

Received November 12, 2020, accepted December 1, 2020, date of publication December 7, 2020,
date of current version December 23, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3042994

A Biologically Constrained Cerebellar Model With Reinforcement Learning for Robotic Limb Control

RONG LIU^{1,2}, (Member, IEEE), QI ZHANG¹, YARU CHEN¹, JIAXING WANG¹,
AND LE YANG³, (Member, IEEE)

¹Liaoning Key Laboratory of Integrated Circuit and Biomedical Electronic System, School of Biomedical Engineering, Dalian University of Technology, Dalian 116024, China

²Affiliated Zhongshan Hospital of Dalian University, Dalian University, Dalian 116001, China

³Department of Electrical and Computer Engineering, University of Canterbury, Christchurch 8020, New Zealand

Corresponding author: Rong Liu (rliu@dlut.edu.cn)

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 81741137, by the Liaoning Provincial Natural Science Foundation of China under Grant 2020-KF-12-04, by the China Postdoctoral Science Foundation Funded Project under Grant 2020M670714, and by the Fundamental Research Funds for the Central Universities.

ABSTRACT The cerebellum is known to be critical for accurate adaptive control and motor learning. It has long been recognized that the cerebellum acts as a supervised learning machine. However, recent evidence shows that cerebellum is integral to reinforcement learning. This paper proposes a biologically plausible cerebellar model with reinforcement learning based on the cerebellar neural circuitry to eliminate the need for explicit teacher signals. The learning capacity of cerebellar reinforcement learning is first demonstrated by constructing a simulated cerebellar neural network agent and a detailed model of the human arm and muscle system in the Emergent virtual environment. Next, the cerebellar model is incorporated in both a simulated arm and a Geomagic Touch device to further verify the effectiveness of the cerebellar model in reaching tasks. Results from these experiments indicate that the cerebellar simulation is capable of driving the “arm plant” to arrive at the target positions accurately. Moreover, by examining the effect of the number of basic units, we find the results are consistent with previous findings that the central nervous system may recruit the muscle synergies to realize motor control. The study described here prompts several hypotheses about the relationship between motor control and learning and may be useful in the development of general-purpose motor learning systems for machines.

INDEX TERMS Cerebellum, cerebellar model, reinforcement learning, robotic limb control.

I. INTRODUCTION

As an important central nervous system, the cerebellum plays an important role in the adaptive movement control [1]. It has often been likened to a neural machine or computer with its precise geometrical array of intrinsic cell types that allow for the integration and organization of movement-related information through both of its afferent systems. Therefore, intensive research on neurophysiology and modeling of the cerebellum were carried out to pave the way for the establishment of artificially intelligent control systems [2].

Since Eccles *et al.* [3] proposed a comprehensive theory about the internal neuron types, connection and function of the cerebellar cortex, the study on developing a functional model of the cerebellum has entered a new stage. Marr [4] and

Albus [5] recognized that the cerebellum played an essential role in error-based supervised learning (SL) and developed a non-fully connected perceptron-like associative memory network called cerebellar model articulation controller (CMAC). Due to its rapid learning convergence and simple structure, the CMAC has been intensively studied [6] and many variants, such as fuzzy CMAC (FCMAC)[7]–[9], recurrent interval type-2 petri CMAC (RIT2PC)[10], have been proposed to address nonlinear problems. Fujita [11] expanded their models by incorporating a dynamic viewpoint and proposed an adaptive-filter model of the cerebellum. Ito [12] proposed a comprehensive functional model in which a cerebellar micro-complex composed of a cortical microzone and a small cell group in cerebellar acted as an adaptive controller. Alternative theories of cerebellar functions have also been proposed such as internal model [13], adjustable pattern generator [14], and tensor geometrization theory [15].

The associate editor coordinating the review of this manuscript and approving it for publication was Nishant Unnikrishnan.

Along this line, various cerebellar models have been developed in the context of robotic control. The key idea in all of these models is that the climbing fiber inputs onto the Purkinje cells provide an error signal that drives learning in the Purkinje cells, such that to some extent, the resulting modified output of the Purkinje cells reduces or eliminates the error in the future. These models assume that the cerebellum is responsible for supervised learning. However, recent studies showed some contrasting evidence against this prominent idea. The anatomical studies indicate that the cerebellum receives sensory information about stimuli impinging on the organism as well as information about the organism's effect. Swain *et al.* [16] posited that this system's organization made it a mediator of reinforcement learning (RL). They reported the evidence from a variety of learning paradigms to show that the cerebellum was integral to reinforcement learning. Therrien *et al.* [17] used a 15° gradual visuomotor rotation task to examine the learning and retention of a reaching skill under error-based and reinforcement paradigms. The results showed that reinforcement schedules produced better learning and retention compared with the error-based counterpart. However, there was significant learning difference between cerebellar group and control group and cerebellar patients also varied in their learning ability. Using mechanistic model, they predicted that cerebellar damage indirectly impaired the RL capability by increasing motor noise. Recently, Yamazaki *et al.* [18] extended the Marr-Albus-Ito model and implemented the cerebellar circuit with RL algorithm successfully. The major limitation of this approach is that there is no direct experimental evidence or rigorous mathematical justification on the feedback inhibition term which is assumed to be fed by the molecular layer interneurons (MLIs).

However, these cerebellar models oversimplified the cerebellum structure. To develop neurorobotics, it is necessary to adopt new discoveries from neuroscience. Moreover, the significance of the cerebellum model is not only to achieve better control of the robot movement, but also to simulate the damage of the cerebellum and establish the mapping relationship between the damage of the cerebellum and motor dysfunction, so as to guide the rehabilitation training of patients with cerebellar disease.

As a result, the modeling of cerebellum should also take the interpretability of the control process into consideration, e.g. the efficacy of neurotransmission among different layers and the learning mechanisms. Therefore, a biologically constrained cerebellum model with RL inspired by results from biology and physiology was proposed in this paper, which may provide a computational basis for cerebellar learning, memory, and movement coordination.

The rest of the paper is organized as follows. Section II presents the cerebellar RL model architecture after a brief introduction of cerebellar cortex as a RL machine. Section III describes the modeling method which is followed by the simulation experiments in Section IV. Section V gives the

system test results. Finally, conclusion and future work are given in the last section, Section VI.

II. CEREbellAR REINFORCEMENT LEARNING MODEL ARCHITECTURE

A. CEREbellAR MODEL ARCHITECTURE

The cerebellum has a very well-defined anatomy with the same basic circuit replicated throughout. Each basic circuit has a homogeneous structure with two inputs and one output as shown in Fig. 1. Within the cerebellum proper, the principal feature and cell type of the cerebellum is the Purkinje cell (PC) whose nearly two dimensional dendritic trees are arrayed perpendicular to the long axis of the lobule. Each set of PCs receives a convergent input from an array of parallel fibers (PFs) which are the axons of the granule cells (GCs), and a private climbing fiber (CF) input originating from the inferior olive (IO) in the brainstem. GCs encode the sensory information received from the mossy fibers (MFs) axons.

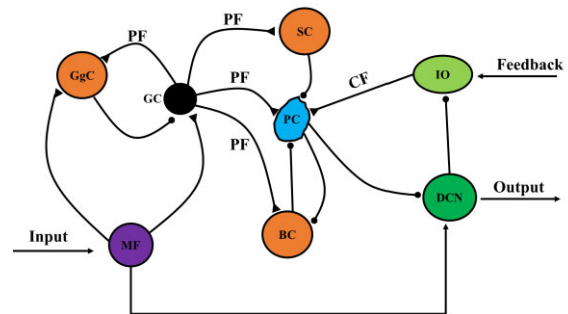


FIGURE 1. Connectivity diagram of the cerebellar circuitry. Excitatory synapses are denoted by triangles and inhibitory synapses by circles. MF: Mossy fibers. DCN: Deep cerebellar nuclei. IO: Inferior Olive. CF: Climbing fiber. GC: Granule Cell. PF: Parallel fiber. PC: Purkinje Cell. GgC: Golgi Cell. SC: Stellate Cell. BC: Basket Cell.

The PC-PF synapses are believed to be related to cerebellar motor learning because of its long-time depression (LTD). As a matter of fact, CF is important for the learning process of the cerebellum. While a given PC receives input from only one climbing fiber, it is considered a very potent exciter of the cell. It is through this strong excitation of PC in specific lobules that particular skeletal motor movements are selected. The output of the PC is inhibitory. As the unique output, PCs inhibit the deep cerebellar nuclei (DCN) that form a link in a regeneratively active cerebro-cerebellar loop to form the motor command. All other connections with PCs are also inhibitory, including stellate cells (SCs), basket cells (BCs). Golgi cells (GgCs) are also inhibitory which receive excitatory input from mossy fibers and synapse on granule cells.

The computational power of the cerebellar cortex as a SL machine has been extensively examined. In this kind of models, the CFs need to provide explicit teacher signals. However, the work by Hausknecht *et al.* [19] showed that the cerebellum using a SL paradigm failed to perform complex tasks well in which explicit teacher signals are not provided. A number of previous studies also showed that CF activities

contain information about directional errors in reaching but they didn't tell an agent how to correct the errors explicitly in terms of joint torques or muscle contractions [18], [20]. In this sense, CF is believed to convey evaluative feedback and cerebellar cortex as a RL machine.

B. CEREBELLAR MODEL IN A RL FRAMEWORK

The essential difference between RL and SL is that explicit “teacher” or “error” information is not required in RL. RL can be defined as the ability to map situations into actions that interact with the environment to maximize a reward [21]. The so-called reward or punishment is the evaluation of the actions taken by the agent. The basic elements of a RL system can be subdivided into five elements, namely, an agent, a policy, a reward or punishment function, a value function and an environment model.

The agent takes an action through a certain policy and then a reward or punishment guiding behavior is obtained through its interaction with the environment. The rewards of a series of actions constitute a value function. There are generally two basic tasks in RL algorithms: policy evaluation and policy improvement [22]. The former calculates the value function according to the current policy, and the latter evaluates the obtained value function in order to update the current policy. The goal of RL is to find an optimal policy for a specific problem, so that the value function obtained under the policy is maximized. According to the basic framework of RL, we designed a robotic limb control system with the cerebellum model as the agent as shown in Fig. 2.

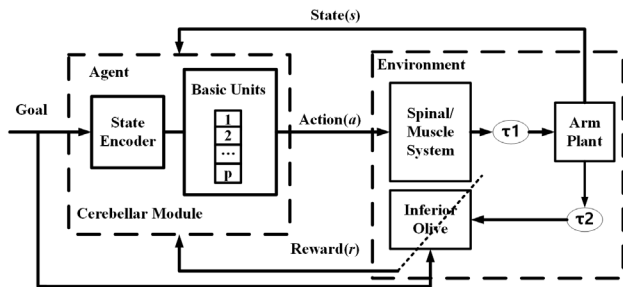


FIGURE 2. Model architecture of a robotic limb control system based on the cerebellar model. The cerebellar module produces signals sent to spinal/muscle system. These signals are then transformed by the spinal/muscle system into torque commands to drive the arm plant. Movement errors are detected by the inferior olive, which update the cerebellar control policy. The delay modules τ_1, τ_2 simulate conduction delays.

The environment mainly includes the spinal/muscle system, the robotic arm plant and inferior olive. The agent, including the basic units and state encoder, perceives the state of the environment s_t and receives the reward signal r_t at time t , and then update policy π_t to select the action a_t . Afterwards, s_t changes to a new state s_{t+1} and evaluates the taken action a_t to generate a new reward r_{t+1} , which will be then sent to the agent again. This process will repeat until the goal of maximizing the reward signals is achieved.

According to Fig. 2, the cerebellar module consists of the state encoder and the basic unit array. The state encoder in the cerebellar module represents the GCs, which provide cerebellum with sparse expansive encoding of the inputs from the spinal cord and motor cortex. Considering the homogeneous structure of the cerebellum, we use basic unit array to establish its structure. In this model, every basic unit receives the same inputs and all the outputs of the basic units are assembled to generate the output of the model. Activation of a basic unit leads to the movement of the arm plant in a direction that is specific to that basic unit, and the magnitude of a basic unit's activity determines the velocity of that movement. Simultaneous activation of selected basic units determines the arm trajectory as the superposition of these movements. Learning processes adjust the subsets of basic units that are selected as well as characteristics of their activity in order to achieve desired movements.

The basic units generate motor commands that are fed to the spinal/muscle system to bring the arm to the specified position. The spinal/muscle system transforms muscle space signals into joint torques.

III. DERIVATION OF THE CEREBELLAR REINFORCEMENT LEARNING MODEL

A. BASIC UNITS ARRAY

Each basic unit has the same structure which contains an artificial neural network (ANN) receiving inputs at each time step and produces an output of nucleus cell firings. Fig. 3 shows the structure of a single basic unit in more details. In one single basic unit, each brown sphere circle represents one PC and the vertical line above it denotes the PC dendrites while the vertical line below it stands for the PC axon. The blue horizontal lines connected with the PC dendrites are PFs. These fibers synapse upon PCs of the model, irrespective of the basic unit in which they are located.

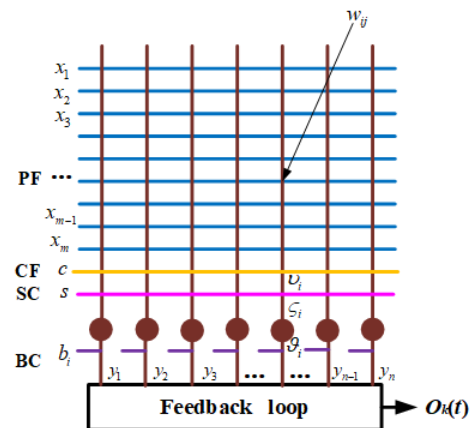


FIGURE 3. The structure of a single basic unit. Each basic unit consists of n PCs with states $y_1(t), \dots, y_n(t)$ at time t and a feedback loop. For the i th PC, the PF input signals are $x_1(t), \dots, x_m(t)$, CF input is c , SC input is s , BC input is b_i , only the connection weights $w_{ij}(1 \leq i \leq n, 1 \leq j \leq m)$ are modified by CF input. The output of the basic unit is determined by the state of the feedback loop and the total amount of PC inhibition of loop.

The yellow and pink lines represent CF and SC respectively. All the PCs of a single basic unit share the same CF and SC inputs. The horizontal purple line that is connected to the PC axon represents BC. The synapses formed between different neurons. The inputs of the basic unit will first determine the status of PF. The activated PF will then transmit the information of the corresponding PC-PF synapses to the PCs. Through the modulation of the intermediate neurons such as BC, SC and CF, the state of PCs will be determined. Finally, the output of each basic unit is determined by the states of all of its PCs as well as its feedback loop.

Different cells feature inhibitory as well as excitatory connections and allow the network to exhibit dynamic temporal behavior. They can be divided into two classes, one representing the state of the neurons and the other denoting the synapse information. Different neurons communicate with each other through synapses. In turn, synapses also produce different effects on neurons according to their activities which are expressed by the parameters listed in Table 1.

TABLE 1. Symbols and notations in the cerebellar model.

Symbol	Quantity	Range	Initial Value
d	number of basic units	a positive integer	6,10
m	number of PFs	a positive integer	108^3
n	number of PCs	a positive integer	8
$x_j(t)$	state of PFs	0 or 1	0
$j=1,2,\dots,m$			
$y_i(t)$	state of PCs	0 or 1	0
$i=1,2,\dots,n$			
$b_i(t)$	state of BC for the i th PC	0 or 1	1
$s(t)$	state of SC	0 or 1	1
$c(t)$	state of CF	0 or 1	0
$w_{ij}(t)$	weight of the synapse forming by the i th PC and j th PF	adjusted	1
ϑ_i	weight of the synapse forming by the i th PC and BC	a certain negative number	-0.1
v_i	weight of the synapse forming by the i th PC and CF	a certain positive number	1
ζ_i	weight of the synapse forming by the i th PC and SC	a certain negative number	-0.1
ϕ	on-threshold value of PC	a certain number	0.8
η	off-threshold value of PC	a certain number	0.6

Suppose each basic unit consists of m PFs and n PCs (also see Fig. 3), the states of PFs and PCs at time t are $x_1(t), \dots, x_m(t)$ and $y_1(t), \dots, y_n(t)$. The i th PC in a basic unit at time t receives the same input signals via multiple excitatory PFs $x_1(t), \dots, x_m(t)$, a single inhibitory BC input $b_i(t)$, a single inhibitory SC input $s(t)$, and a single CF $c(t)$. These inputs are all binary, with 1 and 0 respectively indicating activity and inactivity.

The learned parameter of the network is the synaptic weight between the i th PC and the j th PF. Because of the synaptic plasticity and the property of long-time depression (LTD), we set $w_{ij}(t)$ to be a continuous adjustable variable. ϑ_i , v_i , and ζ_i represent the synapses formed between the i th PC and BC, CF and SC respectively. These weights are often set to be fixed values according to the inhibition or potentiation to the activities of PCs. This model thus mainly involves two types of parameters, the states and weights.

Assuming that there are d basic units in the cerebellar module, the membrane potential $q_{ki}(t)$ of the i th PC in the k th ($k = 1, 2, \dots, d$) unit is determined by the total input to PC at time t , which is equal to

$$q_{ki}(t) = \sum_{j=1}^m x_j(t) \cdot w_{kij}(t) + \vartheta_{ki} \cdot b_i(t) + \zeta_{ki} \cdot s(t) \quad (1)$$

The basic unit number is denoted as subscript k in the notations, such as $w_{kij}(t)$ represents weight of the synapse forming by i th PC and j th PF in the k th unit.

The state of the i th PC in the k th basic unit, denoted by $y_{ki}(t)$, is given by

$$y_{ki}(1) = \begin{cases} 1 & q_{ki}(1) \geq \eta \\ 0 & q_{ki}(1) < \eta \end{cases} \quad (2)$$

$$y_{ki}(t) = \begin{cases} 1 & q_{ki}(t) \geq \phi \parallel (y_{ki}(t-1) = 1 \ \&\& \ q_{ki}(t) > \eta) \\ 0 & q_{ki}(t) \leq \eta \parallel (y_{ki}(t-1) = 0 \ \&\& \ q_{ki}(t) < \phi) \end{cases} \quad (3)$$

where ϕ and η are respectively the on-threshold and off-threshold values of PC, and $\phi > \eta$. The PCs are relatively insensitive to changes in their input except near the on-threshold ϕ , where transitions from the off-state to the on-state occur, and the off-threshold η , where transitions from the on-state to the off-state occur. If the PC is in the off-state, the net input $q_{ki}(t)$ must exceed the ϕ , in order to drive it to the on-state. On the other hand, if the PC is in the on-state, the $q_{ki}(t)$ must drop below the η to force the cell to switch to the off-state.

The effect of PC-PF synapse $w_{kij}(t)$ on PC membrane potential changes with time, and the trend of $w_{kij}(t)$ is more inclined to make the cerebellar module stop the command output. The update formula of $w_{kij}(t)$ is as follows

$$\Delta w_{kij}(t) = \kappa \cdot [1 - y_{ki}(t)]x_j(t) \quad (4)$$

$$w_{kij}(t+1) = w_{kij}(t) + \Delta w_{kij}(t) \quad (5)$$

where κ is a positive constant. The cerebellum produces commands to control arm movement through the spinal muscular system. If the error is within the allowed range, the instructions that cerebellum produces are appropriate. Otherwise, the relevant parameters in the model need to be adjusted through the reinforcement learning model in the inferior olive.

A basic unit’s motor command is generated through the activities of its PCs, which inhibit and modulate the buildup of activity in the feedback loop. When the feedback loop in a unit is in its off-state, the output of the unit is zero. Transition to the on-state is caused by a trigger signal, and the output of a unit is the amount of the proportion of PC in the off-state. In other words, once the feedback loop is triggered, the output of the base unit depends on the state of the PC. However, the opposite process depends only on the PC. Due to the inhibition effect of the PC on the deep cerebellar nucleus, the feedback loop will be turned off when a large fraction of the PC in the basic unit are excited. At this time, the basic unit ceases to produce an output.

The output of the k basic unit $O_k(t)$ is defined by:

$$O_k(t) = \begin{cases} 0 & \text{loop is in off-state} \\ \varepsilon \cdot \left(1 - \left[\frac{\sum_{i=1}^n y_{ki}(t)}{n}\right]\right) & \text{loop is in on-state} \end{cases} \quad (6)$$

where ε is the constant adjustment factor.

B. BASIC UNITS ARRAY

By receiving inputs from multiple classes of MFs and local inhibitory interneurons, the GCs are thought to provide a sparse expansive encoding of the incoming state information. In this model, the sparse expansive encoding is realized by the state encoder. This coding scheme makes use of multiple tiles over the state space. A single tiling partitions the arm end position space into discrete but non-overlapping tiles. Each PF corresponds to a single tile. When the system state falls into a particular tile, the corresponding PF x_j is given an activation level of 1, and all others are set to be 0. In this way, the values of $x_1 \dots x_m$ are determined.

This is realized by the quantization layer in a CMAC network. Each input activates certain fields in the quantization layer. If we use $\mathbf{M}(t) = [x(t), y(t), z(t)]^T$ to denote a 3-dimensional input state, the output of the state encoder can be formulated using a mapping f as:

$$f : \mathbf{M}(t) \rightarrow x_j(t) \quad (7)$$

C. DELAY MODULE

There are different nerve conduction delays when nerve impulses are transmitted away from the neuron along the nerve fiber or axon. In this model, there is a delay τ_1 from the spinal cord system to the arm plant and another delay τ_2 from the arm plant to the IO.

To account for the conduction delay, the eligibility trace is used which simulates a short period of memory process [23]. The concept of the eligibility trace was originally proposed by Klopf from the perspective of cognitive science. Recently, Yagishita [24], He [25] and Brzosko [26] have confirmed the existence of the eligibility trace in the striatum, cortex and hippocampus. When a state is visited, the eligibility trace is used to record the credit of the state, and the value of the credit

decreases over time. When returning the reward value, if the value of the eligibility trace corresponding to the state is not zero, the state is given a certain degree of credit according to its eligibility trace.

In our model, the following replacing trace is used

$$e_s(t) = \begin{cases} \zeta \lambda e_s(t-1) & s \neq s_t \\ 1 & s = s_t \end{cases} \quad (8)$$

where $\zeta(0 \leq \zeta \leq 1)$ is the discount-rate parameter and $\lambda(0 \leq \lambda \leq 1)$ is the decaying parameter. Usually, both of them are in the range between 0 and 1. s_t represents the actual state at time t , $e_s(t)$ and $e_s(t-1)$ are the traces for state s at times t and $t-1$.

$e_s(t)$ is calculated using

$$e_s(t) = \frac{1.088t - 0.9852}{t^2 - 2.066t + 2.099} \quad (9)$$

D. SPINAL/MUSCLE SYSTEM

To carry out a movement, it is necessary to identify what each joint actuator should do at each time instant under certain set of conditions. The spinal/muscle system model is used to convert the motor command from the cerebellum into arm movements. Each basic unit acts alone to generate a specific movement of the arm in the whole workspace. The arm plant trajectory is therefore the result of simultaneous execution of movements generated by multiple basic units.

Let $\mathbf{M}(t)$ and $\Delta\mathbf{M}(t)$ denote the position of the end-effector at time t and its adjustment from t to $t+1$. Simplifying assumptions are introduced that basic unit activity has an instantaneous and linear effect on changes in joint angles. Therefore, we have

$$\Delta\mathbf{M}(t) = \mathbf{F} \cdot \mathbf{O}(t) \quad (10)$$

where \mathbf{F} is a weighting matrix of 3 rows and d columns that summarizes the influence of the d basic units on the end-effector. It represents the mapping between the motor command space and arm’s Cartesian position space. Each column is a weight vector implying how the corresponding basic unit influences the end-effector. \mathbf{F} does not vary with \mathbf{M} means that each basic unit moves the end-effector in a particular direction in the Cartesian space. The output $\mathbf{O}(t)$ is a vector which can be expressed as $\mathbf{O}(t)=[O_1(t), O_2(t), \dots, O_d(t)]^T$. The position of the end-effector for the next time instant would be

$$\mathbf{M}(t+1) = \mathbf{M}(t) + \Delta\mathbf{M}(t) \cdot e_s(\tau) \quad (11)$$

where τ is a constant representing the time delay.

E. ROBOTIC ARM PLANT

The robotic arm platform is a six degree-of-freedom (DOF) haptic device manufactured by Geomagic Touch(Touch, 3D Systems Inc., America) (Fig.4). The first three joints are equipped with driving electrodes and sensors, while the end three joints only have sensing devices. The data communication is realized through the Ethernet Adapter/USB port and

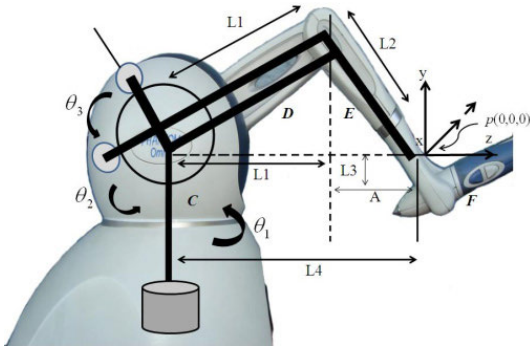


FIGURE 4. Schematic diagram of the Geomagic Touch robotic arm [20].

it allows real-time programming through class ToolKit 3D Touch to work with Visual Studio 2017 [27]. The problem of solving for the relationship between the joint coordinates $\theta(t) = [\theta_1(t), \theta_2(t), \theta_3(t)]^T$ and position of the end-effector $M(t)$ can be summarized as the inverse kinematics [28], [29].

The inverse position kinematics model (IPKM) is used to obtain the joint coordinates $\theta(t)$ as the function of the operational position of $M(t)$, which is expressed as follows:

$$\theta(t) = \begin{bmatrix} -a \tan 2(x, z + L_4) \\ \rho + \psi \\ \rho + \psi + \beta \end{bmatrix} \quad (12)$$

here, we have

$$\begin{aligned} \rho &= \cos^{-1}\left(\frac{L_1^2 + \sigma^2 - L_2^2}{2L_1\sigma}\right), \\ \psi &= a \tan 2(y - L_3, \Delta), \\ \beta &= \cos^{-1}\left(\frac{L_1^2 + L_2^2 - \sigma^2}{2L_1L_2}\right), \\ \sigma &= \sqrt{x^2 + (z + L_4)^2 + (y - L_3)^2}, \\ \Delta &= \sqrt{x^2 + (z + L_4)^2}. \end{aligned} \quad (13)$$

where, $L_1 = L_2 = 135$ mm, $L_3 = 25$ mm and $L_4 = 170$ mm. The driving force of Geomagic Touch is as follows:

$$F = SM - CV \quad (14)$$

among them, F represents the force vector generated by the Geomagic Touch, M and V represents the displacement and velocity vectors of arm respectively. S is the stiffness coefficient and C is the damping coefficient.

F. INFERIOR OLIVARY MODULE

The CF originating from the inferior olive forms an excitatory synapse with PF. It provides feedback information for the cerebellar model to correct the arm movement. The feedback information is used to adjust the memory information, i.e. the weight of the PC-PF synapse. Lots of experimental results showed that the inferior olive-climbing fiber system projecting to the cerebellum plays a critical role in basic associative learning and memory [30]–[33]. For example, anatomical studies on multi-joint limb movements in

monkeys by Thach [34] had shown that it might provide the “reinforcing” input to the cerebellum [35]. As a result, the function of the inferior olive module is implemented in the way of RL as follows.

(1) The state s_t of the cerebellar model is determined based on the arm position. The initial value s_t of all PFs in each basic unit $x_j(0)$ is set to 0. The behavior selection policy under the initial conditions π_0 is

$$\pi_0(a) = pr \{a_0 = a\} = \frac{1}{m} \quad (15)$$

where m is the total number of PFs, that is, the initial probability of choosing each behavior a is the same.

(2) Then, if the action a_t is selected at time t , the next state s_{t+1} and the next reward r_{t+1} are obtained. The value function $V(s_t)$ is updated using

$$V(s_t) = V(s_t) + \alpha[r_{t+1} + \gamma V'(s_{t+1}) - V(s_t)] \quad (16)$$

where α is a constant coefficient, γ is the discount factor. The reward r_{t+1} is determined by ΔD , which is the distance of the current arm position to the target position compared with that of the previous moment. If it is small, the reward r_{t+1} is set to be 0, otherwise be -1 . The value function $V(s_t)$ at time t and the estimated value function $V'(s_{t+1})$ at time $t + 1$ are defined by

$$V(s_t) = E_{\pi_t(a)}\left\{\sum_{i=0}^{\infty} \gamma^i r_{t+i+1} | s_t = s\right\} \quad (17)$$

$$V'(s_{t+1}) = E_{\pi_t(a)}\left\{\sum_{i=0}^{\infty} \gamma^i r_{t+i+1} | s_{t+1} = s\right\} \quad (18)$$

where $E_{\pi_t(a)}\{\}$ represents the expected value of the return value through the policy π_t when behavior a is selected.

(3) Next, the temporal difference (TD) error is calculated using (19), and the behavior selection probability and policy are updated according to

$$\delta_t = r_{t+1} + \gamma V'(s_{t+1}) - V(s_t) \quad (19)$$

$$p(s_t, a_t) = p(s_t, a_t) + \chi \delta_t \quad (20)$$

$$\pi_t(a) = pr \{a_t = a | s_t = s\} = \frac{e^{p(s_t, a_t)}}{\sum_{b=1}^Z e^{p(s_t, b)}} \quad (21)$$

where χ is a step parameter, Z represents the number of optional actions, $p(s_t, a_t)$ denotes the propensity to choose behavior a under the state s at time t .

(4) Afterwards, the evaluative information $c(t)$ that CF transfers is calculated based on δ_t using

$$c(t) = \begin{cases} 1 & \delta_t \geq 0 \\ 0 & \delta_t < 0 \end{cases} \quad (22)$$

(5) Finally, the weight $w_{kij}(t)$ is adjusted according to $c(t)$ via (23):

$$w_{kij}(t) = w_{kij}(t - 1) - \mu \cdot c(t) \cdot x_j(t) \cdot e_s(\tau) \quad (23)$$

where μ is a positive constant. Then the above five steps are repeated until the output of all the basic units $O(t)$ are 0,

which means the end of the cerebellar command generation. After that, ΔD is calculated to compare with a threshold. If ΔD is larger, the cerebellum continues producing commands to control arm according to (1)-(5), otherwise, terminates the whole control process.

IV. SIMULATION RESULTS

The validity and efficiency of the proposed cerebellar model were tested with several simulation experiments. To get an intuitive understanding of the control process, Emergent 8.5.6 (University of Colorado, America), a comprehensive neural network simulator, is first used to illustrate how the cerebellar neural network works to control a biophysically-realistic arm [36]. Then, the proposed cerebellar model was applied to a two-link arm to verify its efficacy in tracking control task to demonstrate the performance of the proposed cerebellar model over the CMAC approach. Finally, a study using a navigation task is also included in this section to demonstrate the disturbance performance of the cerebellar model.

A. EMERGENT SIMULATION EXPERIMENT

To provide a realistic test of our model, Emergent is used to construct a detailed model of the cerebellum and human arm. The arm has 4 degrees of freedom (3 degrees of freedom for the shoulder joint and 1 degree of freedom for the elbow joint) including 12 major muscle groups that can control arm movements as shown in Fig.5. The red, cyan and green balls represent the shoulder joint, the muscle insertion point and the desired position respectively. The blue parts stand for the hand, forearm, upper arm and torso. The muscle groups are attached to different positions of the shoulder joint, upper arm, forearm and hand. We can see the arm reaching over toward the green target. On the first reach, note how the hand overshoots the target a bit (Fig. 5(a)) and then it comes back down closer to the target (Fig. 5(b)).

Fig. 6 shows the activation of neurons when the arm exceeds or reaches the target position with the colored squares represent neurons. The neural network implemented in our model consists of three parts, the input layers, the hidden layers and the output layer. The input layers receive sensory input typically via desired length (TL), current length (L) and velocity (V) of each muscle and target position (TP), current position (HP) and velocity (HV) of the hand. The hidden layers are so called because they do not directly receive sensory input, nor do they directly drive motor output. They play an important role in the classification and processing of information from the input layer, which mainly containing MFs, GCs, GOs, PFs, IOs, BCs, SCs. The output layer PCs have neurons that synapse directly onto muscle control areas and is capable of causing physical movement. The basic cell is a neuron-like unit that stands for a neuron whose output is usually a time-continuous value ranging from 0 to 1. The higher the activation of neurons is, the brighter the color would be. If the output of a neuron is 0, it is not activated and

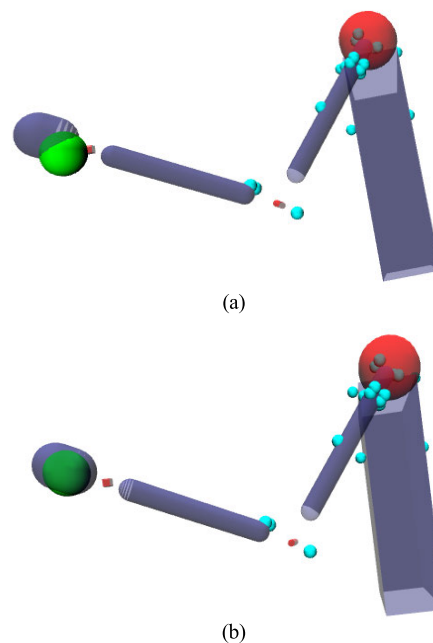


FIGURE 5. Simulation arm (a) arm is beyond target (b) arm reaches target.

colored in white. On the other hand, an output of 1 represents highly activated and colored in yellow.

From Fig. 6(a), we can see as the hand overshoots the target, a subset of the units in the IO layer get activated, enabled by the blue hand position. After correction, the arm is brought to the target position, as shown in Fig. 6(b). At this time, the activations of each muscle in the layer L and TL are consistent. The same applies to the layers of the HP and TP. However, when the arm exceeds the target, these layers are largely different from the target layers as shown in Fig. 6(a).

B. TWO LINK ARM TRACKING CONTROL SIMULATION

A simulated two-link planar arm is used in this simulation experiment to compare the performance of our proposed bionic cerebellum model (BCM) and conventional CMAC in the tracking task. The weight and muscle model of the two-link arm are neglected to simplify the calculation in the simulation. To obtain a reliable conclusion, 30 sets repetitive tracking control experiments were conducted. The two joint trajectories of the two-link arm and the tracking time of the two model control methods were recorded. The methods were developed in MATLAB 2018a (MathWorks, America) on a computer with Intel Core CPU i5-8250U processor of 1.6GHz and 4G of RAM.

In the simulation, the objective is to have the arm joints follow a particular trajectory. The tracking reference mode of the joints of the two-link arm was set to be a prescribed sinusoidal trajectory. Simulation results of the joint angle tracking responses are shown in Fig.7. The solid lines are the reference tracking trajectories of the arm joint angles. The dashed lines and the dotted lines are the joint angles produced by our BCM model and the traditional CMAC model respec-

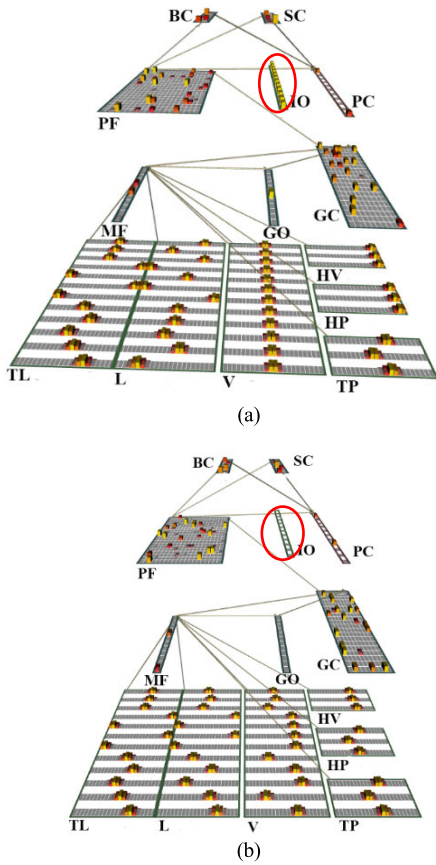


FIGURE 6. The cerebellar neural network. (a) arm is beyond target (b) arm reaches target. TL: the desired length of each muscle, L: the current length of each muscle, V: the velocity of each muscle, TP: the target position of hand, HP: the current position of hand, HV: the velocity of hand, PF: the parallel fiber layer, PC: the Purkinje cell layer, GC: the granule cell layer, IO: the inferior olive layer, MF: the mossy fiber layer, Go: Golgi cell layer, SC: the stellate cell layer, BC: the basket cell layer.

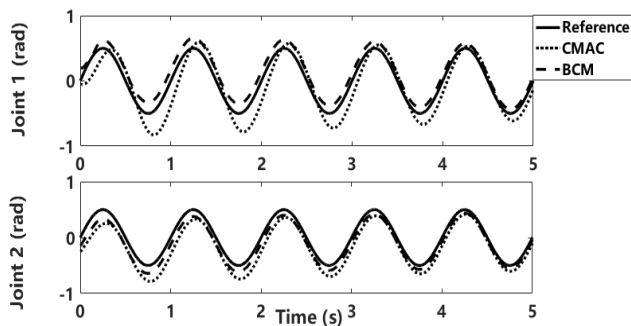


FIGURE 7. The two-link arm joints tracking curves.

tively. From this figure, we can see both control methods are stable, but the BCM has a better tracking accuracy than the CMAC through eliminating the tracking errors. Moreover, the average running time decreases from 25.06 s to 13.64 s when replacing the CMAC model with our BCM model. We could reasonably conclude that our BCM has a faster response speed and higher tracking accuracy.

C. ROBOTIC ARM NAVIGATION CONTROL SIMULATION

To further verify the control effect of the cerebellar model, a simulation experiment was designed to control the simulated two-link arm to move from an initial position to a target position. The control performance with different basic units ($d = 6, d = 10$) were measured to identify the effect of the number of basic unit. The 2-norm of each column of F was set to be 0.2 mm. First, the initial position of the endpoint (30, 40, 20) and target position (-30, -40, -20) are set to be same and thirty trials were conducted. According to Eq. (11), the movement trajectory of the endpoint and the curve of the position difference when the cerebellar model implemented with 6 basic units ($d = 6$) in one trial are shown in Fig. 8. The state of CF in each basic unit computed by Eq. (22) with execution times is shown in Fig.9. For the case of 10 basic units, the results are shown in Fig.10 and Fig. 11 respectively.

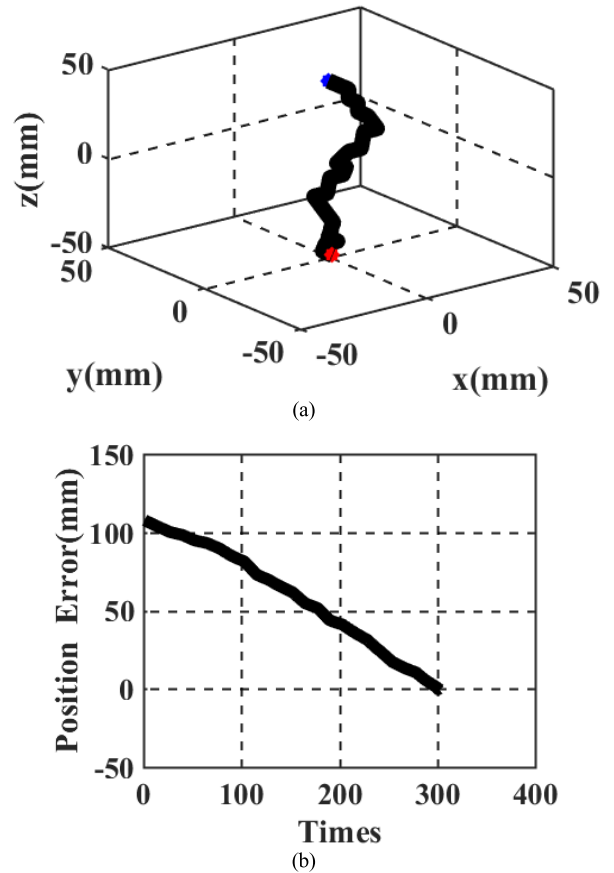


FIGURE 8. The control effect of the cerebellar model with 6 basic units. (a) The trajectory of the endpoint. The initial position of the endpoint is (30, 40, 20) and the desired position is (-30, -40, -20). The red pot and blue dot represent the desired position and initial position respectively. (b) The position error changes with the execution times.

From Fig. 8 and Fig. 10, we can see that during learning, both cerebellar models can constantly adjust the direction of the simulated robotic arm to make it move toward the target position, and finally reach the predetermined target position. All the errors are controlled within 1 mm which proves the

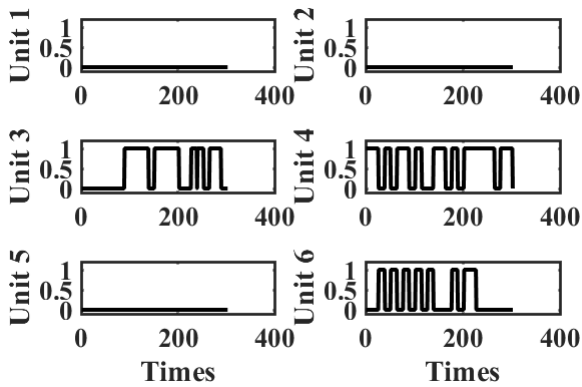


FIGURE 9. The state of CF in each basic unit with execution times for the cerebellar model for the cerebellar model with 6 basic units.

effectiveness of the proposed cerebellar model. Moreover, the results show that the average distance errors for the cerebellar model with 6 and 10 basic units are 0.405 ± 0.137 mm and 0.363 ± 0.197 mm. The use of more basic units appears to lead to less position error. As a result, unilateral *t*-test was applied to investigate the influence of the number of cerebellar model on the task performance. The result showed that there was no significant difference for the position error ($t = 0.944 < 1.65 = t_{0.05}(58)$).

As the number of basic units in the cerebellum increases, there are more strategic directions for the cerebellum to choose, so the number of execution times will increase. The results show that the average execution times for the cerebellar model with 6 and 10 basic units are 352 ± 31 and 368 ± 63 . Unilateral *t*-test result showed that there was significant difference for the execution times ($t = 1.19 < 1.65 = t_{0.05}(58)$). The #3 basic unit (negative X-axis), #4 basic unit (negative Y-axis) and #6 basic unit (negative Z-axis) are more active throughout the learning process since the movement direction of the arm is from the first quadrant to the seventh quadrant as we can see in Fig. 9. The mapping direction of these several basic units is related to the movement direction, so it is activated, and its state value is 1. The inactive or rarely activated basic units, such as #1 (positive X-axis), #2 (positive Y-axis) and #5 (positive Z-axis) are scarcely activated because their mapping direction is opposite to the movement direction. As shown in Fig. 11, expect the #3, #4, #6 basic units, the #8 basic units is also active because it represents the direction vector $(-0.14; -0.14; 0)$ which plays an important role in the motion.

To further investigate the model behavior, we evaluated its performances under unexpected disturbance. At a certain point, an unexpected external force is applied, which will produce a perturbation in the performed the trajectory. We evaluated how the different basis units tuning affected the performances in a noisy environment and how the BCM model deal with the perturbation. For convenience of observation, the trajectory that the arm endpoint moves is in the Y-Z plane (Fig. 12, solid line). When an unexpected perturbation

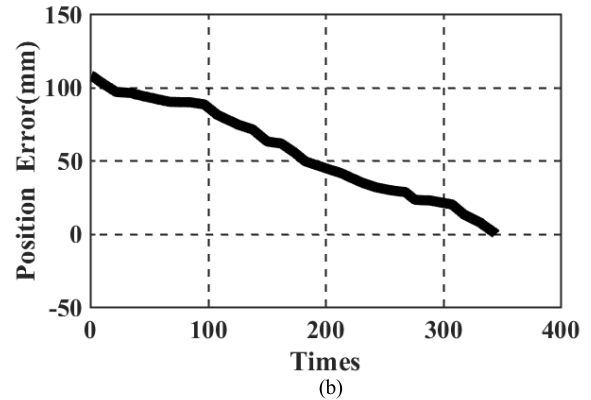
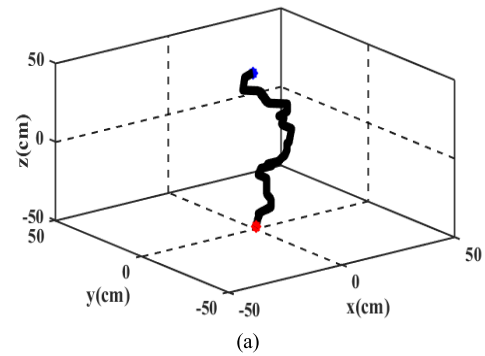


FIGURE 10. The control effect of the cerebellar model (with 10 basic units). (a) The trajectory of the end-effector. The initial position of the end-effector is (30, 40, 20) and the desired position is (-30, -40, -20). The red pot and blue dot represent the desired position and initial position. (b) The change of the position difference with the execution times.

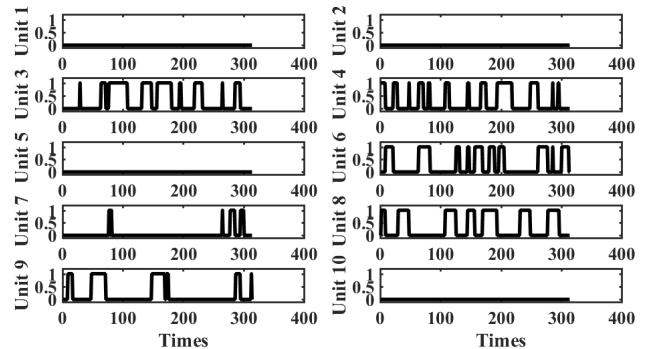


FIGURE 11. The state of CF in each basic unit with execution times for the cerebellar model with 10 basic units.

is virtually added to the arm, the trajectory deviates from the desired one, being deformed toward the left (Fig. 12, dotted line). Note that the trajectory of the arm endpoint was changed when it reached to (0, 10, 20) mm.

Examining the results more closely, as reflected by Fig. 12, when the perturbation appears, reveals that our cerebellar model immediately reacts to the errors. It could rapidly converge in reducing the tracking bias and successfully guide the arm to the target through their online tuning ability.

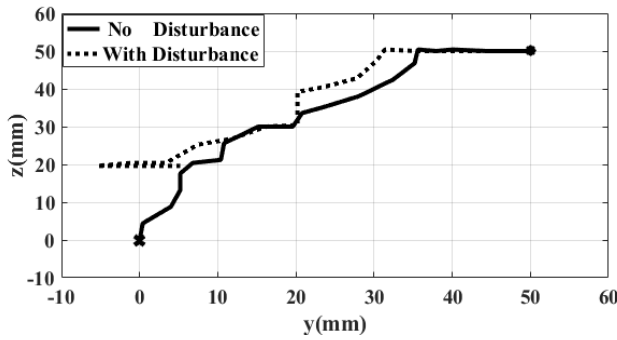


FIGURE 12. Simulation results of position response with disturbance.



FIGURE 13. The running experimental platform of arm control.

V. EXPERIMENT RESULTS

In this study, the proposed cerebellar model was implemented with 10 basic units in a robotic arm, the Geomagic Touch. The bionic arm motion (BCM) control platform is shown in Fig. 13. Four set of trials with different ending points were carried out. In each experiment, the end-effector path is traced and the resulting trajectory in one trial are depicted in Fig. 14. The solid line is the end-effector trajectory with the BCM control model proposed in this paper, and the dotted line stands for the result from CMAC. The star and prism denote the final position for BCM and CMAC respectively, and the starting position is the origin. The results indicate that the end-effector of the robotic arm can accurately reach the target position, and all the trajectories of the BCM are much smoother than those from CMAC, which proves the validity of the proposed cerebellar model.

In addition, each set of the trials were repeated 20 times to calculate the average distance error and target position error. The performance metrics evaluated for this test are summarized in Table 2. Paired *t*-test was applied to investigate the influence of control method on the task performance. The results are significant with $p < 0.05$. The results on the position error revealed that the BCM can control the end-effector to the target position more accurately than the CMAC ($t = 8.94 > t(79)_{0.05} = 1.65$). Regarding the trajectory distance,

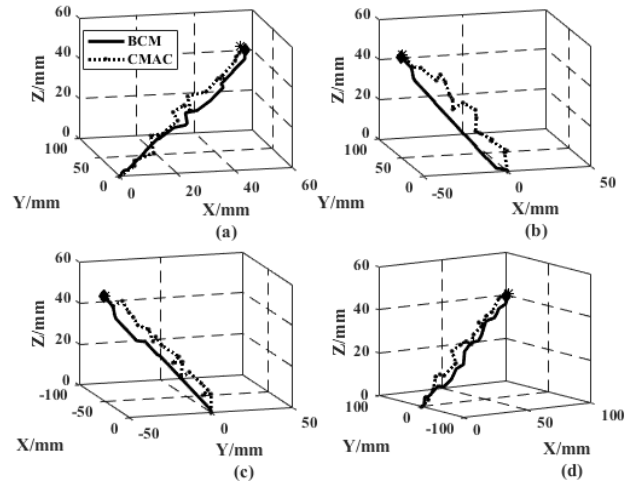


FIGURE 14. Trajectory of Geomagic Touch device controlled by cerebellar model. The start and prim denote the target position of the BCM and the CMAC respectively. The solid and dot curve stands for the movement trajectories of the end-effector. The starting positions of all are the origin. The target point is (a) (50,50,50), (b) (-50,50,50), (c) (-50,-50,50) and (d) (50,-50,50).

TABLE 2. Control performance comparison with BCM and CMAC.

trial set No.	method	average position error (std) (mm)	average distance (std) (mm)
1	BCM	2.91 ± 0.69	87.34 ± 2.11
	CMAC	3.44 ± 0.97	97.57 ± 4.14
2	BCM	2.15 ± 0.33	87.24 ± 1.55
	CMAC	4.99 ± 1.14	101.10 ± 2.86
3	BCM	3.52 ± 0.68	86.13 ± 2.71
	CMAC	5.32 ± 0.98	103.91 ± 6.34
4	BCM	4.16 ± 0.69	85.33 ± 1.81
	CMAC	5.26 ± 0.83	98.37 ± 3.32

there was also a statistical difference between the BCM and the CMAC ($t = 22.6 > t(79)_{0.05} = 1.65$).

VI. CONCLUSION AND FUTURE WOK

The integration of brain science and intelligent technologies will promote the breakthrough and development of brain-like intelligence research, and the understanding of human brain cognitive neural mechanism can bring new enlightenment to the research of artificial intelligence algorithm. This paper takes the cerebellum as the research object, and deeply studies the microscopic structure and mechanism of the inner neurons of the cerebellar cortex. On the basis of the neural computational method, a cerebellar learning model with reinforcement learning was proposed at the neuron level for the control and regulation of the arm movement. In our model, the Purkinje cells are assumed to generate arm-motion commands, and climbing fibers provide the reward or punishment information. Then, the neural computing simulation software Emergent was used to establish the cerebellar neural network and the simulated arm model. The simulation results show that the system can complete the predetermined control tasks, and the motion trajectory is smooth. With the increase of the

number of learning, the position error of the arm to the target position is gradually reduced, indicating the effectiveness of the proposed cerebellar model. At last, the cerebellum model was transplanted into the Geomagic Touch bionic arm robot control system. Experimental tests also showed that the cerebellar model can achieve precise and stable control of the robotic arm.

In this study, we compared the control performance with the cerebellar model using 6 and 10 basic units. The results showed that both could achieve target reaching tasks. Using more basic units can improve the accuracy, but its stability becomes worse and the learning time is longer. This is in accordance with the phenomenon of redundancy in motion control. We assume that this suggests that the central nervous system may reduce the computational complexities of motor control through driving muscle synergies rather than a muscle [37]–[39]. Along this line of consideration, it is supposed that each basic unit generates a specific motor output by selecting a specific pattern of muscle activations. In future work, we will build a more physiologically plausible cerebellar model with the theory of muscle synergy to cover the movement planning with more degrees of freedom which may be useful in the development of general-purpose motor learning systems for machines.

Moreover, combining the cerebellum model with other simulated brain region could yield a more complex and capable model. Of particular interest is the basal ganglia, which is assumed to perform reinforcement learning. Therefore, another important line of work will involve incorporating a wider range of brain regions into computer simulations, as it will not only constrain certain aspects of these models but will allow for the development more sophisticated cerebellar models.

ACKNOWLEDGMENT

The authors would like to thank the technical editor and anonymous reviewers for their valuable and constructive comments.

REFERENCES

- [1] L. N. Miterko, K. B. Baker, J. Beckinghausen, L. V. Bradnam, M. Y. Cheng, J. Cooperrider, M. R. DeLong, S. V. Gornati, M. Hallett, and D. H. Heck, "Consensus paper: Experimental neurostimulation of the cerebellum," *Cerebellum*, vol. 18, no. 6, pp. 1064–1097, Jun. 2019.
- [2] X. Chen, N. Wang, H. Cheng, and C. Yang, "Neural learning enhanced variable admittance control for Human–Robot collaboration," *IEEE Access*, vol. 8, pp. 25727–25737, Jan. 2020.
- [3] J. C. Eccles, M. Ito, and J. Szentágothai, *The Cerebellum as a Neuronal Machine*. Berlin, Germany: Springer, 1967. Accessed: Jan. 1, 1967. [Online]. Available: <https://citations.springer.com/book?doi=10.1007/978-3-662-13147-3>
- [4] D. Marr, "A theory of cerebellar cortex," *J. Physiol.*, vol. 202, no. 2, pp. 437–470, Jun. 1969.
- [5] J. S. Albus, "A theory of cerebellar function," *Math. Biosci.*, vol. 10, nos. 1–2, pp. 25–61, Feb. 1971.
- [6] P.-Y. Chen, K.-H. Chao, and Y.-C. Tseng, "A motor fault diagnosis system based on cerebellar model articulation controller," *IEEE Access*, vol. 7, pp. 120326–120336, Aug. 2019.
- [7] T.-T. Huynh, C.-M. Lin, T.-T.-T. Pham, H.-Y. Cho, and T.-L. Le, "A modified function-link fuzzy cerebellar model articulation controller using a PI-type learning algorithm for nonlinear system synchronization and control," *Chaos, Solitons Fractals*, vol. 118, pp. 65–82, Jan. 2019.
- [8] T.-T. Huynh, C.-M. Lin, T.-L. Le, H.-Y. Cho, T.-T.-T. Pham, N.-Q.-K. Le, and F. Chao, "A new self-organizing fuzzy cerebellar model articulation controller for uncertain nonlinear systems using overlapped Gaussian membership functions," *IEEE Trans. Ind. Electron.*, vol. 67, no. 11, pp. 9671–9682, Nov. 2020.
- [9] J. Guan, C.-M. Lin, G.-L. Ji, L.-W. Qian, and Y.-M. Zheng, "Robust adaptive tracking control for manipulators based on a TSK fuzzy cerebellar model articulation controller," *IEEE Access*, vol. 6, pp. 1670–1679, Dec. 2018.
- [10] T. L. Le, T. T. Huynh, V. Q. Nguyen, C. M. Lin, and S. K. Hong, "Chaotic synchronization using a self-evolving recurrent interval type-2 Petri cerebellar model articulation controller," *Mathematics*, vol. 8, no. 2, pp. 1–15, Feb. 2020.
- [11] M. Fujita, "Adaptive filter model of the cerebellum," *Biol. Cybern.*, vol. 45, no. 3, pp. 195–206, 1982.
- [12] M. Ito, "A new physiological concept on cerebellum," *Revue Neurologique*, vol. 146, no. 10, pp. 564–569, Nov. 1990.
- [13] D. M. Wolpert, R. C. Miall, and M. Kawato, "Internal models in the cerebellum," *Trends Cognit. Sci.*, vol. 2, no. 9, pp. 338–347, Sep. 1998.
- [14] N. E. Berthier, S. P. Singh, A. G. Barto, and J. C. Houk, "Distributed representation of limb motor programs in arrays of adjustable pattern generators," *J. Cognit. Neurosci.*, vol. 5, no. 1, pp. 56–78, Jan. 1993.
- [15] A. Pellionisz, "Tensorial aspects of the multidimensional massively parallel sensorimotor function of neuronal networks," *Prog. Brain Res.*, vol. 76, pp. 341–354, Jan. 1988.
- [16] R. A. Swain, A. L. Kerr, and R. F. Thompson, "The cerebellum: A neural system for the study of reinforcement learning," *Frontiers Neurosci.*, vol. 5, no. 8, pp. 1–6, Mar. 2011, doi: [10.3389/fnbeh.2011.00008](https://doi.org/10.3389/fnbeh.2011.00008).
- [17] A. S. Therrien, D. M. Wolpert, and A. J. Bastian, "Effective reinforcement learning following cerebellar damage requires a balance between exploration and motor noise," *Brain*, vol. 139, no. 1, pp. 101–114, Jan. 2016.
- [18] T. Yamazaki and W. Lennon, "Revisiting a theory of cerebellar cortex," *Neurosci. Res.*, vol. 148, pp. 1–8, Nov. 2019.
- [19] M. Hausknecht, W.-K. Li, M. Mauk, and P. Stone, "Machine learning capabilities of a simulated cerebellum," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 3, pp. 510–522, Mar. 2017.
- [20] S. Kitazawa, T. Kimura, and P.-B. Yin, "Cerebellar complex spikes encode both destinations and errors in arm movements," *Nature*, vol. 392, no. 6675, pp. 494–497, Apr. 1998.
- [21] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *J. Artif. Intell. Res.*, vol. 4, pp. 237–285, Apr. 1996.
- [22] B. Kiumarsi, K. G. Vamvoudakis, H. Modares, and F. L. Lewis, "Optimal and autonomous control using reinforcement learning: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2042–2062, Jun. 2018.
- [23] S. P. Singh and R. S. Sutton, "Reinforcement learning with replacing eligibility traces," *Mach. Learn.*, vol. 22, nos. 1–3, pp. 123–158, 1996.
- [24] S. Yagishita, A. Hayashi-Takagi, G. C. R. Ellis-Davies, H. Urakubo, S. Ishii, and H. Kasai, "A critical time window for dopamine actions on the structural plasticity of dendritic spines," *Science*, vol. 345, no. 6204, pp. 1616–1620, Sep. 2014.
- [25] K. He, M. Huertas, S. Z. Hong, X. Tie, J. W. Hell, H. Shouval, and A. Kirkwood, "Distinct eligibility traces for LTP and LTD in cortical synapses," *Neuron*, vol. 88, no. 3, pp. 528–538, Nov. 2015.
- [26] Z. Brzosko, S. Zannone, W. Schultz, C. Clopath, and O. Paulsen, "Sequential neuromodulation of Hebbian plasticity offers mechanism for effective reward-based navigation," *eLife*, vol. 6, Jul. 2017, Art. no. e27756.
- [27] A. J. Silva, O. A. D. Ramirez, V. P. Vega, and J. P. O. Oliver, "PHANToM OMNI haptic device: Kinematic and manipulability," in *Proc. Electron., Robot. Automot. Mech. Conf. (CERMA)*, Cuernavaca, Mexico, Sep. 2009, pp. 193–198.
- [28] T. Sansanayuth, I. Nilkhamhang, and K. Tungpimolrat, "Teleoperation with inverse dynamics control for PHANToM Omni haptic device," in *Proc. SICE Annu. Conf.*, Akita, Japan, 2012, pp. 2121–2126.
- [29] P. M. Kebrria, A. Khosravi, S. Nahavandi, P. Shi, and R. Alizadehsani, "Robust adaptive control scheme for teleoperation systems with delay and uncertainties," *IEEE Trans. Cybern.*, vol. 50, no. 7, pp. 3243–3253, Jul. 2020, doi: [10.1109/TCYB.2019.2891656](https://doi.org/10.1109/TCYB.2019.2891656).
- [30] R. F. Thompson, J. K. Thompson, J. J. Kim, D. J. Krupa, and P. G. Shinkman, "The nature of reinforcement in cerebellar learning," *Neurobiol. Learn. Memory*, vol. 70, nos. 1–2, pp. 150–176, Jul. 1998.

- [31] F. Naveros, N. R. Luque, E. Ros, and A. Arleo, "VOR adaptation on a humanoid iCub robot using a spiking cerebellar model," *IEEE Trans. Cybern.*, vol. 50, no. 11, pp. 4744–4757, Nov. 2020, doi: 10.1109/TCYB.2019.2899246.
- [32] N. R. Luque, J. A. Garrido, R. R. Carrillo, O. J.-M.-D. Coenen, and E. Ros, "Cerebellarlike corrective model inference engine for manipulation tasks," *IEEE Trans. Syst., Man, Cybern., B, Cybern.*, vol. 41, no. 5, pp. 1299–1312, Oct. 2011.
- [33] P. L. Strick, "The cerebellum: The cerebellum and neural control," *Science*, vol. 229, no. 4713, p. 547, Sep. 1985.
- [34] W. T. Thach, H. P. Goodkin, and J. G. Keating, "The cerebellum and the adaptive coordination of movement," *Annu. Rev. Neurosci.*, vol. 15, no. 1, pp. 403–442, Mar. 1992.
- [35] J. Kim, "Cerebellar circuits and synaptic mechanisms involved in classical eyeblink conditioning," *Trends Neurosciences*, vol. 20, no. 4, pp. 177–181, May 1997.
- [36] B. Aisa, B. Mingus, and R. O'Reilly, "The emergent neural modeling system," *Neural Netw.*, vol. 21, no. 8, pp. 1146–1152, Oct. 2008.
- [37] T. Lencioni, J. Jonsdottir, D. Cattaneo, A. Crippa, E. Gervasoni, M. Rovaris, E. Bizzi, and M. Ferrarin, "Are modular activations altered in lower limb muscles of persons with multiple sclerosis during walking? Evidence from muscle synergies and biomechanical analysis," *Frontiers Hum. Neurosci.*, vol. 10, p. 620, Dec. 2016.
- [38] M. Kristiansen, A. Samani, P. Madeleine, and E. A. Hansen, "Muscle synergies during bench press are reliable across days," *J. Electromyogr. Kinesiol.*, vol. 30, pp. 81–88, Oct. 2016.
- [39] V. C. K. Cheung, A. Turolla, M. Agostini, S. Silvoni, C. Bennis, P. Kasi, S. Paganoni, P. Bonato, and E. Bizzi, "Muscle synergy patterns as physiological markers of motor cortical damage," *Proc. Nat. Acad. Sci. USA*, vol. 109, no. 36, pp. 14652–14656, Sep. 2012.



YARU CHEN is currently pursuing the master's degree with the Dalian University of Technology, Dalian, Liaoning, China. Her current research interests include biomechanics and neurorehabilitation engineering.



JIAXING WANG is currently pursuing the master's degree with the Dalian University of Technology, Dalian, Liaoning, China. His current research interests include cerebellar model, the applications of control theory, and digital signal processing.



RONG LIU (Member, IEEE) received the Ph.D. degree in electrical engineering from The Chinese University of Hong Kong, Hong Kong, China.

She was an Assistant Professor with the Department of Biomedical Engineering, Dalian University of Technology, Dalian, Liaoning, China, from 2006 to 2011, and an Associate Professor, since 2011. From 2010 to 2011, she was also a Research Scientist with the Neuroengineering Laboratory, Department of biomedical engineering, School of Medicine, Johns Hopkins University. She is the author of more than 40 articles and holds five patents. Her research interests include biomedical signal processing and rehabilitation engineering.



QI ZHANG is currently pursuing the master's degree with the Dalian University of Technology, Dalian, Liaoning, China. His current research interests include cerebellar model and the applications of control theory.



LE YANG (Member, IEEE) received the B.S. degree in telecommunication engineering and the M.S. degree in signal and information processing from the University of Electronic Science and Technology of China (UESTC), in 2000 and 2003, respectively, and the Ph.D. degree in electrical and computer engineering (ECE) from the University of Missouri, Columbia, MO, USA, in 2010. From 2003 to 2004, he was a Lecturer with UESTC. From 2004 to 2006, he was a Research Assistant

with the University of Victoria (UVic), working on diversity techniques, performance analysis, and ultra wideband (UWB) communications. From 2006 to 2007, he was with the McMaster University, working on the computational modeling of auditory cortex for studying the neural substrates of tinnitus. From 2011 to 2018, he was an Associate Professor with Jiangnan University, Wuxi, China. In 2019, he joined the University of Canterbury, Christchurch, New Zealand, as a Senior Lecturer with the Department of ECE. His research interests include statistical signal processing with applications to localization and tracking, machine learning, and engineering optimization.

...