Asynchronous stochastic approximations with asymptotically biased errors and deep multi-agent learning

Arunselvan Ramaswamy† arunr@mail.upb.de
Shalabh Bhatnagar*§ shalabh@iisc.ac.in
Daniel E. Quevedo†† *Senior Member, IEEE* dquevedo@ieee.org

***Abstract*—Asynchronous stochastic approximations (SAs) are an important class of model-free algorithms, tools and techniques that are popular in multi-agent and distributed control scenarios. To counter Bellman's curse of dimensionality, such algorithms are coupled with function approximations. Although the learning/control problem becomes more tractable, function approximations affect stability and convergence. In this paper, we present verifiable sufficient conditions for stability and convergence of asynchronous SAs with biased approximation errors. The theory developed herein is used to analyze Policy Gradient methods and noisy Value Iteration schemes. Specifically, we analyze the asynchronous approximate counterparts of the policy gradient (A2PG) and value iteration (A2VI) schemes. It is shown that the stability of these algorithms is unaffected by biased approximation errors, provided they are asymptotically bounded. With respect to convergence (of A2VI and A2PG), a relationship between the limiting set and the approximation errors is established. Finally, experimental results are presented that support the theory.**

***Index Terms*—Asynchronous stochastic approximations, multi-agent learning, networked control systems, distributed control, almost sure boundedness (stability), deep reinforcement learning, neuro-dynamic programming, deep function approximations, asymptotically biased approximation errors.**

I. Introduction

In recent years reinforcement learning (RL) algorithms such as noisy approximate Value Iteration, Q-learning and Policy Gradient Iteration methods have witnessed a colossal resurgence. Many of these algorithms are coupled with function approximators to solve several important problems including, but not limited to, autonomous driving in transportation, process optimization in industrial scenarios and efficient dispersal of health-care services. A neural network with several hidden layers is called a deep neural network (DNN). RL that uses a DNN for function approximation is called DeepRL. The literature around DeepRL is growing rapidly, for example see [19], [20], [30], [18] and [17].

†Software Innovation Lab and Dept. of Computer Science, Paderborn University, Paderborn 33098, Germany.
*Department of Computer Science and Automation and the Robert Bosch Centre for Cyber Physical Systems, Indian Institute of Science, Bangalore - 560012, India.
§ supported by the RBCCPS, IISc, the J.C. Bose National Fellowship from SERB, and a project from the department of Science and Technology under the ICPS grant.
††School of Electrical Engineering and Robotics, Queensland University of Technology (QUT), Brisbane, Australia

Most modern learning and control problems have continuous state and/or action spaces. Thus the use of RL is hindered by **Bellman's curse of dimensionality**. To overcome this curse, learning and control algorithms are often coupled with function approximation. It is worth noting that the previously mentioned resurgence is partly owing to the effectiveness of DNN in function approximation. While optimization problems become tractable, the use of function approximators affects stability (almost sure boundedness) and convergence properties of the algorithms. Further, the optimality of the policies found depends on the **approximation errors**. Such issues are not well studied. **The main contribution of this paper is a complete analysis in terms of the influence of function approximation on stability and convergence (characterizing the limiting set as a function of the approximation errors), in a multi-agent setting**.

While the theory behind traditional RL is mature, there have not been many attempts to analyze DeepRL. Munos analyzed the approximate value and policy iteration algorithms, see [22] and [21]. However the assumptions in [22] and [21] are rather restrictive. Ramaswamy and Bhatnagar [25] studied approximate value iteration methods under significantly weaker assumptions as compared to [22]. However, [25] does not consider the multi-agent scenario. In this paper, we present the framework to develop and analyze large-scale multi-agent RL algorithms. Such algorithms are applicable to industrial process-control, distributed control of microgrids and decentralized resource allocation systems, among others. *It may be noted that in the setting of distributed control and learning, the aforementioned curse of dimensionality problem is particularly pronounced.*

A. Motivation, relevant literature and our contributions

The main motivation for this paper is the development of a general framework for learning and control in large-scale multi-agent settings. In a typical multi-agent architecture, the agents involved need to work towards a common goal by cooperating with each other. Each agent may be asynchronously performing updates, i.e., according to its own local clock, and not that of other agents. While the agents may act independently, their decisions are based on information from other agents. This information is often shared via wireless

communication networks. Bottlenecks in communication resources may lead to (possibly unbounded) delays as well as errors in communication.

Here, we focus on developing a framework which accounts for all of the above constraints. Such a framework would then guide the development of algorithms with behavioural guarantees. For this, we build on tools and techniques developed from the stochastic approximations literature, as in [13], [1], [23], [3], [5], [25] and [2]. Traditional analyses in [13] and [1] do not account for the use of function approximation. Whilst interesting, the multi-agent analyses in [23] do not consider the important question of **algorithmic stability**. Further, the results of [23] do not characterize the limiting set as a function of the approximation errors.

DNN is a popular choice for function approximation among practitioners of RL. It is used to approximate objective functions such as Q-factors, value functions, policies, gradients, etc. Such a DNN is trained in an online manner to minimize some "loss". The network architecture is typically chosen without explicit knowledge of the objective function. Hence, one cannot expect the approximate objective function obtained using a DNN to equal the true objective function everywhere, even after sufficient training. Consequently, the approximation errors are likely to be **biased** (have non-zero mean). Not surprising, these biases may affect the stability and quality of convergence, see Remark 1 for details.

Value and policy iteration are popular RL algorithms that are at once effective and easy to implement. Similar to other RL algorithms, these also suffer from the Bellman's curse of dimensionality. In this paper, we analyze multi-agent value and policy gradient iterations which utilize DNN based function approximators. In related work, documented in [25], value iteration with function approximation is analyzed. However, it does not consider the multi-agent setting. Abounadi et. al. [1] studied the asynchronous version of Q-learning, but without considering function approximation. *To the best of our knowledge the complete analysis of the aforementioned RL algorithms within multi-agent settings, with DNN based function approximation, is new to the literature.* The multi-agent approximate value iteration algorithm is called Asynchronous Approximate Value Iteration (A2VI), and the corresponding policy iteration algorithm is called Asynchronous Approximate Policy Gradient Iteration (A2PG), see Section V for details. With respect to A2VI, online samples are used to get better approximations of the Bellman operator. Simultaneously, these approximations are used in a distributed fixed point iteration. Since A2VI and A2PG operate in online settings, the approximation errors may initially be large. As more samples are obtained, these errors reduce over time, but may, in general, be biased. With respect to A2PG, we consider a generic framework that encompasses distributed implementations of the algorithms that search a given parameterized policy space in order to find the optimal policy. This policy search may be aided by an advantage function, e.g., value function, which is not explicitly considered in the analysis.

The main contributions of this paper are:

1) We show that the stability of the algorithms remains unaffected by error biases, provided these do not grow over time.
2) We provide an explicit relation between the biases and the limiting set. This allows one to trade DNN complexity for solution quality.
3) Although we consider approximation error biases, there are no additional restrictions on the quality of communication. In fact, we present our results under standard (yet general) assumptions on communication as used, e.g., in [13].
4) Our theory is used to analyze the asynchronous approximate counterpart of value (A2VI) and policy gradient (A2PG) iterations.

B. Tool set: Asynchronous Stochastic Approximations

Stochastic approximation algorithms (SAs) encompass a class of iterative algorithms that are model-free and sample-based. SAs find the minimum/maximum of a given objective function through a series of approximations. Traditionally, the approximation errors are expected to vanish over time. In 1951 the first SA was developed by Robbins and Monro [28] for finding a root of a given regression function. The theory of modern SAs was developed by Benaïm [3], Benaïm and Hirsch [4] and Borkar [12]. This theory was extended to SAs with set-valued mean-fields by Benaïm, Hofbauer and Sorin [5] [6], Ramaswamy and Bhatnagar [26], Perkins and Leslie [23], Bianchi et. al. [9], and others. The reader is referred to books by Borkar [11] and Kushner and Yin [16] for a more detailed exposition on this topic.

Although the traditional SA framework can be used to develop and analyze algorithms in RL and stochastic control, it does not encompass multi-agent and distributed scenarios. The latter setting was studied by Borkar [13]. He considered multi-agent algorithms wherein the agents are asynchronous and communications are delayed/ erroneous. Such algorithms are called asynchronous SAs. Many RL algorithms such as Q-learning, value iteration and policy gradient methods have asynchronous counterparts. These algorithms are designed and analyzed using the theory developed in [13] and [1]. The stability issue of asynchronous SAs was studied in [8].

C. Organization

The organization of the remainder of this paper is as follows:

- In Section II we present the assumptions involved in the analysis of asynchronous stochastic approximations with asymptotically bounded biased, errors, *i.e.,* recursion (2).
- In Sections III-A, III-B and III-C, we present a convergence analysis of (2) under the assumptions presented in Section II. The main technical result of this paper, Theorem 1, is enunciated in Section III-B. This result is then moulded through the use of Borkar's balanced step-sizes [13], into the desired statement in Section III-C.
- In Section IV, we show that *the stability of algorithms remains unaffected provided the approximation errors are*

guaranteed to be asymptotically bounded, albeit non-diminishing and possibly biased.

- In Section V-A, our theory is used to understand the long-term behavior of A2VI. *We show that A2VI converges to a fixed point of the perturbed Bellman operator, when Borkar's balanced step-sizes are utilized. We also establish a relationship between these fixed points and the approximation errors.*
- In Section V-B we briefly outline a similar analysis for A2PG. *We show that A2PG converges to a small neighborhood of local minima of the parameterized policy function $\pi(\cdot)$. This neighborhood is shown to be related to the approximation errors.*
- In Section V-C we discuss the verifiability of assumption $(S5)$. Finally, we summarize our contributions in Section VI.
- In Section V-D we present a bird's-eye view of two real-world applications. Specifically, we briefly discuss solutions to vehicle platooning and cumulative consensus, using the theory developed in this paper.
- In Section V-E we present simple experimental results to support our theory.

D. Definitions and Notations

[Set closure] Given $A \subset \mathbb{R}^d$, then $\overline{A}$ is used to represent the closure of $\mathcal{A}$.

[Limiting set] Given $\{x_n\}_{n\geq 0} \subset \mathbb{R}^d$, its limiting set is given by $\bigcap_{N\geq 0} \overline{\{x_n \mid n \geq N\}}$.

[Distance between point and set] Given $x \in \mathbb{R}^d$ and $A \subseteq \mathbb{R}^d$, the distance between x and A is given by: $d(x, A) := \inf\{\|x - y\| \mid y \in A\}$.

[δ-open neighborhood ($N^\delta(\cdot)$)] We define the *δ-open neighborhood* of $A \subset \mathbb{R}^d$ by $N^\delta(A) := \{x \mid d(x, A) < \delta\}$.

[Upper-semicontinuous map] We say that a set-valued map $H : \mathbb{R}^n \to \{\text{subsets of } \mathbb{R}^m\}$ is upper-semicontinuous, if for given sequences $\{x_n\}_{n\geq 1}$ (in $\mathbb{R}^n$) and $\{y_n\}_{n\geq 1}$ (in $\mathbb{R}^m$) such that $x_n \to x$, $y_n \to y$ and $y_n \in H(x_n)$, $n \geq 1$, we have $y \in H(x)$.

[Marchaud Map] A set-valued map $H : \mathbb{R}^n \to \{\text{subsets of } \mathbb{R}^m\}$ is called *Marchaud* if it satisfies the following properties: **(i)** for each $x \in \mathbb{R}^n$, $H(x)$ is a convex and compact set; **(ii)** *(point-wise boundedness)* for each $x \in \mathbb{R}^n$, $\sup_{w\in H(x)} \|w\| < K(1 + \|x\|)$ for some $K > 0$; **(iii)** H is *upper-semicontinuous*.

Let H be a Marchaud map on $\mathbb{R}^d$. The differential inclusion (DI) given by

$$\dot{x} \in H(x) \tag{1}$$

is guaranteed to have at least one solution that is absolutely continuous. The reader is referred to [2] for more details. We say that $\mathbf{x} \in \sum$ if $\mathbf{x}$ is an absolutely continuous map that satisfies (1). The *set-valued semiflow* Φ associated with (1) is defined on $[0, +\infty) \times \mathbb{R}^d$ as:

$\Phi_t(x) = \{\mathbf{x}(t) \mid \mathbf{x} \in \sum, \mathbf{x}(0) = x\}$.

For $B \times M \subset [0, +\infty) \times \mathbb{R}^d$, we define

$$\Phi_B(M) = \bigcup_{t\in B,\ x\in M} \Phi_t(x).$$

II. Assumptions for convergence analysis

In this paper we are interested in the complete analysis of **asynchronous SAs with non-diminishing biased additive errors**. The general iterative structure of such algorithms is given by:

$$x_{n+1}(i) = x_n(i) + a(\nu(n,i))I(i \in Y_n) \left[(\mathcal{A}f)_i(x_{n-\tau_{1i}(n)}(1), \ldots, x_{n-\tau_{di}(n)}(d)) + M_{n+1}(i)\right], \text{ where} \tag{2}$$

1) $x_n = (x_n(1), \ldots, x_n(d)) \in \mathbb{R}^d$, $n \geq 0$.
2) $f : \mathbb{R}^d \to \mathbb{R}^d$ is a Lipschitz continuous objective function.
3) Y_n is a subset of $\{1, 2, \ldots, d\}$ for each $n \geq 0$. It represents the number of agents that are **active** between time-steps $n - 1$ and n.
4) $0 \leq \tau_{ji}(n) \leq n$ is the (stochastic) delay experienced by agent i in receiving information from agent j at time n. In other words, at time n, the information obtained by agent i from agent j is $\tau_{ji}(n)$ time-steps old.
5) $\nu(n, i) = \sum_{m=0}^{n} I(i \in Y_m)$ is the number of times that agent i was active (i.e., updated its component parameter) up until time n. *It may be noted that the time index n in equation (2) represents the global clock, and thus, grows unbounded. We analyze the algorithm with respect to this clock.* Let us say that agent-2 has been updated 34 times when the global clock has been updated 50 times. Then we have $\nu(50, 2) = 34$. Note that the global clock is only required for analysis, the implementation remains fully asynchronous, outside of causal assumptions, see (S2) in Section IV.
6) $\mathcal{A}$ is the approximation operator (eg., DNN), $\{a(n)\}_{n\geq 0}$ is the given step-size sequence and $\{M_{n+1}\}_{n\geq 0}$ is a square integrable Martingale difference noise sequence.

At time-step n, the information regarding agent-j that is available with agent-i is $\tau_{ji}(n)$ steps old. The stochastic delay process τ could be unbounded. However, we make certain standard assumptions on their moments, see $(A2)(v)$ in Section III-B. Further, we assume that the agent-updates are all in the same order of magnitude, asymptotically, see $(S2)$ in Section IV-A. Under these assumptions, we shall show that (2) is stable and convergent.

It is worth noting that multi-agent approximate versions of value and policy gradient iterations, A2VI and A2PI respectively, are structurally identical to (2). We first present an analysis of (2). Later, this analysis is transcribed to obtain the desired theory for A2VI and A2PG, in Sections V-A and V-B respectively. In addition, stronger conclusions are drawn that are specific to A2VI and A2PG. Before proceeding with the analysis, the assumptions involved in the convergence analysis of (2) are listed.

(A1) $f : \mathbb{R}^d \to \mathbb{R}^d$ *is a Lipschitz continuous function with Lipschitz constant* L [1]. *Further,* $\mathcal{A}$ *is such that*

$$\mathcal{A}f(x_n) \in f(x_n) + \overline{B}_\epsilon(0) \tag{3}$$

for all $n \geq N$, *where* N *may be sample path dependent. Note that* $\overline{B}_\epsilon(0)$ *is a closed ball of radius* ϵ *centered at the origin. Here* $\epsilon > 0$ *is a fixed upper bound on the norm of the asymptotic approximation errors.*

(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) $\limsup_{n\to\infty} \sup_{y\in[x,1]} \frac{a(\lfloor yn \rfloor)}{a(n)} < \infty$ *for* $0 < x \leq 1$.

(A3) $\frac{n-\tau_{ij}(n)}{n} \to 1$ *a.s. for every* $1 \leq i < j \leq d$.

(A4) $\sup_{n\geq 0}\|x_n\| < \infty$ *a.s.*, where $\|x_n\|$ is used to represent the Euclidean norm of x_n, i.e., $\|x_n\| = \sqrt{x_n(1)^2 + \ldots + x_n(d)^2}$.

(A5) $\{M_{n+1}\}_{n\geq 0}$ *is a square integrable martingale difference sequence such that*

$$E\left[M_{n+1}(i) \mid \mathcal{F}_n\right] = 0.$$
$$E\left[\|M_{n+1}(i)\|^2 \mid \mathcal{F}_n\right] \leq K(1 + \sup_{m\leq n}\|x_m\|^2),$$

for all n, *where*

$\mathcal{F}_n := \sigma\langle x_m, M_m, Y_m, \tau_{ij}(m); 1 \leq i, j \leq d, m \leq n\rangle$, $1 \leq i \leq d$, $n \geq 0$ *and* $K > 0$ *is some fixed constant.*

We assume that all the agents are asynchronous. However, if we want the algorithm to learn effectively, then certain causal assumptions are necessary. $(A3)$ is one such assumption. Colloquially, $(A3)$ requires that the information delay between agents at time n is in $o(n)$, where $o(\cdot)$ is the standard *Little-O* notation. Without loss of generality, we assume that $\tau_{ii}(n) = 0$ for all i and n. In other words, we assume that an agent does not experience delays in accessing its own local information.

Remark 1. *In typical DeepRL applications, the approximation operator* $\mathcal{A}$ *is a DNN. The objective function* f *is typically one among the following: value function, Q-value function, policy function or Bellman operator. The operator* $\mathcal{A}$ *is trained in an online manner using loss functions that reduce the "approximation errors". The neural network architecture is fixed by the experimenter without complete knowledge of* f. *This certainly limits how well the chosen neural network can approximate* f. *In other words, there may not exist a set of network weights such that the approximation errors are guaranteed to be arbitrarily small. Hence, it is reasonable to merely hope that the errors do not grow over time. This is codified in* $(A1)$ *as* $\limsup_{n\to\infty} \|\mathcal{A}f(x_n) - f(x_n)\| \leq \epsilon$ *a.s. for some fixed* $\epsilon > 0$.

Brief overview of the steps involved in our analysis

- In Section III, convergence properties of (2) are analyzed under the almost sure boundedness assumption $(A4)$. Our analysis proceeds in two stages. In the first stage, presented in Section III-A, it is assumed that $\tau_{ij}(n) = 0$ for all $i,\ j$ and n, *i.e.,* **there are no communication delays**. In the second stage, presented in Section III-B, the effect of communication delays is considered. Specifically, it is shown that the errors due to delayed communications do not affect the analysis in Section III-A.
- In Section IV, we replace $(A4)$ in favor of verifiable conditions which guarantee stability of (2). **We do this by deriving assumptions that imply** $(A4)$. These assumptions are compatible with the conditions listed earlier in this section. Put together, they constitute an analytic framework for studying stability and convergence of (2).

[1] f is Lipschitz continuous *iff* $\exists L > 0$ such that $\forall x, y \in \mathbb{R}^d$ $\|f(x) - f(y)\| \leq L\|x - y\|$.

III. Convergence analysis

We are now ready to analyze the convergence of (2) under $(A1)$-$(A5)$. We begin our analysis by making the additional assumption that there are no communication delays, i.e., $\tau_{ji}(n) = 0\ \forall i, j\ and\ n$. This allows us to focus on the effect of asynchronicity between agents. Then, in Section III-B we show that the analysis in Section III-A is unaffected by the errors due to delayed communications.

A. Analysis assuming zero delays

Assuming $\tau_{ii}(n) = 0$, $\forall i\ and\ n$, equation (2) becomes:

$$\begin{aligned} x_{n+1}(i) = x_n(i) + a(\nu(n,i))I(i \in Y_n) \\ \left[(\mathcal{A}f)_i(x_n(1), \ldots, x_n(d)) + M_{n+1}(i)\right]. \end{aligned} \tag{4}$$

For $n \geq 0$, define $\overline{a}(n) := \max_{i\in Y_n} a(\nu(n,i))$ and $q(n,i) := \frac{a(\nu(n,i))}{\overline{a}(n)} I(i \in Y_n)$. It follows from (A2) that $\sum_{n\geq 0} \overline{a}(n) = \infty$ and $\sum_{n\geq 0} \overline{a}(n)^2 < \infty$. The quantity $\sum_{m=0}^{n} q(m,i)$ captures the fraction of time $\left(\sum_{m=0}^{n} \overline{a}(m)\right)$ that agent i is active. Thus, $q(m, \cdot)$ captures the *relative frequency of the agent updates*. For more details the reader is referred to Borkar [13].

Equation (4) can be further rewritten as follows:

$$\begin{aligned} x_{n+1}(i) = x_n(i) + \overline{a}(n)q(n,i) \\ \left[f_i(x_n(1), \ldots, x_n(d)) + \epsilon_n(i) + M_{n+1}(i)\right], \end{aligned} \tag{5}$$

where, $\epsilon_n = (\epsilon_n(1), \ldots, \epsilon_n(d))$ is the approximation error at stage n, *i.e.,* $\epsilon_n = \mathcal{A}f(x_n) - f(x_n)$. It follows from $(A1)$ that $\limsup_{n\to\infty} \|\epsilon_n\| \leq \epsilon$ for a certain $\epsilon > 0$ fixed. Since we are only interested in the asymptotic behavior of (2), without loss of generality, we may say that $\|\epsilon_n\| \leq \epsilon$ for all $n \geq 0$, even though we only have $\|\epsilon_n\| \leq \epsilon$ for all $n \geq N$ (for sample path dependent N). Note that, $\{x_n\}_{n\geq 0}$ and $\{x_n\}_{n\geq N}$ (subsequence starting at a sample point dependent N) have identical asymptotic properties.

For $n \geq 1$, define $t(0) := 0$, $t(n) := \sum_{m=0}^{n-1} \overline{a}(m)$. For $t \in [t(n), t(n+1))$, define $\overline{x}(t) := x_n$, $\overline{\lambda}(t) := diag(q(n,1), \ldots, q(n,d))$ and $\overline{\epsilon}(t) = \epsilon_n$ for $t \in [t(n), t(n+1))$. The notation $diag(a_1, \ldots, a_d)$ is used to denote the diagonal $d \times d$ matrix given by

$$\begin{bmatrix} a_1 & 0 & \ldots \\ \vdots & \ddots & \\ 0 & & a_d \end{bmatrix}.$$

Let us recall (5) in the following useful form:

$$x_{n+1} = x_n + \overline{a}(n) \lambda(t(n)) \left[f(x_n) + \epsilon_n + M_{n+1} \right]. \tag{6}$$

It follows from $(A4)$, $(A5)$ and $\sum_{n=0}^{\infty} \overline{a}(n)^2 < \infty$, that $\sum_{n \geq 0} \|\overline{a}(n) M_{n+1}\|^2 < \infty$. In other words, the quadratic variation process associated with $\xi_n := \sum_{m=0}^{n} \overline{a}(m)\lambda(t(m))M_{m+1}$, $n \geq 0$, is bounded almost surely. From this we may conclude that the martingale noise sequence, $\{\xi_n\}_{n \geq 0}$, is convergent almost surely. For a proof of the aforementioned, the reader is referred to *Chapter 2* of Borkar [11]. Given the above, the following lemma is immediate.

Lemma 1. *The martingale difference noise sequence* $\xi_n = \sum_{m=0}^{n} \overline{a}(m)\lambda(t(m))M_{m+1}$ *is convergent:* $\lim_{n \to \infty} \xi_n < \infty$ *a.s.*

For $s \geq 0$, define

$$x^s(t) := \overline{x}(s) + \int_s^{s+t} \lambda(\tau) \left(f(\overline{x}(\tau)) + \overline{\epsilon}(\tau) \right) \, d\tau.$$

Then $x^s(\cdot)$ is a solution to the non-autonomous DI

$$\dot{x}(t) \in \lambda(t+s) f(x(t)) + \overline{B}_\epsilon(0)$$

, with $\overline{x}(s)$ as its starting point. It follows from the definitions of $\overline{x}(\cdot)$, $x^s(\cdot)$, and from Lemma 1 that

$$\lim_{s \to \infty} \sup_{t \in [s, s+T]} \|\overline{x}(t) - x^s(t)\| = 0 \ a.s. \tag{7}$$

Therefore, the asymptotic behavior of (2) and (6) can be determined by studying the family of functions given by $\{x^s([0,T]) | s \geq 0, \ T > 0\}$.

For any fixed $T > 0$, the set $\{x^s([0,T]) \mid s \geq 0\}$ can be viewed as a subset of $D([0,T], \mathbb{R}^d)$, [2] equipped with the Skorohod topology.It follows from the Arzela-Ascoli theorem for $D([0,T], \mathbb{R}^d)$ that the above mentioned subset is relatively compact. For more details on Càdlàg spaces, Skorohod topology and the Arzela-Ascoli theorem, the reader is referred to Billingsley [10]. It now follows from (7) that $\{x^s([0,T]) \mid s \geq 0\}$ and $\{\overline{x}([s, s+T]) \mid s \geq 0\}$ have the same limit points in $D([0,T], \mathbb{R}^d)$. Hence, to find any subsequential limit of $\{\overline{x}(s+\cdot) \mid s \geq 0\}$, we merely need to consider the corresponding subsequence in $\{x^s([0,T]) \mid s \geq 0\}$. Finally, since T is arbitrary, $\{\overline{x}(s+\cdot) \mid s \geq 0\}$ is relatively compact in $D([0,\infty), \mathbb{R}^d)$.

Lemma 2. *Almost surely any limit point of* $\{\overline{x}(s+\cdot) \mid s \geq 0\}$ *in* $D([0,\infty), \mathbb{R}^d)$ *is a solution to the non-autonomous DI*

$$\dot{x}(t) \in \Lambda(t) f(x(t)) + \overline{B}_\epsilon(0), \tag{8}$$

where $\Lambda(\cdot)$ *is a* $d \times d$*-dimensional diagonal matrix-valued measurable function with diagonal entries in* $[0,1]$ [3].

[2] $D([0,T], \mathbb{R}^d)$ is used to represent the set of all Càdlàg functions with domain [0,T] and range $\mathbb{R}^d$. This is the set of all functions that are right continuous with left limits.

[3] Note that $\overline{B}_\epsilon(0) = \{x \mid \|x\| \leq \epsilon\}$ is the ϵ-closed ball centred at the origin. The open ball of radius r centered at the origin is represented by $B_r(0)$, and equals $\{x \mid \|x\| < r\}$.

Proof. As in the proof of *Theorem 2, Chapter 7* of Borkar [11], we view $\lambda(\cdot)$ as an element of $\mathcal{V}$, where $\mathcal{V}$ is the space of measurable maps $y(\cdot) : [0,\infty) \to [0,1]^d$ with the coarsest topology that renders continuous, the maps

$$y(\cdot) \to \int_0^t \langle g(s), y(s) \rangle ds,$$

for all $t > 0$, $g(\cdot) \in \mathbb{L}_2([0,T], \mathbb{R}^d)$[4].

Define $\hat{\epsilon}_s(t) := \lambda(t)\overline{\epsilon}(t)$ for all $t \geq 0$. Since $\hat{\epsilon}_s(\cdot)$ is measurable for every $s \geq 0$ and $\sup_{s \geq 0} \|\hat{\epsilon}_s\| < \infty$, we obtain that $\{\hat{\epsilon}_s([0,T]) \mid s \geq 0\}$ is relatively compact in $\mathbb{L}_2([0,T], \mathbb{R}^d)$. If necessary, by choosing a common subsequence of $\{\hat{\epsilon}_s([0,T]) \mid s \geq 0\}$ and $\{\lambda([s, s+T]) \mid s \geq 0\}$, we can show that any limit of $\{\overline{x}(s+\cdot) \mid s \geq 0\}$, in $D([0,T], \mathbb{R}^d)$, is of the form:

$$x(t) = x(0) + \int_0^t \Lambda(\tau) f(x(\tau)) d\tau + \int_0^t \epsilon(\tau) d\tau$$

$$= x(0) + \int_0^t \left[\Lambda(\tau) f(x(\tau)) + \epsilon(\tau) \right] d\tau,$$

where $\epsilon(\cdot)$ and $\Lambda(\cdot)$ are the subsequential limits of $\{\hat{\epsilon}_s([0,T]) \mid s \geq 0\}$ and $\{\lambda([s, s+T]) \mid s \geq 0\}$, respectively. Note that $\|\epsilon(t)\| \leq \epsilon$, for $t \geq 0$, and that $\epsilon(\cdot)$ is the weak limit in $\mathbb{L}_2([0,T], \mathbb{R}^d)$, as $s \to \infty$. Also note that $\Lambda(\cdot)$ is the limit in $\mathcal{V}$, equipped with the coarsest topology described above. □

The above lemma states that algorithm (2) tracks a solution to the non-autonomous DI (8). We needed to associate a DI and not an o.d.e. since the algorithm allows for asymptotically biased approximation errors. The non-autonomous $\Lambda(\cdot)$ is a consequence of asynchronicity. It is clear from the above Proof of Lemma 2 that $\Lambda(\cdot)$ captures the relative update frequencies of the various agents involved in a limiting sense. Recall from the beginning of this section, that $\{\overline{a}(n)\}_{n \geq 0}$ is used to divide the time axis. This plays a crucial part in translating the discrete-time algorithm to a continuous time trajectory, which in turn can be readily analyzed. In this sense, $\frac{\sum_{m=0}^{n} q(m,i)}{\sum_{m=0}^{n} \overline{a}(m)}$ is the fraction of time that agent-i is active with respect to the hypothetical global clock (whose sole function is in facilitating analysis). Clearly, any two agent updates are related to each other through this hypothetical clock.

[4] $\mathbb{L}_2([0,T], \mathbb{R}^d)$ is used to represent the set of all square integrable functions with domain $[0,T]$ and range $\mathbb{R}^d$. In other words,

$$\mathbb{L}_2([0,T], \mathbb{R}^d) = \left\{ f : [0,T] \to \mathbb{R}^d \ \middle| \ \int_0^T \|f(t)\|^2 dt < \infty \right\}.$$

B. Extension to account for delays

Now we are ready to factor in the influence of delays. To do this, we show that the statement of Lemma 2 is true even when $\tau_{ij}(n) > 0$. First, we consider additional restrictions on the step-size sequence and the delay processes $\tau_{ij}(n)$. Under these additional assumptions, we show that the analysis in Section III-A remains unaffected by the errors that arise from delayed communications. Before proceeding, we note that a methodology to deal with the effect of delays separately, was developed by Borkar in 1998, see [13]. We use similar techniques here. In order to avoid redundancies, we only provide additional details. The reader is referred to [13] or [11] for details. Recall the algorithm under consideration:

$$\begin{aligned} &x_{n+1}(i) = x_n(i) + a(\nu(n,i))I(i \in Y_n) \\ &\left[(\mathcal{A}f)_i(x_{n-\tau_{1i}(n)}(1), \ldots, x_{n-\tau_{di}(n)}(d)) + M_{n+1}(i)\right]. \end{aligned} \tag{9}$$

Our analysis makes use of the following additional refinements in $(A2)$:

(A2)(iii) $\sup_{n \geq 0} a(n) \leq 1$.

(A2)(iv) For $m \leq n$, we have $a(n) \leq \kappa a(m)$, where $\kappa > 0$.

(A2)(v) There exists $\eta > 0$ and a non-negative integer-valued random variable $\overline{\tau}$ such that:

(i) $a(n) = o(n^{-\eta})$.

(ii) $\overline{\tau}$ stochastically dominates all $\tau_{kl}(n)$ and satisfies

$$E\left[\overline{\tau}^{1/\eta}\right] < \infty.$$

In what follows we outline the proof of why (9) still tracks a solution to (8) even in the presence of delayed communications. Specifically, it is shown that the "effect" due to delays vanishes in the order of the step-sizes.

Let us consider the following quantity:

$$\begin{aligned} &a(\nu(n,i))I(i \in Y_n) \\ &\left|f_i(x_{n-\tau_{1i}(n)}(1), \ldots, x_{n-\tau_{di}(n)}(d)) - f_i(x_n(1), \ldots, x_n(d))\right|. \end{aligned}$$

There are no error terms due to the approximation operator $\mathcal{A}$, since they are already considered in the analysis presented in Section III-A. Since f is Lipschitz continuous, it is enough to find bounds for the terms

$$a(\nu(n,i))\left|x_n(j) - x_{n-\tau_{ji}(n)}(j)\right| \text{ for every } i \text{ and } j.$$

Clearly, the above term is bounded by

$$a(\nu(n,i)) \sum_{m=n-\tau_{ji}(n)}^{n-1} |x_{m+1}(j) - x_m(j)|.$$

Using (9) and the Lipschitz property of f, we get the following bound:

$$a(\nu(n,i)) \sum_{m=n-\tau_{ji}(n)}^{n-1} Ca(m) \leq Ca(\nu(n,i))\tau_{ji}(n),$$

for some constant $C > 0$. Our task is now reduced to showing that $a(\nu(n,i))\tau_{ji}(n) = o(1)$, which in turn follows from

$$P(\tau_{ji}(n) > n^{\eta} \ i.o.) = 0.$$

The above equation follows from $(A2)(v)$ and the Borel-Cantelli lemma. The following theorem is an immediate consequence of the analysis done hitherto.

Theorem 1. *Under assumptions* $(A1)$-$(A5)$*, the asynchronous approximation algorithm given by (2) has the same limiting set as the non-autonomous DI* (8)*.*

C. Balanced step-size sequences

A drawback in applying the above theorem to practical applications is the fact that the DI (8) is non-autonomous. Further, $\Lambda(\cdot)$ is not exactly known. Borkar [13] solved this problem through the use of "balanced step-size sequences". A step-size sequence $\{a(\nu(n,i))\}_{n \geq 0, 1 \leq i \leq d}$ is balanced if there exist $a_{ij} > 0$ for every pair of i and j such that

$$\lim_{n \to \infty} \frac{\sum_{m=0}^{n} a(\nu(m,i))}{\sum_{m=0}^{n} a(\nu(m,j))} = a_{ij}.$$

Recall that $q(n,i)$ captures the relative update frequency of agent-i with respect to the hypothetical clock used for analysis. The update frequencies of any two agents are related through this clock. Further, in the limiting sense, the relative update frequencies are captured by the Λ process. The reader may note that there is no explicit assumption on the relative update frequency. It is intrinsic to assumptions (A3), (S1)(ii) and (S2). Assumption (A3) states that the time-delays are $o(n)$, and (S2)(ii) states that each agent is updated in the order of magnitude of n, where n is the hypothetical clock which functions as an analytical aid. Typical diminishing square-summable step-size sequences are balanced provided all agents are updated in the same order of magnitude. This can be proven using a combination of the above mentioned assumptions, see [13]. In other words, the step-sizes considered in this paper are balanced.

As the step-sizes are balanced, we get $\Lambda(t) = diag(1/d, \ldots, 1/d)$ for all $t \geq 0$, see Theorem 3.2 of [13] for details. The tracking DI, (8), of Theorem 1 then becomes

$$\dot{x}(t) \in diag(1/d, \ldots, 1/d)f(x(t)) + \overline{B}_{\epsilon}(0). \tag{10}$$

As noted in [1], the qualitative behaviors of $\dot{x}(t) = f(x(t))$ and $\dot{x}(t) = diag(1/d, \ldots, 1/d)f(x(t)) = 1/d \ f(x(t))$ are similar since they only differ in scale. Further, it follows from the upper semi-continuity of chain recurrent sets that the long-term behavior of (10) is similar to that of $\dot{x}(t) = 1/d \ f(x(t))$ for small enough ϵ. Consequently, the long-term behavior of (10) approximates that of $\dot{x}(t) = f(x(t))$.

To summarize, we have shown that asynchronous SAs with asymptotically bounded biased errors track a solution to (10)*, when balanced step-sizes are used.* [5]

IV. STABILITY ANALYSIS

The foregoing analysis required that the iterates be bounded in an almost sure sense. This requirement is hard to ensure in the presence of function approximation and communication delays. It is well known that unbounded approximation errors can affect the stability of the algorithm, see [7]. *In this section, we present a set of sufficient conditions which ensure that* (2) *is stable.*

[5]Recall, from (A1), that ϵ of (10) is the norm-bound on the approximation errors.

A. Stability assumptions

Given $n \geq 0$ and $T > 0$, define $m_T(n) := \max\{m \mid m \geq n,\ t(m) - t(n) \leq T\}$. The goal of this section is to replace the stability assumption, $(A4)$, from Section II with verifiable conditions. These will be combined with the other assumptions to provide a complete analysis of stability and convergence.

(S1) (i) *The step-size sequence is eventually decreasing, i.e., $\exists\ N$ such that $a(n) \geq a(m)$ for all $N \leq n \leq m$.*

(ii) $\lim\limits_{n\to\infty} \frac{\sum_{m=0}^{\lfloor xn \rfloor} a(m)}{\sum_{m=0}^{n} a(m)} = 1$ *uniformly in* $x \in [y, 1]$, *where* $0 < y \leq 1$.

(S2) (i) $\liminf\limits_{n\to\infty} \frac{\nu(n,i)}{n+1} \geq \tau$, *for some* $\tau > 0$.

(ii) $\lim\limits_{n\to\infty} \frac{\sum_{m=\nu(n,i)}^{\nu(m_T(n),i)} a(m)}{\sum_{m=\nu(n,j)}^{\nu(m_T(n),j)} a(m)}$ *exists for all* i, j.

(S3) (i) *For all $n \geq 0$, we have $\|M_{n+1}\| \leq D$ a.s.*

(ii) $\lim\limits_{n\to\infty} \sum_{m=n}^{m_T(n)} a(m) M_{l(m)+1} = 0$, *where $\{l(m)\}_{m\geq 0}$ is an increasing sequence of non-negative integers satisfying $l(m) \geq m$.*

Note that $(S3)$ is stricter than $(A5)$. The analysis in this section assumes $(S3)$ instead of $(A5)$ merely for the sake of clarity. Later, it is shown that assuming $(A5)$ is sufficient. First, we consider a few definitions associated with DI $\dot{x}(t) \in H(x(t))$, relevant to the section:

> ***[Invariant Set]*** *$M \subseteq \mathbb{R}^d$ is invariant for the above DI if for every $x \in M$ there exists a trajectory, $\boldsymbol{x} \in \sum$, such that for $\boldsymbol{x}(0) = x$, $\boldsymbol{x}(t) \in M$, for all $t > 0$. Note that the definition of invariant set used in this paper, is the same as that of positive invariant set in [5] and [11].*
>
> ***[Strongly Positive Invariant Set]*** *$M \subseteq \mathbb{R}^d$ is strongly positive invariant for the above DI if $\Phi_t(M) \subset M$ for $t > 0$.*
>
> ***[Attracting set & fundamental neighborhood]*** *$A \subseteq \mathbb{R}^d$ is attracting, if it is compact and there exists a neighborhood U such that for any $\epsilon > 0$, $\exists\ T(\epsilon) \geq 0$ with $\Phi_{[T(\epsilon),+\infty)}(U) \subset N^{\epsilon}(A)$. Such a U is called the fundamental neighborhood of A. The basin of attraction of A is given by $B(A) = \{x \mid \bigcap_{t\geq 0} \overline{\Phi_{[t,\infty)}(x)} \subset A\}$.*
>
> ***[Attractor set]*** *In addition to being compact, if the attracting set is also invariant, then it is called an attractor.*
>
> ***[Inward directing set, [25]]*** *Given a differential inclusion $\dot{x}(t) \in H(x(t))$, an open set $\mathcal{O}$ is said to be an inward directing set with respect to the aforementioned differential inclusion, if $\Phi_t(x) \subseteq \mathcal{O}$, $t > 0$, whenever $x \in \overline{\mathcal{O}}$. Specifically, if $\mathcal{O}$ is inward directing, then any solution to the DI with starting point at the boundary of $\mathcal{O}$ is "directed inwards", into $\mathcal{O}$.*

(S4) *Associated with $\dot{x}(t) = f(x(t))$ is a compact set Λ, a bounded open neighborhood $\mathcal{U}$ $\left(\Lambda \subseteq \mathcal{U} \subseteq \mathbb{R}^d\right)$ and a function $V : \overline{\mathcal{U}} \to \mathbb{R}^+$ such that*

(i) *$\forall t \geq 0$, $\Phi_t(\mathcal{U}) \subseteq \mathcal{U}$ i.e., $\mathcal{U}$ is strongly positively invariant.*

(ii) *$V^{-1}(0) = \Lambda$.*

(iii) *V is a continuous function such that for all $x \in \mathcal{U} \backslash \Lambda$ and $y \in \Phi_t(x)$ we have $V(x) > V(y)$, for any $t > 0$.*

(S4a) *$\hat{\mathbb{A}}$ is the global attractor of $\dot{x}(t) = f(x(t))$.*

For the analysis we need that one of (S4) and (S4a) is satistfied. $(S4)$ and its variant $(S4a)$ are the key to our stability analysis. The two variants are overlapping yet qualitatively different, thereby covering a multitude of scenarios. Note that the above Lyapunov-based stability conditions are devised based on the ones in [25].

If $(S4)$ is satisfied, then *Proposition 3.25* of Benaïm, Hofbauer and Sorin [5] implies that $\dot{x}(t) = f(x(t))$ has an attractor set $\hat{\mathbb{A}} \subseteq \Lambda$. It also implies that $V^{-1}([0, r])$ is a fundamental neighborhood of $\hat{\mathbb{A}}$, for small values of r. On the other hand, if $(S4a)$ is satisfied, then any compact neighborhood of $\hat{\mathbb{A}}$ is a fundamental neighborhood of it. ***In both cases we can associate an attractor, $\hat{\mathbb{A}}$, and a fundamental neighborhood, $\overline{\mathcal{N}}$, to*** $\dot{x}(t) = f(x(t))$.

Given $\delta > 0$, $\exists\ \epsilon(\delta) > 0$ such that

$$\dot{x}(t) \in f(x(t)) + \overline{B}_{\epsilon(\delta)}(0) \tag{11}$$

has an attractor $\mathbb{A} \subseteq N^{\delta}(\hat{\mathbb{A}})$ *with fundamental neighborhood* $\overline{\mathcal{N}}$. This is a consequence of the upper semicontinuity of attractor sets, see [2] or [5] for details. We will next show that (2) converges to a neighborhood of a local/global attractor of $\dot{x}(t) = f(x(t))$, such as $\hat{\mathbb{A}}$. Further, this neighborhood depends on the approximation errors. Typically, the experimenter decides on the expected accuracy of the algorithm. This accuracy is quantified by δ. Once this accuracy is fixed, the function approximator (DNN) is trained to control the asymptotic errors to $\epsilon(\delta)$. Then one can show that (2) converges to $N^{\delta}(\hat{\mathbb{A}})$.

Before we proceed, we associate the following Lyapunov function to (11): $\tilde{V} : \overline{\mathcal{N}} \to \mathbb{R}_+$ such that $\tilde{V}(x) := \max\{d(y, \mathbb{A})g(t) \mid y \in \Phi_t(x), t \geq 0\}$ and $c \leq g(t) \leq d$ is a strictly increasing function with $c > 0$. Since $\overline{\mathcal{N}}$ is a fundamental neighborhood of $\mathbb{A}$, it follows that $\sup\limits_{x\in\overline{\mathcal{N}}} \tilde{V}(x) < \infty$.

The stability analysis requires choosing two bounded open sets, say $\mathcal{B}$ and $\mathcal{C}$, such that $\mathcal{C}$ is inward directing and $\mathbb{A} \subset \mathcal{B} \subset \overline{\mathcal{B}} \subset \mathcal{C}$. Recall that $\mathbb{A}$ is an attractor of $\dot{x}(t) \in f(x(t)) + \overline{B}_{\epsilon}(0)$ obtained from the definition of $\hat{\mathbb{A}}$ (see $(S4a)$). First, we choose $\mathcal{V}_r$ as $\mathcal{C}$ such that $\overline{\mathcal{V}_r} \subset \mathcal{U}$. This is possible for small values of r. Next, we choose an open $\mathcal{B}$ such that $\mathbb{A} \subset \mathcal{B} \subset \overline{\mathcal{B}} \subset \mathcal{C}$. This is possible since Λ is compact and $\mathcal{C}$ is open.

The following two propositions are necessary for our stability analysis. The reader is referred to [25] for their proofs.

Proposition 1. *For any $r < \sup\limits_{u\in\mathcal{N}} \tilde{V}(u)$, the set $\mathcal{V}_r := \{x \mid \tilde{V}(x) < r\}$ is open relative to $\overline{\mathcal{N}}$. Further, $\overline{\mathcal{V}}_r = \{x \mid \tilde{V}(x) \leq r\}$.*

Proposition 2. *$\mathcal{C}$ is an inward directing set associated with $\dot{x}(t) \in f(x(t)) + \overline{B}_{\epsilon}(0)$.*

Traditionally, the stability of algorithms such as (2) is ensured by projecting the iterates onto a compact set at every stage. If this set is not carefully chosen, then the algorithm may not converge, or converge to an undesirable set. Using

the previously constructed $\mathcal{B}$ and $\mathcal{C}$, we obtain the following **projective counterpart** of (2):

$$\hat{x}_{n+1} = z_n \text{ such that } z_n \in \bigsqcap_{\mathcal{B},\mathcal{C}}(\tilde{x}_n). \tag{12}$$

In the above equation, $\bigsqcap_{\mathcal{B},\mathcal{C}}$ is the projection operator defined as follows: given $\mathcal{B}$ and $\mathcal{C}$ subsets of $\mathbb{R}^d$, the projection map $\bigsqcap_{\mathcal{B},\mathcal{C}} : \mathbb{R}^d \to \{\text{subsets of } \mathbb{R}^d\}$ is given by:

$$\bigsqcap_{\mathcal{B},\mathcal{C}}(x) := \begin{cases} \{x\}, \text{ if } x \in \mathcal{C} \\ \{y \mid d(y,x) = d(x,\overline{\mathcal{B}}),\ y \in \overline{\mathcal{B}}\}, \text{ otherwise;} \end{cases}$$

$\hat{x}_0 = z_0$ such that $z_0 \in \bigsqcap_{\mathcal{B},\mathcal{C}}(x_0)$ and
$\tilde{x}_n(i) = x_n(i) + a(\nu(n,i))I(i \in Y_n)$
$\left[(\mathcal{A}f)_i(x_{n-\tau_{1i}(n)}(1), \ldots, x_{n-\tau_{di}(n)}(d)) + M_{n+1}(i)\right]$.

From the above set of equations, it is clear that the projective iterates $\{\hat{x}_n\}_{n\geq 0} \subseteq \mathcal{C}$. Since $\mathcal{C}$ is bounded by construction (see above), $\sup_{n\geq 0}\|\hat{x}_n\| < \infty$ a.s.

The realization of the projective scheme, (12), depends on finding sets $\mathcal{B}$ and $\mathcal{C}$, whose existence is clear from previous discussions. Since we only require these sets for stability analysis, we need not explicitly construct them. Below, we state our final stability assumption that relates an iteration and its projective counterpart. Its verifiability, within the context of A2VI, is discussed in V-C.

(S5) $\sup_{n\geq N}\|x_n - \tilde{x}_n\| < \infty$ *a.s. for sample path dependent* N.

B. The projective counterpart of (2)

In this section we begin the study of (2) by analysing its projective counterpart (12). In the previous section we used $\hat{x}_n$ to represent the projected iterates to distinguish from the original iterates generated by (2). **In this and the following couple of sections, we simply use x_n for the projected iterates, instead of $\hat{x}_n$s, to reduce clutter.** Due to this transient notational change, (12) becomes:

$$\begin{aligned} \tilde{x}(n+1) &= x_n + D_n\left[\mathcal{A}f(x_n) + M_{n+1}\right], \\ x_{n+1} &= z_n, \text{ where } z_n \in \bigsqcap_{\mathcal{B},\mathcal{C}}(\tilde{x}_{n+1}), \text{ with} \end{aligned} \tag{13}$$

$D_n = diag\left(a(\nu(n,1)I(1 \in Y_n), \ldots, a(\nu(n,d))I(d \in Y_n))\right)$. Note that (13) does not account for delayed communications. However, the modifications neccesary to account delays were presented in Section III-B and can be applied here as well. Without loss of generality, assume that Y_n has cardinality one for all $n \geq 0$. This is a useful trick from Abounadi et al., [1]. There is no loss of generality because the agents being updated at time n can be viewed as being updated sequentially. We thus have, $Y_n = \{\phi_n\}$ with $\phi_n \in \{1, \ldots, d\}$ for all $n \geq 0$. We may rewrite (13) as:

$$x_{n+1} = x_n + D_n\left[f(x_n) + \epsilon_n + M_{n+1}\right] + g_n, \tag{14}$$

where $g_n = \bigsqcap_{\mathcal{B},\mathcal{C}}\left(D_n\left[f(x_n) + \epsilon_n + M_{n+1}\right]\right) - \left(D_n\left[f(x_n) + \epsilon_n + M_{n+1}\right]\right)$. Define
$\mu_n := [I(\phi_n = 1), \ldots, I(\phi_n = d)]$, $\overline{a}(n,i) := a(\nu(n,i))$, $\hat{a}(n) := \overline{a}(n,\phi_n)$, $t(0) := 0$ and $t(n) := \sum_{m=0}^{n-1}\hat{a}(m)$ for $n \geq 1$.

Below we define the trajectories necessary for our analysis. It is suggested that the reader skip these definitions and refer back when required.

$$\begin{aligned} &\mu(t) := \mu_n \text{ for } t \in [t(n), t(n+1)), \\ &D_c(t) := D_n \text{ for } t \in [t(n), t(n+1)), \\ &X_c(t) := x_n \text{ for } t \in [t(n), t(n+1)), \\ &Y_c(t) := \mathcal{A}f(x_n) \text{ for } t \in [t(n), t(n+1)), \\ &G_c(t) := \sum_{m=0}^{n-1} g_m \text{ for } t \in [t(n), t(n+1)), \\ &\epsilon_c(t) := \mu_n\epsilon_n \text{ for } t \in [t(n), t(n+1)), \\ &X_l(t) := \begin{cases} x_n \text{ for } t = t(n) \\ \left(1 - \frac{t-t_n}{\hat{a}(n)}\right) X_l(t(n)) + \\ \left(\frac{t-t_n}{\hat{a}(n)}\right) X_l(t(n+1)) \text{ for } t \in [t(n), t(n+1)), \end{cases} \\ &W_l(t) := \begin{cases} \sum_{m=0}^{n-1} D_m M_{m+1} \text{ for } t = t(n) \\ \left(1 - \frac{t-t_n}{\hat{a}(n)}\right) W_l(t(n)) + \\ \left(\frac{t-t_n}{\hat{a}(n)}\right) W_l(t(n+1)) \text{ for } t \in [t(n), t(n+1)). \end{cases} \end{aligned}$$

We also define the following left-shifted trajectories:

$$\begin{aligned} &X_l^n(t) := X_l(t + t(n)), \\ &X_c^n(t) := X_c(t + t(n)), \\ &Y_c^n(t) := Y_c(t + t(n)), \\ &W_l^n(t) := W_l(t + t(n)), \\ &G_c^n(t) := G_c(t + t(n)) - G_c(t(n)), \\ &\epsilon_c^n(t) := \epsilon_c(t + t(n)), \\ &\mu^n(t) := \mu(t + t(n)), \\ &D_c^n(t) := D_c(t + t(n)). \end{aligned}$$

Note that $D_c^n(t) \leq 1$ and $\|\epsilon_c^n(t)\| \leq \epsilon$ for all $t \geq 0$ and $n \geq 0$. Hence $\{D_c^n([0,T]) \mid n \geq 0\}$ and $\{\epsilon_c^n([0,T]) \mid n \geq 0\}$ are relatively compact in $\mathbb{L}_2([0,T], \mathbb{R}^d)$.

One may view $\{X_l^n([0,T]) \mid n \geq 0\}$ and $\{G_c^n([0,T]) \mid n \geq 0\}$ as subsets of $D([0,T], \mathbb{R}^d)$ equipped with the Skorohod topology. In Lemma 3 below, we show that the aforementioned families of trajectories are relatively compact. As in *Lemma 2* of [25] we only need to show that these families are point-wise bounded and that any two discontinuities are separated by at least $\Delta > 0$.

Lemma 3. *$\{X_l^n([0,T]) \mid n \geq 0\}$ and $\{G_c^n([0,T]) \mid n \geq 0\}$ are relatively compact in $D([0,T], \mathbb{R}^d)$, equipped with the Skorohod topology.*

Proof. As stated earlier, we only need to show that the aforementioned families of trajectories are point-wise bounded and that any two discontinuities are separated by at least $\Delta > 0$. From $(S3)(i)$ we have that $\|M_{n+1}\| \leq D$ a.s. for all $n \geq 0$. Since f is Lipschitz continuous, $F(x) := f(x) + \overline{B}_\epsilon(0)$ is Marchaud. Clearly, $\mathcal{A}f(x_n) \in F(x_n)$ for all $n \geq 0$.

We have the following:

$$\sup_{x\in\overline{C}, y\in F(x)} \|y\| \leq C_1 \text{ for some } C_1 > 0$$

$$\implies \sup_{n\geq 0}\|\tilde{x}_{n+1} - x_n\| \leq \left(\sup_{n\geq 0} a(n)\right)(C_1 + D)$$

$$\implies \sup_{n\geq 0}\|g(n)\| \leq \sup_{n\geq 0}(\|\tilde{x}_{n+1}-x_n\|+d(x_n,\mathcal{B})) \leq C_2$$

for some $0 < C_2 < \infty$ that is independent of n.

Now that the point-wise boundedness property has been proven, it is left to show that any two discontinuities are separated by some $\Delta > 0$. Using arguments similar to the ones found in the proof of *Lemma 2* in [25], we can show that such a constant exists and is given by

$$\Delta = \frac{d}{2\left(D + \sup_{x\in\overline{C}, y\in F(x)}\|y\|\right)},$$

where d is the number of agents in the multi-agent system at hand. □

C. Overview of the strategy involved in stability analysis

Since T in Lemma 3 is arbitrary, $\{X_l^n([0,\infty)) \mid n \geq 0\}$ and $\{G_c^n([0,\infty)) \mid n \geq 0\}$ are relatively compact in $D([0,\infty),\mathbb{R}^d)$. It follows from $(S3)$ that $\{W_l^n([0,\infty)) \mid n \geq 0\}$ is relatively compact in $D([0,\infty)),\mathbb{R}^d)$, and that all limits equal the constant-0-function. If we consider a subsequence $\{m(n)\} \subset \{n\}$ such that $M_{m(n)}$ (Martingale noise) is convergent, then $X_l^{m(n)}([0,T])$ and $X_l^{m(n)}(0) + \int_0^t \left(\mu^{m(n)}(s)f(X_c^{m(n)}(s)) + \epsilon_c^{m(n)}(s)\right) ds + G_c^{m(n)}([0,t])$, $t \in [0,T]$ have identical limits.

Consider a subsequence $\{m(n)\}_{n\geq 0} \subseteq \mathbb{N}$ such that $\{\epsilon_c^{m(n)}([0,T]) \mid n \geq 0\}$ is weakly convergent in $\mathbb{L}_2([0,T],\mathbb{R}^d)$, and such that $\{X_l^{m(n)}([0,T]) \mid n \geq 0\}$ and $\{G_c^{m(n)}([0,T]) \mid n \geq 0\}$ are convergent in $D([0,T],\mathbb{R}^d)$. In addition, this subsequence satisfies the condition that $g_{m(n)-1} = 0$ for all $n \geq 0$. Now, let us suppose that the limit of $\{G_c^{m(n)}([0,T])\}_{n\geq 0}$ is the constant-0-function. Using arguments from Section III-A, we show that the limit of $\{X_l^{m(n)}([0,T]) \mid n \geq 0\}$ is given by:

$$X(0) + \int_0^t (\lambda(s)f(X(s)) + \epsilon(s))\, ds,$$

such that $X(0) \in \overline{\mathcal{C}}$. Hence, the projective scheme (13) tracks a solution to $\dot{x}(t) \in \lambda(t)f(x(t)) + \overline{B}_\epsilon(0)$, where $\lambda(\cdot)$ is some measurable matrix-valued process with only diagonal entries. If balanced step-sizes (see *Theorem 3.2* of Borkar [13]) are used, then (13) tracks a solution to $\dot{x}(t) \in 1/d\ f(x(t)) + \overline{B}_\epsilon(0)$. The asymptotic behaviors of $\dot{x}(t) = f(x(t))$ and $\dot{x}(t) = (1/d)\ f(x(t))$ are similar, *i.e.,* any solution trajectory of both o.d.e.'s, with starting points in $\overline{\mathcal{C}}$, will converge to the attractor $\hat{\mathbb{A}}$. Consequently, any solution trajectory of $\dot{x}(t) \in (1/d)\ f(x(t)) + \overline{B}_\epsilon(0)$ converges to $\mathbb{A}$, provided the starting point is inside $\mathcal{C}$. Recall that $\mathbb{A}$ is an attractor of $\dot{x}(t) \in f(x(t)) + \overline{B}_\epsilon(0)$ with fundamental neighborhood $\overline{\mathcal{N}}$ such that $\mathcal{C} \subset \overline{\mathcal{N}}$. In other words, the projective scheme (13) converges to $\mathbb{A}$ almost surely. Stability of the algorithm under consideration, (2), follows from $(S5)$.

To summarize, there are two important steps in proving stability:
(Step-1) Any limit of $\{X_l^n([0,T])\}_{n\geq 0}$ is of the form

$$X(t) = X(0) + \int_0^t (\mu^* f(X(s)) + \epsilon(s))\, ds + G(t) \text{ for } t \in [0,T],$$

where $\mu^* = diag(1/d,\ldots,1/d)$ and $X(0) \in \overline{C}$.
(Step-2) Show that any limit of $\{G_c^{m(n)}([0,T]) \mid n \geq 0\}$ is the constant 0 function, provided $g_{m(n)-1} = 0$ for all $n \geq 0$, see (14).

D. Stability Result

Define $K := \{n \mid g_{n-1} = 0\}$. The premise of the following two lemmas is that balanced step-sizes (of *Theorem 3.2*, [13]) are used.

Lemma 4. *Without loss of generality, let $\{\epsilon_c^n([0,T])\}_{n\in K}$ be (weakly) convergent in $\mathbb{L}_2([0,T],\mathbb{R}^d)$, with weak limit $\epsilon(\cdot)$. Also let $\{X_l^n([0,T])\}_{n\in K}$ and $\{G_c^n([0,T])\}_{n\in K}$ be convergent in $D([0,T],\mathbb{R}^d)$ as $n \to \infty$, with limits $X(\cdot)$ and $G(\cdot)$ respectively. Then, for $t \in [0,T]$,*

$$X_l^n(t) \to X(0) + \int_0^t (\mu^* f(X(s)) + \epsilon(s))\, ds + G(t). \quad (15)$$

Proof. Since $X_c^n(t) \to X(t)$ for $t \in [0,T]$, we get

$$\int_0^t \mu^* f(X_c^n(s))ds \to \int_0^t \mu^* f(X(s))ds.$$

Note that we have

$$X_l^n(t) = X_l^n(0) + \int_0^t diag(\mu_c^n(s))f(X_c^n(s))ds + W_l^n(t) + G_c^n(t) + \int_0^t \epsilon_c^n(s)ds.$$

Adding and subtracting $\int_0^t \mu^* f(X_c^n(s))ds$ in the above equation, yields:

$$X_l^n(t) = X_l^n(0) + \int_0^t \mu^* f(X_c^n(s))ds + W_l^n(t) + G_c^n(t) + \int_0^t \epsilon_c^n(s)ds + \eta_n(t), \quad (16)$$

where
$\eta_n(t) = \int_0^t diag(\mu_c^n(s))f(X_c^n(s))ds - \int_0^t \mu^* f(X_c^n(s))ds.$

From Assumption $(S3)$ it follows that $\lim_{n\to\infty}\sup_{t\in[0,T]}\|W_l^n(t)\| = 0$. Suppose we show that $\lim_{n\to\infty}\sup_{t\in[0,T]}\|\eta_n(t)\| = 0$, then we can conclude that (16) converges to

$$X(t) = X(0) + \int_0^t \mu^* f(X(s))ds + G(t) + \int_0^t \epsilon(s)ds \text{ as } n \to \infty.$$

Recall that $\epsilon(\cdot)$ is the weak limit of $\{\epsilon_c^n([0,T])\}_{n\in K}$. Thus, it is left to show that $\lim_{n\to\infty}\sup_{t\in[0,T]}\|\eta_n(t)\| = 0$. The proof of this is along the lines of the proof of *Lemma 3.5* in Abounadi et al., [1]. □

Lemma 5. *The $G(\cdot)$ of Lemma 4 is the constant 0 function. As a consequence the projective scheme (14) converges to $\mathbb{A}$.*

Proof. For a proof of this lemma the reader is referred to the proof of *Lemma 3* of [25]. □

Before we state another main result of this paper, we define an internally chain transitive set below:

> ***[Internally chain transitive set]*** *Given $\dot{x}(t) \in H(x(t))$, an invariant set $M \subset \mathbb{R}^d$ is said to be internally chain transitive if M is compact and, for every $x, y \in M$, $\epsilon > 0$ and $T > 0$, we have the following: There exist n and $\Phi^1, \ldots, \Phi^n$ that are n solutions to the differential inclusion $\dot{x}(t) \in H(x(t))$, points $x_1(= x), \ldots, x_{n+1}(= y) \in M$ and n real numbers $t_1, t_2, \ldots, t_n$ greater than T such that: $\Phi^i_{t_i}(x_i) \in N^\epsilon(x_{i+1})$ and $\Phi^i_{[0,t_i]}(x_i) \subset M$ for $1 \le i \le n$. The sequence $(x_1(= x), \ldots, x_{n+1}(= y))$ is called an (ϵ, T) chain in M from x to y.*

Theorem 2. *Under $(A1)$-$(A3)$, $(S1)$-$(S5)$, and when the step-sizes are balanced, iteration (2) is stable ($\sup_{n\ge 0}\|x_n\| < \infty$ a.s.) and converges to a closed connected internally chain transitive invariant set associated with $\dot{x}(t) \in \mu^* f(x(t)) + \overline{B}_\epsilon(0)$.*

Proof. The reader may recall that $\mu^* = diag(1/d, \ldots, 1/d)$. It follows from Lemma 5 that the associated projective iterates, say $\{\hat{x}_n\}_{n\ge 0}$, corresponding to $\{x_n\}_{n\ge 0}$ converge to $\mathbb{A}$. In other words, there exists N, possibly sample path dependent, such that $\hat{x}_n \in \mathcal{C}$ for $n \ge N$. It follows from $(A5)$ that $\sup_{n\ge N}\|x_n\| < \infty$ a.s.

The second part of the statement directly follows from Theorem 1. □

E. Stability assuming $(A5)$ instead of $(S3)$

The statement of Theorem 2 is true when the weaker $(A5)$ is assumed instead of the stricter $(S3)$. The details involved (in a related setup) can be found in *Section 6* of [25]. We merely present the steps involved without any proofs and refer the reader to *Section 6* of [25] for details.

Assumption $(S3)$ was used in Lemma 3 to show that any two discontinuities of $\{X_l^n([0,T]) \mid n \ge 0\}$ and $\{G_c^n([0,T]) \mid n \ge 0\}$ are at least Δ apart. An important step in proving the aforementioned claim with $(A5)$ replacing $(S3)$ is the following auxiliary lemma.

Lemma 6 (*Lemma 5*, [25])**.** *Let $\{t_{m(n)}, t_{l(n)}\}_{n\ge 0}$ be such that $t_{l(n)} > t_{m(n)}$, $t_{m(n+1)} > t_{l(n)}$ and $\lim_{n\to\infty}\left(t_{l(n)} - t_{m(n)}\right) = 0$. Fix an arbitrary $c > 0$ and consider the following:*

$$\psi_n := \left\| \sum_{i=m(n)}^{l(n)-1} a(i) M_{i+1} \right\|.$$

Then $P(\{\psi_n > c\} \ i.o.) = 0$ within the context of the projective scheme given by (14), where $i.o.$ is short for infinitely often.

Colloquially, Lemma 6 states the following: After a lapse of considerable time, there are no significant contributions to jumps in $X_l^n(\cdot)$ or $G_c^n(\cdot)$ from the Martingale difference noise sequence within shrinking time intervals. If we are unable to find a separating Δ, then it can be shown that Lemma 6 is contradicted. Therefore, Theorem 2 is true under the standard, weak assumption on noise imposed by $(A5)$. As a consequence, the following modification of Theorem 2 is immediate.

Theorem 3. *Suppose that $(A1)$-$(A3)$, $(A5)$ and $(S1), (S2), (S4)$, $(S5)$ hold and that balanced step-sizes are used. Then the iteration given by (2) is bounded almost surely (stable) and converges to a closed connected internally chain transitive invariant set associated with $\dot{x}(t) \in 1/df(x(t)) + \overline{B}_\epsilon(0)$.*

V. APPLICATIONS

As stated earlier, sample trajectory based algorithms using value iteration or policy gradients are popular reinforcement learning algorithms. In this section, we consider their asynchronous approximation counterparts. First we adapt noisy value iteration to the multi-agent setting, following which we adapt the policy gradient algorithm. We consider the following adaptation of value iteration for the multi-agent setting:

$$J_{n+1}(i) = J_n(i) + a(\nu(n,i))I(i \in Y_n) \left[(\mathcal{A}T)_i(J_{n-\tau_{1i}(n)}(1), \ldots, J_{n-\tau_{di}(n)}(d)) + M_{n+1}(i)\right], n \ge 0. \quad (17)$$

In the above equation, (i) T is the Bellman operator. (ii) $1 \le i \le d$ is the agent index, and there are d agents in the system. (iii) $J_n := (J_n(1), \ldots, J_n(d))$ is an estimate of the optimal cost-to-go vector at time-step n.

The remaining terms are as defined for (2). Recursion (17) is the *asynchronous approximate value iteration* (A2VI). If the optimal cost-to-go vector associated with agent-i is $J^*(i)$, then $J^* = (J^*(1), \ldots, J^*(d))$ is the optimal cost-to-go vector associated with the d-agent system. The objective is to find J^* in an "asynchronous" manner.

The **policy gradient** algorithm is another important reinforcement learning approach, see [29]. This method assumes a parameterization θ of the policy space π. Finding an optimal policy amounts to finding a $\hat{\theta}$ that locally minimizes a given performance objective such as the Value function. Adapting

the policy gradient algorithm to the multi-agent setting, we get:

$$\theta_{n+1}(i) = \theta_n(i) - a(\nu(i,n))I\{i \in Y_n\} \\ [(\mathcal{A}\nabla_\theta \pi)_i(\theta_{n-\tau_{1i}(n)}(1), \ldots, \theta_{n-\tau_{di}(n)}(d))\nabla_\pi V_\pi(i) \\ + M_{n+1}(i)]. \quad (18)$$

In the above equation, θ is the parameter associated with policy π, $\mathcal{A}$ is the approximation of the policy function gradient, $V_\pi(i)$ is the agent-i value function associated with policy π. Note that required objective-gradient is $\nabla_\theta V_\pi$, which is calculated via the chain rule $\nabla_\theta \pi \times \nabla_\pi V_\pi$. Also note that the gradient with respect to θ is approximated using $\mathcal{A}$. This scheme is useful when the policy parameterization involves non-differentiable neural network architectures such as convolutional neural networks.

In the following sections, the theory hitherto developed is used to present a complete analysis of A2VI and A2PG.

A. Asynchronous approximate value iteration (A2VI)

Consider A2VI given by (17). Let $\epsilon_n = (\mathcal{A}T)J_n - TJ_n$ be the approximation error at stage n, cf. (3). The approximation operator $\mathcal{A}$ could be a deep neural network, or some other function approximator.

Remark 2. *We do not distinguish between stochastic shortest path (no discounting) and infinite horizon discounted cost problems. The definition of the Bellman operator T changes appropriately based on the problem at hand [7].*

The following assumptions are natural:

(AV1) The Bellman operator T is contractive with respect to some weighted max-norm, $\|\cdot\|_\nu$, *i.e.,* $\|Tx - Ty\|_\nu \leq \alpha\|x-y\|_\nu$ for some $0 < \alpha < 1$. Given $\nu = (\nu_1, \ldots, \nu_d)$ such that $\nu_i > 0$ for $1 \leq i \leq d$, the weighted max-norm of any $x = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d$ is given by: $\|x\|_\nu := \max\left\{\frac{|x_i|}{\nu_i} \mid 1 \leq i \leq d\right\}$.

(AV2) T has a unique fixed point J^* and J^* is the unique globally asymptotically stable equilibrium point of $\dot{J}(t) = TJ(t) - J(t)$.

(AV3) $\limsup_{n\to\infty} \|\epsilon_n\|_\nu \leq \epsilon$ for some fixed $\epsilon > 0$.

Given $x \in \mathbb{R}^d$ we make the following simple observations:
(i) $\|x\|_\nu \leq \frac{1}{\min_i \nu_i}\|x\|$.
(ii) $\|x\| \leq \frac{d}{\min_i \nu_i}\|x\|_\nu$.
The following claim is an immediate consequence of these observations.

Claim 1. *T is Lipschitz continuous with some Lipschitz constant $0 < L < \infty$.*

The only difference between (17) and (2) is that in (17) the approximation errors are bounded in the weighted max-norm sense. It is worth noting that the errors could be more generally bounded in the weighted p-norm ($\|\cdot\|_{\omega,p}$) sense [6]. It can be easily shown that $C_l\|x\|_\nu \leq \|x\|_{\omega,p} \leq C_u\|x\|_\nu$, for some $C_l, C_u > 0$, $x \in \mathbb{R}^d$. Hence it is sufficient to work with errors that are bounded in the weighted max-norm sense. In $(AV3)$ we assume $\limsup_{n\to\infty} \|\epsilon_n\|_\nu \leq \epsilon$ a.s., while in $(A1)$ we assume $\limsup_{n\to\infty} \|\epsilon_n\| \leq \epsilon$ a.s. Since $B^\epsilon := \{y \mid \|y\|_\nu \leq \epsilon\}$ is a convex compact subset of $\mathbb{R}^d$ (see *Lemma 7.2* of [25]), the analyses presented in Sections III and IV carry forward verbatim, with B^ϵ replacing $B_\epsilon(0)$.

It follows directly from $(AV2)$ that $(S4a)$ is satisfied. If we show that (17) also satisfies $(S5)$, then we may conclude that the iterates are stable and convergent. For this purpose, we compare the iterates $\{J_n\}_{n\geq 0}$, from (17), to their projective counterparts $\{\hat{J}_n\}_{n\geq 0}$. We can show that $\hat{J}_n \to \mathbb{A}$, where $\mathbb{A}$ is an attractor of $\dot{J}(t) \in 1/d(TJ(t) - J(t)) + B^\epsilon$, contained within a small neighborhood of J^*. This neighborhood is dependent on the approximation errors. Since $\hat{J}_n \to \mathbb{A}$, $\exists N$, possibly sample path dependent, such that $\hat{J}_n \in \mathcal{C}$ for all $n \geq N$. Following the arguments presented in the proof of *Theorem 3* in [25] we can show that

$$\|J_n - \hat{J}_n\|_\nu \leq \max\left\{\|J_N - \hat{J}_N\|_\nu, \left(\frac{2\epsilon}{1-\alpha}\right)\right\},$$

where α is the "contraction constant" associated with the Bellman operator T. In other words, we get that (17) satisfies $(S5)$. The following result is immediate.

Theorem 4. *Suppose that $(AV1)$-$(AV3)$, $(A5)$, $(S1)$ and $(S2)$ are satisfied, and that balanced step-sizes are used. Then (17) is stable and converges to some point in $\{J \mid \|TJ - J\|_\nu \leq d\epsilon\}$, where ϵ is the norm-bound on the approximation errors.*

Proof. From the above discussion, it is clear that A2VI is bounded a.s. (stable). Since balanced step-sizes are used, to study the long-term behavior of A2VI one needs to study $\dot{J}(t) \in \mu^*((TJ)(t) - J(t)) + B^\epsilon$, where $\mu^* = diag(1/d, \ldots, 1/d)$. It follows from *Theorem 2* of *Chapter 6* in [2] that any solution to the aforementioned DI will converge to an equilibrium point of $T(\cdot) + B^{d\epsilon}$, where $B^{d\epsilon} := \{dx \mid x \in B^\epsilon\}$. This is because $\dot{J}(t) \in \mu^*((TJ)(t) - J(t) + B^{d\epsilon})$ and $\dot{J}(t) \in TJ(t) - J(t) + B^{d\epsilon}$ are qualitatively similar and only differ in scale. The equilibrium points of $T + B^{d\epsilon}$ are given by $\{J \mid \|TJ - J\|_\nu \leq d\epsilon\}$. For more details the reader is referred to *Section 7* of [25]. □

We have shown that A2VI is stable as long as the approximation errors are asymptotically bounded. We do not need to distinguish between biased and unbiased errors. Further, we show that A2VI converges to a fixed point of a scaling of the perturbed Bellman operator $1/d\ TJ + B^\epsilon$.

B. Asynchronous approximate policy gradient iteration (A2PG)

The policy gradient iteration method is an important reinforcement learning algorithm developed by Sutton *et al.*, in 2000 [29]. This method relies on a parameterization of the policy space, say $\pi(\theta)$. This parameterization is typically through the use of a deep neural network. Once a parameterization is

[6] Given $\omega = (\omega_1, \ldots, \omega_d)$ such that $\omega_i > 0$ for $1 \leq i \leq d$, and $p \geq 1$, the weighted p-norm of any $x \in \mathbb{R}^d$ is defined by: $\|x\|_{\omega,p} := \left(\sum_{i=1}^d |\omega_i x_i|^p\right)^{1/p}$.

determined, one seeks out a local minimizer $\hat{\theta}$ in the parameter space, in order to find the optimal policy. However, there are several situations wherein one either cannot calculate or does not wish to calculate the exact gradient $\nabla_\theta \pi(\theta_n)$ at every stage. This could be due to the use of a non-differentiable activation function or it could be a consequence of using gradient estimators such as $SPSA$-C [24] (simultaneous perturbation stochastic approximation with a constant sensitivity parameter) or other finite difference methods. In such cases, one would have to deal with a policy gradient scheme with non-diminishing approximation errors. In the present work, we are interested in policy gradient methods within the setting of large-scale distributed systems. A general form of approximate policy gradient methods which satisfy all these conditions is given below:

$$\begin{aligned}&\theta_{n+1}(i) = \theta_n(i) - a(\nu(i,n))I\{i \in Y_n\}\\ &[(\mathcal{A}\nabla_\theta \pi)_i(\theta_{n-\tau_{1i}(n)}(1), \ldots, \theta_{n-\tau_{di}(n)}(d))\nabla_\pi V_\pi(i)\\ &+ M_{n+1}(i)].\end{aligned} \tag{19}$$

We call the above scheme as asynchronous approximate policy gradient iteration or A2PG. As in Section V-A, we can impose natural conditions on the gradient ($\nabla_\theta \pi(\cdot)$), the noise and other parameters of (19). Suppose the approximation errors are asymptotically bounded, then our foregoing analysis can be used to show that the iterates converge to a neighborhood of some local minimizer $\hat{\theta}$. Further, the size of this neighborhood is a function of the approximation errors. For further details on the relationship between the neighborhood and approximation errors, albeit in a centralized set-up, the reader is referred to [24].

C. Verifiability of assumption $(S5)$

In this section, we address the verifiability of assumption $(S5)$. We do not discuss other assumptions, since they deal with the objective function, step-sizes or noise, in a manner that is standard to literature. However, to ensure $(S5)$, one needs to compare the algorithm iterates with a projective scheme. Further, the experimenter is typically uninterested in the projective scheme itself. In this section, we show that $(S5)$ is satisfied for fixed point finding algorithms such as A2VI, provided the objective function is non-expansive.

Recall A2VI and its projective counterpart:

$$\begin{aligned}&J_{n+1} = J_n + a(n)D_n\left[TJ_n - J_n + \epsilon_n\right],\\ &\hat{J}_{n+1} \in \textstyle\bigsqcap_{\mathcal{B},\mathcal{C}}\left(\hat{J}_n + a(n)D_n\left[T\hat{J}_n - J_n + \hat{\epsilon}_n\right]\right).\end{aligned} \tag{20}$$

Unlike in Section V-A, we assume here that T is non-expansive with respect to some norm p, i.e., $p(Tx - Ty) \leq p(x-y)$, for all x, y. It follows from Lemma 5 that the projective scheme converges to $\mathbb{A}$ almost surely. In other words, there exists a sample path dependent N such that $\{\hat{J}_n\}_{n\geq N} \subseteq \mathcal{C}$ a.s. Further, $\hat{J}_{n+1} = \hat{J}_n + a(n)D_n\left[T\hat{J}_n + \hat{\epsilon}_n\right]$ for all $n \geq N$. For $n \geq N$, first, we take the difference between the two iterations in (20). Then, we take the norms on both sides, to get the following:

$$\begin{aligned}p(J_{n+1} - \hat{J}_{n+1}) \leq\ &(1 - a(n))p(J_n - \hat{J}_n)\\ &+ a(n)p(TJ_n - T\hat{J}_n) + a(n)p(\epsilon_n - \hat{\epsilon}_n).\end{aligned}$$

Since T is non-expansive we obtain:

$$p(J_{n+1} - \hat{J}_{n+1}) \leq p(J_n - \hat{J}_n) + a(n)p(\epsilon_n - \hat{\epsilon}_n).$$

For $k \geq 1$, we have:

$$p(J_{N+k} - \hat{J}_{N+k}) \leq p(J_N - \hat{J}_N) + \sum_{n=N}^{N+k-1} a(n)p(\epsilon_n - \hat{\epsilon}_n).$$

As long as $p(\epsilon_n - \hat{\epsilon}_n) \in o(a(n))$, we get:

$$p(J_{N+k} - \hat{J}_{N+k}) \leq p(J_N - \hat{J}_N) + \sum_{n=0}^{\infty} a(n)^2 < \infty.$$

It may be noted that many important RL and MDP algorithms such as Q-learning and Value Iteration are fixed point finding algorithms. In [1], the objective function of Q-learning is shown to be non-expansive. To summarize, the above set of arguments can be used to verify (S5) for approximate asynchronous fixed point finding algorithms with non-expansive objective functions.

D. A brief discussion on real-world applications

The hitherto presented theory has been motivated by the need to understand multi-agent RL and optimization algorithms, in particular, to provide behavioral guarantees when applied to solve real-world problems. In this section, we consider two applications: (a) data-driven vehicular platoon control and (b) cumulative consensus in delayed communication networks. Now, we present bird's-eye views of iterative solutions (to the two problems) based on the main iteration considered herein, (2). As a consequence, the developed theory may be used to draw required conclusions.

[Vehicular platooning] Here, the broad goal is to organize moving vehicles in a rigid string with a fixed inter-car distance and constant velocity [14], [15], [31]. Scalable controllers for the platooning problem are often distributed, with each vehicle acting autonomously. They coordinate with other vehicles via a wireless network, to exchange position and velocity information. Each vehicle uses the latest available information to take local control actions (adjust its speed, position, etc.). While traditional solutions assume a model for vehicular movement, we are interested in a model-free solution using A2VI. In particular, to derive a *distributed model-free data-driven control policy for platooning*.

Platooning can be viewed as a *distributed sequential decision making problem*, that can be associated with a Markov Decision Process (MDP). Solving the "discounted cost problem" for this MDP, in a distributed manner, yields the required "distributed platooning policy". Communication is important for platooning. The theory developed here can be used to derive the following requirement on communication: at any time-step, there is a non-zero probability of successful transmission, although there could be delays. This property ensures that our assumptions on delays are satisfied, see [27] for details. Without loss of generality, the lead vehicle trains a neural network to approximate the Bellman operator, i.e., to minimize the following loss function:

$$\left(\mathcal{A}TJ_{n-\tau} - r_{n-\tau} - \gamma J_{n-\tau}\right)^T\left(\mathcal{A}TJ_{n-\tau} - r_{n-\tau} - \gamma J_{n-\tau}\right), \tag{21}$$

where (i) $\mathcal{A}T$ represents the parameterization of the Bellman operator using a neural network, (ii) $r_{n-\tau}$ represents the reward vector for all vehicles, (iii) $J_{n-\tau} \equiv (J_{n-\tau}(1), \ldots, J_{n-\tau}(d))$ is the current approximation of value function of all d vehicles, available with the lead vehicle, and (iv) τ is the random variable associated with communication delays. The reader may note that we have abused the notation slightly and used a single delay random variable τ, merely for illustration. In reality, vehicles closer to the lead vehicle experience smaller delays as opposed to the ones at the end of the platoon. In addition, each vehicle executes the local A2VI iteration given by (17). *The reader may also note that each car is associated with its own set of local states. The different actions taken in states, across the vehicles, are related to each other through the Bellman operator that operates on the "appended" cost-to-go vector associated with all the vehicles.* At any point in time, the value function update, (21), is only done for states encountered by the platoon at that time. Hence, the Bellman operator is approximated in an asynchronous manner. Our theory can be used to derive stability conditions for the algorithm during training. It can also be used to understand the platooning policy obtained from training, see, e.g., Theorem 4.

Our theory can be used to conclude that the vehicles can be successfully trained to find a near-optimal distributed platooning policy, J^∞, provided the lead vehicle approximates the Bellman operator well. The optimality of the policy found is directly related to the approximation capability of the lead vehicle. Further, the quality of communications has a direct effect on the training time of the vehicles, required to find the platooning policy. The limit J^∞ found after training, is used instead of the optimal cost-to-go vector J^*, to derive the optimal platooning policy. Specifically, $J^\infty(i)$ is used by vehicle-i to derive its optimal policy. Note that $J^\infty(i)$ is itself a vector and the optimal cost-to-go value associated with a vehicle-i state, s, is denoted by $J^\infty(i,s)$. When at state s, the policy of vehicle-i is to pick the action $\underset{a\in\mathcal{A}}{\text{argmax}}\ \mathbb{E}_{s'\sim\mathcal{E}}\left[r(s,a)+\gamma J^\infty(i,s')\right]$. This is in accordance to the classical value iteration paradigm.

[Cummulative consensus with unbounded delays] Here, autonomous agents in a D-agent system cooperate to solve the following optimization problem:

$$\underset{x\in\mathbb{R}^d}{\text{argmin}} \sum_{i=1}^{D} f_i(x), \text{ where,} \tag{22}$$

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

In Section 6 of [27], we present an algorithm that solves (22). This algorithm (Algorithm 2 in [27]) is designed and analyzed using the theory developed here. Let us quickly summarize the ideas of this algorithm. An agent with the maximum computational capacity is first chosen to be the arbiter, who is responsible for bulk of the computations involved in solving the consensus problem. All the other agents perform local computations to support the arbiter. In particular, the arbiter performs the following update:

$$x_{n+1} = x_n - a(\nu(n,1))I(1 \in Y_n)\sum_{i=1}^{D} g_i(\tau),$$

where $g_i(\tau)$ is the descent direction obtained from agent-i, after a delay of τ. As before, τ could vary with the agents. The key requirement on communication, again, is that the probabilities of transmission successes are positive. In [27], we show that the network topology may vary widely over time, and that the agents need not synchronize. These network conditions, obtained using the ideas developed in this paper, are among the weakest in literature. The reader is referred to [27] for details on the algorithm and empirical results.

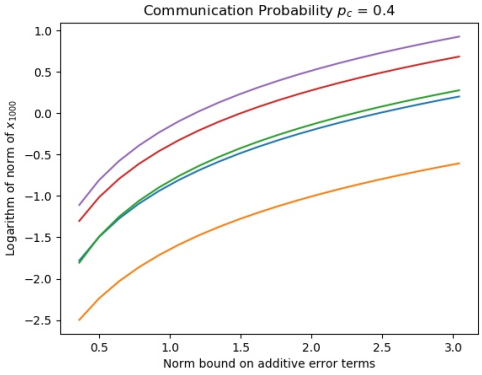

Fig. 1: Five random sample runs with $p_c = 0.4$. $\|(\epsilon_1, \epsilon_2)\|$ is plotted along x-axis and $\log(\|x_{1000}\|)$ is plotted along y-axis.

E. Experimental results

In this section [7], we consider an asynchronous algorithm (given by eq. (2)) to find the minimum of $F : \mathbb{R}^d \to \mathbb{R}^d$, where $d \geq 2$. The function F is defined as $F(x_1, \ldots, x_d) := (F_1(x_1, \ldots, x_d), \ldots, F_d(x_1, \ldots, x_d))$, where $F_1, \ldots F_d : \mathbb{R}^d \to \mathbb{R}$.

[Experimental set-up] For better exposition, we consider an iteration in dimension 2, i.e., $d = 2$. The function F is defined as follows: $F_1(x) := \frac{1}{2}(x^T Ax)(1)$, $F_2(x) := \frac{1}{2}(x^T Bx)(2)$ and $F(x) := (F_1(x), F_2(x))$. The matrices A and B are randomly constructed positive definite matrices of dimension 2×2. A random error vector of norm less than $\epsilon > 0$ is added to the gradient at every step. Each component of this error vector is independent and uniformly distributed in $[0, \epsilon/2]$. It may be noted that $\nabla_x F_1(x) = Ax$ and $\nabla_x F_2(x) = Bx$ for $x \in \mathbb{R}^d$.

Agent-1 runs the following:

$$x_{n+1}(1) = x_n(1) - a(n)\left[A\begin{bmatrix} x_n(1) \\ x_{n-\tau_{2,1}}(2)\end{bmatrix}(1) + \epsilon_1\right],$$

while agent-2 runs the following:

$$x_{n+1}(2) = x_n(2) - a(n)\left[B\begin{bmatrix} x_{n-\tau_{2,1}}(1) \\ x_n(2)\end{bmatrix}(2) + \epsilon_2\right].$$

[7]The reader is referred to [27] for detailed empirical studies on practical applications of the theory from this paper to cumulative consensus.

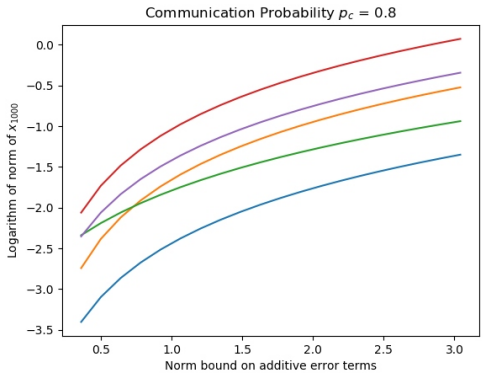

Fig. 2: Five random sample runs with $p_c = 0.8$. $\|(\epsilon_1, \epsilon_2)\|$ is plotted along x-axis and $\log(\|x_{1000}\|)$ is plotted along y-axis.

The above distributed algorithm was run for 1000 iterations using the step-size seqeunce $\{1/(n+10)\}_{n=1}^{1000}$. Since the matrices A and B are positive definite, we expect the limit to be the origin.

At every step the two agents exchange (state) information with probability p_c, and with probability $1 - p_c$ the agents use the available old (state) information. In other words, p_c represents the communication probability in our experiments. Note that we have used symmetric delays for simplicity. The experiments can be easily repeated with asymmetric delays. A direct consequence of this network setup is that the communication delays are random with unbounded support.

Results from the experiments are summarized in Figures 1 and 2. In both the figures, $\|(\epsilon_1, \epsilon_2)\|$ is plotted along x-axis and $\log(\|x_{1000}\|)$ is plotted along y-axis. Thus, any point in the plot represents $(\|(\epsilon_1, \epsilon_2)\|, \log(\|x_{1000}\|))$. Each figure has five differently colored line graphs to represent the five sample runs of the algorithm. For each sample run, the parameters $(x_1(1), x_1(2))$ (initial point) and the matrices A and B are randomly chosen, and the norm bound on additive errors (ϵ_1, ϵ_2) is varied from 0.2 to 3 in steps of 0.1. Fig. 1 illustrates all the experiments with $p_c = 0.4$, and Fig. 2 illustrates all the experiments with $p_c = 0.8$.

In Figures 1 and 2, the agents exchange data 40% and 80% of the time, respectively. It can be seen that the algorithm converged farther from the origin when the additive errors are larger. When $p_c = 0.8$, the algorithm converged to a point close to the origin, even for large additive errors, as compared to $p_c = 0.4$. **The experiments seem to suggest that frequent communications should be used to counter the effect of bias inducing additive errors.** We conclude this section by noting that the experimental set-up considered here is simple yet illustrative. It demonstrates the ability of our algorithm to deal with unbounded communication delays and non-vanishing additive errors. The reader is referred to [27] for detailed empirical studies on practical applications of the theory from this paper.

VI. Summary of our contributions and conclusions

In this paper, we considered a natural extension of asynchronous stochastic approximation algorithms that accommodates the use of function approximators. For this purpose, we considered asynchronous stochastic approximations with asymptotically bounded, and possibly bias inducing, approximation errors. The assumptions and the analyses presented are motivated by the need to understand the current crop of deep reinforcement learning algorithms. We are particularly interested in these algorithms when used within the setting of multi-agent learning and control.

Our framework allows for complete asynchronicity in that each agent is guided by its own local clock. Although the agents are fully asynchronous, we require that the agents are updated, roughly, the same fraction of times, in the long run. Our framework can be used to analyze asynchronous approximate value iteration (A2VI). A2VI is an adaptation of regular value iteration with noise to the setting of large-scale multi-agent learning and control. Here, we showed that A2VI converges to a fixed point of the perturbed Bellman operator when balanced step-sizes are used. We also established a relationship between these fixed points and the approximation errors. Note that the use of function approximators required us to consider the perturbed Bellman operator. We further analyzed a similar adaptation, A2PG, of the classical policy gradient iteration to the multi-agent setting. We briefly discussed how A2PG converges to a small neighborhood of local minima of the parameterized policy function. Again, this neighborhood is directly related to the approximation errors.

An important consequence of our theory is the following: stability of the aforementioned algorithms remains unaffected when the approximation errors are asymptotically bounded, and add bias in the algorithm. Since a function approximator (eg. DNN) is continuously trained, it is reasonable to expect the errors to diminish asymptotically, even though they may not vanish completely. It is worth noting that ours is one of the first theoretical results that can be used to understand the long-term behavior of deep reinforcement learning algorithms within the setting of multi-agent learning and control.

In the future, we would like to make a two-fold extension to our analysis: (i) Allow for multiple timescales and (ii) allow for objective functions that are driven by controlled Markov processes. This will help us analyze other popular algorithms such as Deep Q-Network and deep deterministic policy gradient (a popular actor-critic algorithm). When implementing DeepRL algorithms in an online setting, the learning rate is generally fixed. To this end, we would also wish to explore one and two timescale algorithms with constant step-sizes and function approximations.

References

[1] J. Abounadi, D.P. Bertsekas, and V. Borkar. Stochastic approximation for nonexpansive maps: Application to q-learning algorithms. *SIAM Journal on Control and Optimization*, 41(1):1–22, 2002.

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

[3] M. Benaïm. A dynamical system approach to stochastic approximations. *SIAM J. Control Optim.*, 34(2):437–472, 1996.

[4] M. Benaïm and M. W. Hirsch. Asymptotic pseudotrajectories and chain recurrent flows, with applications. *J. Dynam. Differential Equations*, 8:141–176, 1996.

[5] M. Benaïm, J. Hofbauer, and S. Sorin. Stochastic approximations and differential inclusions. *SIAM Journal on Control and Optimization*, pages 328–348, 2005.

[6] M. Benaïm, J. Hofbauer, and S. Sorin. Stochastic approximations and differential inclusions, part ii: Applications. *Mathematics of Operations Research*, 31(4):673–695, 2006.

[7] D.P. Bertsekas and J.N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1st edition, 1996.

[8] S. Bhatnagar. The Borkar-Meyn theorem for asynchronous stochastic approximations. *Systems & Control Letters*, 60(7):472–478, 2011.

[9] P. Bianchi, W. Hachem, and A. Salim. Constant step stochastic approximations involving differential inclusions: Stability, long-run convergence and applications. *Stochastics*, 91(2):288–320, 2019.

[10] P. Billingsley. *Convergence of probability measures*. John Wiley & Sons, 2013.

[11] V. S. Borkar. *Stochastic Approximation: A Dynamical Systems Viewpoint*. Cambridge University Press, 2008.

[12] V. S. Borkar and S.P. Meyn. The O.D.E. method for convergence of stochastic approximation and reinforcement learning. *SIAM J. Control Optim*, 38:447–469, 1999.

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

[14] F. Dressler, F. Klingler, M. Segata, and R.L. Cigno. Cooperative driving and the tactile internet. *Proceedings of the IEEE*, 107(2):436–446, 2018.

[15] G. Guo and Q. Wang. Fuel-efficient en route speed planning and tracking control of truck platoons. *IEEE Transactions on Intelligent Transportation Systems*, 20(8):3091–3103, 2018.

[16] H. Kushner and G.G. Yin. *Stochastic Approximation and Recursive Algorithms and Applications*. Springer, 2003.

[17] A.S. Leong, A. Ramaswamy, D.E. Quevedo, H. Karl, and L. Shi. Deep reinforcement learning for wireless sensor scheduling in cyber-physical systems. *Automatica (to appear)*, 2019.

[18] Y. Li. Deep reinforcement learning: An overview. *arXiv preprint arXiv:1701.07274*, 2017.

[19] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, A. Graves, M. Riedmiller, A.K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

[20] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, pages 1928–1937, 2016.

[21] R. Munos. Error bounds for approximate policy iteration. In *ICML*, volume 3, pages 560–567, 2003.

[22] R. Munos. Error bounds for approximate value iteration. In *Proceedings of the National Conference on Artificial Intelligence*, volume 20, page 1006, 2005.

[23] S. Perkins and D.S. Leslie. Asynchronous stochastic approximation with differential inclusions. *Stochastic Systems*, 2(2):409–446, 2013.

[24] A. Ramaswamy and S. Bhatnagar. Analysis of gradient descent methods with non-diminishing, bounded errors. *IEEE Transactions on Automatic Control*, 2017. doi:10.1109/TAC.2017.2744598.

[25] A. Ramaswamy and S. Bhatnagar. Conditions for stability and convergence of set-valued stochastic approximations: Applications to approximate value and fixed point iterations. *arXiv preprint arXiv:1709.04673*, 2017.

[26] A. Ramaswamy and S. Bhatnagar. A generalization of the Borkar-Meyn theorem for stochastic recursive inclusions. *Mathematics of Operations Research*, 42(3):648–661, 2017.

[27] A. Ramaswamy, A. Redder, and D.E. Quevedo. Optimization over time-varying networks with unbounded delays. *arXiv preprint arXiv:1912.07055*, 2019.

[28] H. Robbins and S. Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[29] R.S. Sutton, D. A. McAllester, S.P. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.

[30] A. Tamar, Y. Wu, G. Thomas, S. Levine, and P. Abbeel. Value iteration networks. In *Advances in Neural Information Processing Systems*, pages 2154–2162, 2016.

[31] S. Wen and G. Guo. Sampled-data control for connected vehicles with markovian switching topologies and communication delay. *IEEE Transactions on Intelligent Transportation Systems*, 2019.

Arunselvan Ramaswamy is currently a *Junior Research Head* at the *Software Innovation Lab*, Heinz Nixdorf Institute and Dept. of Computer Science, Paderborn University, Germany. He heads research in *Data Science and Artificial Intelligence*. He is also affiliated with the *Intelligent Systems and Machine Learning Group* at Paderborn University.

He completed MSc. (Engg.) in 2012 and Ph.D. in 2017 from the Department of Computer Science and Automation, Indian Institute of Science (IISc), India. *His Ph.D. thesis received special commendation for outstanding research, from IISc*. He also held a Post-Doctoral position at Paderborn University. His primary areas of research include stochastic approximation algorithms, dynamical systems, reinforcement learning, deep learning, stochastic optimal control and stochastic processes.

Shalabh Bhatnagar received his Ph.D from the Indian Institute of Science in 1998. He held postdoctoral positions at the Institute for Systems Research, University of Maryland, College Park, USA, during 1997 to 2000 and at the Vrije Universiteit, Amsterdam, Netherlands, during 2000-2001. He was a Visiting Faculty Member at the Indian Institute of Technology, Delhi in 2001. He joined the Department of Computer Science and Automation, Indian Institute of Science as a Faculty Member in December 2001, where he is a Professor since 2011.

Dr. Bhatnagar's research interests lie in stochastic approximation algorithms, specifically algorithms for optimisation and control of random dynamical systems. He is also interested in applications in the areas of communication and wireless networks, vehicular traffic control as well as autonomous systems. He is a Fellow of the Indian National Science Academy, the Indian Academy of Sciences and the Indian National Academy of Engineering, and was a Senior Associate of the International Centre for Theoretical Physics, Trieste, Italy. He has also served as an Associate Editor for the IEEE Transactions on Automation Science and Engineering.

Daniel E. Quevedo (S'97–M'05–SM'14) received Ingeniero Civil Electrónico and M.Sc. degrees from Universidad Técnica Federico Santa María, Valparaíso, Chile, in 2000, and in 2005 the Ph.D. degree from the University of Newcastle, Australia. He is Professor at the School of Electrical Engineering and Robotics, Queensland University of Technology (QUT), in Australia. Before joining QUT, he established and led the Chair in Automatic Control at Paderborn University, Germany. In 2003 he received the IEEE Conference on Decision and Control Best Student Paper Award and was also a finalist in 2002. He is co-recipient of the 2018 IEEE Transactions on Automatic Control George S. Axelby Outstanding Paper Award.

Prof. Quevedo currently serves as Associate Editor for the *IEEE Control Systems Magazine* and in the Editorial Board of the *International Journal of Robust and Nonlinear Control*. From 2015–2018 he was Chair of the IEEE Control Systems Society *Technical Committee on Networks & Communication Systems*. His research interests are in networked control systems and cyber-physical systems.