33. Neurodynamics

Robert Kozma, Jun Wang, Zhigang Zeng

This chapter introduces basic concepts, phenomena, and properties of neurodynamic systems. it consists of four sections with the first two on various neurodynamic behaviors of general neurodynamics and the last two on two types of specific neurodynamic systems. The neurodynamic behaviors discussed in the first two sections include attractivity, oscillation, synchronization, and chaos. The two specific neurodynamics systems are memrisitve neurodynamic systems and neurodynamic optimization systems.

33.1 Dynamics of Attractor and Analog Networks

An attractor, as a well-known mathematical object, is central to the field of nonlinear dynamical systems (NDS) theory, which is one of the indispensable conceptual underpinnings of complexity science. An attractor is a set towards which a variable moves according to the dictates of a nonlinear dynamical system, evolves over time, such that points get close enough to the attractor, and remain close even if they are slightly disturbed. To well appreciate what an attractor is, some corresponding NDS notions, such as phase or state space, phase portraits, basins of attractions, initial conditions, transients, bifurcations, chaos, and strange attractors are needed to tame some of the unruliness of complex systems.

Most of us have at least some inkling of what *nonlinear* means, which can be illustrated by the most well-known and vivid example of the *butterfly effect* of a chaotic system that is nonlinear. It has prompted the use of the image of tiny air currents produced by a butterfly flapping its wing in Brazil, which are then amplified to the extent that they may influence the building up of a thunderhead in Kansas. Although no one can actually claim that there is such a linkage between Brazilian lepidopterological dynamics and climatology in the Midwest of the USA, it does serve to vividly portray nonlinearity in the extreme.

As the existence of both the nonlinearity and the capacity in passing through different regimes of stability and instability, the outcomes of the nonlinear dynamical system are unpredictable. These different regimes of a dynamical system are understood as different phases governed by different attractors, which means that the dynamics of each phase of a dynamical system are constrained within the circumscribed range allowable by that phase's attractors.

33.1.1 Phase Space and Attractors

To better grasp the idea of phase space, a time series and phase portrait have been used to represent the data points. Time series display changes in the values of variables on the y-axis (or the z-axis), and time on the x-axis as in a time series chart, however, the phase portrait plots the variables against each other and leaves time as an implicit dimension not explicitly plotted. Attractors can be displayed by phase portraits as the long-term stable sets of points of the dynamical system. This means that the locations in the phase portrait towards which the system's dynamics are attracted after transient phenomena have died down. To illustrate phase space and attractors, two examples are employed.

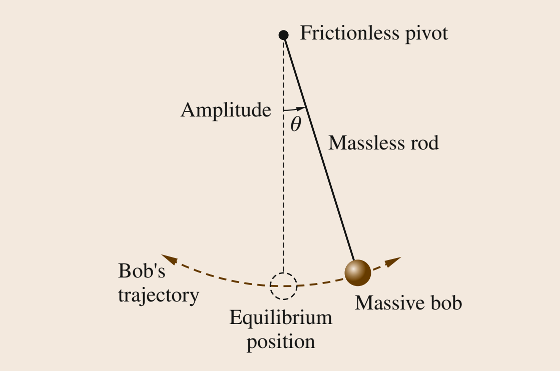

Fig. 33.1 Schematics of an unpushed swing

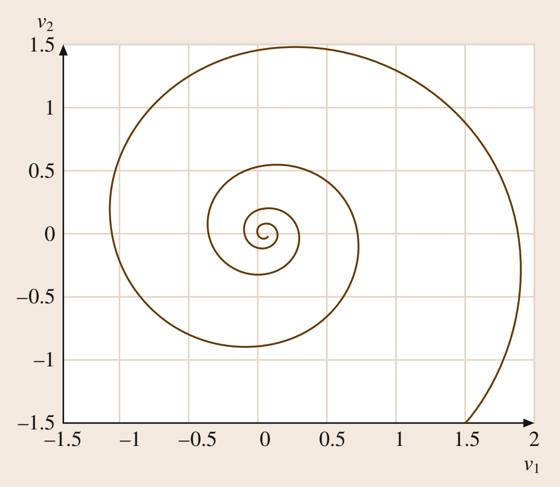

Fig. 33.2 Phase portrait and fixed point attractor of an unpushed swing

Imagine a child on a swing and a parent pulling the swing back. This gives a good push to make the child move forward. When the child is not moving forward, he will move backward on the swing as shown in Fig. 33.1. The unpushed swing will come to rest as shown in the times series chart and phase space. The time series show an oscillation of the speed of the swing, which slows down and eventually stops, that is its flat lines. In phase space, the swing's speed is plotted against the distance of the swing from the central point called a fixed point attractor since it attracts the system's dynamics in the long run. The fixed point attractor in the center of Fig. 33.2 is equivalent to the flat line in Fig. 33.3. The fixed point attractor is another way to see and say that an unpushed swing will come to a state of rest in the long term. The curved lines with arrows spiraling down to the center point in Fig. 33.2 display what is called the basin of attraction for the unpushed swing. These basins of attraction represent various initial conditions for the unpushed swing, such as starting heights and initial velocities.

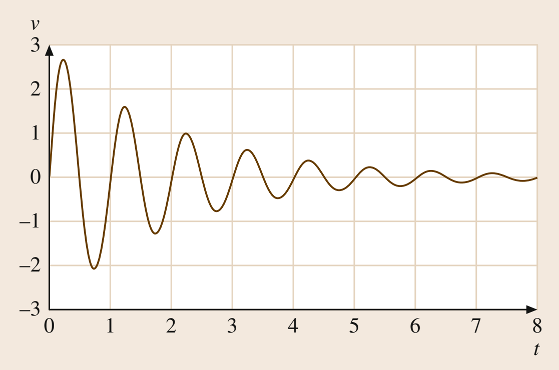

Fig. 33.3 Time series of the unpushed swing

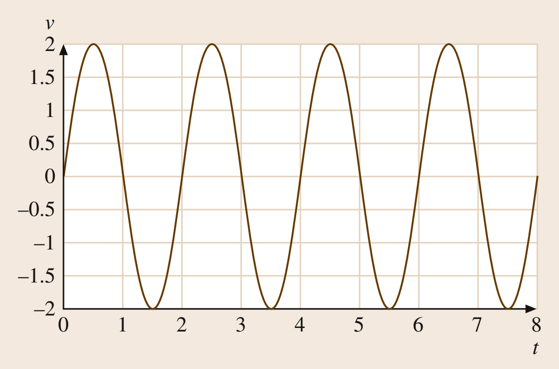

Fig. 33.4 Time series chart of the pushed swing

Now consider another type of a similar dynamical system, this time the swing is pushed each time it comes back to where the parent is standing. The time series chart of the pushed swing is shown in Fig. 33.4 as a continuing oscillation. This oscillation is around a zero value for y and is positive when the swing is going in one direction and negative when the swing is going in the other direction. As a phase space diagram, the states of variables against each other are shown in Fig. 33.5. The unbroken oval in Fig. 33.5 is a different kind of attractor from the fixed point one in Fig. 33.2. This attractor is well known as a limit cycle or periodic attractor of a pushed swing. It is called a limit cycle because it represents the cyclical behavior of the oscillations of the pushed swing as a limit to which the dynamical systems adheres under the sway of this attractor. It is periodic because the attractor oscillates around the same values, as the swing keeps going up and down until the s has a same heights from the lowest point. Such dynamical system can be called periodic for it has a repeating cycle or pattern.

By now, what we have learned about attractors can be summarized as follows: they are spatially displayed phase portraits of a dynamical system as it changes over the course of time, thus they represent the long-term dynamics of the system so that whatever the initial conditions represented as data points are, their trajectories in phase space fall within its basins of attraction, they are attracted to the attractor. In spite of wide usage in mathematics and science, as *Robinson* points out there is still no precise definition of an attractor, although many have been offered [33.2]. So he suggests thinking about an attractor as a phase portrait that attracts a large set of initial conditions and has some sort of minimality property, which is the smallest portrait in the phase space of the system. The attractor has the property of attracting the initial conditions after any initial transient behavior has died down. The minimality requirement implies the invariance or stability of the attractor. As a minimal object, the attractor cannot be split up into smaller subsets and retains its role as what dominates a dynamical system during a particular phase of its evolution.

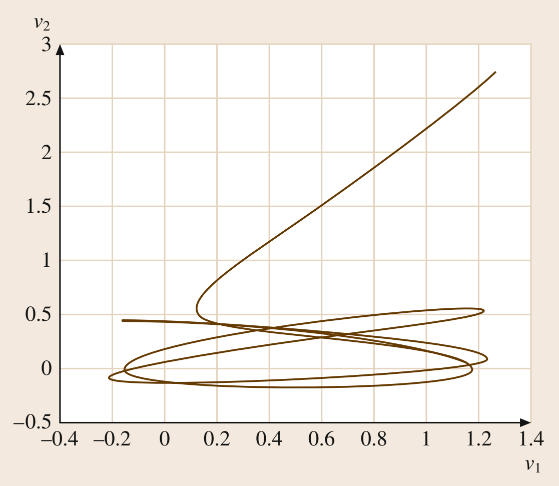

Fig. 33.5 Phase portrait and limit cycle attractor of a pushed swing (after [33.1])

33.1.2 Single Attractors of Dynamical Systems

Standard methods for the study of stability of dynamical systems with a unique attractor include the Lyapunov method, the Lasalles invariance principle, and the combination of thereof. Usually, given the properties of a (unique) attractor, we can realize a dynamical system with such an attractor.

Since the creation of the fundamental theorems of Lyapunov stability, many researchers have gone further and proved that most of the fundamental Lyapunov theories are reversible. Thus, from theory, this demonstrates that these theories are efficacious; i. e., there necessarily exists the corresponding Lyapunov function if the solution has some kind of stability. However, as for the construction of an appropriate V function for the determinant of stability, researchers are still interested. The difference between the existence and its construction is large. However, there is no general rule for the construction of the Lyapunov function. In some cases, different researchers have different methods for the construction of the Lyapunov function based on their experience and technique. Those, who can construct a good quality Lyapunov function, can get more useful information to demonstrate the effectiveness of their theories. Certainly, many successful Lyapunov functions have a practical background. For example, some equations inferred from the physical model have a clear physical meaning such as the mechanics guard system, in which the total sum of the kinetic energy and potential energy is the appropriate V function. The linear approximate method can be used; i. e., for the nonlinear differential equation, firstly find its corresponding linear differential equation's quadric form positive defined V function, then consider the nonlinear quality for the construction of a similar V function.

Grossberg proposed and studied additive neural networks because they add nonlinear contributions to the neuron activity. The additive neural network has been used for many applications since the 1960s [33.3, 4], including the introduction of self-organizing maps. In the past decades, neural networks as a special kind of nonlinear systems have received considerable attention. The study of recurrent neural networks with their various generalizations has been an active research area [33.5–17]. The stability of recurrent neural networks is a prerequisite for almost all neural network applications. Stability analysis is primarily concerned with the existence and uniqueness of equilibrium points and global asymptotic stability, global exponential stability, and global robust stability of neural networks at equilibria. In recent years, the stability analysis of recurrent neural networks with time delays has received much attention [33.18, 19]. Single attractors of dynamical systems are shown in Fig. 33.6.

33.1.3 Multiple Attractors of Dynamical Systems

Multistable systems have attracted extensive interest in both modeling studies and neurobiological research in recent years due to their feasibility to emulate and explain biological behavior [33.20–34]. Mathematically, multistability allows the system to have multiple fixed points and periodic orbits. As noted in [33.35], more than 25 years of experimental and theoretical work has indicated that the onset of oscillations in neurons and in neuron populations is characterized by multistability.

Multistability analysis is different from monostability analysis. In monostability analysis, the objective is to derive conditions that guarantee that each nonlinear system contains only one equilibrium point, and all the trajectories of the neural network converge to it. Whereas in multistability analysis, nonlinear systems are allowed to have multiple equilibrium points. Stable and unstable equilibrium points, and even periodic trajectories may co-exist in a multistable system.

The methods to study the stability of dynamical systems with a unique attractor include the Lyapunov method, the Lasalles invariance principle, and the combination of the two methods. One unique attractor can be realized by one dynamical system, but it is much more complicated for multiple attractors to be realized by one dynamical system or dynamical multisystems because of the compatibility, agreement, and behavior optimization among the systems. Generally, the usual global stability conditions are not adequately applicable to multistable systems. The latest results on multistability of neural networks can be found in [33.36–52]. It is shown in [33.45, 46] that the n-neuron recurrent neural networks with one step piecewise linear activation function can have 2^n locally exponentially stable equilibrium points located in saturation regions by partitioning the state space into 2^n subspaces. In [33.47], mul-

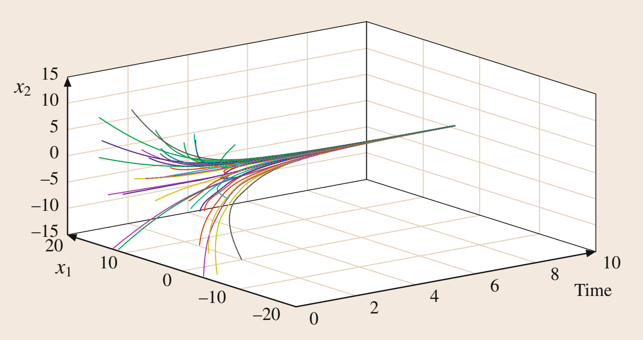

Fig. 33.6 Single attractors of dynamical systems

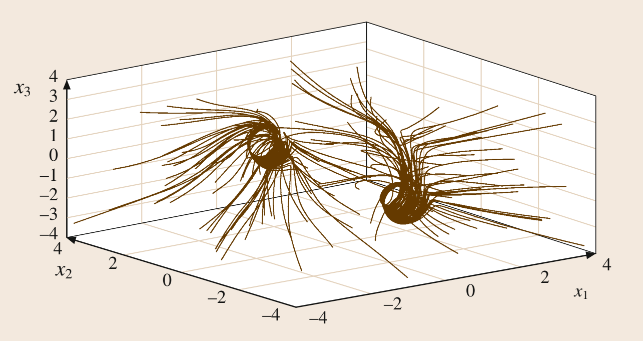

Fig. 33.7 Two limit cycle attractors of dynamical systems

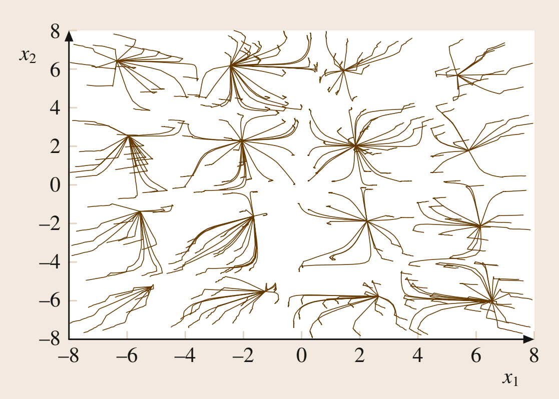

Fig. 33.8 2^4 equilibrium point attractors of dynamical systems

tistability of almost periodic solutions of recurrently connected neural networks with delays is investigated. In [33.48], by constructing a Lyapunov functional and using matrix inequality techniques, a delay-dependent multistability criterion on recurrent neural networks is derived. In [33.49], the neural networks with a class of nondecreasing piecewise linear activation functions with $2r$ corner points are considered. It is proved that the n-neuron dynamical systems can have and only have $(2r+1)^n$ equilibria under some conditions, of which $(r+1)^n$ are locally exponentially stable and others are unstable. In [33.50], some multistability properties for a class of bidirectional associative memory recurrent neural networks with unsaturation piecewise linear transfer functions are studied based on local inhibition. In [33.51], for two classes of general activation functions, multistability of competitive neural networks with time-varying and distributed delays is investigated by formulating parameter conditions and using inequality techniques. In [33.52], the existence of 2^n stable stationary solutions for general n-dimensional delayed neural networks with several classes of activation functions is presented through formulating parameter conditions motivated by a geometrical observation. Two limit cycle attractors and 2^4 equilibrium point attractors of dynamical systems are shown in Figs. 33.7 and 33.8, respectively.

33.1.4 Conclusion

In this section, we briefly introduced what attractors can be summarized as, and phase space and attractors. Furthermore, single-attractor and multiattractors of dynamical systems were also discussed.

33.2 Synchrony, Oscillations, and Chaos in Neural Networks

33.2.1 Synchronization

Biological Significance of Synchronization

Neurodynamics deals with dynamic changes of neural properties and behaviors in time and space at different levels of hierarchy in neural systems. The characteristic spiking dynamics of individual neurons is of fundamental importance. In large-scale systems, such as biological neural networks and brains with billions of neurons, the interaction among the connected neural components is crucial in determining collective properties. In particular, synchronization plays a critical role in higher cognition and consciousness experience [33.53–57]. Large-scale synchronization of neuronal activity arising from intrinsic asynchronous oscillations in local electrical circuitries of neurons are at the root of cognition. Synchronization at the level of neural populations is characterized next.

There are various dynamic behaviors of potential interest for neural systems. In the simplest case, the system behavior converges to a fixed point, when all major variables remain unchanged. A more interesting dynamic behavior emerges when the system behavior periodically repeats itself at period T, which will be described first. Such periodic oscillations are common in neural networks and are often caused by the presence of inhibitory neurons and inhibitory neural populations. Another behavior emerges when the system neither converges to a fixed point nor exhibits periodic oscillations, rather it maintains highly complex, chaotic dynamics. Chaos can be microscopic effect at the cellular level, or mesoscopic dynamics of neural populations or cortical regions. At the highest level of hierarchy, chaos can emerge as the result of large-scale, macroscopic effect across cortical areas in the brain.

Considering the temporal dynamics of a system of interacting neural units, limit cycle oscillations and chaotic dynamics are of importance. Synchronization in limit cycle oscillations is considered first, which illustrates the basic principles of synchronization. The extension to more complex (chaotic) dynamics is described in Sect. 33.2.3. Limit cycle dynamics is described as a cyclic repetition of the system's behavior at a given time period T. The cyclic repetition covers all characteristics of the system, e.g., microscopic currents, potentials, and dynamic variables; see, e.g., the Hodgkin–Huxley model of neurons [33.58]. Limit cycle oscillations can be described as a cyclic loop of the system trajectory in the space of all variables. The state of the system is given as a point on this trajectory at any given time instant. As time evolves, the point belonging to the system traverses along the trajectory. Due to the periodic nature of the movement, the points describing the system at time t and $t+T$ coincide fully. We can define a convenient reference system by selecting a center point of the trajectory and describe the motion as the vector pointing from the center to the actual state on the trajectory. This vector has an amplitude and phase in a suitable coordinate system, denoted as $\xi(t)$ and $\Phi(t)$, respectively. The evolution of the phase in an isolated oscillator with frequency ω_0 can be given as follows

$$\frac{\mathrm{d}\Phi(t)}{\mathrm{d}t} = \omega_0 \,. \tag{33.1}$$

Several types of synchronization can be defined. The strongest synchronization takes place when two (or multiple) units have identical behaviors. Considering limit cycle dynamics, strong synchronization means that the oscillation amplitude and phase are the same for all units. This means complete synchrony. An example of two periodic oscillators is given by the clocks shown in Fig. 33.9a–c [33.59]. Strong synchronization means that the two pendulums are connected with a rigid object forcing them move together. The lack of connection between the two pendulums means the absence of synchronization, i. e., they move completely independently. An intermediate level of synchrony may arise with weak coupling between the pendulums, such as a spring or a flexible band. Phase synchrony takes place when the amplitudes are not the same, but the phases of the oscillations could still coincide. Figure 33.9b–d depicts the case of out-of-phase synchrony, when the phases of the two oscillators are exactly the opposite.

Amplitude Measures of Synchrony

Denote by $a_j(t)$ the time signal produced by the individual units (neurons); $j = 1, \ldots, N$, and the overall signal of interacting units (A) is determined as

$$A(t) = 1/N \sum_{j=1}^{N} a_j(t) \,. \tag{33.2}$$

The variance of time series $A(t)$ is given as follows

$$\sigma_A^2 = \left\langle A^2(t) \right\rangle - \left\langle A(t) \right\rangle^2 \,. \tag{33.3}$$

Here $\langle f(t) \rangle$ denotes time averaging over a give time window. After determining the variance of the individual channels $\sigma_{A_j}^2$ based on (33.3), the synchrony χ_N in the system with N components is defined as follows

$$\chi_N^2 = \frac{\sigma_A^2}{1/N \sum_{i=1}^{N} \sigma_{A_i}^2} \,. \tag{33.4}$$

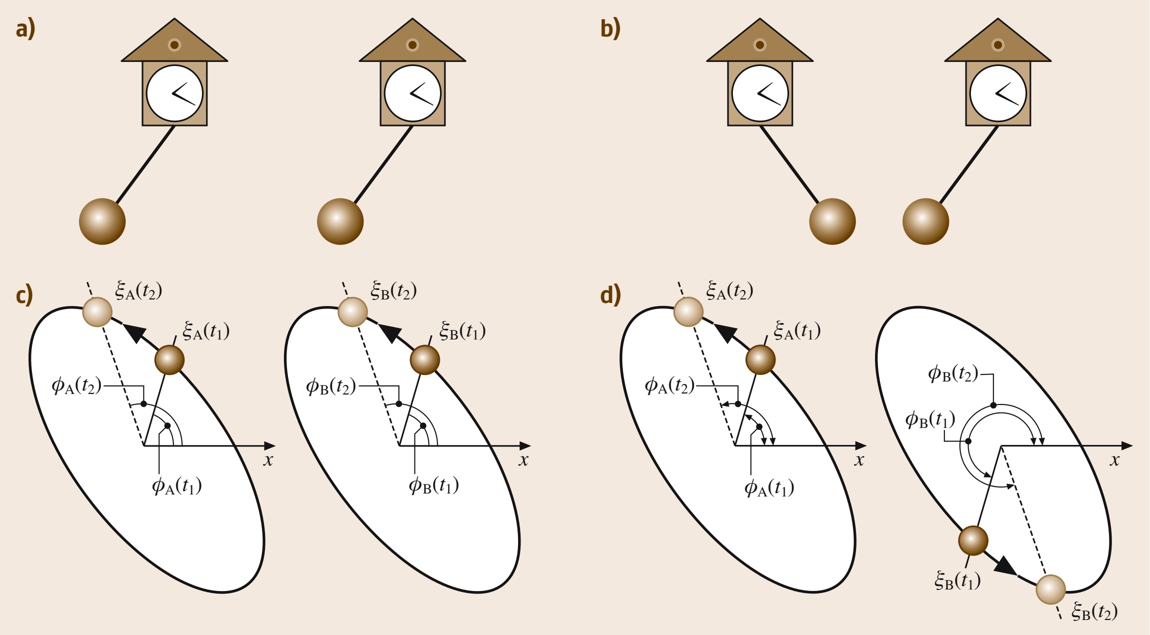

Fig. 33.9a–d Synchronization in pendulums, in phase and out of phase (after [33.59]). *Bottom plots*: Illustration of periodic trajectories, case of in-phase **(a–c)** and out-of-phase oscillations **(b–d)**

This synchrony measure has a nonzero value in synchronized and partially synchronized systems $0 < \chi_N < 1$, while $\chi_N = 0$ means the complete absence of synchrony in neural networks [33.60].

Fourier transform-based signal processing methods are very useful for the characterization of synchrony in time series, and they are widely used in neural network analysis. The Fourier transform makes important assumptions on the analyzed time series, including stationary or slowly changing statistical characteristics and ergodicity. In many applications these approximations are appropriate. In analyzing large-scale synchrony on brain signals, however, alternative methods are also justified. Relevant approaches include the Hilbert transform for rapidly changing brain signals [33.61, 62]. Here both Fourier and Hilbert-based methods are outlined and avenues for their applications in neural networks are indicated. Define the cross correlation function (CCF) between discretely sampled time series $x_i(t)$ and $x_i(t)$, $t = 1, \ldots, N$ as follows

$$\mathrm{CCF}_{ij}(\tau) = \frac{1}{T}\sum_{t=1}^{T-\tau}[x_i(t+\tau) - \langle x_i\rangle][x_j(t) - \langle x_j\rangle] \,. \tag{33.5}$$

Here $\langle x_i\rangle$ is the mean of the signal over period T, and it is assumed that $x_i(t)$ is normalized to unit variance. For completely correlated pairs of signals, the maximum of the cross correlation is 1, for uncorrelated signals it equals 0. The cross power spectral density $\mathrm{CPSD}_{ij}(\omega)$, cross spectrum for short, is defined as the Fourier transform of the cross correlation as follows: $\mathrm{CPSD}_{ij}(\omega) = \mathcal{F}(\mathrm{CCF}_{ij}(\tau))$. If $i = j$, i. e., the two channels coincide, then we talk about autocorrelation and auto power spectral density $\mathrm{APSD}_{ii}(\omega)$; for details of Fourier analysis, see [33.63]. Coherence γ^2 is defined by normalizing the cross spectrum by the autospectra

$$\gamma_{ij}^2(\omega) = \frac{|\mathrm{CPSD}_{ij}(\omega)|^2}{|\mathrm{APSD}_{ii}(\omega)||\mathrm{APSD}_{jj}(\omega)|} \,. \tag{33.6}$$

The coherence satisfies $0 \le \gamma^2(\omega) \le 1$ and it contains useful information on the frequency content of the synchronization between signals. If coherence is close to unity at some frequencies, it means that the two signals are closely related or synchronized; a coherence near zero means the absence of synchrony at those frequencies. Coherence functions provide useful information on synchrony in brain signals at various frequency bands [33.64]. For other information-theoretical characterizations, including mutual information and entropy measures.

Phase Synchronization

If the components of the neural network are weakly interacting, the synchrony evaluated using the amplitude measure χ in (33.4) may be low. There can still be a meaningful synchronization effect in the system, based on phase measures. Phase synchronization is defined as the global entrainment of the phases [33.65], which means a collective adjustment of their rhythms due to their weak interaction. At the same time, in systems with phase synchronization the amplitudes need not be synchronized. Phase synchronization is often observed in complex chaotic systems and it has been identified in biological neural networks [33.61, 65].

In complex systems, the trajectory of the system in the phase space is often very convoluted. The approach described in (33.1), i. e., choosing a center point for the oscillating cycle in the phase space with natural frequency ω_0, can be nontrivial in chaotic systems. In such cases, the Hilbert transform-based approach can provide a useful tool for the characterization of phase synchrony. Hilbert analysis determines the analytic signal and its instantaneous frequency, which can be used to describe phase synchronization effects. Considering time series $s(t)$, its analytic signal $z(t)$ is defined as follows [33.62]

$$z(t) = s(t) + \mathrm{i}\hat{s}(t) = A(t)\mathrm{e}^{\mathrm{i}\Phi(t)} \,. \tag{33.7}$$

Here $A(t)$ is the analytic amplitude, $\Phi(t)$ is the analytic phase, while $\hat{s}(t)$ is the Hilbert transform of $s(t)$, given by

$$\hat{s}(t) = \frac{1}{\pi}\mathrm{PV}\int_{-\infty}^{+\infty}\frac{s(t)}{t-\tau}\mathrm{d}\tau \,, \tag{33.8}$$

where PV stands for the principal value of the integral computed over the complex plane. The analytic signal and its instantaneous phase can be determined for an arbitrary broadband signal. However, the analytic signal has clear meaning only at a narrow frequency band, therefore, the bandpass filter should precede the evaluation of analytic signal in data with broad frequency content.

The Hilbert method of analytic signals is illustrated using actual local field potentials measured over rabbits with an array of chronically implanted intracranial electrodes [33.67]. The signals have been filtered in the theta band (3–7 Hz). An example of time series $s(t)$ is shown in Fig. 33.10a. The Hilbert transform $\hat{s}(t)$ is depicted in Fig. 33.10b in red, while blue color shows $s(t)$. Figure 33.10c shows the analytic

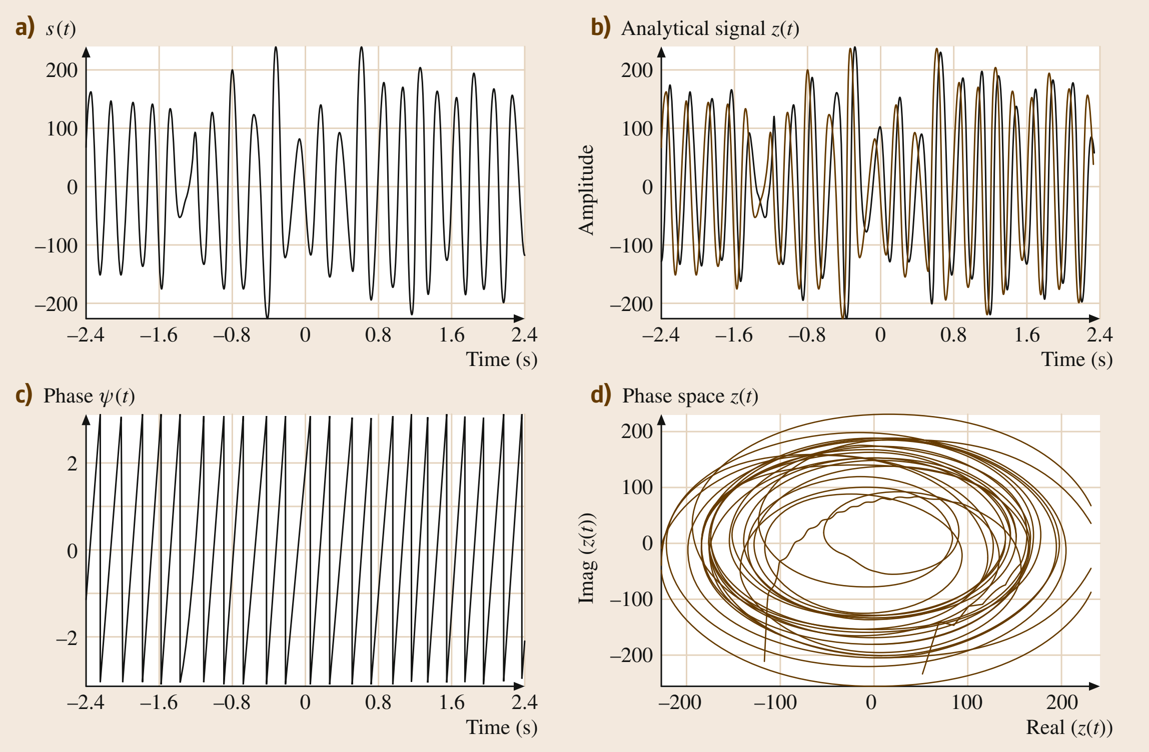

Fig. 33.10a–d Demonstration of the Hilbert analytic signal approach on electroencephalogram (EEG) signals (after [33.66]); **(a)** signal $s(t)$; **(b)** Hilbert transform $\hat{s}(t)$ (*red*) of signal $s(t)$ (*blue*); **(c)** instantaneous phase $\Phi(t)$; and analytic signal in complex plane $z(t)$

phase $\Phi(t)$, and Fig. 33.10d depicts the analytic $z(t)$ signal in the complex plane. Figure 33.11 shows the unwrapped instantaneous phase with bifurcating phase curves indicating desynchronization at specific time instances -1.3 s, -0.4 s, and 1 s. The plot on the right-hand side of Fig. 33.11 depicts the evolution of the instantaneous frequency in time. The frequency is around 5 Hz most of the time, indicating phase synchronization. However, it has very large dispersion at a few specific instances (desynchronization).

Synchronization between channels x and y can be measured using the phase lock value (PLV) defined as follows [33.61]

$$\mathrm{PLV}_{xy}(t) = \left| \frac{1}{T} \int\limits_{t-T/2}^{t+T/2} \mathrm{e}^{\mathrm{i}[\Phi_x(\tau)-\Phi_y(\tau)]} \mathrm{d}\tau \right| . \qquad (33.9)$$

PLV ranges from 1 to 0, where 1 indicates complete phase locking. PLV defined in (33.9) determines an average value over a time window of length T. Note that PLV is a function of t by applying the given sliding window. PLV is also the function of the frequency, which is being selected by the bandpass filter during the preprocessing phase. By changing the frequency band and time, the synchronization can be monitored at various conditions. This method has been applied productively in cognitive experiments [33.68].

Synchronization–Desynchronization Transitions

Transitions between neurodynamic regimes with and without synchronization have been observed and exploited for cognitive monitoring. The Haken–Kelso–Bunz (HKB) model is one of the prominent and elegant approaches providing a theoretical framework for synchrony switching, based on the observations related to bimanual coordination [33.69]. The HKB model invokes the concepts of metastability and multistability as fundamental properties of cognition. In the experiment, the subjects were instructed to follow the rhythm of a metronome with their index fingers in an anti-phase manner. It was observed that by increasing the metronome frequency, the subject spontaneously switched their anti-phase movement to in-phase at a certain oscillation frequency and maintained it thereon even if the metronome frequency was decreased again below the given threshold.

The following simple equation is introduced to describe the dynamics observed: $\mathrm{d}\Delta\Phi/\mathrm{d}t = -\sin(\Phi) - 2\varepsilon\sin(2\Phi)$. Here $\Delta\Phi = \phi_1 - \phi_2$ is the phase difference between the two finger movements, control parameter ε is related to the inverse of the introduced oscillatory frequency. The system dynamics is illustrated in Fig. 33.12 by the potential surface V, where stable fixed points correspond to local minima. For low oscillatory frequencies (high ε), there are stable equilibria at anti-phase conditions. As the oscillatory frequency increases (low ε) the dynamics transits to a state where only the in-phase equilibrium is stable.

Another practical example of synchrony-desynchrony transition in neural networks is given by image processing. An important basic task of neural networks is image segmentation, which is difficult to accomplish with excitatory nodes only. There is evidence that biological neural networks use inhibitory connections for completing basic pattern separation and integration tasks [33.70]. Synchrony between interacting neurons may indicate the recognition of an input. A typical neural network architecture implementing such a switch between synchronous and nonsynchronous states using local excitation and global inhibition is shown in Fig. 33.13. This system uses amplitude difference to

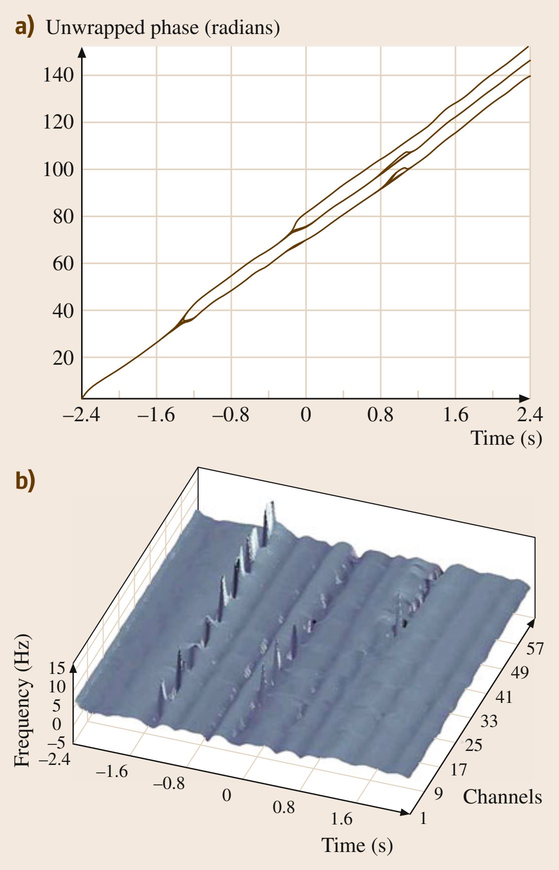

Fig. 33.11a,b Illustration of instantaneous phases; **(a)** unwrapped phase with bifurcating phase curves indicating desynchronization at specific time instances −1.3 s, −0.4 s, and 1 s; **(b)** evolution of instantaneous frequency in time

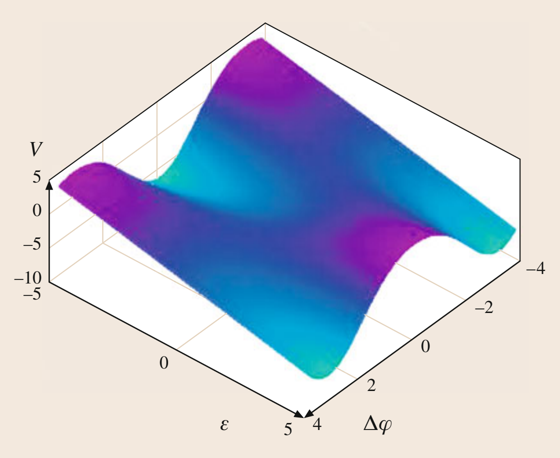

Fig. 33.12 Illustration of the potential surface (V) of the HKB system as a function of the phase difference in radians $\Delta\Phi$ and inverse frequency ε. The transition from anti-phase to in-phase behavior is seen as the oscillation frequency increases (ε decreases)

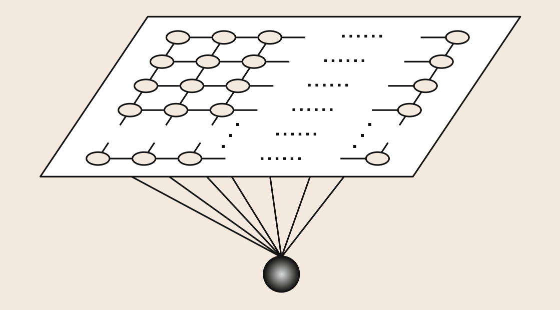

Fig. 33.13 Neural network with local excitation and a global inhibition node (*black*; after [33.70])

measure synchronization between neighboring neurons. Phase synchronization measures have been proposed as well to accomplish the segmentation and recognition tasks [33.71]. Phase synchronization provides a very useful tool for learning and control of the oscillations in weakly interacting neighborhoods.

33.2.2 Oscillations in Neural Networks

Oscillations in Brains

The interaction between opposing tendencies in physical and biological systems can lead to the onset of oscillations. Negative feedback between the system's components plays an important role in generating oscillations in electrical systems. Brains as large-scale bioelectrical networks consist of components oscillating at various frequencies. The competition between inhibitory and excitatory neurons is a basic ingredient of cortical oscillations. The intricate interaction between oscillators produces the amazingly rich oscillations that we experimentally observe as brain rhythms at multiple time scales [33.72, 73].

Oscillations occur in the brain at different time scales, starting from several milliseconds (high frequencies) to several seconds (low frequencies). One can distinguish between oscillatory components based on their frequency contents, including delta (1–4 Hz), theta (4–7 Hz), alpha (7–12 Hz), beta (12–30 Hz), and gamma (30–80 Hz) bands. The above separation of brain wave frequencies is somewhat arbitrary, however, they can be used as a guideline to focus on various activities. For example, higher cognitive functions are broadly assumed be manifested in oscillations in the higher beta and gamma bands.

Brain oscillations take place in time and space. A large part of cognitive activity happens in the cortex, which is a convoluted surface of the six-layer cortical sheet of gyri and sulci. The spatial activity is organized on multiple scales as well, starting from the neuronal level (μm), to granules (mm), cortical activities (several cm), and hemisphere-wide level (20 cm). The temporal and spatial scales are not independent, rather they delicately interact and modulate each other during cognition. Modern brain monitoring tools provide insight to these complex space–time processes [33.74].

Characterization of Oscillatory Networks

Oscillations in neural networks are synchronized activities of populations of neurons at certain well-defined frequencies. Neural systems are often modeled as the interaction of components which oscillate at specific, well-defined frequencies. Oscillatory dynamics can correspond to either microscopic neurons, to mesoscopic populations of tens of thousands neurons, or to macroscopic neural populations including billions of neurons. Oscillations at the microscopic level have been thoroughly studied using spiking neuron models, such as the Hodgkin–Huxley equation (HH). Here we focus on populations of neurons, which have some natural oscillation frequencies. It is meaningful to assume that the natural frequencies are not identical due to the diverse properties of populations in the cortex. Interestingly, the diversity of oscillations at the microscopic and mesoscopic levels can give rise to large-scale synchronous dynamics at higher levels. Such emergent oscillatory dynamics is the primary subject of this section.

Consider N coupled oscillators with natural frequencies ω_j; $j = 1, \ldots, N$. A measure of the synchronization in such systems is given by parameter R, which is often called the *order parameter*. This terminology was introduced by *Haken* [33.75] to describe the emergence of macroscopic order from disorder. The time-varying order parameter $R(t)$ is defined as [33.76]

$$R(t) = |1/N \times \Sigma_{j=1}^{N} \mathrm{e}^{\mathrm{i}\theta_j(t)}| \,. \tag{33.10}$$

Order parameter R provides a useful synchronization measure for coupled oscillatory systems. A common approach is to consider a globally coupled system, in which all the components interact with each other. This is the broadest possible level of interaction. The local coupling model represents just the other extreme limit, i. e., each node interacts with just a few others, which are called its direct neighbors. In a one-dimensional array, a node has two neighbors on its left and right, respectively (assuming periodic boundary conditions). In a two-dimensional lattice, a node has four direct neighbors, and so on. The size of the neighborhood can be expanded, so the connectivity in the network becomes more dense. There is of special interest in networks that have a mostly regular neighborhood with some further neighbors added by a selection rule from the whole network. The addition of remote or nonlocal connections is called rewiring, and the networks with rewiring are small world networks. They have been extensively studied in network theory [33.76–78]. Figure 33.14 illustrates local (top left) and global coupling (bottom right), as well as intermediate coupling, with the bottom left plot giving an example of network with random rewiring.

The Kuramoto Model

The Kuramoto model [33.79] is a popular approach to describe oscillatory neural systems. It implements mean-field (global) coupling. The synchronization in this model allows an analytical solution, which helps to interpret the underlying dynamics in clear mathematical terms [33.76]. Let θ_j and ω_j denote the phase and the inherent frequency of the i-th oscillator. The oscillators are coupled by a nonlinear interaction term depending on their pair-wise phase differences. In the Kuramoto model, the following sinusoidal coupling term has been used to model neural systems

$$\frac{\mathrm{d}\theta_j}{\mathrm{d}t} = \omega_j + \frac{K}{N}\Sigma_{j=1}^{N} \sin(\theta_i - \theta_j), \quad j = 1, \ldots, N\,. \tag{33.11}$$

Here K denotes the coupling strength and $K = 0$ means no coupling. The system in (33.11) and its generalizations have been studied extensively since its first introduction by *Kuramoto* [33.79]. Kuramoto used Lorenztian initial distribution of phases θ defined as: $L(\theta) = \gamma/\{\pi(\gamma^2 + (\omega - \omega_0)^2)\}$. This leads to the asymptotic solution $N \to$ inf and $t \to$ inf for order parameter R in simple analytic terms

$$R = \sqrt{1 - (K_\mathrm{c}/K)} \quad \text{if } K > K_\mathrm{c}, R = 0 \text{ otherwise}\,. \tag{33.12}$$

Here K_c denotes the critical coupling strength given by $K_\mathrm{c} = 2\gamma$. There is no synchronization between the oscillators if $K \leq K_\mathrm{c}$, and the synchronization becomes stronger as K increases at supercritical conditions $K > K_\mathrm{c}$, see Fig. 33.15. Inputs can be used to control synchronization, i. e., a highly synchronized system can be (partially) desynchronized by input stimuli [33.80, 81]. Alternatively, input stimuli can induce large-scale synchrony in a system with low level of synchrony, as evidenced by cortical observations [33.82].

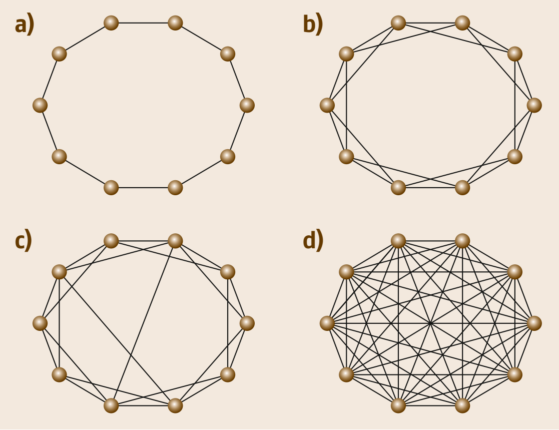

Fig. 33.14a–d Network architectures with various connectivity structures: **(a)** local, **(b)** and **(c)** are intermediate, and **(d)** global (mean-field) connectivity

Neural Networks as Dynamical Systems

A dynamical system is defined by its equation of motion, which describes the location of the system as a function of time t

$$\frac{\mathrm{d}X(t,\lambda)}{\mathrm{d}t} = F(X), \quad X \in \mathbb{R}^n\,. \tag{33.13}$$

Here X is the state vector describing the state of the system in the n-dimensional Euclidean space $X = X(x_1, \ldots, x_n) \in \mathbb{R}^n$ and λ is the vector of system parameters. Proper initial conditions must be specified and it is assumed that $F(X)$ is a sufficiently smooth nonlinear function. In neural dynamics it is often assumed that the state space is a smooth manifold, and the goal is to study the evolution of the trajectory of $X(t)$ in the state space as time varies along the interval $[t_0, T]$.

The Cohen–Grossberg (CG) equation is a general formulation of the motion of a neural network as a dynamical system with distributed time delays in the presence of inputs. The CG model has been studied

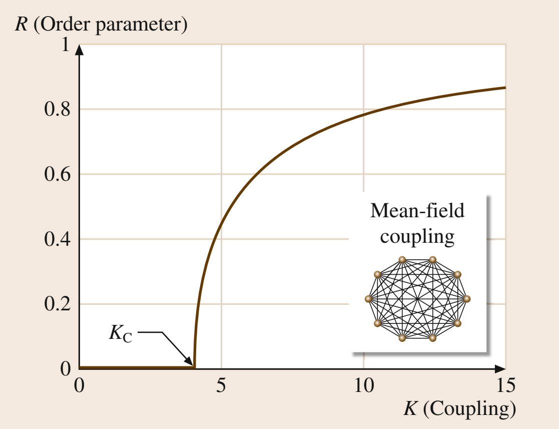

Fig. 33.15 Kuramoto model in the mean-field case. Dependence of order parameter R on the coupling strength K. Below a critical value K_c, the order parameter is 0, indicating the absence of synchrony; synchrony emerges for K above the critical value

thoroughly in the past decades and it served as a starting point for various other approaches. The general form of the CG model is [33.83]

$$\frac{\mathrm{d}z_i(t)}{\mathrm{d}t} = -a_i(x_i(t))\left[b_i(x_i(t)) - \sum_{j=1}^{N} a_{ij}f_j(x_j(t)) - \sum_{j=1}^{N} b_{ij}f_j(x_j(t-\tau_{ij})) + u_j\right], \quad i = 1,\ldots,N\,. \tag{33.14}$$

Here $X(t) = [x_1(t), x_2(t), \ldots, x_N(t)]^\mathsf{T}$ is the state vector describing a neural network with N neurons. Function $a_i(t)$ describes the amplification, $b_i(t)$ denotes a properly behaved function to guarantee that the solution remains bounded, $f_i(x)$ is the activation function, u_i denotes external input, a_{ij} and b_{ij} are components of the connection weight matrix and the delayed connection weight matrix, respectively, and τ_{ij} describes the time delays between neurons, $i, j = 1, \ldots, n$. The solution of (33.14) can be determined after specifying suitable initial conditions.

There are various approaches to guarantee the stability of the CG equation as it approaches its equilibria under specific constraints. Global convergence assuming symmetry of the connectivity matrix has been shown [33.83]. The symmetric version of a simplified CG model has become popular as the Hopfield or Hopfield–Tank model [33.84]. Dynamical properties of CG equation have been studied extensively, including asymptotic stability, exponential stability, robust stability, and stability of periodic bifurcations and chaos. Symmetry requirements for the connectivity matrix have been relaxed, still guaranteeing asymptotic stability [33.85]. CG equations can be employed to find the optimum solutions of a nonlinear optimization problem when global asymptotic stability guarantees the stability of the solution [33.86]. Global asymptotic stability of the CG neural network with time delay is studied using linear matrix inequalities (LMI). LMI is a fruitful approach for global exponential stability by constructing Lyapunov functions for broad classes of neural networks.

Bifurcations in Neural Network Dynamics

Bifurcation theory studies the behavior of dynamical systems in the neighborhood of bifurcation points, i. e., at points when the topology of the state space abruptly changes with continuous variation of a system parameter. An example of the state space is given by the folded surface in Fig. 33.16, which illustrates a cusp bifurcation point. Here $\lambda = [a, b]$ is a two-dimensional parameter vector, $X \in \mathbb{R}^1$ [33.87]. As parameter b increases, the initially unfolded manifold undergoes a bifurcation through a cusp folding with three possible values of state vector X. This is an example of pitchfork bifurcation, when a stable equilibrium point bifurcates into one unstable and two stable equilibria. The projection to the $a - b$ plane shows the cusp bifurcation folding with multiple equilibria. The presence of multiple equilibria provides the conditions for the onset of oscillatory states in neural networks. The transition from fixed point to limit cycle dynamics can described by bifurcation theory.

Neural Networks with Inhibitory Feedback

Oscillations in neural networks are typically due to delayed, negative feedback between neural population. Mean-field models are described first, starting with Wilson–Cowan (WC) oscillators, which are capable of producing limit cycle oscillations. Next, a class of more general networks with excitatory–inhibitory feedback are described, which can generate unstable limit cycle oscillations.

The Wilson–Cowan model is based on statistical analysis of neural populations in the mean-field limit, i. e., assuming that all components of the system fully interact [33.88, 89]. In the brain it may describe a single cortical column in one of the sensory cortices, which in turn interacts with other columns to generate synchronous or asynchronous oscillations, depending on the cognitive state. In its simplest manifestation, the WC model has one excitatory and one inhibitory com-

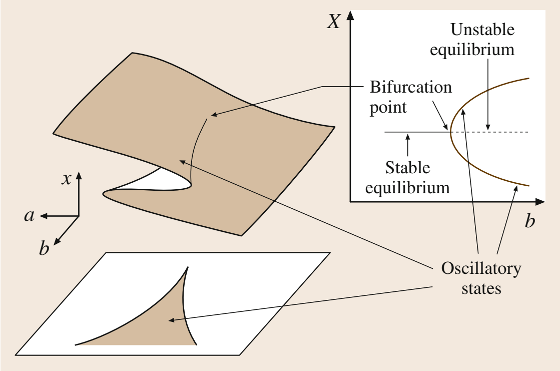

Fig. 33.16 Folded surface in the state space illustrating cusp bifurcation following (after [33.87]). By increasing parameter b, the stable equilibrium bifurcates to two stable and one unstable equilibria

ponent, with interaction weights denoted as w_{EE}, w_{EI}, w_{IE}, and w_{II}. Nonlinear function f stands for the standard sigmoid with rate constant a

$$\frac{\mathrm{d}X_{\mathrm{E}}}{\mathrm{d}t} = -X_{\mathrm{E}} + f(w_{\mathrm{EE}}X_{\mathrm{E}} + w_{\mathrm{IE}}X_{\mathrm{I}} + P_{\mathrm{E}}) \,, \tag{33.15}$$

$$\frac{\mathrm{d}X_{\mathrm{I}}}{\mathrm{d}t} = -X_{\mathrm{I}} + f(w_{\mathrm{EI}}X_{\mathrm{E}} + w_{\mathrm{II}}X_{\mathrm{I}} + P_{\mathrm{I}}) \,, \tag{33.16}$$

$$f(x) = 1/[1 + \mathrm{e}^{-ax}] \,. \tag{33.17}$$

P_{E} and P_{I} describe the effect of input stimuli through the excitatory and inhibitory nodes, respectively. The inhibitory weights are negative, while the excitatory ones are positive. The WC system has been extensively studied with dynamical behaviors including fixed point and oscillatory regimes. In particular, for fixed weight values, it has been shown that the WC system undergoes a pitchfork bifurcation by changing P_{E} or P_{I} input levels. Figure 33.17 shows the schematics of the two-node system, as well as the illustration of the oscillatory states following the bifurcation with parameters $w_{\mathrm{EE}} = 11.5$, $w_{\mathrm{II}} = -2$, $w_{\mathrm{EI}} = -w_{\mathrm{IE}} = -10$, and input values $P_{\mathrm{E}} = 0$ and $P_{\mathrm{I}} = -4$, with rate constant $a = 1$. Stochastic versions of the Wilson–Cowan oscillators have been extensively developed as well [33.90]. Coupled Wilson–Cowan oscillators have been used in learning models and have demonstrated applicability in a number of fields, including visual processing and pattern classification [33.91–93].

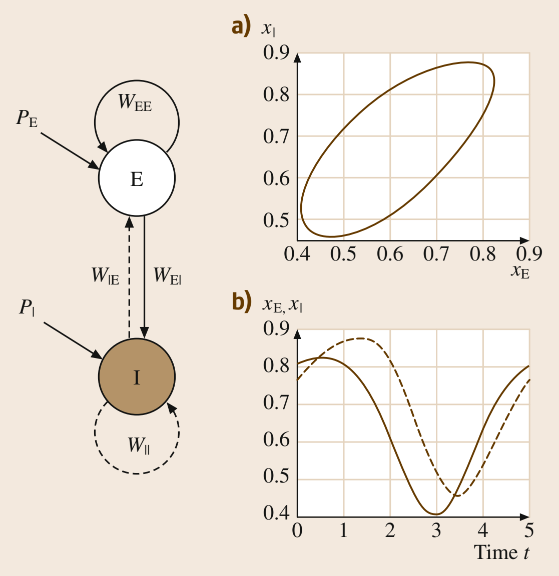

Fig. 33.17 Schematic diagram of the Wilson–Cowan oscillator with excitatory (E) and inhibitory (I) populations; *solid lines* show excitatory, *dashed* show inhibitory connections. The *right panels* show the trajectory in the phase space of $X_{\mathrm{E}} - X_{\mathrm{I}}$ and the time series of the oscillatory signals (after [33.90])

Oscillatory neural networks with interacting excitatory–inhibitory units have been developed in *Freeman* K sets [33.94]. That model uses an asymmetric sigmoid function $f(x)$ modeled based on neurophysiological activations and given as follows

$$f(x) = q\{1 - \exp(-[1/(q(\mathrm{e}^{x} - 1))])\} \,. \tag{33.18}$$

Here q is a parameter specifying the slope and maximal asymptote of the sigmoid curve. The sigmoid has unit gain at zero, and has maximum gain at positive x values due to its asymmetry, see (33.18). This property provides the opportunity for self-sustained oscillations without input at a wide range of parameters. Two versions of the basic oscillatory units have been studied, either one excitatory and one inhibitory unit, or two excitatory and two inhibitory units. This is illustrated in Fig. 33.18. Stability conditions of the fixed point and limit cycle oscillations have been identified [33.95, 96]. The system with two E and two I units has the advantage that it avoids self-feedback, which is uncharacteristic in biological neural populations. Interestingly, the extended system has an operating regime with an unstable equilibrium without stable equilibria. This condition leads to an inherent instability in a dynamical regime when the system oscillates without input. Oscillations in the unstable region have been characterized and conditions for sustained unstable oscillations derived [33.96]. Simulations in the region confirmed the existence of limit cycles in the unstable regime with highly irregular oscillatory shapes of the cycle, see Fig. 33.18, upper plot. Regions with regular limit cycle oscillations and fixed point oscillations have been identified as well, see Fig. 33.18, middle and bottom [33.97].

Spatiotemporal Oscillations in Heterogeneous NNs

Neural networks describe the collective behavior of populations of neurons. It is of special interest to study populations with a large-number of components having complex, nonlinear interactions. Homogeneous populations of neurons allow mathematical modeling in mean-field approximation, leading to oscillatory models such as the Wilson–Cowan oscillators and *Freeman* KII sets. Field models with heterogeneous structure and dynamic

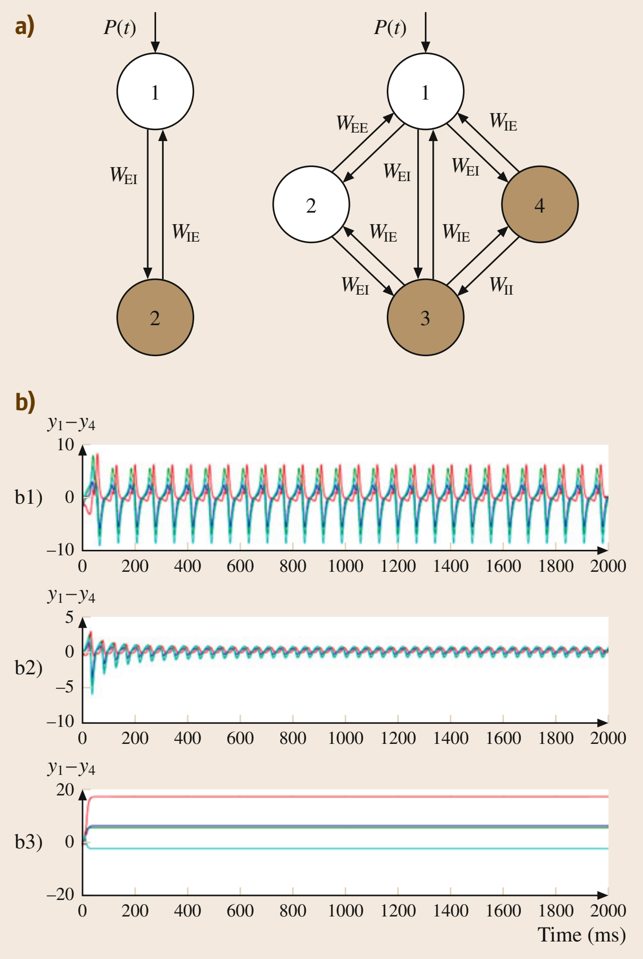

Fig. 33.18a,b Illustration of excitatory–inhibitory models. **(a)** *Left*: simplified model with one excitatory (E) and one inhibitory (I) node. *Right*: extended model with two E and two I nodes. **(b)** Simulations with the extended model with two E and two I nodes; $y_1 - y_4$ show the activations of the nodes; b1: limit cycle oscillations in the unstable regime; b2: oscillations in the stable limit cycle regime; b3: fixed point regime (after [33.97])

variables are of great interest as well, as they are the prerequisite of associative memory functions of neural networks.

A general mathematical formulation views the neuropil, the interconnected neural tissue of the cortex, as a dynamical system evolving in the phase space, see (33.13). Consider a population of spiking neurons each of which is modeled by a Hodgkin–Huxley equation. The state of a neuron at any time instant is determined by its depolarization potential, microscopic current, and spike timing. Each neuron is represented by a point in the state space given by the above coordinates comprising vector $X(t) \in \mathbb{R}^n$, and the evolution of a neuron is given with its trajectory the state space. Neuropils can include millions and billions of neurons; thus the phase space of the neurons contains a myriads of trajectories. Using the ensemble density approach of population modeling, the distribution of neurons in the state space at a given time t is described by a probability density function $p(X, t)$. The ensemble density approach models the evolution of the probability density in the state space [33.98]. One popular approach uses the Langevin formalism given next.

Field Theories of Neural Networks

Consider the stochastic process $X(t)$, which is described by the Langevin equation [33.99]

$$\mathrm{d}X(t) = \mu(X(t))\mathrm{d}t + \sigma(X(t))\mathrm{d}W(t) . \qquad (33.19)$$

Here μ and σ denote the drift and variance, respectively, and $\mathrm{d}W(t)$ is a Wiener process (Brown noise) with normally distributed increments. The probability density $p(X, t)$ of Langevin equation (33.19) satisfies the following form of the Fokker–Planck equation, after omitting higher-order terms

$$\frac{\partial p(X,t)}{\partial t} = -\sum_{i=1}^{n} \frac{\partial}{\partial x_i}[\mu_i(X)p(X,t)] + \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\partial^2}{\partial x_i \partial x_j}[D_{ij}(X)p(X,t)] . \qquad (33.20)$$

The Fokker–Planck equation has two components. The first one is a flow term containing drift vector $\mu_i(X)$, while the other term describes diffusion with diffusion coefficient matrix $D_{ij}(X, t)$. The Fokker–Planck equation is a partial differential equation (PDE) that provides a deterministic description of macroscopic events resulting from random microscopic events. The mean-field approximation describes time-dependent, ensemble average population properties, instead of keeping track of the behavior of individual neurons.

Mean-field models can be extended to describe the evolution of neural populations distributed in physical space. Considering the cortical sheet as a de facto continuum of the highly convoluted neural tissue (the neuropil), field theories of brains are developed using partial differential equations in space and time. The corresponding PDEs are wave equations. Consider a simple one-dimensional model to describe the dynamics of the current density $\Phi(x,t)$ as a macroscopic variable. In the simple case of translational invariance of the connectivity function between arbitrary two points of the domain with exponential decay, the following form of the wave equation is obtained [33.100]

$$\frac{\partial^2\Phi}{\partial t^2}+(\omega_0^2-v^2\Delta)\Phi+2\omega_0\frac{\partial\Phi}{\partial t}$$
$$=\left(\omega_0^2+\omega_0\frac{\partial}{\partial t}\right)S[\Phi(x,t)+P(x,t)]\,. \qquad (33.21)$$

Here $\Delta=\partial^2/x^2$ is the Laplacian in one dimension, $S(.)$ is a sigmoid transfer function for firing rates, $P(x,t)$ describes the effect of inputs; $\omega_0=v/\sigma$, where v is the propagation velocity along lateral axons, and σ is the spatial relaxation constant of the applied exponential decay function [33.100]. The model can be extended to excitatory–inhibitory components. An example of simulations with a one-dimensional neural field model incorporating excitatory and inhibitory neurons is given in Fig. 33.19 [33.101]. The figure shows the propagation of two traveling pulses and the emergence of transient complex behavior ultimately leading to an elevated firing rate across the whole tissue [33.101]. For recent developments in brain field models, see [33.90, 102].

Coupled Map Lattices for NNs

Spatiotemporal dynamics in complex systems has been modeled using coupled map lattices (CML) [33.103]. CMLs use continuous state space and discrete time and space coordinates. In other words, CMLs are defined on (finite or infinite lattices) using discrete time iterations. Using periodic boundary conditions, the array can be folded into a circle in one dimension, or into a torus for lattices of dimension 2 or higher. CML dynamics is described as follows

$$x_{n+1}(i)=(1-\varepsilon)f(x_n(i))+\varepsilon\frac{1}{K}\sum_{k=-K/2}^{K/2}f(x_n(i+k))\,, \qquad (33.22)$$

where $x_n(i)$ is the value of node i at iteration step n, $i=1,\dots,N$; N is the size of the lattice. Note that in (33.22) a periodic boundary condition applies. $f(.)$ is a nonlinear mapping function used in the iterations and ε is the coupling strength, $0\le\varepsilon\le1$. $\varepsilon=0$ means no coupling, while $\varepsilon=1$ is maximum coupling. The CML rule defined in (33.22) has two terms. The first term on the right-hand side is an iterative update of the i-th state, while the second term describes coupling between the units. Parameter K has a special role in coupled map lattices; it defines the size of the neighborhoods. $K=N$ describes mean-field coupling, while smaller K values belong to smaller neighborhoods. The geometry of the system is similar to the ones given in Fig. 33.14. The case of local neighborhood is the upper left diagram in Fig. 33.14, while mean-field coupling is the lower right diagram. Similar rules have been defined for higher-dimensional lattices.

CMLs exhibit very rich dynamic behavior, including fixed points, limit cycles, and chaos, depending on the choice of control parameters, ε, K, and function $f(.)$ [33.103, 104]. An example of the cubic sigmoid function

$$f(x,a)=ax^3-ax+x$$

is shown in Fig. 33.20, together with the bifurcation diagram with respect to parameter a. By increasing the value of parameter a, the map exhibits bifurcations from fixed point to limit cycle, and ultimately to the chaotic regime.

Complex CML dynamics has been used to design dynamic associative memory systems. In CML, each memory is represented as a spatially coherent oscillation and is learnt by a correlational learning rule operating in limit cycle or chaotic regimes. In such systems, both the memory capacity and the basin volume for each memory are larger in CML than in the Hopfield model employing the same learning rule [33.105]. CML chaotic memories reduce the problem of spurious memories, but they are not immune to it. Spurious memories prevent the system from exploiting its memory capacity to the fullest extent.

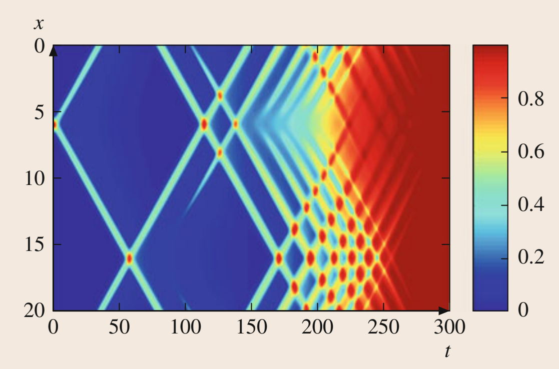

Fig. 33.19 Numerical simulations of a one-dimensional neural field model showing the interaction of two traveling pulses (after [33.101])

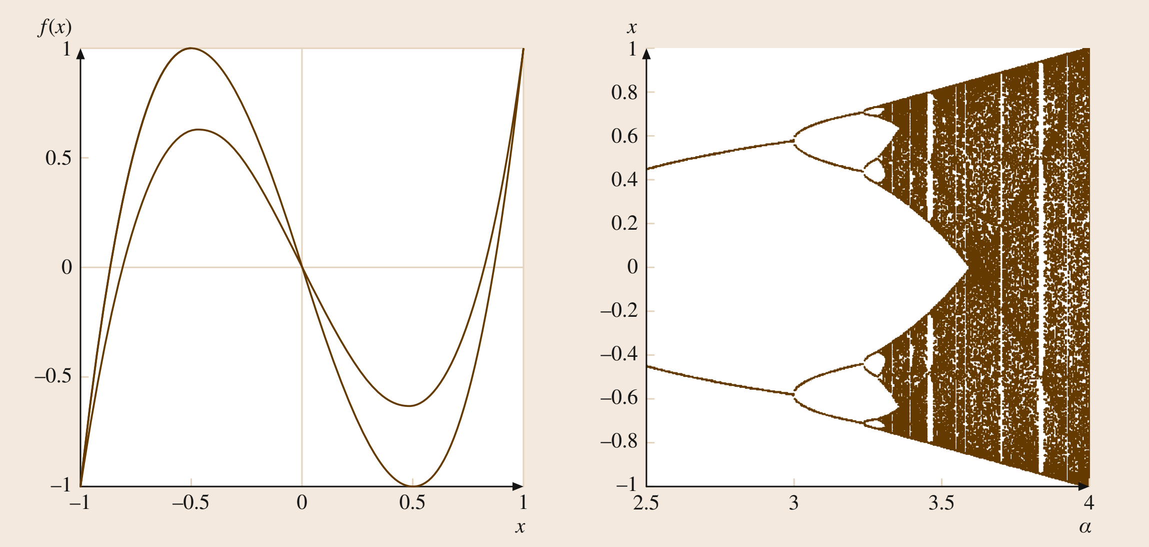

Fig. 33.20a,b Transfer function for CML: **(a)** shape of the cubic transfer function $f(x,a) = ax^3 - ax + x$; **(b)** bifurcation diagram over parameter a

Stochastic Resonance

Field models of brain networks develop deterministic PDEs (Fokker–Planck equation) for macroscopic properties based on a statistical description of the underlying stochastic dynamics of microscopic neurons. In another words, they are deterministic systems at the macroscopic level. Stochastic resonance (SR) deals with conditions when a bistable or multistable system exhibits strong oscillations under weak periodic perturbations in the presence of random noise [33.106]. In a typical SR situation, the weak periodic carrier wave is insufficient to cross the potential barrier between the equilibria of a multistable system. Additive noise enables the system to surmount the barrier and exhibit oscillations as it transits between the equilibria. SR is an example of processes when properly tuned random noise improves the performance of a nonlinear system and it is highly relevant to neural signal processing [33.107, 108].

A prominent example of SR in a neural network with excitatory and inhibitory units is described in [33.109]. In the model developed, the activation rate of excitatory and inhibitory neurons is described by μ_e and μ_i, respectively. The ratio $\alpha = \mu_e/\mu_i$ is an important parameter of the system. The investigated neural populations exhibit a range of dynamic behaviors, including convergence to fixed point, damped oscillations, and persistent oscillations. Figure 33.21 summarizes the main findings in the form of a phase diagram in the space of parameters α and *noise* level. The diagram contains three regions. Region I is at low noise levels and it corresponds to oscillations decaying to a fixed point at an exponential rate. Region II corresponds to high noise, when the neural activity exhibits damped oscillations as it approaches the steady state. Region III, however, demonstrates sustained oscillations for an intermediate level of noise. If a is above a critical value (see the tip of Region III),

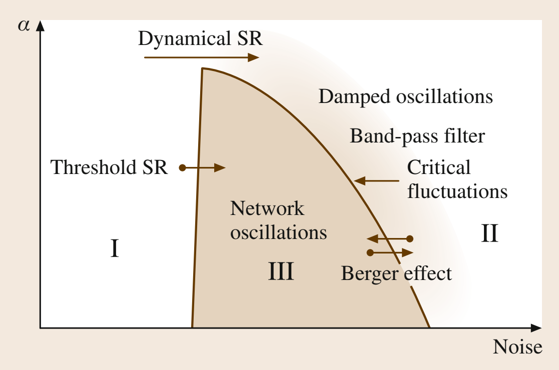

Fig. 33.21 Stochastic resonance in excitatory–inhibitory neural networks; α describes the relative strength of inhibition. Region I: fixed point dynamics. Region II: damped oscillatory regime. Region III: sustained periodic oscillations illustrating stochastic resonance (after [33.109])

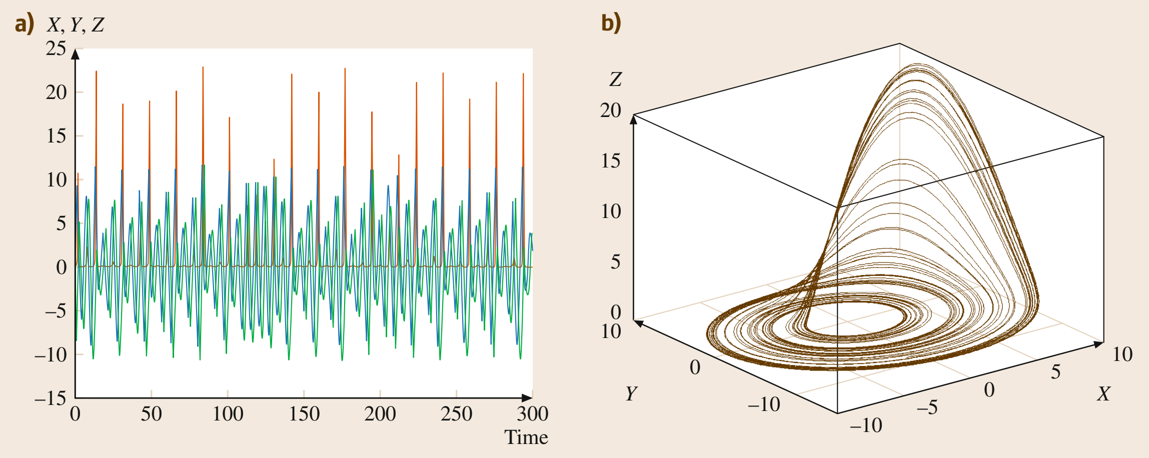

Fig. 33.22a,b Lorenz attractor in the chaotic regime; **(a)** time series of the variables X, Y, and Z; **(b)** butterfly-winged chaotic Lorenz attractor in the phase space spanned by variables X, Y, and Z

the activities in the steady state undergo a first-order phase transition at a critical noise level. The intensive oscillations in Region III at an intermediate noise level show that the output of the system (oscillations) can be enhanced by an optimally selected noise level.

The observed phase transitions may be triggered by neuronal avalanches, when the neural system is close to a critical state and the activation of a small number of neurons can generate an avalanche process of activation [33.110]. Neural avalanches have been described using self-organized criticality (SOC), which has been identified in neural systems [33.111]. There is much empirical evidence of the cortex conforming to the self-stabilized, scale-free dynamics with avalanches during the existence of some quasi-stable states [33.112, 113]. These avalanches maintain a metastable background state of activity.

Phase transitions have been studied in models with extended layers of excitatory and inhibitory neuron populations, respectively. A specific model uses random cellular neural networks to describe conditions with sustained oscillations [33.114]. The role of various control parameters has been studied, including noise level, inhibition, and rewiring. Rewiring describes long axonal connections to produce neural network architectures resembling connectivity patterns with short and long-range axons in the neuropil. By properly tuning the parameters, the system can reside in a fixed point regime in isolation, but it will switch to persistent oscillations under the influence of learnt input patterns [33.115].

33.2.3 Chaotic Neural Networks

Emergence of Chaos in Neural Systems

Neural networks as dynamical systems are described by the state vector $X(t)$ which obeys the equation of motion (33.13). Dynamical systems can exhibit fixed point, periodic, and chaotic behaviors. Fixed points and periodic oscillations, and transitions from one to the other through bifurcation dynamics has been described in Sect. 33.2.2. The trajectory of a chaotic system does not converge to a fixed point or limit cycle, rather it converges to a chaotic attractor. Chaotic attractors, or strange attractors, have the property that they define a fractal set in the state space, moreover, chaotic trajectories close to each other at some point, diverge from each other exponentially fast as time evolves [33.116, 117].

An example of the chaotic Lorenz attractor is shown in Fig. 33.22. The Lorenz attractor is defined by a system of three ordinary differential equations (ODEs) with nonlinear coupling, originally derived for the description of the motion of viscous flows [33.118]. The time series belonging to variables X, Y, Z are shown in Fig. 33.22a for parameters in the chaotic region, while the strange attractor is illustrated by the trajectory in the phase space, see Fig. 33.22b.

Chaotic Neuron Model

In chaotic neural networks the individual components exhibit chaotic behavior, and the goal is to study the order emerging from their interaction. Nerve membranes produce propagating action potentials in a highly nonlinear process which can generate oscillations and bifurcations to chaos. Chaos has been observed in the giant axons of squid and it has been used to study chaotic behavior in neurons. The Hodgkin–Huxley equations can model nonlinear dynamics in the squid giant axon with high accuracy [33.58]. The chaotic neuron model of *Aihara* et al. is an approximation of the Hodgkin–Huxley equation and it reproduces chaotic oscillations observed in the squid giant axon [33.119, 120]. The model uses the following simple iterative map

$$x(t+1) = kx(t) - \alpha f(x(t)) + a\,, \qquad (33.23)$$

where $x(t)$ is the state of the chaotic neuron at time t, k is a decay parameter, α characterizes refractoriness, a is a combined bias term, and $f(y(t))$ is a nonlinear transfer function. In the chaotic neuron model, the log sigmoid transfer function is used, see (33.17). Equation (33.23) combined with the sigmoid produces a piece-wise monotonous map, which generates chaos.

Chaotic neural networks composed of chaotic neurons generate spatio-temporal chaos and are able to retrieve previously learnt patterns as the chaotic trajectory traverses the state space. Chaotic neural networks are used in various information processing systems with abilities of parallel distributed processing [33.121–123]. Note that CMLs also consist of chaotic oscillators produced by a nonlinear local iterative map, like in chaotic neural networks. CMLs define a spatial relationship among their nodes to describe spatio-temporal fluctuations. A class of cellular neural networks combines the explicit spatial relationships similar to CMLs with detailed temporal dynamics using Cohen–Grossberg model [33.83] and it has been used successfully in neural network applications [33.124, 125].

Collective Chaos in Neural Networks

Chaos in neural networks can be an emergent macroscopic property stemming from the interaction of nonlinear neurons, which are not necessarily chaotic in isolation. Starting from the microscopic neural level up to the macroscopic level of cognition and consciousness, chaos plays an important role in neurodynamics [33.82, 126–129]. There are various routes to chaos in neural systems, including period-doubling bifurcations to chaos, chaotic intermittency, and collapse of a two-dimensional torus to chaos [33.130, 131].

Chaotic itinerancy is a special form of chaos, which is between ordered dynamics and fully developed chaos. Chaotic itinerancy describes the trajectory through high-dimensional state space of neural activity [33.132]. In chaotic itinerancy the chaotic system is destabilized to some degree but some traces of the trajectories remain. This describes an itinerant behavior between the states of the system containing destabilized attractors or *attractor ruins*, which can be fixed point, limit cycle, torus, or strange attractor with unstable directions. Dynamical orbits are attracted to a certain attractor ruin, but they leave via an unstable manifold after a (short or long) stay around it and move toward another attractor ruin. This successive chaotic transition continues unless a strong input is received. A schematic diagram is shown in Fig. 33.23, where the trajectory of a chaotic itinerant system is shown visiting attractor ruins. Chaotic itinerancy is associated with perceptions and memories, the chaos between the attractor ruins is related to searches, and the itinerancy is associated with sequences in thinking, speaking, and writing.

Frustrated chaos is a dynamical system in a neural network with a global attractor structure when local connectivity patterns responsible for stable oscillatory behaviors become intertwined, leading to mutually competing attractors and unpredictable itinerancy between brief appearances of these attractors [33.133]. Similarly to chaotic itinerancy, frustrated chaos is related to destabilization of the dynamics and it generates itinerant, wavering oscillations between the orbits of the network, the trajectories of which have been stable with the original connectivity pattern. Frustrated chaos is shown to belong to the family of intermittency type of chaos [33.134, 135].

To characterize chaotic dynamics, tools of statistical time series analysis are useful. The studies may involve time and frequency domains. Time domain analysis

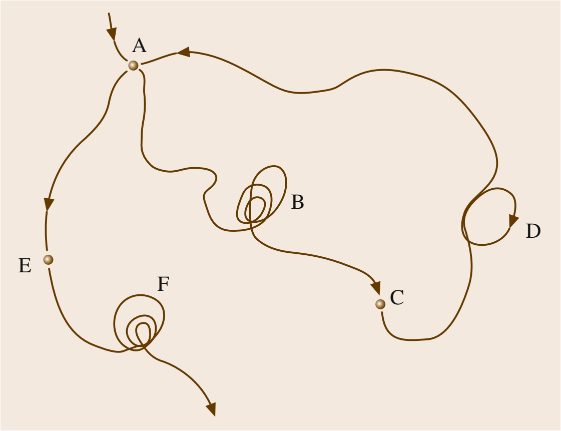

Fig. 33.23 Schematic illustration of itinerant chaos with a trajectory visiting attractor ruins (after [33.132])

includes attractor reconstruction, i. e., the attractor is depicted in the state space. Chaotic attractors have fractal dimensions, which can be evaluated using one of the available methods [33.136–138]. In the case of low-dimensional chaotic systems, the reconstruction can be illustrated using two or three-dimensional plots. An example of attractor reconstruction is given in Fig. 33.22 for the Lorenz system with three variables. Attractor reconstruction of a time series can be conducted using time-delay coordinates [33.139].

Lyapunov spectrum analysis is a key tool in identifying and describing chaotic systems. Lyapunov exponents measure the instability of orbits in different directions in the state space. It describes the rate of exponential divergence of trajectories that were once close to each other. The set of corresponding Lyapunov exponents constitutes the Lyapunov spectrum. The maximum Lyapunov exponent Λ^* is of crucial importance; as a positive leading Lyapunov exponent $\Lambda^* > 0$ is the hallmark of chaos. $X(t)$ describes the trajectory of the system in the phase space starting from $X(0)$ at time $t = 0$. Denote by $X_{\Delta x_0}(t)$ the perturbed trajectory starting from $[X(0) + \Delta x_0]$. The leading Lyapunov exponent can be determined using the following relationship [33.140]

$$\Lambda^* = \lim_{\substack{t \to \infty \\ \Delta x_0 \to 0}} t^{-1} \ln[|X_{\Delta x_0}(t) - X(t)|/|\Delta x_0|] \,, \tag{33.24}$$

where $\Lambda^* < 0$ corresponds to convergent behavior, $\Lambda^* = 0$ indicates periodic orbits, and $\Lambda^* > 0$ signifies chaos. For example, the Lorenz attractor has $\Lambda^* = 0.906$, indicating strong chaos (Fig. 33.24). Equation (33.24) measures the divergence for infinitesimal perturbations in the limit of infinite time series. In practical situations, especially for short time series, it is often difficult to distinguish weak chaos from random perturbations. One must be careful with conclusions about the presence of chaos when Λ^* has a value close to zero. Lyapunov exponents are widely used in brain monitoring using electroencephalogram (EEG) analysis, and various methods are available for characterization of normal and pathological brain conditions based on Lyapunov spectra [33.141, 142].

Fourier analysis conducts data processing in the frequency domain, see (33.5) and (33.6). For chaotic signals, the shape of the power spectra is of special interest. Power spectra often show $1/f^\alpha$ power law behavior in log–log coordinates, which is the indication of scale-free system and possibly chaos. Power-law scaling in systems at SOC is suggested by a linear decrease in log power with increasing log frequency [33.143]. Scaling properties of criticality facilitate the coexistence of spatially coherent cortical activity patterns for a duration ranging from a few milliseconds to a few seconds. Scale-free behavior characterizes chaotic brain activity both in time and frequency domains. For completeness, we mention the Hilbert space analysis as an alternative to Fourier methods. The analytic signal approach based on Hilbert analysis is widely used in brain monitoring.

Emergent Macroscopic Chaos in Neural Networks

Freeman's K model describes spatio-temporal brain chaos using a hierarchical approach. Low-level K sets were introduced in the 1970s, named in the honor of Aharon Kachalsky, an early pioneer of neural dynamics [33.82, 94]. K sets are multiscale models, describing an increasing complexity of structure and dynamics. K sets are mesoscopic models and represent an intermediate level between microscopic neurons and macroscopic brain structures. K-sets are topological specifications of the hierarchy of connectivity in neural populations in brains. K sets describe the spatial patterns of phase and amplitude of the oscillations generated by neural populations. They model observable fields of neural activity comprising electroencephalograms (EEGs), local field potentials (LFPs), and magnetoencephalograms (MEGs) [33.144]. K sets form a hierarchy for cell assemblies with components starting from K0 to KIV [33.145, 146].

K0 sets represent noninteractive collections of neurons forming cortical microcolumns; a K0 set models a neuron population of $\approx 10^3 - 10^4$ neurons. K0 models dendritic integration in average neurons and an asymmetric sigmoid static nonlinearity for axon transmission. The K0 set is governed by a point attractor with zero output and stays at equilibrium except when perturbed. In the original K-set models, K0s are described by a state-dependent, linear second-order ordinary differential equation (ODE) [33.94]

$$ab\, \mathrm{d}^2X(t)/\mathrm{d}t^2 + (a+b)\, \mathrm{d}X(t)/\mathrm{d}t + P(t) = U(t) \,. \tag{33.25}$$

Here a and b are biologically determined time constants. $X(t)$ denotes the activation of the node as a function of time. $U(t)$ includes an asymmetric sigmoid function $Q(x)$, see (33.18), acting on the weighted sum of activation from neighboring nodes and any external input.

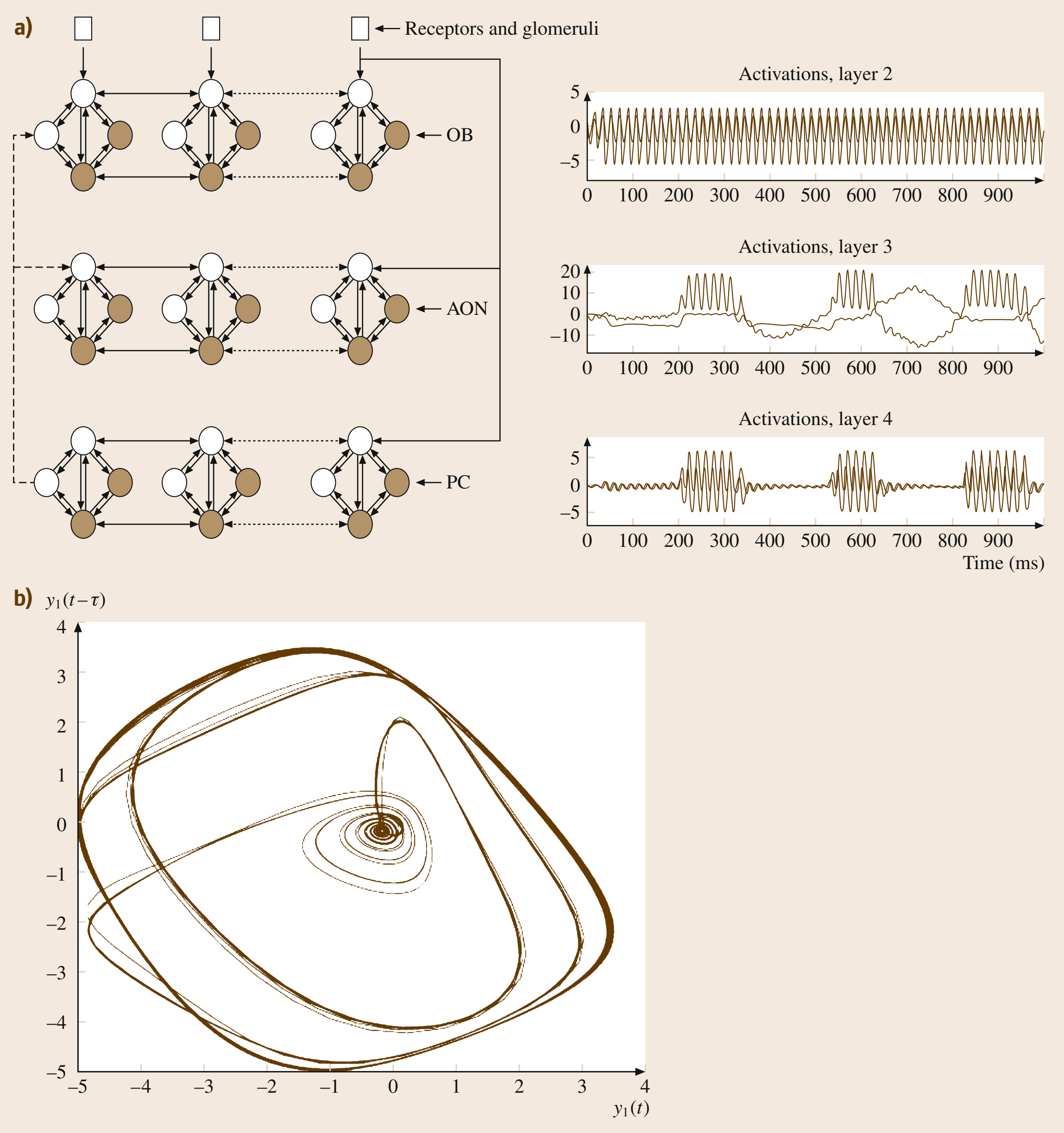

Fig. 33.24a–c KIII diagram and behaviors; **(a)** 3 double layer hierarchy of KIII and time series over each layer, exhibiting intermittent chaotic oscillations, **(b)** phase space reconstruction using delayed time coordinates

KI sets are made of interacting K0 sets, either excitatory or inhibitory with positive feedback. The dynamics of KI is described as convergence to a nonzero fixed point. If KI has sufficient functional connection density, then it is able to maintain a nonzero state of background activity by mutual excitation (or inhibition). KI typically operates far from thermodynamic equilibrium. Neural interaction by stable mutual excitation (or mutual inhibition) is fundamental to understanding brain dynamics. KII sets consists of interacting excitatory and inhibitory KI sets with negative feedback. KII sets are responsible for the emergence of limit cycle oscillation due to the negative feedback between the neural populations. Transitions from point attractor to limit cycle attractor can be achieved through a suitable level of feedback gain or by input stimuli, see Fig. 33.18.

KIII sets made up of multiple interacting KII sets. Examples include the sensory cortices. KIII sets generate broadband, chaotic oscillations as background activity by combined negative and positive feedback among several KII populations with incommensurate frequencies. The increase in nonlinear feedback gain that is driven by input results in the destabilization of the background activity and leads to the emergence of a spatial amplitude modulation (AM) pattern in KIII. KIII sets are responsible for the embodiment of meaning in AM patterns of neural activity shaped by synaptic interactions that have been modified through learning in KIII layers. The KIII model is illustrated in Fig. 33.24 with three layers of excitatory–inhibitory nodes. In Fig. 33.24a the temporal dynamics is illustrated in each layer, while Fig. 33.24b shows the phase space reconstruction of the attractor. This is a chaotic behavior resembling the dynamics of the Lorenz attractor in Fig. 33.22. KIV sets are made up of interacting KIII units to model intentional neurodynamics of the limbic system. KIV exhibits global phase transitions, which are the manifestations of hemisphere-wide cooperation through intermittent large-scale synchronization. KIV is the domain of Gestalt formation and preafference through the convergence of external and internal sensory signals leading to intentional action [33.144, 146].

Properties of Collective Chaotic Neural Networks

KIII is an associative memory, encoding input data in spatio-temporal AM patterns [33.147, 148]. KIII chaotic memories have several advantages as compared to convergent recurrent networks:

1. They produce robust memories based on relatively few learning examples even in noisy environment.
2. The encoding capacity of a network with a given number of nodes is exponentially larger than their convergent counterparts.
3. They can recall the stored data very quickly, just as humans and animals can recognize a learnt pattern within a fraction of a second.

The recurrent Hopfield neural network can store an estimated $0.15N$ input patterns in stable attractors, where N is the number of neurons [33.84]. Exact analysis by *Mceliece* et al. [33.149] shows that the memory capacity of the Hopfield network is $N/(4\log N)$. Various generalizations provide improvements over the initial memory gain [33.150, 151]. It is of interest to evaluate the memory capacity of the KIII memory. The memory capacity of chaotic networks which encode input into chaotic attractors is, in principle, exponentially increased with the number of nodes. However, the efficient recall of the stored memories is a serious challenge. The memory capacity of KIII as a chaotic associative memory device has been evaluated with noisy input patterns. The results are shown in Fig. 33.25, where the performance of Hopfield and KIII memories are compared; the top two plots are for Hopfield nets, while the lower two figures describe KIII results [33.152]. The light color shows recognition rate close to 100%, while the dark color means poor recognition approaching 0. The right-hand column has higher noise levels. The Hopfield network shows the well-known linear gain curve ≈ 0.15. The KIII model, on the other hand, has a drastically better performance. The boundary separating the correct and incorrect classification domains is superlinear; it has been fitted with as a fifth-order polynomial.

Cognitive Implications of Intermittent Brain Chaos

Developments in brain monitoring techniques provide increasingly detailed insights into spatio-temporal neurodynamics and neural correlates of large-scale cognitive processing [33.74, 153–155]. Brains as large-scale dynamical systems have a basal state, which is a high-dimensional chaotic attractor with a dynamic trajectory wandering broadly over the attractor landscape [33.82, 126]. Under the influence of external stimuli, cortical dynamics is destabilized and condenses intermittently to a lower-dimensional, more organized subspace. This is the act of perception when the subject identifies the stimulus with a meaning in the context of its previous experience. The system stays intermittently in the condensed, more coherent state, which gives rise to a spatio-temporal AM activity pattern corresponding to the stimulus in the given context. The AM pattern is meta-stable and it disintegrates as the system returns to the high-dimensional chaotic basal state (less synchrony) Brain dynamics is described as a sequence of phase transitions with intermittent synchronization-desynchronization effects. The rapid emergence of synchronization can be initiated by (Hebbian) neural assemblies that lock into synchronization

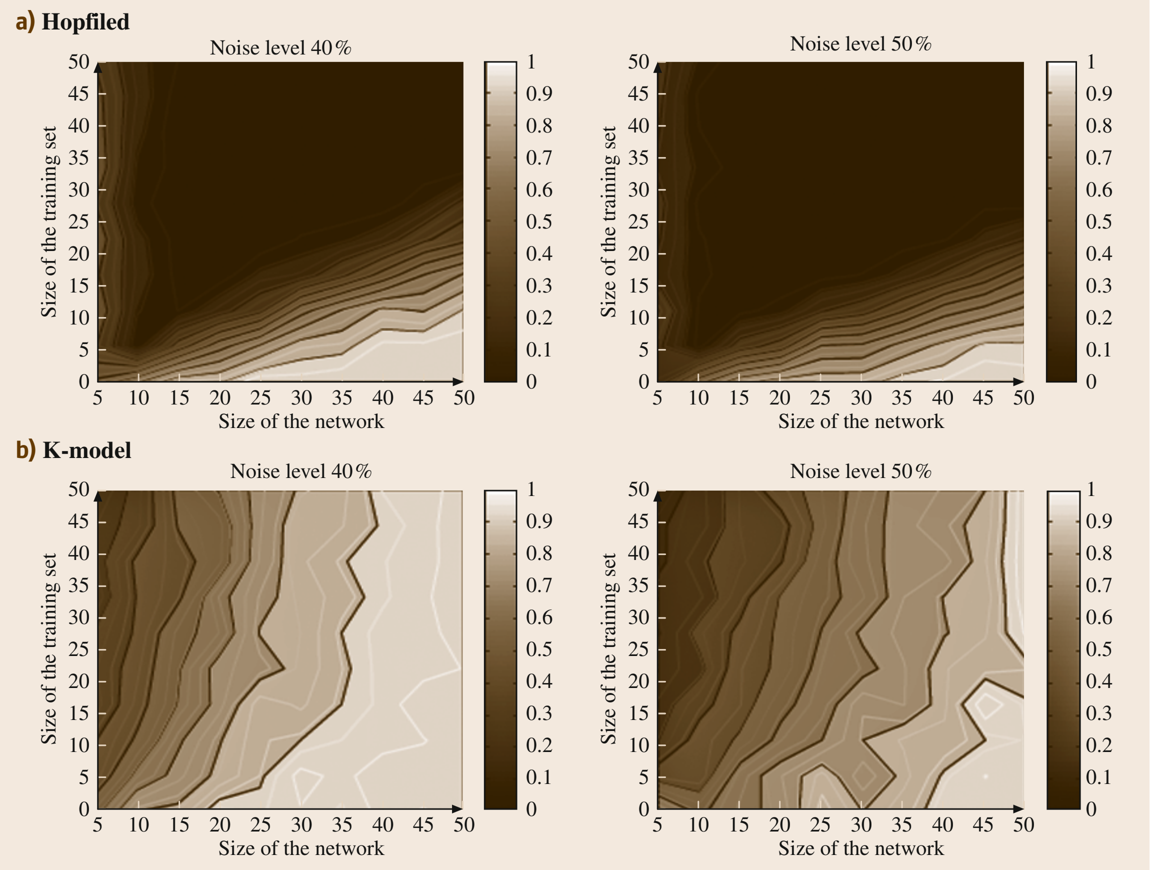

Fig. 33.25a,b Comparison of the memory capacity of **(a)** Hopfield and **(b)** KIII neural networks; the noise level is 40% (*left*); 50% (*right*); the lighter the color the higher the recall accuracy. Observe the linear gain for Hopfield networks and the superlinear (fifth-order) separation for KIII (after [33.152])

across widespread cortical and subcortical areas [33.82, 156, 157].

Intermittent oscillations in spatio-temporal neural dynamics are modeled by a neuropercolation approach. Neuropercolation is a family of probabilistic models based on the theory of probabilistic cellular automata on lattices and random graphs and it is motivated by structural and dynamical properties of neural populations. Neuropercolation constructs the hierarchy of interactive populations in networks as developed in *Freeman* K models [33.94, 144], but replace differential equations with probability distributions from the observed random networks that evolve in time [33.158]. Neuropercolation considers populations of cortical neurons which sustain their background state by mutual excitation, and their stability is guaranteed by the neural refractory periods. Neural populations transmit and receive signals from other populations by virtue of small-world effects [33.77, 159]. Tools of statistical physics and finite-size scaling theory are applied to describe critical behavior of the neuropil. Neuropercolation theory provides a mathematical approach to describe phase transitions and critical phenomena in large-scale, interactive cortical networks. The existence of phase transitions is proven in specific probabilistic cellular automata models [33.160, 161].

Simulations by neuropercolation models demonstrate the onset of large-scale synchronization-desynchronization behavior [33.162]. Figure 33.26 illustrates results of intermittent phase desynchronization for neuropercolation with excitatory and inhibitory populations. Three main regimes can be distinguished, separated by critical noise values $\varepsilon_1 > \varepsilon_0$. In Regime I $\varepsilon > \varepsilon_1$, Fig. 33.26a, the channels are not synchronous and

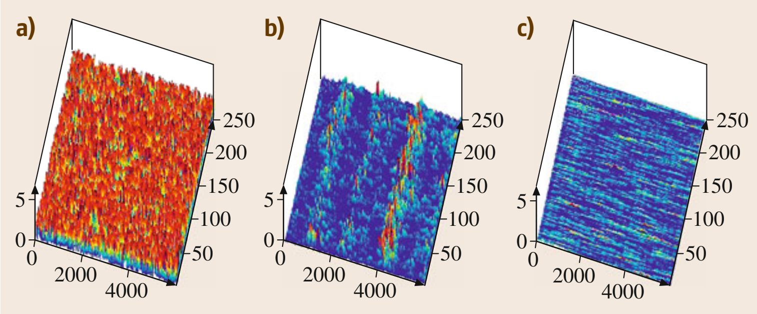

Fig. 33.26a–c Phase synchronization–desynchronization with excitatory–inhibitory connections in neuropercolation with 256 granule nodes; the z-axis shows the pairwise phase between the units. **(a)** No synchrony; **(b)** intermittent synchrony; **(c)** highly synchronized, frozen phase regime (after [33.162])

the phase values are distributed broadly. In Regime II $\varepsilon_1 > \varepsilon > \varepsilon_0$, Fig. 33.26b, the phase lags are drastically reduced indicating significant synchrony over extended time periods. Regime III is observed for high values of $\varepsilon_0 > \varepsilon$, when the channels demonstrate highly synchronized, frozen dynamics, see Fig. 33.26c. Similar transitions can be induced by the relative strength of inhibition, as well as by the fraction of rewiring across the network [33.114, 115, 163]. The probabilistic model of neural populations reproduces important properties of the spatio-temporal dynamics of cortices and is a promising approach for large-scale cognitive models.

33.3 Memristive Neurodynamics

Sequential processing of fetch, decode, and execution of instructions through the classical von Neumann digital computers has resulted in less efficient machines as their ecosystems have grown to be increasingly complex [33.164]. Though modern digital computers are fast and complex enough to emulate the brain functionality of animals like spiders, mice, and cats [33.165, 166], the associated energy dissipation in the system grows exponentially along the hierarchy of animal intelligence. For example, to perform certain cortical simulations at the cat scale even at an 83 times slower firing rate, the IBM team has to employ Blue Gene/P (BG/P), a super computer equipped with 147 456 CPUs and 144 TBs of main memory. On the other hand, the human brain contains more than 100 billion neurons and each neuron has more than 20 000 synapses [33.167]. Efficient circuit implementation of synapses, therefore, is very important to build a brain-like machine. One active branch of this research area is cellular neural networks (CNNs) [33.168, 169], where lots of multiplication circuits are utilized in a complementary metal-oxide-semiconductor (CMOS) chip. However, since shrinking the current transistor size is very difficult, introducing a more efficient approach is essential for further development of neural network implementations.

The memristor was first authorized by *Chua* as the fourth basic circuit element in electrical circuits in 1971 [33.170]. It is based on the nonlinear characteristics of charge and flux. By supplying a voltage or current to the memristor, its resistance can be altered. In this way, the memristor remembers information. In that seminal work, Chua demonstrated that the memristance $M(q)$ relates the charge q and the flux φ in such a way that the resistance of the device will change with the applied electric field and time

$$M = \frac{\mathrm{d}\varphi}{\mathrm{d}q} . \tag{33.26}$$

The parameter M denotes the memristance of a charge controlled memristor, measured in ohms. Thus, the memristance M can be controlled by applying a voltage or current signal across the memristor. In other words, the memristor behaves like an ordinary resistor at any given instance of time, where its resistance depends on the complete history of the device [33.170].

Although the device was proposed nearly four decades ago, it was not until 2008 that researchers from HP Labs showed that the devices they had fabricated were indeed two-terminal memristors [33.171]. Figure 33.27 shows the I–V characteristics of a generic memristor, where memristance behavior is observed for TiO_2-based devices. A TiO_{2-x} layer with oxygen vacancies is placed on a perfect TiO_2 layer, and these layers are sandwiched between platinum electrodes. In metal oxide materials, the switching from R_{off} to R_{on} and vice versa occurs as a result of ion migration, due to the enormous electric fields applied across the nanoscale structures. These memristors have been fabricated using nanoimprint lithography and were successfully integrated on a CMOS substrate in [33.172]. Apart from these metal-oxide memristors, memristance has also been demonstrated using magnetic materials based on their magnetic domain wall motion and spin-torque induced magnetization switching in [33.173].

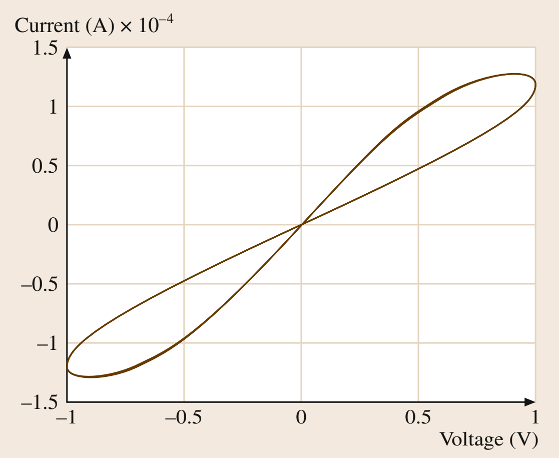

Fig. 33.27 Typical *I–V* characteristic of memristor (after [33.171]). The pinched hysteresis loop is due to the nonlinear relationship between the memristance current and voltage. The parameters of the memristor are $R_{\text{on}} = 100\,\Omega$, $R_{\text{off}} = 16\,\text{K}\Omega$, $R_{\text{init}} = 11\,\text{k}\Omega$, $D = 10\,\text{nm}$, $u_{\text{v}} = 10^{-10}\,\text{cm}^2\,\text{s}^{-1}\,\text{V}^{-1}$, $p = 10$ and $V_{\text{in}} = \sin(2\pi t)$. The memristor exhibits the feature of pinched hysteresis, which means that a lag occurs between the application and the removal of a field and its subsequent effect, just like the feature of neurons in the human brain

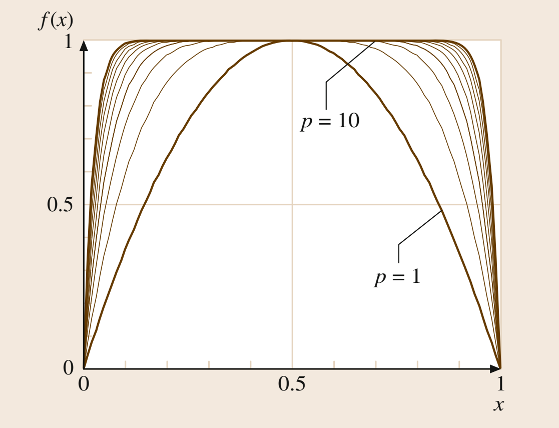

Fig. 33.28 Window function for different integer *p*

Furthermore, several different types of nonlinear memristor models have been investigated [33.174, 175]. One of them is the window model in which the state equation is multiplied by window function $F_p(\omega)$, namely

$$\frac{\mathrm{d}\omega}{\mathrm{d}t} = \mu_{\text{v}} \frac{R_{\text{on}}}{D} \mathrm{i}(t) F_p(\omega) , \tag{33.27}$$

where p is an integer parameter and $F_p(\omega)$ is defined by

$$F_p(\omega) = 1 - \left(2\frac{\omega}{D} - 1\right)^{2p} , \tag{33.28}$$

which is shown in Fig. 33.28.

33.3.1 Memristor-Based Synapses

The design of simple weighting circuits for synaptic multiplication between arbitrary input signals and weights is extremely important in artificial neural systems. Some efforts have been made to build neuron-like analog neural networks [33.178–180]. However, this research has gained limited success so far because of the difficulty in implementing the synapses efficiently. Based on the memristor, a novel weighting circuit was proposed by *Kim* et al. [33.176, 181, 182] as shown in Fig. 33.29. The memristors provide a bridge-like switching for achieving either positive or negative weighting. Though several memristors are employed to emulate a synapse, the total area of the memristors is less than that of a single transistor. To compensate for the spatial nonuniformity and nonideal response of the memristor bridge synapse, a modified chip-in-the-loop learning scheme suitable

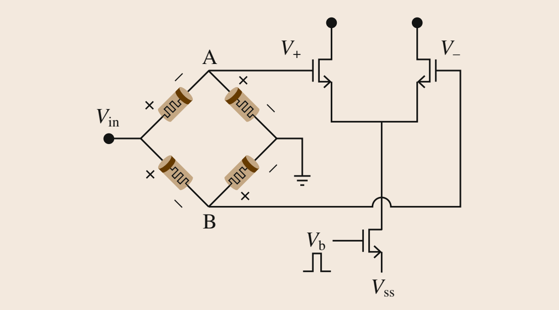

Fig. 33.29 Memristor bridge circuit. The synaptic weight is programmable by varying the input voltage. The weighting of the input signal is also performed in this circuit (after [33.176])

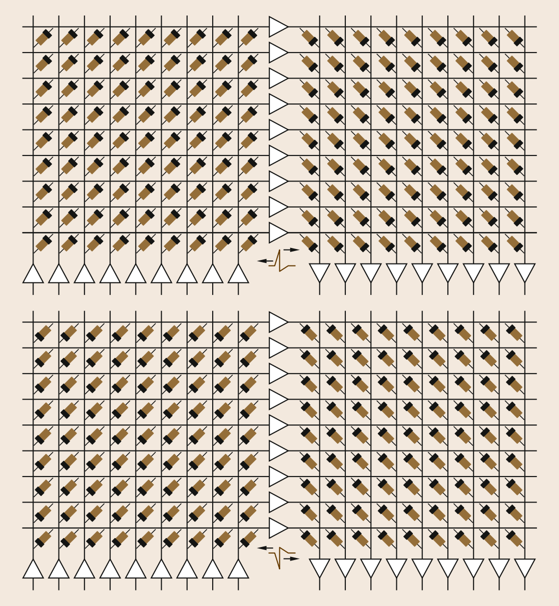

Fig. 33.30 Neuromorphic memristive computer equipped with STDP (after [33.177])

for the proposed neural network architecture is investigated [33.176]. In the proposed method, the initial learning is conducted by software, and the behavior of the software-trained network is learned via the hardware network by learning each of the single layered neurons of the network independently. The forward calculation of single layered neuron learning is implemented through circuit hardware and is followed by a weight updating phase assisted by a host computer. Unlike conventional chip-in-the-loop learning, the need for the readout of synaptic weights for calculating weight updates in each epoch is eliminated by virtue of the memristor bridge synapse and the proposed learning scheme.

On the other hand, spike-timing-dependent learning (STDP), which is a powerful learning paradigm for spiking neural systems because of its massive parallelism, potential scalability, and inherent defect, fault, and failure-tolerance, can be implemented by using a crossbar memristive array combined with neurons that asynchronously generate spikes of a given shape [33.177, 185]. Such spikes need to be sent back through the neurons to the input terminal as in Fig. 33.30. The shape of the spikes turns out to be very similar to the neural spikes observed in realistic biological neurons. The STDP learning function obtained by combining such neurons with memristors is exactly obtained from neurophysiological experiments on real synapses. Such nanoscale synapses can be combined with CMOS neurons which is possible to create neuromorphic hardware several orders of magnitude denser than in conventional CMOS. This method offers better control over power dissipation; fewer constraints on the design of memristive materials used for nanoscale synapses; greater freedom in learning algorithms than traditional design of synapses since the synaptic learning dynamics can be dynamically turned on or off; greater control over the precise form and timing of the STDP equations; the ability to implement a variety of other learning laws besides STDP; better circuit diversity since the approach allows different learning laws to be implemented in different areas of a single chip

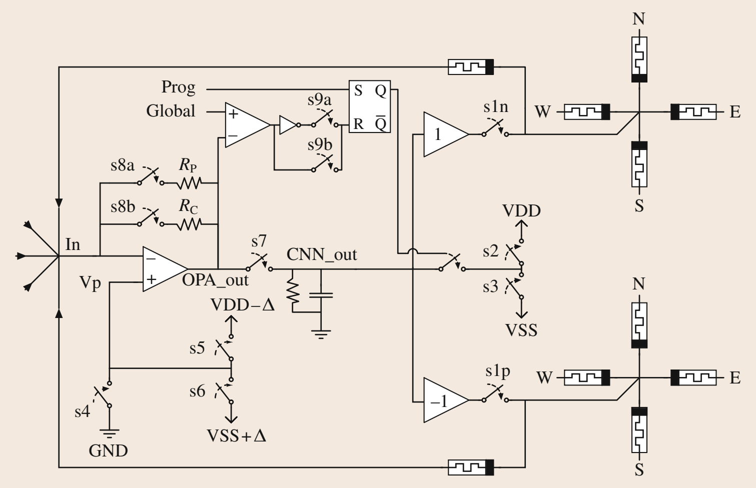

Fig. 33.31 Memristor-based cellular neural networks cell (after [33.183])

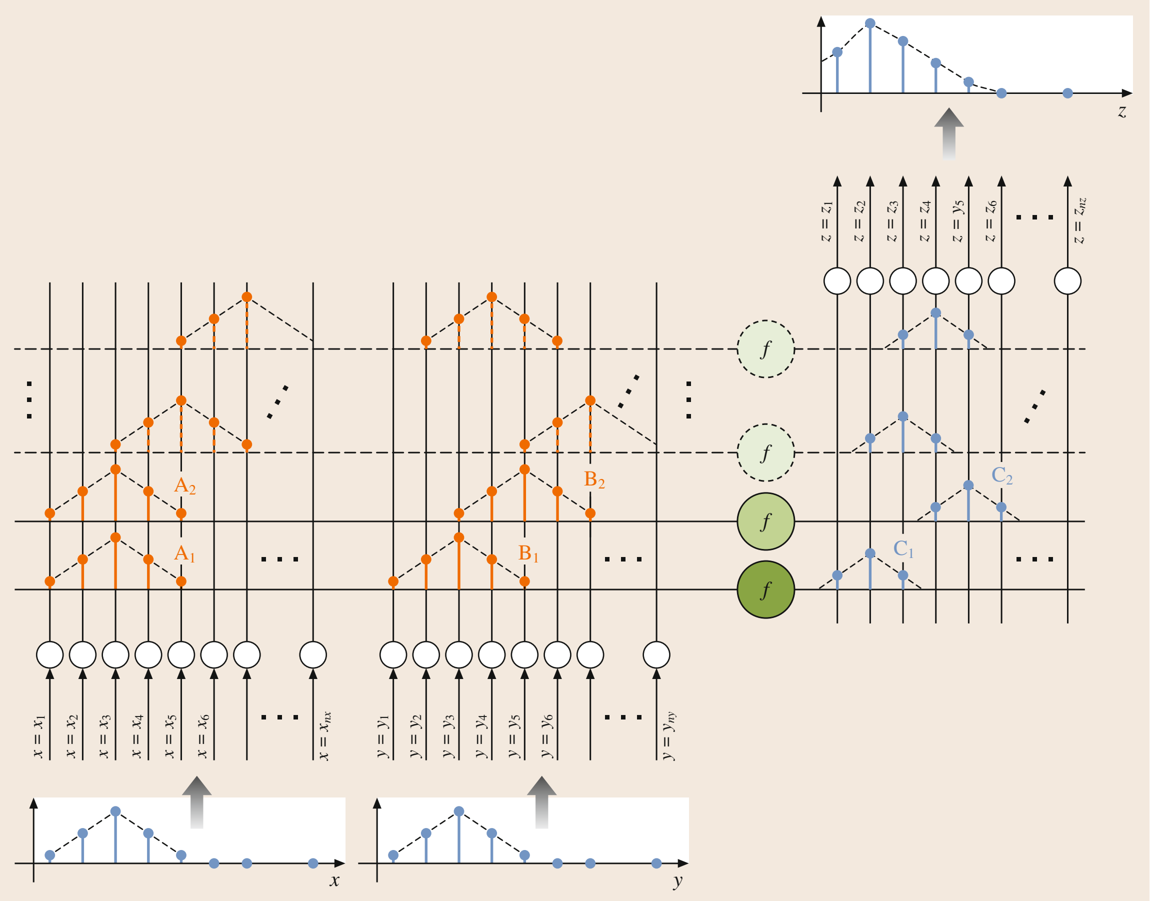

Fig. 33.32 Simple realization of MNN based on fuzzy concepts (after [33.184])

using the same memristive material for all synapses. Furthermore, an analog CMOS neuromorphic design utilizing STDP and memristor synapses is investigated for use in building a multipurpose analogy neuromorphic chip [33.186]. In order to obtain a multipurpose chip, a suitable architecture is established. Based on the technique of IBM 90 nm CMOS9RF, neurons are designed to interface with Verilog-A memristor synapses models to perform the XOR operation and edge detection function.

To make the neurons compatible with such new synapses, some novel training methods are proposed. For instance, *Manem* et al. proposed a variation-tolerant training method to efficiently reconfigure memristive synapses in a trainable threshold gate array (TTGA) system [33.187]. The training process is inspired from the gradient descent machine learning algorithm commonly used to train artificial threshold neural networks known as perceptrons. The proposed training method is robust to the unpredictability of CMOS and nanocircuits with decreasing technology sizes, but also provides its own randomness in its training.

33.3.2 Memristor-Based Neural Networks

Employing memristor-based synapses, some results have been obtained about the memristor-based neural networks [33.183, 184, 188]. As the template weights in memristor-based neural networks (MNNs) are usually known and need to be updated between each template in a sequence of templates, there should be a way to rapidly change the weights. Meanwhile, the MNN cells need to be modified, as the programmable couplings are implemented by memristors which require programming circuits to isolate each other. *Lehtonen* and *Laiho* proposed a new cell of memristor-based cellular neural network that can be used to program the templates. For this purpose, a voltage *global* is input into the cell. This voltage is used to convey the weight of one connection into the cells [33.183]. The level of virtual ground and switches are controlled so that the memristor connected to a particular neighbor is biased above the programming threshold, until it reaches the desired resistance value.

Merrikh-Bayat et al. presented a new way to explain the relationships between logical circuits and artificial neural networks, logical circuits and fuzzy logic, and artificial neural networks and fuzzy inference systems, and proposed a new neuro-fuzzy computing system, which can effectively be implemented via the memristor-crossbar structure [33.184]. A simple realization of MNNs is shown in Figs. 33.32–33.34. Figure 33.32 shows that it is possible to interpret the working procedure of conventional artificial neural network ANN without changing its structure. In this figure, each row of the structure implements a simple fuzzy rule or minterm. Figure 33.33 shows how the activation function of neurons can be implemented when the activation function is modeled by a *t*-norm operator. Matrix multiplication is performed by vector circuit in Fig. 33.34. This circuit consists of a simple memristor crossbar where each of its rows is connected to the virtually grounded terminal of an operational amplifier that plays the role of a neuron with identity activation function. The advantages of the proposed system are twofold: first, its hardware can be directly trained using the Hebbian learning rule and without the need to perform any optimization; second, this system has a great ability to deal with a huge number of input-output training data without facing problems like overtrainging.

Howard et al. proposed a spiking neuro-evolutionary system which implements memristors as plastic connections [33.188]. These memristors provide a learning architecture that may be beneficial to the evolutionary design process that exploits parameter self-adaptation and variable topologies, allow the number of neurons, connection weights, and interneural connectivity pattern to emerge. This approach allows the evolution of networks with appropriate complexity to emerge whilst exploiting the memristive properties of the connections to reduce learning time.

To investigate the dynamic behaviors of memristor-based neural networks, *Zeng* et al. proposed the memristor-based recurrent neural networks (MRNNs) [33.189, 190] shown in Fig. 33.35, where $x_i(.)$ is the state of the i-th subsystem, $f_j(.)$ is the amplifier, M_{fij} is the connection memristor between the amplifier $f_j(.)$ and state $x_i(.)$, R_i and C_i are the resistor and capacitor, I_i is the external input, a_i, b_i are the outputs, $i, j = 1, 2, \ldots, n$. The parameters in this neural network are

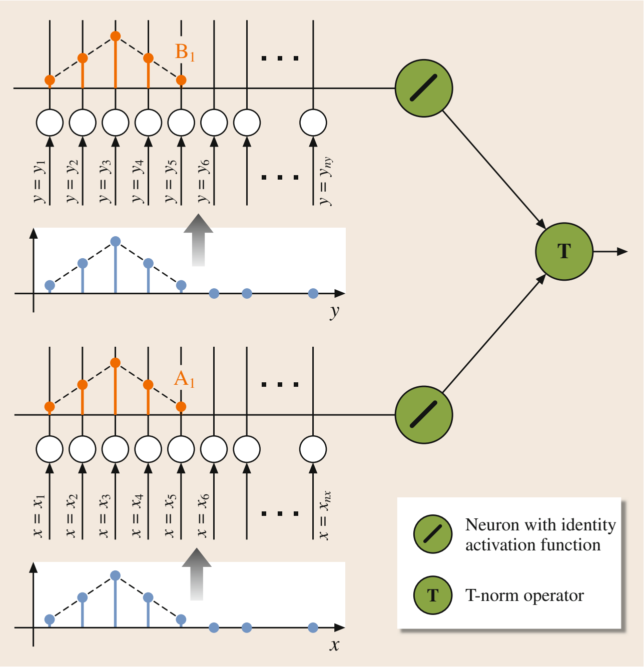

Fig. 33.33 Implementation of the activation function of neurons (after [33.184])

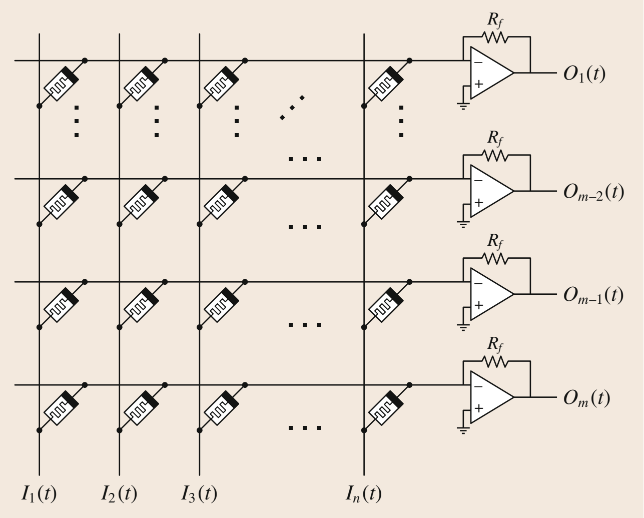

Fig. 33.34 Memristor crossbar-based circuit (after [33.184])

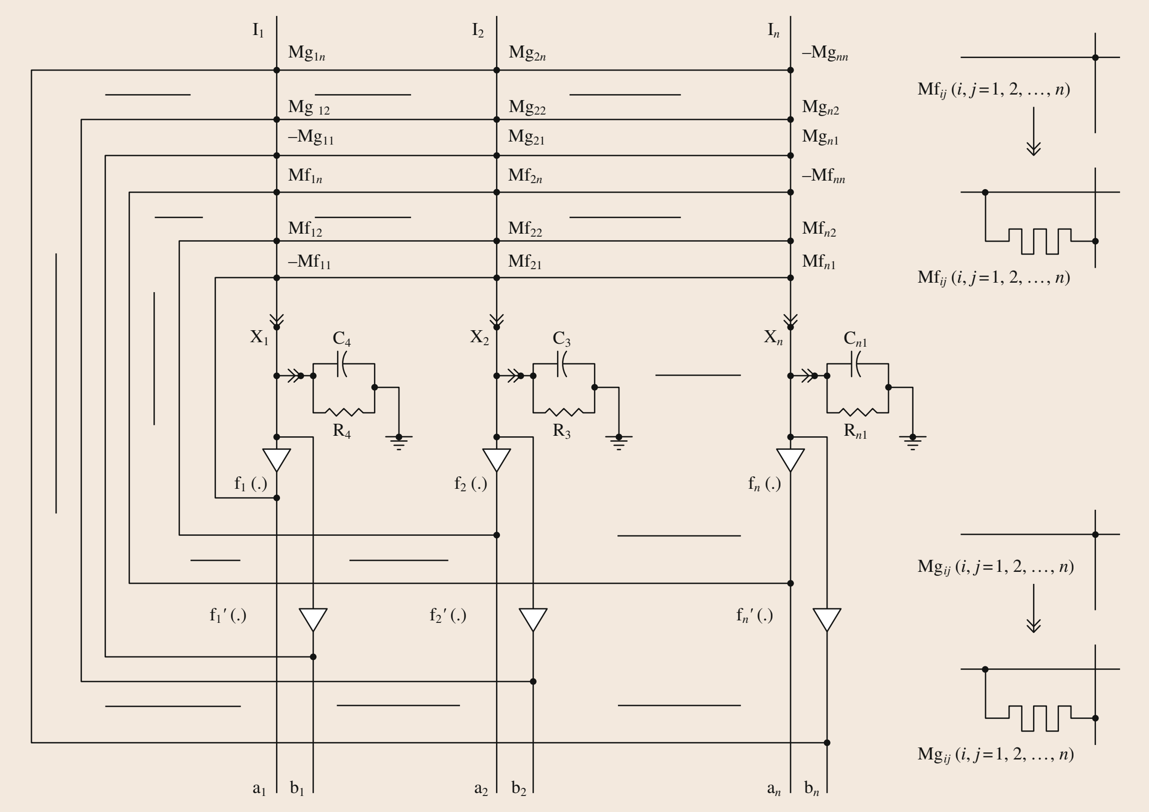

Fig. 33.35 Circuit of a memristor-based recurrent network (after [33.189])

changed according to the state of the system, so this network is a state-dependent switching system. The dynamic behavior of this neural network with time-varying delays was investigated based on the Filippov theory and the Lyapunov method.

33.3.3 Conclusion

Memristor-based synapses and neural networks have been investigated by many scientists for their possible applications in analog, digital information processing, and memory and logic applications. However, the problem, of how to take advantage of the nonvolatile memory of memristors, nanoscale, low-power dissipation, and so on to design a method to process and store the information, which needs learning and memory, into the synapses of the memristor-based neural networks at the dynamical mapping space by a more rational space-parting method, is still an open issue. Further investigation is needed to shorten such a gap.

33.4 Neurodynamic Optimization

Optimization is omnipresent in nature and society, and an important tool for problem-solving in science, engineering, and commerce. Optimization problems arise in a wide variety of applications such as the design, planning, control, operation, and management of engineering systems. In many applications (e.g., online pattern recognition and in-chip signal processing in mobile devices), real-time optimization is necessary or desirable. For such applications, conventional optimization techniques may not be competent due to stringent requirements on computational time. It is computationally challenging when optimization procedures are to be performed in real time to optimize the performance of dynamical systems.

The brain is a profound dynamic system and its neurons are always active from birth to death. When a decision is to be made in the brain, many of its neurons are highly activated to gather information, search memory, compare differences, and make inferences and decisions. Recurrent neural networks are brain-like nonlinear dynamic system models and can be properly designed to imitate biological counterparts and serve as goal-seeking parallel computational models for solving optimization problems in a variety of settings. Neurodynamic optimization can be physically realized in designated hardware such as application-specific integrated circuits (ASICs) where optimization is carried out in a parallel and distributed manner, where the convergence rate of the optimization process is independent of the problem dimensionality. Because of the inherent nature of parallel and distributed information processing, neurodynamic optimization can handle large-scale problems. In addition, neurodynamic optimization may be used for optimizing dynamic systems in multiple time scales with parameter-controlled convergence rates. These salient features are particularly desirable for dynamic optimization in decentralized decision-making scenarios [33.191–194]. While population-based evolutionary approaches to optimization have emerged as prevailing heuristic and stochastic methods in recent years, neurodynamic optimization deserves great attention in its own right due to its close ties with optimization and dynamical systems theories, as well as its biological plausibility and circuit implementability with very large scale integration (VLSI) or optical technologies.

33.4.1 Neurodynamic Models

The past three decades witnessed the birth and growth of neurodynamic optimization. Although a couple of circuit-based optimization methods were developed earlier [33.195–197], it was perhaps *Hopfield* and *Tank* who spearheaded neurodynamic optimization research in the context of neural computation with their seminal work in the mid 1980s [33.198–200]. Since the inception, numerous neurodynamic optimization models in various forms of recurrent neural networks have been developed and analyzed, see [33.201–256], and the references therein. For example, *Tank* and *Hopfield* extended the continuous-time Hopfield network for linear programming and showed their experimental results with a circuit of operational amplifiers and other discrete components on a breadboard [33.200]. *Kennedy* and *Chua* developed a circuit-based recurrent neural network for nonlinear programming [33.201]. It is proven that the state of the neurodynamics is globally convergent and an equilibrium corresponding to an approximate optimal solution of the given optimization problems.

Over the years, neurodynamic optimization research has made significant progress with models with improved features for solving various optimization problems. Substantial improvements of neurodynamic optimization theory and models have been made in the following dimensions:

i) Solution quality: designed based on smooth penalty methods with a finite penalty parameter; the earliest neurodynamic optimization models can converge to approximate solutions only [33.200, 201]. Later on, better models designed based on other design principles can guarantee to state or output convergence to exact optimal solutions of solvable convex and pseudoconvex optimization problems with or without any conditions [33.204, 205, 208, 210], etc.
ii) Solvability scope: the solvability scope of neurodynamic optimization has been expanded from linear programming problems [33.200, 202, 208, 211, 212, 214–219, 223, 242, 244, 251], to quadratic programming problems [33.202–206, 210, 214, 217, 218, 220, 225, 226, 229, 233, 240–243, 247], to smooth convex programming problems with various constraints [33.201, 204, 205, 210, 214, 222, 224, 228, 230, 232, 234, 237, 245, 246, 257], to nonsmooth convex optimization problems [33.235, 248, 250–256], and recently to nonsmooth optimization with some nonconvex objective functions or constraints [33.239, 249, 254–256].
iii) Convergence property: the convergence property of neurodynamic optimization models has been extended from near-optimum convergence [33.200, 201], to conditional exact-optimum global convergence [33.205, 208, 210], to guaranteed global convergence [33.204, 205, 214–216, 218, 219, 222, 226–228, 230, 232, 234, 240, 243, 245, 247, 250, 253, 256, 257], to faster global exponential convergence [33.206, 224, 225, 228, 233, 237, 239, 241, 246, 254], to even more desirable finite-time convergence [33.235, 248, 249, 251, 252, 255], with increasing convergence rate.
iv) Model complexity: the neurodynamic optimization models for constrained optimization are essentially multilayer due to the introduction of instrumen-

tal variables for constraint handling (e.g., Lagrange multipliers or dual variables). The architectures of later neurodynamic optimization models for solving linearly constrained optimization problems have been reduced from multilayer structures to single-layer ones with decreasing model complexity to facilitate their implementation [33.243, 244, 251, 252, 254, 255].

Activation functions are a signature component of neural network models for quantifying the firing state activities of neurons. The activation functions in existing neurodynamic optimization models include smooth ones (e.g., sigmoid), as shown in Fig. 33.36a,b [33.200, 208–210], nonsmooth ones (e.g., piecewise-linear) as shown in Fig. 33.36c,d [33.203, 206], and even discontinuous ones as shown in Fig. 33.36e,f [33.243, 244, 251, 252, 254, 255].

33.4.2 Design Methods

The crux of neurodynamic optimization model design lies in the derivation of a convergent neurodynamic equation that prescribes the states of the neurodynamics. A properly derived neurodynamic equation can ensure that the states of neurodynamics reaches an equilibrium that satisfies the constraints and optimizes the objective function. Although the existing neurodynamic optimization models are highly diversified with many different features, the design methods or principles for determining their neurodynamic equations can be categorized as follows:

i) Penalty methods
ii) Lagrange methods
iii) Duality methods
iv) Optimality methods.

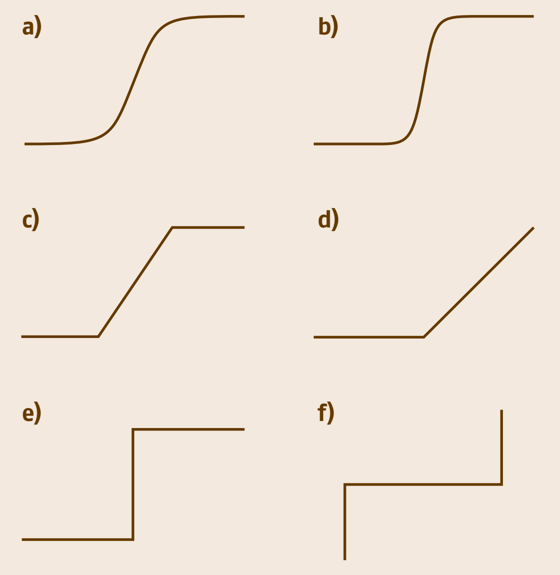

Fig. 33.36a–f Three classes of activation functions in neurodynamic optimization models: smooth in **(a)** and **(b)**, nonsmooth in **(c)** and **(d)**, and discontinuous in **(e)** and **(f)**

Penalty Methods

Consider the general constrained optimization problem

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & g(x) \leq 0, \\ & h(x) = 0, \end{aligned}$$

where $x \in Re^n$ is the vector of decision variables, $f(x)$ is an objective function, $g(x) = [g_1(x), \ldots, g_m(x)]^\mathsf{T}$ is a vector-valued function, and $h(x) = [h_1(x), \ldots, h_p(x)]^\mathsf{T}$ a vector-valued function.

A penalty method starts with the formulation of a smooth or nonsmooth energy function based on a given objective function $f(x)$ and constraints $g(x)$ and $h(x)$. It plays an important role in neurodynamic optimization. Ideally, the minimum of a formulated energy function corresponds to the optimal solution of the original optimization problem. For constrained optimization, the minimum of the energy function has to satisfy a set of constraints. Most early approaches formulate an energy function by incorporating objective function and constraints through functional transformation and numerical weighting [33.198–201]. Functional transformation is usually used to convert constraints to a penalty function to penalize the violation of constraints; e.g., a smooth penalty function is as follows

$$p(x) = \frac{1}{2} \sum_{i=1}^{m} \{[-g_i(x)]^+\}^2 + \sum_{j=1}^{p} [h_j(x)]^2 \ ,$$

where $[y]^+ = \max\{0, y\}$. Numerical weighting is often used to balance constraint satisfaction and objective optimization, e.g.,

$$E(x) = f(x) + wp(x) \ ,$$

where w is a positive weight.

In smooth penalty methods, neurodynamic equations are usually derived as the negative gradient flow of the energy function in the form of a differential equation

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} \propto -\nabla E(x(t)) \,.$$

If the energy function is bounded below, the stability of the neurodynamics can be ensured. Nevertheless, the major limitation is that the neurodynamics designed using a smooth penalty method with any fixed finite penalty parameter can converge to an approximate optimal solution only, as a compromise between constraint satisfaction and objective optimization. One way to remedy the approximated limitation of smooth penalty design methods is to introduce a variable penalty parameter. For example, a time-varying delaying penalty parameter (called temperature) is used in deterministic annealing networks to achieve exact optimality with a slow cooling schedule [33.208, 210].

If the objective function or penalty function is nonsmooth, the gradient has to be replaced by a generalized gradient and the neurodynamics can be modeled using a differential inclusion [33.235, 248, 249, 251, 252, 255]. Two advantages of nonsmooth penalty methods over smooth ones are possible constraint satisfaction and objective optimization with some finite penalty parameters and finite-time convergence of the resulting neurodynamics. Needless to say, nonsmooth neurodynamics are much more difficult to analyze to guarantee their stability.

Lagrange Methods

A Lagrange method for designing a neurodynamic optimization model begins with the formulation of a Lagrange function (Lagrangian) instead of an energy function [33.204, 205]. A typical Lagrangian is defined as

$$L(x,\lambda,\mu) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) + \sum_{j=1}^{p} \mu_j h_j(x) \,,$$

where $\lambda = (\lambda_1,\dots,\lambda_m)^{\mathsf{T}}$ and $\lambda = (\mu_1,\dots,\mu_p)^{\mathsf{T}}$ are Lagrange multipliers, for inequality constraints $g(x)$ and equality constraints $h(x)$, respectively.

According to the saddle-point theorem, the optimal solution can be determined by minimizing the Lagrangian with respect to x and maximizing it with respect to λ and μ. Therefore, neurodynamic equations can be derived in an augmented space

$$\begin{aligned}
\epsilon \frac{\mathrm{d}x(t)}{\mathrm{d}t} &= -\nabla_x L(x(t),\lambda(t),\mu(t)) \,,\\
\epsilon \frac{\mathrm{d}\lambda(t)}{\mathrm{d}t} &= -\nabla_\lambda L(x(t),\lambda(t),\mu(t)) \,,\\
\epsilon \frac{\mathrm{d}\mu(t)}{\mathrm{d}t} &= -\nabla_\mu L(x(t),\lambda(t),\mu(t)) \,,
\end{aligned}$$

where ϵ is a positive time constant. The equilibrium of the Lagrangian neurodynamics satisfy the Lagrange necessary optimality conditions.

Duality Methods

For convex optimization, the objective functions of primal and dual problems reach the same value at their optima. In view of this duality property, the duality methods for designing neurodynamic optimization models begin with the formulation of an energy function consisting of a duality gap between the primal and dual problems and a constraint-based penalty function, e.g.,

$$E(x,y) = \frac{1}{2}(f(x) - f_{\mathrm{d}}(y))^2 + p(x) + p_{\mathrm{d}}(y) \,,$$

where y is a vector of dual variables, $f_{\mathrm{d}}(y)$ is the dual objective function to be maximized, $p(x)$ and $p_{\mathrm{d}}(y)$ are, respectively, smooth penalty functions to penalize the violations of constraints of primal (original) and dual problems. The corresponding neurodynamic equation can be derived with guaranteed global stability as the negative gradient flow of the energy function similarly as in the aforementioned smooth penalty methods [33.216, 218, 222, 226, 258, 259]. Neurodynamic optimization models designed by using duality design methods can guarantee global convergence to the exact optimal solutions of convex optimization problems without any parametric condition.

In addition, using duality methods, dual networks and their simplified/improved versions can be designed for quadratic programming with reduced model complexity by mapping their global convergent optimal dual state variables to optimal primal solutions via linear or piecewise-linear output functions [33.240, 247, 260–263].

Optimality Methods

The neurodynamic equations of some recent models are derived based on optimality conditions (e.g., the

Karush–Kuhn–Tucker condition) and projection methods. Basically, the methods are to map the equilibrium of the designed neurodynamic optimization models to the equivalent equalities given by optimality conditions and projection equations (i. e., all equilibria essentially satisfy the optimality conditions) [33.225, 227, 228]. For several types of common geometric constraints (such as nonnegative constraints, bound constraints, and spherical constraints), some projection operators map the neuron state variables onto the convex feasible regions by using their activation functions and avoid the use of excessive dual variables as in the dual networks, and thus lower the model complexity. For neurodynamic optimization models designed using optimality methods, stability analysis is needed explicitly to ensure that the resulting neurodynamics are stable.

Once a neurodynamic equation has been derived and its stability is proven, the next step is to determine the architecture of the neural network in terms of the neurons and connections based on the derived neurodynamic equation. The last step is usually devoted to simulation or emulation to test the performance of the neural network numerically or physically. The simulation/emulation results may reveal additional properties or characteristics for further analysis or model redesign.

33.4.3 Selected Applications

Over the last few decades, neurodynamic optimization has been widely applied in many fields of science, engineering, and commerce, as highlighted in the following selected nine areas.

Scientific Computing

Neurodynamic optimization models ave been developed for solving linear equations and inequalities and computing inverse or pseudoinverse matrices [33.240, 264–268].

Network Routing

Neurodynamic optimization models have been developed or applied for shortest-path routing in networks modeled by using weighted directed graphs [33.258, 269–271].

Machine Learning

Neurodynamic optimization has been applied for support vector machine learning to take the advantages of its parallel computational power [33.272–274].

Data Processing

The data processing applications of neurodynamic optimization include, but are not limited to, sorting [33.275–277], winners-take-all selection [33.240, 277, 278], data fusion [33.279], and data reconciliation [33.254].

Signal/Image Processing

The applications of neurodynamic optimization for signal and image processing include, but are not limited to, recursive least-squares adaptive filtering, overcomplete signal representations, time delay estimation, and image restoration and reconstruction [33.191, 203, 204, 280–283].

Communication Systems

The telecommunication applications of neurodynamic optimization include beamforming [33.284, 285]) and simulations of DS-CDMA mobile communication systems [33.229].

Control Systems

Intelligent control applications of neurodynamic optimization include pole assignment for synthesizing linear control systems [33.286–289] and model predictive control for linear/nonlinear systems [33.290–292].

Robotic Systems

The applications of neurodynamic optimization in intelligent robotic systems include real-time motion planning and control of kinematically redundant robot manipulators with torque minimization or obstacle avoidance [33.259–263, 267, 293–298] and grasping force optimization for multifingered robotic hands [33.299].

Financial Engineering

Recently, neurodynamic optimization was also applied for real-time portfolio selection based on an equivalent probability measure to optimize the asset distribution in financial investments; [33.255, 300].

33.4.4 Concluding Remarks

Neurodynamic optimization provides a parallel distributed computational model for solving many optimization problems. For convex and convex-like optimization, neurodynamic optimization models are available with guaranteed optimality, expended applicability, improved convergence properties, and reduced model complexity. Neurodynamic optimization approaches have been demonstrated to be effective and efficient for many applications, especially those with real-time solution requirements.

The existing results can still be further improved to expand their solvability scope, increase their convergence rate, or reduce their model complexity. With the view that neurodynamic approaches to global optimization and discrete optimization are much more interesting and challenging, it is necessary to develop neurodynamic models for nonconvex optimization and combinatorial optimization. In addition, neurodynamic optimization approaches could be more widely applied for many other application areas in conjunction with conventional and evolutionary optimization approaches.

References

33.1 R. Abraham: *Dynamics: The Geometry of Behavior* (Aerial, Santa Cruz 1982)

33.2 J. Robinson: Attractor. In: *Encyclopedia of Nonlinear Science*, ed. by A. Scott (Routledge, New York 2005) pp. 26–28

33.3 S. Grossberg: Nonlinear difference-differential equations in prediction and learning theory, Proc. Natl. Acad. Sci. **58**, 1329–1334 (1967)

33.4 S. Grossberg: Global ratio limit theorems for some nonlinear functional differential equations I, Bull. Am. Math. Soc. **74**, 93–100 (1968)

33.5 H. Zhang, Z. Wang, D. Liu: Robust exponential stability of recurrent neural networks with multiple time-varying delays, IEEE Trans. Circuits Syst. II: Express Br. **54**, 730–734 (2007)

33.6 A.N. Michel, K. Wang, D. Liu, H. Ye: Qualitative limitations incurred in implementations of recurrent neural networks, IEEE Cont. Syst. Mag. **15**(3), 52–65 (1995)

33.7 H. Zhang, Z. Wang, D. Liu: Global asymptotic stability of recurrent neural networks with multiple time varying delays, IEEE Trans. Neural Netw. **19**(5), 855–873 (2008)

33.8 S. Hu, D. Liu: On the global output convergence of a class of recurrent neural networks with time-varying inputs, Neural Netw. **18**(2), 171–178 (2005)

33.9 D. Liu, S. Hu, J. Wang: Global output convergence of a class of continuous-time recurrent neural networks with time-varying thresholds, IEEE Trans. Circuits Syst. II: Express Br. **51**(4), 161–167 (2004)

33.10 H. Zhang, Z. Wang, D. Liu: Robust stability analysis for interval Cohen–Grossberg neural networks with unknown time varying delays, IEEE Trans. Neural Netw. **19**(11), 1942–1955 (2008)

33.11 M. Han, J. Fan, J. Wang: A dynamic feedforward neural network based on Gaussian particle swarm optimization and its application for predictive control, IEEE Trans. Neural Netw. **22**(9), 1457–1468 (2011)

33.12 S. Mehraeen, S. Jagannathan, M.L. Crow: Decentralized dynamic surface control of large-scale interconnected systems in strict-feedback form using neural networks with asymptotic stabilization, IEEE Trans. Neural Netw. **22**(11), 1709–1722 (2011)

33.13 Y. Zhang, T. Chai, H. Wang: A nonlinear control method based on anfis and multiple models for a class of SISO nonlinear systems and its application, IEEE Trans. Neural Netw. **22**(11), 1783–1795 (2011)

33.14 Y. Chen, W.X. Zheng: Stability and L_2 performance analysis of stochastic delayed neural networks, IEEE Trans. Neural Netw. **22**(10), 1662–1668 (2011)

33.15 M. Di Marco, M. Grazzini, L. Pancioni: Global robust stability criteria for interval delayed full-range cellular neural networks, IEEE Trans. Neural Netw. **22**(4), 666–671 (2011)

33.16 W.-H. Chen, W.X. Zheng: A new method for complete stability analysis of cellular neural networks with time delay, IEEE Trans. Neural Netw. **21**(7), 1126–1139 (2010)

33.17 H. Zhang, Z. Wang, D. Liu: Global asymptotic stability and robust stability of a general class of Cohen–Grossberg neural networks with mixed delays, IEEE Trans. Circuits Syst. I: Regul. Pap. **56**(3), 616–629 (2009)

33.18 X.X. Liao, J. Wang: Algebraic criteria for global exponential stability of cellular neural networks with multiple time delays, IEEE Trans. Circuits Syst. I **50**, 268–275 (2003)

33.19 Z.G. Zeng, J. Wang, X.X. Liao: Global exponential stability of a general class of recurrent neural networks with time-varying delays, IEEE Trans. Circuits Sys. I **50**(10), 1353–1358 (2003)

33.20 D. Angeli: Multistability in systems with counterclockwise input-output dynamics, IEEE Trans. Autom. Control **52**(4), 596–609 (2007)

33.21 D. Angeli: Systems with counterclockwise input-output dynamics, IEEE Trans. Autom. Control **51**(7), 1130–1143 (2006)

33.22 D. Angeli: Convergence in networks with counterclockwise neural dynamics, IEEE Trans. Neural Netw. **20**(5), 794–804 (2009)

33.23 J. Saez-Rodriguez, A. Hammerle-Fickinger, O. Dalal, S. Klamt, E.D. Gilles, C. Conradi: Multistability of signal transduction motifs, IET Syst. Biol. **2**(2), 80–93 (2008)

33.24 L. Chandrasekaran, V. Matveev, A. Bose: Multistability of clustered states in a globally inhibitory network, Phys. D **238**(3), 253–263 (2009)

33.25 B.K. Goswami: Control of multistate hopping intermittency, Phys. Rev. E **78**(6), 066208 (2008)

33.26 A. Rahman, M.K. Sanyal: The tunable bistable and multistable memory effect in polymer nanowires, Nanotechnology **19**(39), 395203 (2008)

33.27 K.C. Tan, H.J. Tang, W.N. Zhang: Qualitative analysis for recurrent neural networks with linear threshold transfer functions, IEEE Trans. Circuits Syst. I: Regul. Pap. **52**(5), 1003–1012 (2005)

33.28 H.J. Tang, K.C. Tan, E.J. Teoh: Dynamics analysis and analog associative memory of networks with LT neurons, IEEE Trans. Neural Netw. **17**(2), 409–418 (2006)

33.29 L. Zou, H.J. Tang, K.C. Tan, W.N. Zhang: Nontrivial global attractors in 2-D multistable attractor neural networks, IEEE Trans. Neural Netw. **20**(11), 1842–1851 (2009)

33.30 D. Liu, A.N. Michel: Sparsely interconnected neural networks for associative memories with applications to cellular neural networks, IEEE Trans. Circuits Syst. II: Analog Digit, Signal Process. **41**(4), 295–307 (1994)

33.31 M. Brucoli, L. Carnimeo, G. Grassi: Discrete-time cellular neural networks for associative memories with learning and forgetting capabilities, IEEE Trans. Circuits Syst. I: Fundam. Theory Appl. **42**(7), 396–399 (1995)

33.32 R. Perfetti: Dual-mode space-varying self-designing cellular neural networks for associative memory, IEEE Trans. Circuits Syst. I: Fundam. Theory Appl. **46**(10), 1281–1285 (1999)

33.33 G. Grassi: On discrete-time cellular neural networks for associative memories, IEEE Trans. Circuits Syst. I: Fundam. Theory Appl. **48**(1), 107–111 (2001)

33.34 L. Wang, X. Zou: Capacity of stable periodic solutions in discrete-time bidirectional associative memory neural networks, IEEE Trans. Circuits Syst. II: Express Br. **51**(6), 315–319 (2004)

33.35 J. Milton: Epilepsy: Multistability in a dynamic disease. In: *Self- Organized Biological Dynamics Nonlinear Control: Toward Understanding Complexity, Chaos, and Emergent Function in Living Systems*, ed. by J. Walleczek (Cambridge Univ. Press, Cambridge 2000) pp. 374–386

33.36 U. Feudel: Complex dynamics in multistable systems, Int. J. Bifurc. Chaos **18**(6), 1607–1626 (2008)

33.37 J. Hizanidis, R. Aust, E. Scholl: Delay-induced multistability near a global bifurcation, Int. J. Bifurc. Chaos **18**(6), 1759–1765 (2008)

33.38 G.G. Wells, C.V. Brown: Multistable liquid crystal waveplate, Appl. Phys. Lett. **91**(22), 223506 (2007)

33.39 G. Deco, D. Marti: Deterministic analysis of stochastic bifurcations in multi-stable neurodynamical systems, Biol. Cybern. **96**(5), 487–496 (2007)

33.40 J.D. Cao, G. Feng, Y.Y. Wang: Multistability and multiperiodicity of delayed Cohen-Grossberg neural networks with a general class of activation functions, Phys. D **237**(13), 1734–1749 (2008)

33.41 C.Y. Cheng, K.H. Lin, C.W. Shih: Multistability in recurrent neural networks, SIAM J. Appl. Math. **66**(4), 1301–1320 (2006)

33.42 Z. Yi, K.K. Tan: Multistability of discrete-time recurrent neural networks with unsaturating piecewise linear activation functions, IEEE Trans. Neural Netw. **15**(2), 329–336 (2004)

33.43 Z. Yi, K.K. Tan, T.H. Lee: Multistability analysis for recurrent neural networks with unsaturating piecewise linear transfer functions, Neural Comput. **15**(3), 639–662 (2003)

33.44 Z.G. Zeng, T.W. Huang, W.X. Zheng: Multistability of recurrent neural networks with time-varying delays and the piecewise linear activation function, IEEE Trans. Neural Netw. **21**(8), 1371–1377 (2010)

33.45 Z.G. Zeng, J. Wang, X.X. Liao: Stability analysis of delayed cellular neural networks described using cloning templates, IEEE Trans. Circuits Syst. I: Regul. Pap. **51**(11), 2313–2324 (2004)

33.46 Z.G. Zeng, J. Wang: Multiperiodicity and exponential attractivity evoked by periodic external inputs in delayed cellular neural networks, Neural Comput. **18**(4), 848–870 (2006)

33.47 L.L. Wang, W.L. Lu, T.P. Chen: Multistability and new attraction basins of almost-periodic solutions of delayed neural networks, IEEE Trans. Neural Netw. **20**(10), 1581–1593 (2009)

33.48 G. Huang, J.D. Cao: Delay-dependent multistability in recurrent neural networks, Neural Netw. **23**(2), 201–209 (2010)

33.49 L.L. Wang, W.L. Lu, T.P. Chen: Coexistence and local stability of multiple equilibria in neural networks with piecewise linear nondecreasing activation functions, Neural Netw. **23**(2), 189–200 (2010)

33.50 L. Zhang, Z. Yi, J.L. Yu, P.A. Heng: Some multistability properties of bidirectional associative memory recurrent neural networks with unsaturating piecewise linear transfer functions, Neurocomputing **72**(16–18), 3809–3817 (2009)

33.51 X.B. Nie, J.D. Cao: Multistability of competitive neural networks with time-varying and distributed delays, Nonlinear Anal.: Real World Appl. **10**(2), 928–942 (2009)

33.52 C.Y. Cheng, K.H. Lin, C.W. Shih: Multistability and convergence in delayed neural networks, Phys. D **225**(1), 61–74 (2007)

33.53 T.J. Sejnowski, C. Koch, P.S. Churchland: Computational neuroscience, Science **241**(4871), 1299 (1988)

33.54 G. Edelman: *Remembered Present: A Biological Theory of Consciousness* (Basic Books, New York 1989)

33.55 W.J. Freeman: *Societies of Brains: A Study in the Neuroscience of Love and Hate* (Lawrence Erlbaum, New York 1995)

33.56 R. Llinas, U. Ribary, D. Contreras, C. Pedroarena: The neuronal basis for consciousness, Philos. Trans. R. Soc. B **353**(1377), 1841 (1998)

33.57 F. Crick, C. Koch: A framework for consciousness, Nat. Neurosci. **6**(2), 119–126 (2003)

33.58 A.L. Hodgkin, A.F. Huxley: A quantitative description of membrane current and its application to conduction and excitation in nerve, J. Physiol. **117**(4), 500 (1952)

33.59 A. Pikovsky, M. Rosenblum: Synchronization, Scholarpedia **2**(12), 1459 (2007)

33.60 D. Golomb, A. Shedmi, R. Curtu, G.B. Ermentrout: Persistent synchronized bursting activity in cortical tissues with low magnesium concentration: A modeling study, J. Neurophysiol. **95**(2), 1049–1067 (2006)

33.61 M.L.V. Quyen, J. Foucher, J.-P. Lachaux, E. Rodriguez, A. Lutz, J. Martinerie, F.J. Varela: Comparison of Hilbert transform and wavelet methods for the analysis of neuronal synchrony, J. Neurosci. Methods **111**(2), 83–98 (2001)

33.62 W.J. Freeman, L.J. Rogers: Fine temporal resolution of analytic phase reveals episodic synchronization by state transitions in gamma EEGs, J. Neurophysiol. **87**(2), 937–945 (2002)

33.63 G.E.P. Box, G.M. Jenkins, G.C. Reinsel: *Ser. Probab. Stat*, Time Series Analysis: Forecasting and Control, Vol. 734 (Wiley, Hoboken 2008)

33.64 R.W. Thatcher, D.M. North, C.J. Biver: Development of cortical connections as measured by EEG coherence and phase delays, Hum. Brain Mapp. **29**(12), 1400–1415 (2007)

33.65 A. Pikovsky, M. Rosenblum, J. Kurths: *Synchronization: A Universal Concept in Nonlinear Sciences*, Vol. 12 (Cambridge Univ. Press, Cambridge 2003)

33.66 J. Rodriguez, R. Kozma: Phase synchronization in mesoscopic electroencephalogram arrays. In: *Intelligent Engineering Systems Through Artificial Neural Networks Series*, ed. by C. Dagli (ASME, New York 2007) pp. 9–14

33.67 J.M. Barrie, W.J. Freeman, M.D. Lenhart: Spatiotemporal analysis of prepyriform, visual, auditory, and somesthetic surface EEGs in trained rabbits, J. Neurophysiol. **76**(1), 520–539 (1996)

33.68 G. Dumas, M. Chavez, J. Nadel, J. Martinerie: Anatomical connectivity influences both intra- and inter-brain synchronizations, PloS ONE **7**(5), e36414 (2012)

33.69 J.A.S. Kelso: *Dynamic Patterns: The Self-Organization of Brain and Behavior* (MIT Press, Cambridge 1995)

33.70 S. Campbell, D. Wang: Synchronization and desynchronization in a network of locally coupled Wilson–Cowan oscillators, IEEE Trans. Neural Netw. **7**(3), 541–554 (1996)

33.71 H. Kurokawa, C.Y. Ho: A learning rule of the oscillatory neural networks for in-phase oscillation, IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **80**(9), 1585–1594 (1997)

33.72 G. Buzsaki: *Rhythms of the Brain* (Oxford Univ. Press, New York 2009)

33.73 A.K. Engel, P. Fries, W. Singer: Dynamic predictions: Oscillations and synchrony in top-down processing, Nat. Rev. Neurosci. **2**(10), 704–716 (2001)

33.74 W.J. Freeman, R.Q. Quiroga: *Imaging Brain Function with EEG: Advanced Temporal and Spatial Analysis of Electroencephalographic Signals* (Springer, New York 2013)

33.75 H. Haken: Cooperative phenomena in systems far from thermal equilibrium and in nonphysical systems, Rev. Mod. Phys. **47**(1), 67 (1975)

33.76 S.H. Strogatz: Exploring complex networks, Nature **410**(6825), 268–276 (2001)

33.77 O. Sporns, D.R. Chialvo, M. Kaiser, C.C. Hilgetag: Organization, development and function of complex brain networks, Trends Cogn. Sci. **8**(9), 418–425 (2004)

33.78 B. Bollobás, R. Kozma, D. Miklos (Eds.): *Handbook of Large-Scale Random Networks*, Bolyai Soc. Math. Stud., Vol. 18 (Springer, Berlin, Heidelberg 2009)

33.79 Y. Kuramoto: Cooperative dynamics of oscillator community, Prog. Theor. Phys. Suppl. **79**, 223–240 (1984)

33.80 M.G. Rosenblum, A.S. Pikovsky: Controlling synchronization in an ensemble of globally coupled oscillators, Phys. Rev. Lett. **92**(11), 114102 (2004)

33.81 O.V. Popovych, P.A. Tass: Synchronization control of interacting oscillatory ensembles by mixed nonlinear delayed feedback, Phys. Rev. E **82**(2), 026204 (2010)

33.82 W.J. Freeman: The physiology of perception, Sci. Am. **264**, 78–85 (1991)

33.83 M.A. Cohen, S. Grossberg: Absolute stability of global pattern formation and parallel memory storage by competitive neural networks, IEEE Trans. Syst. Man Cybern. **13**(5), 815–826 (1983)

33.84 J.J. Hopfield, D.W. Tank: Computing with neural circuits – A model, Science **233**(4764), 625–633 (1986)

33.85 C.M. Marcus, R.M. Westervelt: Dynamics of iterated-map neural networks, Phys. Rev. A **40**(1), 501 (1989)

33.86 W. Yu, J. Cao, J. Wang: An LMI approach to global asymptotic stability of the delayed Cohen-Grossberg neural network via nonsmooth analysis, Neural Netw. **20**(7), 810–818 (2007)

33.87 F.C. Hoppensteadt, E.M. Izhikevich: *Weakly Connected Neural Networks*, Applied Mathematical Sciences, Vol. 126 (Springer, New York 1997)

33.88 H.R. Wilson, J.D. Cowan: Excitatory and inhibitory interactions in localized populations of model neurons, Biophys. J. **12**(1), 1–24 (1972)

33.89 H.R. Wilson, J.D. Cowan: A mathematical theory of the functional dynamics of cortical and thalamic nervous tissue, Biol. Cybern. **13**(2), 55–80 (1973)

33.90 P.C. Bressloff: Spatiotemporal dynamics of continuum neural fields, J. Phys. A: Math. Theor. **45**(3), 033001 (2011)

33.91 D. Wang: Object selection based on oscillatory correlation, Neural Netw. **12**(4), 579–592 (1999)

33.92 A. Renart, R. Moreno-Bote, X.-J. Wang, N. Parga: Mean-driven and fluctuation-driven persistent activity in recurrent networks, Neural Comput. **19**(1), 1–46 (2007)

33.93 M. Ursino, E. Magosso, C. Cuppini: Recognition of abstract objects via neural oscillators: interaction among topological organization, associative memory and gamma band synchronization, IEEE Trans. Neural Netw. **20**(2), 316–335 (2009)

33.94 W.J. Freeman: *Mass Action in the Nervous System* (Academic, New York 1975)

33.95 D. Xu, J. Principe: Dynamical analysis of neural oscillators in an olfactory cortex model, IEEE Trans. Neural Netw. **15**(5), 1053–1062 (2004)

33.96 R. Ilin, R. Kozma: Stability of coupled excitatory–inhibitory neural populations and application to control of multi-stable systems, Phys. Lett. A **360**(1), 66–83 (2006)

33.97 R. Ilin, R. Kozma: Control of multi-stable chaotic neural networks using input constraints, 2007. IJCNN 2007. Int. Jt. Conf. Neural Netw., Orlando (2007) pp. 2194–2199

33.98 G. Deco, V. K. Jirsa, P. A. Robinson, M. Breakspear, K. Friston: The dynamic brain: From spiking neurons to neural masses and cortical fields, PLoS Comput. Biol. **4**(8), e1000092 (2008)

33.99 L. Ingber: Generic mesoscopic neural networks based on statistical mechanics of neocortical interactions, Phys. Rev. A **45**(4), 2183–2186 (1992)

33.100 V.K. Jirsa, K.J. Jantzen, A. Fuchs, J.A. Scott Kelso: Spatiotemporal forward solution of the EEG and meg using network modeling, IEEE Trans. Med. Imaging **21**(5), 493–504 (2002)

33.101 S. Coombes, C. Laing: Delays in activity-based neural networks, Philos. Trans. R. Soc. A **367**(1891), 1117–1129 (2009)

33.102 V.K. Jirsa: Neural field dynamics with local and global connectivity and time delay, Philos. Trans. R. Soc. A **367**(1891), 1131–1143 (2009)

33.103 K. Kaneko: Clustering, coding, switching, hierarchical ordering, and control in a network of chaotic elements, Phys. D **41**(2), 137–172 (1990)

33.104 R. Kozma: Intermediate-range coupling generates low-dimensional attractors deeply in the chaotic region of one-dimensional lattices, Phys. Lett. A **244**(1), 85–91 (1998)

33.105 S. Ishii, M.-A. Sato: Associative memory based on parametrically coupled chaotic elements, Phys. D **121**(3), 344–366 (1998)

33.106 F. Moss, A. Bulsara, M.F. Schlesinger (Eds.): *The proceedings of the NATO Advanced Research Workshop: Stochastic Resonance in Physics and Biology* (Plenum Press, New York 1993)

33.107 S.N. Dorogovtsev, A.V. Goltsev, J.F.F. Mendes: Critical phenomena in complex networks, Rev. Mod. Phys. **80**(4), 1275 (2008)

33.108 M.D. McDonnell, L.M. Ward: The benefits of noise in neural systems: bridging theory and experiment, Nat. Rev. Neurosci. **12**(7), 415–426 (2011)

33.109 A.V. Goltsev, M.A. Lopes, K.-E. Lee, J.F.F. Mendes: Critical and resonance phenomena in neural networks, arXiv preprint arXiv:1211.5686 (2012)

33.110 P. Bak: *How Nature Works: The Science of Self-Organized Criticality* (Copernicus, New York 1996)

33.111 J.M. Beggs, D. Plenz: Neuronal avalanches in neocortical circuits, J. Neurosci. **23**(35), 11167–11177 (2003)

33.112 J.M. Beggs: The criticality hypothesis: How local cortical networks might optimize information processing, Philos. Trans. R. Soc. A **366**(1864), 329–343 (2008)

33.113 T. Petermann, T.C. Thiagarajan, M.A. Lebedev, M.A.L. Nicolelis, D.R. Chialvo, D. Plenz: Spontaneous cortical activity in awake monkeys composed of neuronal avalanches, Proc. Natl. Acad. Sci. **106**(37), 15921–15926 (2009)

33.114 M. Puljic, R. Kozma: Narrow-band oscillations in probabilistic cellular automata, Phys. Rev. E **78**(2), 026214 (2008)

33.115 R. Kozma, M. Puljic, W.J. Freeman: Thermodynamic model of criticality in the cortex based on EEG/ECoG data. In: *Criticality in Neural Systems*, ed. by D. Plenz, E. Niebur (Wiley, Hoboken 2014) pp. 153–176

33.116 J.-P. Eckmann, D. Ruelle: Ergodic theory of chaos and strange attractors, Rev. Mod. Phys. **57**(3), 617 (1985)

33.117 E. Ott, C. Grebogi, J.A. Yorke: Controlling chaos, Phys. Rev. Lett. **64**(11), 1196–1199 (1990)

33.118 E.N. Lorenz: Deterministic nonperiodic flow, J. Atmos. Sci. **20**(2), 130–141 (1963)

33.119 K. Aihara, T. Takabe, M. Toyoda: Chaotic neural networks, Phys. Lett. A **144**(6), 333–340 (1990)

33.120 K. Aihara, H. Suzuki: Theory of hybrid dynamical systems and its applications to biological and medical systems, Philos. Trans. R. Soc. A **368**(1930), 4893–4914 (2010)

33.121 G. Matsumoto, K. Aihara, Y. Hanyu, N. Takahashi, S. Yoshizawa, J.-I. Nagumo: Chaos and phase locking in normal squid axons, Phys. Lett. A **123**(4), 162–166 (1987)

33.122 K. Aihara: Chaos engineering and its application to parallel distributed processing with chaotic neural networks, Proc. IEEE **90**(5), 919–930 (2002)

33.123 L. Wang, S. Li, F. Tian, X. Fu: A noisy chaotic neural network for solving combinatorial optimization problems: Stochastic chaotic simulated annealing, IEEE Trans. Syst. Man Cybern. B **34**(5), 2119–2125 (2004)

33.124 Z. Zeng, J. Wang: Improved conditions for global exponential stability of recurrent neural networks with time-varying delays, IEEE Trans. Neural Netw. **17**(3), 623–635 (2006)

33.125 M.D. Marco, M. Grazzini, L. Pancioni: Global robust stability criteria for interval delayed full-range cellular neural networks, IEEE Trans. Neural Netw. **22**(4), 666–671 (2011)

33.126 C.A. Skarda, W.J. Freeman: How brains make chaos in order to make sense of the world, Behav. Brain Sci. **10**(2), 161–195 (1987)

33.127 H.D.I. Abarbanel, M.I. Rabinovich, A. Selverston, M.V. Bazhenov, R. Huerta, M.M. Sushchik, L.L. Rubchinskii: Synchronisation in neural networks, Phys.-Usp. **39**(4), 337–362 (1996)

33.128 H. Korn, P. Faure: Is there chaos in the brain? II. experimental evidence and related models, c.r. Biol. **326**(9), 787–840 (2003)

33.129 R. Kozma, W.J. Freeman: Intermittent spatio-temporal desynchronization and sequenced synchrony in ECoG signals, Chaos Interdiscip. J. Nonlinear Sci. **18**(3), 037131 (2008)

33.130 K. Kaneko: *Collapse of Tori and Genesis of Chaos in Dissipative Systems* (World Scientific Publ., Singapore 1986)

33.131 K. Aihara: Chaos in neural networks. In: *The Impact of Chaos on Science and Society*, ed. by C. Grebogi, J.A. Yorke (United Nations Publ., New York 1997) pp. 110–126

33.132 I. Tsuda: Toward an interpretation of dynamic neural activity in terms of chaotic dynamical systems, Behav. Brain Sci. **24**(5), 793–809 (2001)

33.133 H. Bersini, P. Sener: The connections between the frustrated chaos and the intermittency chaos in small Hopfield networks, Neural Netw. **15**(10), 1197–1204 (2002)

33.134 P. Berge, Y. Pomeau, C. Vidal: *Order in Chaos* (Herman, Paris and Wiley, New York 1984)

33.135 Y. Pomeau, P. Manneville: Intermittent transition to turbulence in dissipative dynamical systems, Commun. Math. Phys. **74**(2), 189–197 (1980)

33.136 T. Higuchi: Relationship between the fractal dimension and the power law index for a time series: A numerical investigation, Phys. D **46**(2), 254–264 (1990)

33.137 B. Mandelbrot: *Fractals and Chaos: The Mandelbrot Set and Beyond*, Vol. 3 (Springer, New York 2004)

33.138 K. Falconer: *Fractal Geometry: Mathematical Foundations and Applications* (Wiley, Hoboken 2003)

33.139 T. Sauer, J.A. Yorke, M. Casdagli: Embedology, J. Stat. Phys. **65**(3), 579–616 (1991)

33.140 A. Wolf, J.B. Swift, H.L. Swinney, J.A. Vastano: Determining Lyapunov exponents from a time series, Phys. D **16**(3), 285–317 (1985)

33.141 L.D. Iasemidis, J.C. Sackellares, H.P. Zaveri, W.J. Williams: Phase space topography and the Lyapunov exponent of electrocorticograms in partial seizures, Brain Topogr. **2**(3), 187–201 (1990)

33.142 S. Micheloyannis, N. Flitzanis, E. Papanikolaou, M. Bourkas, D. Terzakis, S. Arvanitis, C.J. Stam: Usefulness of non-linear EEG analysis, Acta Neurol. Scand. **97**(1), 13–19 (2009)

33.143 W.J. Freeman: A field-theoretic approach to understanding scale-free neocortical dynamics, Biol. Cybern. **92**(6), 350–359 (2005)

33.144 W.J. Freeman, H. Erwin: Freeman *k*-set, Scholarpedia **3**(2), 3238 (2008)

33.145 R. Kozma, W. Freeman: Basic principles of the KIV model and its application to the navigation problem, Integr. Neurosci. **2**(1), 125–145 (2003)

33.146 R. Kozma, W.J. Freeman: The KIV model of intentional dynamics and decision making, Neural Netw. **22**(3), 277–285 (2009)

33.147 H.-J. Chang, W.J. Freeman, B.C. Burke: Biologically modeled noise stabilizing neurodynamics for pattern recognition, Int. J. Bifurc. Chaos **8**(2), 321–345 (1998)

33.148 R. Kozma, J.W. Freeman: Chaotic resonance – methods and applications for robust classification of noisy and variable patterns, Int. J. Bifurc. Chaos **11**(6), 1607–1629 (2001)

33.149 R.J. McEliece, E.C. Posner, E. Rodemich, S. Venkatesh: The capacity of the Hopfield associative memory, IEEE Trans. Inf. Theory **33**(4), 461–482 (1987)

33.150 J. Ma: The asymptotic memory capacity of the generalized Hopfield network, Neural Netw. **12**(9), 1207–1212 (1999)

33.151 V. Gripon, C. Berrou: Sparse neural networks with large learning diversity, IEEE Trans. Neural Netw. **22**(7), 1087–1096 (2011)

33.152 I. Beliaev, R. Kozma: Studies on the memory capacity and robustness of chaotic dynamic neural networks, Int. Jt. Conf. Neural Netw., IEEE (2006) pp. 3991–3998

33.153 D.A. Leopold, N.K. Logothetis: Multistable phenomena: Changing views in perception, Trends Cogn. Sci. **3**(7), 254–264 (1999)

33.154 E.D. Lumer, K.J. Friston, G. Rees: Neural correlates of perceptual rivalry in the human brain, Science **280**(5371), 1930–1934 (1998)

33.155 G. Werner: Metastability, criticality and phase transitions in brain and its models, Biosystems **90**(2), 496–508 (2007)

33.156 W.J. Freeman: Understanding perception through neural *codes*, IEEE Trans. Biomed. Eng. **58**(7), 1884–1890 (2011)

33.157 R. Kozma, J.J. Davis, W.J. Freeman: Synchronized minima in ECoG power at frequencies between beta-gamma oscillations disclose cortical singularities in cognition, J. Neurosci. Neuroeng. **1**(1), 13–23 (2012)

33.158 R. Kozma: Neuropercolation, Scholarpedia **2**(8), 1360 (2007)

33.159 E. Bullmore, O. Sporns: Complex brain networks: Graph theoretical analysis of structural and functional systems, Nat. Rev. Neurosci. **10**(3), 186–198 (2009)

33.160 R. Kozma, M. Puljic, P. Balister, B. Bollobas, W. Freeman: Neuropercolation: A random cellular automata approach to spatio-temporal neurodynamics, Lect. Notes Comput. Sci. **3305**, 435–443 (2004)

33.161 P. Balister, B. Bollobás, R. Kozma: Large deviations for mean field models of probabilistic cellular automata, Random Struct. Algorithm. **29**(3), 399–415 (2006)

33.162 R. Kozma, M. Puljic, L. Perlovsky: Modeling goal-oriented decision making through cognitive phase transitions, New Math. Nat. Comput. **5**(1), 143–157 (2009)

33.163 M. Puljic, R. Kozma: Broad-band oscillations by probabilistic cellular automata, J. Cell. Autom. **5**(6), 491–507 (2010)

33.164 S. Jo, T. Chang, I. Ebong, B. Bhadviya, P. Mazumder, W. Lu: Nanoscale memristor device as synapse in neuromorphic systems, Nano Lett. **10**, 1297–1301 (2010)

33.165 L. Smith: *Handbook of Nature-Inspired and Innovative Computing: Integrating Classical Models with Emerging Technologies* (Springer, New York 2006) pp. 433–475

33.166 G. Indiveri, E. Chicca, R. Douglas: A VLSI array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity, IEEE Trans. Neural Netw. **17**, 211–221 (2006)

33.167 Editors of Scientific American: *The Scientific American Book of the Brain* (Scientifc American, New York 1999)

33.168 L. Chua, L. Yang: Cellular neural networks, Theory. IEEE Trans. Circuits Syst. **CAS-35**, 1257–1272 (1988)

33.169 C. Zheng, H. Zhang, Z. Wang: Improved robust stability criteria for delayed cellular neural networks via the LMI approach, IEEE Trans. Circuits Syst. II – Expr. Briefs **57**, 41–45 (2010)

33.170 L. Chua: Memristor – The missing circuit element, IEEE Trans. Circuits Theory **CT-18**, 507–519 (1971)

33.171 D. Strukov, G. Snider, D. Stewart, R. Williams: The missing memristor found, Nature **453**, 80–83 (2008)

33.172 Q. Xia, W. Robinett, M. Cumbie, N. Banerjee, T. Cardinali, J. Yang, W. Wu, X. Li, W. Tong, D. Strukov, G. Snider, G. Medeiros-Ribeiro, R. Williams: Memristor – CMOS hybrid integrated circuits for reconfigurable logic, Nano Lett. **9**, 3640–3645 (2009)

33.173 X. Wang, Y. Chen, H. Xi, H. Li, D. Dimitrov: Spintronic memristor through spin-torque-induced magnetization motion, IEEE Electron Device Lett. **30**, 294–297 (2009)

33.174 Y. Joglekar, S. Wolf: The elusive memristor: Properties of basic electrical circuits, Eur. J. Phys. **30**, 661–675 (2009)

33.175 M. Pickett, D. Strukov, J. Borghetti, J. Yang, G. Snider, D. Stewart, R. Williams: Switching dynamics in titanium dioxide memristive devices, J. Appl. Phys. **106**(6), 074508 (2009)

33.176 S. Adhikari, C. Yang, H. Kim, L. Chua: Memristor bridge synapse-based neural network and its learning, IEEE Trans. Neural Netw. Learn. Syst. **23**(9), 1426–1435 (2012)

33.177 B. Linares-Barranco, T. Serrano-Gotarredona: Exploiting memristive in adaptive spiking neuromorphic nanotechnology systems, 9th IEEE Conf. Nanotechnol., Genoa (2009) pp. 601–604

33.178 M. Holler, S. Tam, H. Castro, R. Benson: An electrically trainable artificial neural network (ETANN) with 10240 *Floating gate* synapsess, Int. J. Conf. Neural Netw., Washington (1989) pp. 191–196

33.179 H. Withagen: Implementing backpropagation with analog hardware, Proc. IEEE ICNN-94, Orlando (1994) pp. 2015–2017

33.180 S. Lindsey, T. Lindblad: Survey of neural network hardware invited paper, Proc. SPIE Appl. Sci. Artif. Neural Netw. Conf., Orlando (1995) pp. 1194–1205

33.181 H. Kim, M. Pd Sah, C. Yang, T. Roska, L. Chua: Neural synaptic weighting with a pulse-based memristor circuit, IEEE Trans. Circuits Syst. I **59**(1), 148–158 (2012)

33.182 H. Kim, M. Pd Sah, C. Yang, T. Roska, L. Chua: Memristor bridge synapses, Proc. IEEE **100**(6), 2061–2070 (2012)

33.183 E. Lehtonen, M. Laiho: CNN using memristors for neighborhood connections, 12th Int. Workshop Cell. Nanoscale Netw. Appl. (CNNA), Berkeley (2010)

33.184 F. Merrikh-Bayat, F. Merrikh-Bayat, S. Shouraki: The neuro-fuzzy computing system with the capacity of implementation on a memristor crossbar and optimization-free hardware training, IEEE Trans. Fuzzy Syst. **22**(5), 1272–1287 (2014)

33.185 G. Snider: Spike-timing-dependent learning in memristive nanodevices, IEEE Int. Symp. Nanoscale Archit., Anaheim (2008) pp. 85–92

33.186 I. Ebong, D. Deshpande, Y. Yilmaz, P. Mazumder: Multi-purpose neuro-architecture with memristors, 11th IEEE Conf. Nanotechnol., Portland, Oregon (2011) pp. 1194–1205

33.187 H. Manem, J. Rajendran, G. Rose: Stochastic gradient descent inspired training technique for a CMOS/Nano memristive trainable threshold gate way, IEEE Trans. Circuits Syst. I **59**(5), 1051–1060 (2012)

33.188 G. Howard, E. Gale, L. Bull, B. Costello, A. Adamatzky: Evolution of plastic learning in spiking networks via memristive connections, IEEE Trans. Evol. Comput. **16**(5), 711–729 (2012)

33.189 S. Wen, Z. Zeng, T. Huang: Exponential stability analysis of memristor-based recurrent neural networks with time-varying delays, Neurocomputing **97**(15), 233–240 (2012)

33.190 A. Wu, Z. Zeng: Dynamics behaviors of memristor-based recurrent neural networks with time-varying delays, Neural Netw. **36**, 1–10 (2012)

33.191 A. Cichocki, R. Unbehauen: *Neural Networks for Optimization and Signal Processing* (Wiley, New York 1993)

33.192 J. Wang: Recurrent neural networks for optimization. In: *Fuzzy Logic and Neural Network Handbook*, ed. by C.H. Chen (McGraw-Hill, New York 1996), pp. 4.1–4.35

33.193 Y. Xia, J. Wang: Recurrent neural networks for optimization: The state of the art. In: *Recurrent Neural Networks: Design and Applications*, ed. by L.R. Medsker, L.C. Jain (CRC, Boca Raton 1999), 13–45

33.194 Q. Liu, J. Wang: Recurrent neural networks with discontinuous activation functions for convex optimization. In: *Integration of Swarm Intelligence and Artifical Neutral Network*, ed. by S. Dehuri, S. Ghosh, S.B. Cho (World Scientific, Singapore 2011), 95–119

33.195 I.B. Pyne: Linear programming on an electronic analogue computer, Trans. Am. Inst. Elect. Eng. **75**(2), 139–143 (1956)

33.196 L.O. Chua, G. Lin: Nonlinear programming without computation, IEEE Trans. Circuits Syst. **31**(2), 182–189 (1984)

33.197 G. Wilson: Quadratic programming analogs, IEEE Trans. Circuits Syst. **33**(9), 907–911 (1986)

33.198 J.J. Hopfield, D.W. Tank: Neural computation of decisions in optimization problems, Biol. Cybern. **52**(3), 141–152 (1985)

33.199 J.J. Hopfield, D.W. Tank: Computing with neural circuits – a model, Science **233**(4764), 625–633 (1986)

33.200 D.W. Tank, J.J. Hopfield: Simple *neural* optimization networks: an A/D converter, signal decision circuit, and a linear programming circuit, IEEE Trans. Circuits Syst. **33**(5), 533–541 (1986)

33.201 M.P. Kennedy, L.O. Chua: Neural networks for nonlinear programming, IEEE Trans. Circuits Syst. **35**(5), 554–562 (1988)

33.202 A. Rodriguez-Vazquez, R. Dominguez-Castro, A. Rueda, J.L. Huertas, E. Sanchez-Sinencio: Nonlinear switch-capacitor *neural* networks for optimization problems, IEEE Trans. Circuits Syst. **37**(3), 384–398 (1990)

33.203 S. Sudharsanan, M. Sundareshan: Exponential stability and a systematic synthesis of a neural network for quadratic minimization, Neural Netw. **4**, 599–613 (1991)

33.204 S. Zhang, A.G. Constantinides: Lagrange programming neural network, IEEE Trans. Circuits Syst. **39**(7), 441–452 (1992)

33.205 S. Zhang, X. Zhu, L. Zou: Second-order neural nets for constrained optimization, IEEE Trans. Neural Netw. **3**(6), 1021–1024 (1992)

33.206 A. Bouzerdoum, T.R. Pattison: Neural network for quadratic optimization with bound constraints, IEEE Trans. Neural Netw. **4**(2), 293–304 (1993)

33.207 M. Ohlsson, C. Peterson, B. Soderberg: Neural networks for optimization problems with inequality constraints: The knapsack problem, Neural Comput. **5**, 331–339 (1993)

33.208 J. Wang: Analysis and design of a recurrent neural network for linear programming, IEEE Trans. Circuits Syst. I **40**(9), 613–618 (1993)

33.209 W.E. Lillo, M.H. Loh, S. Hui, S.H. Zak: On solving constrained optimization problems with neural networks: A penalty method approach, IEEE Trans. Neural Netw. **4**(6), 931–940 (1993)

33.210 J. Wang: A deterministic annealing neural network for convex programming, Neural Netw. **7**(4), 629–641 (1994)

33.211 S.H. Zak, V. Upatising, S. Hui: Solving linear programming problems with neural networks: A comparative study, IEEE Trans. Neural Netw. **6**, 94–104 (1995)

33.212 Y. Xia, J. Wang: Neural network for solving linear programming problems with bounded variables, IEEE Trans. Neural Netw. **6**(2), 515–519 (1995)

33.213 M. Vidyasagar: Minimum-seeking properties of analog neural networks with multilinear objective functions, IEEE Trans. Autom. Control **40**(8), 1359–1375 (1995)

33.214 M. Forti, A. Tesi: New conditions for global stability of neural networks with application to linear and quadratic programming problems, IEEE Trans. Circuits Syst. I **42**(7), 354–366 (1995)

33.215 A. Cichocki, R. Unbehauen, K. Weinzierl, R. Holzel: A new neural network for solving linear programming problems, Eur. J. Oper. Res. **93**, 244–256 (1996)

33.216 Y. Xia: A new neural network for solving linear programming problems and its application, IEEE Trans. Neural Netw. **7**(2), 525–529 (1996)

33.217 X. Wu, Y. Xia, J. Li, W.K. Chen: A high-performance neural network for solving linear and quadratic programming problems, IEEE Trans. Neural Netw. **7**(3), 1996 (1996)

33.218 Y. Xia: A new neural network for solving linear and quadratic programming problems, IEEE Trans. Neural Netw. **7**(6), 1544–1547 (1996)

33.219 Y. Xia: Neural network for solving extended linear programming problems, IEEE Trans. Neural Netw. **8**(3), 803–806 (1997)

33.220 M.J. Perez-Ilzarbe: Convergence analysis of a discrete-time recurrent neural network to perform quadratic real optimization with bound constraints, IEEE Trans. Neural Netw. **9**(6), 1344–1351 (1998)

33.221 M.C.M. Teixeira, S.H. Zak: Analog neural nonderivative optimizers, IEEE Trans. Neural Netw. **9**(4), 629–638 (1998)

33.222 Y. Xia, J. Wang: A general methodology for designing globally convergent optimization neural networks, IEEE Trans. Neural Netw. **9**(6), 1331–1343 (1998)

33.223 E. Chong, S. Hui, H. Zak: An analysis of a class of neural networks for solving linear programming problems, IEEE Trans. Autom. Control **44**(11), 1995–2006 (1999)

33.224 Y. Xia, J. Wang: Global exponential stability of recurrent neural networks for solving optimization and related problems, IEEE Trans. Neural Netw. **11**(4), 1017–1022 (2000)

33.225 X. Liang, J. Wang: A recurrent neural network for nonlinear optimization with a continuously differentiable objective function and bound constraints, IEEE Trans. Neural Netw. **11**(6), 1251–1262 (2000)

33.226 Y. Leung, K. Chen, Y. Jiao, X. Gao, K. Leung: A new gradient-based neural network for solving linear and quadratic programming problems, IEEE Trans. Neural Netw. **12**(5), 1074–1083 (2001)

33.227 X. Liang: A recurrent neural network for nonlinear continuously differentiable optimization over a compact convex subset, IEEE Trans. Neural Netw. **12**(6), 1487–1490 (2001)

33.228 Y. Xia, H. Leung, J. Wang: A projection neural network and its application to constrained optimization problems, IEEE Trans. Circuits Syst. I **49**(4), 447–458 (2002)

33.229 R. Fantacci, M. Forti, M. Marini, D. Tarchi, G. Vannuccini: A neural network for constrained optimization with application to CDMA communication systems, IEEE Trans. Circuits Syst. II **50**(8), 484–487 (2003)

33.230 Y. Leung, K. Chen, X. Gao: A high-performance feedback neural network for solving convex nonlinear programming problems, IEEE Trans. Neural Netw. **14**(6), 1469–1477 (2003)

33.231 Y. Xia, J. Wang: A general projection neural network for solving optimization and related problems, IEEE Trans. Neural Netw. **15**, 318–328 (2004)

33.232 X. Gao: A novel neural network for nonlinear convex programming, IEEE Trans. Neural Netw. **15**(3), 613–621 (2004)

33.233 X. Gao, L. Liao, W. Xue: A neural network for a class of convex quadratic minimax problems with constraints, IEEE Trans. Neural Netw. **15**(3), 622–628 (2004)

33.234 Y. Xia, J. Wang: A recurrent neural network for nonlinear convex optimization subject to nonlinear inequality constraints, IEEE Trans. Circuits Syst. I **51**(7), 1385–1394 (2004)

33.235 M. Forti, P. Nistri, M. Quincampoix: Generalized neural network for nonsmooth nonlinear programming problems, IEEE Trans. Circuits Syst. I **51**(9), 1741–1754 (2004)

33.236 Y. Xia, G. Feng, J. Wang: A recurrent neural network with exponential convergence for solving convex quadratic program and linear piecewise equations, Neural Netw. **17**(7), 1003–1015 (2004)

33.237 Y. Xia, J. Wang: Recurrent neural networks for solving nonlinear convex programs with linear constraints, IEEE Trans. Neural Netw. **16**(2), 379–386 (2005)

33.238 Q. Liu, J. Cao, Y. Xia: A delayed neural network for solving linear projection equations and its applications, IEEE Trans. Neural Netw. **16**(4), 834–843 (2005)

33.239 X. Hu, J. Wang: Solving pseudomonotone variational inequalities and pseudoconvex optimization problems using the projection neural network, IEEE Trans. Neural Netw. **17**(6), 1487–1499 (2006)

33.240 S. Liu, J. Wang: A simplified dual neural network for quadratic programming with its KWTA application, IEEE Trans. Neural Netw. **17**(6), 1500–1510 (2006)

33.241 Y. Yang, J. Cao: Solving quadratic programming problems by delayed projection neural network, IEEE Trans. Neural Netw. **17**(6), 1630–1634 (2006)

33.242 X. Hu, J. Wang: Design of general projection neural network for solving monotone linear variational inequalities and linear and quadratic optimization problems, IEEE Trans. Syst. Man Cybern. B **37**(5), 1414–1421 (2007)

33.243 Q. Liu, J. Wang: A one-layer recurrent neural network with a discontinuous hard-limiting activation function for quadratic programming, IEEE Trans. Neural Netw. **19**(4), 558–570 (2008)

33.244 Q. Liu, J. Wang: A one-layer recurrent neural network with a discontinuous activation function for linear programming, Neural Comput. **20**(5), 1366–1383 (2008)

33.245 Y. Xia, G. Feng, J. Wang: A novel neural network for solving nonlinear optimization problems with inequality constraints, IEEE Trans. Neural Netw. **19**(8), 1340–1353 (2008)

33.246 M.P. Barbarosou, N.G. Maratos: A nonfeasible gradient projection recurrent neural network for equality-constrained optimization problems, IEEE Trans. Neural Netw. **19**(10), 1665–1677 (2008)

33.247 X. Hu, J. Wang: An improved dual neural network for solving a class of quadratic programming problems and its k-winners-take-all application, IEEE Trans. Neural Netw. **19**(12), 2022–2031 (2008)

33.248 X. Xue, W. Bian: Subgradient-based neural networks for nonsmooth convex optimization problems, IEEE Trans. Circuits Syst. I **55**(8), 2378–2391 (2008)

33.249 W. Bian, X. Xue: Subgradient-based neural networks for nonsmooth nonconvex optimization problems, IEEE Trans. Neural Netw. **20**(6), 1024–1038 (2009)

33.250 X. Hu, C. Sun, B. Zhang: Design of recurrent neural networks for solving constrained least absolute deviation problems, IEEE Trans. Neural Netw. **21**(7), 1073–1086 (2010)

33.251 Q. Liu, J. Wang: Finite-time convergent recurrent neural network with a hard-limiting activation function for constrained optimization with piecewise-linear objective functions, IEEE Trans. Neural Netw. **22**(4), 601–613 (2011)

33.252 Q. Liu, J. Wang: A one-layer recurrent neural network for constrained nonsmooth optimization, IEEE Trans. Syst. Man Cybern. **40**(5), 1323–1333 (2011)

33.253 L. Cheng, Z. Hou, Y. Lin, M. Tan, W.C. Zhang, F. Wu: Recurrent neural network for nonsmooth convex optimization problems with applications to the identification of genetic regulatory networks, IEEE Trans. Neural Netw. **22**(5), 714–726 (2011)

33.254 Z. Guo, Q. Liu, J. Wang: A one-layer recurrent neural network for pseudoconvex optimization subject to linear equality constraints, IEEE Trans. Neural Netw. **22**(12), 1892–1900 (2011)

33.255 Q. Liu, Z. Guo, J. Wang: A one-layer recurrent neural network for constrained pseudoconvex optimization and its application for dynamic portfolio optimization, Neural Netw. **26**(1), 99–109 (2012)

33.256 W. Bian, X. Chen: Smoothing neural network for constrained non-Lipschitz optimization with applications, IEEE Trans. Neural Netw. Learn. Syst. **23**(3), 399–411 (2012)

33.257 Y. Xia: An extended projection neural network for constrained optimization, Neural Comput. **16**(4), 863–883 (2004)

33.258 J. Wang, Y. Xia: Analysis and design of primal-dual assignment networks, IEEE Trans. Neural Netw. **9**(1), 183–194 (1998)

33.259 Y. Xia, G. Feng, J. Wang: A primal-dual neural network for online resolving constrained kinematic redundancy in robot motion control, IEEE Trans. Syst. Man Cybern. B **35**(1), 54–64 (2005)

33.260 Y. Xia, J. Wang: A dual neural network for kinematic control of redundant robot manipulators, IEEE Trans. Syst. Man Cybern. B **31**(1), 147–154 (2001)

33.261 Y. Zhang, J. Wang: A dual neural network for constrained joint torque optimization of kinematically redundant manipulators, IEEE Trans. Syst. Man Cybern. B **32**(5), 654–662 (2002)

33.262 Y. Zhang, J. Wang, Y. Xu: A dual neural network for bi-criteria kinematic control redundant manipulators, IEEE Trans. Robot. Autom. **18**(6), 923–931 (2002)

33.263 Y. Zhang, J. Wang, Y. Xia: A dual neural network for redundancy resolution of kinematically redundant manipulators subject to joint limits and joint velocity limits, IEEE Trans. Neural Netw. **14**(3), 658–667 (2003)

33.264 A. Cichocki, R. Unbehauen: Neural networks for solving systems of linear equations and related problems, IEEE Trans. Circuits Syst. I **39**(2), 124–138 (1992)

33.265 A. Cichocki, R. Unbehauen: Neural networks for solving systems of linear equations – part II: Minimax and least absolute value problems, IEEE Trans. Circuits Syst. II **39**(9), 619–633 (1992)

33.266 J. Wang: Recurrent neural networks for computing pseudoinverse of rank-deficient matrices, SIAM J. Sci. Comput. **18**(5), 1479–1493 (1997)

33.267 G.G. Lendaris, K. Mathia, R. Saeks: Linear Hopfield networks and constrained optimization, IEEE Trans. Syst. Man Cybern. B **29**(1), 114–118 (1999)

33.268 Y. Xia, J. Wang, D.L. Hung: Recurrent neural networks for solving linear inequalities and equations, IEEE Trans. Circuits Syst. I **46**(4), 452–462 (1999)

33.269 J. Wang: A recurrent neural network for solving the shortest path problem, IEEE Trans. Circuits Syst. I **43**(6), 482–486 (1996)

33.270 J. Wang: Primal and dual neural networks for shortest-path routing, IEEE Trans. Syst. Man Cybern. A **28**(6), 864–869 (1998)

33.271 Y. Xia, J. Wang: A discrete-time recurrent neural network for shortest-path routing, IEEE Trans. Autom. Control **45**(11), 2129–2134 (2000)

33.272 D. Anguita, A. Boni: Improved neural network for SVM learning, IEEE Trans. Neural Netw. **13**(5), 1243–1244 (2002)

33.273 Y. Xia, J. Wang: A one-layer recurrent neural network for support vector machine learning, IEEE Trans. Syst. Man Cybern. B **34**(2), 1261–1269 (2004)

33.274 L.V. Ferreira, E. Kaszkurewicz, A. Bhaya: Support vector classifiers via gradient systems with discontinuous right-hand sides, Neural Netw. **19**(10), 1612–1623 (2006)

33.275 J. Wang: Analysis and design of an analog sorting network, IEEE Trans. Neural Netw. **6**, 962–971 (1995)

33.276 B. Apolloni, I. Zoppis: Subsymbolically managing pieces of symbolical functions for sorting, IEEE Trans. Neural Netw. **10**(5), 1099–1122 (1999)

33.277 J. Wang: Analysis and design of k-winners-take-all model with a single state variable and Heaviside step activation function, IEEE Trans. Neural Netw. **21**(9), 1496–1506 (2010)

33.278 Q. Liu, J. Wang: Two k-winners-take-all networks with discontinuous activation functions, Neural Netw. **21**, 406–413 (2008)

33.279 Y. Xia, M. S. Kamel: Cooperative learning algorithms for data fusion using novel L1 estimation, IEEE Trans. Signal Process. **56**(3), 1083–1095 (2008)

33.280 B. Baykal, A.G. Constantinides: A neural approach to the underdetermined-order recursive least-squares adaptive filtering, Neural Netw. **10**(8), 1523–1531 (1997)

33.281 Y. Sun: Hopfield neural network based algorithms for image restoration and reconstruction – Part I: Algorithms and simulations, IEEE Trans. Signal Process. **49**(7), 2105–2118 (2000)

33.282 X.Z. Wang, J.Y. Cheung, Y.S. Xia, J.D.Z. Chen: Minimum fuel neural networks and their applications to overcomplete signal representations, IEEE Trans. Circuits Syst. I **47**(8), 1146–1159 (2000)

33.283 X.Z. Wang, J.Y. Cheung, Y.S. Xia, J.D.Z. Chen: Neural implementation of unconstrained minimum

L1-norm optimization—least absolute deviation model and its application to time delay estimation, IEEE Trans. Circuits Syst. II **47**(11), 1214–1226 (2000)

33.284 P.-R. Chang, W.-H. Yang, K.-K. Chan: A neural network approach to MVDR beamforming problem, IEEE Trans. Antennas Propag. **40**(3), 313–322 (1992)

33.285 Y. Xia, G.G. Feng: A neural network for robust LCMP beamforming, Signal Process. **86**(3), 2901–2912 (2006)

33.286 J. Wang, G. Wu: A multilayer recurrent neural network for on-line synthesis of minimum-norm linear feedback control systems via pole assignment, Automatica **32**(3), 435–442 (1996)

33.287 Y. Zhang, J. Wang: Global exponential stability of recurrent neural networks for synthesizing linear feedback control systems via pole assignment, IEEE Trans. Neural Netw. **13**(3), 633–644 (2002)

33.288 Y. Zhang, J. Wang: Recurrent neural networks for nonlinear output regulation, Automatica **37**(8), 1161–1173 (2001)

33.289 S. Hu, J. Wang: Multilayer recurrent neural networks for online robust pole assignment, IEEE Trans. Circuits Syst. I **50**(11), 1488–1494 (2003)

33.290 Y. Pan, J. Wang: Model predictive control of unknown nonlinear dynamical systems based on recurrent neural networks, IEEE Trans. Ind. Electron. **59**(8), 3089–3101 (2012)

33.291 Z. Yan, J. Wang: Model predictive control of nonlinear systems with unmodeled dynamics based on feedforward and recurrent neural networks, IEEE Trans. Ind. Inf. **8**(4), 746–756 (2012)

33.292 Z. Yan, J. Wang: Model predictive control of tracking of underactuated vessels based on recurrent neural networks, IEEE J. Ocean. Eng. **37**(4), 717–726 (2012)

33.293 J. Wang, Q. Hu, D. Jiang: A Lagrangian network for kinematic control of redundant robot manipulators, IEEE Trans. Neural Netw. **10**(5), 1123–1132 (1999)

33.294 H. Ding, S.K. Tso: A fully neural-network-based planning scheme for torque minimization of redundant manipulators, IEEE Trans. Ind. Electron. **46**(1), 199–206 (1999)

33.295 H. Ding, J. Wang: Recurrent neural networks for minimum infinity-norm kinematic control of redundant manipulators, IEEE Trans. Syst. Man Cybern. A **29**(3), 269–276 (1999)

33.296 W.S. Tang, J. Wang: Two recurrent neural networks for local joint torque optimization of kinematically redundant manipulators, IEEE Trans. Syst. Man Cybern. B **30**(1), 120–128 (2000)

33.297 W.S. Tang, J. Wang: A recurrent neural network for minimum infinity-norm kinematic control of redundant manipulators with an improved problem formulation and reduced architectural complexity, IEEE Trans. Syst. Man Cybern. B **31**(1), 98–105 (2001)

33.298 Y. Zhang, J. Wang: Obstacle avoidance for kinematically redundant manipulators using a dual neural network, IEEE Trans. Syst. Man Cybern. B **4**(1), 752–759 (2004)

33.299 Y. Xia, J. Wang, L.-M. Fok: Grasping force optimization of multi-fingered robotic hands using a recurrent neural network, IEEE Trans. Robot. Autom. **20**(3), 549–554 (2004)

33.300 Q. Liu, C. Dang, T. Huang: A one-layer recurrent neural network for real-time portfolio optimization with probability criterion, IEEE Trans. Cybern. **43**(1), 14–23 (2013)