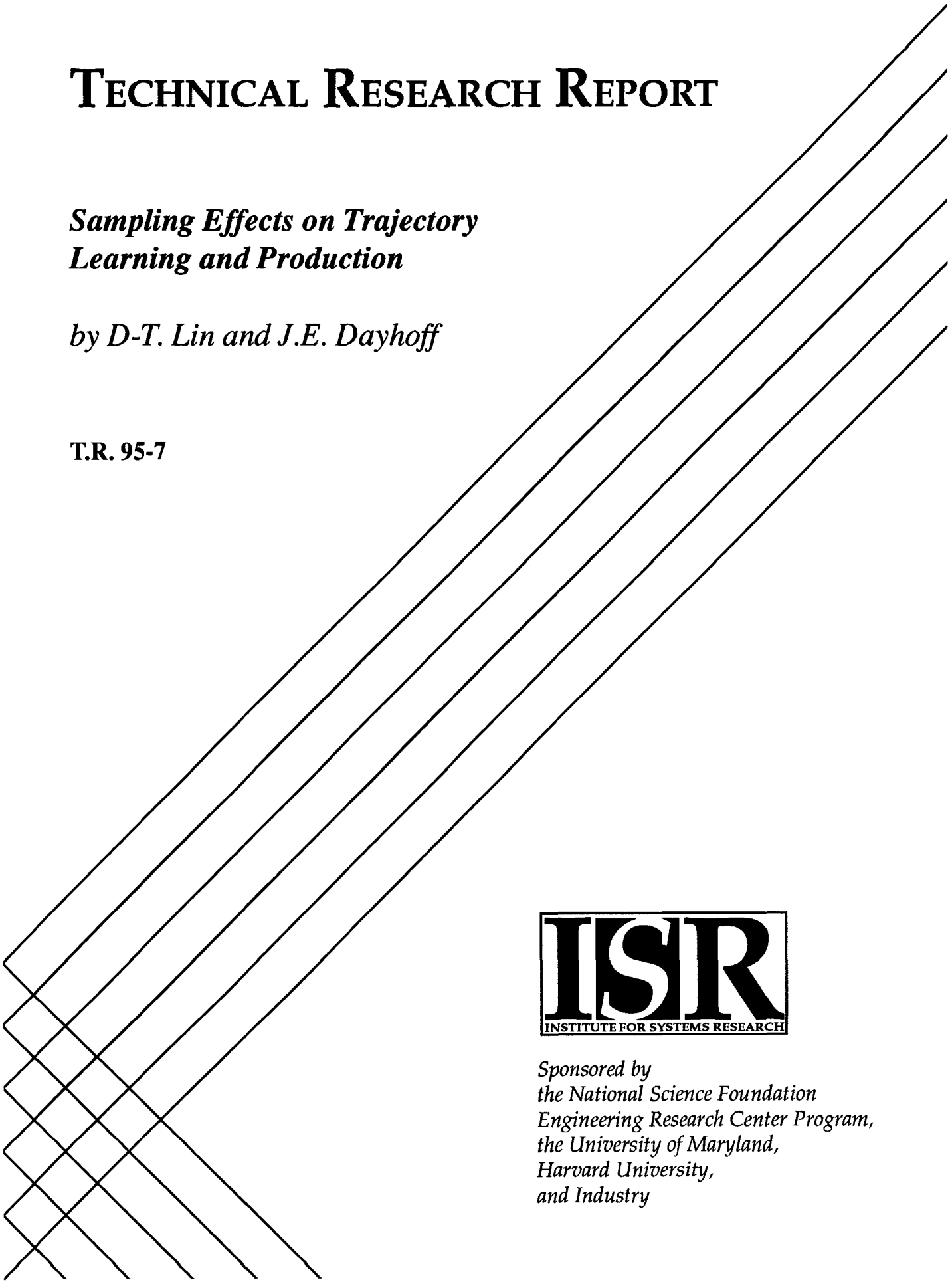
TECHNICAL RESEARCH REPORT

Sampling Effects on Trajectory Learning and Production

by D-T. Lin and J.E. Dayhoff

T.R. 95-7



Sponsored by
the National Science Foundation
Engineering Research Center Program,
the University of Maryland,
Harvard University,
and Industry

Sampling Effects on Trajectory Learning and Production*†

Daw-Tung Lin and Judith E. Dayhoff

Institute for Systems Research

University of Maryland

College Park, MD 20742

Abstract

The time-delay neural network ($TDNN$) and the adaptive time-delay neural network ($ATNN$) are effective tools for signal production and trajectory generation. Previous studies have shown production of circular and figure-eight trajectories to be robust after training. We show here the effects of different sampling rates on the production of trajectories by the ATNN neural network, including the influence of sampling rate on the robustness and noise-resilience of the resulting system. Although fast training occurred with few samples per trajectory, and the trajectory was learned successfully, more resilience to noise was observed when there were higher numbers of

†This work was supported in part by the Institute for Systems Research at the University of Maryland (NSF CDR-88-03012), the Naval Research Laboratory (N00014-90K-2010), and the Applied Physics Laboratory of Johns Hopkins University. The authors would like to thank Dr. Panos Ligomenides for his initiation in this research.

samples per trajectory. The effects of changing the initial segments that begin the trajectory generation were evaluated, and a minimum length of initial segment is required but the location of that segment does not influence the trajectory generation, even when different initial segments are used during training and recall. A major conclusion from these results is that the network learns the inherent features of the trajectory rather than memorizing each point. When a recurrent loop was added from the output to the input of the ATNN, the the training was shown to result in an attractor of the network for a figure-eight trajectory, which involves more complexity due to crossover compared with previous attractor training of a circular trajectory. Furthermore, when the trajectory length was not a multiple of the sampling interval, the trained network generated intervening points on subsequent repetitions of the trajectory, a feature of limit cycle attractors observed in dynamic networks. Thus an effective method of training an individual dynamic attractor into a neural network is extended to more complex trajectories and to show the properties of a limit cycle attractor.

1 Introduction

Sampling rate has the effect of speeding up or slowing down the movement of the pattern along its trajectory in $\mathcal{R}^n$ and of translating forward or backward in time [6]. The pattern will still traverse the same spatial trajectory when a different sampling rate is applied. One of the basic requirements for digital computation in signal or image processing is that the signal be available in digital or binary form. Sampling theory states that a bandlimited signal sampled above its Nyquist rates can be recovered without error by low-pass filtering the sampled signal. Sampling rate effects in neural network models are not well studied as of

yet. In this paper, we study the effect of sampling rate on the training speed and recognition performance for the specific problem of circlar pattern production.

We use an algorithm with modifiable time-delays [8, 2] which is a powerful tool for the dynamic learning of a temporal network and we call this resulting network an Adaptive Time-Delay Neural Network (*ATNN*). The *ATNN* model employs adjustable time delays along the interconnections between two processing units, and both time delays and weights are adjusted according to system dynamics in an attempt to achieve the desired optimization.

The system is schematically illustrated in Figure 1. Assume the task is to emulate the embedding linear or nonlinear function of a plant at time T due to the temporally changing input signals $\hat{x}(t)$. During the training phase, switch $SW1$ is on (closed) and input signals are fed into a three layered network constructed by delay blocks. The inputs are pre-processed if necessary. After inputs are fed forward through the network, the outputs are obtained from the last terminal and compared with the desired outputs of the plant. An error vector E is propagated backwards through the previous layers and used to adjust the weights and delay variables in the intermediate delay blocks. Switch $SW2$ remains off (open) during training. The adaptation of the delays and weights are derived based on the gradient descent method to minimize the energy or cost function during training. Weight modification is based on error back-propagation ([12]) and the mathematical derivation of the time-delay modifications is done with a gradient descent approach [1, 2, 8, 7]. The weights and time-delays are updated step by step proportional to the opposite direction of the error gradient respectively. Processing units do not receive data through a fixed time window, but gather important information from various time delays which are adapted via the learning procedure. With these mechanisms, the network implements the dynamic delays along the

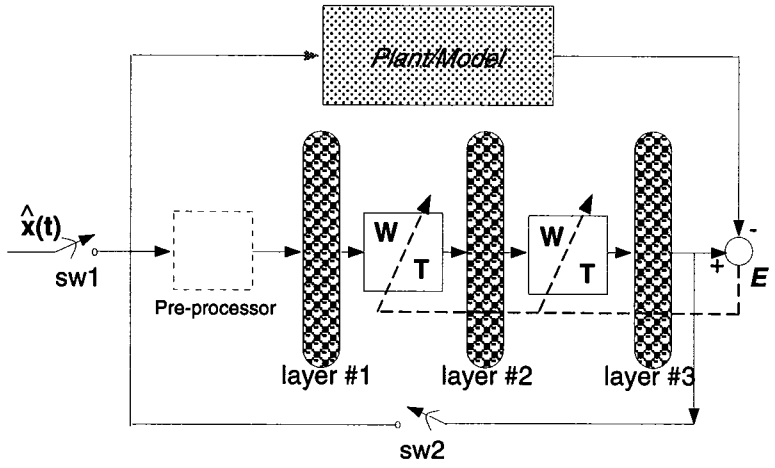


Figure 1: Schematic block diagram of system adaptation.

interconnections of the *ATNN*.

In some experiments, a production or recall phase is used, in which switch $SW1$ is turned off and $SW2$ is on and the outputs are recurrently fed back to the input terminal. The network can then generate a pre-trained trajectory or signal, and can thereby produce a good prediction of a series as complex as chaos. The feedback loop also allows for dynamic self-sustained activity and enables us to train attractors and trajectories into the network. The *ATNN* has been trained to distinguish the temporal properties and spatiotemporal correlations of various input patterns, and to perform predictions and signal generation [8, 9, 11, 10]. The network architecture is simple and efficient, and can achieve good generalization on applications problems.

2 Sampling Rate vs. Training Speed and Performance

The *ATNN* was trained on different sampling data: 64 points per circle, 32 points per circle, 21 and 16 points per circle, all well above the Nyquist rate. The simulation results show

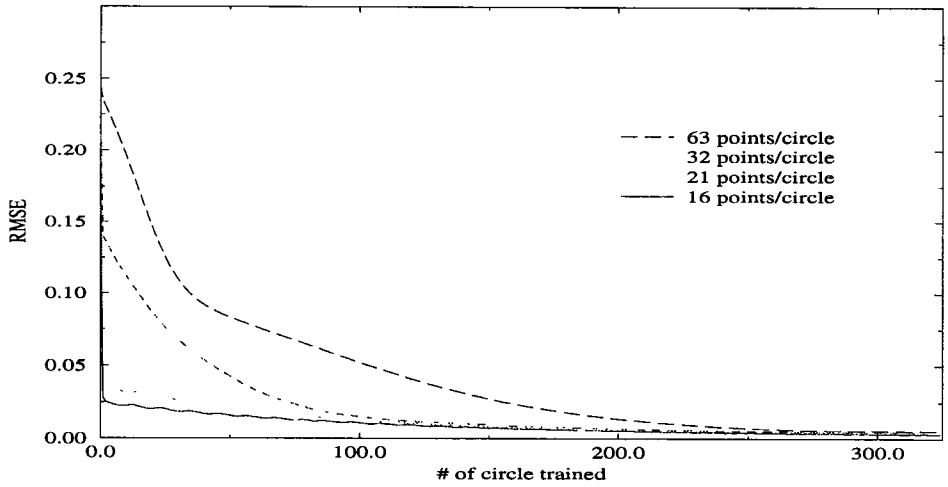


Figure 2: $RMSE$ of various sampling rate during training phase.

(Figure 2) that the network converges faster when the lower sampling rate data is employed. All cases could converge to similar $RMSE$ values shown in Table 1 and 2. The performance of the pattern retrieval during learning progress are shown in Figure 3 to Figure 6 for various sampling rates and iterations of training. Thus the network converges with different speed for different sampling rates. There were also different time delay requirements, as shown in Table 3. In the case with training sequence of 16 points per circle, the maximum time delay of the whole network was 34. The total time delay needed using 64 points per circle was only 17; the network trained on 16 points per circle required twice as much delay. Low sampling rate upgrades the training speed but requires a longer initial segment to recall the patterns.

Notice that the training series has a possibly infinite number of non-repeat points on the circle. Since the training pattern is constructed from $sin2\pi$ and $cos2\pi$ with 0.1 increment and π is an irrational number, each cycle has different points. This corresponds to the property of a limit cycle attractor where successive points generated by a system form a densely filled closed figure [5, 3].

points per circle	# of circle trained				
	1	5	10	25	50
63	2.4437e-01	2.2277e-01	2.0087e-01	1.2952e-01	8.3963e-02
32	2.2863e-01	1.2957e-01	1.1439e-01	7.8265e-02	4.3829e-02
21	2.1029e-01	3.5844e-02	3.1620e-02	2.7754e-02	2.1786e-02
16	1.6010e-01	2.4175e-02	2.2379e-02	2.0574e-02	1.6028e-02

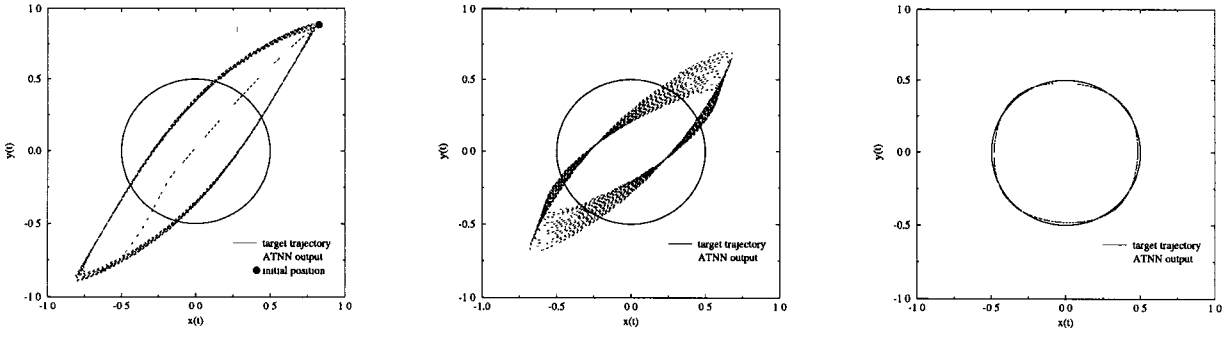
Table 1: $RMSE$ comparison of various sampling rate during training phase

points per circle	# of circle trained				
	100	150	200	250	300
63	5.2819e-02	2.7747e-02	1.3743e-02	7.4201e-03	5.6071e-03
32	1.5177e-02	9.6177e-03	6.8922e-03	5.4706e-03	4.9016e-03
21	1.2701e-02	8.4610e-03	5.6143e-03	4.4714e-03	4.4062e-03
16	1.0880e-02	7.3349e-03	5.7197e-03	4.2824e-03	3.3434e-03

Table 2: $RMSE$ comparison of various sampling rate during training phase

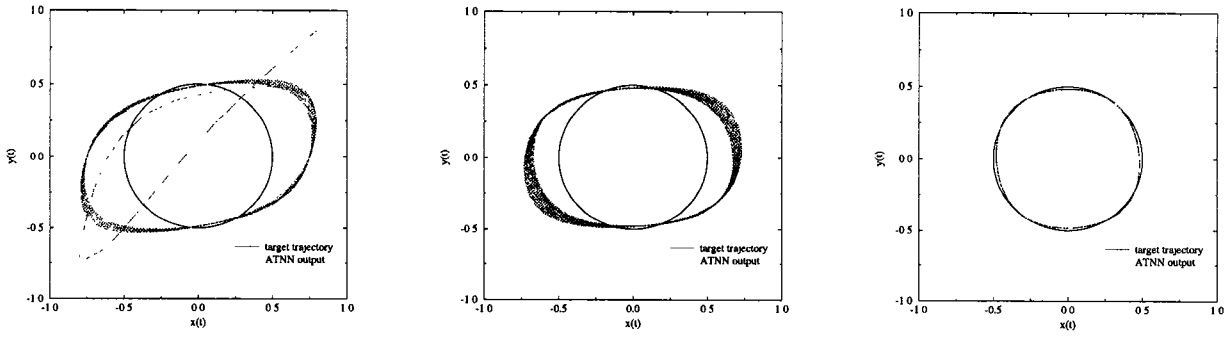
	# of points per circle			
	63	32	21	16
$max(T_1)$	3	8	13	8
$max(T_2)$	14	22	16	26
T_{atnn}	17	30	29	34

Table 3: The resulting time-delay variables after training is complete, where T_1 and T_2 denote the time-delay matrix on layer one and two respectively, and T_{atnn} is the total delay of trained network where $T_{atnn} = max(T_1) + max(T_2)$.



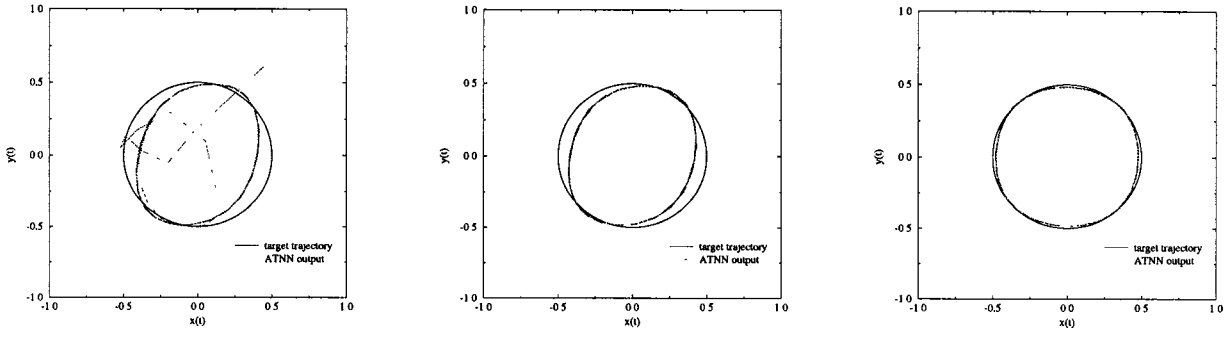
(a) 1 to 8 circles (b) 15 to 30 circles (c) after 250 circles

Figure 3: Performance progress during training when 64 point per circle is employed



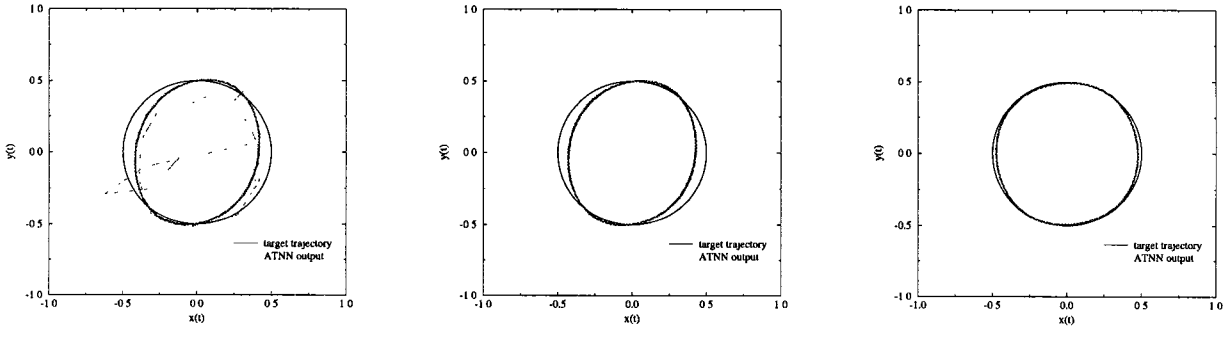
(a) 1 to 8 circles (b) 15 to 30 circles (c) after 150 circles

Figure 4: Performance progress during training when 32 point per circle is employed



(a) 1 to 8 circles (b) 15 to 30 circles (c) after 150 circles

Figure 5: Performance progress during training when 21 point per circle is employed

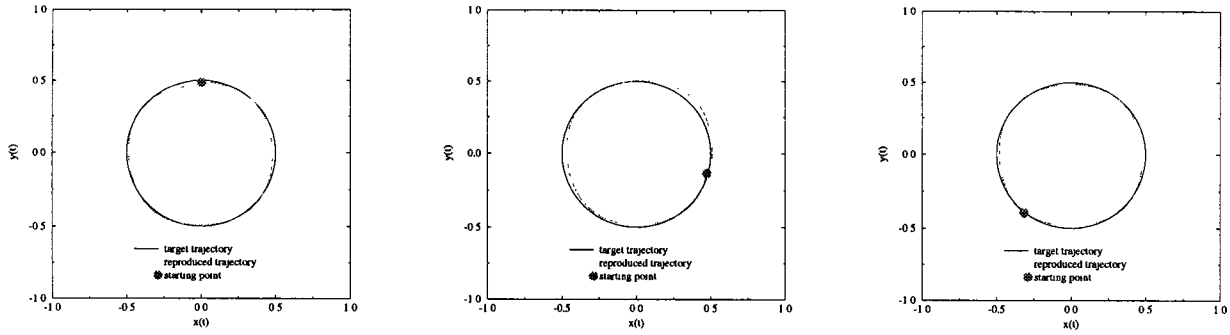


(a) 1 to 8 circles (b) 15 to 30 circles (c) after 150 circles

Figure 6: Performance progress during training when 16 point per circle is employed

points/circle trained	different starting position			average
	(a)	(b)	(c)	
64	0.0113	0.0162	0.0114	0.0130
32	0.0827	0.0170	0.0190	0.0395
21	0.2215	0.1036	0.0729	0.1326
16	0.1741	0.0776	0.0313	0.0943

Table 4: $RMSE$ of reproduced figure under different starting positions when various training sampling rates were employed.

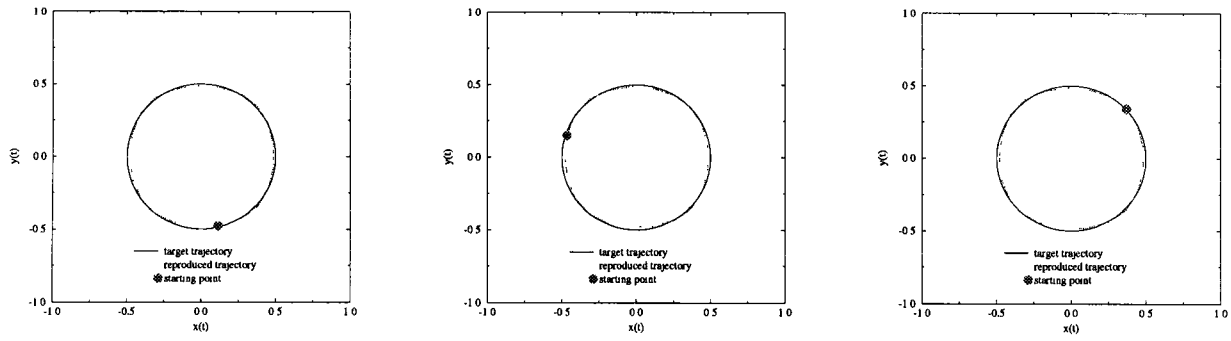


(a) (b) (c)

Figure 7: The results of circle reproduced by $ATNN$ start from different initial position (a) $RMSE = 0.0827$ (b) $RMSE = 0.1700$ (c) $RMSE = 0.0190$ when trained on 32 points per circle pattern.

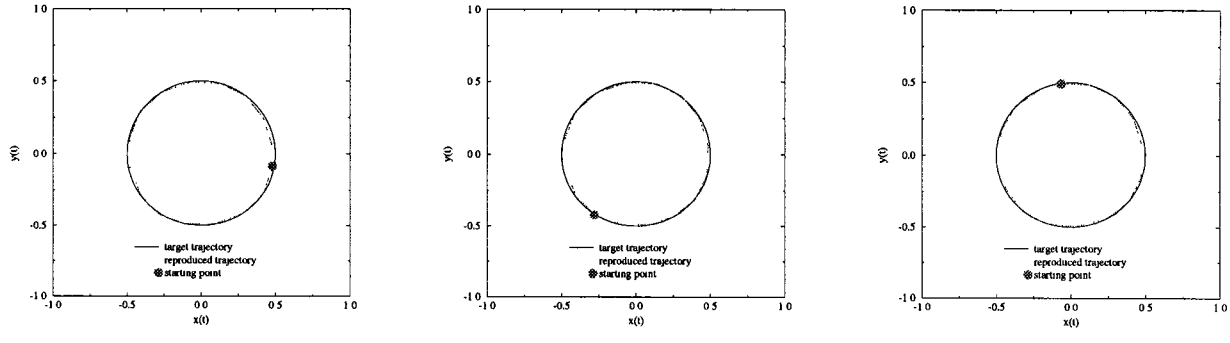
3 Reproduction From Different Starting Positions

Production capability of $ATNN$'s trained on various sampling rates were tested by giving at random different starting positions for each run. The numerical data is summarized in Table 4. The results are plotted in Figure 7, 8 and 9. We can conclude from this table that the performance of networks trained from 64 and 32 point sampling rates are similar but the performance from networks trained on 21 and 16 points are not as good.



(a) (b) (c)

Figure 8: The results of circle reproduced by *ATNN* start from different initial position (a) $RMSE = 0.2215$ (b) $RMSE = 0.1036$ (c) $RMSE = 0.0729$ when trained on 16 points per circle pattern.



(a) (b) (c)

Figure 9: The results of circle reproduced by *ATNN* start from different initial position (a) $RMSE = 0.1741$ (b) $RMSE = 0.0776$ (c) $RMSE = 0.0313$ when trained on 16 points per circle pattern.

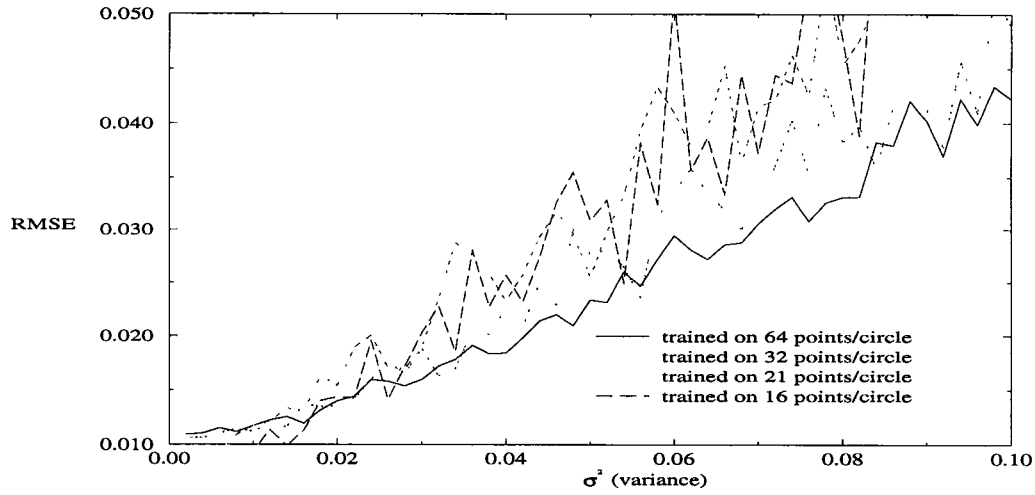


Figure 10: Noise tolerance of network trained from various sampling rate pattern.

4 Noise Tolerance

The noise tolerance is tested by adding different amounts of white noise to patterns and measuring the $RMSE$ of output and desired target. The $RMSE$ curves of each trained network under different sampling rate training patterns with noise variance of 0.001 to 0.1 are plotted in Figure 10. The noise resilience of lower sampling rate training patterns are not as good as that of network trained on 64 points per circle. The lower the sampling rate that is employed, the worse the outcome.

5 Noise Removal as Patterns with Different Sampling Rate are Trained

The noise removal capability was further examined to see if the network can clean up noise even if a low sampling rate is applied. Various amounts of noise were added to the original signal and we provided the network sufficient length of data to evaluate the new position;

σ^2_{noise}	order of circle reproduced						
	1st	2nd	3rd	4th	5th	6th	7th
0.0001	0.0018	0.0124	0.1280	0.7236	0.9886	0.9928	0.9928
0.001	0.0020	0.0123	0.1270	0.7216	0.9885	0.9928	0.9928
0.01	0.0102	0.0220	0.2656	0.8670	0.9794	0.9808	0.9808
0.1	0.0939	0.1433	0.7683	0.9767	0.9809	0.9808	0.9808
0.3	0.3140	0.6989	0.9872	0.9926	0.9928	0.9928	0.9928
0.5	0.5697	0.4375	0.9275	0.9805	0.9809	0.9808	0.9808

Table 5: $RMSE$ of successively generated circles from noisy initial data. The amount of noise was varied: $\sigma^2_{noise} = 0.0001$ *to* 0.5, and 32 points per circle is trained.

the newly generated signal was used to compose further data. The average $RMSE$ of each successive generated circle from the networks trained on 32, 21 and 16 points sampling rate are shown in Table 5, 6 and 7. To show the performance of each network intuitively, the generated patterns along each circle pass are selectively presented in Figures 12 to 17. When a small amount of initial noise was added to test the trajectory, the network could barely maintain reproduction at the second circle (see Figure 12(b), 14(b), and 16(b)), but it can no longer perform well beyond this stage (e.g. in Figure 12(c), 14(c), or 16(c)). When a larger amount of noise was applied, the network could no longer clean up any noise. Examples are shown in Figure 13, 15 and 17. To give a better view of the noise removal capability of various trained networks, the $RMSE$ variation along successive circles generated from different networks is shown in Figure 11. We can conclude that the network trained on lower sampling rates could no longer maintain noise removal since the neighboring points are too far apart from each other and the correlation information is insufficient.

σ^2_{noise}	order of circle reproduced						
	1st	2nd	3rd	4th	5th	6th	7th
0.0001	0.0001	0.0071	0.0221	0.1812	0.8355	1.0516	1.0547
0.001	0.0001	0.0072	0.0239	0.2009	0.8624	1.0523	1.0548
0.01	0.0110	0.0097	0.0312	0.2596	0.8608	1.0520	1.0562
0.1	0.0930	0.0682	0.3088	0.9046	1.0535	1.0562	1.0562
0.3	0.2722	0.1852	0.6787	1.0453	1.0547	1.0548	1.0548
0.5	0.6003	0.5221	0.9934	1.0542	1.0548	1.0548	1.0548

Table 6: $RMSE$ of successively generated circles from noisy initial data. The amount of noise was varied: $\sigma^2_{noise} = 0.0001$ *to* 0.5, and 21 points per circle is trained.

σ^2_{noise}	order of circle reproduced						
	1st	2nd	3rd	4th	5th	6th	7th
0.0001	0.0001	0.0001	0.0058	0.0101	0.0447	0.2267	0.7596
0.001	0.0001	0.0012	0.0060	0.0093	0.0387	0.1985	0.7111
0.01	0.0086	0.0103	0.0121	0.0533	0.2647	0.8100	1.0474
0.1	0.1067	0.0902	0.0578	0.2317	0.7526	1.0395	1.0611
0.3	0.2872	0.2972	0.2615	0.7463	1.0412	1.0611	1.0618
0.5	0.4830	0.4662	0.2583	0.3490	0.8685	1.0418	1.0493

Table 7: $RMSE$ of successively generated circles from noisy initial data. The amount of noise was varied: $\sigma^2_{noise} = 0.0001$ *to* 0.5, and 16 points per circle is trained.

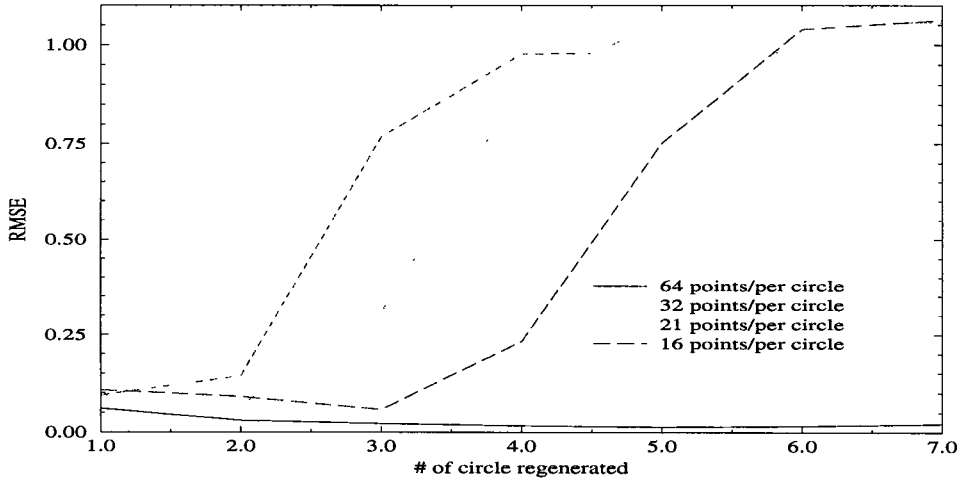
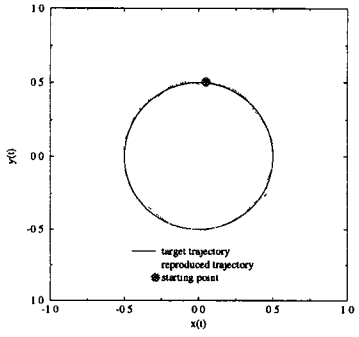
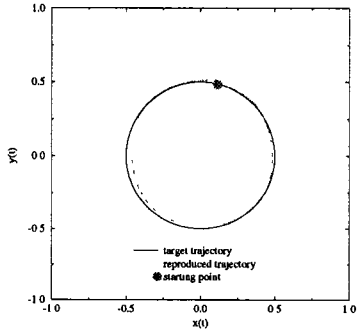


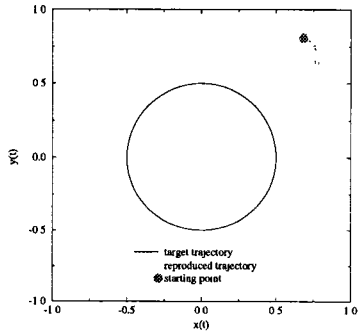
Figure 11: Comparison of $RMSE$ of successive circle reproduced when various sampling rate signals are trained.



(a) 1st circle regeneration, $RMSE = 0.0102$

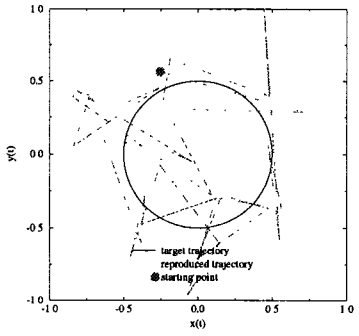


(b) 2nd circle regeneration, $RMSE = 0.0220$

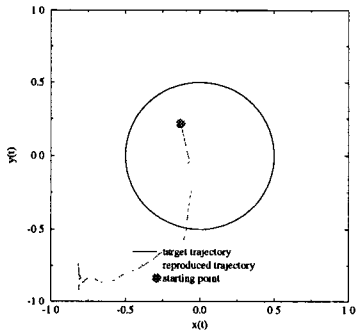


(c) 4th circle regeneration, $RMSE = 0.8670$

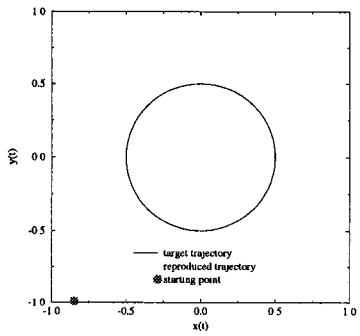
Figure 12: *ATNN* can barely remove small amount of noise (variance $\sigma^2 = 0.01$) at the 2nd sequence but not for the subsequent path when 32 points per circle is trained.



(a) 1st circle regeneration, $RMSE = 0.3140$

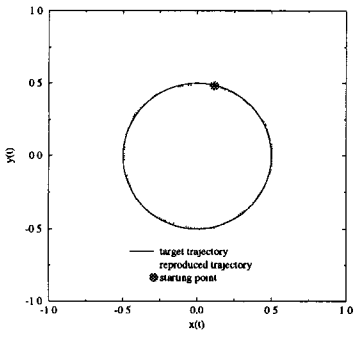


(b) 2nd circle regeneration, $RMSE = 0.6989$

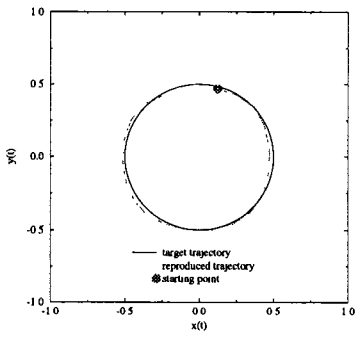


(c) 4th circle regeneration, $RMSE = 0.9926$

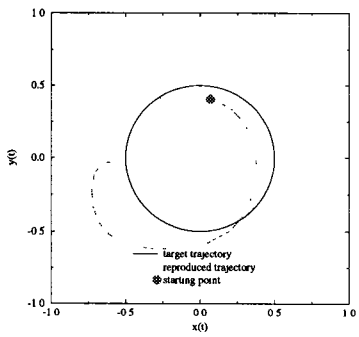
Figure 13: *ATNN* can no longer clean up the noise when larger amount of initial noise is added (variance $\sigma^2 = 0.3$) as 32 points per circle is trained.



(a) 2nd circle regeneration, $RMSE = 0.0097$

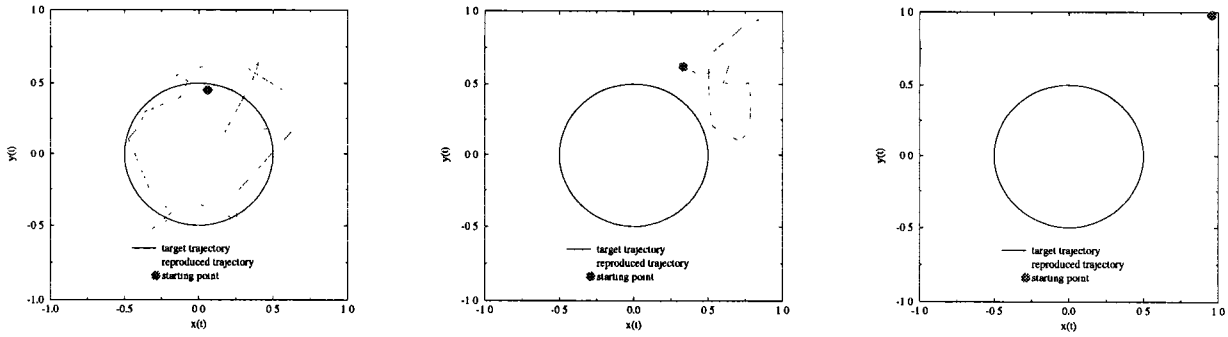


(b) 3rd circle regeneration, $RMSE = 0.0312$



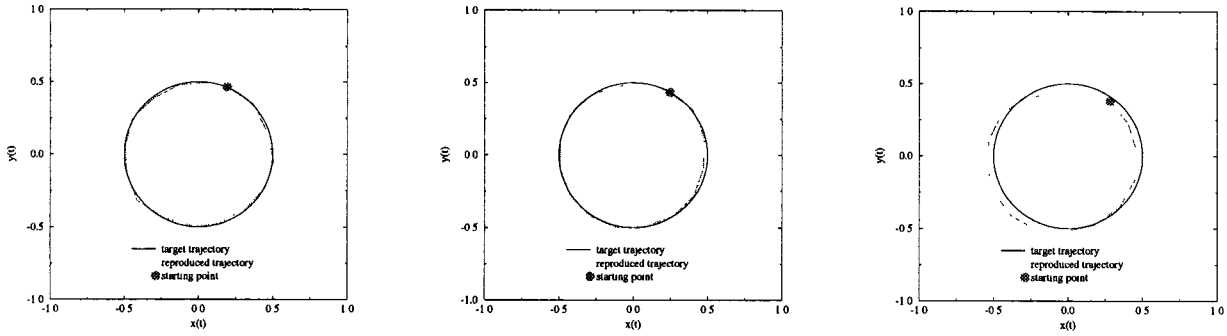
(c) 4th circle regeneration, $RMSE = 0.2596$

Figure 14: *ATNN* can barely remove small amount of noise (variance $\sigma^2 = 0.01$) at the 2nd sequence but not for the subsequent path when 21 points per circle is trained.



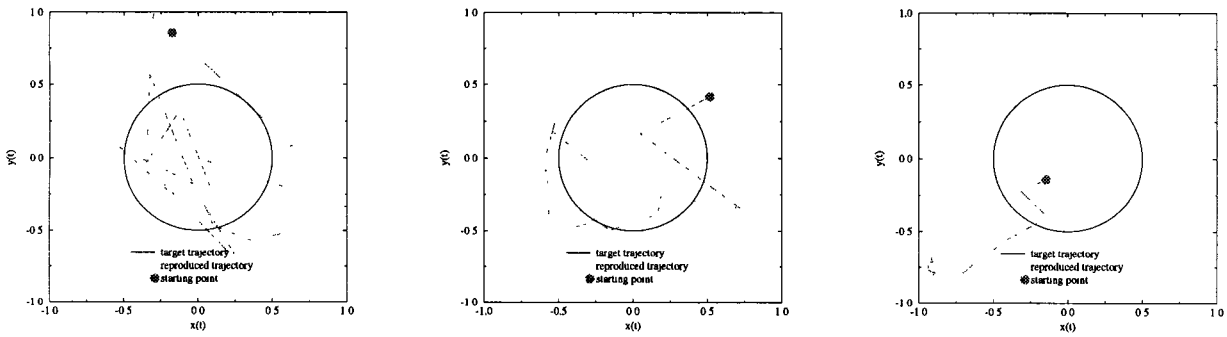
(a) 2nd circle regeneration, $RMSE = 0.1852$ (b) 3rd circle regeneration, $RMSE = 0.6787$ (c) 4th circle regeneration, $RMSE = 1.0453$

Figure 15: $ATNN$ can no longer clean up the noise when larger amount of initial noise is added (variance $\sigma^2 = 0.3$) as 21 points per circle is trained.



(a) 2nd circle regeneration, $RMSE = 0.0103$ (b) 3rd circle regeneration, $RMSE = 0.0121$ (c) 4th circle regeneration, $RMSE = 0.0533$

Figure 16: $ATNN$ can barely remove small amount of noise (variance $\sigma^2 = 0.01$) at the 2nd sequence but not for the subsequent path when 16 points per circle is trained.



(a) 2nd circle regeneration, $RMSE = 0.2972$ (b) 3rd circle regeneration, $RMSE = 0.2615$ (c) 4th circle regeneration, $RMSE = 0.7463$

Figure 17: $ATNN$ can no longer clean up the noise when larger amount of initial noise is added (variance $\sigma^2 = 0.3$) as 16 points per circle is trained.

6 Conclusion

We have studied sampling rate vs. training speed and performance, reproduction from different starting positions, and noise removal as patterns of different sampling rates were trained. Sampling rates are important in determining signal recovery or production, and in determining training speed. Superior performance for trajectory reproduction and noise removal were found at high sampling rates.

This work is consistent to Doya's work in the study of the adaptive neural oscillator using continuous time back-propagation learning [4]. Through their extensive experiments, the network has difficulty to learn and regenerate the input wave form if the period T of the external input is larger. In other words, the net can not catch the dynamics efficiently when the the input oscillates up and down too quickly.

References

[1] S. P. Day and M. Davenport. Continuous-time temporal back-propagation with adaptive time delays. Neuroprose archive, Ohio State University. Accessible on Internet via anonymous ftp on archive.cis.ohio-state.edu, in pub/neuroprose/day.tempora.ps August, 1991.

[2] S. P. Day and M. R. Davenport. Continuous-time temporal back-propagation with adaptive time delays. *IEEE Trans. on Neural Networks*, 4(2):348–354, March 1993.

[3] J.E. Dayhoff, P. Palmadesso, and F. Richards. Developing multiple attractors in a recurrent neural network. In *World Congress on Neural Networks*, volume 4, pages

710–715, San Diego, CA, 1994. INNS Press, New York.

[4] K. Doya and S. Yoshizawa. Adaptive neural oscillator using continuous-time back-propagation learning. *Neural Networks*, 2:375–385, 1989.

[5] B. Doyon, B. Cessac, M. Quoy, and M. Samuelides. Control of the transition of chaos in neural networks with random connectivity. *Internation Journal Bifurcation and Chaos*, 3(2):279–291, 1993.

[6] R. Hecht-Nielsen. *Neurocomputing.* Addison-Wesley, Reading, MA, 1990.

[7] D.-T. Lin, J. E. Dayhoff, and P. A. Ligomenides. Adaptive time-delay neural network for temporal correlation and prediction. In *Intelligent Robots and Computer Vision XI: Biological, Neural Net, and 3-D Methods, Proc. SPIE*, volume 1826, pages 170–181, Boston, November, 1992.

[8] D.-T. Lin, J. E. Dayhoff, and P. A. Ligomenides. A learning algorithm for adaptive time-delays in a temporal neural network. Technical Report SRC-TR-92-59, Systems Research Center, University of Maryland, College Park, Md 20742, May 15 1992.

[9] D.-T. Lin, J. E. Dayhoff, and P. A. Ligomenides. Learning spatiotemporal topology using an adaptive time-delay neural network. In *World Congress on Neural Networks*, volume 1, pages 291–294, Portland, OR, 1993. INNS, New York.

[10] D.-T. Lin, J. E. Dayhoff, and Panos A. Ligomenides. Prediction of chaotic time series and resolution of embedding dynamics with the ATNN. In *World Congress on Neural Networks*, volume 2, pages 231–236, San Diego, CA, 1994. INNS Press, New York.

[11] D.-T. Lin, P. A. Ligomenides, and J. E. Dayhoff. Spatiotemporal topology and temporal sequence identification with an adaptive time-delay neural network. In *Intelligent Robots and Computer Vision XII: Algorithms and Techniques, Proc. SPIE*, volume 2055, pages 536–545, Boston, September, 1993.

[12] J.L. McClelland, D.E. Rumelhart, and the PDP Research Group. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 2. MIT Press, Cambridge, 1986.