

Received 14 September 2022, accepted 25 October 2022, date of publication 4 November 2022, date of current version 28 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3219440

 SURVEY

Neuromorphic Computing for Interactive Robotics: A Systematic Review

MUHAMMAD AITSAM^{ID}, SERGIO DAVIES^{ID}, (Member, IEEE),
AND ALESSANDRO DI NUOVO^{ID}, (Senior Member, IEEE)

Department of Computing, Sheffield Hallam University, S1 1WB Sheffield, U.K.

Corresponding author: Alessandro Di Nuovo (a.dinuovo@shu.ac.uk)

This work was supported in part by the Marie Skłodowska-Curie Action Horizon 2020 for Project “Personalized Robotics as Service Oriented Applications (PERSEO)” under Agreement 955778, and in part by the UK Engineering and Physical Sciences Research Council (EPSRC) with (EMERGENCE) under Grant EP/W000741/1.

ABSTRACT Modelling functionalities of the brain in human-robot interaction contexts requires a real-time understanding of how each part of a robot (motors, sensors, emotions, etc.) works and how they interact all together to accomplish complex behavioural tasks while interacting with the environment. Human brains are very efficient as they process the information using event-based impulses also known as spikes, which make living creatures very efficient and able to outperform current mainstream robotic systems in almost every task that requires real-time interaction. In recent years, combined efforts by neuroscientists, biologists, computer scientists and engineers make it possible to design biologically realistic hardware and models that can endow the robots with the required human-like processing capability based on neuromorphic computing and Spiking Neural Network (SNN). However, while some attempts have been made, a comprehensive combination of neuromorphic computing and robotics is still missing. In this article, we present a systematic review of neuromorphic computing applications for socially interactive robotics. We first introduce the basic principles, models and architectures of neuromorphic computation. The remaining articles are classified according to the applications they focus on. Finally, we identify the potential research topics for fully integrated socially interactive neuromorphic robots.

INDEX TERMS Cognitive robotics, neuromorphic engineering, socially interactive robotics, event-driven computing, biologically-plausible computational intelligence.

I. INTRODUCTION

The biological intelligence of living beings has been an area of focus to explore their capabilities of memorising, thinking, perceiving, and acting accordingly. Among all species, humans have a remarkable capacity to make sound and quick decisions in diverse situations, sometimes based on vague and incomplete information. Humans perform complex behaviours that are important for surviving in dynamic environments. Advances in computational and behavioural neuroscience and embodied cognitive systems provide a baseline to integrate the interdisciplinary approaches for further technological progress in robotics. With increasing efforts of mimicking those functional and structural principles, roboticists have researched how the brain, robot sensors, and actuators

operate together to perform complex tasks in a real-world environment [1]. To acquire more autonomy and operate in the real world, robots should: 1) perceive their environments in real-time, 2) process sparse information with energy efficiency and response latency, 3) behave under changing conditions and acquire self-learning ability.

With the emergence of increasingly powerful computers and sophisticated sensing systems, machine learning algorithms became increasingly capable and have achieved success in several scientific and commercial applications. Recently, advances have been made in deep-learning approaches based on the hierarchical nature of the human vision system [2]. However, the current dominant machine learning (ML) models in robots are far from performing human-like tasks that require precise motor control, fast reaction time and adaptation to external conditions. Besides this, these ML models also lack scalability. Furthermore, the

The associate editor coordinating the review of this manuscript and approving it for publication was Daniel Augusto Ribeiro Chaves^{ID}.

divergence between the human brain and current technology can be exemplified by the fact that a hypothetical clock-based computer running a “human-scale” brain simulation requires approximately 12 Gigawatt of power. By contrast, the actual brain works with just 20 Watt [3]. A major bottleneck that severely limits the up-scaling of intelligent interactive agents is the unnatural discretization of time imposed by mainstream processing and sensing architectures [4], which are based on arbitrary internal clocks. Clock frequencies must be increased to deal with the continuous inputs of the real world. However, very high frequencies prove unfeasible and make large-scale applications of the current hardware inefficient.

To achieve such efficiency, living creatures process the information using spikes, which help them to perceive and act in the real world exceptionally well. A challenge for human-like machine intelligence is to imitate the efficient neuro-synaptic framework of the physical brain. This area of focus has been investigated extensively in recent years and many new technologies and methods are developed which try to mimic the biological behaviour of the human brain which consumes very less energy and acts very fast. One such method is Neuromorphic computing. Neuromorphic computing (also known as brain-inspired computing) is a multidisciplinary research paradigm that investigates large-scale processing systems that support natural neuronal computations through spike-driven communication. Compared to traditional approaches, key advantages of neuromorphic computing are energy efficiency, execution speed and robustness against local failures [5]. Currently, analog-programmable non-volatile memory (NVM) devices such as phase change memory (PCM) [6], resistive RAM (RRAM) [7], conductive bridging RAM (CBRAM) [8], magnetic RAM (STT-MRAM) [9] are the heart of these neuromorphic computing devices. The gradual switching of the resistance level in these devices are the key to neuromorphic computing and robotics applications [10]. Moreover, the neuromorphic design overcomes the distortion of the artificial discretization of time by using asynchronous event-driven computing that matches the temporal evolution of the external world [11]. Inspired by this event-driven type of information processing, emerging hardware and software knowledge in the field of neuroscience and electronics have made it possible to design biologically-inspired machines by using Spiking Neural Networks (SNNs) to model cognitive and interactive capabilities [12].

The interaction between humans and machines is of great relevance for both the field of neuromorphic computing and Robotics. Utilising neuromorphic technologies in robotics, from perception to motor control, is a promising approach to creating robots that can seamlessly integrate into society. In neurorobotics (neuromorphic computing and robotics), bio-inspired sensors are used to efficiently encode sensory signals. It also adapts to different environmental conditions by integrating inputs from multiple sensors and using event-based computation to accomplish desired tasks [13]. Figure 1 is summarising the landscape of neuromorphic

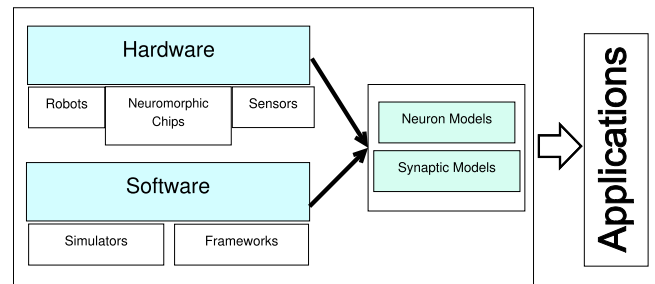


FIGURE 1. Summary of neuromorphic computing and robotics landscape. Hardware and software simulators use specific neuron and synaptic models according to the desired applications.

computing and interactive robotics. Hardware and software simulators use specific neuron and synaptic models according to the desired applications.

However, so far, a comprehensive review of the two fields has not been performed. Researchers have focused on narrow problems and proposed solutions that incrementally build on the mainstream machine learning paradigm rather than radically change it. To foster future research on interactive neuro-robotics, this article focuses on how neuromorphic computing and SNNs can be used for interactive robotics and what impact does it make when SNN models are used along with neuromorphic hardware. We investigated the contributions made in this field both theoretically and practically in terms of hardware and software platforms for the development of neurorobotics solutions. In the next section, we introduce details of spiking neural networks, neuron and synapse models. Section III gives the methodology, criteria and discusses the results of systematic research of the scientific literature on neuromorphic computing and SNNs for interactive robotics. In Section IV we present a comprehensive review of the articles, which are grouped and discussed according to the applications they focus on. Section V presents and briefly discusses neuromorphic hardware, simulators and frameworks that were used in the articles found in the systematic review. Section VI is dedicated to an analysis of shortcomings and possible future directions for research in this field. Section VII gives our conclusions.

II. SPIKING NEURAL NETWORK (SNN)

Neural networks are usually classified into three generations and all of them somehow mimic the multilayer architecture of the human brain, but the behaviour of neurons differs significantly among them [14]. In the first generation, the output of a neuron is binary (0,1) and it is obtained by simple thresholding of the weighted synaptic input. In 1943 McCulloch and Pitts showed that networks of artificial neurons have the ability to do some mathematical and logical computation [15].

With time, another concept came about when researchers developed the backpropagation technique for multilayer perceptron networks. This technique resolves the limitations of prior neural perceptron technique and it is extensively used in deep learning today. This second generation of neural

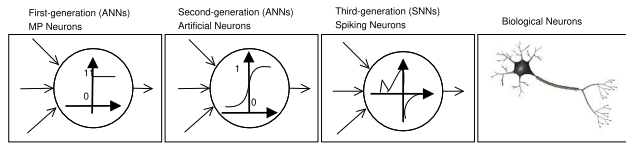


FIGURE 2. Three generations of neural network. Starting from the McCulloch Pitts (MP) neuron model in first generation then artificial neural networks in second generation and spiking neural network in third generation which mimic the behaviour of biological neuron.

networks is also known as Artificial Neural Networks (ANNs). The main difference from the first generation of neural networks is in the neuron output: in ANNs this can be a real number, which is the result of the weighted sum of all the inputs after being passed through a transfer function, usually sigmoidal. Weights are obtained as a result of some machine-learning algorithm ranging from simple linear regression to high-level classification. Modern computational hardware is widely available to evaluate the novel concepts of advanced neural networks, such as learning protocols and inventive architecture.

First and second generations of neural networks have only limited modelling capabilities of their biological counterparts and, in particular, there is no time reference to electrical signals that have been described in their biological neural networks. Moreover, there is still limited research available about biological processes. Processing real-time data, the human brain offers an efficient signal processing in which information is encoded in a number of features related to spikes, including the specific times of events [16]. This idea of simulating neural events led researchers to the development of Spiking Neural Networks (SNNs), which are, so far, the most biologically plausible models.

Figure 2 shows the overview of all three generations of neural networks (NNs). The first generation of neural networks are based on McCulloch Pitts (MP) neuron model, which shows that artificial neurons can perform basic computations. In the 2nd generation of NNs continuous non-linearity function (e.g. Relu, Sigmoid, Tanh etc.) gives promising results in applications like multi-class classification, detection, identification etc. Finally, spiking neural network, also called the third generation of neural networks [17], imitates the action of normal neurons in the brain. The neurons in brains are excitable and produce action potentials, which are also known as spikes. These spikes are basic currency of the brain. They allow neurons to perform computation and to communicate with other neurons. During a spike, a neuron releases a neurotransmitter, a chemical that travels across a synapse before reaching another neuron. Spiking neural networks also operate on spikes; these are discrete events taking place at specific times. SNNs take a spike train as input and produce a spike train at the output. The state variable(s) of a neuron in SNN can change based on the mathematical model of a neuron. If the value of the neuron state variable representing its membrane potential exceeds a predefined threshold, the neuron will send a single impulse (spike) to each post-synaptic neuron [14].

The major advantages of SNNs are the temporal plasticity, reduced computational complexity, and ease of use of neural interfaces [18]. In recent years, the popularity of SNN and its models has increased and several models have been developed for image classifications and object recognition using SNNs. The spiking neural networks are also suitable for a diverse range of applications related to computer vision and robotics, such as classification, clustering, pattern recognition, etc. There are many examples of converting data directly from sensors [19], [20], [21], intelligent systems with controlling manipulators [11], [22], and robots [23], [24] [25]. Moreover, performing detection and recognition tasks [26], [27], [28], and processing numerical data with Neural Engineering Framework (NEF) [29], [30], [31] can also be done with SNNs.

A. LEARNING IN SPIKING NEURAL NETWORKS

The key concepts of spiking neural network operations do not allow the use of classical learning techniques and methods that are appropriate for a conventional neural network. There are still several methods to train an SNN. For unsupervised learning in SNNs, the most famous method is Spike-timing-dependent plasticity (STDP) [32]. In STDP, the synaptic weight is based on the difference in firing time of pre- and post-synaptic neurons. When a pre-synaptic spike arrives before the post-synaptic action potential leads to Long Term Potentiation (LTP), whereas a pre-synaptic spike arrives after post-synaptic spikes leads to Long-term Depression (LTD) of the synapses. The change of the synapse plotted as a function of the relative time of pre- and post-synaptic spikes is known as STDP function or learning window. For supervised learning in SNNs, there are methods like back-propagation [33] where the learning algorithms like SpikeProp [34] and FreqProp [35], demonstrate how the network of spiking neurons with a biologically plausible time constant can perform complex non-linear classification tasks in temporal coding. Another such method is ReSuMe [36], which is suitable not only for movement control but also for other applications like identification and modelling of non-stationary objects. There are also a few models available for reinforcement learning in SNN. One of them is the actor-critic model [37] which uses temporal-difference learning by combining local plasticity rules with global reward signals. This network is capable to solve non trivial grid-world tasks with sparse rewards. There is also a method for reinforcement learning in SNNs through modulation of STDP [38]. This modulation is used as a global reward signal that leads to reinforcement learning.

B. SPIKING NEURAL NETWORK ARCHITECTURES

Like other neural networks, SNNs architectures are majorly divided into four groups [39].

1) FEED-FORWARD NEURAL NETWORKS

It is a classical neural network architecture in which data is transmitted only in one direction, without any cyclic connection, and processing can take place over many hidden

layers. In feed-forward neural networks, which are commonly known as a multilayered network of neurons or Deep Neural Networks (DNNs), information first enters the input node, moves through hidden layers and finally, the results come out through the output nodes. The network does not have any connection to feed the information coming out at the output, back into the network. The majority of modern artificial neural network architectures, such as convolutional neural networks (CNNs) and deep neural networks (DNNs) are feed-forward [40]. In robotics, it is usually adopted for low-level sensory acquisition, such as vision [41], olfactory [42] and tactile sensing [43].

2) RECURRENT NEURAL NETWORKS (RNN)

Unlike feed-forward NNs, recurrent neural networks leverage error correction. In simple words, in RNNs the output from the previous step is fed as an input to the subsequent step. A RNN is a recursive network with a certain structure, such as a linear chain. This mechanism is used by living organisms to process arbitrary input sequences using their internal memory stored within RNNs. Besides this, a Liquid State Machine (LSM) is also a type of recurrent network of spiking neurons where internal connectivity parameters remain static during the training process [44]. In robotics, RNNs are widely used for speech recognition [45], control [1] and planning [46].

3) HYBRID NEURAL NETWORK STRUCTURES

This type of hybrid SNN structure shows some neurons having a feed-forward connection whereas others have recurrent connections. This type of hybrid structures are often used for end-to-end training of SNNs for tasks like object detection and pattern recognition [26]. Experiments with a hybrid approach demonstrated promising results with less computational cost [26].

4) HYBRID NEURAL NETWORK ARCHITECTURES

In these architectures SNNs interact with ANNs. This type of Hybrid ANN-SNN can be trained without conversion; it results in highly accurate networks which are also more computationally efficient than their ANN counterparts [47].

C. SPIKING NEURAL NETWORK MODELS

SNNs are built on the mathematical description of biological neurons. Usually, neuron models are expressed in the form of differential equations. So far, many mathematical descriptions of SNN models have been proposed, processing inhibitory and excitatory inputs using internal state variables. Here, we will discuss some of the most influential spiking neuron models, majorly because of their simplicity and widespread use in robotics applications.

1) HODGKIN-HUXLEY NEURON MODEL

The first bio-inspired neural model was developed by Sir Alan Hodgkin and Sir Andrew Huxley in 1952 [48]. Their mathematical model describes how action potentials in neurons are initiated and how they propagate. This model is a

set of nonlinear differential equations that approximates the electrical characteristics of neurons. They also explained the ionic mechanisms underlying the initiation and propagation of action potentials in giant axons of a squid. The following differential equation of Hodgkin-Huxley model is relating the change in membrane potential to the current flowing across the membrane:

$$C_m \frac{dV_m}{dt} + I_{ion} = I_{ext} \quad (1)$$

The I_{ext} is an externally applied current. C_m is membrane Capacitance whereas V_m is membrane Voltage. Here the ionic current I_{ion} is the combination of three components, a sodium current, a potassium current and small leakage current [49].

2) IZHIKEVICH NEURON MODEL

The Izhikevich neuron model [50] combines the biological plausibility of the Hodgkin-Huxley model and computational efficiency of the integrate-and-fire neuron (LIF) model. This model reproduces spiking and bursting behaviour of known types of cortical neurons [20]. There are only 2 state variables and 2 parameters to tune in order to reproduce the complex behaviour of cortical neurons [50].

This model can be represented as 2-D system of differential equations [50]:

$$\frac{dV_m}{dt} = 0.04 V_m^2 + 5V_m + 140 - u + I_m \quad (2)$$

$$\frac{dU}{dt} = a(bV_m - U) \quad (3)$$

where V_m is the membrane potential, U is the membrane recovery variable I_m is injected bias current or incoming synaptic spikes. a and b are dimensionless variables that tweak the neuron's behaviour based on their values.

3) LEAKY INTEGRATED-AND-FIRE (LIF) NEURON MODEL

This is the most commonly used model because of its simplicity. In the leaky integrated-and-Fire model, each cell has a membrane potential V_m , with its capacitance C_m and a leaky channel that allows current to flow across the membrane with resistance R_m . The charge carriers travelling across the membrane are driven by force voltage V_e . When the voltage in the cell exceeds the threshold value V_{th} , an action potential (impulse, action potential or spike) is generated. After the spike occurs, the voltage is artificially dropped to the reset value V_{reset} . After emitting a spike a neuron may have a "rest" period in which it may not be excited. This time is usually known as "refractory period". The differential equation describes how the membrane potential V_m changes over time ($\frac{dV_m}{dt}$) in the face of an externally applied membrane current I_m is as follows:

$$C_m \frac{dV_m}{dt} = -G_m(V_m - V_e) + I_m \quad (4)$$

A simplified form of the last equation is:

$$\frac{dV_m}{dt} = \frac{-(V_m - V_e) + I_m R_m}{\tau_m} \quad (5)$$

Here V_e , R_m , τ_m are taken to be intrinsic properties of the cell while I_m is the external current and V_m is the membrane potential [51].

D. SYNAPTIC PLASTICITY MODELS

After the selection of an appropriate neural model, the synapse model should be decided to connect neurons inside and among the layer of the spiking neural network. Initially proposed by Hebb in 1949 [52], synaptic plasticity is a mechanism for learning and memory based on theoretical analysis. Depending on the relationship between neural networks and synaptic plasticity, they are roughly classified into two types: spike-based and rate-based [1].

1) RATE-BASED

The most common model of synaptic plasticity that has been proposed over many years is rate-based models. These models use a definition of firing rate that refers to an average of spike-count over time [53]. Here the magnitude of synaptic plasticity is assumed by the rate of pre- and post-synaptic firing over the specific time period. This type of model is typically used for converting conventional artificial neural networks to spiking neural networks by using backpropagation [54], [55]. The ANN is trained with backpropagation technique and then it is converted into an equivalent SNN by relating the activation of ANN units and the firing rate of spiking neurons. The relation between transfer function of spiking neurons and the activation unit of ANN have been thoroughly discussed in [56], [57], and [58]. Rate-based models are successfully used in experiments of a robot-based sensory and/or motor system [59].

2) SPIKE-BASED

A synaptic model with a spike-based learning rule was developed in the early 1980s [60], [61]. Several experiments showed that synaptic plasticity is influenced by the exact time of individual spikes. The phenomenon, which has been named Spike-Time-Dependent-Plasticity (STDP), alters the synaptic weight based on the relative timing of pre- and post-synaptic spikes: a causal relationship (pre- before post-synaptic spike) causes the synaptic weight to increase, while an anti-causal relationship (post- before pre-synaptic spike) causes the opposite effect.

Bienenstock et al. [62] have created a theoretical learning rule called the BCM rule. The basis of this algorithm is that the instantaneous firing rate rather than individual spikes (as for STDP) set the pattern of weight modifications. Izhikevich and Desai [63] have proven an equivalence of the BCM rule and the STDP spike-pair rule, opening both to common theoretical treatment and adding flexibility in specific model implementation choices.

In recent years the STDP learning rule has been successfully implemented in underlying neural learning mechanisms in robotics within both simulated and real environments. Equation 6 gives the precise mathematical definition of

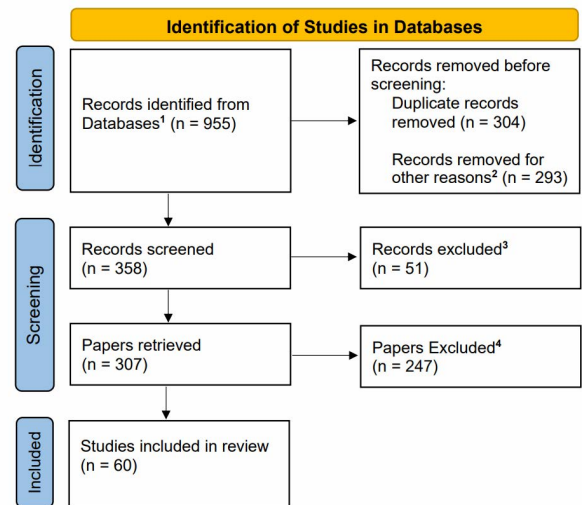


FIGURE 3. 1) Records retrieved in each database: IEEE-Xplorer = 112, Scopus = 594, Web of Science = 249. 2) This review considers paper from 2017 onward, so all publications before 2017 are removed. 3) By reading the meta-data, we found 51 publications that were either focusing on hardware implementation or didn't consider robotics and SNNs. 4) While screening, we labeled all the publications according to their focus, then we excluded all the papers not related to robotics.

STDP learning rule.

$$\text{STDP}(\Delta t) = \begin{cases} A_+ e^{-\Delta t/\tau_+}, & \text{if } \Delta t \geq 0 \\ -A_- e^{\Delta t/\tau_-}, & \text{if } \Delta t < 0 \end{cases} \quad (6)$$

where A_+ and A_- represent the strength of potentiation and depression, respectively. τ_+ and τ_- are positive and negative time constants.

III. SYSTEMATIC REVIEW METHODOLOGY

For this systematic review, we have selected three databases *IEEE-Xplorer*, *Scopus* and *Web-of-Science*. The keywords which we searched selected according to our goal to identify the scientific articles in the intersection of neuromorphic computing and interactive robotics. To this end, we identified two groups of keywords with similar meanings, respectively: 1) “neuromorphic computing”, “spiking neural network” and “brain inspired computing”; 2) “interactive robotics”, “social robotics”, “humanoid robotics”. In group 2, we also included “cognit*” to find cognitive models that may be used to learn and implement interactive behaviour in agents and robots. Finally, we use the pair “brain-inspired computing” AND “robot*”, because we wanted to find alternative methods by using the “brain-inspired computing” keywords but, when combined with the keywords of group 2, it resulted in only 2 articles. We searched all these keywords in the three databases as part of Title, Abstract and Keywords, and downloaded the resulting publications for an initial screening.

We added this meta-data to *Rayyan.ai* [64], an online tool for systematic review. The total number of articles we added was 955. Initially, we removed all duplicates, reducing the total count to 951. After that we decided to limit our systematic review to articles published on or after 2017. This reduced the total publication count to 358. Finally, we labelled all

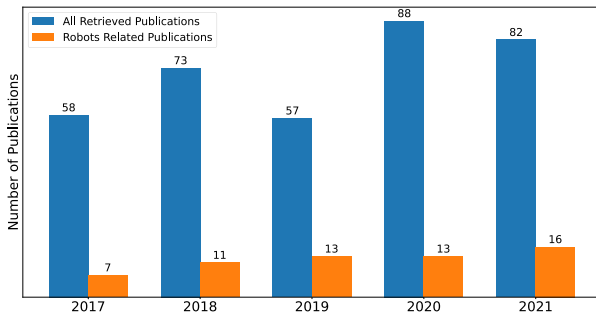


FIGURE 4. Total number of publications per year retrieved with our shortlisted keywords in all three databases, compared with robotics-related publications.

remaining publications by searching our keywords in title, abstract, and keywords. As this review focuses on socially interactive robotics and neuromorphic computing, we shortlisted publications presenting neuromorphic computing techniques to implement a new behaviour for human-robot interaction. In case of ambiguity in the metadata, we also reviewed the body of the article. The review was done independently by the three authors, articles were shortlisted only if two authors agreed that the article was pertinent. The number of publications that remain for review is reduced to 60. Figure 3 summarises the paper selection procedure using a PRISMA chart. Figure 4 shows the distribution per year of the publications retrieved, compared with the number of publications related to robotics which are considered for this systematic review.

IV. NEUROMORPHIC CHIPS/WWW/SIMULATORS AND FRAMEWORKS

In socially interactive neurorobotics, neuromorphic hardware is one of the essential elements for the robot to perform cognitive tasks. With recent advancements in neuroscience and in the chip industry, new neuromorphic hardware is introduced for the simulation of SNNs. In the tables below, we will briefly discuss the neuromorphic chips (Table 1), simulators (Table 2), and frameworks (Table 3) we came across during our review. We also discuss the robots (Table 4) used in our selected publications. There are several other neuromorphic chips (e.g. TrueNorth [65], Brain-drop [66], SyNAPSE [67], FACETS [68], NeuroMem [69], NM500 [70], SynSense [71]), simulators (e.g: Genesis [72], SpikeFun [73]) and humanoid robots (e.g. Pepper [24], Nao [74]) which are not discussed here as they were not used in any of the publications we reviewed. Some of the general-purpose simulators such as *Mayavi* [75] and *CSIM* [76] focus on simulating the environment for models, instead of simulating SNN models. Unlike other simulators, they do not have in-built neuron and/or synaptic models. Besides this, we also found several articles where no neuromorphic hardware is used. Input signals are obtained by sensors on the robot and SNN models are used as processing layers. We discuss more it in the following *Future Direction* section.

V. SNNs IN ROBOTICS APPLICATIONS

The use of SNNs in robotics introduces considerable complexity with limited benefits when performing simple tasks. In cognitive robotics, the goal is to understand the environment and compute the output. Such an approach usually returns useful insights for neural architectures and learned behaviour, especially when dedicated neural hardware is available. So far we have briefly discussed spiking neural networks and their models. In this section, we dive deeper into the applications of SNNs and neuromorphic computing in the field of socially interactive robotics. Robots provide an interesting testbed for SNNs, yet their application requires finding solutions to many problems such as power consumption, action duration, and output fidelity. Through our systematic review, we found 5 major directions in which contributions have been made. Before getting into the details of these applications, we summarise our analysis in Table 5 and Table 6. Here, Table 5 shows the robots and hardware used in the experiments, while Table 6 is about the software or simulation platform used to conduct experiments. These tables guide the reader through the type of robots and platforms that are typically used in experiments related to neurorobotics.

A. SIGNAL ACQUISITION AND PROCESSING APPLICATIONS

The implementation of robotic devices with intelligent sensors has been recognized as an important ingredient in the development of modern-day robots. A robot should plan and execute a series of operations autonomously while adjusting to the surrounding environment in real-time. Image segmentation and identification is one of the major objectives of a robot vision system. The features of the interesting object must be obtained and compared with the reference library for identification [120]. While much research has been conducted on vision-based identification, the combination of vision and non-vision sensors promise improvements in the speed of the recognition process. Rast et al. [28], demonstrated the learning system using an iCub robot and a SpiNNaker system to solve object identification tasks. The SpiNNaker neuromorphic system [121] is a neural network simulation platform, designed for real-time simulations. The details of hardware and chip architecture are given by Furber et al. [77]. The major goal of SpiNNaker is to achieve real-time processing of real-world data. Meanwhile, iCub is a research-grade humanoid robot with 53 degrees of freedom typically used in developing and testing embodied AI algorithms. The most common protocol used to communicate between iCub and host PC is called *Yet Another Robot Platform (YARP)* [122].

Furthermore Rast et al. [28] made several enhancements to the basic networks and showed how they can be used to direct performance towards behaviourally relevant goals. They observed the behaviourally relevant STDP appears to contribute strongly to positive learning as compared to negative learning. Paper demonstrated the integration of neuromorphic chips and humanoid robots to show how such

TABLE 1. Neuromorphic chips used in robotic applications.

Neuromorphic Chips	Description	Reference
1 SpiNNaker	Based on a biologically-inspired, massively parallel computing architecture, SpiNNaker can model and simulate neural networks that contain up to one billion neurons and one trillion synapses in biological real-time. System-on-Chip SpiNNaker is composed of 18 ARM968 processors residing in synchronous islands, surrounded by a lightweight packet-switched asynchronous communications network. It is a general-purpose programmable platform that enables neuroscientists, psychologists, and brain researchers to explore brain function with software neuronal models.	[77]
2 Loihi	Loihi is a 60-mm ² chip that can model spiking neural networks in silicon using Intel’s 14-nm process. It is equipped with a large multicore mesh with 128 neuromorphic cores, three embedded x86 processor cores, and off-chip interfaces that extend the mesh in four planar directions to other chips. There are a number of novel features integrated into the system, including hierarchical connectivity, dendritic compartments, synaptic delays, and programmable synaptic learning rules.	[78]

TABLE 2. Brief description of simulator and platforms used in robotic applications.

Simulator	Description	Reference
1 Brian	Brian is a Python package used to simulate networks of spiking neurons. Users provide mathematical equations which specify neuron and synaptic models, creating groups of neurons that are connected by synapses. In Brian researchers are able to use models that are not already integrated into the simulator.	[79]
2 Nengo	Nengo is Python library for building sophisticated spiking and non-spiking neural simulations with few lines of code. It is highly extensible and flexible. User can define neuron type, learning rules and get input from multiple sources.	[29]
3 NEST	NEST is an open-source simulator for spiking neural network models that focuses on the dynamics of the neural systems instead of the exact morphology of individual neurons. It is a command-line tool that can be used interactively from the Python prompt. NEST simulator also allows the creation of neuron models, connecting neurons with devices and defining synapse properties.	[80]
4 CSIM	CSIM is a general-purpose discrete-event simulation environment designed for modeling complex systems with interdependent elements. This software contains tools for creating hierarchical block diagrams and model libraries covering several domains.	[76]
5 AI-SIMCOG	The Artificial Intelligence Simulator of Cognition (AI-SIMCOG) is a SNN simulator that allows for easy design of complex real-time SNN architectures, their testing on complete autonomous entities in a virtual world, and their export to a physical robot.	[81]
6 NeMo	This spiking neural network simulator simulates networks of Izhikevich neurons on CUDA-enabled GPUs with an aim of achieving real-time simulations of 100,000 realistically connected neurons. This C++ library also has binding for C, Matlab and Python. It is also one of the supported backend for PyNN.	[82]
7 Mayavi	Mayavi is general purpose, cross-platform simulator for 2-D and 3-D data visualization. It has rich user interface with dialogs to interact with all visualizations. Besides this, it has simple scripting interface in Python, including ready to use 3D visualization functionality.	[75]
8 NEURON	NEURON is general purpose spiking neural network simulator use for building models and networks of neurons. It is easy to use command line simulator which provides binary installers for Mac, Linux and Windows.	[83]
9 Gazebo	Gazebo is robot simulator which can simulate robot populations efficiently and accurately in complex environments, both indoors and outdoors. With its strong physics engine, high-quality graphics, and convenient graphical and programmatic interfaces, it’s an excellent package.	[84]

TABLE 3. Description of frameworks we came across during our review.

Framework	Description	Reference
1 Neural Engineering Framework (NEF)	NEF is general methodology that allows to build large scale neural model for cognition. Here user can define the properties of neurons and connection weight between components that will perform the desired function. It works efficiently for feed-forward computations and for recurrent connections. NEF is also been used for model visual attention, inductive reasoning, reinforcement learning etc.	[29]
2 FenNueRobot Framework	FaNueRobot framework is built on top of the NeurCube framework for modeling spatiotemporal brain data. It is a motor control framework for a robot and prosthetic control which integrates an evolving SNN model of the brain and finite automata of neuromuscular behavior of the extensor muscles and forearm flexor during movements.	[85]

TABLE 4. Robots used in major applications.

Robot	Description	Reference
	<p>iCub</p> <p>It is a research-grade humanoid robot which is designed to develop and test embodied AI algorithm. This 1-meter-tall robot is a testbed for many social experiments.</p>	<p>[86]</p>
	<p>Pioneer 3DX</p> <p>It is a general purpose differential-drive mobile robot platform with multiple sensors, an onboard computer, autonomous navigation system, etc. It is typically used for applications like mapping, monitoring and teleoperations.</p>	<p>[87]</p>
	<p>HoLLiE</p> <p>It is a mobile, bi-manual service robot, developed as a part of 'House of living lab' project. HolliE can perform simple tasks like pointing at something specific but has the ability to conduct difficult operations such as doing social interactions.</p>	<p>[88]</p>
	<p>Hexapod</p> <p>It is an insect-like six-legged walking robot which is quite stable and versatile. It is widely used in neurorobotics for experiments like movement, obstacle avoidance etc.</p>	<p>[89]</p>
	<p>UR3</p> <p>It is a collaborative table-top robot, perfect for automated workbench scenarios like assembling tasks. Thanks to its compact nature it is ideal to work in tight spaces.</p>	<p>[90]</p>
	<p>Turtlebot-2</p> <p>It is a low-cost robot kit with open-source software. It gives you enough space to add desired sensors on it. Turtlebot-2 can drive around the house by 3D navigation and have enough power to create interesting applications.</p>	<p>[91]</p>
	<p>Khepera-IV</p> <p>Its is 5.5cm small, differential-wheeled mobile robot that is designed for indoor lab applications. Having high flexibility, Khepera can be used in applications like mapping, teleoperating and navigation.</p>	<p>[92]</p>
	<p>Soundman</p> <p>It is a commercial product designed for the recording of binaural sounds to maximize the perception of spatial effects</p>	<p>[19]</p>
	<p>Schunk SVH</p> <p>It is a five-finger robot hand with gripping system. The hand grip is nearly as perfect as human hand. Elastic gripping surfaces ensure a reliable grip on objects of various shapes and sizes.</p>	<p>[92]</p>
	<p>Cyber Rodent</p> <p>It has two wheels for locomotion, IR proximity sensors as its whiskers, an omnidirectional vision system as its eye and three-color LEDs for emotional communication. Typically used in neurorobotics learning applications.</p>	<p>[94]</p>

TABLE 5. Robots and hardware used in our major applications related to interactive robotics.

Major Applications	Robot	Hardware
Signal Acquisition and Processing	<ul style="list-style-type: none"> • iCub [28][95] • Pioneer3DX [96] 	<ul style="list-style-type: none"> • SpiNNaker [28][95][97] • RGB Camera, Kinect Sensor [22]
Pattern Recognition	<ul style="list-style-type: none"> • Pioneer 3DX [96] • Khepera IV [45] 	<ul style="list-style-type: none"> • RGB Camera [96] • Kinect sensor [98] • Raspberry Pi3 [99]
Speech Recognition	<ul style="list-style-type: none"> • iCub [19] [99] • Soundman [19] 	<ul style="list-style-type: none"> • Microphones [19] • iCub built-in sensors [100]
Motor Control	<ul style="list-style-type: none"> • HoLLiE [31] • Hexapod [23][101] • Roving [102][103] • UR3 [104] • Vector [105] • iRobot [106] • Khepera IV [30] • iCub [100] 	<ul style="list-style-type: none"> • Schunk SVH [107][108][109] • sMEG sensor, Myo sensor [108][109] • HoLLiE arm [31] • FPGA ZEM4310, Emotive Epop headset [23] • DVS, NAS [101] • UR3 with elbow and wrist [104] • Prosthetic hand [85] • Tobii Eye Tracker, YDLIDAR G4, Raspberry Pi 3 [99]
Cognition and Learning	<ul style="list-style-type: none"> • iCub [3] • Neco [110] • Turtlebot [111][112][113][111][114] • Virtual [42][115][116] 	<ul style="list-style-type: none"> • Robotic hand, Myo armband, Epop headset [25][117] • Loihi, Neural Computer Stick2 [111] • Spartan-6, RGB sensor [112] • Raspberry Pi3, PiCamera, PiStrom [116] • Loihi, LIDAR [113]

TABLE 6. Software/simulators and neuron models for all the applications.

Major Applications	Software/ Simulation Platform	Neuron Model
Signal Acquisition and Processing	<ul style="list-style-type: none"> • SpiNNaker [28][95] • CISM [22] 	<ul style="list-style-type: none"> • LIF [28][95][22]
Pattern Recognition	<ul style="list-style-type: none"> • Brian [118] • SIMCOG [99] [30] 	<ul style="list-style-type: none"> • LIF [97] [99] [30]
Speech Recognition	<ul style="list-style-type: none"> • Google ASR, SphinX [19] • EDULT [100] 	<ul style="list-style-type: none"> • LIF [100] • Pulse Neuron [19]
Motor Control	<ul style="list-style-type: none"> • NEF, Nengo [107][108][109] • NEST [31] • MATLAB [23][85] • NeuroNet [102][103] • NeMo [104] • Vector SDK [105] • FeNeuRobot Framework [85] • ROI [106] • SIMCOG [99] [30] 	<ul style="list-style-type: none"> • Manual [107][108][109] • LIF [31][101][85] [99] [100] • Izhikevich [104][103] • aEIF [105] • Pulse Neuron [106]
Cognition and Learning	<ul style="list-style-type: none"> • SIMCOG [115][116] • Nengo [111] • RatSLAM [110] • SNN Simulator [42] • NEF [119] • V-REP [114] 	<ul style="list-style-type: none"> • Created [111] • LIF [115][116][3][119] • Izhikevich [25][117] • SKAN [112] • SRM [110]

a system can learn to recognize and attend to preferred objects without relying on off-line training. SpiNNaker uses External/Internal Event Input-Output (EIEIO) protocol [123] which is the standard protocol of communicating AER data,

as well as general or device-specific commands between heterogeneous platforms. Figure 5 shows the general block diagram of the proposed system. In an extended publication, García et al. [95], showed how neuroanatomically grounded

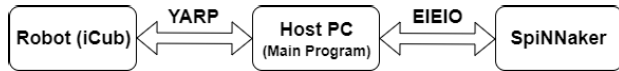


FIGURE 5. General block diagram of iCub-SpiNNaker system. The I/O from the robot is converted into YARP bottles. It is then processed by a host-based EIEIO transceiver which converts the message into spikes to transmit and receive from SpiNNaker.

SNNs for visual attention can be extended with word learning capabilities. After the completion of learning, the robot was able to call the name of the object when visual input was present.

As significant research has been conducted to understand the processing of multi-sensory information, neuroscientists, psycho-physicists, and psychologists suggested models explain how information from different sensors is integrated in order to perceive the environment. The term *convergence-zone* was proposed by Damasio [124], which is one of the plausible models that explain the mechanism of the multi-modal human perception process. Inspired by this, Al-Qaderi and Rad [22] proposed a multi-modal perceptual system for social robotics. Unlike the majority of studies in machine perception which deal with uni-model sensory systems, they proposed a multi-model system which uses concepts from fading memory [21], binding criteria [125], cell assemblies [52], and top-down influences [126]. To access the performance of their system they selected Pioneer 3DX which is a popular research mobile platform. They equipped the robot with an RGB camera, Kinect sensor, directional microphone, and sonar sensor. Experimental results proved that multi-model perceptual systems perform better than uni-modal systems.

B. PATTERN RECOGNITION APPLICATIONS

Visual or pattern recognition is a fundamental component for most robotic systems operating in the real world. A large variety of tasks related to human-robot interaction require visual or pattern recognition. Therefore, a lack of successful recognition is often an impediment to applying robotics systems to real-world situations. In particular, in cognitive robotics visual or pattern recognition is a building block of complex systems including many other components such as pose estimation, grasping and manipulation [127]. So far, this problem has received comparatively little attention and was usually dealt with through methods that require strict supervision during the training phase, such as uncluttered views of the objects or meta-data about object position or orientation towards the camera.

Fanello et al. [27] proposed some improvements in the robot visual perception capabilities with a limited amount of constraints. They modified state-of-the-art coding-pooling pipelines for visual recognition to improve the representation while maintaining real-time performance. Recently, Mansouri-Benssasi and Ye [118] tried to explore all the major applications of bio-inspired SNNs with unsupervised learning using STDP for facial expression recognition. They pre-processed the images with Laplacian of Gaussian (LoG)

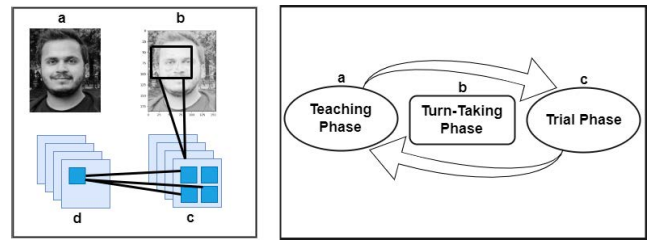


FIGURE 6. Left figure: represents the SNN workflow for facial expression recognition. a) is a raw image b) image with LoG filter and Poisson spike train creation with convolution layer. c) Excitatory layer. d) Inhibitory layer [118]. Right Figure: a) Teaching phase where learner visualize the target action. b) In turn-taking phase, learner extract the nonverbal information. c) In Trial phase, learner confirms the target action [98].

filters to detect the edges and contours of the facial image. After that a spike train is created using a Poisson distribution where the firing rate is directly proportional to the input pixels' intensity. Lastly, they adopt the online STDP [128] to perform unsupervised learning, see Figure 5 (left). The proposed approach was evaluated on two publicly available facial expression datasets and achieved better accuracies to some of the popular methods like Histogram of Oriented Gradients (HOG) features or CNN. In another publication by Al-Qaderi and Rad [96], the multi-modal perceptual system was introduced for efficient facial recognition. The experiments were conducted in real-world scenarios where a robot has to recognize the face of the user while moving around.

Besides this, experiments have been conducted to recognize the motion of the human being in real-time by using SNNs and clustering of direction vectors to analyse the nonverbal information and construct a cognitive system to recognize the target motion and acquire actions [98]. This instructed learning is done in three phases. First is the teaching phase where a teacher is sharing the cognitive environment and the learner visualises and confirms the target action. In the turn-taking phase the learner extract the nonverbal information, and in the final stage, the learner imitates the action. Figure 6 (right) shows the block diagram of this instructed learning.

In another process, proposed by Cyr and Thériault [99], robots learn the relationship between left/right and horizontal/vertical visual stimuli, regardless of the location of the image or their specific pattern composition. Here, different patterns were shown to the robot, and the correct recognition along with appropriate movement was expected from it. Experiments also show that with the different rewarding rules, SNN can adapt its behaviour in real-time.

Similarly, Cyr, Thériault, and Chartier [30] propose a 2-bit task (XOR) with visual compound binary images as input and the left/right action for the output. Here the robot can adapt its behaviour from learning other simpler associative rules during runtime. Furthermore, the impact on the neural architecture was also explored when passing from a 2-bits to a 3-bits task.

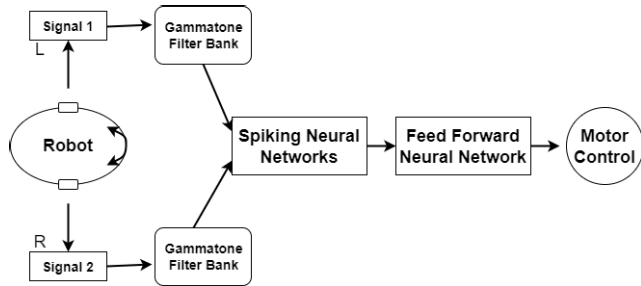


FIGURE 7. The Spike Source Localization (SSL) architecture. Sound signals are received at the left and right receiver of a robot. These sound inputs are decomposed into several frequencies with Gammatone Filterbanks. After that, it is fed to Spiking Neural Networks and then to Feed-Forward neural network for classification. This classification layer produces an output angle that is used to control the motor [19].

C. SPEECH RECOGNITION APPLICATIONS

Speech recognition is the ability of a machine or robot to identify words and phrases in language and convert them into machine convenient language. In most common cases of speech recognition in robots, the voice command is taken through a microphone then processed in a computer, and finally sent back to the robot for action. Sometimes it is hard to recognize speech efficiently due to the noise environment, to solve this problem Davila-Chacon, Liu, and Wermter [19] propose the embodied embedded cognition approach to improve Automatic Speech Recognition (ASR) systems for robots in noisy environments. Sound Source Localization (SSL) was used to locate the direction of the sound source accurately in a short time. Figure 7 represents the SSL architecture. In this approach, before doing an ASR task, the robot orients itself toward the angle where the signal-to-noise ratio (SNR) of speech is maximised for one microphone. Here a spiking neural network was applied to calculate the sound signal angle. The system was tested both on the iCub robot and Soundman, and its performance was measured and used as a baseline for the iCub robot head. The result of this test shows that the proposed ASR with SSL can considerably improve the accuracy of speech recognition in a humanoid robot. Moreover, in the proposed approach where SSL directed the orientation of the robot and selected the appropriate channel as input to the ASR system, showed better results. This was unlike other approaches, where sound signals for both channels (left and right) are averaged before feeding to the ASR system. In the future, speech recognition can also be used as the bootstrapping mechanism to train neural layers as it can perform auditory grouping in frequency and time domain [19].

In another approach, a robot (iCub) was used to perform vestibulo-ocular reflex (VOR) tasks, which applied a spiking cerebellar model comprising an adaptive real-time control loop [100]. It operates as a feed-forward controller that integrates several neural models, certain neural topology and characteristics. The cerebellar model effectively adapts the reflex for a robot by using STDP which generates the eye motor commands to compensate for the head movement of the iCub robot.

D. MOTOR CONTROL APPLICATIONS

The goal of many studies in this field is to develop a robot that can mimic human cognitive and motor behaviour. From doing simple tasks like pointing in different directions to performing advanced learning tasks, movements, grasping and touching actions in robotics have several applications. The method was proposed to generate and control pointing motions for a robot using a spiking neural network. The SNN learns a base motor primitive for pointing at targets (left, right, up, down). The network was able to combine multiple motor primitives to control a robot in real-time to reach a specific point. The performance of the network was evaluated with the help of a humanoid robot (HoLLiE). The board plane with targeted points was placed in front of the robot and the robot has to produce motion towards a target point. The major benefit of this approach is that it is not dependent on the specific kinematic structure of the robot and can be used with different robots [31]. Figure 8 shows the overall architecture of the process. Here the activation pattern is produced by the motion generation layer for four correction primitives and the motion command is given to the motor control system. Similar work is done by Mirus et al. [129]. Here the mobile robot is capable of moving around and detecting objects in an unknown environment.

A different study by Batres-Mendoza et al. [23] presents the development of the real-time locomotion systems for the Hexapod robot using bio-inspired computing. The improved Quaternion based Signal Analysis method (iQSA) method based on quaternion algebra is used for processing and classification [130]. Here the data was collected from 120 users to create a decision rule for the iQSA method. The proposed system has three major parts: the first is a signal acquisition through a Brain-Computer Interface (BCI), which translates electrical signals from the brain into computer input. In the second step, this acquired data is analysed and processed through the iQSA method. Finally, a SNN is used to mimic the Central Pattern Generator (CPG) [131] behaviour to control robot movements. CPGs are a specialised neural network that is capable of creating rhythmic patterns without the use of sensory input. These patterns allow us to coordinate and control repetitive activities, such as chewing, swimming, walking, and running. Figure 9 presents the block architecture of a system. The system was implemented in real-time for performance evaluation. The experimental results show that the robot was able to replicate the gait pattern generated through the user's mental activity with a slight delay.

Similarly, Lele et al. [101] used CPG and coupled it with Dynamic Vision Sensor (DVS) for a prey-tracking scenario in the close-loop robotic system. Here the legs of a Hexapod robot are controlled by a network of spiking neurons. There are six neurons, each controlling one leg, and each spike of these neurons causes movements in the corresponding leg. For the learning of various gaits, a supervision-based weight adaptation algorithm is proposed. The result shows that a maximum of three gaits can be programmed using this

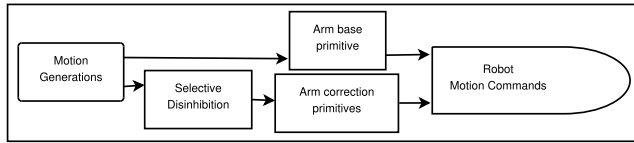


FIGURE 8. Brief block diagram of the motion generation approach. It contains three major components: first, the motion generation layer produces circular activity that creates activation patterns for primitives. Second, the motor control layer has arm base primitive and arm correction primitives for pointing motion and to point to target, respectively. Third, the target layer takes the relative distance between target and base point for selective excitation to activate the correction primitives.

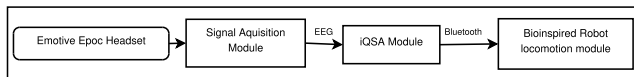


FIGURE 9. Communication architecture between the Brain-Computer Interface (BCI) and the Hexapod robot. The EEG signals are acquired through the Emotive Epos headset. This is then transferred to the iQSA module to determine robot movements. Finally, commands are given to the robot locomotion module via Bluetooth [131].

algorithm with high energy efficiency when implemented on modern neuromorphic hardware.

Alongside with the above-discussed methodologies, the movement of a robotic arm has also remained an area of focus. The model proposed by Zahra, Navarro-Alarcon, and Tolu [104] shows how the controller based on cellular-level guides the motion of the robot arm with real-time data from sensors. The model is inspired by the biological features of the cerebellum, by monitoring the firing rate and pattern of the different groups of neurons in it. This model contains two layers of neurons: one input, one output, with all-to-all connections between them, to provide transformation between two correlated spaces. The network correlates the spatial velocity in the two layers, it acts as a differential map, so the concept of Differential Mapping Spiking Neural Network (DMSNN) is utilised. For learning the STDP is put to use. The model was tested with the UR3 universal robot, where the elbow and shoulder joints are controlled to test the manipulation of the end actuator to a specific position. The experiment shows that such a model can reduce error in a certain direction with fast convergence of learning. Moreover, it can be developed further for real-time adaptive robot control in multiple challenging environments and applications [132], [133]. In addition, another study showed a similar architecture where a virtual robot controlled by a specific SNN could independently learn simpler associative rules [30]. Additional to this, Tieck et al. [107] showed the concept of soft-grasping to control a robotic arm with SNN. The robotic arm is able to grasp objects of different sizes, stiffness and shapes without calculating complex contact point planning or inverse kinematics. This approach required only one example of each grasping motion to train the primitive and it can be used on different robots having similar features of the human hand. In another experiment, they used sEMG signal to activate motion reflexes on a robotic hand. The trained network can classify the sEMG signals and detect finger activation.

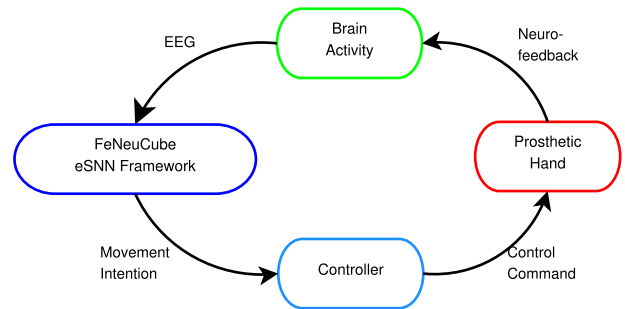


FIGURE 10. The basic block diagram of the prosthetic control through a Brain-Computer Interface (BCI). As shown, the brain provides the EEG signals to the FeNeuCube framework which, in turn, gives instructions to the controller. Finally, the controller forwards the control command to prosthetic hand [85].

The reflexes of the finger are modelled with motion primitives and mapped to a robot kinematics [108], [109].

There are also applications of neurorobotics in the field of medical science. For example, Kumarasinghe et al. [85] presented the proof of concept study for prosthetic control with the Brain-Computer Interface (BCI). They used finite automata theory and a NeuCube evolving SNN architecture. The prosthetic hand was designed to perform tasks such as grasping and touching. Figure 10 shows the basic architecture of the proposed BCI. Here the learning is happening in two stages: the first stage uses an STDP unsupervised learning rule where evolving synaptic connections are formed according to the difference between pre- and post-synaptic neurons. With this approach, SNN cube will activate the same group of neurons when a similar input stimulus is provided. The second stage uses a supervised learning paradigm to update the connection weight between output and hidden layers. The framework (FaNeuRobot) integrates the evolving spiking neural network model of the brain with finite automata that tells the neuromuscular behaviour of forearm muscles and flexor during movement.

Another interesting study focuses on solving movement issues for disabled and elderly people. In most of the existing systems, a user has to control the robot manually which could be difficult for the elder or disabled people. Obo et al. [106] presented a multi-modal interface to control the robot remotely. Besides this, a cognitive platform to control robots based on the concept of the perception-action cycle is also proposed. Here a spiking neural network is used for spatio-temporal modelling of the interaction between the environment and the user. A self-organised neural network based on an unsupervised learning paradigm is used in this system. They developed a seat with pressure sensors on it and the robot movement is dependent on the user's movement while sitting on the seat. Beside this, a stationary eye-tracker (Tobii Eye Tracker 4C) is attached to retrieve the gaze and head pose. Figure 11 shows the summarised proposed system architecture which has two major parts, the perceptual system and the action system. The experimental results show that the teleoperation system can change the sensitivity of the interface according to the operation.

TABLE 7. Learning mode, rule and paradigm in each application.

Major Applications	Online/Offline Learning	Learning Rule	Learning
Signal Acquisition and Processing	<ul style="list-style-type: none"> • Online [28][95][22] 	<ul style="list-style-type: none"> • STDP [28][95] 	<ul style="list-style-type: none"> • Reinforcement [28][95][22]
Pattern Recognition	<ul style="list-style-type: none"> • Online [99][30] • Offline [118][98] 	<ul style="list-style-type: none"> • STDP [118][98] 	<ul style="list-style-type: none"> • Unsupervised [118][98] • Reinforcement [118][98]
Speech Recognition	<ul style="list-style-type: none"> • Online [100] 	<ul style="list-style-type: none"> • STDP [100] 	<ul style="list-style-type: none"> • Supervised, Feedback error learning [100]
Motor Control	<ul style="list-style-type: none"> • Online [107][105][99][30][100] • Offline [31][101][85][23][106][85] 	<ul style="list-style-type: none"> • STDP [104][105][103] 	<ul style="list-style-type: none"> • Supervised [31][101][85][104] • Unsupervised [23][106][85][105] • Feedback-error learning [100]
Cognition and Learning	<ul style="list-style-type: none"> • Online [113][42][114][119] • Offline [112] 	<ul style="list-style-type: none"> • STDP [25][117][115][116][110] 	<ul style="list-style-type: none"> • Reinforcement [115][116][119][114] • Supervised/ Unsupervised [111] • Feedback-error learning [100] • Reinforcement [112][42] • Synaptic learning [110]

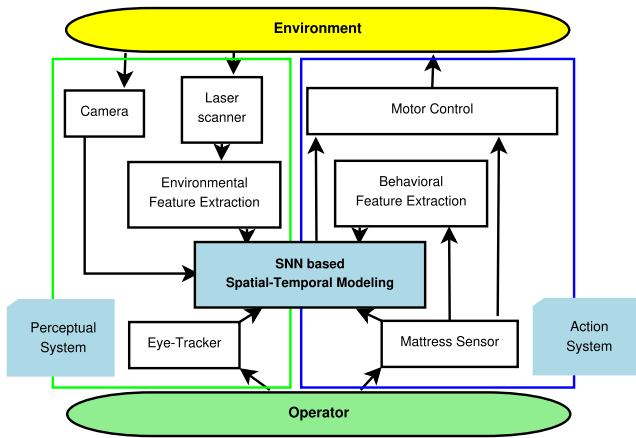


FIGURE 11. shows the summarised system architecture of the system. It has two major parts: 1) the perceptual system, where the information of environmental map find a use to detect the space where a robot can move around. Moreover, the self-organized neural network is utilized to extract perceptual information. 2) the action system, behavioural features in teleoperating are extracted and commands are given to motor control. Based on the perception-action cycle, SNN is used for spatio-temporal modelling.

E. COGNITION AND LEARNING APPLICATIONS

Biological systems generally have a memory, which is defined as the ability to preserve, learn and reproduce past adaptive states. As discussed earlier, synaptic plasticity is an important mechanism of memory and learning on a cellular level. Several mathematical models exist that can simulate cognitive maps, where synaptic plasticity yields the emergence of spatial memory in SNNs. Spatial memory in robots is used for storage and retrieval of information that is used to plan a route to a desired location and to remember where an object is located or where an event occurred [102]. In this subsection instead of discussing the classifications of

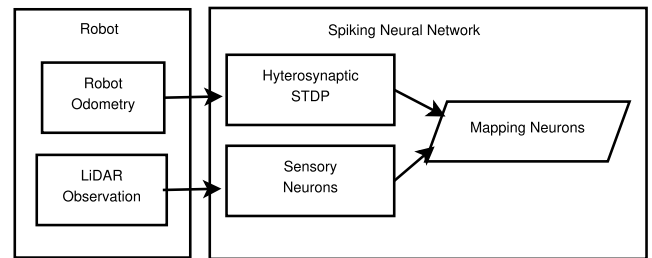


FIGURE 12. The network for real-time mapping with Loihi chip. Here robot and LiDAR are providing inputs to the SNN for mapping [113]. b) The basic block diagram of the prosthetic control through a Brain-Computer Interface (BCI). As shown, the brain provides the EEG signals to the proposed FeNeuCube framework which, in turn, gives instructions to the controller. Finally, the controller forwards the control command to prosthetic hand [85].

learning [1] we discuss the contributions related to spatial memory, learning and using that learning to predict the next action for social robots. Table 7 details learning mode, rule and paradigm in each application.

Spatial mapping is an important component for developing Simultaneous Localization and Mapping (SLAM) in social robots. Tang and Michmizos [113] presented an algorithm that solves the navigation problem for robots by learning the environment using a specialised neural network. This biology-inspired network is integrated with the Loihi neuromorphic chip which interacts with the Robot Operating System (ROS) in real-time, see Figure 12. The robot is equipped with a 360-degree LiDAR sensor. Here the SNN uses Winner-Take-All (WTA) structure and heterosynaptic competitive learning for place field generation and dendritics for reference frame transformation. The algorithm gave an accurate environmental map by using error-free odometer signals from the Gazebo simulator.

Cyr and Theriault [99] used the concept of operant conditioning for a robot to learn the spatial map through reinforcement and punishment. The robot can learn the relationship between different visual stimuli irrespective of their pattern or location on the images. The final results show that when the rewarding rule is changed, the SNN can adapt its behaviour in real-time. They also presented two exploration strategies in a virtual robot controlled by a SNN. A virtual robot controller simulates the thigmotaxis (the movements of one organism either towards or away from the stimulus) and boldness (the propensity to engage in risky behaviour) behaviours. The network performed visual learning tasks solved through an operant conditioning procedure [115], [134]. Similarly, another article presents learning in robots where SNN can implement several variations of learning through classical conditioning with positive or negative reinforcement. The SNN model is implemented on Field-Programmable Gate Array (FPGA) and a Synapto-Dendritic Kernel Adapting Neuron (SKAN) model is used for neural delay [112].

The publication by Zharinov et al. [103] showed the model of spatial memory implemented on SNN. This model was then tested on a robot moving in an environment with neutral and harmful regions. Here the dynamics of the neural population determines the movement of a robot. The STDP learning rule rearranges the SNN coupling and forms spatial memory according to the surroundings. After training, the robot learns to avoid the harmful region. A similar approach was presented by Lobov et al. [102] where the robot has to explore the surroundings and after the training is completed, it should determine where the harmful region is and avoid moving to that region. The proposed network can remap neutral and harmful regions when the dangerous zone moves to another place. Thus, the robot adapts to the changing world. Discussing memory, another model was presented for associative memory in the form of SNN. A mobile robot was used to demonstrate how neuromorphic hardware can reduce energy consumption with the same computing power when implemented on a Loihi neuromorphic chip [111]. Moreover, associative learning tasks were also performed to investigate the abstract relation of the sameness/difference (SD) model in bio-inspired robots. The model uses artificial SNN as the robot's brain-controller and STDP as a learning rule. The experimental results show that robots can learn in different scenarios depending on their previous action and applied reinforcing rules [116]. Improvements are also made in the working memory of robots using SNNs with dynamic synapses. The proposed model was able to refine, overwrite or resist the change in the duration and configuration of incoming stimuli. They showed how local changes in the environment can be accounted for short or long-term changes in synaptic plasticity. The online unsupervised learning with the STDP learning rule is used for working memory of robots so it can perform tasks in evolving environments [105].

Besides this, work has been done in the direction of episodic memory for robots, where a robot can perform more versatile cognitive tasks like exploration, localization, and

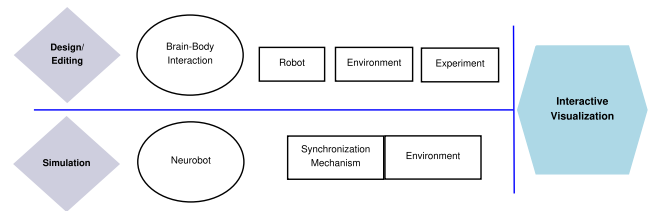


FIGURE 13. Block diagram for a functional overview of the Neurobotics platform. With the design/editing feature, users can conduct neurobotic experiments using the brain model and robot body which can interact in a dynamic environment. These experiments are simulated and the results can be displayed in an interactive fashion.

navigation [110]. For example, if the robot has to make and serve milk tea. For this task, the robot needs to be familiar with the environment through a cognitive map and episodic memory to perform tasks in an appropriate order. Based on the idea of behaviour-based robotics, an algorithm is developed that can robustly perform navigation and exploration tasks. Additionally, another limitation for self-learning in robotics is also explored which is related to having a sense of time. The brain-inspired spatial cognitive system is developed that integrates episodic memory and cognitive map. This system helps the robot to recognize and remember different locations while storing the correct sequence to perform tough tasks. The algorithm was presented to overcome the fundamental issue of designing the topologies and parameters for the SNN. It was evaluated with a simple sensory-motor decision task using evolutionary computation. They showed how some variations in topology and parameters can affect the behaviour of a system. In the experiments, the complexity of the task increased gradually so that the algorithm could evolve with it [42]. This approach can be used in different learning tasks for social robotics where the robot is in a continuous changing environment [135]. Addition to this Huang, Wu, and Qiao [114], shows how emotion plays an important role in participating in value calculation, as an important intrinsic motivation of decision making. The Turtlebot-3 used the model-based decision making approach with emotional intrinsic rewards to solve continuous control problems. Similarly another research illustrated the proof of concept for biologically-plausible socio-emotional robots. The robot was able to execute an amygdala model to determine emotional state from visual input. The model was processed by using SpiNNaker, Loihi and Braindrop neuromorphic chips [119].

As mentioned earlier, developing brain-inspired networks that mimic the complex structure of the brain is a very difficult task. Collaboration between a researcher from biology, neuroscience and computer science helps to design somewhat biologically realistic models of the brain on SNN. To overcome this complexity, the Neurobotics Platform was introduced to easily establish communication between available brain and body models. It is a web-based environment where brain models can connect to a detailed simulation of robot bodies [136]. There are built-in experimental sequences, environments, conditions, robots and brain-body connectors

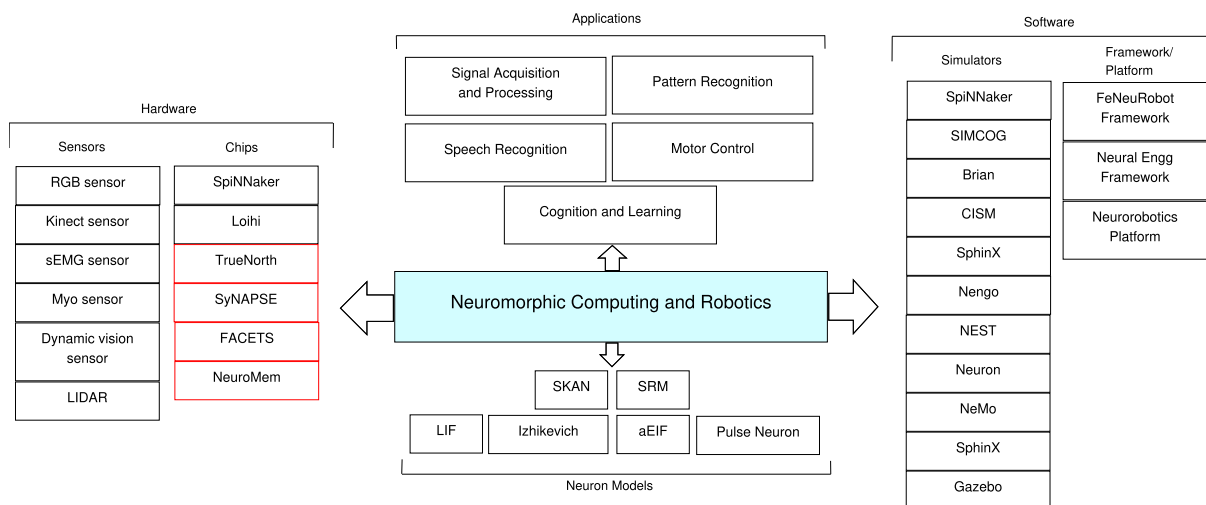


FIGURE 14. An overview figure of the landscape of neuromorphic computing and robotics. It shows major hardware, software, neuron models and applications. The red boxes in chips column shows that they are not used in robotic applications yet.

to help users with less programming background [3]. This project is part of the EU flagship Human-Brain Project (HBP). The functional overview of the Neurorobotics platform is presented in Figure 13.

Figure 14 is an expanded form of Figure 1. It shows the sensors and chips in the hardware section. Here the red outline boxes show that the chip is not used in robotics applications yet. The software section contains a list of simulators, frameworks and platforms. It also shows the neuron models typically used in robotics applications.

VI. FUTURE DIRECTION

In the previous sections, contributions and advancements in SNN-based social robotics have been reviewed in terms of their focused applications. This section focuses on the critical analysis of reviewed articles. We discuss the shortcomings in this area and what are the major areas on which future research should focus.

Humanoid robots: Although many experiments are conducted in the area of neuromorphic computing for robotics, most of them are very basic when it comes to cognitive tasks [105], [110], [99]. Usually, they focus on robots moving around a dedicated environment, constructing the spatial map, or storing some information in its memory block. Few pieces of research considered human involvement in the experiments for detection and identification but they lack human-robot interaction in social context. Besides this, none of the experiments were done with a humanoid robot. So, there is still a need to conduct experiments with a humanoid robot where the robot also interacts with humans instead of just interacting with the environment.

Neuromorphic Chips: As we can see in Table 4, only two types of neuromorphic chips are used in our reviewed papers. Experiments with Intel’s Loihi chip are mostly related to improving the memory or spatial mapping while experiments with the SpiNNaker architecture are focused on behavioural

learning and visual attention. In the experiments with SpiNNaker, the neuromorphic chip is not attached to the robot. The SpiNNaker system receives inputs from the sensors in the robot via the EIEIO protocol for processing and returns action commands. On the other hand, experiments with the Loihi chip don’t contain humanoid robots. In most of the reviewed articles a turtlebot is used, which is equipped with multiple sensors and a Loihi chip. This shows that there are still several open options like using other available neuromorphic chips and conducting experiments in more complex scenarios where the humanoid robot has to make decisions in real-time.

Hardware: Many experiments conducted in our reviewed articles use non-spiking sensors (e.g. web camera is used instead of a retina camera [127]). These sensors send the signal to the processing unit, which converts it to spikes and feeds them to the neuromorphic hardware. This process could get faster and energy efficient if neuromorphic hardware is used instead, which is an important requirement when it comes to social robotics.

Personalization: Another challenge faced by the social neurorobotics field is about personalising the robot. This can only be done through interdisciplinary research of roboticists and neuro-scientists. Usually, roboticists use simplified brain models to make real-time simulations, while neuro-scientists work on detailed brain models which are difficult to embed into the real world due to their high complexity. The community needs more solutions like the Neurorobotics Platform which provides adequate tools to model highly detailed environments, virtual robots, and complex neural networks for both roboticists and neuro-scientists.

Generalised Framework: None of the reviewed articles offers a general-purpose framework that could offer functionalities of training and modelling. Besides this, the available algorithms to convert ANNs to SNNs [137], [138], [57], [58] are still in preliminary stages because accuracy of converted SNN models is much lower than ANNs [139]. The nature of

this situation is that training SNNs for deep networks is notoriously difficult. In the future, advancements in these ANNs to SNNs conversion algorithms are required. Moreover, specialized mechanisms for SNN models performance analysis are lacking. There is a need for better mechanisms to evaluate the computational capabilities such as power consumption and speed, which are of vital importance for social robotic applications.

Ethics: Lastly, some aspects are rarely considered till now, such as ethical aspects in social neurorobotics. The development of social neurorobotics is still in its early stages, which makes it an ideal candidate for proactive and anticipatory ethical reflection. The major concern is *trust and safety* when it needs to be decided whether to use the robot in a social environment or not. Another aspect that might affect the adaptability of social neurorobotics is *data privacy*. Where and how data from the sensors of robots is being processed and how to share this data with another robot in a socially interactive environment. Therefore, research is needed to ensure that the development of neurorobotics is ethical, desirable, and socially acceptable.

VII. CONCLUSION

The neuromorphic computing approach has shown great promise for achieving human-like robotic intelligence in terms of computation, speed, and energy efficiency. This will be greatly beneficial in building robots that can exhibit realistic human-like social interaction. To support further research toward socially interactive neuromorphic robotics, this article delivered a systematic literature review of neuromorphic computing methods and tools that can be applied to improve socially interactive robotics. It first introduced the biological justification of SNNs along with their general architectures. After that, we presented mainstream neuron and synapse models that are used to design SNNs. The reviewed articles were distributed according to the application or problem they are trying to solve and we highlighted five major areas of application (signal acquisition and processing, pattern recognition, speech recognition, motor control, cognition and learning). Finally, the most relevant neuromorphic chips, simulators, frameworks, and robots are discussed in the view of their application to socially interactive robotics. Most of the reviewed articles focus on improvements of the existing techniques for one of the specific applications rather than exploring their full integration to build more capable systems. Only a few pieces of research demonstrated social interaction between robots and humans in real-world scenarios, but robots had several limitations and were able to interact only in a closed-loop fashion. Besides this, the lack of universal training methods and conversion mechanisms for neuromorphic models is also a major challenge for the development of human-like social interactions in neurorobots. To this end, we remark that more interdisciplinary collaborations, especially between roboticists and neuroscientists, are needed to fully develop the merge between neuromorphic computing and social robotics.

REFERENCES

- [1] Z. Bing, C. Meschede, F. Röhrbein, K. Huang, and A. C. Knoll, "A survey of robotics control based on learning-inspired spiking neural networks," *Frontiers Neurobot.*, vol. 12, p. 35, Jul. 2018.
- [2] Y. Bengio, I. Goodfellow, and A. Courville, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2016. [Online]. Available: <https://pubmed.ncbi.nlm.nih.gov/26017442/>
- [3] E. Falotico, L. Vannucci, A. Ambrosano, U. Albanese, S. Ulbrich, J. C. V. Tieck, G. Hinkel, J. Kaiser, I. Peric, O. Denninger, and N. Cauli, "Connecting artificial brains to robots in a comprehensive simulation framework: The neurobotics platform," *Frontiers Neurobot.*, vol. 11, p. 2, Jan. 2017.
- [4] K. Roy, A. Jaiswal, and P. Panda, "Towards spike-based machine intelligence with neuromorphic computing," *Nature*, vol. 575, no. 7784, pp. 607–617, Nov. 2019.
- [5] S. Furber, "Large-scale neuromorphic computing systems," *J. Neural Eng.*, vol. 13, no. 5, Oct. 2016, Art. no. 051001.
- [6] S. R. Nandakumar, M. Le Gallo, I. Boybat, B. Rajendran, A. Sebastian, and E. Eleftheriou, "A phase-change memory model for neuromorphic computing," *J. Appl. Phys.*, vol. 124, no. 15, Oct. 2018, Art. no. 152135.
- [7] P. A. Dananjaya, R. Gopalakrishnan, and W. S. Lew, "RRAM-based neuromorphic computing systems," in *Emerging Non-volatile Memory Technologies*. 2021, pp. 383–414.
- [8] Y. Liu, J. Gao, F. Wu, H. Tian, and T.-L. Ren, "The origin of CBRAM with high linearity, on/off ratio, and state number for neuromorphic computing," *IEEE Trans. Electron Devices*, vol. 68, no. 5, pp. 2568–2571, May 2021.
- [9] P. Rzeszut, J. Chęciński, I. Brzozowski, S. Ziętek, W. Skowroński, and T. Stobiecki, "Multi-state MRAM cells for hardware neuromorphic computing," *Sci. Rep.*, vol. 12, no. 1, pp. 1–11, May 2022.
- [10] A. Chaudhuri, M. Liu, and K. Chakrabarty, "Fault-tolerant neuromorphic computing systems," in *Proc. IEEE Int. Test Conf. (ITC)*, Nov. 2019, pp. 1–10.
- [11] I. K. Schuller and R. Stevens. (Oct. 2015). *Neuromorphic Computing: From Materials to Systems Architecture Roundtable*. [Online]. Available: <https://www.osti.gov/biblio/1283147>
- [12] S. R. Nandakumar, S. R. Kulkarni, A. V. Babu, and B. Rajendran, "Building brain-inspired computing systems: Examining the role of nanoscale devices," *IEEE Nanotechnol. Mag.*, vol. 12, no. 3, pp. 19–35, Sep. 2018.
- [13] C. Bartolozzi, G. Indiveri, and E. Donati, "Embodied neuromorphic intelligence," *Nature Commun.*, vol. 13, no. 1, pp. 1–14, Dec. 2022.
- [14] X. Wang, X. Lin, and X. Dang, "Supervised learning in spiking neural networks: A review of algorithms and evaluations," *Neural Netw.*, vol. 125, pp. 258–280, May 2020.
- [15] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bull. Math. Biophys.*, vol. 5, no. 4, pp. 115–133, 1943.
- [16] W. Guo, M. E. Fouda, A. M. Eltawil, and K. N. Salama, "Neural coding in spiking neural networks: A comparative study for robust neuromorphic systems," *Frontiers Neurosci.*, vol. 15, p. 212, Mar. 2021.
- [17] W. Maass, "Networks of spiking neurons: The third generation of neural network models," *Neural Netw.*, vol. 10, no. 9, pp. 1659–1671, Dec. 1997.
- [18] T. Iakymchuk, A. Rosado-Muñoz, J. F. Guerrero-Martínez, M. Bataller-Mompeán, and J. V. Francés-Víllora, "Simplified spiking neural network architecture and STDP learning algorithm applied to image classification," *EURASIP J. Image Video Process.*, vol. 2015, no. 1, pp. 1–11, Dec. 2015.
- [19] J. Davila-Chacon, J. Liu, and S. Wermter, "Enhanced robot speech recognition using biomimetic binaural sound source localization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 1, pp. 138–150, Jan. 2019.
- [20] E. M. Izhikevich, "Which model to use for cortical spiking neurons?" *IEEE Trans. Neural Netw.*, vol. 15, no. 5, pp. 1063–1070, Sep. 2004.
- [21] W. Maass, T. Natschläger, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural Comput.*, vol. 14, no. 11, pp. 2531–2560, 2002.
- [22] M. Al-Qaderi and A. Rad, "A multi-modal person recognition system for social robots," *Appl. Sci.*, vol. 8, no. 3, p. 387, Mar. 2018.
- [23] P. Batres-Mendoza, E. I. Guerra-Hernandez, A. Espinal, E. Perez-Careta, and H. Rostro-Gonzalez, "Biologically-inspired legged robot locomotion controlled with a BCI by means of cognitive monitoring," *IEEE Access*, vol. 9, pp. 35766–35777, 2021.
- [24] A. K. Pandey and R. Gelin, "A mass-produced sociable humanoid robot: Pepper: The first machine of its kind," *IEEE Robot. Autom. Mag.*, vol. 25, no. 3, pp. 40–48, Sep. 2018.

- [25] T. Zhou and J. P. Wachs, "Early turn-taking prediction with spiking neural networks for human robot collaboration," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers, Sep. 2018, pp. 3250–3256.
- [26] A. Kugele, T. Pfeil, M. Pfeiffer, and E. Chicca, "Hybrid SNN-ANN: Energy-efficient classification and object detection for event-based vision," in *Pattern Recognition (Lecture Notes in Computer Science)*, vol. 13024. Cham, Switzerland: Springer, Sep. 2021, pp. 297–312.
- [27] S. R. Fanello, C. Ciliberto, N. Noceti, G. Metta, and F. Odone, "Visual recognition for humanoid robots," *Robot. Auto. Syst.*, vol. 91, pp. 151–168, May 2017.
- [28] A. D. Rast, S. V. Adams, S. Davidson, S. Davies, M. Hopkins, A. Rowley, A. B. Stokes, T. Wennekers, S. Furber, and A. Cangelosi, "Behavioral learning in a cognitive neuromorphic robot: An integrative approach," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 12, pp. 6132–6144, Dec. 2018.
- [29] S. Sharma, S. Aubin, and C. Eliasmith, "Large-scale cognitive model design using the Nengo neural simulator," *Biologically Inspired Cognit. Archit.*, vol. 17, pp. 86–100, Jul. 2016, doi: [10.1016/j.bica.2016.05.001](https://doi.org/10.1016/j.bica.2016.05.001).
- [30] A. Cyr, F. Thériault, and S. Chartier, "Revisiting the XOR problem: A neurobotic implementation," *Neural Comput. Appl.*, vol. 32, no. 14, pp. 9965–9973, Jul. 2020.
- [31] J. C. V. Tieck, T. Schnell, J. Kaiser, F. Mauch, A. Roennau, and R. Dillmann, "Generating pointing motions for a humanoid robot by combining motor primitives," *Frontiers Neurobot.*, vol. 13, p. 77, Sep. 2019.
- [32] C. Lee, P. Panda, G. Srinivasan, and K. Roy, "Training deep spiking convolutional neural networks with STDP-based unsupervised pre-training followed by supervised fine-tuning," *Frontiers Neurosci.*, vol. 12, p. 435, Aug. 2018.
- [33] J. H. Lee, T. Delbruck, and M. Pfeiffer, "Training deep spiking neural networks using backpropagation," *Frontiers Neurosci.*, vol. 10, p. 508, Nov. 2016.
- [34] S. Bohté, J. Kok, and H. L. Poutre, "SpikeProp: Backpropagation for networks of spiking neurons," in *Proc. ESANN*, 2000, pp. 419–424.
- [35] R. Bogacz, M. W. Brown, and C. Giraud-Carrier, "Frequency-based error backpropagation in a cortical network," in *Proc. IEEE-INNS-ENNS Int. Joint Conf. Neural Networks. IJCNN. Neural Comput., New Challenges Perspect. New Millennium*, Jul. 2000, pp. 211–216.
- [36] F. Ponulak, "Analysis of the ReSuMe learning process for spiking neural networks," *Int. J. Appl. Math. Comput. Sci.*, vol. 18, no. 2, pp. 117–127, 2008.
- [37] W. Potjans, A. Morrison, and M. Diesmann, "A spiking neural network model of an actor-critic learning agent," *Neural Comput.*, vol. 21, no. 2, pp. 301–339, Feb. 2009.
- [38] R. V. Florian, "Reinforcement learning through modulation of spike-timing-dependent synaptic plasticity," *Neural Comput.*, vol. 19, no. 6, pp. 1468–1502, 2007.
- [39] V. Lyashenko. (2020). *Basic Guide to Spiking Neural Networks for Deep Learning*[Cnvr.io]. [Online]. Available: <https://cnvr.io/spiking-neural-networks/>
- [40] A. Zhang, H. Zhou, X. Li, and W. Zhu, "Fast and robust learning in spiking feed-forward neural networks based on intrinsic plasticity mechanism," *Neurocomputing*, vol. 365, pp. 102–112, Nov. 2019.
- [41] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. V. D. Driesche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.
- [42] J. D. Schaffer, "Evolving spiking neural networks for robot sensory-motor decision tasks of varying difficulty," in *Proc. Neuro-Inspired Comput. Elements Workshop*, Mar. 2020, pp. 1–7.
- [43] A. Parvizi-Fard, M. Amiri, D. Kumar, M. M. Iskarous, and N. V. Thakor, "A functional spiking neuronal network for tactile sensing pathway to process edge orientation," *Sci. Rep.*, vol. 11, no. 1, pp. 1–16, Dec. 2021.
- [44] W. A. Kaminski and G. M. Wojcik, "Liquid state machine built of Hodgkin–Huxley neurons," *Informatica*, vol. 15, no. 1, pp. 39–44, Jan. 2004.
- [45] A. Amberkar, P. Awasarmol, G. Deshmukh, and P. Dave, "Speech recognition using recurrent neural networks," in *Proc. Int. Conf. Current Trends towards Converging Technol. (ICCTCT)*, Mar. 2018, pp. 1–4.
- [46] E. Rueckert, D. Kappel, D. Tanneberg, D. Pecevski, and J. Peters, "Recurrent spiking networks solve planning tasks," *Sci. Rep.*, vol. 6, no. 1, pp. 1–10, Aug. 2016.
- [47] J. Schemmel, D. Brüderle, A. Gribl, M. Hock, K. Meier, and S. Millner, "A wafer-scale neuromorphic hardware system for large-scale neural modeling," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2010, pp. 1947–1950.
- [48] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *J. Physiol.*, vol. 117, no. 4, p. 500, Aug. 1952.
- [49] R. Siciliano, "The Hodgkin–Huxley model its extensions, analysis and numerics contents," Univ. Autónoma de Bucaramanga-Colombia, 2012.
- [50] G. Cauwenberghs, M. S. Kyung, E. Weiss, and V. Rangan, "A VLSI implementation: Izhikevich's neuron model," 2009. [Online]. Available: <https://www.semanticscholar.org/paper/A-VLSI-Implementation-%3A-Izhikevich-%27-s-Neuron-Model-Cauwenberghs-Kyung/b17849a4ea2f6742a0ab438c7ca51254833bd725>
- [51] S. Dutta, V. Kumar, A. Shukla, N. R. Mohapatra, and U. Ganguly, "Leaky integrate and fire neuron by charge-discharge dynamics in floating-body MOSFET," *Sci. Rep.*, vol. 7, no. 1, Dec. 2017.
- [52] D. Hebb, *The Organization of Behavior: A Neuropsychological Theory*. Psychology Press, Apr. 2005.
- [53] A. M. Andrew, "Spiking neuron models: Single neurons, populations, plasticity," *Kybernetes*, vol. 32, nos. 7–8, pp. 351–386, Oct. 2003.
- [54] C. Lee, S. S. Sarwar, P. Panda, G. Srinivasan, and K. Roy, "Enabling spike-based backpropagation for training deep neural network architectures," *Frontiers Neurosci.*, vol. 14, p. 119, Feb. 2020.
- [55] Y. Yan, H. Chu, Y. Jin, Y. Huan, Z. Zou, and L. Zheng, "Backpropagation with sparsity regularization for spiking neural network learning," *Frontiers Neurosci.*, vol. 16, p. 340, Apr. 2022.
- [56] P. Dayan and L. Abbott, "Computational and mathematical modeling of neural systems," in *Theoretical Neuroscience*. 2001.
- [57] B. Rueckauer, I.-A. Lungu, Y. Hu, and M. Pfeiffer, "Theory and tools for the conversion of analog to spiking convolutional neural networks," Dec. 2016, *arXiv:1612.04052*.
- [58] B. Rueckauer, I.-A. Lungu, Y. Hu, M. Pfeiffer, and S.-C. Liu, "Conversion of continuous-valued deep networks to efficient event-driven networks for image classification," *Frontiers Neurosci.*, vol. 11, p. 682, Dec. 2017.
- [59] J. López-Randulfe, T. Duswald, Z. Bing, and A. Knoll, "Spiking neural network for Fourier transform and object detection for automotive radar," *Frontiers Neurobot.*, vol. 15, p. 69, Jun. 2021.
- [60] W. Gerstner, R. Ritz, and J. L. van Hemmen, "Why spikes? Hebbian learning and retrieval of time-resolved excitation patterns," *Biol. Cybern.*, vol. 69, nos. 5–6, pp. 503–515, 1993.
- [61] R. Kempter, W. Gerstner, and J. L. Van Hemmen, "Hebbian learning and spiking neurons," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 59, no. 4, pp. 4498–4514, Apr. 1999.
- [62] T. Toyozumi, J. P. Pfister, K. Aihara, and W. Gerstner, "Generalized Bienenstock–Cooper–Munro rule for spiking neurons that maximizes information transmission," *Proc. Nat. Acad. Sci. USA*, vol. 102, pp. 5239–5244, Apr. 2005.
- [63] E. M. Izhikevich and N. S. Desai, "Relating STDP to BCM," *Neural Comput.*, vol. 15, no. 7, pp. 1511–1523, Jul. 2003. [Online]. Available: <https://www.nsi.edu/users/izhikevich>
- [64] M. Ouzzani, H. Hammady, Z. Fedorowicz, and A. Elmagarmid, "Rayyan—A web and mobile app for systematic reviews," *Systematic Rev.*, vol. 5, no. 1, pp. 1–10, Dec. 2016.
- [65] A. S. Cassidy, P. Merolla, J. V. Arthur, S. K. Esser, B. Jackson, R. Alvarez-Icaza, P. Datta, J. Sawada, T. M. Wong, V. Feldman, A. Amir, D. B.-D. Rubin, F. Akopyan, E. McQuinn, W. P. Risk, and D. S. Modha, "Cognitive computing building block: A versatile and efficient digital neuron model for neurosynaptic cores," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Aug. 2013, pp. 1–10.
- [66] A. Necker, S. Fok, B. V. Benjamin, T. C. Stewart, N. N. Oza, A. R. Voelker, C. Eliasmith, R. Manohar, and K. Boahen, "Braindrop: A mixed-signal neuromorphic architecture with a dynamical systems-based programming model," *Proc. IEEE*, vol. 107, no. 1, pp. 144–164, Jan. 2019.
- [67] N. Srinivasa and J. M. Cruz-Albrecht, "Neuromorphic adaptive plastic scalable electronics: Analog learning systems," *IEEE Pulse*, vol. 3, no. 1, pp. 51–56, Jan. 2012.
- [68] A. Saenz. (2009). *Facets: Making Computers Work Like Brains*. [Online]. Available: <https://singularityhub.com/2009/04/13/facets-making-computers-work-like-brains/>
- [69] NeuroNem. (2009). *General Vision Home*. [Online]. Available: <https://www.general-vision.com/>
- [70] J. Kim, "New neuromorphic AI NM500 and its ADAS application," in *Recent Advances in Electrical Engineering and Related Sciences: Theory and Application (Lecture Notes in Electrical Engineering)*, vol. 554. 2020, pp. 3–12.

- [71] (2017). *Synsense*. [Online]. Available: <https://www.synsense-neuromorphic.com/>
- [72] J. M. Bower, D. Beeman, M. Hucka, B. Bower, H. Genesis, and D. Beeman. (2003). *The GENESIS Simulation System*. [Online]. Available: <https://www.researchgate.net/publication/222712683>
- [73] SpikeFun. (2012). *Spikefun V0.90—More Complex Pyramidal Neurons, Demo With 3.52 Billion Synapses*. [Online]. Available: <http://www.digicortex.net/node/31>
- [74] B. Coltin and S. Liemhetcharat, “An introduction to robotics with NAO: A stem integrated, project based approach to learning robotics and computer science,” *Aldebaran Robot.*, 2015. [Online]. Available: https://www.academia.edu/32496373/AN_INTRODUCTION_TO_ROBOTICS_WITH_NAO_A_STEM_INTEGRATED_PROJECT_BASED_APPROACH_TO_LEARNING_ROBOTICS_AND_COMPUTER_SCIENCE_Aligned_to_the_Common_Core_State_Standards_Initiative
- [75] G. Varoquaux, P. Ramachandran, and P. R. Mayavi, “Mayavi: Making 3D data visualization reusable,” in *Proc. 7th Python Sci. Conf. (SciPy)*, 2008, pp. 1–6. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-00502548>
- [76] F. Wang and S. K. Gupta, “CSim: A MOS switch-level simulator,” 2016.
- [77] E. Painkras, L. A. Plana, J. Garside, S. Temple, S. Davidson, J. Pepper, D. Clark, C. Patterson, and S. Furber, “SpiNNaker: A multi-core system-on-chip for massively-parallel neural net simulation,” in *Proc. Custom Integr. Circuits Conf.*, Sep. 2012, pp. 1–4.
- [78] M. Davies, N. Srinivasa, T. H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, and Y. Liao, “Loihi: A neuromorphic manycore processor with on-chip learning,” *IEEE Micro*, vol. 38, no. 1, pp. 82–99, Jan. 2018. [Online]. Available: <https://www.computer.org/micro>
- [79] D. F. M. Goodman and R. Brette, “The brian simulator,” *Frontiers Neurosci.*, vol. 3, pp. 192–197, 2009.
- [80] M.-O. Gewaltig and M. Diesmann, “NEST (neural simulation tool),” *Scholarpedia*, vol. 2, no. 4, p. 1430, 2007.
- [81] A. Cyr, M. Boukadoum, and P. Poirier, “AI-SIMCOG: A simulator for spiking neurons and multiple animats’ behaviours,” *Neural Comput. Appl.*, vol. 18, no. 5, pp. 431–446, Jun. 2009.
- [82] D. S. Hien, N. T. Luong, L. H. Minh, T. T. Phuc, P. T. Trung, B. A. Dong, H. L. T. Thao, N. V. L. Thanh, T. T. A. Tuan, H. H. Trung, N. T. T. Nhan, and D. V. Nga, “Development of quantum device simulator NEMO-VN1,” *J. Phys., Conf.*, vol. 187, Sep. 2009, Art. no. 012088.
- [83] T. Carnevale, “Neuron simulation environment,” *Scholarpedia*, vol. 2, no. 6, p. 1378, 2007.
- [84] K. Takaya, T. Asai, V. Kroumov, and F. Smarandache, “Simulation environment for mobile robots testing using ROS and gazebo,” in *Proc. 20th Int. Conf. Syst. Theory, Control Comput. (ICSTCC)*, Oct. 2016, pp. 96–101.
- [85] K. Kumarasinghe, M. Owen, D. Taylor, N. Kasabov, and C. Kit, “FaNeuRobot: A framework for robot and prosthetics control using the neucube spiking neural network architecture and finite automata theory,” in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers, May 2018, pp. 4465–4472.
- [86] N. G. Tsagarakis, G. Metta, G. Sandini, D. Vernon, R. Beira, F. Becchi, L. Righetti, J. Santos-Victor, A. J. Ijspeert, M. C. Carrozza, and D. G. Caldwell, “iCub: The design and realization of an open humanoid platform for cognitive and neuroscience research,” *Adv. Robot.*, vol. 21, no. 10, pp. 1151–1175, Jan. 2007.
- [87] (2007). *Pioneer 3-DX*. [Online]. Available: <https://www.mobilerobots.com>
- [88] A. Hermann, J. Sun, Z. Xue, S. W. Ruehl, J. Oberlaender, A. Roennau, J. M. Zoellner, and R. Dillmann, “Hardware and software architecture of the bimanual mobile manipulation robot HoLLiE and its actuated upper body,” in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics*, Jul. 2013, pp. 286–292.
- [89] M. Marek and Marekz. (2015). *Design, Construction and Control of Hexapod Walking Robot*. [Online]. Available: <http://excel.fit.vutbr.cz>
- [90] UR3. (2020). *User Manual UR3/CB3 Original Instructions*. [Online]. Available: http://www.universal-robots.com/media/8704/ur5_user_manual_gb.pdf
- [91] S. B. R. A. Report and R. Canosa, “Obstacle detection and avoidance using TurtleBot platform and Xbox Kinect,” Dept. Comput. Sci., Rochester Inst. Technol., 2011.
- [92] J. M. Soares, I. Navarro, and A. Martinoli, “The Khepera IV mobile robot: Performance evaluation, sensory data, and software toolbox,” in *Proc. Robot, 2nd Iberian Robot. Conf.*, 2015, pp. 767–781.
- [93] Schunk SVH, “Translation of original operating manual assembly and operating manual SVH servo-electric 5-finger gripping hand,” Program. Translating Service GMBH, Germany, 2016.
- [94] G. Capi, “Robot task switching in complex environments,” in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics*, Sep. 2007, pp. 1–6.
- [95] D. H. García, S. Adams, A. Rast, T. Wennekers, S. Furber, and A. Cangelosi, “Visual attention and object naming in humanoid robots using a bio-inspired spiking neural network,” *Robot. Auto. Syst.*, vol. 104, pp. 56–71, Jun. 2018.
- [96] M. K. Al-Qaderi and A. B. Rad, “A brain-inspired multi-modal perceptual system for social robots: An experimental realization,” *IEEE Access*, vol. 6, pp. 35402–35424, 2018.
- [97] B. S. Bhattacharya and T. Serrano-Gotarredona, “On- and off-centre pathways in a retino-geniculate spiking neural network on SpiNNaker,” in *Proc. 10th Int. IEEE/EMBS Conf. Neural Eng. (NER)*. Washington, DC, USA: IEEE Computer Society, May 2021, pp. 461–464.
- [98] R. Tanaka, J. Woo, and N. Kubota, “Action acquisition method for constructing cognitive development system through instructed learning,” in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2019, pp. 1–6.
- [99] A. Cyr and F. Theriault, “Spatial concept learning: A spiking neural network implementation in virtual and physical robots,” *Comput. Intell. Neurosci.*, vol. 2019, pp. 1–9, Apr. 2019.
- [100] F. Naveros, N. R. Luque, E. Ros, and A. Arleo, “VOR adaptation on a humanoid iCub robot using a spiking cerebellar model,” *IEEE Trans. Cybern.*, vol. 50, no. 11, pp. 4744–4757, Nov. 2020.
- [101] A. Lele, Y. Fang, J. Ting, and A. Raychowdhury, “An end-to-end spiking neural network platform for edge robotics: From event-cameras to central pattern generation,” *IEEE Trans. Cognit. Develop. Syst.*, vol. 14, no. 3, pp. 1092–1103, Sep. 2021.
- [102] S. A. Lobov, A. I. Zharinov, V. A. Makarov, and V. B. Kazantsev, “Spatial memory in a spiking neural network with robot embodiment,” *Sensors*, vol. 21, no. 8, p. 2678, Apr. 2021.
- [103] A. I. Zharinov, V. A. Makarov, V. B. Kazantsev, and S. A. Lobov, “Spatial memory based on an STDP-driven neural network,” in *Proc. 4th Sci. School Dyn. Complex Netw. Appl. Intellectual Robot. (DCNAIR)*, Sep. 2020, pp. 269–271.
- [104] O. Zahra, D. Navarro-Alarcon, and S. Tolu, “A neurobotic embodiment for exploring the dynamical interactions of a spiking cerebellar model and a robot arm during vision-based manipulation tasks,” 2021, *arXiv:2102.01966*.
- [105] N. Berberian, M. Ross, and S. Chartier, “Embodied working memory during ongoing input streams,” *PLoS ONE*, vol. 16, no. 1, Jan. 2021, Art. no. e0244822.
- [106] T. Obo, R. Hase, K. Kobayashi, K. Sueta, T. Nakano, and D. Shin, “Cognitive modeling based on perceiving-acting cycle in robotic avatar system for disabled patients,” in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020.
- [107] J. C. V. Tieck, K. Secker, J. Kaiser, A. Roennau, and R. Dillmann, “Soft-grasping with an anthropomorphic robotic hand using spiking neurons,” *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 2894–2901, Apr. 2021.
- [108] J. C. V. Tieck, S. Weber, T. C. Stewart, J. Kaiser, A. Roennau, and R. Dillmann, “A spiking network classifies human sEMG signals and triggers finger reflexes on a robotic hand,” *Robot. Auto. Syst.*, vol. 131, Sep. 2020, Art. no. 103566.
- [109] J. C. V. Tieck, S. Weber, T. C. Stewart, A. Roennau, and R. Dillmann, “Triggering robot hand reflexes with human EMG data using spiking neurons,” *Adv. Intell. Syst. Comput.*, vol. 867, pp. 902–916, Jun. 2018.
- [110] H. Tang, R. Yan, and K. C. Tan, “Cognitive navigation by neuro-inspired localization, mapping, and episodic memory,” *IEEE Trans. Cognit. Develop. Syst.*, vol. 10, no. 3, pp. 751–761, Sep. 2018.
- [111] M. Hampo, D. Fan, T. Jenkins, A. Demange, S. Westberg, T. Bihl, and T. Taha, “Associative memory in spiking neural network form implemented on neuromorphic hardware,” in *Proc. Int. Conf. Neuromorphic Syst.*, Jul. 2020, pp. 1–8.
- [112] E. Dumesnil, P.-O. Beaulieu, and M. Boukadoum, “Single SNN architecture for classical and operant conditioning using reinforcement learning,” *Int. J. Cognit. Informat. Natural Intell.*, vol. 11, no. 2, pp. 1–24, Apr. 2017.
- [113] G. Tang and K. P. Michmizos, “Real-time mapping on a neuromorphic processor,” in *Proc. Neuro-Inspired Comput. Elements Workshop*, Mar. 2020, pp. 1–3.

- [114] X. Huang, W. Wu, and H. Qiao, "Computational modeling of emotion-motivated decisions for continuous control of mobile robots," *IEEE Trans. Cognit. Develop. Syst.*, vol. 13, no. 1, pp. 31–44, Mar. 2021.
- [115] A. Cyr, J. Morand-Ferron, and F. Thériault, "Dual exploration strategies using artificial spiking neural networks in a robotic learning task," *Adapt. Behav.*, vol. 29, no. 6, pp. 567–578, Dec. 2021.
- [116] A. Cyr, A. Avarguès-Weber, and F. Thériault, "Sameness/difference spiking neural circuit as a relational concept precursor model: A bio-inspired robotic implementation," *Biologically Inspired Cognit. Archit.*, vol. 21, pp. 59–66, Jul. 2017.
- [117] T. Zhou and J. P. Wachs, "Spiking neural networks for early prediction in human–robot collaboration," *Int. J. Robot. Res.*, vol. 38, pp. 1619–1643, Dec. 2019.
- [118] E. Mansouri-Benssassi and J. Ye, "Bio-inspired spiking neural networks for facial expression recognition: Generalisation investigation," in *Theory and Practice of Natural Computing (Lecture Notes in Computer Science)*, vol. 11324. 2018, pp. 426–437.
- [119] K. D. Fischl, A. B. Cellon, T. C. Stewart, T. K. Horiuchi, and A. G. Andreou, "Socio-emotional robot with distributed multi-platform neuromorphic processing: (Invited presentation)," in *Proc. 53rd Annu. Conf. Inf. Sci. Syst. (CISS)*, Mar. 2019, pp. 1–6.
- [120] S. Devarajan, *Object Identification For Robotic Applications Using Expert Systems*, vol. 7. Katowice, Poland: Uniwersytet slaski, 1996, pp. 343–354. [Online]. Available: <https://digitalscholarship.unlv.edu/rtds/3190>
- [121] S. Furber and P. Bogdan, *SpiNNaker: A Spiking Neural Network Architecture*. Hanover, MA, USA: Now, 2020.
- [122] G. Metta, P. Fitzpatrick, and L. Natale, "YARP: Yet another robot platform," *Int. J. Adv. Robotic Syst.*, vol. 3, pp. 43–48, Mar. 2006.
- [123] University of Manchester. (2015). *Lab Manual Spinnaker Interfacing External Devices*. [Online]. Available: <http://spinnakermanchester.github.io>
- [124] A. R. Damasio, "The brain binds entities and events by multiregional activation from convergence zones," *Neural Comput.*, vol. 1, no. 1, pp. 123–132, Mar. 1989.
- [125] A. Treisman, "The binding problem," *Current Opinion Neurobiol.*, vol. 6, pp. 171–178, Apr. 1996.
- [126] W. Li, V. Piëch, and C. D. Gilbert, "Perceptual learning and top-down influences in primary visual cortex," *Nature Neurosci.*, vol. 7, no. 6, pp. 651–657, Jun. 2004. [Online]. Available: <https://www.nature.com/articles/nn1255>
- [127] R. Chrisley, V. C. Müller, Y. Sandamirskaya, and M. Vincze, "Cognitive robot architectures," in *Proc. Cognit. Robot Archit. CEUR WS*, vol. 1855, 2016, p. 1. [Online]. Available: <https://www.eucognition.org>
- [128] A. Borthakur and T. A. Cleland, "A spike time-dependent online learning algorithm derived from biological olfaction," *Frontiers Neurosci.*, vol. 13, p. 656, Jun. 2019.
- [129] F. Mirus, C. Axenie, T. C. Stewart, and J. Conrad, "Neuromorphic sensorimotor adaptation for robotic mobile manipulation: From sensing to behaviour," *Cognit. Syst. Res.*, vol. 50, pp. 52–66, Aug. 2018.
- [130] E. Bayro-Corrochano, S. Solis-Gamboá, G. Altamirano-Escobedo, L. Lechuga-Gutierrez, and J. Lisarraga-Rodríguez, "Quaternion spiking and quaternion quantum neural networks: Theory and applications," *Int. J. Neural Syst.*, vol. 31, no. 2, Feb. 2021, Art. no. 2050059.
- [131] M. Pitchai, X. Xiong, M. Thor, P. Billeschou, P. L. Mailänder, B. Leung, T. Kulvicius, and P. Manoonpong, "CPG driven RBF network control with reinforcement learning for gait optimization of a dung beetle-like robot," in *Artificial Neural Networks and Machine Learning—ICANN: Theoretical Neural Computation (Lecture Notes in Computer Science Including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11727. 2019, pp. 698–710.
- [132] L. Steffen, A. Liebert, S. Ulbrich, A. Roennau, and R. Dillmann, "Adaptive, neural robot control—Path planning on 3D spiking neural networks," in *Artificial Neural Networks and Machine Learning—ICANN (Lecture Notes in Computer Science including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12397. 2020, pp. 509–520.
- [133] S. Jacob, J. Kwisthout, and S. Thill, "From models of cognition to robot control and back using spiking neural networks," in *Biomimetic and Biohybrid Systems (Lecture Notes in Computer Science including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12413. 2021, pp. 176–191.
- [134] E. Dumesnil, P.-O. Beaulieu, and M. Boukadoum, "Robotic implementation of classical and operant conditioning within a single SNN architecture," in *Proc. IEEE 15th Int. Conf. Cognit. Informat. Cognit. Comput. (ICCI CC)*, Aug. 2016, pp. 322–330.
- [135] K. Doya and T. Taniguchi, "Toward evolutionary and developmental intelligence," *Current Opinion Behav. Sci.*, vol. 29, pp. 91–96, Oct. 2019.
- [136] J. C. V. Tieck, J. Kaiser, L. Steffen, M. Schulze, A. Von Arnim, D. Reichard, A. Roennau, and R. Dillmann, "The neurorobotics platform for teaching—Embodiment experiments with spiking neural networks and virtual robots," in *Proc. IEEE Int. Conf. Cyborg Bionic Syst. (CBS)*, Sep. 2019, pp. 291–298.
- [137] S. Toolbox. (2018). *Spiking Neural Network Conversion Toolbox—SNN Toolbox 0.5.0 Documentation*. [Online]. Available: <https://snntoolbox.readthedocs.io/en/latest/>
- [138] D. Rasmussen, "NengoDL: Combining deep learning and neuromorphic modelling methods," 2018, *arXiv:1805.11144*.
- [139] L. Deng, Y. Wu, X. Hu, L. Liang, Y. Ding, G. Li, G. Zhao, P. Li, and Y. Xie, "Rethinking the performance comparison between SNNs and ANNs," *Neural Netw.*, vol. 121, pp. 294–307, Jan. 2020.



MUHAMMAD AITSAM received the bachelor's degree in electrical engineering with a specialization in electronics from UET Taxila, Pakistan. After that, he got DAAD Master's Scholarship in the field of communication engineering. During master's degree, he got chance to conduct research on emerging technologies like artificial intelligence, machine learning, and data privacy. He recently joined Sheffield Hallam University for the Marie Curie Project Personalized Robotics as Service Oriented Applications (PERSEO). His research interests include machine learning, personalized robotics, and human–robot interaction.



SERGIO DAVIES (Member, IEEE) received the Ph.D. degree in computer science from The University of Manchester, Manchester, U.K. He was a Research Associate with the APT Group, School of Computer Science, The University of Manchester. He is currently a Senior Lecturer with the Department of Computing, Sheffield Hallam University, teaching fundamentals of computing, embedded architectures, and parallel computing. His research interest includes spiking neural networks, and participated in the SpiNNaker Project and the Human–Brain Project (HBP).



ALESSANDRO DI NUOVO (Senior Member, IEEE) is currently a Professor of machine intelligence at Sheffield Hallam University. His research work at SHU is currently focused on computational intelligence for the design, development, and experimental validation of computational models of human cognition in robots (autonomous motor control via mental imagery, finger counting, and number cognition), including their application to human–robot interaction in healthcare and social environments (autism therapy and diagnosis, elderly care, robot companions, and co-workers).

...