International Journal of Neural Systems (2021) 2150028 (19 pages)
© World Scientific Publishing Company
DOI: 10.1142/S0129065721500283

Int. J. Neur. Syst. Downloaded from www.worldscientific.com
by Mr. Reprints Desk on 07/07/22. Re-use and distribution is strictly not permitted, except for Open Access articles.

A Neurorobotic Embodiment for Exploring the Dynamical Interactions of a Spiking Cerebellar Model and a Robot Arm During Vision-Based Manipulation Tasks

Omar Zahra*, David Navarro-Alarcon* and Silvia Tolu†,‡
**The Hong Kong Polytechnic University, Kowloon, Hong Kong*
†*Technical University of Denmark, Kongens Lyngby, Denmark*
‡*stolu@elektro.dtu.dk*

Received 13 April 2021
Accepted 14 April 2021
Published Online 18 May 2021

While the original goal for developing robots is replacing humans in dangerous and tedious tasks, the final target shall be completely mimicking the human cognitive and motor behavior. Hence, building detailed computational models for the human brain is one of the reasonable ways to attain this. The cerebellum is one of the key players in our neural system to guarantee dexterous manipulation and coordinated movements as concluded from lesions in that region. Studies suggest that it acts as a forward model providing anticipatory corrections for the sensory signals based on observed discrepancies from the reference values. While most studies consider providing the teaching signal as error in joint-space, few studies consider the error in task-space and even fewer consider the spiking nature of the cerebellum on the cellular-level. In this study, a detailed cellular-level forward cerebellar model is developed, including modeling of Golgi and Basket cells which are usually neglected in previous studies. To preserve the biological features of the cerebellum in the developed model, a hyperparameter optimization method tunes the network accordingly. The efficiency and biological plausibility of the proposed cerebellar-based controller is then demonstrated under different robotic manipulation tasks reproducing motor behavior observed in human reaching experiments.

Keywords: Spiking neural networks; cerebellum; robotic manipulation; sensor-based control; cellular-level; spiking.

1. Introduction

Neurorobotics is gaining more attention nowadays not only for improving the performance of robot controllers but also to reveal some of the mysteries about how our brains work.[1,2] Limitation of state-of-the-art techniques to monitor spiking activity of all neurons in the brain makes it necessary to develop accurate computational models to verify theories about neural brain mechanisms.[3] Also, the mutual benefit that derives from the joint research in neuroscience and robotics fields[4] enables the development of adaptive biologically inspired controllers that can be used as a basis to explain mechanisms of learning and enhance the performance of robots. Hence, the motor system lies among the most studied for the common interest in both fields. Each of the brain regions contributing to the motor control has distinctive features leading to different roles in the control process.

This study is concerned with modeling an essential complex region in our motor system, the cerebellum.[5,6] The cerebellum is well known to help achieve fine motor control and precise timing and coordination of the movement of the joints to achieve dexterous motion.[7] That was proven by studies of patients with lesions in the cerebellum[8] suffering from clumsy

‡Corresponding author.

Int. J. Neur. Syst. Downloaded from www.worldscientific.com
by Mr. Reprints Desk on 07/07/22. Re-use and distribution is strictly not permitted, except for Open Access articles.

staggering movements similar to a drunken behavior. In robotics field, incorporating a cerebellar model contributes to enhancing the accuracy and precision of robot movements which is critical in many robotic applications like surgery.[9,10]

Various models[11] were built for the cerebellum based upon different theories for its role in our movements. One theory is that cerebellum acts as an inverse model that provides corrections for the motor commands.[12] Another theory is that it acts instead as a forward model to improve the sensory predictions.[13] Further theories were introduced as well to combine the merits of both forward and inverse models.[14–16]

The aim of this work is to develop a controller based on a cerebellar-like model, developed on the cellular-level, to guide the motion of robots with real-time sensory information. The cerebellar model developed in this study is more detailed from the biological perspective to the previously developed model[17,18] to demonstrate the effect of these additional features and its effect on the performance. The controller first builds a sensorimotor differential map through motor babbling and the cerebellum acts as a Smith predictor[19] to correct discrepancies in sensory readings to enhance accuracy and precision of robot movements. Most of the previous studies included a forward and inverse cerebellar model to act together.[9,14,20] In their case, the forward model provided sensory predictions while the inverse model provided corrections to the motor commands. Such hybrid approach is well appreciated from the control perspective but it has less biological evidence compared to the forward cerebellar model.[21]

While developing cellular-level model adds more challenges for tuning and constructing the network, it provides insight into the real working of the biological counterparts and allows to import additional features from the wide repository of studies exploiting neural mechanisms to achieve such features. Building a detailed cellular level model requires utilizing spiking neural network (SNN), the third generation of artificial neural networks (ANN), to give a more faithful representation of the neuronal dynamics which provide them with more complex and realistic firing patterns based on spikes.[22,23] The SNN adds a temporal dimension compared to the previous generations of ANN, which allows for developing more biologically realistic learning mechanisms and a more efficient representation of the spaces/dimensions encoded.[22,24–27]

In Ref. 18, a controller is developed based on an inverse cellular-level cerebellar model to enhance the robot movement. The controller relies on a trajectory planner and inverse kinematics model to generate reference signals for angular position and speed. This limits the ability to learn only the manipulation of the robot based on the given reference value and is not suitable to learn directly from sensory feedback. In Ref. 28, a cerebellar model based on the adaptive filter theory is developed combining some computational neuroscience techniques along with machine learning. Such combination aims to achieve the sensorimotor adaptation,[29] but it does not model the spiking activity in the cerebellum. In Ref. 17, a cellular level cerebellar model is developed in which the teaching signal is provided based on the error in task-space. In all the previous studies, no clear method was identified to tune the network parameters, and the developed networks were used to manipulate only the end-effector in some predefined trajectory or to reach a certain target.

This study contributes to building a detailed spiking cellular-level cerebellar model including more biological features compared to the previous studies. The parameters of the network are set using a Bayesian optimization method to preserve several biological features observed in the cerebellum. This is demonstrated by monitoring the activity and firing rates of the different groups of neurons in the cerebellum, and the output from each layer, which is then compared to those obtained from biological counterparts. The teaching signal is provided as the error in task-space (TS) and demonstrates the ability to adapt to executing different tasks and handling manipulation of deformable objects. Additionally, it shall be noted that interaction with the environment plays a critical role in the emergence of various features in our nervous system and in properly validating the proposed models in comparison with the biological counterparts.[30] Hence, the developed model is a fine compromise between the biological plausibility and accurate predictions of robot states. In future studies, the developed model will be utilized to study lesions in the cerebellum.

In the next sections, the preliminaries are introduced (Sec. 2), then modeling of the cerebellar controller and the hyperparameter optimization method

Int. J. Neur. Syst. Downloaded from www.worldscientific.com
by Mr. Reprints Desk on 07/07/22. Re-use and distribution is strictly not permitted, except for Open Access articles.

is described (Sec. 3). The experimental setup along with the results are explored (Sec. 4). Finally, an analysis of the obtained results is discussed and the main conclusions are derived (Sec. 5).

2. Preliminaries

The motor system is composed of several regions, each of which is responsible for a different function. These areas follow a hierarchical organization, as shown in Fig. 1(a). In such hierarchy,[31,32] both parallel and serial connections provide different behaviors in different situations (i.e. dependent upon sensory feedback).

The higher order areas are responsible for decision making and planning the sequence of motion while coordinating the activity of several limbs.[33] Lower order areas, on the other hand, control muscles while regulating forces and velocities with changes in posture and various interactions with the environment. The higher level tasks start from the cortical association areas and prefrontal cortex (along with the Basal Ganglia), which receive sensory information (from the sensory cortex) to generate an abstract plan for motion and the sequence of execution.[33] This plan is then transformed to motor commands in the motor cortex which send these commands to the brain stem and the cerebellum (as an efference copy). The cerebellum plays a role in the coordination of movements and adjustment of timing to attain fine movements.[7,11] Signals from the motor cortex travel to trigger motor neurons which innervate the skeletal muscles.[34] The motor neurons in the spinal cord control the limbs and the movement of the body, while those in the brain stem control facial and head movements. In relation with the critical role of the cerebellum in both planning and execution of motion, this study focuses on the cerebellar corrections due to noisy sensory readings to obtain fine movements while executing the motor commands generated by the motor cortex to reach a target point in the space. Thus, a computational model of the cerebellum can help improve the performance of robotic controllers. In this work, a biologically inspired control system is built to guide a robot in a serving task after performing motor babbling for several iterations, as shown in Fig. 1. Computational spiking models are developed to both form a coarse sensorimotor map through motor babbling, and to reproduce the cerebellum. The sensory input to the formed map is modulated through the cerebellum to enhance the movement and reduce the deviation from the desired path. The developed cerebellar controller is capable of guiding the robot based on real-time noisy data.

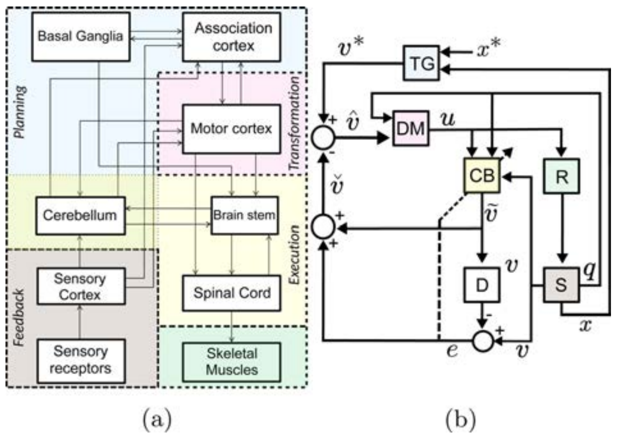


Fig. 1. (a) A simplified schematic of the hierarchical motor system based on studies from the literature.[31,32] The Thalamus is not included for clarity. (b) The block diagram for the proposed cerebellar-based control system for the robot R. Based on the motor command u generated by the differential map (DM), the forward cerebellar model CB provides sensory predictions (i.e. the robot state) for the next cycle. Discrepancy between the actual state observed by the sensors S and the predicted state is used to correct the desired state signal generated by the target generator (TG) before introducing to DM.

3. Methods

3.1. *The cerebellar-based control architecture*

As discussed earlier, various theories were developed about the formation of the cerebellar forward or inverse models. However, more biological evidences support the theory that the cerebellum acts as a forward model based on the behavior observed in the case of cerebellar damage.[21] In this study, the cerebellum acts as a computational forward model to predict the next sensory states based on the desired spatial velocity and the current robot states. The developed model predicts the robot state in TS which differs from other studies where the cerebellar forward model provides predictions in joint-space (JS). Predictions in JS provide a direct reference value to correct the angular positions and velocities, however a reference value for optimal sensory signals from

Int. J. Neur. Syst. Downloaded from www.worldscientific.com
by Mr. Reprints Desk on 07/07/22. Re-use and distribution is strictly not permitted, except for Open Access articles.

joints shall be provided through an inverse model. On the other hand, predictions in TS allow to directly learn from errors related to the executed task without the need for an inverse model which makes it more versatile to learn novel tasks and skills. This model fits in the designed controller to act as a Smith predictor which is known to be capable of handling control schemes with long dead time as shown in Fig. 1. Analogous to control systems, in which dead-time is introduced due to the time needed for sensing, processing of the inputs, computing the control output and actuation, in biological systems the dead-time is introduced due to the time needed for the sensory signals to travel through the nervous system, generate a motor command and travel back for execution to the muscles. In our system, the dead time (expressed as a delay (D)) is caused by slow sensory readings (S) and robot (R) action. In this context, the motor cortex acts as a DM that can generate commands (u) in the JS based on the desired spatial velocity v^* and current joint angles q, and thus acts as an inverse model, as shown in Fig. 2.

This can be expressed as

$$u(t) = f(q(t), v^*(t)), \tag{1}$$

where f represents the inverse mapping formed in the motor cortex to correlate the JS and TS. In case the robot moves from the current end-effector position x to a target position x^*, v^* can be formulated as

$$v^*(t) = \frac{x^*(t) - x(t)}{\|x^*(t) - x(t)\|}. \tag{2}$$

However, due to the delay τ, discussed earlier, along with the imperfection of the map f, the motor command needs a correction before introducing it to the robot. Thus, the cerebellum acts as a forward model to predict spatial velocities upon applying u. While

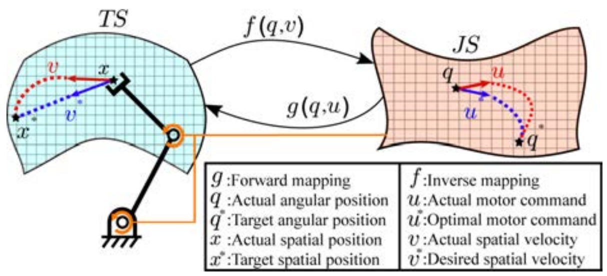

TS
$f(q,v)$
JS
$g(q,u)$
v
x
v^*
x^*
q
u
u^*
q^*
g :Forward mapping
q :Actual angular position
q^* :Target angular position
x :Actual spatial position
x^* :Target spatial position
f :Inverse mapping
u :Actual motor command
u^* :Optimal motor command
v :Actual spatial velocity
v^* :Desired spatial velocity

Fig. 2. The forward and inverse models correlate TS and JS. In this study, DM approximates the inverse model, while CB approximates the forward model.

the robot motion can be expressed as

$$v(t) = g(q(t), u(t - \tau)), \tag{3}$$

where map g represents the actual differential kinematics of the robot, and $v(t)$ is the spatial velocity of the end-effector upon executing the command $u(t - \tau)$ while the robot joint configuration is $q(t)$.

The spatial velocity predictions generated by the cerebellum $\tilde{v}(t)$ can then be expressed as

$$\tilde{v}(t) = \tilde{g}(q(t), u(t - \tau), v(t), v^*(t)), \tag{4}$$

where $\tilde{g}$ represents the forward model built by the cerebellum to approximate the map g. The model approximations are improved during training relying on the feedback error e, where $e(t) = v(t) - \tilde{v}(t)$. Hence, the error in sensory signals and discrepancy from the predicted value is used to modulate the desired spatial velocity before introducing to the DM, as detailed later in Sec. 3.3. So, DM makes use of the predictions provided by the cerebellum to correct the anticipated error in sensory readings due to the dead-time effect:

$$u(t) = f(q(t), v^*(t), \tilde{v}(t), \tilde{v}(t + \tau)). \tag{5}$$

Taking into consideration both the error from previous trials and the sensory prediction, the controller is enabled to correct the next motor command in an indirect way. This is carried out by adding the error from previous attempts to the predictions of the robot state to make up for the expected error:

$$\check{v}(t) = \tilde{v}(t + \tau) + e(t). \tag{6}$$

Finally, the corrected prediction $\check{v}(t)$ is compared to the desired velocity v^*:

$$\hat{v}(t) = 2v^*(t) - \check{v}(t). \tag{7}$$

To put it in other words, $\hat{v}$ is the sensory signal that can be introduced to the DM to give better estimations and make up for the delay in the feedback cycle.

$$u(t) = f(q(t), \hat{v}(t)). \tag{8}$$

3.2. *The differential mapping SNN*

A two-layer SNN, one input and one output layer, are connected via all-to-all plastic synapses (weights can change) to provide the transformation between two correlated spaces. The network encodes the current joint angles and the spatial velocity of the end-effector in the input layer and encodes the angular

Int. J. Neur. Syst. Downloaded from www.worldscientific.com
by Mr. Reprints Desk on 07/07/22. Re-use and distribution is strictly not permitted, except for Open Access articles.

velocity of the joints in the output layer. As the network correlates mainly the velocities in two spaces, it acts as a differential map; it is named as Differential Mapping Spiking Neural Network (DMSNN).[35] For a robotic manipulator of n degrees of freedom (DOF), TS is represented by n assemblies of neurons, where each assembly encodes a corresponding dimension.

Similarly, for m DOF of JS, m assemblies are needed. Hence, the input layer is made up of $n + m$ assemblies. The assemblies $l^{v_{1:n}}$ encode the n-dimensional spatial velocity, while the assemblies $l^{q_{1:m}}$ encode the m-dimensional joints' angular positions. At the output, the assemblies $l^{\dot{q}_{1:m}}$ encode the m-dimensional joints' angular velocities as depicted in Fig. 3. The two layers of the network are connected via all-to-all plastic synapses (both excitatory and inhibitory). The inhibitory synapses regulate the motor (output) neurons activity to maintain stable learning and hence avoid the unbounded increase in the weights of the excitatory synapses. At the output layer, local inter-inhibitory connections (i.e. within each assembly) are added with low inhibition to proximal neurons and higher inhibition to the distal neurons.

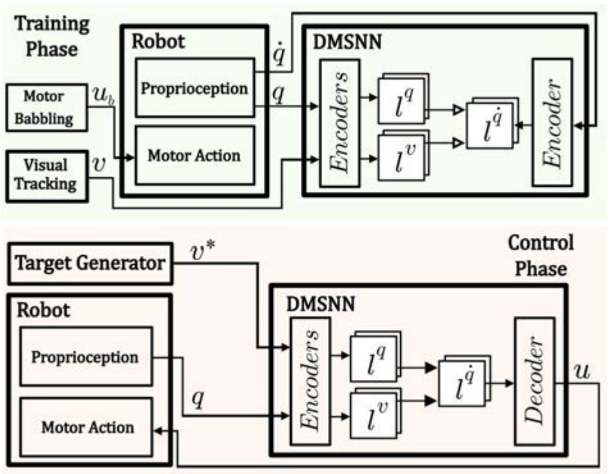


Fig. 3. Input (sensory) neurons are connected to output (motor) neurons through excitatory and inhibitory plastic synapses. The signals introduced to each neuron assembly are depicted. The motion is guided during the training of the network through motor babbling actions in JS based on motor commands u_b. During training, the values of angular positions q, angular velocities $\dot{q}$, and spatial velocities v are encoded and fed to the corresponding assemblies. During the control phase, encoder readings q and updated desired spatial velocity v^* are encoded and fed to the input layer, and the robot is controlled by decoding motor commands from the activity in the output layer. For arrows between the input and output layers, the hollow arrow heads refer to plastic synapses, while solid ones refer to nonplastic synapses.

Modulation of the plastic synapses occurs during training the network, as shown in Fig. 3, where random target angles q_d are generated for motor babbling. The robot joints move towards q_d linearly in JS based on the error calculated as the difference between the desired joint angles and current one q. The internal (i.e. *proprioception*) and external sensors (i.e. *exterioception*) provide the necessary data to both the sensory and motor layers while training DM. These variables are encoded based on the preferred/central value ψ_c defined for each neuron, and a distribution that allows all the neurons in the assembly to contribute to what is known as *population coding*[36]). The neurons' firing rates are defined by the Gaussian tuning curve which can be expressed as

$$\Theta_i(t) = \exp\left(\frac{-\|\psi - \psi_c\|^2}{2\sigma^2}\right), \tag{9}$$

where ψ is the variable's value, and σ is the radius calculated based on the number of neurons per assembly N_l, and the defined range of values of each variable. The synaptic weights are modulated accordingly forming a proper DM. After training ends, the control phase starts where DM is ready to execute a coarse robotic servoing task. The corresponding values of the variable are encoded and introduced to assemblies $l^{q_{1:m}}$ and $l^{v_{1:n}}$, and the output is then decoded from the activity of $l^{\dot{q}_{1:m}}$. The decoding scheme in this case is the *central neuron*[36]:

$$\psi_{\text{est}} = \frac{\Sigma\psi_i \cdot \Theta_i}{\Sigma\Theta_i}, \tag{10}$$

where ψ_i is the defined central value of neuron i in l^Ψ assembly, and ψ_{est} is the decoded/estimated output value . However, DM is formed as a coarse map which lacks in both the precision and accuracy needed for fine control of the robot.

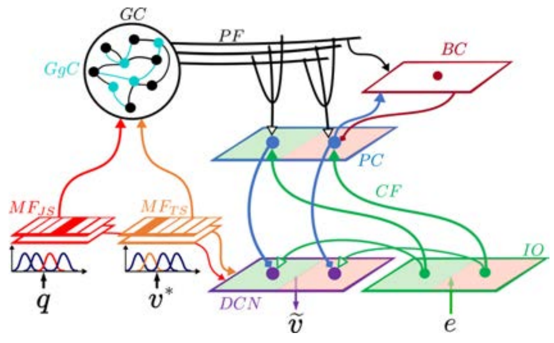
3.3. *The proposed forward cerebellar-like model*

The cerebellum is composed of three layers. The *Granule layer*, which is the innermost layer, is made up mainly of granule cells, which counts up to 80% of the neurons in the brain.[37] It contains as well the *Mossy fiber* axons and the Golgi cells. The *Purkinje*

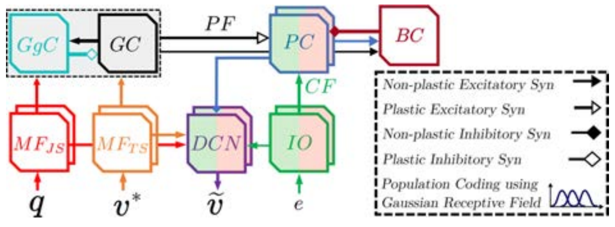
Int. J. Neur. Syst. Downloaded from www.worldscientific.com
by Mr. Reprints Desk on 07/07/22. Re-use and distribution is strictly not permitted, except for Open Access articles.

layer is the middle layer and contains the Purkinje cells which are featured by the distinctive firing pattern and considered a key component for learning to occur in the cerebellum. The *Molecular layer* is the outermost layer containing the axons extended from the granule cells to purkinje cells (known as parallel fibers) intersecting with axons extended from the *inferior olive* (known as climbing fibers). The molecular layer also contains the basket and stellate cells. The cerebellar computational model developed, as shown in Fig. 4, acts to rectify the sensory readings before introducing to the DM which replicates the function of the motor cortex. This enhances the motor commands generated by DM by accounting for sensory discrepancies in the previous cycles as explained in the two previous subsections. The cerebellar forward model is developed based on the cerebellar microcircuit.

The *Mossy Fibers* (MF) encode the current angular position of the joints and the desired Cartesian velocity. These two variables are chosen to provide the essential information to define the state of the robot in both JS and TS. Thus, MF consists of n_{MF} assemblies of neurons, and each assembly encodes a dimension of one of the variables. The angular position is encoded by n_{JS} assemblies forming the $\mathrm{MF}_{\mathrm{JS}}$ group, where n_{JS} is the number of DOFs studied. The desired Cartesian velocity is encoded by n_{TS} assemblies forming the $\mathrm{MF}_{\mathrm{TS}}$ group, where n_{TS} is the number of Cartesian DOF studied. The synapses connecting between MF, Granular cells (GC) and Golgi cells (GgC) shall lead to a sparse coding of robot states. The Inferior olive (IO) provides the teaching signal, which in this study is the TS error, through the climbing fibers (CF) to Purkinje Cells (PC). The error (e) is defined as the discrepancy between the actual and the predicted spatial velocities. The plastic synapses connecting GC to PC, known as parallel fibers PF, are modulated initially under the effect of the teaching signals from CF. These signals evoke activity in PC to provide the desired correction, and thus allows to encode such corrections at the corresponding state of the robot.



(a)



(b)

Fig. 4. (a) A cellular and (b) functional schematic diagrams of the cerebellum. First, inputs from both JS and TS are introduced to MF. After that, spikes from MF travel to GC for sparse encoding of robot states. Signals from GC travel then through PF to PC, which also receives teaching signals from IO. Finally, PC inhibits neurons in DCN from which the output is decoded.

The MF connects to Deep Cerebellar Nuclei (DCN) through excitatory synapses to maintain a basal spiking activity. The PC connects to DCN through inhibitory synapses to allow for the right value to be decoded from the activity of these neurons, while IO connects to DCN through excitatory plastic connections studied to provide fast convergence of learning.[38]

The PC, and similarly IO and DCN, consists of n_{TS} neurons' groups with each group consisting of two assemblies for positive and negative change for each DOF.

After the DM training goes for several iteration, i.e. till the coarse control map is formed, the training then starts for the cerebellum to build the corrective mapping.

In Ref. 39, a study was conducted on infants between 6–12 months to monitor the neural activity in the motor cortex (M1) while reaching targets. It was observed that the activity shifts from a diffused state across M1 to a focused one as the age of infants increases. Additionally, insufficient data is reported about the development of the cerebellum in infants at that age.[40] Thus, it is safe to assume that in infants the development of the cerebellum starts later than the motor cortex and its contribution increases with

Int. J. Neur. Syst. Downloaded from www.worldscientific.com
by Mr. Reprints Desk on 07/07/22. Re-use and distribution is strictly not permitted, except for Open Access articles.

the increase in its size which is reflected by performing fine movements and exhibiting some motor skills.[41]

Encoding in MF. Similar to encoding in DM, the input values to MF is first encoded employing population coding, with the current introduced to the ith neuronal unit in the jth MF assembly ($\Theta_{\mathrm{MF}_{i,j}}(t)$) can be calculated using the following equation:

$$\Theta_{i,j}^{\mathrm{MF}}(t) = \exp\left(\frac{-\|\theta_j^{\mathrm{MF}} - \theta_{i,j}^{\mathrm{MF}}\|^2}{2\sigma_{\mathrm{MF}_j}^2}\right), \tag{11}$$

where θ_j^{MF} is the input to the jth MF assembly, and σ_{MF_j} is the radius for the Gaussian distribution, and the variable ranges from $\theta_{\mathrm{MF}_{j_{\min}}}$ to $\theta_{\mathrm{MF}_{j_{\max}}}$. $\theta_{\mathrm{MF}_{i,j}}$ is the preferred/central value of ith neuron in jth MF assembly. Both $\theta_{\mathrm{MF}_{i,j}}$ and σ_{MF_j} are adjusted using the *self-organization* algorithm (SOA), where the data previously collected for babbling Ξ is used to adjust the values of $\theta_{\mathrm{MF}_{i,j}}$ for all neurons. **SOA:** The algorithm is an adaptation of self organising maps to reorganize the linearly initialized set θ_{MF_j} (i.e. equally spaced from $\theta_{\mathrm{MF}_{j_{\min}}}$ to $\theta_{\mathrm{MF}_{j_{\max}}}$) for better encoding based on the density of data in specific regions. It was chosen for the biologically appealing properties which fits with the theme of this study. The algorithm depends on three concepts: competition, cooperation, and adaptation.[42,43] A best matching unit β is chosen for each central value from the set θ_{MF_j} based on the *competition* with other units/neurons. β is picked based on the Euclidean distance from a random data sample ξ (from the set Ξ):

$$\beta = \arg\min \kappa(\theta_{\mathrm{MF}_{\kappa,j}} - \xi)^2, \tag{12}$$

where the value of $\theta_{\mathrm{MF}_{\beta,j}}$ and that of the neighboring units, emphasizing the *cooperation* concept, at an instant k are updated (i.e. *adaptation*) such that

$$\theta_{\mathrm{MF}_{i,j}}(k+1) = \theta_{\mathrm{MF}_{i,j}}(k) + \rho(k)\nu_{i\beta}(k)(\xi - \theta_{\mathrm{MF}_{i,j}}(k)), \tag{13}$$

where ρ is the learning rate and ν is the neighboring/proximity function given by

$$\nu_{i\beta}(k) = \exp\left(\frac{-\|i-\beta\|^2}{2\vartheta^2(k)}\right) \tag{14}$$

and ϑ is the radius gauging the proximity of the neighborhood. Both ρ and ϑ decay exponentially over the whole period K:

$$\rho(k) = \rho_\circ \exp\left(\frac{-k}{K}\right), \quad \vartheta(k) = \vartheta_\circ \exp\left(\frac{-k}{K}\right), \tag{15}$$

where $\rho_\circ$ and $\vartheta_\circ$ are the initial values for ρ and ϑ, respectively. After running the SOA for K iterations, the radius $\sigma_{\mathrm{MF}_{i,j}}$ is set for every neuron in MF separately such that

$$\sigma_{\mathrm{MF}_{i,j}} = \sigma_{\mathrm{MF}_{i,j}} - \sigma_{\mathrm{MF}_{i+1,j}} \tag{16}$$

with the last radius (i.e. $\sigma_{\mathrm{MF}_{N_l,j}}$) set to have the same radius as the previous one (i.e. $\sigma_{\mathrm{MF}_{N_l-1,j}}$). **Sparse encoding in GC.** The application of the SOA to MF allows for a more distinctive input to the GC and better encoding of the robot states. Both GC and MF are connected through excitatory synapses to the Golgi Cells (GgC), which is connected to the GC through plastic inhibitory connections. The recurrence through the reciprocal connections between GC and GgC along with the connections to the PC allows for sparse encoding of the robot state space and can interpreted as a Liquid State Machine (LSM) as argued in Ref. 44. It is also claimed that the inhibitory action of GgC allows to have a minimum number of neurons in GC active at the same time and thus higher sparsity and better encoding of the states.[8]

Each neuron from GC is connected to a randomly picked neuron from each assembly MF, and hence each neuron from GC shall be connected to n_{MF} neurons. Furthermore, MF connects via excitatory synapses (all-to-all connections) to DCN. **Synaptic connections to PC.** GC connects to PC via plastic excitatory projections (which are known as Parallel Fibers (PF)). The parameters of the network shall be tuned such that PF are modulated only when IO neurons are active. The projections from IO to PC, known as Climbing Fibers CF, are one-to-one synapses to ensure that neurons belonging to the same group (i.e. the same DOF) and direction (i.e. positive/negative changes) connect to each other, and hence ensuring that the right members of CF are modulated. IO connects through excitatory synaptic connections to DCN. The activity of

Int. J. Neur. Syst. Downloaded from www.worldscientific.com
by Mr. Reprints Desk on 07/07/22. Re-use and distribution is strictly not permitted, except for Open Access articles.

neurons in IO is given by

$$\Theta_{\mathrm{IO}+_j} = \begin{cases} \Theta_{\mathrm{IO}_{\max}} & e_{\mathrm{pred}} > \Upsilon, \\ 0 & e_{\mathrm{pred}} \leq \Upsilon, \end{cases} \tag{17}$$

$$\Theta_{\mathrm{IO}-_j} = \begin{cases} 0 & e_{\mathrm{pred}} \geq -\Upsilon, \\ \Theta_{\mathrm{IO}_{\max}} & e_{\mathrm{pred}} < -\Upsilon, \end{cases} \tag{18}$$

where $\Theta_{\mathrm{IO}+_j}$ and $\Theta_{\mathrm{IO}-_j}$ describe the mean firing rates of the two opposite directions of the jth DOF encoded by the IO assemblies, while $\Theta_{\mathrm{IO}_{\max}}$ describes the maximum firing rate of neurons in IO. A threshold value Υ is defined to avoid an overlapping oscillatory activity around the reference value. PC $\rightarrow$ DCN: Similarly, the connections between PC and DCN follow the same concept to allow for activation of the right group of neurons. The cerebellar output is decoded from the activity of DCN neurons:

$$\tilde{v}_j = \frac{\Sigma\Theta_{\mathrm{DCN}+_{i,j}} - \Sigma\Theta_{\mathrm{DCN}-_{i,j}}}{\Theta_{\mathrm{DCN}_{\max}} \times n_{\mathrm{DCN}}} \times v_{\max_j}, \tag{19}$$

where $\tilde{v}_j$ gives the anticipated/predicted velocity for the jth DOF, $\Theta_{\mathrm{DCN}+_{i,j}}$ and $\Theta_{\mathrm{DCN}-_{i,j}}$ are the firing rates of the i_{th} neuron in the two opposing directions assemblies of the jth DOF, and n_{DCN} is the number of neurons in each DCN assembly. $\Theta_{\mathrm{DCN}_{\max}}$ is the maximal firing rate observed in DCN. $v_{\max_j}$ defines the maximum speed for the jth DOF.

3.4. *Optimization of the network parameters*

To have the cerebellar microcircuits employ more features of the biological counterparts, an optimization process is applied to finely tune the parameters. The optimization allows to set the values of the hyperparameters $\mathcal{H}$ of the network to meet specific goals for that mean. Such goals are set to tune the layer by layer of the cerebellar model through the defined objective function $f_i(\mathcal{H}_i)$ to be minimized, where $\mathcal{H}_i \subset \mathcal{H}$. Each objective function consists of a set of objectives $\mathcal{O}_i$ weighed by a vector $w^i_{\mathcal{O}}$, such that $f_i(\mathcal{H}_i) = \mathcal{O}_i w^i_{\mathcal{O}}$. Tuning the model layer by layer allows to properly define and monitor the expected output from each layer. Moreover, this allows to simplify the optimization and avoid the complexities accompanying choosing a large number of hyperparameters to be optimized simultaneously.

Objective 1 (Uniform firing in MF). The input signals coming from the DM is first introduced to the MF. Thus, the first step is to make sure that the firing pattern in MF is suitable to be introduced to the next layer. The parameters affecting the firing are the neuron parameters and the amplitude of the input current to the neurons. The criteria selected are the maximum firing rate of the neurons and the pattern of firing. For the Gaussian distribution of the input current, it is expected to have a corresponding Gaussian distribution, as well, for the firing rates in the MF layer. Thus, a Gaussian distribution fitting, via maximum likelihood estimation,[45] is applied to the number of spikes released from the neurons to compare the mean value of the curve G_{mean} to the expected output (i.e. the neuron $\mathrm{MF}_{\mathrm{nearest}}$ whose central value is the closest to the input value). Hence, the two objectives are defined $\mathcal{O}_1 = [\mathcal{O}^1_1, \mathcal{O}^2_1]$ and can be formulated as

$$\begin{aligned} \mathcal{O}^1_1 &= \frac{\sum_{n=1}^{N_{\mathrm{test}}} \left|fr^{\mathrm{MF}}_{\max}(n) - fr^{\mathrm{MF}}_{\mathrm{desired}}\right|}{N_{\mathrm{test}}}, \\ \mathcal{O}^2_1 &= \frac{\sum_{n=1}^{N_{\mathrm{test}}} \left|G_{\mathrm{mean}}(n) - \mathrm{MF}_{\mathrm{nearest}}(n)\right|}{N_{\mathrm{test}}}. \end{aligned} \tag{20}$$

With $w^1_{\mathcal{O}} = [0.5, 0.5]$, the first objective function is defined as $f_1(\mathcal{H}_1) = \mathcal{O}_1 w^1_{\mathcal{O}}$.

Objective 2 (Sparsity in GC). The output from MF is then introduced to the GC. As mentioned in the previous subsection, GC and GgC act together to give a distinctive output for each input from the MF, and thus allowing for sparse coding of the input. Consequently, f_2 is designed to ensure that a minimum number of GC neurons is active and checks as well for the uniqueness of the activity pattern for every input. Additionally, the firing rate is defined to resemble the neuron activity in the biological counterpart. With these four objectives to be satisfied, $\mathcal{O}_2 = [\mathcal{O}^1_2, \mathcal{O}^2_2, \mathcal{O}^3_2, \mathcal{O}^4_2]$, the firing rates are defined in a way similar to that in MF, where

$$\begin{aligned} \mathcal{O}^1_2 &= \frac{\sum_{n=1}^{N_{\mathrm{test}}} \left|\mathrm{fr}^{\mathrm{GC}}_{\max}(n) - \mathrm{fr}^{\mathrm{GC}}_{\mathrm{desired}}\right|}{N_{\mathrm{test}}}, \\ \mathcal{O}^2_2 &= \frac{\sum_{n=1}^{N_{\mathrm{test}}} \left|\mathrm{fr}^{\mathrm{GgC}}_{\max}(n) - \mathrm{fr}^{\mathrm{GgC}}_{\mathrm{desired}}\right|}{N_{\mathrm{test}}}. \end{aligned} \tag{21}$$

The objective $\mathcal{O}^3_2$ targets minimizing the number of active neurons as much as possible, but also ensures that there is still active neurons in GC (i.e. at least one neuron is active). During each iteration, the number of spikes triggered by each neuron is recorded in the set S^{GC}_n after the nth test trial, to be used to

Int. J. Neur. Syst. Downloaded from www.worldscientific.com
by Mr. Reprints Desk on 07/07/22. Re-use and distribution is strictly not permitted, except for Open Access articles.

define the firing rates and the active neurons as well. Neurons with the a firing rate greater than third of the desired firing rate (i.e. firing rate greater than $0.33\text{fr}^{\text{GC}}_{\text{desired}}$) are kept and the rest are discarded, then S^{GC}_n is updated accordingly. This allows to consider only the neurons that would affect the learning in PC. The difference between the optimal number of firing neurons (chosen as 1 in this case) and the number of active neurons λ^{GC}_n is recorded for each trial n in a set Λ^{GC}_n. To penalize the state in which all neurons in GC are inactive, a large number/score ϕ is returned, which is formulated as follows:

$$\Lambda^{\text{GC}}_n = \begin{cases} \phi, & \lambda^{\text{GC}}_n = 0, \\ |\lambda^{\text{GC}}_n - 1|, & \lambda^{\text{GC}}_n > 0, \end{cases} \tag{22}$$

$$\mathcal{O}^3_2 = \frac{\sum_{n=1}^{N_{\text{test}}} |\Lambda^{\text{GC}}_n - 1|}{N_{\text{test}}}. \tag{23}$$

The objective $\mathcal{O}^4_2$ describes the uniqueness of the output obtained across the N_{test} trials, where the repetition of spiking of a neuron across many trials is penalized. To simplify the computations, from each trial n the neuron with maximum firing rate is recorded, and then, the number of repetitions of each of the neurons in the set is computed. The mean μ_{mean} and maximum μ_{max} number of repetitions is finally computed to formulate $\mathcal{O}^4_2$ as

$$\mathcal{O}^4_2 = 0.3\mu_{\text{max}} + 0.7\mu_{\text{mean}}. \tag{24}$$

With $w^2_{\mathcal{O}} = [0.1, 0.1, 0.2, 0.6]$, the second objective function is defined as $f_2(\mathcal{H}_2) = \mathcal{O}_2 w^2_{\mathcal{O}}$.

Objective 3 (Proper firing rates in PC). The neurons of PC are known to have distinguishable firing patterns in response to different inputs. The input from GC tends to trigger simple spikes (SS), while the input from IO/CF triggers complex spikes (CS) in PC. These two spiking patterns differ from each other in many aspects, but in this study only their respective firing rates are considered. Moreover, the assemblies of neurons within each group tend to have an alternating activity for different directions of motion. Thus, three objectives are defined, $\mathcal{O}_3 = [\mathcal{O}^1_3, \mathcal{O}^2_3, \mathcal{O}^3_3]$. The first two objectives can be formulated as

$$\mathcal{O}^1_3 = \frac{\sum_{n=1}^{N_{\text{test}}} |\text{fr}^{\text{PC}}_{\text{SS}}(n) - \text{fr}^{\text{SS}}|}{N_{\text{test}}}, \tag{25}$$

$$\mathcal{O}^2_3 = \frac{\sum_{n=1}^{N_{\text{test}}} |\text{fr}^{\text{PC}}_{\text{CS}}(n) - \text{fr}^{\text{CS}}|}{N_{\text{test}}}. \tag{26}$$

To construct the formula to describe the third objective $\mathcal{O}^3_3$, firstly the activity of the PC is compared to a threshold value (chosen as the maximum firing frequency of the simple spikes) with those above and below the threshold assigned as active/true and inactive/false, respectively. In this case, the most desirable state is to only have one group of neurons active per DOF, which is represented by an XNOR logic gate:

$$A(\text{PC}_{\pm j}) = \begin{cases} 1, & \text{fr}^{\text{PC}_{\pm j}} > \text{fr}^{\text{PC}}_{\text{SS}}, \\ 0, & \text{fr}^{\text{PC}_{\pm j}} \leq \text{fr}^{\text{PC}}_{\text{SS}}, \end{cases} \tag{27}$$

where $\text{fr}^{\text{PC}_{\pm j}}$ is the mean firing rate of neuron in either of the two assemblies of neurons for each DOF j. $\mathcal{O}^3_3$ can then be expressed as

$$\mathcal{O}^3_3 = \frac{1}{n_{\text{TS}}} \sum_{j=1}^{n_{\text{TS}}} A(\text{PC}_{+j}) \odot A(\text{PC}_{-j}). \tag{28}$$

With $w^3_{\mathcal{O}} = [0.2, 0.2, 0.6]$, the third objective function is defined as $f_3(\mathcal{H}_3) = \mathcal{O}_3 w^3_{\mathcal{O}}$.

Objective 4 (Proper output from DC). After optimization is carried for the previous hyperparameters, this last objective function is directed to test the operation of the network while optimizing the parameters affecting the activity in the DCN. The robot, in a simulation environment, repeats a chosen motion toward a target four times, and both the error while moving e_{pred} and the execution time Δ are recorded. The error in this case is formulated as

$$e_{\text{pred}} = \left|\arccos\left(\frac{\tilde{\mathbf{v}} \cdot \mathbf{v}}{\|\tilde{\mathbf{v}}\|\|\mathbf{v}\|}\right)\right|. \tag{29}$$

Consequently, $\mathcal{O}_4$ consists of six weighted objectives, where $\mathcal{O}_4 = [\mathcal{O}^1_4, \mathcal{O}^2_4, \mathcal{O}^3_4, \mathcal{O}^4_4, \mathcal{O}^5_4, \mathcal{O}^6_4]$. The objective $\mathcal{O}^1_4$ acts to keep the mean value of the error e_{pred} across the four trials as minimum as possible, and an inverted firing pattern in the opposing groups of neurons within the DCN and when compared to PC as well. The objectives $\mathcal{O}^2_4$ and $\mathcal{O}^3_4$ are set to promote the decrease in the error e_{pred} and the execution time Δ, respectively, as the training proceeds. To achieve this, each of these objectives is assigned a value of 1 at the beginning of the training, and the variables (mean value of e_{pred} and Δ) are compared to those from the previous trial, to deduct 0.33 in case of a decrease in the value of the variable. Hence, in case of a consistent decrease in the error and execution time from one trial to another, reflecting a successful

Int. J. Neur. Syst. Downloaded from www.worldscientific.com
by Mr. Reprints Desk on 07/07/22. Re-use and distribution is strictly not permitted, except for Open Access articles.

learning process, the values of these objectives would return a value zero, which is the absolute minimum in this case. The objective $\mathcal{O}_4^4$ adjusts the firing rate of DCN neurons in the desired range such that

$$\mathcal{O}_4^4 = \frac{\sum_{n=1}^{N_{\text{test}}} |\text{fr}_{\max}^{\text{DCN}}(n) - \text{fr}_{\text{desired}}^{\text{DCN}}|}{N_{\text{test}}}. \tag{30}$$

Similar to PC, in DCN the opposing groups of neurons shall be set to fire in an alternating manner, which is formulated as

$$A(\text{DCN}_{\pm j}) = \begin{cases} 1, & \text{fr}^{\text{DCN}_{\pm j}} > \text{fr}^{\text{DCN}}, \\ 0, & \text{fr}^{\text{DCN}_{\pm j}} \leq \text{fr}^{\text{DCN}}, \end{cases} \tag{31}$$

where fr^{DCN} is the mean firing rate of DCN, and $\text{fr}^{\text{DCN}_{\pm j}}$ is the mean firing rate of neuron in either of the two assemblies of neurons for each DOF j. $\mathcal{O}_4^5$ can then be expressed as

$$\mathcal{O}_4^5 = \frac{1}{n_{\text{TS}}} \sum_{j=1}^{n_{\text{TS}}} A(\text{DCN}_{+j}) \odot A(\text{DCN}_{-j}). \tag{32}$$

Additionally, the objective $\mathcal{O}_4^6$ ensures that the activity in the assemblies of DCN opposes that of the corresponding ones for same DOF in PC (i.e. fire in an inverted manner):

$$\mathcal{O}_4^6 = \frac{1}{n_{\text{TS}}} \sum_{j=1}^{n_{\text{TS}}} A(\text{DCN}_{\pm j}) \odot A(\text{PC}_{\pm j}). \tag{33}$$

With $w_{\mathcal{O}}^4 = [0.3, 0.1, 0.1, 0.1, 0.2, 0.2]$, the fourth objective function is defined as $f_4(\mathcal{H}_4) = \mathcal{O}_4 w_{\mathcal{O}}^4$. It can be concluded that: (i) $\mathcal{O}_{1-3}$ act to optimize the firing properties of neuronal groups and ensure that each group executes the assigned role. (ii) $\mathcal{O}_4$ involves directly minimizing the error in predictions e_{pred} (as defined by subobjectives $\mathcal{O}_4^{1-3}$) along with optimizing the firing properties of DCN. For a proper neurorobotic embodiment, both the biological features and the robot-environment interaction are essential to create a basis for neuroscientific studies.

3.5. *Bayesian optimization for the objectives*

To meet these objectives, Bayesian Optimization (BO) is utilized to optimize the defined objective functions.[46] These functions would be very costly and time consuming to optimize through manual or random searching methods due to the high dimensionality and stochasticity of the search space. BO develops a probabilistic model for the objective function to facilitate the evaluation of the objective function while making use of the history of previous trials to guide the optimization process, as explained later in this subsection. Thus, the optimal solution is sought to minimize each objective function to obtain the optimal hyperparameters h_i^*, such that

$$h_i^* = \underset{h_i \in \mathcal{H}_i}{\arg\min} f_i(h_i). \tag{34}$$

The main constituents through which a BO method is identified are the regression model and the acquisition function. The probabilistic regression model *surrogates* the objective function (and referred to usually as the surrogate model). This probabilistic model is initiated with some random evaluations to guide the algorithm, starting from complete uncertainty (prior), and develops as more evaluations are stored in the history to give better future evaluations and decrease the uncertainty (posterior). The surrogate model is expressed as $\mathcal{S}_i = P(\mathcal{L}_i|\mathcal{H}_i)$ representing the mapping of the $\mathcal{H}_i$ hyperparameters to a probability of a loss/score $\mathcal{L}_i$ for an objective function f_i. The acquisition function (also known as selection function) allows for selecting appropriate candidates to improve the surrogate model while exploring for the optimum values for the hyperparameters. Hence, a proper choice of the acquisition functions guarantees a balance between exploration and exploitation without getting trapped in a local minima. In this study, an *Adaptive Tree Parzen Estimator* (ATPE) is chosen as a regression model,[47] and the *Expected Improvement* (EI) as the acquisition function.

The standard TPE is known to fit for optimization problems where either a mixture of discrete and continuous hyperparameter spaces are to be studied or when the hyperparameters are contingent upon each other, and it is chosen for the latter reason. In TPE, rather than describing the posterior $P(\mathcal{L}_i|\mathcal{H}_i)$, it describes instead $P(\mathcal{H}_i|\mathcal{L}_i)$, relying on Bayes rule such that

$$P(\mathcal{L}_i|\mathcal{H}_i) = \frac{P(\mathcal{H}_i|\mathcal{L}_i)P(\mathcal{L}_i)}{P(\mathcal{H}_i)}. \tag{35}$$

The TPE targets building two separate hierarchical processes, $P(\mathcal{H}_i|\mathcal{L}_i \in \mathcal{U})$ and $P(\mathcal{H}_i|\mathcal{L}_i \in \mathcal{D})$, where the sets $\mathcal{U}$ and $\mathcal{D}$ contain the highest and lowest values of $\mathcal{L}_i$, respectively, observed so far reference to a defined threshold value $\mathcal{L}_i^*$. This threshold value

Int. J. Neur. Syst. Downloaded from www.worldscientific.com
by Mr. Reprints Desk on 07/07/22. Re-use and distribution is strictly not permitted, except for Open Access articles.

is decided based on a predefined percentage γ such that $P(\mathcal{L}_i < \mathcal{L}_i^*) = \gamma$. The likelihoods $\mathcal{U}$ and $\mathcal{D}$ are modelled via kernel density estimators (which in this case is the *Parzen estimator*). The Parzen estimator PE allows to represent a function through a mixture of kernels K, which are continuous distributions, to be expressed as

$$P(\mathcal{H}) = \frac{1}{N_p \eta} \sum_{j=1}^{N_p} K \frac{\mathcal{H} - \mathcal{H}_j}{\eta}, \tag{36}$$

where N_p is the number of kernels used for the approximation, η is the bandwidth of each kernel, and K is chosen to be a normal distribution. Modeling $\mathcal{U}$ and $\mathcal{D}$ gives a way to choose hyperparameters for the next observations that are more likely to return lower values for the objective functions (in the case of minimization of objective functions).

Although TPE has less time complexity compared to other BO methods (as Gaussian Process BO), TPE does not model interaction/correlations among the hyperparameters. The ATPE addresses this drawback by judging from Spearman correlation[48] between the hyperparameters which parameters to vary and which parameters to lock to achieve a more efficient exploration. Also, among the drawbacks of TPE is that it has a fixed value for γ and a fixed number of candidates introduced to the acquisition function to predict the next candidate optimal solution, which were introduced initially while solving some specific problems.[46] ATPE introduced empirically concluded formulas based on the cardinality of the search spaces for the hyperparameters to give better values for these two variables.

In this study, the *Expected Improvement* EI is chosen as the acquisition function, to maximize the ratio $P(\mathcal{H}_i|\mathcal{L}_i \in \mathcal{D})/P(\mathcal{H}_i|\mathcal{L}_i \in \mathcal{U})$. The EI generates a probability of obtaining a better solution than the current optimum solution and the amount of expected improvement as well, and consequently, favors bigger improvements. Thus, the EI outperforms the *maximum probability of improvement*, which estimates the probability of an improvement to occur, but not the amount of improvement.[49] Also, EI was found to outperform *upper/lower confidence bound* as compared in Ref. 50. The basic formula for the EI is[46]

$$\mathrm{EI}_{\mathcal{L}_i^*}(\mathcal{H}_i) = \int_{-\infty}^{\mathcal{L}_i^*} (\mathcal{L}_i^* - \mathcal{L}_i) P(\mathcal{L}_i|\mathcal{H}_i)\, d\mathcal{L}_i. \tag{37}$$

By applying Bayes rule (Eq. (35)) and substituting in Eq. (37), the EI can be written as[46]:

$$\mathrm{EI}_{\mathcal{L}_i^*}(\mathcal{H}_i) = \frac{\gamma \mathcal{L}_i^* \mathcal{D}(\mathcal{H}_i) - \mathcal{D}(\mathcal{H}_i) \int_{-\infty}^{\mathcal{L}_i^*} P(\mathcal{L}_i) d\mathcal{L}_i}{\gamma \mathcal{D}(\mathcal{H}_i) + (1-\gamma)\mathcal{U}(\mathcal{H}_i)}, \tag{38}$$

$$\mathrm{EI}_{\mathcal{L}_i^*}(\mathcal{H}_i) \propto \left(\gamma + \frac{\mathcal{U}(\mathcal{H}_i)}{\mathcal{D}(\mathcal{H}_i)}(1-\gamma)\right)^{-1}. \tag{39}$$

From formula (39), it can be concluded that EI acts to maximize the ratio $\mathcal{D}(\mathcal{H}_i)/\mathcal{U}(\mathcal{H}_i)$ and thus leading to introducing better candidates for the next search while still maintaining a trade-off between exploration and exploitation.

3.6. *Neuron model*

To model the spiking activity of the neurons in the cerebellum in real-time, the simple neuron model developed by Izhikevich is chosen for its ability to reproduce different firing patterns.[51] The model provides a decent biological plausibility at a relatively low computational cost. The following differential equations describe the model:

$$\dot{\mathcal{V}} = f(\mathcal{V},\mathcal{U}) = 0.04\mathcal{V}^2 + 5\mathcal{V} + 140 - \mathcal{U} + I, \tag{40}$$

$$\dot{\mathcal{U}} = g(\mathcal{V},\mathcal{U}) = a(b\mathcal{V} - \mathcal{U}). \tag{41}$$

Additionally, the membrane potential is reset after triggering a spike such that

$$\text{if } \mathcal{V} \geq 30 \text{ mV}, \quad \text{then}$$
$$\mathcal{V} \leftarrow c, \quad \mathcal{U} \leftarrow (\mathcal{U} + d), \tag{42}$$

where $\mathcal{V}$ (in mV) is the membrane potential and $\mathcal{U}$ is the variable that acts to lower the neuron's membrane potential (which is also known as the recovery variable). The parameter a (in ms^{-1}) decides the time scale of $\mathcal{U}$ (i.e. the decay rate), and b (dimensionless) describes the sensitivity of the neuron before triggering a spike (i.e. sensitivity of $\mathcal{U}$ to the sub-threshold $\mathcal{V}$). The parameter c (in mV) describes the reset value of $\mathcal{V}$ after a spike is triggered, and d (in mV) gives the reset value of $\mathcal{U}$ after triggering spike. The external currents introduced to the neurons are described by I.

3.7. *Synaptic connections*

To model the plastic connections in both DM and CB, the Spike-timing-dependent plasticity (STDP)

Int. J. Neur. Syst. Downloaded from www.worldscientific.com
by Mr. Reprints Desk on 07/07/22. Re-use and distribution is strictly not permitted, except for Open Access articles.

learning rule is chosen. In STDP, the change in the synaptic weight depends on the spikes relative timing in both the pre and post synaptic neurons.[52] For the CB, the antisymmetric STDP[53] modulates the weight of the plastic synapses, and is formulated as

$$\Delta\varepsilon_{ij} = \begin{cases} -S_a \exp(-\Delta t/\tau_a) & \Delta t \leq 0, \\ S_b \exp(-\Delta t/\tau_b) & \Delta t > 0, \end{cases} \tag{43}$$

where the two coefficients S_a and S_b decide the amount of the decrease and increase of the synaptic weight, respectively. τ_a and τ_b determine the time windows through which synaptic weights decrease and increase, respectively.

The symmetric STDP is applied for the plastic synapses in DM,[54] and can be formulated as

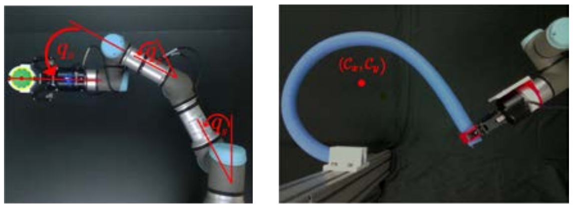
$$\Delta\varepsilon_{ij} = S(1 - (\Delta t/\tau_1)^2) \times \exp(|\Delta t|/\tau_2), \tag{44}$$

where S determines the amount of change in synaptic weights, while the ratio between the τ_1 and τ_2 adjusts the time window for increasing and decreasing the synaptic weights. Δt is the difference between the timing of spikes at post-synaptic and pre-synaptic neurons, respectively, such that $\Delta t = t_{\text{post}} - t_{\text{pre}}$.

4. Results

4.1. *Setup*

A UR3 universal robot is used to test the developed cerebellar controller in two experiments. The shoulder and elbow joints are controlled in an experiment to test the manipulation of the end-effector to a desired position as shown in Fig. 5(a), while the elbow and wrist joints are controlled to test the manipulation of a deformable object, as shown in Fig. 5(b). The shoulder and elbow joints for the first experiment are defined to have ranges of $q_S = [-170°, -135°]$ and $q_E = [-60°, 0°]$, respectively.



(a) (b)

Fig. 5. The robot setups to manipulate (a) the end-effector and (b) the deformable object.

While the elbow and wrist joints for the first experiment are defined to have ranges of $q_E = [-45°, -20°]$ and $q_W = [-210°, -180°]$, respectively. A camera is fixed on top of the robot to track the end-effector position and the deformable object through color filtration. The proprioceptive readings (i.e. joints' positions and velocities) of the robot joints are obtained from motor encoders. The information required for motor babbling is recorded by giving commands to move to random joint angles linearly in JS within he defined ranges, through a script-based programming language developed for universal robots.

For the end-effector manipulation task, the robot is instructed to move to only 100 points in the JS and the collected data during motion is then used to train DM, while for the deformable object case the robot is instructed to move to 300 points as small increments in joint angles may lead to a big change in the centroid of the deformable object and hence more rich data is needed. The spiking neural networks are developed using NeMo package[55] which allows simulating the network using GPU. In this study, a computer with i7-6700K CPU and a GeForce GTX 1080Ti is used to build the networks and control the robot. Each iteration takes 80 ms, where the GPU allows running the simulation in realtime (i.e. 1 s in simulation is equal to 1 s in real world clock). The simulation can be accelerated by using a more powerful GPU[56] or a neuromorphic chip,[57,58] however 80 ms allows a proper averaging of the output for a moderate movement speed without jerky motions.

4.2. *Optimization results*

After running the optimization described in the two previous subsections, the chosen parameters are obtained as shown in Table 1. The optimization is run on a robot in the simulation environment to avoid wear of the mechanical parts and for safety, but the final experiments are conducted on a real robot. Neurons in MF achieve a maximum firing rate of 62 Hz which is comparable to 60 ± 35 Hz spontaneous firing rate observed at excitatory synapses connecting MF to GC, and an average firing rate (for active neurons only) of 40 ± 6 Hz compared to 20 ± 21.[59] The activity in each assembly is shown in Fig. 6(a). For the neurons in GC the sparsity is achieved by having 6 ± 3 neurons only active at the same time for a certain input, while maintaining a maximum firing

Int. J. Neur. Syst. Downloaded from www.worldscientific.com
by Mr. Reprints Desk on 07/07/22. Re-use and distribution is strictly not permitted, except for Open Access articles.

Table 1. CB network parameters.

Values of neuronal parameters

Area	a	b	c	d	$\mathcal{N}$
MF	0.2	0.17	−59.	14.	40
GC	0.22	0.25	−55.	7.	1500
GgC	0.16	1.15	−66.	16.	7
PC	1.74	1.24	−59.	6.	12
BC	0.95	0.4	−68.	16.	70
IO	0.02	0.25	−65.	6.	12
DCN	0.45	0.08	−56.	17.	12

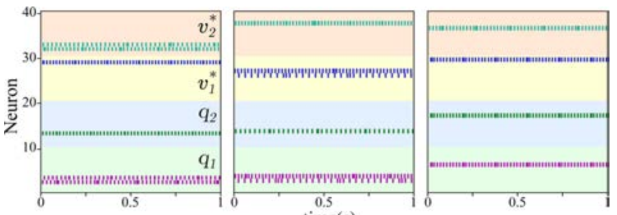
Values of synaptic parameters

Projection	T	V	ω_{init}	$\omega_{\max}$
MF → GC	Rnd	4	3.6	—
MF → GgC	Rnd	1	0.6	—
MF → DCN	A2A	1.0	9.0	—
GC → GgC	Prob	0.01	0.3	—
GgC → GC	Prb	0.5	−9.8	−15.0
GC → PC	Prb	0.8	0.01	24.0
IO → PC	O2O	—	43.0	—
IO → DCN	A2A	1.0	0.47	—
PC → DCN	O2O	—	−13.0	—
PC → BC	Prb	0.4	2.4	—
GC → BC	Prb	0.3	3.2	—
BC → PC	Prb	0.5	−45.4	—

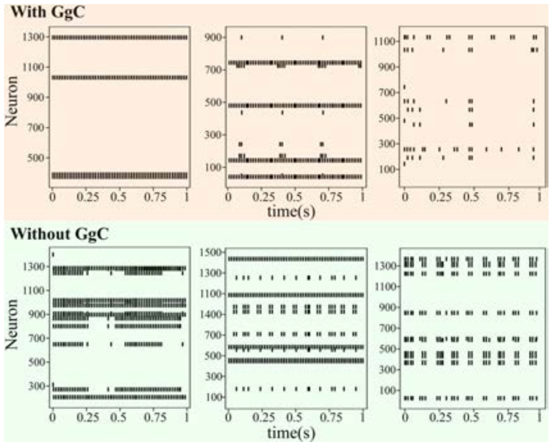
Notes: * The "T" column gives the connections' type from A to B in "Projection" with a value "V" for the parameter. "Rnd" implies that "V" neurons randomly picked from A connect to one neuron in B. "Prb" implies that for each neuron in A there is a probability V to connect to each neuron in B. "A2A" denotes All-to-All connections, where all neurons from A connect to all neurons in B. "O2O" denotes One-to-One connections, where a neuron in A connects to a corresponding neuron in B. $\mathcal{N}$ is the total number of neurons of a specific type. ω_{init} and $\omega_{\max}$ are the initial and maximum value of synaptic weights, respectively.

rate of 86 ± 8 Hz compared to 106 ± 65 Hz reported in Ref. 59 for maximal firing rate in GC while at locomotion state. The sparse activity is achieved by the aid of firing in GgC and the plasticity in synapses between GC and GgC, to suppress the undesirable activity and thus limiting number of active neurons.

The parameters of PC neurons are close to that of bistable neurons demonstrated by Izhikevich,[60] where analysis of the PC properties in Ref. 61 provides indications of bistable behavior, as shown in



(a)

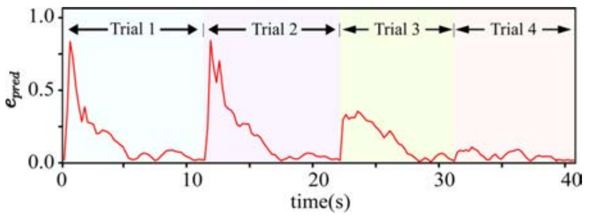


(b)

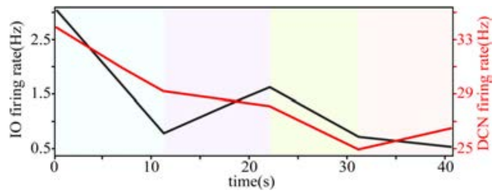
Fig. 6. (Color online) Firing in (a) MF and (b) GC with and without inhibition from GgC. Each assembly in MF is highlighted by a different color.

Fig. 7(c). Hence, the neurons in PC fire simple spikes with a firing rate of 70 ± 7 Hz (i.e. when no input is introduced from IO through the CF) which is comparable to an average of firing rate of 50 Hz.[62] In case of complex spikes a firing rate of 160 ± 14 Hz is achieved, which is mentioned to achieve in biological counterpart to firing rates up to 400 Hz.[62] While the modeled PC fire slower than the real cells, however the big difference in the firing rates of simple and complex spikes facilitates the choice of adequate learning parameters for proper modulation of the PF synapses. It shall be noted from Fig. 7(d) that higher firing rates are achieved in the beginning, corresponding to bigger errors (and thus higher activity in IO), then the rate of activity decreases. After several trials, the strength of PF synapses increases, and hence the firing rate increases. The biostability achieved in PC relies on the inhibitory activity from BC to be able to adjust the firing rates. While BC is not included in the optimization objectives, the obtained parameters refer to the fast spiking activity which suits its interinhibitory role.[51,63] The

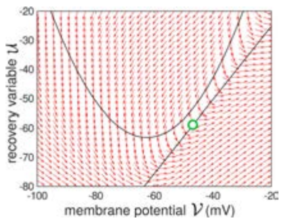
Int. J. Neur. Syst. Downloaded from www.worldscientific.com by Mr. Reprints Desk on 07/07/22. Re-use and distribution is strictly not permitted, except for Open Access articles.

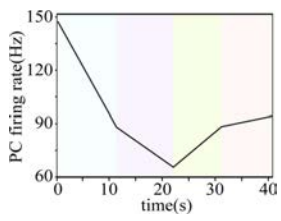


(a)



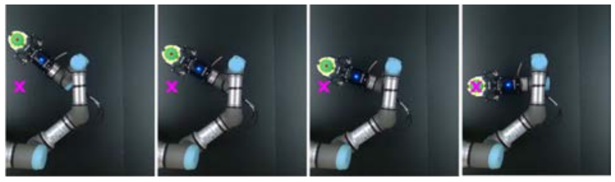
(b)



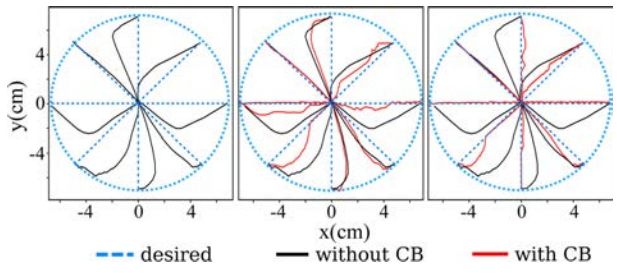


(c) (d)

Fig. 7. (a) The error in cerebellar predictions e_{pred}, as defined in Eq. (29), as the learning proceeds for four trials, (b) the average firing rates of IO and DCN, (c) the phase portrait and (d) firing rates of PC.



(a)



(b)

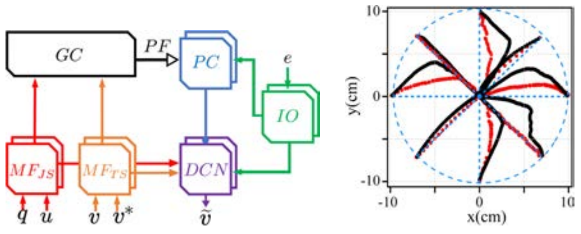
Fig. 8. (Color online) (a) A representative motion of target reaching. (b) The robot reaches the eight radial targets starting from the center of the drawn circle with targets apart 45° from each other. The performance is shown for reaching to targets at the beginning, in the middle and at the end of learning for six trials from left to right. The black and red trails are for reaching without and with cerebellum, respectively. The blue color refers to the desired path.

spikes generated by IO is characterized by a low rate (<3 Hz) as seen in Fig. 7(b), which is comparable to a firing rate 1 Hz reported,[64] however a large number of CF synapses connects between each neuron of PC to a corresponding neuron of IO. This allows this low frequency stimulus to trigger high frequency spikes in PF. The maximum firing rate in DCN is 40 ± 8 Hz, which is comparable to the instantaneous firing rate $30 \sim 50$ Hz reported in Ref. 65.

4.3. *Radial reaching*

In Ref. 66, patients with cerebellar damage attempt to reach radially towards targets at the same distance starting from the same central point with the aim of testing the smoothness of movements of these patients and joint motion coordination compared to healthy persons. Similarly, the robot in this study is commanded to reach radially towards target points equally distributed in an eight-angled star shape with a 45° internal angle and distant from the center 7 cm each. It can be concluded from the robot movements in the eight directions while relying only on DM in the left panel in Fig. 8(b) that always the biggest error is presented in the first generated motor commands while the robot moves from rest. Thus, a small distance is defined for reaching to demonstrate the subsequent effects of the from-rest estimation error. The cerebellar network training is done by repeating the reaching motion to each of the eight targets only six times with an improvement in the reaching motion as shown in the right panel in Fig. 8(b) compared to what obtained after only three repetitions as shown in the middle panel. The maximum deviation recorded for repeating the reaching movement (after training ends) 10 times in each direction is reduced from ca. 21 mm to ca. 8 mm, and reduced by a mean value of ca. 310% (from ca. 11.8 mm to ca. 3.8 mm), while the time for executing reaching movements is reduced by a mean value of ca. 235% (from ca. 26 s to ca. 11 s). The training and testing are done one by one for each direction. This outperforms the network developed in Ref. 17, which can be considered as a suboptimal cerebellar network compared to the

Int. J. Neur. Syst. Downloaded from www.worldscientific.com
by Mr. Reprints Desk on 07/07/22. Re-use and distribution is strictly not permitted, except for Open Access articles.



(a) (b)

Fig. 9. (a) The network proposed in Ref. 17 and (b) the results of radial reaching based after training the network for eight repetitions.

network developed in this study where less biological similarity is achieved through trial and error tuning of the hyperparameters. The suboptimal network developed in Ref. 17, as shown in Fig. 9(b), improved the output from DM by only a reduction of 55% in the maximum deviation and 120% in execution time, after training for eight repetitions. Thanks to the added neuron groups, GgC and BC, better similarity to firing properties and more precise predictions are achieved with the aid of the hyperparameter optimization technique utilized.

4.4. *Deformable object manipulation*

To demonstrate the ability of the developed cerebellar controller to facilitate learning different skills, an experiment is set for the robot to manipulate a deformable object. The contour of the deformable object is observed based on the color, and the moments are calculated such that

$$\mathcal{M}_{ij} = \sum_{x_1}\sum_{x_2} {x_1}^{(i)} {x_2}^{(j)} \varphi(x_1, x_2), \tag{45}$$

where φ gives the intensity of pixels. The centroid $\mathcal{C}_x = (\mathcal{C}_x, \mathcal{C}_y)$ is then calculated in pixels based on the moment such that $\mathcal{C}_x = \mathcal{M}_{10}/\mathcal{M}_{00}$ and $\mathcal{C}_y = \mathcal{M}_{01}/\mathcal{M}_{00}$ as explained in Ref. 67.

The centroid of the object is then converted to the world coordinates using the intrinsic parameters of the camera[68]:

$$\begin{aligned} x_1 &= (\mathcal{C}_x - \mathcal{P}_x)\frac{x_3}{\mathcal{F}}, \\ x_2 &= (\mathcal{C}_y - \mathcal{P}_y)\frac{x_3}{\mathcal{F}}, \end{aligned} \tag{46}$$

where $\mathcal{C}_x$ and $\mathcal{C}_y$ denote to the components of the centroid position in pixels, x_3 is the object's depth away from the camera, $\mathcal{P}_x$ and $\mathcal{P}_y$ denote the principal point and $\mathcal{F}$ is the focal length.

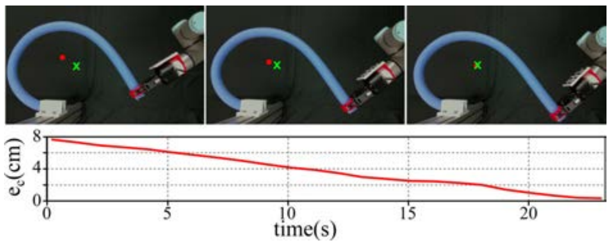


Fig. 10. (Color online) The upper panel shows the shifting of the deformable object centroid (red dot) towards the target point (green cross). The lower panel demonstrates the almost linear decrease in the norm of the distance between the centroid and the target. The plot includes only the reaching guided by the proposed model as reaching under guidance of DM, without CB, failed.

Similar to the motor babbling in case of end-effector tracking, in this case motor babbling is carried out and the corresponding value of centroid is recorded for the joint values to be used for training DM. It shall be noted that using the same parameters as those used in the previous experiment fails to develop a proper map to guide the robot. Thus, DM alone fails to drive the robot to manipulate the object properly. However, the cerebellar model and the control architecture allow to drive the robot and develop the plastic synapses properly to guide the robot motion to deform the object properly as shown in Fig. 10. The robot is given 10 random targets in the studied work-space which are at least 5 cm apart from each other, only three of which are reached using the DM with an average final error of around 7 mm, while the 10 targets are reached successfully with an average final error of less than 4 mm is achieved. While this study considers using the centroid as one feature to manipulate the deformable object, it may not be the optimal choice to achieve such task. Other features can be explored for future studies to give better candidates to reduce the high-dimensional data needed to characterize a deformable object.[69]

5. Discussion and Conclusions

In this study, a biomimetic control system is developed based on the detailed microcircuit of the cerebellum. A spiking cellular-level forward cerebellar model is integrated with a differential map to help improve the motion in terms of speed and deviation from the desired path. The learning is supervised by

Int. J. Neur. Syst. Downloaded from www.worldscientific.com
by Mr. Reprints Desk on 07/07/22. Re-use and distribution is strictly not permitted, except for Open Access articles.

a teaching signal based on task-based sensory feedback, in contrast to most studies in the literature relying on joint-based errors as mentioned earlier. The forward model, acting as a Smith Predictor, then compares the expected output with the actual one to build an anticipation of error for the next cycle and provide corrections to the sensory feedback to the motor-cortex-like differential mapping network, where the spatial motion plan is converted into motor commands. Both the angular and spatial accelerations are not included in this study as the robot moves at moderate speed with a light structure, hence the motion dynamics is not an effective factor.

While the optimization is carried out utilizing the simulated robot, both the data collection for DM and the testing/control phase are executed using the real robot. The optimization takes an average of 300 iterations to obtain the best candidates for each objective, which can still be handled by a real robot given that safety procedures are taken into account. The ability of the network to learn by babbling without any prior knowledge and improve the accuracy by learning from errors in TS makes the network a good candidate to control robots with complex kinematics. However, it shall be noted that a proper choice of the features learnt by the network would be crucial to reduce of the amount of the babbling data collected and the time needed to learn the task properly.

The network parameters are optimized using an ATPE-based Bayesian Optimization. The optimization is carried out step by step to avoid the complexity of handling all the parameters at the same time. This allows to achieve the desirable performance while still maintaining the biological properties of the network components. This allows for future studies to use the detailed computational model to study cases with cerebellar damage in which monitoring the activity of all the neurons simultaneously is not feasible.

The obtained results show that cerebellum acts to reduce the deviation from the target path and the execution time. Additionally, it demonstrates the ability to learn new skills such as deformable object manipulation based on the error in task performance. The developed model demonstrates the ability to reduce the error in a certain direction in only few repetitions indicating the fast convergence of learning, and the suitability to be further developed for real-time adaptive robot control in multiple scenarios and applications.

Moreover, it shall be noted from the conducted experiments that the estimation error in DM is not uniform across the map. This is related to the way of collecting and introducing data for training. This is analogous to having areas mapping different body parts in the motor cortex having different density of neurons depending on how frequent and accurate are the motions generated by each body part. Hence, providing sensory corrections from CB allows DM to improve the quality of the motor output and points to the possibility of transfer of learning by having the cerebellum correcting the motion, and thus improving the quality of the training data introduced to the motor cortex. The radial reaching experiment obtained with the robot arm displays a fair similarity with the observations in Ref. 66.

This study considers only the planar motion at the end-effector as those carried out to test the motion of patients suffering from cerebellar damage. Future studies shall include nonplanar motions and incorporate dynamics/forces acting at the end-effector.[20] Additionally, to scale up the proposed network, several considerations have to be taken into account: (i) DM scaling, similar to scaling up of MF, PC, IO, and DCN, would be simply add additional assemblies corresponding to the increased DOFs. (ii) scaling GC, and consequently GgC and BC, would depend on the size and dimensions of the defined workspace, as well as the precision needed to encode the states of the robot. (iii) for more complex mappings and larger workspaces it may be essential to divide the latter into subspaces that can be approximated by multiple forward models.

Future models would include more features, where developing highly detailed models would allow identifying cerebellar dysfunctions and studying lesions at the cellular-level, which may not be possible using current state-of-art techniques. A more detailed model for the motor cortex, with optimized hyperparameters, would be included rather than the currently simplistic model of DMSNN.

Acknowledgments

This research work was supported in part by the Research Grants Council Under Grant 14203917, in part by PROCORE-France/Hong Kong Joint

Int. J. Neur. Syst. Downloaded from www.worldscientific.com
by Mr. Reprints Desk on 07/07/22. Re-use and distribution is strictly not permitted, except for Open Access articles.

Research Scheme Under Grant F-PolyU503/18, in part by the Key-Area Research and Development Program of Guangdong Province 2020 Under Project 76, in part by the Jiangsu Industrial Technology Research Institute Collaborative Research Program Scheme Under Grant ZG9V, and in part by PolyU Under Grants 252047/18E, ZZHJ, and UAKU.

References

1. H. J. Chiel and R. D. Beer, The brain has a body: Adaptive behavior emerges from interactions of nervous system, body and environment, *Trends Neurosci.* **20**(12) (1997) 553–557.
2. M. Rucci, D. Bullock and F. Santini, Integrating robotics and neuroscience: Brains for robots, bodies for brains, *Adv. Robot.* **21**(10) (2007) 1115–1129.
3. J. L. Krichmar and G. M. Edelman, Machine psychology: Autonomous behavior, perceptual categorization and conditioning in a brain-based device, *Cerebral Cortex* **12**(8) (2002) 818–830.
4. G. M. Edelman, Learning in and from brain-based devices, *Science* **318**(5853) (2007) 1103–1105.
5. J. C. Eccles, *The Cerebellum as a Neuronal Machine* (Springer Science & Business Media, 2013).
6. R. L. Buckner, The cerebellum and cognitive function: 25 years of insight from anatomy and neuroimaging, *Neuron* **80**(3) (2013) 807–815.
7. J. Porrill, P. Dean and S. R. Anderson, Adaptive filters and internal models: Multilevel description of cerebellar function, *Neural Networks* **47** (2013) 134–149.
8. E. R. Kandel, J. H. Schwartz, T. M. Jessell, M. B. T. Jessell, S. Siegelbaum and A. Hudspeth, *Principles of Neural Science* (McGraw-Hill, New York, 2000), p. 4.
9. N. R. Luque, J. A. Garrido, R. R. Carrillo, S. Tolu and E. Ros, Adaptive cerebellar spiking model embedded in the control loop: Context switching and robustness against noise, *Int. J. Neural Syst.* **21**(5) (2011) 385–401.
10. S. Tolu, M. Vanegas, J. A. Garrido, N. R. Luque and E. Ros, Adaptive and predictive control of a simulated robot arm, *Int. J. Neural Syst.* **23**(3) (2013) 1350010.
11. D. M. Wolpert, R. C. Miall and M. Kawato, Internal models in the cerebellum, *Trends Cognitive Sci.* **2**(9) (1998) 338–347.
12. M. Kawato, K. Furukawa and R. Suzuki, A hierarchical neural-network model for control and learning of voluntary movement, *Biol. Cybernet.* **57**(3) (1987) 169–185.
13. T. Ishikawa, S. Tomatsu, J. Izawa and S. Kakei, The cerebro-cerebellum: Could it be loci of forward models? *Neurosci. Res.* **104** (2016) 72–79.
14. D. M. Wolpert and M. Kawato, Multiple paired forward and inverse models for motor control, *Neural Networks* **11**(7–8) (1998) 1317–1329.
15. T. Honda, S. Nagao, Y. Hashimoto, K. Ishikawa, T. Yokota, H. Mizusawa and M. Ito, Tandem internal models execute motor learning in the cerebellum, *Proc. Natl. Acad. Sci.* **115**(28) (2018) 7428–7433.
16. C. Corchado, A. Antonietti, M. C. Capolei, C. Casellato and S. Tolu, Integration of paired spiking cerebellar models for voluntary movement adaptation in a closed-loop neuro-robotic experiment. A simulation study, *2019 IEEE Int. Conf. Cyborg and Bionic Syst.* (IEEE, 2019).
17. O. Zahra, D. Navarro-Alarcon and S. Tolu, Vision-based control for robots by a fully spiking neural system relying on cerebellar predictive learning, arXiv:2011.01641.
18. I. Abadía, F. Naveros, J. A. Garrido, E. Ros and N. R. Luque, On robot compliance: A cerebellar control approach, *IEEE Trans. Cybernet.* (2019).
19. S. Tolu, M. C. Capolei, L. Vannucci, C. Laschi, E. Falotico and M. V. Hernandez, A cerebellum-inspired learning approach for adaptive and anticipatory control, *Int. J. Neural Syst.* **30**(1) (2020) 1950028.
20. M. C. Capolei, E. Angelidis, E. Falotico, H. H. Lund and S. Tolu, A biomimetic control method increases the adaptability of a humanoid robot acting in a dynamic environment, *Front. Neurorobot.* **13** (2019) 70.
21. A. J. Bastian, Moving, sensing and learning with cerebellar damage, *Current Opinion in Neurobiol.* **21**(4) (2011) 596–601.
22. W. Maass, Networks of spiking neurons: The third generation of neural network models, *Neural Networks* **10**(9) (1997) 1659–1671.
23. S. Ghosh-Dastidar and H. Adeli, Spiking neural networks, *Int. J. Neural Syst.* **19**(4) (2009) 295–308.
24. S. Ghosh-Dastidar and H. Adeli, A new supervised learning algorithm for multiple spiking neural networks with application in epilepsy and seizure detection, *Neural Networks* **22**(10) (2009) 1419–1431.
25. R. Hu, Q. Huang, H. Wang, J. He and S. Chang, Monitor-based spiking recurrent network for the representation of complex dynamic patterns, *Int. J. Neural Syst.* **29**(8) (2019) 1950006.
26. Y. Todo, Z. Tang, H. Todo, J. Ji and K. Yamashita, Neurons with multiplicative interactions of nonlinear synapses, *Int. J. Neural Syst.* **29**(8) (2019) 1950012.
27. S. R. Kheradpisheh and T. Masquelier, Temporal backpropagation for spiking neural networks with one spike per neuron, *Int. J. Neural Syst.* **30**(6) (2020) 2050027.
28. M. C. Capolei, N. A. Andersen, H. H. Lund, E. Falotico and S. Tolu, A cerebellar internal models control architecture for online sensorimotor adaptation of a humanoid robot acting in a

Int. J. Neur. Syst. Downloaded from www.worldscientific.com
by Mr. Reprints Desk on 07/07/22. Re-use and distribution is strictly not permitted, except for Open Access articles.

dynamic environment, *IEEE Robot. Automat. Lett.* **5**(1) (2020) 80–87.

29. D. Navarro-Alarcon, J. Qi, J. Zhu and A. Cherubini, A Lyapunov-stable adaptive method to approximate sensorimotor models for sensor-based control, *Front. Neurorobotics* **14**(59) (2020).
30. J. L. Krichmar and G. M. Edelman, Brain-based devices for the study of nervous systems and the development of intelligent machines, *Artific. Life* **11**(1–2) (2005) 63–77.
31. G. J. Mogenson, D. L. Jones and C. Y. Yim, From motivation to action: Functional interface between the limbic system and the motor system, *Progress Neurobiol.* **14**(2–3) (1980) 69–97.
32. J. Merel, M. Botvinick and G. Wayne, Hierarchical motor control in mammals and machines, *Nat. Commun.* **10**(1) (2019) 1–12.
33. J. F. Kalaska and D. J. Crammond, Cerebral cortical mechanisms of reaching movements, *Science* **255**(5051) (1992) 1517–1523.
34. N. Stifani, Motor neurons and the generation of spinal motor neurons diversity, *Front. Cellular Neurosci.* **8** (2014) 293.
35. O. A. Zahra, S. Tolu and D. Navarro-Alarcon, Differential mapping spiking neural network for sensor-based robot control, *Bioinspiration & Biomimetics* (2021).
36. S. Amari *et al.*, *The Handbook of Brain Theory and Neural Networks* (MIT Press, 2003).
37. S. Herculano-Houzel, The human brain in numbers: A linearly scaled-up primate brain, *Front. Human Neurosci.* **3** (2009) 31.
38. N. R. Luque, J. A. Garrido, R. R. Carrillo, E. D'Angelo and E. Ros, Fast convergence of learning requires plasticity between inferior olive and deep cerebellar nuclei in a manipulation task: A closed-loop robotic simulation, *Front. Comput. Neurosci.* **8** (2014) 97.
39. R. Nishiyori, S. Bisconti, S. K. Meehan and B. D. Ulrich, Developmental changes in motor cortex activity as infants develop functional motor skills, *Development. Psychobiol.* **58**(6) (2016) 773–783.
40. H. T. Chugani, Imaging brain metabolism in the newborn, *J. Child Neurol.* **33**(13) (2018) 851–860.
41. R. C. Knickmeyer, S. Gouttard, C. Kang, D. Evans, K. Wilber, J. K. Smith, R. M. Hamer, W. Lin, G. Gerig and J. H. Gilmore, A structural mri study of human brain development from birth to 2 years, *J. Neurosci.* **28**(47) (2008) 12176–12182.
42. T. Kohonen, *Self-Organizing Maps*, Ser. Information Sciences (Springer, Berlin, Heidelberg, 2001).
43. O. Zahra and D. Navarro-Alarcon, A self-organizing network with varying density structure for characterizing sensorimotor transformations in robotic systems, *Annual Conf. Towards Autonomous Robotic Systems* (Springer, 2019), pp. 167–178.
44. T. Yamazaki and S. Tanaka, The cerebellum as a liquid state machine, *Neural Network* **20**(3) (2007) 290–297.
45. D. Cousineau, S. Brown and A. Heathcote, Fitting distributions using maximum likelihood: Methods and packages, *Behavior Res. Methods Instruments & Comput.* **36**(4) (2004) 742–756.
46. J. Bergstra, R. Bardenet, Y. Bengio and B. Kégl, Algorithms for hyper-parameter optimization, *25th Annual Conf. Neural Information Processing Systems* (*NIPS 2011*), Vol. 24, Neural Information Processing Systems Foundation, 2011.
47. B. Arsenault, Adaptive tree parzen estimator, https://github.com/electricbrainio.
48. J. H. Zar, Spearman rank correlation, *Encyclopedia Biostat.* **7** (2005).
49. E. Brochu, V. M. Cora and N. De Freitas, A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning, arXiv:1012.2599.
50. E. Merrill, A. Fern, X. Fern and N. Dolatnia, An empirical study of bayesian optimization: Acquisition versus partition, *J. Machine Learn. Res.* **22**(4) (2021) 1–25.
51. E. M. Izhikevich, Simple model of spiking neurons, *IEEE Trans. Neural Networks* **14**(6) (2003) 1569–1572.
52. D. Buonomano and T. Carvalho, Spike-timing-dependent plasticity (STDP), in *Encyclopedia of Neuroscience*, ed. L. R. Squire (Oxford, Academic Press, 2009), pp. 265–268.
53. K. Buchanan and J. Mellor, The activity requirements for spike timing-dependent plasticity in the hippocampus, *Front. Synaptic Neurosci.* **2** (2010) 11.
54. M. A. Woodin, K. Ganguly and M.-M. Poo, Coincident pre-and postsynaptic activity modifies gabaergic synapses by postsynaptic changes in cl-transporter activity, *Neuron* **39**(5) (2003) 807–820.
55. A. K. Fidjeland, E. B. Roesch, M. P. Shanahan and W. Luk, Nemo: A platform for neural modeling of spiking neurons using gpus, *2009 20th IEEE Int. Conf. Application-Specific Syst.*, *Architectures and Processors* (IEEE, 2009), pp. 137–144.
56. A. K. Fidjeland and M. P. Shanahan, Accelerated simulation of spiking neural networks using gpus, *The 2010 Int. Joint Conf. Neural Networks* (*IJCNN*) (IEEE, 2010), pp. 1–8.
57. G. Indiveri, B. Linares-Barranco, T. J. Hamilton, A. Van Schaik, R. Etienne-Cummings, T. Delbruck, S.-C. Liu, P. Dudek, P. Häfliger, S. Renaud *et al.*, Neuromorphic silicon neuron circuits, *Front. Neurosci.* **5** (2011) 73.
58. F. Galán-Prado, A. Morán, J. Font, M. Roca and J. L. Rosselló, Compact hardware synthesis of stochastic spiking neural networks, *Int. J. Neural Syst.* **29**(8) (2019) 1950004.

Int. J. Neur. Syst. Downloaded from www.worldscientific.com
by Mr. Reprints Desk on 07/07/22. Re-use and distribution is strictly not permitted, except for Open Access articles.

59. K. Powell, A. Mathy, I. Duguid and M. Häusser, Synaptic representation of locomotion in single cerebellar granule cells, *Elife* **4** (2015) e07290.
60. E. M. Izhikevich, Which model to use for cortical spiking neurons? *IEEE Trans. Neural Networks* **15**(5) (2004) 1063–1070.
61. J. D. Engbers, F. R. Fernandez and R. W. Turner, Bistability in purkinje neurons: Ups and downs in cerebellar research, *Neural Networks* **47** (2013) 18–31.
62. L. Squire, D. Berg, F. E. Bloom, S. Du Lac, A. Ghosh and N. C. Spitzer, *Fundamental Neurosci.* (Academic Press, 2012).
63. N. V. Povysheva, A. V. Zaitsev, G. Gonzalez-Burgos and D. A. Lewis, Electrophysiological heterogeneity of fast-spiking interneurons: Chandelier versus basket cells, *PloS One* **8**(8) (2013) e70553.
64. P. J. Mathews, K. H. Lee, Z. Peng, C. R. Houser and T. S. Otis, Effects of climbing fiber driven inhibition on purkinje neuron spiking, *J. Neurosci.* **32**(50) (2012) 17988–17997.
65. E. J. Lang and T. A. Blenkinsop, Control of cerebellar nuclear cells: A direct role for complex spikes? *The Cerebellum* **10**(4) (2011) 694–701.
66. P. Fortier and J. Kalaska, Cerebellar activity during reaching 199 rapid self-paced alternating movements (Schieber, 2002).
67. M.-K. Hu, Visual pattern recognition by moment invariants, *IRE Trans. Inf. Theory* **8**(2) (1962) 179–187.
68. P. Sturm, *Pinhole Camera Model* (Springer, Boston, MA, US, 2014), pp. 610–613.
69. D. Navarro-Alarcon, Y.-H. Liu, J. G. Romero and P. Li, On the visual deformation serving of compliant objects: Uncalibrated control methods and experiments, *Int. J. Robot. Res.* **33**(11) (2014) 1462–1480.