Robotica (2023), **41**, pp. 690–712
doi:10.1017/S0263574722001539

RESEARCH ARTICLE

Autonomous robot navigation based on a hierarchical cognitive model

Jianxian Cai[1,2], Fenfen Yan[1,*], Yan Shi[1,2], Mengying Zhang[3] and Lili Guo[1,2]

[1]Institute of Disaster Prevention, Sanhe 065201, China, [2]Hebei Key Laboratory of Seismic Disaster Instrument and Monitoring Technology, Sanhe 065201, China, and [3]College of Electronic Science and Control Engineering, Institute of Disaster Prevention, Sanhe 065201, China
*Corresponding author. E-mail: 337856275@qq.com

Received: 14 May 2022; **Revised:** 9 September 2022; **Accepted:** 26 September 2022;
First published online: 15 November 2022

Keywords: hierarchical cognitive model, complex environment, mobile robot navigation, cognitive learning

Abstract

We propose a hierarchical cognitive navigation model (HCNM) to improve the self-learning and self-adaptive ability of mobile robots in unknown and complex environments. The HCNM model adopts the divide and conquers approach by dividing the path planning task into different levels of sub-tasks in complex environments and solves each sub-task in a smaller state subspace to decrease the state space dimensions. The HCNM model imitates animal asymptotic properties through the study of thermodynamic processes and designs a cognitive learning algorithm to achieve online optimum search strategies. We prove that the learning algorithm designed ensures that the cognitive model can converge to the optimal behavior path with probability one. Robot navigation is studied on the basis of the cognitive process. The experimental results show that the HCNM model has strong adaptability in unknown and environment, and the navigation path is clearer and the convergence time is better. Among them, the convergence time of HCNM model is 25 s, which is 86.5% lower than that of HRLM model. The HCNM model studied in this paper adopts a hierarchical structure, which reduces the learning difficulty and accelerates the learning speed in the unknown environment.

1. Introduction

The ability to autonomous navigation, in which the robot can learn and accumulate knowledge in real environments for selecting optimal behavior autonomously, is a prerequisite for mobile robots to perform tasks smoothly in all applications [1]. Currently, several excellent methods have been proposed for autonomous navigation, such as fuzzy logic [2], genetic algorithms [3, 4], random trees [5], and neural networks [6–8]. However, these methods usually need to assume complete environmental configuration information, which has to be adapted by agents in a large number of practical applications. Therefore, how to improve the self-learning ability and adaptability of robot navigation in an unknown environment has become a key technology for scholars to study. In particular, the self-learning ability and self-adaptability of robots are determined by the intelligence level of a robot's perception and response to the environment determines. Cognition and learning ability are the main ways in which humans and animals acquire knowledge, and it is also a significant sign of their intelligence. Therefore, many scholars have begun to simulate biological cognition and behavioral learning models in order to conduct extensive research on mobile robot navigation systems [9–12].

Reinforcement learning algorithm is the optimal strategy to approach the target by interactive learning by maximizing the cumulative reward of agents in the environment. The algorithm is a model of "closed-loop learning" paradigm. Reinforcement learning is applied to the learning ability of robots in unknown environments. At present, reinforcement learning has been widely used in robot autonomous navigation and has achieved many important results. Xu *et al*. [13]. proposed a reactive navigation method for mobile robots based on reinforcement learning and successfully applied it to the CIT-AVT-VI mobile robot platform; Wen *et al.* [14] designed Q-learning obstacle avoidance algorithm based on KEF-SLAM for NAO robot autonomous walking in unknown environment; Cherroun *et al.* [15] studied autonomous navigation based on fuzzy logic and reinforcement learning. Although the reinforcement learning-based navigation method can successfully navigate autonomously, the reinforcement learning Q-learning learning information is stored in the Q table and needs to be updated continuously.

In 1938, Skinner first proposed the concept of operational conditional reflection (OCR) and thus created the theory of OCR. Referring to Pavlov' s concept of "reinforcement," he divided "reinforcement" into positive reinforcement and negative reinforcement. Positive reinforcement increased the response probability of organisms to stimuli, and negative reinforcement increased the response of organisms to eliminate the stimuli. Stimulus produces response, which affects the probability of stimulus, which is the core of Skinner's operational conditioned reflex theory [16]. Since the mid-1990s, Heisenberg *et al.* have been focusing on the computational theory and model of Skinner's OCR [17]. Moreover, Touretzky *et al.* have further developed the computational model of Skinner's OCR theory [18]. Later, many scholars have carried out extensive research on the computational model of operational conditioned reflexes [19–32]. Robots are showing more self-learning ability and self-adaptability, similar to organisms. Zhang *et al.* proposed the Cyr-an outstanding representative, which enables biological agents to learn from the results of previous actions by means of operational conditioned reflexes [19]. Ruan *et al.* Have carried out a series of studies on the operational conditioned reflex model [24–32], including the design and calculation model based on the operational conditioned reflex mechanism, combined with probabilistic automata, neural network, and extended Kalman filter. In 2013, Ruan *et al.* designed a bionic learning model based on operating conditioned reflex learning automata. When applied to self-balancing control and autonomous navigation of two-wheeled robots, the results show that robots can learn autonomously like animals, and their adaptability is better than reinforcement learning. In 2016, Ruan *et al.* designed Skinner-Ransac algorithm based on Skinner's operating conditioned reflex principle and extended Kalman filter, which can simultaneously realize positioning and map creation (SLATM), and the pose estimation results of slam can meet the needs of mobile robot autonomous navigation. In 2018, Cai Jianxian *et al.* designed a cognitive development model for autonomous navigation based on biological cognition and development mechanism. This model can enable robots to simulate animals to automatically acquire knowledge and accumulate experience from unknown environments and acquire the skills of autonomous navigation through cognitive development.

Based on the idea of operational conditioned reflex cognitive model and hierarchy proposed by Ruan *et al.* [24–32], this paper constructs a cognitive model based on operational conditioned reflex and hierarchy to solve the self-learning and adaptive problems of mobile robot autonomous navigation system in unknown complex environment. An operational conditioned reflex cognitive model uses the idea of dividing and conquering; it divides the navigation tasks of mobile robots in complex environments into sub-tasks at different levels and solves each sub-task in a small state subspace in order to reduce the dimension of the state space. A cognitive learning algorithm is designed to simulate the thermodynamic process and achieve the optimal navigation strategy for an online search. Based on the formation of cognitive processes, we investigated autonomous path planning processes of mobile robots. The results show that the hierarchical structure can reduce the learning difficulty and accelerate the learning speed of mobile robots in unknown and complex environments. At the same time, robots can automatically acquire knowledge and accumulate knowledge from the environment, like animals. Experience gradually forms, develops, and perfects the robot's autonomous path planning skills.

2. Basic principles of hierarchical reinforcement learning

Path planning of mobile robots involves finding an optimal path, which enables the robot to reach the target point without collision and optimize performance indicators, such as distance, time, and energy consumption. Distance is the most commonly used criteria. In order to better accomplish this task,

https://doi.org/10.1017/S0263574722001539 Published online by Cambridge University Press

in the process of reinforcement learning, robots need to perform different actions in order to obtain more information and experience and to promote higher future returns. On the other hand, they need to accumulate and execute the current actions with the highest returns according to their own experience. This is called the tradeoff between exploration and exploitation [33]. Too little exploration hinders the convergence of the system in regard to the optimal strategy and too much exploration leads to a higher dimension of the general state space. The size of the state space affects the convergence rate of the bionic learning system. When the state space is too large, mobile robots can spend a lot of time exploring; the learning efficiency is consequently low and it is difficult to converge. In order to solve this problem, a hierarchical reinforcement learning structure is proposed [34]. The aim is to reduce the learning difficulty of the reinforcement learning systems in a complex environment through hierarchical learning. The core idea of the hierarchical reinforcement learning method is to use an abstract method to divide the whole task into different levels of sub-tasks and solve each sub-task in a small state subspace, in order to obtain reusable sub-task strategies and speed up the solution to the problem.

State space decomposition, temporal abstraction, and state abstraction are commonly used techniques in hierarchical reinforcement when a robot is learning to achieve hierarchy. The state space decomposition method divides the state space into several different subspaces and solves the task in the subspace of lower dimensions. The temporal abstraction principle involves grouping the action space set in order to realize the execution of multi-step actions in the process of agent reinforcement learning, thus reducing the consumption of computing resources. The state abstraction method ignores the state space variables that are not related to a sub-task, which reduces the dimensionality of the state space. Although the above three methods layer the system with different methods, they also realize the function of reducing the complexity of the system state space and accelerating the learning speed of the system.

Sequential decision-making problems are usually modeled by Markov Decision Processes (MDP) [35]. When the execution of action strategies extends from a point in time to continuous time, an MDP model also extends to a Semi-Markov Decision Processes (SMDP) model. An SMDP model can solve the problem of learning, which needs to complete action execution in multiple time steps and make up for the deficiency of reinforcement learning, which only assumes that an action is completed in a single time step in the framework of the MDP model. The Bellman optimal equation of value function, based on the SMDP model, is shown in Eq. (1), the Bellman optimal equation of state-action pair function is shown in Eq. (2), and the Q-learning iteration equation is shown in Eq. (3):

$$V^{*}(s) = \max_{a\in A}\left[R(s,a) + \sum_{s',\tau}\gamma^{\tau}P\left(s',\tau\,|s,a\right)V^{*}\left(s'\right)\right] \tag{1}$$

$$Q^{*}(s,a) = R(s,a) + \sum_{s',\tau}\gamma^{\tau}P\left(s',\tau\,|s,a\right)\max_{a\in A}Q^{*}\left(s',a'\right) \tag{2}$$

$$Q_{k+1}(s,a) = (1-\alpha)\,Q_{k}(s,a) + \alpha\left[r_{t} + \gamma r_{t+1} + \cdots + \gamma^{\tau-1}r_{t+\tau-1} + \gamma^{\tau}\max_{a\in A}Q^{*}\left(s',a'\right)\right] \tag{3}$$

In the formula, τ is the random waiting time, indicating the time interval after the agent executes action a in state s; $P(s',\tau|s,a)$ is the transition probability from action a in state s to action a after the agent waits for time τ; and $R(s,a) = E[r_t + \gamma r_{t+1} + \cdots + \gamma^{\tau}r_{t+\tau}]$ is the corresponding reward value.

HRL problems are often modeled based on the SDMP model. HRL (Hierarchical Reinforcement Learning) adopts the strategy of divide and conquer; it divides the planning tasks of agents into sub-tasks at different levels and solves each sub-task in a smaller state subspace, thus realizing the function of reducing the dimension of the state space.

3. Hierarchical structure cognitive model design

3.1. Structure of the cognitive model

The complexity of the working environment, the size of the state space, and the size of the behavior space of a mobile robot affect the robot's learning rate. When the state space is too large, mobile robots

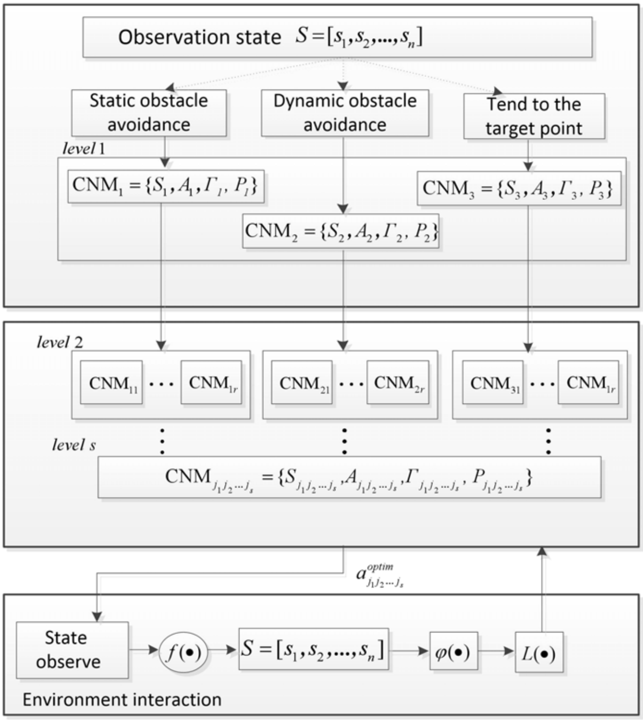

Observation state $S=[s_1,s_2,...,s_n]$
Static obstacle avoidance
Dynamic obstacle avoidance
Tend to the target point
level 1
$\text{CNM}_1=\{S_1,A_1,\Gamma_1,P_1\}$
$\text{CNM}_2=\{S_2,A_2,\Gamma_2,P_2\}$
$\text{CNM}_3=\{S_3,A_3,\Gamma_3,P_3\}$
level 2
CNM_{11} ⋯ CNM_{1r}
CNM_{21} ⋯ CNM_{2r}
CNM_{31} ⋯ CNM_{1r}
level s
$\text{CNM}_{j_1j_2\cdots j_s}=\{S_{j_1j_2\cdots j_s},A_{j_1j_2\cdots j_s},\Gamma_{j_1j_2\cdots j_s},P_{j_1j_2\cdots j_s}\}$
$a^{optim}_{j_1j_2\cdots j_s}$
State observe
$f(\bullet)$
$S=[s_1,s_2,...,s_n]$
$\varphi(\bullet)$
$L(\bullet)$
Environment interaction

Figure 1. *Hierarchical cognitive model.*

will spend a lot of time exploring, resulting in the "dimension disaster" problem; when the behavior space is too large, mobile robots need to try to learn many times, which makes cognitive model learning inefficient and difficult to converge to the optimal strategy. So robots need to acquire the ability to learn in complex and unknown environments.

In view of these practical problems, considering that the path planning task of mobile robots is carried out in an unknown environment with static or dynamic obstacles, the obstacle avoidance problem for static and dynamic obstacles should be considered in the design of the hierarchical cognitive model. In this paper, the path planning tasks of mobile robots are divided into three basic sub-tasks: "static obstacle avoidance" for static obstacles, "dynamic obstacle avoidance" for dynamic obstacles, and "targeting point" motion. Decomposed into small-scale state subspaces, each sub-task is solved in each state subspace independently. Furthermore, considering that the bionic strategy is a trial-and-error learning process, the mobile robot performs complex behaviors. Therefore, the complex behavior strategy is decomposed into a series of simple behavior strategies and independent learning training. The structure of the hierarchical structure cognitive model of the design is shown in Fig. 1.

In Fig. 1, the HCNM learning model includes two functional layers: sub-task selection and behavior decomposition. The sub-task selection layer includes three sub-task selection modules, "static obstacle avoidance," "dynamic obstacle avoidance," and "trend target point." According to the observed environmental state decision output, the corresponding sub-task is selected and the mobile robot is then selected according to the sub-task. The planning strategy selects the behavior and executes it; the behavior set of the sub-task selection layer adopts a roughly divided form. According to the selection result of the sub-task's behavior, the behavior decomposition layer is further finely divided and the number of layers of the refined score is determined according to the degree of complexity of the actual behavior.

Therefore, the HCNM learning model is actually composed of multiple single cognitive systems. If the hierarchical cognitive model is regarded as consisting of seven parts, $\text{HCNM} = <A, S, \Gamma, P, f, \phi, L>$, all CNM learning systems share elements {f, ϕ, L}, and other elements {S, A, Γ, P} are

https://doi.org/10.1017/S0263574722001539 Published online by Cambridge University Press

applied to the corresponding CNM learning system, respectively. Assuming that the number of optional behaviors in the behavior set is r, there are three CNM learning systems: $\{\mathrm{CNM}_1, \mathrm{CNM}_2, \mathrm{CNM}_3\}$ for Level 1 (sub-task selection layer) of the HCNM learning model; Level 2 is the first level of the behavior decomposition layer of the HCNM learning model, and there are 3r CNM learning systems: $\{\mathrm{CNM}_{11}, \ldots, \mathrm{CNM}_{1r}, \mathrm{CNM}_{21}, \ldots, \mathrm{CNM}_{2r}, \mathrm{CNM}_{31}, \ldots, \mathrm{CNM}_{3r}\}$. By analogy, there are 3rs CNM learning systems for the Level s layer. Through the s layer, the corresponding behavior a is selected and used as the environment, and new state information s is observed. Based on this, the orientation evaluation φ of the environment is obtained, and the updating L of learning knowledge is completed. In this way, after many rounds of interactive learning with the environment, our model can make the most optimal decision.

(1) The symbols in the HCNM learning model are defined as follows:

i. f: The state transition function of the HCNM learning model, $f\colon S(t) \times A(t)|P \to S(t+1)$. It shows that the state $s(t+1) \in S$ of $t+1$ time is determined by the state $s(t) \in S$ at time t and the probabilistic operation $\alpha(t)|P \in A$ at time t, which is independent of the state and operation before t time.

ii. ϕ: The HCNM learning model orientation mechanism, $\phi = \{\phi_1, \phi_2 \ldots, \phi_{\mathrm{n}}\}, \phi_i \in \phi$, represents the orientation value of the state $s_i \in S$, which represents the tendency of the state to update the probability vector Pi, satisfying: $0 < \phi_i < 1$.

iii. L: The operation conditioned reflex learning mechanism of the HCNM learning model, $L\colon (t) \to (t+1)$. Update and adjust the probability according to $P(t+1) = L[P(t), \phi(t), a(t)]$.

(2) The elements of the HS-CNM learning models belonging to different CNM learning systems are as follows:

Level 1: S_1, S_2, and S_3 represent the internal discrete state set of three sub-tasks, respectively: $S_{i'} = \{s_{i'i} | \, i' = 1, 2, 3, i = 1, 2, \cdots, n\}$, $S_{i'}$ is a non-empty set composed of all possible discrete states of the control system, and $s_{i'i} \in S_{i'}$indicates that the HCNM system is in the first state at a certain time.$A_{i'}$ represents the behavior set of Level 1 hierarchy tasks: $A_{i'} = \{\alpha_{i'j_1} | j_1 = 1, 2, \cdots, r\}$, $\alpha_{i'j_1}$ is the j_1 operation of sub-task i'. $\Gamma_{i'}$ represents a set of conditional state-action random mappings for Level 1 sub-tasks. It means that $\mathrm{CNM}_{i'}$ implements operation $\alpha_{i'j_1} \in A_{i'}$ according to probability $P_{i'}$ under the condition that the state is $s_{i'i} \in S_{i'}$. $P_{i'}$ denotes the probability vector of the operation behavior set $A_{i'}, p_{i'j_1} \in P_{i'} (j_1 = 1, \ldots., r)$ and denotes the probability value of the implementation of operation behavior α_{j_1}, satisfying: $0 < p_{j_1} < 1, \sum_{j1=1}^{r} p_{j_1} = 1$.

Level 2: $\mathrm{CNM}_{i'j_1} = \{S_{i'j_1}, A_{i'j_1}, \Gamma_{i'j_1}, P_{i'j_1}\}$: The output behavior of Level 1 is $\alpha_{i'j_1}$ as the internal state set of the $\mathrm{CNM}_{i'j_1}$ learning system in the Level 2 layer: $S_{i'j_1} = \{a_{i'j_1}\}$. $A_{i'j_1}$ represents the set of operation behavior of the Level2 level CNM learning system: $A_{i'j_1} = \{\alpha_{i'j_1j_2} | j_2 = 1{,}2, \cdots{,}r\}$. $\Gamma_{i'j_1}$ represents the conditional state of the Level2 level CNM learning system – the random mapping set of operation behavior: $\Gamma_{i'j_1}$: $\{a_{i'j_1} \to a_{i'j_1j_2}(P_{i'j_1}), j_2 = (1, \ldots{,}r)\}$. $P_{i'j_1}$ denotes the probability vector of operation behavior set $A_{i'j_1}$ and $p_{i'j_1j_2} \in P_{i'j_1}$ denotes the probability value of operation behavior set $\alpha_{i'j_1j_2}$, which satisfies: 0<$p_{i'j_1j_2}$<1,$\sum_{j_2=1}^{r} p_{i'j_1j_2} = 1$.

The Levels layer $\mathrm{CNM}_{i'j_1j_2\cdots j_{s-1}} = \{S_{i'j_1j_2\cdots j_{s-1}}, A_{i'j_1j_2\cdots j_{s-1}}, \Gamma_{i'j_1j_2\cdots j_{s-1}}, P_{i'j_1j_2\cdots j_{s-1}}\}$: The operation behavior of the Level(s-1) layer is $\alpha_{i'j_1j_2\cdots j_{s-2}}$ as the internal state set of the CNM learning system in the Levels layer: $S_{i'j_1j_2\cdots j_{s-1}} = \{a_{i'j_1j_2\cdots j_{s-1}}\}$. $A_{i'j_1j_2\cdots j_{s-1}}$ represents the set of operation behaviors of the Levels level CNM learning system: $A_{i'j_1j_2\cdots j_{s-1}} = \{\alpha_{j_1j_2\cdots j_{s-1}} | j_{s-1} = 1{,}2, \cdots{,}r\}$.

$\Gamma_{i'j_1j_2\cdots j_{s-1}}$ represents the conditional state of the Levels CNM learning system – the random mapping set of operation behavior $\Gamma_{i'j_1j_2\cdots j_{s-1}}$: $\{a_{i'j_1j_2\cdots j_{s-1}} \to a_{i'j_1j_2\cdots j_s}(P_{i'j_1j_2\cdots j_{s-1}}), j_{s-1} = (1, \ldots{,}r)\}$. $P_{i'j_1j_2\cdots j_{s-1}}$ denotes the probability vector of the operation behavior set $A_{i'j_1j_2\cdots j_{s-1}}$ and $p_{i'j_1j_2\cdots j_{s-1}} \in P_{i'j_1j_2\cdots j_{s-1}}$ denotes the probability value of implementing the operation behavior $\alpha_{i'j_1j_2\cdots j_{s-1}}$, which satisfies: 0<$p_{i'j_1j_2\cdots j_{s-1}}$<1, $\sum_{i'=1}^{3} \sum_{j_{s-1}=1}^{r} p_{i'j_1j_2\cdots j_{s-1}} = 1$.

The working process of the HCNM learning model can be briefly summarized as follows: $t=0$ time; the state condition signal s_i first activates the CNM learning system of the Level1 layer to determine the sub-tasks to be performed. Then, it moves into the behavior decomposition layer and chooses the refined behavior. The initial probability of behavior selection is the same; that is, $p_{i'j_1} = \frac{1}{r}\left(p_{i'j_1} \in P,\ j_1 = 1, \ldots, r\right); p_{i'j_1j_2} = \frac{1}{r}\left(p_{i'j_1j_2} \in P_{i'j_1},\ j_1 = 1, \ldots, r,\ j_2 = 1, \ldots, r\right)$. According to the probability vector $P_{i'} = (p_{i'1}, p_{i'2}, \ldots, p_{i'r})$, the learning system randomly selects an operation behavior (assumed to be $a_{i'j_1}$) from the behavior set $A_{i'}$ and transfers it to the next layer. Behavior $a_{i'j_1}$ acts as the internal state signal of the next layer of the CNM learning system, activates the corresponding $\mathrm{CNM}_{i'j_1}$ learning system, and then, the CNM learning system randomly selects a behavior (assumed to be $a_{i'j_1j_2}$) from the operation behavior set $A_{i'j_1}$, according to the probability vector $P_{i'j_1} = [p_{i'11}, p_{i'2}, \ldots, p_{i'r}]$, and continues to transport it to the next layer. Similar operations range from Level 2 to Level s, and ultimately output behavior $a_{i'j_1j_2\cdots j_{s-1}}$ as a control signal acting on the control system. After several rounds of trial learning, the optimal decision $a^{optim}_{i'j_1j_2\cdots j_s}$ of each sub-task is finally learned.

In order to illustrate the learning process of the HCNM learning model more clearly, the following definition is given:

Definition 1: Behavior path: From Level 1 to Level s, Sequence $a_{i'j_1}, a_{i'j_1j_2}, \ldots, a_{i'j_1j_2\cdots j_{s-1}}$, which is composed of behaviors selected by the CNM learning system, is defined as a behavior path, expressed in $\phi_{i'j_1j_2\cdots j_{s-1}}$.

Remark 1: Behavior path selection probability: t time defines the behavior path $\phi_{i'j_1j_2\cdots j_{s-1}}$. The probability of being selected is $q_{i'j_1j_2\cdots j_{s-1}}$, which satisfies:

$$q_{i'j_1j_2\cdots j_{s-1}}(t) = p_{i'j_1}(t)\, p_{i'j_1j_2}(t) \ldots p_{i'j_1j_2\cdots j_{s-1}}(t) \tag{4}$$

Definition 2: The orientation values of behavior paths are defined as follows:

$$\varphi_{i'j_1j_2\cdots j_{s-1}}(t) = \left| \frac{e^{\gamma\chi(t)} - e^{-\gamma\chi(t)}}{e^{\gamma\chi(t)} + e^{-\gamma\chi(t)}} \right| \tag{5}$$

Among $\chi(t) = \dot{e}(t) + \zeta e(t)$ and $e(t) = s(t) - s_d$, s_d is the expected state value, when orientation value $\varphi_{i'j_1j_2\cdots j_{s-1}}(t)$ is zero, it indicates that learning performance is good and the orientation degree in this state is high. When orientation value $\varphi_{i'j_1j_2\cdots j_{s-1}}(t)$ equals 1, it indicates that learning performance is poor and the orientation degree in this state is low.

A behavior path is a control strategy. After the behavior path $\phi_{i'j_1j_2\cdots j_{s-1}}$ acts on the control system, it will be fed back to an orientation value $\varphi_{i'j_1j_2\cdots j_{s-1}}(t)$ to measure the orientation degree of the learning system to the behavior path. The learning system adjusts the behavior probability vector $\{P_{i'j_1}, \ldots, P_{i'j_1j_2\cdots j_{s-1}}\}$ corresponding to the CNM learning system according to the orientation value. Repeat the above learning process until you find the optimal behavior path $\phi_{i'j_1^*j_2^*\cdots j_{s-1}^*}$. The optimal behavior path satisfies the following inequality:

$$\min E\left\{\varphi_{i'j_1^*j_2^*\cdots j_N^*}(t)\right\} > \max E\left\{\varphi_{i'j_1j_2\cdots j_{s-1}}(t)\right\} \tag{6}$$

Among them, $\varphi_{i'j_1j_2\cdots j_N}(t)$ represents other path (non-optimal) orientation values, and $\max\left(|j_1^* - j_1|, |j_2^* - j_2|, \ldots, |j_{s-1}^* - j_{s-1}|\right) > 0$.

3.2. Learning algorithm and convergence proof

3.2.1. Learning algorithm design

In both psychodynamics and biothermodynamics, a cognitive process can be regarded as a thermodynamic process that can be studied thermodynamically. Thermodynamic methods can be used to study it. Therefore, this paper combines a Monte Carlo-based simulated annealing algorithm to design

https://doi.org/10.1017/S0263574722001539 Published online by Cambridge University Press

a cognitive learning algorithm. A simulated annealing algorithm is a probabilistic algorithm with an approximate global optimum. According to the Metropolis criterion, the probability of reaching the equilibrium of energy in a particle at temperature T is $\exp{(\Delta E)}/K_B * T$. In the equation, E represents the internal energy of a particle at temperature T, ΔE is the variation of energy in a particle, and KB is the Boltzmann constant.

$$p\left(\alpha_{i' j_1 j_2 \cdots j_{s-1}}(t)\, | s_i(t)\right) = \exp\left[\frac{\varphi_{i' j_1 j_2 \cdots j_{s-1}}(t+1) - \varphi_{i' j_1 j_2 \cdots j_{s-1}}(t)}{K_B T}\right] \tag{7}$$

Furthermore, suppose that the state of t is $s_i(t)$, the operation behavior $\alpha_{i' j_1 j_2 \cdots j_{s-1}}$ is implemented and the state transition is $t+1$ time state $s_j(t+1)$. According to Skinner's OCR Theory, if the difference between the orientation values of state $s_j(t+1)$ and state $s_i(t)$ is $\varphi_{i' j_1 j_2 \cdots j_{s-1}}(t+1) - \varphi_{i' j_1 j_2 \cdots j_{s-1}}(t) > 0$, the probability $p(\alpha_{i' j_1 j_2 \cdots j_{s-1}}(t)|s_i(t))$ of implementing operational behavior $\alpha_{i' j_1 j_2 \cdots j_{s-1}}$ in state $s_i(t)$ tends to increase in later learning and, vice versa, $\varphi_{i' j_1 j_2 \cdots j_{s-1}}(t+1) - \varphi_{i' j_1 j_2 \cdots j_{s-1}}(t) < 0$, the probability $p(\alpha_{i' j_1 j_2 \cdots j_{s-1}}(t)|s_i(t))$ tends to decrease. Therefore, based on the idea of simulated annealing, the cognitive learning algorithm is designed as Eq. (8).

$$p\left(\alpha_{i' j_1 j_2 \cdots j_{s-1}}(t)\, | s_i(t)\right) = \exp\left[\frac{\varphi_{i' j_1 j_2 \cdots j_{s-1}}(t+1) - \varphi_{i' j_1 j_2 \cdots j_{s-1}}(t)}{K_B T}\right] \tag{8}$$

When the generated random number $\delta \in [0, 1]$ is less than $p(\alpha_{i' j_1 j_2 \cdots j_{s-1}}(t)|s_i(t))$, the random action $\alpha_{i' j_1 j_2 \cdots j_{s-1}}$ is chosen, whereas the action with the largest orientation value is selected according to the strategy. The isometric cooling strategy is adopted to cool the temperature $T = \lambda_T^n T_0$. The temperature decreases regularly, and the speed of change is slow. In the Equation, T_0 represents the initial temperature and n indicates the total number of iterations. λ_T is the value between [0,1]; the greater the value of λ_T, the slower the annealing rate.

The implementation of the simulated annealing strategy is as follows:

Step 1: Initialization of initial temperature T and iteration number n;

Step 2: Acquires the state i of the current solution and generates a new state j;

Step 3: The random number $\delta \in [0, 1]$ is generated, and the probability p of accepting the new solution j with the current solution i and temperature control parameter T is calculated according to Eq. (4). When $\delta < p$, accepting the new solution to the current problem is not acceptable.

Step 4: If the end condition is satisfied, the optimal solution is output; otherwise, Step 5 is executed.

Step 5: If each temperature T reaches n times, then according to the annealing strategy, the temperature T is cooled down to Step 1, and the temperature T after cooling is taken as the initial temperature of this learning; otherwise, the temperature T is changed to Step 2 to continue learning.

At the beginning of learning, the value of temperature T is larger and the probability of choosing a non-optimal solution is higher. With the increase in learning times and time itself, the value of temperature T becomes smaller and the probability of choosing the optimal solution increases. When $T \to 0$, the non-optimal solution will not be selected and the global optimal solution will be found.

3.2.2. Proof of convergence of the learning algorithm

Suppose that the optimal path vector corresponding to the optimal path $\phi_{i'^* j_1^* j_2^* \cdots j_{s-1}^*}$ is: $V_{i'^* j_1^* j_2^* \cdots j_{s-1}^*}(t) = \{v_{i'^* j_1^* j_2^* \cdots j_{s-1}^* 1}(t), v_{i'^* j_1^* j_2^* \cdots j_{s-1}^* 2}(t), \ldots, v_{i'^* j_1^* j_2^* \cdots j_{s-1}^* r}(t)\}$.

Lemma 1: *If the orientation value of each action in the orientation value vector is calculated by the reference Eq. (5), the following inequalities are satisfied for the optimal learning system* $\mathrm{CNM}_{\mathrm{i}'^* j_1^* j_2^* \cdots j_{s-1}^*}$ *:*

$$E\left\{v_{i'^* j_1^* j_2^* \cdots j_{s-1}^*}(t)\right\} > E\left\{v_{i'^* j_1^* j_2^* \cdots j_{s-2}^* i_s}(t)\right\} \tag{9}$$

Among them, $i'^* = 1, 2, 3; i_s, j_{s-1}^* = 1, 2, \ldots, r \quad i_s \neq j_{s-1}^*$.

https://doi.org/10.1017/S0263574722001539 Published online by Cambridge University Press

Proof: In order to establish proof (8), we first give the following definition:

Definition 3: Assuming that the behavior path $\phi_{{i'}_{j_1j_2\cdots j_{s-1}}}$ at time t is selected and the orientation value of the path $\varphi_{{i'}_{j_1j_2\cdots j_{s-1}}}$ is calculated from the feedback information of the system, the current orientation degree of path $\phi_{{i'}_{j_1j_2\cdots j_{s-1}}}$ is defined as: $u_{{i'}_{j_1j_2\cdots j_{s-1}}}(t)=\varphi_{{i'}_{j_1j_2\cdots j_{s-1}}}$, time t, and the current orientation value of other paths $\phi_{{i''}_{i_1i_2\ldots i_{s-1}}}(\exists k, i_k \neq j_k)$ is defined as:

$$u_{{i''}_{i_1i_2\ldots i_{s-1}}}(t)=\varphi_{{i''}_{i_1i_2\ldots i_{s-1}}}\left(\tau_{{i''}_{i_1i_2\ldots i_{s-1}}}\right) \tag{10}$$

Among them, $\tau_{{i''}_{i_1i_2\ldots i_{s-1}}}$ is the nearest time chosen by the path $\phi_{{i''}_{i_1i_2\ldots i_{s-1}}}$, and $\varphi_{{i''}_{i_1i_2\ldots i_{s-1}}}(\tau_{{i''}_{i_1i_2\ldots i_{s-1}}})$ indicates the orientation value of $\tau_{{i''}_{i_1i_2\ldots i_{s-1}}}$ time.

Remark 2: In layer s CNM$_{{i'}_{j_1j_2\cdots j_{s-1}}}$ of the HS-CNM learning model, the orientation values of all the behaviors constitute vectors: $V_{{i'}_{j_1j_2\cdots j_{s-1}}}(t)=\{v_{{i'}_{j_1j_2\cdots j_{s-1}1}}(t), v_{{i'}_{j_1j_2\cdots j_{s-1}2}}(t), \ldots, v_{{i'}_{j_1j_2\cdots j_{s-1}r}}(t)\}$. Each part of $V_{j_1j_2\cdots j_{s-1}}(t)$ is constructed as follows:

Layer s:

$$v_{{i'}_{j_1j_2\cdots j_{s-1}}}(t)=u_{{i'}_{j_1j_2\cdots j_{s-1}}}(t) \tag{11}$$

Layer $s-1$:

$$v_{{i'}_{j_1j_2\cdots j_{s-1}}}(t)=\max\left\{v_{{i'}_{j_1j_2\cdots j_{s-1}1}}(t), v_{{i'}_{j_1j_2\cdots j_{s-1}2}}(t), \ldots, v_{{i'}_{j_1j_2\cdots j_{s-1}r}}(t)\right\} \tag{12}$$

According to Eqs. (10) and (11), for the optimal learning system CNM$_{{i'^*}_{j_1^*j_2^*\cdots j_{s-1}^*}}$ at level s, the orientation value of the internal operation behavior satisfies:

$$E\{v_{{i'^*}_{j_1^*j_2^*\cdots j_{s-1}^*i_s}}(t)\}=E\{u_{{i'^*}_{j_1^*j_2^*\cdots j_{s-1}^*i_s}}(t)\}=E\{\varphi_{{i'^*}_{j_1^*j_2^*\cdots j_{s-1}^*i_s}}(\tau_{{i'^*}_{j_1^*j_2^*\cdots j_{s-1}^*i_s}})\}iN=1,2,\ldots .r \tag{13}$$

Eq. (13) can be obtained:

$$E\left\{v_{{i'^*}_{j_1^*j_2^*\cdots j_{s-1}^*}}(t)\right\}=E\left\{\varphi_{{i'^*}_{j_1^*j_2^*\cdots j_{s-1}^*}}\left(\tau_{{i'^*}_{j_1^*j_2^*\cdots j_{s-1}^*}}\right)\right\}>E\left\{\varphi_{{i'^*}_{j_1^*j_2^*\cdots j_{s-1}^*i_s}}\left(\tau_{{i'^*}_{j_1^*j_2^*\cdots j_{s-1}^*i_s}}\right)\right\}=E\left\{v_{{i'^*}_{j_1^*j_2^*\cdots j_{s-1}^*i_s}}(t)\right\} \tag{14}$$

Among them, $i_{\mathrm{s}}=1{,}2{,}\ldots{,}r$; $i_s \neq j_{s-1}^*$.

It can be seen from the cognitive model of the s level that the orientation value of the operation behavior is satisfied Eq. (8).

Next, we analyze the orientation value of the CNM$_{{i'^*}_{j_1^*j_2^*\cdots j_{s-2}^*}}$ operation behavior in the s-level optimal learning system, which can be obtained by Eqs. (8) and (14):

$$\begin{aligned}E\left\{v_{{i'^*}_{j_1^*j_2^*\cdots j_{s-2}^*j_{s-1}^*}}(t)\right\}&=\max_{j_{s-1}}E\left\{v_{{i'^*}_{j_1^*j_2^*\cdots j_{s-2}^*j_{s-1}^*}}(t)\right\}=E\left\{v_{{i'^*}_{j_1^*j_2^*\cdots j_{s-2}^*j_{s-1}^*}}(t)\right\}\\&=E\left\{\varphi_{{i'^*}_{j_1^*j_2^*\cdots j_{s-2}^*j_{s-1}^*}}\left(\tau_{{i'^*}_{j_1^*j_2^*\cdots j_{s-2}^*j_{s-1}^*}}\right)\right\}\end{aligned} \tag{15}$$

$$\begin{aligned}E\left\{v_{{i'^*}_{j_1^*j_2^*\cdots j_{s-2}^*j_{s-2}^*i_{s-1}}}(t)\right\}&=\max_{l_s}E\left\{v_{{i'^*}_{j_1^*j_2^*\cdots j_{s-2}^*j_{s-2}^*i_{s-1}l_s}}(t)\right\}=E\left\{v_{{i'^*}_{j_1^*j_2^*\cdots j_{s-2}^*j_{s-2}^*i_{s-1}l_s}}(t)\right\}\\&=E\left\{\varphi_{{i'^*}_{j_1^*j_2^*\cdots j_{s-2}^*j_{s-2}^*i_{s-1}l_s}}\left(\tau_{{i'^*}_{j_1^*j_2^*\cdots j_{s-2}^*j_{s-2}^*i_{s-1}l_s}}\right)\right\}\end{aligned} \tag{16}$$

This can be obtained with Eqs. (15) and (16):

$$\begin{aligned}E\left\{v_{{i'^*}_{j_1^*j_2^*\cdots j_{s-2}^*j_{s-1}^*}}(t)\right\}&=E\left\{\varphi_{{i'^*}_{j_1^*j_2^*\cdots j_{s-2}^*j_{s-1}^*}}\left(\tau_{{i'^*}_{j_1^*j_2^*\cdots j_{s-2}^*j_{s-1}^*}}\right)\right\}\\&>E\left\{v_{{i'^*}_{j_1^*j_2^*\cdots j_{s-2}^*j_{s-2}^*i_{s-1}}}(t)\right\}=E\left\{\varphi_{{i'^*}_{j_1^*j_2^*\cdots j_{s-2}^*j_{s-2}^*i_{s-1}l_s}}\left(\tau_{{i'^*}_{j_1^*j_2^*\cdots j_{s-2}^*j_{s-2}^*i_{s-1}l_s}}\right)\right\}\end{aligned} \tag{17}$$

where, $l_{\mathrm{s}}, j_{\mathrm{s}}$-1 $=1{,}2{,}\ldots{,}r$; i_{s} $1=1{,}2{,}\ldots{,}r$; $i_{s-1} \neq j_{s-2}^*$.

https://doi.org/10.1017/S0263574722001539 Published online by Cambridge University Press

Following the same steps, we can obtain the following conclusions:

$$E\left\{v_{i'^*_{j_1^* j_2^* \cdots j_{s-1}^* j_{s-1}^*}}(t)\right\} > E\left\{v_{i'^*_{j_1^* j_2^* \cdots j_{s-1}^* i_s}}(t)\right\} \tag{18}$$

where, i_s-1 = 1,2,. . .,r; $i_s \neq j_{s-1}^*$, and lemma is proved.

Theorem 1: *When the probability of the operation satisfies* $0 < p_{\mathrm{i}'_{j_1 j_2 \cdots j_{s-1}}} < 1$, *the optimal path* $\phi_{\mathrm{i}'^*_{j_1^* j_2^* \cdots j_{s-1}^*}}$ *is selected according to probability* $q_{\mathrm{i}'_{j_1 j_2 \cdots j_{s-1}}}(t) \approx 1$.

It has been proven that the probability of the operation behavior is satisfied: $p_{\mathbf{min}} \leq p_{i'_{j_1 j_2 \cdots j_{s-1}}} \leq p_{\mathbf{max}}$, where pmax is close to 1 and pmax is close to 0. From lemma 1, it can be seen that the orientation value corresponding to the optimal operation behavior is also the largest, so the occurrence probability of the operation behavior corresponding to the optimal action path is satisfied: $\lim_{t\to\infty} p_{i'^*_{j_1^* j_2^* \cdots j_{s-1}^*}} = p_{\mathbf{max}}$. Then, the probability corresponding to the optimal action path is satisfied:

$$\begin{aligned}
&\lim_{t\to\infty} q_{i'^*_{j_1^* j_2^* \cdots j_{s-1}^*}}(t) = \lim_{t\to\infty}\left\{p_{i'^*}(t), p_{i'^*_{j_1^*}}(t), \ldots, p_{i'^*_{j_1^* j_2^* \cdots j_{s-1}^*}}(t)\right\}\\
&= \lim_{t\to\infty} p_{i'^*}(t) \lim_{t\to\infty} p_{i'^*_{j_1^*}}(t) \ \ldots \ \lim_{t\to\infty} p_{i'^*_{j_1^* j_2^* \cdots j_{s-1}^*}}(t)\\
&= p_{\mathbf{max}}\, p_{\mathbf{max}} \ \ldots \ p_{\mathbf{max}} = (p_{\mathbf{max}})^s
\end{aligned} \tag{19}$$

This can be obtained with type Eqs. (15) and (16), that Eq. (18) is $\lim_{t\to\infty} q_{i'^*_{j_1^* j_2^* \cdots j_{s-1}^*}}(t) \approx 1$, the theorem is proved.

4. Realization of robot autonomous navigation

4.1. Robot State and Behavior Classification

4.1.1. Sub-task status division

The purpose of the robot path planning is to make the robot reach the target point safely and without collisions, having departed from the starting point. It is necessary to consider the distance information of the obstacles, as well as the movement state and the position and distance information of the target point. These changing pieces of environmental information constitute a large state space, which seriously affects the learning efficiency of the robot. According to the hierarchical and abstract strategy of the hierarchical cognitive model, the path planning task of a mobile robot is divided into three basic sub-tasks $S = \{S_1, S_2, S_3\}$: static obstacle avoidance, dynamic obstacle avoidance, and moving toward the target point. The state space is divided into three small-scale spaces in order to improve the learning efficiency of the robot.

Assuming that the robot can turn freely in a narrow environment without touching any obstacles, the radius of rotation of the robot is not considered in the navigation algorithm, and the robot is simplified to a particle. The relationship between robots, obstacles, and target points is shown in Fig. 2.

The environmental information around the mobile robot is mainly detected by the camera. In order to simplify the problem, the detection range of the sensor is divided into three areas: left, front, and right. Thus, the environmental state between the robot and the obstacle can be expressed through distance information in three directions. In addition, we do not pay attention to the specific location information and determine the distance of the robot itself and all the obstacles, but rather only to the approximate distance range and relative position direction of the nearest obstacle. Based on this, we define the state space of the three sub-tasks.

Definition 4: The static obstacle avoidance task mainly considers the robot moving towards the target point while avoiding obstacles. Therefore, the minimum distance measurements of static obstacles in three areas detected by sensors, the angle between the moving direction of the mobile robot and the direction of target point, and the distance between the mobile robot and the target point are taken as the input state information of the static obstacle avoidance sub-task module:

$$S_1 = \left\{d_{sr\_l}, d_{sr\_f}, d_{sr\_r}, d_{r\_tar}, \theta\right\} \tag{20}$$

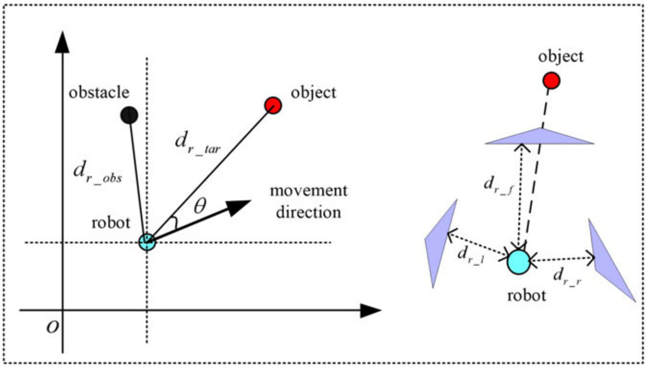


Figure 2. *The relationship between robots, obstacles, and target points.*

In the Eq. (20), d_{sr_l} is the distance from the left side of the robot to the static obstacle; d_{sr_f} is the distance between the front and the static obstacles of the robot; d_{sr_r} is the distance from the right side of the robot to the static obstacle; d_{r_tar} is the distance between robots and target points; and θ is the angle between the moving direction and the target point of the robot.

The distance from static obstacles in three directions is discretized into N (Near) and F (Far), as shown in Eq. (21).

$$\mathrm{d_{sr\_l}, d_{sr\_f}, d_{sr\_r}\}} = \begin{cases} \mathrm{N, d_s \leq min\,(d_{sr\_l}, d_{sr\_f}, d_{sr\_r}) < d_m} \\ \mathrm{F, min\,(d_{sr\_l}, d_{sr\_f}, d_{sr\_r}) > d_m} \end{cases} \tag{21}$$

$\mathrm{d_s}$ represents the minimum dangerous distance, when any detection distance is less than $\mathrm{d_s}$ represents the mobile robot obstacle avoidance failure. $\mathrm{d_m}$ represents the maximum safe distance, and the robot can walk safely at the maximum speed when the detection distance in all directions is greater than $\mathrm{d_m}$.

The distance between the robot and the target point is also discretized into N and F, and the angle between the direction of motion and the target point is discretized into zero and non-zero, {zero, not zero}.

Definition 5: The dynamic obstacle avoidance task mainly considers avoiding dynamic obstacles while avoiding collisions with static obstacles. Therefore, the minimum distance measurements of obstacles in three areas detected by sensors, the moving direction of dynamic obstacles, and the location information of dynamic obstacles are taken as the input state information of the dynamic obstacle avoidance sub-task module:

$$S_2 = \left\{d_{sr\_l}, d_{sr\_f}, d_{sr\_r}, d_{dr\_l}, d_{dr\_f}, d_{dr\_r}, \theta_d\right\} \tag{22}$$

In the Eq. (22), d_{dr_l} is the distance from the left side of the robot to the dynamic obstacle; $d_{dr_f}$is the distance from the front of the robot to the dynamic obstacles; d_{dr_r} is the distance from the right side of the robot to the dynamic obstacles; and θ_d is the angle between the direction of motion of a robot and the direction of motion of a moving obstacle.

The distance from the dynamic obstacle in three directions is also discretized into n (Near) and F (Far) whose form is the same as Eq. (21).

A virtual rectangular coordinate system is established, with the robot as the origin, and the direction of the robot's motion and the dynamic obstacle are used as the x-axis, as shown in Fig. 3.

The angle between the robot motion direction and the dynamic obstacle movement direction is discretized as Eq. (23).

$$\theta_{\mathrm{d}} = \begin{cases} \text{danger}, & \theta_{\mathrm{ds}} \leq |\theta_{\mathrm{d}}| \leq \pi \\ \text{safe}, & \text{else} \end{cases} \tag{23}$$

In Eq. (23), θ_{ds} represents the minimum risk angle.

https://doi.org/10.1017/S0263574722001539 Published online by Cambridge University Press

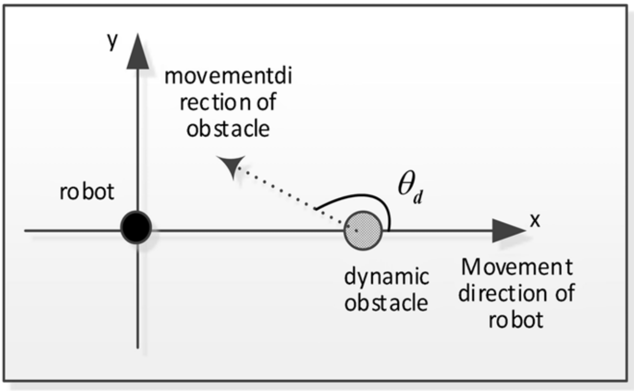


Figure 3. Virtual rectangular coordinate system.

Definition 6: A target-oriented task mainly considers the way in which a robot can move radially towards a target point along an optimal path. Therefore, the distance between the mobile robot and the target point, and the angle between the direction of the mobile robot and the direction of the target point are taken as the input state information of the sub-task module of the target-oriented task:

$$S_3 = \{d_{r\_tar}, \theta\} \tag{24}$$

The form of discretization is the same as in Definition 5.

4.1.2. Sub-task status division

The control variables of the robot are linear speed v and robot motion angle $\Delta\theta$. In the robot behavior control structure, the goal of obstacle avoidance and orientation can be achieved by determining the appropriate v and $\Delta\theta$. Therefore, v_i and $\Delta\theta$ are the behavior a of the robot.

$$a = \{v, \Delta\theta\} \tag{25}$$

When performing path planning tasks, the robot operation as follows:

If the robot is far from the static and dynamic obstacles, then the robot can directly navigate to the target at its maximum speed and perform the sub-task of moving toward the target.

If there are static obstacles near the robot, the robot tries to move left or right along the nearest obstacle in the direction of the target; that is, moving along the left or right static obstacles to perform the static obstacle avoidance task.

If there are dynamic obstacles near the robot, the robot tries to move left or right along the nearest dynamic obstacle, that is moving along the left or right dynamic obstacle to perform the dynamic obstacle avoidance task.

Considering that the behavior of the sub-task selection layer is roughly divided, all sub-tasks adopt the same behavior set $A_1 = A_2 = A_3 = \{a_1, a_2, a_3\}$.

Remark 3: The behavior set A of the robot in the sub-task selection layer is:

$$A_1 = A_2 = A_3 = \{a_1, a_2, a_3\} \tag{26}$$

a_1: The robot rotates +20° and advances 100 mm. a_2: The robot rotates −20° and advances 100 mm. a_3: The robot advances 200 mm.

+: Clockwise rotation. -:Counterclockwise rotation.

The behavior of the coarse selection of the sub-task selection layer is refined by one level.

Remark 4: The behavior refinement layer set is shown in Table I.

https://doi.org/10.1017/S0263574722001539 Published online by Cambridge University Press

Table I. *Layer thinning behavior set.*

Level **1 layer**	***Level*** **2 layer**	**Explanation**
Static obstacle avoidance	A_{11}	a_{111}: rotate+22°, advance50 mm;
A_1	$\{a_{111}, a_{112}, a_{113}\}$	a_{112}: rotate+15°, advance100 mm;
$\{a_1, a_2, a_3\}$		a_{113}: rotate+10° advance 100 mm
	A_{12}	a_{121}: rotate−22°, advance50 mm;
	$\{a_{121}, a_{122}, a_{123}\}$	a_{122}: rotate−15°, advance100 mm;
		a_{123}: rotate−10°, advance100 mm
Dynamic obstacle avoidance	A_{21}	A_{211}: rotate+25°, advance50 mm;
A_2	$\{a_{211}, a_{212}, a_{213}\}$	a_{212}: rotate+15°, advance50 mm;
$\{a_1, a_2, a_3\}$		A_{213}: rotate+10°, advance50 mm
	A_{22}	A_{221}: rotate−25°, advance50 mm;
	$\{a_{221}, a_{222}, a_{223}\}$	a_{222}: rotate−15°, advance50 mm;
		A_{223}: rotate−10° advance50 mm
	A_{23}	A_{231}: stand still;
	$\{a_{231}, a_{232}, a_{2233}\}$	a_{232}: rotate+5°, advance50 mm;
		A_{233}: rotate−5°, advance50mm
Tending to target	A_{31}	A_{311}: rotate+10°, advance100 mm;
A_3	$\{a_{111}, a_{112}, a_{113}\}$	a_{312}: rotate+5°, advance100 mm;
$\{a_1, a_2, a_3\}$		A_{313}: advance100 mm
	A_{32}	A_{321}: rotate−10°, advance100 mm;
	$\{a_{121}, a_{122}, a_{123}\}$	a_{322}: rotate−5°, advance100 mm;
		A_{323}: advance100 mm

4.2. Three-dimensional simulation experiment

4.2.1. Test environment and parameter settings

The computer operating environment settings for all the simulation experiment are as follows: the computer processor is Intel(R) Core(TM) i7-4790, the computer frequency of 3.6 GHz, and the computer RAM is 8 GB. The simulation software adopts MobotSim in this paper. A three-dimensional simulation experiment is carried out based on the software platform. The robot is approximately circular, with a diameter of 0.5 m. The robot is marked in red, and the target point is a smaller yellow circle, as shown in Fig. 4. There are obstacles in the environment: five in total from the starting point to the destination. The shapes and sizes of the obstacles are different. Three obstacles surround the robot in three directions and the other two obstacles surround the destination in two directions. The starting position of the robot is in the lower left corner of the environment, and the target is located in the upper right corner of the environment. The time interval of the time step is 0.1 s, and the speed of the robot center is 0.2 m/s. The mobile robot system can detect the distance of obstacles in different directions through detection devices and can sense the current location and target location information. It can then avoid obstacles through autonomous learning, and find the best or a better path through which to reach the target point.

The initial conditions of the experiment are as follows: In the probability renewal Eq. (4), the initial temperature $K_B T = 10{,}000$, the cooling coefficient $\lambda_T = 0.9$; the minimum dangerous distance $d_s = 0.1$ m, the maximum safe distance $d_m = 0.9$m; the maximum speed $V_{\max} = 0.15$ m/s; and the initial selection probability of each action in the behavior set is the same.

4.2.2. Experimental results and analysis

First, in the static obstacle environment, the feasibility of autonomous learning ability of cognitive model based on operational conditioned reflex and hierarchical structure is tested. Figure 5 shows the simulation results of path planning in a static environment. Figure 5 shows the simulation result of path planning

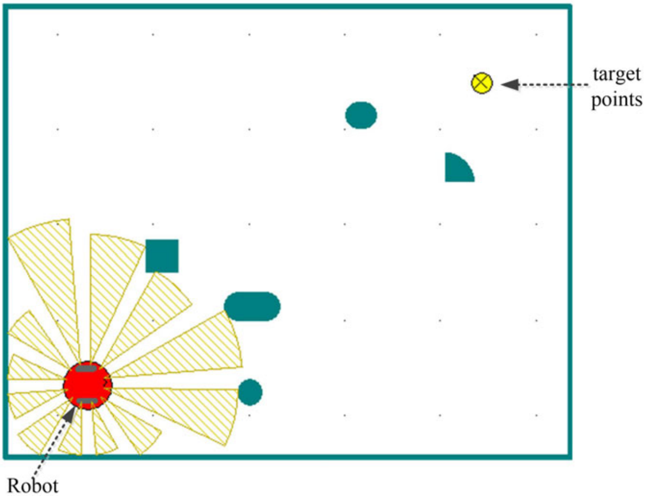


Figure 4. *2D simulation environment.*

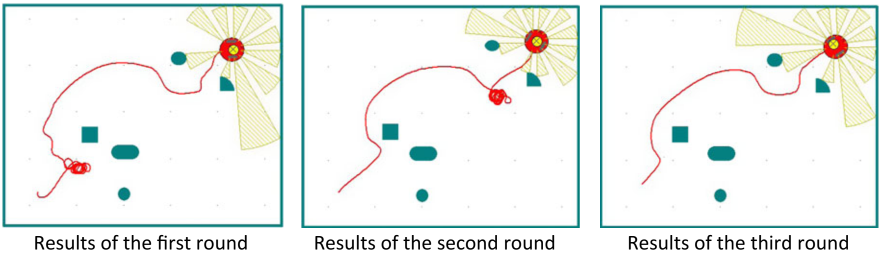

Results of the first round
Results of the second round
Results of the third round

Figure 5. *Results of the three-dimensional simulation experiments.*

in a static environment. From the results of the first round of experiments, it can be seen that the robot has not yet learned and has not accumulated enough knowledge about the environment. Therefore, the path appears to be more chaotic. Several failed attempts were made, in which the robot collided with obstacles but, after a period of exploration and further attempts, the robot can solve the predicament and continue to move toward the destination. The results of the second round of experiments show that the number of collisions is reduced, and the time it takes to solve the problem is shortened after the collisions occur. This shows that, after cognitive learning, the robot has a certain level of environmental experience. From the results of the third round of experiments, we can see that the robot has learned a relatively optimal path without collision. This shows that, after three rounds of experiments, the robot has grasped the general situation of the starting point, end point, and distribution of obstacles and can make a relatively stable judgment. The results of the follow-up experiments also show that the robot's judgment of the route has been amended, indicating that the model converges and the learning is over. Experiment results shown in Fig. 5 depict the way in which the trained mobile robot can reach the target point from the starting point without collisions and find an almost optimal path.

From Definitions 1 and 2, we can know the behavior path and behavior path selection probability. The histogram shown in Fig. 6 depicts the evolution process of each behavior probability. From the simulation results shown in Fig. 6, it can be seen that, at the end of the first round of learning, the probability values of each behavior are no longer equal, but a gap is not obvious. When the second round of learning is carried out, the probability of one behavior occurring continues to increase, while the probability of other behaviors occurring gradually decreases. After three rounds of learning, the probability of one behavior occurring is close to 1, while the probability of other behaviors is close to

https://doi.org/10.1017/S0263574722001539 Published online by Cambridge University Press

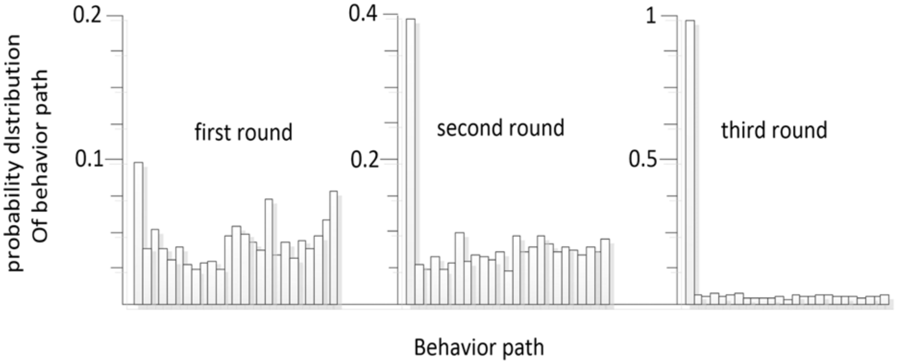


Figure 6. *Probability evolution process of the static obstacle avoidance behavior path.*

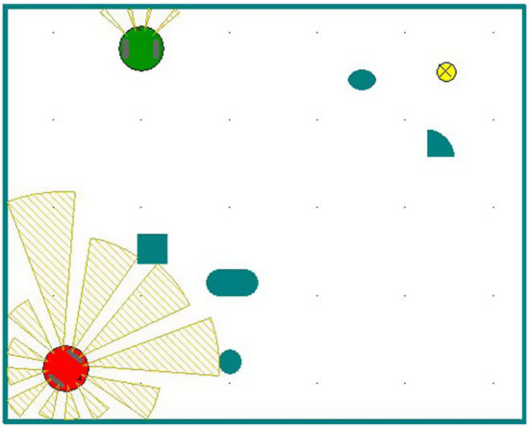


Figure 7. *Environment after adding dynamic obstacles.*

0. This shows that robots have learned to choose the most favorable behavior for themselves by learning and starting to choose behavior autonomously.

Changing the environmental information, adding a dynamic obstacle, and replacing it with a robot moving at a uniform speed v, as shown in Fig. 7, means that the best behavior learned before is no longer adapted to the new environment. Due to the complexity of environmental information, the robot needs to explore more. Therefore, the cooling coefficient is greatly increased $\lambda_T = 0.95$, slowing down the cooling rate and increasing the randomness of the robot action selection at the initial stage of training. Because the randomness of action selection is increased, the success rate is low in the initial stages. After four rounds of learning, the optimal path is acquired. Figure 8 is the result of the fourth round of learning. As can be seen in the graph, when the robot is about to reach point A, the sub-task of avoiding dynamic obstacles is selected to avoid collision with dynamic obstacles at point A. After reaching point B, the dynamic obstacle avoidance sub-task is completed. The static obstacle avoidance sub-task is selected again according to the environmental state. The robot is in a safe environment at point C. Therefore, the static obstacle avoidance sub-task is ended and the sub-task tending to the target point is selected. The histogram shown in Fig. 9 shows the evolutionary process of each behavior probability. After the fourth round of learning, the robot has learned the new optimal behavior. Compared to the experimental results shown in Figs. 5 and 6, the hierarchical model designed has a certain generalization ability for the autonomous path planning of robots. Comparing Figs. 6 and 9, it can be seen that without dynamic obstacles, the robot converges after three rounds of learning, and after adding dynamic obstacles, the robot converges after four rounds of learning. This shows even if the environmental information

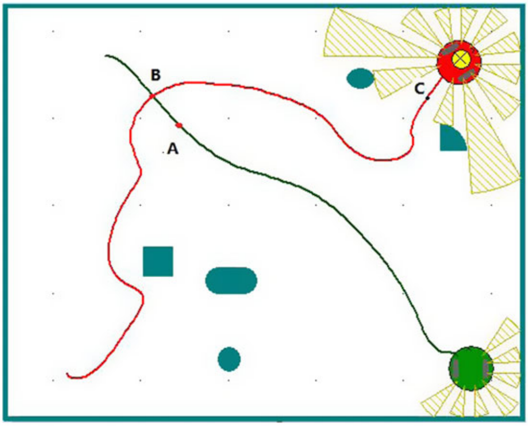


Figure 8. *Dynamic obstacle avoidance path.*

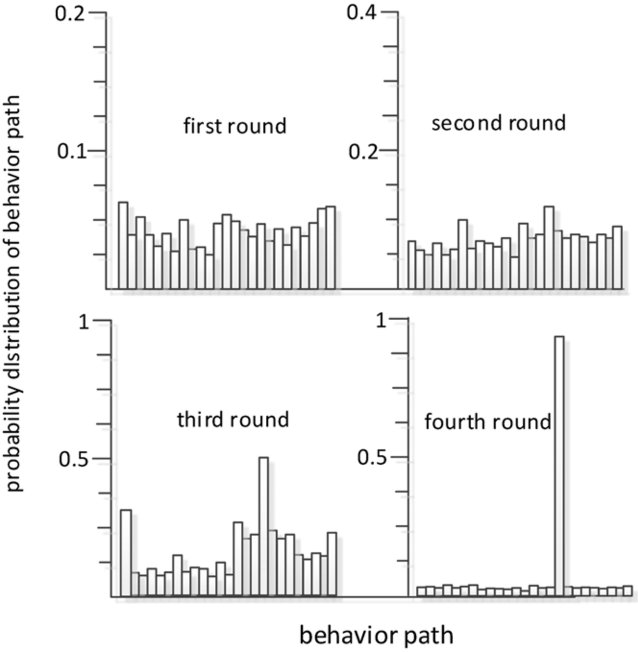


Figure 9. *Probabilistic evolution process of the dynamic obstacle avoidance behavior path.*

changes, the learning convergence speed does not decrease significantly and the robots can adapt to the environment quickly and rediscover the rules, which show the learning ability and adaptability similar to animals and can solve the dimensionality disaster problem in a complex environment.

In order to further test the autonomous navigation performance of the robot, the experiment that the robot faces both dynamic and static obstacles when it meets with dynamic obstacles is added. As shown in Fig. 10, the experimental results show that the robot encounters dynamic obstacles at point A and static obstacles at point C at point B, and the robot can avoid dynamic and static obstacles at the same time and successfully reach the target point.

In order to prove the rapidity of the hierarchical cognitive model, the results are compared with the single cognitive model. We can see that the probability of successful navigation is 80% by using single structure based on twenty times experiments; however, the hierarchical structure is about 94%.

https://doi.org/10.1017/S0263574722001539 Published online by Cambridge University Press

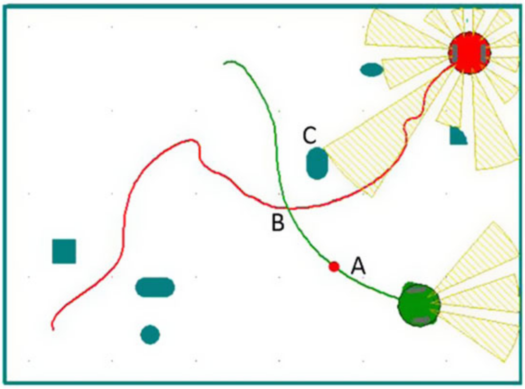


Figure 10. *Autonomous navigation results facing dynamic and static obstacles.*

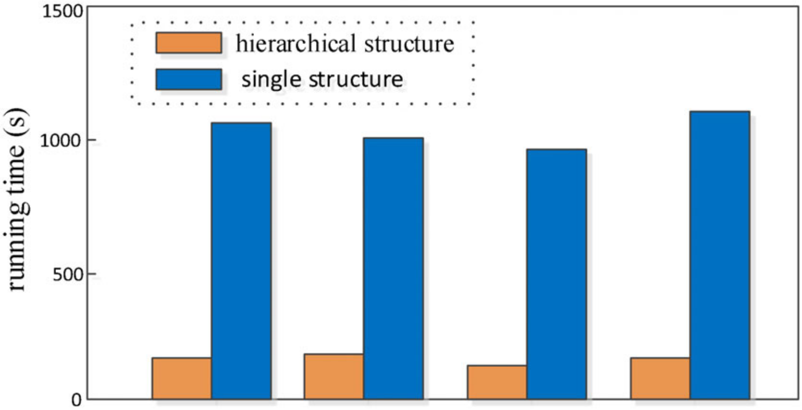


Figure 11. *The comparison of running time on hierarchical structure and single structure.*

The results of four times successful navigation were extracted from the experimental results, and their running times were compared as shown in Fig. 11.

According to the above experiments, it can be concluded that although the cognitive model of single structure can successfully realize robot autonomous navigation, its convergence speed is much slower than that of hierarchical structure. From the practical point of view, the hierarchical structure is better than the single structure.

In order to further prove the validity of the design model HCNM, we compared it with the similar hierarchical reinforcement learning model (HRLM). The similarity between the two models is that they are hierarchical, reward-based, "trial-and-error" learning. The difference is that the learning information of Q-learning which is the main algorithm reinforcement learning is stored in the Q table, which needs to update continuously. However, the learning algorithm of this paper can obtain autonomously the environmental information. The experimental results are shown in Figs. 12 and 13.

When the robot can navigate to the target point autonomously, it represents convergence. The time from the beginning of robot navigation to the end of navigation is the convergence time. Figures 12 and 13 show that HCNM is better than HRLM both in experimental effects and convergence time. The final path of HCNM model is clearer, and the effect is better. From the point of view of convergence speed, the

https://doi.org/10.1017/S0263574722001539 Published online by Cambridge University Press

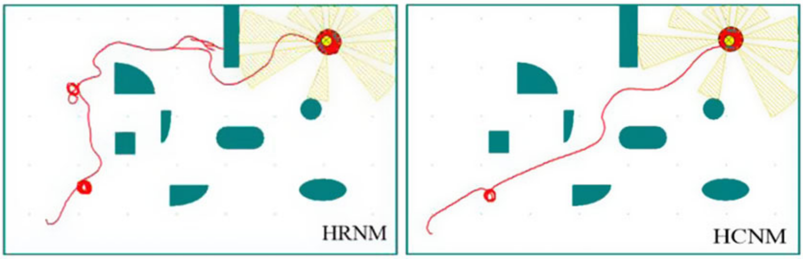


Figure 12. *The navigation of HCNM and HRLM model.*

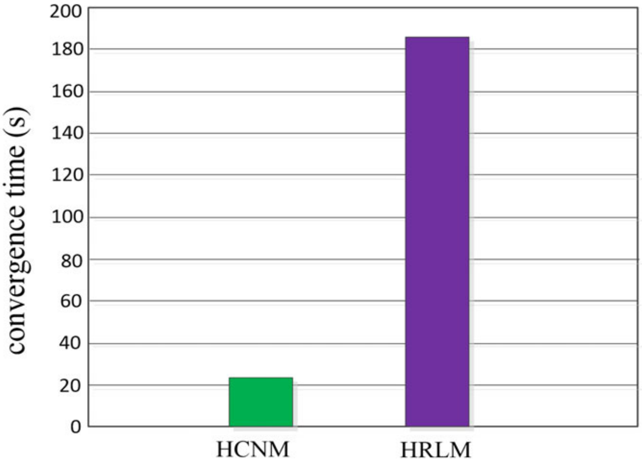
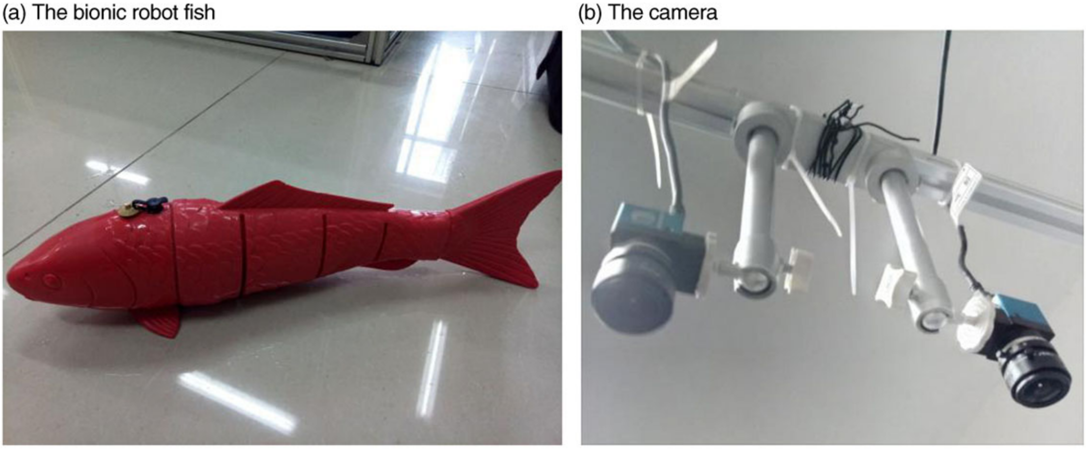


Figure 13. *The convergence time of two models.*


(a) The bionic robot fish

(b) The camera

Figure 14. *Experimental system of a bionic robot fish.*

convergence time of the HCNM is significantly reduced, which indicates that the convergence speed of HCNM is faster than HRLM. In addition, in this experiment, there are obstacles on the three sides of the target point, which increase the difficulty of navigation. The results show that the robot can successfully reach the target point through autonomous learning.

https://doi.org/10.1017/S0263574722001539 Published online by Cambridge University Press

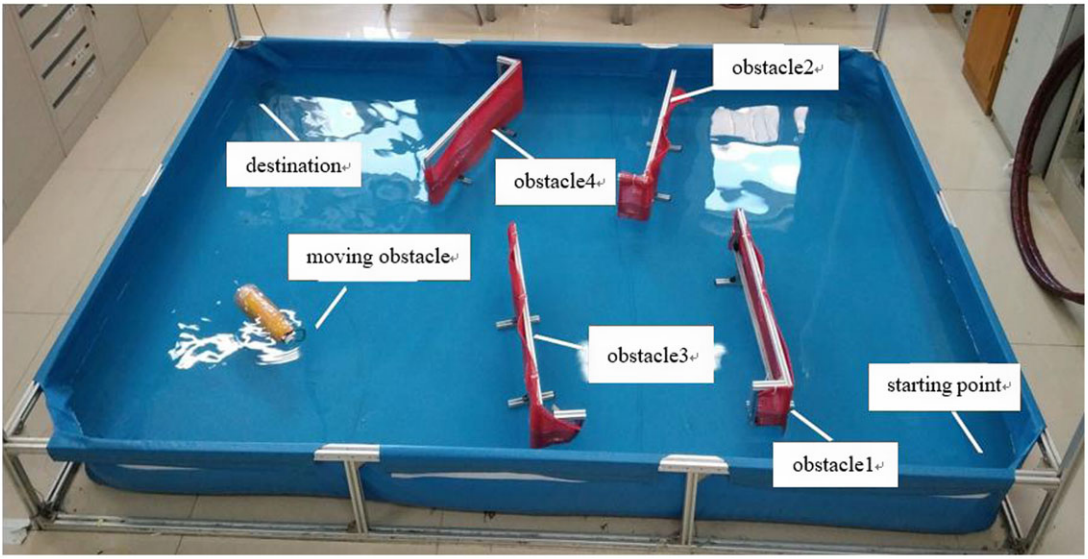


Figure 15. *Top view of the experimental environment.*

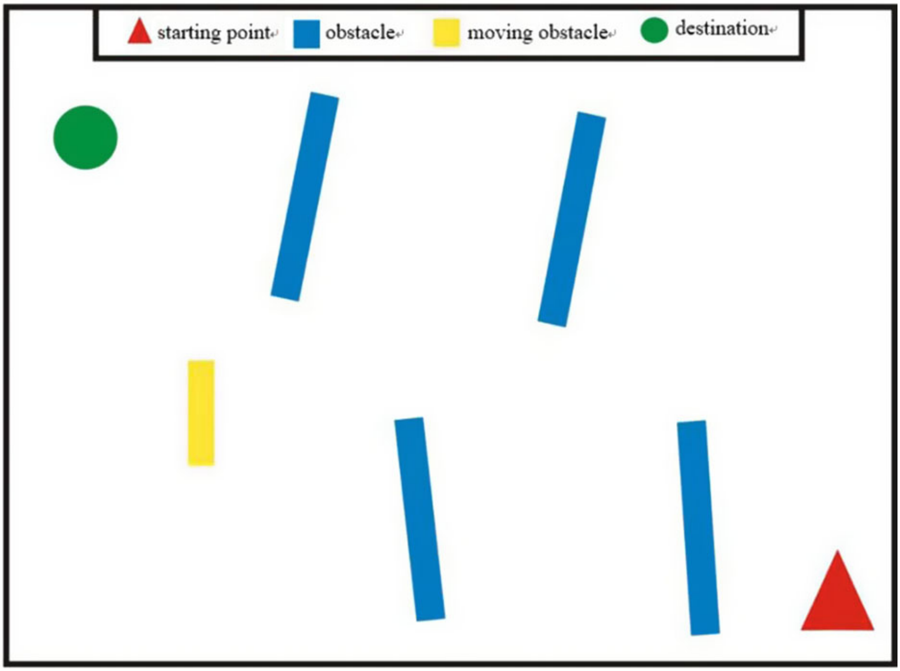


Figure 16. *Schematic diagram of the experimental environment simulation.*

4.3. Physical experiment

A simple biomimetic robotic fish (also known as a mechanical fish, artificial fish, or fish-like robot) is used in the physical experiments. As shown in Fig. 14(a), the robot is a lightweight mobile robot, which is suitable for navigation experiments in underwater environments. Robots perceive the environment through external cameras and the camera coverage must cover the entire site. Therefore, two cameras located in the center of the navigation environment are set up. The cameras are Mercury MER-040-60UC

https://doi.org/10.1017/S0263574722001539 Published online by Cambridge University Press

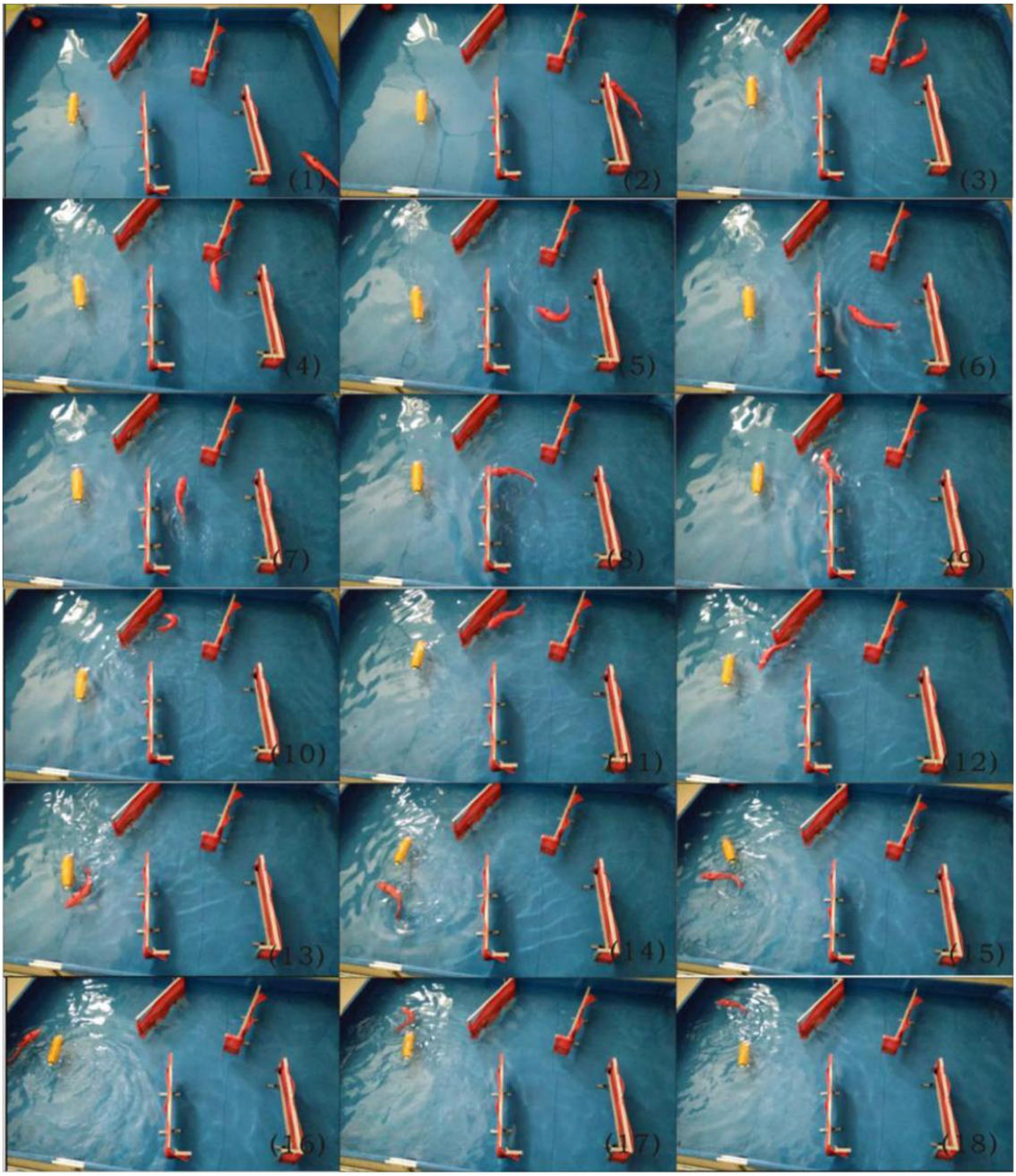


Figure 17. *Obstacle avoidance and navigation process for robotic fish.*

models, as shown in Fig. 14(b). Visual positioning can be accomplished by calculating the coordinate system established on the site. The position of the bionic fish is tracked in real time.

The kinematic model [30] of the robotic fish is approximated by a polynomial and sinusoidal synthesis, as shown in formula (27).

$$\boldsymbol{f}_{\boldsymbol{B}}\left(\boldsymbol{x},\boldsymbol{t}\right)=\left(\boldsymbol{c}_1\boldsymbol{x}+\boldsymbol{c}_2\boldsymbol{x}^2\right)\sin\left(\omega\boldsymbol{t}+\boldsymbol{k}\boldsymbol{x}\right) \tag{27}$$

where $\boldsymbol{f}_{\boldsymbol{B}}(\boldsymbol{x},\boldsymbol{t})$ is the lateral displacement of fish body, $\boldsymbol{c}_1$ is the primary coefficient of fish wave amplitude envelope. $\boldsymbol{c}_2$ is the quadratic coefficient of fish wave amplitude envelope, $\boldsymbol{k}$ is the wave number of fish

https://doi.org/10.1017/S0263574722001539 Published online by Cambridge University Press

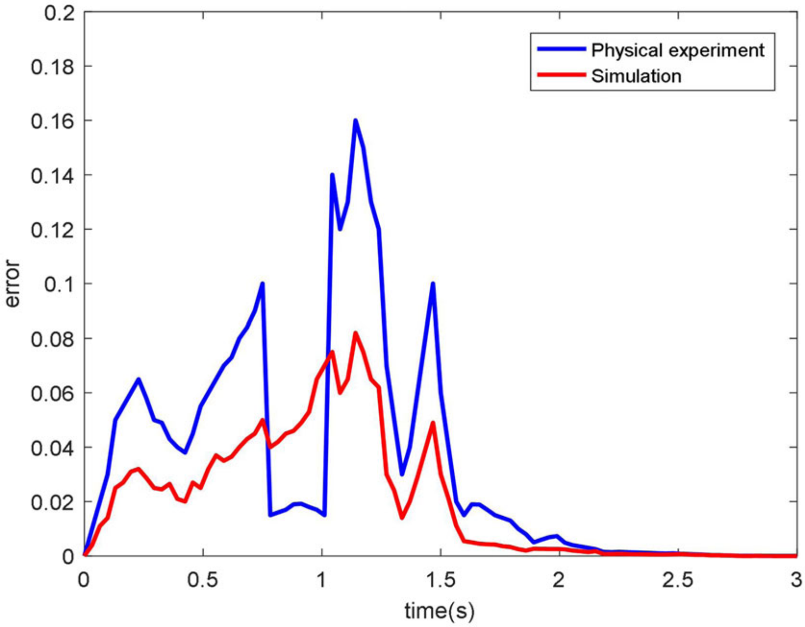


Figure 18. *Comparison of error between simulation and physical experiment.*

wave ($\boldsymbol{k} = 2\pi/\lambda$, λ is the wavelength of fish body wave), ω is the wave frequency of fish body ($\omega = 2\pi \boldsymbol{f} = 2\pi/\boldsymbol{T}$).

The robot experiment takes place in a 3 m × 2 m × 0.35 m cuboid fish pond environment. Considering the size of the fish and the experimental environment, four static obstacles and one dynamic obstacle are set up, which make the environment more complex. The bionic fish starts cruising from the lower right corner of the space and the destination is set at the upper left side of the initial position. The top view of the experimental environment is shown in Fig. 15. The schematic diagram of the experimental environment is shown in Fig. 16. After the bionic fish starts to move, it traverses obstacles from the starting point of the environment to its destination. If the robot fish touches a wall, touches obstacles, or arrives at its destination, the experiment ends.

A video recording of the obstacle avoidance navigation process of the robot is made. Figure 17 shows some screenshots of a successful experiment.

When the robot deviates from the target point, there is navigation error. The degree and number of robots deviating from the target point are the magnitude of navigation error. We compare the navigation errors between the simulation experiment and the physical experiment as shown in Fig. 18. As shown in Fig. 18, there is a deviation between physical experiments and simulation experiments, and the simulation results are closer to the expected navigation in both the navigation accuracy and the convergence velocity. The main reason is that the robotic fish is seen as a particle in simulation experiment; however, the size of the robot fish cannot be ignored in physics experiments. However, simulation results provide many valuable references for physics experiments, such as learning methods and learning directions.

5. Conclusion

According to the Skinner OC mechanism, a hierarchical cognitive model is constructed to improve the self-learning and self-adaptive ability of mobile robots in unknown and complex environments. The cognitive model adopts the idea of dividing and conquering; it divides the navigation tasks of mobile robots in complex environments into sub-tasks at different levels and solves each sub-task in a smaller

https://doi.org/10.1017/S0263574722001539 Published online by Cambridge University Press

state subspace to reduce the dimension of the state space. The cognitive learning algorithm imitates the thermodynamic process to design and realize an online search, in order to determine the best navigation strategy.

Based on the formation of the cognitive process, the autonomous path planning activities of mobile robots were studied. The experimental results show that: (1) The cognitive model designed can enable mobile robots to successfully avoid obstacles in the environment and reach their goals via a better path. It can automatically acquire knowledge and accumulate experience from the environment, just like animals. Through cognitive learning, we have gradually formed, developed, and perfected autonomous path planning skills. (2) In time-varying and complex dynamic environments, robots can still quickly acquire the optimal path. The adaptive ability of robots shows that the hierarchical structure can reduce the learning difficulty of mobile robots in unknown complex environments, accelerate the learning speed, and avoid dimension disasters. (3) In order to prove the accuracy and rapidity of the hierarchical cognitive model, we compared it with the single cognitive model. The experimental results show that in the same complex environment, the navigation success rate of hierarchical structure is 94%, which is 14% higher than that of single structure. The running time of the layered structure is 200 s, 850 s higher than that of the single structure. (4) To prove the effectiveness of the design model HCNM, we compared it with a similar HRLM. The experimental results show that, in the same complex environment, HCNM navigation path is clearer and better than HRLM in experimental effect and convergence time, and the convergence time is accelerated by 160 s.

The significance of this paper is mainly in two aspects. On one hand, based on the operating conditional reflection mechanism of humans and animals, the paper designs cognitive learning behaviors similar to humans and animals for robot, enabling them to achieve autonomous navigation. It has certain guiding significance for artificial intelligence, robot technology, and cognitive science. On the other hand, compared with the operation-conditioned reflex learning model, the hierarchical cognitive model designed in this paper divides the navigation task of the mobile robot in the complex environment into different levels of sub-tasks through the hierarchical structure, which reduces the learning difficulty of the robot in the unknown environment and accelerates the learning speed of the robot. However, this method also has some limitations. After adding dynamic obstacles, the method needs four rounds of learning to converge, and the convergence speed is reduced, but the reduction is not obvious. In addition, for the selection of new paths, the hierarchical cognitive model needs to constantly try and make mistakes to find the best path. After the robot navigation method based on deep learning trains enough samples, the trained model can complete the navigation task of the new path without retraining. In the follow-up work, the application of deep learning methods in mobile robot navigation will be the next focus of the author.

Author contributions. Jianxian Cai and Fenfen Yan conceived and designed the study. Jianxian Cai and Yan Shi conducted the simulation experiments. Mengying Zhang and Lili Guo conducted the physical experiments. Jianxian Cai wrote the article.

Financial support. This work was supported by the Fundamental Research Funds for the Central Universities (J.X. and F.Y., grant numbers ZY20180112), Research and practice project of higher education reform in Hebei Province from 2019 to 2020 (J.X., grant numbers 2019GJJG471), Scientific Research Plan Projects for The National Key Research and Development Program of China (J.X., grant numbers 2018YFC1503801), the Scientific Research Project Item of Hebei Province Education Department (J.X., grant numbers QN2018317), The second batch of new engineering research and practice projects of the Ministry of Education (J.X., grant numbers E-SXWLHXLX20202607), 2020 education research and teaching reform project of College of disaster prevention technology (J.X., grant numbers JY2020A12).

Conflicts of interest. The authors declare none.

References

[1] R. Kanmani, M. Malathi, S. Rajkumar, R. Thamaraiselvan and M. Rahul, “Autonomous navigation and obstacle avoidance of a microbus,” *J. Adv. Robot. Syst*. **118**(18), 1579–1585 (2013).

[2] B. Srine and A. M. Alimi, “TYPE-2 fuzzy logic controller design using a real-time PSO algorithm applied to “IROBOT CREATE robot,” *Int. J. Robot. Automat*. **30**(4), 384–396 (2015).

https://doi.org/10.1017/S0263574722001539 Published online by Cambridge University Press

[3] C. Wang, Y. C. Soh, H. Wang and H. Wang, "A Hierarchical Genetic Algorithm for Path Planning in a Static Environment with Obstacles," **In:** *Proc. IEEE Conference on Electrical and Computer Engineering*, Honolulu, USA (2002) pp. 1652–1657.

[4] L. Wang, C. M. Luo, M. Li and J. X. Cai, "Trajectory planning of an autonomous mobile robot by evolving ant colony system," *Int. J. Robot. Automat.* **32**(4), 406–413 (2017).

[5] A. Pandey, R. K. Sonkar, K. K. Pandey and D. R. Parhi, "Path Planning Navigation of Mobile Robot with Obstacles Avoidance Using Fuzzy Logic Controller," **In:** *Proc. 8th IEEE Conference on Intelligent System and Control*, Coimbatore, India (2014) pp. 39–41.

[6] V. Tom and S. Branko, "Selective topological approach to mobile robot navigation with recurrent neural networks," *Int. J. Robot. Automat.* **30**(5), 441–446 (2015).

[7] S. X. Yang and C. Luo, "A neural network approach to complete coverage path planning," *IEEE Trans. Syst. Man Cybern.* **34**(1), 718–724 (2004).

[8] C. Luo and S. X. Yang, "A bioinspired neural network for real-time concurrent map building and complete coverage robot navigation in unknown environments," *IEEE Trans. Neural Netw.* **99**(7), 1279–1298 (2008).

[9] J. Shao, J. Y. Yang and C. H. Shi, "Autonomous land vehicle path planning based on learning classifiler system in narrow environments," *Inform. Comput.* **40**(3), 413–417 (2011).

[10] J. J. Ni, X. Y. Li, M. G. Hua and S. X. Yang, "Bioinspired neural network-based Q-learning approach for robot path planning in unknown environments," *Int. J. Robot. Autom.* **31**(6), 464–474 (2016).

[11] X. G. Ruan, P. F. Liu and X. Q. Zhu, "Q - learning environmental cognition method based on odor reward guidance," *J. Tsinghua Univ. (Nat. Sci. Edn.)* **61**(3), 254–260 (2021).

[12] X. Chen and E. K. Abdelkader, "Neural inverse reinforcement learning in autonomous navigation," *Robot. Auton. Syst.* **84**(10), 1–14 (2016).

[13] X. Xu. *Enhanced Learning and Approximate Dynamic Programming*. 1st edn., Science Press, Beijing (2010).

[14] S. Wen, X. Chen, C. Ma, H. K. Lam and S. Hua, "The Q-learning obstacle avoidance algorithm based on EKF-SLAM for NAO autonomous walking under unknown environments," *Robot. Auton. Syst.* **72**, 29–36 (2015).

[15] L. Cherroun and M. Boumehraz, "Fuzzy logic and reinforcement learning based approaches for mobile robot navigation in unknown environment," *J. Meas. Cont.* **9**(1), 109–117 (2013).

[16] B. F. Skinner. *The Behavior of Organisms: An Experimental Analysis* (Appleton-Century Company, New York, 1938) pp. 110–150.

[17] M. Heisenberg, R. Wolf and B. Brembs, "Flexibility in a single behavioral variable of Drosophila," *Learn. Memory* **8**(1), 1–10 (2001).

[18] D. S. Touretzky, N. D. Daw and E. J. Tira-Thompson, "Combining Configural and TD Learning on a Robot," **In:** *Proceedings of 2th IEEE Conference on Development and Learning*, Los Alamitos, CA, USA (2002) pp. 47–52.

[19] X. Zhang, X. G. Ruan, Y. Xiao and J. Huang, "Sensorimotor self-learning model based on operant conditioning for two-wheeled robot," *J. Shanghai Jiaotong Univ. (Sci.)* **22**(2), 148–155 (2017).

[20] X. Zhang, X. Ruan, H. Zhang, L. Liu, C. Han and L. Wang, "Mobile robot's sensorimotor developmental learning from orientation and curiosity," *IEEE Access* **8**, 178117–178129 (2020), doi: 10.1109/ACCESS.2020.3027571.

[21] S. Dominguez, E. Zalama, J. García-Bermejo and P. J., "Robot Learning in a Social Robot," **In:** From Animals to Animats 9, Lecture Notes in Computer Science, vol. **4095** (2006) pp. 691–702.

[22] C. Chang and P. Gaudiano, "Application of biological learning theories to mobile robot avoidance and approach behaviors," *J. Compl. Syst.* **1**(1), 79–114 (1998).

[23] D. Gutnisky and B. Zanutto, "Learning obstacle avoidance with an operant behavior model," *Artif. Life* **10**(1), 65–81 (2004).

[24] X. G. Ruan, L. Z. Dai and N. G. Yu, "Autonomous operant conditioning automata," *Control Theory Appl.* **29**(11), 1452–1457 (2012).

[25] X. G. Ruan and J. Chen, "Operant conditioning reflex learning control scheme based on SMC and Elman network," *Control Decis.* **26**(9), 1398–1406 (2011).

[26] Y. Y. Gao, X. G. Ruan and S. Hongjun, "Operant conditioning learning automatic and its application on robot balance control," *Control Decis.* **28**(6), 930–934 (2013).

[27] J. Chen, X. G. Ruan and L. Z. Dai, "Behavior cognition computational model based on cerebellum and basal ganglia mechanism," *Patt. Recognit. Artif. Intell.* **25**(1), 29–36 (2012).

[28] X. G. Ruan and X. Wu, "The skinner automaton: a psychological model formalizing the theory of operant conditioning," *Sci. China Technol. Sci.* **56**(11), 2745–2761 (2013).

[29] X. G. Ruan, J. Huang and Q. W. Fan, "A learning model based on operant conditioning principles," *Control Decis.* **6**, 1016–1020 (2014).

[30] J. Huang, X. G. Ruan, N. G. Yu, Q. W. Fan, J. M. Li and J. X. Cai, "A cognitive model based on neuromodulated plasticity," *Comput. Intel. Neurosc.* **2016**, 1–15 (2016).

[31] X. G. Ruan, R. Y. W. D.Y.Wang and X. Y. Li, "Monocular vision slam of mobile robot based on Skinner-RANSAC," *J. Beijing Univ. Technol.* **42**(09), 1281–1285 (2016).

[32] J. X. Cai, X. G. Ruan, N. G. Yu, J. Chai and X. Q. Zhu, "Autonomous navigation of mobile robot based on cognitive development," *Comput. Eng.* **44**(01), 9–16 (2018).

[33] D. Rasmussen, A. Voelker and C. Eliasmith, "A neural model of hierarchical reinforcement learning," *PLoS One* **12**(7), 1–39 (2017).

https://doi.org/10.1017/S0263574722001539 Published online by Cambridge University Press

[34] Z. Li, A. Narayan and T. Y. Leong, “An Efficient Approach to Model-Based Hierarchical Reinforcement Learning,” **In:** *Proceedings of 31th AAAI Conference on Artificial Intelligence*, California, USA (2017) pp. 3583–3589.

[35] S. Sagnik and M. Prasenjit, “Semi-Markov decision processes with limiting ratio average rewards,” *J. Math. Anal. Appl.* **455**(1), 864–871 (2017).

Cite this article: J. Cai, F. Yan, Y. Shi, M. Zhang and L. Guo (2023). “Autonomous robot navigation based on a hierarchical cognitive model”, Robotica **41**, 690–712. https://doi.org/10.1017/S0263574722001539