Experimental Study of Direct Adaptive SPSA Control System With Diagonal Recurrent Neural Network Controller

Xiao D. Ji
Dept. of Electrical Engineering
The University of Memphis
Memphis, TN 38152

Babajide. O. Familoni
Dept. of Electrical Engineering
The University of Memphis
Memphis, TN 38152

Abstract- A direct adaptive simultaneous perturbation stochastic approximation (SPSA) control system with a diagonal recurrent neural network (DRNN) as controller was examined by simulation. Different hidden number DRNN's were used in SPSA system to study the relationship between the performance and neural network architecture and parameters. Results were compared with those of SPSA using forward neural network (FNN) controller. Study shows that direct adaptive SPSA control system with DRNN has simpler architecture, smaller size of parameter vector and faster convergence rate. The system has steady-state error and is sensitive to SPSA coefficients and termination condition. For real-time trajectory control purpose, further improvement of direct adaptive SPSA approach is required.

INTRODUCTION

Nonlinear adaptive control system design is a challenge in control theory. When one deals with an unknown and/or uncertain nonlinear system, in general, one may use neural networks to identify and/or control the system. To perform neural network control, one needs to train (in general, off-line) an inverse neural network (INN) as controller. It is difficult to train the INN since the system is unknown. An ideal scheme is direct adaptive neural network control system. Spall[5] explored a neural network based simultaneous perturbation stochastic approximation (SPSA) approach to estimate the gradient of the performance function of an unknown nonlinear system. Direct adaptive SPSA approach does not require any prior knowledge of the unknown system and does not need a separated training phase. SPSA direct adaptive control system will converge to an optimal neural network parameter set, if it exists. Original SPSA approach use a forward neural network (FNN) as controller. The parameter vector size is generally large, for example, an $\aleph^4_{2,20,10,1}$ network has 280 elements in the parameter vector. Its increased computational cost results in a slow performance measure period (i.e., sampling period), which is very important for a real-time control application.

As is wellknown, recurrent neural network (RNN) has some advantages over FNN such as faster convergence, more accurate mapping ability etc., but it is difficult to apply the gradient-descent method to update the neural network weights in RNN. Ku et all [2,3] proposed the DRNN schema that captures the dynamic behavior of a system and since it is not full connected, training is expected to be much faster than RNN. DRNN with time delay has RNN behavior but simple connections and it is easy to apply gradient-descent method. Therefore, in this experimental study, a diagonal recurrent neural network (DRNN) is first time used in SPSA control system. Several different number of hidden nodes cases were simulated to examine SPSA approach. The results were compared with those of FNN SPSA scheme.

BACKGROUND

Consider the problem of finding a root θ^* of the gradient equation

$$g(\theta) \equiv \frac{\partial L(\theta)}{\partial \theta} = 0 \tag{1}$$

for some differentiable loss function (or call it energy function, cost function) $L: R^p \to R^1$. There are many methods for finding θ^*. In the case where L is observed in the presence of noise, a stochastic approximation algorithm of the generic Kiefer-Wolfowitz/Blum type is appropriate. It is based on finite difference methods which require 2p (noisy) measurements of L at each iteration. The estimated $\hat{\theta}$ at the $(k+1)^{th}$ iteration is

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

where the gain sequence { a_k } satisfies certain conditions and $\hat{g}_k(.)$ is the estimated gradient at the k[th] iteration.

In SPSA method, the $\hat{g}_k(.)$ is estimated by the "simultaneous perturbation " method: let $\Delta_k \in R^p$ be a vector of p mutually independent mean-zero random variables { $\Delta_{k1}, \Delta_{k2}, \ldots \Delta_{kp}$ } satisfying certain conditions [6], furthermore, let { Δ_k } be a mutually independent sequence with Δ_k independent of $\hat{\theta}_0, \hat{\theta}_1, \ldots \hat{\theta}_k$. We make two measurements

$$y_k^{(+)} = L(\hat{\theta}_k + c_k \Delta_k) + \varepsilon_k^{(+)} \quad (3.a)$$

$$y_k^{(-)} = L(\hat{\theta}_k - c_k \Delta_k) + \varepsilon_k^{(-)} \quad (3.b)$$

where $\varepsilon_k^{(+)}$ and $\varepsilon_k^{(-)}$ represent measurement noise terms that satisfy

$$E(\varepsilon_k^{(+)} - \varepsilon_k^{(-)} | f_k, \Delta_k) = 0 \quad \text{a.s.} \quad \forall k$$

$$f_k \equiv \{\hat{\theta}_0, \hat{\theta}_1, \ldots \hat{\theta}_k\} \quad (4)$$

Then the estimation of g(.) at k[th] iteration is

$$\hat{g}(\hat{\theta}_k) = \begin{bmatrix} \dfrac{y_k^{(+)} - y_k^{(-)}}{2c_k \Delta_{k1}} \\ :::: \\ \dfrac{y_k^{(+)} - y_k^{(-)}}{2c_k \Delta_{kp}} \end{bmatrix} \quad (5)$$

The name "simultaneous perturbation" as applied to this method arises from the fact that all elements of the $\hat{\theta}_k$ vector are being varied simultaneously. If there is little change in several successive iterations, the algorithm terminated and the last iteration $\hat{\theta}_k$ is the optimum θ*. Note that this estimate needs only two measurements instead of 2p in the usual finite difference approximation.

Spall has shown that under certain conditions the bias in $\hat{g}_k(.)$ as estimation of g(.) goes to 0 as $k \rightarrow \infty$ and $\hat{\theta}_k$ converges almost surely (a.s.) to θ*[4,5,6]. The basic steps for implementing the SPSA are in [9].

EMPIRICAL STUDIES

Spall[4,5] proposed two kind adaptive control system with SPSA algorithm: direct adaptive control (DA) and self-turning adaptive control (STA). When virtually nothing is known about the plant the DA approach is appropriate but STA requires that some prior information exists about the plant.

Figure 1 shows the block diagrams of DA scheme which was employed in our study. An unknown nonlinear plant,

$$y(k+1) = 0.8 \times \sin(2y(k)) + 1.2 \times u(k),$$

was used for our study. The desired plant output is sinusoidal. FNN and DRNN neural network controllers were employed for comparison purposes. In both cases, tanh(.) function was employed as sigmoidal function and each iteration involved three measurements (samplings).

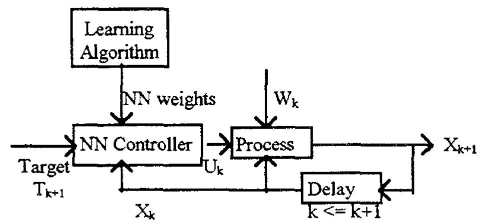

Figure 1. Block diagram for DA control system

In the FNN case, the neural network controller (NNC) is $\aleph^4_{2,20,10,1}$ and the size of the parameter vector $\hat{\theta}$ is p = 280. Based on the FNN algorithm, it needs 250 multiplications, 280 additions and 30 sigmoidal function calculations for each measurement.

Simulation results show that it takes longer to simulate each measurement and the plant output convergence is slower. Figure 2 shows a simulation result for first 2000 measurements. The termination condition is $\| \Delta\theta \| < 0.01$ and the successive iteration number is 100.

In DRNN case, the NNC is $\aleph^3_{2,5,1}$, the size of the parameter vector $\hat{\theta}$ is p=25 with biases. It needs 20 multiplications, 20 additions and 5 sigmoidal function calculations for each measurement. Figure 3 shows a simulation result of the DRNN case with the same termination condition as the FNN case. At 1500 measurements the system satisfied the termination condition and the SPSA was terminated. Figure 4 and Figure 5 show final results of FNN and DRNN case after the SPSA algorithm, respectively. Note that in both cases there are steady state errors and it seems DRNN has a larger identification error.

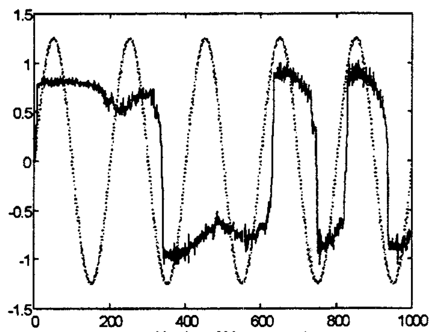

Figure 2. System output during SPSA procedure with a $\aleph^4_{2,20,10,1}$ FNN as NNC

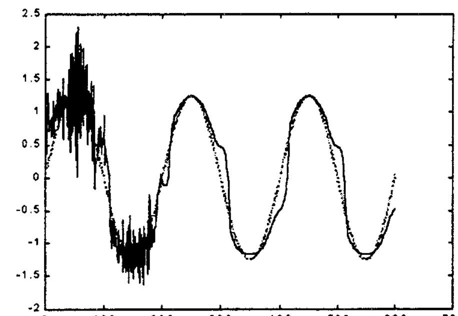

Figure 4. System output after SPSA procedure with a $\aleph^4_{2,20,10,1}$ FNN as NNC

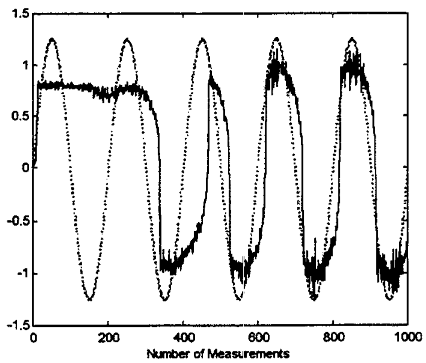

Figure 3. System output during SPSA procedure with a $\aleph^3_{2,5,1}$ DRNN as NNC

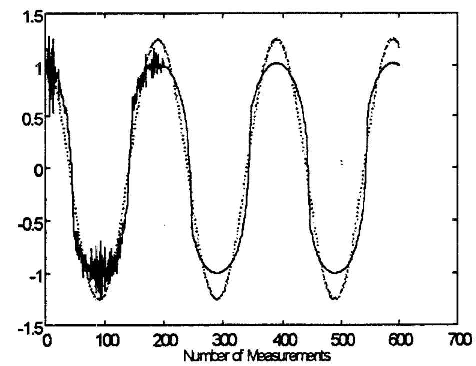

Fig. 5. System output after SPSA procedure with a $\aleph^3_{2,5,1}$ DRNN as NNC

DISCUSSION

Different configurations of DRNN ($\aleph^3_{2,7,1}$, $\aleph^3_{2,9,1}$) were investigated. The results were not significantly different, therefore, the simplest one was chosen.

Since FNN and DRNN have different net architecture with different parameters, it is difficult to compare the convergence speed directly, but considering the computational cost for each measurement, one can see that DRNN converges faster than FNN.

The choice of the gain sequences $a_k = a(k+1)^{-\alpha}$ and $c_k = c(k+1)^{-\gamma}$ are critical to the performance of SPSA. Spall[29] pointed out that α 0.602 and γ 0.101 are practically effective values and **a** and **c** should be determined by experiments. In general, starting at a value of one and decreasing or increasing this value if, respectively, the algorithm seems to be behaving erratically or too conservatively. As a rule-of-thumb, with the Bernoulli ± 1 distribution for the elements of Δ_k, it is found that **c** can be set at a level approximately equal to the standard deviation of the measurement noise in $y(\theta)$. Our study proved these conclusions are true in DRNN case.

$\hat{\theta}_0$ should be close to the optimal θ^* to ensure $\hat{\theta}_k$ does not get stuck in a local minimum of L_k [8,9]. Chin [1] discussed a technique by which SPSA can be used as a global optimizer for arbitrary initial conditions. An appropriate $\hat{\theta}_0$ will lead to better and faster results.

When the network parameters change by a given tolerance for *several consecutive iterations*, one can terminate the SPSA algorithm and use the fixed neural network as controller. Our study shows that the termination condition is critical to the final performance of the system from the control engineering point of view. In general, after SPSA the system has steady-state error and it is termination condition dependent. One needs to determine what is the optimal number of iterations for terminating the SPSA.

During SPSA procedure, the system is in direct adaptive control but not after SPSA. When system parameters or structure changes (the common case in adaptive control problem) or the desired output changes, the SPSA should be turned on again to perform adaptive control function. In practical application, one must determine the condition to turn on or off the SPSA algorithm if necessary.

Our simulation shows that the neural network based SPSA algorithm is good for a smooth trajectory signal but not for rapidly changing trajectory signal (e.g., a random signal) or a constant signal. It would limit its application in some particular cases.

CONCLUSION

SPSA approach has good direct adaptive control scheme in the sense of statistical modeling and control. Therefore, it can be applied to unknown nonlinear systems in the presence of noise. The system is stable and converges to an optimal state under certain conditions of SPSA coefficients. Our study showed that for trajectory control problem, in general, SPSA has some steady state error, and it is parameter (α, γ, a, and c) sensitive. A good performance after SPSA also depends on the termination condition and the desired trajectory signal.

These implies that a good performance SPSA design is case dependent and requires trial and error method to find the optimal performance coefficients. To improve direct adaptive SPSA control system for real-time trajectory control purpose is our next study topic.

REFERENCE

[1] D. C, Chin, "A more efficient global optimization algorithm based on Styblinski and Tang." *Neural Networks*, pp. 573-574, 1994.

[2] C.-C. Ku and K.Y. Lee, "Diagonal Recurrent Neural Network-Based Control: Convergence and Stability." *Proc. of ACC,* pp 3340 - 3345, Baltimore, June, 1994.

[3] C.-C. Ku and K.Y. Lee, "Diagonal recurrent neural networks for dynamic systems control." *IEEE Trans. Neural Networks. Vol. 6, No. 1,* pp 144-155, Jan. 1995.

[4] J. Spall and J. Cristion, "Nonlinear Adaptive Control Using Neural Networks: Estimation with a Smoothed Form of Simultaneous Perturbation Gradient Approximation." *Proc. of ACC,* pp 2560 - 2564, Baltimore, June, 1994.

[5] J. Spall, and J. Cristion, "Nonlinear adaptive control using neural networks: estimation with a smoothed form of simultaneous perturbation gradient approximation." *Statistica Sinica Vol. 4.* pp. 1-27, 1994.

[6] J. Spall, "Multivariate stochastic approxima-tion using a simultaneous perturbation gradient approximation. *IEEE trans. Auto. Contr. Vol. 37, No. 3,* pp. 332- 341, March, 1992.

[7] J. Spall and D. C. Chin, "A model-free approach to optimal signal light timing for system-wide traffic control." *Proce. of IEEE CDC,* pp. 1868-1875. Lake Buena Vista, 1994.

[8] J. Spall, "Developments in stochastic optimiz - ation algorithms with gradient approximations based on function measurements. *Proce. of the 1994 winter simulation conference*. pp. 207-213.

[9] J. Spall, "Implementation of the simultaneous perturbation algorithm for stochastic optimization. *submitted to American Statistician, Sept. 1995.*