

TECHNICAL RESEARCH REPORT

Randomized Difference Two-Timescale Simultaneous
Perturbation Stochastic Approximation Algorithms for
Simulation Optimization of Hidden Markov Models

*by Shalabh Bhatnagar, Michael C. Fu,
Steven I. Marcus, Shashank Bhatnagar*

T.R. 2000-13



ISR develops, applies and teaches advanced methodologies of design and analysis to solve complex, hierarchical, heterogeneous and dynamic problems of engineering technology and systems for industry and government.

ISR is a permanent institute of the University of Maryland, within the Glenn L. Martin Institute of Technology/A. James Clark School of Engineering. It is a National Science Foundation Engineering Research Center.

Web site <http://www.isr.umd.edu>

Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE 2000		2. REPORT TYPE		3. DATES COVERED 00-00-2000 to 00-00-2000	
4. TITLE AND SUBTITLE Randomized Di erence Two-Timescale Simultaneous Perturbation Stochastic Approximation Algorithms for Simulation Optimization of Hidden Markov Models				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) University of Maryland, College Park,Institute for Systems Research,College Park,MD,20742				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT We propose two nite di erence two-timescale simultaneous perturbation stochastic approximation (SPSA) algorithms for simulation optimization of hidden Markov models. Stability and convergence of both the algorithms is proved. Numerical experiments on a queueing model with high dimensional parameter vectors demonstrate orders of magnitude faster convergence using these algorithms over related (N + 1)-Simulation nite di erence analogues and another Two-Simulation nite di erence algorithm that updates in cycles.					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT Same as Report (SAR)	18. NUMBER OF PAGES 27	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

Randomized Difference Two-Timescale Simultaneous Perturbation Stochastic Approximation Algorithms for Simulation Optimization of Hidden Markov Models *

Shalabh Bhatnagar
Institute for Systems Research
University of Maryland, College Park, MD 20742
shalabh@isr.umd.edu, fax: (301) 314-9920

Michael C. Fu
The Robert H. Smith School of Business
and Institute for Systems Research
University of Maryland, College Park, MD 20742
mfu@rhsmith.umd.edu, fax: (301) 314-9157

Steven I. Marcus
Department of Electrical and Computer Engineering
and Institute for Systems Research
University of Maryland, College Park, MD 20742
marcus@isr.umd.edu, fax: (301) 314-9920

Shashank Bhatnagar
Department of Physics
University of Addis Ababa, P. O. Box 21926, Code 1000
Addis Ababa, Ethiopia
shashank_bhatnagar@hotmail.com, fax: +251 (1) 550655

June 1, 2000

Abstract

We propose two finite difference two-timescale simultaneous perturbation stochastic approximation (SPSA) algorithms for simulation optimization of hidden Markov models. Stability and convergence of both the algorithms is proved. Numerical experiments on a queueing model with high dimensional parameter vectors demonstrate orders of magnitude faster convergence using these algorithms over related $(N + 1)$ -Simulation finite difference analogues and another Two-Simulation finite difference algorithm that updates in cycles.

Key Words Simulation optimization, two-timescale SPSA algorithms, hidden Markov models.

*This research was supported in part by the NSF under Grant DMI-9713720, by the Semiconductor Research Corporation under Grant 97-FJ-491 and by DoD contract Number MDA90497C3015.

1 Introduction

A popular approach to simulation optimization of discrete event systems with continuous-valued parameters is based on stochastic approximation [1], [20], [22], [27]. Gradient descent stochastic approximation algorithms are typically used to perform function optimization in cases where the function to be optimized is difficult to compute analytically [16]. Several stochastic approximation schemes that have been used for optimization of long run average performance measures suffer from the drawback that they require data aggregation (for averaging) over regeneration epochs [13], [15]. These epochs can be very infrequent (particularly for large systems/networks), making the scheme extremely slow in practice. In [12], an algorithm that updates the parameter after a fixed number of customers is presented and convergence proved. However, the algorithms in [12], [13] and [15] all require the availability of direct gradient estimates, and are all based on infinitesimal perturbation analysis [19]. In [23] and [24], various stochastic approximation algorithms governed by finite difference estimates (as well as direct gradient estimates) were considered for optimizing a steady-state performance measure with respect to a scalar parameter in a single-server queue.

In [3], [4] and [5], a more general setting for vector parameters and long-run average performance measures is considered, in which the parameter is updated at deterministic instants that are obtained using two timescales; a faster timescale at which the system evolves, and a slower timescale at which the parameter is updated. Specifically, in [4], the parameter is updated at deterministically increasing time instants that are in turn obtained using two timescales (see also [2]). On the other hand, in [5], the parameter is updated at every instant using coupled iterations that are governed by different timescales. However as with any other forward finite difference scheme [23], [24], these schemes also require $(N + 1)$ parallel simulations for an N -vector parameter. Thus, if the parameter dimension (N) is large, the corresponding number of simulations required to obtain the optimum parameter is large as well. A proposed alternative in [4] uses only two parallel simulations at any instant by moving the algorithm in bigger loops or cycles. This, however, results in slow convergence.

Spall [28] proposed a stochastic approximation technique that requires only two simulations for a parameter vector of any dimension and updates all parameter components at every instant. This technique came to be known as simultaneous perturbation stochastic approximation (SPSA), since it simultaneously perturbs the various parameter components randomly, most commonly by using independent and identically distributed (i.i.d.), symmetric Bernoulli random variables in the two simulations, and uses the estimates thus obtained for updating the parameter. It has been applied in various contexts; for instance, see [17] for an application of SPSA to optimization of discrete event systems. Most of the work in discrete event systems (except [17]) is on low-dimensional problems (cf. [4], [24],[23], [12],[21]). In [28] and [22], a general idea for high-dimensional problems is proposed.

Motivated by all of the above considerations, in this paper we develop SPSA variants (that we call SPSA-1 and SPSA-2) of the two-timescale algorithms of [4] and [5] (respectively) for optimizing high-dimensional parameters in hidden Markov models. The algorithm SPSA-1 uses only two parallel simulations and updates the parameter at increasing time instants as in [4]. The algorithm SPSA-2 also uses only two parallel simulations but has the added advantage (over the algorithm in [5]) that it allows for data aggregation over a fixed number of instants in between two successive

updates of the parameter for better performance. Moreover, the algorithm in [5] was only for ordinary Markov processes and not hidden Markov models (which is a more general setting) [14] considered here. We prove the convergence of both of these schemes and numerically demonstrate the algorithms on a feedback queueing network with high-dimensional parameters. These schemes are found to converge orders of magnitude faster than their $(N + 1)$ -Simulation analogues in [4] and [5], and also the Two-Simulation algorithm of [4] that moves in cycles.

Hidden Markov models arise in several queueing and stochastic control applications. To illustrate a very simple instance of a hidden Markov model, consider an $M/G/1$ queue. Let $\{q_n\}$ represent the queue length process observed at customer arrival epochs. Similarly, let $\{r_n\}$ represent the sequence of residual service times of the customers in service at these epochs. Then the joint process $\{(q_n, r_n)\}$ is Markov. In most real life applications only the process $\{q_n\}$ is observed whereas $\{r_n\}$ is not. Thus in this example, $\{q_n\}$ represents a hidden Markov model.

In [7], we applied algorithms SPSA-1 and a special case of SPSA-2 for the closed loop feedback control of available bit rate (ABR) service in asynchronous transfer mode (ATM) networks, by considering parameterized feedback policies. A finite state setting was considered there and as a result there was no problem with stability of the schemes. We develop these algorithms in this paper in the framework of hidden Markov models with an unbounded state space, and therefore stability issues are explicitly addressed. The convergence of SPSA-1 in the finite state setting of [7] was proven in [6]. The convergence analysis of SPSA-2 in any setting has not been shown earlier.

The rest of the paper is organized as follows: In the next section, we formulate the optimization problem, present the assumptions on the system and give a result on tightness of the stationary measures for the hidden Markov model. In Section 3, we present our SPSA algorithms (SPSA-1 and SPSA-2) and compare their performance with their corresponding analogues in [4] and [5]. In Sections 4 and 5, we present the detailed convergence analyses for both algorithms. In Section 6, numerical results comparing the SPSA algorithms with those of [4] and [5] for a simple queueing system are presented. Finally, Section 7 provides the concluding remarks.

2 The Optimization Problem

The process that we seek to optimize is an $\mathcal{R}^d$ - valued parameterized (with parameter $\theta \in \mathcal{R}^N$) hidden Markov model (HMM) represented by $\{Y(j), j \geq 0\}$ and is given by the set of coupled iterations

$$X(j+1) = F(X(j), Y(j), \xi(j), \theta), \quad (2.1)$$

$$Y(j+1) = G(X(j), Y(j), \eta(j), \theta), \quad (2.2)$$

$j \geq 0$. Here the state process $\{X(j)\}$ is $\mathcal{R}^q$ - valued and is unobserved or hidden. Further, $\{\xi(j)\}$, $\{\eta(j)\}$ are i.i.d. sequences in $\mathcal{R}^l$ and $\mathcal{R}^s$, respectively, and are mutually independent of one another. The maps $F : \mathcal{R}^q \times \mathcal{R}^d \times \mathcal{R}^l \times \mathcal{R}^N \rightarrow \mathcal{R}^q$ and $G : \mathcal{R}^q \times \mathcal{R}^d \times \mathcal{R}^s \times \mathcal{R}^N \rightarrow \mathcal{R}^d$ are measurable. Also, $\theta \triangleq (\theta_1, \dots, \theta_N)^T \in \mathcal{R}^N$ represents the parameter to be tuned in order to minimize a certain cost function $J(\theta)$ (defined below). We will assume that θ takes values in a compact set $C \subset \mathcal{R}^N$, and that C is of the form $C \triangleq \prod_{i=1}^N [\theta_{i,\min}, \theta_{i,\max}]$. We assume that the joint process $\{(X(j), Y(j))\}$ is ergodic Markov for every fixed θ , and has stationary distribution $\mu_\theta(dx, dy) \in \mathcal{P}(\mathcal{R}^{q+d})$. Let $\nu_\theta(dy)$ be the marginal of this stationary distribution in $\mathcal{P}(\mathcal{R}^d)$. Here, $\mathcal{P}(\dots)$ represents the space

of all probability measures on ‘...’. Also, let $p_\theta(x, y; dx', dy')$ represent the transition kernel of $\{(X(j), Y(j))\}$. Let $h : \mathcal{R}^d \rightarrow \mathcal{R}$ be a given bounded and continuous cost function. Our aim then is to find a θ in the set C that minimizes the average cost

$$J(\theta) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{j=0}^{N-1} h(Y(j)).$$

Since $\{(X(j), Y(j))\}$ is ergodic Markov (for fixed θ), the above limit exists and

$$J(\theta) = \int h(y) \nu_\theta(dy).$$

Remark As shown in [8], very general classes of random processes have an HMM representation if one is permitted time nonhomogeneity viz., replace F and G in (2.1)-(2.2) above with F_j , G_j , $j \geq 0$. Also, $\{\xi(j)\}$ and $\{\eta(j)\}$ can be taken to be uniformly distributed on $[0, 1]$ without loss of generality.

We now proceed with the rest of the model. As mentioned earlier, the parameter θ in (2.1)-(2.2) has to be tuned in order to minimize $J(\theta)$. Let $\tilde{\theta}_n \in C$ represent the parameter value at instant n . Thus in particular, we will consider the following dynamics:

$$X(j+1) = F(X(j), Y(j), \xi(j), \tilde{\theta}_j), \quad (2.3)$$

$$Y(j+1) = G(X(j), Y(j), \eta(j), \tilde{\theta}_j), \quad (2.4)$$

$j \geq 0$. Let $\mathcal{G}_n \triangleq \sigma(X(j), Y(j), \xi(j), \eta(j), \tilde{\theta}_j, j \leq n)$, $n \geq 0$, represent the natural-filtration generated by $\{X(j), Y(j), \xi(j), \eta(j), \tilde{\theta}_j, j \geq 0\}$. Note that since $\{\tilde{\theta}_j\}$ could be any arbitrarily defined parameter sequence, $\{(X(j), Y(j))\}$ defined simply by (2.3)-(2.4) would in general not be Markov. We thus force $\{(X(j), Y(j))\}$ to be Markov by assuming that any sequence $\{\tilde{\theta}_j, j \geq 0\}$ satisfies

$$P(X(n+1) \in A, Y(n+1) \in B \mid \mathcal{G}_n) = p_{\tilde{\theta}_n}(X(n), Y(n); A, B), \quad (2.5)$$

for any A, B , Borel in $\mathcal{R}^q$ and $\mathcal{R}^d$ respectively. We call any such $\{\tilde{\theta}_n\}$ satisfying (2.5), an **M-Sequence**.

Next, we list the following assumptions on our system. Assumptions (A1), (A2) and (A3) are needed to prove convergence of our first algorithm SPSA-1, while (A1), (A2') and (A3) are required for convergence of algorithm SPSA-2. Both algorithms SPSA-1 and SPSA-2 are presented in Section 3.

Assumptions

(A1) The average cost $J(\theta)$ is continuously differentiable.

(A2) The map $(\theta, x, y) \in \mathcal{R}^{N+q+d} \rightarrow p_\theta(x, y; dx', dy') \in \mathcal{P}(\mathcal{R}^{q+d})$ is continuous.

(A2') For any $h \in C(\mathcal{R}^{q+d})$ vanishing at ∞ ,

$$\lim_{\|(x,y)\| \rightarrow \infty} \int p_\theta(x, y; dx', dy') h(x', y') = 0$$

uniformly over $\theta \in C$.

(A3) Liapunov Stability Condition: There exist nonnegative $V \in C(\mathcal{R}^{q+d})$, $K \subset \mathcal{R}^{q+d}$ compact and $\epsilon_0 > 0$ such that under any M-Sequence $\{\tilde{\theta}_n\}$,

1. $\lim_{\|(x,y)\| \rightarrow \infty} V(x,y) = \infty$,
2. $\sup_n E[V(X(n), Y(n))^2] < \infty$,
3. $E[V(X(n+1), Y(n+1)) \mid \mathcal{G}_n] \leq V(X(n), Y(n)) - \epsilon_0$,
whenever $(X(n), Y(n)) \notin K$, $n \geq 0$.

Note that a sufficient condition for (A1) to be satisfied is that the parameterized stationary distribution $\mu_\theta(dx, dy)$ of the ergodic Markov process $\{(X(n), Y(n))\}$ be continuously differentiable in the parameter θ . In [29], certain sufficiency conditions for showing the latter have been given. Assumptions (A2) and (A2') are technical conditions that are satisfied routinely by most systems [4], [5]. Finally, Assumption (A3) is required to ensure that the system remains stable, and is a standard assumption. In [3], the algorithms of [4] and [5] were applied to a closed loop feedback control problem in asynchronous transfer mode (ATM) networks and all the above assumptions were verified for the system (also see [6] for verification of Assumptions (A1), (A2) and (A3) for a similar system).

Recall that a sequence $\{\hat{\mu}_n\}$ of probability measures on S is tight if for each $\epsilon > 0$, there exists a compact set $K_\epsilon \subset S$ such that $\hat{\mu}_n(K_\epsilon) > 1 - \epsilon$ for all n . The following lemma shows that the sets of all parameterized stationary distributions $\{\mu_\theta, \theta \in C\}$ and their marginals $\{\nu_\theta, \theta \in C\}$ are tight and crucially uses Assumption (A3) for its proof. This result is required later in the analysis.

Lemma 2.1 $\{\mu_\theta, \theta \in C\}$ (resp., $\{\nu_\theta, \theta \in C\}$) is compact in $\mathcal{P}(\mathcal{R}^{q+d})$ (resp., $\mathcal{P}(\mathcal{R}^d)$) and the map $\theta \rightarrow \mu_\theta$ (resp., ν_θ) is continuous.

Proof Follows in exactly the same manner as Lemma 2.1 of [4]. □

3 Two-Timescale SPSA Algorithms

In this section, we shall present our two-timescale SPSA algorithms and compare their performance with corresponding two-timescale finite difference algorithms in [4] and [5]. In order to put things in proper perspective and to clearly bring out the advantages of our SPSA algorithms, we shall first begin with the $(N+1)$ -Simulation finite difference stochastic approximation algorithm of [4] that we refer to as $(N+1)$ -Simulation FDSA-1 and its corresponding Two-Simulation alternative (proposed in that paper) referred here as Two-Simulation FDSA-1. We shall then present our first SPSA algorithm (SPSA-1). Later, we shall illustrate the $(N+1)$ -Simulation finite difference algorithm of [5] that we refer to as $(N+1)$ -Simulation FDSA-2, followed by its generalized SPSA version (SPSA-2). We shall then briefly compare the performance of all these algorithms and argue the reasons for the superior performance of SPSA-1 and SPSA-2 over the algorithms in [4] and [5].

3.1 The Algorithms

The algorithms presented here are called two-timescale algorithms since they are governed with two step-size sequences (or timescales) $\{a(n)\}$ and $\{b(n)\}$ defined below. Let $\delta > 0$ be a fixed small constant. Let $\pi_i(x) \triangleq \min(\max(\theta_{i,\min}, x), \theta_{i,\max})$, $i = 1, \dots, N$, denote the point closest to $x \in \mathcal{R}$ in the interval $[\theta_{i,\min}, \theta_{i,\max}] \subset \mathcal{R}$, $i = 1, \dots, N$, and $\pi(\theta)$ be defined by the vector $\pi(\theta) \triangleq (\pi_1(\theta_1), \dots, \pi_N(\theta_N))^T$. Then $\pi(\theta)$ is a projection of $\theta \in \mathcal{R}^N$ onto the set C . Define

sequences $\{a(n)\}$ and $\{b(n)\}$ as follows: $a(0) = b(0) = 1$, $a(i) = i^{-1}$, $b(i) = i^{-\alpha}$, $i \geq 1$, and with $\frac{1}{2} < \alpha < 1$. Then clearly,

$$\frac{a(n+1)}{a(n)}, \frac{b(n+1)}{b(n)} \rightarrow 1, \text{ as } n \rightarrow \infty, \quad (3.1)$$

$$\sum_n a(n) = \sum_n b(n) = \infty, \sum_n a(n)^2, \sum_n b(n)^2 < \infty, a(n) = o(b(n)). \quad (3.2)$$

Define $\{n_m, m \geq 0\}$ as follows: $n_0 = 1$ and $n_{m+1} = \min\{j > n_m \mid \sum_{i=n_m+1}^j a(i) \geq b(m)\}$, $m \geq 1$. Then $\{n_m\}$ represents a deterministically increasing sequence of points. In SPSA-1 (as also $(N+1)$ -Simulation FDSA-1 and Two-Simulation FDSA-1), $\{n_m\}$ defines the parameter update instants of the algorithm.

Note that any finite difference stochastic approximation scheme ordinarily requires $(N+1)$ parallel simulations for an N -vector parameter. The two-timescale stochastic approximation algorithm in [4] for an N -vector parameter (which we call $(N+1)$ -Simulation FDSA-1) is thus as follows:

$(N+1)$ -Simulation FDSA-1

The first simulation corresponds to $\{(X(j), Y(j))\}$ and is governed by $\{\tilde{\theta}_j\}$ that is in turn defined by $\tilde{\theta}_j = \theta(m)$, for $n_m \leq j < n_{m+1}$. The remaining N parallel simulations are represented by $\{(X_i(j), Y_i(j))\}$, $i = 1, \dots, N$, and are respectively governed by $\{\hat{\theta}^i(j)\}$, $i = 1, \dots, N$, with $\hat{\theta}^i(j) = \theta(m) + \delta e_i$, for $n_m \leq j < n_{m+1}$, and where e_i is the unit vector with 1 in the i th direction. The algorithm then is as follows: For $i = 1, \dots, N$,

$$\theta_i(m+1) = \pi_i \left(\theta_i(m) + \sum_{j=n_m+1}^{n_{m+1}} a(j) \left(\frac{h(Y(j)) - h(Y_i(j))}{\delta} \right) \right). \quad (3.3)$$

It is clear that one would require $N+1$ parallel simulations using this algorithm. An alternative scheme was proposed in [4] that uses only two parallel simulations at any instant. This, it achieves by moving the algorithm in ‘cycles’, in each of which only one component is updated. This scheme that we call Two-Simulation FDSA-1 is as follows:

Two-Simulation FDSA-1

The first simulation here corresponds to $\{(X(j), Y(j))\}$ and is governed by $\{\tilde{\theta}_j\}$ where the parameter $\tilde{\theta}_j$ is the N -vector $\tilde{\theta}_j = (\tilde{\theta}_{j,1}, \dots, \tilde{\theta}_{j,N})^T$ with $\tilde{\theta}_{j,i} = \theta_i(m)$, for $n_{Nm+i-1} \leq j < n_{Nm+i}$, $i = 1, \dots, N$, $m \geq 0$ (with N being the dimensionality of the parameter vector), where $\tilde{\theta}_{j,i}$ (resp. $\theta_i(m)$) is the i th component of $\tilde{\theta}_j$ (resp. $\theta(m)$). The second simulation is now represented as $\{(\bar{X}(j), \bar{Y}(j))\}$ and is governed by $\{\hat{\theta}_j\}$ defined by $\hat{\theta}_j = \theta(m) + \delta e_i$, for $n_{Nm+i-1} \leq j < n_{Nm+i}$, $i = 1, \dots, N$, $m \geq 0$. The algorithm is as follows: For $i = 1, \dots, N$,

$$\theta_i(m+1) = \pi_i \left(\theta_i(m) + \sum_{j=n_{Nm+i-1}+1}^{n_{Nm+i}} a(j) \left(\frac{h(Y(j)) - h(\bar{Y}(j))}{\delta} \right) \right). \quad (3.4)$$

Thus using this scheme, the whole parameter is updated once every n_{Nm} steps instead of the n_m steps required for one update using the $(N+1)$ -Simulation FDSA-1 version (3.3) of it. Next, we present our first randomized difference SPSA algorithm (SPSA-1).

Let for any $i \geq 0$, $\Delta(i) \in \mathcal{R}^N$ be a vector of mutually independent and mean zero random variables $\{\Delta_{1,i}, \dots, \Delta_{N,i}\}$, (viz., $\Delta(i) \triangleq (\Delta_{1,i}, \dots, \Delta_{N,i})^T$) taking values in a compact set $E \subset \mathcal{R}^N$

and having a common distribution. We assume that these random variables satisfy Condition (B) below.

Condition (B) There exists a constant $\bar{K} < \infty$, such that for any $l \geq 0$, and $i \in \{1, \dots, N\}$, $E \left[\Delta_{l,i}^{-2} \right] \leq \bar{K}$.

Further, we assume that $\{\Delta(i)\}$ is a mutually independent sequence with $\Delta(i)$ independent of $\sigma(\theta(l), l \leq i)$, the filtration generated by the sequence of parameter updates. Condition (B) is a standard condition in SPSA algorithms. Minor variants of this condition are for instance available in [28], [11]. Note that distributions like Gaussian and Uniform are precluded while using Condition (B). An important consequence of $E \left[\Delta_{l,i}^{-2} \right] < \infty$ is that $P(\Delta_{l,i} = 0) = 0$.

We now proceed with our first algorithm, SPSA-1, wherein we use only two parallel simulations and update all parameter components every n_m instants by perturbing all of these simultaneously along random directions in the two simulations.

SPSA-1

Consider two parallel simulations $\{(X^k(j), Y^k(j))\}$, $k = 1, 2$, respectively governed by $\{\tilde{\theta}_j^k\}$, $k = 1, 2$ as follows: For the process $\{(X^1(j), Y^1(j))\}$, we define $\tilde{\theta}_j^1 = \theta(m) - \delta \Delta(m)$, for $n_m < j \leq n_{m+1}$, $m \geq 0$. The parameter sequence $\{\tilde{\theta}_j^2\}$ for $\{(X^2(j), Y^2(j))\}$ is similarly defined by $\tilde{\theta}_j^2 = \theta(m) + \delta \Delta(m)$, for $n_m < j \leq n_{m+1}$, $m \geq 0$. In the above, $\theta(m) \triangleq (\theta_1(m), \dots, \theta_N(m))^T$ is the value of the parameter update that is governed by the following recursion equations. For $i = 1, \dots, N$,

$$\theta_i(m+1) = \pi_i \left(\theta_i(m) + \sum_{j=n_m+1}^{n_{m+1}} a(j) \left(\frac{h(Y_j^1) - h(Y_j^2)}{2\delta \Delta_{m,i}} \right) \right), \quad (3.5)$$

$m \geq 0$. It will be shown in the proof of Theorem 4.1 that the sequence $\{n_m\}$ is an exponentially increasing sequence. Thus algorithms $(N+1)$ -Simulation FDSA-1, Two-Simulation FDSA-1 and SPSA-1 update parameters at exponentially increasing time instants. Hence, using these algorithms, subsequent parameter updates become less frequent as time progresses. The algorithm of [5] (cf. $(N+1)$ -Simulation FDSA-2 below) on the other hand, uses coupled iterations with two-timescales and updates the whole parameter at every instant (even though it requires $N+1$ parallel simulations at any instant). We present this algorithm next.

$(N+1)$ -Simulation FDSA-2

Let $\{(X(n), Y(n))\}$ be governed by $\{\theta(n)\}$ (where $\theta(n) \triangleq (\theta_1(n), \dots, \theta_N(n))^T$ is the n th update of parameter θ) and which is updated according to equations (3.6) below. Let us also define N additional parallel simulations as follows: For $i = 1, \dots, N$, let $\{(X^i(n), Y^i(n))\}$ be governed by $\{\theta(n) + \delta e_i\}$, where e_i is the unit vector with 1 in the i th direction. In the following, the sequences $\{Z(n)\}$ and $\{Z^i(n)\}$, $i = 1, \dots, N$, perform weighted averages of the cost function values and are defined as in the last two equations in (3.6) below. Let $Z(0) = Z^i(0) = 0$, $i = 1, \dots, N$. Then, for $i = 1, \dots, N$,

$$\begin{aligned} \theta_i(n+1) &= \pi_i \left(\theta_i(n) + a(n) \left[\frac{Z(n) - Z^i(n)}{\delta} \right] \right) \\ Z(n+1) &= Z(n) + b(n)(h(Y(n)) - Z(n)) \\ Z^i(n+1) &= Z^i(n) + b(n)(h(Y^i(n)) - Z^i(n)) \end{aligned} \quad (3.6)$$

It is clear that one requires $N + 1$ parallel simulations in this manner. Finally, we present our next SPSA algorithm (SPSA-2) which requires only two parallel simulations as in SPSA-1 but which also allows for data aggregation over a fixed number L of epochs in between two successive parameter updates for better performance. We will explain the last part in detail later.

SPSA-2

Let $\{(X^-(l), Y^-(l))\}$ and $\{(X^+(l), Y^+(l))\}$ be the two parallel simulations. These depend on parameter sequences $\{\theta(n) - \delta\Delta(n)\}$ and $\{\theta(n) + \delta\Delta(n)\}$ respectively in the manner explained below: Let $L \geq 1$ be a given fixed integer. We extract double sequences $\{(X_m^-(n), Y_m^-(n))\}$ and $\{(X_m^+(n), Y_m^+(n))\}$, $n \geq 0$, $m = 0, 1, \dots, L - 1$, from the two parallel simulations in the following manner. Write l as $l = nL + m$, where $n \geq 0$ and $m \in \{0, 1, \dots, L - 1\}$. Now, set $X_m^-(n) \triangleq X^-(nL + m)$ and $Y_m^-(n) \triangleq Y^-(nL + m)$. Similarly, $X_m^+(n) \triangleq X^+(nL + m)$ and $Y_m^+(n) \triangleq Y^+(nL + m)$ respectively. Now, for $m = 0, 1, \dots, L - 1$, $(X_m^-(n), Y_m^-(n))$ is governed by the parameter $\theta(n) - \delta\Delta(n)$. Similarly, for $m = 0, 1, \dots, L - 1$, $(X_m^+(n), Y_m^+(n))$ is governed by the parameter $\theta(n) + \delta\Delta(n)$. We also define two double sequences $\{Z_m^-(n)\}$ and $\{Z_m^+(n)\}$, $n \geq 0$, $m = 0, 1, \dots, L - 1$, in recursions (3.7) for averaging the cost function. Let $Z_0^-(0) = Z_1^-(0) = \dots = Z_L^-(0) = 0$ and $Z_0^+(0) = Z_1^+(0) = \dots = Z_L^+(0) = 0$. Then, for $i = 1, \dots, N$, $n \geq 0$,

$$\theta_i(n + 1) = \pi_i \left(\theta_i(n) + a(n) \left[\frac{Z_L^-(n) - Z_L^+(n)}{2\delta\Delta_{n,i}} \right] \right),$$

where, for $m = 0, 1, \dots, L - 1$,

$$\begin{aligned} Z_{m+1}^-(n + 1) &= Z_m^-(n + 1) + b(n)(h(Y_m^-(n + 1)) - Z_m^-(n + 1)), \\ Z_{m+1}^+(n + 1) &= Z_m^+(n + 1) + b(n)(h(Y_m^+(n + 1)) - Z_m^+(n + 1)), \end{aligned} \quad (3.7)$$

with $Z_0^-(n + 1) = Z_L^-(n)$ and $Z_0^+(n + 1) = Z_L^+(n)$. Note again that one requires only two parallel simulations in this manner as opposed to $N + 1$ earlier.

Remark Note that for $L = 1$, the algorithm SPSA-2 is simply as follows: Let $\{(X^-(n), Y^-(n))\}$ and $\{(X^+(n), Y^+(n))\}$ be the two parallel simulations respectively governed by $\{\theta(n) - \delta\Delta(n)\}$ and $\{\theta(n) + \delta\Delta(n)\}$. Then, for $i = 1, \dots, N$,

$$\begin{aligned} \theta_i(n + 1) &= \pi_i \left(\theta_i(n) + a(n) \left[\frac{Z^-(n) - Z^+(n)}{2\delta\Delta_{n,i}} \right] \right), \\ Z^-(n + 1) &= Z^-(n) + b(n)(h(Y^-(n)) - Z^-(n)), \\ Z^+(n + 1) &= Z^+(n) + b(n)(h(Y^+(n)) - Z^+(n)), \end{aligned} \quad (3.8)$$

with $Z^-(0) = Z^+(0) = 0$. We observed in the numerical experiments that the algorithm (3.8) (corresponding to $L = 1$) did not exhibit good performance when the parameter dimension is high. This could probably be due to the fact that in the latter case, the system does not adapt as quickly to the new parameter update before it changes again. By selecting $L > 1$, one can effectively take care of this problem by holding the parameter fixed for L instants, thus giving the system sufficient time to adapt to the new parameter update. The choice of L is completely arbitrary though. In the numerical experiments for instance, where we consider the parameter vectors to be 10 and 40-dimensional respectively, the value of L is chosen as 100.

3.2 Comparison of Algorithms

We can classify the five algorithms broadly into two categories - those that update the parameter over time instants (or their multiples) of increasing separation n_m , $m \geq 1$, and those that update the parameter at regular intervals. Let us first consider the algorithms in the first category. These comprise $(N + 1)$ -Simulation FDSA-1, Two-Simulation FDSA-1 and SPSA-1. As already stated earlier, $(N + 1)$ -Simulation FDSA-1 requires $N + 1$ parallel simulations at every instant but updates the whole parameter vector once every n_m instants. On the other hand, Two-Simulation FDSA-1 uses only two simulations and updates the parameter in cycles of n_{Nm} , $m \geq 1$ instants, where N is the parameter dimension. SPSA-1, however, updates the full parameter every n_m instants and still requires only two parallel simulations for doing so. Thus SPSA-1 has the combined advantages of $(N + 1)$ -Simulation FDSA-1 and Two-Simulation FDSA-1. Moreover SPSA-1 tracks trajectories of the ordinary differential equation (o.d.e.) (4.7) as does $(N + 1)$ -Simulation FDSA-1. It was shown in [4] that Two-Simulation FDSA-1 tracks trajectories of an o.d.e. that is similar to (4.7) but with a factor $1/N$ multiplying its RHS. This factor essentially serves to slow down the rate of convergence of the algorithm (3.4).

Our next set of algorithms viz., $(N + 1)$ -Simulation FDSA-2 and SPSA-2 update parameters after every fixed number of instants. In particular, $(N + 1)$ -Simulation FDSA-2 updates the full parameter vector every instant while requiring $N + 1$ parallel simulations for the same. SPSA-2, on the other hand, requires only two parallel simulations and updates the parameter after every fixed number 'L' of instants. This number ($L \geq 1$) is chosen arbitrarily. SPSA-2 thus allows for data aggregation in between successive parameter update epochs (for better performance) while requiring only two parallel simulations at any instant. It will be shown in Section 5 that SPSA-2 tracks the trajectories of the o.d.e. (4.7) on the slower timescale. A similar result was shown in [5] for $(N + 1)$ -Simulation FDSA-2. In Section 6, we shall consider a simple queueing network with parameter vectors of dimensions 10 and 40 respectively. We found that both SPSA-1 and SPSA-2 exhibit significantly superior performance than the algorithms in [4] and [5] mentioned here. We shall consider in Section 6 the following step-size sequences $\{a(n)\}$ and $\{b(n)\}$: $a(0) = b(0) = 1$, $a(n) = 1/n$ and $b(n) = 1/n^{2/3}$, $n \geq 1$. For these sequences, $n_0 = 1$, $n_{50} \approx 2.5 \times 10^4$, $n_{100} \approx 4.3 \times 10^5$ etc. Thus, a possible disadvantage in using SPSA-1 is for large systems/networks that require several updates of the parameter before convergence is achieved, since in using this scheme, successive parameter updates are held fixed over intervals of increasing sizes and thus the parameter is updated less often as time progresses. This is not the case with algorithm SPSA-2 where we hold the parameter fixed only for a fixed number L of epochs before updating it. This intuition is also confirmed in Section 6 where we show numerical experiments with parameters of dimensions 10 and 40 respectively. We observed that when the parameter dimension is 10, SPSA-1 outperforms SPSA-2. However, when the same is increased to 40, it is SPSA-2 that performs better than SPSA-1.

As already observed, the algorithms SPSA-1 and SPSA-2 are computationally superior than their corresponding variants. This is however achieved by generating N i.i.d. random variables $\Delta_{n,1}, \dots, \Delta_{n,N}$, that satisfy Condition (B). In particular, one could select these to be i.i.d., Bernoulli distributed (as we do in our numerical experiments in Section 6) viz., $\Delta_{n,i} = \pm 1$, w.p. $1/2$, $i = 1, \dots, N$. It will become clear in the convergence analysis in the next two sections that it is these randomizations and the particular form of the gradient estimates that are primarily responsible for both of these schemes using only two parallel simulations at any instant as against

$N + 1$. The presence of $\Delta_{n,i}$ in the denominator of the gradient estimate term on the RHS in the update equation for the i th component $\theta_i(n)$ of $\theta(n)$ in (3.5), and the first equation in (3.7), essentially ensures that the update for the i th component occurs only in the i th gradient direction (with the rest of them averaging to zero). Generating N i.i.d., Bernoulli random variables or in general those satisfying Condition (B), is far more computationally simpler than generating N parallel simulations, the latter requires in particular simulating N independent parallel systems. It will be shown in the next two sections that algorithms (3.5) and (3.7) asymptotically track the stable points of the o.d.e. (4.7).

Finally, in [6], the algorithm SPSA-1 was analyzed in the context of rate based feedback flow control in available bit rate (ABR) service in asynchronous transfer mode (ATM) networks. However, because of the finite state setting there, questions about stability of the scheme did not arise. This is however not the case here. Our state space is unbounded and hence we require Liapunov stability assumptions on the system to ensure tightness.

4 Convergence Analysis of SPSA-1

By a Polish space, we mean a complete separable metric space. Let S be a Polish space with complete metric $\hat{d}$. Also, let $\mathcal{P}(S)$ be the space of probability measures on S . Let $C_b(S)$ be the space of all bounded and continuous functions on S . A countable sequence of functions $\{f_i\} \subset C_b(S)$ is called a separating class for $\mathcal{P}(S)$ if for any probability measures $\mu, \nu \in \mathcal{P}(S)$, $\int f_i d\mu = \int f_i d\nu$, $i \geq 1$, implies $\mu = \nu$. It is easy to show (cf. Ch.2 of [9]) that such a separating class of functions exists for any such $\mathcal{P}(S)$. Also, a class of functions $\{f_\alpha, \alpha \in I\} \subset C_b(S)$ is called a convergence determining class for $\mathcal{P}(S)$ if for any sequence of measures $\{\mu_n\}_{n=1}^\infty \in \mathcal{P}(S)$, $\int f_\alpha d\mu_n \rightarrow \int f_\alpha d\mu_\infty$ $\forall \alpha \in I$, implies $\mu_n \rightarrow \mu_\infty$ in $\mathcal{P}(S)$.

Now define a metric ρ on $\mathcal{P}(S)$ as follows: For any $\mu, \nu \in \mathcal{P}(S)$,

$$\rho(\mu, \nu) = \sum_{n=1}^{\infty} 2^{-n} \left| \int f_n d\mu - \int f_n d\nu \right|,$$

where $\{f_n\}$ is a separating class of functions for $\mathcal{P}(S)$. The topology induced by ρ is called the Prohorov topology. It can be shown that under ρ , the space $\mathcal{P}(S)$ is complete if and only if S is compact and not otherwise. Thus ρ is not complete if S is not compact. Now consider the following metric $\bar{\rho}$ defined in the following manner: For $\epsilon > 0$ and Borel $A \subset S$, let $A^\epsilon = \{x \in S \mid \hat{d}(x, A) < \epsilon\}$. For $\mu, \nu \in \mathcal{P}(S)$, define

$$\bar{\rho}(\mu, \nu) = \inf\{\epsilon > 0 \mid \mu(A) \leq \nu(A^\epsilon) + \epsilon, \nu(A) \leq \mu(A^\epsilon) + \epsilon, \text{ for all Borel } A \subset S\}.$$

The metric $\bar{\rho}$ is called the Prohorov metric. It retains the same topology as the Prohorov topology and is in addition complete. Moreover, $\mathcal{P}(S)$ is separable under the Prohorov topology thereby making the space $\mathcal{P}(S)$ Polish without the requirement that S be compact (in addition to being Polish). It can also be shown that if S is a compact Polish space, then so is $\mathcal{P}(S)$. In what follows, we shall assume that $\mathcal{P}(S)$ is equipped with the Prohorov topology and is thus Polish.

Consider now the stochastic approximation scheme SPSA-1. Recall that $\{(X^k(j), Y^k(j))\}$, $k = 1, 2$, are the two parallel simulations respectively governed by $\{\tilde{\theta}_j^k\}$, $k = 1, 2$, with $\tilde{\theta}_j^1 = \theta(m) - \delta \Delta(m)$

and $\tilde{\theta}_j^2 = \theta(m) + \delta\Delta(m)$ respectively, for $n_m < j \leq n_{m+1}$, $m \geq 0$. Also, the dynamics of the simulations $\{(X^k(j), Y^k(j))\}$, $k = 1, 2$, is governed by equations of type (2.3)-(2.4), with $\xi(j)$, $\eta(j)$ replaced with analogously defined $\xi^k(j)$, $\eta^k(j)$, $k = 1, 2$, respectively independent of one another.

In this section, we assume (A1), (A2) and (A3) (defined in Section 2) for our system. Let the filtration be represented by $\mathcal{F}_n \triangleq \sigma(X^k(j), Y^k(j), \xi^k(j), \eta^k(j), \bar{\theta}_j, \bar{\Delta}_j, k = 1, 2; j \leq n)$, where $\bar{\theta}_j = \theta(m)$ and $\bar{\Delta}_j = \Delta(m)$ for $n_m \leq j < n_{m+1}$, $m \geq 0$. For $m \geq 0$, define $\mathcal{P}(\mathcal{R}^{q+d})$ -valued random variables $\{\mu_m^k\}$, $k = 1, 2$, for $f \in C_b(\mathcal{R}^{q+d})$ (the space of bounded and continuous functions on $\mathcal{R}^{q+d}$) by

$$\int f d\mu_m^k = \frac{\sum_{i=n_m+1}^{n_{m+1}} a(i) f(X^k(i), Y^k(i))}{\sum_{i=n_m+1}^{n_{m+1}} a(i)},$$

$m \geq 0$, $k = 1, 2$. Let $\tilde{\mathcal{R}}^{q+d} = \mathcal{R}^{q+d} \cup \{\infty\}$ denote the one point compactification of $\mathcal{R}^{q+d}$ and let δ_∞ denote the Dirac measure at $\{\infty\}$. In other words,

$$\delta_\infty(x) = \begin{cases} 1 & \text{if } x = \{\infty\} \\ 0 & \text{otherwise} \end{cases} \quad (4.1)$$

Using the natural embedding of $\mathcal{R}^{q+d}$ into $\tilde{\mathcal{R}}^{q+d}$, one may view $\{\mu_m^k, m \geq 0\}$, $k = 1, 2$, as random sequences in $\mathcal{P}(\tilde{\mathcal{R}}^{q+d})$, a compact Polish space. The proof of Theorem 4.1 (below) differs at several places from that of a similar result in [6] since as already stated earlier, the latter was shown for finite state spaces which is not the case here. Moreover, Theorem 4.1 here, also serves to show stability of the proposed scheme, instead of just being a step in the convergence analysis as in [6].

Theorem 4.1 Almost surely, $(\mu_m^1, \mu_m^2, \theta(m), \Delta(m))$, $m \geq 0$, converges in $\mathcal{P}(\mathcal{R}^{k+d})^2 \times C \times E$ to the compact set $\{(\mu_{\theta-\delta\Delta}, \mu_{\theta+\delta\Delta}, \theta, \Delta) \mid \theta \in C, \Delta \in E\}$.

Proof The proof proceeds through several steps. For $f \in C_b(\mathcal{R}^{q+d})$, define sequences $\{Z_k(m), m \geq 1\}$, $k = 1, 2$, by

$$Z_k(m) = \sum_{j=0}^{m-1} b(j)^{-1} \left[\sum_{i=n_j+1}^{n_{j+1}} a(i) \left(f(X^k(i), Y^k(i)) - E[f(X^k(i), Y^k(i)) \mid \mathcal{F}_{i-1}] \right) \right]. \quad (4.2)$$

Then $\{Z_k(m), \mathcal{F}_{n_m}\}$, $k = 1, 2$, are zero mean, square integrable martingale sequences. Let us represent their quadratic variation processes by $\{< Z_k > (m)\}$, $k = 1, 2$, respectively. Then by definition,

$$< Z_k > (m) = \sum_{j=0}^{m-1} E[(Z_k(j+1) - Z_k(j))^2 \mid \mathcal{F}_{n_j}] + E[Z_k(0)^2], \quad (4.3)$$

with $< Z_k > (\infty) = \lim_{m \rightarrow \infty} < Z_k > (m)$. For ease of exposition, let us denote $F_k(i) \triangleq f(X^k(i), Y^k(i))$, $k = 1, 2$, in this part of the proof. Note that

$$\begin{aligned} E[(Z_k(j+1) - Z_k(j))^2 \mid \mathcal{F}_{n_j}] &= E \left[\frac{1}{b(j)^2} \left(\sum_{i=n_j+1}^{n_{j+1}} a(i) (F_k(i) - E[F_k(i) \mid \mathcal{F}_{i-1}]) \right)^2 \mid \mathcal{F}_{n_j} \right] \\ &= E \left[\frac{1}{b(j)^2} \sum_{i=n_j+1}^{n_{j+1}} a(i)^2 (F_k(i) - E[F_k(i) \mid \mathcal{F}_{i-1}])^2 \mid \mathcal{F}_{n_j} \right] \end{aligned}$$

$$+E \left[\frac{1}{b(j)^2} \sum_{i,l=n_j+1, i \neq l}^{n_{j+1}} a(i)a(l) (F_k(i) - E[F_k(i) | \mathcal{F}_{i-1}]) (F_k(l) - E[F_k(l) | \mathcal{F}_{l-1}]) | \mathcal{F}_{n_j} \right].$$

The second term on the RHS above can be written as

$$\begin{aligned} & E \left[\frac{1}{b(j)^2} \sum_{i,l=n_j+1, i > l}^{n_{j+1}} E[a(i)a(l) (F_k(i) - E[F_k(i) | \mathcal{F}_{i-1}]) (F_k(l) - E[F_k(l) | \mathcal{F}_{l-1}]) | \mathcal{F}_{i-1}] | \mathcal{F}_{n_j} \right] \\ & + E \left[\frac{1}{b(j)^2} \sum_{i,l=n_j+1, i < l}^{n_{j+1}} E[a(i)a(l) (F_k(i) - E[F_k(i) | \mathcal{F}_{i-1}]) (F_k(l) - E[F_k(l) | \mathcal{F}_{l-1}]) | \mathcal{F}_{l-1}] | \mathcal{F}_{n_j} \right] \\ & = 0 \text{ a.s.,} \end{aligned}$$

since for $i > l$, $\mathcal{F}_l \subseteq \mathcal{F}_{i-1}$ and similarly $\mathcal{F}_i \subseteq \mathcal{F}_{l-1}$ for $i < l$. Thus

$$\begin{aligned} & E \left[(Z_k(j+1) - Z_k(j))^2 | \mathcal{F}_{n_j} \right] \\ & = \frac{1}{b(j)^2} \sum_{i=n_j+1}^{n_{j+1}} a(i)^2 E \left[\left(f(X^k(i), Y^k(i)) - E[f(X^k(i), Y^k(i))] \right)^2 | \mathcal{F}_{n_j} \right]. \end{aligned} \quad (4.4)$$

Now since f is a bounded and continuous function, there exist constants $K_1, K'_1 > 0$ such that

$$< Z_k > (m) \leq K_1 \sum_{j=0}^{m-1} \frac{1}{b(j)^2} \sum_{i=n_j+1}^{n_{j+1}} a(i)^2 \leq K'_1 \sum_{j=0}^{m-1} \frac{1}{b(j)^2} \left(\frac{1}{n_j} - \frac{1}{n_{j+1}} \right),$$

the latter inequality follows from the fact that $\sum_{i=0}^n a(i)^2 \approx 1 - \frac{1}{n}$. Also note that $\sum_{i=0}^n a(i) \approx \ln(n)$, and $\sum_{i=0}^n b(i) \approx n^{1-\alpha}$. Now, from the definition of $\{n_m, m \geq 0\}$, $\sum_{i=0}^{n_{m+1}} a(i) \approx \sum_{i=0}^m b(i)$. Thus, $\ln(n_{m+1}) \approx m^{1-\alpha}$. Hence, $n_{m+1} \approx \exp(am^{1-\alpha})$ for some $a > 0$. Thus, $< Z_k > (\infty) < \infty$. Hence, by Proposition V11.2.3(c) of [26], $\{Z_k(m), m \geq 1\}$, $k = 1, 2$, are a.s. convergent martingale sequences. Now let us consider $\{Z_1(m)\}$. From the fact that $\sum_{j=n_m+1}^{n_{m+1}} a(j)/b(m) \rightarrow 1$, as $m \rightarrow \infty$ and from (3.1),

$$\int \left(f(x, y) - \int f(x', y') p_{\theta(m) - \delta \Delta(m)}(x, y; dx', dy') \right) \mu_m^1(dx, dy) \rightarrow 0 \text{ a.s.} \quad (4.5)$$

Now, since the above holds for all $f \in C_b(\mathcal{R}^{q+d})$, it holds in particular for those $f \in C_b(\mathcal{R}^{q+d})$ that are in a countable convergence determining class of $\mathcal{P}(\tilde{\mathcal{R}}^{q+d})$. Hence, outside a set of measure zero, any limit point of $(\mu_m^1, \theta(m) - \delta \Delta(m))$, $m \geq 0$, in $\mathcal{P}(\tilde{\mathcal{R}}^{q+d}) \times C$ must be of the form $(b\delta_\infty + (1-b)\mu, \theta - \delta \Delta)$. In the above, $b \in [0, 1]$, where when $b < 1$, μ must satisfy

$$\int f(x, y) \mu(dx, dy) = \int \int f(x', y') p_{\theta - \delta \Delta}(x, y; dx', dy') \mu(dx, dy).$$

Thus for $b < 1$, μ must be of the form $\mu_{\theta - \delta \Delta}$. For $b = 1$, μ is arbitrary and hence can be set to be $\mu_{\theta - \delta \Delta}$ itself. Now, note that if in the definition of the sequences $\{Z^k(m)\}$, $k = 1, 2$, the function $f(\cdot, \cdot)$ is replaced by the Liapunov function $V(\cdot, \cdot)$, the sequences $\{Z^k(m)\}$ continue to be martingale sequences. Also, from Assumption (A3.2) and (4.4), it is clear that the quadratic

variation process of such a martingale would converge as well. We then obtain (4.5) with function V replacing f . Now define

$$\phi_{\theta-\delta\Delta}(x, y) \triangleq \int V(x', y') p_{\theta-\delta\Delta}(x, y; dx', dy') - V(x, y),$$

with $\phi_{\theta-\delta\Delta}(\{\infty\}) = -\epsilon_0$, $\forall \theta \in C$. Here ϵ_0 is the same as in Assumption (A3.3). Now, as a consequence of Assumption (A3.3), the map $(\theta, x, y) \rightarrow \phi_{\theta}(x, y) : C \times \tilde{\mathcal{R}}^{q+d} \rightarrow \mathcal{R}$ is upper semicontinuous and bounded from above. Further, if $(\mu_m^1, \theta(m) - \delta\Delta(m)) \rightarrow (b\delta_\infty + (1-b)\mu_{\theta-\delta\Delta}, \theta - \delta\Delta)$ along a subsequence, then from (4.5) with f replaced by V , we have,

$$0 = \limsup_{m \rightarrow \infty} \int \phi_{\theta(m)-\delta\Delta(m)} d\mu_m^1 \leq (1-b) \int \phi_{\theta-\delta\Delta} d\mu_{\theta-\delta\Delta} - b\epsilon_0, \quad (4.6)$$

along the same subsequence. Now, it is clear from the definitions of $\phi_{\theta-\delta\Delta}$ and $\mu_{\theta-\delta\Delta}$ above that

$$\int \phi_{\theta-\delta\Delta} d\mu_{\theta-\delta\Delta} = 0.$$

Thus from (4.6), we have $0 \leq -b\epsilon_0$, which cannot happen unless $b = 0$. A similar argument holds for the sequence $\{\mu_m^2\}$. Thus $\{\mu_m^k\}$, $k = 1, 2$, are tight in $\mathcal{P}(\mathcal{R}^{q+d})$ and have limit points of the form $\mu_{\theta-\delta\Delta}^1$ or $\mu_{\theta+\delta\Delta}^2$. Now by Lemma 2.1, the maps $\theta \rightarrow \mu_{\theta-\delta\Delta}^1$ and $\theta \rightarrow \mu_{\theta+\delta\Delta}^2$, $k = 1, 2$, are continuous. The claim now follows from the fact that any continuous image of a compact set is compact. $\square$

The final step is to show convergence of the algorithm (3.5) to the set of local minima. The o.d.e. technique is commonly used to prove convergence of stochastic approximation algorithms. Here, we show that the algorithm (3.5) asymptotically converges to the stable points of the o.d.e. (4.7). For any function $H : \mathcal{R}^N \rightarrow \mathcal{R}$, let $\nabla H(x) \triangleq [\nabla_1 H(x), \dots, \nabla_N H(x)]^T$ represent the gradient of H at the point $x \in \mathcal{R}^N$. Let $\tilde{Z}(t) \triangleq (\tilde{Z}_1(t), \dots, \tilde{Z}_N(t)) \in \mathcal{R}^N$, with $\tilde{Z}_i(t)$, $i = 1, \dots, N$, satisfying the o.d.e.

$$\dot{\tilde{Z}}_i(t) = \tilde{\pi}_i(-\nabla_i J(\tilde{Z}(t))), \quad t \geq 0, \quad \tilde{Z}(0) \in C, \quad (4.7)$$

where for any bounded, continuous, real valued function $v(\cdot)$,

$$\tilde{\pi}_i(v(y)) = \lim_{0 < \eta \rightarrow 0} \left(\frac{\pi_i(y + \eta v(y)) - \pi_i(y)}{\eta} \right).$$

For $x = (x_1, \dots, x_N)^T$, let $\tilde{\pi}(x) = (\tilde{\pi}_1(x_1), \dots, \tilde{\pi}_N(x_N))^T$. The operator $\tilde{\pi}(\cdot)$ forces the o.d.e. (4.7) to evolve within the constraint set C . Let $K \triangleq \{\theta \in C \mid \tilde{\pi}(\nabla J(\theta)) = 0\}$.

We recall here a key result from [18] stated as Lemma 4.1 below. Consider an o.d.e. in $\mathcal{R}^N$

$$\dot{x}(t) = F(x(t)), \quad (4.8)$$

which has an asymptotically stable attracting set $\bar{G}$. Let $\bar{G}^\epsilon$ denote the ϵ -neighborhood of $\bar{G}$ viz., $\bar{G}^\epsilon \triangleq \{x \mid \exists x' \in \bar{G} \text{ s.t. } \|x - x'\| \leq \epsilon\}$, where $\|\cdot\|$ represents the sup norm. For $T > 0$, $\gamma > 0$, say that $y(\cdot)$ is a (T, γ) -perturbation of (4.8) if there exist real numbers $0 = T_0 < T_1 < T_2 < \dots$, such that $T_{i+1} - T_i \geq T$, $\forall i$, and on each interval $[T_i, T_{i+1}]$, there exists a solution $x^i(\cdot)$ of (4.8) such that

$$\sup_{t \in [T_i, T_{i+1}]} |x^i(t) - y(t)| < \gamma.$$

The following result is adapted from [18], pp.339. The proof of this can be found in the appendix of [10].

Lemma 4.1 For given $\epsilon > 0$, $T > 0$, there exists a $\bar{\gamma}$ such that for all $\gamma \in [0, \bar{\gamma}]$, any (T, γ) -perturbation of (4.8) converges to $\bar{G}^\epsilon$. $\square$

For fixed $\eta > 0$, let $K^\eta \triangleq \{\theta \in C \mid \exists \theta' \in K \text{ s.t. } \|\theta - \theta'\| \leq \eta\}$ represent the set of points within a distance η of the set K . As a direct consequence of Lemma 4.1, for any given $\eta, T > 0$, $\exists \bar{\gamma} > 0$ s.t. $\forall \gamma \in [0, \bar{\gamma}]$, any (T, γ) -perturbation of (4.7) shall converge to K^η . Finally, Theorem 4.2 shows that given $\eta > 0$, there exists a $\bar{\delta} > 0$ such that the algorithm SPSA-1 for all $\delta \leq \bar{\delta}$, converges to K^η a.s. We only sketch the proof of this theorem here since the details are similar to that in [6].

Theorem 4.2 Given $\eta > 0$, $\exists \bar{\delta} > 0$ such that for any $\delta \in (0, \bar{\delta}]$, the algorithm (3.5) converges to K^η almost surely.

Proof (sketch) Note that the algorithm (3.5) can be written as:

$$\theta_i(m+1) = \pi_i \left(\theta_i(m) + b(m) \frac{\sum_{j=n_m+1}^{n_{m+1}} a(j) \left(\frac{h(Y_j^1) - h(Y_j^2)}{2\delta\Delta_{m,i}} \right)}{b(m)} \right).$$

Now from the fact that $\sum_{j=n_m+1}^{n_{m+1}} a(j)/b(m) \rightarrow 1$ as $m \rightarrow \infty$ and using the conclusions of Theorem 4.1, the algorithm can be shown to asymptotically exhibit the same behaviour as the following algorithm:

$$\theta_i(m+1) = \pi_i \left(\theta_i(m) + b(m) \left(\frac{J(\theta(m) - \delta\Delta(m)) - J(\theta(m) + \delta\Delta(m))}{2\delta\Delta_{m,i}} \right) \right). \quad (4.9)$$

Now construct martingale sequences $\{N_i(p)\}$, $i = 1, \dots, N$, as follows: For $i = 1, \dots, N$,

$$\begin{aligned} N_i(p) = & \sum_{j=0}^p b(j) \left(\frac{J(\theta(j) - \delta\Delta(j)) - J(\theta(j) + \delta\Delta(j))}{2\delta\Delta_{j,i}} \right. \\ & \left. - E \left[\frac{J(\theta(j) - \delta\Delta(j)) - J(\theta(j) + \delta\Delta(j))}{2\delta\Delta_{j,i}} \mid \mathcal{F}'_j \right] \right), \end{aligned}$$

where, for $j \geq 1$, $\mathcal{F}'_j \triangleq \sigma(\theta(0), \theta(1), \dots, \theta(j), \Delta(0), \Delta(1), \dots, \Delta(j-1))$ represents the filtration and the expectation $E[\cdot]$ is w.r.t. the common expectation of $\Delta_{j,i}$. Note that $\Delta(j)$ is independent of $\mathcal{F}'_j$. It can now be easily shown (see [6]) that sequences $\{N_i(p)\}$, $i = 1, \dots, N$, converge a.s., as $p \rightarrow \infty$. Hence, one could replace algorithm (4.9) by the following equivalent algorithm: For $i = 1, \dots, N$,

$$\theta_i(m+1) = \pi_i \left(\theta_i(m) + b(m) E \left[\frac{J(\theta(m) - \delta\Delta(m)) - J(\theta(m) + \delta\Delta(m))}{2\delta\Delta_{m,i}} \mid \mathcal{F}'_j \right] \right). \quad (4.10)$$

Finally, it can be shown that as $\delta \rightarrow 0$,

$$\left| E \left[\frac{J(\theta(j) - \delta\Delta(j)) - J(\theta(j) + \delta\Delta(j))}{2\delta\Delta_{j,i}} \mid \mathcal{F}'_j \right] - \nabla_i J(\theta(j)) \right| \rightarrow 0.$$

Thus (4.10) can be shown to asymptotically exhibit analogous behaviour as the following algorithm:
For $i = 1, \dots, N$,

$$\theta_i(m+1) = \pi_i(\theta_i(m) + b(m)(-\nabla_i J(\theta(m)))) .$$

The last finally can be shown to asymptotically track the stable points of the o.d.e. (4.7). $\square$

Remark Note that K is the set of all critical points of (4.7), and not just the set of local minima. However, points in K that are not local minima will be unstable equilibria and since our algorithm is of the gradient descent type, it will converge a.s. to the η -neighborhood of $K_0(\triangleq$ the set of local minima of $J(\cdot)$) $\subset K$.

5 Convergence Analysis of SPSA-2

Consider now the stochastic approximation scheme SPSA-2. We assume (A1), (A2') and (A3) in this section. Recall that $\{(X^-(l), Y^-(l))\}$ and $\{(X^+(l), Y^+(l))\}$ are the two parallel simulations respectively governed by $\{\tilde{\theta}(l) - \delta\tilde{\Delta}(l)\}$ and $\{\tilde{\theta}(l) + \delta\tilde{\Delta}(l)\}$, where $\tilde{\theta}(l) = \theta\left(\left[\frac{l}{L}\right]\right)$ and $\tilde{\Delta}(l) = \Delta\left(\left[\frac{l}{L}\right]\right)$, and where $\left[\frac{l}{L}\right]$ represents integral part of $\frac{l}{L}$. In other words, if l has the form $l = nL + m$, where $m \in \{0, 1, \dots, L-1\}$ and n is an integer, then $\left[\frac{l}{L}\right] = n$. Note that by definition, $\{\tilde{\theta}(l)\}$ (resp. $\{\tilde{\Delta}(l)\}$) takes values in the compact set C (resp. E). Define $\mathcal{P}(\mathcal{R}^q \times \mathcal{R}^d \times C \times E)$ -valued processes $\{\mu_n^-\}$ and $\{\mu_n^+\}$ by

$$\mu_n^-(A_1 \times A_2 \times B \times D) = \frac{1}{n} \sum_{m=0}^{n-1} I\{X^-(m) \in A_1, Y^-(m) \in A_2, \tilde{\theta}(m) \in B, \tilde{\Delta}(m) \in D\}$$

and

$$\mu_n^+(A_1 \times A_2 \times B \times D) = \frac{1}{n} \sum_{m=0}^{n-1} I\{X^+(m) \in A_1, Y^+(m) \in A_2, \tilde{\theta}(m) \in B, \tilde{\Delta}(m) \in D\},$$

for Borel sets $A_1 \subset \mathcal{R}^q$, $A_2 \subset \mathcal{R}^d$, $B \subset C$ and $D \subset E$. Let $\tilde{\mathcal{R}}^{q+d} = \mathcal{R}^{q+d} \cup \{\infty\}$ denote the one point compactification of $\mathcal{R}^{q+d}$. The following theorem establishes tightness of sequences $\{\mu_n^-\}$ and $\{\mu_n^+\}$ in $\mathcal{P}(\mathcal{R}^q \times \mathcal{R}^d \times C \times E)$. The proof follows in a somewhat similar manner as the proof of Theorem 4.1. Here one first shows under the martingale stability theorem of [25], pp.53 and Assumption (A2') that $\{\mu_n^-\}$ and $\{\mu_n^+\}$ are tight in $\mathcal{P}(\tilde{\mathcal{R}}^{q+d} \times C \times E)$ which is a compact Polish space. Finally, using the Liapunov function $V(\cdot, \cdot)$ in Assumption (A3) and again the martingale stability theorem of [25], pp.53, one shows that $\{\mu_n^-\}$ and $\{\mu_n^+\}$ are in fact tight in $\mathcal{P}(\mathcal{R}^q \times \mathcal{R}^d \times C \times E)$ itself. We do not present the proof of Theorem 5.1 here so as to avoid repetition.

Theorem 5.1 Almost surely, $\{\mu_n^-\}$ and $\{\mu_n^+\}$ are tight sequences in $\mathcal{P}(\mathcal{R}^q \times \mathcal{R}^d \times C \times E)$. $\square$

We now proceed with the rest of the analysis. Let for $k \geq 1$, $\mathcal{F}'_k \triangleq \sigma(\theta(0), \theta(1), \dots, \theta(k), \Delta(0), \Delta(1), \dots, \Delta(k-1))$. Then $\Delta(k)$ is independent of $\mathcal{F}'_k$, $\forall k \geq 1$. Define sequences $\{N_i^-(p), p \geq 1\}$, $\{N_i^+(p), p \geq 1\}$; $i = 1, \dots, N$, as follows:

$$N_i^-(p) = \sum_{j=0}^p a(j) \left(\frac{J(\theta(j)) - \delta\Delta(j)}{\Delta_{j,i}} - E \left[\frac{J(\theta(j)) - \delta\Delta(j)}{\Delta_{j,i}} \mid \mathcal{F}'_j \right] \right),$$

and

$$N_i^+(p) = \sum_{j=0}^p a(j) \left(\frac{J(\theta(j) + \delta \Delta(j))}{\Delta_{j,i}} - E \left[\frac{J(\theta(j) + \delta \Delta(j))}{\Delta_{j,i}} \mid \mathcal{F}'_j \right] \right).$$

Then, we have

Lemma 5.1 For every $i = 1, \dots, N$, $\{N_i^-(p)\}$ and $\{N_i^+(p)\}$ converge a.s.

Proof Follows in a similar manner as Lemma A.2 of [6]. $\square$

Now let $s(0) = 0$, $s(n) = \sum_{i=0}^{n-1} a(i)$, $n \geq 0$. Let $\Delta_{t,i} = \Delta_{n,i}$, for $t \in [s(n), s(n+1)]$, $n \geq 1$. Further, let $\Delta(t) = (\Delta_{t,1}, \dots, \Delta_{t,N})^T$. Recall that for any bounded, continuous, real valued function $v(\cdot)$,

$$\tilde{\pi}_i(v(y)) = \lim_{0 < \eta \rightarrow 0} \left(\frac{\pi_i(y + \eta v(y)) - \pi_i(y)}{\eta} \right).$$

Also, for $x = (x_1, \dots, x_N)^T \in \mathcal{R}^N$, $\tilde{\pi}(x) = (\tilde{\pi}_1(x), \dots, \tilde{\pi}_N(x))^T$. Consider the following o.d.e.: For $i = 1, \dots, N$,

$$\dot{\theta}_i(t) = \tilde{\pi}_i \left(E \left[\frac{J(\theta(t) - \delta \Delta(t)) - J(\theta(t) + \delta \Delta(t))}{2\delta \Delta_{t,i}} \right] \right), \quad (5.1)$$

where, the operator $E[\cdot]$ in (5.1) represents the expectation w.r.t. the common c.d.f. of $\{\Delta_{t,i}\}$. Recall that L represents the number of data aggregation epochs in between two successive parameter updates in SPSA-2. Let $c(i) = b(i)L$. It is easy to see that

$$\sum_i c(i) = \infty, \quad \sum_i c(i)^2 < \infty, \quad a(i) = o(c(i)).$$

Here, we shall consider $\{a(i)\}$ and $\{c(i)\}$ to be the two step-size sequences (as opposed to $\{a(i)\}$ and $\{b(i)\}$ in SPSA-1). Let $t(0) = 0$, $t(n) = \sum_{i=0}^{n-1} c(i)$, $n \geq 1$. Let $z^-(\cdot), z^+(\cdot) : [0, \infty) \rightarrow \mathcal{R}$ and $\bar{\theta}(\cdot) : [0, \infty) \rightarrow C$ denote the continuous functions obtained by setting $z^-(t(n)) = Z_L^-(n)$, $z^+(t(n)) = Z_L^+(n)$, $\bar{\theta}(t(n)) = \theta(n)$ respectively $\forall n$, with linear interpolation on $[t(n), t(n+1)]$, $n \geq 0$. Consider the system of o.d.e.'s

$$\begin{aligned} \dot{\theta}(t) &= 0, \\ \dot{z}^-(t) &= J(\theta(t) - \delta \Delta(t)) - z^-(t), \\ \dot{z}^+(t) &= J(\theta(t) + \delta \Delta(t)) - z^+(t). \end{aligned} \quad (5.2)$$

We now have

Theorem 5.2 For any $T, \delta > 0$, $(z^-(t(n) + \cdot), z^+(t(n) + \cdot), \theta(t(n) + \cdot))$ is a bounded (T, δ) -perturbation of (5.2) for n sufficiently large.

Proof Note that the algorithm SPSA-2 (cf. (3.7)) can be rewritten as follows: For $i = 1, \dots, N$,

$$\theta_i(n+1) = \pi_i \left(\theta_i(n) + a(n) \left[\frac{Z_L^-(n) - Z_L^+(n)}{2\delta \Delta_{n,i}} \right] \right),$$

$$Z_L^-(n+1) = Z_L^-(n) + c(n) \frac{1}{L} \sum_{m=0}^{L-1} (h(Y_m^-(n+1)) - Z_m^-(n+1)),$$

$$Z_L^+(n+1) = Z_L^+(n) + c(n) \frac{1}{L} \sum_{m=0}^{L-1} (h(Y_m^+(n+1)) - Z_m^+(n+1)). \quad (5.3)$$

From $\{Z_m^-(n)\}$ and $\{Z_m^+(n)\}$, $n \geq 0$, $m \in \{0, 1, \dots, L-1\}$ defined in (3.7), we obtain sequences $\{Z^-(l)\}$ and $\{Z^+(l)\}$ as follows: First write l as $l = nL + m$ for some $n \geq 0$ and $m \in \{0, 1, \dots, L-1\}$. Then set $Z^-(l) = Z^-(nL + m) \triangleq Z_m^-(n)$. Similarly set $Z^+(l) = Z^+(nL + m) \triangleq Z_m^+(n)$. Now from the second and third equations in (3.7), note that $Z^-(l+1)$ (resp. $Z^+(l+1)$) is the convex combination of $h(\cdot)$ and $Z^-(l)$ (resp. $Z^+(l)$). Thus $\{Z^-(l)\}$ (resp. $\{Z^+(l)\}$) are uniformly bounded sequences with upper bound depending on h and $\|Z^-(0)\|$ (resp. $\|Z^+(0)\|$). Now $\{Z_L^-(n)\}$ (resp. $\{Z_L^+(n)\}$) is just a subsequence of $\{Z^-(l)\}$ (resp. $\{Z^+(l)\}$) and hence is uniformly bounded (irrespective of the value of L) as well. Before we proceed further, let us look at the term $\frac{1}{L} \sum_{m=0}^{L-1} Z_m^-(n+1)$ on the RHS of the second equation in (5.3). We will show that it has the same

asymptotic behaviour as $Z_L^-(n)$. A similar argument shall hold for the term $\frac{1}{L} \sum_{m=0}^{L-1} Z_m^+(n+1)$ on the RHS of the third equation in (5.3). First note that the terms $Z_0^-(n+1), Z_1^-(n+1), \dots, Z_{L-1}^-(n+1)$ are all governed by the same parameter update viz., $(\theta(n+1) - \delta\Delta(n+1))$. Now recall that for $m \in \{0, 1, \dots, L-1\}$, $Z_m^-(n+1) = Z^-((n+1)L + m)$. For notational simplicity let $(n+1)L = k$ in the rest of the proof. Now from the second equation in (3.7),

$$Z^-(k+1) = (1 - b(n))Z^-(k) + b(n)h(Y^-(k))$$

Similarly,

$$\begin{aligned} Z^-(k+2) &= (1 - b(n))Z^-(k+1) + b(n)h(Y^-(k+1)) \\ &= (1 - b(n))^2 Z^-(k) + (1 - b(n))b(n)h(Y^-(k)) + b(n)h(Y^-(k+1)) \end{aligned}$$

Proceeding in this manner, one obtains

$$\begin{aligned} Z^-(k+L-1) &= (1-b(n))^{L-1} Z^-(k) + (1-b(n))^{L-2} b(n)h(Y^-(k)) + (1-b(n))^{L-3} b(n)h(Y^-(k+1)) + \dots \\ &\quad + (1-b(n))b(n)h(Y^-(k+L-3)) + b(n)h(Y^-(k+L-2)) \end{aligned}$$

Now,

$$\begin{aligned} \frac{1}{L} \sum_{m=0}^{L-1} Z^-(k+m) &= \frac{1}{L} [(1 + (1-b(n)) + \dots + (1-b(n))^{L-1}) Z^-(k) \\ &\quad + (b(n) + (1-b(n))b(n) + \dots + (1-b(n))^{L-2} b(n)) h(Y^-(k)) \\ &\quad + (b(n) + (1-b(n))b(n) + \dots + (1-b(n))^{L-3} b(n)) h(Y^-(k+1)) \\ &\quad + \dots + b(n)h(Y^-(k+L-2))] \\ &= \frac{1}{L} \left[\left(\frac{1 - (1-b(n))^L}{b(n)} \right) Z^-(k) + (1 - (1-b(n))^{L-1}) h(Y^-(k)) + (1 - (1-b(n))^{L-2}) h(Y^-(k+1)) + \right. \\ &\quad \left. \dots + b(n)h(Y^-(k+L-2)) \right] \end{aligned} \quad (5.4)$$

Now applying standard martingale arguments (as in Lemma 4.3 of [5]) in the second (resp. third) equation in (3.7), the algorithm (3.7) can be shown to behave asymptotically (as $n, k \rightarrow \infty$) in the

same manner as an analogous algorithm that has $J(\theta(n+1) - \delta\Delta(n+1))$ (resp. $J(\theta(n+1) + \delta\Delta(n+1))$) in place of the terms $h(Y^-(k)), h(Y^-(k+1)), \dots, h(Y^-(k+L-2))$ (resp. $h(Y^+(k)), h(Y^+(k+1)), \dots, h(Y^+(k+L-2))$). Thus one can in particular consider the latter algorithm in place of (3.7). Upon simplification now (with a slight abuse of notation), the RHS of (5.4) can be replaced by

$$\begin{aligned} &= \frac{1}{L} \left[\left(\frac{1 - (1 - b(n))^L}{b(n)} \right) Z^-(k) + LJ(\theta(n+1) - \delta\Delta(n+1)) \right. \\ &\quad \left. + \frac{(1 - b(n))^L - 1}{b(n)} J(\theta(n+1) - \delta\Delta(n+1)) \right]. \end{aligned}$$

Now,

$$\begin{aligned} \left\| \frac{1}{L} \sum_{m=0}^{L-1} Z^-(k+m) - Z^-(k) \right\| &\leq \left\| \frac{1}{L} \sum_{m=0}^{L-1} Z^-(k+m) - \frac{1 - (1 - b(n))^L}{Lb(n)} Z^-(k) \right\| \\ &\quad + \left\| \left(\frac{1 - (1 - b(n))^L}{Lb(n)} - 1 \right) Z^-(k) \right\| \\ &\leq \left\| LJ(\theta(n+1) - \delta\Delta(n+1)) + \frac{(1 - b(n))^L - 1}{b(n)} J(\theta(n+1) - \delta\Delta(n+1)) \right\| \\ &\quad + \left\| \left(\frac{1 - (1 - b(n))^L}{Lb(n)} - 1 \right) Z^-(k) \right\|. \end{aligned}$$

Consider a real valued function $f(x) = \left(\frac{1 - (1 - x)^L}{x} \right)$. By L'Hospital's rule, it is easy to see that

$\lim_{|x| \rightarrow 0} f(x) = L$. It is thus clear that $\lim_{b(n) \rightarrow 0} \left(\frac{1 - (1 - b(n))^L}{b(n)} \right) = L$. Hence,

$$\left\| \frac{1}{L} \sum_{m=0}^{L-1} Z^-(k+m) - Z^-(k) \right\| \longrightarrow 0, \text{ as } k \rightarrow \infty.$$

Finally, the first equation in (5.3) can be rewritten as

$$\theta_i(n+1) = \pi_i \left(\theta_i(n) + c(n) \frac{a(n)}{c(n)} \left[\frac{Z_L^-(n) - Z_L^+(n)}{2\delta\Delta_{n,i}} \right] \right) \quad (5.5)$$

Moreover, since $a(n) = o(c(n))$, applying standard arguments [10] to (5.5) and the second and third equations in (5.3), one obtains the claim. $\square$

Define $\hat{z}^-(\cdot), \hat{z}^+(\cdot) : [0, \infty) \rightarrow \mathcal{R}$ and $\hat{\theta}(\cdot) : [0, \infty) \rightarrow C$ by $\hat{z}^-(s(n)) = Z_L^-(n)$, $\hat{z}^+(s(n)) = Z_L^+(n)$, $\hat{\theta}(s(n)) = \theta(n)$ respectively $\forall n$, with linear interpolation on intervals $[s(n), s(n+1)]$, $n \geq 0$.

Lemma 5.2 For any $T, \delta > 0$, $\hat{\theta}(s(n) + \cdot)$ is a bounded (T, δ) -perturbation of (5.1) for sufficiently large n .

Proof Rewrite the first equation in (3.7) as follows: For $i = 1, \dots, N$,

$$\begin{aligned} \theta_i(m+1) &= \pi_i(\theta_i(m) + a(m)E \left[\frac{J(\theta(m) - \delta\Delta(m)) - J(\theta(m) + \delta\Delta(m))}{2\delta\Delta_{m,i}} \mid \mathcal{F}_m \right] \\ &\quad + \eta_1(m) + \eta_2(m)), \end{aligned} \quad (5.6)$$

where,

$$\eta_1(m) = a(m) \left(\frac{J(\theta(m) - \delta\Delta(m)) - J(\theta(m) + \delta\Delta(m))}{2\delta\Delta_{m,i}} \right. \\ \left. - E \left[\frac{J(\theta(m) - \delta\Delta(m)) - J(\theta(m) + \delta\Delta(m))}{2\delta\Delta_{m,i}} \mid \mathcal{F}'_m \right] \right),$$

and

$$\eta_2(m) = a(m) \left(\frac{Z_L^-(m) - Z_L^+(m)}{2\delta\Delta_{m,i}} - \frac{J(\theta(m) - \delta\Delta(m)) - J(\theta(m) + \delta\Delta(m))}{2\delta\Delta_{m,i}} \right).$$

Now, $\eta_1(m)$ becomes asymptotically negligible by Lemma 5.1 and $\eta_2(m)$ vanishes asymptotically as a consequence of Lemma 4.1 applied to Theorem 5.2. The algorithm (3.7) can then be viewed as a discretization of the o.d.e. (5.1) except that as mentioned earlier, it has in addition asymptotically diminishing error terms $\eta_1(m)$ and $\eta_2(m)$. Now a standard argument as in pp.191-194 of [20], proves the claim. $\square$

Lemma 5.3 For any $\theta(m) \in C$, for all $i = 1, \dots, N$,

$$\lim_{\delta \downarrow 0} \left| E \left[\frac{J(\theta(m) - \delta\Delta(m)) - J(\theta(m) + \delta\Delta(m))}{2\delta\Delta_{m,i}} \mid \mathcal{F}'_m \right] - \nabla_i J(\theta(m)) \right| = 0.$$

Proof Follows in a similar manner as Lemma A.5 of [6]. $\square$

Recall that the set $K \triangleq \{\theta \in C \mid \tilde{\pi}(\nabla J(\theta)) = 0\}$ is the asymptotically stable attractor set for the o.d.e. (4.7) with $J(\cdot)$ itself serving as the strict Liapunov function. Further, $K^\eta \triangleq \{\theta \in C \mid \|\theta - \theta_0\| < \eta, \theta_0 \in K\}$ represents the set of points in C that are within an η -distance from the set K . We now have

Lemma 5.4 Given $\eta > 0$, there exists $\delta_0 > 0$ such that for all $\delta \in (0, \delta_0]$, K^η is an asymptotically stable attractor set for the o.d.e. (5.1).

Proof As already mentioned, $J(\cdot)$ itself serves as a strict Liapunov function for (4.7) outside the set K . Now by Lemmas 5.2 and 5.3, for sufficiently small δ , $J(\cdot)$ will also serve as a strict Liapunov function for (5.1) outside the set K^η . $\square$

Finally, we come to the main result of this section.

Theorem 5.3 Given $\eta > 0$, there exists $\delta_0 > 0$ such that for all $\delta \in (0, \delta_0]$, $\theta(n) \rightarrow K^\eta$ a.s.

Proof Follows from Lemmas 4.1, 5.2 and 5.4. $\square$

This completes the convergence analysis of both the algorithms SPSA-1 and SPSA-2.

6 Numerical Results

In this section, we demonstrate our algorithms SPSA-1 and SPSA-2 by means of a simple queueing system and numerically compare their performance with the algorithms in [4] and [5], $(N+1)$ -Simulation FDSA-1, Two-Simulation FDSA-1 and $(N+1)$ -Simulation FDSA-2. We consider the two-node queueing network shown in Figure 1 below.

There are two external arrival streams (one each) to the two nodes. Arrivals to the nodes from these streams follow independent Poisson processes with rates λ_1 and λ_2 . The service times are

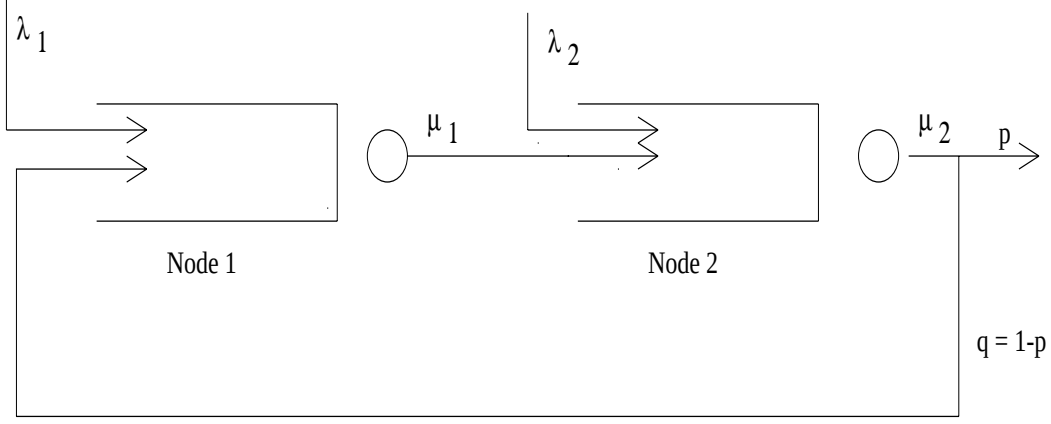


Figure 1: Queueing Network

exponentially distributed with rates $\mu_1(\theta_1)$ and $\mu_2(\theta_2)$, respectively, where θ_1 and θ_2 are parameter vectors at the two nodes. The exact dependence of μ_1 and μ_2 on θ_1 and θ_2 , respectively, is given below. A customer after service at Node 1 joins the queue at Node 2. After service at Node 2, a customer either departs with probability p or is fed back to Node 1 with the remainder probability of $1 - p$. Let $W_j^i(\theta_i)$, $i = 1, 2$, $j = 1, 2, \dots$, represent the waiting time of the j th customer at the i th node when the parameter value for the i th node is θ_i . Our aim is to find the optimum (joint) parameter vector (θ_1, θ_2) within the constraint set C which minimizes the sum of the stationary mean waiting times in the two queues. The constraint set C is defined as follows: Given $M > 0$, each component θ_i^j , $i = 1, 2$, $j = 1, \dots, M$, takes values in the interval $[\theta_{i,\min}^j, \theta_{i,\max}^j]$ and so the set C is defined as

$$C = [\theta_{1,\min}^1, \theta_{1,\max}^1] \times \dots \times [\theta_{1,\min}^M, \theta_{1,\max}^M] \times [\theta_{2,\min}^1, \theta_{2,\max}^1] \times \dots \times [\theta_{2,\min}^M, \theta_{2,\max}^M].$$

We assume that both θ_1 and θ_2 are vectors of the same dimension M . Note that $M = N/2$. Thus, $\theta_i \triangleq (\theta_i^1, \dots, \theta_i^M)^T$, $i = 1, 2$, and the whole parameter vector is represented as $\theta \triangleq (\theta_1^1, \dots, \theta_1^M, \theta_2^1, \dots, \theta_2^M)^T$. Let $\bar{\theta} \triangleq (\bar{\theta}_1^1, \dots, \bar{\theta}_1^M, \bar{\theta}_2^1, \dots, \bar{\theta}_2^M)^T$ represent the target parameter (we will explain this in a moment). The dependence of the service times on the parameters has the following form:

$$\mu_i(\theta_i) = \frac{\bar{\mu}_i}{\left(1 + \prod_{j=1}^M |\theta_i^j - \bar{\theta}_i^j|\right)}, \quad i = 1, 2,$$

where $\bar{\mu}_i$, $i = 1, 2$, is assumed to be constant. Note that the cost, which is the sum of the stationary mean waiting times in the two queues, will be minimized if the service rates are maximized. The latter clearly occurs at $\theta = \bar{\theta}$. Thus we know that the optimum for our problem lies at $\bar{\theta}$. Let $\theta_i^j(0)$, $i = 1, 2$, $j = 1, \dots, M$, represent the initial (starting) values of the parameter components.

For the simulation experiments, we select the following step-size sequences $(\{a(n)\})$ and $(\{b(n)\})$ for all the five schemes: $a(0) = b(0) = 1$, $a(n) = 1/n$, $b(n) = 1/n^{2/3}$, $n \geq 1$. Moreover, for SPSA-2, we choose $L = 100$ in the experiments. Thus data is aggregated in SPSA-2 over 100 instants in between two successive parameter updates. For both SPSA-1 and SPSA-2, we choose random variables $\Delta_{n,i}$, $i = 1, \dots, N$, $n \geq 0$, to be i.i.d., Bernoulli distributed viz., $\Delta_{n,i} = \pm 1$ w.p. $1/2$, $i = 1, \dots, N$, $n \geq 0$. We now present the simulation results. We consider the following set up for

all the three algorithms.

$$\lambda_1 = 0.2,$$

$$\lambda_2 = 0.1,$$

$$\bar{\mu}_1 = 87,$$

$$\bar{\mu}_2 = 92,$$

$$p = 0.4,$$

$$\theta_{i,\min}^j = 0.1, i = 1, 2, j = 1, \dots, M,$$

$$\theta_{i,\max}^j = 0.6, i = 1, 2, j = 1, \dots, M,$$

$$\bar{\theta}_i^j = 0.3, i = 1, 2, j = 1, \dots, M,$$

$$\theta_1^j(0) = 0.2, j = 1, \dots, M,$$

$$\theta_2^j(0) = 0.4, j = 1, \dots, M.$$

We consider two values of M for our experiments: $M = 5$ and $M = 20$. Thus the parameter vectors we consider in the simulations using the two schemes have dimensions 10 and 40 respectively. We consider a total of 3×10^5 data aggregation epochs for all the five schemes. The corresponding total number of parameter updates for each algorithm (for $M = 5$ and 20 respectively) is shown in Table 1 below.

Table 1: Number of Parameter Updates in 3×10^5 epochs

Algorithm	No. of Parameter Updates for $M = 5$	No. of Parameter Updates for $M = 20$
$(N + 1)$ -FDSA-2	3×10^5	3×10^5
SPSA-2	3×10^3	3×10^3
SPSA-1	92	92
$(N + 1)$ -FDSA-1	92	92
Two-FDSA-1	$9\frac{2}{10}$	$2\frac{12}{40}$

Note that whereas there are 3×10^5 parameter updates in $(N + 1)$ -Simulation FDSA-2, the corresponding number in SPSA-2 is 3×10^3 (since $L = 100$). Moreover, the corresponding number in SPSA-1 and $(N + 1)$ -Simulation FDSA-1 is only 92. This is so because for the step-size sequences considered in the experiments viz., $a(0) = b(0) = 1$, $a(n) = 1/n$, $b(n) = 1/n^{2/3}$, $n \geq 1$, the values n_{92} and n_{93} in $\{n_m, m \geq 1\}$ (defined after (3.2)) are as follows: $n_{92} \approx 2.94 \times 10^5$ and $n_{93} \approx 3.09 \times 10^5$ respectively. Finally, in Two-Simulation FDSA-1, the number of parameter updates for $M = 5$ (mentioned in Table 1) is $9\frac{2}{10}$. It is written in this manner to indicate that in addition to the 9 times that the whole parameter is updated in 3×10^5 data aggregation epochs, the first two components of the parameter vector are also updated for a 10th time. Similarly, for $M = 20$, the number of parameter updates in Two-Simulation FDSA-1 is $2\frac{12}{40}$.

In what follows, we shall compare the performance of our SPSA algorithms (SPSA-1 and SPSA-2) with the algorithms of [4] and [5] (viz., $(N + 1)$ -Simulation FDSA-1, Two-Simulation FDSA-1 and $(N + 1)$ -Simulation FDSA-2), in terms of speed of convergence. We choose the Euclidean distance between the current parameter update and the target parameter value as the performance metric and plot that w.r.t. the number of data aggregation epochs for all the five schemes. The

Euclidean distance $d(\theta, \bar{\theta})$ is defined by

$$d(\theta, \bar{\theta}) = \left((\theta_1^1 - \bar{\theta}_1^1)^2 + \dots + (\theta_1^M - \bar{\theta}_1^M)^2 + (\theta_2^1 - \bar{\theta}_2^1)^2 + \dots + (\theta_2^M - \bar{\theta}_2^M)^2 \right)^{1/2}.$$

We performed five independent replications of each experiment using different seeds. In Figures 2 and 3, the mean trajectories from these experiments are plotted for all five schemes, for both the 10-dimensional and the 40-dimensional parameter cases respectively. The standard error from these replications for the five schemes for both cases was computed at the end of these simulations, and is indicated in Table 2.

As expected, algorithms SPSA-1 and SPSA-2 show significantly better performance than the rest of the algorithms. Also, as expected, Two-Simulation FDSA-1 shows the worst performance amongst the five algorithms. It is interesting to observe that for the 10-dimensional parameter case, SPSA-1 shows the best performance amongst the five algorithms. Also, for this case, the performance of $(N + 1)$ -Simulation FDSA-2 is nearly as bad as that of Two-Simulation FDSA-1. However, for the 40-dimensional parameter case, it is SPSA-2 (and not SPSA-1) that shows the best performance. Moreover, the performance of $(N + 1)$ -Simulation FDSA-2 for this case is close to that of $(N + 1)$ -Simulation FDSA-1, and is considerably better than that of Two Simulation FDSA-1.

The above seems to indicate that for parameter dimensions that are not very high, SPSA-1 (resp. variants of SPSA-1 in [4]) performs better than SPSA-2 (resp. $(N + 1)$ -Simulation analogue of SPSA-2 in [5]). However, for cases where the parameter dimension is high, the opposite is true. The reason for this could be that SPSA-1 and its variants require that the parameter be held fixed over intervals of increasing size. Also, higher dimensional parameters would typically require several updates before convergence is achieved.

We observed that when $L = 1$ (i.e., no data aggregation in between successive parameter updates), SPSA-2 does not exhibit good performance. As already mentioned in Section 3, this is probably because of the fact that since SPSA-2 uses only two parallel simulations, the system is unable to adapt to the new parameter update before it changes again. Data aggregation over L epochs (for $L > 1$) on the other hand leads to additional averaging and hence improved performance. The choice of L is completely arbitrary though. We observed, however, that the performance somewhat degrades when L is either too low or too high.

We observed that using SPSA-1, for $M = 5$ (i.e., number of dimensions of the parameter = 10), the Euclidean distance between the current update and the optimum parameter became less than 0.10 (on an average of five replications) from its 28th parameter update onwards (after only 3183 data aggregation epochs). Using SPSA-2, the same is achieved from its 296th parameter update (after 2.96×10^4 data aggregation epochs). However, using SPSA-1 (after running the algorithm long enough), it was observed that for $M = 20$ (i.e., number of dimensions of the parameter = 40), the same distance became less than 0.10 after its 102nd parameter update (after nearly 4.75×10^5 data aggregation epochs). The same is achieved in SPSA-2 from its 829th update onwards (or after only 8.29×10^4 data aggregation points).

It should be noted that for the same number of data aggregation epochs, SPSA-1, SPSA-2 and Two-Simulation FDSA-1 require the least number of simulations. In particular, for 3×10^5 data aggregation epochs, the algorithms SPSA-1, SPSA-2 and Two-Simulation FDSA-1, each require

$2 \times 3 \times 10^5$ simulations for both the 10-parameter and the 40-parameter vectors, whereas, $(N + 1)$ -Simulation FDSA-1 and $(N + 1)$ -Simulation FDSA-2 require $11 \times 3 \times 10^5$ simulations (each) for the 10-parameter and $41 \times 3 \times 10^5$ simulations for the 40-parameter cases, respectively.

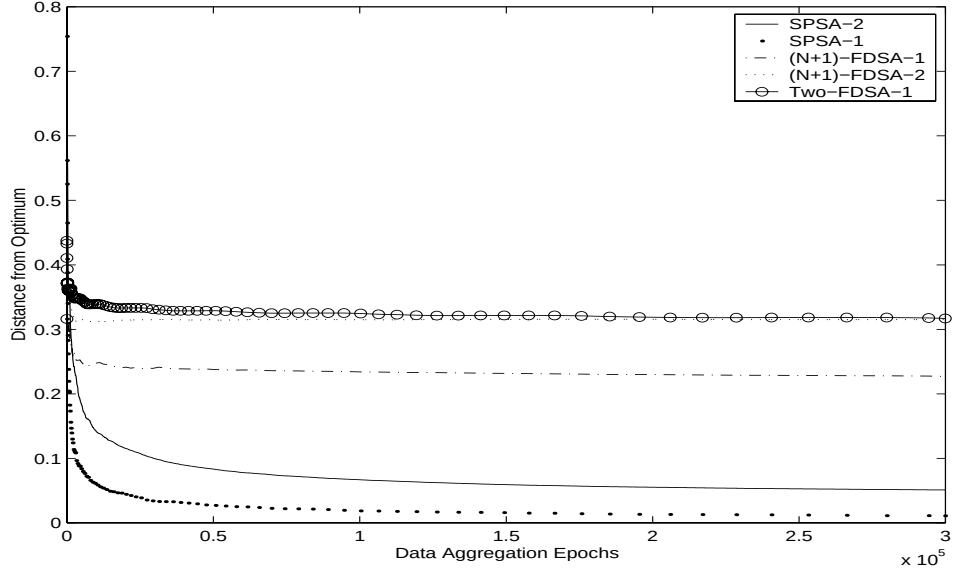


Figure 2: 10-Parameter Vector

Table 2: Distance after 3×10^5 Data Aggregation Epochs

Algorithm	Mean $\pm$ Standard Error (10-Parameter Vector)	Mean $\pm$ Standard Error (40-Parameter vector)
SPSA-1	0.011 ± 0.001	0.121 ± 0.03
SPSA-2	0.051 ± 0.02	0.063 ± 0.02
$(N + 1)$ -FDSA-1	0.227 ± 0.03	0.492 ± 0.06
$(N + 1)$ -FDSA-2	0.315 ± 0.003	0.507 ± 0.05
Two-FDSA-1	0.317 ± 0.04	0.782 ± 0.05

On a Sun Ultra10 Unix workstation, for $M = 5$, it took about 2 minutes using SPSA-1 for the Euclidean distance from optimum to become less than 0.05. SPSA-2 required about 3-4 minutes for the same. On the other hand, for $M = 20$, SPSA-1 took about 15 minutes, while SPSA-2 required only about 5 minutes for the same to happen. $(N + 1)$ -Simulation FDSA-1 and $(N + 1)$ -Simulation FDSA-2 (along with Two-Simulation FDSA-1) took orders of magnitude more time than SPSA-1 and SPSA-2. For $M = 20$, after almost 6.5×10^7 data aggregation epochs and running for nearly 21 hours, the separation from optimum of $(N + 1)$ -Simulation FDSA-1 was about 0.44, while that of $(N + 1)$ -Simulation FDSA-2 was about 0.42. As expected, Two-Simulation FDSA-1 showed the worst performance. For $M = 20$, after $8\frac{20}{40}$ parameter updates (or nearly 5×10^8 data aggregation epochs) and running for almost 18 hours, the Euclidean distance from optimum using Two-Simulation FDSA-1 was still about 0.70. Thus, our simulation experiments confirm that both algorithms SPSA-1 and SPSA-2 presented here, perform orders of magnitude faster than the

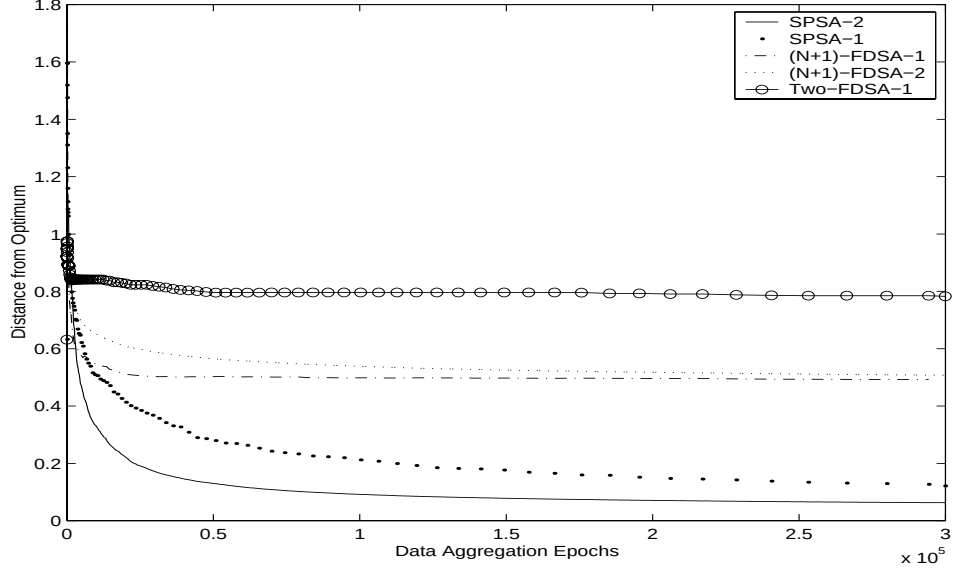


Figure 3: 40-Parameter Vector

algorithms $(N + 1)$ -Simulation FDSA-1, Two-Simulation FDSA-1 and $(N + 1)$ -Simulation FDSA-2 of [4] and [5].

7 Conclusions

We developed two simultaneous perturbation stochastic approximation (SPSA) algorithms (SPSA-1 and SPSA-2) for simulation optimization of hidden Markov models. Both of these algorithms use only two parallel simulations (each) and are generalized variants of the two-timescale stochastic approximation algorithms of [4] and [5] ($(N + 1)$ -Simulation FDSA-1, Two-Simulation FDSA-1 and $(N + 1)$ -Simulation FDSA-2) respectively. Whereas SPSA-1 updates the parameter over time instants of increasing separation as in [4], SPSA-2 updates once after every fixed number of instants. The latter is a generalization of the algorithm in [5] that updates the parameter at every instant; and shows improved performance. The convergence analysis for both the algorithms was presented. We conducted numerical experiments with parameters of different dimensions on a two node queueing network model with feedback using both the SPSA algorithms, the Two-Simulation algorithm of [4] and its $(N + 1)$ -Simulation analogue and the $(N + 1)$ -Simulation algorithm of [5]. We found that the SPSA algorithms converge orders of magnitude faster than the rest.

References

- [1] S. Andradóttir. *Simulation Optimization*. Chapter 9 in Handbook of Simulation, ed. J. Banks, Wiley, 1998.
- [2] J. D. Bartusek and A. M. Makowski. On stochastic approximation driven by sample averages: convergence results via the ODE method. Technical Report, Institute for Systems Research, University of Maryland, http://www.isr.umd.edu/TechReports/ISR/1994/TR_94-4/, 1994.
- [3] S. Bhatnagar. Multiscale stochastic approximation schemes with applications to ABR service in ATM networks. Doctoral dissertation, Dept. of Electrical Engineering, Indian Institute of Science, Bangalore, India, July 1997.
- [4] S. Bhatnagar and V. S. Borkar. Multiscale stochastic approximation for parametric optimization of hidden Markov models. *Probability in the Engineering and Informational Sciences*, 11:509–522, 1997.
- [5] S. Bhatnagar and V. S. Borkar. A two time scale stochastic approximation scheme for simulation based parametric optimization. *Probability in the Engineering and Informational Sciences*, 12:519–531, 1998.
- [6] S. Bhatnagar, M. C. Fu, and S. I. Marcus. Optimal multilevel feedback policies for ABR flow control using two timescale SPSA. Technical Report, Institute for Systems Research, University of Maryland, http://www.isr.umd.edu/TechReports/ISR/1999/TR_99-18/ (shortened version submitted for journal publication), 1999.
- [7] S. Bhatnagar, M. C. Fu, and S. I. Marcus. *Two timescale SPSA algorithms for rate-based ABR flow control*. Chapter 27 in System Theory: Modeling, Analysis and Control, ed. T.Djaferis and I.Schick, Kluwer Academic, 1999.
- [8] V. S. Borkar. On white noise representations in stochastic realization theory. *SIAM J. Control and Optim.*, 31:1093–1102, 1993.
- [9] V. S. Borkar. *Probability Theory: An Advanced Course*. Springer Verlag, New York, 1995.
- [10] V. S. Borkar. Stochastic approximation with two time scales. *Systems and Control Letters*, 29:291–294, 1997.
- [11] H. F. Chen, T. E. Duncan, and B. P.-Duncan. A Kiefer-Wolfowitz algorithm with randomized differences. *IEEE Trans. Autom. Contr.*, 44(3):442–453, 1999.
- [12] E. K. P. Chong and P. J. Ramadge. Optimization of queues using an infinitesimal perturbation analysis - based stochastic algorithm with general update times. *SIAM J. Contr. and Optim.*, 31(3):698–732, 1993.
- [13] E. K. P. Chong and P. J. Ramadge. Stochastic optimization of regenerative systems using infinitesimal perturbation analysis. *IEEE Trans. on Autom. Contr.*, 39(7):1400–1410, 1994.

- [14] R. J. Elliott, L. Aggoun, and J. B. Moore. *Hidden Markov Models: Estimation and Control*. Springer-Verlag, New York, 1995.
- [15] M. C. Fu. Convergence of a stochastic approximation algorithm for the $GI/G/1$ queue using infinitesimal perturbation analysis. *J. Optim. Theo. Appl.*, 65:149–160, 1990.
- [16] M. C. Fu. Optimization via simulation: a review. *Annals of Oper. Res.*, 53:199–248, 1994.
- [17] M. C. Fu and S. D. Hill. Optimization of discrete event systems via simultaneous perturbation stochastic approximation. *IIE Trans.*, 29(3):233–243, 1997.
- [18] M. W. Hirsch. Convergent activation dynamics in continuous time networks. *Neural Networks*, 2:331–349, 1987.
- [19] Y. C. Ho and X. R. Cao. *Perturbation Analysis of Discrete Event Dynamical Systems*. Kluwer, Boston, 1991.
- [20] H. J. Kushner and D. S. Clark. *Stochastic Approximation Methods for Constrained and Unconstrained Systems*. Springer Verlag, New York, 1978.
- [21] H. J. Kushner and F. J. Vazquez-Abad. Stochastic approximation methods for systems over an infinite horizon. *SIAM J. Contr. and Optim.*, 34(2):712–756, 1996.
- [22] H. J. Kushner and G. G. Yin. *Stochastic Approximation Algorithms and Applications*. Springer Verlag, New York, 1997.
- [23] P. L’Ecuyer, N. Giroux, and P. W. Glynn. Stochastic optimization by simulation: Numerical experiments with the $M/M/1$ queue in steady-state. *Management Science*, 40(10):1245–1261, 1994.
- [24] P. L’Ecuyer and P. W. Glynn. Stochastic optimization by simulation: Convergence proofs for the $GI/G/1$ queue in steady state. *Management Science*, 40(11):1562–1578, 1994.
- [25] M. Loève. *Probability Theory, vol. 2, 4th ed.* Springer Verlag, New York, 1977.
- [26] J. Neveu. *Discrete Parameter Martingales*. North Holland, Amsterdam, 1975.
- [27] G. C. Pflug. *Optimization of Stochastic Models: The Interface Between Simulation and Optimization*. Kluwer Academic, 1996.
- [28] J. C. Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Trans. Autom. Cont.*, 37(3):332–341, 1992.
- [29] F. J. Vazquez-Abad and H. J. Kushner. Estimation of the derivative of a stationary measure with respect to a control parameter. *J. Appl. Prob.*, 29:343–352, 1992.