A Simplified Cerebellar Model with Priority-based Delayed Eligibility Trace Learning for Motor Control

Vui Ann Shim, Chris Stephen Naveen Ranjit, Bo Tian, Miaolong Yuan, and Huajin Tang

Abstract—The study of cerebellum has resulted in a common agreement that it is implicated in motor learning for movement coordination. Learning governed by error signal through synaptic eligibility traces has been proposed to be a learning mechanism in cerebellum. In this paper, we extend this idea and suggest a simplified and improved cerebellar model with priority-based delayed eligibility trace learning rule (S-CDE) that enables a mobile robot to freely and smoothly navigate in an environment. S-CDE is constructed in a brain-based device which mimics the anatomy, physiology, and dynamics of cerebellum. The input signal in terms of depth information generated from a simulated laser sensor is encoded as neuronal region activity for velocity and turn rate control. A priority-based delayed eligibility trace learning rule is proposed to maximize the usage of input signals for learning in synapses on Purkinje cell and cells in the deep cerebellar nuclei of cerebellum. Error signal generation and input signal conversion algorithms for turn rate and velocity are designed to facilitate training in an environment containing turns of varying curvatures. S-CDE is tested on a simulated mobile robot which had to randomly navigate maps of Singapore and Hong Kong expressways.

Index Terms—Brain-based devices, cerebellum, synaptic eligibility trace, error signals, motor control, motor learning

I. INTRODUCTION

The study of cerebellum in neuroscience, neurophysiology, and neuroimaging has resulted in a commonly accepted fact that implicates the cerebellar system in motor learning for movement coordination [1]–[3]. A plethora of experiments [4], such as the study of vestibuloocular reflex [5], [6], optokinetic eye movement [7], saccadic eye movement [8], arm movements [9], cursor tracking [10], eye blink conditioning [11], etc., has been carried out to explain the neuroanatomy, dynamics, and functionality of cerebellum. These studies have motivated the development of cerebellum-based computational models with the aim to verify the observations in neuroscience and other related experiments, to provide a new sort of brain study through simulated brain models, or to develop intelligent robots with self-learning behaviors [12]–[17].

It has been proposed that cerebellum learns reflex movement coordination by a predictive controller [18]. Specifically, error signal, which is the reflexive motor commands, has been suggested to regulate the predictive control or learning in the cerebellar system through synaptic eligibility traces. The error signals are transmitted from the inferior olive (IO) to cerebellar regions via climbing fibers [19]. With the predictive controller, the cerebellum produces motor control commands earlier than the reflex responses.

Inspired by the above findings, McKinstry et al. [20] proposed a cerebellum-based computational model with delayed eligibility trace learning rule (here dubbed as CDE) that learns to predict corrective motor control actions based on the experiences of reflex responses. This learning rule determines the eligibility of connections between neuronal areas to be plastic or non-plastic. It is inspired by the motor timing studies which suggested that synaptic eligibility traces in the cerebellum are triggered by motion onset [18]. The eligibility of plastic connections is determined by suprathreshold presynaptic activity after a fixed delay. The strength of the plastic connections is subsequently changed in relation to the eligibility trace as well as error signal generated from IO. The change in the plastic connections results in a trained cerebellum that avoids the impending error signal, thus functions as a predictive controller.

CDE is developed in a brain-based device (BBD) [21]. BBD is a synthetic neural model which incorporates features of neuroanatomy, neurophysiology, and dynamic of vertebrates, and is embodied in robotic platforms. Specifically, CDE is suggested to govern the plasticity of synapses onto Purkinje cell (PC) and deep cerebellar nuclei (DCN) and is used to learn proper motor responses, given certain visual cues, such that error signals are avoided. The error signals are generated using inputs from IR sensors, to govern the learning in PC and DCN. Also, a laser range finder is used to detect oncoming collisions and initiate collision avoidance behavior. This model has been implemented in a Segway robot and tested in several curved courses. The results demonstrated a smooth traversal of curved paths, with each path comprised of similar turn curvatures.

However, CDE suffers from several limitations. First, the complexity of CDE, which consists of 28 neural areas, 27,688 neurons, about 1.6 million synaptic connections, and three input modalities, may limit its practicality on real machines. Second, CDE is inept at effective motor learning in environments with varied turn curvatures due to its limited utilization of synaptic inputs for learning. Third, there is a limit to the volatility of input stimuli, beyond which learning becomes impaired. This limitation is caused by an insufficient eligibility trace decay rate in the delayed eligibility trace learning rule.

Even through CDE suffers the aforementioned limitations and it was only tested in simple and short curved courses, its
potential to be extended to a complete self-learning navigation system [22]–[26] is very promising. First, CDE has adaptive self-learning capability with skills trained by error signal. With this capability, learning is possible in any environment. Second, the learning capability can be enhanced by simply increasing the size of the neuronal regions. This enables efficient learning in complex and large environments. Third, CDE in the BBD framework can be easily expanded by connecting cerebellum areas with other brain regions. This may allow the robot to have a higher cognitive capability.

With regard to the aforementioned strengths and limitations, we propose a simplified and improved cerebellar model with priority-based delayed eligibility trace learning rule (S-CDE). First, to simplify the model, we utilize a simulated laser sensor to generate environmental depth information. This is the only input modality. This dramatically reduces the complexity of the model since visual processing in CDE imposes extravagant complexity on the system compared to its cerebellar portion. Second, to improve the learning capability, a priority-based delayed eligibility trace learning rule is suggested to maximize the usage of input signals for synaptic learning on plastic connections to PC and DCN areas. This is done by introducing a mechanism to prematurely re-trigger eligibility traces upon encountering more salient synaptic inputs. Third, an increased eligibility trace decay rate is used to allow for increased input volatility. The proposed model is developed in a brain-based device and tested in a simulated mobile robot which had to randomly and smoothly navigate maps of the Singapore expressway, Hong Kong expressway, and oval running track. The training efficiency, adaptability in different environments, and generalization of learning behaviors in environments with different complexities are investigated.

The rest of the paper is organized as follows. Section II presents the literature review related to cerebellar models in robotic fields. The proposed modifications are given in Section III. Section IV describes the experimental details and simulation results. Finally, conclusion and future work are given in last section.

II. LITERATURE REVIEW

Cerebellum has been commonly accepted that it is implicated in motor learning or motor coordination [27], [28] even through recent findings argued that it could also involve in cognitive processes [29], [30] and emotional control [31], [32].

The earliest cerebellar models were based on Perceptron models proposed by Marr [33] and Albus [34]. These models assumed that PC cells fire complex spikes in response to climbing fibers and simple spikes in response to parallel fibers [9]. Since then, many models have been proposed based on the Marr and Albus’s models [35]–[37] with the aim to study the functional mechanism of cerebellum. An earlier review of cerebellar models and its functionality can be referred to [1].

In [9], computational models for four regions of cerebellum based on feedback error learning mechanism were proposed. The feedback was used as the error signal for motor learning. The error signal was controlled by a feedback controller which converts the trajectory error into motor command error. Similar to S-CDE, error signals are used to train the cerebellum and different regions of the cerebellum learn the predictive control for movement coordination. However, this model only investigates the functional roles of different regions of the cerebellum and does not take into account the motion control for robot navigation. In [38], a computational model for intermediate parts of cerebellum was constructed for limbs movement coordinations. This is an extension to Kawato’s model [9] in order to realize the smooth constrained motion. This model contributes to the use of cerebellar system for control of a single arm. However, the manipulation in this model does not contribute to a fine motion control for effective navigation.

In [39], a cerebellar model was constructed by integrate-and-fire spiking neurons for controlling robotic arm. The spike-timing dependent plasticity connection was established between parallel fiber and PC. Besides, the synaptic plasticity was driven by the activity in IO region. This concept is identical to the learning governed by error signal implemented in S-CDE, however different methods of error signal generation have been employed. Furthermore, delay was introduced in the sensorimotor pathways. This delay define the time of delivery of torques to the limb, thus is different from the delay concept in S-CDE.

In [40], a computational model was constructed to study the correlation between plasticity in cerebellar cortex and adaptive conditioned response timing. Similar to S-CDE, the synaptic plasticity of the cerebellar system is accounted as a main learning for motor control. However, this study only involves plasticity changes and does not take into account the eligibility of plasticity. In the implementation, only simple obstacle avoidance tasks are achieved by this model while S-CDE can learn for a more complex behaviour such as effective robot navigation.

CDE [20] is a cerebellar model that directly addresses the robot navigation problem. Due to its practicality in robotic platform as well as biologically plausible properties, CDE has gained our great interests. The characteristics, strengths, weaknesses, and suggestion of improvement of this model have been described in the Introduction section.

III. A SIMPLIFIED AND IMPROVED CEREBELLAR MODEL FOR MOTOR CONTROL

The proposed method consists of three significant differences from CDE. First, a simplified version with one input modality is proposed. Second, an improved learning rule using eligibility trace is suggested to enable an efficient learning in environments with different turn curvatures. Furthermore, the input volatility is increased for fast adaptation to salient inputs. Third, the conversion of input signal and the generation of error signal are modified to allow a better learning in environments with varied turn curvatures.

A. The System Architecture

The system architecture of a simplified cerebellar model with priority-based delayed eligibility trace learning rule (S-CDE) is presented in Fig. 1. S-CDE can be divided into a
sensory input layer, a preprocessing layer, a cerebellar layer, and an output motor layer. The laser sensor is the only input modality of S-CDE. This is one of the main differences compared to the CDE model which consists of three input modalities. This modification eliminates the need for extensive processing of visual cues from a camera, thus, reducing the computational complexity. Besides, the derivation of error signals from the laser input also eliminates the requirement of other input modalities such as infra-red sensors. Furthermore, the depth information from a laser scanner allows efficient navigation in a typical office environment [41].

The laser scans for objects in an environment. The output of the laser is processed in the preprocessing layer such that the input data is converted into four different streams (ES-Turn, ES-Velo, SC-Turn, and SC-Velo). Note that ES is error signal and SC is signal conversion, each in an appropriate semantic format for the sub-region it is being fed into. These signals are then fed into the cerebellar layer to trigger corresponding neuronal responses. The cerebellar layer can be further divided into symmetrical Turn and Velo regions for handling turn rate and velocity computations respectively.

In the cerebellar layer, the preprocessed sensory inputs are fed to the PN-Turn and PN-Velo areas. Besides, error signals are derived from a subset of the sensory input and are subject to preprocessing before being fed to IO regions (IO-Turn and IO-Velo). The turn and velocity regions are symmetrical to one another. PN areas (PN-Turn and PN-Velo) are linked to PC (PC-Turn and PC-Velo) and DCN (DCN-Turn and DCN-Velo) areas via plastic connections. PC controls DCN via disinhibition through its inhibitory connections, which in turn provide predictive control signals for turn rate and velocity to motor areas (Motor-Turn and Motor-Velo). Error signals from IO govern motor learning in the cerebellar regions (IO→PC and IO→DCN) and initially drive motor output in early

![System architecture](image_url)

**TABLE I: Properties of the projection in S-CDE.** Projection refers to the connection from a region to another region. Arborization (Arbor) is the projection shape. Noted that the projection probability of 1.0 implemented in the experiment. rect\((he,wi)\) is the arborization’s rectangular shape with a height (he) and width (wi) while nontopo gives any two neuronal units a probability of being connected. The initial synaptic weights (c) are uniformly generated between the maximum and minimum values. \(\alpha\) is the learning rate for plastic connections.

<table>
<thead>
<tr>
<th>Projection</th>
<th>Arbor</th>
<th>(c_{ij})</th>
<th>(\alpha)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ES-Turn→IO-Turn</td>
<td>rect(1,1)</td>
<td>1.0</td>
<td>0</td>
</tr>
<tr>
<td>IO-Turn→PC-Turn</td>
<td>rect(1,1)</td>
<td>(0.22,0.27)</td>
<td>0</td>
</tr>
<tr>
<td>IO-Turn→DCN-Turn</td>
<td>rect(1,1)</td>
<td>(0.50,0.55)</td>
<td>0</td>
</tr>
<tr>
<td>IO-Turn→Motor-Turn</td>
<td>rect(1,1)</td>
<td>(0.60,0.65)</td>
<td>0</td>
</tr>
<tr>
<td>SC-Turn→PC-Turn</td>
<td>rect(1,1)</td>
<td>1.0</td>
<td>0</td>
</tr>
<tr>
<td>PC-Turn→DCN-Turn</td>
<td>rect(1,1)</td>
<td>(-1.47,-1.42)</td>
<td>0</td>
</tr>
<tr>
<td>DCN-Turn→Motor-Turn</td>
<td>rect(1,1)</td>
<td>(0.38,0.43)</td>
<td>0</td>
</tr>
<tr>
<td>PN-Turn→PC-Turn</td>
<td>nontopo</td>
<td>(0.50,0.55)</td>
<td>-0.08</td>
</tr>
<tr>
<td>PN-Turn→DCN-Turn</td>
<td>nontopo</td>
<td>(0.50,0.55)</td>
<td>0.04</td>
</tr>
<tr>
<td>ES-Velo→IO-Velo</td>
<td>rect(1,1)</td>
<td>1.0</td>
<td>0</td>
</tr>
<tr>
<td>IO-Velo→PC-Velo</td>
<td>rect(1,1)</td>
<td>(0.21,0.26)</td>
<td>0</td>
</tr>
<tr>
<td>IO-Velo→DCN-Velo</td>
<td>rect(1,1)</td>
<td>(0.50,0.55)</td>
<td>0</td>
</tr>
<tr>
<td>IO-Velo→Motor-Velo</td>
<td>rect(1,1)</td>
<td>(0.60,0.65)</td>
<td>0</td>
</tr>
<tr>
<td>SC-Velo→PC-Velo</td>
<td>rect(1,1)</td>
<td>1.0</td>
<td>0</td>
</tr>
<tr>
<td>PC-Velo→DCN-Velo</td>
<td>rect(1,1)</td>
<td>(-1.50,-1.45)</td>
<td>0</td>
</tr>
<tr>
<td>DCN-Velo→Motor-Velo</td>
<td>rect(1,1)</td>
<td>(0.36,0.41)</td>
<td>0</td>
</tr>
<tr>
<td>PN-Velo→PC-Velo</td>
<td>nontopo</td>
<td>(0.50,0.55)</td>
<td>-0.08</td>
</tr>
<tr>
<td>PN-Velo→DCN-Velo</td>
<td>nontopo</td>
<td>(0.50,0.55)</td>
<td>0.04</td>
</tr>
</tbody>
</table>
modules. This sensor streams \{L\} are the raw data fed into preprocessing layer.

C. Preprocessing Layer

The laser streams are then preprocessed, such that the sensory input is converted into four different streams, each in an appropriate semantic format for the sub-region it is being fed into. In SC-Turn region, L is capped at \( R \) and then it is normalized (\( T_{SC} = -(L/R) + 1 \)). The normalized signal \( T_{SC} \) denotes that obstacles are nearby when the signal has a higher value and vice versa. \( T_{SC} \) is used as the activity in SC-Turn region such that neuronal units can be activated when obstacles present. In ES-Turn region, the input signal is capped at 0.5\( R \), and normalized (\( T_{ES} = -(L/0.5R) + 1 \)). The \( T_{ES} \) is the turn error signal. \( T_{ES} = 0 \) signifies that no object is within a specific range while \( T_{ES} = 1 \) represents that an object is nearby.

The calculation for activity in SC-Velo is presented in Algorithm 1. The activity is determined by the highest normalized input signal (\( peakUnit \) in the algorithms) and the activity of its neighbouring units (\( peakUnit \pm i \)) is decreased linearly. This calculation signifies that if no object is detected (\( \varphi = 0 \)), the activity is zero. Then, maximum velocity can be outputted by velocity computation (described in Section III E). On the other hand, the activity is peak if a closest object is detected by any of the sensor units (\( i = 1, \ldots, n \)). These activities will decrease the speed calculated by the velocity computation. The calculation for activity in ES-Velo is identical to Algorithms 1, with the exception that the range of laser is capped at 0.5\( R \).

D. Cerebellum Layer

1) Neuronal Response: Standard neuronal dynamics that are implemented in BBDs are employed in S-CDE [21]. The synaptic connections can be either plastic or non-plastic and voltage-dependent or voltage-independent. In S-CDE, only voltage-independent connections are implemented as suggested in CDE [20]. Defining \( j \) as the parent node and \( i \) as the child node (for example, for a PN→PC connection, PN is the parent and PC is the child), the voltage-independent connection from \( j \) to \( i \) is formulated as:

\[
VI_{ij}(t) = c_{ij}s_j(t)
\]  

where \( c_{ij} \) is the synaptic weight between unit \( i \) and \( j \), and \( s_j \) is the activation state of unit \( j \). The activity of the neuronal areas are updated as:

\[
s_i(t + 1) = \phi \left( \tanh \left( \sum_{j=1}^{N}(VI_{ij}(t)) + \omega s_i(t) \right) \right)
\]  

\[
\phi(x) = \begin{cases} 
0; & \text{if } x < \sigma_{i}^{fire} \\
x; & \text{otherwise} 
\end{cases}
\]  

where \( N \) is the total number of synapses onto unit \( i \), \( \omega \) is the persistence of unit activity, and \( \sigma_{i}^{fire} \) is the firing threshold of unit \( i \).

2) Cerebellar Modules: IO transmits the error signal to cerebellar regions (IO → PC and IO → DCN) via climbing fiber. The IO activities are triggered by ES-Turn and ES-Velo. For connection from IO → PC, activity of IO is denoted by \( s_i \) (parent node) while activity of PC is \( s_i \) (child node). \( s_i \) is obtained using equation 2.

PN is a neuron population that delivers afferents to the cerebellum. In the implementation, the activity of PN (\( s_i \) as child node) is determined by the activity of SC-Turn and SC-Velo (\( s_j \) as parent node). For the connection between PN and PC (PN → PC), PN is a parent node while PC is a child node. The total activity of PC neurons is the summation of the activity caused by IO and PN. The depression of PC synapses causes disinhibition of DCN neurons which are used to drive motor activity.

3) Priority-based Delayed Eligibility Trace Learning Rule: Eligibility trace is a basic concept of reinforcement learning which records the parameters that associate an event as eligible for undergoing learning changes [42]. This concept has been employed in [18] which suggested that eligibility trace exists in pursuit learning of eye movements. In [20], Mc Kinstry et al. extended this concept and proposed delayed eligibility trace learning rule in synaptic weight learning for cerebellar system.
to determine an eligibility of a synapse to be plastic and non-plastic.

By investigating the properties of the delayed eligibility trace learning rule, two limitations are observed. First, it is inept at effective motor learning in environments with varied turn curvatures due to the additional condition imposed that once eligibility for plasticity of a synapse has been triggered, subsequent input over that synapse is ignored for some time. This creates the possibility of neglecting important inputs during the time when input is ignored, thus impairing the learning process. Due to the impaired learning process, the effectiveness of learned predictive motor control is limited. Second, there is a limit to the volatility of input stimuli, beyond which learning becomes impaired. This volatility is inversely proportional to the distance between turns in a path. The input volatility limit is determined by the onset of consecutive eligibility trace triggers during traversal of a rapid series of turns. This limitation is caused by an insufficient eligibility trace decay rate.

**Algorithm 2** Pseudo-code for priority-based delayed eligibility trace learning rule

\[
\begin{align*}
&\text{if activity of unit } j \text{ is less than the activity threshold } \epsilon_j \text{ then} \\
&P_j(t+1) = 0 \\
&\text{else if } \epsilon_j (t - \text{delay}) \geq \epsilon \text{ then} \\
&P_j(t+1) = s_j(t - \text{delay}) \\
&\epsilon = s_j(t - \text{delay}) \\
&\text{else if activity of unit } j \text{ is less than the activity threshold } \\
&P_j(t+1) = 0.6 \times P_j(t) \\
&\text{else if activity of unit } j \text{ is greater than the activity threshold } \\
&P_j(t+1) = s_j(t - \text{delay}) \\
&\epsilon = s_j(t - \text{delay}) \\
&\text{end if}
\end{align*}
\]

To overcome these limitations, a prioritized-learning concept is integrated into the learning rule. The calculation of the priority-based delayed eligibility trace learning rule is presented in Algorithm 2. This algorithm consists of three conditions. First, \( P_j(t) \) is set to zero if the simulation cycle \( t \) is less than the predefined delay value, \( \text{delay} \). Secondly, the eligibility trace is set to zero if \( P_j(t+1) = 0 \). Finally, the eligibility trace is triggered when the activity of unit \( j \) is greater than the activity threshold \( \epsilon \).

**E. Motor Output layer**

1) **Turn Rate Computation:** Turn rate (°/cycle) is updated every cycle as a function of activity in the Motor-Turn area. Activity in the area is interpreted using population vector decoding [43]. Each neuronal unit in Motor-Turn has a preferred turn-rate magnitude and direction. For explanatory purposes, neuronal units in Motor-Turn are indexed as \( t_1 \) to \( t_{100} \). Units \( t_1 \) to \( t_{50} \) have a rightward preference of direction, and the preferred turn-rate magnitude of each unit grows linearly with its index. Units \( t_{51} \) to \( t_{100} \) have a leftward preference of direction, and the preferred turn-rate magnitude of each unit shrinks linearly as its index increases. To convert the activity in Motor-Turn in this semantic context to a specific turn rate, a combination of symmetric difference and population vector decoding techniques are used, expressed in equation 5. The resulting vector is the nett asymmetric activity in Motor-Turn, indexed as \( \tilde{a}_1 \) to \( \tilde{a}_{50} \), where preferred turn-rate magnitude grows with the index, negative values indicate a leftward contribution, and positive values indicate a rightward contribution. The turn rate is calculated as shown in equation 6, where \( n \) is the size of \( \tilde{a} \) and \( \gamma \) is a constant defining the maximum turn rate.

\[
\tilde{a} = \begin{bmatrix}
a_1 \\ \vdots \\ a_{50}
\end{bmatrix} = \begin{bmatrix}
t_1 \\ \vdots \\ t_{50}
\end{bmatrix} - \begin{bmatrix}
t_{100} \\ \vdots \\ t_{51}
\end{bmatrix}
\] (5)

\[
\text{TurnRate} = \sum_{i=1}^{n} \left\{ i \times \frac{\gamma}{100} (\tilde{a}_i) \right\}
\] (6)

2) **Velocity Computation:** Velocity (pixels/cycle) is updated every cycle as a function of activity in the Motor-Velo area. Activity in the area is interpreted using population vector decoding. For explanatory purposes, neuronal units in Motor-Velo are indexed as \( v_1 \) to \( v_{100} \), and each has a preferred amount of braking, the magnitude of which grows linearly with its index. The conversion of activity in Motor-Velo in
this semantic context to a specific value is expressed in the equations below.

\[
Velocity = \begin{cases} 
V_{max} & \text{if } \sum_{i=1}^{n} v_i = 0 \\
\|V_{max} - \beta(V_{max} - 1)\| & \text{otherwise}
\end{cases}
\]  

(7)

\[
\beta = \frac{n \sum_{i=1}^{n} (i \times v_i)}{100 \sum_{j=1}^{n} v_j}
\]  

(8)

where \( \beta \) is a braking coefficient that controls the amount of braking used by the robot ranging from 0 to 1, \( n \) is the size of Motor-Velo, \( v_i \) is the \( i \)th unit of Motor-Velo, and \( V_{max} \) is a constant defining the maximum velocity. This formulation prevents the velocity from ever reaching zero, thus imposing a minimum velocity of 1 pixel/cycle.

\[ \text{Velocity} = \begin{cases} 
V_{max} & \text{if } \sum_{i=1}^{n} v_i = 0 \\
\|V_{max} - \beta(V_{max} - 1)\| & \text{otherwise}
\end{cases} \]  

(7)

\[ \beta = \frac{n \sum_{i=1}^{n} (i \times v_i)}{100 \sum_{j=1}^{n} v_j} \]  

(8)

\[ \text{Velocity} = \begin{cases} 
V_{max} & \text{if } \sum_{i=1}^{n} v_i = 0 \\
\|V_{max} - \beta(V_{max} - 1)\| & \text{otherwise}
\end{cases} \]  

(7)

\[ \beta = \frac{n \sum_{i=1}^{n} (i \times v_i)}{100 \sum_{j=1}^{n} v_j} \]  

(8)

The simulated mobile robot is given an innate behavior to move forward at a maximum speed of 6 pixels/cycle. When the robot is close to a border of a path, error signals are generated that induce velocity and turn rate corrections to avoid a collision. When the robot is too close to a border of a path, the robot rotates in place until it is able to continue moving from its current coordinate. When a junction is detected, the robot randomly chooses to turn or moving forward. For junction detection, all coordinates of the junctions in the maps are prerecorded. A junction is detected when an agent is located on one of the recorded coordinates. The turning angle is also predefined such that the agent can turn to an appropriate junction. During this process, the cerebellar model is shut off.

\[ \text{Velocity} = \begin{cases} 
V_{max} & \text{if } \sum_{i=1}^{n} v_i = 0 \\
\|V_{max} - \beta(V_{max} - 1)\| & \text{otherwise}
\end{cases} \]  

(7)

\[ \beta = \frac{n \sum_{i=1}^{n} (i \times v_i)}{100 \sum_{j=1}^{n} v_j} \]  

(8)

\[ \text{Velocity} = \begin{cases} 
V_{max} & \text{if } \sum_{i=1}^{n} v_i = 0 \\
\|V_{max} - \beta(V_{max} - 1)\| & \text{otherwise}
\end{cases} \]  

(7)

\[ \beta = \frac{n \sum_{i=1}^{n} (i \times v_i)}{100 \sum_{j=1}^{n} v_j} \]  

(8)

\[ \text{Velocity} = \begin{cases} 
V_{max} & \text{if } \sum_{i=1}^{n} v_i = 0 \\
\|V_{max} - \beta(V_{max} - 1)\| & \text{otherwise}
\end{cases} \]  

(7)

\[ \beta = \frac{n \sum_{i=1}^{n} (i \times v_i)}{100 \sum_{j=1}^{n} v_j} \]  

(8)

**F. Robot Behavior**

The simulated mobile robot is given an innate behavior to move forward at a maximum speed of 6 pixels/cycle. When the robot is close to a border of a path, error signals are generated that induce velocity and turn rate corrections to avoid a collision. When the robot is too close to a border of a path, the robot rotates in place until it is able to continue moving from its current coordinate. When a junction is detected, the robot randomly chooses to turn or moving forward. For junction detection, all coordinates of the junctions in the maps are prerecorded. A junction is detected when an agent is located on one of the recorded coordinates. The turning angle is also predefined such that the agent can turn to an appropriate junction. During this process, the cerebellar model is shut off.

**IV. SIMULATION RESULTS**

**A. Experiment Setup**

S-CDE was developed in a brain-based device (called agent S-CDE in the rest of the paper) and tested in a simulated mobile robot which randomly and smoothly traversed maps of the Singapore (SG map) expressway, Hong Kong expressway (HK map), and oval running TRack (TR map) (see Fig. 3). The SG map has narrow paths averaging 40 pixels in width and is made up of gradual as well as sharp turns. Total dimension is 1600×900 pixels. The HK map is about three times larger, has wide paths averaging 60 pixels in width, and is made up of mostly gradual turns. Total dimension is 4642×4037 pixels. The TR map has narrow paths averaging 40 pixels in width and is made up of a simple and gradual turn. Total dimension is 1208×535 pixels. The simulation was executed in MATLAB on a PC with a 3.4 GHz processor. In the experiment, the following settings were used: \( \epsilon_o=1.5, \gamma=30, V_{max}=6, R=35, \) and \( n=100. \)

The simulation involved training and testing phases. In the training phase, a complete simulation cycle consisting of spontaneous dynamic input stimuli generation, computations for all neuronal regions, weight adjustments, and simulated movement took approximately 18 ms. In the testing phase, a complete simulation cycle consisting of spontaneous dynamic input stimuli generation, computations for all neuronal regions, and simulated movement took approximately 8 ms. As a comparison, two agents namely agent CDE and agent reflexive...
Fig. 3: The paths traveled by all agents in different maps during testing phase in one of the simulation runs. The agents started from a same position and randomly navigated the map. The agents stopped after 10,000 cycles of movements. The top row shows the paths travelled by agent (a) S-CDE with IO (b) CDE with IO and (c) reflexive in SG map. The middle row shows the paths travelled by agent (d) S-CDE with IO (e) CDE with IO and (f) reflexive in HK map. The bottom row shows the path travelled by agent (g) S-CDE with IO (h) CDE with IO and (i) reflexive in TR map.

were included in the simulation.

Agent CDE is similar to agent S-CDE except that it uses delayed eligibility trace learning rule instead of priority-based delayed eligibility trace learning rule. It is also different from the cerebellar model suggested in [20] in terms of system architecture, processing of the sensory data, calculation of the error signal, computation for the motor output, size of the neuronal regions, and setting of initial connection strength. We are no mean to reproduce the results in [20], thus many of those settings are determined based on experimental investigations. Agent reflexive is a simulated agent using a reflexive motor controller which is purely driven by error signals from IO and has no predictive capabilities. This was achieved by lesioning its DCN → Motor connections.

The experiments were carried out as follows. First, in the training phase, experiments were conducted to train agent S-CDE in SG map with different length of the delay. The delay values tested were 0, 2, 5 and 8 cycles for all experiments. The delay value resulting in the lowest motor error after 10,000 cycles of traversing SG map in the training phase is the calibrated delay value. After training, the testing phase (without lesioning IO connections) was carried out. Furthermore, in order to test the training efficiency, all connections originating from IO were lesion (IO → PC, IO → DCN, IO → Motor). Motor error for each cycle is quantified by taking the ratio of the strength of the error signal generated in the preprocessing layer to the maximum possible strength of the error signal. The experiments were then repeated to calibrate the delay for agent CDE in SG map. The calibrated delay was used in the testing phase of other maps. The testing in SG map represents the training effectiveness in a familiar environment. Following that, the adaptability of agent S-CDE and agent CDE was determined. As both agents were previously trained in SG map, their adaptability was evaluated by tasking them
to traverse and adapt to HK map, which has wider paths and different turn curvatures. Again, agent reflexive was used as a control. Next, the generalization of the learned motor control was evaluated by tasking them to traverse and adapt to TR map which is a simple environment with constant and simple turn curvatures. The above experiments were repeated with training in HK map and testing in all maps and training in TR map and testing in all maps. All of these experiments were repeated 20 times, and the line curve plotted the mean results while the box plot presented the statistical results.

**B. Navigation Results**

The performance in terms of motor error for training in SG map and testing in other maps for the three agents was shown in Fig. 2. It was observed that agent S-CDE with 5 cycles of delays generated less motor error at the second half of training phase. This indicates that agent S-CDE successfully learns to traverse smoothly the SG map. The agent reflexive, which did not involve any learning, generated the highest values of motor error, indicating that agent S-CDE possesses a certain degree of learning in all delays. In testing phase, we kept the IO part in both agent S-CDE and CDE to allow optimal navigation. It was observed that agent S-CDE outperformed agent CDE and reflexive in all maps. Besides, the performance of agent CDE was better than agent reflexive (Figs. 2(c)-2(e)). This indicates that agents S-CDE and CDE successfully learn the predictive control. In a new environment, error signals generated in IO regions are important for the agent to adapt to a new environment or relearn an extinction of a learned behavior. Besides, the prioritized-based learning in agent S-CDE denotes a better learning skill.

In another testing phase, the IO parts were disconnected and agent S-CDE governed the motion simply using immediate depth information of the environment. In this case, agent S-CDE generated less motor error than agent CDE and reflexive (Figs. 2(f)-2(h)). This illustrates that agent S-CDE successfully learns to predict the control actions to navigate smoothly in the path with varied turn curvatures. For agent CDE without IO, a poorer performance was observed. This suggests that agent CDE fails to effectively learn a proper predictive control and it requires error signals to support the predictive motor control even after training for decent traversal of environments with varied turn curvatures.

In Fig. 2(g), agent S-CDE demonstrated effective predictive motor control while traversing HK map despite only having prior experience in SG map. This shows the adaptability of agent S-CDE in different environments. Agent CDE also demonstrated a certain degree of adaptability with a better motor error than agent reflexive. However, its performance was inferior to agent S-CDE. It was observed that motor error of the agents was lower during traversal of HK map than in SG map. This is because HK map has a wider path than SG map, thus is easier to traverse. Next, the adaptability of the motor learning in simple environment with a simple and gradual turn curvature was shown in Fig. 2(h). This experiment shows the generalization of the motor learning in which learning in a complex environment may result in a smooth traversal in a simple environment. The figure showed that agent S-CDE outperformed other agents as indicated by near zero motor error. Agent CDE also demonstrated a good generalization and adaptability in this environment with a lower motor error than agent reflexive.

For a clearer visualization of the navigation performance, Fig. 3 showed the path travelled by different agents with IO in all three maps during testing phase. In SG map, agent S-CDE
showed a smooth traversal along the path except in several edges that need sharp turning. In those edges, the motion was awkward, but returned to smooth after a short traversal. For agent CDE and reflexive, the motions were awkward in most of the traversal, but a better traversal was observed in the pathways of agent CDE. A near similar performance was observed when agents travelled in HK map. In TR map, agent S-CDE demonstrated smooth and stable pattern of traversal along the course, agent CDE showed a slight non-smooth traversal in certain regions but kept itself in the middle of the course. For agent reflexive, its traversal was awkward in certain regions and always drove near to the border of the path.

Next, the border analysis was presented. When traversing the course of a map, it may happen that agents fail to adjust their turning rate and velocity appropriately in order to avoid collision or being too close to a border. This happens especially when a sharp turning is required. Besides, when agents move in an awkward pattern, they often meet a border. In these experiments, collision is disallowed. When an agent is too close to a border of a path, the learning is stopped and an innate behavior is activated to allow the agents to rotate in place until it is able to continue moving from its current coordinate. Thus, the border detection rate denotes another performance criterion when traversing the maps. A smooth traversal can only be achieved when the agents keep themselves in the middle of the course or avoid to be too close to a border of a path. From the robotic point of view, this assessment is more consistent with real navigation since the first concern of mobile robots is to avoid obstacles and keep them save from collisions.

Fig. 4 showed the border detection rate over cycles for training and testing phases for all agents in different environments. Border detection rate generated by agent (a) S-CDE and (b) CDE under different settings of delays during training phase in SG map. An averaging statistical result for border detection rate during the testing phase in (c) SG map (d) HK map and (e) TR map generated by different agents using the best delay setting.

Fig. 6: Border detection rate for testing in different environments after training in HK map. An averaging statistical results for border detection rate during the testing phase in (a) SG map (b) HK map and (c) TR map generated by different agents (with IO) using the best delay setting.
agents with longer delays obtained a lower border detection rate. During the testing phase, agent S-CDE with IO showed lowest border detection rate followed by agent CDE with IO and agent reflexive. Comparing border detection rate in SG and HK maps, a lower border detection rate was obtained for all agents in HK map due to a wider path. In TR map, no border was detected throughout the simulations for all agents since no sharp turn is required.

Next, we performed the training in HK map and testing in other environments. This is to ensure that the learning is not dependent on any particularly map. Fig. 5 showed the performance in terms of motor error during training and testing phases in HK map. The calibrated delays for agents S-CDE and CDE were 5 cycles and 2 cycles respectively. With the present of IO, agent S-CDE and CDE showed a better performance than agent reflexive in all maps during testing phase as shown in Figs. 5(c)-5(e). The present of IO allows continuous learning by triggering the eligibility trace when an unfamiliar occasion or environment is detected. This occasion serves as a salient feature to determine the amount of learning required in synaptic strength for plastic connections. The priority-based delayed eligibility trace learning rule gives every salient feature to trigger the learning behavior, resulting in a better adaptation to environments. In Fig. 5g, agent S-CDE successfully learned the predictive control in HK map, thus generating proper motor commands to adapt to this environment after removing the IO part from its cerebellum. For adaptability in an unfamiliar environment, the agents were placed in SG (Fig. 5(f)) and TR (Fig. 5(h)) maps for random exploration. The testing in SG and TR maps showed that agent S-CDE demonstrated a better adaptability in new environments compared to other agents. Besides, agent S-CDE learned a general motor control which can be adapted in significantly different environments. For agent CDE, it generated a lower motor error in TR map, but not in SG map compared to agent reflexive. This indicates that a complex environment gives enough input data with varied features for effective learning. When agent CDE was put in the TR map, it was able to adapt its motor control to that environment.

The border detection rate for testing in different environments after training in HK map was shown in Fig. 6. These results were consistent with the results obtained in the training in SG map. In SG and HK maps, agent S-CDE with IO demonstrated the lowest border detection rate. Besides, no border was detected by all agents in TR map.

Next, TR map was served as a training environment while SG and HK maps as testing environments. The result was presented in Fig. 7. During the training phase, both agents showed a decrement in motor error over cycles. This is expected as the cerebellar model learns the predictive motor control based on the error signal, thus resulting in a lower motor error at the later stages of cycles. With IO connection, both agent S-CDE and CDE achieved lower motor error during testing in all maps. However, when IO parts were disconnected in the testing phase, the performance was dropped in all maps. Agent S-CDE only showed a better performance than agent reflexive (Fig. 7h) in TR map, while the performance of agent CDE was inferior to agent reflexive in all maps (Figs. 7f-

detection rate decreased especially at the early stage of the training cycles. This is proportional to the motor error as a lower motor error denotes a better alignment of the agents in the middle of the course. For agent S-CDE, a short delay (0 and 2 cycles) showed a fast adaptation to avoid border while a longer delay (5 and 8 cycles) showed a gradual adaptation to avoid border. However, at the end of the training cycles,
7h). This may be attributed to the fact that the input data for effective learning should be varied and diverse in order for the cerebellar system to generalize well the predictive control. This shows that the cerebellar model fails to learn a proper motor control when the input data is monotonous.

The border detection rate for testing in different environments after training in TR was shown in Fig. 8. Both agent S-CDE and CDE with IO achieved better performance than agent reflexive. Even through the agents fail to generalize the navigation capability after training in a simple and monotonous map, optimal performance can be achieved by allowing IO to continuously guide for subsequent training. Similar to the previous results, no border was detected by all agents in TR map.

C. Discussions

With regard to the delay calibration experiments, an explanation for the difference in optimal delay between agent S-CDE and agent CDE is proposed as follows. Agent CDE ignores subsequent input for some time once input surpasses a threshold. If a long delay is used, agent CDE would only learn based on the initiation of a turn and block input arising from traversing the rest of the turn. A short enough delay would enable agent CDE to learn based on input from several points during the turn. However, using such a short delay severely limits the predictive capability of CDE. This results in abrupt corrective motor control and would thus be prone to higher motor error rates and ineffective motor learning in an environment with varied turn curvatures. Agent S-CDE, on the other hand, is able to base its learning on input received throughout a turn due to its re-triggering mechanism. Its learned motor responses respond at the start of a turn and continues throughout the turn. It is thus more likely to result in lower motor error rates. The experimental results corroborate this hypothesis. The initial drop in motor error common to all delay values can be attributed to learned motor control for traversing straight paths and slightly gradual turns. At this point in training all the agents are relatively bad at medium and sharp turns, which improves as training progresses according to each agent’s delay setting, hence the stratified motor error rates.

From the results of the training effectiveness experiments, there are evident that agent CDE was incapable of effective predictive motor control in either environment. A hypothesis for this is proposed as follows. Due to the limited predictive capability previously mentioned, predictive motor control signals for medium and sharp turns would have been generated too close to an impending collision to make a significant difference. Also, the difference in motor error rates for agent CDE between the end of the delay calibration experiment and the training effectiveness experiments suggests that effectiveness of agent CDE is dependent on the presence of error signals from IO supplementing outputs from the predictive motor controller. As such, the experimental results validate the initial prediction of CDE’s dependence on error signals after training for traversal of environments with varied turn curvatures. On the other hand, the performance of agent S-CDE in all maps during the training effectiveness experiment was comparable to that achieved at the end of the delay calibration experiment, which strongly suggests effective retention of learned predictive responses, and demonstrates the robustness of such responses in an unfamiliar environment.

D. Synaptic Changes and Neuronal Activities

In this section, the learning of the synaptic efficacy of the plastic connections was examined. The changes in synaptic weights over cycles were summed in order to provide a clearer visualization to the synaptic weight changes. In Fig. 9, it was observed that the sum of the synaptic weight changes in PN→PC regions (for both Turn and Velo) is greater in SG and HK map than in TR map. This can be attributed to the fact that SG and HK maps provide various input patterns with varied turn curvatures while TR map only consists of a simple and gradual turn curvature. Besides, the turn rate at the later stages of training was increased in all environments, denoting a greater adaptation to the environments. For the corresponding velocity (Fig. 9(f)), HK map showed a higher velocity due to a wider course and SG map demonstrated a lower velocity due to a narrow and complex environment. These demonstrates that the learning with priority-based delay eligibility trace is sufficient to adapt to complex environments.

Fig. 10 showed the activities of different cerebellar regions of agent S-CDE during training in SG map. The agent passed the first time a section of the course of SG map at cycles 105-204. After a few thousand cycles of training, the agent passed the same course at cycles 9274-9373, and its activities were recorded. Before training (Fig. 10(a)), the laser input activated the neurons in PN regions and the white pixels denote the border of a course seen by the robot while the black pixels denote the open areas seen by the robot. ROI regions showed a near similar activity patterns as PN regions. This represents that the error signal, which activates the IO regions, responses to the views seen by the robot. Furthermore, Motor regions demonstrated similar activity patterns as IO regions, and there was no activity in DCN regions. This denotes that the motor commands generated by Motor regions are governed by error signal instead of cerebellar model. In other words, the cerebellar model does not play an important role in controlling the motions of the robot. After training (Fig. 10(b)), no activity was observed in IO regions. Besides, motor regions showed similar activity patterns as DCN regions. This indicates that the motions are controlled by the DCN regions instead of error signal. The PN regions inhibited the activities of DCN regions, thus, showing reverse activity patterns as DCN regions.

V. Conclusion and Future Work

In this paper, a simplified and improved cerebellar model for predictive motor control has been presented. S-CDE has a
Fig. 8: Border detection rate for testing in different environments after training in TR map. An averaging statistical results for border detection rate during the testing phase in (c) SG map (d) HK map and (e) TR map generated by different agents.

Fig. 9: Sum of the synaptic weight change for plastic connections of PN→PC and PN→DCN for both Velo and Turn areas generated by agent S-CDE when learning in different maps. Total of the synaptic weight change over cycles for connection of (a) PN-Turn→PC-Turn (b) PN-Velo→PC-Velo (c) PN-Turn→DCN-Turn and (d) PN-Velo→DCN-Velo. The corresponding (e) turning rate and (f) velocity which are controlled by the cerebellar model and output by Motor-Turn and Motor-Velo regions after responding to the input cues of the environments.

simpler architecture due to the reduction of input modalities. It has demonstrated effective motor learning in environments with varied turn curvatures because of the introduction of the priority-based delayed eligibility trace learning rule and the increased eligibility trace decay rate. Despite the success of S-CDE, it still has several limitations requires further investigation. Firstly, S-CDE is only effective in static environments as it does not possess the necessary architectural components to differentiate between dynamic and static elements of an environment. Also, the fixed delay used in eligibility traces is only optimal over a certain range of speeds. A dynamically-calibrated delay would alleviate this limitation. With further development, it would be possible to integrate S-CDE with a navigational component for autonomous exploration applications, and thus merits further research especially when implementing in a real robotic platform.

ACKNOWLEDGMENT

This work was supported by Agency for Science, Technology, and Research (A*STAR), Singapore under SERC Grant 12251 00002.

REFERENCES

Fig. 10: Neuronal activities of the cerebellum regions on a same section of the course of SG map during early (105-204 cycles) and later (9274-9373 cycles) stages of training. Horizontal axis is the simulation cycles of the training and vertical axis is the activity of neurons in cerebellum regions (100 neurons per region). The activities are represented in the range of [0 1] with black pixel denotes 0 activity (no activity) and white pixel denotes activity of 1 (high activity). As an example, the white pixels in the upper half of Motor-Turn region provide a leftward contribution while the white pixels in the lower half of Motor-Turn region provide a rightward contribution.

IEEE TRANSACTIONS ON AUTONOMOUS MENTAL DEVELOPMENT


Vui Ann Shim received the B.Eng. degree in electrical and mechatronics engineering from the Universiti Teknologi Malaysia, and the Ph.D. in electrical and computer engineering from the National University of Singapore in 2008 and 2012, respectively. He is currently working as a Scientist in the Institute for Infocomm Research, A*STAR, Singapore. His research interests include computational intelligence, computational neuroscience, multi-objective optimization, and robotics.

Chris Stephen Naveen Ranjit is currently pursuing his bachelor’s degree in computer science, Nanyang Technological University, Singapore. His research interests include computational neuroscience and robotics.

Bo Tian received the B.Eng. degree in mechanical engineering from Tsinghua University and M.Eng. in mechanical engineering from National University of Singapore, in 2007 and 2010, respectively. He is currently a research engineer with the Institute for Infocomm Research under Agency for Science, Technology and Research Singapore. His current research interests include biologically inspired machine vision and robot navigation.

Miaolong Yuan received his PhD degree in Mechanical Engineering from Huazhong University of Science and Technology (HUST), China in 1997. During 1997-1999, he worked in HUST as a lecturer. During 1999-2000, he worked as a senior software engineer in Asahi Hi-tech Co. Ltd, Japan. From 2000 to 2008, he was a research fellow in Nanyang Technological University and National University of Singapore. Since 2008, he has been working at A*STAR Institute for Infocomm Research as a scientist. His research interests include computer graphics and computer vision, machine learning, augmented reality and robotics.

Huaqin Tang received B.Eng. degree from Zhejiang University, M.Eng. degree from Shanghai Jiao Tong University, China, and Ph.D. degree in electrical and computer engineering from National University of Singapore, Singapore, in 1998, 2001, and 2005, respectively. Dr. Tang was an R&D Engineer with STMicroelectronics, Singapore, from 2004 to 2006. From 2006 to 2008, he was a Postdoctoral Fellow with Queensland Brain Institute, University of Queensland, Australia. He is currently a Research Scientist and Group Leader of Robotic Cognition, Research Singapore. He has published one monograph (Springer-Verlag 2007) and over 30 international journal papers. His current research interests include neural computation, neuromorphic systems, cognitive computing, and neuro-robotics. He is an Associate Editor of IEEE Transactions on Neural Networks and Learning Systems and Editorial Board Member of Frontiers in Robotics and AI. He serves as guest editor for Neurocomputing (2014) Special Issue on Brain Inspired Models of Cognitive Memory, and for IEEE Trans. on Neural Networks and Learning Systems (2015) Special Issue on Learning in Neuromorphic Systems and Cyborg Intelligence. He serves as organizing committee member for Asia Pacific Symposium of Intelligent and Evolutionary Systems (IES) 2014 and Program Chair for IEEE International Conference on Cybernetics and Intelligent Systems (CIS) 2015.