Proceedings of the American Control Conference
San Diego, California • June 1999

ON THE USE OF SIMULTANEOUS PERTURBATION STOCHASTIC APPROXIMATION FOR NEURAL NETWORK TRAINING

A.VANDE WOUWER, C. RENOTTE, M. REMY

Laboratoire d'Automatique, Faculté Polytechnique de Mons, 31 Boulevard Dolez, 7000 Mons, Belgium
(email: vdw@autom.fpms.ac.be)

Abstract – Learning, i.e., estimation of weights and biases in neural networks, involves the minimization of a quadratic error criterion, a problem which is usually solved using back-propagation algorithms. This study, which is essentially experimental, aims at assessing the potential of first- and second-order simultaneous perturbation stochastic approximation (SPSA) algorithms to handle this minimization problem. To this end, several application examples in identification and control of nonlinear dynamic systems are presented. Test results, corresponding to training of neural networks possessing different structures and sizes, are discussed in terms of efficiency, accuracy, ease of use (parameter tuning), and implementation.

Keywords – stochastic approximation, neural networks, system identification, model predictive control.

1. Introduction

In recent years, neural networks (NNs) have attracted much attention for their potential to address a number of difficult problems in modeling and controlling nonlinear systems; see, e.g., [3], [16] and the references therein.

The most common NN structure is the multilayer feedforward (or static) NN, which is proved to be a universal approximator of static nonlinearities [2]. Learning, i.e., estimation of weights and biases, involves the minimization of a quadratic output error criterion. This non-convex optimization problem is usually handled using back-propagation (BP) [8] - or generalized delta rule - in which the error evaluated at the output layer is propagated back through the hidden layers and the input layer.

On the other hand, modeling of nonlinear dynamic systems involves more complex NN structures, e.g., time delay networks, recurrent networks, and interconnected neural networks and dynamic elements or submodels. For several of these NN structures, the static BP method can be extended, e.g., dynamic back-propagation [6, 7] and back-propagation through time [18] for recurrent networks. Recently, Ayoubi [1] proposed a NN structure with distributed dynamic linear elements embodied within the elementary processing units (called dynamic multilayer perceptron – DMLP) and developed a generalized dynamic delta-rule.

Although extensions of the BP approach can be developed for those NN structures, the resulting procedure can be more complicated to implement and to use. In addition, as NNs and dynamic elements could, in principle, be interconnected in arbitrary configurations, it is appealing to compute the gradient of the quadratic output error criterion in a more straightforward, numerical way. However, as neural networks usually involve a large number of unknown parameters, the evaluation of the criterion gradient by varying the parameters one at a time, as it is required in finite difference approximations, would be extremely costly.

In contrast to standard finite differences, the simultaneous perturbation (SP) gradient approximation proposed by Spall [9] makes use of a very efficient technique based on a simultaneous (random) perturbation in all the parameters. This approach was first used for gradient estimation in a first-order stochastic approximation (SA) algorithm [9], and more recently for Hessian estimation in an accelerated second-order SPSA algorithm [10, 11].

These algorithms seem particularly well suited to the NN learning problem, and in this connection, Spall and Cristion developed a direct adaptive control scheme using a feedforward NN controller [13, 14, 15]. In this case, the SPSA-based learning algorithm has the advantage that it does not require the prior knowledge of a process model to evaluate the criterion gradient. Since then, this direct adaptive control scheme has been the subject of several simulation studies, e.g., [4, 5].

This study is essentially experimental and, based on several application examples, aims at assessing the potential of the above-mentioned first- and second-order SA algorithms to address the problem of parameter estimation in NNs. From a practical point of view, the important issues of convergence, accuracy, computational load, ease of use (tuning), and simplicity of implementation are discussed.

The paper is organized as follows. Section 2 introduces the basic principles of the SA algorithms considered in this study. In Section 3, these algorithms are applied to the training of two NN configurations used for modeling nonlinear dynamic systems, e.g., a series-parallel NN and a DMLP. In addition, a control application is discussed, in which the parameters of a predictive PID controller, implemented in the form of a feedforward NN, are adjusted using SPSA. Finally, Section 4 is devoted to some conclusions.

0-7803-4990-6/99 $10.00 © 1999 AACC

2. The SA algorithms

We consider the problem of minimizing a mean-square output error criterion with respect to a vector θ of unknown parameters (typically the weights and biases of a NN)

$$\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i(\theta))^2 \qquad (1)$$

where y_i is the desired output and $\hat{y}_i$ is the output produced by the NN depending on the parameters θ. In this work, the minimization problem (1) is handled using several SA algorithms, which have been implemented in MATLAB .m files.

The first-order 1SPSA algorithm is given by the following core recursion for the parameter vector θ [9]

$$\hat{\theta}_k = \hat{\theta}_{k-1} - a_k \, \hat{g}_k(\hat{\theta}_{k-1}) \qquad (2)$$

in which a_k is a positive scalar gain coefficient satisfying certain conditions, and $\hat{g}_k(\hat{\theta}_{k-1})$ is an approximation of the criterion gradient obtained by varying all the elements of $\hat{\theta}_{k-1}$ simultaneously, i.e.,

$$\hat{g}_k(\theta_{k-1}) = \begin{bmatrix} \dfrac{J(\hat{\theta}_{k-1} + c_k\Delta_k) - J(\hat{\theta}_{k-1} - c_k\Delta_k)}{2c_k\Delta_{k1}} \\ \cdots \\ \dfrac{J(\hat{\theta}_{k-1} + c_k\Delta_k) - J(\hat{\theta}_{k-1} - c_k\Delta_k)}{2c_k\Delta_{kp}} \end{bmatrix} \qquad (3)$$

where c_k is a positive scalar and Δ_k is a user-generated zero-mean random vector satisfying certain regularity conditions (typically, Δ_k is symmetrically Bernouilly distributed).

Important to note is that this gradient estimate differs from usual finite difference approximations in that the number of criterion evaluations is not proportional to the number of unknown parameters. Instead, only two evaluations of the criterion are required.

The recursion (2) can also be based on a smoothed gradient approximation [13]

$$G_k = \rho_k G_{k-1} + (1-\rho_k)\hat{g}_k(\hat{\theta}_{k-1}), \qquad G_0 = 0 \qquad (4)$$

where $\hat{g}_k(\hat{\theta}_{k-1})$ is the simultaneous perturbation gradient estimate (3) and $0 \le \rho_k \le 1$. This procedure denoted 1SPSA-GS is analogous to the momentum approach in BP. Alternatively, if direct evaluations of the criterion gradient are available (usually with some added noise), recursion (2) defines a stochastic gradient (SG) algorithm, denoted 1SGSA in the following.

The second-order algorithms 2SPSA (and 2SGSA) are based on the following two core recursions, one for the parameter vector θ, the second for the Hessian $H(\theta)$ of the criterion [11].

$$\hat{\theta}_k = \hat{\theta}_{k-1} - a_k \, \overline{\overline{H}}_k^{-1} \, \hat{g}_k(\hat{\theta}_{k-1}), \qquad \overline{\overline{H}}_k = f_k(\overline{H}_k) \qquad (5)$$

$$\overline{H}_k = \frac{k}{k+1}\overline{H}_{k-1} + \frac{1}{k+1}\hat{H}_k \qquad (6)$$

where $\hat{H}_k$ is a per-iteration estimate of the Hessian matrix, which is computed from gradient approximations (or direct evaluations) using a simultaneous perturbation approach, $\overline{H}_k$ is a simple sample mean, and f_k is a mapping designed to cope with possible non-positive-definiteness of $\overline{H}_k$.

Again, the algorithm requires only a small number of function evaluations - at least four criterion evaluations to construct the gradient and Hessian estimates, or three gradient evaluations in the SG case - independent of the number of unknown parameters.

Important implementation issues of these SA algorithms include initialization, choice of the gain sequences $\{a_k\}$ and $\{c_k\}$ (usually these sequences are chosen in the form $a_k = a(A+k+1)^{-\alpha}$ and $c_k = c(k+1)^{-\gamma}$), gradient/Hessian averaging and step rejection; see [11].

Usually, 1SPSA appears as an efficient, robust algorithm, but suffers from the classical drawback of first-order algorithms, i.e., a slowing down in the convergence as an optimum is approached. 2SPSA includes second-order effects with the aim of accelerating convergence. However, tuning of the several algorithm coefficients is a more delicate task, particularly in noisy environment.

3. Numerical Results

3.1 System identification using a series-parallel NN

First, a NN in series-parallel mode (or feedforward time delay NN) is used as a prediction model for a process given in [6]

$$y(k) = \frac{y(k-1)y(k-2)y(k-3)u(k-2)(y(k-3)-1)+u(k-1)}{1+y(k-2)^2+y(k-3)^2} \qquad (7)$$

where u(k) and y(k) denote the input and output sequences, respectively.

Obviously, this simple example does not require any special learning procedure since the application of BP is straightforward. Hence, first- and second-order methods as implemented in the MATLAB NN Toolbox could certainly be used. Our objective at this stage is to assess the relative performance of the SA algorithms presented in Section 2.

The training set consists of 997 data produced by (7) using an input sine wave with increasing frequency (from 0.1 to 10 Hz). Test sets used for model validation are produced with other input signals, e.g., a switchback signal. Based on the assumption that we know the correct system orders, the process is modeled using a NN with 5 inputs $[y(k-1), y(k-2), y(k-3), u(k-1), u(k-2)]$, one hidden layer and one output. In order to select the number of nodes in the hidden layer n_{hl} and to define "reference" results, we first use a Levenberg-Marquardt (LM) algorithm to estimate the parameter sets corresponding to NNs with n_{hl} ranging from 1 to 15. As a result, we see that the quadratic error criterion is monotonically decreasing and that the resulting NNs have good generalization (validation) properties. Compromising model accuracy and size (a measure of this compromise is given by Akaike's criterion), the "best" model corresponds to a NN with $n_{hl} = 7$.

In all these cases, 1SPSA, 1SPSA-GS and 2SPSA perform successfully. Generally, the SPSA algorithms can be rated as follows with regards to speed of convergence and accuracy: (1) 2SPSA - (2) 1SPSA-GS - (3) 1SPSA. In the case of NN with $n_{hl} = 7$ and a minimization of the error criterion carried out with 2SPSA, identification and cross-validation results are illustrated in Fig. 1.

As the number of nodes in the hidden layer increases from 1 to 15, the results obtained with the three SPSA algorithms become almost the same, i.e., the accuracy of the results produced by 1SPSA and 1SPSA-GS gradually improves, the improvement being more noticeable for 1SPSA, while the accuracy of the results produced by 2SPSA, which initially also improves, slightly deteriorates for larger values of n_{hl}. We can only conjecture about this last observation, which could be interpreted as a deterioration of the Hessian estimate due to NN over-parametrization.

Convergence and accuracy are usually better and less sensitive to the NN structure when the gradient information, which is readily available in this example, is used in 1SGSA or 2SGSA algorithms. Our numerical experiments show that 1SGSA produces results equivalent to standard BP, while 2SGSA slightly supersedes BP with momentum and adaptive learning rate, as implemented in the MATLAB NN Toolbox. As compared to SP-algorithms, 1SGSA and 2SPSA display similar performance.

From a computational point of view, the estimation and regularization of the Hessian estimate used in 2SPSA can become very costly for larger numbers of parameters. In our application, the CPU time required by 2SPSA ranges from 5 to 10 times the CPU required by 1SPSA, so that, in terms of efficiency, the use of 2SPSA might be questionable.

Following an idea of Spall [12], an attempt is made to reduce the computational expense by evaluating only a diagonal estimate of the Hessian matrix. Indeed, we observe a reduction of about 40% in the computation time, which is due to savings in the evaluation of the Hessian estimate, as well as in the recursion on θ that only requires a trivial matrix inverse. The performance, in terms of rate of convergence and accuracy, remains almost unchanged, which demonstrates that the diagonal Hessian estimate still captures potential large scaling differences in the elements of θ. This algorithm variation is denoted 2SPSA-DH.

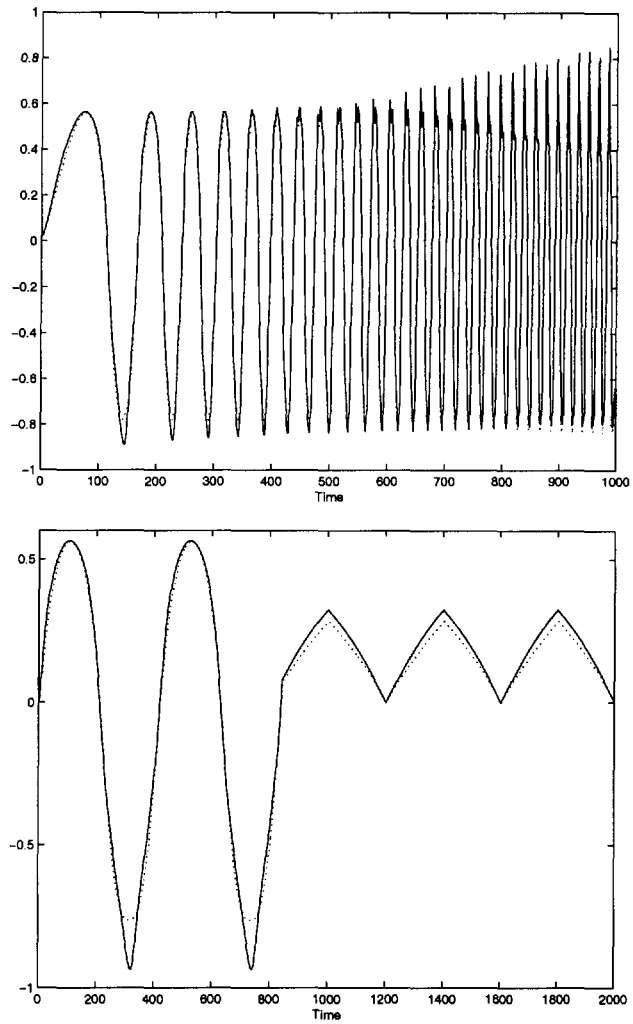

Figure 1. Identification (above) and cross-validation (below) results (solid line: real system output; dotted line: NN output)

Table I compares, for ten independent runs starting from the same initial parameter estimates θ_0, the results obtained with each algorithm in the case of a NN with $n_{hl} = 7$. This table gives the minimum (best), maximum (worst) and average values of the mean-square error criterion after 3000 iterations. It is apparent that algorithms based on a SP-approximation of the gradient and/or the Hessian produce more "dispersed" results, i.e., display a larger deviation between the worst and best cases than the SG(BP)-algorithms. Figure 2 shows the average mean-square error curves for the SP-algorithms. Note that, even though the average performance of 1SPSA and 1SPSA-GS are virtually identical, the results produced by 1SPSA are more dispersed from one run to another; see Table I.

Table I. Mean-square errors obtained after 3000 iterations (10 independent runs – NN with n_{hl} = 7)

	Best RMS	Worst RMS	Average RMS
1SPSA	0.0051	0.023	0.012
1SPSA-GS	0.011	0.019	0.013
2SPSA	0.0015	0.0030	0.0021
2SPSA-DH	0.0012	0.0067	0.0025
1SGSA	0.0016	0.0016	0.0016
2SGSA	0.000092	0.00030	0.00014
BP	0.0016	0.0016	0.0016
Adaptive BP-GS	0.00023	0.00023	0.00023
LM	0.00000024	0.00000024	0.00000024

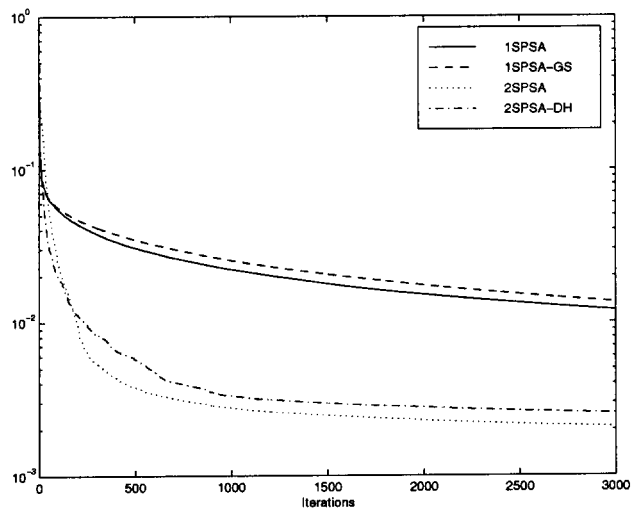

Figure 2. Average mean-square error curves (NN with n_{hl}=7)

3.2 System identification using a DMLP

DMLPs incorporate dynamic linear elements within each elementary processing units and so, allow nonlinear dynamic systems to be modeled without the need for recursion. As the number of unknown parameters grows rapidly with the NN input dimension (and so, in a recurrent NN, with the number of delayed inputs), DMLPs will be particularly advantageous in reducing the size of the unknown parameter set.

We consider the problem of modeling a nonlinear process given in [17]

$$y(k) = \frac{0.875y(k-1) + u(k-1)}{1 + y^2(k-1)} \qquad (8)$$

using a DMLP with 1 input, 12 nodes in the hidden layer and 1 output. The hidden and output nodes are associated with first-order dynamic elements, so that there are n_p = 50 unknown parameters to estimate.

Figure 3 shows identification results obtained with 1SPSA-GS. Although these results are satisfactory, it can be observed that the dynamics of the process is not very accurately captured, which might appear paradoxical since DMLPs are, in principle, well-suited to this problem. In fact, SP-algorithms experience difficulties with this NN structures, and the gradient information, which is readily available in this application, must be used to get better performance.

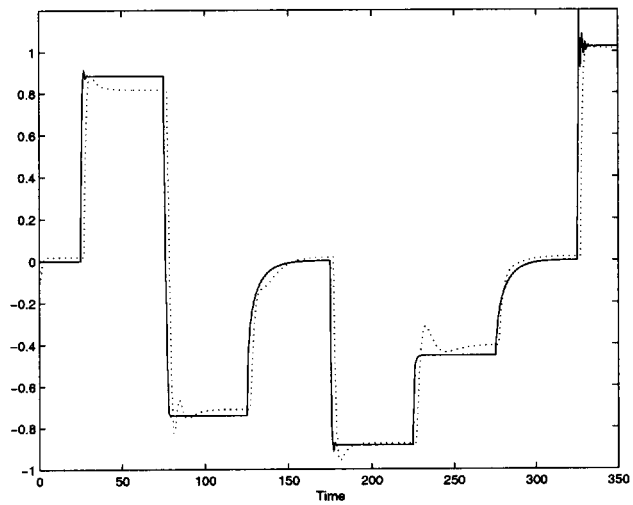

Figure 3. Identification results using 1SPSA-GS (solid line: real system output; dotted line: NN output)

3.3 A model-based predictive PID controller

A predictive PID control scheme, as depicted in Fig. 4, is designed for the process (8) considered in section 3.2. The PID controller is implemented using a feedforward NN with three inputs $\left[e(k), \sum_{i=1}^{k} e(i), e(k) - e(k-1)\right]$, one hidden layer with 2 nodes and one output. The 11 parameters of this NN controller are adjusted in order to minimize a receding horizon criterion in the form

$$J = \sum_{i=1}^{N} \left(y^r(k+i) - y^m(k+i)\right)^2 + \sum_{i=1}^{N} \lambda \Delta u(k+i-1)^2 \qquad (9)$$

In this expression, $y^r(k)$ is the output of a second-order model reference with $\xi = 0.8$ and $\omega_n = 0.1$, while $y^m(k)$ is the output of a system emulator, which can be in the form of an NN model (for the sake of simplicity, the emulator is assumed perfect here) and provides a prediction of the process output over a horizon N = 25. The weighting factor $\lambda = 0.1$ penalizes the control increments.

In this application, the requirements on the optimization algorithm are: (a) small computational costs so that the criterion minimization can take place within a sampling period, (b) robustness and reliability so that, even in the presence of perturbations, at least a lower value of the error criterion can be achieved. On the other hand, accuracy in estimating the controller parameters is not determinant. With this view, the minimization of the receding horizon criterion (9) is performed using 1SPSA-GS, with a maximum of 20 iterations per sampling interval. Figure 5

illustrates the excellent tracking capabilities of the NN PID controller.

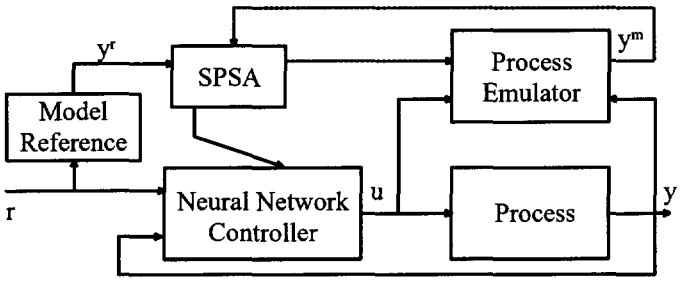

Figure 4. Model predictive control scheme

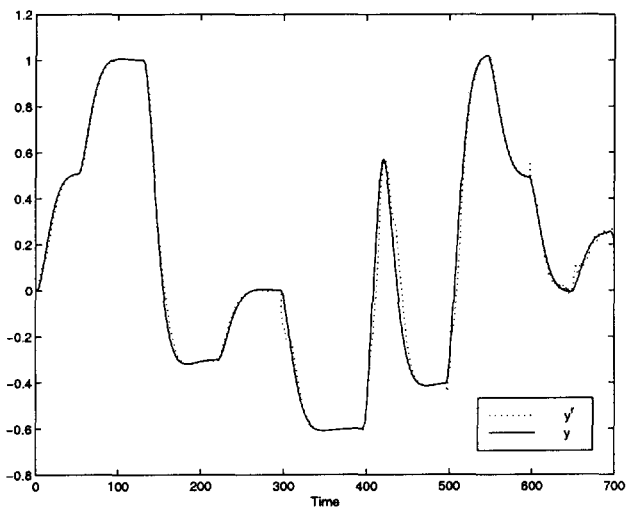

Figure 5. Performance of the NN predictive controller

4. Concluding Remarks

The SP-approach devised by Spall [9-11] is a very powerful technique, which allows an approximation of the gradient and/or the Hessian of the objective function to be computed by effecting simultaneous random perturbations in all the parameters. This paper shows the potential of SGSA and SPSA algorithms to efficiently address the parameter estimation problem in NNs, and in this connection, discusses several test-results in nonlinear system identification and control. Remarkable properties of these algorithms is their simplicity of implementation and ease of use. Especially, the application of SPSA algorithms in the area of direct and indirect control (or inverse modeling) is very appealing.

Acknowledgements - The authors are very grateful to Dr. James Spall for his insightful comments and suggestions.

References

[1] Ayoubi M., *Nonlinear System Identification Based on Neural Networks with Locally Distributed Dynamics and Application to Technical Processes*, VDI-Verlag, Düsseldorf, (1996).

[2] Funahashi K.-I., "On the Approximate Realization of Continuous Mappings by Neural Networks", *Neural Networks* **2**, 183-192, (1989).

[3] Hunt K.J., Sbardaro D., Zbikowski R., Gawthrop P.J., "Neural Network for Control Systems – A Survey", *Automatica* **28**, 1083-1112, (1992).

[4] Ji X.D., Familoni B.O., "Experimental Study of Direct Adaptive SPSA Control System with Diagonal Recurrent Neural Network Controller", *Proc. IEEE SoutheastCon'96*, 525-528, (1996).

[5] Maeda Y., Figueiredo R.J.P., "Learning Rules for Neuro-Controller via Simultaneous Perturbation", *IEEE Trans. Neural Networks* **8**, 1119-1130, (1997).

[6] Narendra K.S., Parthasarathy K., "Identification and Control of Dynamical Systems Using Neural Networks", *IEEE Trans. Neural Networks* **1**, 4-27, (1991).

[7] Narendra K.S., Parthasarathy K., "Gradient Methods for the Optimization of Dynamical Systems Containing Neural Networks", *IEEE Trans. Neural Networks* **2**, 252-262, (1991).

[8] Rumelhart D.E., Hinton G.E., Williams R.J., "Learning Representations by Back-propagating Errors", *Nature* **323**, 533-536, (1986).

[9] Spall J.C., "Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation", *IEEE Trans. Automat. Contr.* **37**, 332-341, (1992).

[10] Spall J.C., "Accelerated Second-Order Stochastic Optimization Using Only Function Measurements, *Proc. IEEE Conference on Decision and Control*, 1417-1424, (1997).

[11] Spall J.C., "Adaptive Stochastic Approximation by the Simultaneous Perturbation Method", submitted to *IEEE Trans. Automat. Contr.*, (1998).

[12] Spall J.C., personal communication, (1999).

[13] Spall J.C., Cristion J.A., "Nonlinear Adaptive Control Using Neural Networks: Estimation with Smoothed Form of Simultaneous Perturbation Gradient Approximation", *Statistica Sinica* **4**, 1-27 (1994).

[14] Spall J.C., Cristion J.A., "A Neural Network Controller for Systems with Unmodeled Dynamics with Applications to Wastewater Treatment", *IEEE Trans. Syst., Man, Cybern.* **27**, 369-375 (1997).

[15] Spall J.C., Cristion J.A., "Model-Free Control of Nonlinear Stochastic Systems with Discrete-Time Measurements", *IEEE Trans. Automat. Contr.* **43**, 1198-1210 (1998).

[16] Suykens J., Vandewalle J., De Moor B., *Artificial Neural Networks for Modelling and Control of Non-Linear Systems*, Kluwer, (1996).

[17] Tan Y., "An Architecture for Adaptive Neural Control", *Journal A* **34**, 12-16, (1993).

[18] Werbos P.J., "Backpropagation Through Time: What it does and how to do it", *Proc. IEEE* **78**, 1550-1560, (1990).