Controlling Chaos with Artificial Neural Network: Numerical Studies and Experiments

István Z. Kiss and Vilmos Gáspár*

Institute of Physical Chemistry, University of Debrecen, 4010 Debrecen, P. O. Box 7, Hungary

Received: April 6, 2000; In Final Form: June 16, 2000

Although there are a number of theoretically suggested chaos control methods using artificial neural networks (ANN), experimental tests are still lacking. In this paper, we report on experimental chaos control during the electrochemical dissolution of copper in phosphoric acid. The neural network implementation of simple proportional and recursive feedback algorithms is presented.

Introduction

Chaotic systems are highly susceptible to control by using small perturbations to a system constraint. Feedback methods have been applied to taming chaos in magnetoelastic and hydrodynamic systems, electric circuits, lasers, chemical reactions, and tissues of heart and brain in vitro.[1] The well-known Ott−Grebogi−Yorke (OGY) algorithm[2] and its variations are based on the notion that for controlling chaos one needs to know only the local dynamics in a small (linear) range around a fixed point (corresponding to the targeted unstable periodic orbit, UPO), on some *n*-dimensional surface-of-section (Poincaré section) of the phase space trajectories. The OGY method has been further developed by Petrov et al.[3] and Rhode et al.[4] They studied the dynamics of the system around the fixed point under the effect of random perturbations, then the goal dynamics has been targeted by using a control rule with empirically determined constants in a control formula.

In experimental settings, however, application of these methods is often troublesome because of noise and shift in the system constraints.[5] It seemed inevitable to develop a better technique so as to make chaos control a routine (automated) procedure.

Artificial neural networks are widely used in chemometrics, especially when the evaluation of experimental data requires complex, nonlinear fitting. A comprehensive review on the topic has been published by Sumpter et al.[6] It has been shown first by Alsing et al.[7] that chaos control can be implemented by using a nonlinear fitting procedure with an ANN. In their numerical work, however, they just simply fitted the data to the OGY formula by an ANN and applied the trained network to control chaos. A strategy developed later by Lebender et al.[8] takes advantage of the nonlinearity built in an ANN, and it also works outside the linear region of the fixed point. However, training of the ANN required an additional numerical fitting procedure. Konishi et al.[9] have also developed an on-line chaos controller that, however, works only for fixed point with eigenvalues in a given range or it requires a complicated procedure to construct an error function for the ANN.[10] The method of Bakker et al.[11] fits a global model to discrete time series data and controls chaos using the trained network. This method has been experimentally applied to control the chaotic motion of a pendulum.

These methods have not been tested on real chemical systems so far. Our goal was to develop ANN algorithms as simple as possible and test their effectiveness in experiments.

This paper is structured as follows. First, a brief introduction to artificial neural networks is presented. Then we devise a simple strategy for controlling chaos by taking advantage of a well-known feature of artificial neural networks: they can "learn" the linear or nonlinear rules embedded even in a noisy data set. The proposed method is first tested for taming chemical chaos in a simple three-variable model, the chaotic Autocatalator.[12] We show that after learning the map-based representation of the chaotic dynamics at a given range of an accessible control parameter, the trained network can be readily applied to controlling chaos based on the simple proportional feedback (SPF) algorithm by Peng et al.[13] The network is then modified so as to implement the so-called recursive proportional feedback (RPF) method for controlling chaos.[14] The improved ANN algorithm is applied to a model for the respiratory behavior of a diffusively coupled two-cell system.[15,16] Finally, we test the suggested method on an experimental system, the chaotic electrodissolution of copper in concentrated phosphoric acid electrolyte.[17−21]

Artificial Neural Networks

From a practical point of view, an artificial neural network is simply a computer program that transforms an *m*-variable input into an *n*-variable output. The units of the network are the so-called neurons that are connected, for example, to a feed-forward network, in which the information is processed through several layers such as input layer, hidden layer(s), and output layer.[22] A neuron collects inputs from all of the units of the upper layer and transforms its net input into an output. The total input I_j to unit j is the weighted sum of the output of all neurons ($i = 1, 2, ..., n$) connected to it:

$$I_j = \theta_j + \sum_i w_{ji} o_i \tag{1}$$

where o_i is the output of the ith unit, w_{ji} is a weight characterizing the "strength" of connection between units j and i, and θ_j is a threshold value. The total input I_j is transformed into an output o_j by a transfer function:

$$o_j = \{1 + \exp(-I_j)\}^{-1} \tag{2}$$

* To whom correspondence should be addressed.

10.1021/jp001316w CCC: $19.00
Published on Web 08/04/2000

The network is trained by iteratively correcting the weights w_{ji} so as to produce the previously specified output values (the target sets) for as many input sets as possible. The error of a network for a given training set is calculated by the following equation:

$$E = \frac{1}{2}\sum_j (o_j^{\text{out}} - t_j)^2 \tag{3}$$

where o_j^{out} and t_j are, respectively, the actual and desired (target) output values of neurons in the output layer. During the training session the weights are updated according to a "learning rule". In our calculations we apply the Langevin-type error back propagation using dynamic learning parameters:

$$\Delta w_{ji}^{n+1} = -\eta_n \frac{\partial E}{\partial w_{ji}} + \alpha \Delta w_{ji}^n + \beta_n \sigma(1) \tag{4}$$

where Δw_{ji}^n is the change in the w_{ji} value at the nth iteration, η_n is the learning rate, $0 < \alpha < 1$ is a constant called momentum, β_n is a training parameter, and $\sigma(1)$ is a Gaussian-noise with zero expected value and unit standard deviation. The derivatives $\partial E/\partial w_{ji}$ are calculated by the back-propagation algorithm. After each input set has been passed through the network (after one epoch) η_n and β_n are updated according to eq 5

$$\eta_{n+1} = \begin{Bmatrix} 0.999\eta_n & \text{if } E_{n+1} \geq E_n \\ \eta_n\left(1 + \dfrac{1-0.9999}{10}\right) & \text{otherwise} \end{Bmatrix}$$
$$\beta_{n+1} = 0.999\beta_n \tag{5}$$

The initial values of β_n and η_n were set to 2.0 and 0.1, respectively, while $\alpha = 0.5$. After the weights are adjusted, the error is recalculated by doing a forward pass through the network again. This procedure is repeated until the error of the network is less than an acceptable value or the number of epochs exceeds a preset limit. Details of the learning algorithm can be found in the manual of JETNET program package[23] and references cited therein.

Controlling Chaos with ANN

Numerical Studies. The steps of the envisioned simplest strategy for controlling low-dimensional chaotic systems with an artificial neural network are as follows: (1) Collect a small set of next-return values $X_n(p)$ and $X_{n+1}(p)$ of a monitored variable to a previously defined Poincaré section of the phase−space trajectories at different values of an accessible parameter p. (2) By using the collected values of next iterates at each value of p (the input set), train an ANN to compute the corresponding values of control parameter p (the output set). (3) To control chaos, apply the trained network to calculate p_{n+1} for the next period (the output value) by using $X_n(p_n)$ and $X_F(p_0)$ as input values, where p_n and p_0 are values of the control parameter, respectively, during the cycle and before control is turned on.

If the Poincaré section of the attractor moves in the phase space when the control parameter p is changed, the OGY algorithm must be modified. Dressler and Nitsche[24] showed that the correction δp_n depends not only on the deviation of the system from the fixed point but also on the correction δp_{n-1} during the previous cycle. The result is a recursive proportional feedback (RPF) algorithm. As shown later, the simple strategy above can be easily modified for the implementation of the RPF method.

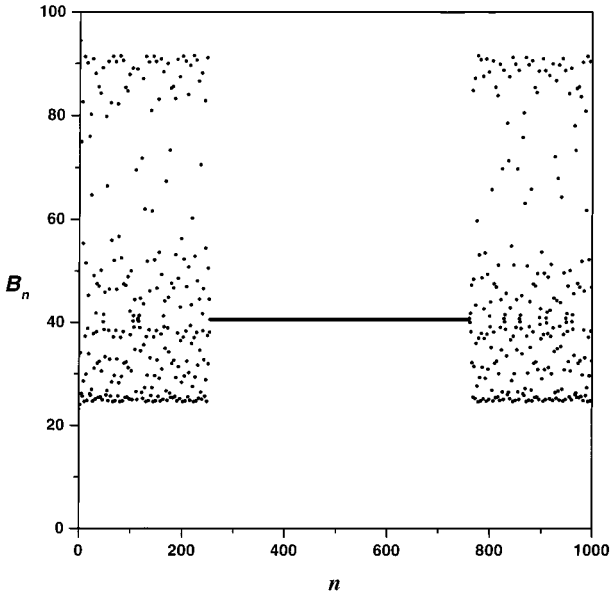

Figure 1. Chaos in the three-variable Autocatalator at $\mu = 0.154$ is controlled by using a 2−6−6−1 feed forward neural network. Control ($B_F = 40.568$) is switched on at $n = 250$ and turned off at $n = 750$.

Simple Proportional Feedback by ANN Algorithm. In Figure 1, a previously trained feed-forward ANN is applied to controlling chaos in the three-variable Autocatalator[12] that is a prototype model for chaotic behavior observed in isothermal chemical systems. Rate equations of the model are

$$\frac{da}{dt} = \mu(\kappa + c) - a - ab^2$$
$$\sigma\frac{db}{dt} = a + ab^2 - b$$
$$\delta\frac{dc}{dt} = b - c \tag{6}$$

where a, b, and c are dimensionless variables, τ is dimensionless time, and $\mu = 0.154$, $\kappa = 65$, $\sigma = 5 \times 10^{-3}$, and $\delta = 2 \times 10^{-2}$ are parameters. B_n is the value of the monitored variable b at the nth return to a Poincaré section ($c = 15$, $dc/dt > 0$).

The differential equations were solved at 32 different values of control parameter μ ranging from 0.1535 to 0.15505 ($\Delta\mu = 5 \times 10^{-5}$). At a given μ, the values of B_n and B_{n+1} were monitored after a transient period $t = 10$. A particular pair of data has been recorded if the value of B_n was within a predefined range ($36 < B_n < 43$). The full set of input data was compiled by using the first 26 pairs of such B_n values at each μ, while the targeted set of outputs contained the corresponding μ values. The appropriate size of the network has been determined by using the standard procedures of neural computing. In this case we have found the best performance by using a 2−6−6−1 feed forward network.

Control is achieved (Figure 1) by changing the value of μ such that the fixed point $B_F = 40.568$ (corresponding to the unstable periodic orbit) is targeted on each return. The value of μ for the next period is calculated by the trained network such that the inputs are B_n and B_F, while the output is the sought value of μ_{n+1}.

Recursive Proportional Feedback by ANN Algorithm. The chaotic respiratory behavior of a two-cell *Klebsiella aerogenes* bacterial culture can be described with a model suggested by

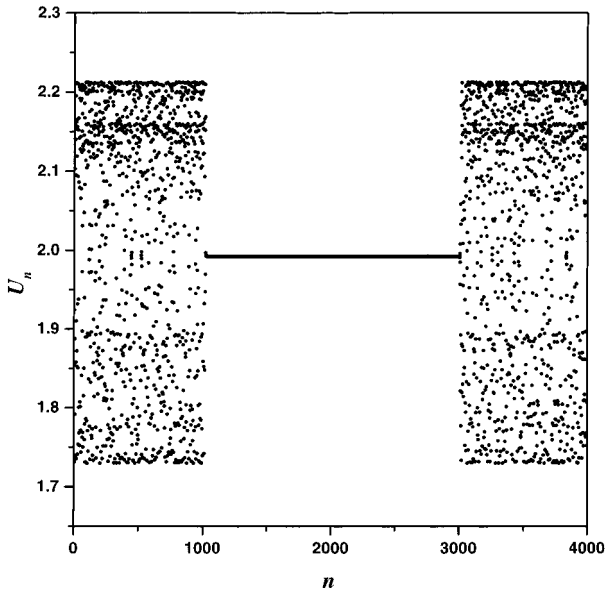

Figure 2. Chaos in the model for the respiratory behavior of a two-cell system at $a_1 = 8.95$ is controlled by using a 3−8−8−1 feed forward network. Control ($U_F = 1.9921$) is switched on at $n = 1000$ and turned off at $n = 3000$.

Degn and Harrison:[15,16]

$$\frac{du_1}{dt} = b_1 - u_1 - \frac{u_1 v_1}{1 + q_1 u_1^2} + D_u c(u_2 - u_1)$$

$$\frac{dv_1}{dt} = a_1 - \frac{u_1 v_1}{1 + q_1 u_1^2} + D_v c(v_2 - v_1)$$

$$\frac{du_2}{dt} = b_2 - u_2 - \frac{u_2 v_2}{1 + q_2 u_2^2} - D_u(u_2 - u_1)$$

$$\frac{dv_2}{dt} = a_2 - \frac{u_2 v_2}{1 + q_2 u_2^2} - D_v(v_2 - v_1) \qquad (7)$$

where u_1, v_1, u_2, and v_2 are variables, t is time, and $a_2 = 8.9$, $b_1 = b_2 = 11$, $q_1 = q_2 = 0.5$, $c = 4 \times 10^3$, $D_u = 1 \times 10^{-5}$, and $D_v = 1 \times 10^{-3}$ are parameters. The control parameter is a_1, while U_n is the value of the monitored variable u_2 at the nth return to a Poincaré section ($v_2 = 13.45$, ($dv_2/dt > 0$).

The differential equations were solved by randomly varying the value of control parameter: $a_1 = 8.9 + 2 \times 10^{-3}$random(50) where random(50) denotes a random integer between 0 and 50. The values of U_{n-1}, U_n, and $a_{1,n-1}$ were monitored. The data were recorded if the value of U_n was within a predefined range $|U_n - U_F| < 0.01$ around the targeted fixed point ($U_F = 1.9921$). The full set of input data was compiled by using 500 recorded values, while the targeted set of outputs contained the corresponding $a_{1,n}$ values. The best performance has been found by using a 3−8−8−1 feed forward network.

Control is achieved (Figure 2) by changing the value of a_1 such that the fixed point $U_F = 1.9921$ is targeted on each return. The value of $a_{1,n+1}$ for the next period is calculated by the trained network such that the inputs are U_n, U_F, and $a_{1,n}$, while the output is the sought value of $a_{1,n+1}$ for the next cycle.

Experiments. Details of the experimental setup and procedures have been published in an earlier publication.[21] A standard three-electrode electrochemical cell was used. The cell contained 70 cm^3 orthophosphoric acid (85%, Merck or Spektrum-3D) thermostated at -17.5 ± 0.1 °C and was connected to a computer controlled potentiostat (Electroflex EF451). The potential between a rotating copper disk electrode of 5 mm diameter (anode) and a saturated calomel electrode (SCE) was measured and set with a resolution of 0.2 and 0.01 mV, respectively, for simple potentiostatic and chaos control experiments. The output current signal between the rotating disk electrode (RDE) and a platinum sheet counter electrode of 5 cm^2 area (cathode) was fed into a 12-bit A/D converter built into the potentiostat. Sampling frequencies of 100 and 200 Hz were applied, respectively, for data acquisition and control experiments. A period-doubling route to chaos can be observed at about 1750 rpm if 130 Ω external serial resistance is connected to the system such that the total ohmic resistance of the cell is between 200 and 204 Ω. Chaotic current oscillations can be observed under potentiostatic conditions at −17.5 °C near 1750 rpm and 510 mV.

We have shown in our earlier work[21] that the dynamics of the system on the Poincaré section can be represented by a one-dimensional next-return map f. In our case, a reconstructed phase space can be obtained using the time-delay method with an embedding dimension of two and 0.5 s delay. The next return map I_{n+1} vs I_n has been generated by using successive current values on the Poincaré section defined as $I_t = I(t - 0.5\text{ s})$ and $dI(t)/dt > 0$), where $I(t)$ is the measured current at time t. The iterates on the next-return map are given by $I_{n+1} = f(I_n,V)$, where I_n and I_{n+1} are next-return values of the monitored variable (current) at a given value of an accessible parameter V (voltage). At $V = V_0$ the map f generates chaotic dynamics around the fixed point $I_F(V_0) = f(I_F(V_0),V_0)$.

For this experimental system, the simplest ANN algorithm for controlling chaos was applied as follows. (i) We change V randomly around V_0 at every step $V_n = V_0 + \Delta V(-1 + 2$ ran(1)), where ran(1) denotes a random number between 0 and 1 and ΔV is the maximum allowed perturbation (0.15 mV). During this step we collected the data triplets $\{I_n, I_{n+1}, \delta V_n\}$ if the I_n and I_{n+1} values were near the fixed point $I_F(V_0)$. The approximate value of I_F (0.900 ± 0.001 mA) was determined from the one-dimensional map using a linear least-squares fit around the fixed point. We collected the data triplets $\{I_n, I_{n+1}, \delta V_n\}$ if $|I_n - I_F| < \epsilon$ or $|I_{n+1} - I_F| < \epsilon$, and $|I_{n+1} - I_n| < 2\epsilon$ were found to be valid with ϵ set to 0.015 mA. (ii) When enough data triplets (31 in this case) was collected, we constructed a 2−6−1 ANN with two inputs (I_n, I_{n+1}) and one output (δV_n). The network was trained (3500 epoch) to calculate the parameter value at which the phase point in the map is moved from I_n to I_{n+1}. (iii) To control chaos, we applied the trained network to calculate δV_n for the next period by using $I_n(V_{n-1})$ and $I_F(V_0)$ as inputs. The potential perturbation calculated by the network was applied when $|I_n - I_F| < 2\epsilon$.

We note that for simple technical reasons, namely, the design of our computer-controlled potentiostat, we trained the network to calculate δV_n values rather than $V_n = V_0 + \delta V_n$ values. In Figure 3, time series current data (left axis) and the applied potential perturbations (right axis) are plotted. The control algorithm was turned on at 56.2 s. However, the first potential perturbation was applied only at 67.2 s when $|I_n - I_F| < 2\epsilon$. The period-one orbit was stabilized until the control was turned off at 115.9 s, then the behavior became chaotic again.

The average potential perturbation is greater than zero (0.09 mV) which shows that at the applied V_0 the controlled fixed point ($I_{F,ANN}$) is slightly different from the desired value I_F. This deviation can be explained by some systematic error during the training session of the ANN. By changing the desired fixed point

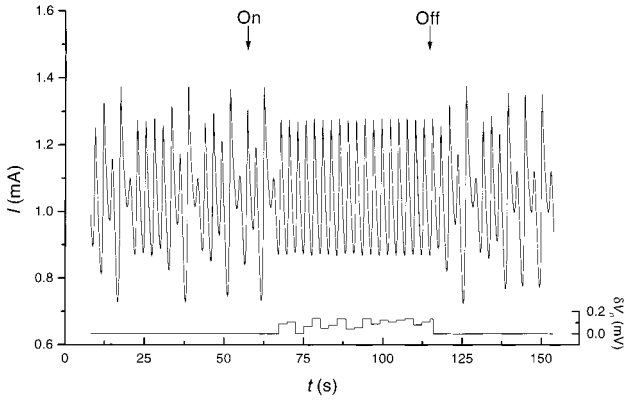

Figure 3. The ANN implementation of SPF. Current (left axis) vs time for an interval when control for stabilizing period-one has been switched on at 56.2 s and switched off at 115.9 s. The chaotic behavior was observed at 1721 rpm, 509 mV, and −17.5 °C. I_F was set to 0.9000 mA. The ANN (2−6−1) was trained using 31 $\{I_n, I_{n+1}, \delta V_n\}$ data triplets collected near the fixed point. Potential perturbations (right axis) were applied if $|I_n - I_F| < 0.030$ mA.

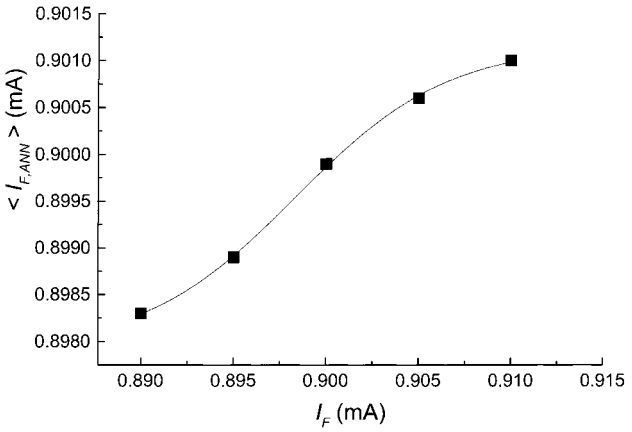

Figure 4. The average of controlled fixed points $\langle I_{F,ANN}\rangle$ vs the desired fixed points I_F. The average of controlled fixed points $\langle I_{F,ANN}\rangle$. The average was calculated using 30 controlled fixed points. The curve is a fitted sigmoidal function of [(0.8979−0.9012)/(1 + exp((*x*-0.8993)/0.0041)) + 0.9012]. Experimental conditions are given in Figure 3.

to stabilize I_F we calculated the average of the controlled fixed points (Figure 4). The sigmoidal curve fitted on the points resulted from the sigmoidal scaling functions of the network and was also observed in our numerical studies. We expected the network to control the desired fixed points in the range from 0.89 to 0.91; however, the ANN was able to control chaos only in the range of 0.898−0.901. Outside this region the ANN was not able to control chaos but held the system near the fixed points for some iterations. The small control region can be interpreted by plotting the average potential perturbations during control vs the controlled fixed points (Figure 5). The ANN controls chaos in almost half of the training region $V_0 \pm 0.5\Delta V$ but obviously does not control outside this range. Our efforts to increase the training region were unsuccessful because the ANN could not be trained with enough accuracy.

The most exciting feature of our algorithm is its ability to extend the chaos control algorithm to more dimensions and more complicated systems. We demonstrate this feature by implementing the recursive proportional feedback algorithm as well. This algorithm can be applied if I_{n+1} depends on I_n,V_n, and V_{n-1} as well. The ANN implementation of RPF requires the following steps: Step (i) is the same as before but instead of collecting $\{I_n, I_{n+1}, \delta V_n\}$ data triplets we now collect $\{I_n, I_{n+1}, \delta V_{n-1}, \delta V_n\}$ datasets. In step (ii) we construct an ANN with three inputs

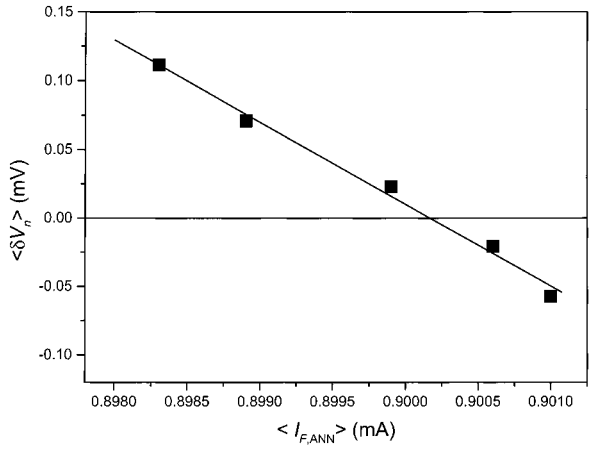

Figure 5. The average values of potential perturbations $\langle \delta V_n\rangle$ vs the average values of the controlled fixed points $\langle I_{F,ANN}\rangle$. The averages were made from 30 points. The straight line is the least-squares linear fit to the points. The intersection of this line with $\langle \delta V_n\rangle = 0$ line gives the actual value of the fixed point $I_F = 0.9002$ mA. Experimental conditions are given in Figure 3.

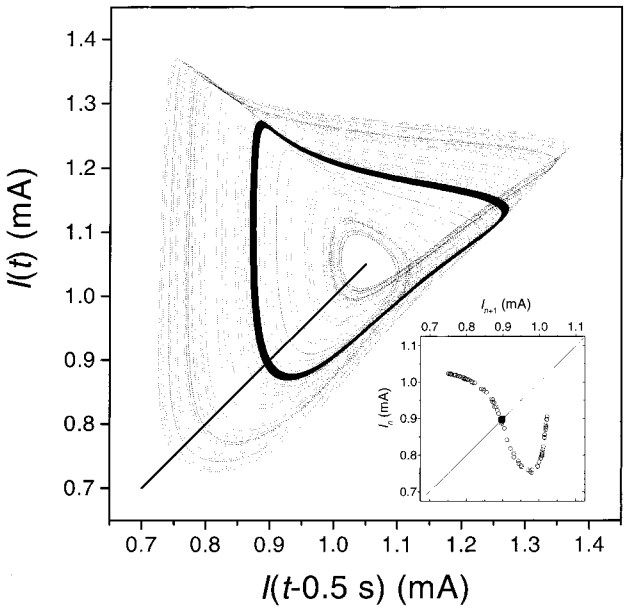

Figure 6. The ANN implementation of RPF. Stabilized period one orbit embedded in the chaotic attractor. The next-return map has been generated by using the successive $I(t - 0.5$ s) values on the Poincaré section (thick line). Superimposed on the map (inset) are the next-return values (solid squares) while control is being implemented. The applied potential in the uncontrolled system is 515.5 mV, rotation rate 1800 rpm, $I_F = 0.9040$ mA. The ANN (3−4−1) was trained using 32 $\{I_n, I_{n+1}, \delta V_{n-1}, \delta V_n\}$ data sets collected near the fixed point.

$\{I_n, I_{n+1}, \delta V_{n-1}\}$ and one output δV_n. To control chaos in step (iii), the inputs to the ANN are $\{I_n, I_F(V_0), \delta V_n\}$.

Stabilized period-one orbit embedded in the reconstructed chaotic attractor of the copper-phosphoric acid system is shown in Figure 6. The next return map has been generated by using successive current values on the Poincaré section as earlier. Superimposed on the map are the next-return values (solid squares) while control is being implemented. The data collection took approximately 15 min, while the ANN training took about 5 min, so the whole procedure could be repeated easily. This algorithm requires only the approximate location of the fixed point and was successfully applied to control the chaotic electrodissolution of copper in concentrated phosphoric acid electrolyte.

Conclusions

The ANN implementation of SPF and RPF algorithms was successfully applied to control chaos in simple models and in an electrochemical reaction as well. An ANN is trained by exploring the dynamics of the system under random perturbations (training session) and then is applied to controlling chaos (control session). Advantages of using an ANN are the following: (a) the inherent nonlinearity built into ANN can be exploited to control nonlinear processes, (b) the procedure does not require a large number of complex calculations, and (c) because of its simplicity, the strategy can be easily applied to similar problems in a wide arena of research fields ranging from physics to biology.

The algorithms presented in this paper are based on one-dimensional next-return maps and low-dimensional chaotic attractors. ANN methods have been tested to control high-dimensional chaos but only in numerical models. Application to real systems as well as extending these algorithms to track unstable periodic orbits requires extensive experimental work in the future.

Acknowledgment. This work was supported by the following Hungarian Research Grants: OTKA T017784, T025375, and FKFP 0455/1997.

References and Notes

(1) Shinbrot, T. *Adv. Phys.* **1995**, *44*, 73.
(2) Ott, E.; Grebogi, C.; Yorke, J. A. *Phys. Rev. Lett.* **1990**, *64*, 1196.
(3) Petrov, V.; Mihaliuk, E.; Scott, S. K.; Showalter, K. *Phys. Rev. E* **1995**, *51*, 3988.
(4) Rhode, M. A.; Rollins, R. W.; Dewald, H. D. *Chaos* **1997**, *7*, 653.
(5) Petrov, V.; Gáspár, V.; Masere, J.; Showalter, K. *Nature* **1993**, *361*, 240.
(6) Sumpter, B. G.; Getino, C.; Noid, D. W. *Annu. Rev. Phys. Chem.* **1994**, *45*, 439.
(7) Alsing, P. M.; Gavrielides, A.; Kovanis, V. *Phys. Rev. E* **1994**, *49*, 1225.
(8) Lebender, D.; Müller, J.; Schneider, F. W. *J. Phys. Chem.* **1995**, *99*, 4992.
(9) Konishi, K.; Kokame, H. *Phys. Lett. A* **1995**, *206*, 203.
(10) Konishi, K.; Kokame, H. *Physica D* **1996**, *100*, 423.
(11) Bakker, R.; Schouten, J. C.; Takens, F.; van den Bleck, C. M. *Phys. Rev. E* **1996**, *54*, 3545.
(12) Peng, B.; Scott, S. K.; Showalter, K. *J. Phys. Chem.* **1990**, *94*, 5423.
(13) Peng, B.; Petrov, V.; Showalter, K. *J. Phys. Chem.* **1991**, *95*, 4957.
(14) Rollins, R. W.; Parmananda, P.; Sherard, P. *Phys. Rev. E* **1993**, *47*, R780.
(15) Degn, H.; Harrison, D. E. F. *J. Theor. Biol.* **1969**, *22*, 238.
(16) Lengyel, I.; Epstein, I. R. *Chaos* **1991**, *1*, 69.
(17) Glarum, S. H.; Marshall, J. H. *J. Electrochem. Soc.* **1985**, *132*, 2872.
(18) Albahadily, F. N.; Schell, M. *J. Chem. Phys.* **1988**, *88*, 4312.
(19) Vidal, R.; West, A. C. *J. Electochem. Soc.* **1995**, *142*, 2682.
(20) Kiss, I. Z.; Gáspár, V.; Nyikos, L. *J. Phys. Chem. A* **1998**, *102*, 99.
(21) Kiss, I. Z.; Gáspár, V.; Nyikos, L.; Parmananda, P. *J. Phys. Chem. A* **1997**, *101*, 8668.
(22) Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. *Nature* **1986**, *323*, 533.
(23) Peterson, C.; Rögnvaldsson, T.; Lönnblad, L. *Comp. Phys. Commun.* **1994**, *81*, 185.
(24) Dressler, U.; Nitsche, G. *Phys. Rev. Lett.* **1992**, *68*, 1.