PERGAMON

Neural Networks 12 (1999) 825–836

Neural
Networks

www.elsevier.com/locate/neunet

A learning algorithm for oscillatory cellular neural networks

C.Y. Ho[a,*], H. Kurokawa[b]

[a]*Department of Electronic Engineering, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon, Hong Kong, People's Republic of China*
[b]*Tokyo Engineering University, 1401-1, Katakuramachi, Hachioji-shi, Tokyo, 192 Japan*

Received 19 August 1998; received in revised form 13 April 1999; accepted 13 April 1999

Abstract

We present a cellular type oscillatory neural network for temporal segregation of stationary input patterns. The model comprises an array of locally connected neural oscillators with connections limited to a 4-connected neighborhood. The architecture is reminiscent of the well-known cellular neural network that consists of local connection for feature extraction. By means of a novel learning rule and an initialization scheme, global synchronization can be accomplished without incurring any erroneous synchrony among uncorrelated objects. Each oscillator comprises two mutually coupled neurons, and neurons share a piecewise-linear activation function characteristic. The dynamics of traditional oscillatory models is simplified by using only one plastic synapse, and the overall complexity for hardware implementation is reduced. Based on the connectedness of image segments, it is shown that global synchronization and desynchronization can be achieved by means of locally connected synapses, and this opens up a tremendous application potential for the proposed architecture. Furthermore, by using special grouping synapses it is demonstrated that temporal segregation of overlapping gray-level and color segments can also be achieved. Finally, simulation results show that the learning rule proposed circumvents the problem of component mismatches, and hence facilitates a large-scale integration. © 1999 Elsevier Science Ltd. All rights reserved.

Keywords: Neural oscillators; Temporal segmentation; Time lag; Global desynchronization; Connectedness matrix

Nomenclature

T	positive feedback synapse of oscillator
$H(U)$	piecewise linear activation function of neurons
U_1,U_2	outputs for mutually coupled neurons
w_{12},w_{21}	weight synapses
I_1,I_2	external inputs for mutually coupled neurons U_1 and U_2, respectively
τ_1,τ_2	time constants
ϕ_{learn},ϕ_{target}	phase of learning and target neurons, respectively
ϵ_1,ϵ_2	learning coefficients
T^0_{learn}	initial value of positive feedback synapse
ϕ^{tar}_{ij}	target phase for neuron *ij*
O_{ij}	oscillator at grid *ij*
O_{learn},O_{target}	learning and target oscillators
$W_{ij,kl}$	connectedness matrix for binary image
$GW_{ij,kl}$	connectedness matrix for *gray-level* image
$CW_{ij,kl}$	connectedness matrix for *color* image

1. Introduction

Human perception shows the capability of simultaneously combining various sensory fields from different cortical areas to form a high level interpretation of an object. Experimental results performed on the olfactory cortex of rabbits (Skarda & Freeman, 1987) showed that synchronization among different cortical attractors emerging from a chaotic substratum is an efficient way of storing and recall of memory and of information processing. While phase synchrony among different cortical attractors can be perceived as a high level interpretation of the different characteristics of an object including its smell, size, orientation, color, etc., phase asynchrony among different cortical attractors can be interpreted as the existence of multiple objects in the scene. Temporal phase locking phenomenon seems to exist between the striate cortices across two brain hemispheres (Engel, König, Kreiter & Singer, 1991), and it might constitute a feasible mechanism for solving the binding solution (Malsburg & Schneider, 1986). Malsburg and Schneider showed that by temporal segregating spatially isolated patterns into different phases, pattern segmentation is realizable with locally coupled neural oscillators. Thus when coherent objects share a common phase, they are logically binded together.

[*] Corresponding author. Tel.: +852-2788-9132; fax: +852-2788-7791.
E-mail address: eemurphy@cityu.edu.hk (C.Y. Ho)

PII: S0893-6080(99)00035-0

The capability of temporal segmentation of spatially isolated binary patterns via oscillatory neural networks has been discussed by Malsburg and Schneider (1986), König and Schillen (1991), Schillen and König (1991), Wang (1995), and Campbell and Wang (1996). Their learning rules are based on the connectedness of adjacent oscillating pixels while uncorrelated segments are separated by at least one white pixel. Overlapping segments, however, collapsed into one large segment even if the segments were not correlated. The problem is mainly due to the limited binary representation of image (pattern) pixels, i.e. the presence and absence of objects. If an input image is presented in *gray-level* or in *color* domain, overlapping pixels could be readily discerned in accordance with their gray intensity or their R, G, B values. According to Wang (1995) *uncorrelated* objects are discerned by assigning a different, but *uncontrollable* phase to each oscillator group. Meanwhile, accidental synchronization among non-overlapping objects cannot be avoided. This problem of erroneous synchrony is alleviated by Campbell and Wang (1996) by using a *global separator* that accelerates the speed of the leading oscillator, and thus safeguards the *phase delay* between non-overlapping objects. Campbell et al. showed that the *global separator* mechanism, which requires an immense number of interconnections between every neural oscillator and the *global separator*, is able to desynchronize oscillator groups with small phase shifts within one cycle. By adopting relaxation oscillator, the architecture is further generalized (Terman & Wang, 1995; Wang & Terman, 1997; Campbell & Wang, 1998) to handle real *gray-level* images. However due to a limited phase resolution between different objects, their dynamic systems can segregate only a limited number of patterns with a fixed set of parameters. Add to that objects in the image scene are randomly ‘popped out’ regardless of their respective geometry relations.

Recent years have seen a growing interest in the applications of oscillatory neural networks for information processing. However, a large-scale implementation of the aforementioned oscillatory network model, which consists of immense global interconnections, seems to be difficult, if not impossible. In practice, global connection among an $L \times M$ array of neural oscillators is a nightmare when L and M become large. To date, ultra-large-scale integration (ULSI) technology allows a fan-in/fan-out of up to a few hundreds, however, with much of the chip space being used for routing. A more realistic approach is the implementation with connections limited to a cell’s direct neighborhood akin to the structure of cellular neural networks (Chua & Yang, 1988). It has been shown that by using a network of locally connected nonlinear elements, global features can be extracted with connections confined to nearest neighbors. This is in sharp contrast to the requirements of a *global separator* and global connections for global synchronization and for the prevention of spurious desynchronization.

In contrast to the previous works that tried to model human vision from a pure biological point of view, we shall on the other hand, strive to find an efficient way of implementing an artificial visual device that realized real time image segregation. It is strongly believed that the idea of *global separator* cannot be readily accepted for large-scale applications. In this article, we shall demonstrate that the aforementioned problems of erroneous synchrony and the need of a *global separator* can be circumvented by using a cellular type oscillatory neural network. By synchronizing *stationary* input objects into different common phases, the model is capable of temporally segregating objects in the sensory scene, and of maintaining the geometry relations among different objects. Our proposed oscillatory cellular neural network (OCNN) architecture comprises an array of simple neural oscillators with interconnections limited to the nearest neighborhood. This overcomes a bottleneck that would be created by the immense number of interconnections between a *global separator* and the oscillators in devices such as optical silicon retinas.

As is suggested by Campbell and Wang (1998), the maximum number of groups that can be segregated is related to the maximum time difference within each group. Our model has a better performance since the phase dispersion of each converged oscillator group can be controlled within a very high precision (less than 0.01°). The problem of spurious synchronization of overlapping but uncorrelated segments in conventional oscillator model is solved by the introduction of a fast time-scale *connectedness matrix* (CM) that signifies the *grouping* between adjacent oscillators. In comparison with the global separator approach, these merits will be, however, offset by a longer convergence time for object segmentation. In essence we shall use an oscillatory model that consists of one and only one plastic synapse in order to reduce the complexity for hardware implementation, even if its analogy with biological counterpart is less evident. The model is basically similar to the oscillator proposed by Amari (1972) that comprises inhibitory and excitatory synapses. However this simplified structure enables a simpler learning rule that can easily control the oscillating frequency and phase via a single positive feedback synapse. In contrast to our previous work (Kurokawa, Ho & Mori, 1997), the learning rule proposed in this context deploys the time-delayed phase error between the target and learning oscillators. This alleviates the problem of unsteady oscillations due to nonlinear phase error. In particular the phase error between learning and target oscillators can be entrained to be arbitrarily small with steady oscillations. This time-delayed learning approach can be further generalized to *gray-level* and *color* pattern segmentation whereas spatially overlapping *gray-level* and *color* segments are entrained to oscillate with the same prescribed frequency except with a different phase shift.

In the following section, we shall describe the properties of temporal segregation of binary images. The idea of sequential image segregation that retains the geometry relations among different objects in the image scene is elaborated. In Section 3, we shall briefly discuss the

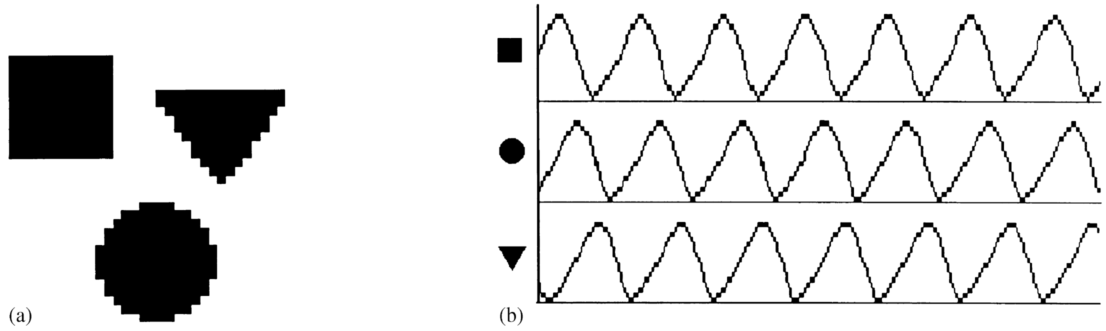

Fig. 1. (a) Binary input of 3 polygons. (b) Desired oscillator output temporal sequence.

characteristics of the proposed oscillator model. Its dynamic range and the oscillating criterion will also be addressed. A time-delayed learning rule is then proposed to control the phase (*time*) delay between two oscillators. In Section 4, we shall present a cellular type oscillatory neural network model that comprises local connections confined to a 4-connected neighborhood. A wave-propagation-like phase initialization scheme is conceived so as to eliminate any erroneous synchrony during temporal segmentation and to maintain the geometry relations among different objects. By assigning the initial phase relationship of the oscillator array into an ordered manner, it is demonstrated that sequential object segmentation can be achieved when the time lag between successive objects is made proportional to their Euclidean distances. In Section 5, temporal segregation of overlapping *gray-level* and *color* patterns is investigated. It is manifested that the problems of erroneous synchrony in conventional oscillatory networks can be circumvented by temporally segregating overlapping *gray-level* or *color* segments with respect to their *connectedness matrix* (CM) values. Further in Section 6, validity of the proposed oscillatory cellular neural network is examined via extensive numerical simulations.

2. Sequential image segmentation

For sequential image segmentation, our objective here is to temporally synchronize correlated objects from a left to right (right to left, or top to bottom) manner where stationary objects are sequentially popped out with different phases. In the previous studies (Malsburg & Schneider, 1986; Campbell & Wang, 1996; Wang & Terman, 1997), correlated objects are successfully popped out in a random manner regardless of their relative displacements. This context introduces a special phase initialization scheme so that subsequent pattern segregation can retain the geometry relations among the objects in the sensory scene.

Generally speaking, scene segmentation can be divided into two steps; namely grouping and segregation. Conventional image segmentation techniques heavily rely on special heuristics to perform grouping while pattern segregation is based on their connectedness or on multiple thresholding like pixel histogram (Kohler, 1981). In fact, there exists no particular criterion if grouping should be performed before or after segregation. A successful grouping eases subsequent segregation while a prior meaningful segregation helps to identify object boundary for grouping. In this article, we shall assume that all objects in the sensory scene are *stationary* and have uniform luminosity for the sake of illustration, and thus (i) binary objects *grouping* can be realized according to their connectedness; (ii) *gray-level* objects *grouping* can be realized according to their gray intensity; and (iii) *color* objects *grouping* can be achieved according to their fuzzified color domain will be described later.

Having the aforementioned assumptions for grouping, all we needed to do is temporal segregation in accordance with object connectedness. In order to maintain the geometry information among different objects, one can arrange the phases of coherent objects in an ordered phase relationship so that the leftmost (rightmost) object should always be leading. If a *global* target oscillator is present, this can be realized by initializing the phase of the *i*th column oscillators ϕ_i of an $L \times M$ oscillator array with

$$\phi_i(t) = \phi_{\text{target}}(t - ki\tau_s), \qquad (1)$$
$$k \text{ is an integer constant, and } i = 1, 2, \ldots, M$$

where τ_s is the time-step, and thus the phases of successive column oscillators differ by k time-step. Upon initialization, the phase relation of the oscillatory array is similar to ripple propagating from a left to right manner with end-around shift. When an image is presented to the oscillatory array with each image pixel being represented by an oscillator, correlated object pixels will be most likely oscillating with similar phase. If the correlated phases are averaged out, the synchronized phases can be divided into different groups as depicted in Fig. 1. Consider a 32 × 32 binary image shown in Fig. 1(a) with 3 polygons, namely a square, a circle and a triangle. Depending on user's practice, the temporal pop-out sequence with proper thresholding could possibly be ■ → ▼ → ●, ■ → ● → ▼, or ▼ → ● → ■ etc. Suppose the desired temporal output sequence is ■ → ● → ▼, output oscillator responses of each correlated group will be similar to that shown in Fig. 1(b). In the pattern each pixel is

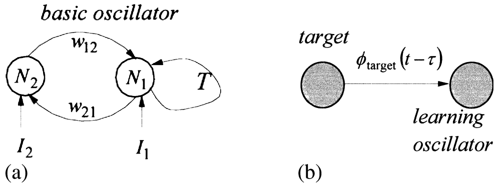

Fig. 2. 2-neuron oscillator model and learning mechanism: (a) oscillator model; (b) learning mechanism.

represented by an oscillator, and all correlated oscillators are synchronized with no phase error while different oscillator groups are desynchronized. The phase difference between successive oscillator group determines the time lap of the 'pop-out' sequence. In our model, however, the maximum time lap is governed by the oscillator period. Finally it should be noted that due to the cyclic effect of the oscillators, the 'pop-out' sequence of the patterns is ■ → ● → ▼ → ■ → ●...

3. Dynamics of neural oscillator model

Fig. 2(a) shows the basic neural oscillator model that composed of 2 neurons. Dynamics of the neurons N_1 and N_2 is governed by

$$\tau_1 \frac{dU_1}{dt} = -U_1 + w_{12}H(U_2) + TH(U_1) + I_1, \tag{2a}$$

$$\tau_2 \frac{dU_2}{dt} = -U_2 + w_{21}H(U_1) + I_2, \text{ and} \tag{2b}$$

$$H(U_i) = \begin{cases} 1 & \text{if } U_i \geq 1.0 \\ (1+U_i)/2 & \text{if } |U_i| < 1.0, \qquad i = 1, 2, . \\ 0 & \text{if } U_i \leq -1.0 \end{cases} \tag{2c}$$

where U_1 and U_2 are the internal states of the neurons. Parameter T is the positive feedback synapse for N_1, $H(U)$ the piecewise linear function that limits the dynamic range of the proposed model, and w_{12} and w_{21} the weight connections. τ_1 and τ_2 are the time constants, and I_1 and I_2 are the external inputs of neurons N_1 and N_2, respectively. Without loss of generality, one can set $\tau_1 = \tau_2 = 1.0$. The model is basically similar to that proposed by Amari (1972) with the exception that one and only one plastic synapse T is needed for frequency and phase learning, and the sigmoid function being replaced by a piecewise linear function. The reason that we adopt the oscillator structure depicted in Fig. 2 is mainly due to its simplicity and feasibility for hardware implementation. By imposing the constraints

$$(w_{12} + T)/2 + I_1 = w_{21}/2 + I_2 = 0, \tag{3}$$

it can be shown that at least one of the equilibrium points (U_1, U_2) is fixed at (0,0). In the special case when $I_1 = I_2 = 0$, (U_1, U_2) asymptotically converges to (0,0) (see Appendix A), which is the only fixed point, and the oscillator is non-oscillating. When $I_1 = -I_2 \neq 0$, the oscillator becomes oscillating with its frequency and phase controlled by T. It can be shown analytically (see Appendix B) that the oscillator is oscillating if the positive synapse $T > 2/H'(0)$ when $I_1 = -I_2 \neq 0$. In order to realize the oscillator model into silicon, we have to pay attention to the dynamic ranges of the outputs U_1 and U_2 since they are controlled by a nonlinear integration function. One can show (see Appendix C) that by imposing the constraints Eq. (3), the dynamic ranges of U_1 and U_2 are well within the typical power supply voltage of common integrated circuits.

3.1. Time-delayed learning algorithm

In contrast to the studies (Malsburg & Schneider, 1986; Wang, 1995; Campbell & Wang, 1996) that used the amplitude difference between two oscillators as the learning parameter, we adopt the phase difference here since the control on the phase of individual oscillator can be more precise. Consider a coupled oscillator pair depicted in Fig. 2(b), the objective is to synchronize the learning oscillator O_{learn} with the target oscillator O_{target}. This is accomplished by varying the positive plastic synapse T_{learn} with

$$T_{\text{learn}}(t) = T^0_{\text{learn}} + \varepsilon_1 \int_0^t \Delta\phi(t)\, dt + \varepsilon_2 \Delta\phi(t), \tag{4a}$$

$$\Delta\phi(t) = \phi_{\text{learn}}(t) - \phi_{\text{target}}(t - \tau), \text{ and} \tag{4b}$$

$$\phi_x = \begin{cases} \cos^{-1}\left(U_{1,x}/\sqrt{(U_{1,x})^2 + (U_{2,x})^2}\right) & \text{if } U_{2,x} \geq 0 \\ -\cos^{-1}\left(U_{1,x}/\sqrt{(U_{1,x})^2 + (U_{2,x})^2}\right) & \text{if } U_{2,x} < 0 \end{cases}, \tag{4c}$$

$x \in \{\text{learn}, \text{target}\}$.

The parameters ϕ_{learn} and ϕ_{target} are, respectively, the phase of the learning oscillator and that of the target oscillator. T^0_{learn} is the initial value of positive feedback for O_{learn}, and τ is the user defined time lag between the learning oscillator and the target oscillator. Parameters ε_1 and ε_2 are constants that control the learning behavior of the system. Upon convergence, $\Delta\phi(t) = 0$, and the learning oscillator will be oscillating at the same frequency as that of the target oscillator except with a time delay, τ. In contrast to our previous work (Kurokawa et al., 1997), the control parameter is substituted with Eq. (4b) due to the non-sinusoidal oscillating trajectory of the model. The second term in Eq. (4a) circumvents the problem of parameter *mismatches* in T^0_{learn} while the third one tackles the phase errors between the learning and target oscillators.

Now suppose that the phase of the learning oscillator is leading, the feedback synapse T_{learn} will be increased, and this will effectively slow down the oscillation (Kurokawa et al., 1997). Since $\Delta\phi(t)$ in Eq. (4b) is positive, T_{learn} will be increased and the oscillation is retarded, and vice versa. Thus by using the learning rule stated in Eq. (4), the frequency and phase of an oscillator can always be

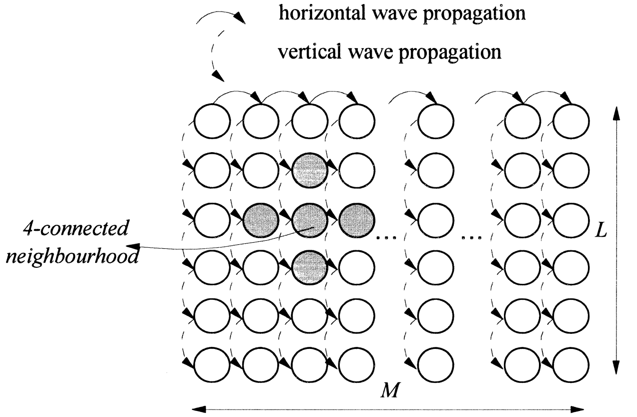
horizontal wave propagation
vertical wave propagation
4-connected neighbourhood
...
...
L
M

Fig. 3. Initialization of the phases of oscillatory cellular neural networks without using global target.

synchronous with the target oscillator. Hereafter we shall consider the characteristics of an extensively coupled oscillatory array for image segregation.

4. Oscillatory cellular neural networks with local learning rule

The architecture of an oscillatory cellular neural network (OCNN) is shown in Fig. 3. Each cell is basically an oscillator depicted in Fig. 2(a). Without loss of generality, every oscillator has connections confined to its 4-connected neighboring cells. In essence every oscillator will refer to the oscillator on its left or upper row as its target oscillator for frequency learning. Thus the information needed for each oscillator is necessarily local. Since all the oscillators directly or indirectly refer to the leftmost oscillator in the top row as its target oscillator, this oscillator should maintain uppermost oscillation stability.

4.1. Initialization of the phase of OCNN with local learning rule

As described in Section 2, in order to retain the geometry relations among different objects in the image scene, it is indispensable to initialize the phase of the oscillatory array as a ripple propagating from the left to right. Fig. 3 shows an $L \times M$ oscillatory array. Assuming that the time-step of oscillation is given by τ_s, each oscillator in the top row is trained to differ in k time-step with respect to the oscillator on its left-hand-side. Oscillators in the same column are, however, trained to oscillate with the same phase. Upon initialization the phase behavior of the oscillatory array will then be similar to a ripple periodically propagating from the left to right. The local learning rule takes on the following form with

$$\phi_{1j}^{\rm tar}(t) = \phi_{1j-1}(t - k\tau_s) \qquad \forall j = \{2,\ldots,M\}, \text{ and} \tag{5a}$$

$$\phi_{ij}^{\rm tar}(t) = \phi_{i-1j}(t) \qquad \forall i = \{2,\ldots,M\}. \tag{5b}$$

The learning strategy is basically similar to that with a global target Eq. (1). However, the global target for oscillator O_{ij} (at the ith row and jth column) is now replaced with its neighboring oscillator, namely O_{ij-1} or O_{i-1j}, and a local learning rule can then be deployed.

4.2. Temporal segregation of OCNN via connectedness matrix

After initialization the oscillatory array is now ready for sequential object segmentation. Based on the idea that objects are usually segmented according to their connectedness, a special learning rule for pattern synchronization/desynchronization is suggested. When an input image is presented to the network, neighboring pixels with nonzero input have $I_1 = -I_2 = 1.0$ (oscillating) while those with zero inputs will have $I_1 = I_2 = 0.0$ (non-oscillating with asymptotic fixed points at (0,0)). Connections among neighboring pixels are limited to a 4-connected neighborhood defined by the *connectedness matrix* (CM) $W_{ij,kl}$ with

$$W_{ij,kl} = J_{ij} \times J_{kl} \qquad \forall k,l \in |k-i| + |l-j| \le 1, \tag{6a}$$

$$W_{ij,kl} = 1 \qquad \text{if } k,l = i,j, \tag{6b}$$

where $J_{ij} = 1$ if O_{ij} is oscillating, i.e. with nonzero inputs I_1 and I_2. These synapses can be interpreted as the dynamic weights discussed by Malsburg and Schneider (1986) and Wang and Terman (1997) with fast time-scale. The local neighborhood structure of the cellular array is indicated in Fig. 3. Thus for binary images, the connectedness between adjacent pixels is defined by $W_{ij,kl}$. By the same token, one can simply modify the CM so that the proposed OCNN model can be applied to *gray-level* or *color* image segregation.

Here for binary image segregation, the target phase for oscillator O_{ij} is calculated after the rule

$$\phi_{ij}^{\rm tar} = \begin{cases} \cos^{-1}\left(U_{ij}^1/\sqrt{(U_{ij}^1)^2 + (U_{ij}^2)^2}\right) & \text{if } U_{ij}^2 \ge 0 \\ -\cos^{-1}\left(U_{ij}^1/\sqrt{(U_{ij}^1)^2 + (U_{ij}^2)^2}\right) & \text{if } U_{ij}^2 < 0 \end{cases}, \tag{7a}$$

where

$$U_{ij}^1(t) = \frac{\sum_{k,l} W_{ij,kl} U_{1,kl}(t-1)}{\sum_{k,l} W_{ij,kl}}, \tag{7b}$$

$$U_{ij}^2(t) = \frac{\sum_{k,l} W_{ij,kl} U_{2,kl}(t-1)}{\sum_{k,l} W_{ij,kl}}, \tag{7c}$$

$\forall k,l \in |k-i| + |l-j| \le 1$, and

$$T_{\rm learn}(t) = T_{\rm learn}^0 + \varepsilon_2 \Delta\phi(t), \tag{7d}$$

Eq. (7) aims at averaging the phases among *grouped*

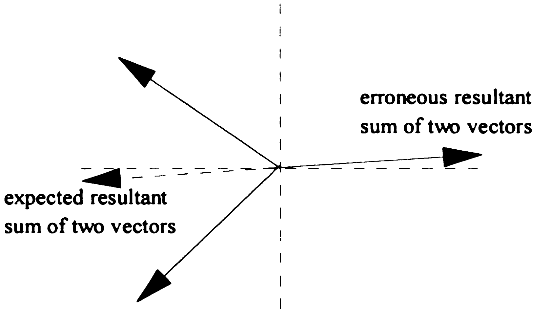

Fig. 4. Resultant of averaging two phases.

oscillating pixels, and hence upon convergence, 'connected' input pixels will be oscillating at the same phase. The synchronized phase among correlated segments will be approximately equal to the average of connected neighboring phases during initialization, and the time delay between uncorrelated segments will then be proportional to their Euclidean distance in the original input image. Eq. (4a) is replaced by Eq. (7d) with which the integration term is equated to zero. This forces all the converged objects to have the same 'T' (i.e. same frequency), and their mutual phase difference can be maintained over time. In contrast to the previous approach (Kurokawa et al., 1997), which calculates the average of the sum of 5 phases in a 4-connected neighborhood, Eq. (7) first estimates the resultant vectors of summing U_1 and U_2 and then converts it into the proper phase. This vector operating approach circumvents the problem of misleading phase sum of two angles when both are greater than $\pi/2$, as is depicted in Fig. 4.

5. Overlapping image segregation

In conventional oscillatory neural networks, temporal segregation is applicable only to spatially separated segments while overlapping segments will be merged into a single large segment. The problem is mainly due to the representation of input image in a limited binary domain. By using pixel classification, Wang and Terman (1997) successfully applied their models to segregate real *gray-level* images. In this article we shall extend the idea to include both *gray-level* images and *color* images with the R, G, B values of each pixel presented.

5.1. Gray-level image segregation

The connectedness matrix (CM) introduced in Section 4.2 effectively groups correlated objects together for phase averaging. As far as one can discern the boundary of two objects, CM can then be used to *group* overlapping *gray-level* or *color* patterns in the image scene. To segregate *gray-level* images, heuristic threshold techniques have been used to classify different regions (Kohler, 1981). Other approaches use a growing technique to categorize similar pixels into a connected region if they satisfy prescribed conditions. As we have mentioned in Section 4, the *connectedness matrix* (CM) implicitly realizes region classification where physically connected pixels are categorized as one object. For *gray-level* image, we adopt a thresholding technique for its simplicity and ease of implementation. Consider a *gray-level connectedness matrix* (GCM) $GW_{ij,kl}$, the connection between neighboring oscillators is determined after pixel classification with simple thresholding:

$$GW_{ij,kl} = J_{ij} \times J_{kl} G_{ij,kl} \qquad \forall k,l \in |k-i| + |l-j| \le 1, \tag{8a}$$

$$GW_{ij,kl} = 1 \quad \text{if } k,l = i,j, \tag{8b}$$

$$G_{ij,kl} = 1 \qquad \text{if } O_{ij} \text{ and } O_{kl} \text{ are of the same gray-level} \tag{8c}$$

$$\text{and 0 otherwise}$$

The GCM effectively *groups* segments with roughly the same gray intensity together if they are connected, and subsequent phase averaging operation will be conducted within isolated groups. Physically overlapped segments will, however, be discerned provided that they are of different gray intensity. This strategy helps to improve the architecture of OCNN, and the same learning rule Eq. (7) can be applied to the segregation of *gray-level* input images by replacing $W_{ij,kl}$ with $GW_{ij,kl}$.

5.2. Color image segregation

Color image segregation has a very different nature compared to *gray-level* image segregation. One of major difficulties is the diversity of color range that can be presented by the color components. Human vision perceives electromagnetic light beams by 4 types of photochemical transducers. These include 3 retinal cone cells for color and one retinal rod cell for light at low levels. The 3 cone cells contain different types visual pigment, namely 'Red', 'Blue' and 'Green' pigment. The cone cells detect 3 primary colors, and the brain mixes these colors into a wide color range that we perceive (Szaflarish, 1998). For a color description, the luminosity is usually separately expressed, and luminosity and chrominance can be described separately. If the objects have a uniform luminosity, one can fuzzy classifiy the chrominance of an object according to the continuous varying color spectrum.

The idea of fuzzy color classification has been studied by Benoit, Foulloy, Galichet and Mauris (1994). By using a single 80C196 micro-controller, the author demonstrated that it is possible to classify a colored surface into different color membership functions. Throughout the article, we shall assume that the image has a uniform luminosity, and make use of a fuzzy approach to classify the image pixels into different color clusters, and this is then followed by

Table 1
RGB settings for different color spectrums

R	G	B	Color
0	0	0	Black
255	0	0	Red
0	128	0	Green
255	128	0	Orange
0	255	0	Green
255	255	0	Yellow
255	0	255	Violet
255	128	255	Pink
0	255	255	Azure
255	255	255	White

pattern segregation. Due to the limited color range of machine vision, the fuzzy approach used in the article assumes that the input color can be effectively classified into 7 colors ‘Violet’, ‘Blue’, ‘Azure’, ‘Green’, ‘Yellow’, ‘Orange’ and ‘Red’. The RGB values for different color spectrum is shown in Table 1 while the values are fuzzified with the fuzzy rules summarized in Table 2. The RGB values can be interpreted as ‘HIGH’, ‘MEDIUM’ or ‘LOW’. Input image is first categorized into different color clusters after the fuzzy membership functions described in Fig. 5. In this article, fuzzy clustering is implemented by the fuzzy system toolbox described by Beale and Demuth (1994). With the fuzzy rules state in Table 2, the input color spectrum in Fig. 6(a) can be fuzzy classified to 7 color groups as shown in Fig. 6(b).

Similar to the *gray-level connectedness matrix* described in Section 5.1, connectedness between neighboring color pixels can be defined by a *color connectedness matrix* (CCM) $CW_{ij,kl}$ with

$$CW_{ij,kl} = J_{ij} \times J_{kl} \cdot C_{ij,kl} \qquad \forall k, l \in |k-i| + |l-j| \le 1, \tag{9a}$$

$$CW_{ij,kl} = 1 \quad \text{if } k,l = i,j, \tag{9b}$$

$$C_{ij,kl} = 1 \quad \text{if } O_{ij} \text{ and } O_{kl} \text{ are of the same color, and 0 otherwise} \tag{9c}$$

where $J_{ij} = 1$ if O_{ij} is oscillating, i.e. with nonzero inputs I_1 and I_2. While the input R, G, B values of a color pixel is crisp, their combinations are fuzzy classified into a crisp color group. This strategy simplifies the overall architecture of the OCNN, and by replacing $W_{ij,kl}$ with $CW_{ij,kl}$ the learning rule Eq. (7) can also be applied to the segregation of color input images.

Table 2
Fuzzy rules for color segmentation

Black	If none of Red, Green and Blue is HIGH
Red	If Red is HIGH, and Green and Blue are LOW
Orange	If Red is HIGH, Green is MEDIUM and Blue is LOW
Yellow	If Red and Green are HIGH, and Blue is LOW
Green	If Red and Blue are LOW, and Green is HIGH
Azure	If Red is LOW, and Green and Blue are HIGH
Blue	If Red and Green are LOW, and Blue is HIGH
Violet	If Red and Blue are HIGH, and Green is LOW
White	If none of Red, Green and Blue is LOW

6. Numerical simulation results

In Section 3, we have shown that the phase of an oscillator can be controlled via the positive feedback T. It can be analytically proven that T should be larger than $2/H'(0)$ in order to fulfil the conditions for oscillating. In the article, all the initial values of Ts of the oscillatory array are set to 4.0, with other parameters take on the values of $\epsilon_1 = 0.00001$, $\epsilon_2 = 0.015$, $\tau_1 = \tau_2 = 1.0$, and $I_1 = -I_2 = 1.0$. By substituting the values into Eq. (2), one immediately obtains $w_{12} = -6.0$, and $w_{21} = 2.0$.

6.1. Convergence of a coupled oscillator pair

In Fig. 2(b) two oscillators are coupled together where a single oscillator is trained to synchronous with the *time* lag version of target oscillator. In the following experiments, Ts of both the target and learning oscillators are set to 4.0. All the differential equations are calculated by using 4th order Runge–Kutta method. Table 3 shows the simulation results of the convergence time for the learning oscillator with a random initial phase delay of approximately $\{-30, 0, +30\}$ time-step with respect to the target oscillator. With each initial phase delay, the average convergence time for 10 random desired phase delays of $[-50, +50]$ time-step is recorded. It is demonstrated that an extremely small value of phase error of less than 0.01° can be achieved. If a much larger phase error, say 5° can be tolerated, the convergence time can be roughly halved. Noted that throughout the article, the convergence time is measured in the unit of cycle. The metric is a more favorable metric since the period of OCNN depends on the circuit parameters. It is obvious that the convergence time for multiple-point sampling is faster than that for zero crossing (single-point) learning. In the following simulations, multiple-point updating approach for T is used in order to differentiate the advantages of continuous time learning over zero crossing learning.

6.2. Convergence time of oscillator array with different sizes

Different sizes of oscillatory array are used to test the convergence speed of initialization. In this experiment, all the positive synapse Ts are set to 4.0 with ks being set to 3.

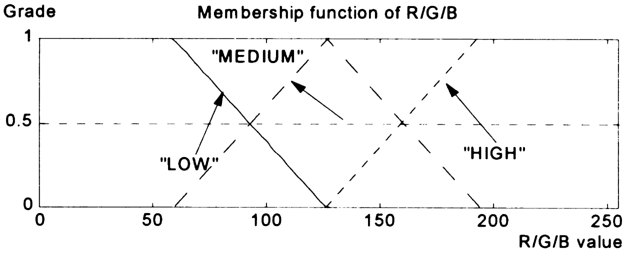

Fig. 5. Membership functions of RGB settings.

The criterion for convergence here is that the average T of the oscillators should have an error of less than 0.1% and the initialization process with local learning rule described in Section 4.1 is completed. As shown in Fig. 7, it is obvious that it takes a longer time to propagate the ripple along the column than along the row. This phenomenon can be easily understood since the initialization process needs to control the phase difference between adjacent columns to k time steps while the oscillators are synchronized for each column. In fact we have found that even when some of the Ts are mismatched (up to a discrepancy of 20%), the *long* convergence time in the initialization process can absorb these discrepancies without further prolonging the convergence time.

6.3. Binary pattern segregation via oscillatory array

A 6 × 13 oscillator array with phase initialization of the 4th column depicted in Fig. 8 is used to segment an input binary image with characters ‘C’, ‘I’, ‘T’ and ‘Y’. All the nonzero input pixels corresponding to the characters have $I_1 = -I_2 = 1.0$, and $I_1 = I_2 = 0.0$ otherwise. T of every oscillator and k takes on the value of 4.0 and 3, respectively. Follow the initialization scheme described in Section 4.1, the characters are temporally segmented and have maximum amplitudes at $t \approx 3$, 11, 21, and 33, respectively. And the phase (time) delay between successive characters is proportional to their Euclidean distance. Convergence of the oscillator array for initialization needs approximately 137 cycles, and an additional 44.2 cycles for pattern segmentation. This is compared to the case with a dedicated global target that uses 63.4 cycles for initialization and 44.2 cycles for pattern segregation. Note that the convergence time for oscillatory cellular neural network without global target is longer due to the ripple propagation property in the initialization process. Meanwhile the convergence time for pattern segmentation should be essential the same since the same local learn rule is used. The convergence time for subsequent pattern segregation, however, will be proportional to the value of k since it determines the phase dispersion of each ‘connected’ coherent object.

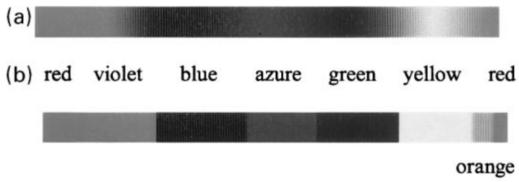

Fig. 6. (a) Input color spectrum. (b) Output color segmentation.

6.4. Gray-level pattern segregation

Similar to the segregation of binary image, overlapping gray-level patterns can be temporally segregated by the proposed OCNN. The process is basically the same as binary image segregation with an initialization process, and then follows by the segregation process. The only exception is the replacement of CM by *gray-level connectionedness matrix* (GCM). Here we assume that the input image has a uniform luminosity, and the solution can be reduced to the temporal segregation of overlapping patterns.

Fig. 9(a) shows a 25 × 32 *gray-level* image of 3 overlapping polygons similar to that shown in Fig. 1(a). All nonzero input pixels corresponding to the patterns have $I_1 = -I_2 = 1.0$, and $I_1 = I_2 = 0.0$ otherwise. T of every oscillator and k takes on the value of 4.0 and 1, respectively. Initialization of the oscillatory array needs about 863 cycles, and the convergence time for Ts (nominal value 4.0) of pattern ‘■’ segregation, which is shown in Fig. 9(b), takes approximately 827 cycles. The number of cycles needed for ‘●’ and ‘▼’ is 1173, and 940 respectively. Meanwhile, the converged phase behavior of the 3 patterns is summarized in Fig. 9(c). As far as convergence time is concerned, the overhead for the calculation of the *gray-level* connectedness matrix $GW_{ij,kl}$ is roughly equivalent to that for CW. Meanwhile, convergence time for *gray-level* pattern segregation is essentially the same as for binary pattern segmentation. Compared to the oscillatory model (Campbell & Wang, 1996) with *global separator*, our convergence speed is far slower. The difference is mainly due to the implicit learning activities during the long refractory period for those models described by Malsburg and Schneider (1986), Wang (1995), and Campbell and Wang (1996). In particular we have discovered that the synchronization process proceeds even during the refractory period (Campbell & Wang, 1996). One of the possible solutions to remedy the long learning time in our model is to use a faster time constants τ_1 and τ_2.

6.5. Color pattern segregation via oscillatory array

Similar to the segregation of *gray-level* image, *color* patterns can be temporally segmented by the proposed OCNN. The process is basically the same as *gray-level* image segregation with the only exception here is the substitution of CM by *color connectionedness matrix* (CCM). Since we have assumed a uniform luminosity and no more than one *color* segment will be placed at the same physical location, the solution can be reduced to the temporal segregation of overlapping patterns. As far as the connectedness of different *color* patterns can be discerned via the fuzzy clustering approach described in Section 5.2, overlapping

Table 3
Convergence time for coupled oscillators

Initial phase shift (time-step)	No. of sampling points	Phase error < 5°	Phase error < 1°	Phase error < 0.5°	Phase error < 0.1°	Phase error < 0.01°
30	1	37.04	56.64	66.08	86.96	94.36
0	1	35.62	58.00	68.01	91.39	100.34
−30	1	41.94	62.29	73.82	98.29	107.90
Average		38.20	58.98	69.30	92.21	100.87
30	108	27.57	33.31	35.65	41.37	43.52
0	108	36.86	43.18	45.23	49.87	51.09
−30	108	26.20	33.54	36.80	40.96	43.08
Average		30.21	36.68	39.23	44.07	45.90

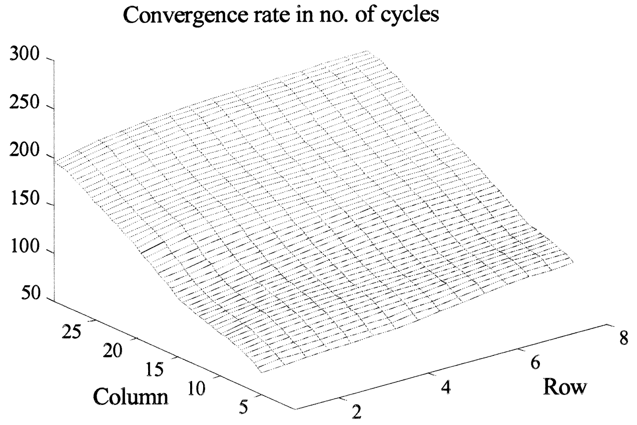

Fig. 7. Convergence speed of initialization for different sizes of oscillatory array.

color segments can be temporally segregated in accordance with the CCM Eq. (9).

As far as convergence time is concerned, the overhead for the calculation of the *color connectedness matrix* $CW_{ij,kl}$ is roughly equivalent to that for *CW*. Meanwhile, convergence time for *color* pattern segmentation is similar to that for *gray-level* pattern segmentation.

7. Discussions and conclusion

In this article, we have developed an oscillatory cellular neural network (OCNN) model for binary, *gray-level* and *color* image segregation. Global synchronization and global desynchronization of input images are accomplished by means of local connections confined to a 4-connected neighborhood. A novel time-delayed learning rule is conceived to precisely control the phases of individual oscillating

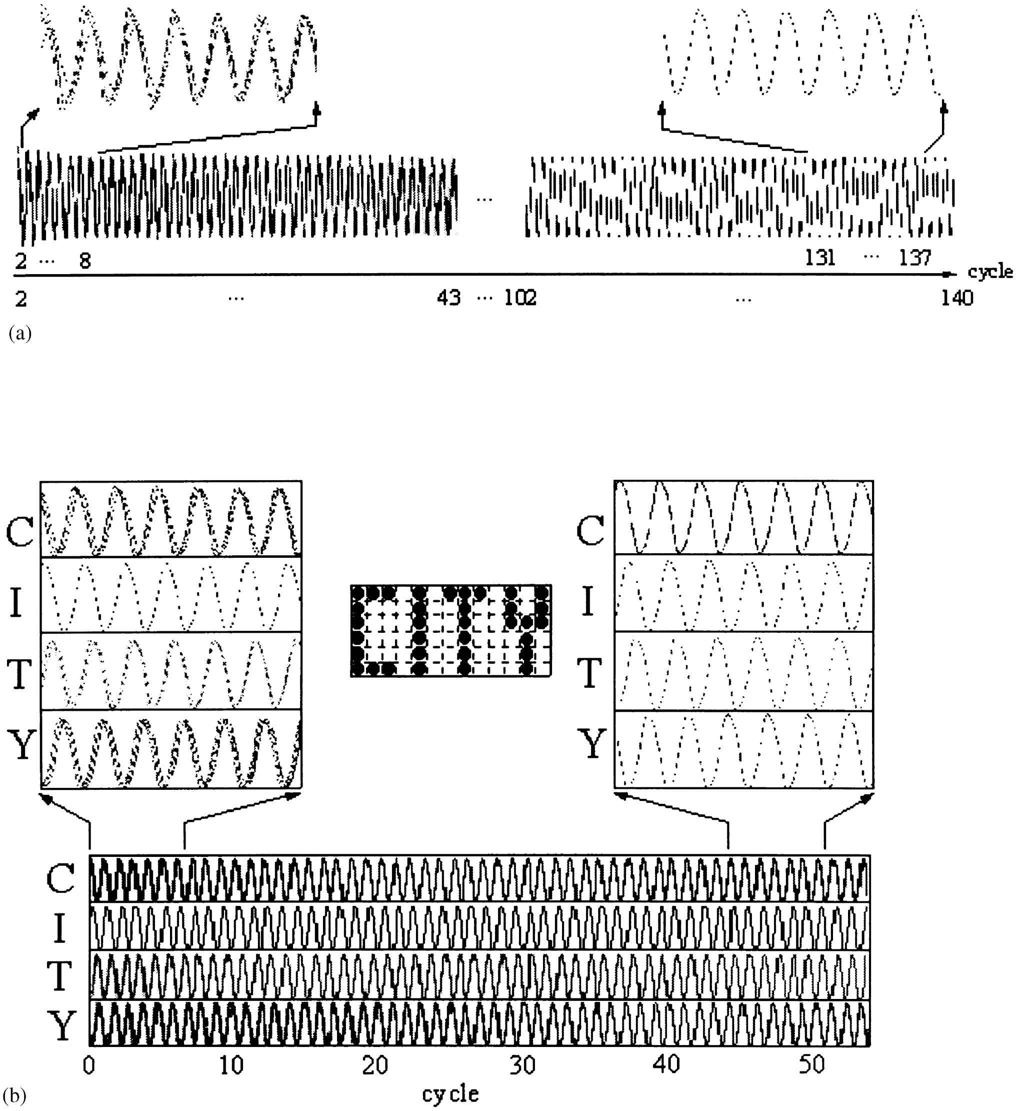

Fig. 8. (a) Initialization of OCNN (4th column) for sequential segregation. (b) Temporal segmentation of binary input image after initialization.

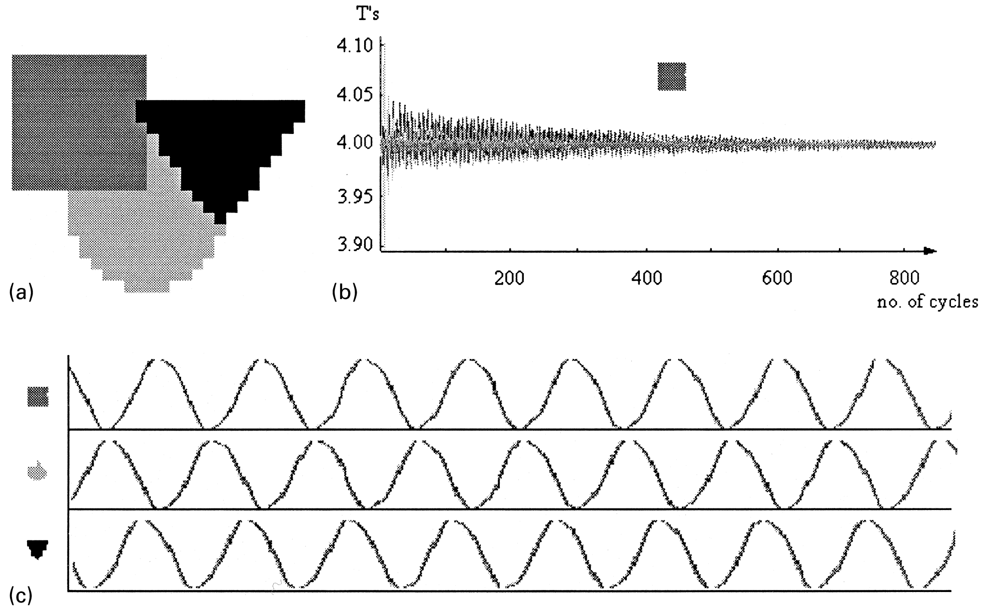

Fig. 9. (a) 25 × 32 *gray-level* image; (b) Convergence example of pattern '■'; (c) Converged phase patterns of 25 × 32 *gray-level* image.

segment. Uncorrelated segments in the input image are temporally segregated with phase difference in proportional to their Euclidean distance. Most importantly, the geometry relations among uncorrelated objects in the image scene are transformed into a temporal 'pop-out' sequence that retains the Euclidean distance information of the objects. In the article, we have introduced the idea of connectedness matrix for the *grouping* of correlated segments. Subsequent segregation process has shown to be much simplified, and the architecture of the OCNN is generalized to cope with binary, *gray-level* and *color* images. Simulation results have demonstrated the performance and validity of the proposed model even if some of the parameters are unmatched. Although the convergence speed of the network is quite slow compared to previous work, the simplicity and robustness of the model facilitate a large-scale integration of the network array. Currently we are investigating different learning methods to speed up the convergence time of the network.

Acknowledgements

We sincerely thank the two anonymous referees for their valuable comments. This project was supported in part by CityU strategic grant 7000749-570.

Appendix A. Stability behavior of neural oscillator

Fig. A1 depicts the solutions for the simultaneous equations $\mathrm{d}U_1/\mathrm{d}t = 0$ and $\mathrm{d}U_2/\mathrm{d}t = 0$. Interceptions of the two curves correspond to the equilibrium points of the neural oscillator. As shown in Fig. A1, there is only one interception, namely, the origin (0,0), which is the only stable state for non-oscillating.

Appendix B. Oscillating behavior of mutually coupled neurons

Consider the macroscopic behavior of two mutually coupled neurons shown in Fig. 2(a). The dynamics is governed by

$$f_1(U_1, U_2) = \dot{U}_1 = -U_1 + w_{12}H(U_2) + TH(U_1) + I_1, \quad (10)$$

$$f_2(U_1, U_2) = \dot{U}_2 = -U_2 + w_{21}H(U_1) + I_2. \quad (11)$$

Due to the dynamic interactions of the two neurons, the trajectories of the coupled neurons cannot be explicitly estimated. However, thanks to the theory of differential

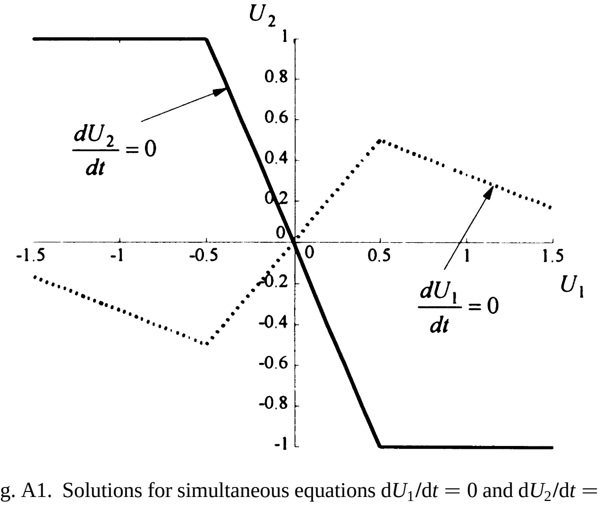

Fig. A1. Solutions for simultaneous equations $\mathrm{d}U_1/\mathrm{d}t = 0$ and $\mathrm{d}U_2/\mathrm{d}t = 0$.

equations, the behavior of the equilibrium state (U_1,U_2) can be examined by imposing constraints on the Jacobian matrix of (10) and (11) (Amari, 1972):

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial U_1} & \frac{\partial f_1}{\partial U_2} \\ \frac{\partial f_2}{\partial U_1} & \frac{\partial f_2}{\partial U_2} \end{bmatrix}. \tag{12}$$

If the trace Tr_J of J is negative and the determinant D_J of J is positive (the eigenvalues of J have negative real parts), then the system is stable, and equilibrium state exists. On the other hand, when any one of the aforementioned constraints does not satisfy, the system becomes oscillatory:

$$CI : Tr_J = \frac{\partial f_1}{\partial U_1} + \frac{\partial f_2}{\partial U_2} = -2 + T{\cdot}H'(0) < 0 \tag{13}$$

$$CII : D_J = \frac{\partial f_1}{\partial U_1}\frac{\partial f_2}{\partial U_2} - \frac{\partial f_1}{\partial U_2}\frac{\partial f_2}{\partial U_1}$$

$$= 1 - T{\cdot}H'(0) - w_{12}w_{21}(H'(0))^2 > 0. \tag{14}$$

Substituting the constraints imposed from Eq. (3), the condition for oscillatory behavior reduces to $T > 2/H'(0)$. As $H'(0) = 0.5$, the feedback synapse T should always be greater than 4.0 for oscillatory.

Appendix C. Dynamic range of neural oscillator

In practice, one has to pay attention to the dynamic range of the neurons for real-time applications so as not to saturate the output of components. After Eq. (2), the dynamic ranges of U_1 and U_2 can be estimated as

$$w_{12} + I_1 = U_1|_{\min.} \le U_1 \le U_1|_{\max} = T + I_1 \tag{15}$$

$$I_2 = U_2|_{\min.} \le U_2 \le U_2|_{\max.} = w_{21} + I_2, \tag{16}$$

where $w_{12} < 0$, $w_{21} > 0$, and $T > 0$. By substituting the constraints in Eq. (3), one obtains

$$-T_1 - I_1 \le U_1 \le T + I_1, \tag{17}$$

$$I_2 \le U_2 \le -I_2, \tag{18}$$

where $I_2 < 0$, $I_1 > 0$. In case when values of $T = 4.0$, $I_1 = -I_2 = 1.0$, the dynamic ranges of U_1 and U_2 can be calculated as

$$-5 \le U_1 \le 5, \tag{19}$$

$$-1 \le U_2 \le 1. \tag{20}$$

Obviously, the ranges of U_1 and U_2 are well within the typical supply voltage of common integrated circuits.

References

Amari, S. (1972). Characteristics of random nets of analog neuron-like elements. *IEEE Trans. On Systems, Man, and Cybernetics*, *SMC-2*, 643–657.

Beale, M., & Demuth, H. (1994). Fuzzy Systems Toolbox for use with MATLAB. PWS Publishing Company, Boston, Massachusetts.

Benoit, L., Foulloy, L., Galichet, S., & Mauris, G. (1994). Fuzzy sensor for the perception of color. *Proc. of IEEE International Conference on Fuzzy Systems*, *3*, 2008–2013.

Campbell, S. R., & Wang, D. L. (1996). Synchronization and desynchronization in a network of locally coupled Wilson–Cowan oscillators. *IEEE Trans. on Neural Networks*, *7*, 541–554.

Campbell, S. R., & Wang, D. L. (1998). Relaxation oscillators with time delay coupling. *Physica D*, *111*, 151–178.

Chua, L. O., & Yang, L. (1988). Cellular neural networks: theory. *IEEE Trans. On Circuits and Systems*, *35*, 1257–1272.

Engel, A. K., König, P., Kreiter, A. K., & Singer, W. (1991). Interhemispheric synchronization of oscillatory neuronal responses in cat visual cortex. *Science*, *252*, 1177–1179.

Kohler, R. (1981). A Segmentation System based on Thresholding. *Comput. Graphics Image Process.*, *15*, 319–338.

König, P., & Schillen, T. B. (1991). Stimulus-dependent assembly formation of oscillatory responses: I. synchronization. *Neural Computation*, *3*, 155–166.

Kurokawa, H., Ho, C. Y., & Mori, S. (1997). A learning rule of the oscillatory neural network for in-phase oscillation. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, *E80*, 1585–1594.

Malsburg, C. V., & Schneider, W. (1986). A neural cocktail-party processor. *Biol. Cybern.*, *54*, 29–40.

Schillen, T. B., & König, P. (1991). Stimulus-dependent assembly formation of oscillatory responses: II. desynchronization. *Neural Computation*, *3*, 166–178.

Skarda, C. A., & Freeman, W. J. (1987). How brains make chaos in order to make sense of the world. *Behavioral and Brain Sciences*, *10*, 161–195.

Szaflarish, D.M. (1998). How we see: the first steps of human vision, .

Terman, D., & Wang, D. L. (1995). Global competition and local cooperation in a network of neural oscillators. *Physica D*, *81*, 148–176.

Wang, D. L. (1995). Emergent synchrony in locally coupled neural oscillators. *IEEE Trans. on Neural Networks*, *6*, 941–948.

Wang, D. L., & Terman, D. (1997). Image segmentation based on oscillatory correlation. *Neural Computation*, *9*, 805–836.