Proceedings of the American Control Conference
Chicago, Illinois • June 2000

NEURAL MODELING AND CONTROL OF A HEAT EXCHANGER BASED ON SPSA TECHNIQUES

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

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

Abstract – The aim of this paper is twofold: first, we consider a variation of the first-order simultaneous perturbation stochastic approximation (SPSA) algorithm developed by J. Spall [5-6] which makes use of several numerical artifices, including adaptive gain sequences, gradient smoothing and a step rejection procedure, to enhance convergence and stability. Second, we present numerical studies on a non-trivial test-example, i.e., the water cooling of sulfuric acid in a two-tank system. This numerical evaluation includes the development of a neural model as well as the design of a model-based predictive neural PID controller.

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

1. Introduction

Over the past several years, neural networks (NNs) have been increasingly applied to the identification and control of nonlinear systems; see, e.g., [2], [7] and the references therein.

A basic model structure for static nonlinearities is the multilayer feedforward NN, in which learning, i.e., estimation of weights and biases, involves the minimization of a quadratic output error criterion $J(\theta)$ using back-propagation (BP) [4], an analytical procedure in which the error evaluated at the output layer is propagated back through the hidden layers and the input layer.
Although the BP method can be generalized for more complex NN structures, which are useful in modeling dynamic nonlinear systems, the resulting algorithms are usually more complicated to implement and more computationally demanding. Hence, it is appealing to develop a more straightforward, numerical procedure for computing the gradient of the quadratic output error criterion. However, as NNs 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 conventional finite difference approximations, would be extremely costly.

In contrast to standard finite differences, the simultaneous perturbation (SP) approximation of the gradient proposed by Spall [5] makes use of a very efficient technique based on a simultaneous (random) perturbation in all the parameters and requires only two evaluations of the criterion. This approach has first been applied to gradient estimation in a first-order stochastic approximation (SA) algorithm [5], and more recently to Hessian estimation in an accelerated second-order SPSA algorithm [6].

In a previous work [9], the authors have applied the above-mentioned first- and second-order SA algorithms (1SPSA and 2SPSA) to several test-examples in order to evaluate their potential to address the problem of parameter estimation in NNs. Efficiency, simplicity of implementation and very modest computational costs make 1SPSA particularly attractive, even though it suffers from the classical drawback of first-order algorithms, i.e., a slowing down in the convergence as an optimum is approached.

In this study, a variation of this first-order algorithm is considered which makes use of adaptive gain sequences, gradient smoothing and a step rejection procedure, to enhance convergence and stability. To demonstrate the algorithm efficiency and versatility, attention is focused on a realistic application example, e.g. the development of a predictive control scheme for a two-tank cooling system. This predictive controller is based on three main components [2] :

- a process emulator in the form of a neural state space model [7], which generates prediction of the future process outputs over a specified horizon;
- a NN controller, with a PID-like input-output parametrization [8];
- an optimization procedure to train the NN model (off-line) and the NN controller (on-line).

The remainder of this paper is organized as follows. Section 2 introduces the basic principle of the SPSA algorithm used throughout this study. In section 3, the model-based predictive control scheme is described in some details, including the structure of the process emulator and controller. In section 4, these tools are applied to the problem of controlling the output temperature of sulfuric acid in a two-tank cooling system. The performance of this control scheme in the face of non measurable disturbances in the acid inlet temperature and noisy output acid temperature measurements is investigated. Finally, section 5 is devoted to some concluding remarks.

0-7803-5519-9/00 $10.00 © 2000 AACC

2. A first-order SPSA algorithm

Consider the problem of minimizing a, possibly noisy, objective function $J(\theta)$ with respect to a vector θ of unknown parameters (in this study, the weights and biases of a NN).

1SPSA is given by the following core recursion for the parameter vector θ [5-6]

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

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

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

where c_k is a positive scalar and $\Delta_k = (\Delta_{k1}, \Delta_{k2}, \ldots, \Delta_{kp})^T$ with symmetrically Bernouilli distributed random variables $\{\Delta_{ki}\}$.

In its original formulation, 1SPSA makes use of decaying gain sequences $\{a_k\}$ and $\{c_k\}$ in the form

$$a_k = \frac{a}{(A+k+1)^{\alpha}}, \qquad c_k = \frac{c}{(k+1)^{\gamma}} \qquad (3)$$

Numerical experiments show that the algorithm may therefore get stuck somewhere in the parameter space if the criterion value becomes significantly worse (due to a poor current gradient approximation, a non-convex optimization problem,...) and the gain sequences are too small to recover from this situation.

In order to enhance convergence and stability, the use of an adaptive gain sequence for parameter updating is considered in this study, i.e.,

$$\begin{aligned} a_k &= \eta\, a_{k-1}, & \eta &\geq 1, & \text{if } J(\theta_k) &< J(\theta_{k-1}) \\ a_k &= \mu\, a_{k-1}, & \mu &\leq 1, & \text{if } J(\theta_k) &\geq J(\theta_{k-1}) \end{aligned} \qquad (4)$$

where, typically, $\eta = 1.05$ and $\mu = 0.7$.

In addition to gain attenuation when the value of the criterion becomes worse, "blocking" mechanisms [6] are also applied, i.e., the current step is rejected and, starting from the previous parameter estimate, a new step is accomplished (with a new gradient evaluation and a reduced updating gain).

A constant gain sequence $c_k = c$ is used for gradient approximation, the value of c being selected so as to overcome the influence of (numerical or experimental) noise.

Finally, a gradient smoothing (GS) procedure is implemented, i.e., gradient approximations are averaged across iterations in the following way

$$G_k = \rho_k\, G_{k-1} + (1-\rho_k)\,\hat{g}_k(\hat{\theta}_{k-1}),\ 0 \leq \rho_k \leq 1,\ G_0 = 0 \quad (5)$$

where, starting with a typical value of $\rho = 0.95$, ρ_k is decreased in a way similar to (4) when step rejection occurs (i.e. $\rho_k = \mu\rho_{k-1}$ with $\mu \leq 1$) and is reset to its initial value after a successful step.

As the following simple numerical example illustrates, the use of these numerical artifices, i.e., adaptive gain sequences, step rejection procedure and gradient smoothing, significantly improves the effective practical performance of the algorithm (which, in the following, is denoted "adaptive 1SP-GS").

Test-example: we consider the problem of modeling a nonlinear process given by

$$y(k) = \frac{0.875y(k-1) + u(k-1)}{1 + y^2(k-1)} \qquad (6)$$

using a dynamic multilayer perceptron (DMLP) [1] with 1 input, 4 nodes in the hidden layer and 1 output. The hidden and output nodes are associated with second-order dynamic elements, so that there are $n_p = 33$ unknown parameters to estimate.

Table 1 compares, in terms of number of iterations, computational load (normalized CPU with the CPU required by 1SPSA as reference) and mean square error (RMS), the performance of

- The original 1SPSA algorithm (equations 1-3 with a step rejection procedure);
- Adaptive 1SP-GS (equations 1, 2, 4, 5);
- Adaptive 1GBP-GS (same as above with a gradient evaluated analytically using generalized back-propagation);

Table 1. Computational statistics

	Iterations	CPU	RMS
1SPSA	8000	1	0.00326 - 0.00431
Adaptive 1SP-GS	8000	1.3	0.00309
Adaptive 1GBP-GS	600	7.2	0.00310

Clearly, generalized back-propagation (GBP) is very efficient in terms of number of iterations required to achieved a certain level of accuracy, but it is computationally expensive as it requires the solution of dynamic sensitivity equations at each iteration. On the other hand, 1SPSA has very modest computational requirements, but produces relatively dispersed results (0.00326 – 0.00431 represents the range of values obtained from 10 independent runs starting from the same initial parameter estimates). The main advantage of our algorithm is that it retains the very modest computational requirement of 1SPSA and usually provides less dispersed, more accurate results.

3. NN modeling and control

When modeling nonlinear dynamic systems, several alternative NN architectures can be used (see, e.g. [2]). For the application considered in section 4, a neural state space model, as introduced by Suykens [7], is selected

$$\begin{aligned}\hat{x}_{k+1} &= W_{AB}\tanh(V_A\,\hat{x}_k + V_B\,u_k + \beta_{AB})\\ \hat{y}_k &= W_{CD}\tanh(V_C\,\hat{x}_k + V_D\,u_k + \beta_{CD})\end{aligned} \qquad (7)$$

As illustrated in Fig. 1, neural state space models are recurrent NNs. The dimensions of the weight matrices and bias vectors are $W_{AB} \in \Re^{n \times n_{hx}}$, $V_A \in \Re^{n_{hx} \times n}$, $V_B \in \Re^{n_{hx} \times m}$, $\beta_{AB} \in \Re^{n_{hx}}$, $W_{CD} \in \Re^{\ell \times n_{hy}}$, $V_C \in \Re^{n_{hy} \times n}$, $V_D \in \Re^{n_{hy} \times m}$, $\beta_{CD} \in \Re^{n_{hy}}$, where n, m, l, n_{hx} and n_{hy} are the number of states, inputs, outputs and hidden neurons, respectively.

These unknown parameters are estimated off-line from a set of N real system outputs by minimizing an output error least-square (OLS) criterion, i.e.

$$\min_{\theta} I(\theta) = \min_{\theta} \sum_{i=1}^{N} (y_i - \hat{y}_i(\theta))^2 \qquad (8)$$

Once this task has been achieved, the neural state space model can be used as a process emulator in a model-based predictive control scheme [2] (Fig. 2). The NN state space model generates prediction of future process outputs over a specified prediction horizon, which allows a quadratic performance criterion to be defined, i.e.

$$J = \sum_{i=N_1}^{N_2} \left(y^r(k+i) - \hat{y}(k+i)\right)^2 + \lambda \sum_{i=1}^{N_2} \Delta u(k+i-1)^2 \qquad (9)$$

where $y^r(k)$ is the output of a model reference, N_1 and N_2 define the horizons over which the tracking errors $e(k) = y^r(k) - \hat{y}(k)$ and control increments $\Delta u(k)$ are considered. The weighting factor λ penalizes the control increments.

The control signal u(k) can be produced in two ways:

- An on-line optimization routine is applied to the minimization of J with respect to the control moves u(k) over the prediction horizon, and the optimal control signal is directly applied to the process;
- The minimization of J is accomplished with respect to the weights of a feedforward NN controller, which in some sense mimics the action of the on-line optimization routine.

In this study, the second approach has been chosen as it is conceptually more robust. Indeed, the NN controller keeps track of the optimization procedure, and can produce a control signal even during periods of time where optimization cannot be performed satisfactorily (e.g., in an on-line application, higher priority events and interrupts can prevail on optimization). In our implementation, controller training occurs only during the transient phases, in which the system is sufficiently excited by the input signals. Optimization is suspended in steady-state phases to avoid detuning the NN controller, which produces the input signal applied to the process.

Following the line of thought in [8], the NN controller parametrization is chosen in a similar way as in a classical PID controller, i.e., with three inputs $i_1(k) = e(k)$, $i_2(k) = \sum_{i=1}^{k} e(i)$ and $i_3(k) = e(k) - e(k-1)$, one hidden layer with n_{hl} nodes and one output. As only a weighted sum of $i_1(k)$, $i_2(k)$, $i_3(k)$ is needed to mimic the PID algorithm, the biases of the NN controller are set to zero.

Of course, the prediction of the NN emulator is usually not perfect, and the effect of modeling errors can be treated as an additive disturbance which can be estimated at the kth sampling instant in a manner similar to Dynamic Matrix Control (DMC), i.e.,

$$\begin{aligned}y^m(k) &= \hat{y}(k) + d(k-1)\\ d(k) &= y(k) - \hat{y}(k)\end{aligned} \qquad (10)$$

$y^m(k)$ is substituted for $\hat{y}(k)$ in the expression of the performance criterion (9), which allows modeling errors to be compensated.

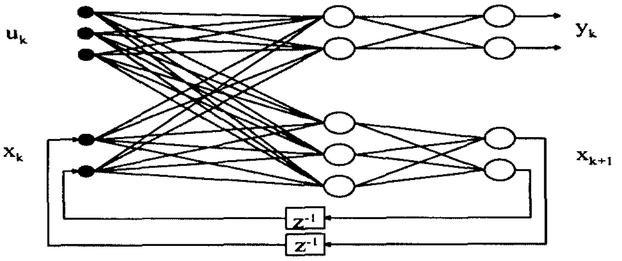

Figure 1. Neural state space model

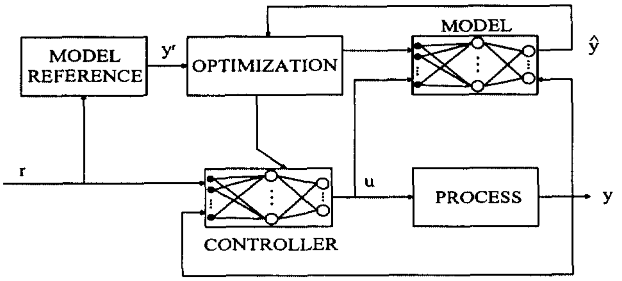

Figure 2. Model-based predictive neural control scheme

4. Case study: cooling of sulfuric acid in a two-tank system

In the remaining of this study, the usefulness of our adaptive 1SP-GS algorithm is illustrated with a numerical application, i.e., neural modeling and control of a two-tank system used to cool sulfuric acid with a countercurrent water stream [3].

4.1 Process description

A nonlinear mathematical model of the cooling system (see Fig. 3.) is used for producing simulated data. This model can be derived by expressing the energy balance on each tank, i.e.,

$$M_{T,1}C_{pa}\dot{T}_{a,1} = \dot{M}_w C_{pw}(T_{w,2} - T_{w,1}) + \dot{M}_a C_{pa}(T_{a,in} - T_{a,1})$$

$$M_{T,2}C_{pa}\dot{T}_{a,2} = \dot{M}_w C_{pw}(T_{w,in} - T_{w,2}) + \dot{M}_a C_{pa}(T_{a,1} - T_{a,2}) \qquad (11)$$

where $M_{T,i}$ is the weight of acid in tank i (i =1, 2), C_{pa} (C_{pw}) is the acid (water) specific heat, $\dot{M}_a$ ($\dot{M}_w$) is the acid (water) mass flow rate, $T_{a,i}$ ($T_{w,i}$) is the acid (water) temperature from tank i.

Heat transfer is modeled through a log-mean delta T, i.e.,

$$\dot{M}_w C_{pw}(T_{w,1} - T_{w,2}) = k_1 A_1 \frac{(T_{a,1} - T_{w,1}) - (T_{a,1} - T_{w,2})}{\ln\dfrac{(T_{a,1} - T_{w,1})}{(T_{a,1} - T_{w,2})}}$$

$$\dot{M}_w C_{pw}(T_{w,2} - T_{w,in}) = k_2 A_2 \frac{(T_{a,2} - T_{w,2}) - (T_{a,2} - T_{w,in})}{\ln\dfrac{(T_{a,2} - T_{w,2})}{(T_{a,2} - T_{w,in})}} \qquad (12)$$

where k_i is the heat transfer coefficient in tank i and A_i is the coil heat transfer area in tank i.

Parameter values and steady-state operating conditions are listed in Table 2.

Table 2. Model parameters

$M_{T,1} = 4351$ kg	$A_1 = 6.4$ m^2
$M_{T,2} = 4351$ kg	$A_2 = 8.9$ m^2
$C_{p,a} = 1.506$ kJ/kg K	$T_{a,in} = 447$ K
$C_{p,w} = 4.183$ kJ/kg K	$\dot{M}_a = 1.26$ kg/s
$k_1 = 1.136$ kJ/m^2 s K	$T_{w,in} = 293$ K
$k_2 = 0.738$ kJ/m^2 s K	$\dot{M}_w = 0.97$ kg/s

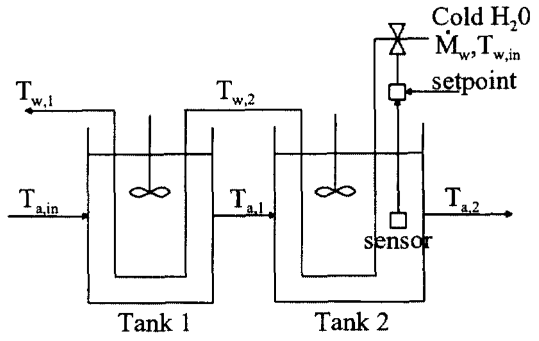

Figure 3. Two-tank cooling process

As sulfuric acid is assumed to come from an upstream unit, the feed temperature $T_{a,in}$ varies and is considered as a non-measured disturbance.

4.2 NN modeling

The 40 weights and biases of a neural state space model with n = 2, m = 1, *l* = 1, $n_{hx} = 5$, $n_{hy} = 2$ are estimated by minimizing an OLS criterion (8). The training set consists of 2270 data produced by applying steps of various amplitudes and durations in the cooling water stream. During these experiments, the inlet acid temperature is constant (Table 2). The evolution of the criterion for ten independent runs starting with random initial estimates (Fig. 4) illustrates the good performance of the adaptive 1SP-GS algorithm. Figure 5 shows some cross-validation results demonstrating the good model agreement.

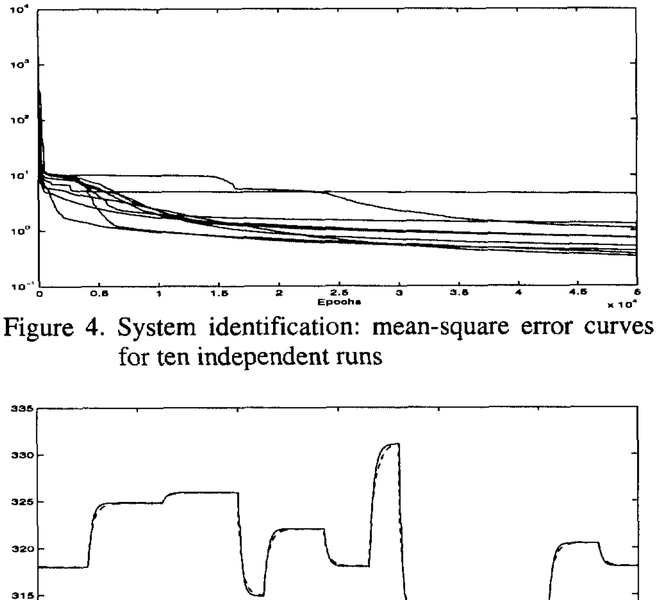

Figure 4. System identification: mean-square error curves for ten independent runs

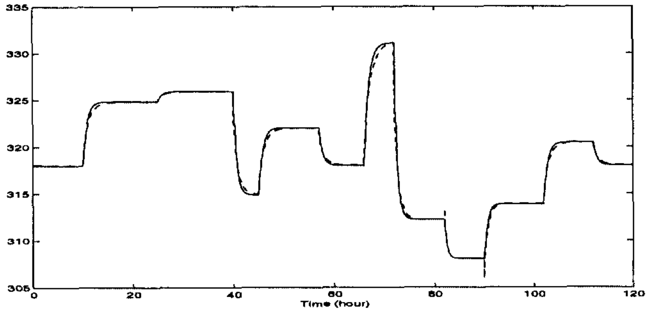

Figure 5. Cross-validation results (solid line: process output; dotted line: NN output)

4.3 NN predictive control

A NN PID controller with $n_{hl} = 2$ is used to control the acid temperature from the second tank $T_{a,2}(t)$ by acting on the cooling water stream $\dot{M}_w(t)$. The 8 unknown NN weights are estimated by minimizing the performance criterion (9) with $N_1 = 1$, $N_2 = 20$ and the output $y^r(k)$ of a second-order model reference with $\xi = 0.8$ and $\omega_n = 4$.

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 100 iterations per sampling interval. Figure 6 illustrates the excellent tracking capabilities of the NN PID controller.

However, in the face of disturbances in the acid feed temperature, the prediction of the NN process emulator deteriorates, which results in large tracking errors. Figure 7 shows the effect of step disturbances of –4 K in t = 15 hrs and +8K in t = 42 hrs. The effect of these non measured disturbances can be compensated using (10), as illustrated in Fig. 8. The effect of measurement noise with a standard deviation of 0.75 K is also depicted in Fig. 9. In all these cases, the predictive NN PID control scheme displays very satisfactory performance.

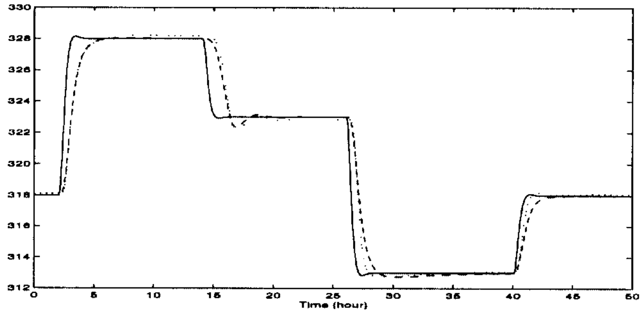

Figure 6. Tracking capabilities of the NN PID (solid line: model reference; dotted line: NN process emulator; dashed line: process output)

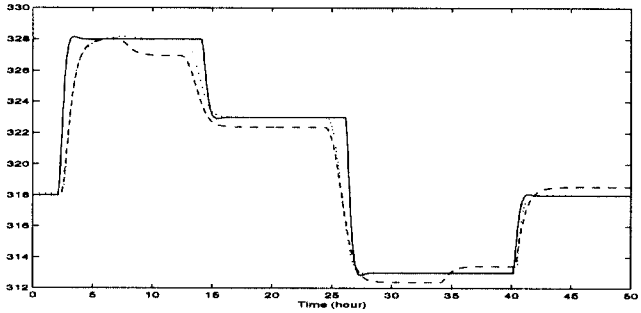

Figure 7. Effect of non measurable disturbances in the acid feed temperature (key as in Fig. 6).

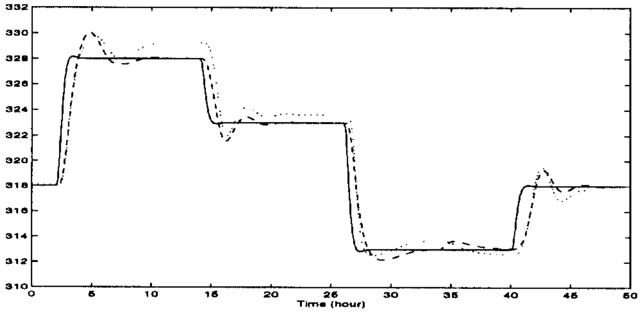

Figure 8. Modeling error compensation (key as in Fig. 6).

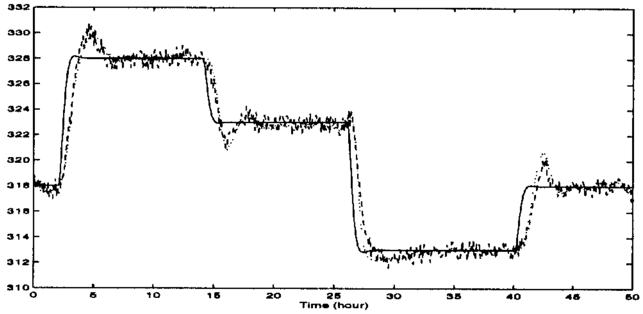

Figure 9. Effect of measurement noise (key as in Fig. 6).

5. Conclusion

The SP-approach devised by Spall [5-6] is a very powerful technique, which allows an approximation of the gradient of the objective function to be computed by effecting simultaneous random perturbations in all the parameters. In this study, a variation of the first-order SP algorithm is described and evaluated with a realistic numerical application. Especially, a predictive NN PID control scheme is developed, which shows promising (simulation) results.

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

[3] Jenson V.G., Jeffreys G.V., *Mathematical Methods in Chemical Engineering*, Academic Press, (1977).

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

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

[6] Spall J.C., "Adaptive Stochastic Approximation by the Simultaneous Perturbation Method", *IEEE Trans. Automat. Contr.* **45**, (in Press, 2000), (in condensed form in *Proc. IEEE CDC*, 3872-3879, (1998)).

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

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

[9] Vande Wouwer A., Renotte C., Remy M., "On the Use of Simultaneous Perturbation Stochastic Approximation for Neural Network Training", *Proc. ACC*, 388-392, (1999).