Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

An Efficient Adaptive Optimization Scheme

Elias B. Kosmatopoulos * **Markos Papageorgiou** *

* *Dynamic Systems and Simulation Laboratory, Department of Production & Management Engineering, Technical University of Crete, Chania 73100, Greece, tel: +30-28210-37306, kosmatop|markos@dssl.tuc.gr*

Abstract: Adaptive optimization schemes based on stochastic approximation principles such as the Random Directions Kiefer-Wolfowitz (RDKW), the Simultaneous Perturbation Stochastic Approximation (SPSA) and the Adaptive Fine-Tuning (AFT) algorithms possess the serious disadvantage of not guaranteeing efficient transient behaviour due to their requirement for using random or random-like perturbations of the current parameter vector. The use of random or random-like perturbations may lead to particularly large values of the objective function, a fact that prevents the wide application of these algorithms to controller fine-tuning and adaptive and learning control where efficient transient performance is a prerequisite; in these applications, there may be cases where a small perturbation of a "good" parameter vector may lead to an unacceptable – or, even worse, unstable – closed-loop behavior. In this paper, we introduce and analyze a new algorithm for alleviating this problem. Mathematical analysis establishes efficient transient performance and convergence of the proposed scheme under a general set of assumptions. Simulation results demonstrate the efficiency of the proposed scheme.

Keywords: Adaptive Optimization, Stochastic Approximation, Simultaneous Perturbation Stochastic Approximation (SPSA), Kiefer-Wolfowitz procedure, Adaptive Fine-Tuning

1. INTRODUCTION

Many controller fine-tuning as well as adaptive and learning control problems can be formulated as optimization problems where the objective function to be optimized cannot be computed directly (i.e., its analytical form is not known) but only estimated via observations. More formally, a large class of these problems can be shown to be equivalent to the problem of finding an algorithm for updating the vector of controller parameters $\theta_k \in \Re^{n_\theta}$ so that an appropriately defined *objective function*

$$J_k \equiv J(\theta_k, x_k) \tag{1}$$

converges as close as possible to one of its local minima. Here, J denotes the *unknown* nonlinear objective function which is assumed to [1] be C^m, $m \geq 2$, bounded-from-below and unbounded for unbounded θ_k; J_k denotes the objective function measurement at the k-th algorithm iteration; $x_k \in \Re^{n_x}$ is the vector of exogenous variables; and n_θ, n_x denote the dimensions of the vectors θ_k, x_k, respectively. In a typical case, J corresponds to an appropriately defined *performance index*, while the entries of the vector x_k may correspond to signals that are not available for measurement (e.g, sensor noise, un-measurable disturbances, etc.) as well as signals that are available for measurement (e.g., system states, measurable disturbances, reference signals, demand, etc.).

[1] Throughout this paper, a function f is said to be C^m, where m is a positive integer, if it is uniformly continuous and its first m derivatives are uniformly continuous.

Stochastic Approximation (SA) methods such as the Kiefer-Wolfowitz (KW) [13, 2], the Random Directions Kiefer-Wolfowitz (RDKW) [16], the Simultaneous Perturbation Stochastic Approximation (SPSA) [18, 20] and the Adaptive Fine-Tuning [15] algorithms are directly applicable to the above problem since – contrary to conventional optimization methods – they can efficiently deal with the problem of optimization of functions that cannot be computed directly but only estimated via observations. A common element in the aforementioned SA algorithms is the need for evaluating the objective function at random or pseudo-random perturbations $\Delta\theta+\theta$ around the current vector θ. For instance, 2-sided RDKW and SPSA assume an iterative scheme of the form

$$\theta_k = \theta_{k-3} - \beta_k \frac{J_{k-2} - J_{k-1}}{2\alpha_k} r_k \tag{2}$$

where $\theta_{k-2} = \theta_{k-3} + \alpha_k d_k, \theta_{k-1} = \theta_{k-3} - \alpha_k d_k$ are the perturbation points, α_k, β_k are slowly-decaying to zero step-sizes and r_k, d_k are random vectors.

Unfortunately, the objective function values J_{k-2}, J_{k-1} at the perturbation points may be particularly large. To see this, assume for simplicity that $x_k = 0$ for all k and consider the case where the current value of the parameter vector θ_k satisfies $|\nabla J(\theta_k)| = B$ with B being a positive constant; then it can be seen that the objective function at the next iteration satisfies

$$J(\theta_{k+1}) > J(\theta_k) + \alpha_k B, \text{ with probability P}$$

where $P > 0$ is a non-negligible probability. In other words, the aforementioned SA algorithms may introduce large

978-1-1234-7890-2/08/$20.00 © 2008 IFAC **10.3182/20080706-5-KR-1001.0335**

"spikes" in the objective function during the optimization process, especially when θ_k is away from a local minimum of J. This fact is actually one of the main reasons that SA algorithms have not found a wide application to controller fine-tuning and adaptive and learning control, since in these applications efficient transient performance is a prerequisite; it is worth noticing that, in these applications there may be cases where a small perturbation of a "good" parameter vector may lead to an unacceptable – or, even worse, unstable – closed-loop behavior (see e.g. [15]).

Therefore, there is a need for the construction of algorithms for updating θ_k which, on the one hand, retain the nice convergence properties of the aforementioned SA algorithms and, on the other hand, guarantee that the objective function values satisfy at each algorithm iteration

$$J_k \leq J_{k-1} + \epsilon_k$$

where ϵ_k is, in the worst case, a small positive term; as already presented above, in the case of the aforementioned SA algorithms the term ϵ_k can be particularly large.

In this paper, we propose and analyze a new algorithm that addresses the above problem. More precisely, as we establish by using rigorous arguments, at each iteration of the proposed algorithm J_k is *strictly decreasing* outside a subset centered at a local minimum of J; the magnitude of this subset depends on the magnitude of the exogenous disturbances, the estimation accuracy $x_k - \bar{x}_k$ (where $\bar{x}_k$ denotes a estimation/prediction of x_k) and the approximation accuracy of the estimator used by the proposed algorithm (which increases with the number of algorithm iterations); moreover, we establish that the proposed algorithm guarantees convergence under a general set of assumptions. The basic difference of the proposed algorithm as compared to KW, RDKW and SPSA is the use of a function approximator/estimator for the approximation of the unknown objective function and for the estimation of the effect of candidate perturbations. We note also that, in the case where estimates/predictions of the exogenous signals x_k are available, the proposed algorithm can incorporate this knowledge, something that it is not possible in the majority of the SA methodologies.

We close this section by noticing that simulation experiments (see section 3) demonstrate the validity of our theoretical results.

1.1 Notations

The notation $\text{vec}(A, B, \dots,)$, where $A, B, \dots$ are scalars, vectors or matrices, is used to denote a vector whose elements are the entries of $A, B, C, \dots$ (taken columnwise). Z denote the set of nonnegative integers. For a vector $x \in \Re^n$, $|x|$ denotes the Euclidean norm of x (i.e., $|x| = \sqrt{x^\tau x}$), while for a matrix $A \in \Re^{n^2}$, $|A|$ denotes the induced matrix norm of A. $\dim(x)$ denotes the dimension of the vector x. Finally, in order to avoid definition of too many variables, constants, etc, we will use the following notation: If $f_\alpha(x)$ is a function parametrized by the nonnegative parameter α, we will say that f_α is $O(\alpha)$ (symbolically and with some abuse of notation $f_\alpha = O(\alpha)$, if there exists a strictly increasing scalar, at least C^1, function g satisfying $g(0) = 0, g(\alpha) > 0,\ \forall \alpha \neq 0$, such that $|f_\alpha(x)| \leq g(\alpha), \forall x$. Note that our definition of $O(\cdot)$ differs from the usual "order of" definition. For instance, it is not difficult for someone to see that if a function f is $O(a)$ then, according to our definition, it will be also $O(a^2), O(a^3)$, etc.

2. THE PROPOSED ALGORITHM AND ITS CONVERGENCE PROPERTIES

As we have already noticed in the Introduction, the proposed approach uses a function approximator for the estimation of the unknown objective function J. More precisely, we use a linear in the parameters function approximator for the estimation of the unknown function J as follows (here $k-1$ denotes the current algorithm iteration):

$$\hat{J}_k(\theta_\ell, \bar{x}_\ell) = \vartheta_k^\tau \phi_k(\theta_\ell, \bar{x}_\ell),\ \ell \leq k \tag{3}$$

where $\hat{J}_k(\theta_\ell, \bar{x}_\ell)$ denotes the estimate of $J(\theta_\ell, x_\ell)$, ϕ_k denotes a nonlinear vector of regressor terms (which is assumed to be a smooth function of its arguments), ϑ_k denotes the vector of estimator tunable parameters (e.g., in the case where the estimator (3) is a neural network, ϑ_k denotes the vector of neural network weights),

$$L_k \equiv \dim(\phi_k(\theta, x)) = \min\{k+1, L\} \tag{4}$$

with L being a user-defined positive integer and ϑ_k is calculated according to

$$\vartheta_k = \arg\min_{\vartheta} \frac{1}{2} \sum_{\ell=\ell_k}^{k-1} \left(J_k - \vartheta^\tau \phi_k(\theta_\ell, \bar{x}_\ell)\right)^2 \tag{5}$$

where $\ell_k = \min\{0, k - L_k - T_h\}$ with T_h being a user-defined nonnegative integer. The vector $\bar{x}_\ell$ denotes an estimate (or prediction in case of future measurements, i.e., in the case where $\ell = k$) of the vector x_ℓ of actual exogenous signals.

To better understand the meaning of the estimator (3) assume for the time-being that the exogenous vector x_k is exactly known; then, if the regressor terms ϕ_k are chosen to belong to a family of Universal Approximators (e.g. polynomials, neural networks, etc) it can be seen using standard arguments from the theory of function approximation (see e.g., [14, 10, 11, 17] and the references therein) that the choice for ϑ_k according to (5) guarantees that

$$J_k = \hat{J}_k + \nu_k \tag{6}$$

where ν_k is a term that can be made arbitrarily small (for k large enough) provided that the "dimension" L of the regressor vector ϕ_k is large enough. In the sequel, we will use the notation N_L to denote the upper bound of the magnitude of the term ν_k (see Theorem 1 for a formal definition of N_L). We close this parenthesis by noticing that, since in the general case the exogenous vector x_k is not exactly known, equation (6) should be replaced by the following equation

$$J_k = \hat{J}_k + \nu_k + O(x_k - \bar{x}_k) \tag{7}$$

[**Availability of** $\bar{x}_k$] The proposed algorithm assumes that an estimate – or prediction – $\bar{x}_k$ of the vector x_k is available. In many cases of control applications such an assumption is realistic since the entries of x_k may correspond to system states and exogenous inputs which are

available for measurement or can be estimated/predicted using appropriate estimation algorithms (see the simulation section for such an example). However, there may be cases where such an assumption is not realistic; in this case it can be readily seen that all the results of the paper are still valid by setting $\bar{x}_k = 0$ and $c_{\bar{x}} = c_x$, where $c_{\bar{x}}, c_x$ are defined in Theorem 1 below.

The algorithm considered in this paper takes the following form: let α_k be a user-defined scalar positive sequence and $\Delta\theta_k^{(j)} \in \{-\alpha_k, +\alpha_k\}^{n_\theta}$, $j \in \{1, \ldots, K\}$ denote a collection of $K \geq n_\theta$ vectors of *candidate perturbations*, satisfying $\forall j \in \{1, \ldots, K\}$

$$\left| \left[\phi_{k-L_k+1}^{(k)}, \ldots, \phi_{k-1}^{(k)}, \phi_k(\theta_{k-1} \pm \Delta\theta_k^{(j)}, \bar{x}_k) \right]^{-1} \right| \leq \frac{\Xi_1}{\alpha_k} \quad (8)$$

with $\phi_\ell^{(k)} = \phi_k(\theta_\ell, \bar{x}_\ell)$, and

$$\left| \left[\Delta\theta_k^{(1)}, \ldots, \Delta\theta_k^{(K)} \right]^{-1} \right| \leq \frac{\Xi_2}{\alpha_k} \quad (9)$$

where Ξ_1, Ξ_2 are finite positive constants independent of α_k. Let also e_i denote the unit vector in the i-th direction (i.e., $e_{ii} = 1$ and $e_{ij} = 0$ for all $j \neq i$) and

$$\widehat{\nabla J}_k = \frac{\text{vec}\left(\hat{J}_k(\theta_{k-1} + c_k e_i, \bar{x}_k) - \hat{J}_k(\theta_{k-1}, \bar{x}_k) \right)}{c_k}$$

where c_k is a user-defined positive scalar sequence. Then $\Delta\theta_k \equiv \theta_k - \theta_{k-1}$ is chosen according to

$$\Delta\theta_k = \arg \min_{\pm\Delta\theta_k^{(j)}, j \in \{1,\ldots,K\}} \left(\pm\Delta\theta_k^{(j)} \right)^\tau \widehat{\nabla J}_k \quad (10)$$

The key idea of the proposed algorithm is to use a nonlinear estimator of the form (3)-(5) to estimate the effect of the candidate perturbations $\pm\Delta\theta_k^{(j)}$ to the objective function J and choose the perturbation that leads to the maximum – estimated – decrease of J. Condition (8) is imposed to make sure that the regressor vector ϕ_k is *persistently exciting*, see e.g., [12], a crucial property for the efficiency of the proposed algorithm. Finally, condition (9) is imposed to make sure that there exists at least one candidate perturbation $\pm\Delta\theta_k^{(j)}$ that leads to a non-negligible decrease of J.

Note that condition (8) renders the problem of finding $\Delta\theta_k^{(j)}$ a computationally hard problem. Fortunately, as we will see in Propositions 3 and 4, under appropriate choice of the regressor vector ϕ_k and selection of the vectors $\Delta\theta_k^{(j)}$ according to an appropriate random or pseudo-random generator, the constraint (8) holds with probability 1. In other words, under appropriate selection of the regressor vector ϕ_k and the random generator for producing $\Delta\theta_k^{(j)}$, there is no practical need to check the computationally "heavy" condition (8).

Our first result establishes efficient algorithm performance and convergence in the case of bounded – but otherwise arbitrary – exogenous signals.

Theorem 1. Suppose that

(A1) $|x_k| < c_x, |x_k - \bar{x}_k| < c_{\bar{x}}, |x_k - x_{k-1}| < c_{\Delta x}\ \forall k \in Z$ where $c_x, c_{\bar{x}}, c_{\Delta x}$ are finite nonnegative constants.

(A2) The algorithm (3)-(10) admits a solution satisfying (8), (9) for each $k \in Z$.

(A3) The algorithm (3)-(10) guarantees that $|\theta_k| \leq c_\theta < \infty, \forall k \in Z$.

(A4) The user-defined sequences α_k, c_k satisfy $\alpha_k \geq \alpha > 0$, $\bar{c} > c_k \geq \underline{c} > 0$ for all k.

Let also

$$\vartheta_k^* = \arg\min_{\vartheta} \sup_{x:|x|\leq c_x, \theta:|\theta|\leq c_\theta} |J(\theta, x) - \vartheta^\tau \phi_k(\theta, x)|$$

$$\nu_k(\theta, x) = J(\theta, x) - (\vartheta_k^*)^\tau \phi_k(\theta, x)$$

and

$$N_k = \sup_{x:|x|\leq c_x, \theta:|\theta|\leq c_\theta} |\nu_k(\theta, x)|$$

Then, the algorithm (3)-(10) guarantees that J_k is strictly decreasing (i.e., $J_k < J_{k-1}$) as long as $\theta_{k-1} \notin D_k$ where

$$D_k = \{\theta : |\nabla J(\theta, x_k)| \leq \varepsilon_k\}$$

with $\varepsilon_k = \frac{1}{\alpha_k} \max\left\{ c_k O(1) + O(N_k) + O(c_{\bar{x}}), \alpha_k^2 O(1) + O(c_{\Delta x}) \right\}$; moreover, θ_k converges to the subset D in finite time and remains there thereafter, where

$$D = \{\theta : |\nabla J(\theta, x_k)| \leq \varepsilon\}$$

with $\varepsilon = \frac{1}{\alpha} \max\left\{ \bar{c} O(1) + O(N_L) + O(c_{\bar{x}}), O(c_{\Delta x}) \right\}$ and $N_L = \sup_{k \geq L} \nu_k(\theta_k, x_k)$.

[**Assumptions (A1)-(A4)**] Assumption (A1) requires that the exogenous signal x_k is uniformly bounded; note that the proposed algorithm does not require knowledge of the bounds $c_x, x_{\bar{x}}, c_{\Delta x}$. We notice here that in many control applications such an assumption is quite realistic since x_k represents bounded signals such as exogenous disturbances, demand, reference signals, etc. However, there may be cases where such an assumption is not realistic; we are currently working on extending the results of this paper to the case where the exogenous signal x_k may be unbounded. Assumption (A2) is quite difficult to verify for a general choice of the regressor vector ϕ_k; however, as we establish in Propositions 3 and 4, if ϕ_k is chosen to be either a polynomial or a neural network of a specific structure, then assumption (A2) is trivially satisfied if the candidate perturbations $\Delta\theta_k^{(j)}$ are Bernoulli-like random terms. Assumption (A3) is imposed in order to avoid lengthy technicalities in the proof of our main results. It is not difficult for someone to see that all of the results of this paper are valid if we remove assumption (A3) and use a a projection mechanism as in [16, 5] or a resetting mechanism as in [9] for keeping θ_k bounded; similarly to [16, 5, 9] it can be seen that the introduction of such mechanisms does not destroy the performance and convergence properties of the proposed algorithm. Finally, assumption (A4) is a standard SA assumption on updating schemes with fixed step-sizes (see e.g. [4]).

[**Efficiency**] To better understand the efficiency of the proposed algorithm as compared to standard SA algorithms let $|\nabla J(\theta_{k-1}, x_k)| = B_k$; then, according to Theorem 1, we have that the proposed algorithm guarantees that

$$J_k < \begin{cases} J_{k-1} & \text{if } B_k > \varepsilon_k \\ J_{k-1} + \alpha_k B_k + O(c_x) n_\theta \alpha_k^2 + O(c_{\Delta x}) & \text{otherwise} \end{cases}$$

On the other hand, it it is not difficult for someone to see that in the case of standard SA algorithms, during the perturbation phase we have that the following equality holds with a non-negligible probability:

$$J_k = J_{k-1} + \alpha_k B_k + O(c_x)\, n_\theta \alpha_k^2 + O(c_{\Delta x})$$

In other words, in the case where θ_{k-1} is quite far from the local minimum of J, we have that the standard SA algorithms cannot avoid large "spikes" during their perturbation phase, while the proposed algorithm rejects candidate perturbations that may lead to such large "spikes".

The next result establishes the properties of the proposed algorithm if additionally to the boundedness of exogenous signals, we assume that these signals as well as the estimation error $x_k - \bar{x}_k$ are zero-mean. Note that in this case the sequences α_k, c_k are chosen to be slowly decaying to zero terms.

Theorem 2. Suppose that (A1)-(A3) hold and additionally that the following assumptions hold:

(A4′) The user-defined sequences α_k, c_k satisfy $\lim_{k\to\infty} \alpha_k = 0$, $\sum_{k=0}^{\infty} \alpha_k = \infty$, $\sum_{k=0}^{\infty} \alpha_k^2 < \infty$, $\sum_{k=0}^{\infty} \alpha_k c_k < \infty$, $\lim_{k\to\infty} \alpha_k / c_k = 0$.

(A5) $E[x_k - \bar{x}_k | G_{k-1}] = 0, E[x_k - x_{k-1} | G_{k-1}] = 0$ where G_k denotes the σ-field generated by $\{x_0, \ldots, x_k, \bar{x}_0, \ldots, \bar{x}_k, \Delta\theta_0^{(j)}, \ldots, \Delta\theta_k^{(j)}\}$.

Suppose moreover that T_h is chosen according to

$$T_h = k - L_k - \bar{L} \tag{11}$$

where $\bar{L}$ is any positive integer satisfying $\bar{L} \geq L$. Then, the algorithm (3)-(10) guarantees that there exists a positive integer $\bar{k}$ such that J_k is strictly decreasing if $\theta_{k-1} \notin D_k, k < \bar{k}$ and $\theta_{k-1} \in D_{\bar{k}}$ for all $k \geq \bar{k}$; moreover,

$$\lim_{k\to\infty} |\theta_k - \theta^*| = O(N_L), \text{ with probability 1} \tag{12}$$

where θ^* is any vector satisfying

$$E[\nabla J(\theta^*, x_k) | G_k] \to 0$$

[Assumptions (A4′), (A5)] Assumption (A4′) is a standard assumption on SA algorithms with vanishing gains. Assumption (A5) is also a quite standard assumption in SA (see e.g. [1]); note that in the case where x_k does not depend on θ_ℓ for $\ell < k$, assumption (A5) reduces to the requirement that $x_k - \bar{x}_k, x_k$ are zero-mean processes with bounded variance.

In the next two propositions we show that if the estimator (3) is chosen to be either an Incremental-Extreme Learning Machine (I-ELM) [10, 11] or a Polynomial-like approximator [14] and, moreover, the candidate perturbations $\Delta\theta_k^{(j)}$ are Bernoulli-like random terms, then there is no need to check the computationally heavy condition (8); moreover, Propositions 3 and 4 establish that the term N_L in the proofs of Theorems 1 and 2 can be made arbitrarily small by increasing the "size" L of the estimator (3).

Proposition 3. Suppose that each algorithm iteration the regressor vector ϕ_k is selected according to

$$\phi_{k,i}(\theta, x) = S(A_i^\tau \text{vec}(\theta, x) + b_i), i \in \{1, \ldots, L_k\} \tag{13}$$

where $S(\cdot)$ is an invertible smooth nonlinear function and the vectors A_i and the real parameters b_i are randomly generated (with A_i, b_i being zero-mean). Moreover assume that $\Delta\theta_k^{(j)}$ are random zero-mean vectors in $\{-\alpha_k, +\alpha_k\}^{n_\theta}$ satisfying [2] (9). Then, condition (8) is satisfied with probability 1. Moreover, the term N_L in Theorems 1 and 2 satisfies

$$N_L = O(1/L)$$

Proposition 4. Suppose that each algorithm iteration the regressor vector ϕ_k is selected according to

$$\begin{aligned}\phi_{i,k}(\theta, x) = S(\theta_1)^{d_{i,1}^\theta} \ldots S(\theta_{n_\theta})^{d_{i,n_\theta}^\theta} \qquad i \in \{1, \ldots, L_k\} \\ \times \bar{S}(x_1)^{d_{i,1}^x} \ldots \bar{S}(x_{n_x})^{d_{i,n_x}^x}\end{aligned} \tag{14}$$

where S is any smooth monotone function and $d_{i,j}^\theta, d_{i,j}^x$ are nonnegative integers such that

(TA1) $\bar{S}(\bar{x}_{k,1})^{d_{i,1}^x} \ldots \bar{S}(\bar{x}_{k,n_x})^{d_{i,n_x}^x} \neq 0$, $\forall k, i$ and moreover the integers $d_{i,j}^\theta$ in (14) are such that $\exists j \in \{1, \ldots, n_\theta\} : d_{i,j}^\theta > 0$, $\forall i \in \{1, \ldots, L_k\}$.

Moreover assume that $\Delta\theta_k^{(j)}$ are random zero-mean vectors in $\{-\alpha_k, +\alpha_k\}^{n_\theta}$ satisfying (9). Then, condition (8) is satisfied with probability 1. Moreover, the term N_L in Theorems 1 and 2 satisfies

$$N_L = O(1/L)$$

[Choice of Algorithm Parameters] Contrary to other applications of function approximators where the dimension L of an approximator of the form (3) should be large enough to guarantee that it can approximate nonlinear functions over the whole input set, this is not the case here: in the case of the proposed algorithm it is sufficient that the approximator has enough regressor terms to come up with an approximation of the unknown function J over a small neighborhood around the most recent vector θ_k. The size of this small neighborhood is proportional to α_k and therefore, roughly speaking, the smaller is the magnitude of α_k the less is the number of regressor terms L needed in the proposed algorithm. The above claim can be established by using the results of e.g., [17] where it is seen that the number of regressor terms needed to accommodate a desired approximation accuracy is proportional to the size of the set over which the approximation is taking place and by replacing the definition of the term N_L in Theorem 1 by the following one: $N_L = \sup_{\ell > L, \ell \in \{k-L,k\}} \nu_k(\theta_k, x_k)$.

Having the above in mind, a relatively small (as compared to other applications of function approximators) number L of regressor terms should suffice for efficient algorithm performance; similarly, since the approximation required is over a small neighborhood of the current value of θ_k a small time-window (determined by the parameter T_h in (5)) should be chosen. As a matter of fact, in all practical applications of algorithms using functions approximators for optimization purposes, (see [15]) as well as in various applications where we tested the proposed algorithm, a choice for L, T_h according to $L \approx 2(n_\theta + n_x), T_h = 50$ was found to produce quite satisfactory results. Moreover, in the case where a polynomial approximator is used, we found that it suffices to use a polynomial approximator of maximum order equal to 3 with randomly chosen poly-

[2] A choice $\Delta\theta_k^{(j)} = \alpha_k \Delta_k^{(j)}$ where $\Delta_k^{(j)}$ are Bernoulli random vectors satisfies (9); see also [3, 23] for construction of zero-mean random or random-like sequences that satisfy condition (9).

nomial terms at each algorithm iteration (i.e., $\sum_j d^{\theta}_{i,j} + \sum_\ell d^{x}_{i,\ell} = 3$ in (14) with $d^{\theta}_{i,j}, d^{x}_{i,j}$ randomly chosen).

We close this section, by noticing that for the choice of the step-sizes α_k, c_k similar rules as the ones apply in standard SA algorithms can be used.

3. SIMULATION EXPERIMENTS

In this section, we present simulation results on the application of the proposed algorithm to the fine-tuning of the urban traffic control strategy TUC as applied to a complex urban traffic network (traffic network of Chania, Greece). As reported in [15], SPSA failed to produce any meaningful results for this particular application, while the AFT algorithm, although in most scenarios was found to improve considerably the controller performance, it suffered from serious performance problems (spiky behavior at its perturbation phase) in some scenarios; in this section, we compare the performance of the proposed algorithm with the one of AFT for one of these scenarios. Next we briefly present some details regarding the particular application; it is worth noticing that the simulation setting is the same as the one of [15] used for the evaluation of the AFT algorithm. Due to space limitations, some details of the simulation experiments are not presented here; the interested reader is referred to [15] for a more detailed description.

Control System: The general form of the TUC controller [6, 15] can be described as follows:

$$g(t) = H(Lz(t-1)) \tag{15}$$

where $t = 0, 1, ...,$ denotes the discrete time-index with sampling time period equal to the traffic cycle time, g denotes the control input applied at the t-th cycle and z is a vector of traffic measurements. The control input corresponds to the green times of the junctions' stages and the vector z corresponds to a nonlinear vector function of the average occupancies (averaged over the last cycle) of the network links. The nonlinear operator H is used to guarantee that the control decisions satisfy minimum and maximum allowable green time constraints as well as that the sum of green times of the stages of each junction adds up to the cycle time minus the lost times (intergreens) of the particular junction; the entries of the matrix L correspond to the tuneable controller parameters. It is worth noticing that in the particular application treated here, $g \in \Re^{42}$ and $x \in \Re^{71}$ and thus the matrix L has a total of $42 \times 71 = 2982$ entries.

Traffic Network: Figure 1 displays the Chania urban traffic network, a typical urban traffic network containing all possible varieties of complex junction staging. The junctions are represented by nodes and the links are represented by arrows. Each network link corresponds to a particular junction stage.

Traffic Network Simulation: The macroscopic traffic simulation tool METACOR [8] was used for the simulation experiments. The reader is referred to [15] for more details on the network simulation setting.

Traffic Demand Scenarios: Ten different *basic traffic demand scenarios* $D_{i,t}$ (where $D_{i,t}$ is a vector whose entries are the number of vehicles entering the network origins at the t-th interval for the i-th scenario) with duration equal to 14 hours were designed based on actual measurements; each day of simulation a random perturbation (with average 5% of the nominal values) of one of the basic scenarios was used. It is worth noticing that the basic demand scenarios corresponded to highly congested traffic conditions and, moreover, that the variance among these scenarios was particularly high (namely, $|E[D_{i,t} - E[D_{i,t}]]| \approx 0.5E[D_{i,t}]$).

Performance Index: The average daily mean speed of the whole traffic network (in km/h) was used.

Tunable Parameters: As in [15], the 544 most important entries of the matrix L in (15) were fine-tuned. See [15] for more details.

Algorithm Design Parameters: The choice of the proposed algorithm design parameters was exactly same as that for the AFT algorithm used for the same application in [15]. Due to space limitations, the reader is referred to [15] for more details.

Estimation of Exogenous Signals: The exogenous vector x_k in this particular application corresponds to the traffic demand. As it was shown in [15], a low-dimension, noisy estimate of the traffic demand can be constructed based on traffic measurements at the networks origins; see [15] for more details.

Initial Controller Parameters: The initial L matrix used was designed by using TUC's methodology (which is based Linear-Quadratic control principles) assuming large uncertainty on the traffic network characteristics.

Figure 2 demonstrates the performance of the AFT (upper plot) and the proposed (lower plot) algorithms for the experiment setting described above. In Figure 2, the percentage of the mean speed improvement (as compared to the performance obtained using the initial L matrix for the particular demand scenario) is plotted.

Please notice that the AFT algorithm preserves a "spiky" behavior, especially at days 58 and 102, where its performance is more than 40% worse than that obtained using the initial L matrix; it is worth noticing that these big negatives spikes correspond to the emergence of grid-locks at the traffic network, where practically the vehicles stay blocked for hours.

By inspecting Figure 2 it is clearly seen that the proposed algorithm avoids the aforementioned "spiky" behavior of AFT while maintaining efficient convergence behavior. Also, note that the convergence of the algorithm is faster than the one of the AFT.

REFERENCES

[1] D. P. Bertsekas and J. N. Tsitsiklis (2000). Gradient convergence in gradient methods with errors, *SIAM Journal in Optimization*, 10, 627-642.

[2] J. R. Blum (1954). Multidimensional stochastic approximation methods. *Ann. Math. Statist.*, 25 , 737-744.

[3] S. Bhatnagar, M.C. Fu, S.I. Marcus, and I.J. Wang (2003). Two-Timescale Simultaneous Perturbation Stochastic Approximation Using Deterministic Per-

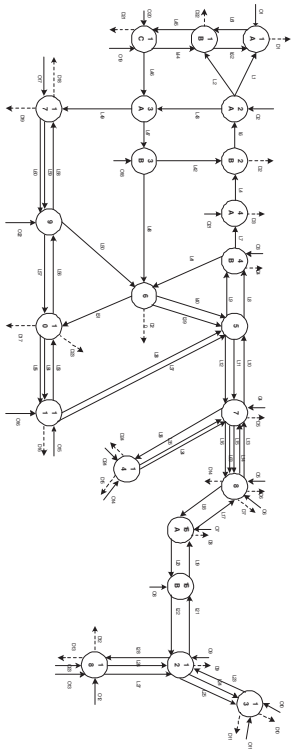

Fig. 1. The traffic network of the city of Chania.

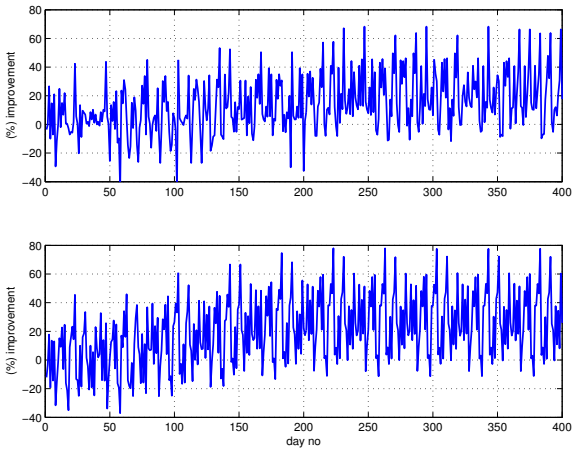

Fig. 2. Mean speed improvement (%) of the AFT algorithm (upper plot) and the proposed algorithm (lower plot).

turbation Sequences. *ACM Transactions on Modeling and Computer Simulation*, 13, 180-209.

[4] V.S. Borkar and S.P. Meyn (2000). O.D.E. method for convergence of stochastic approximation and reinforcement learning. *SIAM Journal in Optimization*, 38, 447-469.

[5] B. Deylon (1996). General results on the convergence of stochastic algorithms. *IEEE Transactions on Automatic Control*, 41, 1245-1255.

[6] C. Diakaki,V. Dinopoulou, K. Aboudolas, M. Papageorgiou, E. Ben-Shabat, E. Seider, and A. Leibov, "Extensions and new applications of the traffic signal control strategy TUC," *Transportation Research Record* no. 1856, pp. 202-216, 2003.

[7] J. Dippon and J. Renz (1997). Weighted means in stochastic approximation of of minima. *SIAM Journal of Control and Optimization*, 35, 1811-1827.

[8] N. Elloumi, H. Haj-Salem, and M. Papageorgiou, "METACOR: A macroscopic modeling tool for urban corridors," *TRISTAN II (Triennal Symposium on Transportation Analysis)*, Capri, Italy, Vol.1, pp. 135-150, June 23-28, 1994.

[9] L. Gerencer (1999). Convergence rate of moments in stochastic approximation with simultaneous perturbation gradient approximation and resetting. *IEEE Transactions on Automatic Control*, 44, 894-905.

[10] G.-B. Huang, L. Chen and C.K. Sew (2006). Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Transaction on Neural Networks*, 17, 879-892.

[11] G.-B. Huang and L. Chen (2007). Convex incremental extreme learning machine. *Neurocomputing*, 70, 3056-3062.

[12] P. A. Ioannou and J. Sun (1995). *Stable and Robust Adaptive Control*, Englewood Cliffs, NJ: Prentice Hall.

[13] J. Kiefer and J. Wolfowitz (1952). Stochastic estimation of the maximum of a regression function. *Ann. Math. Statist.*, 462-466.

[14] E. B. Kosmatopoulos, M. M. Polycarpou, M. A. Christodoulou, and P. A. Ioannou (1995). High-order neural network structures for identification of dynamical systems. *IEEE Transactions on Neural Networks*, vol. 6, 422-431.

[15] E. B. Kosmatopoulos, M. Papageorgiou, A. Vakouli and A. Kouvelas (2007). Adaptive fine-tuning of nonlinear control systems with application to the urban traffic control strategy TUC. *IEEE Transactions on Control Systems Technology*, vol. 15, no. 6, pp. 991-1002, November 2007.

[16] H. K. Kushner and D. S. Clark (1978). *Stochastic Approximation Methods for Constrained and Unconstrained Systems*. New York: Springer.

[17] V. Maiorov and R.S. Meir, (1998). Approximation bounds for smooth functions in $C(\Re^d)$ by neural networks and mixture networks. *IEEE Transactions on Neural Networks*, 9, 969-978.

[18] J.C. Spall (1992). Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, 37, 332-341.

[19] J.C. Spall and D.C. Chin, "Traffic-Responsive Signal Timing for System-Wide Traffic Control," *Transportation Research C*, vol. 5, pp. 153-163, 1997.

[20] J.C. Spall (2000). Adaptive stochastic approximation by the simultaneous perturbation method, *IEEE Transactions on Automatic Control*. 45 1839-1853.

[21] H. Robbins and D. Siegmund (1971) A convergence theorem for nonnegative almost supermartingales and some applications. *Optimizing Methods in Statistics* J. S. Rustari, ed., Academic Press, New York, 233-257.

[22] W. Muttay Wonham (1979). *Linear Multivariable Control*, Springer-Verlag, New York.

[23] X. Xiong, I.-J. Wang and M.C. Fu (2002). An asymptotic analysis of stochastic approximation with deterministic perturbation sequences. *Proceedings of the Winter Simulation Conference*, San Diego, CA, 285-291 .