Biointerface Research in Applied Chemistry
Platinum Open Access Journal (ISSN: 2069-5837)

*Article*
**Volume 12, Issue 5, 2022, 6618 - 6631**
https://doi.org/10.33263/BRIAC125.66186631

# Spatio-Temporal Pattern Representation from AI Inspired Brain Model in Spiking Neural Network

**Narinder Pal Singh [1], Partha Sarathy [1], Archana Mantri [3], Gurjinder Singh [3], Debarshi Ghosh [3], Thakur Gurjeet Singh [4,*], Nitin Kumar Saluja [3], Rashpinder Kaur [3]**

[1] Chitkara School of Art & Design, Chitkara University, Punjab, India; narinder.singh@chitkara.edu.in (N.P.S.); parthasarathy@chitkara.edu.in (P.S.);

[3] Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India; archana.mantri@chitkara.edu.in (A.M.); gurjinder.singh@chitkara.edu.in (G.S.); debarshi.ghosh@chitkara.edu.in (D.G.); nitin.saluja@chitkara.edu.in (N.K.S.); rashpinder.kaur@chitkara.edu.in (R.K.);

[4] Chitkara College of Pharmacy, Chitkara University, Punjab, 140401, India; gurjeet.singh@chitkara.edu.in (T.G.S.);

[*] Correspondence: gurjeet.singh@chitkara.edu.in; gurjeetthakur@gmail.com (T.G.S.);

Scopus Author ID 57226012795

**Received: 26.05.2021; Revised: 20.09.2021; Accepted: 24.09.2021; Published: 19.11.2021**

**Abstract:** Neuronal population activity in the brain is the combined response of information in the spatial domain and dynamics in the temporal domain. Modeling such Spatio-temporal mechanisms is a complex process because of the complexity of the brain and the limitations of the hardware. In this paper, we demonstrate how information processing principles adapted from the brain can be used to create a brain-inspired artificial intelligence (AI) model and represent Spatio-temporal patterns. The same is demonstrated by designing the tiny brain using spiking neural networks, where activated neuronal populations represent information in the spatial domain and transmitting signals represent dynamics in the temporal domain. Spatially located sensory neurons excited by input visual stimuli further activate motor neurons to trigger a motor response that causes behavior modification of the robotic agent. Initially, an isolated brain network is simulated to understand the excitation part from sensory to motor neurons while plotting waveform between membrane potential and time. The response of the network to stimulate robot body movements is also plotted to demonstrate representation. The simulation shows how the response of particular visual stimuli modifies behavior and helps us understand the body and brain synchronization. The perceived environment and resultant behavior response allow us to study body interaction with the environment.

**Keywords:** brain-inspired AI; spatio-temporal patterns; Spiking Neural Networks; neurorobotics; feature detection; behaviour modification.

## 1. Introduction

The universe evolves and exists within space-time and has complex creations like the human brain [1,2]. The human brain can constantly learn, take actions, sense, interpret results, and modify behavior accordingly. Our brain utilizes deep learning to create or destroy synapses between spatially located neurons by exchanging data means of electrical impulses and representing whole connections as deep knowledge in space-time [3-5]. But it is almost impossible to model human brain dynamics as it contains 86 billion neurons, each with an average of 7,000 synapses. The modern-day computational capabilities, though advanced exponentially, are insufficient to simulate the whole human brain. But we can adopt a simpler approach by simulating and modeling sub-organizational levels of the brain or modeling a simple nervous system like *C. elegans* [6-8] organism to understand complex systems. Instead of accurately creating the brain, we can adopt brain dynamics principles to create a brain-inspired AI [9-12] model to represent complex Spatio-temporal patterns.

Modeling brain-inspired AI can be employed to develop intelligent bio-realistic systems, which can help us cure and investigate the brain and help us understand life more clearly[13,14]. This may also reduce experiments on animals to some extent. With simulation, we can study diseases and how they affect the human body and can understand life more closely.

The paper is structured as follows: Spiking neural networks introduced in Section II. Experimentation detail, tools, and methodology are explained in Section III. It also introduces our approach to develop visual perception, the neuron model, brain modeling, robot and environment setup, simulation, and functional implementation. Section IV presents the simulation results of our approach. Finally, Section V concludes the paper and provides research applications.

## 2. Materials and Methods

### *2.1. Spiking Neural Network.*

Spiking Neural Networks (SNNs)[15–17] are our main building blocks. These are also called the third generation neuron network through which brain-like behaviors can be ascertained, and it has capabilities to encode and decode Spatio-temporal information; it offers a solution for achieving information processing for cognitive psychology. From the bio-realistic perspective, neuronal information is done through electrical signals or spikes, as these networks are more biologically plausible. The information does not flow continuously, but networks operate using time-dependent discrete spike events. These consecutive spikes form Spatio-temporal patterns. SNNs are more energy-efficient than traditional AI networks or machine learning models because of asynchronous spike-based computations [18–22]. Furthermore, SNNs can be trained unsupervised using Spike Timing Dependent Plasticity (STDP) [23,24] for deep learning to build structures for deep knowledge representation.

### *2.2. Experimentational details.*

This paper presented a way to encode vision features as Spatio-temporal patterns and further decode them for robotic agent behavior modification. We developed a tiny brain with a spiking neural network with structural and dynamics properties inspired by bio-realistic brains. In this Spatio-temporal system, spatially located neurons are electrically excitable cells and transmit spikes to each other across the synapses and represent the Spatio-temporal pattern of events while communicating.

In the closed-loop environment, the following steps demonstrate our methodology: Develop connected neurons with synapses to form neuronal populations; Simulate the brain with a set of neuronal populations; Implement vision perception and stimulus capability in a robotic agent to behave like living creatures; Simulate and connect the brain and the robot in a closed-loop environment; Perceive natural image patches and detect features using robot eyes (cameras); Activate the sensory neuronal population and detect features of the detected image patch according to the associated coordinates; Connect with the respective motor neuron to the sensory neuronal population to excite them and stimulate movement electrically.

To perceive the environment, robot cameras are used because digital cameras function like an eye, where the image sensor behaves as a retina. In vivo axons of retinal neurons deliver visual information in the form of spikes to the different brain areas to perform several functions. In this approach, the natural image patch perceived by the robot camera excites the neuronal population. In vivo, the somatic motor system responsible for muscle movements controls skeletal muscles to move bones around joints. A neural signal stimulates muscle fiber by generating an action potential. Similarly, in current experimentation, the simulated brain produces signals to control the robot. Here brain motor neurons voltage is directly linked with robot joints rotation movement. Every input stimulus changes environmental perception for the agent. So, while performing simulation, we can observe robot interactions with its environment and can check how future sensory influences the robot's actions, thus allowing us to study brain-body synchronization. We can investigate the brain's responses to input stimuli. Thus this research bridges robotics and neuroscience, facilitating seamless communication by means of Spatio-temporal patterns between these two disciplines. The experiment was designed so that a robot perceives some images through the camera, responds to visual stimuli, and switches behavior.

We know that brain-inspired AI is a multidisciplinary field encompassing neuroscience, computer science, engineering, and many others. We used the interdisciplinary platform called Neurorobotics Platform (NRP) [25,26] developed for the Human Brain Project (HBP)[27,28]. NRP simulates experiments in closed perception-action loops with Closed Loop Engine (CLE), where Transfer functions (TFs) are used to represent and connect the brain/robot simulation to robot/brain stimulation.

In NRP, all the communications are handled by the Robot Operating System (ROS) [29] topics. ROS is a well-known middleware in the robotics field.

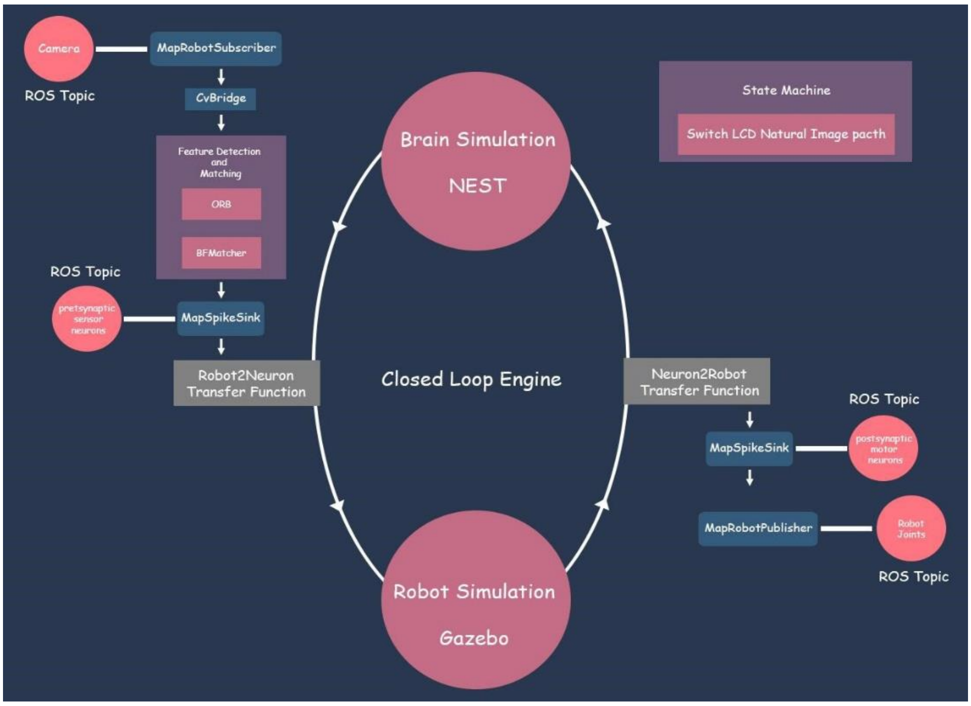


**Figure 1.** Schematic diagram of the brain-body interaction experiment.

It can interface with many physics' libraries, robot simulators, and hardware. ROS node is an executable that is used to transfer data and communicate with other nodes within single or multiple machines. ROS nodes communicate with each other with topics. ROS messages are a data type used to subscribe or publish to a topic. Nodes can publish and subscribe messages to a topic to perform two-way communication. When one publisher node publishes data, the subscribers subscribe to published topics to receive data.

Figure 1 demonstrates body-brain interaction in a closed loop. For establishing realistic experiments in NRP, brain stimulation is coupled with Gazebo [30] for physics simulation and can be used for the physical simulation of robotic agents and the environment. The Gazebo provides a robot designer and accurate simulation. The Gazebo has capabilities for algorithm testing and training AI agents in realistic scenarios.

The simulation of the robot population in complex environments is also possible. Gazebo bundled with excellent 3d UI navigation, physics engine like Bullet, high-quality 3d graphics. Gazebo communicates with simulated robots through the ROS with asynchronous event-based communication through Topics.

### *2.3. Visual perception.*

Image features are mathematical representations of key areas. Visual perception systems need to understand the important aspects of an image. Eventually, these are the vector representations of the visual content, which allow us to perform mathematical operations on them. Features play a vital role in many computer vision problems like motion estimation, object detection, segmentation, image alignment, and a lot more. The selection of an efficient feature detector algorithm is very important. There are several doctor algorithms presented, like Scale Invariant Feature Transform (SIFT)[31], Speed up Robust Feature (SURF)[32], Binary Robust Independent Elementary Features (BRIEF)[33], and Oriented FAST and Rotated BRIEF (ORB) [34]. But ORB is the open-source, fastest, good matching performance, and efficient choice over SIFT and SURF [35]. For matching Brute-Force Matcher in the OpenCV library, It is a very simple matcher that does distance calculations between descriptor one feature from the input image with all source image features.

### *2.4. Neuron model.*

A vivo neuron cell is composed of 3 functional units, those are dendrites, soma, and axon. The soma is a cell body and is the spherical part of the neuron. It is a central processing unit that performs an important nonlinear processing step. While dendrites is an ‘input device and axon is an output device. Suppose a signal coming through dendrites exceeds a certain threshold, soma spikes. That spike delivers to other neurons with an axon where synapse is a neuronal junction, that transmission of electric impulses between two nerve cells that connect the axon of one neuron to another neuron. Suppose a neuron sends a signal across a synapse that is received by connected neurons. In that case, the transmitting neuron is called a presynaptic neuron, and the receiving neuron is the postsynaptic neuron [36].

Potential difference u(t) of postsynaptic neurons can be recorded with an intracellular electrode. That measures the difference between the inner part of the neuron and the outer part. This potential difference is called the membrane potential. If there is no input spike, then the neuron is at rest, called constant membrane potential urest. Received spike change potential of the neuron. Suppose membrane potential reaches its threshold value, neuron fire and decays back to urest. The positive or negative change in the potential declares excitatory or inhibitory synapse.

Current computational resources are capable of simulating the Hodgkin–Huxley neuron model [37,38] dynamic behavior having various ion channels of the soma, dendrites, and hundreds of spatial compartments and can reproduce the neuron having all features like all ionic channels, synapses. But in this way, we may lose mathematics control. On the other side, over-simplified modeling may lose biological realism.

So for spiking neuron network modeling, we selected the integrate-and-fire (IAF) neuron model [39–43]. Because of its conceptual simplicity, the IAF model can be used to explore some general principles of neurodynamics, where it can either be stimulated by a spike input of synaptic input from presynaptic neurons or external current.

A schematic diagram of the Integrate-and-fire model is presented in Figure 2. This diagram shows a network with three neurons, one presynaptic neuron s0 and two postsynaptic neurons m0 and m1. The soma circuit of neurons comprises a Resistor Capacitor pair driven by the current I(t). As both components are connected in a parallel manner, the RC circuit charged by I(t) can be written as I(t) = IR + IC, here IC is responsible to charges the capacitor C and IR is the divided current that passes through the resistor R. According to Ohm's law, the amount of current passing through two points is directly proportional to the voltage across two points and inversely proportional to the resistance of the circuit. Suppose u is the voltage across the resistor, then IR = u/R. On the other hand, the capacitance of a capacitor is defined as the charge stored per unit potential difference change, If q is the charge from capacitance definition C = q/u and Capacitive current IC = C (du/dt).

Thus, I(t) = IR + IC can be written as:

$$I(t) = \frac{u(t)}{R} + C\frac{du}{dt} \tag{1}$$

By multiplying the equation by R

More generally, we write:

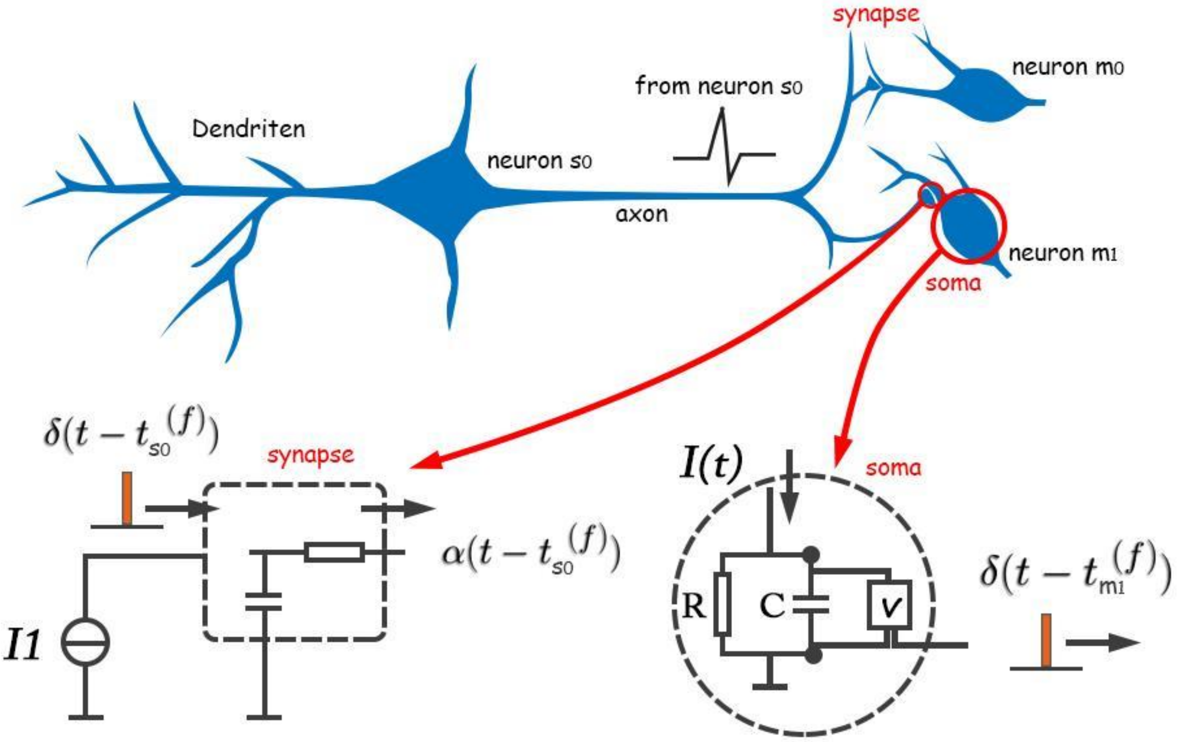


**Figure 2.** Schematic diagram of the integrate-and-fire model having soma and synapse circuit.

$$\tau_m \frac{du}{dt} = -u(t) + RI(t) \tag{2}$$

In this equation, u(t) is the potential difference or membrane potential at time t, $\tau_m$=RC the membrane time constant and R is the membrane resistance. Equation (2) describes how membrane potential behaves as a function of time when current is injected into the neuron.

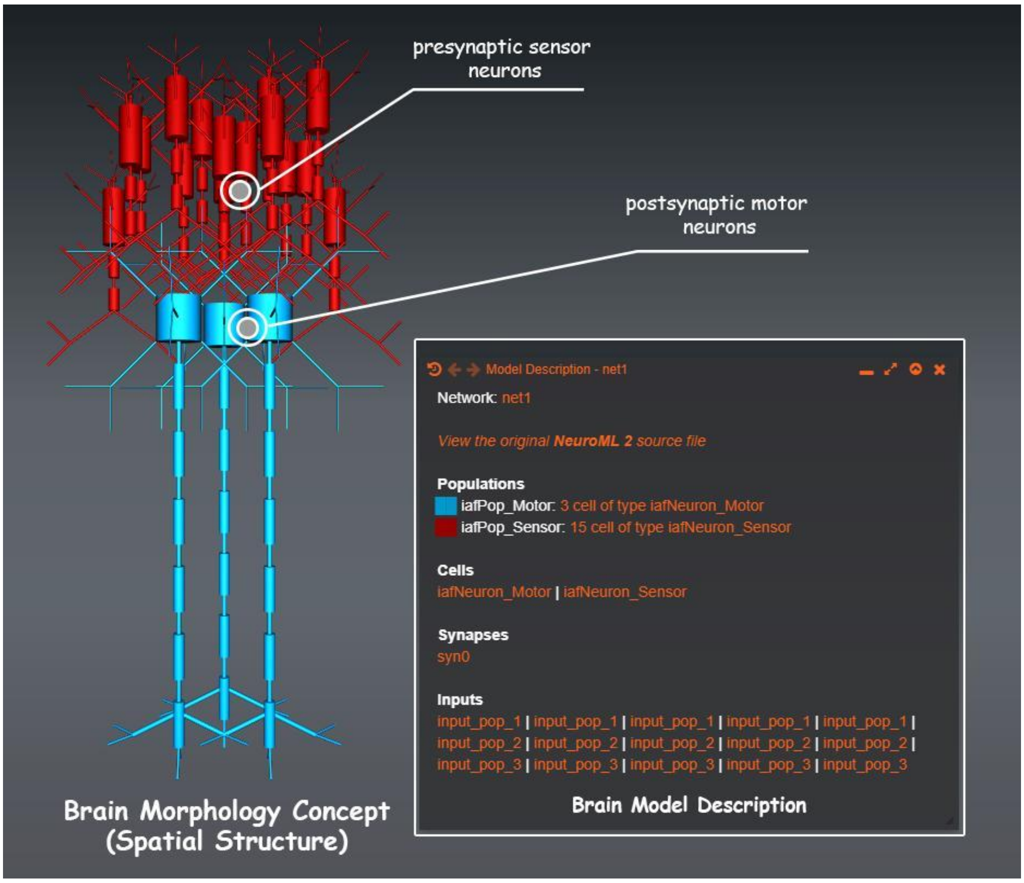
A presynaptic spike $\delta(t - t_{S0}(f))$ from neuron s0 is low-pass filtered at the synapse between presynaptic neuron s0 and post postsynaptic neuron m1 generates an input spike $\alpha\ (t - t_{S0}(f))$.

Once membrane potential u(t) across the capacitance reaches threshold $v$ i.e., u(t) = $v$ at time tm1(f) $\alpha\ (t - t_{m1}(f))$ output spike is recorded. After generating spikes, neurons enter a resting state for a duration $\tau_{rest}$. During the resting period, the membrane potential remains constant a $u_{rest}$.

So, according to equation (2), the IAF model's current state is characterized by membrane potential that receives excitatory or inhibitory spikes by synaptic inputs. Spikes generated with IAF neurons are discrete events in time rather than continuous values.

### *2.5. Brain model.*

The brain model comprises 3 presynaptic sensories and a postsynaptic motor neuronal population, where a set of IAF neurons is bundled in a single neuronal population. Each neuron in the sensory neuronal population is connected with a single motor neuron. There are several tools that exist to develop and simulate a functioning virtual model of a brain circuit using a spiking neural network having different levels of detail. In our case NEST [44] simulator with PyNN[45] (Simulator-independent language for building neuronal network models ) is used because NEST is currently supported inside NRP through the PyNN abstraction layer and runs within a distributed and parallel environment.


presynaptic sensor neurons
postsynaptic motor neurons
Brain Morphology Concept (Spatial Structure)
Model Description - net1
Brain Model Description

Network: net1

*View the original **NeuroML 2** source file*

**Populations**
🟦 iafPop_Motor: 3 cell of type iafNeuron_Motor
🟥 iafPop_Sensor: 15 cell of type iafNeuron_Sensor

**Cells**
iafNeuron_Motor | iafNeuron_Sensor

**Synapses**
syn0

**Inputs**
input_pop_1 | input_pop_1 | input_pop_1 | input_pop_1 | input_pop_1 |
input_pop_2 | input_pop_2 | input_pop_2 | input_pop_2 | input_pop_2 |
input_pop_3 | input_pop_3 | input_pop_3 | input_pop_3 | input_pop_3

**Figure 3.** Initial brain morphology concept designed in Open Source Brain Platform.

For initial brain morphology concept development Figure 3, initial analysis, testing, and simulation, we used the Open Source Brain (OSB) online platform [46]. With OSB we can share and collaboratively develop computational models of neural systems. It has various tools like 3D visualization, analysis, and models simulation directly over the internet browser. PyNN /NEST models are easily converted into NeuroML[47,48] format that is supported in Open Source Brain.

In OSB each neuron is placed in space with a 3d coordinate system as associated 3-dimensional vector keys. We also implemented a pulse generator with NeuroML as an input device connected with each sensor neuron. The pulses typically have a duration between 1ms and 2ms and an amplitude of about 100 mV, and it's also called action potentials or spikes. The action potential is the elementary unit of signal transmission. We don't consider forms of the pulse. Since isolated spikes do not look similar, so the spike's anatomy does not carry any information. But it is considered as a series of events in time. In other words, a time when a particular neuron spike and the total count of the spike within the given window matter. These events occur at regular or irregular intervals—a sequence of pulses emitted by a single neuron from a spike train. In spikes, train action potentials are well separated because even a strong spike can not excite a second spike immediately after or during the first one.

Let us consider 2 presynaptic neurons s1, s2 which transmit signals and excites to the connected postsynaptic neuron m. Suppose neuron s1 spikes continuously at time t1(1), t1(2),...., and neuron s2 fires at t2(1),t2(2),… . Each input spike from presynaptic neurons excites a postsynaptic potential ϵm1, ϵm2, respectively. So equation (3) is the total change in the potential of neuron m is approx the sum of all postsynaptic potentials.

$$u_m(t) = \sum\sum \in_{ms} \left(t - t_s^f\right) + u_{rest} \tag{3}$$

To investigate the total change in the membrane potential of postsynaptic neurons we simulated the network and demonstrated in Figure 4 having sensory neuronal populations Ps1(n0, n1,,n2, n3,,n4), Ps2(n5, n6,,n7, n8,,n9) and Ps3(n10, n11,,n12, n13,,n14) and motor neuronal populations Pm1(n0), Pm2(n1) and Pm3(n2). In this figure, a plot of membrane potential vs. time is demonstrated. Where presynaptic neurons of sensory populations Ps1, Ps2, Ps3 are excited with a pulse generator as an input device, those further excite postsynaptic neurons of motor populations Pm1, Pm2, Pm3. To differentiate excitation activities of different ìpopulations, a delay of 10ms is provided in every pulse generator. Note that this was done only for visualization purposes, so we recorded a spike train at motor neurons after simulation. In the simulation, we tested if we stop pulse generators for populations Ps2 and Ps3, spike sequence from Pm2, and Pm3 also stop. Stopping some neurons in a particular population lets us stop input pulse generators for n7, n8, n9 in population Ps2. The recorded frequency of waveform at the n1 of Pm2 also decreases, meaning a lower firing rate. That demonstrates that spatially located neurons spike activity in time; the firing rate of presynaptic neurons affects postsynaptic neuron activation. After testing and analyzing, the network was redeveloped in NEST.

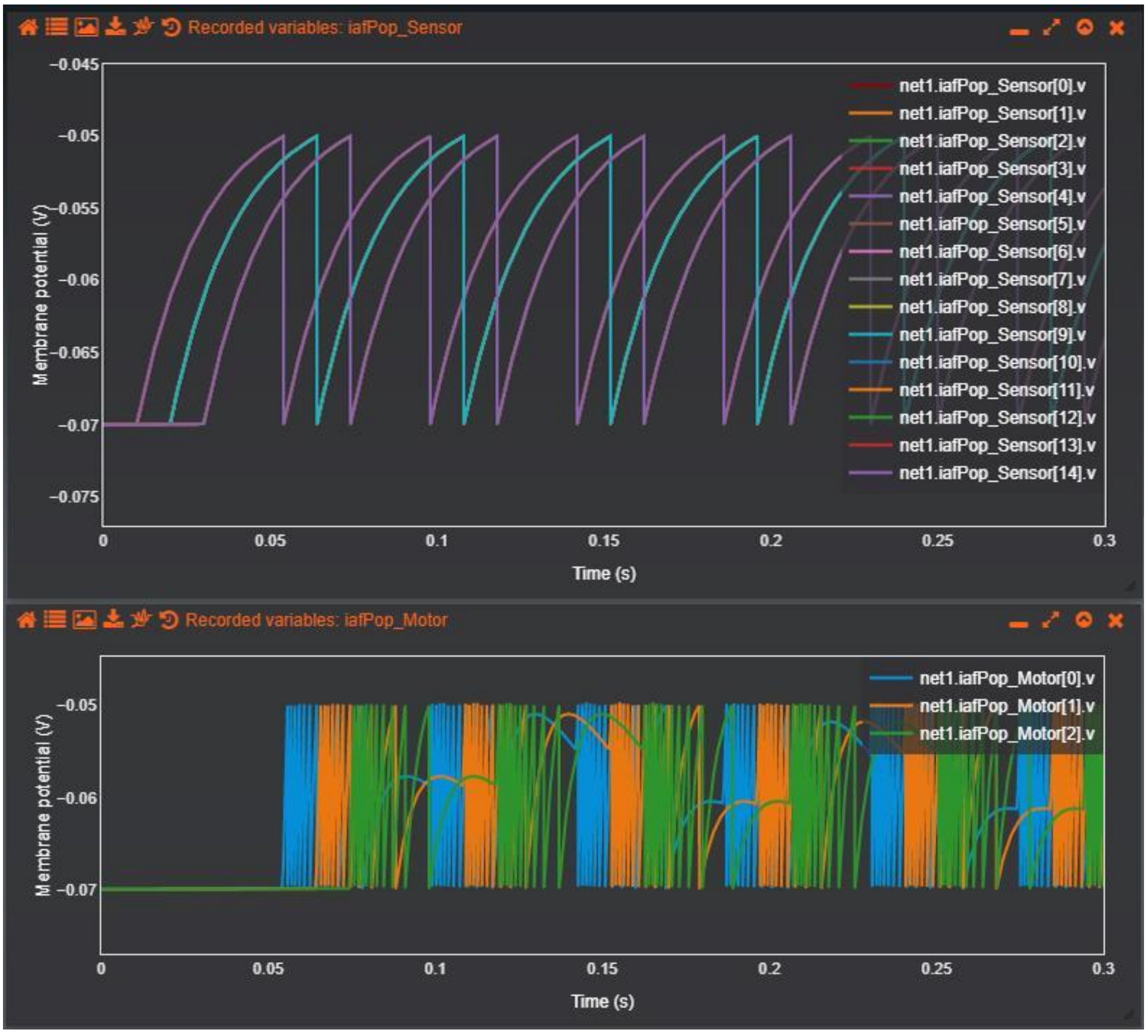


**Figure 4.** Membrane potential vs. Time plot for Sensor and Motor neurons in OSB simulation.

### *2.6. iCub robot.*

For deploying realistic humanoid embodiments in the experiment, we selected an iCub robot [49] is an Open Source robot. The motivation behind its selection is its humanoid design. It provides human-like manipulators that play a vital role in embodied cognition and brain-inspired AI research [50–55]. iCub height is 104 cm, like a child. The upper body of the robot has been allocated 3d degrees of freedom (DOF). Hands have 9 DOF, each with 5 independent fingers. Each leg has 6 DOF for providing bipedal locomotion. From a sensory point of view, It is equipped with two 320X240@60Hz cameras positioned at the eyes, microphones to perceive its current environment, have gyroscopes and accelerometers, and force/torque sensors. Each joint is instrumented with positional sensors, using absolute position encoders. It also has lines of LEDs for representing facial expressions.

We used a gazebo plugin to control each joint and receive images from cameras. Joints are controlled by two PIDs, positions, and velocities. The gazebo plugin publishes the encoder's values to a ROS topic. Other nodes subscribe to published topics to provide translation and rotation movements to the robot.

### *2.7. Environment.*

The experiment environment is performed in a 3-dimensional virtual space that contains a floor, a virtual screen, and some decorative elements. iCub is placed in front of the virtual screen so that cameras attached with robot eyes can continuously take the pictures and publish them on the ROS topic. We use NRP state machines to generate events that influence the environment. State Machine can control and monitor any properties that are being published on ROS topics. Like spiking activity, sensor output, simulation time. So we used it to switch virtual screen Natural image patches after some interval of time.

### *2.8. Functional implementation.*

The functional implementation shows in Figure 5. where iCub cameras are published on the ROS topic, which is mapped with Robot2Neuron TFs through MapRobotSubscriber (parameters functional mapping decorator to the robot). OpenCV can not read images directly from ROS Topic. For that, we used CVBridge. Visual perception python program loops over 3 natural image patches displaying handshake, hand-waving, and kick illustrations. These images are stored in an array and compare their features with the camera input image. Feature extraction implemented with ORB and matching with Brute-Force Matcher. Sensory Populations Ps1, Ps2, Ps3 mapped with features I1(f0, f1,..., f4), I2(f5, f6,..., f9) and I3(f10, f11,..., f14), where I1, I2, I3 are input natural image patches having f0, f1, f2,....., f14 features. When features (f0, f1,..., f4) matches the input image I1, it excites mapped population Ps1 neurons using associated keys. Neurons are stored in an associated array with 3d coordinate keys and mapped with image features using 2d coordinates keys. Poisson generators generate spikes to activate presynaptic neurons according to detected image features for assessing sensory neurons MapSpikeSink (Parameters functional mapping decorators to the neural network).

Brain python programs control brain functionality. We constructed a feedforward network of 18 neurons, 15 sensories (5 in each population), and 3 motor IAF neurons. Neurons are stored in an array that follows a 3d coordinates key scheme. Each sensory population, i.e., Ps1, Ps2, and Ps3 mapped with motor neurons in population Pm1, Pm2, and Pm3, respectively. Brain implementation is done with PyNN and simulation handled by the NEST simulator.

Neuron2Robot TFs side postsynaptic motor neurons mapped with a behavior modifier python program using MapSpikeSink decorator. In the behavior modifier program, we access the voltage of actor neurons, the amplified voltage to provide input signals to robot joints. Populations Pm1, Pm2, and Pm3 produce a set of motor responses that cause behavior modification. In the mathematics Set terminology if B1, B2 and B3 are behaviors reflection of R0, R1,..., Rn responses, the behaviors can be declared as B1 = {R0, R1, R2}, B2 = {R3, R4, R5} and B3 = {R6, R7, R8}. If the robot's current behavior is B1, after getting motor responses R3, R4, and R5, behavior B1 is smoothly modified and switched to B2. It was a mixing of responses from two behaviors. For example, the robot gets R3, R4, R5, R6, R7, and R8 responses that result in hybrid behavior Bh = B2 ∪ B3. Bh will have both properties from B2 and B3. On the other hand, if the robot is in any behavior and stops getting a response belongs to that behavior. The robot produces only single one-degree movements, not behavior itself; suppose the current behavior of the robot is B3. It stops getting R6 and R7 responses, so R8 results in one joint movement, so the robot will perform single-joint movements and not resultant behavior.

So, it’s important to receive a number of motor responses equal to the required responses for proper behavior modification. MapRobotPublisher decorator is used for publishing modified joint movements as a ROS topic.

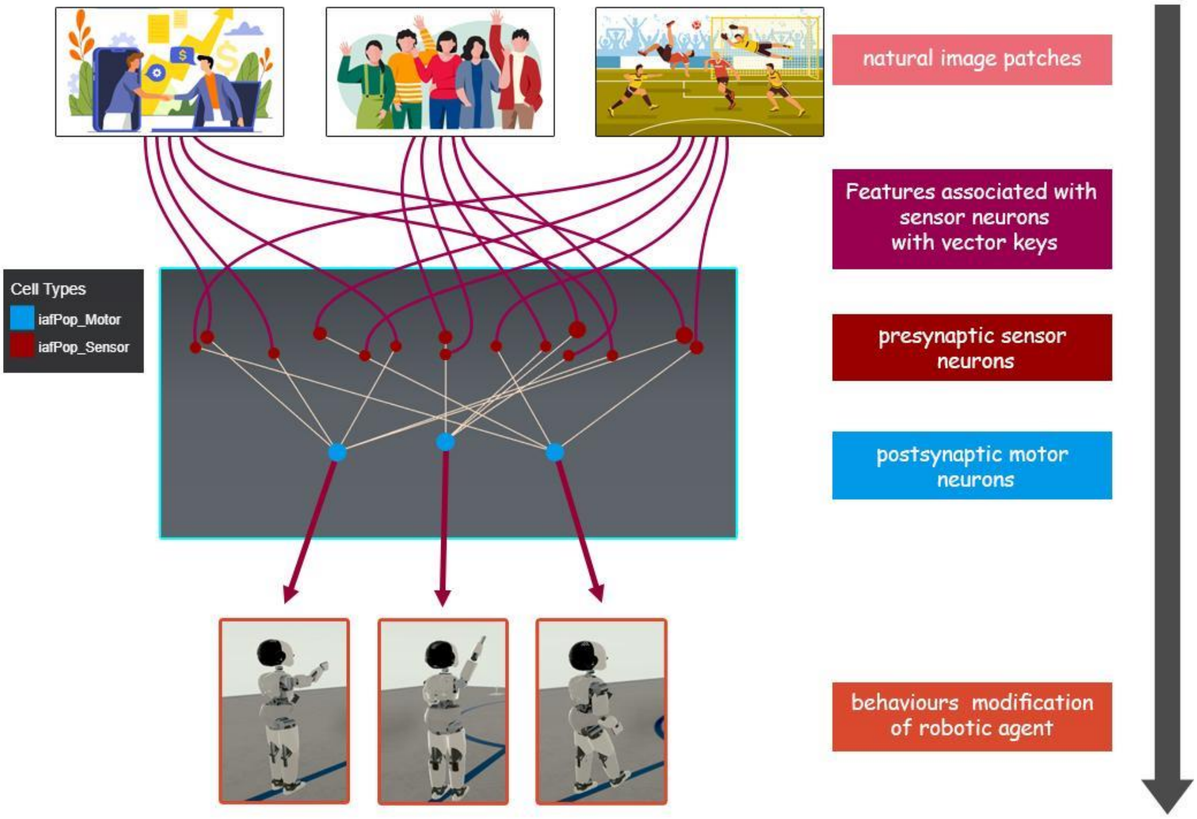

natural image patches
Features associated with sensor neurons with vector keys
presynaptic sensor neurons
postsynaptic motor neurons
behaviours modification of robotic agent
Cell Types
iafPop_Motor
iafPop_Sensor

**Figure 5.** Spatio-temporal data processing from Visual perception to behaviors modification.

Published joint position and velocity topics are used to modify iCub behavior. Perceived natural image patch causes stimulate joint movement and modify behaviors.

## 3. Results and Discussion

Figure 6 presents the result of the experiment. Here 6(A) are the screenshots from experimentation taken while iCub perceiving natural image patches and passing through different behaviors. To support the experiment, three behaviors were designed, B1(handshake), B2(hand-waving), and B3(kick). In 6(B), a spike train of motor neurons is captured in a time window. 6(C) is a plot of joint positions that are the result of a single response. Initially, the robot perceives a blank virtual screen and remains in an idle state. As soon as the state machine switches image I1 and displays it in the virtual screen, the visual system matches the features and excites spatially located sensory neurons, and a Temporal sequence or spike train starts at neuron m0, demonstrating mo fully activated. Activation m0 produces motor responses; here, we just plotted {R0} in the form of joint movement. While staying in behavior B1 the robot body motion offset the robot's initial pose and changed future perception of the environment. Next, when it perceives image I2, A spike train for m1 starts and produces respective motor responses, single movement plotted to represent {R3} response. As its realistic simulation, iCub is not fixed with any external constraint. Behavior movement in any part also causes disturbance in the initial pose and robot cameras to start detecting some wrong features, which activates wrong neurons. We can notice while inputting image I2; the robot detects wrong coordinates of features that produce some wrong spikes and responses. Before the input image, I3 robot forcibly resets so that it can smoothly switch the robot to behavior B3. So visual stimuli encoded in space-time patterns with spatially located neurons spike activity in time to produce movement stimuli. Here we can notice response is recorded only on active neurons. So, these spike trains are the result of the temporal dynamics of the spatial neurons. Vision perception represented Spatio-temporal patterns and applied that representation to modify robot behavior.

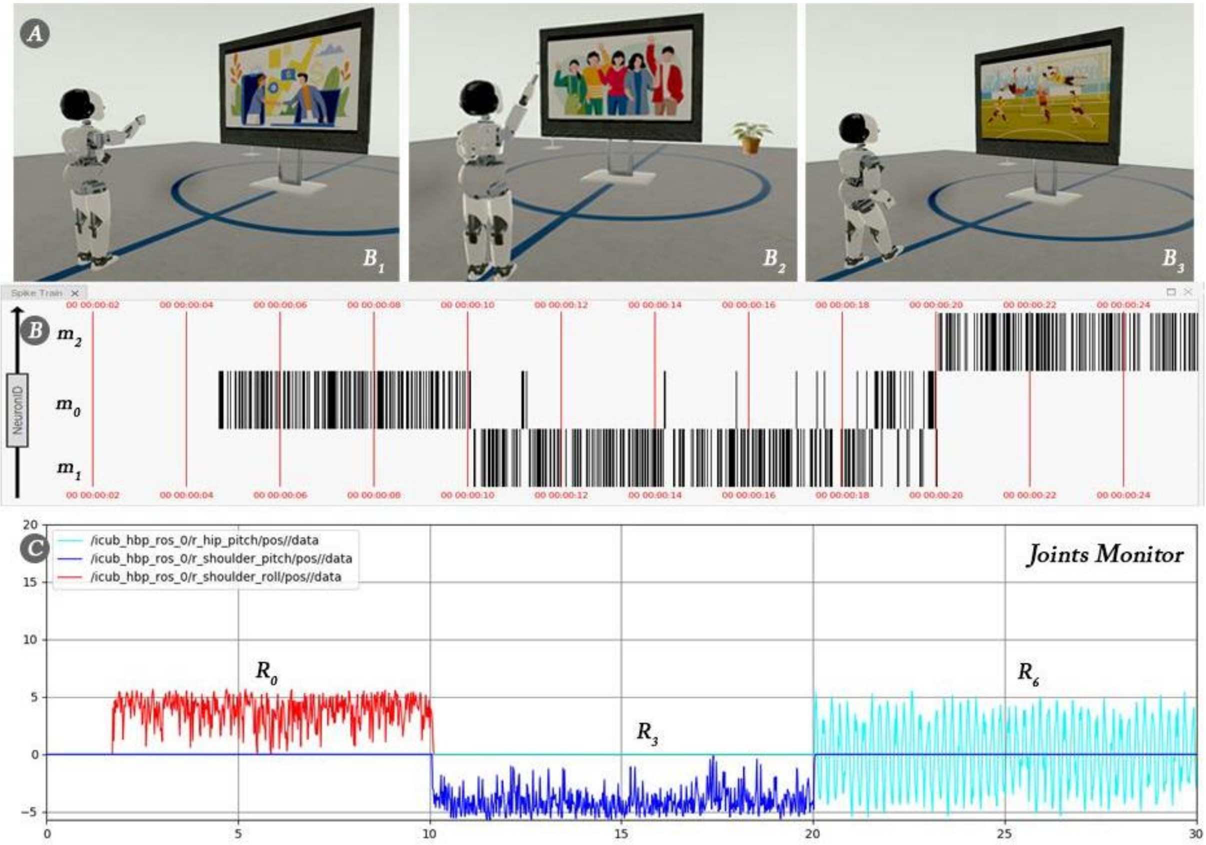


**Figure 6.** Experiment screenshots from top to down, reflected behaviors, spike train, robot joint state.

## 4. Conclusions

This paper presents an approach to design the Spatio-temporal system that represents information processing in both space and time based on the three-dimensional geometric structure of nervous systems and the temporal evolution of activity patterns. This is implemented with spatially located neurons, transmit spikes to each other across the synapses in time, and represent Spatio-temporal patterns of events while communicating. The experiment is designed with the Neurorobotics Platform as part of the Human Brain Project to demonstrate the concept. Input visual stimuli cause simulated brains to produce signals that stimulate robot body movements and change the robot's perception of the environment. It allows us to observe robot interaction with the environment because a robot simulated as a real robot also influences its future sensory input, which greatly helps us understand how the body and brain work together to change its behavior to a given stimulus.

## Funding

This research received no external funding.

## Acknowledgments

The authors would like to acknowledge the Chitkara University Research & Innovation Network (CURIN) for offering computational facilities in their laboratory.

## Conflicts of Interest

The authors declare no conflict of interest.

## References

1. Einstein, A. Einstein's Booklet. In *Relativity*, Princeton University Press: 2019; 7-9.
2. Serdahely, W.J. A brief history of time: From the big bang to black holes. *Journal of Near-Death Studies* **1990**, *9*, 123-131, https://doi.org/10.1007/bf01074212.
3. Townsend, R.G.; Gong, P. Detection and analysis of spatiotemporal patterns in brain activity. *PLoS Comp. Biol.* **2018**, *14*, e1006643, https://doi.org/10.1371/journal.pcbi.1006643.
4. Kasabov, N.K. *Time-space, spiking neural networks and brain-inspired artificial intelligence*; Springer: 2019.
5. Kasabov, N. Deep Learning in Spiking Neural Networks for Brain-Inspired Artificial Intelligence. In Proceedings of Proceedings of the 19th International Conference on Computer Systems and Technologies, Ruse, Bulgaria; ACM, 2018.
6. Szigeti, B.; Gleeson, P.; Vella, M.; Khayrulin, S.; Palyanov, A.; Hokanson, J.; Currie, M.; Cantarelli, M.; Idili, G.; Larson, S. OpenWorm: an open-science approach to modeling Caenorhabditis elegans. *Front. Comput. Neurosci.* **2014**, *8*, 137, https://doi.org/10.3389/fncom.2014.00137.
7. Sarma, G.P.; Lee, C.W.; Portegys, T.; Ghayoomie, V.; Jacobs, T.; Alicea, B.; Cantarelli, M.; Currie, M.; Gerkin, R.C.; Gingell, S.; Gleeson, P.; Gordon, R.; Hasani, R.M.; Idili, G.; Khayrulin, S.; Lung, D.; Palyanov, A.; Watts, M.; Larson, S.D. OpenWorm: overview and recent advances in integrative biological simulation of Caenorhabditis elegans. *Philosophical Transactions of the Royal Society B: Biological Sciences* **2018**, *373*, 20170382, https://doi.org/10.1098/rstb.2017.0382.
8. Cohen, N.; Denham, J.E. Whole animal modeling: piecing together nematode locomotion. *Current Opinion in Systems Biology* **2019**, *13*, 150-160, https://doi.org/10.1016/j.coisb.2018.12.002.
9. Wang, L.; Alexander, C.A. Brain Science and Brain-inspired Artificial Intelligence: Advances and Trends. *Journal of Computer Sciences and Applications* **2019**, *7*, 56-61, https://doi.org/10.12691/jbms-8-1-1.
10. Wang, Y.; Lu, J.; Gavrilova, M.; Fiorini, R.; Kacprzyk, J. *Brain-Inspired Systems (BIS): Cognitive Foundations and Applications*; 2018, https://doi.org/10.1109/SMC.2018.00177.
11. Hu, B.; Guan, Z.H.; Chen, G.; Chen, C.L.P. Neuroscience and Network Dynamics Toward Brain-Inspired Intelligence. *IEEE Transactions on Cybernetics* **2021**, https://doi.org/10.1109/tcyb.2021.3071110.
12. Naghshvarianjahromi, M.; Majumder, S.; Kumar, S.; Naghshvarianjahromi, N.; Deen, M.J. Natural Brain-Inspired Intelligence for Screening in Healthcare Applications. *IEEE Access* **2021**, *9*, 67957-67973, https://doi.org/10.1109/access.2021.3077529.
13. Stefanovski, L.; Meier, J.M.; Pai, R.K.; Triebkorn, P.; Lett, T.; Martin, L.; Bülau, K.; Hofmann-Apitius, M.; Solodkin, A.; McIntosh, A.R. From genes to network models of Alzheimer's disease: Biological framework for multi-scale brain simulation with The Virtual Brain. **2020**, https://doi.org/10.31234/osf.io/4t5qa.
14. Geminiani, A.; Pedrocchi, A.; D'Angelo, E.; Casellato, C. *Spiking cerebellar model with damaged cortical Neural population reproduces human ataxic behaviors in perturbed upper limb reaching*; 2017; https://doi.org/10.1109/NER.2017.8008396.
15. Taherkhani, A.; Belatreche, A.; Li, Y.; Cosma, G.; Maguire, L.P.; McGinnity, T.M. A review of learning in biologically plausible spiking neural networks. *Neural Networks* **2020**, *122*, 253-272, https://doi.org/10.1016/j.neunet.2019.09.036.
16. Tavanaei, A.; Ghodrati, M.; Kheradpisheh, S.R.; Masquelier, T.; Maida, A. Deep learning in spiking neural networks. *Neural Networks* **2019**, *111*, 47-63, https://doi.org/10.1016/j.neunet.2018.12.002.
17. Jang, H.; Skatchkovsky, N.; Simeone, O. Spiking Neural Networks—Part I: Detecting Spatial Patterns. *IEEE Communications Letters* **2021**, *25*, 1736-1740, https://doi.org/10.1109/lcomm.2021.3050207.
18. Paugam-Moisy, H.; Bohte, S.M. Computing with spiking neuron networks. *Handbook of natural computing* **2012**, *1*, 1-47.
19. Kumarasinghe, K.; Kasabov, N.; Taylor, D. Deep learning and deep knowledge representation in Spiking Neural Networks for Brain-Computer Interfaces. *Neural Networks* **2020**, *121*, 169-185, https://doi.org/10.1016/j.neunet.2019.08.029.
20. Kasabov, N.; Capecci, E. Spiking neural network methodology for modelling, classification and understanding of EEG spatio-temporal data measuring cognitive processes. *Information Sciences* **2015**, *294*, 565-575, https://doi.org/10.1016/j.ins.2014.06.028.
21. Marks, S. Immersive visualisation of 3-dimensional spiking neural networks. *Evolving Systems* **2017**, *8*, 193-201, https://doi.org/10.1007/s12530-016-9170-8.

22. Lisitsa, D.; Zhilenkov, A. *Prospects for the development and application of spiking neural networks*; 2017; https://doi.org/10.1109/EIConRus.2017.7910708.
23. Caporale, N.; Dan, Y. Spike timing–dependent plasticity: a Hebbian learning rule. *Annu. Rev. Neurosci.* **2008**, *31*, 25-46, https://doi.org/10.1146/annurev.neuro.31.060407.125639.
24. Brzosko, Z.; Mierau, S.B.; Paulsen, O. Neuromodulation of spike-timing-dependent plasticity: past, present, and future. *Neuron* **2019**, *103*, 563-581, https://doi.org/10.1016/j.neuron.2019.05.041.
25. Knoll, A.; Gewaltig, M.-O. Neurorobotics: A strategic pillar of the Human Brain Project. *Science Robotics* **2016**, 2-3.
26. Hinkel, G.; Groenda, H.; Vannucci, L.; Denninger, O.; Cauli, N.; Ulbrich, S. A domain-specific language (DSL) for integrating neuronal networks in robot control. 2015; 9-15.
27. Amunts, K.; Knoll, A.C.; Lippert, T.; Pennartz, C.M.A.; Ryvlin, P.; Destexhe, A.; Jirsa, V.K.; D'Angelo, E.; Bjaalie, J.G. The Human Brain Project—Synergy between neuroscience, computing, informatics, and brain-inspired technologies. *PLoS Biol.* **2019**, *17*, e3000344, https://doi.org/10.1371/journal.pbio.3000344.
28. Salles, A.; Bjaalie, J.G.; Evers, K.; Farisco, M.; Fothergill, B.T.; Guerrero, M.; Maslen, H.; Muller, J.; Prescott, T.; Stahl, B.C. The human brain project: responsible brain research for the benefit of society. *Neuron* **2019**, *101*, 380-384, https://doi.org/10.1016/j.neuron.2019.01.005.
29. Quigley, M.; Conley, K.; Gerkey, B.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; Ng, A.Y. ROS: an open-source Robot Operating System. 2009; 5.
30. Koenig, N.; Howard, A. Design and use paradigms for Gazebo, an open-source multi-robot simulator. In Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566), 28 Sept.-2 Oct. 2004; 2149-2154.
31. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision* **2004**, *60*, 91-110, https://doi.org/10.1023/b:visi.0000029664.99615.94.
32. Bay, H.; Tuytelaars, T.; Van Gool, L. SURF: Speeded Up Robust Features. European Conference on Computer Vision {ECCV} 2006; 404-417.
33. Calonder, M.; Lepetit, V.; Strecha, C.; Fua, P. BRIEF: Binary robust independent elementary features. 2010; 778-792.
34. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. ORB: An efficient alternative to SIFT or SURF. In Proceedings of 2011 International Conference on Computer Vision, 6-13 Nov. 2011; 2564-2571.
35. Karami, E.; Prasad, S.; Shehata, M. Image matching using SIFT, SURF, BRIEF and ORB: performance comparison for distorted images. *arXiv preprint arXiv:1710.02726* **2017**.
36. Riley, M.A.; Holden, J.G. DYNAMICS OF COGNITION. In *Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition*, Paninski, L., Naud, R., Kistler, W.M., Gerstner, W., Eds. Cambridge University Press: Cambridge, 2014, 417–420, https://doi.org/10.1017/CBO9781107447615.019.
37. Hodgkin, A.L.; Huxley, A.F. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology* **1952**, *117*, 500-544, https://doi.org/10.1113/jphysiol.1952.sp004764.
38. Hu, X.; Liu, C. Dynamic property analysis and circuit implementation of simplified memristive Hodgkin–Huxley neuron model. *Nonlinear Dynamics* **2019**, *97*, 1721-1733, https://doi.org/10.1007/s11071-019-05100-8.
39. Gerstner, W.; Kistler, W.M. *Spiking neuron models: Single neurons, populations, plasticity*; Cambridge university press: 2002.
40. Meffin, H.; Burkitt, A.N.; Grayden, D.B. An Analytical Model for the 'Large, Fluctuating Synaptic Conductance State' Typical of Neocortical Neurons In Vivo. *J. Comput. Neurosci.* **2004**, *16*, 159-175, https://doi.org/10.1023/b:jcns.0000014108.03012.81.
41. Burkitt, A.N. A Review of the Integrate-and-fire Neuron Model: I. Homogeneous Synaptic Input. *Biol. Cybern.* **2006**, *95*, 1-19, https://doi.org/10.1007/s00422-006-0068-6.
42. Cessac, B.; Viéville, T. On dynamics of integrate-and-fire neural networks with conductance based synapses. *Front. Comput. Neurosci.* **2008**, *2*, 2, https://doi.org/10.3389/neuro.10.002.2008.
43. Teeter, C.; Iyer, R.; Menon, V.; Gouwens, N.; Feng, D.; Berg, J.; Szafer, A.; Cain, N.; Zeng, H.; Hawrylycz, M.; Koch, C.; Mihalas, S. Generalized leaky integrate-and-fire models classify multiple neuron types. *Nature Communications* **2018**, *9*, 709, https://doi.org/10.1038/s41467-017-02717-4.
44. Gewaltig, M.-O.; Diesmann, M. Nest (neural simulation tool). *Scholarpedia* **2007**, *2*, 1430, https://doi.org/10.4249/scholarpedia.1430.
45. Davison, A.P.; Brüderle, D.; Eppler, J.M.; Kremkow, J.; Muller, E.; Pecevski, D.; Perrinet, L.; Yger, P. PyNN: A Common Interface for Neuronal Network Simulators. *Frontiers in Neuroinformatics* **2008**, *2*, 11, https://doi.org/10.3389/NEURO.11.011.2008.
46. Gleeson, P.; Cantarelli, M.; Marin, B.; Quintana, A.; Earnshaw, M.; Sadeh, S.; Piasini, E.; Birgiolas, J.; Cannon, R.C.; Cayco-Gajic, N.A. Open source brain: a collaborative resource for visualizing, analyzing,

simulating, and developing standardized models of neurons and circuits. *Neuron* **2019**, *103*, 395-411, https://doi.org/10.1016/j.neuron.2019.05.019.

47. Gleeson, P.; Steuber, V.; Silver, R.A.; Crook, S. NeuroML. In *Computational systems neurobiology*, Springer: 2012; 489-517.
48. Gleeson, P.; Davison, A.P. Relationship between NeuroML and PyNN. *F1000Research* **2020**, *9*.
49. Metta, G.; Sandini, G.; Vernon, D.; Natale, L.; Nori, F. Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems. **2008**.
50. Kreiser, R.; Leite, V.R.C.; Serhan, B.; Bartolozzi, C.; Glover, A.; Sandamirskaya, Y. An on-chip spiking neural network for estimation of the head pose of the icub robot. *Front. Neurosci.* **2020**, *14*, 551, https://doi.org/10.3389/fnins.2020.00551.
51. Hu, Y.; Mombaur, K. Bio-inspired optimal control framework to generate walking motions for the humanoid robot iCub using whole body models. *Applied Sciences* **2018**, *8*, 278, https://doi.org/10.20944/preprints201712.0099.v1.
52. Zhao, J.; Risi, N.; Monforte, M.; Bartolozzi, C.; Indiveri, G.; Donati, E. Closed-Loop Spiking Control on a Neuromorphic Processor Implemented on the iCub. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* **2020**, *10*, 546-556, https://doi.org/10.1109/jetcas.2020.3040390.
53. Singh, G.; Mantri, A.; Sharma, O.; Kaur, R. Virtual reality learning environment for enhancing electronics engineering laboratory experience. *Computer Applications in Engineering Education* **2021**, *29*, 229-243, https://doi.org/10.1002/cae.22333.
54. Faridi, H.; Tuli, N.; Mantri, A.; Singh, G.; Gargrish, S. A framework utilizing augmented reality to improve critical thinking ability and learning gain of the students in Physics. *Computer Applications in Engineering Education* **2021**, *29*, 258-273, https://doi.org/10.1002/cae.22342.
55. Gargrish, S.; Kaur, D.P.; Mantri, A.; Singh, G.; Sharma, B. Measuring effectiveness of augmented reality-based geometry learning assistant on memory retention abilities of the students in 3D geometry. *Computer Applications in Engineering Education* **2021**, https://doi.org/10.1002/cae.22424.