PFC/JA-96-35

Fast Numerical Integration of Relaxation Oscillator Networks Based on Singular Limit Solutions

Linsay, Paul S.; Wang*, DeLiang

October 1996

Plasma Fusion Center
Massachusetts Institute of Technology
Cambridge, MA 02139

*Department of Computer and Information Science and Center for Cognitive Science, The Ohio State University, Columbus, OH 43210-1277

This work was supported in part by the Office of Naval Research. Reproduction, translation, publication, use and disposal, in whole or part, by or for the United States government is permitted.

Submitted for publication in: Physica D.

Fast Numerical Integration of Relaxation Oscillator Networks Based on Singular Limit Solutions

Paul S. Linsay
Plasma Fusion Center, NW17-225
Massachusetts Institute of Technology
Cambridge, MA 02139

DeLiang Wang
Department of Computer and Information Science and Center for Cognitive Science
The Ohio State University
Columbus, OH 43210-1277

Abstract

Relaxation oscillations exhibiting more than one time scale arise naturally from many physical systems. This paper proposes a method to numerically integrate large systems of relaxation oscillators. The numerical technique, called the singular limit method, is derived from analysis of relaxation oscillations in the singular limit. In such limit, system evolution gives rise to time instants at which fast dynamics takes place and intervals between them during which slow dynamics takes place. A full description of the method is given for LEGION (locally excitatory globally inhibitory oscillator networks), where fast dynamics, characterized by jumping which leads to dramatic phase shifts, is captured in this method by iterative operation and slow dynamics is entirely solved. The singular limit method is evaluated by computer experiments, and it produces remarkable speedup compared to other methods of integrating these systems. The speedup makes it possible to simulate large-scale oscillator networks.

PACS numbers: 02.60.Cb, 84.35, 87.10

Keywords: neural networks, relaxation oscillators, LEGION, singular limit method, numerical integration.

1. Introduction

Relaxation oscillations comprise a large class of nonlinear dynamical systems, and arise naturally from many physical systems such as mechanics, biology, and engineering [6] [23]. Such oscillations are characterized by intervals of time during which little happens, followed by short intervals of time during which considerable changes take place. In other words, a relaxation oscillation system exhibits more than one time scale. Among the most well-known is the van der Pol oscillator [22] [6], which admits two time scales when a parameter of the system is chosen to be very large. The periodic trajectory of the van der Pol oscillator is composed of four pieces, two slow ones interleaving with two fast ones.

Neurophysiological experiments have revealed that neural oscillations exist in the visual cortex and other brain areas [3] [7] [17]. The experimental findings can be summarized as the following: (1) Neural oscillations are triggered by appropriate sensory stimulation, and thus the oscillations are stimulus-dependent; (2) Long-range synchrony with zero phase-lag occurs if the stimuli appear to form a coherent object; (3) No synchronization occurs if the stimuli appear to be unrelated. These intriguing data are consistent with the *temporal correlation* theory [11] [24] [25], which states that in perceiving a coherent object the brain links various feature detecting neurons via temporal correlation among the firing activities of these neurons. A natural implementation of the *temporal correlation* theory is to use neural oscillators, whereby each oscillator represents some feature of the object, such as a pixel. This special form of temporal correlation is called *oscillatory correlation* [28] [21], whereby each object is represented by synchronization of the oscillator group corresponding to the object and different objects in a scene or image are represented by different oscillator groups which are desynchronized from each other. Since the discovery of coherent oscillations in the brain, neural oscillations and synchronization of oscillator networks have been extensively studied. A variety of models (see [21] for many references) are proposed to simulate biological data as well as to explore oscillatory correlation as an engineering approach to attack the problem of perceptual organization and image analysis.

As observed by Sporns et al. [20] and Wang [26], in order for oscillatory correlation to be effective for achieving visual scene analysis, it is critically important that synchronization is achieved with local coupling only, because a globally (all-to-all) connected network does not reveal the geometrical relation among sensory features, which is essential for visual perception. While most of the models proposed so far rely on global coupling to achieve synchronization, recent studies have shown that relaxation oscillators can achieve rapid synchronization among an oscillator population with just local excitatory coupling [18] [28] [21]. In particular, Terman and Wang [21] have proven that global synchronization with local coupling is a robust property of relaxation oscillator networks. Additionally, by using a global inhibitory mechanism, such networks are shown to be capable of rapid desynchronization among different oscillator groups. The network architecture thus formed is referred to as LEGION (Locally Excitatory Globally Inhibitory Oscillator Networks) [28]. To our knowledge, LEGION is the only oscillator network that has be rigorously shown to be capable of both rapid synchronization and desynchronization. Thus, LEGION provides an elegant computational mechanism for the oscillatory correlation theory.

Although the rate of synchrony and desynchrony in LEGION is high in terms of oscillation cycles, numerical simulations are still very expensive when dealing with real images typically having 256x256 pixels or more. When pixels map to oscillators in one-to-one correspondence, analyzing an image typically entails integration of hundreds of thousands of differential equations. The nature of the relaxation oscillators used in LEGION complicates the situation: integration steps can be chosen relatively large to speed up integration during time periods when little change occurs in the system, but integration steps must be small during short periods when large changes happen quickly. From the numerical analysis point of view, a LEGION network is a stiff set of equations [1] [15]. A natural way of dealing with this kind of stiffness is to use various techniques of adaptive steps [15]. Although the use of adaptive steps can speed up integration considerably, it is still by far not enough to deal with systems of sizes of real images.

In this paper, we propose a numerical method to integrate large systems of relaxation oscillators, in particular LEGION networks. The central idea is to solve the system in the singular limit when the system evolves on a slow time scale, and approximate the system when it evolves on a fast time scale. This is possible because the system, in the singular limit, naturally exhibits instants of fast dynamics that divide time into intervals of slow dynamics. We note that analytical results on LEGION are established in the singular limit [21], thus our method, called the *singular limit method*, well approximates the dynamics of the relaxation oscillator networks. The singular limit method results in a great deal of speedup compared to traditional methods integrating these systems.

In Sect. 2, we provide the definition of a LEGION network, and Sect. 3 describes the singular limit method. Computer experiments and comparative evaluations are given in Sect. 4, and some discussions are provided in Sect. 5. Though our analysis focuses on the LEGION networks, similar analysis may be applied to other networks of relaxation oscillators.

2. LEGION Network

Our following description of a LEGION network closely follows Wang and Terman [29], which is an extension of the model of Terman and Wang [21]. Each oscillator i in a LEGION network is defined as a feedback loop between an excitatory variable x_i and an inhibitory variable y_i:

$$\dot{x}_i = 3x_i - x_i^3 + 2 - y_i + I_i H(p_i - \theta) + S_i + \rho \tag{1a}$$

$$\dot{y}_i = \varepsilon(\gamma(1 + \tanh(x_i/\beta)) - y_i) \tag{1b}$$

Here I_i represent external stimulation to the oscillator, and H stands for the Heaviside function, defined as $H(v) = 1$ if $v \geq 0$ and $H(v) = 0$ if $v < 0$. The term S_i denotes the overall input from other oscillators in the network, and ρ denotes the amplitude of Gaussian noise. The mean of the noise term is set to $-\rho$, which is used to reduce the chance of self-generating oscillations; this will become clear in the paragraph below. The primary purpose of including noise is to segregate different input patterns.

The parameter ε is chosen to be a small positive number. Thus, if coupling and noise are ignored and I is a constant, (1) defines a typical relaxation oscillator, similar to the van der Pol oscillator. The x-nullcline of (1) is a cubic function and the y-nullcline is a sigmoid function. If $I > 0$ and $H = 1$, these nullclines intersect only at a point along the middle branch of the cubic when β is small. In this case, the oscillator produces a stable periodic orbit for all sufficiently small values of ε, and is referred to as *enabled* (see Fig. 1A). The periodic solution alternates between *silent* and *active* phases of near steady-state behavior. As shown in Fig. 1A, the silent and the active phases correspond to the left branch (LB) and the right branch (RB) of the cubic, respectively. Compared to motion within each phase, the transition between the two phases takes place rapidly (thus called *jumping*). The parameter γ determines relative times that the periodic solution spends in these two phases. A larger γresults in a relatively shorter time in the active phase. If $I \bullet H < 0$, the two nullclines of (1) intersect at a stable fixed point along the left branch of the cubic (see Fig. 1B). In this case no oscillation occurs, and the oscillator is called *excitable*. An excitable oscillator does not oscillate but can be induced to oscillate by stimulation. An oscillator is *stimulated* if $I > 0$, and *unstimulated* if $I \leq 0$. Because of this dependency on I, the oscillations are stimulus-dependent. The parameter β determines the steepness of the sigmoid, and is always chosen to be small so that the sigmoid is close to a binary function (see Fig. 1).

The variable p_i, referred to as the *lateral potential* of the oscillator i, is defined as:

$$\dot{p}_i = \lambda\,(1 - p_i)\,H[\sum_{k \in N(i)} T_{ik}\, H(x_k - \theta_x) - \theta_p] - \mu\,\varepsilon\, p_i \qquad (2)$$

where $\lambda > 0$ is a parameter, T_{ik} is the permanent connection weight (explained later) from oscillator k to i, and $N(i)$ is a set of oscillators called the *neighborhood* of i. Both θ_p and θ_x are thresholds, and θ_x is chosen between LB and RB. If the weighted sum the oscillator i receives from $N(i)$ exceeds θ_p, p_i approaches 1 on a (fast) time scale determined by λ which is assumed to be $O(1)$.

If this weighted sum is below θ_p, p_i relaxes to 0 on a slow time scale determined by ε, with μ being $O(1)$. With p_i initialized to 1, it follows that p_i will drop below the threshold θ in (1a) unless i is able to receive a large enough lateral excitation from its neighborhood. Note that in order to maintain a high potential a large number of neighbors of i must all exceed θ_x at the same time in their oscillations.

The motivation behind the lateral potential is to remove noisy fragments on an image. Through lateral interactions in (2) the oscillators that can maintain high potentials are those that lie at the center of an oscillator block corresponding to an object. These oscillators are called *leaders*. If an object, being a noisy fragment, is too small no oscillator in its corresponding block can become a leader, and the whole block will stop oscillating after a beginning period. This is because the Heaviside function in (1a) will become 0 unless the potential is maintained high.

A typical LEGION network for segmentation is two dimensional. The simplest case is a 2-D grid, shown in Fig. 2, where each oscillator is connected only with its four immediate neighbors except on the boundaries where no wrap-around is used. Generally speaking $N(i)$ should be larger, however, and when modeling a neuronal network T_{ij} should take on the form of a Gaussian distribution with the distance between i and j. The coupling term S_i in (1) is given by

$$S_i = \sum_{k \in N(i)} W_{ik}\, H(x_k - \theta_x) - W_z\, H(z - \theta_{xz}) \qquad (3)$$

where W_{ik} is the *dynamic* connection weight from k to i. The summation neighborhood in (3) may be chosen differently from that in (2), but the same choice suffices in this paper.

The dynamic weights W_{ik}'s are formed on the basis of permanent weights T_{ik}'s according to the mechanism of dynamic normalization [27] [29]. Dynamic normalization ensures that each oscillator has equal overall weights of dynamic connections, W_T, from its neighborhood. We note that weight normalization is not a necessary condition for the LEGION dynamics to work [21], but it improves the quality of synchronization. Moreover, based on external stimulation W_{ik} can be properly determined in one step at the beginning.

Finally, W_z in (3) is the weight of inhibition from the global inhibitor z, defined as

$$\dot{z} = \phi\,(\sigma_\infty - z) \tag{4}$$

where ϕ is a parameter. The quantity $\sigma_\infty = 1$ if $x_i \geq \theta_{zx}$ for at least one oscillator i, and $\sigma_\infty = 0$ otherwise. Hence θ_{zx} represents a threshold. If σ_∞ equals 1, $z \rightarrow 1$.

The lateral potential makes it possible to distinguish three types of stimulated oscillators: leaders as discussed above, *followers*, and *loners*. Followers are those oscillators that can be recruited to jump up by leaders in the same block. Loners are those oscillators which belong to the noisy fragments. It is clear that loners will not be able to jump up beyond an initial period, because they can neither develop into leaders and thus jump up by themselves, nor be recruited to jump up because they are not in an oscillator block corresponding to a major image region. The collection of all noisy regions corresponding to loners is called the *background*, which is generally discontiguous and not uniform.

Wang and Terman [29] have proven a number of rigorous results regarding the system (1)-(4). These analytical results together imply that loners will no longer be able to oscillate after an initial time period and the asymptotic behavior of a leader or a follower is precisely the same as the network obtained by simply eliminating all the loners. Thus similar analysis in Terman and Wang [21] applies, and implies that after a number of oscillation cycles a block of oscillators corresponding to a single major image region will oscillate in synchrony, while any two oscillator blocks corresponding to two major regions will desynchronize from each other. This behavior is established for a robust range of parameters. Regarding the speed of computation, in the singular limit the number of cycles required for full segmentation is no greater than the number of major regions plus one.

3. Singular Limit Method

For static external stimulation I, the system (1)-(4) can be reduced to an iterated map. This follows from the observation that for typical system parameters the equations of motion simplify and no actual integration is required in the singular limit $\varepsilon \to 0$. The analysis of the system (1)-(4) has been fully carried out in the singular limit [21] [29]. Thus the analytical statements about the system are established when the system is taken to the singular limit.

Consider a typical oscillator in the singular limit, and ignore the noise term in (1a) for the moment. Let $I_T = I \bullet H(p - \theta) + S$, where I_T is the total input to the oscillator. The x-nullcline has left and right knees at LK = $(-1, I_T)$ and RK = $(1, I_T+4)$, respectively. As shown in Fig. 3, when $I_T > 0$ the limit cycle trajectory consists of four pieces: two slow pieces lying on LB or RB and two fast pieces connecting LB and RB. The slow pieces are obtained by introducing a slow time scale $t' = \varepsilon t$ and setting $\varepsilon = 0$ in (1):

$$0 = 3x - x^3 + 2 - y + I_T \tag{5a}$$

$$\dot{y} = \gamma(1 + \tanh(x/\beta)) - y \tag{5b}$$

Thus the two slow pieces follow the x-nullcline. Motion on the right and left branches is constrained to the intervals $x = [1,2]$ and $x = [-2, -1]$, respectively. With a small value of β as required in the analysis [21] [29], the tanh term in (1b) is approximately either 1 or -1, i.e., it can be treated as a bipolar value. As a consequence the equations of motion for the oscillator in the slow system reduce to

$$\dot{y} = 2\gamma - y \qquad \text{on RB} \tag{6a}$$

$$\dot{y} = -y \qquad \text{on LB} \tag{6b}$$

The solutions to these equations are trivial.

The fast pieces are obtained simply by setting $\varepsilon = 0$ in (1):

$$\dot{x} = 3x - x^3 + 2 - y + I_T \tag{7a}$$

$$\dot{y} = 0 \tag{7b}$$

Relative to motion on the two branches, the two fast pieces correspond to instantaneous jumps from LK to a right corner of the limit cycle at RC = $(2, I_T)$ or from RK to a left corner at LC = $(-2, I_T + 4)$, as illustrated in Fig. 3 (cf. Fig. 1A).

To sum, the trajectory of the oscillator in the singular limit reduces to motion determined by (6) when the oscillator is in either the active phase or the silent phase, and instantaneous jumping between the two phases. Since within either phase $\dot{x} = 0$, motion is entirely determined by the slow variable y, defined in (6). There is no need to solve the equations of (1). Given the value of y and the branch of the nullcline, it is straightforward to compute x if its value is needed (see later discussions regarding displaying system output).

For the LEGION network, we observe that the coupling term S_i does not change between jumping instants, while it changes at jumping instants. In (3), θ_x is chosen between the two outer branches of the cubic [21] [29], so that $H(x_k - \theta_x)$ is 1 if oscillator k is on RB, and 0 if k is on LB. Thus the Heaviside function does not change its value unless a jump occurs. Furthermore, the parameter ϕ is $O(1)$ so that the dynamics of z is on the fast time scale. The only possible time for z to change its value is when a jump occurs. Thus, S_i remains a constant between two consecutive jumping instants.

For the lateral potential p_i, the argument to the outer Heaviside function is a constant between jumping instants. The choices for λ and μ entail that p_i increases on the fast time scale and decreases on the slow time scale. In the fast system, (2) reduces to

$$\dot{p}_i = \lambda\,(1 - p_i)\,H[\sum_{k \in N(i)} T_{ik}\,H(x_k - \theta_x) - \theta_p] \qquad (8)$$

which approaches 1 if the outer Heaviside is 1 and remains unchanged otherwise. If $\lambda < 1$, as chosen in [29], p_i updates its value with a slightly slower rate than jumping of oscillators so that the lateral potential of an oscillator is updated after all oscillators have jumped. This condition is used here. In the slow system, (2) reduces to

$$\dot{p}_i = 0 \qquad \text{if outer } H = 1 \qquad (9a)$$

$$\dot{p}_i = -\mu\,p_i \qquad \text{if outer } H = 0 \qquad (9b)$$

Here (9a) holds because the outer H does not change its value in the slow system. Solving (9b) is trivial. It is possible that the lateral potential drops to below θ in (1a), and leads to a nullcline shift during the evolution of the slow system. But precise times of updating a potential do not matter, and we can limit such updates to when jumps occur. The only effect of doing so is to speed up or slow down the evolution of an oscillator by a fraction of an oscillator cycle.

The above analysis concludes that all the relevant information to an oscillator changes only at the times when the oscillators jump from one branch to the other. At these times, the couplings between the oscillators turn on and off, the global inhibitor updates its value, and so do lateral potentials. The time instants at which jumping occurs naturally divide time into intervals, within each of which the total input to an oscillator, I_T, remains a constant. Thus within each interval the system can be solved. To numerically solve the system, we only need to know when a jump occurs and which branch an oscillator is on in order to set the couplings correctly. In other words, we only need to compute when jumps will occur to correctly model a LEGION network. The following numerical integration is now obvious.

Singular Limit Method of LEGION

```
0. Initialization
        0.1 Set each $I_i$ according to external input, and $p_i = 1$.
        0.2 Form dynamic weight $W_{ij}$ according to permanent weight $T_{ij}$, $I_i$ and $I_j$.
        0.3 Randomly start oscillators on LB by choosing $y_i(0)$ randomly
              in $[I_i, 2\gamma + I_i]$.
        0.4 Set $z(0) = 0$.

1. For each oscillator compute the time to its closest knee (left or right), $T_i$.

2. Find the oscillator, $m$, with the shortest time, $T_{\min}$, to its knee.
        2.1 Advance each oscillator by $T_{\min}$.
        2.2 Jump $m$ to its opposite branch.

3. Jumping. Iterate until no jumping occurs in an iteration
        For each oscillator $i$ do the following:
        3.1 Update its nullcline by computing its $I_T$.
        3.2 If $i$ is beyond its updated knee, $i$ jumps to its opposite branch.

4. For each oscillator update its lateral potential.

5. Go to 1 until done.
```

Some remarks on these steps.

[Step 0.3] We note that in choosing these initial conditions it is possible to get antiphase solutions with only local excitatory coupling, as pointed out by Kopell and Somers [10]. However, the basin of attraction to antiphase solutions is much smaller than to synchronous solutions, and antiphase behavior is rarely seen in our simulations. To avoid the situation entirely, one can start all the oscillators on LB only from LK up to the lowest excited knee. This way, there will be no accidental antiphase behavior between oscillators in the same block. But to limit initial conditions in a tight zone may slow down the segmentation process because the rebound mechanism discussed in the Remark on Step 3.2 below can cause accidental synchrony temporarily.

[Step 1 & Step 2.1] There is no need to actually compute the times, which involves a logarithm. The solution to Eq. (6) is

$$y(t) = [y(0) - y_F] \exp(-t) + y_F$$

On the slow time scale $y(t)$ relaxes to y_F where $y_F = 0$ on LB and $y_F = 2\gamma$ on RB. Jumping occurs when $y(t_K) = y_K$, its value at the knee, where $y_K = I_T$ on LB and $y_K = I_T + 4$ on RB. The time to reach the knee is given by the expression,

Table 1. Oscillator State Conditions

Stable limit cycle	$\left.\begin{array}{l} y(0) > y_K > y_F,\ \text{LB} \\ y_F > y_K > y(0),\ \text{RB} \end{array}\right\} \Rightarrow v > 1$	
Stable fixed point	$\left.\begin{array}{l} y(0) > y_F > y_K,\ \text{LB} \\ y_K > y_F > y(0),\ \text{RB} \end{array}\right\} \Rightarrow v < 0$ $\left.\begin{array}{l} y_F > y(0) > y_K,\ \text{LB} \\ y_K > y(0) > y_F,\ \text{RB} \end{array}\right\} \Rightarrow 0 < v < 1$	$\Rightarrow \left\{\begin{array}{l} y_F > y_K \ \&\ y(0) > y_K,\ \text{LB} \\ y_K > y_F \ \&\ y_K > y(0),\ \text{RB} \end{array}\right.$
Jump	$\left.\begin{array}{l} y_F > y_K > y(0),\ \text{LB} \\ y(0) > y_K > y_F,\ \text{RB} \end{array}\right\} \Rightarrow v > 1$ $\left.\begin{array}{l} y_K > y(0) > y_F,\ \text{LB} \\ y_F > y(0) > y_K,\ \text{RB} \end{array}\right\} \Rightarrow 0 < v < 1$ $\left.\begin{array}{l} y_K > y_F > y(0),\ \text{LB} \\ y(0) > y_F > y_K,\ \text{RB} \end{array}\right\} \Rightarrow v < 0$	in all cases $\left\{\begin{array}{l} y_K > y(0),\ \text{LB} \\ y(0) > y_K,\ \text{RB} \end{array}\right.$

$$t_K = \ln(v) \qquad \text{and} \qquad v = \frac{y(0)-y_F}{y_K-y_F}$$

The order of $y(0)$, y_K, and y_F and the branch determine what an oscillator can do. The conditions are summarized in Table 1. The iteration of the map needs to take these state conditions into account when determining the next step.

Step 1 is determined by the limit cycle oscillators, hence $T_{\min} = \ln(v_{\min})$ for all $v > 1$.

Once we have found $T_{\min}$, then it is simple to update the trajectory of each oscillator

$$y(T_{\min}) = \frac{y(0) - y_F}{v_{\min}} + y_F$$

So, no computing of logarithms or exponentials is required. For the lateral potential,

$$p(T_{\min}) = p(0) \qquad \text{if outer } H = 1$$

$$\mathrm{p}(T_{\min}) = \frac{p(0)}{v_{\min}^{\mu}} \qquad \text{if outer } H = 0$$

The parameter θ in (1a) can be chosen carefully so that μ is set to 1. In this case, $p(T_{\min}) = p(0)/v_{\min}$. The criterion for choosing θ should be that it takes about an oscillation period to decay the value of p from 1 to below θ. See the Remark on Step 5 for how to estimate the period of oscillations. Again, no computing of exponentials is required.

Since we only track the slow variable, y, we need a binary variable for each oscillator to keep track of whether it is on LB or RB. Similarly, each oscillator has its own value of I_T which must be maintained carefully to correctly compute y_K.

[Step 2.2] This step reflects the role of noise in desynchronization. As discussed in [21] [29], the noise term in (1a) serves to desynchronizes the oscillators of different blocks that are very close to each other, thus reducing the chance of accidental synchrony. We approximate the effect of noise by jumping only the oscillator which arrives at a knee first, which is done in Step 2.2. The noise term is omitted in (5) and (7) because noise prohibits analytic solutions. Thus, Step 2.2 embodies the major effect of the noise term, even though noise is not explicitly included.

[Step 3] This iterative step is where much of the action in the LEGION network takes place. Although much is happening, it is on the fast time scale, and this step in the singular limit corresponds to a single time instant. The actual jumping of an oscillator involves reversing the binary variable that records which branch the oscillator is on, and does not change its y value. When an oscillator jumps (up or down), its input to the oscillators in its neighborhood needs to be updated. Also, its input to the global inhibitor and the global inhibitor, z, itself are updated immediately. In other words, the jumping of the oscillator will shift the cubics of its neighboring oscillators and possibly others through z. The cubic shifts may trigger further cubic shifts and jumps, and phase shifts spread out.

This step terminates eventually when in an iteration step no jumping has occurred. This termination condition is correct on the basis of the observation that I_T of an oscillator remains the same if no oscillator has jumped in the previous step, and it will remain the same in the next step, and so on. To repeat, although the step may take many iterations, it takes an instant in real time.

In the test of Step 3.2, we allow a certain error window when considering if an oscillator is beyond its knee. This will reduce unnecessary computing when oscillators are synchronized but do not have exactly the same y values due to floating point errors and so on. Also, we do not consider the "turn-around" mechanism which, as discussed in [21], facilitates the process of desynchronization. In order for turn-around to take effect, several parameters must be chosen carefully.

We note that global inhibition, in addition to serving to desynchronize, also has a role to synchronize [21] [29]. This happens when the sudden release of global inhibition raises the x-nullcline of every oscillator, and oscillators from different blocks may jump up simultaneously and thus synchronize. This mechanism of synchronization is often referred to as *rebound* [14].

[Step. 4] According to (8), p_i is set to 1 if the outer Heaviside in (8) is 1. Otherwise there is no change to p_i.

[Step. 5] A useful criterion for ending the numerical computation is the total time after n cycles of oscillation as measured by

$$\tau_n = \sum_{j=1}^{n} \ln(v_j)$$

where v_j is the total phase shift in the jth cycle. The times on LB and RB are given by (see the Remark on Steps 1 & 2.1)

$$\tau_L = \ln(\frac{I_T + 4}{I}) \qquad \tau_R = \ln(\frac{I - 2\gamma}{I_T - 2\gamma + 4})$$

respectively. Here it is assumed that a typical enabled oscillator travels between I and $I_T + 4$, where $I_T = I + W_T - W_z$. A typical set of parameters (see Sect. 4): $\gamma = 6.5$, $I = 0.2$, $W_z = 1.5$, and $W_T = 8$, gives $I_T = 6.7$, $\tau_L \approx 3.98$, $\tau_R \approx 1.72$, and a total period $\tau = 5.7$. According to Wang and Terman [29], the system (1)-(4) exhibits a segmentation capacity, C, the maximum number of segments that can be separated. The segmentation capacity roughly corresponds to

$$C = \lceil \tau / \tau_R \rceil, \qquad \text{if } \tau_L \geq \tau_R. \tag{10}$$

According to the analysis on the speed of LEGION segmentation mentioned in Sect. 2, a sensible stopping criterion is $\tau_n = (1 + C)\bullet\tau$.

The above method is derived without considering the x variable. However, the x values better indicate the phases of the oscillators. Thus, the x variable should be used to display the oscillator activity, as done in [21] [29]. Given a value of y and which branch the oscillator is on, the x value can be found by solving the following cubic function

$$3x - x^3 + 2 - y + I_T = 0$$

Let $y' = y - I_T$. After straightforward calculations (see [19]), we have for $4 \geq y' \geq 0$

$$x = \begin{cases} 2\cos(\frac{1}{3}\omega+\frac{2}{3}\pi) & \text{on LB} \\ 2\cos(\frac{1}{3}\omega) & \text{on RB} \end{cases}$$

where $\cos\omega = -(y'-2)/2$. Note that I_T has different values when the oscillator is on different branches. For $y' > 4$ or $y' < 0$, we have

$$x = \sqrt[3]{-\frac{1}{2}(y'-2)+\frac{1}{2}\sqrt{y'^2-4y'}} + \sqrt[3]{-\frac{1}{2}(y'-2)-\frac{1}{2}\sqrt{y'^2-4y'}}$$

Since x values are needed only for display purposes, some computing time can be saved when the cubic function is approximated by a piecewise linear function leading to:

$$x = \begin{cases} \frac{-y}{4} - 1 & \text{on LB} \\ \frac{-y}{4} + 2 & \text{on RB} \end{cases}$$

4. Computer Experiments

To compare with traditional numerical methods, we have conducted extensive computer experiments on the singular limit method. Recently, Wang and Terman presented simulations of a 50x50 LEGION network, as defined in Sect. 2, using the fourth-order Runge-Kutta method. The Runge-Kutta method is commonly used in simulating relaxation oscillator networks (see among others [18] [21] [5]). To have a controlled comparison, we experimented with the same network using the same input pattern as in Wang and Terman [29]. The input image to the network consists of three objects, designated as the "sun", a "tree", and a "mountain", which are mapped to a 50x50 grid shown in Fig. 4A. Each little box in Fig. 4A corresponds to an oscillator in the LEGION network. To test the utility of the lateral potential in eliminating noisy regions, Fig. 4A is corrupted with a 20% noise so that each uncovered (unstimulated) box has a 20% chance to be covered, resulting in the image of Fig. 4B. Fig. 4B is the input to the network, and is exactly the same as in [29].

The network configuration is the same as in [29]. In the simulation, $N(i)$ is simply the four nearest-neighbors with no boundary wrap-around. For a stimulated oscillator $I = 0.2$, and for an unstimulated oscillator $I = 0$. The permanent connections between any two neighboring oscillators are set to 2.0, and the total dynamic connection W_T is set to 8.0. Notice that if oscillator i is unstimulated, $W_{ik} = W_{ki} = 0$ for all k according to dynamic normalization, and i cannot oscillate. In [29] the amplitude ρ was set to 0.02, representing a 10% noise level compared to the external input. The following values are used for the other parameters in (1)-(4): $\varepsilon = 0.02$, $\beta = 0.1$, $\gamma =$ 6.5, $\theta = 0.001$, $\lambda = 0.1$, $\theta_x = -0.5$,$\theta_p = 7.0$, $\mu = 0.25$, $W_z = 1.5$, $\phi = 3.0$, and $\theta_{zx} = \theta_{xz} = 0.1$. When using the singular limit method, the following parameters are not needed: ε, β, λ, θ_x, ϕ, θ_{zx}, θ_{xz}. The rest of the parameters have the same values as in [29].

Figure 4C-4F show the instantaneous activity of the network at different times of dynamic evolution. Here, a black circle represents the x activity of an oscillator, and its diameter is proportional to $(x-x_{\min})/(x_{\max}-x_{\min})$, where $x_{\min}$ and $x_{\max}$ are the minimum and maximum x values of all the oscillators, respectively. Fig. 4C is the snapshot at the beginning of system evolution, included here to show the oscillators' random initial positions on LB. Fig. 4D is a snapshot shortly after Fig. 4C. Synchronization and desynchronization are shown clearly: all the oscillators corresponding to the "sun" and its immediate neighbors are entrained and in the active phase; all the other stimulated oscillators are in the silent phase. Thus the noisy "sun" object is separated from the rest of the image. Fig. 4E shows another snapshot shortly after Fig. 4D. Now the oscillators corresponding to the noisy "tree" are in the active phase, and desynchronized from the rest. At yet another time shown in Fig. 4F, the noisy "mountain" has its turn to be in the active phase and separates from the rest of the image. The successive "pop-out" of these three segments continue in a stable periodic manner until the input image is withdrawn.

To illustrate the role of the lateral potential, the same network has been simulated with the Heaviside term in (1a) set to 1. Typical snapshots of the network are given in Fig. 4G-4K. Fig. 4G shows the random initial positions of the oscillators on LB. After a short beginning period, the network exhibits a stable periodic behavior with four segments. These four segments are successively shown in Fig. 4H-4K. It is clear that, without the lateral potential, the network cannot distinguish between major regions and noisy fragments, and as a result, major regions generally cannot be segmented apart.

To show the entire process of synchronization and desynchronization, Fig. 5A displays temporal activity of all the stimulated oscillators for the case including the lateral potential. Unstimulated oscillators are omitted from the display because they remain excitable always. The oscillators corresponding to each noisy object are combined in the display, and thus should appear like a single oscillator when they are synchronized. The upper three traces of Fig. 5A represent the three oscillator blocks corresponding to the three objects, respectively. All the loners corresponding to the background are combined together, and their activities are shown in the fourth trace. Because of the inability to develop high potentials, these loners quickly stop oscillating even though they are enabled at the beginning. The bottom trace of Fig. 5A shows the activity of the global inhibitor. Due to the display resolution, some time instants at which the global inhibitor is inactive (i.e. $z = 0$) are not captured in the figure. These instants occur when a block jumps down while another block jumps up. From Fig. 5A, it is clear that synchronization within each block and desynchronization between different blocks are both achieved in less than two cycles.

For the purpose of comparison, Fig. 5B gives the corresponding temporal activity of the network when the fourth-order Runge-Kutta method is used (see [29]). Fig. 5A and Fig. 5B have the same duration with respect to the slow variable y, from $t = 0$ to $t = 36$. The number of cycles is comparable in these two cases. Because of $\varepsilon \to 0$, synchronization in Fig. 5A is perfect. Also it takes three cycles to separate the "sun" block from the "mountain" block in Fig. 5B, but only one cycle in Fig. 5A. This is because the former depends on noise to segregate two blocks that happen to be synchronized, and is thus not as reliable as Step 2 of our method. In a sense, the singular limit method is more faithful to mathematical analysis of LEGION [21] [29] because ε is taken to the singular limit. But it cannot be chosen too small in other methods in order to avoid exceedingly slow execution.

Table 2. Performance Comparison

	Runge-Kutta method	**Singular limit method**		
		cubic nullcline	piecewise linear	no x recording
With potential	1467	13	9	6
Without potential	1102	10	6	4.5

Table 2 provides performance comparisons between the singular limit method and the fourth-order Runge-Kutta method. The performance has been evaluated on an HP 735 workstation, and the numbers in Table 2 are computing times automatically recorded in seconds. We distinguish two cases, one including the lateral potential and the other excluding the potential. As expected, the inclusion of the potential slows computation by about more than 30%, and this is so for all cases. In our method, we further distinguish between the case where a cubic equation is solved in finding a x value for display and the one where a piecewise linear equation is solved. The piecewise approximation cuts down computing time by about 40%. Overall, the singular limit method is more than 100 times faster than the fourth-order Runge-Kutta method. When the piecewise approximation is used, our method is more than 160 times faster.

Since the recording of x values is for display purposes only, it is useful to compare the two methods when the computing time spent on generating system output is excluded from the comparison. Table 2 also lists the times for the singular limit method when no x recording is performed. The times for the Runge-Kutta method stay about the same with or without x recording, since writing display values occupies only a tiny fraction of the overall execution. The speedup in this case is 245 times. This speedup better reflects the underlying computing methods for numerical integration.

An important property of LEGION is that it exhibits a segmentation capacity. To verify whether this property holds in the same fashion, we reproduced an experiment in [29], where an image with nine major regions is presented to a 30x30 network. These nine binary patterns together form **OHIO STATE**, as shown in Fig. 6A. Then 10% noise is added to the input in the same way as in Fig. 4. The resulting input is shown in Fig. 6B, which is the same as in [29]. The parameter choices are exactly the same as in [29]: the same parameter values are used as in Fig. 4 except for $\gamma = 8.0$. The results of the simulation are given in Fig. 6C-6H, in the same format as in Fig. 4. Fig. 6C shows the snapshot when the network starts. Shortly after that, the network segments the input into five segments, which pop out alternately as shown in Fig. 6D-6H. Out of the five segments, three are *simple* (Fig. 6D-6F), each corresponding to a single block, and two are *congregate* (Fig. 6G and Fig. 6H), each corresponding to more than one block. The behavior of the network is fully consistent with that reported in [29], both yielding a segmentation capacity of 5. This capacity is consistent with the analysis in Sect. 3. Notice that Equation (10) points to ways by which the capacity can be increased by varying certain parameters, say, increasing γ. Notice also that synchrony between different blocks in a congregate segment is caused by rebound.

5. Discussion

The relaxation oscillator used in LEGION is similar to the van der Pol oscillator, and its dynamics is also similar to numerous other oscillators that have been proposed to model neuronal behavior. Exemplar models include the FitzHugh-Nagumo equations [4] [13], and the Morris-Lecar equation [12]. These models can all be viewed as simplifications of the Hodgkin-Huxley equations [9], which describe the membrane potential dynamics and spike generation of a typical neuron. We thus expect that the singular limit method proposed here can be applied to simulating a variety of relaxation oscillator networks.

The singular limit method differs from the original system (1)-(4) in three main aspects. First, the method needs a central clock to keep time, resulting in a synchronous update for all units of the network. We note, however, that synchronous update is necessary for all iterative numerical methods of integrating differential equations. Second, Step 2 in our method needs comparison among a set of values. The original version did not compare among oscillators, but instead let them evolve on the slow time scale until one reaches a jumping point first. The computation in Step 2 amounts to winner-take-all competition, which has been extensively studied in neural networks. See [2] [8] [16] among others for different versions of winner-take-all networks, all of which are parallel computation. Third, the external input to an oscillator is considered a constant between two consecutive jumping times in the singular limit method, whereas it can vary all the time in the original version. Because of this, for image segmentation our method is limited to input images that do not vary constantly.

To speed up computation, Wang and Terman have proposed an algorithm based on LEGION dynamics [29]. Their algorithm observes the jumping behavior, and keeps track of the ordering between alternating blocks. However, there is no temporal evolution in the algorithm, and no tracking of the detailed oscillator position on its trajectory. Also the right branch in a limit cycle becomes a single point, RK. As a result, the algorithm represents only a crude approximation to LEGION dynamics. In contrast, the singular limit method is firmly based on the analysis of LEGION dynamics in the singular limit $\varepsilon \rightarrow 0$. Other approximations to the original equations are all well founded, such as approximating the tanh term by a bipolar function when β is small. Thus, the singular limit method is a highly accurate numerical approximation to LEGION dynamics.

To conclude, we have presented a numerical method, the singular limit method, to integrate LEGION networks. The method is derived based on the analysis of network behavior in the singular limit. Our method is a combination of analytical solutions and iterative operations, leading to a great deal of speedup compared to commonly used integration methods. The remarkable speedup makes it possible to model large-scale LEGION networks and to analyze real images by oscillatory dynamics. The method may be applied to other types of relaxation oscillator networks.

Acknowledgments. The authors thank D. Terman for useful discussions and critical reading of an earlier draft. PSL was supported in part by grants from ONR and DOE. DLW was supported in part by ONR, NSF, and an ONR Young Investigator Award.

References

[1] R.L. Burden and J.D. Faires, Numerical analysis, 4th ed. (PWS-KENT, Boston, MA, 1989).

[2] R.L. Didday, A model of visuomotor mechanisms in the frog optic tectum, Math. Biosci. 30 (1976) 169-180.

[3] R. Eckhorn, et al., Coherent oscillations: A mechanism of feature linking in the visual cortex, Biol. Cybern. 60 (1988) 121-130.

[4] R. FitzHugh, Impulses and physiological states in models of nerve membrane, Biophys. J. 1 (1961) 445-466.

[5] D. Golomb and J. Rinzel, Clustering in globally coupled inhibitory neurons, Physica D (in press).

[6] J. Grasman, Asymptotic methods for relaxation oscillations and applications (Springer-Verlag, New York, 1987).

[7] C.M. Gray, P. König, A.K. Engel, and W. Singer, Oscillatory responses in cat visual cortex exhibit inter-columnar synchronization which reflects global stimulus properties, Nature 338 (1989) 334-337.

[8] S. Grossberg, Adaptive pattern classification and universal recoding: I. Parallel development and coding of neural feature detectors, Biol. Cybern. 23 (1976) 121-134.

[9] A.L. Hodgkin and A.F. Huxley, A quantitative description of membrane current and its application to conduction and excitation in nerve, J. Physiol. (Lond.) 117 (1952) 500-544.

[10] N. Kopell and D. Somers, Anti-phase solutions in relaxation oscillators coupled through excitatory interations, J. Math. Biol. 33 (1995) 261-280.

[11] P.M. Milner, A model for visual shape recognition, Psychol. Rev. 81(1974) 521-535.

[12] C. Morris and H. Lecar, Voltage oscillations in the barnacle giant muscle fiber, Biophys. J. 35 (1981) 193-213.

[13] J. Nagumo, S. Arimoto, and S. Yoshizawa, An active pulse transmission line simulating nerve axon, Proc. IRE 50 (1962) 2061-2070.

[14] D.H. Perkel and B. Mulloney, Motor pattern production in reciprocally inhibitory neurons exhibiting post-inhibitory rebound, Science 185 (1974) 181-183.

[15] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, Numerical Recipes in C, 2nd ed. (Cambridge University Press, New York, 1992).

[16] D.E. Rumelhart and D. Zipser, Feature discovery by competitive learning, in Parallel distributed processing, vol. 1, D.E. Rumelhart and J.L. McClelland, eds (MIT Press, Cambridge, MA, 1986) 115-193.

[17] W. Singer and C.M. Gray, Visual feature integration and the temporal correlation hypothesis, Ann. Rev. Neursci. 18 (1995) 555-586.

[18] D. Somers and N. Kopell, Rapid synchrony through fast threshold modulation, Biol. Cybern. 68 (1993) 393-407.

[19] M.R. Spiegel, Mathematical handbook of formulas and tables, Schaum's Outline Series (McGraw-Hill, New York, 1991).

[20] O. Sporns, G. Tononi, and G.M. Edelman, Modeling perceptual grouping and figure-ground segregation by means of active re-entrant connections, Proc. Natl. Acad. Sci. USA 88 (1991) 129-133.

[21] D. Terman and D.L. Wang, Global competition and local cooperation in a network of neural oscillators, Physica D 81 (1995) 148-176.

[22] B. van der Pol, On 'relaxation oscillations', Philos. Mag.. 2 (1926) 978-992.

[23] F. Verhulst, Nonlinear differential equations and dynamical systems (Springer-Verlag, Berlin Heidelberg, 1990).

[24] C. von der Malsburg, The correlation theory of brain function, Internal Report 81-2, Max-Planck-Institute for Biophysical Chemistry (1981).

[25] C. von der Malsburg and W. Schneider, A neural cocktail-party processor, Biol. Cybern. 54 (1986) 29-40.

[26] D.L. Wang, Pattern recognition: Neural networks in perspective, IEEE Expert 8 (August, 1993) 52-60.

[27] D.L. Wang, Emergent synchrony in locally coupled neural oscillators, IEEE Trans. Neural Net. 6 (1995) 941-948.

[28] D.L. Wang and D. Terman, Locally excitatory globally inhibitory oscillator networks, IEEE Trans. Neural Net. 6 (1995) 283-286.

[29] D.L. Wang and D. Terman, Image segmentation based on oscillatory correlation, Neural Comp. (in press); see also Tech. Rep. 19, OSU Center for Cognitive Science (1996).

Figure Caption

Figure 1. Nullclines and trajectories of a single relaxation oscillator. **A** Behavior of an enabled oscillator. The limit cycle is shown with a bold curve, and its direction of motion is indicated by arrowheads. LB and RB denote the left branch and the right branch of the cubic, respectively. LK and RK denote the left knee and the right knee of the cubic, respectively. **B** Behavior of an excitable oscillator. The oscillator approaches the stable fixed point P_I.

Figure 2. Architecture of a two dimensional LEGION network. An oscillator, indicated by an open circle, connects with four nearest neighbors with no wrap-around on the boundary and a global inhibitor is indicated by the filled circle.

Figure 3. Limit cycle trajectory in the singular limit. In addition to the notations in Fig. 1A, LC and RC indicate the left and right corners of the limit cycle, respectively.

Figure 4. A An image with three patterns, as mapped to a 50x50 LEGION network. If a square is entirely covered by the input, the corresponding oscillator receives external input; otherwise, the oscillator receives no external input. **B** An input image resulting from **A** corrupted with 20% noise. **C** A snapshot at the beginning of system evolution. **D-F** Subsequent snapshots taken shortly after the system starts. The above simulation includes the lateral potential, whereas the simulation in G-K does not. **G** A snapshot at the beginning of system evolution for the simulation without the lateral potential. **H-K** Subsequent snapshots taken shortly after **G**.

Figure 5. Temporal activity of every stimulated oscillator. **A** Results generated from the singular limit method. **B** Results generated from the fourth-order Runge-Kutta method (from [29]). In either **A** or **B**, the upper three traces give the combined x activities of the three oscillator blocks indicated by their respective labels. The fourth one gives the x activities of the loners, and the bottom one gives the activity of the global inhibitor. The oscillator activity is normalized in the figure, and the global inhibitor is displayed accordingly. Both simulations correspond to the same period from $t = 0$ to $t = 36$.

Figure 6. A An image with nine patterns, as mapped to a 30x30 LEGION network. See the Fig. 4 legend for notations. **B** An input image resulting from **A** corrupted with 10% noise. **C** A snapshot at the beginning of system evolution. **D-H** Subsequent snapshots taken shortly after **C**.

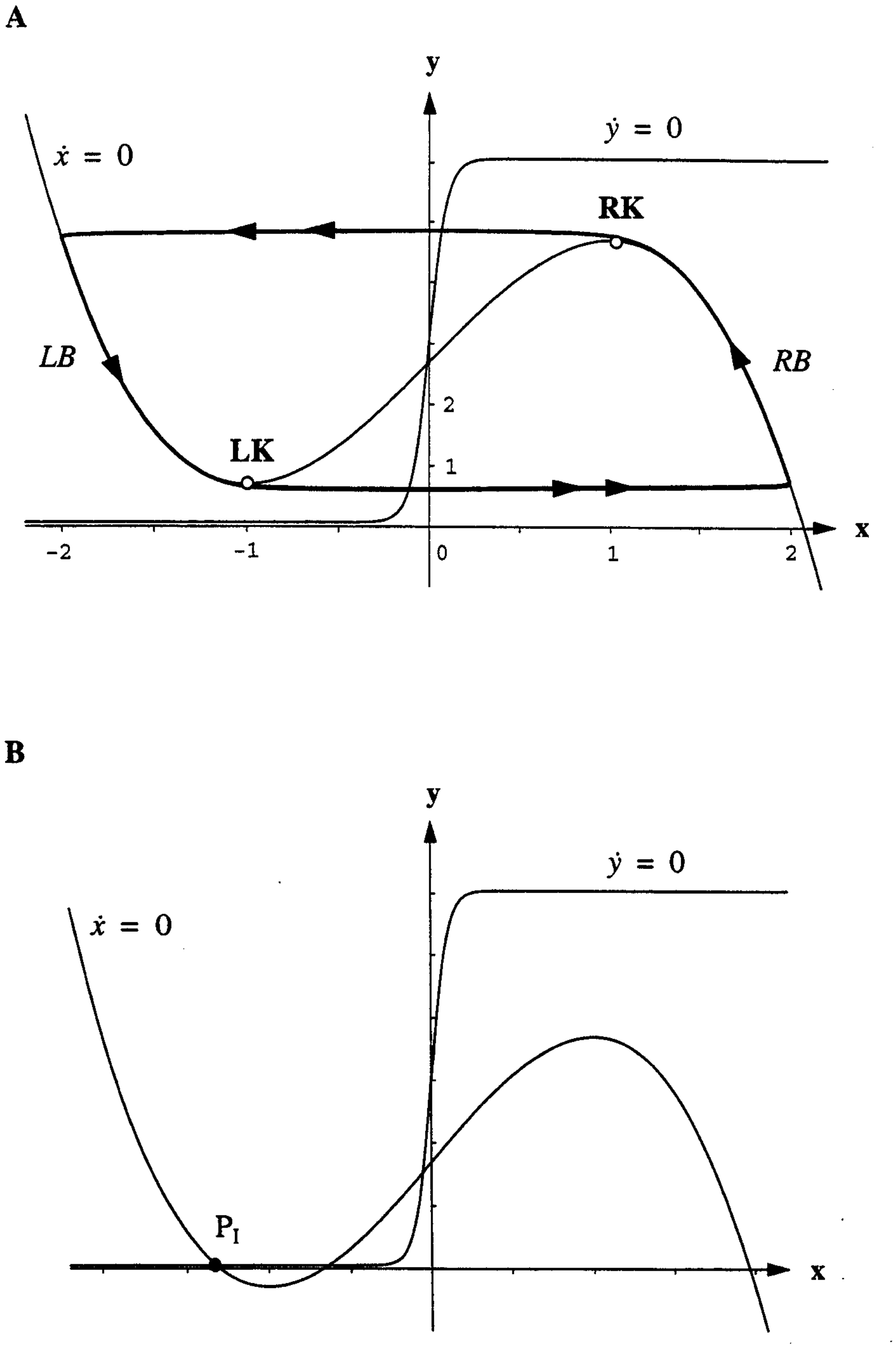


Fig.1

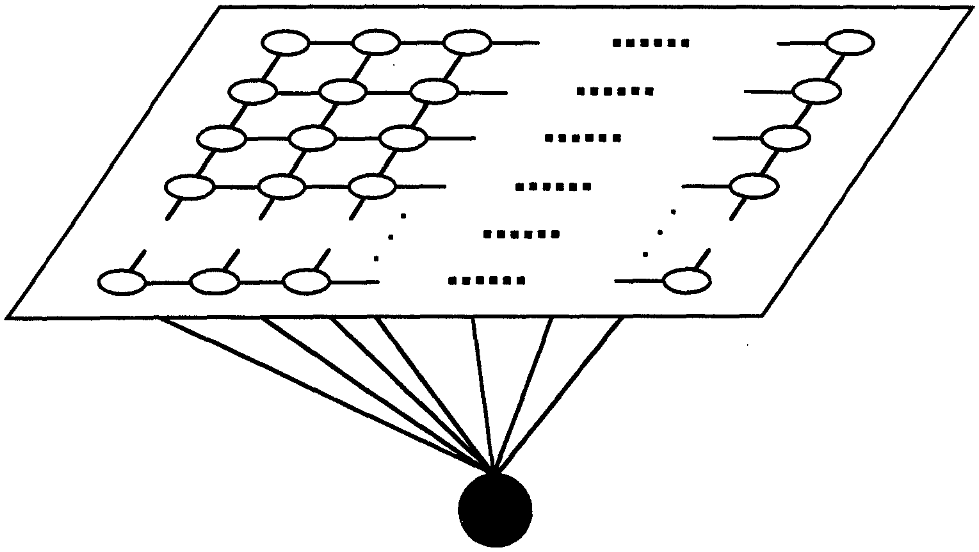


Fig.2

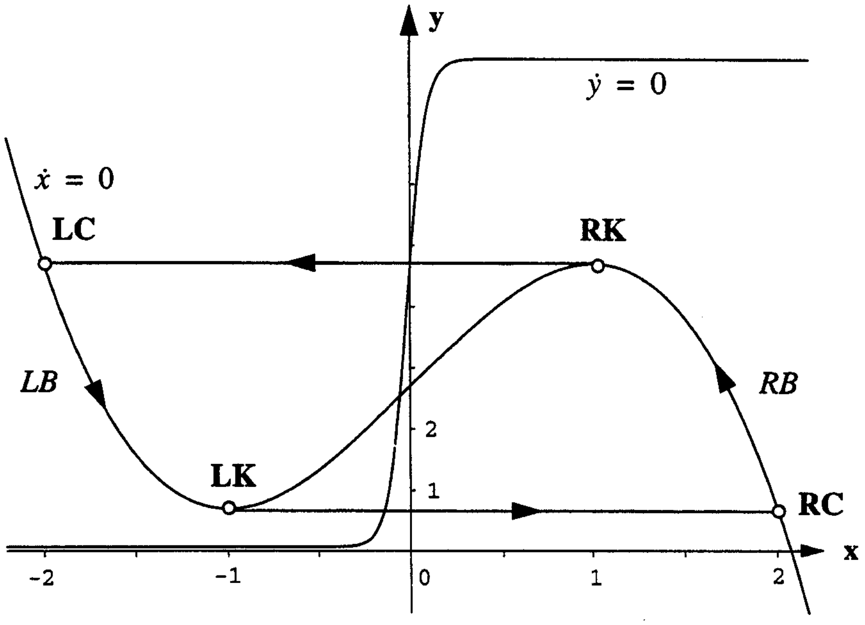
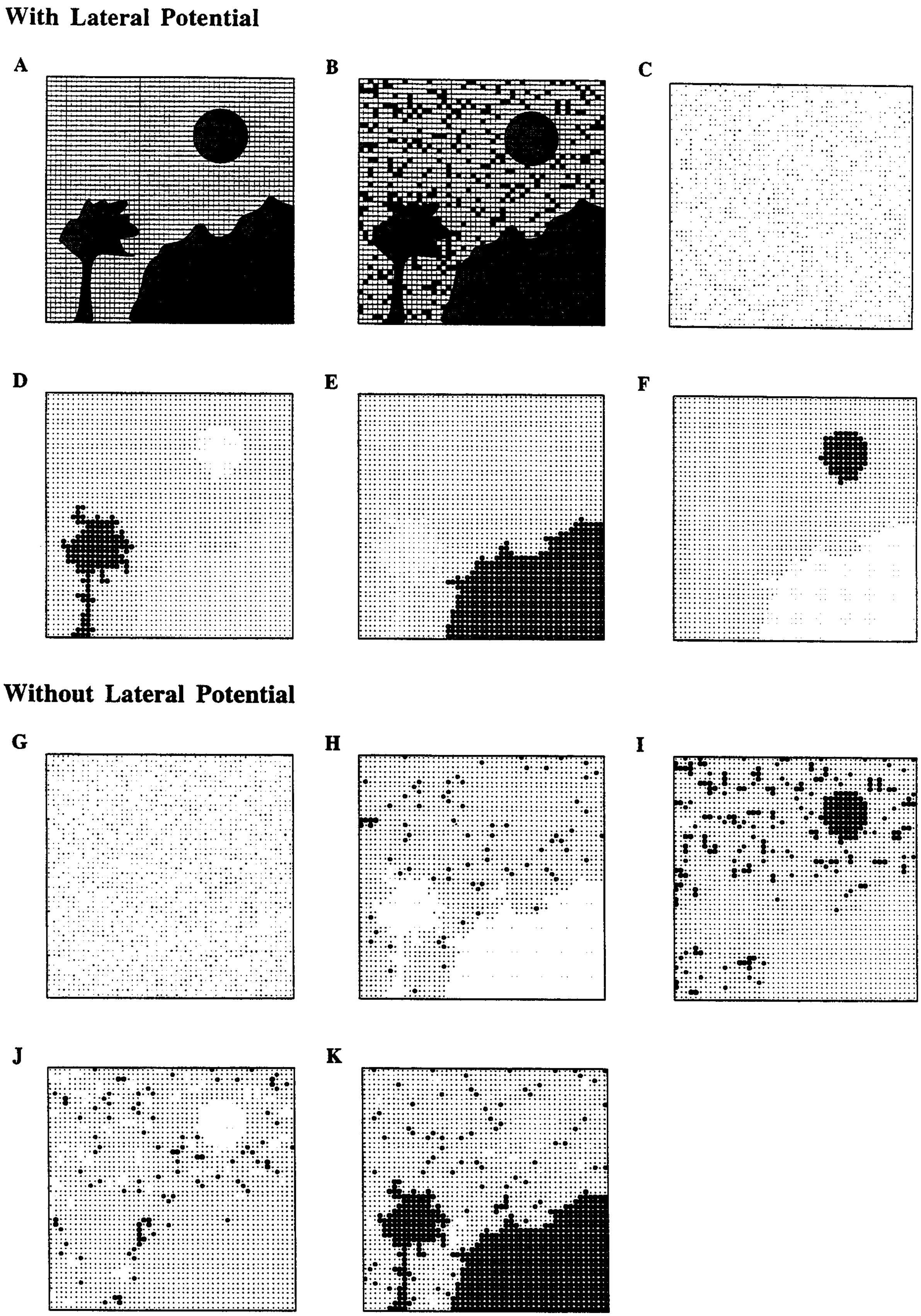


Fig. 3



With Lateral Potential

Without Lateral Potential

Fig. 4

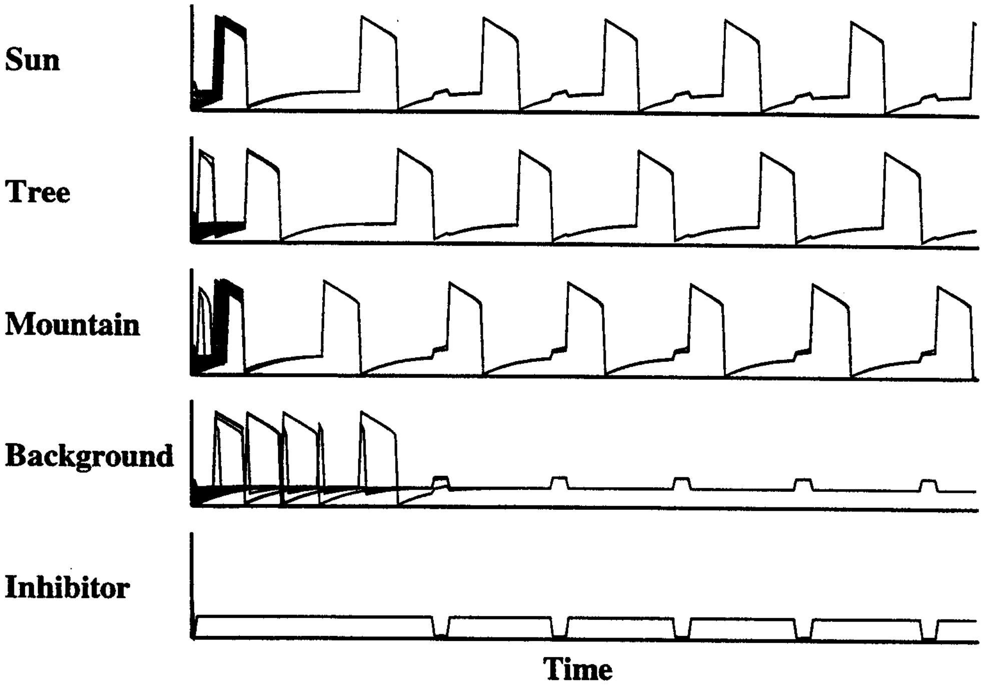


Fig.5A

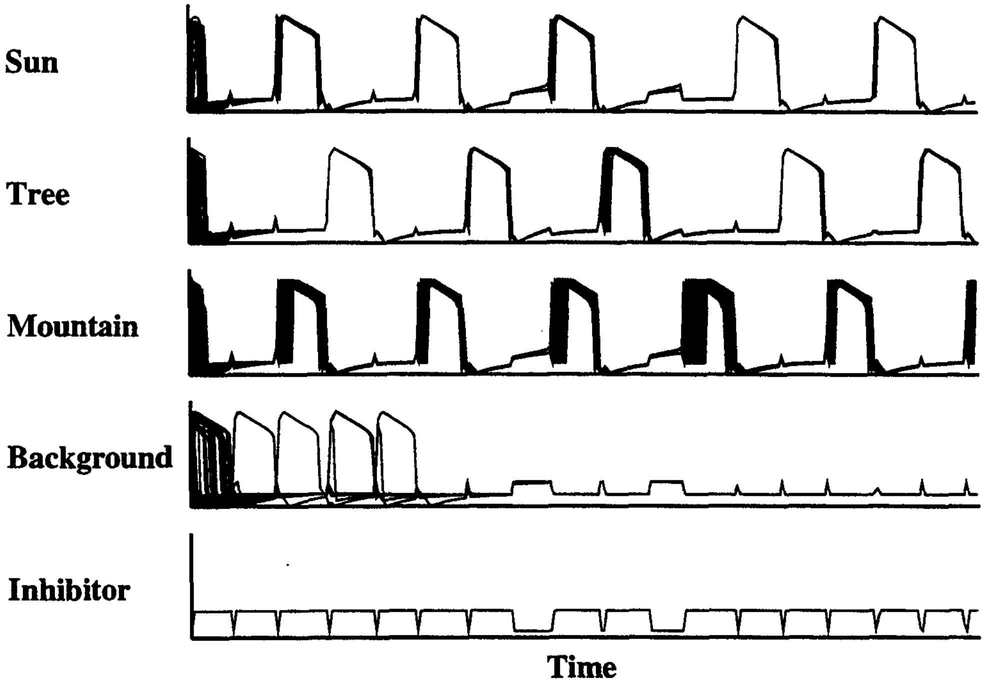


Fig. 5B

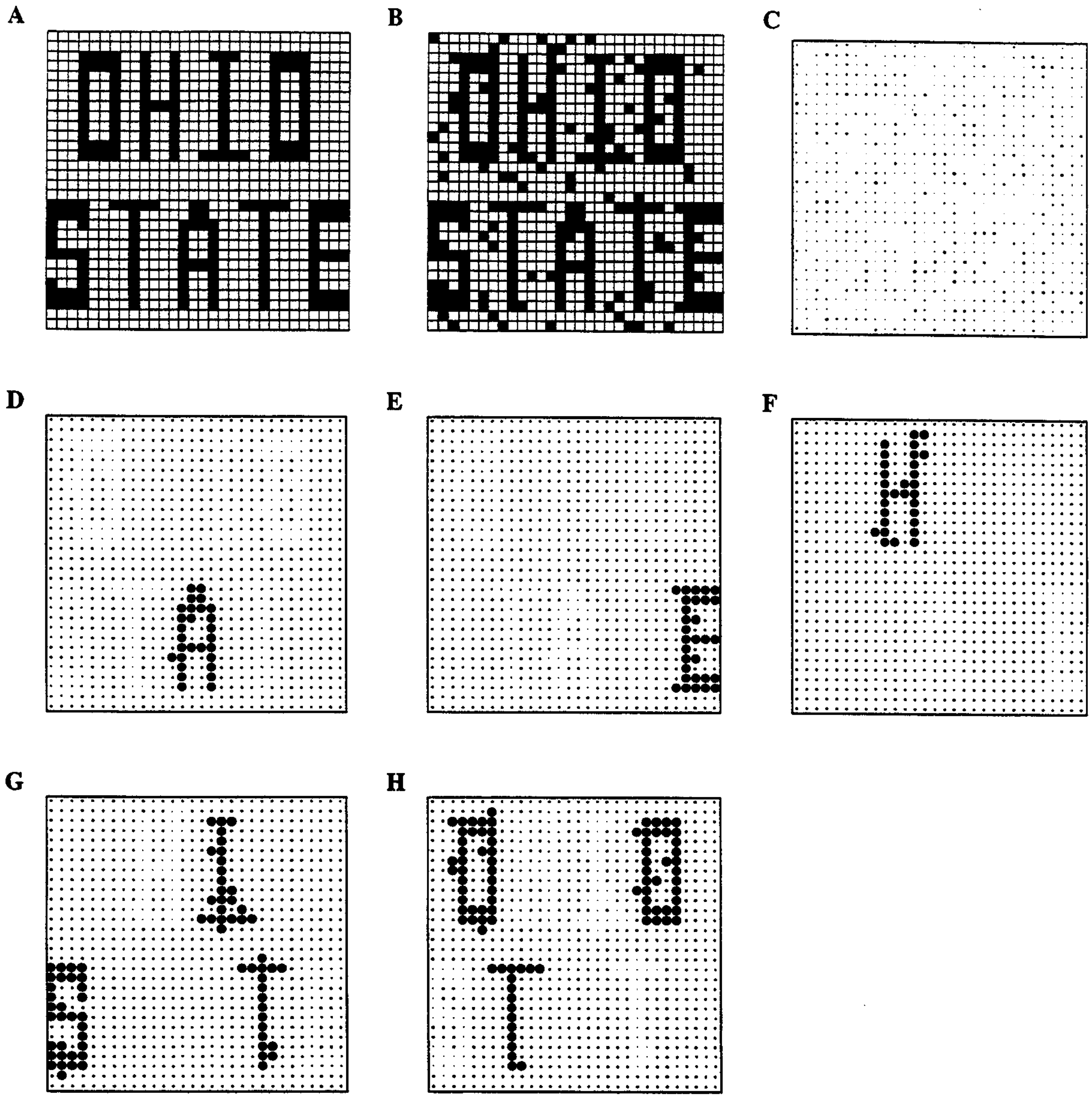


Fig. 6