Optimization over time-varying networks with unbounded delays

Arunselvan Ramaswamy
Adrian Redder
Danie E. Quevedo

December 19, 2019

Abstract
Solving optimization problems in multi-agent systems (MAS) involves information exchange between agents. These solutions must be robust to delays and errors that arise from an unreliable wireless network which typically connects the MAS. In today’s large-scale dynamic Internet Of Things style multi-agent scenarios, the network topology changes and evolves over time. In this paper, we present a simple distributed gradient based optimization framework and an associated algorithm. Convergence to a minimum of a given objective is shown under mild conditions on the network topology and objective. Specifically, we only assume that a message sent by a sender reaches the intended receiver, possibly delayed, with some positive probability. To the best of our knowledge ours is the first analysis under such weak general network conditions. We also discuss in detail the verifiability of all assumptions involved. This paper makes a significant technical contribution in terms of the allowed class of objective functions. Specifically, we present an analysis wherein the objective function is such that its sample-gradient is merely locally Lipschitz continuous. The theory developed herein is supported by numerical results. Another contribution of this paper is a consensus algorithm based on the main framework/algorithm. The long-term behavior of this consensus algorithm is a direct consequence of the theory presented. Again, we believe that ours is the first consensus algorithm to account for unbounded stochastic communication delays, in addition to time-varying networks.

1 INTRODUCTION
Systems consisting of multiple autonomous agents that interact with each other to solve local and global problems are called Multi-agent Systems (MAS). The
agents involved may operate on different spatial and temporal scales, although in a cooperative manner. Examples include, smart electricity grids and buildings, vehicular and mobile networks, and the Internet of Things (IoT). Typically, MAS are large-scale in nature and spread over large geographical areas. It is therefore convenient and cost-effective to facilitate agent interactions through the use of wireless communication networks. Although cost effective and easy to set-up, wireless networks are prone to delays and errors.

The agents interact and cooperate with each other to solve a system-level problem, while performing local computations and taking local decisions. System-level problems such as multi-agent learning, distributed networked control, consensus, etc., may be solved by viewing them as distributed optimization problems. Here, the classical optimization paradigm is not applicable. This is because it assumes the presence of a central entity that has access to all available problem data. In the scenario considered herein, a central entity may or may not exist. When present, it does not have access to all problem data. It may however request and obtain data from other agents in the system. As stated earlier, all communications are over a lossy delayed wireless network, whose topology and quality of service (QoS) may be time-varying. The reader is referred to [12] for a survey of distributed optimization algorithms for networked control problems.

In this paper, we are interested in solving the following optimization problem in a decentralized manner:

$$x^* = (x^*_1, \ldots, x^*_D) = \arg\min_{x \in \mathbb{R}^d} F(x), \quad (1)$$

where $D$ is the number of agents in the MAS, $F$ is a given stochastic global objective function, $x^*_i$ is the local optimal decision variable of agent-$i$ for $1 \leq i \leq D$, and $d$ is the sum of the system dimensions of the $D$ agents. Starting from an initial random estimate $x^0_i$, agent-$i$ iteratively refines its estimates such that it converges to $x^*_i$, together with other agents simultaneously. Put together, these local optima minimize the global objective.

Broadly speaking, the distributed delayed gradient-based optimization procedure presented here entails the following. At time $n$, agent-$i$ uses local estimates $x_{m(j)}^j$, $j \neq i$, $0 \leq m(j) \leq n$, obtained from other agents via the wireless network. Ideally, agent-$i$ would like to have $x^n_j$, $j \neq i$. However, due to network delays and errors, it may only have access to older estimates, hence we have the condition that $0 \leq m(j) \leq n$. Due to the large system size, every pair of agents may not be connected by a dedicated wireless channel. Instead, information is relayed through other agents in the system, leading to further delays. Put succinctly, we show that the optimization problem can be solved through distributed delayed gradient descent, provided there is a positive probability of successful information exchange between every pair of agents. This probability may vary over time, thereby allowing for time-varying network topologies.

The literature on distributed optimization is too long to list. We only mention a few recent ones and compare them to the results found herein. An accelerated algorithm to tackle distributed optimization problems wherein local functions themselves are a sum, is studied in [9]. In [21], a Kiefer–Wolfowitz style distributed descent algorithm is developed to solve non-convex and non-smooth optimization problems in a distributed manner. Although [21] considers
time-varying network topologies, the assumptions on network quality and topology are strict. Specifically, they assume that (a) the associated network graph is doubly stochastic at every time-step, (b) the probability of successful transmissions across any link is lower bounded by a fixed constant that does not vary over time, and (c) every pair of agents must be periodically connected by a direct link/channel. On the other hand, in this paper, we merely assume that there is a positive probability associated with inter-agent information exchange, that may vary over time. While we do assume that the objective function is differentiable, we may drop this assumption and use Kiefer-Wolfowitz style finite difference terms, similar to [21], or SPSA-C terms similar to [16], instead. The reader is referred to Remark 2 for a short discussion on this. In another related work, [23], the network topology switching process is assumed to have a unique stationary distribution. Although we do not explicitly consider network switching, our assumptions imply that there is no requirement for the aforementioned unique stationary distribution. There may be multiple limiting distributions or the switching process may even oscillate.

Another problem considered in this paper is that of consensus in MAS. Here, the agents seek to find a common control objective starting from a random one, through interactions and local computations. The cumulative consensus problem is given by:

$$x^* = \arg\min_{x \in \mathbb{R}^d} \sum_{i=1}^{D} f_i(x),$$

where $f_i$ is a local objective of agent-i that is not accessible to others. This problem is studied in [19] under the added complexity that only delayed local gradients are available. Further, these delays may be stochastic and unbounded. Subgradient based consensus algorithms are studied in [13], constraints and time-delays are tackled in [11], and rate of convergence is studied in [14]. The theory developed herein, to tackle distributed optimization, is utilized to develop a simple distributed gradient-based consensus algorithm. It accommodates unbounded stochastic delays, even with respect to information exchanged between agents. This latter type of delay is not considered in [19]. Again, we show that consensus can be achieved in the presence of time-varying network topologies.

### 1.1 Our Contributions

We have developed simple sufficient conditions for convergence of distributed (approximate) gradient-based algorithms, used to solve large-scale distributed stochastic optimization problems. For this, we have used tools from Stochastic Approximation Algorithms [3, 6] and Viability Theory [1]. The theory accounts for stochastic unbounded delays in the information exchanged between agents, that usually arises from the underlying wireless network. The analysis can be trivially extended to accommodate for computational delays in obtaining local gradients. This theory naturally yields an algorithm that uses distributed delayed gradients. In Remark 2 we discuss how this algorithm can be extended to use Kiefer-Wolfowitz style gradient approximations instead of exact gradients.

Convergence is shown under very mild conditions on communication. Specifically, we assume that there is a positive probability of successful, possibly delayed, information exchange between agents. We also assume that the transmissions across different channels are independent. Unlike assumptions that are
popular in literature, these conditions naturally account for a time-varying network topology. They also accommodate multi-hop networks that may connect agents. In cases where the topology change is governed by a switching process, our algorithm is guaranteed to converge even when this process has multiple stationarities. Finally, the theory is used to develop an easy to implement consensus algorithm.

In the next section, we set up the problem of interest and state it formally. We then present the general form of an iteration, that uses delayed gradient information, to solve a given stochastic distributed optimization problem. This iteration will later form the basis for the development of the previously mentioned algorithms.

2 Distributed optimization over time-varying networks

We consider a $D$-agent system denoted by $V := \{1, \ldots, D\}$. The collective goal of this $D$-agent system is to minimize a given global stochastic objective function $F$, in a cooperative manner. As stated before, the minimizer $x^*$ of $F$, is composed of $x^*_i$, $1 \leq i \leq D$, where $x^*_i$ is interpreted as a local optimal decision variable with respect to agent-$i$. The decision space of agent-$i$ is given by $\mathbb{R}^{d_i}$, such that $\sum_{i=1}^{D} d_i = d$. Each agent starts with a completely random estimate $x^0_i \in \mathbb{R}^{d_i}$, of $x^*_i$, and refines this iteratively through gradient-based searches in its decision space. Specifically, the descent direction is given by the partial derivative taken with respect to the $x_i$-component, i.e., $\nabla x_i F$. To calculate this partial derivative, at every stage, it requires estimate information from other agents. This information is obtained using a wireless network that is prone to losses and delays. In other words, at time $n$, agent-$i$ refines its estimate using the following gradient descent step:

$$x^{n+1}_i = x^n_i - a(n) \nabla x_i F(x^{m(1)}_1, \ldots, x^{m(D)}_D), \quad (2)$$

(i) $F : \mathbb{R}^d \rightarrow \mathbb{R}$ is the given stochastic cost function,
(ii) $x^{m(j)}_j$ is the $m(j)$th estimate of agent-$j$, $1 \leq j \leq D$, with $0 \leq m(j) \leq n$.
(iii) $\nabla x_i (\cdot)$ denotes the partial derivative with respect to vector $x_i$.
(iv) $\{a(n)\}_{n \geq 0}$ is the given step-size sequence.

In (2) the delay $m(j)$ is anywhere between 0 and $n$ ($0 \leq m(j) \leq n$) to account for information delays. Due to the large system size, each agent-pair may not be connected by a dedicated wireless channel. Instead, information is exchanged by forwarding it through other agents along a path connecting them, which causes further delays.

Graph view of MAS. A MAS that is connected by a wireless network can be represented by a graph. There is one node for every agent in the MAS. Two nodes are connected by an edge if and only if the corresponding agents can communicate directly. Communication between a pair of agents is considered “direct” when there is no involvement from other agents. In real-world scenarios this may be realized when a pair is connected by a dedicated wireless channel, or a multi-hop network that does not contain other agents. Further, the channels may be unidirectional which are represented using directional edges. Hence, in
the most general case, a MAS is represented by a digraph (directed graph) \[22\]. The channels themselves, in general, may be modeled as hidden Markov fading channels \[20\]. It may be noted that a wide range of communication models are accommodated by the framework presented herein. Further, we allow for time-varying network topologies and QoS, to account for real-world scenarios wherein physical wear-and-tear, introduction of new resources, etc., happen routinely.

In this paper, we are interested in the development of an analytical framework, and an associated algorithm, to solve the distributed optimization problem in MAS with unreliable communication, time-varying network topology and QoS. For this, among others, we develop simple verifiable practical assumptions on the wireless network, that ensure convergence of distributed delayed gradient based optimization algorithms.

**Stochastic Global Objective.** In \[2\], the form of the stochastic objective function \(F\) is given by:

\[
F(x) := E_\xi[f(x, \xi)],
\]

where \(f : \mathbb{R}^d \times S \to \mathbb{R}\) is a real-valued function and \(\xi\) is an \(S\) valued random variable. Typically, \(\xi\) is the randomness that arises due to environmental and network-related fluctuations. We shall broadly call \(\xi\) as the network noise. In this paper, we assume that all agents may only observe noisy samples of the stochastic objective, at any point in time. Note that stochasticity in the objective function arises due to the random variable \(\xi\). Further, when two agents observe the function at the same time, the sample obtained is influenced by the same realization of \(\xi\). In words, say that agent-i wishes to evaluate \(\nabla x_i F\) at \(\tilde{x}_i\). At the same time agent-j, \(j \neq i\), wishes to evaluate \(\nabla x_j F\) at \(\tilde{x}_j\). Further, say that the sample realization of the network noise is \(\xi_0\). Then, the agent-i partial-derivative is \(\nabla x_i F(\tilde{x}_i, \xi_0)\), while the agent-j one is \(\nabla x_j F(\tilde{x}_j, \xi_0)\). It must however be noted that the analysis presented herein can be readily extended to accommodate local fluctuations that may vary with the observing agent. The reader is referred to Remark 3 for a brief discussion on this.

**Autonomy and asynchronicity.** An important trait of MAS is the autonomy with which the agents involved operate. Agent-i performs the above partial derivative calculations at every time-step to search for a local optimal in its decision space. For this, it requires estimates from other agents that may be delayed or lost on account of the unreliable communication network. Another important aspect of a MAS is that the agents are asynchronous, i.e., the agents may not synchronize their clocks. The implication of this is that the time-step, previously mentioned, is with respect to the local clock of agent-i. However, convergence of any optimization algorithm operating in this setting requires some causal assumptions on the relative update frequency of the agents. For clarity, we carry out our analysis with respect to a hypothetical global clock, which is faster than all agent clocks and the network clock. In particular, for every tick of the hypothetical clock the agent/network clocks tick at most once. Although delay was captured in \(2\), it does not account for asynchronicity. We therefore rewrite \(2\) to accommodate for asynchronicity and to change notations for convenience.
\[ x_{i}^{n+1} = x_{i}^{n} - a(\nu(n,i))I(i \in Y^{n}) \bigg( \nabla_{x_{i}} f(x_{1}^{n-\tau_{1i}(n)}, \ldots, x_{D}^{n-\tau_{Di}(n)}, \xi^{n}) + \epsilon_{i}^{n} \bigg), \] (3)

where
(i) \( \xi^{n} \)'s are i.i.d. \( S \)-valued samples, due to the stochastic communication network.
(ii) \( Y^{n} \subset \{1, 2, \ldots, D\} \) represents the subset of agents that update their estimates between times \( n-1 \) and \( n \). Note that \( n \) is the time index with respect to the hypothetical global clock and \( Y^{n} \) accounts for asynchronicity by tracking agent updates. Say that the agent-i clock did not tick between \( n-1 \) and \( n \) (no update of the local estimate), then \( i \notin Y^{n}, I(i \in Y^{n}) = 0 \) and \( x_{i}^{n+1} = x_{i}^{n} \).
(iii) \( 0 \leq \tau_{ji}(n) \leq n \) is the (stochastic) delay experienced by agent-i in receiving the agent-j estimate, at time \( n \).
(iv) \( \nu(n,i) = \sum_{m=0}^{n} I(i \in Y^{n}) \) is the number of times that agent-i updated its local estimate until time \( n \). For example, if agent-2 has updated its estimate 3 times when the global clock has ticked 10 times, then \( \nu(10,2) = 3 \).
(v) \( \epsilon_{i}^{n} \) is a stochastic additive error term that arises due to sampling and/or gradient estimations.

The reader may recognize (3) as a distributed asynchronous sample-based noisy gradient descent scheme. The global clock indexed by \( n \) is hypothetical and only used for analysis. In the following section, we present a quick convergence analysis of (3). Specifically, we present conditions on \( \tau_{ji}(n) \), for \( 1 \leq i,j \leq D \) and \( n \geq 0 \), such that (3) converges a minimum of \( F \). Clearly, the delay random variables \( \tau_{ji}(n) \) depend on the characteristics of the underlying communication network. Additionally, it depends on the “communication protocol” used (who communicates and when).

Till now, we have not focused on the connecting wireless network, specifically on the graph associated with the MAS. In Section 3, we use the theory developed in Section 2 to present sufficient conditions on the MAS-graph/wireless network, to ensure convergence of (3). This is one of the major contributions of this paper. Hitherto requirements, in literature, on the MAS-graph are too strong and often impractical, see Section 1. We essentially show convergence under the mild assumption that, at any time, the wireless network must connect any two agents with positive probability, possibly with delays.

**Communication Protocol (CP).** In this paper, the set of rules governing the exchange of information, in a MAS, using the underlying wireless network is called the communication protocol. For example, the CP presented here is the following. At every tick of the network clock, agent-i forwards a list containing the latest available estimates from all agents to its neighbors in the MAS-graph. Put succinctly, it forwards \( \hat{x}_{ji} \), \( 1 \leq j \leq D \), where \( \hat{x}_{ji} \) is the agent-j estimate available with agent-i, that is possibly old, see Section 4 for details. In addition to network quality and topology, communication delays are also affected by the CP used. In general, the CP must be topology sensitive, since leveraging topological knowledge can greatly reduce delays and hasten the rate of convergence, of distributed optimization algorithms. In this paper, however, we present a topology agnostic CP, merely for the sake of simplicity.
and generality. In Section 4, we only present assumptions on the network and all requirements on the CP used, are intrinsic.

In the following section, we present the sufficient conditions for convergence, analyze the behavior of (3), under them. In Section 4, we present a distributed stochastic optimization algorithm and an associated communication protocol. In Section 5, we present experimental results for the algorithm presented. Finally, in Section 6, we present a consensus algorithm motivated by the theory developed in Sections 3, complete with relevant empirical evidence.

3 Analysis of (3)

First, we rewrite (3) in the following form to make it more conducive for analysis:

\[ x_{i}^{n+1} = x_{i}^{n} - a(n, i) f(i \in Y^{n}) \]

\[ = \left( E \left[ \nabla_{x} f(x_{1}^{n-\tau_{i}(n)}, \ldots, x_{D}^{n-\tau_{D}(n)}, \xi^{n}) | \mathcal{F}^{n} \right] + \left( \epsilon_{i}^{n} + M_{n+1}^{i} \right) \right), \quad (4) \]

where

(a) \( F_{0} := \sigma(x^{0}, \epsilon^{0}, Y^{0}) \) and \( F_{n} := \sigma(x^{m}, \epsilon^{m}, Y^{m}, \xi^{k} | m \leq n, k < n) \) for \( n \geq 1 \).

(b) \( M_{n+1}^{i} := \nabla_{x} f(x_{1}^{n-\tau_{i}(n)}, \ldots, x_{D}^{n-\tau_{D}(n)}, \xi^{n}) - E \left[ \nabla_{x} f(x_{1}^{n-\tau_{i}(n)}, \ldots, x_{D}^{n-\tau_{D}(n)}, \xi^{n}) | \mathcal{F}^{n} \right] \).

To see that (4) and (3) are equivalent, we observe that \( E \left[ \nabla_{x} f(x_{1}^{n-\tau_{i}(n)}, \ldots, x_{D}^{n-\tau_{D}(n)}, \xi^{n}) | \mathcal{F}^{n} \right] = \nabla_{x} F(x_{1}^{n-\tau_{i}(n)}, \ldots, x_{D}^{n-\tau_{D}(n)}, \xi^{n}) \) for \( n \geq 0 \) and \( 1 \leq i \leq D \). The aforementioned equality follows trivially from the definition of the filtration, \( \{ \mathcal{F}^{n} \}_{n \geq 0} \).

As stated earlier, \( \{ \xi^{n} \}_{n \geq 0} \) are \( \mathcal{S} \)-valued i.i.d. (independent and identically distributed) samples. They represent network and environment fluctuations. In Remark 3, we briefly discuss the case where different agents are affected by different local network fluctuations, i.e., agent-i is affected by \( \xi_{n}^{i} \) at time \( n \).

3.1 Assumptions on (3)

(A1) (i) \( \nabla_{x} f \) is continuous.

(ii) It is locally Lipschitz continuous in the \( x \)-coordinate and may change with the second coordinate.

(A2) The step-size sequence \( \{ a(n) \}_{n \geq 0} \) satisfies the following conditions:

(i) \( \sum_{n \geq 0} a(n) = \infty \) and \( \sum_{n \geq 0} a(n)^{2} < \infty \).

(ii) \( \lim_{n \to \infty} \sup_{y \in [x, 1]} \frac{a(y/n)}{a(n)} < \infty \) for \( 0 < x \leq 1 \).

(iii) \( \sup_{n \geq 0} a(n) \leq 1 \).

(iv) For \( m \leq n \), we have \( a(n) \leq \kappa a(m) \), where \( \kappa > 0 \).
(v) There exists $1/2 < \eta < 1$ and a non-negative integer-valued random variable $\tau$ such that:
(a) $a(n) = o(n^{-\eta})$.
(b) $\tau$ stochastically dominates all $\tau_{kl}(n)$ and satisfies
$$E\left[\frac{\tau^1}{\eta}\right] < \infty.$$ 

(A3) $\sup_{n \geq 0} \|x^n\| < \infty$ a.s.

(A4) $\liminf_{n \to \infty} \frac{\mu(n,i)}{n} > 0$ for $1 \leq i \leq D$.

(A5) Almost surely, $\limsup_{n \to \infty} \|e^n\| \leq \epsilon$ for some fixed $\epsilon > 0$.

3.2 Remarks on the assumptions

Remark 1. Let us say that we are given a distributed delayed gradient-based iterative optimization algorithm. We first check whether:
(i) the local updates of this algorithm are described by (3);
(ii) the information delays due to the associated communication protocol and network delays/losses satisfy (A2)(v). If these two conditions are satisfied, then the analyses associated with (3) can be applied to understand its behavior. The other assumptions are typically satisfied in many real-world scenarios. In Section 4, we refine (A2)(v) to obtain a practical verifiable condition (A6). This refinement encompasses the requirements on the network topology and QoS, and CP.

Remark 2. Assumption (A5) requires that the stochastic additive error terms be norm-bounded asymptotically. These error terms may arise due to sampling or due to the use of approximate gradients instead of exact ones. Let us suppose that some or all $\nabla_x f\left(x_1^{n-\tau_1(n)}, \ldots, x_D^{n-\tau_D(n)}, \xi^n\right)$, $1 \leq i \leq D$, are unavailable or cannot be easily computed. Further, let us suppose that each agent may sample the objective function. Then a SPSA-C style \cite{16} gradient approximator may be used in its stead. This is given by
$$\frac{f\left(\cdot + \Delta, \xi^n\right) - f\left(\cdot - \Delta, \xi^n\right)}{2c^2},$$
where $0 < c < \infty$ is a perturbation parameter, and $\Delta$ is a $d$-dimensional random perturbation vector with $i^{th}$ component $\Delta_i$. Note that since $\tau_i(n)$ is assumed to be zero without loss of generality, $x_i^n = x_i^{n-\tau_i(n)}$. The approximation errors associated with using this gradient estimator are shown to be in $o(c^2)$, see \cite{16} for details. Further, these errors are captured by the $e$ sequence, and satisfy (A5). In particular, the analysis and the algorithms presented can be readily extended to approximate gradient methods.

Remark 3. In (3), each agent is affected by the same realization of network fluctuations. In reality, however, network fluctuations may be local and vary from agent to agent. Hence, $\xi^n$ in (3) must be replaced by $\xi_{ni}^n$ (specific to agent-$i$). Let us assume that $\{\xi_{ni}^n\}_{n \geq 0, 1 \leq i \leq D}$ are independent. Then, we may define
The assumptions (A1)–(A5) are motivated by the theory developed by [17]. To apply the theory from [17], for the analysis of (3), we present a couple of technical lemmata. Lemma 1 is needed, since unlike [17], we allow for objectives such that \( \nabla x f \) is locally Lipschitz continuous in the \( x \)-coordinate. This is because local Lipschitz continuity is a property that is easy to verify. Further, the constant associated with every \( \hat{x} \in \mathbb{R}^d \) may vary with \( \xi \). Through this lemma, we guarantee the existence of a global Lipschitz constant, that does not vary with \( x \) or \( \xi \), under the following restrictions: \( x \) belongs to a compact convex subset \( K \subset \mathbb{R}^d \) (restrict the decision search space to a sample path dependent compact convex subset), and \( S \) is such that some one-point-compactification theorem is applicable. Note that we state the theorem for compact \( S \) and discuss how its proof can be extended to general \( S \) using one-point-compactification theorems. Lemma 2 shows that the errors arising from the use of gradient samples, as opposed to expected gradients, vanish asymptotically.

To apply the theory from [17], we rewrite (4) in the following form:

\[
x^{n+1} = x^n - \pi(n)\lambda^n \\
(\nabla_x f(x_1^{n-\tau_1(n)}, \ldots, x_D^{n-\tau_D(n)}, \xi^n) + c^n + M^{n+1}),
\]

where \( \pi(n) := \max_{i \in Y^n} a(\nu(n, i)) \), \( q(n, i) := \frac{a(\nu(n, i))I(i \in Y^n)}{\pi(n)} \) and

\[
\lambda^n := \begin{bmatrix} Q^n_1 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & Q^n_D \end{bmatrix}_{d \times d},
\]

\[
Q^n_i := \begin{bmatrix} q(n, i) & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & q(n, i) \end{bmatrix}_{d_i \times d_i}.
\]

Recall that there are \( D \) agents in the system, the dimension of agent-\( i \) is \( d_i \) and \( \sum_{i=1}^D d_i = d \). For \( 1 \leq i \leq d \) and \( n \geq 0 \), \( Q^n_i \) is a diagonal matrix of dimension \( d_i \times d_i \), such that every diagonal element equals \( q(n, i) \). For \( n \geq 0 \), \( \lambda^n \) is a \( d \times d \) diagonal matrix obtained by placing the \( Q^n_i \) matrices along its diagonal. The term \( q(n, i) \) can be interpreted as the step-size of agent-\( i \) at time \( n \), in relation...
to the maximum step-size of an active agent \( \pi(n) \). The \( \lambda^n \) matrix is used to capture the transient relative update of various agents with respect to \( \pi(n) \). The various agent updates are related to each other via this maximum step-size. To analyze the discrete-time algorithm \( \text{\cite{[3]}} \), we use \( \{\pi(n)\}_{n \geq 0} \) to divide the time axis, and obtain a corresponding trajectory in the “continuous domain”. Here, the “limit” of \( \lambda^n \) captures the asymptotic relative update frequency of the agents.

**Lemma 1.** Given a compact convex set \( K \subset \mathbb{R}^d \), there exists \( 0 < L < \infty \) such that \( \|\nabla_x f(y, \xi) - \nabla_x f(z; \xi)\| \leq L\|x - y\| \forall x, y \in K \) and \( \xi \in S \). Further, if \( S \) is compact, then \( L \) is independent of \( \xi \), although may depend on \( K \).

**Proof.** From (A1)(ii) we get for every \( z \in K \) and \( \xi \in S \), there exists \( \infty > r(z) > 0 \) and \( \infty > L(z, \xi) > 0 \) such that \( \|\nabla_x f(y, \xi) - \nabla_x f(z; \xi)\| \leq L(z, \xi)\|y - z\| \) for every \( y \in B_{r(z)}(z) \) and \( \xi \in S \). Without loss of generality, \( L(z, \xi) \) is such that \( \|\nabla_x f(y^1, \xi) - \nabla_x f(y^2, \xi)\| \leq L(z, \xi)\|y^1 - y^2\| \) for every \( y^1, y^2 \in B_{r(z)}(z) \). Since \( K \) is compact, the open cover \( \{B_{r(z)/2}(z) \mid z \in K\} \) has a finite sub-cover given by \( S_c := \{B_{r(z)/2}(z^j) \mid 1 \leq j \leq M, z^j \in K\} \). Further, define \( \hat{L} := \max_{1 \leq j \leq M} L(z^j, \xi) \).

To prove the statement of the lemma, we consider some \( y^1, y^2 \in K \). It follows from the convexity assumption that \( y^1 \) and \( y^2 \) are connected by a straight line \( \ell \subset K \). It follows from the discussion above, that we can find a sequence of sets, \( B_{r(z^i)}(z^i1), \ldots, B_{r(z^im)}(z^im) \) from \( S_c \), covering the aforementioned line \( \ell \), such that \( m \leq M \), \( y^1 \in B_{r(z^i)}(z^i1) \), \( y^2 \in B_{r(z^im)}(z^im) \), and \( B_{r(z^i)}(z^i) \cap B_{r(z^{i+1})}(z^{i+1}) \neq \emptyset \) for \( 1 \leq i \leq m - 1 \). Now, we construct a sequence \( z^0, \ldots, z^m \) such that \( z^0 = y^1 \), \( z^m = y^2 \) and \( z^i \in B_{r(z^i)}(z^i) \cap B_{r(z^{i+1})}(z^{i+1}) \cap \ell, 1 \leq i \leq m - 1 \). Using this sequence, we get the following:

\[
\|\nabla_x f(y^1, \xi) - \nabla_x f(y^2, \xi)\| \leq \sum_{i=0}^{m-1} \|\nabla_x f(z^{i+1}, \xi) - \nabla_x f(z^i, \xi)\|, \\
\|\nabla_x f(y^1, \xi) - \nabla_x f(y^2, \xi)\| \leq \sum_{i=0}^{m-1} \hat{L}\|z^{i+1} - z^i\| \text{ and} \\
\|\nabla_x f(y^1, \xi) - \nabla_x f(y^2, \xi)\| \leq m\hat{L}\|y^1 - y^2\|.
\]

If we define \( \hat{L} := M\hat{L} \), then it may depend on both \( K \) and \( \xi \).

From the above discussions, we can find a minimum \( L(\xi) \) such that

\[
\|\nabla_x f(y^1, \xi) - \nabla_x f(y^2, \xi)\| \leq L(\xi)\|y^1 - y^2\|
\]

holds for every \( y^1, y^2 \in K \) with \( L(\xi) < \infty \) for all \( \xi \in S \). Note that \( L(\xi) \) depends on \( K \), but this dependency is omitted in the notation for the sake of clarity in presentation.

It is now left to show that there exists a constant that depends only on \( K \) and not on \( \xi \). To do this, we show that the map \( \xi \mapsto L(\xi) \) is continuous. Consequently, we may choose sup \( L(\xi) \) as the \( \xi \)-independent Lipschitz constant.

First, we need to show that it satisfies: \( \liminf_{\xi^n \to \xi} L(\xi^n) = L(\xi) \). To see this, observe that:

\[
\|\nabla_x f(y^1, \xi) - \nabla_x f(y^2, \xi)\| \leq \|\nabla_x f(y^1, \xi^n) - \nabla_x f(y^2, \xi^n)\| + \epsilon^n, \quad (6)
\]
where $\epsilon_n = \| \nabla_x f(y^1, \xi) - \nabla_x f(y^1, \xi_n) \| + \| \nabla_x f(y^2, \xi) - \nabla_x f(y^2, \xi_n) \|$. Clearly, $\lim_{\xi_n \to \xi} \epsilon_n = 0$, additionally recall that $\| \nabla_x f(y^1, \xi_n) - \nabla_x f(y^2, \xi_n) \| \leq L(\xi_n) \| y^1 - y^2 \|$. Now, taking $\liminf$ on both sides of (5), we get

$$\| \nabla_x f(y^1, \xi) - \nabla_x f(y^2, \xi) \| \leq \liminf_{\xi_n \to \xi} L(\xi_n) \| y^1 - y^2 \|.$$ 

It follows from the minimality of $L(\xi)$ that $L(\xi) \leq \liminf_{\xi_n \to \xi} L(\xi_n)$. It is left to show that $L(\xi) \geq \liminf_{\xi_n \to \xi} L(\xi_n)$. Let us assume the contrary. Then, there exists $\delta_0 > 0$ such that $L(\xi) + \delta_0 < \liminf_{\xi_n \to \xi} L(\xi_n)$. Hence there exists $N$ such that $\forall \ n \geq N$ $L(\xi) + \delta_0 < L(\xi_n)$. It follows from the minimality of $L(\xi_n)$ that there exists $\tilde{y}, \tilde{z} \in \mathcal{K}$ such that $\| \nabla_x f(\tilde{y}, \xi_n) - \nabla_x f(\tilde{z}, \xi_n) \| > (L(\xi_n) - \delta_0/2) \| \tilde{y} - \tilde{z} \|$. Taking $\liminf$ on both sides, we get $\| \nabla_x f(\tilde{y}, \xi) - \nabla_x f(\tilde{z}, \xi) \| > (L(\xi) - \delta_0/2) \| \tilde{y} - \tilde{z} \|$. This contradicts the minimality of $L(\xi)$. Hence, $\liminf_{\xi_n \to \xi} L(\xi_n) = L(\xi)$.

We make a further claim that $\lim_{\xi_n \to \xi} L(\xi_n)$ exists and equals $L(\xi)$. For this, we show that every sub-sequence of $\{ L(\xi_n) \}_{n \geq 0}$, such that $\xi_n \to \xi$, has a further subsequence whose limit is $L(\xi)$. Given a subsequence $\{ L(\xi_n^{(m)}(n)) \}_{n \geq 0}$, $\{ m(n) \}_{n \geq 0} \subset \{ n \geq 0 \}$, it follows from the above discussion that $\liminf_{\xi_n^{(m)}(n) \to \xi} L(\xi_n^{(m)}(n)) = L(\xi)$. Hence it has a further subsequence that converges to $L(\xi)$. In other words, the map $\xi \mapsto L(\xi)$ is continuous.

Now define $L := \max_{\xi \in \mathcal{S}} L(\xi)$. It follows from the continuity of the $\xi \mapsto L(\xi)$ map, that $L < \infty$. The statement of the lemma is now immediate. \hfill \Box

The above lemma states that the local Lipschitz property of $\nabla_x f$ is global provided we restrict the decision space to compact convex subsets of $\mathbb{R}^d$. For clarity in presentation, we assume that $\mathcal{S}$ is compact. In addition to being realistic, we do not lose any generality by making this assumption. This is because one can use one-point compactification theorems when $\mathcal{S} = \mathbb{R}^d$.

Define $c_n := \sum_{m=0}^{n-1} \pi(m) \lambda^m M^{m+1}$. The sequence $\{ \zeta_n \}_{n \geq 0}$ must be considered due to the use of samples instead of expected values. We will show that $\{ (\zeta_n, F^n) \}_{n \geq 0}$ is a zero-mean square integrable Martingale difference sequence, and use the Martingale convergence theorem to show that it converges almost surely. This is important, since this implies that the asymptotic behavior of $\{ \zeta \}$ is identical to the version that uses expected values.

**Lemma 2.** Almost surely $\lim_{n \to \infty} \zeta_n$ exists, where $\zeta_n = \sum_{m=0}^{n-1} \pi(m) \lambda^m M^{m+1}$ for $n \geq 0$.

**Proof.** To prove the above statement, we first show that $\{ (\zeta_n, F^n) \}_{n \geq 0}$ is a zero-mean square integrable Martingale difference sequence. Next, we show that its associated quadratic variation process is bounded almost surely. The convergence of $\zeta_n$ is then a direct consequence of the Martingale convergence theorem. \hfill \Box

Before proceeding, we make a few important observations. First, we note that it follows from (A3), that $x^n \in B_R(0)$ for some sample path dependent $R < \infty$. Using Lemma 1 we can associate $L$ with $B_R(0)$. Hence, for
\[ z \in B_R(0), \text{ we get:} \]
\[ \|\nabla_x f(z, \xi) - \nabla_x f(0, \xi)\| \leq L\|z\|, \text{ and} \]
\[ \|\nabla_x f(z, \xi)\| \leq \|\nabla_x f(0, \xi)\| + L\|z\|. \]

Since \( S \) is compact, we get that
\[ \|\nabla_x f(z, \xi)\| \leq K(1 + \|z\|), \tag{7} \]
where \( K < \infty \) is sample path dependent. For our second observation, we note that
\[ \sum_{n \geq 0} \pi(n)^2 < \infty \text{ as a consequence of (A2)(i) and (A4).} \]
Finally, we observe that the diagonal entries in \( \lambda^n \) are all less than 1.

It follows from the definition of \( M_n+1 \) that \( \mathbb{E}\zeta^n = 0 \). It follows from the above made observations and [7] that \( \mathbb{E}\|\zeta^n\|^2 < \infty \) a.s. Hence, we have that \( \zeta^n \) is a zero-mean square integrable Martingale difference sequence with filtration \( F_n \). To show that the associated quadratic variation process is bounded, it is enough to show that
\[ \sum_{n \geq 0} \|\zeta^{n+1} - \zeta^n\|^2 < \infty \text{ a.s.} \]
But this is a direct consequence of the above made observations. Hence, it follows from the Martingale Convergence Theorem that \( \lim_{n \to \infty} \zeta^n \) exists a.s. \( \square \)

An important component of our analysis is the verifiability of our assumptions. There are three of them whose demonstrability is unclear: (A1)(ii), (A2)(v) and (A3). We clarify the verifiability of (A1)(ii) in the following section. We defer the discussion on the verifiability of (A2)(v) till the end Section 4. This is because, we require the exact assumption on the network topology and quality, for this discussion. Finally, note that we do not discuss the verifiability of (A3). The readers may however refer to Section 5 of [17] for verifiable sufficient conditions for (A3).

### 3.4 On the verifiability of (A1)(ii)

Assumption (A1)(ii) requires that \( \nabla_x f \) be locally Lipschitz continuous in the \( x \)-coordinate. This is a gross weakening of similar ones found in literature that require global Lipschitz property, see for e.g., [5]. Since twice continuous differentiability is a sufficient condition for the local Lipschitz property, (A1)(ii) is easily verifiable. Additionally, in literature, the global Lipschitz constant is assumed to be independent of the second coordinate (\( \xi \) coordinate). However, (A2)(ii) allows for the local Lipschitz constant to vary with \( \xi \).

On the surface, it seems surprising that (A1)(ii) is able to allow for a two-fold weakening, without compromising verifiability. Typical assumptions in literature do not exhibit these two traits simultaneously. For (A1)(ii), this is not surprising in the light of Lemma 1. In particular, its proof leverages two ideas. First, it uses the stability assumption (A3) to move from local to global Lipschitzness, by restricting the set of interest, to a sample path dependent compact convex set obtained from (A3). Second, it eliminates the dependency on \( \xi \) by assuming, without loss of generality, that \( S \) is compact. This is done because \( S \) is typically \( \mathbb{R}^n, n \geq 1 \), or more generally a locally compact space. In this
case, one uses Alexandroff’s one point compactification theorem \cite{7} to find a homeomorphic map into a subspace \( S_\ast \) of a compact space. Also, \( f \) is easily extended to be viewed as a map from \( \mathbb{R}^d \times S_\ast \). Then, instead of viewing samples as arising in \( S_\ast \), one uses corresponding samples in \( S_\ast \). Now that \( S_\ast \) is compact, the statement of Lemma \cite{1} holds.

### 3.5 Convergence Analysis of \( [3] \)

We are finally ready to analyze \( [3] \). As stated earlier, our assumptions and lemmata are developed and presented so that the theory from \cite{17} can be used in the analysis. The reader must note that the notations used in this section are consistent those in \cite{17}. To avoid redundancies, we do not repeat overlapping proofs and refer the reader to corresponding ones from \cite{17}.

At time \( n \), agent-i performs gradient descent in its local decision space using \((x_1^{n-\tau_{i}(n)}, \ldots, x_D^{n-\tau_{i}(n)})\), where \( x_j^{n-\tau_{i}(n)} \) is the latest agent-j estimate that is available with agent-i at time \( n \). The error incurred at time \( n \) with respect to agent-i, \( e_i^n \), is given by:

\[
e_i^n := a(\nu(n, i)) I(i \in Y^n) \times \|\nabla_x f(x_1^n, \ldots, x_D^n, \xi^n) - f(x_1^{n-\tau_{i}(n)}, \ldots, x_D^{n-\tau_{i}(n)}, \xi^n)\|.
\]

It follows from Lemma \cite{4} that \( \nabla_x f \) is globally Lipschitz continuous when restricted to \( \overline{B_R}(0) \), where \( x^n \in \overline{B_R}(0) \) for \( n \geq 0 \), and \( 0 < R < \infty \) is obtained from (A3). Hence,

\[
e_i^n \leq a(\nu(n, i)) L \|x^n - (x_1^{n-\tau_{i}(n)}, \ldots, x_D^{n-\tau_{i}(n)})\|. \tag{8}
\]

To bound the right hand side of the above equation, it is enough to consider \( \|x_j^n - x_j^{n-\tau_{i}(n)}\|, 1 \leq j \leq D \), separately. Using assumptions (A3), (A2)(iv) we get that almost surely

\[
\|x_j^n - x_j^{n-\tau_{i}(n)}\| \leq \sum_{m=n-\tau_{i}(n)}^{n-1} \|x_j^{m+1} - x_j^m\| \leq C \tau_i(n),
\]

for some sample path dependent \( C < \infty \). To proceed with the analysis, we need the following technical lemma.

**Lemma 3.** \( P(\tau_{ji}(n) > n^9 \, i.o.) = 0 \).

**Proof.** To prove the above statement, we show that \( \sum_{n \geq 0} P(\tau_{ji}(n) > n^9) < \infty \) and invoke the Borel-Cantelli lemma \cite{8}.

Recall that (A2)(v) requires the existence of \( \eta \) such that \( \mathbb{E}\tau^{1/\eta} < \infty \) and \( a(n) \in o(n^{-\eta}) \). It is easy to see one can choose \( \eta \) such that \( \mathbb{E}\tau^{1/(\eta-\delta)} < \infty \), \( 1/2 \leq \eta - \delta \) and (A2)(v) hold, using a very small \( \delta \). Let us define \( \beta := \eta - \delta \), then \( \mathbb{E}\tau^{1/\beta} < \infty \). Also, recall that \( \tau_{i}(n) \) is stochastically dominated by \( \tau \) for \( n \geq 0 \) and \( 1 \leq i,j \leq D \). Hence,

\[
P(\tau_{ji}(n) > n^9) \leq P(\tau > n^9).
\]

13
It follows from the Markov inequality that

\[ P(\tau > n^\eta) \leq \frac{E\tau^{1/\beta}}{n^{\eta/\beta}}, \]

and that

\[ \sum_{n \geq 0} P(\tau_{ji}(n) > n^\eta) \leq E\tau^{1/\beta} \sum_{n \geq 0} \frac{1}{n^{\eta/\beta}}. \]

We have that \( \{\frac{1}{n^{\eta/\beta}}\}_{n \geq 1} \) is a summable sequence, since \( \beta < \eta \) by construction. Hence we get that \( \sum_{n \geq 0} P(\tau_{ji}(n) > n^\eta) < \infty \), as required.

Informally, the above lemma states that there is a sample path dependent \( 0 < N < \infty \) such that \( \tau_{ij}(n) \leq n^\eta \) for all \( n \geq N \). From (A2)(v)(a) and (A4), we get that almost surely \( a(\nu(n,i))\tau_{ij}(n) = o(1) \) for \( n \geq N \). If we combine these observations with (8), we get that \( e^n \in o(1) \) for \( n \geq N \). In other words, the errors associated with delays vanish asymptotically.

The remainder of the analysis, in this section, overlaps with the analysis presented in Section 4 of [17]. Hence, we merely outline the important steps involved and refer the reader to [17] for details.

Using (A1)–(A5) and hitherto presented discussions, we get that (8) converges to a closed connected invariant set associated with

\[ \dot{x}(t) \in \Lambda(t)\nabla_x F(x(t)) + B_x(0), \]

where \( \epsilon \) is the asymptotic norm-bound on the additive error terms. It may be noted that the \( \Lambda \) process may be thought of as capturing the relative update frequency of the agents involved, in the asymptotic sense.

It follows from (A4) that all the agents are updated in the same order of magnitude. Hence, for \( 1 \leq i, j \leq D \), one can ensure that:

\[
\lim_{n \to \infty} \frac{\sum_{m=0}^{n} a(\nu(m,i))}{\sum_{m=0}^{n} a(\nu(m,j))} \quad \text{exists.}
\]

When the above equation is satisfied, the step-size sequence is said to be “balanced”. An important role of (A4) is to ensure that the given steps-size sequence is “balanced”. Then, it follows from the theory developed in [4] that

\[
\Lambda(t) = \begin{pmatrix} 1/D & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & 1/D \end{pmatrix}, \quad t \geq 0.
\]

The diagonal matrix in the above equation is denoted by \( \text{diag}(1/D, \ldots, 1/D) \). We may now conclude that (3) converges to a closed connected invariant set associated with

\[ \dot{x}(t) \in \text{diag}(1/D, \ldots, 1/D)\nabla_x F(x(t)) + B_x(0). \quad (9) \]

Before we present our main result, let us state Theorem 2 from Chapter 6 of [1]:

**Let** \( H \) **be a upper semicontinuous set-valued map, from a compact** \( K \subset \mathbb{R}^d \) **to** \( \mathbb{R}^d \).
such that $H(x)$ is compact convex for every $x \in \mathbb{R}^d$. Further, let $x(\cdot)$ be a solution to $\dot{x}(t) \in H(x(t))$ that converges to $x^* \in \mathcal{K}$. Then, $x^*$ is an equilibrium point of $H$.

As stated earlier, the asymptotic behavior of (3) and (9) are identical. It follows from (A3) that there exists $R < \infty$ such that $x^n \in B_R(0)$ for all $n \geq 0$. Since any limit-point of the $x^n$ sequence belongs to $B_R(0)$, a solution to (9) tracked by (3), converges to $B_R(0)$. The conditions of the above stated theorem are satisfied, and (9) converges to an equilibrium point of $diag(1/D, \ldots, 1/D) \nabla_x F + B_\epsilon(0)$. In particular, (3) converges to $x^\infty$ such that $0 \in diag(1/D, \ldots, 1/D) \nabla_x F(x^\infty) + B_\epsilon(0)$. Hence (3) converges to a small neighborhood of the minimum values of $F$. Further, this neighborhood is a function of $\epsilon$, the asymptotic norm-bound on the additive error terms. The following theorem is an immediate consequence of the hitherto presented discussions. Again, the reader is referred to Section 4 of [17] for details.

**Theorem 1.** Under $(A1)-(A5)$, the general asynchronous optimization algorithm given by (3) converges to a small neighborhood of the minimum set of $F$. The size of this neighborhood is a function of the asymptotic norm-bound on the additive error terms.

**Remark 4.** If $\epsilon = 0$ in (A5), i.e., there are no additive errors, then we get that (3) converges to $x^\infty$, an equilibrium point of $diag(1/D, \ldots, 1/D) \nabla_x F$. Since, $diag(1/D, \ldots, 1/D) \nabla_x F(x^\infty) = 0$, we get that $\nabla_x F(x^\infty) = 0$. In other words, when $\epsilon = 0$, iteration (3) converges to a minimum of $F$. As $\epsilon$ increases, the algorithm is expected to converge farther from the minimum set. Therefore, when $\epsilon \uparrow \infty$ a minimum of $F$ cannot be found.
then \(N^{in}(i) = N^{out}(i)\) for every \(i\). If every agent can communicate with every other agent directly, then \(N^{in}(i) = N^{out}(i) = \mathcal{V} \setminus \{i\}\).

We assume that agent-\(i\) maintains the latest available estimates, \(\hat{X}_i := (\hat{x}_{1i}, \ldots, \hat{x}_{Di})\), from other agents in the system. At time \(n\), \(\hat{x}_{ij} = x^m_{i(j)}\) for some \(0 \leq m_{i(j)} \leq n\) and \(1 \leq j \leq D\). To facilitate consistent updates for \(\hat{X}_i\), \(1 \leq i \leq D\), each agent appends its estimate with a local timestamp, before transmitting across the network. Without loss of generality, agent-\(i\) has instant access to its local estimates, i.e., \(\hat{x}_{ii} = x^n_{i}\) at time \(n\). Also without loss of generality, we assume that the agents exchange information in accordance to the network clock, although they update local estimates in accordance to local clocks. Specifically, agent-\(i\) sends \(\hat{X}_i\) to every agent in \(\mathcal{N}^{out}(i)\) and receives \(\{\hat{X}_j \mid j \in \mathcal{N}^{in}(i)\}\) at every tick of the network clock. Using \(\{\hat{X}_j \mid j \in \mathcal{N}^{in}(i)\}\) and the time-stamps contained within them, agent-\(i\) updates its list \(\hat{X}_i\). Note that both local clocks and network delays lead to asynchronicity between agents. Let us codify these ideas into the following algorithm.

### Algorithm 1: Optimization algorithm w.r.t. agent-\(i\)

- Initialize local decision variable estimate \(\hat{x}_i\);
- Synchronize with other agents to initialize \(\hat{X}_i\);
- for every tick of the network clock do
  - if the local clock also ticked then
    - 1. Obtain network sample \(\xi\);
    - 2. \(\hat{x}_i \leftarrow \hat{x}_i - a(n(i))\nabla_x f(\hat{X}_i, \xi)\);
    - 3. Update \(\hat{x}_{ii}\) of \(\hat{X}_i\) to the new \(\hat{x}_i\) and append the local timestamp \(\nu(n,i)\);
  - end
  - 4. Send \(\hat{X}_i\) to agent-\(j\), for \(j \in \mathcal{N}^{out}(i)\);
  - 5. Receive \(\hat{X}_j\) from agent-\(j\), for \(j \in \mathcal{N}^{in}(i)\);
  - 6. Update \(\hat{X}_i\) using \(\{\hat{X}_j \mid \hat{X}_j \text{ successfully received}, j \in \mathcal{N}^{in}(i)\}\);
- end

Since we assume imperfect communication, the estimate-list sent by agent-\(j\) may not reach agent-\(i\). In other words, in step 5 of Algorithm 1, agent-\(i\) may only receive a strict subset of \(\hat{X}_j\)’s from its neighbors. Using this, it updates \(\hat{x}_{ki}\), the estimate associated with agent-\(k\), to that \(\hat{x}_{kj}\) which is appended with the largest \(\nu(m,k)\) (local timestamp of agent-\(k\)), that is also greater than the \(\nu(m,k)\) associated with the current estimate \(\hat{x}_{ki}\), where \(m \leq n\), \(1 \leq k \leq D\) and \(j \in \mathcal{N}^{in}(i)\).

**Remark 5.** Algorithm 1 states that all agents send and receive information in accordance to the network clock (steps 4 and 5). However, this requirement can be easily relaxed to obtain identical convergence results. In fact, agent-\(i\) is only required to send information every time its “list” \(\hat{X}_i\) is updated, or in the case of repeated failed transmissions. The list is then updated, at every tick of the local clock and when information is received from its neighbors.
4.1 Network requirement (A6) and convergence of Algorithm 1

Before proceeding, let us introduce a few useful graph theoretical terms [22]. A path $P$ is a sequence of vertices such that any vertex is connected to the next vertex by an edge, i.e., $P = i_0, \ldots, i_m$ and $(i_j, i_{j+1}) \in E$ for $0 \leq j \leq m - 1$. The length of a path is given by the number of edges in it, and an edge is trivially a length-1 path. We use $iPj$ to represent a path whose first vertex is $i$ and last vertex is $j$. We use $p(iPj, x)$ to represent the probability of successfully transmitting information $x$ from agent-$i$ to agent-$j$, using path $P$, in the shortest possible time. In other words, $p(iPj, x)$ represents the probability that every link along $P$ transmits successfully, one after the other. If the wireless channels are modeled as Markov fading channels then $p(iPj, x)$ may change over time.

To apply the analysis from Section 3.5 to understand Algorithm 1, we need the following simple assumption on the underlying transmissions.

**(A6)**

(i) For every $1 \leq i, j \leq D$, there exists a path connecting agent-$i$ to agent-$j$, $iP^i_j$, such that $p(iP^i_j, x) > 0$.

(ii) At any time, the success or failure of transmissions along different edges are independent.

Informally (A6) requires that: (a) there is a positive probability of transmitting information between any two agents and (b) the success / failure of transmissions over one edge does not interfere with transmissions over the others. Clearly (A6)(ii) is the harder requirement, as compared to (A6)(i), to ensure. One way, is by letting the channel frequencies be sufficiently separated, so as to cause zero/minimal interference with each other. In typical real-world applications such as industries, the network topology is fixed and the channel quality varies with the system state. In particular, at any time-step, transmissions along different channels are independent when conditioned on the current system state, see [15] for details. The independence requirement in (A6)(ii) can be readily relaxed to the aforementioned conditional independence for the industrial application.

To apply the analysis from Section 3.5 to understand Algorithm 1, we must verify (A1)–(A5). Since there are no additive errors, (A5) is trivially satisfied. (A1) is satisfied by design. If the experimenter enforces that the agents must synchronize their clocks with the network clock, then (A4) is satisfied. While this enforcement is standard, there are several other ways to ensure (A4). Note that (A4) merely requires that all agents are updated in the same order of magnitude. This condition is stated by requiring that the number of agent-$i$ updates is in $\Theta(n)$, $1 \leq i \leq D$. Note that the $\Theta$ notation is used to indicate that the two quantities involved have similar asymptotic growth. Since, the experimenter is typically at liberty to choose the step-size sequence, (A2)(i)-(iv) are trivially satisfied. We refer the reader to sufficient conditions in [17] to ensure (A3). It is now left to show that (A2)(v) is satisfied.

**Lemma 4.** Under (A6), Algorithm 1 satisfies (A2)(v).

**Proof.** We prove the statement of the lemma by finding a random variable $\tau$ that stochastically dominates $\tau_{ij}(n)$ such that $\mathbb{E}\tau^2 < \infty$, $1 \leq i, j \leq D$ and
\( n \geq 0 \). Then, \((A2)(v)\) is satisfied for any \( \eta \in (1/2, 1) \). Fix \( 1 \leq i, j \leq D \) and \( n \geq 0 \). Recall that \( \tau_{ij}(n) \) denotes the age of the agent-i-estimate available at agent-j at time \( n \), i.e., the age of \( \hat{x}_{ij} \in \hat{X}_j \). Let \( \mathcal{P} \) be the shortest length path satisfying \((A6)(i)\), and let \( l \) be its length. We define a random variable \( \hat{\tau}_{ij}(n) \), to denote the age of the latest agent-i-estimate that was successfully transmitted using \( \mathcal{P} \) in the shortest possible time, i.e., in \( l \) steps. Since \( \tau_{ij}(n) \) denotes the age of the last agent-i-estimate that was successfully transmitted, we have that \( \tau_{ij}(n) \leq \hat{\tau}_{ij}(n) \) a.s. Hence, it is enough to find a random variable \( \tau \) that stochastically dominates \( \hat{\tau}_{ij}(n) \).

Let \( A_{ij}^t \) denote the event that the \( \hat{X}_i \)-vector sent at time \( k \) (of the network clock) was successfully transmitted via \( \mathcal{P} \) in \( l \) steps. Then,

\[
P(\hat{\tau}_{ij}(n) > m) = P\left( \bigcap_{n-m \leq t \leq n-l} (A_{ij}^t)^c \right),
\]

where \( (A_{ij}^t)^c \) denotes the complement of event \( A_{ij}^t \). We claim that the events \( \{A_{ij}^t \mid n-m \leq t \leq n-l\} \) are independent. To see this, we begin by observing that \( A_{ij}^t \) unfolds over time starting at \( t \) and ending at most before \( t+l \). Further, at any given time instance each of these events can be associated with different edges. The independence of the above mentioned set of events now follows from the independence of the associated edges, i.e., \((A6)(ii)\). Now, equation (10) becomes:

\[
P(\hat{\tau}_{ij}(n) > m) = \prod_{n-m \leq t \leq n-l} P\left( (A_{ij}^t)^c \right) \leq p^{m-l+1}.
\]

It follows from \((A6)(i)\) that there is \( p \), such that \( \sup_{n-m \leq t \leq n-l} P\left( (A_{ij}^t)^c \right) \leq p < 1 \). Hence we get,

\[
P(\hat{\tau}_{ij}(n) > m) \leq p^{m-l+1}.
\]

Please note that the above inequality is independent of \( n \). Further, without loss of generality, the above inequality holds for every \((i,j)\) pair with the same \( p \) and \( l \).

Let us define \( \tau \) by describing its cumulative distribution function as follows: \( P(\tau \geq 0) := 1 \) and \( P(\tau \geq m) := p^{m-l} \), \( m \geq 1 \). It now follows from (11) that \( \tau \) stochastically dominates \( \tau_{ij}(n) \) for all \( 1 \leq i, j \leq D \) and \( n \geq 0 \).

It is left to show that \( E\tau^2 < \infty \). To do this, we observe that:

\[
E\tau = \sum_{m=0}^{\infty} m P(\tau \geq m) \leq \sum_{m=0}^{\infty} mp^{m-l}.
\]

The required result now follows from the simple observation that:

\[
\sum_{m=0}^{\infty} mp^{m-l} \leq \int_0^{\infty} x^{(l/p)-(x+l)} \, dx < \infty.
\]

Hence, we have shown that Algorithm 1 satisfies the assumptions required to apply Theorem 1. The following is therefore an immediate consequence of the aforementioned theorem and Remark 4.

**Theorem 2.** Algorithm 1 finds a minimum of \( F \) when \((A1), (A2)(i-iv), (A3) \) and \((A6)\) are ensured.
5 Empirical Results

We now present experimental results to substantiate the hitherto developed theory. We consider a 16-agent system such that $\mathbb{R}^2$ is the local decision space of every agent in the system. The agents are connected by a wireless network whose topology is time-varying. This is simulated by randomly choosing, at every step, between $N = 4$ digraphs $\{G_n \equiv (V, E_n)\}_{n=1}^4$. Since we assume full asynchronicity, it is important to note that the network topology is randomly switched at every tick of the network clock. For $1 \leq i \leq 16$, define $\rho_i \equiv a_i/b_i$, where $a_i$ is the number of agent-i clock ticks for every $b_i$ network clock ticks. We call $\rho_i$ the $i$th tick-ratio. For our experiments $\{\rho_i\}_{i=1}^{16}$ are chosen to be independent random variables, each sampled uniformly from $[1/5, 1]$. Note that $\{\rho_i\}_{i=1}^{16}$ is used to simulate the asynchronicity between various agents in the system, and with the wireless network. We choose them in $[1/5, 1]$, since we want to satisfy (A4).

For added complexity, we constrain each of the 4 network topologies to contain exactly 8 edges, i.e., $|E_n| = 8$, $1 \leq n \leq 4$. Each edge represents a communication link that is modeled as a Markov fading channel with state space $Z = \{z_0, z_1, \ldots, z_{K-1}\}$, parameterized by tuples $(T, p, e)$, where $T$ denotes the transition probability matrix, $p$ denotes the steady state probability vector $p$ (defined by the equilibrium condition $p^T T = p^T$), and $e$ denotes the crossover probability vector, see [10] for details. In each channel state $z_d$, fading results in communication drop outs with probability in accordance to the $d$-th component of the crossover probability vector $e$.

Let us say that all agents try to simultaneously minimize the perturbed “average network distance”. The average is taken over a network switching process involving 4 different topologies. Then, the global objective function is given by:

$$F(x) := E_\xi f(x, \xi),$$

where $f(x, \xi) := x^T [L_\xi + 0.1 I] x; x = (x_1, \ldots, x_{16})$ is the appended decision vector; $L_\xi$ is the Laplacian matrix associated with the graph $G_\xi \equiv (V, E_\xi); \xi \in \{1, \ldots, 4\}$ is the random variable associated with network switching; and $I$ is a square matrix of order 32. The Laplacian $L_\xi$ associated with a network topology is positive semi-definite. When perturbed by $0.1I$, it becomes positive definite. We work with this perturbed “average network distance” measure to force the associated optimization problem to have a unique global minimum. Working with a perturbed Laplacian is common in literature, see [2,18]. There are several ways to perturb a Laplacian, we do the above as it is simple, and is sufficient to illustrate the properties of Algorithm 1.

The analyses from Sections 3.5 and 4 are applicable, provided assumption (A6) is satisfied for the experimental setup described above. Convergence to the minimum of $F$ is then a consequence of Theorem 2. To show that (A6) is satisfied, we consider the union graph $\hat{G} := (V, \bigcup_{n=1}^4 E_n)$. At the tick of the network clock, the sample of $\xi$ determines which among the 4 available network topologies is to be activated. For $\hat{G}$, this means that a subset of edges $E_\xi$, corresponding to topology $G_\xi$, is “active”. This means that all possible transmissions, at that time, are along these active edges. There is no information transmission along the remaining “inactive” edges. We however, take the quasi-
equivalent view that transmissions along inactive edges are also possible, only to fail with probability 1. This view allows us to move from time-varying network topologies to considering a static network topology, given by $\hat{G}$, that does not change over time. $(A6)(i)$ is satisfied when $\hat{G}$ is strongly connected. Recall that a digraph is strongly connected when any vertex is reachable from any other vertex. $(A6)(ii)$ is ensured if we assume that all channels are uncorrelated. In practice, this assumption is satisfied when the channel frequencies are sufficiently separated.

Fig. 1 illustrates the union-graph of the four network topologies used in our experiments. Algorithm 1 is run for 5000 steps, using the following step-size sequence:

$$a(\nu(n, i)) = \frac{1}{\nu(n, i) + 50}, \text{ } n \leq 5000 \text{ and } 1 \leq i \leq 16$$

The initial estimates/decision variables of all 16 agents are randomly chosen from the annulus centered at the origin with inner ring radius of 500 and outer ring radius of 1000. The Laplacian matrices were perturbed such that the origin is the unique minimizer of $F$. The results of our experiments are illustrated in Figures 2 and 3. The various agent estimate-trajectories, in $\mathbb{R}^2$, are illustrated in Fig. 2. The estimates converge to the local optima, which put together minimize the “average network distance”. This is illustrated by the convergence of the perturbed “average network distance” towards the origin, in Fig. 3.

6 Special Case: Cumulative Consensus

Consensus is an important optimization problem that commonly arises in multi-agent systems. Popular forms of consensus include cumulative and average consensus. In this section, we focus on the cumulative consensus problem. Here, the D-agent system cooperatively tries to solve the following global optimization
problem:

\[
\arg\min_{x \in \mathbb{R}^d} \sum_{i=1}^{D} f_i(x),
\]

where \( f_i : \mathbb{R}^d \to \mathbb{R} \) is a local function that is only accessible to agent-i; \( x \equiv (x(1), \ldots, x(D)) \) and \( x(i) \) is interpreted as a local control variable of agent-i.

The global objective function \( F \) is such that \( F : x \mapsto \sum_{i=1}^{D} f_i(x) \).

We begin by noticing the key differences between problems (1) and (12). In (1), noisy samples of the global objective function is available to all agents. In (12), each agent only observes its local objective, and in this sense the global objective is partially observed. On the surface, the consensus problem may seem harder than (1) due to partial observability. However, we present simple modifications to Algorithm 1 to solve the consensus problem. Further, we show
that the analyses from Section 3.5 and 4 describe completely the behavior of this new modified algorithm.

The solution to the cumulative consensus presented in this section involves choosing one of the agents as an “arbiter”, who is then responsible for all computations involved in solving the given consensus problem. All other agents in the system support the arbiter by supplying it with information required to perform the aforementioned computations. A completely distributed approach, in the vein of [3], where the computational load is shared by all agents, will turn out to be a simple extension of the arbiter-approach. We present the former for the sake of simplicity and clarity. It may be noted that the arbiter-approach is susceptible to the “single point of failure” problem. However, it should also be noted that the arbiter-approach uses much lesser resources than the fully distributed approach. This is because all agents need only to communicate with the arbiter, not with each other.

Remark 6. Suppose that all paths connecting any two vertices in the associated communication graph pass through a single vertex/agent. Then, that agent is clearly a single point of failure even in a distributed approach. For this case, the arbiter approach is in fact optimal when the said agent is chosen as the arbiter.

Motivated by (A1), we make the following assumption regarding the local objectives:

(C) For $1 \leq i \leq D$, $\nabla f_i$ is locally Lipschitz continuous.

Similar to Algorithm 1, we associate a digraph $\mathcal{G}$ with the multi-agent system at hand. We assume that $\mathcal{G}$ satisfies (A6) and let agent-1 be the arbiter. Clearly agent-1 is strongly connected to every other agent. In the centralized paradigm, the arbiter is assumed to have complete knowledge of all local objectives $\{f_i\}_{i=1}^D$, and it may solve the consensus problem as follows:

$$x^{n+1} = x^n - a(n) \sum_{i=1}^D \nabla f_i(x^n). \tag{13}$$

However, as we are in a decentralized setting, agent-1 only has knowledge of $\nabla x f_1$. It must obtain all additional gradients from other agents in the MAS, if it is to execute an iteration similar to (13). We have assumed that the network is unreliable, and that every pair of agents may not be connected by a dedicated wireless channel. Hence, gradient information obtained by agent-1, from agent-j, $j \neq 1$, may be old. At time $n$, instead of (13), agent-1 (arbiter) executes:

$$x^{n+1} = x^n - a(n) \sum_{i=1}^D g^j_i(n),$$

where $g^j_i(n)$ is the, possibly delayed, agent-j gradient information available with the arbiter at time $n$.

Similar to Algorithm 1, agent-i updates and forwards the estimate-vector $\hat{X}_i := (\hat{x}_i, g^2_i, \ldots, g^D_i)$; where $\hat{x}_i$ is the latest estimate of the solution to the consensus problem available with agent-i, and $g^j_i$ is the latest agent-j gradient information that is available with agent-i. Specifically, at time $n$, $g^j_i(n) = \ldots$
∇f_j(x^n−τ(i,j)), for some delay 0 ≤ τ(i,j) ≤ n, for 2 ≤ j ≤ D. At every tick of the network clock, agent-i sends its estimate vector \( \hat{X}_i \) to all agents belonging to \( N^{out}(i) \). At the same time, it receives estimate-vectors \( \{ (\hat{x}_j, g^2_j, \ldots, g^D_j) \mid j \in N^{in}(i) \} \) from its “in-neighbors”. Due to the unreliable nature of the wireless network, it may only receive information from a strict subset of in-neighbors. This is then used to update \( \hat{X}_i \) at the next tick of the agent-i clock, similar to step-6 in Algorithm 1. Finally, agent-i calculates \( \nabla f_i(\hat{x}_i) \) and updates \( g_{ji} \) to this new gradient value. In the case of the arbiter, \( g_{ji} \), for 2 ≤ j ≤ D are updated and the following descent step is executed:

\[
x^{n+1} = x^n - a(n,1)I(1 \in Y_n) \sum_{i=1}^D g^1_{ji}
\]

(14)

Then, the arbiter revises \( \hat{x}_1 \) to \( x^{n+1} \) and \( g^1_1 \) to \( \nabla f_1(x^{n+1}) \). Although the agents forward information in accordance to the network clock, local computations and updates are governed by local clocks. This naturally leads to asynchronicity between agents, that is captured by the \( Y_n \) process in (14). Please note that we have considered asynchronicity for the sake of generality and completeness. To understand the main ideas of this section, the reader may simply omit the \( Y_n \) process, and assume that the agents are synchronized their clocks.

If we consider the gradient information used in (14), then \( g^1_1 = \nabla f_j(x^n−\tau_j) \), for some delay 0 ≤ \( \tau_j \) ≤ n, 1 ≤ j ≤ D. Each delay random variable \( \tau_j \) can be decomposed into \( \tau^1_j \) and \( \tau^2_j \), i.e., \( \tau_j = \tau^1_j + \tau^2_j \). Here \( \tau^1_j \) is the number of steps taken by \( x^n−\tau_j \) to reach agent-j from agent-1, and \( \tau^2_j \) is the time taken for \( \nabla f_j(x^n−\tau_j) \) to return to agent-1. Since the events associated with \( \tau^1_j \) and \( \tau^2_j \) are disjoint in time, it follows from (A6)(ii) that \( \tau^1_j \) and \( \tau^2_j \) are independent. Using arguments similar to the proof of Lemma 4, we can show that there is a random variable \( \tau \) that stochastically dominates \( \tau^1_j \) and \( \tau^2_j \), for 1 ≤ j ≤ D, such that \( \mathbb{E}\tau^2 < \infty \). Hence \( 2\tau \) stochastically dominates \( \tau_1 \) and \( \tau^2_j \), for 1 ≤ j ≤ D, such that \( \mathbb{E}(2\tau)^2 < \infty \), and (14) satisfies (A2)(ii). Now, we can readily apply analyses from Sections 3.5 and 4 to understand its long-term behavior, and the following theorem is immediate.

**Theorem 3.** Under (C), (A2)(i - iv), (A3), (A4) and (A6) the above described consensus algorithm, (14), achieves consensus.

For easy reference, below we present the consensus algorithm, obtained by modifying Algorithm 1.

**Remark 7.** Choosing the right agent as the arbiter is problem dependent. It is best to choose one based on “accessibility”, i.e., choose an agent that is connected by multiple short paths composed of reliable channels, to all other agents. On the other hand, an agent that is connected to most agents by a single unreliable path is a bad candidate. In this section, we have chosen agent-1 as the arbiter as we don’t assume anything beyond strong connectivity of the underlying graph.

### 6.1 Empirical Results: Algorithm 2

Although Algorithm 2 is expected to have similar empirical behavior to Algorithm 1, we present quick results of an experiment, to corroborate the discussion from Section 6. In our experiment, we considered a 16-agent system such that
Algorithm 2: Consensus Algorithm

Synchronize with other agents and initialize $\hat{\mathbf{x}}_1$;

for every tick of the network clock do

1. Send $\hat{\mathbf{x}}_i$ to agent-$j$, for $j \in \mathcal{N}^{\text{out}}(i)$;
2. Receive $\hat{\mathbf{x}}_j$ from agent-$j$, for $j \in \mathcal{N}^{\text{in}}(i)$;

if the local clock also ticked then

3. (i) Agent-$i$ ($i \neq 1$) updates $\hat{x}_i$ and $g^j_i$, $2 \leq j \leq D$, using
   \{$\hat{\mathbf{x}}_j \mid j \in \mathcal{N}^{\text{in}}(i)$\} (ii) $g^j_i \leftarrow \nabla f_i(\hat{x}_i)$;
4. Agent-$1$ updates $g^j_1$, $2 \leq j \leq D$, using \{$\hat{\mathbf{x}}_j \mid j \in \mathcal{N}^{\text{in}}(1)$\} (ii)
   $\hat{x}_1 \leftarrow \hat{x}_1 - a(\nu(n, 1)) \sum_{j=1}^{D} g^j_1$ (iii) $g^1_1 \leftarrow \nabla f_1(\hat{x}_1)$;

end

end

the common control variable belongs to $\mathbb{R}^{32}$. The underlying communication network used, is identical to the one from Section 5. Also, the asynchronicity between agents is simulated using the procedure explained in Section 5.

The local objective of agent-$i$ is given by:

$$f_i(x) = x^T A_i x + b_i^T x + c_i, \quad (15)$$

where $A_i \in \mathbb{R}^{32 \times 32}$ is a symmetric positive definite matrix, $b_i \in \mathbb{R}^{32}$ and $c_i \in \mathbb{R}$. The matrices $A_i$, the vectors $b_i$ and the scalars $c_i$ are all randomly generated.

We therefore have the following cumulative consensus problem:

$$\arg\min_{x \in \mathbb{R}^{32}} \sum_{i=1}^{16} f_i(x).$$

Its analytic solution is given by:

$$x = -\frac{1}{2} \left( \sum_{i=1}^{16} A_i \right)^{-1} \sum_{i=1}^{16} b_i. \quad (16)$$

The arbiter (Agent 1) uses the following step-size sequence to update the estimates (step-4, Algorithm 2).

$$a(\nu(n, 1)) = \frac{1}{\nu(n, 12)} + 150, \quad n \leq 2500 \quad (17)$$

The initial estimate is randomly chosen from a ball of radius 10 centered at the origin, and the algorithm is run for 2500 iterates. Fig. 4 illustrates the performance of Algorithm 2 when applied to solve the above described consensus problem. In the figure: (a) the dotted line represents the analytical solution to the cumulative consensus problem, (b) the dark blue line illustrates the convergence of the cumulative objective function to the analytical solution, and (c) all the other colored lines illustrate the convergence of local objective functions, of all agents in the MAS.
Figure 4: The dotted line represents the analytical solution to the consensus problem and the dark-blue line illustrates the convergence of the cumulative objective function, over 2500 iterates, to the analytical minimum. Convergence of local objective functions is also illustrated using different colors.

7 Conclusions

In this paper, we considered the problem of distributed optimization in large-scale multi-agent systems, connected by an unreliable wireless network. We presented an analytical framework to understand the long-term behavior of distributed delayed (approximate) gradient-based optimization algorithms. The theory developed, allows for a larger class of objective functions, as compared to similar results from the literature. Specifically, the presented analyses requires that the objective functions have locally Lipschitz continuous gradients. In addition to being easy to verify, this condition encompasses highly non-linear objectives. The framework also accommodates the use of approximate gradients instead of exact ones. A major contribution of this paper is in the development of practical verifiable requirements on network quality and topological properties. In particular, we merely require that the probability of successful information exchange between agents is positive. These probabilities may vary over time and there may be delays involved in the information exchange process. This is due to the unreliable nature of the underlying wireless network, in addition to possessing time-varying topological properties. In particular, when the topological switching is governed by some stochastic process, our theory holds water even when the switching process is non-stationary. To the best of our knowledge, ours is the first analyses under such weak conditions. The theory was also used to develop a simple easy-to-implement optimization algorithm that uses distributed delayed gradient information.

Finally, the theory was also used to develop an algorithm to solve the consensus problem from control. To the best of our knowledge, ours is the first algorithm, and associated analysis, that considers truly unbounded stochastic delays for information exchange between agents, in addition to time-varying
topologies.

In the future, we would like to conduct a thorough study of the rate of convergence of such algorithms. In particular, we would like to study the impact of both network topology and quality on the rate of convergence. Information exchange is an imperative aspect of the algorithms presented. However, it may be expensive. We also plan to study the optimal amount of information that needs to be transmitted, for solving the optimization problem. We would explore the possibility of developing a dynamic adaptive policy for information exchange.

References

[1] J-P Aubin and Arrigo Cellina. *Differential inclusions: set-valued maps and viability theory*, volume 264. Springer Science & Business Media, 2012.

[2] RB Bapat, SJ Kirkland, and S Pati. The perturbed laplacian matrix of a graph. *Linear and Multilinear Algebra*, 49(3):219–242, 2001.

[3] Michel Benaîm, Josef Hofbauer, and Sylvain Sorin. Stochastic approximations and differential inclusions. *SIAM Journal on Control and Optimization*, 44(1):328–348, 2005.

[4] Vivek S Borkar. Asynchronous stochastic approximations. *SIAM Journal on Control and Optimization*, 36(3):840–851, 1998.

[5] Vivek S Borkar. Stochastic approximation with ‘controlled Markov’noise. *Systems & control letters*, 55(2):139–145, 2006.

[6] Vivek S Borkar. *Stochastic approximation: a dynamical systems viewpoint*, volume 48. Springer, 2009.

[7] Glen E Bredon. *Topology and geometry*, volume 139. Springer Science & Business Media, 2013.

[8] Rick Durrett. *Probability: Theory and Examples*, volume 49. Cambridge university press, 2019.

[9] Hadrien Hendrikx, Francis Bach, and Laurent Massoulié. Asynchronous accelerated proximal stochastic gradient for strongly convex distributed finite sums. *arXiv preprint arXiv:1901.09865*, 2019.

[10] Hong Shen Wang and N. Moayeri. Finite-state Markov channel-a useful model for radio communication channels. *IEEE Transactions on Vehicular Technology*, 44(1):163–171, 1995.

[11] Wenying Hou, Zongze Wu, Minyue Fu, and Huanshui Zhang. Constrained consensus of discrete-time multi-agent systems with time delay. *International Journal of Systems Science*, 49(5):947–953, 2018.

[12] Angelia Nedic and Ji Liu. Distributed optimization for control. *Annual Review of Control, Robotics, and Autonomous Systems*, 1:77–103, 2018.

[13] Angelia Nedic and Asuman Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48, 2009.
[14] Angelia Nedić and Asuman Ozdaglar. Convergence rate for consensus with delays. *Journal of Global Optimization*, 47(3):437–456, 2010.

[15] Daniel E Quevedo, Anders Ahlen, and Karl H Johansson. State estimation over sensor networks with correlated wireless fading channels. *IEEE Transactions on Automatic Control*, 58(3):581–593, 2012.

[16] Arunselvan Ramaswamy and Shalabh Bhatnagar. Analysis of gradient descent methods with nondiminishing bounded errors. *IEEE Transactions on Automatic Control*, 63(5):1465–1471, 2018.

[17] Arunselvan Ramaswamy, Shalabh Bhatnagar, and Daniel E Quevedo. Asynchronous stochastic approximations with asymptotically biased errors and deep multi-agent learning. *IEEE Transactions Automatic Control* (cond. accept), arXiv:1802.07935v2, 2019.

[18] Israel Rocha and Vilmar Trevisan. A fiedler-like theory for the perturbed laplacian. *Czechoslovak Mathematical Journal*, 66(3):717–735, 2016.

[19] Benjamin Sirb and Xiaojing Ye. Decentralized consensus algorithm with delayed and stochastic gradients. *SIAM Journal on Optimization*, 28(2):1232–1254, 2018.

[20] William Turin and Robert Van Nobelen. Hidden Markov modeling of fading channels. In *VTC’98. 48th IEEE Vehicular Technology Conference. Pathway to Global Wireless Revolution (Cat. No. 98CH36151)*, volume 2, pages 1234–1238. IEEE, 1998.

[21] Yinghui Wang, Wenxiao Zhao, Yiguang Hong, and Mohsen Zamani. Distributed subgradient-free stochastic optimization algorithm for nonsmooth convex functions over time-varying networks. *SIAM Journal on Control and Optimization*, 57(4):2821–2842, 2019.

[22] Douglas Brent West et al. *Introduction to graph theory*, volume 2. Prentice hall Upper Saddle River, NJ, 1996.

[23] George Yin, Le Yi Wang, and Thu Nguyen. Switching stochastic approximation and applications to networked systems. *IEEE Transactions on Automatic Control*, 2018.