

Chapter 2

Non-linear neuro-inspired circuits and systems: processing and learning issues

Abstract In this chapter the main elements useful for the design and realization of the neural architectures reported in the following chapters will be presented. Considering spiking and non-spiking neurons, the models used for implementing each of them, the synaptic models, the basic learning and plasticity algorithms and the network architectures will be introduced and analysed. The key elements that led to their selection and application in the developed neuro-inspired systems will be discussed briefly.

2.1 Introduction

To model neuro-inspired circuits and systems, we need basic blocks that can be combined with each other, to develop complex networks. In the following chapters, we will present neuro-inspired models designed to solve tasks that range from the control of locomotion to the learning of spatial memory and behavioural sequences. Depending on the peculiar characteristics of the modelled system, different basic processing units were considered. In the following sections, all the ingredients needed will be introduced and discussed.

As far as the neural models are considered, we can identify two elementary processing units: spiking and non-spiking neurons. Spiking neuron models are used to develop bio-inspired neural networks that try to mimic the information transfer that occurs in the brain which is mainly based on spiking events. The use of spikes improves the robustness of the network to noise because the spike can be more easily propagated through neurons even under low signal-to-noise conditions. Networks based on spiking neurons can perform different types of computations from classification to working-memory formation and sequence discrimination [1, 32, 17]. To connect spiking neurons with each other, different synaptic models can be considered [12, 27]. The synaptic transfer function can be either a simple weight or a more complex system with one or several state variables. Furthermore, the parameters of the synaptic model can be subject to a learning process because

it is important to store the acquired knowledge coming from the environment in the neural structure [6, 9].

Even if the spiking activity is the most common way to exchange information between in-vivo neurons, there are also examples of non-spiking neurons that regulate through slow dynamics specific rhythmic activities like in the Mollusk *Clione* mollusk [10]. They are also common in insect nervous systems, mainly to regulate motor activity [13] [29]. These neurons work like pulse generators and can also maintain a plateau when needed.

Even in presence of spiking neurons, there is the possibility to sum the activity of a group of spiking/bursting neurons using higher level indicators as for instance the mean firing rate that is a quantitative variable able to indicate the level of activity in a population of neurons. This strategy, exploited in several works, is part of the so-called *neural field* approach [18, 24].

The design of the basic blocks, together with the topology of the neural networks are relevant aspects to be addressed also in view of a software/hardware implementation of the architectures. Several solutions could be taken into account to develop neuromorphic circuits [20, 34, 14]: microcontrollers, FPGA, FPAA, integrated circuits and GPU accelerators are some of the potential devices that can be successfully considered.

2.2 Spiking neural models

Among the different models available for modelling spiking neurons, we considered different options: the leaky integrate-and-fire (LIF), the Izhikevich's model and the Morris-Lecar resonant neuron.

2.2.1 Leaky integrate-and-fire model

The leaky integrate-and-fire model is one of the first computational models applied to develop neuro-inspired networks [33][15]. Its main advantage consists of the simplicity of the equations. In fact, the system works like a leaky integrator. The time evolution of the membrane potential $V_m(t)$ of each neuron is described by the following equation:

$$C_m \dot{V}_m(t) = -g_L(V_m(t) - V_L) - I_{syn}(t) \quad (2.1)$$

where C_m is the membrane capacitance (typical values adopted are: 0.5 nF for excitatory cells and 0.2 nF for inhibitory cells); g_L is the leak conductance (0.025 μ S for excitatory cells and 0.02 μ S for inhibitory cells), and V_L is the resting potential (-70 mV for both excitatory and inhibitory cells). When the membrane potential of a neuron reaches a threshold (i.e. -50 mV) a spike occurs and the membrane potential

returns to a reset potential (i.e. -60 mV). The last term $I_{syn}(t)$ represents the total synaptic input current of the cell.

2.2.2 Izhikevich's neural model

Izhikevich's neural model, proposed in [22], is well known in the literature and offers many advantages from the computational point of view. The model is represented by the following differential equations:

$$\begin{aligned} \dot{v} &= 0.04v^2 + 5v + 140 - u + I \\ \dot{u} &= a(bv - u) \end{aligned} \quad (2.2)$$

with the spike-resetting

$$\text{if } v \geq 0.03, \text{ then } \begin{cases} v \leftarrow c \\ u \leftarrow u + d \end{cases} \quad (2.3)$$

where v is the membrane potential of the neuron, u is a recovery variable and I is the synaptic current. The values assigned to the parameters a, b, c and d will vary as they are depending on the particular neural structures that will be introduced. Varying the neuron parameters we can obtain several different behaviours [22].

2.2.3 Resonant neurons

In some applications it can be useful to enhance neuron sensitivity to the input stimulus timing or phase. For instance a stimulus, independently of its intensity cannot, to a certain extent, elicit any response if endowed with a given frequency or phase. Neurons able to show such a pattern of activity are called resonators. The Morris-Lecar neural model shows a robust tunable resonant behaviour [31, 19] through the following dynamics:

$$\begin{cases} \dot{V} = k_f [I + g_l(V_f - V) + g_k w(V_k - V) + \\ + g_{Ca} m_\infty(V)(V_{Ca} - V)] \\ \dot{\omega} = k_f [\lambda(V)(\omega_\infty(V) - \omega)] \end{cases} \quad (2.4)$$

where

$$\begin{aligned} m_\infty(V) &= \frac{1}{2} \left(1 + \tanh \frac{V - V_1}{V_2} \right) \\ \omega_\infty(V) &= \frac{1}{2} \left(1 + \tanh \frac{V - V_3}{V_4} \right) \\ \lambda(V) &= \frac{1}{3} \cosh \frac{V - V_3}{2V_4} \end{aligned}$$

V and ω are the state variables of the system, I is the input and $V_i, V_t, V_k, V_{Ca}, g_l, g_k, g_{Ca}, I$ and k_f are parameters of the model whose typical values are indicated in Table 2.1.

Table 2.1 Parameters of the Morris-Lecar neurons.

V_1	V_2	V_3	V_4	V_t	V_k
-1.2	18	2	30	-60	-84
g_l	g_k	V_{Ca}	g_{Ca}	I	k_f
2	8	120	4.4	[60.5 61.5]	[0.687 2.5]

2.3 Synaptic models

Neurons are connected through synapses; the synaptic model transforms the spiking dynamics of the pre-synaptic neuron into a current that excites the post-synaptic one. The mathematical response of the synapses to a pre-synaptic spike can be ruled by the following equation:

$$\varepsilon(t) = \begin{cases} Wt/\tau \exp(1-t/\tau), & \text{if } t > 0 \\ 0, & \text{if } t < 0 \end{cases} \quad (2.5)$$

where t is the time elapsed since the emitted spike, τ is the time constant and W is the efficiency of the synapse. This last parameter can be modulated with experience. This model represents the impulse response of the pre-synaptic neuron; it can be cumulated if multiple spikes are emitted within the active window, and in relation to the chosen time constant.

2.3.0.1 Synaptic adaptation through learning

The Spike-timing-dependent-plasticity STDP can reproduce Hebbian learning in biological neural networks [8, 30]. The algorithm works on the synaptic weights, modifying them according to the temporal sequence of occurring spikes. The updating rule can be expressed by the following formula:

$$\delta W = \begin{cases} A^+ \exp(\delta t/\tau^+), & \text{if } \delta t < 0 \\ -A^- \exp(\delta t/\tau^-), & \text{if } \delta t > 0 \end{cases} \quad (2.6)$$

where δt is the time delay between pre- and post-synaptic spikes. In this way the synapse is reinforced if the pre-synaptic spike happens before the post-synaptic one; it is weakened in the opposite situation. Parameters τ_+ and τ_- represent the slope

of exponential functions, whereas positive constants A_+ and A_- represent the maximum variation of the synaptic weight.

2.3.1 Synaptic model with facilitation

Short-term synaptic plasticity is characterized by different phenomena and mechanisms. Short-term facilitation is one of these mechanisms that contribute to the evaluation of the synaptic efficacy. A facilitating dynamics can be modeled with the following dynamical system [21]:

$$\dot{u} = (U - u)/F + k\delta(t - t_n) \quad (2.7)$$

where U is a constant which the membrane potential tends to, F is related to the fading memory and k is a gain factor. $\delta(t - t_n)$ represents the spike emitted at time t_n by the pre-synaptic neuron.

2.4 The liquid state network

Besides simple networks using spiking neurons, the main structure able to generate neural dynamics important for the models presented in the following chapters is the Liquid state network (LSN). This structure was introduced taking, from the one side, inspiration from specific parts of the insect brain, like mushroom bodies (MBs; see chapter 1), and on the other side, to already existing architectures and related algorithms. From the latter side, among the different kinds of neural networks used for solving problems like navigation [31], multi-link system control [16] and classification, a lot of interest was devoted to Reservoir computing, which mainly includes two different approaches: Echo State Network (ESN) and Liquid State Machines (LSM) [23, 25]. In previous studies non-spiking Recurrent Neural Networks were used to model the MBs' memory and learning functions [3]. The core of the proposed architecture, inspired by the biology of MBs, resembles the LSM architecture. It consists of a large collection of spiking neurons, the so-called liquid layer, receiving time-varying inputs from external sources as well as recurrent connections from other nodes of the same layer. The recurrent structure of the network turns the time-dependent input into spatio-temporal patterns in the neurons. These patterns are read-out by linear discriminant units. In the last years LSM became a reference point for replicating brain functionalities. However, there is no guaranteed way to analyse the role of each single neuron activity for the overall network dynamics: the control over the process is very weak. This apparent drawback is a consequence of the richness of the dynamics potentially generated within the liquid layer. The side advantage is that the high dimensional complexity can be concurrently exploited

through several projections (the read-out maps) to obtain non-linear mappings useful for performing different tasks at the same time.

In details, the architecture used in the following applications consists of a lattice of Izhikevich's class I neurons. An important characteristics of this structure is the local connectivity that is a relevant added value in view of a hardware implementation of the model. Considering that, as recently found, in insects memory-relevant neural connections are excitatory [11] the network used in the following models is composed by excitatory (75%) and inhibitory (25%) neurons. Moreover, in our model the synaptic weight values are randomly distributed between -0.5 and 0.5 whereas the input weights are fixed to 1. The generation of the synaptic connections within the lattice is based on a probability that is a function of the distance $d_{i,j}$ between the presynaptic (i) and postsynaptic (j) neurons.

$$P_{ij} = k * C_{i,j} \quad (2.8)$$

where $C_{inh,inh} = 0.2, C_{inh,exc} = 0.8, C_{exc,inh} = 0.4, C_{exc,exc} = 0.6$. and

$$\begin{aligned} k &= 1 \quad \text{if } d_{i,j} \leq 1 \\ k &= 0.5 \quad \text{if } 1 < d_{i,j} \leq 2 \\ k &= 0 \quad \text{if } d_{i,j} > 2 \end{aligned} \quad (2.9)$$

The parameters $C_{i,j}$ have been chosen according to [25]. The distance is calculated considering the neurons distributed on a regular grid with toroidal boundary conditions. The distance $d_{i,j} = 1$ is considered for both horizontally and vertically adjacent neurons. The time constant of the synaptic model was randomly chosen among the values $\tau = 2.5, 5, 15$ and $25ms$. This variability showed to improve the dynamics that can be generated inside the network within the processing time window. The rich dynamics from the LSN is collected into a readout map, whose weights are randomly initialized around zero and are subject to learning. The neurons of the readout map, called Sum neurons, possess a linear activation function and are massively connected with the LSN.

2.4.1 Learning in the LSN

Inputs are provided to the network as currents that, through a sparse connection, reach the hidden lattice (i.e. the liquid layer). The multiple read-out maps can be learned considering the error between the network output, collected through each Sum neuron for each read-out map, and the target signal.

To enslave the dynamics of the LSN to follow a given target, depending on the model needs, a supervised learning method can be performed. A simple solution consists in a batch algorithm, based on the Moore-Penrose Pseudo-inverse computing method, that determines all the weights needed in the read-out map using all the available data in a single iteration. If either the dataset is not completely avail-

able since the beginning, or a more biologically grounded solution is preferred, an incremental approach can be adopted.

In this case, for a given sum neuron s , the weight value at each integration step depends on the lattice activity and on the error between the current output and the desired target value. This can be summarized in the following equation:

$$W_{i,j}^s(t+1) = W_{i,j}^s(t) + \eta * Z_{i,j}(t) * E^s(t) \quad (2.10)$$

where η is the learning factor, $Z_{i,j}(t)$ is the synaptic output of the neuron (i, j) of the lattice at time t and $E^s(t)$ is the error between the desired target and the neuron s . Similar results could be obtained cumulating the error in each presented sample and updating the weights at the end of the presentation. Moreover, to stabilize the learning process, an error threshold ($E_{th} = 10^{-8}$) is imposed to avoid a negligible weight update for each sample presentation.

2.5 Non-spiking neurons for locomotion

Non-spiking neurons are perfect candidate for reproducing rhythmic movements that are commonly observed in motor activities. One of the main paradigms of neural networks tailored to work as locomotion controllers is the Central Pattern Generator (CPG) [4].

Among the large number of different implementations of the CPG paradigm, the following architecture will be used in the next chapters.

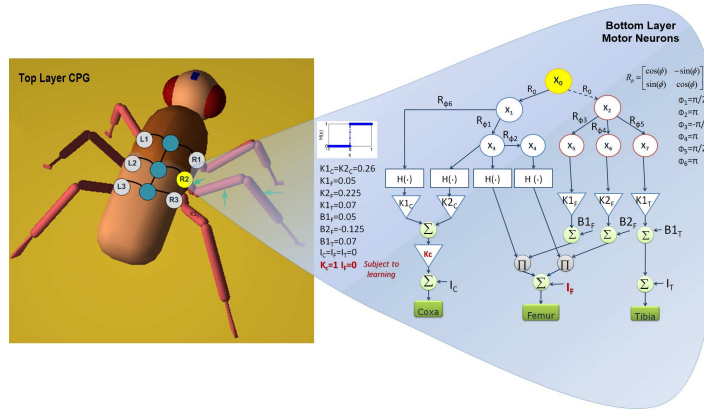


Fig. 2.1 Neural network scheme: the top layer generates a stable gait pattern, whereas the bottom layer consists of additional sub-networks generating the specific reference signals for the leg joints. The network devoted to control a middle leg is reported. The parameters adapted during the learning process for the middle legs are indicated in red [2].

The basic cell characterizing the CPG network is described by the following equations:

$$\begin{cases} \dot{x}_{1,i} = -x_{1,i} + (i + \mu + \varepsilon)y_{1,i} - s_1 y_{2,i} + i_1 \\ \dot{x}_{2,i} = -x_{2,i} + s_2 y_{1,i} + (i + \mu - \varepsilon)y_{2,i} + i_2 \end{cases} \quad (2.11)$$

with $y_i = \tanh(x_i)$ and the parameters for each cell: $\mu = 0.23$, $\varepsilon = 0$, $s_1 = s_2 = 1$, $i_1 = i_2 = 0$, the system will generate a stable limit cycle [8]. μ is chosen to approximate the dynamics to a harmonic (non-spiking) oscillation. A suitable modulation of these parameters can modify the dynamics of the cell, in order to show also a rhythmic spiking activity. The CPG network is built by connecting neighbouring cells via links expressing rotational matrices $R(\phi)$, as follows:

$$\dot{x}_i = f(x_i, t) + k \sum_{j \neq i} (R(\phi_{i,j})x_j - x_i) \text{ with } i, j = 1, \dots, n \quad (2.12)$$

where the summation involves all the neurons j which are nearest neighbours to the neuron i ; n is the total number of cells; $f(x_i, t)$ represents the reactive dynamics of the i -th uncoupled neurons as reported in eq. (2.11) and k is the strength of the connections. The sum of these terms performs diffusion on adjacent cells and induces phase-locking as a function of rotational matrices [28].

The CPG structure main role is to generate a given stable phase shift among a number of oscillating cells. Of course an interface to be adapted to the peculiarities of the robot kinematics. Moreover, once the robot has walked, each leg has basically two distinct phases: the stance phase (when the leg is on the ground and supports the weight of the body), and the swing phase (when the leg is lifted-off and recovers). In the case of a *Drosophila*-like hexapod simulated robot, the network controlling one of the middle legs is sketched in Fig. 2.1. The CPG neuron identified with the label *R2* is connected through rotational matrices with different phases to *Top layer CPG*, a network of motor neurons arranged implementing eq.(2.12) in a directed tree graph, using eq. (2.11) as the constituting neuron model. The blocks $H(\bullet)$ in Fig. 2.1 are Heaviside functions, used to distinguish, within the limit cycle, between the stance and swing phases: this allows to associate suitable modulation parameters to each part of the cycle, depending on the morphology of the leg. The signals are finally merged to generate the position control command for the coxa, femur and tibia joints. A detailed discussion on the CPG structure and behaviours can be found in a previous study [2].

2.6 Conclusions

In this chapter the ingredients that will be used to develop the neuro-inspired models are introduced. Different neuron models have been considered including spiking and non-spiking ones. Synaptic connections are fundamental to develop networks of neurons. Simple synaptic models have been considered including learning mechanisms that modify the synaptic efficiency depending on the pre- and post- synaptic neuron activity. A first example of complete network was briefly discussed illustrat-

ing the learning mechanism needed to enslave the internal dynamics using read-out maps that can be associated to different behavioural responses. Non-spiking neurons finally presented with particular attention to the models used to develop locomotion controllers for bio-inspired robots.

References

1. Abbott, L., DePasquale, B., Memmesheimer, R.: Building functional networks of spiking model neurons. *Nature Neuroscience* **19**(3), 350–355 (2016)
2. Arena, E., Arena, P., Patané, L.: CPG-based locomotion generation in a *Drosophila*-inspired legged robot. In: *Biorob 2012*, pp. 1341 – 1346. Roma, Italy (2012)
3. Arena, E., Arena, P., Strauss, R., Patané, L.: Motor-skill learning in an insect inspired neuro-computational control system. *Frontiers in Neurobotics* **11**, 12 (2017). DOI 10.3389/fnbot.2017.00012. URL <http://journal.frontiersin.org/article/10.3389/fnbot.2017.00012>
4. Arena, P.: The central pattern generator: A paradigm for artificial locomotion. *Soft Computing* **4**(4), 251–265 (2000). URL www.scopus.com. Cited By :19
5. Arena, P., Caccamo, S., Patané, L., Strauss, R.: A computational model for motor learning in insects. In: *International Joint Conference on Neural Networks (IJCNN)*, pp. 1349–1356. Dallas, TX, August 4-9 (2013)
6. Arena, P., De Fiore, S., Patané, L., Pollino, M., Ventura, C.: Stdp-based behavior learning on tribot robot. In: *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 7365, pp. 1–11 (2009). DOI 10.1117/12.821380
7. Arena, P., Fortuna, L., Frasca, M., Patané, L.: A CNN-based chip for robot locomotion control. *IEEE Trans. on Circuits and Systems I* **52**(9), 1862–1871 (2005)
8. Arena, P., Fortuna, L., Frasca, M., Patané, L.: Learning anticipation via spiking networks: application to navigation control. *IEEE Transaction on Neural Networks* **20**(2), 202–216 (2009)
9. Arena, P., Patané, L.: Simple sensors provide inputs for cognitive robots. *IEEE Instrumentation and Measurement Magazine* **12**(3), 13–20 (2009). DOI 10.1109/MIM.2009.5054548
10. Arshavsky, Y.I., Beloozerova, I.N., Orlovsky, G.N., Panchin, Y.V., Pavlova, G.A.: Control of locomotion in marine mollusc *clione limacina* iii. on the origin of locomotory rhythm. *Experimental Brain Research* **58**(2), 273–284 (1985)
11. Barnstedt, O., David, O., Felsenberg, J., Brain, R., Moszynski, J., Talbot, C., Perrat, P., Waddell, S.: Memory-relevant mushroom body output synapses are cholinergic. *Neuron* **89**(6), 1237–1247 (2017). DOI 10.1016/j.neuron.2016.02.015
12. Brette, R., Rudolph, M., Carnevale, T., Hines, M., Beeman, D., Bower, J.M., Diesmann, M., Morrison, A., Goodman, P.H., Harris, F.C., Zirpe, M., Natschläger, T., Pecevski, D., Ermentrout, B., Djurfeldt, M., Lansner, A., Rochel, O., Vieville, T., Muller, E., Davison, A.P., El Boustani, S., Destexhe, A.: Simulation of networks of spiking neurons: A review of tools and strategies. *Journal of Computational Neuroscience* **23**(3), 349–398 (2007)
13. Büschges, A., Wolf, H.: Nonspiking local interneurons in insect leg motor control. i. common layout and species-specific response properties of femur-tibia joint control pathways in stick insect and locust. *Journal of Neurophysiology* **73**(5), 1843–1860 (1995). URL <http://jn.physiology.org/content/73/5/1843>
14. Chen, Q., Wang, J., Yang, S., Qin, Y., Deng, B., Wei, X.: A real-time {FPGA} implementation of a biologically inspired central pattern generator network. *Neurocomputing* **244**, 63 – 80 (2017). DOI <https://doi.org/10.1016/j.neucom.2017.03.028>
15. Compte, A., Brunel, N., Goldman-Rakic, P., Wang, X.: Synaptic mechanisms and network dynamics underlying spatial working memory in a cortical network model. *Cerebral Cortex* **10**, 910–923 (2000)

16. Cruse, H.: MMC - a new numerical approach to the kinematics of complex manipulators. *Mechanisms and Machine Theory* **37**, 375–394 (2002)
17. Durstewitz, D., Seamans, J., Sejnowski, T.: Neurocomputational models of working memory. *Nature Neuroscience* **19**(3), 1184 – 1191 (2000)
18. Erlhagen, W., Bicho, E.: The dynamic neural field approach to cognitive robotics. *Journal of Neural Engineering* **3**(3), R36 (2006)
19. Hoppensteadt, F., Izhikevich, E.: *Brain Theory and Neural Networks*, vol. 181–186, MIT press Cambridge edn. Arbib MA (2002)
20. Indiveri, G., Linares-Barranco, B., Hamilton, T., van Schaik, A., Etienne-Cummings, R., Delbruck, T., Liu, S.C., Dudek, P., Hfliger, P., Renaud, S., Schemmel, J., Cauwenberghs, G., Arthur, J., Hynna, K., Folorosele, F., SAGHI, S., Serrano-Gotarredona, T., Wijekoon, J., Wang, Y., Boahen, K.: Neuromorphic silicon neuron circuits. *Frontiers in Neuroscience* **5** (73), 1–23 (2011). DOI 10.3389/fnins.2011.00073
21. Izhikevich, E., Desai, N., Walcott, E., Hoppensteadt, F.: Bursts as a unit of neural information: selective communication via resonance. *TRENDS in Neurosciences* **26**(3), 161–167 (2003)
22. Izhikevich, E.M.: Which model to use for cortical spiking neurons? *IEEE Transactions on Neural Networks* **15**(5), 1063–1070 (2004)
23. Jaeger, H.: The "echo state" approach to analysing and training recurrent neural networks. GMD-Report German National Research Institute for Computer Science **148** (2001)
24. Johnson, J.S., Spencer, J.P., Luck, S.J., Schonher, G.: A dynamic neural field model of visual working memory and change detection. *Psychological Science* **20**(5), 568–577 (2009). DOI 10.1111/j.1467-9280.2009.02329.x
25. Maass, W., Natschlger, T., Markram, H.: Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural Computation* **14**(11), 2531–2560 (2002)
26. Morris, C., Lecar, H.: Voltage oscillations in the barnacle giant muscle fiber. *Biophysical Journal* **35**, 193–213 (1981)
27. Morrison, A., Diesmann, M., Gerstner, W.: Phenomenological models of synaptic plasticity based on spike timing. *Biological Cybernetics* **98**(6), 459–478 (2008)
28. Seo, K., Slotine, J.: Models for global synchronization in cpg-based locomotion. In: *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pp. 281–286 (2007)
29. Siegler, M.V.: Nonspiking interneurons and motor control in insects. *Advances in Insect Physiology* **18**, 249 – 304 (1985). DOI [http://dx.doi.org/10.1016/S0065-2806\(08\)60042-9](http://dx.doi.org/10.1016/S0065-2806(08)60042-9). URL <http://www.sciencedirect.com/science/article/pii/S0065280608600429>
30. Song, S., Miller, K.D., Abbott, L.F.: Competitive Hebbian learning through spike-timing-dependent plasticity. *Nature Neuroscience* **3**, 919–926 (2000)
31. Tani, J.: Model-based learning for mobile robot navigation from the dynamical systems perspective. *IEEE Transactions System, Man and Cybernetics (Part B)* **26**(3), 421–436 (1996)
32. Thalmeier, D., Uhlmann, M., Kappen, H.J., Memmesheimer, R.M.: Learning universal computations with spikes. *PLOS Computational Biology* **12**(6), 1–29 (2016). DOI 10.1371/journal.pcbi.1004895. URL <https://doi.org/10.1371/journal.pcbi.1004895>
33. Tuckwell, H.: *Introduction to theoretical neurobiology*. Cambridge UP (1988)
34. Wang, R., Cohen, G., Stiefel, K., Hamilton, T., Tapson, J., van Schaik, A.: An fpga implementation of a polychronous spiking neural network with delay adaptation. *Frontiers in Neuroscience* **7**(14), 1–14 (2013). DOI 10.3389/fnins.2013.00014