A2 to the first two terms; in (ii) we apply PS1(A) to the last term as $\mathbb{E}_r\,\tilde{\bar{\mathbf{x}}}^{r+1} = \bar{\mathbf{x}}^{r+1}$ and merge the other two terms by using the fact that $\left\|\tilde{\bar{\mathbf{x}}}^r - \bar{\mathbf{x}}^r\right\|^2 \le \frac{1}{N}\left\|\tilde{\mathbf{x}}^r - \mathbf{x}^r\right\|^2$ and $\mathbf{x}$ is a sub-vector of $\mathbf{y}$; in (iii) we apply the update steps of $\tilde{\mathbf{y}}^r$ in (8); in (iv) we apply PD1 to the first term and bound the third term by

$$\begin{aligned}
&\left\|u^r_{\ell,y}\right\|^2 \overset{(PD3)}{\le} (C_x^2 + C_v^2)\left\|\nabla f(\tilde{\mathbf{x}}^r)\right\|^2 \qquad (13)\\
&\le 2(C_x^2 + C_v^2)\cdot\left(\left\|\nabla f(\tilde{\bar{\mathbf{x}}}^r)\right\|^2 + \left\|\nabla f(\tilde{\mathbf{x}}^r) - \nabla f(\tilde{\bar{\mathbf{x}}}^r)\right\|^2\right)\\
&\overset{(A2)}{\le} 2(C_x^2 + C_v^2)\cdot\left(\left\|\nabla f(\tilde{\bar{\mathbf{x}}}^r)\right\|^2 + L_f^2\left\|\tilde{\mathbf{x}}^r - \tilde{\bar{\mathbf{x}}}^r\right\|^2\right)\\
&= 2(C_x^2 + C_v^2)\cdot\left(\left\|\nabla f(\tilde{\bar{\mathbf{x}}}^r)\right\|^2 + L_f^2\left\|(I-R)\cdot\tilde{\mathbf{x}}^r\right\|^2\right)
\end{aligned}$$

Finally, substitute the above result into Δ^{r+1} in (12) and apply PD4, then Lemma 1(A) is proved.

B.1.2. Case I: Lemma 1(B)

In Lemma 1(B), the key step is to bound:

$$\begin{aligned}
\Delta^{r+1} &= \mathbb{E}_r\left[\tilde{\mathcal{E}}^{r+1}\right] - \mathcal{E}^{r+1}\\
&= \underbrace{\mathbb{E}_r\left[\tilde{\mathcal{E}}^{r+1}\right] - f(\mathbb{E}_r\,\tilde{\mathbf{x}}^{r+1}) - \left\|(I-R)\cdot\mathbb{E}_r\,\tilde{\mathbf{y}}^r\right\|^2}_{\Delta_A} \qquad (14)\\
&\quad + \underbrace{f(\mathbb{E}_r\,\tilde{\mathbf{x}}^{r+1}) + \left\|(I-R)\cdot\mathbb{E}_r\,\tilde{\mathbf{y}}^r\right\|^2 - \mathcal{E}^{r+1}}_{\Delta_B}.
\end{aligned}$$

First, we bound Δ_A, which is the same as Δ^{r+1} in the previous case:

$$\begin{aligned}
\Delta_A &\le (1+\frac{L_f}{2N})\cdot\mathrm{Var}_r(\eta'^r_\ell\,\tilde{u}^r_\ell + \eta'^r_g\,\tilde{u}^r_g)\\
&\overset{(PS2)}{\le} (1+\frac{L_f}{2N})\cdot(\eta'^r_\ell)^2(B_\ell\left\|\mathbb{E}_r\,\tilde{u}^r_\ell\right\|^2 + \sigma_\ell^2)\\
&\quad + (1+\frac{L_f}{2N})\cdot(\eta'^r_g)^2(B_g\left\|\mathbb{E}_r\,\tilde{u}^r_g\right\|^2 + \sigma_g^2)\\
&\overset{(i)}{\le} (1+\frac{L_f}{2N})\cdot(\eta'^r_\ell)^2(B_\ell C_1 + \sigma_\ell^2)\\
&\quad + (1+\frac{L_f}{2N})\cdot(\eta'^r_g)^2(B_g\left\|(I-R)\cdot\tilde{\mathbf{y}}^r\right\|^2 + \sigma_g^2),
\end{aligned}$$

where in (i) we apply PS1(B) to bound the first term and PD1 to bound $\left\|\mathbb{E}_r\,\tilde{u}^r_g\right\|^2$.

We then bound Δ_B by:

$$\begin{aligned}
\Delta_B &\overset{(A2)}{\le} \left\langle\nabla f(\bar{\mathbf{x}}^{r+1}), \mathbb{E}_r\,\tilde{\bar{\mathbf{x}}}^{r+1} - \bar{\mathbf{x}}^{r+1}\right\rangle\\
&\quad + \frac{L_f}{2}\left\|\mathbb{E}_r\,\tilde{\bar{\mathbf{x}}}^{r+1} - \bar{\mathbf{x}}^{r+1}\right\|^2\\
&\quad + 2\left\langle(I-R)\cdot\mathbf{y}^{r+1}, (I-R)\cdot(\mathbb{E}_r\,\tilde{\mathbf{y}}^{r+1} - \mathbf{y}^{r+1})\right\rangle\\
&\quad + \left\|(I-R)\cdot(\mathbb{E}_r\,\tilde{\mathbf{y}}^{r+1} - \mathbf{y}^{r+1})\right\|^2\\
&\overset{(i)}{\le} \left\langle\nabla f(\bar{\mathbf{x}}^{r+1}) - \nabla f(\tilde{\bar{\mathbf{x}}}^r), \mathbb{E}_r\,\tilde{\bar{\mathbf{x}}}^{r+1} - \bar{\mathbf{x}}^{r+1}\right\rangle\\
&\quad + \left\langle\nabla f(\tilde{\bar{\mathbf{x}}}^r), \mathbb{E}_r\,\tilde{\bar{\mathbf{x}}}^{r+1} - \bar{\mathbf{x}}^{r+1}\right\rangle\\
&\quad + \left(1+\frac{L_f}{2N}\right)\left\|\mathbb{E}_r\,\tilde{\mathbf{y}}^{r+1} - \mathbf{y}^{r+1}\right\|^2\\
&\quad + 2\left\langle(I-R)\cdot(\mathbf{y}^{r+1} - \tilde{\mathbf{y}}^r), (I-R)\cdot(\mathbb{E}_r\,\tilde{\mathbf{y}}^{r+1} - \mathbf{y}^{r+1})\right\rangle\\
&\quad + 2\left\langle(I-R)\cdot\tilde{\mathbf{y}}^r, (I-R)\cdot(\mathbb{E}_r\,\tilde{\mathbf{y}}^{r+1} - \mathbf{y}^{r+1})\right\rangle\\
&\overset{(10)}{\le} \frac{\beta_2}{2}\left\|\nabla f(\tilde{\bar{\mathbf{x}}}^r)\right\|^2 + \frac{\beta_3}{2}\left\|(I-R)\cdot\tilde{\mathbf{y}}^r\right\|^2\\
&\quad + \frac{\beta_1(2+L_f^2)}{2}\left\|\mathbf{y}^{r+1} - \tilde{\mathbf{y}}^r\right\|^2\\
&\quad + \left(1 + \frac{L_f}{2N} + \frac{1}{\beta_1} + \frac{\beta_2+\beta_3}{2\beta_2\beta_3}\right)\left\|\mathbb{E}_r\,\tilde{\mathbf{y}}^{r+1} - \mathbf{y}^{r+1}\right\|^2,
\end{aligned}$$

where in (i) we use the fact that $\mathbf{x}$ is a sub-vector of $\mathbf{y}$ and combine the two norms; add and subtract $\nabla f(\tilde{\bar{\mathbf{x}}}^r)$ to the first term, and add and subtract $(I-R)\cdot\tilde{\mathbf{y}}^r$ to the third term.

To bound the last two terms in the above relation, we have:

$$\begin{aligned}
&\left\|\mathbf{y}^{r+1} - \tilde{\mathbf{y}}^r\right\|^2 = \left\|\eta'^r_\ell\,u^r_{\ell,y} + \eta'^r_g\,u^r_g\right\|^2\\
&\le 2(\eta'^r_\ell)^2\cdot(C_x^2 + C_v^2)\cdot\left(\left\|\nabla f(\tilde{\bar{\mathbf{x}}}^r)\right\|^2 + L_f^2\left\|(I-R)\cdot\tilde{\mathbf{x}}^r\right\|^2\right)\\
&\quad + 2(\eta'^r_g)^2\left\|(I-R)\cdot\tilde{\mathbf{y}}^r\right\|^2,
\end{aligned}$$

where we apply (13) to bound $u^r_{\ell,y}$ and PD1 to bound u^r_g. And we have

$$\begin{aligned}
&\left\|\mathbb{E}_r\,\tilde{\mathbf{y}}^{r+1} - \mathbf{y}^{r+1}\right\|^2\\
&= \left\|\eta'^r_\ell(\mathbb{E}_r\,\tilde{u}^r_{\ell,y} - u^r_{\ell,y}) + \eta'^r_g(\mathbb{E}_r\,\tilde{u}^r_g - u^r_g)\right\|^2\\
&\overset{(PS1)}{=} (\eta'^r_\ell)^2\cdot\left(\left\|\mathbb{E}_r\,\tilde{u}^r_{\ell,y}\right\|^2 + \left\|u^r_{\ell,y}\right\|^2 - 2\left\langle\mathbb{E}_r\,\tilde{u}^r_{\ell,y}, u^r_{\ell,y}\right\rangle^2\right)\\
&\overset{(PS1)}{\le} (\eta'^r_\ell)^2\cdot\left(C_1 + (1-2C_2)\cdot\left\|u^r_{\ell,y}\right\|^2 + 2\sigma_G^2\right)\\
&\overset{(13)}{\le} 2(\eta'^r_\ell)^2\cdot(C_x^2 + C_v^2)\cdot(1-2C_2)\cdot\left\|\nabla f(\tilde{\bar{\mathbf{x}}}^r)\right\|^2\\
&\quad + 2L_f^2\cdot(\eta'^r_\ell)^2\cdot(C_x^2 + C_v^2)\cdot(1-2C_2)\cdot\left\|(I-R)\cdot\tilde{\mathbf{x}}^r\right\|^2\\
&\quad + (\eta'^r_\ell)^2\cdot(C_1 + 2\sigma_G^2).
\end{aligned}$$

substitute the above results to (14), we have:

$$\begin{aligned}
&\mathbb{E}_r\left[\tilde{\mathcal{E}}^{r+1}\right] - \mathcal{E}^{r+1}\\
&\le C'_{11}\left\|\nabla f(\tilde{\bar{\mathbf{x}}}^r)\right\|^2 + C'_{12}\left\|(I-R)\cdot\tilde{\mathbf{y}}^r\right\|^2\\
&\quad + (1+\frac{L_f}{2N})\cdot(\eta'^r_g)^2\cdot\sigma_g^2 + (1+\frac{L_f}{2N})\cdot(\eta'^r_\ell)^2\cdot\sigma_\ell^2\\
&\quad + (B_\ell(1+\frac{L_f}{2N}) + C_{17})\cdot(\eta'^r_\ell)^2\cdot C_1 + (\eta'^r_\ell)^2\cdot C_{17}\sigma_G^2,
\end{aligned}$$

where we define the following constants

$$\begin{aligned}
C'_{11} &:= \frac{\beta_2}{2} + \beta_1\cdot(2+L_f)\cdot C_{18} + 2C_{17}C_{18}\cdot(1-2C_2),\\
C'_{12} &:= (1+\frac{L_f}{2N})\cdot(\eta'^r_g)^2 B_g + 2L_f^2 C_{17}C_{18}\cdot(1-2C_2)\\
&\qquad + \frac{\beta_2}{2} + \beta_1\cdot(2+L_f)\cdot(C_{18}\cdot L_f^2 + (\eta'^r_g)^2),\\
C_{15} &:= C_{14}B_\ell + \sum_{r=0}^{t} C_{17}\cdot(\eta'^r_\ell)^2,\\
C_{16} &:= \sum_{r=0}^{t} C_{17}\cdot(\eta'^r_\ell)^2,\\
C_{17} &:= 1 + \frac{L_f}{2N} + \frac{1}{\beta_1} + \frac{\beta_2+\beta_3}{2\beta_2\beta_3},
\end{aligned}$$

$C_{18} := (\eta_\ell'^r)^2 \cdot (C_x^2 + C_v^2)$.

Plug it into (12), and apply PD4, then Lemma 1(B) is proved.

B.2. Case II and III

Case II: For Case II, $\tau_g = Q\tau_\ell > 0$. Let us denote the states at r^{th} global sampling time instance as $(\cdot)^r := (\cdot)(r\tau_g)$, where the q^{th} local sampling time between two consecutive global sampling instance as $(\cdot)^{r,q} := (\cdot)(r\tau_g + q\tau_\ell)$, then the system can be written as:

$$\begin{aligned}
\mathbf{x}^{r,q+1} &= \mathbf{x}^{r,q} - \eta_\ell'^{r,q} \cdot \tilde{u}_{\ell,x}^{r,q} - \eta_g'^{r,q} \cdot \tilde{u}_{g,x}^{r} \\
\mathbf{v}^{r,q+1} &= \mathbf{v}^{r,q} - \eta_\ell'^{r,q} \cdot \tilde{u}_{\ell,v}^{r,q} - \eta_g'^{r,q} \cdot \tilde{u}_{g,v}^{r} \\
\mathbf{z}^{r,q+1} &= \mathbf{z}^{r,q} - \eta_\ell'^{r,q} \cdot \tilde{u}_{\ell,z}^{r,q},
\end{aligned} \tag{15}$$

where $\eta_\ell'^{r,q} = \tau_\ell \eta_\ell^{r,q}$, $\eta_g'^{r,q} = \tau_\ell \eta_g^{r,q}$. Note that we have $(\cdot)^{r,Q} = (\cdot)^{r+1,0}$. In this case, we have the following result for the stochastic system:

Lemma 2 *Suppose the deterministic system satisfies PD1 - PD4, with stochastic controllers satisfy PS1(A) - PS3, and consider the discretization Case II with $\tau_g = Q\tau_\ell > 0$. Then we have the following:*

$$\begin{aligned}
\mathbb{E}[\tilde{\mathcal{E}}^t] - \mathcal{E}^0 \leq & -\sum_{r=0}^{t-1} (\gamma_1^r - C_{21}^r) \cdot \mathbb{E}[\left\|\nabla f(\tilde{\mathbf{x}}^r)\right\|^2] \\
& - \sum_{r=0}^{t-1} (\gamma_2^r - C_{22}^r) \cdot \mathbb{E}[\|(I-R)\cdot \tilde{\mathbf{y}}^r\|^2] \\
& + C_{23}(t)\sigma_g^2 + C_{24}(t)\sigma_\ell^2,
\end{aligned} \tag{16}$$

where $\{C_{2i}\}_{i=1}^4$ are some coefficients related to $L, L_f, C_x, C_v, B_\ell, B_g, \eta_\ell'^{r,q}, \eta_\ell'^{r,q}$, and Q.

This result is similar to Lemma 1(A). The proof of this lemma is given in Sec. B.2.1.

Also, a similar result with the LCFL satisfies PS1(B) can be proved following similar steps as Lemma 1(B) and Lemma 2. We omitted these derivations to avoid repetition.

Case III: For Case III, $\tau_\ell = K\tau_g > 0$. Let us denote the states at r^{th} local sampling time instance as $(\cdot)^r := (\cdot)(r\tau_\ell)$, where the k^{th} global sampling time between two consecutive local sampling time instances as $(\cdot)^{r,k} := (\cdot)(r\tau_\ell + k\tau_g)$, then the system can be written as:

$$\begin{aligned}
\tilde{\mathbf{x}}^{r,k+1} &= \tilde{\mathbf{x}}^{r,k} - \eta_\ell'^{r,k} \cdot \tilde{u}_{\ell,x}^{r} - \eta_g'^{r,k} \cdot \tilde{u}_{g,x}^{r,k} \\
\tilde{\mathbf{v}}^{r,k+1} &= \tilde{\mathbf{v}}^{r,k} - \eta_\ell'^{r,k} \cdot \tilde{u}_{\ell,v}^{r} - \eta_g'^{r,k} \cdot \tilde{u}_{g,v}^{r,k} \\
\tilde{\mathbf{z}}^{r,k+1} &= \tilde{\mathbf{z}}^{r,k} - \eta_\ell'^{r,k} \cdot \tilde{u}_{\ell,z}^{r},
\end{aligned} \tag{17}$$

where $\eta_\ell'^{r,k} = \tau_g \eta_\ell^{r,k}$, $\eta_g'^{r} = \tau_g \eta_g^{r,k}$. Note that $(\cdot)^{r,K} = (\cdot)^{r+1,0}$.

A similar result to Case I can be shown for Case III:

Lemma 3 *Suppose the deterministic system satisfies PD1 - PD4, with stochastic controllers satisfy PS1(A) - PS3, and consider the discretization Case III with $\eta_\ell = K\eta_k > 0$. Then we have the following:*

$$\begin{aligned}
\mathbb{E}[\tilde{\mathcal{E}}^t] - \mathcal{E}^0 \leq & -\sum_{r=0}^{t-1} (\gamma_1^r - C_{31}^r) \cdot \mathbb{E}[\left\|\nabla f(\tilde{\mathbf{x}}^r)\right\|^2] \\
& - \sum_{r=0}^{t-1} (\gamma_2^r - C_{32}^r) \cdot \mathbb{E}[\|(I-R)\cdot \tilde{\mathbf{y}}^r\|^2] \\
& + C_{33}(t)\sigma_g^2 + C_{34}(t)\sigma_\ell^2,
\end{aligned} \tag{18}$$

where $\{C_{3i}\}_{i=1}^4$ are some coefficients related to $L, L_f, C_x, C_v, B_\ell, B_g, \eta_\ell'^{r,q}, \eta_\ell'^{r,q}$, and K.

The proof follows the similar steps as in Case I and Case II so we omit it due to page limitation.

B.2.1. Proof of Lemma 2

The proof follows the similar steps as in Case I. We first break down the difference between the energy functions of the consecutive communications as:

$$\begin{aligned}
\mathbb{E}_{r,0}\left[\tilde{\mathcal{E}}^{r+1,0} - \tilde{\mathcal{E}}^{r,0}\right] &= \underbrace{\mathbb{E}_{r,0}\left[\tilde{\mathcal{E}}^{r+1,0}\right] - \mathcal{E}^{r+1,0}}_{\Delta^{r+1}} \\
&+ \underbrace{\mathcal{E}^{r+1,0} - \tilde{\mathcal{E}}^{r,0}}_{\text{term I}},
\end{aligned} \tag{19}$$

Then the key is to bound Δ^{r+1}. We proceed by the following:

$$\begin{aligned}
\Delta^{r+1} &= \mathbb{E}_{r,0}\left[f(\tilde{\mathbf{x}}^{r+1,0}) - f(\bar{\mathbf{x}}^{r+1,0})\right] \\
&\quad + \mathbb{E}_{r,0}\left[\left\|(I-R)\cdot\tilde{\mathbf{y}}^{r+1,0}\right\|^2 - \left\|(I-R)\cdot\mathbf{y}^{r+1,0}\right\|^2\right] \\
&\overset{(A2)}{\leq} (1 + \frac{L_f}{2N}) \cdot \mathbb{E}_{r,0}\left\|\mathbf{y}^{r+1,0} - \tilde{\mathbf{y}}^{r+1,0}\right\|^2 \\
&\quad + \mathbb{E}_{r,0}\left[\langle \nabla f(\bar{\mathbf{x}}^{r+1,0}), \tilde{\mathbf{x}}^{r+1,0} - \bar{\mathbf{x}}^{r+1,0}\rangle\right] \\
&= (1 + \frac{L_f}{2N}) \cdot \mathbb{E}_{r,0}\left\|\mathbf{y}^{r+1,0} - \tilde{\mathbf{y}}^{r+1,0}\right\|^2 \\
&\quad + \mathbb{E}_{r,0}\left[\langle \nabla f(\bar{\mathbf{x}}^{r+1,0}) - \nabla f(\tilde{\mathbf{x}}^{r,0}), \tilde{\mathbf{x}}^{r+1,0} - \bar{\mathbf{x}}^{r+1,0}\rangle\right] \\
&\quad + \mathbb{E}_{r,0}\left[\langle \nabla f(\tilde{\mathbf{x}}^{r,0}), \tilde{\mathbf{x}}^{r+1,0} - \bar{\mathbf{x}}^{r+1,0}\rangle\right] \\
&\overset{(10)}{\leq} (1 + \frac{L_f}{2N}) \cdot \mathbb{E}_{r,0}\left\|\mathbf{y}^{r+1,0} - \tilde{\mathbf{y}}^{r+1,0}\right\|^2 \\
&\quad + \frac{\beta_1}{2}\left\|\nabla f(\bar{\mathbf{x}}^{r+1,0}) - \nabla f(\tilde{\mathbf{x}}^{r,0})\right\|^2 + \frac{\beta_2}{2}\left\|\nabla f(\tilde{\mathbf{x}}^{r,0})\right\|^2 \\
&\quad + \frac{\beta_1 + \beta_2}{2\beta_1\beta_2}\mathbb{E}_{r,0}\left\|\tilde{\mathbf{x}}^{r+1,0} - \bar{\mathbf{x}}^{r+1,0}\right\|^2.
\end{aligned}$$

We need to bound each term separately. Notice that we have

$$\left\|\tilde{\mathbf{x}}^{r,q} - \bar{\mathbf{x}}^{r,q}\right\|^2 \leq \frac{1}{N}\left\|\tilde{\mathbf{x}}^{r,q} - \mathbf{x}^{r,q}\right\|^2,$$

$$\left\|\tilde{\mathbf{x}}^{r,q} - \mathbf{x}^{r,q}\right\|^2 \leq \left\|\tilde{\mathbf{y}}^{r,q} - \mathbf{y}^{r,q}\right\|^2 \leq \left\|\tilde{\mathbf{s}}^{r,q} - \mathbf{s}^{r,q}\right\|^2.$$

Therefore, we first bound the first term and the last term in

the above equation by:

$$
\begin{aligned}
&\mathbb{E}_{r,0}\left\|\tilde{\mathbf{s}}^{r,q}-\mathbf{s}^{r,q}\right\|^2 \qquad (20)\\
&=\mathbb{E}_{r,0}\left\|\sum_{q_1=0}^{q}\eta_\ell'^{r,q_1}(\tilde{u}_\ell^{r,q_1}-u_\ell^{r,q_1})+q\eta_g'^{r}(\tilde{u}_g^{r,0}-u_g^{r,0})\right\|^2\\
&\overset{(i)}{\le} 2q\cdot\sum_{q_1=0}^{q}(\eta_\ell'^{r,q_1})^2\cdot\mathbb{E}_{r,0}\left\|\tilde{u}_\ell^{r,q_1}-u_\ell^{r,q_1}\right\|^2\\
&\quad+2q^2\cdot(\eta_g'^{r})^2\cdot\mathbb{E}_{r,0}\left\|\tilde{u}_g^{r,0}-u_g^{r,0}\right\|^2\\
&\overset{(ii)}{\le} 4q\cdot\sum_{q_1=0}^{q}(\eta_\ell'^{r,q_1})^2\cdot\mathbb{E}_{r,0}\left\|\tilde{u}_\ell^{r,q_1}-\mathbb{E}_{r,q_1}\tilde{u}_\ell^{r,q_1}\right\|^2\\
&\quad+4q\cdot\sum_{q_1=0}^{q}(\eta_\ell'^{r,q_1})^2\cdot\mathbb{E}_{r,0}\left\|\mathbb{E}_{r,q_1}\tilde{u}_\ell^{r,q_1}-u_\ell^{r,q_1}\right\|^2\\
&\quad+2q^2\cdot(\eta_g'^{r})^2\cdot\mathrm{Var}_{r,0}(\tilde{u}_g^{r,0})\\
&\overset{(iii)}{\le} 4q\cdot\sum_{q_1=0}^{q}(\eta_\ell'^{r,q_1})^2\cdot\left(B_\ell\left\|\mathbb{E}_{r,q_1}\tilde{u}_\ell^{r,q_1}\right\|^2+\sigma_\ell^2\right)\\
&\quad+4qL^2\cdot\sum_{q_1=0}^{q}(\eta_\ell'^{r,q_1})^2\cdot\mathbb{E}_{r,0}\left\|\tilde{\mathbf{s}}^{r,q_1}-\mathbf{s}^{r,q_1}\right\|^2\\
&\quad+2q^2\cdot(\eta_g'^{r})^2\cdot\left(B_g\left\|\mathbb{E}_{r,0}\tilde{u}_g^{r,0}\right\|^2+\sigma_g^2\right),
\end{aligned}
$$

where in (i) we plug in the update (15) and apply Cauchy–Schwarz inequality; in (ii) we add and subtract $\mathbb{E}_{r,q_1}\tilde{u}_\ell^{r,q_1}$ to the first term and apply PS1 to the second term; in (iii) we apply PS2 to the first and third terms and apply PD2 to the second term. Note that same as (13), we have

$$
\begin{aligned}
&\left\|\mathbb{E}_{r,q}\tilde{u}_\ell^{r,q}\right\|^2\le C_f\left\|\nabla f(\tilde{\mathbf{x}}^{r,q})\right\|^2\\
&\le 2C_f\left(\left\|\nabla f(\tilde{\mathbf{x}}^{r,q})\right\|^2+\left\|\nabla f(\tilde{\mathbf{x}}^{r,q})-\nabla f(\tilde{\mathbf{x}}^{r,q})\right\|^2\right)\\
&\le 2C_f\left(\left\|\nabla f(\tilde{\mathbf{x}}^{r,q})\right\|^2+L_f^2\left\|\tilde{\mathbf{x}}^{r,q}-\tilde{\mathbf{x}}^{r,q}\right\|^2\right)\\
&= 2C_f\left(\left\|\nabla f(\tilde{\mathbf{x}}^{r,q})\right\|^2+L_f^2\left\|(I-R)\cdot\tilde{\mathbf{x}}^{r,q}\right\|^2\right)
\end{aligned}
\qquad (21)
$$

Applying (21) to the first term and PD1 to the last term, recursively apply (20) to the second term in (20), we obtain:

$$
\begin{aligned}
&\mathbb{E}_{r,0}\left\|\tilde{\mathbf{s}}^{r,q}-\mathbf{s}^{r,q}\right\|^2 \qquad (22)\\
&\le\sum_{q_1=0}^{q}C_{45}^{r,q_1}\left(B_\ell C_f\left\|\nabla f(\tilde{\mathbf{x}}^{r,q_1})\right\|^2+\sigma_\ell^2\right)\\
&\quad+\sum_{q_1=0}^{q}C_{45}^{r,q_1}B_\ell C_f L_f^2\left\|(I-R)\cdot\tilde{\mathbf{x}}^{r,q_1}\right\|^2\\
&\quad+\frac{2q^3\cdot(\eta_g'^{r})^2}{1-4qL^2\cdot(\eta_\ell'^{r,0})^2}\cdot\left(B_g\left\|(I-R)\cdot\mathbf{y}^{r,0}\right\|^2+\sigma_g^2\right),
\end{aligned}
$$

where we define $C_{45}^{r,q}:=\frac{4q\cdot(\eta_\ell'^{r,q})^2}{1-4qL^2\cdot(\eta_\ell'^{r,q})^2}$.

Next, we bound the second term by:

$$
\begin{aligned}
&\left\|\nabla f(\bar{\mathbf{x}}^{r+1,0})-\nabla f(\tilde{\mathbf{x}}^{r,0})\right\|^2\overset{(A2)}{\le}L_f^2\left\|\bar{\mathbf{x}}^{r+1,0}-\tilde{\mathbf{x}}^{r,0}\right\|^2\\
&\overset{(15)}{=}L_f^2\left\|\frac{\mathbb{1}_N^T}{N}\sum_{q=0}^{Q-1}\eta_\ell'^{r,q}u_\ell^{r,q}\right\|^2\\
&\overset{(i)}{\le}\frac{QL_f^2}{N}\sum_{q=0}^{Q-1}(\eta_\ell'^{r,q})^2\left\|u_\ell^{r,q}-\mathbb{E}_{r,q}\tilde{u}_\ell^{r,q}+\mathbb{E}_{r,q}\tilde{u}_\ell^{r,q}\right\|^2\\
&\overset{(ii)}{\le}\frac{2QL_f^2}{N}\sum_{q=0}^{Q-1}(\eta_\ell'^{r,q})^2\left\|\mathbb{E}_{r,q}\tilde{u}_\ell^{r,q}\right\|^2\\
&\quad+\frac{2QL_f^2}{N}\sum_{q=0}^{Q-1}(\eta_\ell'^{r,q})^2L^2\left\|\mathbf{s}^{r,q}-\tilde{\mathbf{s}}^{r,q}\right\|^2,
\end{aligned}
\qquad (23)
$$

where in (i) we apply Cauchy–Schwarz inequality; in (ii) we first apply Cauchy–Schwarz inequality and then apply PD2. Further plug (21) and (22) into (23), we have:

$$
\begin{aligned}
&\left\|\nabla f(\bar{\mathbf{x}}^{r+1,0})-\nabla f(\tilde{\mathbf{x}}^{r,0})\right\|^2\\
&\le\frac{4QC_fL_f^2}{N}\sum_{q=0}^{Q-1}(\eta_\ell'^{r,q})^2\left(\left\|\nabla f(\tilde{\mathbf{x}}^{r,q})\right\|^2+L_f^2\left\|(I-R)\cdot\tilde{\mathbf{x}}^{r,q}\right\|^2\right)\\
&\quad+\frac{2QL_f^2}{N}\sum_{q=0}^{Q-1}(\eta_\ell'^{r,q})^2\\
&\quad\times\left(L^2\sum_{q_1=0}^{q}C_{45}^{r,q_1}\left(B_\ell C_f\left\|\nabla f(\tilde{\mathbf{x}}^{r,q_1})\right\|^2+\sigma_\ell^2\right)\right.\\
&\quad+\sum_{q_1=0}^{q}C_{45}^{r,q_1}B_\ell C_fL_f^2\left\|(I-R)\cdot\tilde{\mathbf{x}}^{r,q_1}\right\|^2\\
&\quad\left.+\frac{2q^3\cdot(\eta_g'^{r})^2}{1-4qL^2\cdot(\eta_\ell'^{r,0})^2}\cdot\left(B_g\left\|(I-R)\cdot\mathbf{y}^{r,0}\right\|^2+\sigma_g^2\right)\right),
\end{aligned}
\qquad (24)
$$

Substitute (22) and (24) into Δ^{r+1} in (19), then apply PD4, Lemma 2 is proved.

C. Algorithm Design: a Case Study

In this part, we take the gradient tracking algorithm as an example to illustrate how the framework can be applied to design new algorithms for different applications. In specific, we modify the stochastic local and consensus controllers for different applications. Then we verify PS1 - PS3 for the stochastic controllers and PD1-PD4 for the deterministic system, so that we can apply Theorem 1 to obtain the final convergence result and optimize the hyperparameters. Finally we conduct additional numerical experiments to verify these convergence results.

C.1. Gradient-tracking Based Stochastic Algorithm

We start with the deterministic gradient tracking algorithm described in (4) as baseline. First, we consider adopting the stochastic gradient, which results in the Distributed Stochastic Gradient Tracking (DSGT) algorithm (Lu et al.,

2019), with the following updates:

$$\begin{aligned}\mathbf{x}^+ &= \mathbf{x} - W\mathbf{x} - \alpha\mathbf{v},\\ \mathbf{v}^+ &= \mathbf{v} - W\mathbf{v} + (\tilde{\nabla} f(\mathbf{x}) - \tilde{\nabla} f(\mathbf{z})),\\ \mathbf{z}^+ &= \mathbf{x},\end{aligned} \tag{25}$$

where the LCFL for auxiliary state $\mathbf{v}$ are replaced by the difference of stochastic gradients $\tilde{\nabla} f(\cdot)$ estimated with a subset of samples.

Then, we consider the randomized communication scheme, where each communication connection between the agents has a p failure rate at each round of communication. Which result in gradient tracking on dynamic directed communication graph (D^{2}GT):

$$\begin{aligned}\mathbf{x}^+ &= \mathbf{x} - \eta_g' \tilde{W}\mathbf{x} - \alpha\mathbf{v},\\ \mathbf{v}^+ &= \mathbf{v} - \eta_g' \tilde{W}\mathbf{v} + (\nabla f(\mathbf{x}) - \nabla f(\mathbf{z})),\\ \mathbf{z}^+ &= \mathbf{x},\end{aligned} \tag{26}$$

where $\tilde{W}$ is a stochastic weight matrix satisfies

$$\tilde{W}_{ij} = \tilde{W}_{ji} := \begin{cases} W_{ij}/(1-p), & \text{w.p. } 1-p\\ 0, & \text{w.p. } p\end{cases}.$$

For the third application, we consider adopting the Gaussian mechanism (Abadi et al., 2016) to provide DP guarantee for the local data. The resulting DP-DSGT algorithm is:

$$\begin{aligned}\mathbf{x}^+ &= \mathbf{x} - \eta_g' \tilde{W}\cdot(\mathbf{x} + \mathbf{w}_x) - \alpha\cdot \mathrm{clip}(\mathbf{v}, \beta_x),\\ \mathbf{v}^+ &= \mathbf{v} - \eta_g' \tilde{W}\cdot(\mathbf{v} + \mathbf{w}_v) + \mathrm{clip}(\tilde{\nabla} f(\mathbf{x}) - \tilde{\nabla} f(\mathbf{z}), \beta_v),\\ \mathbf{z}^+ &= \mathbf{x},\end{aligned} \tag{27}$$

where $\tilde{W}$ is the same as the one in (26), $\mathbf{w}_x \sim \mathcal{N}(0, \sigma_x^2 I), \mathbf{w}_v \sim \mathcal{N}(0, \sigma_v^2 I)$ are the privacy noises, and β_x, β_v are the clipping thresholds.

C.2. Theoretical Analysis

In this part, we show how the proposed framework helps analyze the stochastic algorithms. It is easy to verify PD1-PD3. We can also verify PD4 for the deterministic algorithm with $\gamma_1^r = \mathcal{O}(\alpha^r), \gamma_2^r = \mathcal{O}(\alpha^r)$, cf. (Lu et al., 2019):

Lemma 4 ((Lu et al., 2019) Lemma 4) *With the energy function $\mathcal{E}(t)$ defined in (6), we have*

$$\mathcal{E}^{r+1} - \mathcal{E}^r \le -c_1\alpha \left\|\nabla f(\bar{\mathbf{x}}^r)\right\|^2 - c_2\alpha \left\|(I-R)\cdot \mathbf{y}^r\right\|^2,$$

where c_1 and c_2 are some constants depending on C_g, L_f, N.

For DSGT, only the LCFL has stochasticity. By assuming the stochastic gradients are unbiased and has bounded variance, i.e.,

$$\mathbb{E}\tilde{\nabla} f(\mathbf{x}) = \nabla f(\mathbf{x}),\ \mathbb{E}\left\|\tilde{\nabla} f(\mathbf{x}) - \nabla f(\mathbf{x})\right\| \le \sigma^2.$$

then PS1(A) is satisfied; PS2 is satisfied with $B_\ell = 0, \sigma_\ell = 2\sigma$; and PS3 is also satisfied. Therefore, apply Lemma 1(A), we obtain the following convergence result:

$$\begin{aligned}\mathbb{E}[\tilde{\mathcal{E}}^t] - \mathcal{E}^0 \le &-\sum_{r=0}^{t-1}\mathcal{O}(\alpha^r)\cdot\mathbb{E}[\left\|\nabla f(\tilde{\mathbf{x}}^r)\right\|^2]\\ &-\sum_{r=0}^{t-1}\mathcal{O}(\alpha^r)\cdot\mathbb{E}[\|(I-R)\cdot\tilde{\mathbf{y}}^r\|^2] + C_{14}(t)\sigma_\ell^2,\end{aligned}$$

where $C_{14} = \sum_{r=0}^{t-1}(\alpha^r)^2\cdot(1+\frac{L_f}{2N})$. Therefore, we can choose $\alpha^r = \mathcal{O}(1/\sqrt{r})$, then the algorithm converges with

$$\begin{aligned}&\mathbb{E}\left[\left\|\nabla f(\tilde{\mathbf{x}}^{r_1})\right\|^2 + \|(I-R)\cdot\tilde{\mathbf{y}}^{r_1}\|^2\right]\\ &= \mathcal{O}\left(\frac{1}{\sum_{r=0}^{t-1}\alpha^r}\right)\mathcal{E}^0 + \mathcal{O}\left(\frac{\sum_{r=0}^{t-1}(\alpha^r)^2}{\sum_{r=0}^{t-1}\alpha^r}\right)\sigma_\ell^2.\end{aligned}$$

with rate $\mathcal{O}(\log(t)/\sqrt{t})$. This recovers the convergence result in (Lu et al., 2019).

For D^{2}GT, only the GCFL has stochasticity. We can verify that PS1(A) is satisfied, PS2 is satisfied with $B_g = p/(1-p), \sigma_g = 0$, and PS3 is also satisfied. Therefore, apply Lemma 1(A), it requires $C_{12}^r = B_g\cdot(\eta_g')^2\cdot(1+\frac{L_f}{2N}) < c_2\alpha^r$. So we can choose $\alpha = \mathcal{O}(1), \eta_g' = \mathcal{O}\left(\sqrt{B_g c_2 \alpha^r\cdot(1+\frac{L_f}{2N})}\right)$, and we obtain the following convergence result:

$$\mathbb{E}\left[\left\|\nabla f(\tilde{\mathbf{x}}^{r_1})\right\|^2 + \|(I-R)\cdot\tilde{\mathbf{y}}^{r_1}\|^2\right] = \mathcal{O}\left(\frac{1}{\sum_{r=0}^{t-1}\alpha^r}\right)\mathcal{E}^0.$$

with rate $\mathcal{O}(1/t)$.

For DP-DSGT, both controllers have stochasticities. We can verify that PS1(B) is satisfied, with $C_1 = 2(\beta_x + \beta_v)$. For C_2, σ_G can be derived with similar technique in (Chen et al., 2020). For PS2, we can verify that $B_\ell = 0, \sigma_\ell = 2\sigma$ and $B_g = p/(1-p), \sigma_g = \sigma_x + \sigma_v$. If we assume $\beta_x \ge \|\mathbf{v}\|^2$ and $\beta_v \ge \left\|\tilde{\nabla} f(\mathbf{x}) - \tilde{\nabla} f(\mathbf{z})\right\|^2$ for all $t \in [0,\infty)$, then PS1(A) is satisfied. Applying Lemma 1(B), we obtain:

$$\begin{aligned}\mathbb{E}[\tilde{\mathcal{E}}^t] - \mathcal{E}^0 \le &-\sum_{r=0}^{t-1}(\gamma_1^r - C_{11}')\cdot\mathbb{E}[\left\|\nabla f(\tilde{\mathbf{x}}^r)\right\|^2]\\ &-\sum_{r=0}^{t-1}(\gamma_2^r - C_{12}')\cdot\mathbb{E}[\|(I-R)\cdot\tilde{\mathbf{y}}^r\|^2]\\ &+ C_{13}(t)\sigma_g^2 + C_{14}(t)\sigma_\ell^2 + C_{15}(t)C_1 + C_{16}(t)\sigma_G^2.\end{aligned}$$

where

$$C_{11}' = \mathcal{O}((\alpha^r)^2),\quad C_{12}' = \mathcal{O}((\alpha^r)^2 + (\eta_g'^r)^2),$$

$$\{C_{1i}\}_{i=3}^6 = \mathcal{O}(\sum_{r=0}^{t-1}(\alpha^r)^2),\ \sigma_x^2 = \Omega\left(\frac{C_1 d_x^2 t\cdot(1-p)}{N\epsilon^2}\right),$$

$$\sigma_v^2 = \Omega\left(\frac{C_1 d_v^2 t\cdot(1-p)}{N\epsilon^2}\right),$$

σ_x, σ_v are chosen for the algorithm to provide (ϵ,δ)-DP guarantee, cf. (Abadi et al., 2016)[Definition 1, Theorem 1]:

Definition 2 ((ϵ, δ)**-DP)** *An algorithm $\mathcal{M}$ is (ϵ, δ)-DP if*

$$P(\mathcal{M}(\mathcal{D}) \in \mathcal{S}) \leq e^{\epsilon} P(\mathcal{M}(\mathcal{D}') \in \mathcal{S}) + \delta, \qquad (28)$$

where $\mathcal{D}$ and $\mathcal{D}'$ are neighboring datasets, $\mathcal{S}$ is an arbitrary subset of outputs of $\mathcal{M}$.

Theorem 2 (Privacy of DP-DSGT) *There exist constants u and v so that given the number of iterations t, for any $\epsilon \leq u(1-p)^2 t$ with p as communication dropout rate, Algorithm DP-DSGT is (ϵ, δ)-differentially private for any $\delta > 0$ if $\sigma^2 \geq v \frac{C_1^2 (1-p) T \ln(\frac{1}{\delta})}{N\epsilon^2}$.*

Optimizing $p, \alpha, t, \beta_x, \beta_v, \sigma_x, \sigma_v$, we obtain the final convergence rate $\mathcal{O}(\frac{\sqrt{d_x + d_v}}{N\epsilon})$.

C.3. Numerical Results

In this subsection, we provide numerical results for implementations of the three algorithms discussed in the previous subsection. We verify the convergence speed derived from the previous subsection for each algorithm.

In the experiments, we consider optimizing the non-convex regularized logistic regression problem:

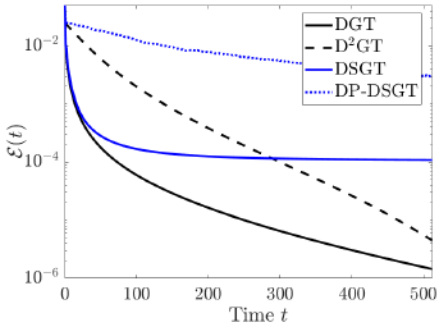
$$f_i(\mathbf{x}; (\mathbf{a}_i, b_i)) = \log(1 + \exp(-b_i \mathbf{x}^T \mathbf{a}_i)) + \sum_{d=1}^{d_x} \frac{\beta \alpha (\mathbf{x}[d])^2}{1 + \alpha(\mathbf{x}[d])^2},$$

where $\mathbf{a}_i$ denotes the features and b_i denotes the labels of the dataset on the i^{th} agent. We set the number of agent $N = 200$ and each agent has local dataset of size 1000. We use an Erdős–Rényi random graph with density 0.5 for the network and the weight matrix is select as $W := 0.9A^T A / \max\{A^T A\}$

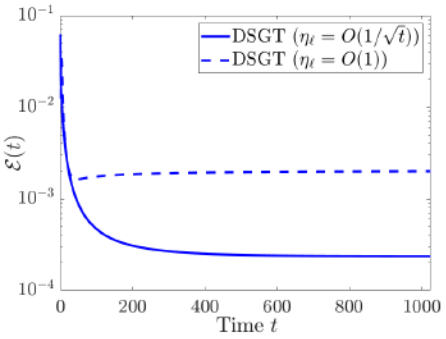
For DSGT and DP-DSGT, we use batch size 10 to estimate the stochastic gradients; for D^{2}GT, and DP-DSGT, we choose the communication dropout rate $p = 0.9$. The clipping threshold β_x, β_v are set as the average of local controller's magnitude of the DSGT algorithm and σ_x, σ_v are chosen following (McMahan et al., 2018) with $(\epsilon, \delta) = (4, 10^{-5})$ at $t = 128$.

The result is shown in Figure 4a. It can be observed that D^{2}GT has the same convergence rate as DGT with a constant slow down, while DSGT and DP-DSGT have slower convergence rates. These results match with the theoretical results in the previous subsection.

In addition, we provided another example demonstrating the necessity of the $\mathcal{O}(1/\sqrt{t})$ rate for DSGT. We run the DSGT algorithm with batch size 2 to estimate the stochastic gradients. In one setting we choose $\alpha = \mathcal{O}(1)$ and $\alpha = \mathcal{O}(1/\sqrt{t})$ in the other setting. The result is shown in Figure 4b. We can see that with improperly chosen constant stepsize, DSGT will not converge.



(a) The convergence of the Energy function $\mathcal{E}(t)$ of DGT, D^{2}GT, DSGT and DP-DSGT.



(b) Energy function $\mathcal{E}(t)$ of DSGT with different decreasing and constant stepsizes $\eta'_\ell(t)$.

Figure 4: The performance of DGT, D^{2}GT, DSGT and DP-DSGT.