This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2021.3128179, IEEE Transactions on Cognitive and Developmental Systems

A Computational Developmental Model of Perceptual Learning for Mobile Robot

Dongshu Wang, Kai Yang and Jianbin Xin

***Abstract*—During the perceptual learning of a mobile robot, how to improve the efficiency of its behavior decision is a great challenge. The existing methods suffer from the less flexibility and low efficiency. This paper proposes a novel methodology to address these problems. It mimics the principle that the human's brain can still think, even when he/she neither receives any sensory stimulus nor performs any action, during off-task process, a gated self-organization mechanism is used to trigger the "brain" of the mobile robot to work, analyze the scenarios that it encountered in the former tasks to decide the best moving direction, then store the sensed scenario information by means of the lateral excitation of the internal neurons in a developmental network, and set up the weight connections from the internal area to high-level decision-making area of the developmental network. In the following perceptual learning, if the robot encounters a similar scenario, it can make a better and faster behavior decision. Therefore, this methodology can greatly enhance the efficiency of behavior decision with the limited training samples, without re-training in facing a new scenario. Most importantly, this methodology makes the robot continuously improve its intelligence through the autonomous learning, even in off-line state. Extensive simulation and experiment results of the mobile robot perceptual learning demonstrate its potential. Since it is task-nonspecific, the same learning principles are potentially suitable for other fields. To the best of our knowledge, it is the first trial to apply the transfer learning through the lateral excitation mechanism of internal neurons to the robot field with the emergent representation.**

***Index Terms*—Behavior decision, off-task process, lateral excitation, developmental network, gated self-organization mechanism, transfer**

I. Introduction

DURING the perceptual learning (PL) of a mobile robot, how to enhance the efficiency of its behavior decision, will generate great influence on its following behavior. The core problem for a mobile robot to make future behavior decision is how to determine the moving direction, which will certainly affect its next behavior decision.

To decide the moving direction of the mobile robot, a substantial amount of research has been employed [11], [26]. Generally, these research can be divided into two categories: conventional methods and intelligent methods. A large number of traditional behavior decision methods, such as cell decomposition algorithm [43], roadmap approach [30], artificial potential field approach [31], mathematical programming [6], probability navigation function [8], hybrid approaches [38], and other methods [2], have been used in the mobile robot. But these methods have such main drawbacks as high computational cost, demanding for precise information about the environment, requiring an accurate sensing mechanism for real-time implementation, specific behavioral rules designed for particular task, namely, once the task changes, new rules have to be re-designed, etc. Therefore, they are less preferred in real-time implementation.

Recently, with the rapid development of the intelligent technology, many intelligent approaches have been widely used in mobile robot's behavior decision-making, such as genetic algorithm [34], fuzzy full-state feedback control [57], particle swarm optimization [46], [47], neural network [7], [36], firefly algorithm [33], artificial immune system [3], krill herd algorithm [37], ant colony optimization [20], artificial bee colony algorithm [25], grey wolf colony optimization [37], bacterial foraging optimization algorithm [13], cuckoo search algorithm [39], shuffled frog leaping algorithm [12], invasive weed optimization [32], harmony search algorithm [21], bat algorithm [42], etc. However, just like the conventional approaches, these intelligent methods also exist several disadvantages, such as longer computational time, complicated design, necessary learning phase and numerous learning samples, requiring large memory, etc.

Generally speaking, the existing behavior decision-making approaches of the mobile robot suffer from the common limitations: these approaches generally solve the robot behavior decision-making in particular scenarios, once the scenario changes, the behavior decision strategy needs re-designing, leading to a poor flexibility. Furthermore, these methods generally need considerable computation cost, resulting in low efficiency [10].

Since the scenarios that the robot percepts have certain similarity, in the following perceptual learning, it may encounter a similar scenario. If the robot can learn from its previous experience, it will certainly promote its efficiency of the behavior decision.

A. Transfer in the PL

Repeated performance of tasks causes the long-lasting improved sensitivity to the trained input, a phenomenon called perceptual learning (PL), which investigates how experience can alter the manner we perceive sights, sounds, smells, tastes, and touch [9]. One of the trademark characteristics of perceptual learning is its stimulus and specificity for different

Dongshu Wang is with School of Electrical Engineering, Zhengzhou University, Zhengzhou, Henan, 450001, China. (email: wangdongshu@zzu.edu.cn)

Kai Yang is with Henan Branch of China Mobile Communications Group Co. Ltd, Zhengzhou, Henan, 450021, China. (e-mail: 1092813671@qq.com)

Jianbin Xin is with School of Electrical Engineering, Zhengzhou University, Zhengzhou, Henan, 450001, China. Corresponding author (e-mail: j.xin@zzu.edu.cn).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2021.3128179, IEEE Transactions on Cognitive and Developmental Systems

aspects of task and stimulus configuration, such as stimulus orientation or curvature, retinotopic position and ocularity.

A large number of recent researches, however, illustrate that specificity is not an innate feature of the perceptual learning, but a function of experimental paradigms [9], [35]. For instance, a substantial amount of researches had been done by Yu and his colleagues who proposed novel "double training" method to make visual perceptual learning of different tasks transfer to untrained positions and orientations/directions. Visual perceptual learning can transfer to a novel position if this novel position is trained in a secondary task which has no influence on the performance of primary learning task [48], [49], [54]. Sometimes when double training is executed at the same position, learning can also transfer to new positions [50]. Moreover, learning also transfers to a novel direction when the second task is performed at the novel direction to remove feature specificity [55], [56], [58]. Their works indicate that position specificity is not necessarily a real characteristic of the direction learning, and visual perceptual learning is at least in some situations a high-level procedure which occurs beyond the retinotopic and feature selective visual areas. So these researches elaborate on the current neurophysiological argue about the brain sites of orientation learning and help to delineate the general principle of the perceptual learning.

Uka et al. [44] investigated whether behavioral sensitivity and neuronal features transfer across visual fields through the depth-discrimination task, behavioral results with two monkeys suggested that although learning was specific to the trained hemifield, the transfer learning which caused the fast learning at the new position occurred across the visual hemifields. They further presented that even neuronal features could transfer across hemispheres. Using an orientation discrimination task, Jeter et al [14] demonstrated that specificity followed after extensive training, while the earliest stages of perceptual learning exhibited substantial transfer to a novel position and an opposite direction.

Moreover, based on the integrated re-weighting theory model of PL, Sotiropoulos et al [41] even presented a model including an important characteristic: dynamic weighting of retinotopic position specific vs position-independent representations. This dynamic performance-monitoring model unified a substantial amount of psychophysical data on transfer of PL, such as the short-vs-long staircase effect, and many results from the double training literature.

More transfer of learning have been observed in early stage of training with shorter training sessions [1], using coarse discrimination [15], change in learning rate [17], pretesting and subliminal exposure [22], double training [24], TPE training [59], or mixed different stimuli and tasks [60].

B. Off-task Process

Off-task processes in the developmental network are the neural interactions during the times when the network is not attending to any stimulus and task [40]. As for the human being, in the off-task process, he/she gets no input stimulus and does not output any behavior, but the human brain can still work, playback and re-study the past things by the memory replay mechanism [4], [5], [18]. This re-study can generate a guiding role on his/her follow-up behavior. Fig. 1 provides a general pattern of the transfer in PL: training a perceptual task in the 1st condition accompanied by exposure to the second condition causes the transfer of learning effects to the second condition [40]. Here, exposure means receiving (being exposed to) any sensory stimulus that has at least one characteristic (such as location, orientation, etc) in common with the transfer condition [40]. The transfer model proposed in this work is motivated by this general pattern.

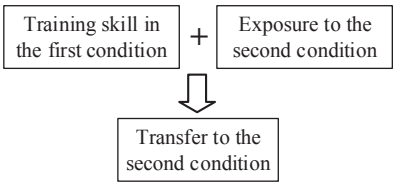


Fig. 1. General mode observed in transfer studies.

C. Contribution of This Work

Recent neurophysiological evidences [23], [61], [27], [19] show that perceptual learning is associated with the neuronal changes not in the sensory cortical regions, but in higher regions correlated to decision making. For instance, Law and Gold [23] observed that in discriminating the motion direction, behavioral enhancement was associated with the plasticity of the neurons in lateral intraparietal cortex (LIP, a decision area), instead of the sensory area MT (middle temporal cortex) that was also active during the task.

Since PL is associated with neuronal changes in higher regions correlated to the decision making, it is very natural to generate an idea to introduce the PL to the behavior decision-making of mobile robot. Unfortunately, till now, PL is widely used in the pattern recognition fields [14], [44], [48], while a few in the robot field. Our previous work [45] has studied the effect of PL on the navigation of an artificial agent, combining with the reinforcement learning through dopamine and serotonin. This work will investigate the transfer mechanism of the moving direction of a mobile robot when it is exposed in a similar scenario during an environment exploration, which is a typical PL process.

Simulating the thinking principle of the human brain during the off-task process, in the intervals of the PL, i.e., neither perceptual input nor action output, in this paper, a gated self-organization mechanism is used to trigger the "brain" of the mobile robot to work, i.e., to re-analyze and investigate the scenarios that it met in former tasks, and determine the best moving direction corresponding to that scenario. Then, it establishes connection between the perceived scenario information and the behavior decision (determining the best moving direction in this work), through the weight connections of the developmental network (DN).

Moreover, the lateral excitation mechanism of the internal neurons of the DN is used to enroll more surrounding neurons

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2021.3128179, IEEE Transactions on Cognitive and Developmental Systems

to store the similar scenario information. In the subsequent environmental PL, when the robot encounters a similar scenario, the robot can make a better and faster decision and transfer to the best moving direction, in accordance with the knowledge learned in the previous off-task processes.

Therefore, this methodology can improve the accuracy and efficiency of the behavior decision with the limited training samples, and without re-training in facing a new scenario, embodying the capacity of autonomous mental development. Furthermore, this methodology is task-nonspecific, and can be used into other perceptual learning. As far as we know, it is the first work to use the transfer mechanism by means of the lateral excitation of the internal neurons in a DN to the behavior decision-making of a robot.

The following sections are arranged as below. Section II introduces the DN and architecture of the motivated developmental network (MDN). Section III illustrates the principle of the off-task process and Section IV explains the principle of transfer learning. Section V and section VI design three simulations and real experiment to demonstrate the potential of the proposed method. Section VII concludes this paper with a discussion on future work.

II. Theory of Developmental Network

A. Algorithm of the Developmental Network

Developmental Network (DN) is an emergent network which can serve as a model of the brain-mind that lives and learns autonomously in the open-ended, dynamics, real physical world [51], [52], [53]. In general, a simplest DN has three layers, the internal layer Y as a "bridge", its sensory layer X as a "bank" relative to the Y and its motor layer Z as another "bank" relative to Y, as illustrated in Fig. 2. In principle, the layer X can model any sensory modality (e.g., vision, audition, and touch, etc). The motor layer Z serves as input or output ports. When the DN is in a status of supervised learning, i.e., training, Z serves as the input layer. Otherwise, Z gives a motor output to drive effectors to act on the external world.

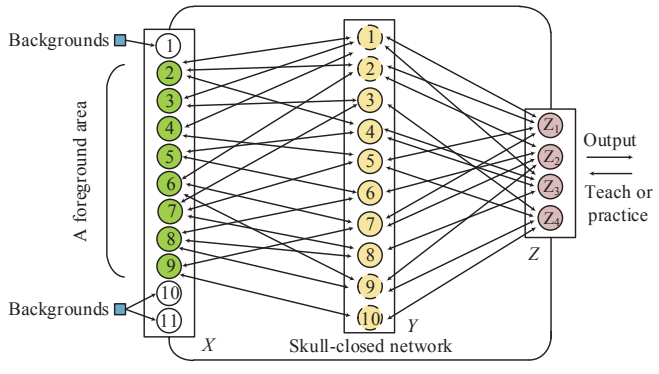


Fig. 2. Schematic diagram of the basic DN. Source: from [52].

The detailed training procedure of the DN can be described as follows.

1) At time $t = 0$, for each layer A in $\{X, Y, Z\}$, initialize its adaptive part $N = (V, G)$ and the response vector $\mathbf{r}$, where V contains all the synaptic weight vectors and G stores all the neuronal ages.

2) At time $t = 1, 2, ...$, for each layer A in $\{X, Y, Z\}$, do the following two steps repeatedly forever:

a) Every layer A computes using the layer function f

$$(\mathbf{r}',\ N') = f(\mathbf{b}, \mathbf{t}, N) \tag{1}$$

where f is the unified layer function described in the following equation (2), $\mathbf{b}$ and $\mathbf{t}$ are layer's bottom-up and top-down inputs from current network response $\mathbf{r}$, respectively; and $\mathbf{r}'$ is its new response vector.

b) For each layer A in $\{X, Y, Z\}$, A replaces: $N \leftarrow N'$ and $\mathbf{r} \leftarrow \mathbf{r}'$.

Each neuron in layer A has a weight vector $\mathbf{v} = (\mathbf{v}_b, \mathbf{v}_t)$, corresponding to the layer input $(\mathbf{b}, \mathbf{t})$, if both bottom-up part and top-down part are applicable to the layer. Otherwise, the missing part of the two should be dropped from the notation. Its pre-response energy is the sum of two normalized inner product:

$$\mathbf{r}(\mathbf{v}_b, \mathbf{b}, \mathbf{v}_t, \mathbf{t}) = \frac{\mathbf{v}_b}{||\mathbf{v}_b||} \cdot \frac{\mathbf{b}}{||\mathbf{b}||} + \frac{\mathbf{v}_t}{||\mathbf{v}_t||} \cdot \frac{\mathbf{t}}{||\mathbf{t}||} = \dot{\mathbf{v}} \cdot \dot{\mathbf{p}} \tag{2}$$

where $\dot{\mathbf{v}}$ is the unit vector of the normalized synaptic vector $\mathbf{v} = (\dot{\mathbf{v}}_b, \dot{\mathbf{v}}_t)$, and $\dot{\mathbf{p}}$ is the unit vector of the normalized input vector $\mathbf{p} = (\dot{\mathbf{b}}, \dot{\mathbf{t}})$.

To simulate the lateral inhibition (winner takes all) within each layer A, only top-k winners fire and update. Considering $k = 1$, the winner neuron j is identified by:

$$j = \arg\max_{1 \le i \le c} \mathbf{r}(\mathbf{v}_{bi}, \mathbf{b}, \mathbf{v}_{ti}, \mathbf{t}) \tag{3}$$

where c is the neuron number in the layer A. For $k = 1$, only the single winner fires with response value $y_j = 1$ and all other neurons in A do not fire with response value $y_j = 0$.

All the connections in the DN are learned incrementally based on Hebbian learning — co-firing of the pre-synaptic activity $\dot{\mathbf{p}}$ and the post-synaptic activity y of the firing neuron. Here, the layer Y is taken as an example, since other layers learn in a similar way. If the pre-synaptic end and the post-synaptic end fire together, the synaptic vector of the neuron has a synapse gain $y\dot{\mathbf{p}}$. Other non-firing neurons do not modify their weights. When a neuron j fires, its weight is updated by a Hebbian-like mechanism:

$$\mathbf{v}_j \leftarrow \omega_1(n_j)\mathbf{v}_j + \omega_2(n_j) y_j \dot{\mathbf{p}} \tag{4}$$

where $\omega_2(n_j)$ is the learning rate depending on the firing age n_j of the neuron j and $\omega_1(n_j)$ is the retention rate with $\omega_1(n_j) + \omega_2(n_j) \equiv 1$. The simplest version of $\omega_2(n_j)$ is $1/n_j$. The age of the winner neuron j is incremented $n_j \leftarrow n_j + 1$.

B. Motivated Developmental Network (MDN)

Fig. 3 shows the architecture of the MDN with the serotonin and dopamine systems. Functions of dopamine and serotonin have not been completely understood [16], since serotonin and dopamine have different effects in different parts of human

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2021.3128179, IEEE Transactions on Cognitive and Developmental Systems

body [28]. In this paper, roles of the dopamine and serotonin are associated with reward and punishment, respectively. While those of the glutamate and GABA are linked with excitatory signal and inhibitory signal, respectively.

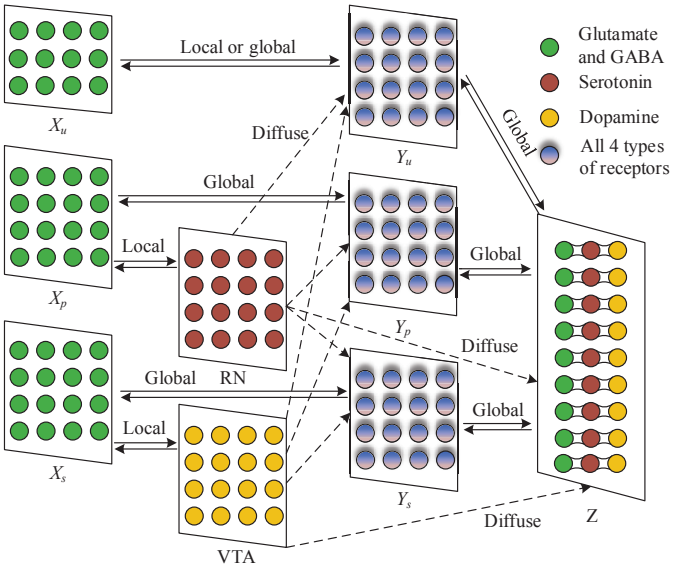


Fig. 3. Schematic diagram of the MDN with the serotonin and dopamine subsystems.

As illustrated in Fig. 3, in the MDN, the sensory layer X can be represented as X=(X_u, X_p, X_s), where X_u represents an unbiased input, while X_p and X_s denote the pain input and sweet input, respectively. Fig. 3 shows that Raphe Nuclei (RN) and Ventral Tegmental Area (VTA) can respectively release the serotonin and dopamine, which can diffuse into other parts of the human brain. RN and VTA can be regarded as the serotonin and dopamine versions of X_p and X_s, respectively.

Similarly, layer Y can also be denoted by three sub-layers: Y=(Y_u, Y_p, Y_s). Neurons in layers Y_p and Y_s receive the inputs from X_p and X_s, respectively. Neurons in Y_u have glutamate, serotonin and dopamine receptors. The role of GABA neurotransmitters is simulated by the top-k competition. Considering the roles of the serotonin and dopamine, the new learning rate of the neurons in layer Y can be depicted as follows [45]:

$$\omega_2(n_j) = \min((1 + \alpha_{RN} + \alpha_{VTA})\frac{1}{n_j}, 1) \tag{5}$$

where α_{RN} and α_{VTA} represent the parameters associated with RN and VTA, respectively. If the serotonin and dopamine do not work in the neuromodulatory system, then α_{RN}=α_{VTA}=0, hence the new learning rate will return its original form $\omega_2(n_j) = 1/n_j$. Since the learning rate $\omega_2(n_j)$ and retention rate $\omega_1(n_j)$ of the neurons meet the condition of $\omega_1(n_j)$+$\omega_2(n_j)\equiv$1, generally the learning rate $\omega_2(n_j) \leq 1$. In Eq. (5), if $(1 + \alpha_{RN} + \alpha_{VTA})/n_j >$1, the function "min" is used to limit its value.

The motor layer Z can be denoted by a series of neurons $Z = (z_1, z_2, \cdots, z_m)$, where m is the number of the neuron in layer Z. Corresponding to the three sub-layers in layers X and Y, each z_i also has three neurons, i.e., $z_i = (z_{iu}, z_{ip}, z_{is})$, as shown in Fig. 3, where z_{iu}, z_{ip} and z_{is} (i=1,2,$\cdots$,m) represent the unbiased, pain and sweet motors, respectively. Whether an action i is activated depends not only on the response of z_{iu}, but also on those of z_{ip} and z_{is}. z_{ip} and z_{is} show how much negative value and positive value are associated with the i-th action, respectively, according to the past experience. They form a triplet for the pre-response energy of each motor neuron i, coming from glutamate, serotonin and dopamine, respectively.

Modeling the motor neuron's internal interactions of the three different types of neurotransmitter, the total pre-response energy of a motor neuron is determined as follows:

$$z_i = z_{iu}(1 - \alpha z_{ip} + \beta z_{is}) \tag{6}$$

where both α and β are positive constants.

Then the j-th motor neuron with the highest pre-response energy can be activated according to the top-k competition mechanism:

$$j = \arg \max_{1 \leq i \leq m} \{z_i\} \tag{7}$$

Other neurons in layer Z cannot fire.

As for the MDN, its detailed description, such as the learning rate of internal neurons, etc, can be found in [45].

III. Learning in the Off-task Process

In the off-task process, there is neither sensory stimulus nor behavior output, and the central nervous system of the mobile robot, i.e., the MDN, is in a state of rest. In this state, how to trigger the central nervous system to work again, is the first problem to be investigated in this part. In this section, a gated self-organization mechanism is used to address this problem.

Since there is no sensory stimulus and behavior output in the off-task processes, in order to trigger the MDN to work again, the MDN must have an external input, either from the sensory layer X or motor layer Z. But there is no sensory input from the layer X, therefore, we have to design a manner to realize an input from the motor layer. Considering that there is certain similarity among the scenes the agent explored, the motor neurons highly activated in the previous works have higher probabilities to fire again. According to this principle, a gated self-organization mechanism as explained in detail below, is designed, namely, in the time period set, the motor neurons with higher activated times in previous work are more likely to be selected as the activated neurons. The motor neurons with their fire times higher than the threshold are selected to be the input to the MDN to trigger it to work again during the off-task processes, then the agent re-studies the scenes encountered in the previous work and determines the optimal behavior decision by the lateral excitation of the internal neurons of the MDN.

A. The Gated Self-organization Mechanism

Whether or not a motor neuron can be activated in the off-task processes is determined by a function of the amount of recent exposure of the MDN to the concept that the neuron

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2021.3128179, IEEE Transactions on Cognitive and Developmental Systems

denotes. In this paper, the concept corresponds to the moving direction of the agent.

Probability of a motor neuron activated in the off-task processes, under the condition of no other neuron activated in the same layer, can be expressed by a monotonically increasing function of the amount of recent exposure to the corresponding concept, i.e.,

$$p(z_i = 1|\exists j \neq i, z_j = 1) = 0$$
$$p(z_i = 1|\forall j \neq i, z_j = 0) = \frac{2}{1 + e^{-\gamma_i}} - 1 \tag{8}$$

where $\gamma_i \geq 0$ represents the amount of recent exposure to the concept that the ith motor neuron denotes. To simulate the lateral inhibition in the motor layer, the conditional probability in Eq. (4) represents the lateral inhibition in the motor layer, and it assures that a motor neuron can not be activated if there is already another neuron activated in the same layer.

Concretely, the amount of exposure to the ith concept, γ_i, in Eq. (4), can be calculated as follows:

$$\gamma_i = \frac{n_{z_i}}{\sum_{i=1}^{8} n_{z_i}} \tag{9}$$

where n_{z_i} represents the activated times of the ith neuron in layer Z, and $\sum_{i=1}^{8} n_{z_i}$ denotes the total activated times of the motor neurons in layer Z, where 8 represents the 8 types of movement directions that the agent can choose, as explained in the following Section IV A.

The principle of the gated self-organization mechanism can be depicted as follows: ranking the motor neurons in descending order in light of the activated probabilities, then activate the former k neurons with nonzero probabilities. For instance, assuming k=3, it is assumed that the activated probabilities of the former three motor neurons are nonzero, and the order of the three motor neurons according to their probabilities is [neuron1, neuron2, neuron4], then the MDN algorithm circulates the following process three times: use the sequence of the activated motor neurons as the input from layer Z to layer Y, activate the internal neurons in layer Y in terms of the top-k competition mechanism, perform the lateral excitation of the internal neurons to activate surrounding neurons, construct the new synaptic weights among the neurons in layers Y and Z. Here the first circulation of the MDN algorithm is used to be an example to explain the process clearly. The input from layer Z to layer Y can be denoted as z_1=[1, 0, 0, 0, 0, 0, 0, 0, 0], then the pre-response energies of the internal neurons in layer Y can be calculated, and the internal neurons with nonzero pre-response energies is activated, all these activated internal neurons are associated with the first movement direction of the agent, thus they only link with the first neuron, namely, z_1 in layer Z. Then pre-response energies of these internal neurons are multiplied by a linearly declining function in terms of the rank of the neurons:

$$\frac{k - r_i}{k} z_i \rightarrow z_i \tag{10}$$

where $0 \leq r_i < k$ denotes the rank of the neuron in terms of the pre-response energy, and the internal neuron with the highest pre-response energy has the rank of 0, the neuron with the second highest pre-response energy has the rank of 1, and the rest can be deduced in the same manner.

B. Lateral Excitation in the Internal Neurons of the MDN

During the off-task process, these reactivated motor neurons make up the top-down input of the MDN, and the synaptic weights from the motor layer Z to internal layer Y in the last work are adopted, then the pre-response energies of the internal neurons are calculated by multiplying the top-down input and the corresponding synaptic weights. The k internal winner neurons are selected by means of the top-k competition mechanism to be activated. These activated internal neurons perform the lateral excitation to activate more surrounding neurons to store new information. In the lateral excitation, the neighbouring neurons of the k winner neurons are also permitted to be activated and modify their synaptic weights. These pre-response energies of the new fire neighbouring neurons are determined by those of the winner neurons, multiplied by an exponentially declining function of their distance to the winner neurons:

$$e^{\frac{-d^2}{2}} z_{winner} \rightarrow z_i \tag{11}$$

where d=1 denotes the distance between the immediate four neighbors (up, down, left, right) and the winner neuron, and $d = \sqrt{2}$ denotes the distance between the diagonal four neighbors and the winner neuron, and the rest distances can be deduced in the same manner, as shown in Fig. 4.

$2\sqrt{2}$	$\sqrt{5}$	2	$\sqrt{5}$	$2\sqrt{2}$
$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$
2	1		1	2
$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$
$2\sqrt{2}$	$\sqrt{5}$	2	$\sqrt{5}$	$2\sqrt{2}$

Fig. 4. Scheme of the lateral excitation of the internal neurons in the MDN. It is noteworthy that only 24 neurons surrounding the winner neuron fire. Actually, the fire scope of the winner neurons can be set in light of the actual requirement.

Then these new fire neurons modify their activated times and connection weights. In the off-task process, these new fire neurons can memorize the similar environment knowledge with the neurons that activate them. The environment similarity s defined in the following Eq. (12) is used to determine which neuron to store which similar environment knowledge. In other words, the environment knowledge corresponding to the highest similarity are stored in the nearest surrounding

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2021.3128179, IEEE Transactions on Cognitive and Developmental Systems

neurons (d=1 in Fig. 4), when all the nearest four neurons have been occupied, other environmental knowledge with relatively low similarity are stored to the nearer surrounding neurons ($d = \sqrt{2}$ in Fig. 4), and the rest can be deduced in the same manner. During the following perceptual learning of the mobile robot, if it encounters a similar environment, it can determine the movement direction quickly, according to the relation between the environment knowledge and the corresponding movement direction that the robot has learned in the off-task processes.

IV. How the Lateral Excitation Cause Transfer during Off-task Process

In the training phase, robot executes stimulus-specific (corresponding to the specific location relationship among the robot, target and the obstacles in this work, depicted by the following Eq. (10)) and concept-specific (corresponding to the specific moving direction of the robot) learning, e.g., $L_{1,1}D_1$, as shown in Fig. 5 (A), where D_1 denotes the moving direction "1" of the robot (the robot has 8 moving directions, as depicted in the following Section V. A), and $L_{1,1}$ denotes the first location relationship among the robot, target and the obstacle, corresponding to the moving direction "1". $L_{1,2}$ denotes the second location relationship among the robot, target and the obstacle, corresponding to the moving direction "1", and the rest can be deduced in the same manner. Each moving direction of the robot has 19 typical location relationships, as shown in Fig. 6. This phase establishes the synaptic connections between the specific location relation and the specific moving direction of the robot.

During the perceptual learning, at each time step, the robot's brain (MDN) transforms the environment location information to corresponding input information p, according to the following Eq. (10), and calculate the similarity between this new input scenario and the one already learned. Assuming the current neuron weight is v, the similarity s can be computed as follows:

$$s = \frac{v}{||v||} \cdot \frac{p}{||p||} \tag{12}$$

If the similarity is no less than a threshold already set, this scenario is regarded as already learned, and the robot can make a decision from the learned knowledge. Otherwise, it is regarded as unknown, and the robot also makes a decision based on its experience (the knowledge it has learned), and moves along this direction to continue the following task. It is noteworthy that this direction maybe not the best and the robot may encounter a variety of new scenarios in its following task.

In the off-task processes, the robot will re-study these new scenarios that it has encountered in the former work and determine the best moving direction in accordance with the following section Determining the Best Moving Direction. Then it will store these new scenario information and their corresponding moving direction by means of the lateral excitation mechanism depicted in the above section. Thus the MDN generates new weight connections, that is to say, it connects

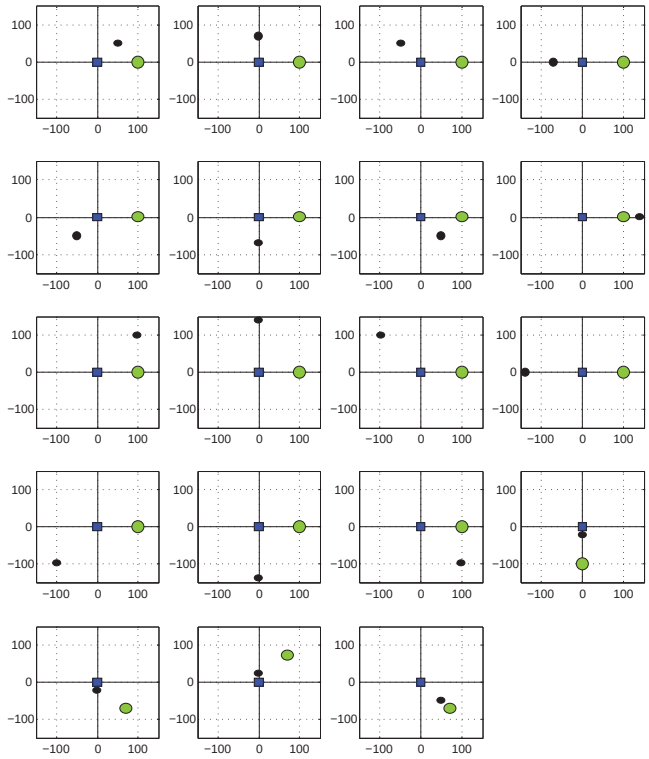


Fig. 6. The 19 typical location relationship among the robot, target and obstacle, corresponding the moving direction "1", where the blue rectangle denotes the agent, green circle denotes the target, and black circle denotes the obstacle, consistent with the description in the following Fig. 7. The location relationship of other moving directions can be deduced with the same manner.

these scenario information with the best moving directions. In the following perceptual learning, when the robot encounters (exposes to) the similar scenario, it can make a better decision (transfer to a better moving direction in this work) based on the experience obtained in the pervious off-task processes. With this mechanism, the robot will become more and more smart.

A. Example: Transfer via the Off-task Processes

Assuming at a time step, the new input scenario (saved in L_m as shown in Fig. 5 (A)) is regarded as a new scenario. In the off-task process, the robot calculates and determines the best moving direction corresponding to this new scenario L_m, in terms of the following section Determining the Best Moving Direction. Assuming the best moving direction is direction "1". At the same time, assuming the neuron denoted by $L_{1,19}$ fire in the top-k competition in the internal neurons, then this new scenario L_m will be stored to the neuron (depicted by the grey neuron in Fig. 5 (B)) which rounds the neuron $L_{1,19}$, according to the mechanism depicted in the above section Lateral Excitation in the Internal Neurons of the MDN. Thus the robot saves this relationship, i.e., the new scenario L_m and the corresponding moving direction "1", to the MDN. Similarly, another new scenario L_n can be stored to the grey neuron that rounds the neuron $L_{8,2}$, hence the robot connects the scenario L_n and the corresponding moving direction "2".

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2021.3128179, IEEE Transactions on Cognitive and Developmental Systems

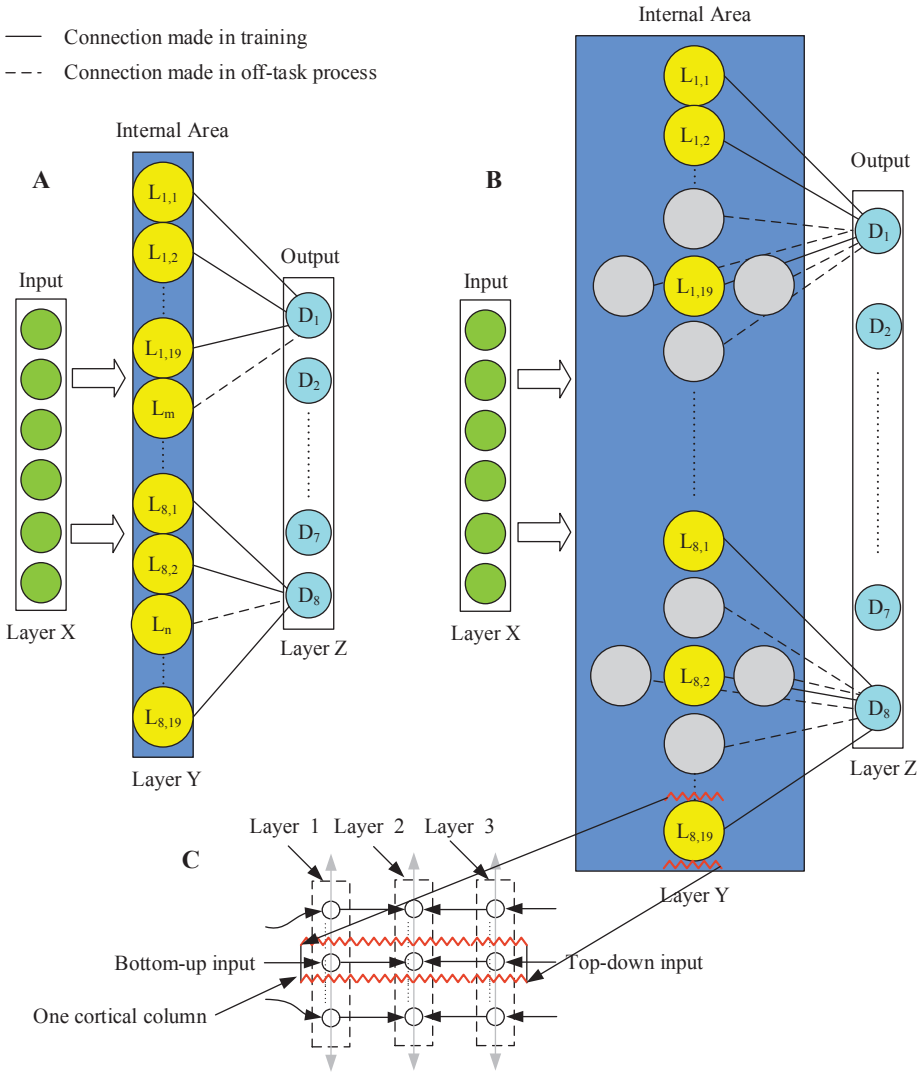


Fig. 5. Scheme of principle of the lateral excitation causing the transfer learning in the off-task process. (A) Transfer of the moving directions in the MDN. (B) Fire more neurons in the internal layer Y. (C) A neuron column in the internal layer Y magnified.

In the following perceptual learning, if the robot encounters a similar scenario, it can choose a better moving direction based on the knowledge it has learned in the off-task processes, with the limited training samples and without re-training. Therefore, this mechanism can make the robot to acquire more and more knowledge from the limited training samples, and increase its intelligence gradually, even in the off-line state, embodying the ability of the autonomous mental development just like the human beings.

B. Determining the Best Moving Direction

Choosing a feasible behavior decision a under the present state $S_1 : (x_1, x_2, ..., x_n)$, and this decision a causes the robot to transform its state to a new state $S_2 : (x_1^{'}, x_2^{'}, ..., x_n^{'})$. Assuming the state of the destination that the robot should reach is $S^{'} : (x_1^{''}, x_2^{''}, ..., x_n^{''})$, through calculating the Manhattan distance between the new state S_2 and the destination $S^{'}$ to evaluate the decision a, then we can achieve each evaluation score for each decision a in $A : (a_1, a_2, ..., a_n)$ of the state S_1. The decision with the highest score is chosen as the best decision in the state S_1.

$$score = \frac{1}{|S_2 - S^{'}|} = \frac{1}{\sum_{i=1}^{n} |x_i^{'} - x_i^{''}|} \tag{13}$$

It is noteworthy that (x_1, x_2, ... , x_n) have different meanings in different situations. The element number n depends on the concrete situation. In this work, (x_1, x_2) can be seen as the horizontal and vertical coordinates of the agent.

Next section, three simulation experiments in static and dynamic environments are designed to demonstrate the potential of the proposed methodology.

V. Simulation Experiments

In this section, several entities are used to illustrate the effect of the method designed above. One of the entities is an agent

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2021.3128179, IEEE Transactions on Cognitive and Developmental Systems

directed by the MDN to think and act. Except the agent, there exist several obstacles and one target in the environment. In the movement, the simulated agent will receive the reward from the dopamine if it comes near the target and punishment from the serotonin if it closes to the obstacles. Hence, the simulated agent can determine a feasible trajectory to reach the target under the double constraints of serotonin and dopamine. The agent, however, have to learn this behavior by means of the trial and error experiences.

A. Input and Output

Following entities can be defined, an agent (a), one target (t) and an obstacle (o), as shown in Fig. 7, the agent is represented by a blue square, the target is depicted by a green circle, while the obstacle is denoted by a black circle. From the Fig. 7, the position and orientation relations among the three entities can be represented by the following formula:

$$\begin{aligned}\theta_t =& \arctan(a_x - t_x, a_y - t_y)\\ d_t =& \sqrt{(a_x - t_x)^2 + (a_y - t_y)^2}\\ \theta_o =& \arctan(a_x - o_x, a_y - o_y)\\ d_o =& \sqrt{(a_x - o_x)^2 + (a_y - o_y)^2}\\ x_u =& \{\cos\theta_t, \sin\theta_t, \cos\theta_o, \sin\theta_o, \frac{d_t}{d_t + d_o}, \frac{d_o}{d_t + d_o}\} \end{aligned} \quad (14)$$

where (a_x, a_y), (t_x, t_y), (o_x, o_y) denote the coordinates of the agent, target and obstacle, respectively; θ_t (θ_o) represents the angle between the heading of the agent and direction of the target (obstacle), and d_t (d_o) indicates the distance between the agent and the target (obstacle). x_u denotes the sensory input of the MDN. During its movement, the agent can execute one of the 8 possible actions, namely, it can move along each of the cardinal or inter-cardinal directions. So the output of the MDN is one of the 8 movement directions.

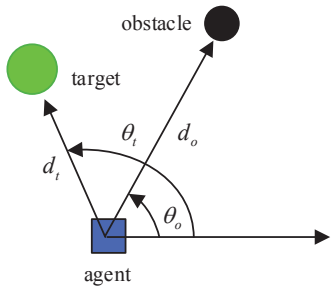


Fig. 7. Scheme of the relation of the position and orientation among the agent, target and obstacle.

B. Simulation Setup

In the MDN model, the reward value α can be defined as follows:

$$\alpha = \begin{cases} 0 & d_t < d_{2t} \\ \frac{1}{d_{1t} - d_{2t}} d_t - \frac{d_{2t}}{d_{1t} - d_{2t}} & d_{1t} < d_t < d_{2t} \\ 1 & d_t > d_{1t} \end{cases} \quad (15)$$

where d_{1t} is the original distance between the agent and the target, d_{2t} denotes the distance when the agent reaches the target, and d_t represents the real-time distance between them.

The punishment value β can be defined as follows:

$$\beta = \begin{cases} 0 & d_o > d_s \\ -\frac{1}{d_s - d_{ms}} d_o + \frac{d_s}{d_s - d_{ms}} & d_{ms} < d_o < d_s \\ 1 & 0 < d_o < d_{ms} \end{cases} \quad (16)$$

where d_s represents the scan range of the agent, d_o means the real-time distance between the agent and the obstacle, while d_{ms} denotes the minimal safe distance between them.

Based on the values of α and β, the unique movement direction of the agent can be determined by the Eq. (2).

At each step, the agent can make a decision based on the memorized environment information, but there exist error between the actual position/orientation relation and the position/orientation relation identified by the agent. If the actual sensory input of the MDN is defined as $p = \{p_1, p_2, p_3, p_4, p_5, p_6\}$, the synaptic weights of the activated neurons in internal layer Y is $w = \{w_1, w_2, w_3, w_4, w_5, w_6\}$, then the recognition error of the agent can be defined as follows:

$$e = \sum_{i=1}^{6} |p_i - w_i| \quad (17)$$

In the following three sections, one static scenario, one dynamic scenario and a comparative environment are conducted to testify whether the knowledge learned by the agent in the off-task processes can lead to the change of its behavior in the following perceptual learning.

C. Simulation in the Static Environment

In the static simulated environment, except the agent and the target, there exist 13 obstacles. First, the MDN model is trained with the training samples. After training, distribution of the internal neurons that memorize the environment knowledge in the layer Y, is illustrated in Fig. 8, where each small blue square represents an internal neuron, and totally, there are 152 (8*19) internal neurons memorize the environment knowledge, where 8 denotes the 8 types of run directions, and 19 represents the 19 types of typical location relationship described in Fig. 6. The horizontal and vertical coordinates in Fig. 8 denote the number of neurons (10000=100*100, arranged in 100 rows and 100 columns) in the internal layer Y.

After the MDN is trained, the agent performs the first run in the static environment to reach the target, and the trajectory in the first run is represented by the blue curve with "+", as shown in Fig. 9, and the horizontal and vertical coordinates in the Fig. 9 denote the size (1000*1000) of the simulation environment. The agent spends 187 steps to reach the target, and its behavior decision is made based on the original 152 training data of the MDN. After the first run, the agent enters the off-task process, to memorize the new environment knowledge learned in the first run.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2021.3128179, IEEE Transactions on Cognitive and Developmental Systems

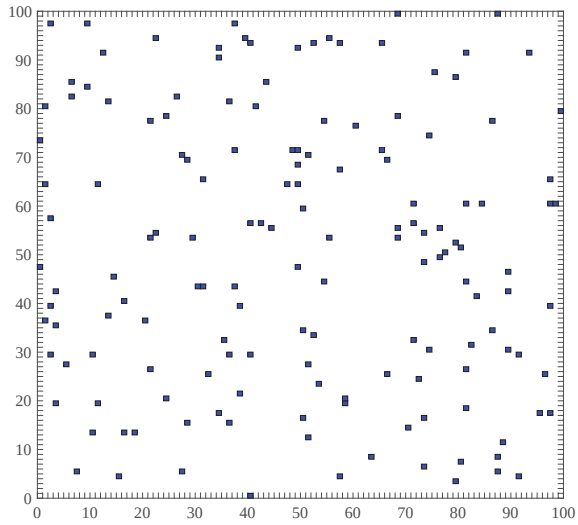


Fig. 8. Distribution of the internal neurons which memorize the environment knowledge in layer Y in the static environment.

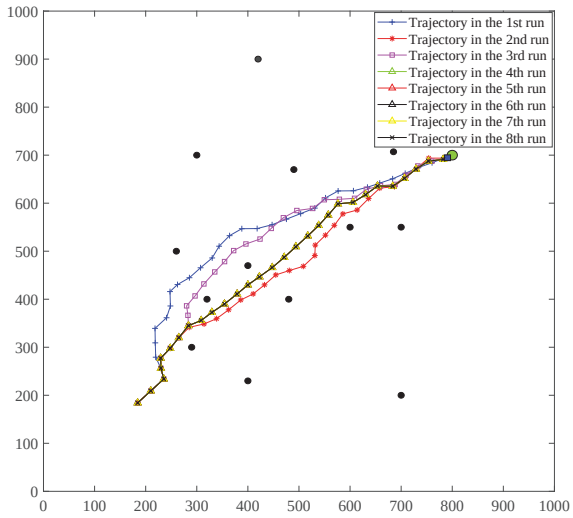


Fig. 9. Trajectories of the agent in the eight movements in static environment.

After the first off-task process, the agent performs the second run to reach the same target in the same environment. The trajectory in the second run is denoted by the red curve with "$*$", as shown in Fig. 9. During this movement, the agent spends 176 steps to reach the same target, and it moves along a different trajectory to reach the target. This phenomenon is rooted in the fact that after the first run, in the off-task process, the agent memorizes the new environment knowledge encountered in its first run. That is to say, the agent abstracts the similar location features and memorizes them in the new activated neurons. Before it performs the second run, the agent has learned new environment knowledge. Therefore, during the following run, if the agent encounters a similar or same environment, it can make a different decision from the pervious run. So in the second run, the agent moves along a different trajectory. Similarly, in the second run, since the agent moves along a different trajectory, it learns new environment knowledge again. After the second run, during the off-task process, the agent memorizes the new environment knowledge in the new activated neurons again. Before it executes the third run, the agent has learned new environment knowledge again. Consequently, the agent moves along a different trajectory denoted by the purple curve in the third run, and it spends 181 steps to reach the same target. Similarly, before the fourth run, the agent has carried out three off-task processes, thus it selects a different trajectory represented by a green curve in the 4th run. During the fifth run, the agent moves along almost the same trajectory as the one in the fourth run, and the agent spends 171 steps to reach the same target. Because the agent is more and more familiar with the environment, then it learns less and less new environment knowledge during reaching the target, from the 5th run, the agent has learned all the environment information, hence the last three trajectories are the same as the fifth one.

Fig. 10 displays the numbers of internal neurons that memorize the environment knowledge after each run. From Fig. 10, we can see that the agent has learned new environment knowledge during the former 5 runs. Meanwhile, the new environment knowledge that the agent learns in each run becomes less and less. This can be attributed to the fact that the obstacles and target are all static, as the run goes on, the agent becomes more and more familiar with the environment, therefore the new environment knowledge it learns becomes less and less.In the last three runs, since the agent has not learned new environment knowledge, the numbers of the internal neurons to memorize new knowledge remain unchanged.

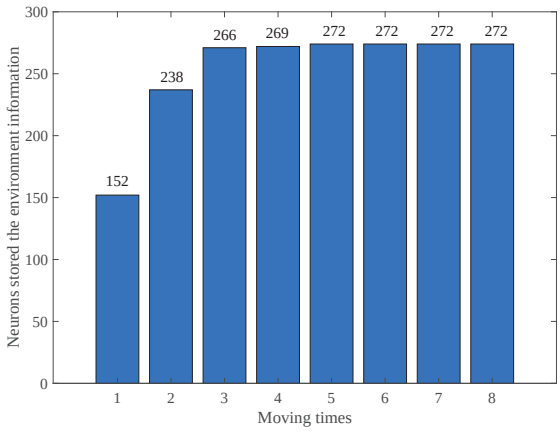


Fig. 10. Numbers of the internal neurons which memorize the environment knowledge in layer Y after the 8 movements.

Fig. 11 shows the distribution of the internal neurons that memorize the environment knowledge in layer Y after the 8th run. Fig. 11 shows that a large amount of blue squares (denoting the internal neurons) get together, which can boil down to the fact that the lateral excitation of the internal neurons in layer Y of the MDN can activate more surrounding

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2021.3128179, IEEE Transactions on Cognitive and Developmental Systems

neurons to memorize new similar environment knowledge. In the following perceptual learning, if the agent encounters a similar scenario, it can determines its run direction quickly by means of the transfer learning, thus to improve the efficiency of its behavior decision. Most important, the agent can continuously improve its intelligence even in the off-line states, i.e., the off-task processes.

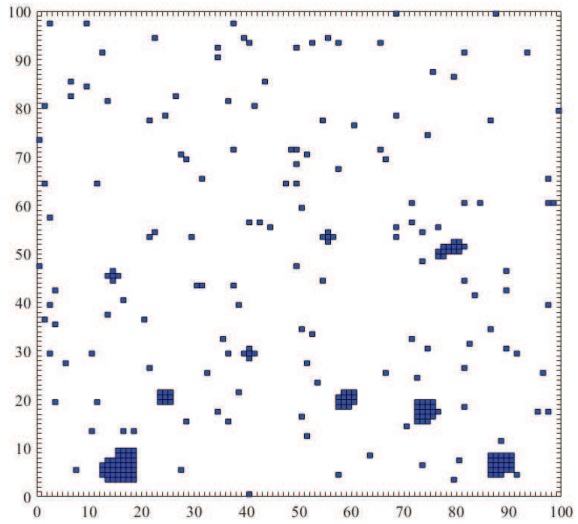


Fig. 11. Distribution of the internal neurons which memorize the environment knowledge in layer Y in static environment after the 8th movement.

Fig. 12 provides the broken lines of the recognition error by the agent after each run. During each run, the recognition error can be calculated by the Eq. (13) and the corresponding results are offered as follows: 0.8602, 0.3663, 0.2179, 0.2444, 0.2310, 0.2310, 0.2310 and 0.2310. Each point on the broken lines represents a recognition error of the agent at each step. From Fig. 12, we can see that at the former three runs, after each run, the recognition error will decrease a little. At the third run, the recognition error gets the smallest value, i.e., 0.2179. At the last four runs, the recognition errors are about 0.23. With the run going on, the agent becomes more and more familiar with the environment, so the recognition error becomes smaller and smaller. After the fourth run, the agent has become very familiar with the environment. Therefore, the recognition errors of the agent keep almost unchanged after the 4th run.

D. Simulation in the Dynamic Environment

Similar with the simulation in the static environment, applying the lateral excitation in the off-task process of the agent to the dynamic environment, we can get the similar results, as displayed from Fig. 13 to Fig. 16.

In the dynamic environment, besides the static obstacles, there are three dynamic obstacles denoted by the unfilled circles with a radium 6. To compare the effect of the transfer learning in the sequential tasks, we set the following experiment process: the agent executes the first run in a static scenario (the target and all the obstacles are static), and from

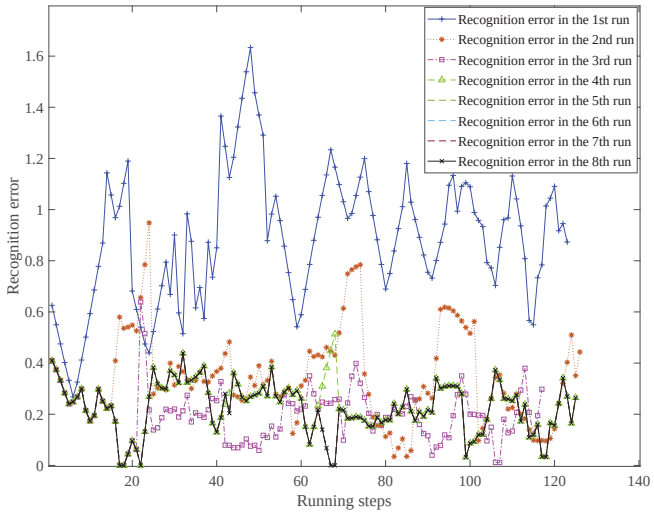


Fig. 12. Scheme of the average recognition error of the agent in the 8 movements in the static environment.

the 2nd to the 19th run in dynamic scenarios (dynamic target and three dynamic obstacles, other obstacles are static), while the 20th run still in a static scenario, same as the 1st run. Thus we can compare the changes of the trajectories of the agent after the off-task process. The trajectories in the 20 runs are displayed in the Fig. 13.

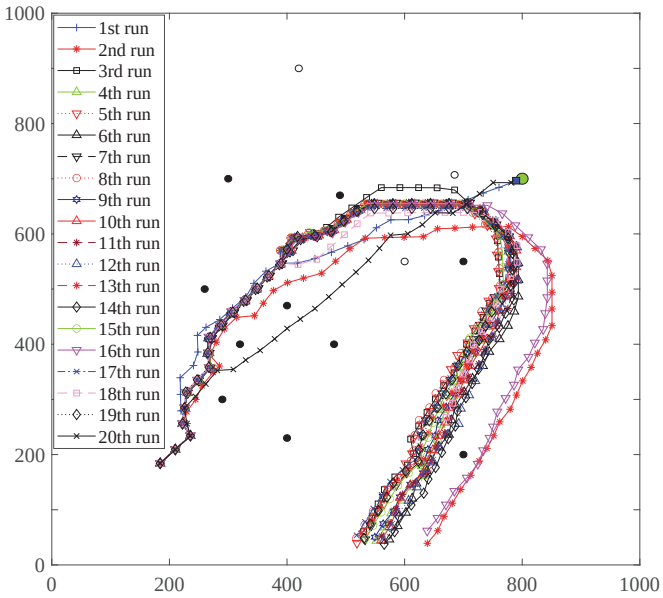


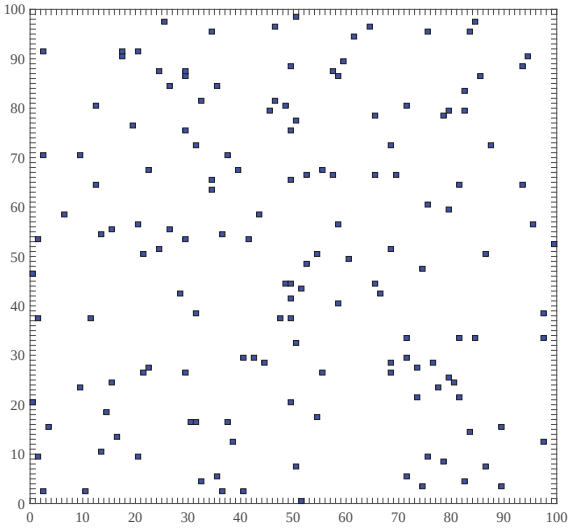
Fig. 13. Trajectories of the agent in the 20 movements in dynamic environment.

Fig. 13 shows that after 4 times learning in the off-task process, the agent grasps more knowledge about the scenario, so from the 5th run, it can catch up the target faster (shorter moving paths), compared with the 1st run. Concretely, the running steps of the agent in the former 5 runs are 187, 314, 140, 180 and 171, respectively. Compared with the 1st run, the running step in the 5th run reduces 16 steps, which indicates

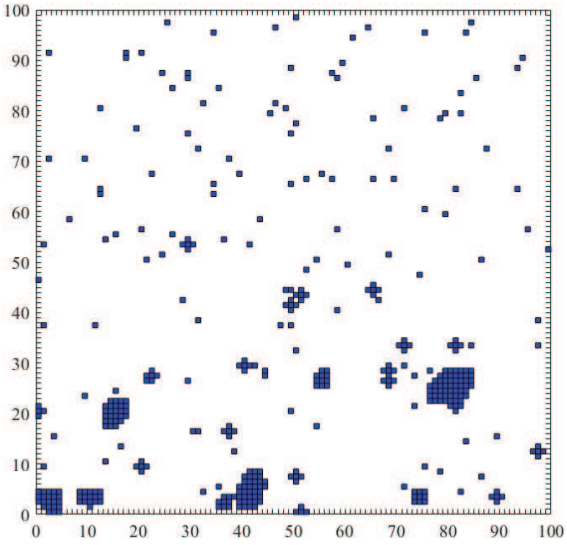
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2021.3128179, IEEE Transactions on Cognitive and Developmental Systems

that the agent has learned new knowledge in the former 4 runs, and the transfer learning happened in the dynamic environment.

Fig. 14 provides the distribution of the neuron memorizing the environment knowledge before and after the 20 runs. It is obvious that there are some neurons gathering together, which indicate the effect of the lateral excitation of the internal neurons on the MDN again, and the knowledge stored in the neurons are new environment knowledge that the agent learned during the off-task processes. Compared with the Fig. 11, intuitively, there are more internal neurons aggregate together, which indicates that in the dynamic environment, there are more internal neurons needed to memorize the new environment knowledge.



(a)



(b)

Fig. 14. Distribution of the internal neurons which memorize the environment knowledge in the layer Y in dynamic environment (a) before the movement (b) after the 20th movement.

In dynamic environment, number of neurons that memorize new environment knowledge will increase with the running times, as shown in Fig. 15. Since the environment is always changing, during each run, the agent will encounter new environment information and learn new knowledge. Correspondingly, the numbers of the neuron to store new knowledge in the internal layer increase approximately logarithmically.

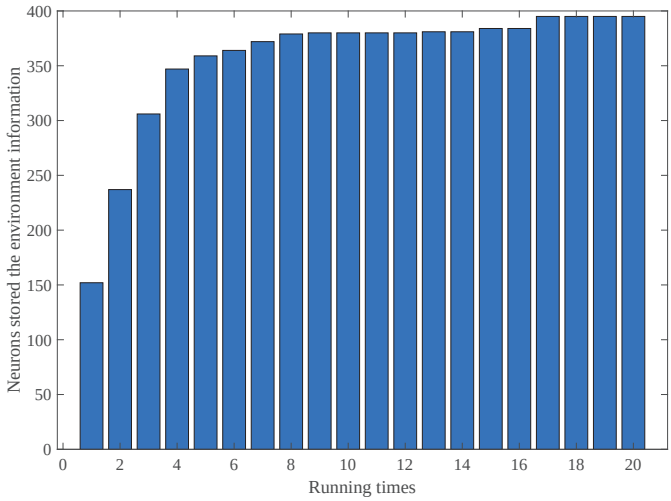


Fig. 15. Number change of the internal neurons which memorize the environment knowledge in layer Y in the 20 movements in dynamic environment.

For simplicity and clearity, Fig. 16 only provides the broken lines of the recognition error in the 1st and 20th runs in dynamic environment. The recognition errors in the 20 runs are 0.8859, 0.5336, 0.3493, 0.2720, 0.2374, 0.2426, 0.2008, 0.2086. 0.2018, 0.2059, 0.2137, 0.2146, 0.2081, 0.2153, 0.1999, 0.2543, 0.1988, 0.2186, 0.1936 and 0.2440, respectively. Since the agent moves in different trajectory in each run, and it achieves new environment knowledge after each run, hence the recognition error are always changing. But with the increase of the run times, the agent becomes gradually familiar with the changing environment, thus the recognition error reduces accordingly.

E. Comparative Experiment

To further demonstrate the potential of the methodology proposed in this work, this section compares the final trajectories of the three methods, i.e., the MDN proposed in this work, original DN [52], and DQN [29]. All these three methods independently run five times to compare the final trajectories, as shown in Fig. 17, where the green triangle denotes the destination, and the blue pentagram denotes the agent. During the run, the performance parameters of the three algorithms are recoded and shown in Table 1.

From Fig. 17 and Table 1, we can see that compared with the original DN, the proposed MDN algorithm has fewer moving steps (32 vs 44), and the final average recognition error is lower than that of the original DN (0.2458 vs 0.2986). As the proposed MDN method has the ability of transfer learning, it can be seen from the data in the 4th column in the Table 1 that the number of training samples needed by the proposed

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2021.3128179, IEEE Transactions on Cognitive and Developmental Systems

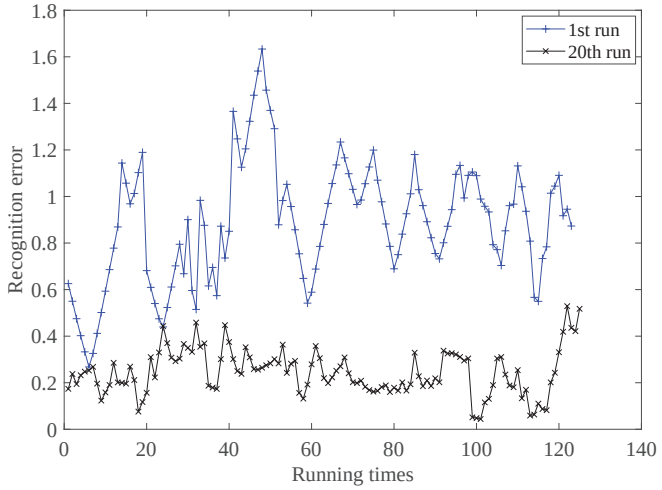


Fig. 16. Schematic diagram of the recognition error in the 1st and 20th runs of the agent in the dynamic environment.

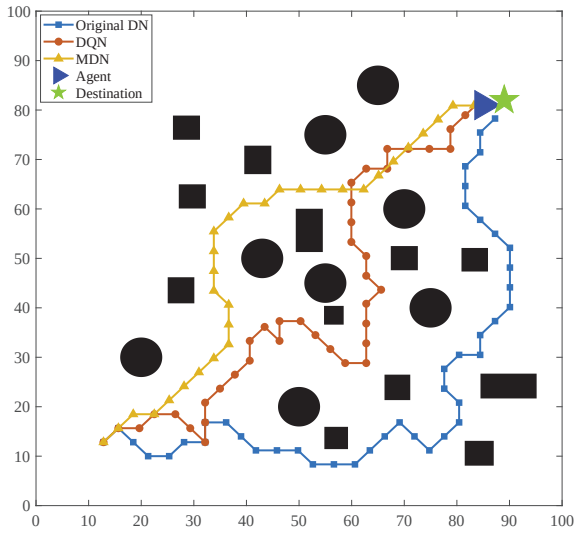


Fig. 17. Schematic diagram of the performance comparison among the original DN, DQN and our proposed MDN method.

TABLE I
EXPERIMENT RESULTS COMPARISON AMONG THE THREE METHODS.

Methods	Running steps	Recognition error	Training samples	Run times to get stable path
MDN	32	0.2458	150	4
Original DN	44	0.2986	500	no
DQN	42	no	0	29

MDN method is far less than that of the original DN. In cases with a small amount of training samples, the MDN method finally can still find a feasible short path, marked by triangles in the Fig. 17. It is rooted in the fact that the proposed MDN method has the unique cognitive transfer ability, when the agent encounters similar environmental information, it will temporarily memorize this information. During the off-task process, the agent can perform the transfer learning according to the similar characteristics, and generate new knowledge inside the MDN, that is, the amount of knowledge inside the MDN presents an incremental pattern, thus the agent can constantly improve its decision-making ability. Consequently, the agent only needs four runs to get a stable path. This index does not exist in the original DN, because the original DN chooses the same path in each run, and there is no adjustment of decision.

For the complex static environment, compared with the DQN, the proposed MDN algorithm not only has the less running steps to reach the target (32 vs 42), but also the faster convergence speed. That is, the MDN needs only 4 attempts to find the stable path, while the DQN needs 29 attempts. But the advantage of the DQN is that it does not need training samples, and it can learn decisions through continuous interaction with the environment. The proposed MDN method needs training by a small number of samples, but it can continuously improve its learning ability by means of interacting with the environment, thus accelerating its convergence speed.

VI. EXPERIMENT WITH THE ROS SYSTEM

To demonstrate the effect of the transfer learning, a real robot, as shown in Fig. 18, is used to do the experiment. It uses the laser radar to sense the environment and odometer to record the location. Therefore, the real-time location and velocity of the agent can be achieved.

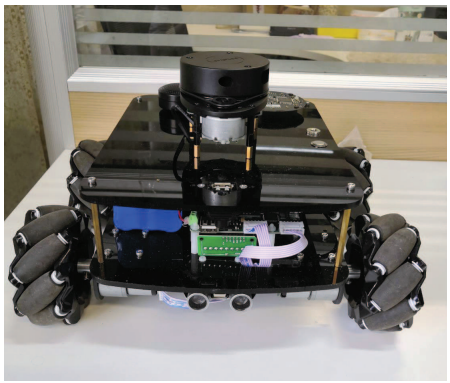


Fig. 18. Mobile robot used in the experiment.

There are 12 static obstacles in the environment. The initial positions of the robot and the static obstacles are shown in Fig. 19. Initial position of the robot is located at the (0, 0) of the global coordinate system (i.e., the odometer coordinate system), and the robot is set to be the origin of the coordinate system, and its running direction is set to be the X axis of the coordinate system, while the direction perpendicular to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2021.3128179, IEEE Transactions on Cognitive and Developmental Systems

the running direction of the robot is set to be the Y axis. This coordinate system is fixed on the robot and moves with the robot's running.

Fig. 19. Initial positions of the agent and the obstacles in the physical experiment.

After starting the robot, the MDN is trained to control the robot. Because the experiment is performed in a static scenario, after training, distribution of the internal neurons which memorize the environment knowledge is the same as that in the Fig. 8.

The experiment environment is set to be 5×5, and the target is set to be located at (3.5, -2.7). During the movement, the robot can learn new environment knowledge continuously. The run trajectories of the robot are drawn and shown in Fig. 20. The blue square denotes the robot, while the black circles and black rectangles denote the obstacles. Since the robot encounters similar environment information in each run, and learns these new environment knowledge in the off-task process, thus its run trajectory in each run is different. Meanwhile, the trajectory turns to be better gradually. Fig. 21 provides the monitoring interface of the robot at its initial run.

After the 8th run, the distribution of the neurons that store the environment knowledge is shown in Fig. 22. Fig. 22 displays that lateral excitation of the internal neurons in layer Y indeed happens during the off-task processes.

Fig. 23 provides the neurons which store the environment knowledge during the robot's running. We can see that with the increase of the robot running, knowledge stored in the internal neurons increases gradually and tends to be stable, which denotes that the new knowledge the robot learned becomes less and less.

Moreover, Fig. 24 displays the broken lines of the recognition error in the 8 runs. The recognition error in the 8 runs are 0.4597, 0.3325, 0.3216, 0.2897, 0.2928, 0.2634, 0.2683 and 0.2618, respectively. From Fig. 24, we can see that the recognition error reduces gradually and tends to be stable. Because the environment is static and the robot moves in the same scenario each time, so it is more and more familiar with the environment, leading to smaller and smaller recognition

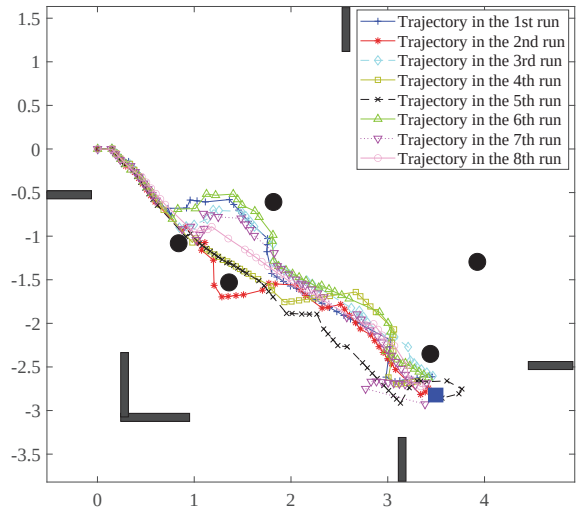


Fig. 20. Run trajectories of the robot in the 8 runs in the physical experiment.

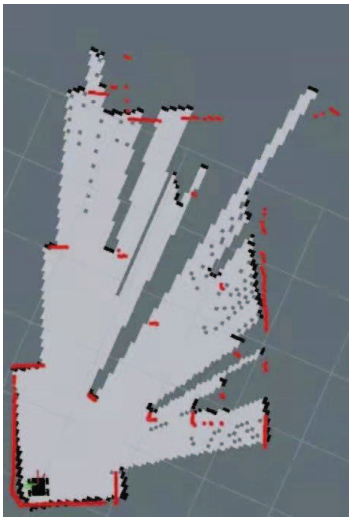


Fig. 21. The monitoring interface of the robot during its run in the physical experiment.

error. During the 5th run, the recognition errors increase a little, which can be attributed to the fact that the robot collides with an obstacle, leading to it deviating from the original trajectory, thus its moving steps also increases accordingly.

Taken together, we can achieve the results shown in Table 2. Table 2 shows that with the run times increase, basically, the total running steps of the robot reduces gradually, and the neurons that store the environment knowledge increases, while the recognition error decreases. These results show that after each run, the robot executes the off-task process to learn new knowledge. In the following run, when the robot encounters a similar scenario, it can decide its run direction by the transfer learning. Based on the knowledge it has learned, the robot can make quick and better decision.

The above four experiments demonstrate that the robot indeed achieves the incremental learning ability through the transfer learning.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2021.3128179, IEEE Transactions on Cognitive and Developmental Systems

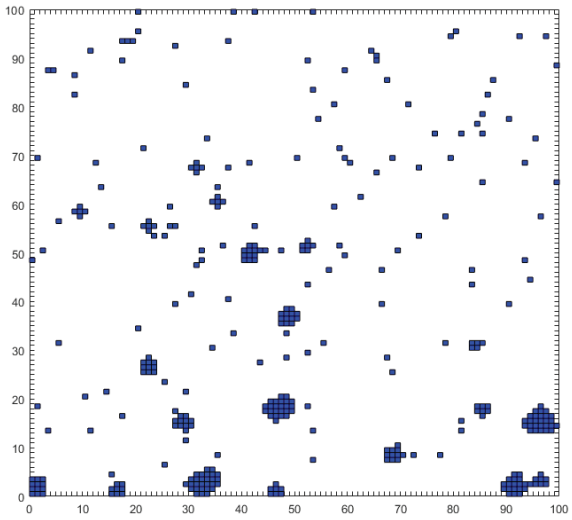


Fig. 22. Distribution of the internal neurons memorized the environment knowledge after 8 movements in the physical experiment.

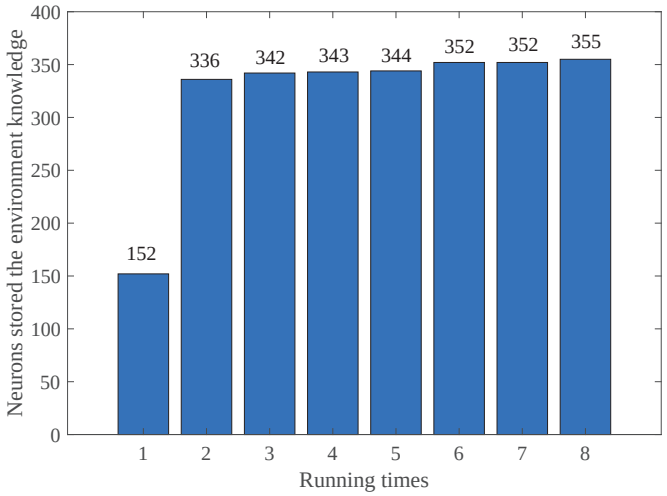


Fig. 23. Number change of the internal neurons which memorize the environment knowledge in internal layer Y in the 8 runs in the physical experiment.

TABLE II
COMPARISON AMONG THE 8 RUNS IN THE EXPERIMENT

Running times	Total running steps	Neurons stored knowledge	Recognition error
1st movement	51	152	0.4597
2nd movement	48	336	0.3325
3rd movement	46	342	0.3216
4th movement	43	343	0.2897
5th movement	56	344	0.2928
6th movement	46	352	0.2634
7th movement	46	352	0.2683
8th movement	39	355	0.2618

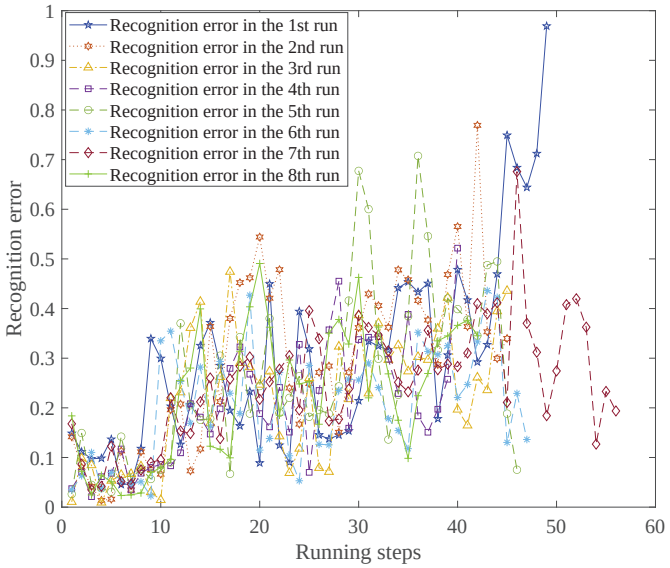


Fig. 24. Scheme of the recognition error in the 8 runs of the robot in the experiment.

VII. CONCLUSION AND FUTURE WORK

To promote the efficiency of behavior decision of the mobile robots, this paper first designs a gated self-organization mechanism to trigger the "brain" of the mobile robot to work again in off-line states, then the lateral excitation in the internal neurons of the MDN is adopted to activate more surrounding neurons to memorize the scenario and the corresponding movement direction during the off-task processes. In the following environment perceptual learning, if the robot encounters a similar or same situation, it can determine its movement direction quickly, according to the knowledge it has learned in the off-task processes, thus to promote the efficiency of the behavior decision. Moreover, the proposed algorithm can continuously improve the intelligence of the mobile robots, even in their off-line states. Results of an autonomous robot navigation in static and dynamic simulation environments, as well as that of the physical experiment, demonstrate the potential of the proposed algorithm.

But this paper only studies the effect of the proposed algorithm in simple environments. In the future, more complex scenarios will be designed to illustrate the advantages of the proposed algorithm. Furthermore, in determining the activated motor neurons in layer Z, only the activated motor neurons in the last previous work are considered, and those in former works have not considered. Theoretically, all the activated motor neurons in the previous works should have chance to fire in the off-task processes. Therefore, how to investigate the influence of the time in calculating the fire probability of the motor neurons, is another important research direction. In the physical experiment, collision occurs occasionally, at the same time, the cumulative detection error of the lidar affects the performance of the trajectory. Therefore, how to perfect the algorithm and decrease the influence of detection error of

the lidar simultaneously, is another important and interesting research point.

ACKNOWLEDGMENTS

This research is supported by the National Natural Science Funds of China under Grant No. 62173309, 62173311, 61873245, Natural Science Funds of Henan Province under Grant No. 202300410483, Scientific Problem Tackling of Henan Province under Grant No. 192102210256, and the Henan Province College Youth Backbone Teacher Project under Grant No. 2021GGJS001.

REFERENCES

[1] K. C. Aberg, E. M. Tartaglia, and M. H. Herzog. Perceptual learning with chevrons requires a minimal number of trials, transfers to untrained directions, but does not require sleep. *Vision Research*, 49:2087–2094, 2009.

[2] S. A. Ahmed, A. V. Topalov, N. G. Shakev, and V. L. Popov. Model-free detection and following of moving objects by an omnidirectional mobile robot using 2d range data. *IFAC PapersOnLine*, 51-22:226–231, 2018.

[3] M. Akram and A. Raza. Towards the development of robot immune system: A combined approach involving innate immune cells and t-lymphocytes. *Biosystems*, 172:52–67, 2018.

[4] L. Aubin, M. Khamassi, and B. Girard. *Prioritized Sweeping Neural DynaQ with Multiple Predecessors, and Hippocampal Replays*. Springer, Cham, Switzerland, 2018.

[5] R. Caze, M. Khamassi, L. Aubin, and B. Girard. Hippocampal replays under the scrutiny of reinforcement learning models. *Journal of neurophysiology*, 120(6):2877–2896, 2018.

[6] S. Choi, E. Kim, K. Lee, and S. Oh. Real-time nonparametric reactive navigation of mobile robots in dynamic environments. *Robotics and Autonomous Systems*, 91:11–24, 2017.

[7] W. Fang, F. Chao, L. Yang, C. M. Lin, C. Shang, C. Zhou, and Q. Shen. A recurrent emotional cmac neural network controller for vision-based mobile robots. *Neurocomputing*, 334:227–238, 2019.

[8] S. Hacohen, S. Shoval, and N. Shvalb. Applying probability navigation function in dynamic uncertain environments. *Robotics and Autonomous Systems*, 87:237–246, 2017.

[9] H. Harris, M. Gliksberg, and D. Sagi. Generalized perceptual learning in the absence of sensory adaptation. *Current Biology*, 22(19):1813–1817, 2012.

[10] W. He, Z. Li, and C. L. P. Chen. A survey of human-centered intelligent robots: Issues and challenges. *IEEE/CAA Journal of Automatica Sinica*, 4(4):602–609, 2017.

[11] B. Hermans, P. Patrinos, and G. Pipeleers. A penalty method based approach for autonomous navigation using nonlinear model predictive control. *IFAC PapersOnLine*, 51-20:234–240, 2018.

[12] A. Hidalgo-Paniaguaa, M. A. Vega-Rodriguez, J. Ferruz, and N. Pavon. Mosfla-mrpp: Multi-objective shuffled frog-leaping algorithm applied to mobile robot path planning. *Engineering Applications of Artificial Intelligence*, 44:123–136, 2015.

[13] M. A. Hossain and I. Ferdous. Autonomous robot path planning in dynamic environment using a new optimization technique inspired by bacterial foraging technique. *Robotics and Autonomous Systems*, 64:137–141, 2015.

[14] P. E. Jeter, B. A. Dosher, S. Liu, and Z. Lu. Specificity of perceptual learning increases with increased training. *Vision Research*, 50:1928–1940, 2010.

[15] P. E. Jeter, B. A. Dosher, A. Petrov, and Z. Lu. Task precision at transfer determines specificity of perceptual learning. *Journal of Vision*, 9:1–13, 2009.

[16] S. Kakade and P. Dayan. Dopamine: generalization and bonuses. *Neural Networks*, 15(6):549–559, 2002.

[17] F. Kattner, A. Cochrane, C. R. Cox, T. E. Gorman, and C. S. Green. Perceptual learning generalization from sequential perceptual training as a change in learning rate. *Current Biology*, 27:840–846, 2017.

[18] M. Khamassi and B. Girard. Modeling awake hippocampal reactivations with model-based bidirectional search. *Biological Cybernetics*, 114:231–248, 2020.

[19] H. Kumano and T. Uka. Neuronal mechanisms of visual perceptual learning. *Behavioural Brain Research*, 249:75–80, 2013.

[20] P. B. Kumar, C. Sahu, and D. R. Parhi. A hybridized regression-adaptive ant colony optimization approach for navigation of humanoids in a cluttered environment. *Applied Soft Computing*, 68:565–585, 2018.

[21] S. Kundu and D. R. Parhi. Navigation of underwater robot based on dynamically adaptive harmony search algorithm. *Behavioural Brain Research*, 8(2):125–146, 2016.

[22] G. Lange and P. D. Weerd. Limited transfer of visual skill in orientation discrimination to locations treated by pre-testing and subliminal exposure. *Vision Research*, 143:103–116, 2018.

[23] C. T. Law and J. I. Gold. Neural correlates of perceptual learning in a sensory motor, but not a sensory cortical area. *Nature Neuroscience*, 11(4):505–513, 2008.

[24] J. Liang, Y. Zhou, M. Fahle, and Z. Liu. Limited transfer of long-term motion perceptual learning with double training. *Journal of Vision*, 15(10):1–9, 2015.

[25] T. Lin, C. Chen, and C. Lin. Wall-following and navigation control of mobile robot using reinforcement learning based on dynamic group artificial bee colony. *Journal of Intelligent & Robotic Systems*, 92:343–357, 2018.

[26] E. S. Low, P. Ong, and K. C. Cheah. Solving the optimal path planning of a mobile robot using improved q-learning. *Robotics and Autonomous Systems*, 115:143–161, 2019.

[27] T. Mastropasqua, J. Galliussi, D. Pascucci, and M. Turatto. Location transfer of perceptual learning: Passive stimulation and double training. *Vision Research*, 98:93–102, 2015.

[28] K. E. Merrick. A comparative study of value systems for self-motivated exploration and learning by robots. *IEEE Transactions on Autonomous Mental Development*, 2(2):119–131, 2010.

[29] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, and M. G. Bellemare. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015.

[30] J. C. Mohanta and A. Keshari. A knowledge based fuzzy-probabilistic roadmap method for mobile robot navigation. *Applied Soft Computing Journal*, 79:391–409, 2019.

[31] U. Orozco-Rosas, K. Picos, and O. Montiel. Hybrid path planning algorithm based on membrane pseudo-bacterial potential field for autonomous mobile robots. *IEEE Access*, 7:156787–156803, 2019.

[32] M. R. Panda, S. Dutta, and S. Pradhan. Hybridizing invasive weed optimization with firefly algorithm for multi-robot motion planning. *Arabian Journal for Science and Engineering*, 43(8):4029–4039, 2018.

[33] B. K. Patle, A. Pandey, A. Jagadeesh, and D. R. Parhi. Path planning in uncertain environment by using firefly algorithm. *Defence Technology*, 14(6):691–701, 2018.

[34] B. K. Patle, D. R. Parhi, A. Jagadeesh, and S. K. Kashyap. Matrix-binary codes based genetic algorithm for path planning of mobile robot. *Computers & Electrical Engineering*, 67:708–728, 2018.

[35] M. Pavlovskaya and S. Hochstein. Perceptual learning transfer between hemispheres and tasks for easy and hard feature search conditions. *Journal of Vision*, 11(1):1–13, 2011.

[36] V. L. Popov, S. A. Ahmed, A. V. Topalov, and N. G. Shakev. Development of mobile robot target recognition and following behaviour using deep convolutional neural network and 2d range data. *IFAC PapersOnline*, 51-30:210–215, 2018.

[37] D. C. Rao, M. R. Kabat, P. K. Das, and P. K. Jena. Cooperative navigation planning of multiple mobile robots using improved krill herd. *Arabian Journal for Science and Engineering*, 43:7869–7891, 2018.

[38] E. Roszkowska, P. Dulewicz, and L. Janiec. Hierarchical hybrid control for multiple mobile robot systems. *IFAC PapersOnLine*, 52-8:452–457, 2019.

[39] M. Saraswathia, G. B. Murali, and B. B. V. L. Deepak. Optimal path planning of mobile robot using hybrid cuckoo search-bat algorithm. *Procedia Computer Science*, 133:510–517, 2018.

[40] M. Solgi, T. Liu, and J. Weng. A computational developmental model for specificity and transfer in perceptual learning. *Journal of Vision*, 13(1):ar. 7, 1–23, 2013.

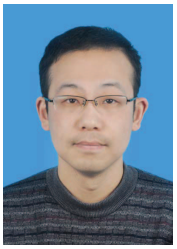
[41] G. Sotiropoulos, A. R. Seitz, and P. Series. Performance-monitoring integrated reweighting model of perceptual learning. *Vision Research*, 152:17–39, 2018.

[42] P. Surez, A. Iglesias, and A. Glvez. Make robots be bats: Specializing robotic swarms to the bat algorithm. *Swarm and Evolutionary Computation*, 44:113–129, 2019.

[43] T. P. Tunggal, A. Supriyanto, R. N. M. Zaidatur, I. Faishal, I. Pambudi, and T. Iswanto. Pursuit algorithm for robot trash can based on fuzzy-cell decomposition. *International Journal of Electrical and Computer Engineering*, 6(6):2863–2869, 2016.
[44] T. Uka, R. Sasaki, and H. Kumano. Change in choice-related response modulation in area mt during learning of a depth-discrimination task is consistent with task learning. *Journal of Neuroscience*, 32(40):13689–13700, 2012.
[45] D. Wang, Y. Duan, and J. Weng. Motivated optimal developmental learning for sequential tasks without using rigid time-discounts. *IEEE Transactions on Neural Networks and Learning Systems*, 29(10):4917–4931, 2018.
[46] D. Wang, D. Tan, and L. Liu. Particle swarm optimization algorithm: An overview. *Soft Computing*, 22(2):387–408, 2018.
[47] D. Wang, H. Wang, and L. Liu. Unknown environment exploration of multi-robot system with the fordpso. *Swarm and Evolutionary Computation*, 26:157–174, 2016.
[48] R. Wang, L. Cong, and C. Yu. The classical tdt perceptual learning is mostly temporal learning. *Journal of Vision*, 13(5):1–9, 2013.
[49] R. Wang, J. Zhang, S. A. Klein, D. M. Levi, and C. Yu. Task relevancy and demand modulate double-training enabled transfer of perceptual learning. *Vision Research*, 61:33–38, 2012.
[50] R. Wang, J. Zhang, S. A. Klein, D. M. Levi, and C. Yu. Vernier perceptual learning transfers to completely untrained retinal locations after double training: A "piggybacking" effect. *Journal of Vision*, 14(13):1–12, 2014.
[51] J. Weng. Why have we passed neural networks no not abstract well. *Natural Intelligence: the INNS Magazine*, 1(1):13–22, 2011.
[52] J. Weng. *Natural and Artificial Intelligence: Introduction to computational brain-mind*. BMI press, Okemos, Michigan, USA, 2012.
[53] J. Weng, J. McClelland, A. Pentland, O. Sporns, I. Stockman, M. Sur, and E. Thelen. Autonomous mental development by robots and animals. *Science*, 291(5504):599–600, 2001.
[54] L. Xiao, J. Zhang, R. Wang, S. A. Klein, D. M. Levi, and C. Yu. Complete transfer of perceptual learning across retinal locations enabled by double training. *Current Biology*, 18(24):1922–1926, 2008.
[55] X. Xie and C. Yu. Double training downshifts the threshold vs. noise contrast (tvc) functions with perceptual learning and transfer. *Vision Research*, 152:3–9, 2018.
[56] Y. Xiong, X. Xie, and C. Yu. Location and direction specificity in motion direction learning associated with a single-level method of constant stimuli. *Vision Research*, 119:9–15, 2016.
[57] X. Yu, W. He, H. Li, and J. Sun. Adaptive fuzzy full-state and output-feedback control for uncertain robots with output constraint. *IEEE Transactions on Systems, Man and Cybernetics: Systems*, http://dx.doi.org/10.1109/TSMC.2019.2963072, 2020.
[58] J. Zhang and Y. Yang. Perceptual learning of motion direction discrimination transfers to an opposite direction with tpe training. *Vision Research*, 99:93–98, 2014.
[59] J. Zhang and Y. Yang. Perceptual learning of motion direction discrimination transfers to an opposite direction with tpe training. *Vision Research*, 99:93–98, 2014.
[60] J. Zhang, G. Zhang, L. Xiao, S. A. Klein, D. M. Levi, and C. Yu. Rule-based learning explains visual perceptual learning and its specificity and transfer. *Journal of Neuroscience*, 30:12323–12328, 2010.
[61] T. Zhang, L. Xiao, S. A. Klein, D. M. Levi, and C. Yu. Decoupling location specificity from perceptual learning of orientation discrimination. *Vision Research*, 50:368–374, 2010.

Dongshu Wang received his Bachelors degree in Mechanical Manufacture Technique and Equipment in 1996, Masters degree in Mechanical Manufacture and Automation in 2002, and PhD in Control Theory and Control Engineering in 2006 from Northeastern University, China. His research domain is autonomous mental development, artificial intelligence, machine learning and bio-inspired computation. Currently, he is an Associate Professor in School of Electrical Engineering, Zhengzhou University, Zhengzhou, China.

Kai Yang received his Bachelors degree in Automation from Zhengzhou University of Aeronautics in 2017 and MSc degree in Control Science and Engineering from Zhengzhou University in 2021 both in China. Now he works in Henan Branch of China Mobile Communications Group Co. Ltd, Zhengzhou. His research domain includes artificial intelligence, robot intelligent control.



Jianbin Xin obtained BSc. degree in Electrical Engineering from Xidian University in 2007 and MSc degree in Control Science and Engineering from Xian Jiaotong University in 2010 both in China. In 2015, he received Ph.D. degree from Department of Maritime and Transport Technology from Delft University of Technology, the Netherlands.

Currently, he is an Associate Professor in School of Electrical Engineering, Zhengzhou University, Zhengzhou, China. His research interests include planning and control of smart logistics systems and cooperative robots.