Model-Free Control of Nonlinear Stochastic Systems with Discrete-Time Measurements

James C. Spall, *Senior Member, IEEE*, and John A. Cristion

***Abstract*—Consider the problem of developing a controller for general (nonlinear and stochastic) systems where the equations governing the system are unknown. Using discrete-time measurements, this paper presents an approach for estimating a controller without building or assuming a model for the system (including such general models as differential/difference equations, neural networks, fuzzy logic rules, etc.). Such an approach has potential advantages in accommodating complex systems with possibly time-varying dynamics. Since control requires some mapping, taking system information, and producing control actions, the controller is constructed through use of a function approximator (FA) such as a neural network or polynomial (no FA is used for the unmodeled system equations). Creating the controller involves the estimation of the unknown parameters within the FA. However, since no functional form is being assumed for the system equations, the gradient of the loss function for use in standard optimization algorithms is not available. Therefore, this paper considers the use of the simultaneous perturbation stochastic approximation algorithm, which requires only system measurements (not a system model). Related to this, a convergence result for stochastic approximation algorithms with time-varying objective functions and feedback is established. It is shown that this algorithm can greatly enhance the efficiency over more standard stochastic approximation algorithms based on finite-difference gradient approximations.**

***Index Terms*— Direct adaptive control, gradient estimation, nonlinear systems, simultaneous perturbation stochastic approximation.**

I. Introduction

ADAPTIVE control procedures have been developed in a variety of areas for controlling systems with imperfect information about the system (e.g., manufacturing process control, robot arm manipulation, aircraft control, etc.). Such procedures are typically limited by the need to assume that the forms of the system equations are known (and usually linear) while the parameters may be unknown. In complex physical, socioeconomic, or biological systems, however, the forms of the system equations (typically nonlinear) are often unknown as well as the parameters, making it impossible to determine the control law needed in existing adaptive control procedures. This provides the motivation for developing a control procedure that does not require a model for the underlying system. In this way, we expand the range of problems for which "formal" automatic control methods can apply. It is obvious that one should not be limited by customary model-based approaches given the wide range of successful "model-free" controllers in nature. Humans, for example, have little problem solving certain control problems that would vex even the must sophisticated model-based automatic controllers.

By definition, an automatic controller requires some function mapping that takes current (and maybe past) information about the system and produces control actions to affect future system performance. The approach here uses the system measurements to determine the control function without the need to estimate or assume a separate model for the system. The approach here is based on using a function approximator (FA) to represent the controller (no FA—or other mapping such as fuzzy logic rules base—is used for the system). Associated with any FA will be a set of parameters that must be determined, which will be one of the key aspects of this paper. Popular FA's include, for example, polynomials, multilayered feed-forward or recurrent neural networks, splines, wavelet networks, and trigonometric series. By results such as the well-known Stone–Weierstrass approximation theorem (e.g., Rudin [38, pp. 146–153, 176, 205]), it can be shown that many of these techniques share the property that they can be used to approximate any continuous function to any degree of accuracy (of course, this is merely an existence result, so experimentation must usually be performed in practice to ensure that the desired level of accuracy with a given FA is being achieved). Each FA technique tends to have advantages and disadvantages, some of which are discussed in Poggio and Girosi [36], Lane *et al.* [22], and Chen and Chen [8] (e.g., polynomials have a relatively easy physical interpretability, but the number of parameters to be determined grows rapidly with input dimension or polynomial order). Since the approach of this paper is generic, the methods will be presented without regard to which type of FA is to be used, although we will demonstrate the approach using polynomials and neural networks.

Others have considered the problem of developing controllers based on FA's when there is minimal information about the system equations. The majority of such techniques are indirect control methods in the sense that a second FA is introduced to model the open-loop behavior of the system. This open-loop FA is typically determined in a system identification process from sample input–output data on the system prior to operating the system in closed-loop and constructing the controller FA (with neural networks as the FA's; see, e.g., Narendra and Parthasarathy [31], Pao *et al.* [34], or Sartori

Manuscript received November 8, 1996; revised November 14, 1997. Recommended by Associate Editor, E. Yaz. This work was supported in part by the JHU/APL IRAD Program and the U.S. Navy under Contract N00024-98-D-8124.

The authors are with Johns Hopkins University, Applied Physics Laboratory, Laurel, MD 20723-6099, USA (e-mail: james.spall@jhuapl.edu).

Publisher Item Identifier S 0018-9286(98)05809-7.

and Antsaklis [45]). In contrast, the direct control approach here does not require any open-loop system identification, instead constructing the one (controller) FA while the system is operating in closed-loop. Thus we avoid the need for open-loop "training" data, which may be difficult or expensive to obtain, and the need to estimate perhaps twice as many parameters (for constructing two FA's). Further, the approach here offers potential advantages in being better able to handle changes in the underlying system dynamics (since it is not tied to a prior system model) and being more robust in the face of widely varying control inputs (i.e., the indirect approach may perform poorly for closed-loop controls outside of the range of open-loop controls used in the prior identification step).

Let us briefly discuss how the approach here contrasts with other "model-free" approaches. We are aware of many claims made in the literature that a particular control technique is "model-free." However, we question most of these claims on the basis of the hidden or implicit system modeling required. (We wish to raise this issue not because there is anything inherently wrong with these other approaches, but to help clarify that our use of "model-free" is to be taken literally.) For example, fuzzy controllers are frequently claimed as model-free; nevertheless, in all fuzzy controllers there is a requirement for a rules base (or associative memory matrix) that describes the dynamics of the system in a linguistic-type fashion. Although such information is not in the classical form of differential or difference equations, it is still a representation of the dynamics that seems to qualify as a model. Similar arguments can be made for other controllers claimed as model-free (e.g., some neural network controllers).

Although the model-free approach here is appropriate for many practical systems, it is generally inappropriate for systems where a reliable system model can be determined. One reason, of course, is that with a reliable model, the controller will generally achieve optimal control more quickly (fewer suboptimal "training" steps). Further, a reliable model allows in some cases for theoretical analysis of such issues as stability and controllability and for the calculation of state estimates for use in system performance monitoring and feedback to the controller. (We say "in some cases" because in the stochastic discrete-time setting, there are currently almost no practically useful stability results for adaptive nonlinear systems.) Sanner and Slotine [43], Levin and Narendra [23], [24], Jagannathan *et al.* [16], Fabri and Kadirkamanathan [13], and Ahmed and Anjum [1] are examples of approaches that rely on controller FA's but introduce stronger modeling assumptions (e.g., deterministic systems or specific knowledge of how the controller enters the system dynamics) as a means of performing a stability analysis. However, for systems where only a flawed (if any) model is available, attempts to do such analysis can lead to suboptimal (or worse) controllers and faulty stability and controllability analysis. It is such cases that are of interest here.

As we will show, it is not possible in our model-free framework to obtain the derivatives necessary to implement standard gradient-based search techniques (such as back-propagation) for estimating the unknown parameters of the FA. We will, therefore, consider stochastic approximation (SA) algorithms based only on measurements of the system as it operates in closed-loop. Usually such algorithms rely on well-known finite-difference approximations to the gradient (for examples of such algorithms in control, see Saridis [44, pp. 375–376] or Bayard [2]). The finite-difference approach, however, can be very costly in terms of the number of system measurements required, especially in high-dimensional problems such as estimating an FA parameter vector, (which may easily have dimension of order 10^2 or 10^3). Further, real-time implementation of finite-difference methods would often suffer since the underlying system dynamics may change during the relatively long period in which measurements are being collected for one gradient approximation (see Section IV here). We will, therefore, consider an SA algorithm based on a "simultaneous perturbation" method (Spall [46]), which is typically much more efficient than the finite-difference SA algorithms in the amount of data required. In particular, the simultaneous perturbation approximation requires only one or two system measurements versus the hundreds (or more) typically required in finite-difference methods. A special case of the control approach here—focusing on the "direct approximator" method (see Section II below), perfect state measurements, and a neural network as the FA—is considered in Spall and Cristion [52]. Some applications of the simultaneous perturbation optimization method in control are given in Maeda and De Figueiredo [26] (robotics), Koch *et al.* [17] (integrated transit/traffic control), and Nechyba and Xu [32] (human-machine interface control). Note that the general convergence result presented here is relevant to most of these applications.

The remainder of this paper is organized as follows. Section II describes two related methods (based on different levels of prior information about the system) for using FA's to control nonlinear systems. This section also describes why it is not possible to determine the gradient of the loss function, in contrast to the approaches of Narendra and others mentioned above where they introduce an additional FA to model the open-loop system. Section III discusses the SA approach to FA parameter estimation using the simultaneous perturbation method and presents a theoretical result on the convergence of the estimate. Section IV presents numerical studies on two different nonlinear systems, and Section V offers some concluding remarks and areas for further investigation.

II. Overview of Approach to Control Without System Model

A. The System and Generic Form of Controller

We consider general dynamic processes, typically involving nonlinear dynamics and stochastic effects. It is assumed that a sequence of discrete-time measurements of the process is available and that the goal is to choose a corresponding sequence of controls to optimize a function of future system measurements. We let the sequence of discrete-time measurements be

$$y_1, y_2, y_3, \cdots, \tag{1a}$$

with corresponding controls

$$u_0, u_1, u_2, \cdots, \tag{1b}$$

(so u_k affects y_{k+1}, y_{k+2}, etc.). In general, we assume no particular analytical structure (e.g., state-space, NARMAX, continuous- or discrete-time process evolution, etc.) behind the process generating the measurements. Based on information contained within measurements and controls up to y_k and u_{k-1}, our goal is to choose a control u_k in a manner such that we minimize some loss function related to the next measurement y_{k+1} (or to the next specified number of measurements). Often, this loss function will be one that compares y_{k+1} against a target value t_{k+1}, penalizing deviations between the two. Without sacrificing generality, most of this paper will focus on this target-tracking control problem, although the approach would apply in other (say, optimal control) problems as well, as discussed briefly in Section II-B.[1]

In the approach here, a function approximator (e.g., neural network or polynomial) will be used to produce the control u_k, as outlined in Section I. We will consider an FA of fixed structure across time (e.g., the number of layers and nodes in a neural network is fixed), but allow for underlying parameters in the FA (e.g., connection weights in a neural network) to be updated. Since past information will serve as input to the control, this fixed structure requires that u_k be based on only a fixed number of previous measurements and controls (in contrast to using all information available up to time k for each k). Suppose our "sliding window" of previous information available at time k, say I_k, contains M previous measurements and N previous controls; akin to Parisini and Zoppoli [35], the choice of M and N reflects a tradeoff between carrying along a large quantity of potentially relevant information and the corresponding requirements for a more complex FA. Thus, when the system is operating without the FA parameters being updated, we have from (1a) and (1b)

$$I_k = \{y_k, y_{k-1}, \cdots, y_{k-M+1}; u_{k-1}, u_{k-2}, \cdots, u_{k-N}\}$$

(when the FA parameters are being updated, the set I_k of M and N previous measurements and controls will contain "test" values, as discussed in Section III-A).

We will consider two methods to the problem of a controlling system in the face of uncertainty about the system dynamics governing the evolution of sequence (1a), as illustrated in Fig. 1(a) and 1(b) for the target-tracking problem in the important special case where $I_k = \{y_k\}$ (for the more general definition of I_k, as given above, the figures would be modified in an obvious way; in addition, the process may include direct-state feedback, which is not shown). In the direct approximation method of Fig. 1(a), the output of the FA will correspond directly to the elements of the u_k vector, i.e., the inputs to the FA will be I_k ($=y_k$ here) and t_{k+1} and the output will be u_k. This approach is appropriate when there is no known analytical structure generating the measurements. In contrast, the self-tuning method of Fig. 1(b)

[1]The method here attempts to find the best controller given such characteristics as degree of controllability, relative number of elements in u_k, and y_{k+1}, etc. Since the system may inherently be less than fully controllable, perfect tracking may be unachievable in even deterministic systems.

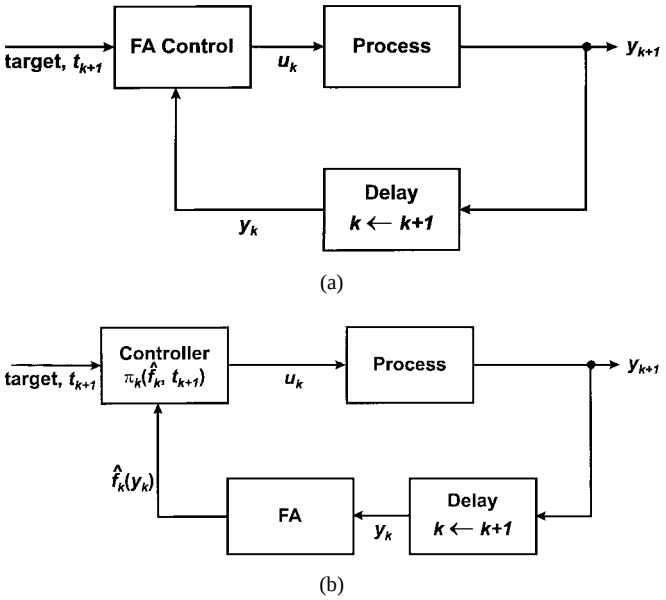

Fig. 1(a) Control system with FA as direct approximator to optimal u_k when $I_k = \{y_k\}$. (b) Self-tuning control system with FA as approximator to $f_k(I_k)$ when $u_k = \pi_k(f_k(I_k), t_{k+1})$ and $I_k = \{y_k\}$.

requires some prior information about this structure. In particular, it requires that enough information be available to write $u_k = \pi_k(f_k(I_k), t_{k+1})$, where $\pi_k(\cdot)$ is a known control law that depends on some unknown function $f_k(\cdot)$ that is approximated by the FA. As demonstrated in Section IV, a very important type of process to which this second method can apply is an affine-nonlinear (i.e., generalized bilinear) system such as that in Chen [7] and Dochain and Bastin [11]. As we will see in Section IV, when reliable prior information is available, the self-tuning method of Fig. 1(b) may yield a controller superior to the direct approximation method of Fig. 1(a).

B. Formulation of Estimation Problem for Determining FA

We now introduce some of the principles involved in determining the FA for use in generating the control u_k. Section III will provide a more detailed discussion on the estimation procedure and an associated convergence proof.

Associated with the FA-generating u_k will be a parameter vector θ_k that must be estimated (e.g., the connection weights in a neural network). Recall that we assume that the FA structure is fixed across time. Hence the problem of finding the optimum control function at time k is equivalent to finding the $\theta_k \in R^p$, p not a function of k, that minimizes some loss function $L_k(\theta_k)$ (and the optimal control value would be the output of u_k after the optimal θ_k has been found). A common loss is the one-step-ahead quadratic tracking error

$$L_k(\theta_k) = E\left[(y_{k+1} - t_{k+1})^T A_k (y_{k+1} - t_{k+1}) + u_k^T B_k u_k\right] \tag{2}$$

where A_k and B_k are positive semi-definite matrices reflecting the relative weight to put on deviations from the target and on the cost associated with larger values of u_k. The approach of this paper would also apply with nonquadratic

and/or nontarget-tracking loss functions. Such functions might arise, e.g., in constrained optimal control problems where we are trying to minimize some cost (without regard to a specific target value) and penalty functions or projections are used for certain values of y_{k+1} and/or u_k to reflect problem constraints (e.g., Sadegh [41]). For convenience, however, the remainder of the paper will illustrate points with the target-tracking problem exemplified by (2). Note that although (2) is a one-time-step error function, much of the adaptive control literature focuses on minimizing a loss function over an infinite horizon; Saridis [44, pp. 291–296] and Moden and Soderstrom [29] are two of a number of references that discuss the relationship between the two approaches. Note also that if the unconditional loss function $L_k(\theta_k)$ were replaced by a conditional loss as is sometimes seen in the control literature (e.g., (2) replaced by an expected tracking error conditional on previous measurements and controls), the same optimal θ_k would typically result. This follows since under standard conditions justifying the interchange of a derivative and an integral (e.g., Fleming [14, pp. 237–239]) $\partial L_k^{\text{cond}}/\partial\theta_k = 0$ implies $E(\partial L_k^{\text{cond}}/\partial\theta_k) = \partial E(L_k^{\text{cond}})/\partial\theta_k = \partial L_k/\partial\theta_k = 0$ at the optimal θ_k, where L_k^{cond} represents the conditional loss.

With a control of the form in Fig. 1(a) or (b), the problem of minimizing $L_k(\theta_k)$ implies that for each k we are seeking a (minimizing) solution θ_k^* to

$$g_k(\theta_k) \equiv \frac{\partial L_k}{\partial \theta_k} = \frac{\partial u_k^T}{\partial \theta_k} \cdot \frac{\partial L_k}{\partial u_k} = 0.$$

Since the functions governing the system are incompletely known, the term $\partial L_k/\partial u_k$ is not generally computable. Hence, $g_k(\theta_k)$ is not generally available in either of the methods in Fig. 1(a) and (b).[2] Thus the standard gradient descent algorithm (e.g., back-propagation—see, Narendra and Parthasarathy [31]), or any other algorithm involving $g_k(\theta_k)$ or a direct noisy measurement of $g_k(\theta_k)$, is not feasible.

Because gradient-descent-type algorithms are not generally feasible in the model-free setting here, we consider a stochastic approximation (SA) algorithm of the form

$$\hat{\theta}_k = \hat{\theta}_{k-1} - a_k(\text{gradient approx.})_k \tag{3}$$

to estimate $\{\theta_k\}$, where $\hat{\theta}_k$ denotes the estimate at the given iteration, $\{a_k\}$ is a scalar gain sequence satisfying certain regularity conditions, and the gradient approximation is such that it does not require full knowledge of the form of the process equations. The next section is devoted to describing in more detail the gradient-free SA approach to this problem.

[2]This contrasts with implementations of so-called indirect feedback controllers (e.g., Narendra and Parthasarathy [31, Sec. 6]), where a separate FA is used to model the unknown system dynamics and the identification and adaptive control are performed as if the FA model was identical in structure to the true system dynamics. One special case where $g_k(\theta_k)$ can be computed is in the self-tuning setting of Fig. 1(b) where $u_k(\cdot)$ is known to enter the process addictively (since $\partial L_k/\partial u_k$ then does not depend on unknown dynamics); such additive control models are considered, e.g., in Sanner and Slotine [43] for continuous time and Jagannathan *et al.* [16] for discrete time. Of course, in the more general setting of direct approximation control [Fig. 1(a)] $g_k(\theta_k)$ would still be unavailable since such adaptivity is not assumed known.

III. Parameter Estimation by Simultaneous Perturbation Stochastic Approximation: Implementation and Convergence

This section is divided into three sections. The first gives a summary of how simultaneous perturbation SA (SPSA) is used in implementing the control strategies of Fig. 1(a) and (b). The second section establishes conditions under which the FA parameter estimates from SPSA converge to the optimal weight values for the given structure of the FA. The final section provides some comments on the regularity conditions of the convergence result and considers the setting where there is no asymptotically unique (time-invariant) optimal parameter vector, as would often occur, say, when the underlying system has nonstationary dynamics.

A. Overview of the Approach

Recall that we are seeking the FA parameter vector at each time point that minimizes $L_k(\theta_k)$, i.e., we are seeking a θ_k^* such that $g_k(\theta_k^*) = 0$. Recall also that since gradient-based search algorithms are not applicable, we will consider a gradient-free SA-based approach.

Spall [46] gives a detailed analysis of the SPSA approach to optimization in the classical setting of a time-invariant loss function $L(\cdot)$ and corresponding fixed minimum. It is shown that the SPSA algorithm has the usual almost sure (a.s.) convergence and asymptotic normality properties of finite-difference SA (FDSA) algorithms of the Kiefer–Wolfowitz form but that the asymptotic normality result indicates that SPSA can achieve the same level of asymptotic accuracy as FDSA with only $1/p$ the number of system measurements in many practical problems. This is of particular interest in FA's for nonlinear, multivariate problems since p can easily be on the order of 10^2 or 10^3. Of course, in the control setting here the loss function $L_k(\cdot)$ is generally time-varying, and hence it cannot be automatically assumed that the results of Spall [46] apply. We, therefore, will present conditions under which the SPSA estimation error $\hat{\theta}_k - \theta_k^*$ converges a.s. to zero as in the time-invariant loss setting. Unfortunately, it does not appear possible to similarly produce an asymptotic normality result for the time-varying $L_k(\cdot)$ setting, which would provide a formal proof that asymptotically SPSA achieves the same level of accuracy as FDSA with only $1/p$th the number of measurements for the gradient approximations. This follows from the fact that the limiting mean and variance of the asymptotic normal distribution in Spall [46] are based on the (fixed) values of the second and third derivatives of the loss function and, of course, such fixed values do not exist in the time-varying setting here. Nevertheless, we feel that the general advantage of SPSA in the fixed loss function case together with the a.s. convergence of SPSA established below for the time-varying $L_k(\cdot)$ setting provide ample theoretical evidence of the advantage of SPSA over the standard FDSA approach in this control problem. This will be augmented by empirical evidence in Section IV.

In line with (3), the SPSA algorithm has the form

$$\hat{\theta}_k = \hat{\theta}_{k-1} - a_k \hat{g}_k(\hat{\theta}_{k-1}) \tag{4}$$

where $\hat{g}_k(\hat{\theta}_{k-1})$ is the simultaneous perturbation approximation to $g_k(\hat{\theta}_{k-1})$. Although several variations are possible (see below), the core approximation is such that the ℓth component of $\hat{g}_k(\hat{\theta}_{k-1})$, $\ell = 1, 2, \cdots, p$, is given by

$$\hat{g}_{k\ell}(\hat{\theta}_{k-1}) = \frac{\hat{L}_k^{(+)} - \hat{L}_k^{(-)}}{2c_k\Delta_{k\ell}} \tag{5}$$

where

- $\hat{L}_k^{(\pm)}$ are estimated values of $L_k(\hat{\theta}_{k-1} \pm c_k\Delta_k)$ using the observed $y_{k+1}^{(\pm)}$ and $u_k^{(\pm)}$, e.g., for $L_k(\theta_k)$ as in (2), $\hat{L}_k^{(\pm)} = (y_{k+1}^{(\pm)} - t_{k+1})^T A_k (y_{k+1}^{(\pm)} - t_{k+1}) + u_k^{(\pm)T} B_k u_k^{(\pm)}$.
- $u_k^{(\pm)}$ are controls based on an FA with parameter vector $\theta_k = \hat{\theta}_{k-1} \pm c_k\Delta_k$, where $\Delta_k = (\Delta_{k1}, \Delta_{k2}, \cdots, \Delta_{kp})^T$ is a random vector. Usually, the $\{\Delta_{ki}\}$ are independent, bounded, symmetrically distributed (about zero) random variables $\forall k, i$, identically distributed at each k, with $E(\Delta_{ki}^{-2})$ uniformly bounded $\forall k, i$, although the conditions for convergence below are stated more generally (see Section III-C for comments on regularity conditions).
- $y_{k+1}^{(\pm)}$ are measurements based on $u_k^{(\pm)}$.
- $\{c_k\}$ is a sequence of positive numbers satisfying certain regularity conditions (typically $c_k \to 0$ or $c_k = c \forall k$, depending on whether the system equations are stationary or nonstationary, as discussed in Sections III-B and III-C).

The key fact to observe is that at any iteration only two measurements are needed to approximate $g_k(\cdot)$ (i.e., the numerators in $\hat{g}_k(\cdot)$ are the same for all p components, reflecting the simultaneous perturbation about all p elements in $\hat{\theta}_{k-1}$). This is in contrast to the standard FDSA approach where $2p$ measurements are needed (i.e., for the ℓth component of the gradient approximation, the quantity Δ_k is replaced by a vector with a positive constant in the ℓth place and zeroes elsewhere; see, e.g., Ruppert [39]). A variation on the gradient approximation in (5) is to average several gradient approximations, with each vector in the average being based on a new (independent) value of Δ_k and a corresponding new pair of measurements; this may enhance the performance of the algorithm in a high-noise setting as discussed in Spall [46] and Spall and Cristion [52], even at the expense of the additional loss function evaluations. A further variation on (5) is to smooth the gradient approximation across time by a weighted average of the previous and current gradient estimates (analogous to the "momentum" approach in backpropagation); such smoothing can sometimes improve the performance of the algorithm (see Spall and Cristion [51] for a thorough discussion of smoothing in SPSA-based direct adaptive control).

A slightly more fundamental modification is to replace the two-measurement gradient approximation in (5) with the one-measurement form

$$\hat{g}_{k\ell}(\hat{\theta}_{k-1}) = \frac{\hat{L}_k^{(+)}}{c_k\Delta_{k\ell}} \tag{6}$$

as discussed in Spall [47]. Although [47] shows that (5) remains generally preferable to (6) in terms of overall efficiency of optimization based on loss function measurements (even though (5) uses twice the number of L_k measurements), (6) has advantages in highly nonstationary systems. This follows from the relationship of (5) or (6) to the underlying gradient $g_k(\hat{\theta}_{k-1})$: if the dynamics change significantly, (5) may be a poor approximation to the gradient, while the instantaneous approximation (6) always provides a quantity that [to within $O(c_k^2)$] is an unbiased estimate of the gradient. A guideline for when one should consider the instantaneous form (6) is when condition C3) from the proposition is not satisfied for (5); this condition is a stochastic analogue of a well-known condition for nonlinear optimization. Thus, although the focus here is on the "standard" SP gradient approximation in (5), the closely related one-measurement form in (6) may be preferable in some cases.

There are several ways in which a practical control strategy can be developed using the SPSA algorithm in (4) and (5) [with obvious analogues for (4) and (5)]. These differences result from whether or not it is desired to produce a "nominal" y_{k+1} based on a control with updated $\theta_k = \hat{\theta}_k$ (only $y_{k+1}^{(\pm)}$ are required in implementing (4) and (5), which use $\theta_k = \hat{\theta}_{k-1} \pm c_k\Delta_k$) and whether or not it is possible to reset the system from a given state value to the previous state value. Spall and Cristion [52] discuss these strategies in greater detail for neural network-based control when one has direct measurements of the states in a nonlinear state-space model; these strategies could be readily extended to the more general setting here. As an illustration of one of these strategies, suppose that we wish to produce a sequence of system measurements that includes nominal measurements, i.e., the sequence is $\{y_0, y_1^{(+)}, y_1^{(-)}, y_1, y_2^{(+)}, y_2^{(-)}, y_2, \cdots\}$, and that the system cannot be readily reset from one state to the previous state (which is the usual case). Then, as in Section II-A, each of $u_k^{(+)}$, $u_k^{(-)}$, and u_k are produced using the previous M measurements and N controls, which comprise the information sets $I_k^{(+)}$, $I_k^{(-)}$, and I_k (say), respectively (note that the elements of I_k here will differ from those in Section II-A where no parameter update is taking place). Thus, for example, if $M = N = 2$ and the most recent measurement is $y_{k+1}^{(+)}$, then the next control $u_k^{(-)}$ is based on $\theta_k = \hat{\theta}_{k-1} - c_k\Delta_k$, t_{k+1}, and $I_k^{(-)} = \{y_{k+1}^{(+)}, y_k, u_k^{(+)}, u_{k-1}\}$.

To keep the notation in this paper relatively simple, the discussion focuses on the case where θ is changed at every discrete-time point during the "training" process of building the controller. The same basic ideas can be applied when θ is changed less frequently (say, to allow transient effects to decay). An example of such a setting is the vehicle traffic control problem in Spall and Chin [50] where θ (i.e., the control function) is changed on a daily basis even though traffic-responsive control actions (the control function outputs) are changed much more frequently (perhaps minute by minute).

One must also pick an initial θ, $\hat{\theta}_0$ to initialize the controller. Random initialization is one possibility (see Section IV). However, in some real-world systems, it will be useful to exploit simulations and/or other prior information to provide a more precise initial θ. With a simulation, one could implement the control strategy above using the simulation as a proxy for the real system. The final trained value for θ could then serve as the initial θ for use with the real system (in the idealized

limiting case where the simulation was run for a large number of SPSA iterations and was a perfect representation of the real system, and where the real system was nonstationary, this initial θ would be the optimal $\theta = \theta^*$ based on the convergence theory in Section III-B). If historical data are available on "reasonable" controls and responses, then one can use standard offline optimization (e.g., back-propagation) to train θ so that $u_0(\cdot)$ is a control function that can effectively reproduce the historical input–outputs. So if the historical data represented a prior suboptimal control strategy, the SPSA approach would allow for a graceful transition to a more nearly optimal controller.

B. The Convergence of the Weight Estimate $\hat{\theta}_k$

Let us now present conditions such that, as in the $L_k(\cdot) = L(\cdot)$ case of Spall [46], [47], $\hat{\theta}_k$ will converge a.s. for the case of varying $L_k(\cdot)$. The proposition below applies to either the two- or one-measurement approximation form in (5) or (6). Note that general convergence results such as in Benveniste *et al.* [4, Part II] do not directly apply here due to the time-varying loss function and underlying system evolution. Unlike Spall [46], the connection to the true gradient is not used explicitly in the regularity conditions here [this is especially relevant when (5) is used since the system evolution between $\hat{L}_k^{(+)}$ and $\hat{L}_k^{(-)}$ complicates the interpretation of $\hat{g}_k(\hat{\theta}_{k-1})$ as an estimate of $g_k(\hat{\theta}_{k-1})$]; hence, certain conditions related to $\hat{g}_k(\hat{\theta}_{k-1})$ being a nearly unbiased estimator of the gradient (e.g., mean zero on the measurement noise difference) are not used here, while other conditions imposing "gradient-like" behavior are imposed [C3) and C4)]. Section III-C provides some discussion on the conditions relative to the control problem here. Note that having the optimal parameters converge (i.e., $\theta_k^* \to \theta^*$) does not imply the (say) pointwise convergence of $L_k(\cdot)$ to some fixed $L(\cdot)$; in fact $L_k(\theta_k^*)$ may be perpetually varying even when $\theta_k^* = \theta^* \forall k$, as discussed in Section III-C below. Also note that the result below differs considerably from the convergence result in Spall and Cristion [51], which requires stronger conditions (e.g., uniformly bounded increments for the iterates) and is oriented to the "smoothed" SP gradient approximation. We let $\|\cdot\|$ denote any vector norm, i.o., represent "infinitely often," $(\theta^*)_i$ and $(\theta - \theta^*)_i$ represent the ith components of the indicated vectors (notation chosen to avoid confusion with time subscript k), and

$$\bar{g}_k(\hat{\theta}_{k-1}) = E(\hat{g}_k(\hat{\theta}_{k-1}) \mid \hat{\theta}_{k-1}).$$

C1) $a_k, c_k > 0\ \forall k$; $a_k \to 0$, $c_k \to 0$ as $k \to \infty$; $\sum_{k=1}^{\infty} a_k = \infty$, $\sum_{k=1}^{\infty} (a_k/c_k)^2 < \infty$.

C2) Δ_k is symmetrically distributed about 0 and for some $\rho > 0$ and $\forall k, \ell$, $E[(\hat{L}_k^{(+)}/\Delta_{k\ell})^2] \le \rho$.

C3) For some $K < \infty$ any $\rho > 0$ and for each $k \ge K$ suppose that if $\|\theta - \theta^*\| \ge \rho$, there exists a $\delta_k(\rho) > 0$ such that $(\theta - \theta^*)^T \bar{g}_k(\theta) \ge \delta_k(\rho)$ where $\delta_k(\rho)$ satisfies $\sum_{k=1}^{\infty} a_k \delta_k(\rho) = \infty$.

C4) For each $i = 1, 2, \cdots, p$, and any $\rho > 0$, $P(\{\bar{g}_{ki}(\hat{\theta}_{k-1}) \ge 0 \text{ i.o.}\} \cap \{\bar{g}_{ki}(\hat{\theta}_{k-1}) < 0 \text{ i.o.}\} \mid \{|\hat{\theta}_{ki} - (\theta^*)_i| \ge \rho\ \forall k\}) = 0$.

C5) For any $\tau < 0$ and nonempty $S \subseteq \{1, 2, \cdots, p\}$, there exists a $\rho'(\tau, S) > \tau$ such that

$$\limsup_{k \to \infty} \left| \frac{\sum_{i \notin S} (\theta - \theta^*)_i \bar{g}_{ki}(\theta)}{\sum_{i \in S} (\theta - \theta^*)_i \bar{g}_{ki}(\theta)} \right| < 1 \text{ a.s.}$$

for all $|(\theta - \theta^*)_i| < \tau$ when $i \notin S$ and $|(\theta - \theta^*)_i| \ge \rho'(\tau, S)$ when $i \in S$.

Proposition: Let conditions C1) through C5) hold, and suppose there exists a θ^* such that $\theta_k^* \to \theta^*$ as $k \to \infty$. Then

$$\hat{\theta}_k - \theta^* \to 0 \text{ a.s.} \tag{7}$$

Proof: The proof will proceed in three parts. First we will show that $\tilde{\theta}_k \equiv \hat{\theta}_k - \theta^*$ does not diverge in magnitude to ∞ on any set of nonzero measure. Second, we will show that $\tilde{\theta}_k$ converges a.s. to some random vector; third we show that this random vector is the constant zero, as desired.

Part 1: Letting $M_k = a_k \bar{g}_k(\hat{\theta}_{k-1})$ and $M_k' = a_k(\hat{g}_k(\hat{\theta}_{k-1}) - \bar{g}_k(\hat{\theta}_{k-1}))$, we can write

$$\tilde{\theta}_{k+1} + \sum_{j=1}^{k} M_j = \tilde{\theta}_1 - \sum_{j=1}^{k} M_j'. \tag{8}$$

Since C1) and C2) imply that $\{\sum_{j=1}^{k} M_j'\}$ represents a martingale sequence (in k)

$$E\left\| \sum_{j=1}^{k} M_j' \right\|^2 \le \sum_{j=1}^{k} E\|M_j'\|^2 < \infty$$

where the finiteness follows from C1) and C2). Then by (8) and the martingale convergence theorem

$$\tilde{\theta}_{k+1} + \sum_{j=1}^{k} M_j \overset{\text{a.s.}}{\to} X \tag{9}$$

where X is some integrable random vector.

Let us now show that $P(\limsup_{k \to \infty} \|\tilde{\theta}_k\| = \infty) = 0$. Since the arguments below apply along any subsequence, we will for ease of notation replace the "lim sup" with "lim" without loss of generality. We will show that the event $\{\|\tilde{\theta}_k\| \to \infty\}$ has probability zero in a multivariate extension to scalar arguments in Blum [5] and Evans and Weber [12]. Furthermore, suppose that the limiting quantity of the unbounded elements in $\tilde{\theta}_k$ is $+\infty$ (trivial modifications cover a limiting quantity including $-\infty$ limits). For τ, S, and $\rho'(\tau, S)$ as in C5), the event of interest $\{\|\tilde{\theta}_k\| \to \infty\}$ can be represented as

$$\bigcup_{S} \{\tilde{\theta}_{ki} \to \infty\ \forall i \in S\}$$

$$\subseteq \bigcup_{\tau > 0, S} \left\{ \left\{ \{\tilde{\theta}_{ki} \ge \rho'(\tau, S)\ \forall i \in S, \tilde{\theta}_{ki} \le \tau\ \forall i \notin S, \right.\right.$$

$$\left.\left. k \ge K(\tau, S)\} \bigcap \limsup_{k \to \infty} \{M_{ki} < 0\ \forall i \in S\} \right\} \tag{10a}$$

$$\bigcup \left\{ \{\tilde{\theta}_{ki} \to \infty\ \forall i \in S\} \bigcap \liminf_{k \to \infty} \{M_{ki} < 0\ \forall i \in S\}^c \right\} \Bigg\} \tag{10b}$$

where $K(\tau,S)<\infty$ and the superscript c denotes set complement. We now analyze the two principal events shown in (10a) and (10b). For the event (10a), we know that there exists a subsequence $\{k_0,k_1,k_2,\cdots\}$ such that $\{\tilde{\theta}_{k_j i}\geq \rho'(\tau,S)\ \forall i\in S\}\cap\{M_{k_j i}<0\ \forall i\in S\}$ is true. But, from C5), this implies that $\tilde{\theta}_{k_j}^T\bar{g}_{k_j+1}(\hat{\theta_{k_j}})<0\ \forall j$ sufficiently large, contradicting C3). Hence the first event on the right-hand side of (10) has probability zero for any τ, S. Now consider the second principal event, as shown in (10b). From (9), we know that for almost all sample points, $\sum_{k=1}^{\infty}M_{ki}\to-\infty\ \forall i\in S$ must be true. But this implies from C4) that for no $i\in S$ can $M_{ki}\geq 0$ occur i.o. However, at each k, the event $\{M_{ki}<0\ \forall i\in S\}^c$ is composed of the union of $2^{\dim(S)}-1$ events, each of which has $M_{ki}\geq 0$ for at least one $i\in S$. This, of course, requires that $M_{ki}\geq 0$ i.o. for at least one $i\in S$, which creates a contradiction. Hence, the probability of the event in (10b) is zero for any S. Taking the union over τ,S [shown in (10a and (10b)] of the zero-probability events yields $P(\|\tilde{\theta}_k\|\to\infty)=0$, completing Part 1 of the proof.

Part 2: To show that $\tilde{\theta}_k$ converges a.s. to a unique (finite) limit, we show that

$$P\left(\liminf_{k\to\infty}\tilde{\theta}_{ki}<a<b<\limsup_{k\to\infty}\tilde{\theta}_{ki}\right)=0\qquad\forall i \tag{11}$$

for any $a<b$. There exist two subsequences, one with convergence to a point $<a$ and one with convergence to a point $>b$. From (9) and the conclusion of Part 1, each of these subsequences has a sub-subsequence $\{k_{j_l}\}$ such that

$$\limsup_{l\to\infty}\left|\sum_{k=1}^{k_{j_l}}M_{ki}\right|<\infty\ \text{a.s.} \tag{12}$$

Supposing that the event within the probability statement of (11) is true, we know from C4) and (9) that for any $\rho>0$ and corresponding sample point we can choose $m>n$ sufficiently large so that for each i and combined sub-subsequence (from both sub-subsequences mentioned above)

$$\left|\sum_{k=k_{j_n}}^{k_{j_{m-1}}}M_{ki}\right|\leq\rho \tag{13a}$$

$$\left|\tilde{\theta}_{k_{j_m}i}-\tilde{\theta}_{k_{j_n}i}+\sum_{k=k_{j_n}}^{k_{j_{m-1}}}M_{ki}\right|\leq\frac{b-a}{3} \tag{13b}$$

$$\tilde{\theta}_{k_{j_n}i}<a<b<\tilde{\theta}_{k_{j_m}i}. \tag{13c}$$

Picking $\rho<(b-a)/3$ implies by (13a) and (13b) that

$$\left|\tilde{\theta}_{k_{j_n}i}-\tilde{\theta}_{k_{j_m}i}\right|\leq 2(b-a)/3.$$

However, (13c) requires that

$$\tilde{\theta}_{k_{j_m}i}-\tilde{\theta}_{k_{j_n}i}>b-a$$

which is a contradiction. Hence, (11) holds, completing the proof of Part 2.

Part 3: Let us now show that the unique finite limit from Part 2 is zero. From (12) this follows if

$$P\left(\lim_{k\to\infty}\tilde{\theta}_k\neq 0,\left\|\sum_{k=1}^{\infty}M_k\right\|<\infty\right)=0. \tag{14}$$

Suppose the event in the probability of (14) is true and let $I\subseteq\{1,2,\cdots,p\}$ represent those indexes i such that $\tilde{\theta}_{ki}\not\to 0$ as $k\to\infty$. Then, by the convergence in Part 2 there exists for almost any sample point in the underlying sample space some $0<a<b<\infty$ and $K(a,b)<\infty$ (dependent on the sample point) such that $\forall k\geq K$, $0<a\leq|\tilde{\theta}_{ki}|\leq b<\infty$ when $i\in I$ $(I\neq\emptyset)$ and $|\tilde{\theta}_{ki}|<a$ when $i\in I^c$. From C3), and taking $\delta_k=\delta_k(a)$, it follows that

$$\sum_{k=K+1}^{n}a_k\sum_{i\in I}\tilde{\theta}_{k-1,i}\bar{g}_{ki}(\hat{\theta}_{k-1})\geq\sum_{k=K+1}^{n}a_k\delta_k. \tag{15}$$

But since C4) implies that $\bar{g}_{ki}(\hat{\theta}_{k-1})$ can change sign only a finite number of times (except possibly on a set of sample points of measure zero) and since $|\tilde{\theta}_{ki}|\leq b$, we know from (15) that for at least one $i\in I$

$$\limsup_{n\to\infty}\left|\frac{\sum_{k=K+1}^{n}a_k\delta_k}{\sum_{k=K+1}^{n}M_{ki}}\right|<\infty. \tag{16}$$

From C3), we have $\sum_{k=K+1}^{\infty}a_k\delta_k=\infty$. Then by (16) $|\sum_{k=K+1}^{\infty}M_{ki}|=\infty$. Since, for the $a<b$ above, there exists such a K for each sample point in a set of measure one (recalling that $\hat{\theta}_k$ converges a.s. by Part 2), we know from the above discussion that there also exists at least one $i\in I$ (i possibly dependent on the sample point) such that $|\sum_{k=K+1}^{\infty}M_{ki}|=\infty$. Since I has a finite number of elements, $|\sum_{k=1}^{\infty}M_{ki}|=\infty$ with probability greater than zero for at least one i . However, this is inconsistent with the event in (14), showing that the event does in fact have probability zero. This completes Part 3, which completes the proof. □

C. Comments on Regularity Conditions for Proposition and Extension to Perpetually Varying θ_k^*

This section provides some interpretation of the above regularity conditions and discusses the feasibility of a unique limiting θ^* existing together with some discussion on SA algorithms when θ_k^* is not converging to any fixed θ^*.

Condition C1) presents some standard conditions on the SA gains (as discussed below, however, in a system with nonstationary dynamics—where θ_k^* does not converge—it is generally best to not satisfy this condition). C2) ensures that the variability of $\hat{g}_k(\cdot)$ is not so large as to potentially cause divergence of the algorithm. Note that this condition is closely related to the important bounded inverse moments condition for $\Delta_{k\ell}$ in SPSA, since by Holder's inequality, C2) holds if $E[(\hat{L}_k^{(+)})^{2+\delta}]$ and $E[\Delta_{k\ell}^{-2-\delta'}]$ are both bounded for certain δ, $\delta'>0$. This bounded inverse moments condition prevents, e.g., taking $\Delta_{k\ell}$ as uniformly or normally distributed. However, taking $\Delta_{k\ell}$ as symmetrically Bernoulli-distributed to satisfy this condition has proven effective in our numerical studies, and, in fact, is shown in Sadegh and Spall [42] to be the optimal choice of distribution in static optimization

problems based on asymptotic principles. C3) ensures that if we are at a value $\hat{\theta}_{k-1}$ not at θ^*, the estimate $\hat{g}_k(\cdot)$ is, on average, sufficiently steep (as well as pointing toward θ^*) so that there will be a tendency to push the next value $\hat{\theta}_k$ toward θ^*.[3] One case where C3) may be violated is where (5) is used and there are strong system transients between the times associated with $\hat{\theta}_{k-1}+c_k\Delta_k$ and $\hat{\theta}_{k-1}-c_k\Delta_k$ (in such cases, it may be desirable to lengthen the time interval between changes in θ to allow the transients to decay). Note that the nonuniformity (in k) that is allowed for $\delta_k(\rho)$ permits $L_k(\cdot)$ to "flatten out" in some fixed region around θ^* as k gets large provided that this flattening does not occur too fast.[4] C4) is a very weak condition that says if $\hat{\theta}_k$ is uniformly bounded away from θ^*, then it cannot be bouncing around in a manner that causes the elements of the mean of $\hat{g}_k(\cdot)$ to change sign an infinite number of times. C5) is another weak condition that ensures for k sufficiently large that each element of $\bar{g}_k(\theta)$ makes a nonnegligible contribution to products of the form $(\theta-\theta^*)^T\bar{g}_k(\theta)$ [see C3)] when $(\theta-\theta^*)_i \neq 0\ \forall i$. A sufficient condition for C5) is that for each i, $\bar{g}_{ki}(\theta)$ is uniformly (in k) bounded >0 and $<\infty$ when $|(\theta-\theta^*)_i| \geq \rho > 0\ \forall i$.

The assumption in the proposition that there exists a fixed θ^* such that $\theta_k^* \to \theta^*$ is reasonable for a range of applications. In fact, the even stronger assumption that $\theta_k^* = \theta^*\ \forall k$ holds in many applications, including settings where $L_k(\theta)$ is perpetually varying at any θ (as results from, among other things, a time-varying target t_k). For example, stationary dynamics and measurement processes result in a control law that can generally be expressed as $u_k = u(\cdot)$ for some $u(\cdot)$ not indexed by k (see, e.g., Nijmeijer and van der Schaft [33, Ch. 14]). Then, there will generally exist a θ^* that yields the best possible approximation to $u(\cdot)$ under a mean-square-type loss function (this θ^* may not be unique unless the FA is minimal in some sense—see e.g., Sussman [53]). The more general condition of the proposition, $\theta_k^* \to \theta^*$, allows for a system with transient effects.

Let us close this section with a brief discussion of the constant gain setting where we take $a_k = a > 0$ and/or $c_k = c > 0\ \forall k$. It is well known that SA algorithms with such gains are better able to track a time-varying root (θ_k^*) than the usual decaying gain algorithms (see e.g., Kushner and Huang [19], Benveniste and Ruget [3], Macchi and Eweda [25], or Benveniste *et al.* [4, pp. 120–164]), which is relevant when θ_k^* is nonconvergent. This is likely to occur, say, when the process or measurement dynamics are perpetually time-varying due to cyclic behavior or when they change due to a failure or wear and tear of components in the system. In fact, because of their ease of use, such constant gains are sometimes applied in SA (or SA-type) algorithms even when $\theta_k^* = \theta^*\ \forall k$ (see, e.g., Kushner and Huang [19] or Kuan and Hornik [18]), although it is known that they preclude the formal a.s. convergence of decaying gain algorithms.

IV. Empirical Studies

This section presents the results of numerical studies on two different nonlinear systems. In the first study we present results for controlling a wastewater treatment system, where the dynamics are in the so-called affine-nonlinear form. This study includes a comparison of direct approximation (DA) versus self-tuning (ST) controllers, SPSA versus FDSA estimation algorithms, and one-measurement SPSA [see (6)] versus two-measurement SPSA [see (5)]. In the second study we consider a system where the noise is not additive and where the control only begins to have an effect after a certain time lag. For this second study we compare two different FA's: a neural network and a polynomial. This study also briefly examines the Polyak–Ruppert iterate averaging technique.

Because of time-varying dynamics, the first study uses constant SA gains $(a_k = a, c_k = c)$ for the estimation algorithms. The dynamics in the second study are time-invariant; hence decaying gains are used, which fulfills the requirements for convergence given in the proposition of Section III-B.

A. Wastewater Treatment System

This section presents the results of a study on a wastewater treatment system from Dochain and Bastin [11].[5] Our interest here is to compare the SPSA (one- and two-measurement forms) and FDSA estimation algorithms as well as the DA and ST control approaches of Fig. 1(a) and (b). Models for similar wastewater treatment systems may also be found in the bioremediation literature (e.g., Cardello and San [6]). This is a model of affine-nonlinear multiplicative control form (e.g., Chen [7]).

In this wastewater treatment system, influent wastewater is first mixed (as determined by a controller) with a dilution substance to provide a mixture with a desired concentration of contaminants. This diluted mixture is then sent to a second tank at a controlled flow rate. In the second tank the mixture goes through an anaerobic digestion process, where the organic material in the mixture is converted by bacteria into by-products such as methane (Metcalf and Eddy [28, p. 420]).

[3] This condition does not preclude $\hat{\theta}_k$ from converging to a local minimizing point θ^* (the same issue arises, of course, in gradient search algorithms such as back-propagation); Chin [9] discusses a technique by which SPSA can be used as a global optimizer for arbitrary initial conditions. An alternate global optimizing approach that seems likely to apply here is described in Yakowitz [56] for the FDSA algorithm. We have yet to investigate such approaches in control problems, instead relying on the standard approach of experimenting with different initial conditions to see that we are converging to the same minimum. In fact, this issue is not always critical since in some systems converging to only a local minimum may offer performance that is sufficiently improved for the allowable resources expended.

[4] In some special cases, the conditional expectation $\bar{g}_k(\cdot)$ can be replaced by the true gradient. Then condition C3) resembles the well-known "steepness" condition in stochastic approximation derived from Lyapunov theory (e.g., Lai [21] and Ruppert [40]). Two of the special cases are when the one-measurement form (6) is used for $\hat{g}_k(\cdot)$ or when the system can be "reset" when the two-measurement form (5) is used (i.e., the system can be placed at the same state prior to generating $u_k^{(+)}$ and $u_k^{(-)}$, as, say, with some robotic systems). In these special cases $\bar{g}_k(\hat{\theta}_k)$ equals $g_k(\hat{\theta}_k)$ plus an $O(c_k^2)$ bias that can be absorbed into $\delta_k(\rho)$.

[5] More detailed information and additional references on this wastewater treatment system model may be found in Spall and Cristion [52]. That paper considers a model with process noise but no measurement noise. A similar DA controller model was used except for a time-invariant target vector. Also included in the study of Spall and Cristion [52] were the effects of gradient averaging at each iteration (not considered here). The model here, however, has a greater degree of nonstationarity.

Therefore, the system consists of two controls (the mix of wastewater/dilution substance and the input flow rate) and two states (an effluent depolluted water and methane gas, which is useful as a fuel). Since this system relies on biological processes, the dynamics are nonlinear and usually time-varying (Cardello and San [6]). Also, the system is subject to constraints (e.g., the input and output concentrations, the methane gas flow rate, and the input flow rate all must be greater than zero), which presents an additional challenge in developing a controller for the system. (Note that Dochain and Bastin [11] controlled only the output substrate concentration or the methane production rate—not both—using the input flow rate as their only control, and they used an indirect controller where a general form for the model of the wastewater treatment system was assumed to be known with unknown parameters that were estimated.)

The study here is based on $A_k = A$ (a constant weighting matrix) and $B_k = 0$ in the loss function (2) (i.e., a minimum variance regulator). The performance of the technique will mainly be evaluated by presenting an estimate of the root-mean-square (rms) error for the measurements, i.e., an estimate of $\{E[(y_k - t_k)^T A(y_k - t_k)]\}^{1/2}$. For our studies, we used a two-dimensional diagonal weight matrix A with a value .01 as the first diagonal element and .99 as the second diagonal element (reflecting the relative emphasis to be given to methane production and water purity, respectively). We will also present some results on the actual (versus measured) rms state tracking error. The (feedforward) neural networks (NN's) considered here have nodes that are scaled logistic functions (i.e., $1/(1+e^{-z})$ for input z). Each node takes as an input (z) the weighted sum of outputs of all nodes in the previous layer plus a bias weight not connected to the rest of the network as in Chen [7]. For the weight estimation we will consider the SPSA and the FDSA algorithms. For the SPSA algorithms we take the perturbations Δ_{ki} to be Bernoulli ± 1 distributed, which satisfies the relevant regularity conditions mentioned in Section III.

The nonstationary model we used for producing the measurements closely follows that of Dochain and Bastin [11, eqs. (8) and (9)] with the addition of additive (independent) process and measurement noise, i.e.,

$$\begin{pmatrix} x_{k+1,1} \\ x_{k+1,2} \end{pmatrix} = \begin{pmatrix} 1+\mu_k T & 0 \\ -.3636T & 1 \end{pmatrix} \begin{pmatrix} x_{k1} \\ x_{k2} \end{pmatrix} + \begin{pmatrix} -Tx_{k1} & 0 \\ -Tx_{k2} & T \end{pmatrix} \times \begin{pmatrix} u_{k1} \\ u_{k1}u_{k2} \end{pmatrix} + \begin{pmatrix} w_{k1} \\ w_{k2} \end{pmatrix} \text{ (state)} \tag{17a}$$

$$\mu_k = \frac{(.4 + .15\sin(2\pi k/96))x_{k2}}{.4 + x_{k2}} \quad \text{(bacterial growth rate)} \tag{17b}$$

$$y_k = x_k + v_k \text{ (measurement)} \tag{17c}$$

where $w_k \sim N(0, \sigma_w^2 I)$, $v_k \sim N(0, \sigma_v^2 I)$, and the sampling period is $T = .5$. The DA control algorithm, of course, has no knowledge of the model (17a)–(17c). The ST control algorithm has some prior information about the form of the model, namely that the state equation (17a) has the general affine-nonlinear form shown, but with no knowledge of the functional form of the nonzero/nonone elements appearing as the components of the two 2×2 matrices in (17a) (the "one" elements are a result of the transformation of a differential equation to a difference equation). For the target sequence t_k we used a periodic square wave, with values $(.97, .13)^T$ for the first 48 time points and $(1, .1)^T$ for the second 48 time points, where the second (water purity) component is as in Fig. 4 in Dochain and Bastin [11] (we also varied the first component [methane production rate] to provide for time variation in both components). The controllers for both the DA and ST methods were NN's with two hidden layers, one of 20 nodes, and one of 10 nodes (as in Narendra and Parthasarathy [31] and Chen [7]). The inputs to the controller were the current and most recent state $(M = 2)$ and the most recent control $(N = 1)$, yielding a total of eight input nodes for the DA controller (the target vector for the next state was also included) and six input nodes for the ST controller. In the notation of Narendra and Parthasarathy [31], an $N_{8,20,10,2}$ network was used for the DA controller, which has $180 + 210 + 22 = 412$ weights to be estimated, and an $N_{6,20,10,5}$ network was used for the ST controller, which has $140 + 210 + 55 = 405$ weights to be estimated.

Fig. 2 shows the main results for our study of the model in (17a)–(17c), based on the procedure outlined in Section III-A with the two-measurement form for $\hat{g}_k(\cdot)$ in (5) (including the generation of the optional nominal state for purposes of plotting the weighted total system rms error). As with a practical wastewater system, there is no system resetting in the course of the SPSA estimation (see Section III-A). The rms error curves in the figure are based on the sample mean of ten independent runs, where the elements of $\hat{\theta}_0$ for each run were generated randomly (and independently) from a uniform $(-.1, .1)$ distribution for the DA controller and a uniform $(-.01, .01)$ distribution for the ST controller. We chose $x_0 = (.5, 1.6375)^T$, process noise $\sigma_w = .001$, and measurement noise $\sigma_v = .01$, so the initial-weighted total system rms error is 1.51 (which includes the effects of the measurement noise) and the minimum achievable long-run weighted total system rms error is some number greater than .01 (see footnote 6). To further smooth the resulting error curves and to show typical performance (not just case-dependent variation), we applied an expanding window smoother (which allows for rapid changes in the early iterations and little change in later iterations) to the error values based on the average of ten runs. The curves shown in the figure are based on this combination of across-realization averaging and across-iteration smoothing.

For this nonstationary system, we used constant SA gains of the form $a_k = a$ and $c_k = c$ with $a, c > 0$. We attempted to tune a and c in each algorithm to approximately maximize the rate of decay in weighted total system rms error (as would typically be done in practice); the values satisfied $.001 \le a \le .5$ and $.005 \le c \le .1$. The choice of a, c is important for adequate performance of the algorithm (analogous to choosing the step-size in back-propagation). For example, choosing a too small may lead to an excessively slow reduction in error, while choosing an a too large may cause the system to go unstable (so for practical problems, where *a priori* "tuning" may not be feasible, it might be appropriate to begin with a relatively small a and gradually increase it until there is an

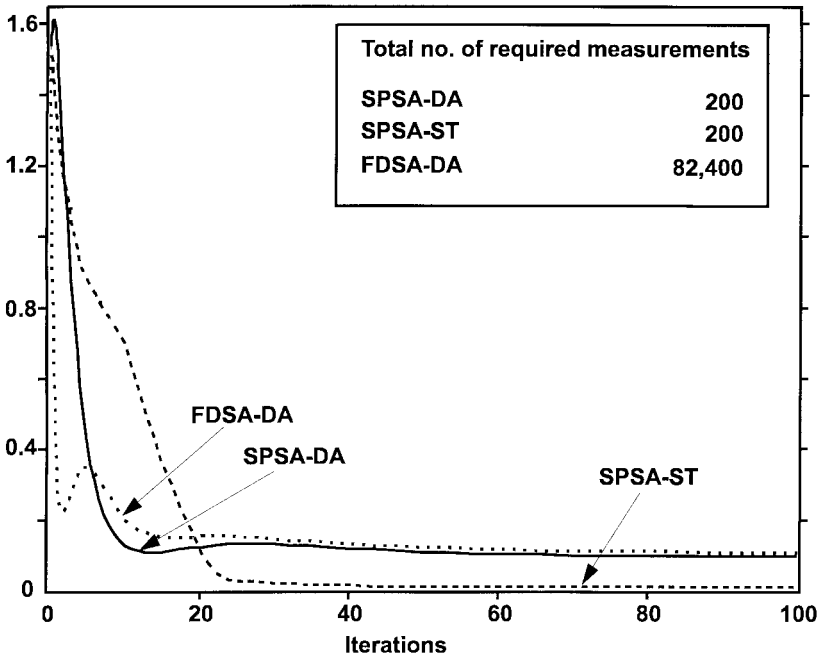

Fig. 2. Root mean square errors for DA and ST controllers and a relative number of required loss function measurements in wastewater treatment system.

adequate convergence rate but little chance of going unstable; Spall [49] also includes some practical guidelines for choosing the gain sequences).

Fig. 2 shows that all of the SPSA and FDSA algorithms yield controllers with decreasing weighted total system rms error over time.[6] We see that the overall performance of SPSA-DA is somewhat better than FDSA-DA. Further, SPSA-ST has the best terminal performance, reflecting the value of the additional information used. The critical observation to make here is that the SPSA algorithms achieved their performance with a large savings in data: each iteration of the SPSA algorithms required only two measurements, while each iteration of the FDSA algorithm needed 824 measurements. Hence Fig. 2 illustrates that the SPSA algorithms yielded a slightly lower level of rms error than the standard FDSA algorithm with a 412-fold savings in measurements. The data savings seen in Fig. 2 is typical of that for a number of other studies involving SPSA and FDSA that we have conducted on model (17a)–(17c) as well as on other nonlinear models (see Spall and Cristion [52]); in fact, even greater data savings are typical with more complex NN's (as might be needed in higher dimensional systems). Note also that the curves in Fig. 2 have the typical shape of many optimization algorithms in that there is a sharp initial decline followed by a slow decline. Hence, the rms error is reduced over 90 percent, (which may be all that is required in some applications), by the SPSA algorithms (DA and ST) within approximately 20 iterations. In terms of relative performance, the same pattern holds for actual rms state (versus measurement) tracking error. For example, the rms state error for SPSA-ST at 20 iterations was 0.0389 (versus 0.0981 for the measured tracking error shown in Fig. 2), at 50 iterations was 0.0070 (0.0128), and at 100 iterations was 0.0059 (0.0127). Of course, in a real system the state tracking error would not be measurable.

In Fig. 2 we see that the SPSA-DA algorithm has slightly better overall performance than the FDSA-DA algorithm. This appears to be a result of the significant nonstationary dynamics shown in (17b). Since the FDSA-DA algorithm requires 412 times more measurements than SPSA-DA at each iteration, the system dynamics change more over the course of one gradient approximation. Therefore, the FDSA-DA algorithm will have inherent difficulties in achieving the same performance as the SPSA-DA algorithm since in SPSA the dynamics change only a negligible amount over the course of a gradient approximation. In contrast, the wastewater study in Spall and Cristion [52] has a smaller level of nonstationarity, and consequently the FDSA-DA and the SPSA-DA algorithms have more nearly equivalent overall rms errors performances (of course, FDSA-DA still requires many times the number of measurements of SPSA-DA to achieve this performance).

As a final study on this system, we evaluated the one-measurement SPSA form (6) in the DA context. After 100 iterations, the rms error was 0.195, somewhat greater than 0.0983 for the two-measurement form, but still much improved from the 1.51 initial error at half the cost of the two-measurement form.[7] It is expected that the ideal application for (6) is in systems with even greater nonstationarity where the underlying dynamics change significantly at each measurement point.

[6] It does not appear possible to analytically know the minimum achievable total measured rms errors for each algorithm since they involve the combined effects of the process and measurement noise as well as the nonstationary dynamics and requisite nondecaying SA gains (which preclude formal convergence of the $\hat{\theta}_k$). As an approximate indication of these lower bounds, SPSA-DA and SPSA-ST achieved total measured rms errors of 0.0845 and 0.0104, respectively, after 10 000 iterations versus values of 0.0983 and 0.0127 at 100 iterations.

[7] After 10 000 iterations, the rms error for the one-measurement form was 0.0889, relatively close to the error of 0.0845 for two-measurement SPSA.

B. Nonadditive Noise Model

The second model we consider is one where the noise is not additive and the control is added to the dynamics. In particular, as in Yaz [56], the data are generated according to

$$y_{k+1} = \begin{pmatrix} -.5 & .3 \\ 0 & 1.1 \end{pmatrix} y_k + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} u_k + \begin{pmatrix} \|y_k\| \\ 0 \end{pmatrix} w_k, \quad y_k, u_k \in R^2 \quad (18)$$

where w_k is an independent scalar Bernoulli $\pm.5$ noise process and $\|\cdot\|$ denotes the Euclidean norm. Aside from the multiplicative (possibly unstable) mode in which the noise enters (18), this model is interesting since only one of the two control elements affects the system, and since the first element of y_k can only be affected by a control after a delay of one time period. We used a periodic square wave target sequence, where $t_k = (1,0)^T$ for the first five iterations of the period and $t_k = (-1,0)^T$ for the second five iterations, which yields a long-run best possible rms error of $1/\sqrt{2}$ based on the same quadratic loss function considered in Section IV-A, with a diagonal weight matrix A a with a value .5 for both of the diagonal elements and $B = 0$. Since this system has time-invariant dynamics (and fixed $\theta_k^* = \theta^*$), the proposition of Section III is relevant here.

In this study we looked at two different function approximators using the DA method [which, of course, assumes no knowledge of the dynamics in (18)]. As in Fig. 1(a), only the most recent measurement (and next target) are fed into the controller (i.e., $M = 1$, $N = 0$). One FA was a $N_{4,5,5,2}$ neural network, which has $25 + 30 + 12 = 67$ weights to be estimated (including bias weights). The hidden layer nodes were hyperbolic tangent functions (i.e., $(e^z - e^{-z})/(e^z + e^{-z})$ for input z) and the output nodes were linear functions. The second FA was a third-order polynomial, which has 70 parameters to be estimated (close to the 67 weights used for the NN FA so as to maintain an approximate equivalence in complexity of the FA).

Table I shows the results of the study with model (18). The rms errors were calculated as in the wastewater treatment study above, with values formed from the same averaging/interpolation scheme. The decaying SA gains were of the form $a_k = a/k^{.602}$ and $c_k = c/k^{.101}$ with $.1 \le a \le .5$ and $.3 \le c \le 5$ (the gains satisfy condition C1) of the proposition; although not used here, an a_k of the form $a_k = a/(k + A)^{.602}$, $A > 0$, usually provides superior practical performance as discussed in Spall [49]). We also used an average of four individual SP gradient approximations for each gradient estimate to enhance algorithm performance given the relatively large noise level (even at the expense of the six additional measurements required per iteration). Numerical analysis of the iterates $\hat{\theta}_k$ (for both the polynomial and the NN) indicate convergence to a fixed θ^* as predicted by the proposition.

As seen in Table I, both the NN and polynomial FA's produced good results, although the polynomial was slightly better in the long run. We also looked at higher order polynomials and NN's in controlling this system, but the orders chosen here seemed to work well. In fact, when using a fourth-order polynomial function, the controller would often drive the system into an unstable mode. Higher order NN's, on the other hand, were able to keep the system under control, but performed no better than the NN presented here. The lower order FA's may have performed better in the system of (18) because of the inherent instability of the system (see Yaz [57]); in higher order FA's, there is a greater possibility of poorly initialized parameters (or combinations of parameters) that may cause the system to go unstable.

TABLE I
RMS ERRORS FROM NONADDITIVE NOISE MODEL (MINIMUM ACHIEVABLE RMS = 0.707)

Iteration Number	Neural Net	Polynomial
0	1.500	1.500
10	1.530	0.971
50	0.755	0.778
100	0.727	0.746
1000	0.719	0.708

We also considered averaging the iterates $\hat{\theta}_k$ over time in the context of the polynomial DA controller (so the controller used the averaged value instead of the most recent iterate). This averaging method has been shown theoretically to yield asymptotically minimum variance estimates in the general Robbins–Monro SA setting with nontime-varying loss function (Polyak and Juditsky [37]) and to offer improved performance in some SPSA settings (Dippon and Renz [10] and Maryak [27]). By its nature, of course, averaging seems most appropriate for systems that have stationary—or perhaps asymptotically very slowly time-varying—dynamics (e.g., the case of the proposition). In our primary study, we initialized the averaging at the 50th iteration (so as to ignore the initial parameter estimates, which will typically not be close to the optimal θ^*). Unfortunately, however, the averaged results were slightly poorer than the nonaveraged results (e.g., at iteration 100 the rms value was 0.761 and at iteration 1000 it was .709 versus 0.746 and 0.708, respectively, for the nonaveraged results in Table I). This slightly poorer performance was consistent even as we varied aspects of the study. For example, if the initial point for averaging to commence was changed from iteration 50 to both higher and lower values or if the target sequence was changed to a constant, the averaging approach consistently yielded a slightly higher rms error than the nonaveraging approach. The averaging method should be most useful in practical finite-sample situations when the iterate is bouncing approximately uniformly around the solution. However, we found that the latest SPSA-DA iterates consistently produced better rms results than the earlier iterates (which were not bouncing "uniformly" around θ^*); so by using past iterates, the averaging method appears to be folding in too many relatively poor values. A similar result is discussed in Wang [54, p. 37] and Maryak [27]. The numerical results here are in contrast to the numerical results of Kushner and Yang [20], where it is shown that the averaging scheme yields significant improvements in a Robbins–Monro (noncontrol) setting. We expect that in certain other control problems, this type of averaging may be more effective and

may offer significant improvements over the nonaveraging implementations.

V. Summary and Conclusions

This paper has considered the problem of controlling nonlinear stochastic systems with unknown process and measurement equations. Our approach differs fundamentally from conventional methods in adaptive control: rather than modeling the system and performing a stability analysis en route to building a controller, this method avoids the construction of an open-loop model and focuses directly on regulating the system via the construction of a closed-loop control function. So, the approach here addresses the shortcoming noted in Narendra and Parthasarathy [31, p. 19] that "At present, methods for directly adjusting the control parameters based on the output error (between the plant and reference [target] outputs) are not available." The approach encompasses two different methods—DA control and ST control—where DA applies when there is very little information available about the system dynamics and ST applies when some (still incomplete) information is available.

Since we are not assuming full knowledge of the structure of the equations describing the system, it is not possible to calculate the gradient of the loss function for use in standard gradient-descent-type search algorithms. Therefore, we describe a stochastic approximation-based method for the estimation of the controller, which is based on a "simultaneous perturbation" approximation (Spall [46]). This method relies on observing the system at one or (usually) two levels of the control to each iteration of the algorithm. Both theoretical and empirical evidence indicate that this SPSA method for weight estimation is much more efficient (in terms of the number of system measurements needed) than more standard Kiefer–Wolfowitz-type SA methods based on finite-difference approximations to the gradient.

There remain several open problems to address to further enhance the applicability of the approach here. One, perhaps, is to develop general conditions for stability and controllability, although the extent to which these conditions could be checked in a specific application is limited by the model-free framework. Further, it seems that very little work has been done on such issues for general nonlinear, stochastic, discrete-time systems (although for deterministic systems, one may consult Mousa *et al.* [30], Nijmeijer and van der Schaft [33, Ch. 14], or references mentioned in Section I). Essentially, we feel that stability analysis should not be a necessary aspect in building all controllers since that would prevent the solution of many real-world problems. In practice, systems can often be monitored for anomalous behavior and sometimes shut down or converted to a default control if instabilities are a threat. Another issue is to develop ways to increase the rate at which the required parameter estimates approach the globally optimal values. This is especially relevant in systems where precise control is needed within a short time since SPSA has the property of bringing the iterate to within the vicinity of the optimum in relatively few time points but then taking a long time to complete the convergence to the optimum (this property, of course, is common to all first-order search algorithms, including, e.g., standard gradient descent). A number of techniques have been proposed to accelerate the convergence of SA algorithms or to enhance convergence to a global minimum (see, e.g., Spall [48], Chin [9], or Yakowitz [55]), and it would be of interest to explore the applicability of such techniques to SPSA in a control context. Constraints are usually handled on a problem-dependent basis (such as the wastewater example in Section IV-A), but general approaches with SPSA are described in Sadegh [41] and Fu and Hill [15]; these approaches have yet to be implemented in a control context. Another open problem is one common to many applications of function approximators: namely, to develop guidelines for determining the optimal (at least approximately) structure for the FA, e.g., optimal number of hidden layers and nodes in a neural network. Related to this is the problem of allowing for the FA structure to change if, say, the number of controller inputs or outputs change (Nechyba and Xu [32] present an approach for neural networks). Although solving any of the above problems would enhance the SPSA-based approach to control, the approach as it currently stands still has broad applicability to many practical systems where little is known about the equations describing the system.

Acknowledgment

The authors appreciate insightful comments from Dr. L. Gerenscer of the Hungarian Academy of Sciences and Dr. I.-J. Wang of JHU/APL, especially regarding the main convergence result.

References

[1] M. S. Ahmed and M. F. Anjum, "Neural-net-based self-tuning control of nonlinear plants," *Int. J. Contr.*, vol. 66, pp. 85–104, 1997.
[2] D. S. Bayard, "A forward method for optimal stochastic nonlinear and adaptive control," *IEEE Trans. Automat. Contr.*, vol. 36, pp. 1046–1053, 1991.
[3] A. Benveniste and G. Ruget, "A measure of the tracking capability of recursive stochastic algorithms with constant gains," *IEEE Trans. Automat. Contr.*, vol. AC-27, pp. 639–649, 1982.
[4] A. Benveniste, M. Metivier, and P. Priouret, *Adaptive Algorithms and Stochastic Approximation*. New York: Springer-Verlag, 1990.
[5] J. R. Blum, "Approximation methods which converge with probability one," *Ann. Math. Stat.*, vol. 25, pp. 382–386, 1954.
[6] R. J. Cardello and K.-Y. San, "The design of controllers for batch bioreactors," *Biotech. Bioeng.*, vol. 32, pp. 519–526, 1988.
[7] F. C. Chen, "Back-propagation neural networks for nonlinear self-tuning adaptive control," *IEEE Contr. Syst. Mag.*, pp. 44–48, Apr. 1990.
[8] T. Chen and H. Chen, "Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems," *IEEE Trans. Neural Networks*, vol. 6, pp. 911–917, 1997.
[9] D. C. Chin, "A more efficient global optimization algorithm based on Styblinski and Tang," *Neural Networks*, pp. 573–574, 1994.
[10] J. Dippon and J. Renz, "Weighted means in stochastic approximation of minima," *SIAM J. Contr. Optimiz.*, vol. 35, pp. 1811–1827, 1997.
[11] D. Dochain and G. Bastin, "Adaptive identification and control algorithms for nonlinear bacterial growth systems," *Automatica*, vol. 20, pp. 621–634, 1984.
[12] S. N. Evans and N. C. Weber, "On the almost sure convergence of a general stochastic approximation procedure," *Bull. Austral. Math. Soc.*, vol. 34, pp. 335–342, 1986.
[13] S. Fabri and V. Kadirkamanathan, "Dynamic structure neural networks for stable adaptive controls of nonlinear systems," *IEEE Trans. Neural Networks*, vol. 7, pp. 1151–1167, 1996.
[14] W. Fleming, *Functions of Several Variables*. New York: Springer-Verlag, 1977.
[15] M. C. Fu and S. D. Hill, "Optimization of discrete event systems via simultaneous perturbation stochastic approximation," *Trans. Inst. Indust. Engineers*, vol. 29, pp. 233–243, 1997.

[16] S. Jagannathan, F. L. Lewis, and O. Pastravanu, "Discrete-time model reference adaptive control of nonlinear dynamical systems using neural networks," *Int. J. Contr.*, vol. 64, pp. 217–239, 1996.
[17] M. I. Koch, D. C. Chin, and R. H. Smith, "Network-wide approach to optimal signal light timing for integrated transit vehicle and traffic operations," in *Proc. 7th National Conf. Light Rail Transit*, vol. 2. National Academy of Sciences Press, 1997, pp. 126–131.
[18] C. M. Kuan and K. Hornik, "Convergence of learning algorithms with constant learning rates," *IEEE Trans. Neural Networks*, vol. 2, pp. 484–489, 1991.
[19] H. J. Kushner and H. Huang, "Asymptotic properties of stochastic approximations with constant coefficients," *SIAM J. Contr. Optimiz.*, vol. 19, pp. 87–105, 1981.
[20] H. J. Kushner and J. Yang, "Stochastic approximation with averaging and feedback: Rapidly convergent on-line algorithms," *IEEE Trans. Automat. Contr.*, vol. 40, pp. 24–34, 1995.
[21] T. L. Lai, "Stochastic approximation and sequential search for optimum," in *Proc. Berkeley Conf. in Honor of Jerzy Newman and Jack Kiefer*, vol. 2, L. M. LeCam and R. A. Olshen Eds. Belmont, CA: Wadsworth, 1985, pp. 557–577.
[22] S. H. Lane, D. A. Handelman, and J. J. Gelford, "Theory and development of higher order CMAC neural networks," *IEEE Contr. Syst. Mag.*, vol. 12, pp. 23–30, Apr. 1992.
[23] A. U. Levin and Narendra, "Control of nonlinear dynamical systems using neural networks: Controllability and stabilization," *IEEE Trans. Neural Networks*, vol. 4, pp. 192–206, 1993.
[24] A. U. Levin and Narendra, "Control of nonlinear dynamical systems using neural networks—Part ii: Observability, identification, and control," *IEEE Trans. Neural Networks*, vol. 7, pp. 30–42, 1996.
[25] O. Macchi and E. Eweda, "Second order convergence analysis of stochastic adaptive linear filtering," *IEEE Trans. Automat. Contr.*, vol. 28, pp. 76–85, 1983.
[26] Y. Maeda and R. J. P. De Figueiredo, "Learning rules for neurocontroller via simultaneous perturbation," *IEEE Trans. Neural Networks*, vol. 8, pp. 1119–1130, 1997.
[27] J. L. Maryak, "Some guidelines for using iterate averaging in stochastic approximation," in *Proc. IEEE Conf. Decision Control*, 1997, pp. 2287–2290.
[28] Metcalf and Eddy, Inc., *Wastewater Engineering: Treatment, Disposal, and Reuse*, 3rd ed. New York: McGraw-Hill, 1972.
[29] P. E. Moden and T. Soderstrom, "Stationary performance of linear stochastic systems under single step optimal control," *IEEE Trans. Automat. Contr.*, vol. AC-27, pp. 214–216, 1982.
[30] M. S. Mousa, R. K Miller, and A. N. Michel, "Stability analysis of hybrid composite dynamical systems: Descriptions involving operators and difference equations," *IEEE Trans. Automat. Contr.*, vol. AC-31, pp. 603–615, 1986.
[31] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Networks*, vol. 1, pp. 4–26, 1990.
[32] M. C. Nechyba and Y. Xu, "Human-control strategy: Abstraction, verification, and replication," *IEEE Contr. Syst. Mag.*, vol. 17, pp. 48–61, Oct. 1997.
[33] H. Nijmeijer and A. J. van der Schaft, *Nonlinear Dynamical Control Systems*. New York: Springer-Verlag, 1990.
[34] Y.-M. Pao, S. M. Phillips, and D. J. Sobajic, "Neural-net computing and the intelligent control of systems," *Int. J. Contr.*, vol. 56, pp. 263–289, 1992.
[35] T. Parisini and R. Zoppoli, "Neural approximations for multistage optimal control of nonlinear stochastic systems," *IEEE Trans. Automat. Contr.*, vol. 41, pp. 889–895, 1996.
[36] T. Poggio and F. Girosi, "Networks for approximation and learning," in *Proc. IEEE*, vol. 78, pp. 1481–1497, 1990.
[37] B. T. Polyak and A. B. Juditsky, "Acceleration of stochastic approximation by averaging," *SIAM J. Contr. Optimiz.*, vol. 30, pp. 838–855, 1992.
[38] W. Rudin, *Principles of Mathematical Analysis*. New York: McGraw-Hill, 1964.
[39] D. Ruppert, "Kiefer-Wolfowitz procedure," *Encyclopedia of Statistical Science*, vol. 4, S. Kotz and N. L. Johnson, Eds. New York: Wiley, 1983, pp. 379–381.
[40] D. Ruppert, "A Newton-Raphson version of the multivariate Robbins-Monro procedure," *Ann. Stat.*, vol. 13, pp. 236–245, 1985.
[41] P. Sadegh, "Constrained optimization via stochastic approximation with a simultaneous perturbation gradient approximation," *Automatica*, vol. 33, pp. 889–892, 1997.
[42] P. Sadegh and J. C. Spall, "Optimal random perturbations for stochastic approximation using a simultaneous perturbation gradient approximation," *IEEE Trans. Automat. Contr.*, to be published.
[43] R. M. Sanner and J.-J. Slotine, "Gaussian networks for direct adaptive control," *IEEE Trans. Neural Networks*, vol. 3, pp. 837–863, 1992.
[44] G. N. Saridis, *Self-Organizing Control of Stochastic Systems*. New York: Marcel Dekker, 1977.
[45] M. A. Sartori and P. J. Antsaklis, "Implementation of learning control systems using neural networks," *IEEE Contr. Syst. Mag.*, vol. 12, pp. 49–57, Apr. 1992.
[46] J. C. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *IEEE Trans. Automat. Contr.*, vol. 37, pp. 332–341, 1992.
[47] ———, "A one-measurement form of simultaneous perturbation stochastic approximation," *Automatica*, vol. 33, pp. 109–112, 1997.
[48] ———, "Accelerated second-order stochastic optimization using only function measurements," in *Proc. IEEE Conf. Decision Control*, 1997, pp. 1417–1424.
[49] ———, "Implementation of the simultaneous perturbation algorithm for stochastic optimization," *IEEE Trans. Aero. Electronic Syst.*, vol. 34, pp. 817–823.
[50] J. C. Spall and D. Chin, "Traffic-responsive signal timing for system-wide traffic control," *Transp. Res.-C*, vol. 5, pp. 153–163, 1997.
[51] J. C. Spall and J. A. Cristion, "Nonlinear adaptive control using neural networks: Estimation based on a smoothed form of simultaneous perturbation gradient approximation," *Statistica Sinica*, vol. 4, pp. 1–27, 1994.
[52] J. C. Spall and J. A. Cristion, "A neural network controller for systems with unmodeled dynamics with applications to wastewater treatment," *IEEE Trans. Syst. Man, Cybern.*, Part B, vol. 27, pp. 369–375, 1997.
[53] H. J. Sussman, "Uniqueness of the weights for minimal feedforward nets with a given input–output map," *Neural Networks*, vol. 5, pp. 589–593, 1992.
[54] I.-J. Wang, "Analysis of stochastic approximation and related algorithms," Ph.D. dissertation, School of Electrical and Computer Engineering, Purdue Univ., 1996.
[55] S. Yakowitz, "A globally convergent stochastic approximation," *SIAM J. Contr. Optimiz.*, vol. 31, pp. 30–40, 1993.
[56] E. Yaz, "A control scheme for a class of discrete nonlinear stochastic systems," *IEEE Trans. Automat. Contr.*, vol. AC-32, pp. 77–80, 1987.

James C. Spall (S'82–M'83–SM'90) joined Johns Hopkins University, Applied Physics Laboratory, in 1983 and was appointed to the Principal Professional Staff in 1991. He also teaches in the Johns Hopkins School of Engineering. He has published many articles in the areas of statistics and control and holds two U.S. patents.

In 1990, Dr. Spall received the Hart Prize as Principal Investigator of the most outstanding independent research and development project at JHU/APL. He is an Associate Editor for the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, a Contributing Editor for the *Current Index to Statistics*, and he served as editor and coauthor for the book *Bayesian Analysis of Time Series and Dynamic Models*. He is a member of the American Statistical Association and a fellow of the engineering honor society Tau Beta Pi.

John A. Cristion received the B.S. degree in electrical engineering from Drexel University, Philadelphia, PA, in 1986 and the M.S. degree in electrical engineering from Johns Hopkins University, Baltimore, MD, in 1991.

Since 1986, he has been with the Johns Hopkins University, Applied Physics Laboratory, where his current research interests include sonar signal processing, biomedical engineering, higher order statistical signal processing, and statistical modeling. He has published in the areas of controls, sonar signal processing, and biomedical engineering on subjects such as epileptic seizure detection, polygraph analysis, parameter estimation, beam-forming, and neural networks.

Mr. Cristion is a member of Tau Beta Pi and Eta Kappa Nu.