Chapter 7

OSCILLATION RESPONSES IN A CHAOTIC RECURRENT NETWORK

Judith E. Dayhoff

Complexity Research Solutions, Inc.
Silver Spring, MD

Peter J. Palmadesso

Plasma Physics Division
Naval Research Laboratory
Washington, D.C.

Fred Richards

Dynamic Software Solutions
Alexandria, VA

I. INTRODUCTION

Dynamic neural networks are capable of prolonged self-sustained activity patterns, in which each neuron has an activation level that can change over time, and different neurons can be at different levels, with different changes over time. An enormous repertoire of self-sustained activity patterns is possible, due to the wide variety of oscillations that networks can engage in. Oscillations include n-state oscillations (repeating finite sequences of states), limit cycles (quasiperiodic), and chaos. An infinitude of different oscillations is possible for each type. The resulting set of possible oscillations and activity patterns has potential for increasing the capacity and capability of neural networks for computational purposes. Whereas feedforward neural networks have extensive applications in recognition and control, with many highly successful performances in applications, their output is a fixed vector. Single-layer networks, such as those studied by Hopfield, have fixed vectors (stable states) as output. In contrast, dynamic networks can produce a wide variety of oscillations as output, and enhanced computational properties are expected to be realized by dynamic neural networks that oscillate.

Recently, neurobiological investigators have suggested that the brain may use chaotic behavior and oscillatory states to store and recognize patterns, and that chaotic states are "waiting" states, whereas simpler oscillators are recognition and response states [Yao and Freeman, 1990, Yao *et al*, 1991, Freeman *et al.*, 1988]. Theoretical investigations have shown a tremendous variety of oscillatory states possible, and systematic ways to produce chaos in a network, along with a progression of oscillatory states moving a network from a fixed point to chaos [Sompolinsky *et al.*, 1988, Doyon *et al.*, 1993, Palmadesso and Dayhoff, 1994]. Some investigators have researched computations that use limit cycles, strange attractors, chaos, or transient behaviors in associative memory or information processing models [Kumagai *et al.* , 1996, Yao, 1991, Chapeau-Blondeau, 1993, Moreira and Auto, 1993, Dmitriev *et al.*, 1993, Dmitriev and Kuminov, 1994, Hjelmfelt and Ross, 1994, Wang, 1996, Lin *et al.*, 1995, Dayhoff *et al.,* 1998]. These studies make it natural to suggest that new paradigms using oscillations as final states are likely to enhance the development of powerful methods for information processing with artificial neural networks. This prospect has received increasing research interest recently and requires more characterization of dynamic network activities.

Here we study chaotic neural networks that can be used to produce pattern-to-oscillation maps. First, chaotic behavior is developed in a sparsely connected neural network with random weights, as described by Doyon [Doyon *et al.*, 1993], [Doyon *et al.*, 1994]. To the chaotic network, an external pattern is applied and the network usually locks into a simpler dynamic attractor, consisting of a limit cycle or simple n-state oscillator. A range of intensity for the applied pattern has been considered, and the resulting attractor changes as the intensity is increased. An increase in intensity eventually results in a stable fixed-point attractor. When different patterns are applied with the same intensity, they evoke different oscillations and differing dynamic activities. Adjusting the pattern intensity helps to produce a pattern-to-oscillation map. This approach has promise for developing paradigms in which the evoked attractor represents a memory, pattern class, or optimization result. The neural network model used in this chapter differs from that of other investigators cited above, except for previous work by Samuelides and coworkers [Cessac, 1994, Cessac *et al.*, 1994, Cessac, 1995, Cessac and Quoy, 1995, Doyon *et al.*, 1993, Quoy *et al.*, 1995], who have presented a variety of theoretical results for the network's dynamics without an applied external pattern, and have found positive results on Hebbian weight adjustment in the presence of applied external patterns. Here we present results on the effects of variations in pattern strengths, the adaptation of the pattern strength variable, and the responses of the network to noisy patterns.

In Section 2, we illustrate a progression to chaos in a network with no stimulus patterns [Doyon *et al.*, 1993]. A series of different attractors is obtained. Different initial states usually do not evoke different attractors. The application of external patterns to the network is addressed in Section 3. Effects of increasing pattern strengths are shown, and capacity and uniqueness of evoked oscillations is illustrated. Some resilience to pattern noise is attained. Dynamic adjustment of pattern strength is described in Section 4, which results in a pattern-to-oscillation map that is unique and results in a low complexity oscillator in response to each pattern. Further characteristics of the pattern-to-oscillation map, including resilience to pattern noise, are discussed in Section 5. The impact of this type of approach is discussed in Section 6.

II. PROGRESSION TO CHAOS

Dynamic attractors can be developed in a random neural network, and a progression from a single fixed-point stable state to a chaotic oscillation can be obtained. The construction of the network is as follows. A random network with full or sparse interconnections is configured as a single layer of neurons, where closed-loop connections are allowed and weights are set at random. A multiplier g can be applied to all of the weights at the same time, and, when g is increased sufficiently, chaotic behavior occurs [Sompolinsky *et al.*, 1988, Doyon *et al.*, 1993]. Thus the network can be modulated into a chaotic state. The neural units are simple biologically inspired units, performing a weighted sum.

$$a_j(t+1) = f\left(\sum_{i=1}^{N} g w_{ji} a_i(t)\right) \tag{1}$$

where $a_j(t)$ = activation of unit j at time t, w_{ji} = weight to unit j from unit i, N = the number of processing units, g a multiplier, and f a nonlinear squashing function. The squashing function used in our experiments was

$$f(x) = \left[\frac{1}{1+e^{-x}} - 0.5\right] * 2 \tag{2}$$

Reciprocal connections did not have to be the same (e.g., $w_{ij} \neq w_{ji}$), and self-loops were suppressed in our experiments as a simplification. The activation values could vary from -1.0 to 1.0, and the parameter g is a multiplier for all weights and can be set at any value greater than zero. The interconnections and weights were chosen at random. Networks were denoted as (N, K), where N was the total number of processing units, and K the number of incoming interconnections for each unit. The K units that sent interconnections to each unit were selected at random, and the values of the weights were selected from a uniform random distribution [-1,1], with the random variable divided by K, as specified in Doyon *et al.*, [1993]. With this model, we have examined a variety of paths from single fixed-point attractors to chaos.

The parameter g is a multiplier for all weights. Thus, the original set of weights becomes amplified or de-amplified depending on whether g>1 or g<1. A stronger set of driving forces is then presented to each neuron as g increases above 1.The incoming sum for neuron j is

$$S_j = \sum g w_{ji} a_i \tag{3}$$

and the modulated weight is gw_{ji}. The neuron then performs the squashing function to determine its next activation value:

$$a_j(t+1) = f(S_j) \tag{4}$$

The parameter g can also be considered as a scaling of the x-axis in the squashing function. Organizing equation (1) differently, we get

$$R_j = \sum_{i=1}^{N} w_{ij} a_i(t) \tag{5}$$

where R_j is now the incoming sum for unit j, and

$$a_j(t+1) = f_g R_j \tag{6}$$

where f_g = f (gx), a sigmoid squashing function with a re-scaled horizontal axis. Here the weight is not modulated by g, but the horizontal scale of the sigmoid is modulated by g. In both contexts, g is key to producing chaos.

Figure 1 shows the symmetric sigmoid function, with the x=y line. Figure 2 shows the function f_{10}, which compresses the horizontal direction of the squashing function so that two pockets form, bounded in part by the x=y line. This curvature in turn causes there to be two absorbing states, at the upper and lower intersections, for the single neuron case [Dayhoff, 1998]. It is thus not surprising that dynamic attractors - oscillations and chaos - develop at increased values of g.

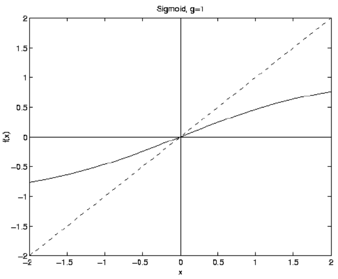


Figure 1. Symetric sigmoid function, with the x=y line

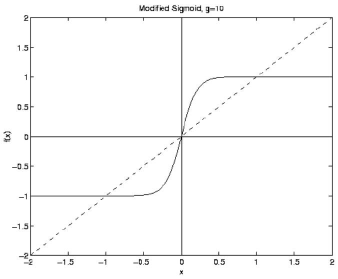


Figure 2. Modified sigmoid function with g=10

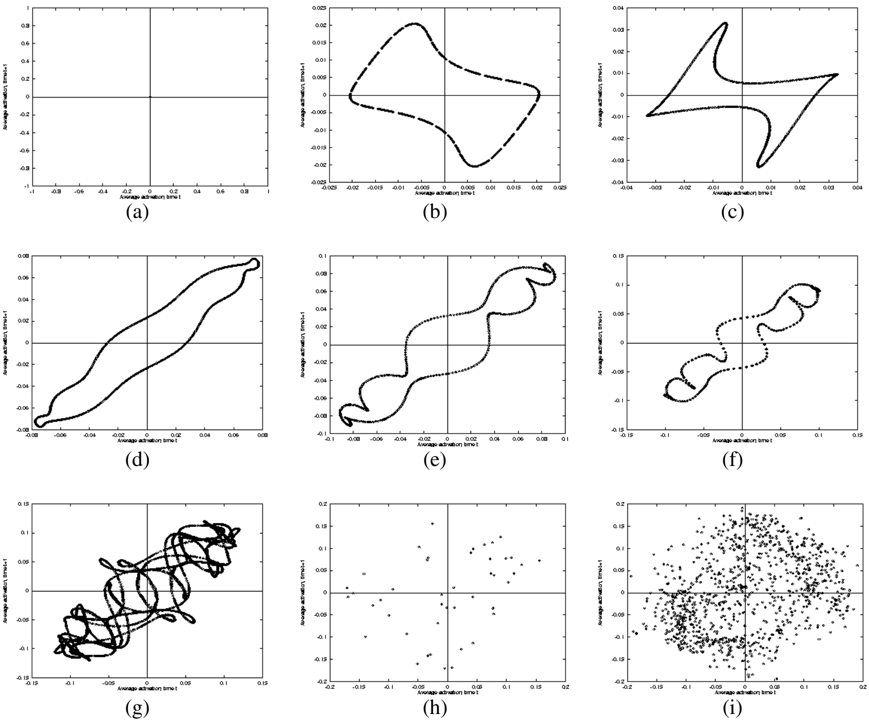
A network with 64 units, each with 16 incoming connections, was constructed with random initial weights. Transitions from fixed-point attractors to chaotic attractors were observed as g was increased starting from numbers below 1.0. Figure 3 illustrates such a progression, from a fixed point attractor to a chaotic attractor, with average activation a(t+1) graphed as a function of a(t), to form a map of the dynamics. Use of the average activation in the plot was chosen to project the many dimensions (64 activation levels) to a single measured observation over time.

For low values of g (e.g., g=0.9), a single fixed-point attractor was observed, shown in Figure 3(a), which has a single point at (0,0). When g was increased to 1.0, a limit cycle appeared (Figure 3(b)), consisting of a dense set of points along a closed loop. When g was increased to 1.1, the four corners of the closed loop became pointed, and the span of the graph increased from +/- 0.025 to +/- 0.04 (Figure 3(c)). When g was increased to 1.2, the limit cycle took on a new shape along the x=y diagonal, with an expansion in range to within +/- 0.08 (Figure 3(d)). In Figure 3(e), g was increased to 1.3, and the limit cycle developed several wiggles, and expanded in range to +/- 0.1. When g=1.4 (Figure3 (f)), the wiggles appear deeper, and the span increases to within +/- 0.15. When g was increased to 1.5, the previously observed closed loop appears to have changed to a figure that roughly outlines the previous loop several times, with different variations each time, and gives the appearance of scribbling (Figure 3(g)). When g=1.6, a locking occurs into a finite-state oscillator (Figure 3(h)). When g=1.7 (Figure 3(i)), chaotic or irregular behavior occurs. This type of progression shows the tremendous range and complexity of dynamic attractors and the ability to exert some control over their appearance, through varying the single parameter g. Figure 3 illustrates a progression for one network of 64 neurons; Doyon and colleagues [Doyon *et al.*, 1993] have characterized four types of paths to chaos in networks of 128 neurons. In some cases, increasing the number of iterations calculated allowed activity that appeared chaotic to resolve into a limit cycle. Sometimes the number of transients in these cases would be too large to be practical to implement in a computational application. Thus, for practical use to be made of dynamic neural networks, a "chaotic" response would be considered to be behavior that appeared to be chaotic in a limited predefined time frame

A. ACTIVITY MEASUREMENTS

We have experimented with measurements that reflect the type of activity observed in the maps drawn in Figure 3. An activity measurement was computed by an algorithm that uses the average activation over the entire network over a period of time, usually 1000 iterations of Equation (1). The range of the graph of a(t*)* versus a(t+1*)*, as in Figure 3, is defined to cover the points generated by the *n* iterations used. The graph is then divided into a 10 by 10 grid, with evenly spaced lines. The percentage of non-empty grid squares is the grid coverage. Grid coverage tends to reflect the types of activity shown in Figure 3, and thus can be used as an appropriate measurement or reflection of the network's activity.

Chaotic behavior such as in Figure 3(i) tends to have coverage values in the range >73. Limit cycles tend to be in the range 43-57, and n-state oscillations, such as Figure 3(h) are in a range >2, usually small. The coverage measurement can be used in the method shown next for adjusting the pattern strengths in the pattern-to-oscillation map. Other measurements of activity could be substituted for grid coverage. Ideally, the best activity measurement would be always high when chaotic, irregular behavior occurs and progressively lower as the network dynamics progresses to a fixed point. Although the grid coverage usually adheres to this pattern, there are exceptions.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 3. Progession from fixed point to chaos in a random (64, 16) network. The horizontal axis is average activation at time t+1, a(t-1) and the vertical axis is average activation at time t, a(t). (a) g=0.9, (b) g= 1.0, (c) g=1.1, (d) g=1.2, (e) g=1.3, (f) g=1.4, (g) g =1.5, (h) g=1.6, (i) g=1.7

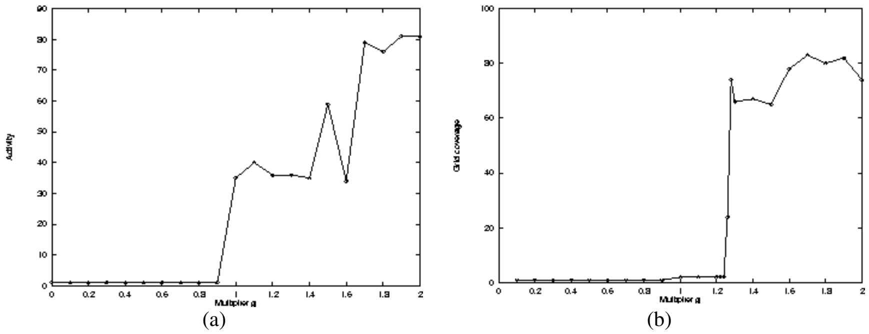
Figure 4 shows the coverage measurement graphed versus the value of the multiplier g, for two different networks. The coverage measurement is 1 for small values of g, and then increases until the coverage value is consistently at a high level, generally above about 70. Usually, once a high level of coverage is attained, reflecting irregular activity or chaos, higher values of g continue to evoke high coverage levels. The lowest value of g for which a high coverage level is attained is an indicator that the network is at the "edge of chaos".

B. DIFFERENT INITIAL STATES

It is computationally interesting to find a paradigm whereby a pattern-to-oscillation map can be produced, because then the initial state of the network could be set from a pattern vector, and the evoked oscillation could represent a pattern class or other computational answer evoked by the initial pattern. We first explored whether different initial states of the network would produce different attractors in the random networks described above, but were not satisfied with the potential of this approach, as described next.

We generated a series of random networks with a (64,16) configuration, as described in Section 2. A series of random pattern vectors were used as initial states to the network. The resulting attractor was then observed. In most cases, only one attractor was observed, which was reached from a wide variety of initial states. The initial states in this experiment were generated at random from a [-1,1] uniform distribution. Sometimes there were two attractors (limit cycles or n-state oscillations), reached from different initial states, but they were symmetric with one another, having a 180 degree rotational symmetry about the origin. Figure 5 illustrates such an example. About half of the initial states (taken at random) evoked each attractor. Although different initial conditions could evoke different (but symmetric) limit cycles in this case, this scenario does not offer enough flexibility to discriminate patterns by the limit cycles they evoke.

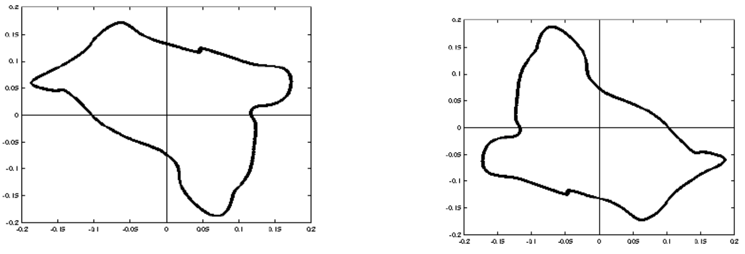
Occasionally, different attractors were observed from different initial states during our simulations of networks with random weights. This circumstance was very rare among our observations, and in these cases the parameter g was first tuned to be very near a bifurcation point. Figure 6 shows such an example. Figure 6(a) shows an 8-state oscillator. This occurs at g=1.72 in a (64,16) random network. Figure 6(b) shows the same network except that a different initial state was selected at random. The 8 points appear to have bifurcated into 8 rings, which are asymmetric about the origin. A third random initial state evoked the limit cycle in 6 (c), which is symmetric to that in 6 (b). It was also possible to evoke an oscillator symmetric to that of 6(a) with other initial states. Although different initial conditions evoke different attractors in this case, this scenario also does not offer enough flexibility to discriminate many patterns by the limit cycles they evoke. However, this does not rule out the possibility of creating intricate and desired distinctions between different types of initial states according to which attractor is evoked if suitable weight adjustments are made.



(a)

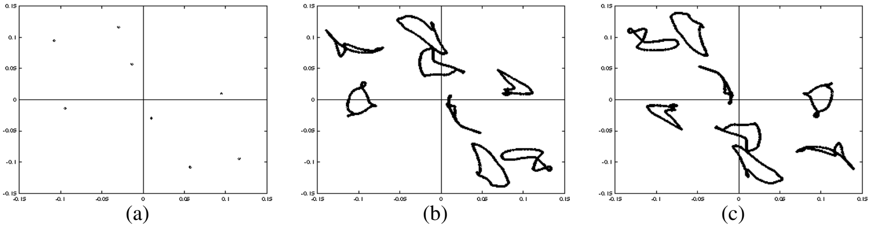
(b)

Figure 4. Activity level – in this case, coverage level – as a function of g, for two different networks. (a) For this network, a bifurcation was exhibited when g = 0.9, to generate an oscillation instead of a fixed point. The map appeared chaotic when g reached 1.7, and for higher values of g. (b) For this network, a rapid set of bifurcations occurred to generate oscillations at g = 1.26 and the coverage of the map was above 60 until g = 1.6, where chaotic behavior occurred. Both networks were (64,16) with randomly assigned weights as described in Section A.



(a) (b)

Figure 5. Symmetric attractors evoked by different initial states in the same network. The initial state of the network was set so that neuron activation levels matched a random pattern vector
E $(a_i(0) \leftarrow e_i)$.



(a) (b) (c)

Figure 6. A case where different initial states lead to different attractors, from a random (64,16) network. (a) An 8-state oscillator (b) A limit cycle with 8 closed loops (c) A limit cycle symmetric to part (a).

III. EXTERNAL PATTERNS

Next, we treat a pattern vector as an external stimulus, to overcome the limitations in flexibility encountered when patterns are applied as initial states. To include an external stimulus, the updating equation for the neurons (1) can be modified as follows:

$$a_j(t+1)=f\left(\sum_{i=1}^{N} g w_{ji} a_i(t)+\alpha e_i\right) \qquad (7)$$

where E=$(e_1,e_2,\ldots,e_n)$ is the external input pattern. The input E is then applied at every time step, and its strength is modulated by the multiplier α. The vector E can then be assigned as a pattern to be classified, and a pattern-to-oscillation map can be generated.

The network is initially put in a chaotic oscillation. The chaotic net does not have an external stimulus, and updates by (1). To produce the chaotic network, the parameter g is increased until the network reaches chaotic behavior. Typically, we do not increase g more than is necessary to produce chaotic behavior, so the network can be said to be at the "edge of chaos". An external input E is then applied to the chaotic net, and the network uses (7) to update. The externally applied input usually "locks" the chaotic network into a simpler attractor, often a limit cycle. This scenario can thus be called "attractor locking".

The attractor that results depends on the characteristics of the externally applied pattern (for the same network) and the weights and configuration of the network. A chaotic network was updated for varying amounts of time without an external input, and regardless of when E was applied, the same dynamic attractor was observed. Figure 7 shows the results when a chaotic behaving network receives an external stimulus pattern. The graph of Figure 7(a) shows activity of the chaotic network, without any applied pattern. The other graphs show the results of applying different external patterns. The patterns were generated at random from a uniform distribution [-1,1].

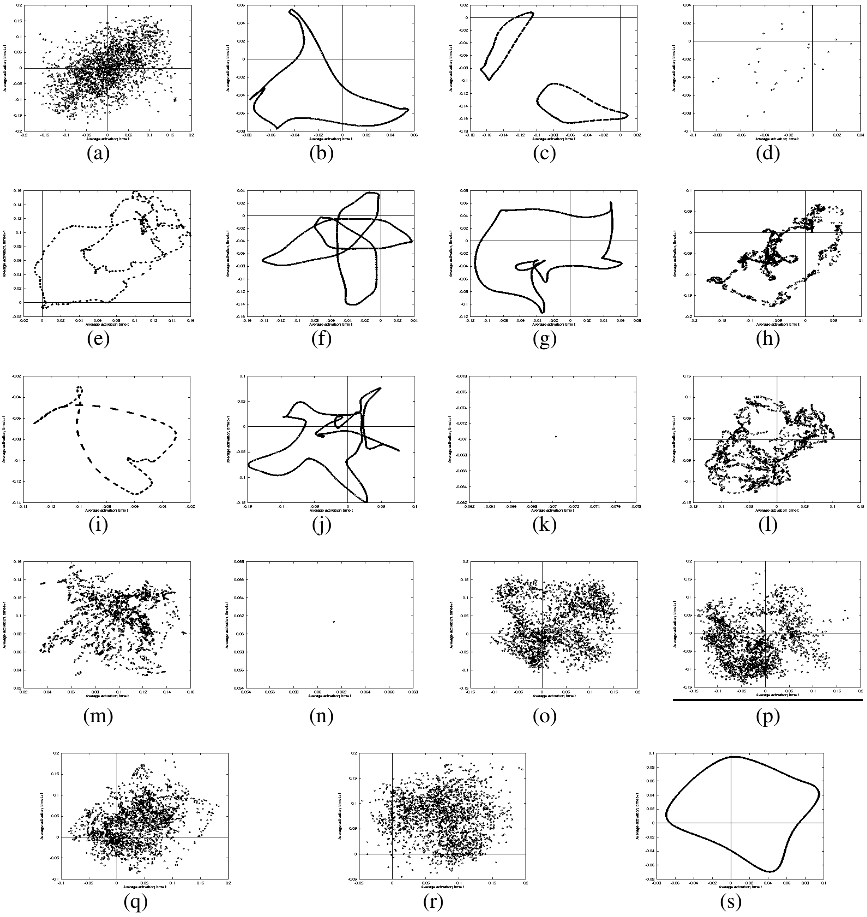
A. PROGRESSION FROM CHAOS TO A FIXED POINT

Each pattern has a strength, α, as shown in equation (7). As the pattern strength α is increased, the dynamics of the network's activity moves through a progression to a fixed point. In all cases examined, a sufficiently high α produced a fixed point. Figure 8 shows the progression of a network's dynamics as a function of α ((a)-(i)), from chaos (a) to a fixed point (i). One external pattern, at increasing strengths, was applied to the same chaotic network.

Figure 9(a) shows activity measurements as a function of α for the same network. Figure 9(b) is based on another random network, with a different random pattern. The activity measurement always starts high (generally >*73*), which is indicative of chaotic oscillation. When α is increased, there is a point at which the activity measurement enters a mode that has limit cycles or finite state oscillations (usually activity values of about 2-57). At sufficiently high α, the activity becomes a fixed point. These transitions happen at different values for α for the two examples shown in Figure 9.

B. QUICK RESPONSE

Figure 10 shows the results of applying an external pattern to a network, including transients before a limit cycle is obtained. All points were plotted after application of the external pattern. The transients were included in the figure to illustrate that sometimes very few transients occur after the external pattern is applied and before the limit cycle is attained. The possibility for fast classification to occur when a network is initially in a chaotic state has been suggested by Freeman [Yao and Freeman, 1990], in reference to olfactory neural models that wait for stimuli in a chaotic state and lock into a simpler oscillation after a stimulus is received. Because such fast classification of patterns with chaotic networks is of interest, we explored the relationship between the number of transients and the value of the pattern strength, using the (64,16) networks described in Section 2.



(a) (b) (c) (d)

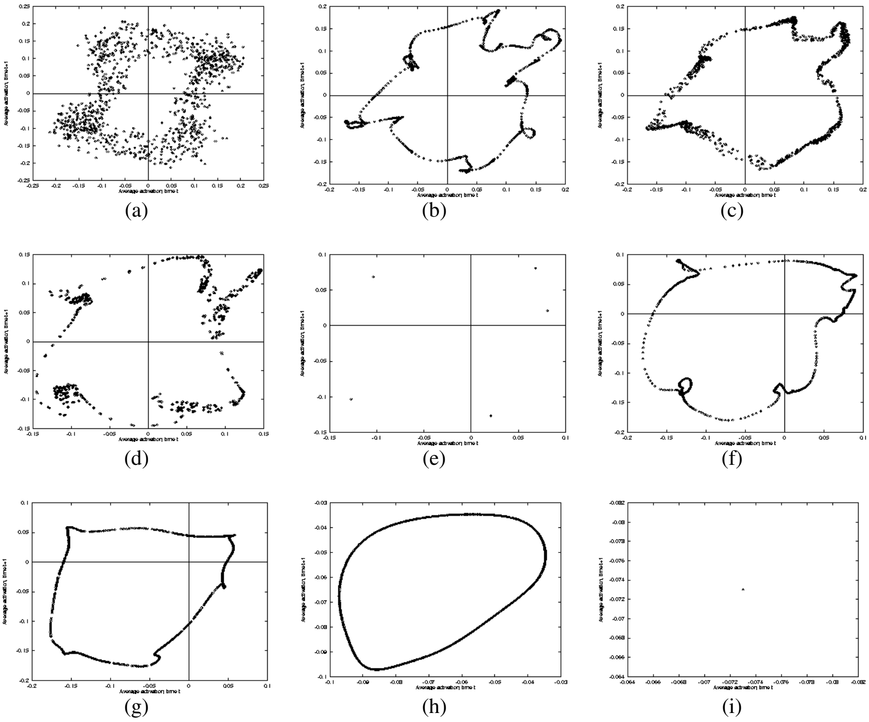
(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n) (o) (p)

(q) (r) (s)

Figure 7. A chaotic network has 18 different patterns applied, with different results on the dynamics. (a) Activity of the chaotic network, before an external pattern is applied. Multiple g is set just above the value where chaotic activity occurs. (b-s) Activity of the network after 18 different patterns were applied. Evoked dynamics is highly unique. Most graphs show recognizable low-order dynamics – limit cycles and n-state oscillations – but some show chaotic (irregular) behavior and others show fixed points only. Graphs have the horizontal axis as average activation at time *t* + *1*, a(*t* – *1*), and the vertical axis as average activation at time *t*, a(*t*). The patterns were generated from a uniform random distribution [-1, 1].

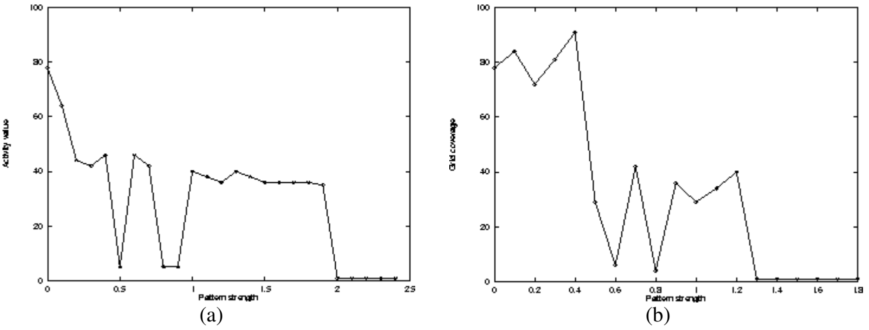


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 8. Progression of maps for the same pattern applied to the same chaotic network, at increasing pattern strengths.



(a) (b)

Figure 9. Activity measurement (grid coverage) as a function of pattern strength for two different patterns, applied to two different chaotic networks.

Figure 11 shows results on a random (64,16) network, where the number of observed transients is graphed as a function of pattern strength. The decrease in the number of transients above $\alpha=1.4$ correlates with a limit cycle (ring) that shrinks into a fixed point as the pattern strength increases further. In the region where $\alpha < 1.4$, there is a path from the first limit cycle, obtained at high strength, to chaos, which undergoes several transitions. Figure 11 shows a graph of number of transients as a function of α. In Figure 11, there is a peak at $\alpha=1.4$ and a decrease to $\alpha=1.2$. This observation leads to the hypothesis that there are specific conditions under which the number of transients before an attractor is reached is low, leading to fast responses and/or fast recognition.

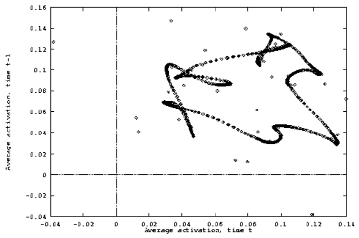


Figure 10. Results of applying an external pattern to the network

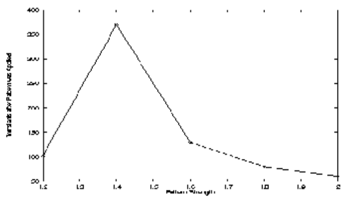


Figure 11. Observed transients as function of pattern strengths

IV. DYNAMIC ADJUSTMENT OF PATTERN STRENGTH

Our goal was to produce a pattern-to-oscillation map in which the evoked oscillation satisfies a criterion that indicated it is a relatively simple oscillation. We define a Criterion A as follows.

Criterion A:

-- A repeating sequence of states, e.g., a finite n-state oscillation ($n>1$).

OR

-- A limit cycle (quasi-periodic).

In Figure 7, most of the evoked attractors satisfied this criterion. Several did not, and remained chaotic in appearance (Figures 7(l)(m)(o)(p)(q)(r)). It is also possible to evoke a fixed point (Figure 7(n)). Here there is no *a priori* knowledge of the best choice for α. To produce the map, a chaotic network is used. Thus, the initial parameters are (1) the network weights, randomly assigned, and (2) the connection configuration. A value of g_c for the multiplier in (1) was set just above the transition to chaotic behavior. A set of external patterns is then applied to the network, one at a time. Initially, a value for the pattern strength α_b is chosen. External patterns are then applied, each with strength α_b, as in Equation (3), with results similar to Figure 7.

The value of α_b can be chosen so that a set of random test patterns usually evokes oscillations that meet Criterion A above. Any evoked oscillation that fits criterion A is considered to be the result of the pattern-to-oscillation map.

For patterns that evoke oscillations that do not fit Criterion A, we do not yet have a suitable oscillation for the pattern-to-oscillation map. So, a procedure is used to adjust the pattern strength until an oscillation that fits Criterion A is evoked. In the end, each pattern has its own strength parameter, and each oscillation evoked fits criterion A.

The adjustment procedure is constructed based on the following observations about an activity value as a function of α.

Values a, b, c, d can be set so that

1. If activity $\leq d$ then α is too high.
2. If activity $\geq c$ (chaotic), then α is too low.
3. If activity is between a and b, then a limit cycle or n-state oscillation is assumed present.

If the activity of the network is represented by the coverage value, then good cutoff values are at d=1, and approximately c=73, a=42, and b=53. If coverage value is used for the activity measurement, then Observation 3 is usually correct, and the limit cycle or n=state oscillation is assumed present.

```
1.  Choose $\alpha_{init}$. Try to choose a value so that most typical patterns evoke limit
    cycles or n-state oscillations.
2.  Set $\delta$ to a small value > 0 and << $\alpha_{init}$
3.  Set a and b so that [a,b] is the desired window for activity measurements.

For each pattern i, do the following

4.  Set $\alpha_i(1) \leftarrow \alpha_{init}$
5.  Set $i_{hi} = i_{low} = 0$
6.  Set k=0, iteration number
7.  Increment k:= k+1
8.  Measure the activity m(k) when the pattern i is applied at strength $\alpha_i(k)$. If a
    $\leq$ m(k) $\leq$ b, then done with pattern i
9.  If m(k) < a then $i_{low} = k$
10. If m(k) > b then $i_{hi} = k$
11. If $i_{low} > 0$ and $i_{hi} > 0$ then
            $\alpha_i(k+1) = \frac{\alpha_i(i_{low}) + \alpha_i(i_{hi})}{2}$
12. If m(k) > b and $i_{low} = 0$ then
            $\alpha_i(k+1) \leftarrow \alpha_i(k) + \delta$
13. If m(k) < a and $I_{hi} = 0$ then
            $\alpha_i(k+1) \leftarrow \alpha_i(k) - \delta$
14. Go to 7 for next iteration
```

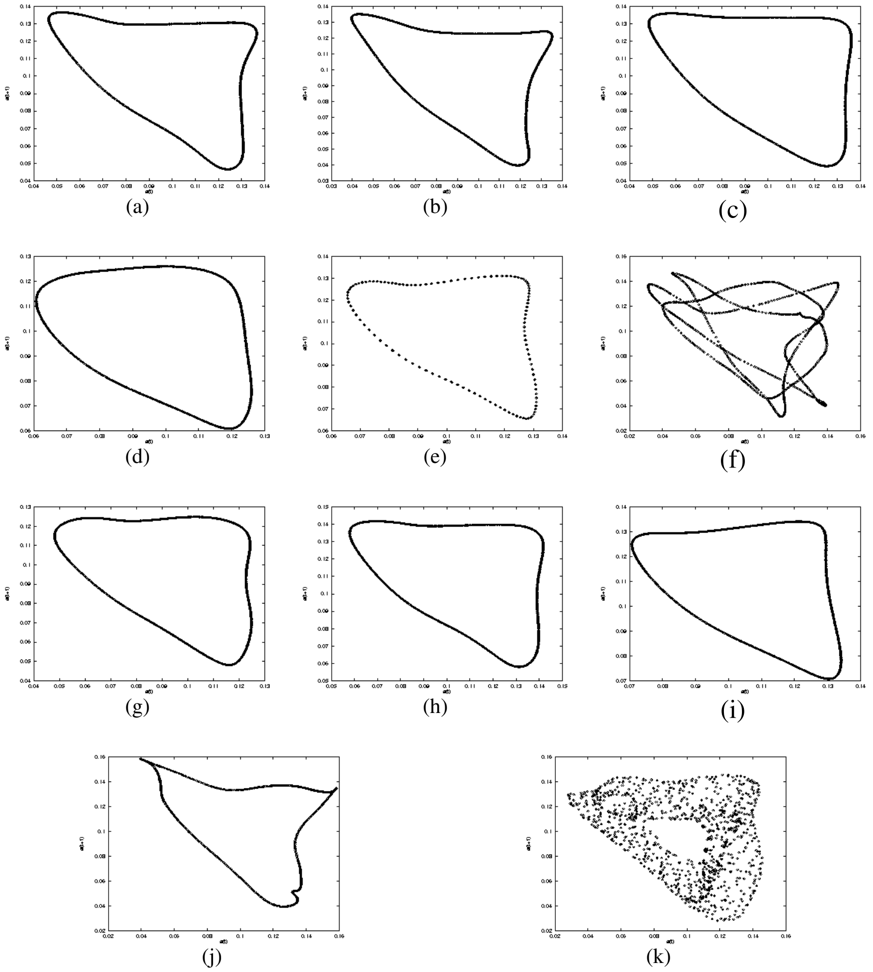
Figure 12. Strength Adjustment Algorithm

An algorithm for pattern strength adjustment is given in Figure 12. This algorithm only uses parameters a and b, and targets [a,b] as the window for desired activity. If the activity measurement is initially too large or too small, the pattern strength is changed and a new activity measurement is taken. If this does not fall into the desired range [a,b], then the amount of change is adjusted appropriately to eventually reach the desired range.

V. CHARACTERISTICS OF THE PATTERN-TO-OSCILLATION MAP

Resilience to pattern noise has been observed in oscillations evoked by external patterns. A set of specific patterns was simulated and used as base patterns. To each base pattern, ten noisy patterns were generated by adding small random numbers to each entry. The resulting oscillation was then observed, to see how the evoked oscillations compared among the noisy versions of the same base pattern.

Figure 13 shows the result for one base pattern. The base pattern was simulated at random and applied with a strength of 1.6. The same α was used for the noisy versions of the base pattern. Figure 14 shows the same results but with a different form of display, in which successive points generated by the network are connected. Figure 14 thus reflects more of the dynamic action of the network over a progression of times. These plots were generated to see whether the building of each figure was with the same type of sequence of points, and in all cases show that the figure, roughly triangular, was constructed by sampling three points over each 360 degree traversed. Figure 14(f) in which a triangular-shaped limit cycle has split into two triangles, has the same sampling pattern of three points per 360 degrees.



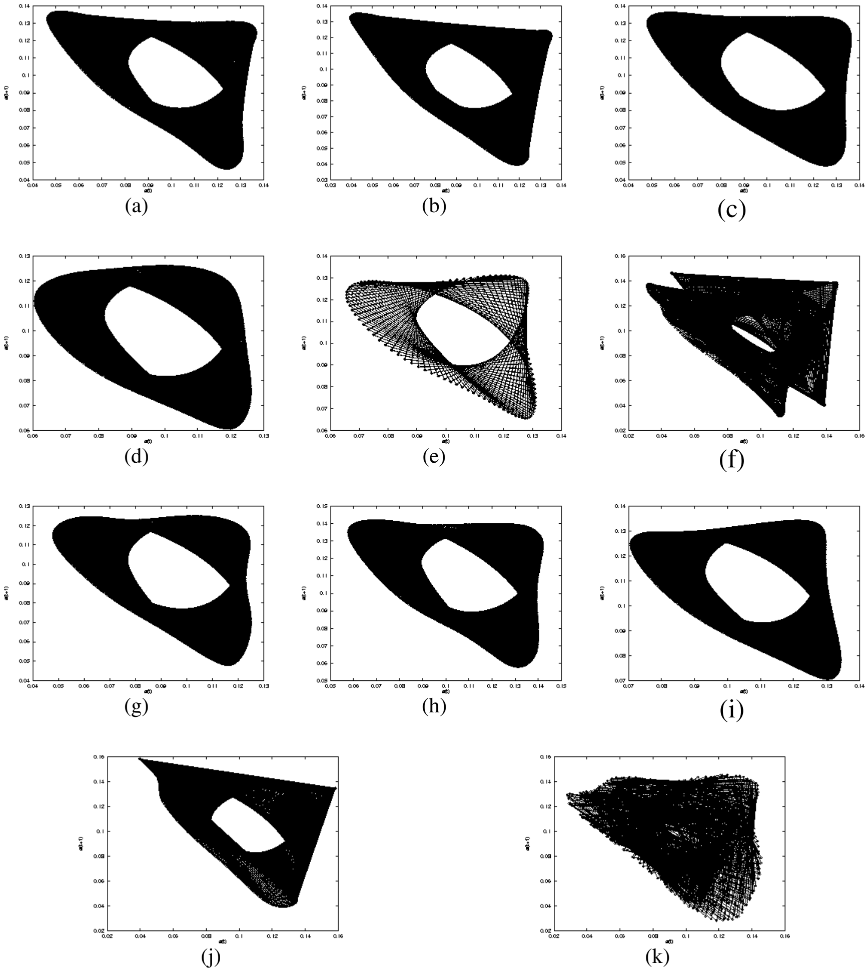
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k)

Figure 13. A chaotic network has different external patterns applied, each at strength $\alpha = 1.6$. (a) The base pattern, with 64 entries, each from a uniform random distribution (-1:1). (b-k) The base pattern with 5% noise added, to make 10 different variations of the pattern.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k)

Figure 14. A chaotic network has different external patterns applied. The map is drawn with successively generated points interconnected. (a) The base pattern, with 64 entries, each from a uniform random distribution (-1:1). (b-k) The base pattern with 5% noise added, ten times to generate ten variations of the base pattern. The pattern strength $\alpha = 1.6$, and the attractors are the same as in Figure 13.

Figure 15 shows the base pattern and its ten variations, plotted together. Successive entries of the pattern vector are marked on the horizontal axis, with values in the vertical axis. The pattern entries were generated at random from a [-1,1] distribution, and the added noise was generated from a uniform random distribution [-0.05,0.05], adding 5% noise.

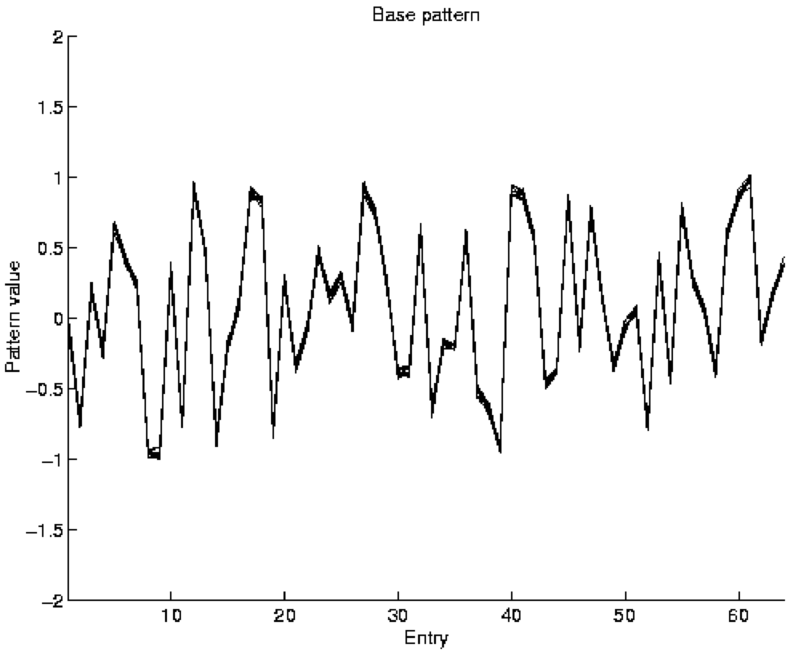
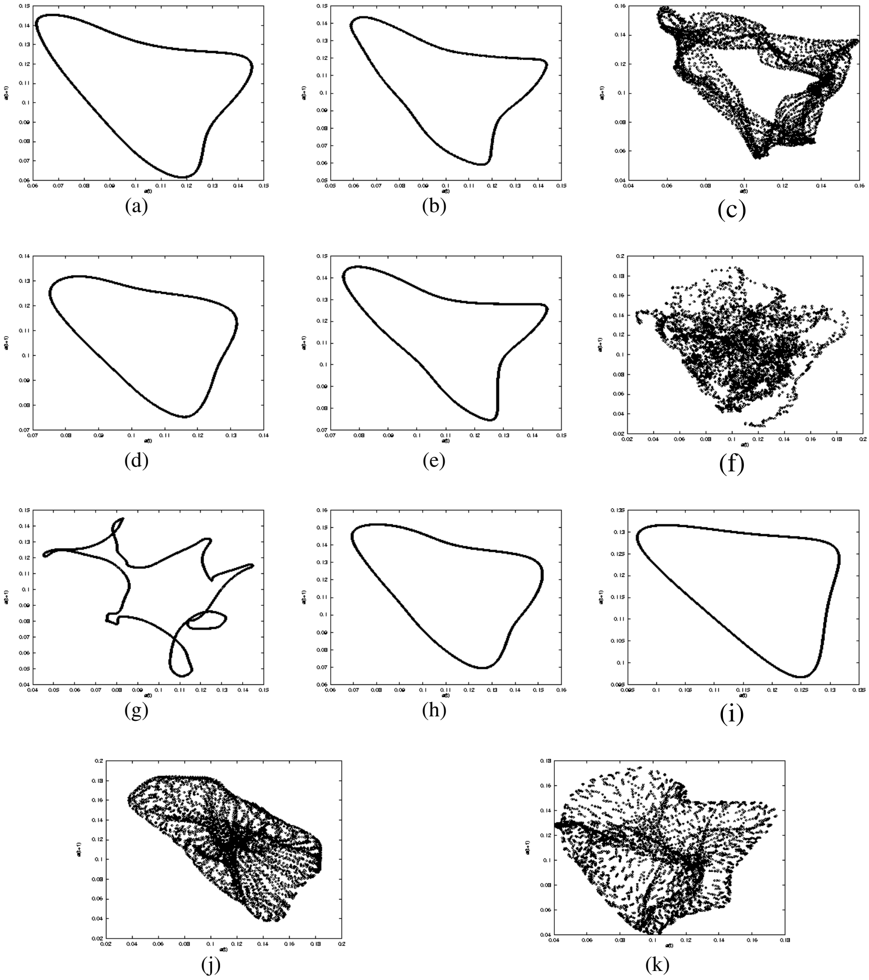


Figure 15. The base pattern and its ten variations, plotted together.

Figures 16, 17, and 18 show applications of the same base pattern as in Figures 13 and 14, with the same ten variations, to the same network, with varied pattern strengths. In Figure 16, a pattern strength of 1.4 was used, which is weaker than that in Figure 13. There is an increase in the variation among the evoked attractors in Figure 16 compared with Figure 13.

In Figure 17, a pattern strength of 1.8 was used, which is stronger than that in Figure 13. The variation in the evoked attractors was less. In Figure 18, a considerably weaker strength (1.2) was used, resulting in large amounts of variation. These preliminary results suggest that different values for the pattern strength change the map's resilience to noise, and that the oscillations show less variation at a stronger pattern strength than at weaker pattern strengths. For these figures (Figures 13-14, 16-18), the pattern strength evoked a limit cycle which was the result of the last bifurcation from the final fixed point, as in Figure 8(h).



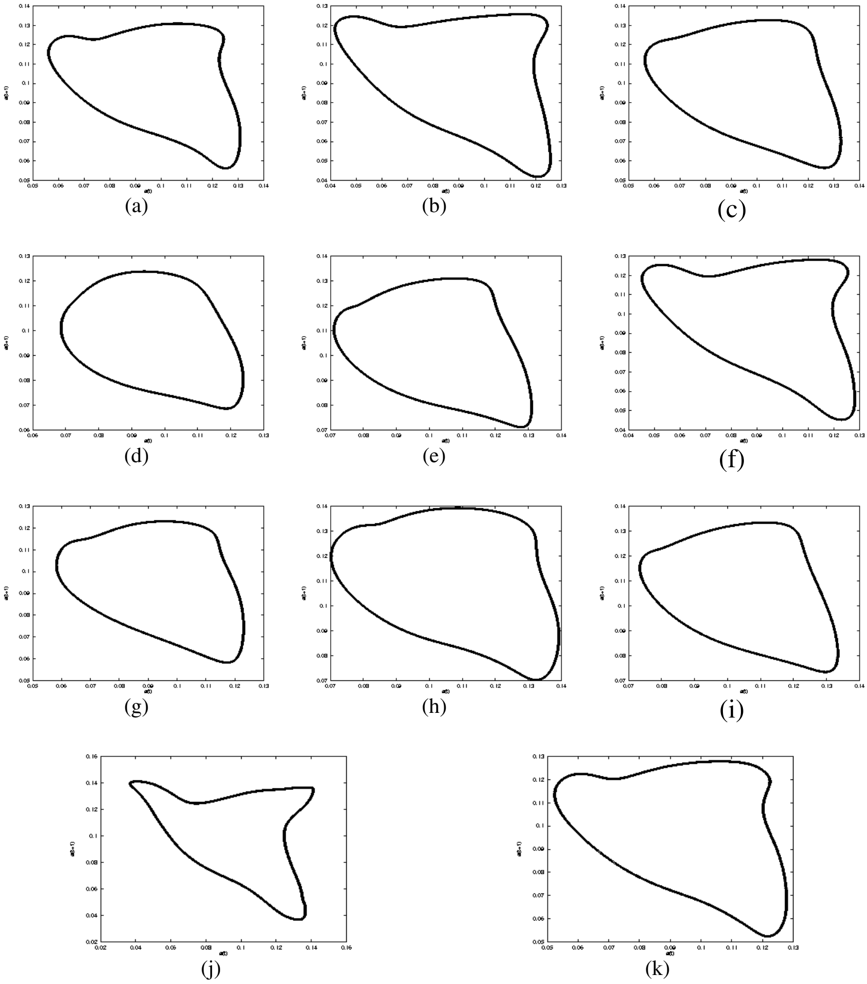
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k)

Figure 16. The base pattern (a) and its ten variations (b-k). Here the pattern strength α was 1.4, a decrease compared with Figure 13.



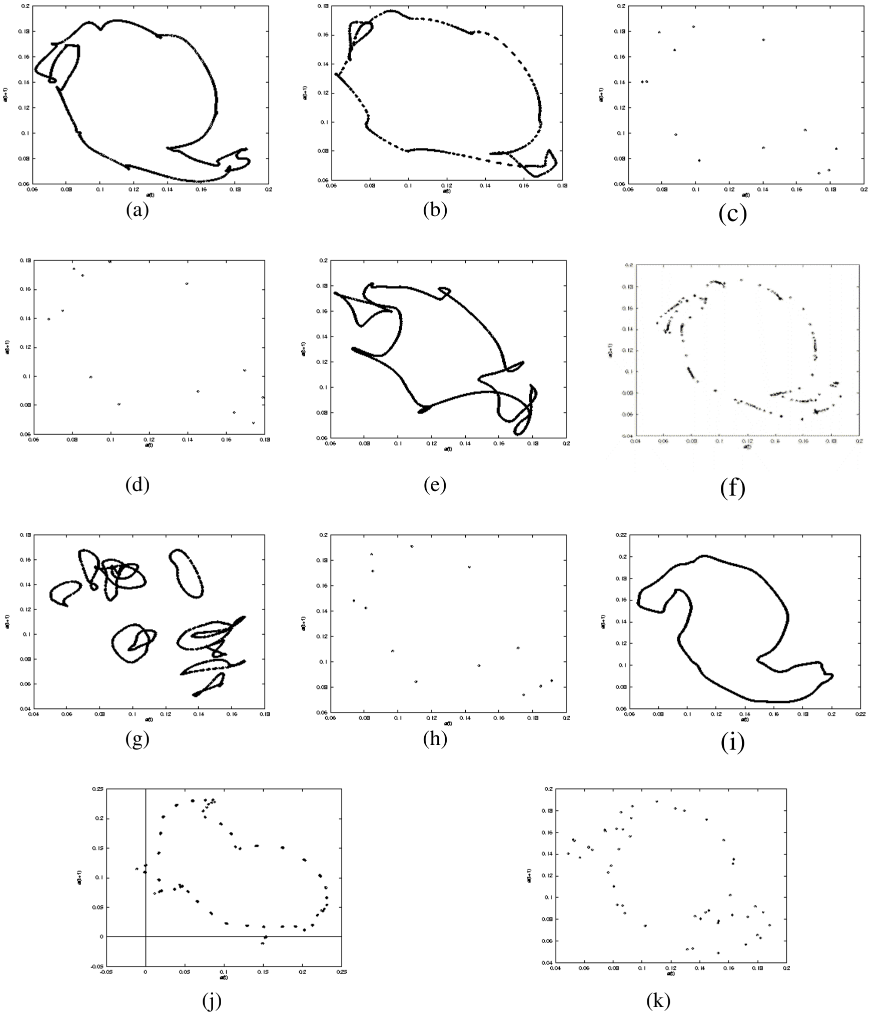
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k)

Figure 17. The base pattern (a) and its ten variations (b-k). Here the patterns strength α was 1.8, an increase compared with Figure 14. The variations between the evoked attractors is less compared with Figure 14, where α was 1.6.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k)

Figure 18. The base pattern (a) and its ten variations (b-k). Here the pattern strength α was 1.2, a large decrease compared with Figure 14. The variations between the evoked attractors is considerably higher compared with Figure 14, where α was 1.6.

VI. DISCUSSION

The use of an evoked oscillation when an external pattern is applied to a chaotic network forms the groundwork for a potentially powerful approach to associative memory, recognition, and other potential computational applications. A wide range of different pattern inputs can evoke unique dynamic attractors, and the entry into the attractor often occurs quite rapidly. Thus there is the potential for performing useful computations, such as pattern classification, where the attractor that is evoked would represent the pattern class, associated memory, or other computational result.

Capacity is high with the model shown here, in the sense that many different patterns can each evoke a different attractor. Whereas the Hopfield network showed a capacity of about 0.15n attractors (memories) for n neurons, the number of oscillation attractors that could be used in our approach far exceeds n.

A previous study has applied a Hebbian-like learning approach to reduce the dynamics from chaos to quasi-periodic attractors in the presence of an external pattern. The network's chaos is reduced during the learning of a pattern to gain a specific response of a limit cycle to the pattern [Quoy *et al.*, 1995]. Capacity, however, was found to be seriously limited in this study, although changes in neural architecture are hoped to increase capacity. We propose here that the adjustment of α, along with Hebbian-like training, could possibly increase the capacity of such a training paradigm. Higher pattern strength α tends to evoke simpler periodic or quasi-periodic attractors. Noise resilience is demonstrated through Figures 13-18. The same pattern, with added noise, evokes similar attractors. Higher values of pattern strength α can cause the attractors to be even more similar, whereas lower values of pattern strength tend to disperse the similarities. Quantitative analysis of attractor similarity has been introduced, but the graphical presentation shown here (Figures 13-14, 16-18) was chosen to show a more complete comparison of the attractors that are evoked.

Recognition of attractors is a problem for which feedforward time-delay networks would be naturally appropriate. Previous study has shown that different closed trajectories can be learned, recognized and generated by TDNNs and ATNNs [Lin *et al.*, 1995]. Generalization capabilities would reflect the ability of the network to recognize the boundary between one class of applied pattern and another, when new external patterns are applied. This could be tested in a combined system with a TDNN or ATNN recognizer as a post-processor.

Previous investigations have addressed the activity of the chaotic networks with a mean theory approach [Sompolinsky *et al.*, 1988]. The result of using a fixed external pattern (a "bias") with a fixed strength, and varying the multiplier g, has been analyzed [Cessac, 1994a,b]. It is found that the distance from the highest g that evokes a fixed point to the lowest g that evokes chaos is surprisingly small, and diminishes as the number of neurons n grows, with fully connected random networks. The evolution of the neurons is a white noise in the thermodynamic limit [Cessac, 1995].

Cessac [Cessac, 1994a] shows an absolute stability criterion for discrete time neural networks. In the thermodynamic limit, critical lines were found to divide planar graphs of the average bias of the external pattern versus g into areas with one fixed point, two stable points, one fixed and one strange attractor, and one strange attractor. In finite size systems, the line of destabilization and the appearance of chaos are not identical: there is an intermediate periodic and quasi-periodic regime corresponding to the quasi-periodic route to chaos observed in finite neural networks. Doyon and colleagues [Doyon *et al.*, 1993] show a quasi-periodic route to chaos in a class of finite random networks. We have illustrated one route for a 64-neuron network in this chapter (Figure 3).

Research on dynamic neural networks is highly motivated because there is potential for gaining superior generalization and pattern classification. Dynamical systems can have complex basin boundaries, including fractal boundaries [Ott, 1993], and the dynamic neural networks presented here are examples of such dynamic systems. Fractal basin boundaries and other complicated boundaries could enable a recognition scheme to place arbitrarily complex boundaries between different pattern classes, with a tremendous amount of fine structure. The tailoring of basin boundaries to fit the solution of a recognition problem has so far been considered only for limited cases with a different neural network paradigm [Hao *et al.*, 1994, Venkatesh *et al.*, 1990, Dayhoff and Palmadesso, 1998] and fractal boundaries were not taken into account.

Dynamic neural networks have an extensive armamentarium of behaviors, including dynamic attractors - finite-state oscillations, limit cycles, and chaos - as well as fixed-point attractors (stable states) and the transients that arise between attractor states. In our experiments, this tremendous spectrum of differing activities develops naturally as a result of the network's processing units, asymmetric weights, and closed-loop interconnections. Components that could oscillate or individually produce chaos did not have to be built into the network to insure the presence of dynamics. The resulting networks are enormously flexible, capable of prolonged self-sustained activity, and able to undergo progressions of oscillations that are controlled by modulating parameters and applied patterns.

Although the neural architectures studied here are artificial, they are inspired by biological structures. Furthermore, the ability for self-sustained activity is clear in biology, as neurons in the brain have recorded "spontaneous" activity, and animals can maintain ongoing awareness, consciousness, and mental activity. The extent to which a capability for self-sustained activity and changing oscillations contributes to these biological and behavioral abilities is as yet unknown. We propose that underlying oscillations, changes in oscillation complexity, and modulated progressions of oscillations may contribute to biological activities such as awareness, mental transitions, mental representations, consciousness, and high-level tasks. If so, this chapter shows a simplified and abstracted model that represents such neural oscillations, their modulation by externally applied patterns, and progressions between simple fixed states, more complex oscillations, and chaos.

ACKNOWLEDGMENTS

J. Dayhoff was supported by the Naval Research Laboratory (N00014-93-K-2019), the National Science Foundation (CDR-88-03012 and BIR9309169), the AFOSR Summer Faculty Research Program, AFOSR SREP contract 97-0850, ITT contract 1400256, and the Office of Naval Research. P. Palmadesso and F. Richards acknowledge support from the Office of Naval Research.

REFERENCES

Cessac, B., Doyon, B., Quoy, M., Samuelides, M., Mean-field equations, bifurcation map and chaos in discrete time neural networks, *Physica D*, 74, 24, 1994.

Cessac, B., Absolute stability criterion for discrete time neural networks, *Journal of Physics A*, 27, L297, 1994.

Cessac, B., Occurrence of chaos and at line in random neural networks, *Europhysics Letters*, 26(8), 577, 1994.

Cessac, B., Increase in complexity in random neural networks, *Journal de Physique I*, 5, 409, 1995.

Cessac, B., Quoy, M., Influence of learning in chaotic neural networks, Technical Report 713/12/95, Universitat Bielefeld Postfach 10 01 31 33501 Bielefeld, BiBos at PHYSIK.UNI-Biefeld.DE, 1995.

Chapeau-Blondeau, F., *Chaos Solitons Fractals*, 3(2), 133, 1993.

Dayhoff, J. E., Palmadesso, P. J., Capacity for basin flexibility in dynamic binary networks, *Proceedings of World Congress on Neural Networks* (WCNN), 1, 365, 1995.

Dayhoff, J. E., Palmadesso, P. J., Richards, F., Lin, D.-T., Patterns of dynamic activity and timing in neural network processing, *Neural Networks and Pattern Recognition*, Omidvar and Dayhoff, Eds., Academic Press, Boston, 1998.

Dmitirev, A. S., Kuminov, D., Chaotic scanning and recognition of images in neural-like systems with learning, *J. Commun. Technol. Electron.*, 39(8), 118, 1994.

Dmitriev, A. S. et al, The simple neural-like system with chaos, *Proc. Int. Conf. Noise in Physical Syst. and 1/f Fluctuations*, Kyoto, Japan, 501, 1993.

Doyon, B., Cessac, B., Quoy, M., Samuelides, M., Control of the transition of chaos in neural networks with random connectivity, *International Journal of Bifurcation and Chaos*, 3(2), 279, 1993.

Doyon, B., Cessac, B., Quoy, M., Samuelides, M., On bifurcation and chaos in random neural networks, *Acta Biotheoretica*, 42, 215, 1994.

Freeman, W. J., Yao, Y., Burke, B., Central pattern generating and recognizing in olfactory bulb: a correlation learning rule, *Neural Networks*, 1, 277, 1988.

Hao, J., Tan, S., Vandewalle, J., A new approach to the design of discrete Hopfield associative memories, *Journal of Artificial Neural Networks*, 1(2), 247, 1994.

Hjelmfelt, A., Ross, J., Pattern Recognition, chaos, and multiplicity in neural networks and excitable system, *Proc. Nat. Academy Sci.*, USA, 91, 63, 1994.

Kumagai, T., Hashimoto, R., Wada, M., Learning of limit cycles in discrete-time neural network, *Neurocomputing*, 13, 1, 1996.

Lin, D.-T., Dayhoff, J. E., Ligomenides, P. A., Trajectory production with the adaptive time-delay neural network, *Neural Networks*, 8, (3): 447-461, 1995.

Moreira, J. E., Auto, D. M., Intermittency in a neural network with variable threshold, *Europhys. Lett.*, 21(6), 639, 1993.

Ott, E., *Chaos in Dynamical Systems*, Cambridge University Press, Cambridge, 1993.

Palmadesso, P. J., Dayhoff, J. E., Attractor locking in a chaotic neural network: stimulus patterns evoke limit cycles*, Proceedings of the World Congress on Neural Networks (WCNN)*, 1:254-257, 1995.

Quoy, M., Doyon, B., Samuelides, M., Dimension reduction by learning in a discrete time chaotic neural network, *Proceedings of World Congress on Neural Networks* (WCNN), 1:300-303, 1995.

Samuelides, M., Doyon, B., Cessac, B., Quoy, M., Spontaneous dynamics and associative learning in an asymmetric recurrent random neural network, *Annals of Mathematics and Artificial Intelligence*.

Sompolinsky, H., Crisanti, A., Sommers, H. J., Chaos in random neural networks, *Phys. Rev. Let.*, 61(3), 259, 1988.

Venkatesh, S. S., Pancha, G., Psaltis, D., Sirat, G., Shaping attraction basins in neural networks, *Neural Networks*, 3:613-623, 1990.

Wang, L., Oscillatory and chaotic dynamics in neural networks under varying operating conditions, *IEEE Trans. on Neural Networks*, 7(6), 1382, 1996.

Yao, Y., Freeman, Model of biological pattern recognition with spatially chaotic dynamics, *Neural Networks*, 3(2), 153,1990.

Yao, Y., Freeman, W. J., Burke, B., Yang, Q., Pattern recognition by a distributed neural network: an industrial application, *Neural Networks*, 4, 103, 1991.