Giacomo Indiveri and Yulia Sandamirskaya

The Importance of Space and Time for Signal Processing in Neuromorphic Agents

The challenge of developing low-power, autonomous agents that interact with the environment

©ISTOCKPHOTO.COM/JUST_SUPER

Digital Object Identifier 10.1109/MSP.2019.2928376
Date of current version: 29 October 2019

Artificial neural networks (ANNs) and computational neuroscience models have made tremendous progress, enabling us to achieve impressive results in artificial intelligence applications, such as image recognition, natural language processing, and autonomous driving. Despite this, biological neural systems consume orders of magnitude less energy than today's ANNs and are much more flexible and robust. This adaptivity and efficiency gap is partially explained by the computing substrate of biological neural processing systems that is fundamentally different from the way today's computers are built. Biological systems use in-memory computing elements operating in a massively parallel way rather than time-multiplexed computing units that are reused in a sequential fashion. Moreover, the activity of biological neurons follows continuous-time dynamics in real, physical time instead of operating on discrete temporal cycles abstracted away from real time.

Here, we present neuromorphic processing devices that emulate the biological style of processing by using parallel instances of mixed-signal analog/digital circuits that operate in real time. We argue that this approach brings significant advantages in efficiency of computation and show examples of embodied neuromorphic agents that use such devices to interact with the environment and exhibit autonomous learning.

Introduction

Tremendous progress in machine learning and computational neuroscience is leading to the development of neural processing algorithms that may have a far-reaching impact on our daily lives [1]. For example, recently developed deep and convolutional neural network algorithms can be trained to perform remarkably well in pattern-recognition tasks, in some cases even outperforming humans. Typically, these algorithms run on conventional computing systems based on the von Neumann architecture and, consequently, are commonly implemented using large and power-hungry platforms, sometimes distributed across multiple machines in server farms. The power required to run these algorithms and achieve impressive results is orders of magnitude larger than the power

 1053-5888/19©2019IEEE

used by biological nervous systems that once served as inspiration for the ANNs. This high power consumption is not sustainable for the future needs of ubiquitous computing at scale.

The reasons for the large gap in energy costs between artificial and natural neural processing systems are still being investigated; however, it is clear that one fundamental difference lies in the way the elementary computing processes are organized: Conventional computers use Boolean logic, bit-precise digital representations, and time-multiplexed and clocked operations. Nervous systems, on the other hand, carry out robust and reliable computation using analog components that are inherently noisy and operate in continuous time and activation domains. These components typically communicate among each other with all-or-none discrete events (spikes), thus, using a combination of analog computation and digital communication. Moreover, they form distributed, event-driven, and massively parallel systems, and they feature a considerable amount of adaptation, self-organization, and learning, with dynamics that operate on a multitude of timescales.

One promising approach for bridging the efficiency gap between the biological and artificial neural systems is to develop a new generation of ultralow-power and massively parallel computing technologies that are optimally suited to implement neural network architectures that use the same principles of computation used by the brain. This "neuromorphic engineering" approach [2] was originally proposed in the early 1990s [3] but now takes advantage of the progress made in very large-scale integration (VLSI) technologies to scale up the number of neurons and synapses per device, as well as the development in computational neuroscience to implement more complex, spike-based signal processing and learning architectures. Neuromorphic circuits are typically designed using analog, digital, or mixed-mode analog/digital CMOS transistors, and are well suited for exploiting the features of emerging memory technologies and new memristive materials [4]–[6]. Similar to the biological systems they model, neuromorphic systems process information using energy-efficient, asynchronous, event-driven methods [7]; they are often adaptive and fault-tolerant, and they can be flexibly configured to display complex behaviors by combining multiple instances of simpler elements.

Remarkable neuromorphic computing platforms have been developed in the past for modeling cortical circuits, solving pattern recognition problems, and implementing machine learning tasks [8]–[19], as well as for accelerating the simulation of computational neuroscience models [15], [17]. In parallel, impressive, full-custom dedicated accelerators have been proposed for implementing convolutional and deep network algorithms following the more conventional digital design flow [20], [21]. While these systems significantly reduce the power consumption in a wide variety of neural network applications compared to conventional computing approaches, they still fall short of reproducing the power, size, and adaptivity of biological neural processing systems that can produce adaptive behavior that consumes just a few watts of power. Indeed, developing low-power, compact, and autonomous electronic agents that can interact with the environment in real time is still an open challenge. Such agents must be capable to extract relevant information from the signals they sense, adapt to the changes and uncertain conditions present in the external inputs and internal states, and learn to produce context-dependent behaviors for carrying out goal-directed procedural tasks.

We argue that to address this challenge and find a computational substrate that minimizes size and power consumption in real-time behaving systems, the computing hardware in which computation is realized needs to match the computing principles underlying autonomous adaptive behavior. In particular, to fully benefit from the emulation of biological neural processing systems, it is important to preserve two of their fundamental characteristics: the explicit representation of time and the explicit use of space, instantiating dedicated physical circuits for each neuron/synapse element and avoiding the use of time multiplexing. In the following, we first present the design principles for building neuromorphic electronic systems that make use of these representations and then show how such explicit representation of time and space matches a computing framework of dynamic neural fields that embody principles of autonomous behavior generation and learning [22], [23]. Finally, we demonstrate successful realizations of autonomous neural-dynamic architectures in neuromorphic hardware, emphasizing the role of using physical time and space representation in hardware for efficiency of the autonomous neuronal controllers.

Design principles for building biologically plausible neuromorphic processors

Dedicated neuronal circuits in hardware neural-processing systems

In an effort to minimize silicon-area consumption, digital neuromorphic processors typically use time-multiplexing techniques to share circuits that simulate neural dynamics for modeling multiple neurons [16], [18], [19]. This requires that the shared circuits continuously transfer their state variables to and from an external memory block at each update cycle (therefore, burning extra power for the data transfer). The faster the transfer and cycle rate, the larger the number of neurons that can be simulated per time unit. In addition, if these circuits need to model processes that evolve continuously over natural time, such as the leak term of a leaky integrate and fire (I&F) neuron model, it is necessary to include a clock to update the related state variables periodically and manage the passing of time (thus, adding extra overhead and power consumption).

Unlike digital simulators of neural networks, analog neuromorphic circuits use the physics of silicon to directly emulate neural and synaptic dynamics [2]. In this case, the state variables evolve naturally over time, and "time represents itself" [3], bypassing the need to have clocks and extra circuits to manage the representation of time. Furthermore, since the state variable memory is held in the synapse and neuron capacitors, there is no need to transfer data to extra memory blocks, dramatically saving energy that would otherwise be required to shift the neuron state variables back and forth from the memory. Examples of neuromorphic architectures that follow this

mixed-signal approach include the Italian Istituto Superiore di Sanità (ISS) learning attractor neural network that has plasticity and a long-term memory chip (LANN-21) [8]; ISS "final learning attractor neural network" (F-LANN) chip [9]; Georgia Tech learning-enabled neuron array integrated circuit (LENA-IC) [10]; University of California San Diego integrate-and-fire array transceiver (IFAT) architecture [11]; Stanford University, California, Neurogrid system [12]; University of Zürich recurrent online learning spiking (ROLLS) neuromorphic processor [14]; and University of Zürich dynamic neuromorphic asynchronous processor- scalable (DYNAP-SE) chip [13].

In these devices, the analog synapse and neuron circuits have no active components [2]. The circuits are driven directly by the input "streaming" data. Their synapses receive input spikes, and their neurons produce output spikes at the rate of the incoming data. So, if they are not processing data, there is no energy dissipated per synaptic operation (SOP) and no dynamic power consumption. Therefore, this approach is particularly attractive in the case of applications in which the signals have sparse activity in space and time. Under these conditions, most neurons would be silent at any one time, thus, bringing the system power consumption to the minimum.

A quantitative comparison of the specifications of these devices is presented in Table 1. The spike-based learning algorithms that some of these devices implement are based on the basic version of spike-timing-dependent plasticity (STDP) [24] or the more elaborate spike-timing and rate-dependent plasticity (STRDP) [25].

While this approach is very power efficient, it is necessary to instantiate and place one distinct physical neuromorphic circuit per emulated synapse or neuron, therefore, requiring a physical area that scales with the number of synapses and neurons in the network. This was a serious limiting factor with older VLSI fabrication processes. However, technology scaling pushed by Moore's law, and the emergence of novel nanoscale memory device technologies that can be used to implement synaptic and neural dynamics [5], [6], bring renewed interest to this approach and make it very competitive compared to the classical one of resorting to more compact digital designs based on shared time-multiplexed circuits, as was done, for example, for the IBM TrueNorth [16] and Intel Loihi [19] neuromorphic processors.

A clear difference between the use of pure digital circuits versus mixed-signal analog/digital ones is the higher sensitivity to noise and variability of the latter. The higher robustness of digital circuits is given by their feature of restoring signals to high or low states at each processing stage. This ensures accurate representation of signals and is a crucial requirement for deterministic and reproducible computation using the standard computing paradigms based on Boolean logic. However, animal brains are an existence proof that noisy and analog computational elements can, indeed, be used to perform reliable computation in sensory processing tasks at much lower power. Within this context, mixed-signal neuromorphic processors represent a promising approach that can potentially lead to robust computation by using the same strategies used by real nervous systems, such as adaptation, learning, and smoothing over space (for example, with population coding [26]) and time (through the low-pass filtering property of transferring the function of the physical computing elements).

Natural time in hardware neural processing systems

For the approach based on parallel instances of mixed analog/digital circuits as described previously, the most power-efficient way of processing time-varying signals is to use circuit time constants that are well matched to those of the dynamics of the signals that need to be processed. For example, real-world "natural" events and signals, such as speech, biosignals measured from the body, human gestures, and motor and sensory signals measured from roving robots, would require the synapse and neural circuits to have time constants in the range of 5–500 ms. It is important to realize that although the speed of the individual synapse or neuron computing elements in such architectures is very slow (for example, compared to digital circuits), the response time of the overall system can be extremely fast. This is due to the fact that the distributed parallel processing nodes in the system will be affected by device mismatch and have different initial conditions; so, upon the arrival of an external input, there will be many neurons very close to their firing threshold, and they will produce an output with a latency that is much shorter than their natural integration time constant. While these fast reaction times can be achieved with any (mismatched and matched) system, for example, via stochastic resonance, neural

Table 1. The quantitative comparison of mixed-signal neuromorphic processor specifications.

	LANN-21 [8]	F-LANN [9]	LENA-IC [10]	IFAT [11]	NeuroGrid [12]	ROLLS [14]	DYNAP-SE [13]
Technology	0.6 μm	0.6 μm	0.35 μm	90 nm	0.18 μm	0.18 μm	0.18 μm
Supply Voltage	3.3 V	3.3 V	2.4 V	1.2 V	3 V	1.8 V	1.8 V
Core Area	10 mm^2	68.9 mm^2	25 mm^2	139 μm^2	170 mm^2	51.4 mm^2	7.5 mm^2
Neurons/Core	21	128	100	2,000	65,636	256	256
Synapses/Core	129	16,384	30,000	N/A	4,096	128,000	16,000
Fan-In/Fan-Out	21/21	128/128	100/100	N/A	N/A	256/256	64/4,000
Synaptic Weight	Capacitor	Capacitor	>10 bits	8 bits	13-bit shared	Capacitor	1 + 1 bit
Online Learning	STRDP	STRDP	STDP	No	No	STRDP	No
Neurons/mm^2	174	N/A	27	7,142	360	1,089	812
Energy per SOP	N/A	N/A	10 pJ	22 pJ	31.2 pJ	77 fJ	17 pJ

fJ: femtojoule; pJ: picojoule.

networks that use dedicated circuits per neuron and that are affected by mismatch do not need to implement explicit random number generators and cycle through each time-multiplexed neuron to inject the noise needed to achieve this effect.

The relatively long time constants required by this approach are not easy to realize using analog CMOS technology. Standard analog circuit design techniques either lead to bulky and silicon-area expensive solutions or fail to meet this condition, resorting to modeling neural dynamics at accelerated timescales [27]. One elegant solution to this problem is to combine the use of subthreshold circuit design techniques [3] with a current-mode design [28]. A very compact subthreshold log-domain circuit that can reproduce biologically plausible synaptic and neural temporal dynamics and that has been used in a wide variety of neuromorphic applications [25], [29]–[31] is the differential pair integrator (DPI) circuit [32], depicted in Figure 1. The analytic transfer function of this circuit can be derived following a translinear-loop log-domain analysis [2], [28]. From Figure 1(b), it follows that

$$\tau\left(1+\frac{I_g}{I_{\text{out}}}\right)\frac{d}{dt}I_{\text{out}}+I_{\text{out}}=\frac{I_g I_{\text{in}}}{I_\tau}-I_g, \tag{1}$$

where I_{out} represents the synaptic or neuron dynamic variable, I_{in} the input signal, I_τ a user-defined leakage current, and I_g a global gain term useful, for example, for modeling spike-based learning, intrinsic plasticity, and synaptic scaling. This is a nonlinear differential equation. However, under the reasonable assumptions of nonnegligible input currents, such that $I_{\text{in}} \gg I_\tau$, and observing that, for such conditions, the output current I_{out} will eventually grow to be $I_{\text{out}} \gg I_g$, this equation simplifies to

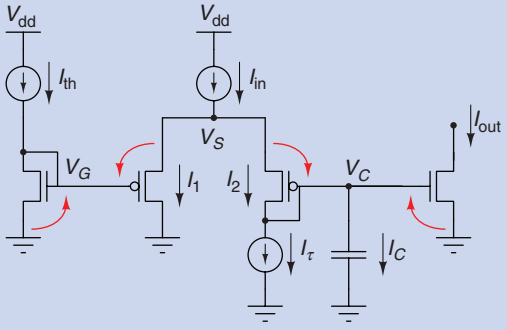
$$\tau\frac{d}{dt}I_{\text{out}}+I_{\text{out}}=\frac{I_g}{I_\tau}I_{\text{in}}. \tag{2}$$

The circuit time constant is defined at $\tau \triangleq CU_T/\kappa I_\tau$. Long time constants are achieved by using a combination of large capacitors and small currents. Even though it is possible to implement such long time constants without sacrificing large amounts of silicon real estate area, for example, using high dielectric-constant materials, memristive devices, and active circuits that minimize leakage currents [31], there is a limit to the maximum time constants that can be implemented at the level of the single neural or synaptic components, and to the temporal scales that they can deal with for processing slow-changing signals. This is a problem that biological neural processing systems also face and that can be solved by using network dynamics to model attractors and long-term memory phenomena.

Signal processing in neuromorphic hardware on behavioral timescales

Biological nervous systems are capable of controlling behavior and processing sensory inputs in an adaptive and robust manner over a multitude of timescales that extend well beyond those of the single synapse and neuron time constants [33]. Indeed, in such systems, there is a multitude of neural circuits and mechanisms that underlies the ability to process and generate temporal patterns over behavioral timescales [34]. A common circuit motive found throughout multiple parts of the cortex that is believed to subserve these functions is the "winner take all" (WTA) network [35], [36], which is a specific type of attractor network [37]. Theoretical studies have shown that such networks provide elementary units of computation that can stabilize and denoise the neuronal dynamics [35], [38], [39]. These theoretical considerations have been validated in neuromorphic hardware systems to generate robust behavior in closed sensorimotor loops [40]. However, to extend these models and hardware neural processing circuits to more complex systems, such as autonomous agents that can make decisions and generate goal-directed behavior, it is necessary to develop higher-level control strategies and theoretical frameworks compatible with mixed signal neuromorphic hardware, and endowing neuromorphic architectures with compositionality and modularity. The core challenge is to design neuronal networks for neuromorphic hardware that can create and stabilize a neuronal state that can, for example, drive movements of a robot that unfold at arbitrary timescales.

Recurrently connected neural populations can, indeed, create sustained activation to keep a neuronal state active for macroscopic, behavioral time intervals, providing a model of working memory [41]. Such sustained neuronal states do not need to lead to static behavior but should create an attractor in the behavioral space that generates the desired behavior. This desired behavior can be periodic or correspond to a fixed-point attractor [42].



(a)

Circuit Equations	Translinear Loop
$I_{\text{out}} = I_0 e^{\frac{\kappa V_C}{U_T}}$	$I_g \cdot I_1 = I_2 \cdot I_{\text{out}}$
$I_C = C\frac{dV_C}{dt}$	$I_g \cdot I_1 = (I_\tau + I_C) \cdot I_{\text{out}}$
$I_{\text{in}} = I_1 + I_\tau + I_C$	$I_g \cdot (I_{\text{in}} - I_\tau - I_C) = (I_\tau + I_C) \cdot I_{\text{out}}$
$I_C = C\frac{U_T}{\kappa I_{\text{out}}}\frac{dI_{\text{out}}}{dt}$	

(b)

FIGURE 1. (a) The DPI circuit diagram; the red arrows show the translinear loop considered for the log-domain analysis. (b) The circuit and translinear loop equations used to derive the circuit transfer function. The term I_0 in the equation represents the transistor dark current, U_T the thermal voltage, and κ the subthreshold slope factor [3].

These attractor-based representations can sustain a variable duration of actions and perceptual states and help to bridge the neuronal and behavioral timescales in a robust way [23], [41]. The recurrent neural network dynamics that can be used for this purpose are described by the following equation [22]:

$$\tau \frac{d}{dt} u(x,t) = -u(x,t) + h + I(x,t) + \int f(u(x',t)) w(|x-x'|) dx'. \tag{3}$$

Here, $u(x, t)$ is a neuronal activation function defined across a perceptual or motor space, x, that is encoded by a neuronal population. Such a dimension, x, could represent a continuous sensory feature, such as color or orientation of a visual stimulus, or a motor variable, such as a hand position or location in space. The term h in (3) is a negative resting level of the neuronal population; the term $I(x, t)$ is an external input, and $f(\cdot)$ is a sigmoid nonlinearity that smoothly filters population activity. The final integral term in the equation expresses the lateral connectivity in the neuronal population with activity described by the activation function $u(x, t)$. In particular, the integral is a convolution shaped by the interaction kernel $w(|x - x'|)$ that only depends on the distance between two positions. The interaction kernel has a "Mexican hat" form of the type

$$w(|x-x'|) = A_{\text{exc}} e^{\frac{(x-x')^2}{2\sigma_{\text{exc}}^2}} - A_{\text{inh}} e^{\frac{(x-x')^2}{2\sigma_{\text{inh}}^2}}, \tag{4}$$

where $A_{\text{exc/inh}}$ and $\sigma_{\text{exc/inh}}$ are the amplitude and spread of the excitatory and inhibitory parts of the kernel. Such a pattern of lateral connections creates a soft WTA dynamic in the neuronal population: If they are supported by neighboring (in the behavioral space) neurons, the neurons that get activated first stay activated and inhibit other neurons in the population. This formalization of the dynamics of recurrent neural populations is known as a dynamic neural field (DNF) [43], [44]. Through decades, DNF theory was developed into a framework for a neuronal basis of cognition [22] that has been recently applied to the control of cognitive robotic agents [23], [45].

The DNF dynamics described by (3) can be simulated on a computer using the Euler or Runge-Kutta method for the numerical integration of the integro-differential equation. Such simulations were developed in the past using MATLAB (https://dynamicfieldtheory.org/cosivina/) and C++ [46] (https://cedar.ini.rub.de) software libraries, and they usually require substantial computing power since the dense recurrent connections in large neuronal fields do not allow for parallelizing architectures efficiently. In some cases, when interaction kernels are separable and unchanging, the computation can be brought into Fourier space and run more efficiently. However, even in these cases, the performance of large-scale DNF architectures on a conventional computer is still unsatisfactory for real-world robotic applications. To the contrary, implementation of DNFs in neuromorphic hardware enables running these networks in real time, since computation is performed in parallel using hardware connections instead of simulating convolutions and numerical integration over time.

Figure 2(a) shows a scheme of a WTA/DNF implementation with spiking neurons. Here, red circles designate neurons, a larger number of which form an excitatory pool that represents the behavioral variable; the smaller pool of neurons forms an inhibitory group that realizes the inhibitory part of interaction kernel in (3). Red lines are excitatory, and blue lines are inhibitory connections (shown for one of the inhibitory neurons each).

The stable attractors created by such WTA dynamics are critical to enable behavior learning in a closed sensory-motor loop in which the sensory input changes continually as the agent generates action. To learn a mapping between a sensory state and its consequences, or a precondition and an action, the sensory state, before the action, needs to be stored in a neuronal representation. This can be achieved by creating a reverberating activation in the neuronal population that can be sustained for the duration of the action, even if the initial input ceases. The sustained activity can be used to update sensorimotor mappings when a rewarding or punishing signal is obtained [45], [47].

Implementing on-chip learning in neuromorphic processors

Learning is probably the most important aspect of neural processing systems: It allows us to train ANNs to perform pattern classification and recognition tasks, and in biological neural systems, it allows animals to form memories and create associations. Most importantly, however, it endows agents with the capability to adapt to changes in the statistics of the input signals or changes in the properties of their actuators across different timescales.

Usually, in machine learning, the supervised-, unsupervised-, and reinforcement-learning procedures are distinguished. All three cases of learning must be addressed, both in biological neural systems and on neuromorphic hardware that emulates those principles, using spiking neuronal dynamics and spike-based learning rules [48], [49]. One line of research, thus, tries to adapt the learning methods and training procedures developed for "rate-based," continuous-valued ANNs to spiking neural networks, which do not support nonlocal computing across synaptic weights or the computation of derivatives [50]–[52]. The hardware principles that we present here will support these developments. In our own work, additionally, we aim to develop learning architectures that enable the adaptation of synaptic weights and other network parameters when the neuronal network is used to produce behavior. Such "online" continual learning relies on stabilized representations of behavioral states (that include both perceptual states and action intentions) and their consequences or preconditions in attractor networks, presented in the "Signal Processing in Neuromorphic Hardware on Behavioral Timescales" section. The weight update mechanism itself is not critical and can, then, be implemented by simple Hebbian learning mechanisms.

In artificial neural processing systems, learning typically involves the update of synaptic weight values. In hardware, this can require significant amounts of extra resources: In time-multiplexed systems, these resources are typically in the form of state memory, memory-transfer bandwidth, and power. These

extra resources can be especially significant if the storage is done using memory banks that are not on the processor die (for example, dynamic random-access memory in large-scale systems). On the other hand, the overhead needed to implement learning in mixed-signal neuromorphic architectures that place multiple instances of synaptic circuits per emulated synapse is very small. For example, Figure 3, depicting the chip micrograph of the ROLLS device [14], shows how the synapses that comprise learning algorithms occupy approximately the same area of the nonplastic synapses that have fixed programmable weights and short-term plasticity (STP) dynamics. Since each of these synapse circuits can be stimulated individually by the chip input spikes, they can operate in parallel using slow dynamics (for example, driven by picoampere currents), without having to transfer state memory at high speeds to external memory banks. Here, the area used by the parallel circuits enables us to save bandwidth and power in implementing neural dynamics. Furthermore, the feature of implementing biologically plausible time constants and making use of explicit natural timescales enables us to use the fast digital AER [7] circuits for stimulating multiple synapses in parallel.

In our work, we used the ROLLS device in Figure 3 to enable learning in WTA networks to implement a hardware model of sequence learning [53] as well as learning maps [54] and sensorimotor mappings [55] in neuromorphic agents. Specifically, we used interconnected populations of spiking neurons in a WTA fashion, creating excitatory recurrent connections between neurons that encode similar features. When a group of such interconnected neurons is activated, the excitatory connections create an activation flow between neurons that can sustain itself after the initiating input yields. Such sustained activation forms an attractor that can drive down-stream structures, resulting in a movement of the agent; the time structure of this movement can be decoupled from the dynamics and timing of sensory input. Moreover, learning between sustained attractor states can be triggered when a rewarding or error signal is perceived.

The rather dense WTA connectivity requires parallel processing to be efficiently computed. Moreover, nonlinearities of the DNF dynamics, which are crucial for its attractor properties, need to be processed for a large number of neurons in parallel. The dedicated neuronal circuits in neuromorphic devices, such as the one in Figure 3, lead to a direct implementation of the neuronal attractor networks in the wiring on the chip, with local synapses and neurons performing signal processing (such as filtering and smoothing), symbolic computation (selecting stable attractors), and state-holding (memory storage). Such implementation is more efficient compared to hardware with time multiplexing and virtual time, in terms of energy, because it does not require moving the state data (neuronal activation and weight values) between separate memory blocks and processing ones. We, thus, argue that neuromorphic devices that feature real-time processing with timescales matching the task, and with a massively parallel network of analog computing elements, when matched with

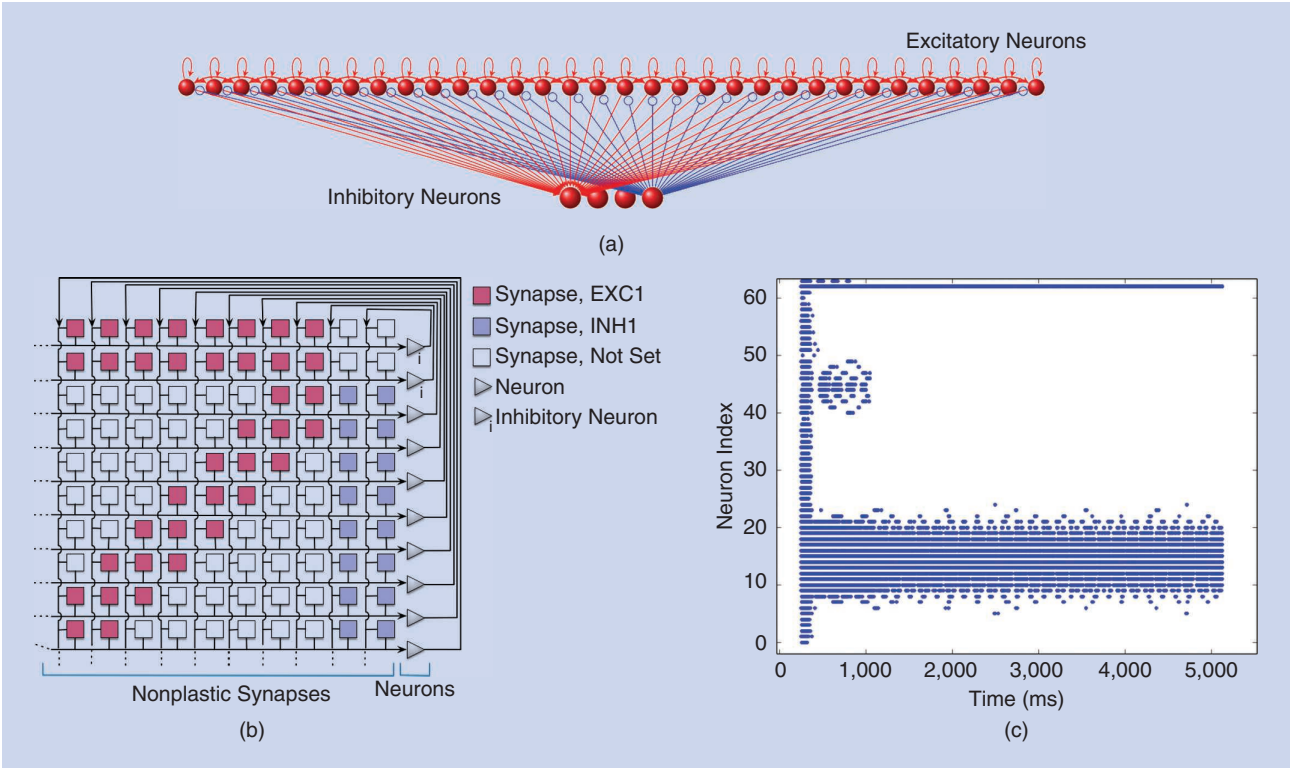


FIGURE 2. (a) The neuronal connectivity realizing a WTA/DNF (red are excitatory connections and blue are inhibitory connections; only one set of excitatory-to-inhibitory and inhibitory-to-excitatory connections is shown). (b) The same connectivity is realized in nonplastic synapses on a mixed-signal neuromorphic processor. (c) The illustration of the noise-reducing properties of the DNF dynamics: a raster plot of spiking activity in a WTA architecture implemented on a neuromorphic processor [14]. The distractor input is suppressed after 1 s.

a neuronal architecture of attractor-based population dynamics, can lead to efficient implementations of neuromorphic systems capable of generating cognitive behavior. In the following sections, we present representative examples of such neuromorphic agents.

Signal processing for neuromorphic agents

In this section, we describe neuronal architectures that make use of the properties of neuromorphic devices with explicit representation of neuronal substrate and real time, using concepts of attractor dynamics in neuronal populations as a means to program the devices to implement desired functionalities. We describe, in detail, two seminal architectures: a realization of a closed-loop navigation controller inspired by a Braitenberg vehicle, and a sequence-learning architecture that makes use of plastic on-chip synapses to store sequences of observations. We review a few more examples where on-chip plasticity was used to learn along with behavior. Both these architectures were realized on the ROLLS neuromorphic processor [14]. Despite the small size of the network that was implemented, the neuromorphic processor is capable of producing robust behavior on a robotic agent in real time.

Closed-loop sensorimotor control

The first neuromorphic architecture demonstrates how the behavior of a simple roving agent can be controlled by a spiking neuronal network. Such behavior can be generated by a very simple neuronal system as has been demonstrated by V. Braitenberg with his classical "vehicles" that generate meaningful behavior in a structured environment based on the simple wiring of their sensors and motors [56]. According to this view, the most basic ability required to generate goal-directed behavior is the capacity to differentiate between different sensory inputs. In simple terms, the agent must be able to tell one sensory state from another one. In terms of the neuronal architecture, this means that the system needs to represent combinations of visual features and their spatial locations to select the spatial target for a movement. Note that such combinations are also generated in deep convolutional neural network architectures in different feature maps. The spatial component, however, typically gets lost across layers of a network that is trained for a recognition task. The first part of our architecture achieves such differentiation or representation of sensory states (note, we are showing a small-scale example, here, on a chip with 256 spiking neurons).

Figure 4 presents a neuronal architecture that realizes one of Braitenberg's controllers on the neuromorphic device ROLLS. We use a WTA network to represent visual input obtained by a neuromorphic, event-based camera DVS mounted on a miniature robotic vehicle, Pushbot. In particular, neurons in two WTAs receive an external spike for each event in their receptive field: The first WTA population ("Target" in the figure) has weak lateral interactions and can represent several potential targets in the field of view of the robot. The second

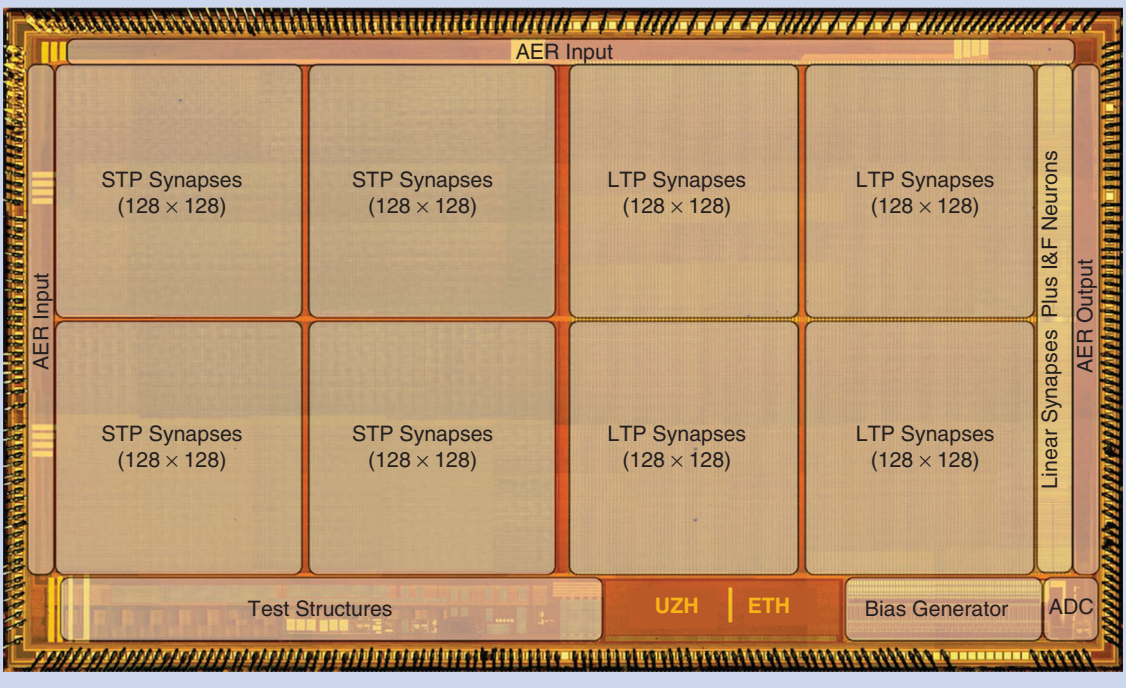


FIGURE 3. The chip micrograph of the ROLLS neuromorphic processor. Silicon-neuron integrate and I&F circuits are placed at the right of the die area and connected to an array of 256 × 512 synapses. Half of the synapse array is composed of learning synapses that exhibit long-term plasticity (LTP), and the other half is composed of fixed-weight synapses that exhibit STP. Input and output spikes are routed to the synapses from the neurons via asynchronous digital address-event representation (AER) circuits. On-chip DACs (bias generators) can be programmed to set the bias values of the analog synapse and neuron circuits. On-chip analog-to-digital converters (ADCs) are used to read out the currents from the current-mode DPI circuits that emulate synaptic dynamics. UZH: Universität Zürich; ETH: Eidgenössische Technische Hochschule Zürich.

WTA population ("Target Selective") has strong lateral excitation and inhibition and selects a single target. Each physical neuron on the ROLLS that is assigned to belong to the Target WTA network is stimulated by visual events from the upper half of the DVS: Each neuron in the Target WTA observes a vertical column of the DVS frame. Because of the soft WTA connectivity, activity bumps emerge in the target WTA that correspond to locations of salient visual inputs in the field of vision (FoV) of the robot. In this simple example, the target direction is set by a blinking LED on the second robot that can be observed in the upper half of the FoV. Objects in the lower half of the FoV are considered to be obstacles and drive two "Obstacles" neuronal populations, shown in the upper part of the architecture in Figure 4: Objects in the left or right half of the FoV activate the left or right obstacle population, respectively. Thus, in the visual part of the controller, we differentiate between obstacles and targets based on the location of the input on the vertical axis of the FoV, and we differentiate between locations (or directions) of input based on the horizontal coordinate of the dynamic vision sensor (DVS) events.

Note that the physical instantiation of neurons on the ROLLS chip makes this architecture simple and elegant: No time-multiplexing and state-machine logic are needed; in fact, no software-based arbitration or "algorithm" is used for processing. Instead, we have created a closed-loop dynamical system that processes sensory inputs and creates a representation in real, physical time. By connecting events from the camera to different neurons through synaptic weights that realize the receptive fields of these neurons, we represent the visual input in the neuronal substrate. In this

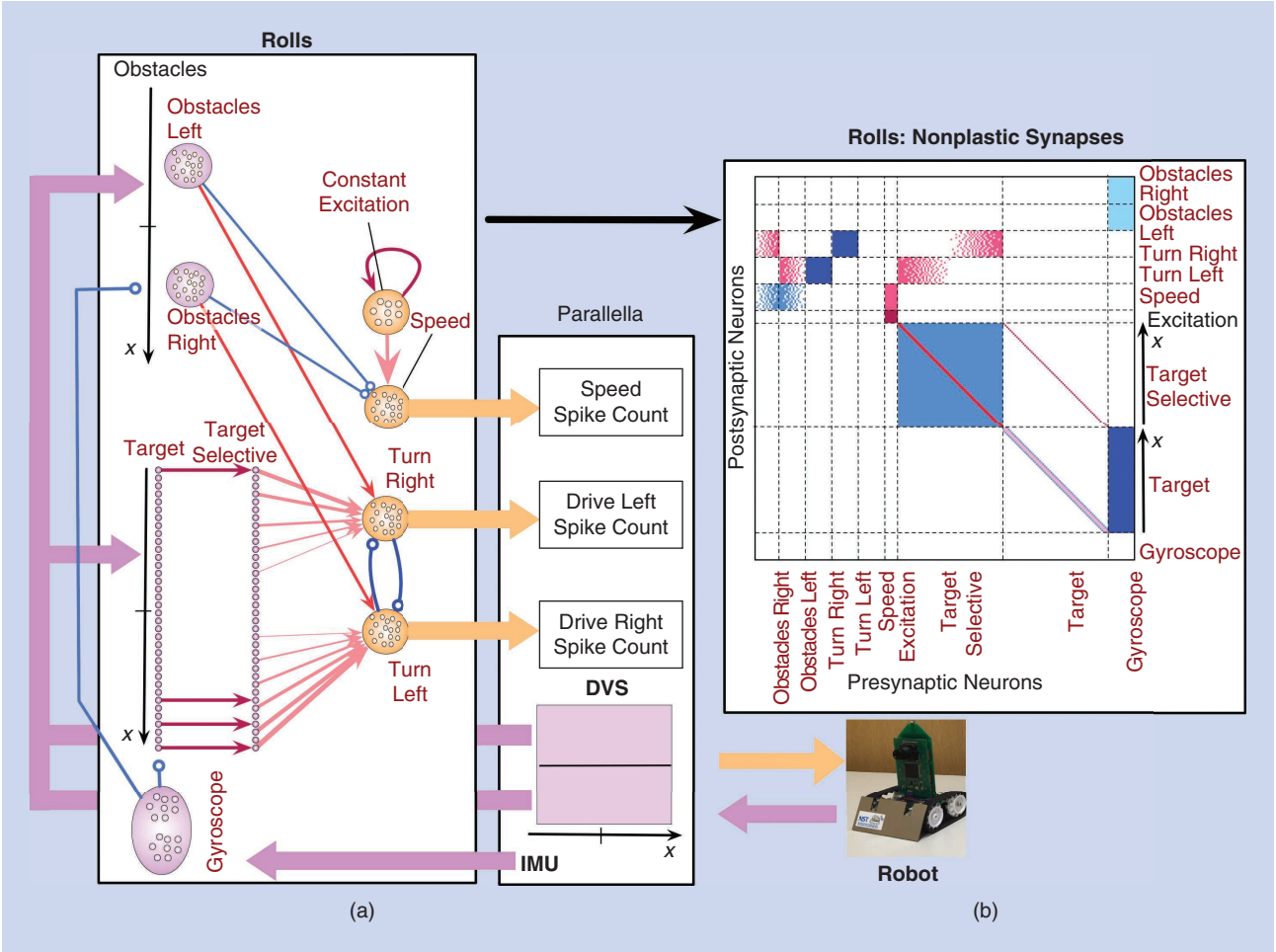


FIGURE 4. (a) The schematics of the architecture for obstacle avoidance and target acquisition on the ROLLS chip. The architecture is realized on the ROLLS neuromorphic device with 256 adaptive, mixed-signal, analog-digital, and leaky I&F neurons. The interface to the robot is established using a miniature computer board, Parallella. The output of the robot's DVS is collected on the Parallella, and the DVS events are streamed to neurons on the ROLLS that represent horizontal coordinates of the events caused by either a target (the upper half of the FoV) or an obstacle (the lower half of the FoV). The second target neuronal population has a strong WTA connectivity and selects a single target location. The motor populations (the orange neuron groups) control the robot, setting its turning speed (to the right or left from its heading direction) and translation speed, respectively. The latter speed population is inhibited if an obstacle is detected in one of the obstacle populations. The left or right turning speed is set by the excitatory connections from the obstacles and target populations. Spikes from the motor neurons are collected on the Parallella board and set the motor speeds of the robot (proportional to the instantaneous firing rate). The gyroscope population inhibits the visual neurons when the robot is turning to compensate for additional events that are caused by turning ("saccadic suppression"). For more details, see [57] and [74]. (b) The same architecture is shown as a connectivity matrix of nonplastic synapses on the ROLLS chip. Red are excitatory and blue are inhibitory synapses. IMU: inertial measurement unit.

representation, each active neuron signals that there is some input coming from the portion of the sensor that it observes. If several objects are present in front of a camera, several neurons will be active. In the selective target population, moreover, the WTA connectivity makes sure that a single target location is selected and that distractors are suppressed [57]–[59], [74].

On the motor side, the neurons that initiate the turning movement of the robot are connected to the obstacle left/right populations. These motor neurons, if active, cause the robot to turn either to the right or left. Thus, each of the two motor populations in Figure 4 activates a motor primitive that causes the robot's movement. In particular, the rotational velocity of the robot is set proportionally to the firing rate of the respective (left or right) motor population. The presence of an obstacle also makes the robot slow down; activity in the "Obstacles" populations inhibits the "Speed" neuronal population. The firing rate of this latter population sets the robot's forward speed.

Overall, the simple neuronal architecture in Figure 4 enables the robot to drive around, avoiding obstacles and approaching targets [57]–[59], [74]. This behavior is produced by a dynamical system created by the sensorimotor "embodiment" of the robot that is situated in an environment, and the neuronal system that closes the behavior loop, connecting perception to motor control. This controller does not include any algorithmic arbitration: The sensory signals (DVS events and gyroscope measurements) directly drive neurons and, eventually, the motors. The neuronal architecture ensures that the correct behavior is produced. The fact that the neurons on the ROLLS chip are instantiated physically makes such efficient, direct connection possible, reducing demands on the memory and arbitration that are usually managed by the CPU.

The neuronal architecture presented here does not include any learning. Indeed, this architecture is wired for a particular task, although the precise neuronal parameters can be found by a machine-learning procedure [60] instead of human labor tuning. In the next section, we present a neuronal architecture on the ROLLS chip that includes learning using plastic on-chip synapses. This architecture is one of our recent examples that shows how representations—of temporal or spatial behaviorally-relevant patterns, such as sequences or maps—can be learned using principles of attractor dynamics in hardware with explicit, physical representation of space and time.

Learning and processing sequences in the neuromorphic device

Recently, we implemented the basic serial-order sequence-learning model based on DNFs [61] on the ROLLS chip [53]. In this architecture, a WTA population represents the content of the items in a sequence (here, the location of a target on the horizontal axis of the DVS's FoV). Other neuronal populations represent serial order positions along the sequence: the first, second, and third ordinal group, and so on. Each group of these ordinal neurons is connected to the content WTA with plastic synapses that are updated when an ordinal neuron and a WTA neuron are active at the same time, according to a local synaptic plasticity rule. Groups of memory neurons, driven by the ordinal neurons, create sustained activation and keep track of the position in the sequence during periods of sequential transitions, when ordinal groups are inhibited. This inhibition is induced by a condition-of-satisfaction (CoS) system: a group of neurons that is activated when an item in the sequence has been successfully executed [61], [62].

In [54], we show how the sequence of states can be learned and replayed by a robotic agent, whose sensors and motors are interfaced to the neuromorphic device implementing the sequence learning architecture in Figure 5. In this example, again, the physical identity of neurons is important to be able to efficiently, directly distribute activity patterns in the neuronal architecture. Moreover, the ability of the attractor networks to bridge different lengths of time is critical here: Note that during sequence learning and production, each item in a sequence can take different—and unknown in advance—time intervals. Neuronal dynamics can sustain these long time intervals because stable activity bumps are created in the WTA neuronal population and in the ordinal groups. No clock is needed to keep track of time in these neuronal systems explicitly, although storing typical durations of actions is also possible in the neuronal population dynamics [63].

Another example of a neurodynamic architecture built using attractor networks and allocating dedicated circuits for each synapse/neuron in the network is the neuromorphic self-localization and map-formation architecture presented in [54], [64] and [65]. In this system, first, a WTA neuronal population keeps track of the correct heading of the agent, performing path integration based on the motor commands sent to the robot. Another neuronal population keeps track of the position of the robot in an environment. The plastic synapses of the ROLLS chip are used to learn a simple map of the environment as the robot navigates it and senses walls with a bumper. The map is effectively an association between the representation of position (and orientation) of the robot and a neuronal representation of collisions. Learning happens in the moment when the event induced by the collision with an object activates the "Collision" population of neurons. Coactivation of these neurons and neurons that represent the current position in real time leads to an increase of the plastic synapses between these two groups, forming a collisions map as the robot explores the environment. When the robot revisits the place of a previously experienced collision, but no collision happens there, the "Collision" population will be activated through the plastic synapses, anticipating a collision. However, the firing rate of this population will be lower than during an actual collision, when neurons are stimulated by the sensory input. This causes synaptic depression in the plastic synapses and leads to a gradual forgetting of the associations that are no longer valid.

The same principles of explicit space and time representation have been used to develop a neuromorphic proportional-integral (PI) controller that triggers the learning of a mapping from the target speed of a robot to a motor command that

realizes this speed [55]. Such mapping is an instantiation of a simple inverse model, which is learned using plastic synapses on the ROLLS chip. Here, learning is triggered when the PI controller (realized using the same principles of population dynamics and attractor WTA networks) converges and activates a zero-error neuronal population. Again, in this architecture, sensory inputs drive neuronal representations in real time, and learning is activated autonomously.

In all these neuromorphic architectures, a neuronal system is designed that connects sensors and motors to solve a particular task. Learning is reserved for certain portions of these architectures; for example, a mapping between a sensory and motor space, or between a representation of the ordinal position and perceptual content. The WTA connectivity of neuronal populations that represent perceptual states ensures that representations are localized and stabilized, limiting learning in time and in space. We have developed more architectures that include learning; for example, in [64], we show how on-chip plasticity combined with a WTA structure can also be used to learn to differentiate patterns in an unsupervised manner, while [55]

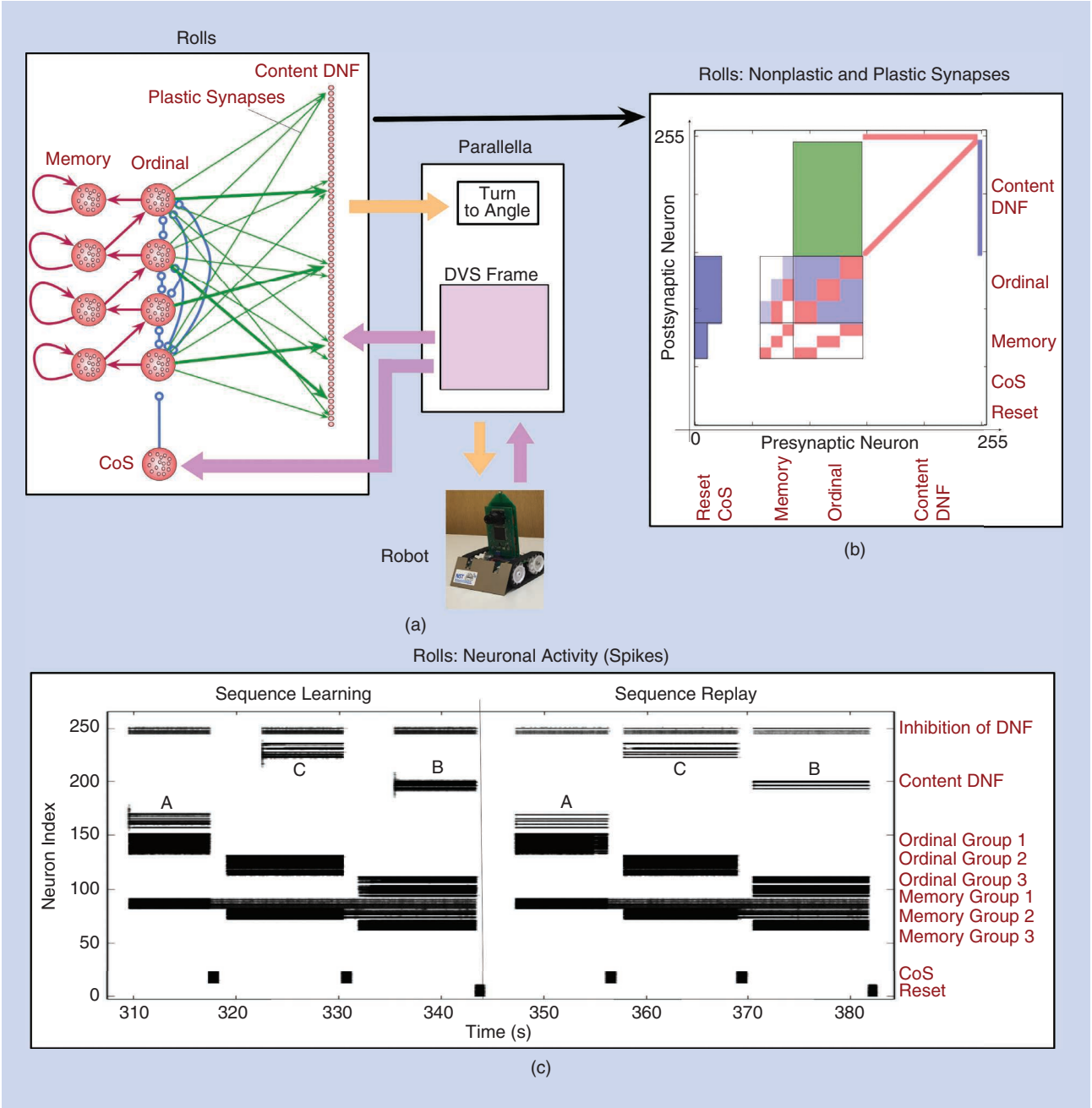


FIGURE 5. (a) The schematics of the architecture for sequence learning on the ROLLS chip. (b) Both plastic and non-plastic synapses. Plastic synapses are in the green block, and non-plastic synapses are red (excitatory) or blue (inhibitory). (c) A raster plot of neuronal activity in a sequence-learning experiment in which the robot observes three inputs, stores their order in plastic on-chip synapses, and replays the items. See [54] and the main text for details.

and [59] show how neuronal populations can be used to perform arithmetic operations, such as the summation and subtraction required, for example, to realize a PI controller in a neural network, or to perform reference frame transformations [59]. The latter work emphasizes that some computation that can easily be solved on a conventional von Neumann computing platform might require rethinking and the design of a whole neuronal architecture in neuromorphic computing devices.

For any given task that we have solved using neuromorphic computing and discussed here, a conventional software solution can be developed. Indeed, the digital computing architecture can be used to realize any computation, since it is Turing-complete. There are two main reasons why we believe that developing neuromorphic agents is important, nevertheless. First, the cost of computation becomes a nonnegligible issue as computing is scaled up in terms of the amount of sensory information that needs to be processed in real time on a compact device with limited power. Thus, even if more precise digital solutions exist for the problems of map formation, sequence learning, and motor control, and that employ digital computers and conventional software, truly neuromorphic solutions that use physics to realize neuronal computing can solve these tasks using several orders of magnitude less power. Of course, due to today's limitations in the number of artificial neurons realized in hardware, the resulting neuronal solutions provide a lower precision, which, however, is sufficient for many tasks that include "soft" actuators and imprecise sensors, in particular where closed-loop control can be used to keep the behavior at a tolerable accuracy.

The second reason lies in the homogeneity and adaptivity of the neuromorphic computing architectures. Indeed, all architectures that we briefly reviewed here use just a handful of neuronal structures—WTAs and plastic connections between them—to realize many different functions: representation, feature binding, attention, memory formation, decision making, reference frame transformation, and so forth. Indeed, in the theoretical framework of DNFs, it has been shown that most cognitive processing can be done with similar building blocks that realize attractor dynamics in neuronal populations [22]. Augmented with simple Hebbian learning, intrinsic plasticity, and homeostasis, such neuronal architectures can autonomously adapt to new situations and be easily "reprogrammed" to produce new behaviors just by using demonstration or self-generated supervision signals [66]–[69]. Combined with the ability to represent continuous variables in an adaptive, task-dependent way, neuromorphic computing steps outside the Turing computing framework and, as the hardware matures and supports larger networks, could lead to more scalable and robust solutions to tasks that involve interaction with the real world.

Conclusions

Neuromorphic computing represents a promising paradigm for artificial cognitive systems that may become an enabling element for the design of power-efficient, real-time, and compact solutions for electronic devices that process sensory data and generate behavior in dynamic, real-world environments. To minimize power consumption, it is important to match the dynamics of neuromorphic computing elements to the timescale of the physical processes that drive them and that the devices control. The impact of this approach is significant because, in addition to autonomous robots, "cognitive agents" represent low-power, always-on embedded sensory-computing systems (for example, in cell phones or wearable devices), intelligent wireless sensors (for Internet of Things applications), and intelligent microbiosensors (for personalized medicine, brain–machine interfaces, and prosthetics), and other types of microscale autonomous devices that extract structure from the data they acquire through their interactions with the environment and make decisions on how and when to act (to transmit information, power-up further processing stages, and so forth).

Although many of the organizing principles of nervous systems have successfully been adopted for the design of artificial intelligence systems, two of the most powerful ones (that is, that of letting time represent itself, and that of using the physical instantiations of parallel circuits rather than single time-multiplexed ones) are not yet fully exploited. Here, we presented examples of systems that implement these principles and described a computational approach that extends the temporal processing abilities of such devices to arbitrary timescales using attractor dynamics and the DNF framework. We discussed several examples of neurodynamic architectures that make use of the physical instantiation of spiking neurons in hardware and their real-time dynamics, matched to biological timescales. We presented architectures that were designed based on the available sensory and motor systems and the given task as spiking neural networks that can be realized in neuromorphic hardware. In some cases, when such spiking neural network implementation is not straightforward, other methods can be used to either learn the architecture from examples of the desired behavior, recorded with a teleoperated sensorimotor plant using, for example, deep neural networks, or to approximate the dynamical system of the neuronal controller with spiking neurons using the neural engineering framework [70].

Many more examples are being proposed for using dedicated circuits for neurons and synapses to take advantage of emerging memory technologies based on memristive devices [71]. These novel types of architectures differ from the more classical type of deep network or convolutional network accelerators mainly for breaking the von Neumann bottleneck [72] by having distributed memory elements that act also as computing devices. This in-memory computing approach enables dramatic power savings. For example, the (pure CMOS-based) in-memory computing DYNAP-SE processor presented in Table 1 is at least a factor of 100 times more power efficient than Spinnaker2 (for example, compare the power consumption figures in [13] versus [73]), which represents one the most recent state-of-the-art spiking neural network accelerators implemented following the classical synchronous logic, time-multiplexed, microprocessor approach.

We believe the examples presented in this article represent the first steps toward the realization of power-efficient neuromorphic systems and robust computing architectures

optimally suited for tasks in which behavior needs to be generated by autonomous neuromorphic agents in real time, while staying adaptive in complex environments.

Acknowledgments

Many of the concepts presented here were inspired by the discussions held at the CapoCaccia Cognitive Neuromorphic Engineering Workshop. The quantitative comparison in Table 1 was performed by Mohammad Ali Sharif Shazileh. This work received funding from the European Research Council under grant 724295 (Neuro Agents) and the Swiss National Science Foundation project Ambizione (grant PZ00P2_168183_1).

Authors

Giacomo Indiveri (giacomo@ini.uzh.ch) received his M.Sc. degree in electrical engineering and Ph.D. degree in computer science from the University of Genoa, Italy, and his habilitation degree on neuromorphic engineering at ETH Zürich in 2006. He is the director of the Institute of Neuroinformatics of the University of Zürich and ETH Zürich, and he holds a professor position at the University of Zürich, Switzerland. He is a recipient of two European Research Council (ERC) grants: he was awarded an ERC Starting Grant in 2011 and an ERC Consolidator Grant in 2017.

Yulia Sandamirskaya (ysandamirskaya@ini.uzh.ch) received a degree in physics from Belarusian State University, Minsk, and a Dr. rer. nat. degree from the Institute for Neural Computation at the Ruhr-Universität Bochum, Germany. She is a group leader in the Institute of Neuroinformatics of the University of Zürich and ETH Zürich. Her group, Neuromorphic Cognitive Robots, develops neurodynamic architectures for embodied cognitive agents. In particular, she studies memory formation, motor control, and autonomous learning in spiking and continuous neural networks, realized in neuromorphic hardware interfaced to robotic sensors and motors. She is the chair of the European Society for Cognitive Systems and the coordinator of the NEUROTECH project that supports and develops the neuromorphic computing community in Europe.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015. doi: 10.1038/nature14539.

[2] E. Chicca, F. Stefanini, C. Bartolozzi, and G. Indiveri, "Neuromorphic electronic circuits for building autonomous cognitive systems," *Proc. IEEE*, vol. 102, no. 9, pp. 1367–1388, Sept. 2014. doi: 10.1109/JPROC.2014.2313954.

[3] C. Mead, "Neuromorphic electronic systems," *Proc. IEEE*, vol. 78, no. 10, pp. 1629–1636, Oct. 1990. doi: 10.1109/5.58356.

[4] G. Indiveri, B. Linares-Barranco, T. Hamilton, A. van Schaik, R. Etienne-Cummings, T. Delbruck, S.-C. Liu, P. Dudek et al., "Neuromorphic silicon neuron circuits," *Front. Neurosci.*, vol. 5, pp. 1–23, May 2011. doi: 10.3389/fnins.2011.00073.

[5] S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, and W. Lu, "Nanoscale memristor device as synapse in neuromorphic systems," *Nano Lett.*, vol. 10, no. 4, pp. 1297–1301, Mar. 2010. doi: 10.1021/nl904092h.

[6] S. Saighi, C. Mayr, B. Linares-Barranco, T. Serrano-Gotarredona, H. Schmidt, G. Lecerf, J. Tomas, J. Grollier et al., "Plasticity in memristive devices," *Front. Neurosci.*, vol. 9, no. 51, Mar. 2015. doi: 10.3389/fnins.2015.00051.

[7] S.-C. Liu, T. Delbruck, G. Indiveri, A. Whatley, and R. Douglas, *Event-Based Neuromorphic Systems*. Hoboken, NJ: Wiley, 2014.

[8] E. Chicca, D. Badoni, V. Dante, M. D'Andreagiovanni, G. Salina, L. Carota, S. Fusi, and P. Del Giudice, "A VLSI recurrent network of integrate-and-fire neurons connected by plastic synapses with long term memory," *IEEE Trans. Neural Netw.*, vol. 14, no. 5, pp. 1297–1307, Sept. 2003. doi: 10.1109/TNN.2003.816367.

[9] M. Giulioni, P. Camilleri, V. Dante, D. Badoni, G. Indiveri, J. Braun, and P. Del Giudice, "A VLSI network of spiking neurons with plastic fully configurable 'stop-learning' synapses," in *Proc. 2008 15th IEEE Int. Conf. Electronics, Circuits, and Systems*, pp. 678–681. doi: 10.1109/ICECS.2008.4674944.

[10] S. Brink, S. Nease, P. Hasler, S. Ramakrishnan, R. Wunderlich, A. Basu, and B. Degnan, "A learning-enabled neuron array IC based upon transistor channel models of biological phenomena," *IEEE Trans. Biomed. Circuits Syst.*, vol. 7, no. 1, pp. 71–81, Feb. 2013. doi: 10.1109/TBCAS.2012.2197858.

[11] J. Park, S. Ha, T. Yu, E. Neftci, and G. Cauwenberghs, "A 65k-neuron 73-mevents/s 22-pj/event asynchronous micro-pipelined integrate-and-fire array transceiver," in *Proc. 2014 IEEE Biomedical Circuits and Systems Conf. (BioCAS)*, pp. 675–678. doi: 10.1109/BioCAS.2014.6981816.

[12] B. V. Benjamin, P. Gao, E. McQuinn, S. Choudhary, A. R. Chandrasekaran, J. Bussat, R. Alvarez-Icaza, J. Arthur et al., "Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations," *Proc. IEEE*, vol. 102, no. 5, pp. 699–716, Apr. 2014. doi: 10.1109/JPROC.2014.2313565.

[13] S. Moradi, N. Qiao, F. Stefanini, and G. Indiveri, "A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (DYNAPs)," *IEEE Trans. Biomed. Circuits Syst.*, vol. 12, no. 1, pp. 106–122, Feb. 2018. doi: 10.1109/TBCAS.2017.2759700.

[14] N. Qiao, H. Mostafa, F. Corradi, M. Osswald, F. Stefanini, D. Sumislawska, and G. Indiveri, "A re-configurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128k synapses," *Frontiers Neuroscience*, vol. 9, no. 141, pp. 1–17, Apr. 2015. doi: 10.3389/fnins.2015.00141.

[15] J. Schemmel, D. Bruderle, A. Grubl, M. Hock, K. Meier, and S. Millner, "A wafer-scale neuromorphic hardware system for large-scale neural modeling," in *Proc. 2010 IEEE Int. Symp. Circuits and Systems*, pp. 1947–1950. doi: 10.1109/ISCAS.2010.5536970.

[16] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam et al., "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Sci.*, vol. 345, no. 6197, pp. 668–673, Aug. 2014. doi: 10.1126/science.1254642.

[17] S. Furber, F. Galluppi, S. Temple, and L. Plana, "The SpiNNaker project," *Proc. IEEE*, vol. 102, no. 5, pp. 652–665, May 2014. doi: 10.1109/JPROC.2014.2304638.

[18] C. Frenkel, J. Legat, and D. Bol, "A 0.086-mm^2 9.8-pj/sop 64k-synapse 256-neuron online-learning digital spiking neuromorphic processor in 28nm CMOS," *IEEE Trans. Biomed. Circuits Syst.*, vol. 13, no. 1, pp 145–158, Feb. 2019. doi: 10.1109/TBCAS.2018.2880425.

[19] M. Davies, N. Srinivasa, T. H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi et al., "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, Jan.-Feb. 2018. doi: 10.1109/MM.2018.112130359.

[20] F. Conti and L. Benini, "A ultra-low-energy convolution engine for fast brain-inspired vision in multicore clusters," in *Proc. 2015 Design, Automation and Test Europe Conf. Exhibition,* pp. 683–688. doi: 10.7873/DATE.2015.0404.

[21] A. Aimar, H. Mostafa, E. Calabrese, A. Rios-Navarro, R. Tapiador-Morales, I.-A. Lungu, M. B. Milde, F. Corradi et al., "Nullhop: A flexible convolutional neural network accelerator based on sparse representations of feature maps," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 3, pp. 644–656, Mar. 2019. doi: 10.1109/TNNLS.2018.2852335.

[22] G. Schöner and J. Spencer, Eds., *Dynamic Thinking: A Primer on Dynamic Field Theory*. London: Oxford Univ. Press, 2015.

[23] Y. Sandamirskaya, S. K. Zibner, S. Schneegans, and G. Schöner, "Using dynamic field theory to extend the embodiment stance toward higher cognition," *New Ideas Psychology*, vol. 31, no. 3, pp. 322–339, Dec. 2013. doi: 10.1016/j.newideapsych.2013.01.002.

[24] M. R. Azghadi, N. Iannella, S. Al-Sarawi, G. Indiveri, and D. Abbott, "Spike-based synaptic plasticity in silicon: Design, implementation, application, and challenges," *Proc. IEEE*, vol. 102, no. 5, pp. 717–737, May 2014. doi: 10.1109/JPROC.2014.2314454.

[25] S. Mitra, S. Fusi, and G. Indiveri, "Real-time classification of complex patterns using spike-based learning in neuromorphic VLSI," *IEEE Trans. Biomed. Circuits Syst.*, vol. 3, no. 1, pp. 32–42, Feb. 2009. doi: 10.1109/TBCAS.2008.2005781.

[26] W. Ma, J. Beck, P. Latham, and A. Pouget, "Bayesian inference with probabilistic population codes," *Nat. Neurosci.*, vol. 9, no. 11, pp. 1432–1438, Oct. 2006. doi: 10.1038/nn1790.

[27] J. Schemmel, D. Brüderle, K. Meier, and B. Ostendorf, "Modeling synaptic plasticity within networks of highly accelerated I&F neurons," in *Proc. 2007 IEEE Int. Symp. Circuits and Systems*, pp. 3367–3370. doi: 10.1109/ISCAS.2007.378289.

[28] C. Tomazou, F. Lidgey, and D. Haigh, Eds., *Analogue IC design: The Current-Mode Approach*. Stevenage, U.K.: Peregrinus, 1990.

[29] T. Rost, H. Ramachandran, M. P. Nawrot, and E. Chicca, "A neuromorphic approach to auditory pattern recognition in cricket phonotaxis," in *Proc. 2013 European Conf. Circuit Theory and Design (ECCTD)*, pp. 1–4. doi: 10.1109/ECCTD.2013.6662247.

[30] A. Banerjee, S. Kar, S. Roy, A. Bhaduri, and A. Basu, "A current-mode spiking neural classifier with lumped dendritic nonlinearity," in *Proc. 2015 IEEE Int. Symp. Circuits and Systems, (ISCAS)*, pp. 714–717. doi: 10.1109/ISCAS.2015.7168733.

[31] N. Qiao, C. Bartolozzi, and G. Indiveri, "An ultralow leakage synaptic scaling homeostatic plasticity circuit with configurable time scales up to 100 ks," *IEEE Trans. Biomed. Circuits Syst.*, vol. 11, no. 6, pp. 1271–1277, Dec. 2017. doi: 10.1109/TBCAS.2017.2754383.

[32] C. Bartolozzi and G. Indiveri, "Synaptic dynamics in analog VLSI," *Neural Comput.*, vol. 19, no. 10, pp. 2581–2603, Oct. 2007. doi: 10.1162/neco.2007.19.10.2581.

[33] C. V. Buhusi and W. H. Meck, "What makes us tick? Functional and neural mechanisms of interval timing," *Nat. Rev. Neurosci.*, vol. 6, no. 10, pp. 755–765, Sept. 2005. doi: 10.1038/nrn1764.

[34] J. J. Paton and D. V. Buonomano, "The neural basis of timing: Distributed mechanisms for diverse functions," *Neuron*, vol. 98, no. 4, pp. 687–705, May 2018. doi: 10.1016/j.neuron.2018.03.045.

[35] R. J. Douglas and K. A. Martin, "Neuronal circuits of the neocortex," *Annu. Rev. Neurosci.*, vol. 27, pp. 419–451, July 2004. doi: 10.1146/annurev.neuro.27.070203.144152.

[36] R. Douglas and K. Martin, "Recurrent neuronal circuits in the neocortex," *Curr. Biol.*, vol. 17, no. 13, pp. R496–R500, July 2007. doi: 10.1016/j.cub.2007.04.024.

[37] N. Brunel, *Network Models of Memory*. Amsterdam, The Netherlands: Elsevier, 2004.

[38] W. Maass, "On the computational power of winner-take-all," *Neural Computation*, vol. 12, no. 11, pp. 2519–2535, Nov. 2000.

[39] U. Rutishauser, R. Douglas, and J. Slotine, "Collective stability of networks of winner-take-all circuits," *Neural Comput.*, vol. 23, no. 3, pp. 735–773, Feb. 2011. doi: 10.1162/NECO_a_00091.

[40] E. Neftci, J. Binas, U. Rutishauser, E. Chicca, G. Indiveri, and R. Douglas, "Synthesizing cognition in neuromorphic electronic systems," *Proc. Nat. Academy Sci. United States America*, vol. 110, no. 37, pp. E3468–E3476, Sept. 2013. doi: 10.1073/pnas.1212083110.

[41] J. S. Johnson, J. P. Spencer, and G. Schöner, "Moving to higher ground: The dynamic field theory and the dynamics of visual cognition," *New Ideas Psychology*, vol. 26, no. 2, pp. 227–251, Aug. 2008. doi: 10.1016/j.newideapsych.2007.07.007.

[42] V. Mante, D. Sussillo, K. V. Shenoy, and W. T. Newsome, "Context-dependent computation by recurrent dynamics in prefrontal cortex," *Nature*, vol. 503, no. 7474, pp. 78–84, Nov. 2013. doi: 10.1038/nature12742.

[43] S. Amari, "Dynamics of pattern formation in lateral-inhibition type neural fields," *Biological Cybern.*, vol. 27, no. 2, pp. 77–87, June 1977. doi: 10.1007/BF00337259.

[44] H. Wilson and J. Cowan, "A mathematical theory of the functional dynamics of cortical and thalamic nervous tissue," *Biological Cybern.*, vol. 13, no. 2, pp. 55–80, Sept. 1973. doi: 10.1007/BF00288786.

[45] Y. Sandamirskaya, "Dynamic neural fields as a step toward cognitive neuromorphic architectures," *Front. Neurosci.*, vol. 7, pp. 276–289, Jan. 2014. doi: 10.3389/fnins.2013.00276.

[46] O. Lomp, M. Richter, S. K. U. Zibner, and G. Schöner, "Developing dynamic field theory architectures for embodied cognitive systems with cedar," *Front. Neurorobot.*, vol. 10, pp. 1–18, Nov. 2016. doi: 10.3389/fnbot.2016.00014.

[47] Y. Sandamirskaya J. Conradt, "Learning sensorimotor transformations with dynamic neural fields," in *Proc. ICANN 2013: Artificial Neural Networks and Machine Learning*, vol. 8131, pp. 248–255. New York: Springer.

[48] R. Gütig and H. Sompolinsky, "The tempotron: A neuron that learns spike timing-based decisions," *Nat. Neurosci.*, vol. 9, no. 3, pp. 420–428, Feb. 2006. doi: 10.1038/nn1643.

[49] W. Gerstner and W. Kistler, *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge, U.K.: Cambridge Univ. Press, 2002.

[50] E. O. Neftci, C. Augustine, S. Paul, and G. Detorakis, "Event-driven random back-propagation: Enabling neuromorphic deep learning machines," *Front. Neurosci.*, vol. 11, pp. 1–18, June 2017. doi: 10.3389/fnins.2017.00324.

[51] R. Urbanczik and W. Senn, "Learning by the dendritic prediction of somatic spiking," *Neuron*, vol. 81, no. 3, pp. 521–528, Feb. 2014. doi: 10.1016/j.neuron.2013.11.030.

[52] J. C. Thiele, O. Bichler, and A. Dupret, "Event-based, timescale invariant unsupervised online deep learning with STDP," *Front. Comput. Neurosci.*, vol. 12, pp. 1–13, June 2018. doi: 10.3389/fncom.2018.00046.

[53] R. Kreiser, D. Aathmani, N. Qiao, G. Indiveri, and Y. Sandamirskaya, "Organizing sequential memory in a neuromorphic device using dynamic neural fields," *Front. Neuromorphic Eng.*, vol. 12, pp. 1–17, Nov. 2018. doi: 10.3389/fnins.2018.00717.

[54] R. Kreiser, P. Pienroj, A. Renner, and Y. Sandamirskaya, "Pose estimation and map formation with spiking neural networks: Towards neuromorphic SLAM," in *Proc. 2018 IEEE/Robotics Society Japan Int. Conf. Intelligent Robots and Systems (IROS)*, pp. 2159–2166. doi: 10.1109/IROS.2018.8594228.

[55] S. Glatz, R. Kreiser, J. N. P. Martel, N. Qiao, and Y. Sandamirskaya, "Adaptive motor control and learning in a spiking neural network, fully realized on a mixed-signal analog/digital neuromorphic processor," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA) 2019*. [Online]. Available: https://arxiv.org/abs/1810.10801

[56] V. Braitenberg, *Vehicles: Experiments in Synthetic Psychology*. Cambridge, MA: MIT Press, 1984.

[57] M. B. Milde, A. Dietmüller, H. Blum, G. Indiveri, and Y. Sandamirskaya, "Obstacle avoidance and target acquisition in mobile robots equipped with neuromorphic sensory-processing systems," in *Proc. 2017 IEEE Int. Symp. Circuits and Systems, (ISCAS)*, pp.1–4. doi: 10.1109/ISCAS.2017.8050984.

[58] M. B. Milde, D. Neil, A. Aimar, T. Delbruck, and G. Indiveri, "ADaPTION: Toolbox and benchmark for training convolutional neural networks with reduced numerical precision weights and activation," unpublished.

[59] H. Blum, A. Dietmüller, M. Milde, J. Conradt, G. Indiveri, and Y. Sandamirskaya, "A neuromorphic controller for a robotic vehicle equipped with a dynamic vision sensor," in *Proc. Robotics: Science and Systems (RSS 2017)*, pp.1–9. doi: 10.15607/RSS.2017.XIII.035.

[60] E. Neftci, J. Binas, E. Chicca, G. Indiveri, and R. Douglas, "Systematic construction of finite state automata using VLSI spiking neurons," in *Biomimetic and Biohybrid Systems*, T. Prescott, N. Lepora, A. Mura, and P. Verschure, Eds. Berlin: Springer-Verlag, 2012, pp. 382–383.

[61] Y. Sandamirskaya and G. Schöner, "An embodied account of serial order: How instabilities drive sequence generation," *Neural Netw.*, vol. 23, no. 10, pp. 1164–1179, Dec. 2010. doi: 10.1016/j.neunet.2010.07.012.

[62] M. Luciw, S. Kazerounian, K. Lakhmann, M. Richter, and Y. Sandamirskaya, "Learning the condition of satisfaction of an elementary behavior in dynamic field theory," *Paladyn, J. Behavioral Robotics*, vol. 6, no. 1, pp. 180–190, Jan. 2015. doi: 10.1515/pjbr-2015-0011.

[63] B. Duran and Y. Sandamirskaya, "Learning temporal intervals in neural dynamics," *IEEE Trans. Cogn. Develop. Syst.*, vol. 10, no. 2, pp. 359–372, June 2018. doi: 10.1109/TCDS.2017.2676839.

[64] R. Kreiser, T. Moraitis, Y. Sandamirskaya, and G. Indiveri, "On-chip unsupervised learning in winner-take-all networks of spiking neurons," in *Proc. 2017 IEEE Biomedical Circuits and Systems Conf., (BioCAS)*, pp. 424–427. doi: 10.1109/BIOCAS.2017.8325168.

[65] R. Kreiser, M. Cartiglia, J. N. Martel, J. Conradt, and Y. Sandamirskaya, "A neuromorphic approach to path integration: A head direction spiking neural network with vision-driven reset," in *Proc. 2018 IEEE Int. Symp. Circuits and Systems, (ISCAS)*, pp. 1–5. doi: 10.1109/ISCAS.2018.8351509.

[66] C. Strub, G. Schöner, F. Wörgötter, and Y. Sandamirskaya, "Dynamic neural fields with intrinsic plasticity," *Front. Comput. Neurosci.*, vol. 11, pp. 74, pp. 74–87, Aug. 2017. doi: 10.3389/fncom.2017.00074.

[67] D. Lobato, Y. Sandamirskaya, M. Richter, and G. Schöner, "Parsing of action sequences: A neural dynamics approach," *Paladyn, J. Behavioral Robotics*, vol. 6, pp. 119–135, May 2015. doi: 10.1515/pjbr-2015-0008.

[68] Y. Sandamirskaya and T. Storck, "Learning to look and looking to remember: A neural-dynamic embodied model for generation of saccadic gaze shifts and memory formation," in *Artificial Neural Networks*, P. Koprinkova-Hristova, V. Mladenov, and N. K. Kasabov, Eds. Berlin: Springer, 2015, pp. 175–200.

[69] S. Kazerounian, M. Luciw, M. Richter, and Y. Sandamirskaya, "Autonomous reinforcement of behavioral sequences in neural dynamics," in *Proc. 2013 Int. Joint Conf. Neural Networks (IJCNN)*, pp. 1–8. doi: 10.1109/IJCNN.2013.6706877.

[70] C. Eliasmith and C. Anderson, *Neural Engineering: Computation, Representation, and Dynamics in Neurobiological Systems*. Cambridge MA: MIT Press, 2004.

[71] M. Payvand, M. Nair, L. Müller, and G. Indiveri, "A neuromorphic systems approach to in-memory computing with non-ideal memristive devices: From mitigation to exploitation," *Faraday Discussions*, vol. 213, pp. 487–510, July 2018. doi: 10.1039/C8FD00114F.

[72] G. Indiveri and S.-C. Liu, "Memory and information processing in neuromorphic systems," *Proc. IEEE*, vol. 103, no. 8, pp. 1379–1397, Aug. 2015. doi: 10.1109/JPROC.2015.2444094.

[73] C. Liu, G. Bellec, B. Vogginger, D. Kappel, J. Partzsch, F. Neumärker, S. Höppner, W. Maass et al., "Memory-efficient deep learning on a spinnaker 2 prototype," *Front. Neurosci.*, vol. 12, pp. 840–855, Nov. 2018. doi: 10.3389/fnins.2018.00840.

[74] M. B. Milde, H. Blum, A. Dietmüller, D. Sumislawska, J. Conradt, G. Indiveri, and Y. Sandamirskaya, "Obstacle avoidance and target acquisition for robot navigation using a mixed signal analog/digital neuromorphic processing system." *Front. Neurorobotics*, vol. 11, no. 28, 2017. doi: 10.3389/fnbot.2017.00028.

SP