International Journal of Innovative
Computing, Information and Control
Volume **18**, Number **4**, August **2022**

ICIC International ©2022 ISSN 1349-4198
pp. **1163–1180**

REAL-TIME BEHAVIOR DECISION OF MOBILE ROBOT BASED ON THE DELIBERATE/REACTIVE ARCHITECTURE

Jian Xu[1], Liang Wang[1], Qilong Kou[1], Tao Fang[1], Dan You[1], Leiyue Zhou[1] and Yadong Zhang[2,*]

[1]State Grid Luoyang Electric Power Supply Company
No. 259, Kaiyuan Street, Luolong District, Luoyang 471000, P. R. China
{ 2922685831; 2402339697; 56330478; 404582151; 1505615356; 531860654 }@qq.com

[2]Patent Examination Cooperation (Henan) Center of the Patent Office, CNIPA
No. 196, Ping'an Street, Zhengdong New District, Zhengzhou 450016, P. R. China
*Corresponding author: 2461514178@qq.com

Received February 2022; revised May 2022

Abstract. *Excellent behavior decision is a prerequisite for a mobile robot to perform various tasks. Nowadays, a large number of behavior decision methods have been proposed and achieved good performance in static and dynamic environments. Unfortunately, the real scene faced by the mobile robot is a dynamically changing, unstructured environment. Except the static obstacles and regular moving obstacle in the environment, it is also accompanied by some sudden events, such as sudden appearing unknown uncertain obstacles, emergency obstacles approaching the robot with high speed. In order for the mobile robot to complete the specific tasks, the robot must have the capability to handle the emergency events. Aiming at the dynamic complex environments with sudden events, this paper proposes a novel real-time decision making model based on the scalable neural network. It adopts a deliberate/reactive hybrid control architecture to detect the dynamics of obstacles in real time, determine the type of obstacles, and direct the moving state of the robot according to the detected information, so that the robot can make safe and excellent moving decisions in a dynamic and complex environment with sudden events. The deliberate/reactive hybrid control architecture integrates the advantages of the hierarchical architecture and reactive system structure, and the deliberate sub-structure conducts the environment modeling and planning by consulting the hierarchical architecture, while the reactive sub-structure realizes the quick response to the local environment information, and overcomes the uncertainty of the dynamic changes in the environment. Simulation and physical experiments verify the potential of the real-time decision making model.*
Keywords: Mobile robot, Behavior decision, Deliberate/reactive control, Neural network

1. **Introduction.** With the development of the science and technology, mobile robots are getting more and more applications in diverse fields, such as industry [1], agriculture [2], military field [3], medical field [4], and household service [5]. To complete the different complex tasks, the mobile robots must have excellent behavior decision ability. Therefore, how to improve the behavior decision ability of the mobile robot has been a topic of intense research in the recent past.

In the past decades, a substantial amount of methods, such as reinforcement learning [6], depth search principle [7], particle filter algorithm [8], model predictive control [9], finite state machine [10], support vector machine [11], EEG Data-based [12], have been proposed to make corresponding behavior decisions for the mobile robots.

DOI: 10.24507/ijicic.18.04.1163

Despite all these researches have achieved good results in behavior decision of the mobile robots, most of the methods focus on conventional application scenes which involve stationary obstacle, and regular or irregular moving obstacles. Nevertheless, there is the need to dynamically re-organize the robot's work scenarios in co-existence with more complex scenes, involving the unknown uncertain obstacles and/or emergency obstacles in the environment. To deal with these more complex scenes, numerous robot behavior decision methods have been proposed, such as the rescheduling policies [13], hidden Markov model [14], and probabilistic approach [15].

Recently, with the rapid progress of the intelligence science, many artificial intelligence methods, such as neural network [16], particle swarm optimization [17], ant colony optimization [18], fuzzy logic inferencing [19], immuno-inspired method [20], and grey wolf colony optimization [21], are employed to enhance the behavior decision ability of the mobile robots. Among these intelligent methods, due to the excellent ability to approximate any nonlinear system, neural network has become one of the important tools for behavior decision of the mobile robot [22,23].

For the general static and dynamic environment, neural network can direct the robot to execute good behavior decision. However, for environments with the unknown uncertain obstacles and high-speed emergency obstacles, the neural networks will be compromised in terms of safety performance.

With this in mind, a novel real-time decision-making model based on the scalable neural network (SNN) is proposed in this paper. It adopts a deliberate/reactive hybrid architecture to detect the dynamics of obstacles in real time, determine the types of obstacles, and guide the robot's moving in light of the detected scene information; thus, the robot can make safe and effective behavioral decisions in a dynamic and complex environment with sudden events. Results of the simulation and physical experiment illustrate that the real-time decision model proposed in this paper has excellent behavior decision ability.

The remainder of the paper is organized as follows. Section 2 describes the general overview of the real-time decision-making model proposed in this work. Section 3 introduces the theory of the SNN model. This section also explains the generating process of the training samples for the SNN model. Sections 4 and 5 design the simulation and physical experiments to testify the behavior decision performance of the SNN model in dynamic and complex environments respectively, and analyze the corresponding experiment results, while the last section concludes this paper with a discussion on future work.

2. General Overview of the Real-Time Decision System.

2.1. Localization of the mobile robot. Since the mobile robot works in 2D environment, its pose can be depicted by three variables, i.e., (x, y, θ), where (x, y) is its 2D Cartesian coordinates, and θ is its heading direction. Because the real mobile robot used in the following physical environment is a differential drive robot, here we provide its dead reckoning model, as shown in Figure 1.

In Figure 1, θ_1 is the heading between the neighbouring two poses, ΔD is the offset of the robot, and R is its turning radius, L denotes the distance between the two wheels, v denotes the linear speed, while v_l and v_r represent the speed of the left wheel and the right wheel, respectively. d is the excess offset of the right wheel over the left wheel. ω is used to denote the angular speed of the robot, according to the relations among the variables, the following expressions can be achieved:

$$\theta_1 = \frac{d}{L},\ v = \frac{v_l + v_r}{2},\ R = \frac{v}{\omega} = \frac{v_l + v_r}{2\omega},\ \Delta D = \frac{R}{\theta_1},\ \theta_1 \neq 0 \tag{1}$$

Based on the above expressions, the pose formula of the mobile robot can be achieved:

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} -\Delta D/\theta_1 \sin\theta + \Delta D/\theta_1 \sin(\theta+\theta_1) \\ -\Delta D/\theta_1 \cos\theta - \Delta D/\theta_1 \cos(\theta+\theta_1) \\ \theta_1 \end{bmatrix} \tag{2}$$

where $\left(x', y', \theta'\right)$ represents the pose of the current moment, while (x, y, θ) denotes the pose of the last moment. Particularly, when the mobile robot moves along a line, $\theta_1 = 0$, then the pose of the robot changes into

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \Delta D \cos\theta \\ \Delta D \sin\theta \\ 0 \end{bmatrix} \tag{3}$$

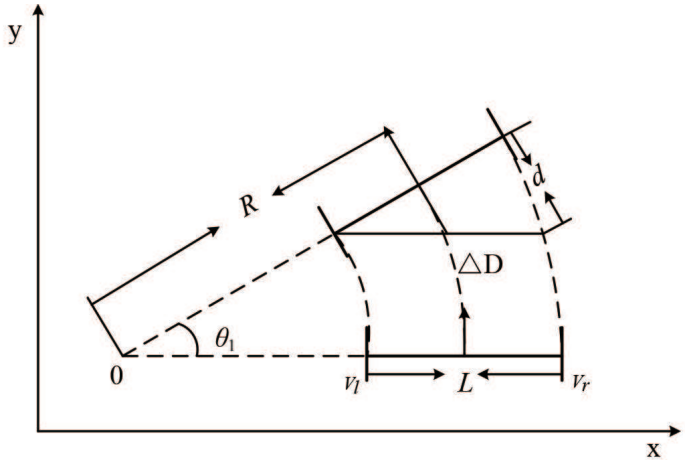


FIGURE 1. Scheme of the dead reckoning model of the differential drive robot

2.2. **Global coordinate system construction.** Based on the data from the odometer sensor, the robot can obtain the position information relative to the start point by means of a dead reckoning. In this work, a lidar is used to detect the obstacles. According to the data obtained by the lidar, the global coordinate system is constructed, as shown in Figure 2.

In Figure 2, $x_0o'y_0$ denotes the coordinate system of the mobile robot, while xoy is the global coordinate system. Axis y_0 represents the movement direction of the robot, θ is the angle between the longitudinal axis of the robot and the axis of the global coordinate system, φ is the angle between the robot and one obstacle, and di is the distance between the robot and the obstacle.

In Figure 2, the robot's position can be obtained by the dead reckoning, assuming that at current time, the robot's position in global coordinate system is $(x_{o'}, y_{o'})$. In the robot coordinate system $x_0o'y_0$, the angle and distance between the obstacle and the robot are known, and then the coordinates of the obstacle can be calculated. Further, its coordinates (x_1, y_1) in the coordinate system $x_1o'y_1$ can also be achieved. After the coordinate transformation, the coordinates (x, y) of the obstacle in the global coordinate system can be obtained by the following Equation (4).

$$\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} r\sin\varphi \\ r\cos\varphi \end{bmatrix}, \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} y_0\cos\theta \\ y_0\sin\theta \end{bmatrix}, \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x_1 + x_{o'} \\ y_1 + y_{o'} \end{bmatrix} \tag{4}$$

With the similar way, the coordinates of other obstacles and the target in the global coordinate system can be achieved finally.

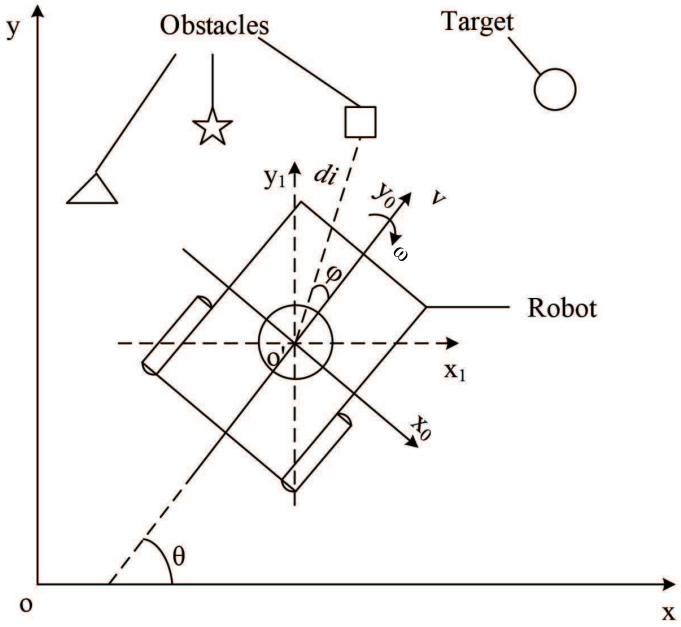


FIGURE 2. Scheme of the robot coordinate system and its kinematic model

2.3. **Description of the environment status.** During the robot moving, change of the environment can directly affect the motion of the robot, and the robot will make corresponding motion decisions according to the environment information. To facilitate the study of real-time decision making of the mobile robot, a set IM of the discrete state events is used to classify the environment. The set IM can be described as follows:

$$IM = \langle A, B, C, D, E \rangle$$

where event A denotes that the distance between the robot and the nearest obstacle is not less than the first safety distance μ, i.e., $A = \{\Phi_A | \mu \leq |q - q_{obs}|\}$, where q and q_{obs} denote the coordinates of the robot and the nearest obstacle to it, respectively. Event B denotes the obstacle that its moving state has been detected, and its moving velocity is not greater than that of the robot. In addition, event B includes 3 sub-events, namely, $B = \langle B_1, B_2, B_3 \rangle$, where B_1, B_2 and B_3 denote the static obstacles, the regular moving obstacles, and the irregular moving obstacles, respectively. Event C denotes the robot reaching the target. Event D denotes the unknown uncertain obstacles that suddenly appear, and it has 2 sub-events, $D = \langle D_1, D_2 \rangle$, where D_1 denotes the unknown uncertain obstacles whose distance to the robot is not greater than the second safety distance ϵ, i.e., $D_1 = \{\Phi_{D_1} || q - q_{obsd}| \leq \epsilon\}$, where q_{obsd} denotes the coordinates of the unknown uncertain obstacles, and the sub-event D_2 denotes the unknown uncertain obstacles whose distance to the robot is greater than the second safety distance ϵ, i.e., $D_2 = \{\Phi_{D_2} | \epsilon < |q - q_{obsd}|\}$. Event E denotes the emergency obstacles which move towards the robot and their moving velocity is greater than that of the robot, $E = \langle E_1, E_2 \rangle$, the sub-events E_1 denotes the emergency obstacles whose distance to the robot is not greater than the first safety distance μ, i.e., $E_1 = \{\Phi_{E_1} || q - q_{obse}| \leq \mu\}$, where q_{obse} denotes the coordinate of the emergency obstacle, and sub-event E_2 denotes the emergency obstacles whose distance to the robot is greater than the first safety distance μ, i.e., $E_2 = \{\Phi_{E_2} | \mu < |q - q_{obse}|\}$.

2.4. **Real-time decision system with the deliberate/reactive structure.** In the environment described by the discrete state set IM, safety of the robot cannot be ensured only relying on the neural network which does not consider the obstacle's types and

velocity, the unknown uncertain obstacles (corresponding to the event D) and the emergency obstacles approaching the robot with high velocity (corresponding to the event E). Aiming at the complex environment, the real-time decision system is introduced and works together with the neural network, and the output of the neural network is a partial of the final decision outputs. This real-time decision system is based on the deliberate/reactive hybrid structure, which integrates the advantages of the hierarchical architecture and reactive system structure, it divides the system into two sub-structures, i.e., deliberate and reactive, as shown in Figure 3, where the deliberate sub-structure conducts the environment modeling and planning by consulting the hierarchical architecture, while the reactive sub-structure realizes the quick response to the local environment information, and overcomes the uncertainty of the dynamic changes in the implementation process. Concretely, it can realize three main functions of the mobile robot: environment perception, environment cognition and motion control.

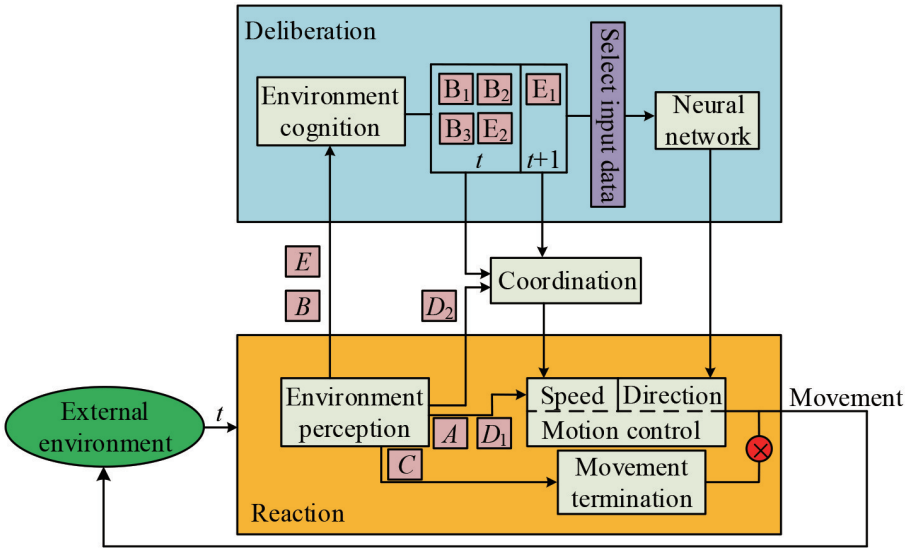


FIGURE 3. Scheme of the real-time decision system under the deliberation/reaction control architecture

The deliberate sub-architecture has the ability of environment cognition and the neural network is included in it, while reactive sub-architecture has the ability of environment perception and motion control. There is a coordination module between the two sub-architectures to coordinate the diverse events, and then output the corresponding velocity to control the robot.

The environment perception module directly faces the external environment, detects and collects the environment data through the sensors. Except collecting the environment data, the environment perception module also has a simple ability to identify the environment status and this ability can be depicted as follows.

1) According to the distance between the robot and the nearest obstacle, the environment perception module judges whether the current environmental status of the robot is event A, if so, the robot velocity will be increased to 2 times of its current velocity. Otherwise, the robot moves with its current velocity.

2) In terms of the distance between the robot and the target, the environment perception module judges whether the robot reaches the target, i.e., the environmental status belongs to event C, if so, the behavior decision task is completed, and the robot stops.

3) A simple environment library is established, and the detected obstacles are classified into two categories, namely, the known obstacles and the unknown uncertain obstacles

suddenly appeared. The known obstacles have been saved in the environment library and their position information is transmitted to the environment cognition module in the deliberate sub-structure. When an unknown obstacle appears, this obstacle is further classified into two events, D_1 and D_2, according to the distance between it and the robot. If it belongs to the event D_1, the robot emergently brakes and re-plans its moving behavior according to current environment information. If it belongs to event D_2, the position information of the obstacle is transmitted to the coordinator. Meanwhile, it is saved to the environment library in the environment perception module.

The environment cognition module locates in the deliberate sub-architecture, and mainly processes the known obstacles from the environment perception module, namely, the events B and E, further classifies these obstacles and forecasts their moving trajectories. Moreover, the environment cognition module records the global coordinates of the known obstacles. For each obstacle, it uses a specific sequence to record the coordinates of the obstacle at different times. Then the following Equation (5) is used to calculate the moving velocity and direction of the obstacles, and modifies them as shown in Table 1.

$$v_{obji} = \frac{\sqrt{\left(x_{t_{i+1}} - x_{t_i}\right)^2 + \left(y_{t_{i+1}} - y_{t_i}\right)^2}}{t_{i+1} - t_i}, \ \alpha_i = \tan^{-1}\left[\frac{y_{t_{i+1}} - y_{t_i}}{x_{t_{i+1}} - x_{t_i}}\right] \tag{5}$$

Table 1. Classification table of different kinds of obstacle

Obstacle type	Speed	Direction
Static	$v_{obji} = 0$	$\alpha_i = 0$.
Regular moving	$v_{obji} \leq v_{rob}$	$\alpha_i = m$, m is a constant.
Irregular moving	$v_{obji} \leq v_{rob}$	$\alpha_i = rand$, $rand$ is a random.
Emergency	$v_{obji} > v_{rob}$	$\alpha_i = n$, n is a constant.

The environment cognition module further classifies the obstacle denoted by the events B and E, as shown in Figure 3. Obstacles in event B can be further classified into three sub-events, namely, B_1, B_2 and B_3. Similarly, obstacles in event E can be further classified into two sub-events, namely, E_1 and E_2, where event E_1 denotes the emergency obstacle close to the robot, so its position at next moment can be calculated with the above Equation (5), and then the decision system puts the position of this obstacle at moment $t+1$ and positions of other obstacles at moment t together and considers comprehensively.

Neural network module also locates in the deliberate module, and it has been trained already. Its main function is to control the robot's movement direction according to the input information. In the selection process of obstacles, the neural network gives priority to the nearest obstacle.

The coordination module locates between the deliberate sub-architecture and reactive sub-architecture, and it receives the obstacle information classified already. Its main function is to control the robot's speed.

1) For the events B_1, B_2, B_3, D_2 and E_2, the robot's moving speed keeps unchanged.

2) For the event E_1, the robot increases its speed to twice of the current speed, and accelerates to avoid the emergency obstacles, through combining with the neural network.

Since the real-time decision-making system uses the neural network to predict the robot's moving direction, in next section, the neural network used in this work will be designed.

3. Scalable Neural Network (SNN) Model.

3.1. Architecture of the SNN.

As shown in Figure 4, the SNN has three layers: sensory layer X, internal layer Y and motor layer Z. Layer Y works as a "bridge" to connect the two "banks", i.e., the sensory layer X and motor layer Z. Layer Z corresponds to the output layer, and has diverse functions in different working status: if the SNN is supervised, layer Z works as another input; if the SNN is tested, the layer Z outputs the corresponding decision in light of the current sensory information.

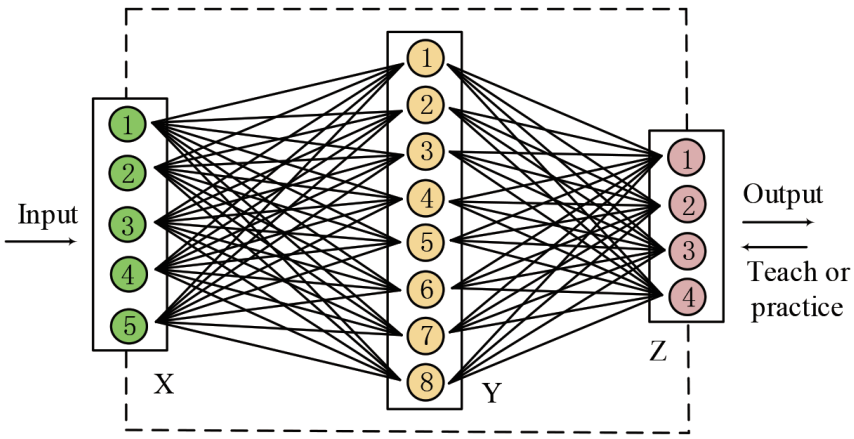


FIGURE 4. Scheme of the architecture of the SNN. Only 8 neurons in internal layer Y are illustrated, actually, number of the neurons in the internal layer Y is increasable according to the practical requirement, that is, its architecture is scalable.

3.2. Theory of the SNN.

Theory of the SNN can be depicted as follows.

1) At time $t = 0$, for each layer of the SNN, its adaptive part $N = (W, G)$ and the response vector $\mathbf{r}$ are initialized, where W is the synaptic weights, and G denotes the neuronal age (denoting its activated times).

2) At time $t = 1, 2, \ldots$, the SNN is trained. During the training, layer X receives the sensory signals and transforms to the layer Y. For each sensory input, the layer Y fires a corresponding neuron. To fire a neuron, the area function f is applied to computing the pre-response energy and updating the adaptive part N and the response vector $\mathbf{r}$, as follows:

$$\left(\mathbf{r}', N'\right) = f(b, N) \tag{6}$$

where $\mathbf{r}'$ represents the new response vector of the fire neuron, and b denotes the bottom-up input from the sensory layer X. Particularly, assuming that the ith neuron in layer Y fires, then the adaptive part N and the response vector $\mathbf{r}$ of this fire neuron can be modified accordingly:

$$W_{xy}(i) = b, \;\; n_i = n_i + 1 \tag{7}$$

where $W_{xy}(i)$ denotes the synaptic weights between the ith neuron in the layer Y and the layer X, and n_i denotes its age. Next, the ith component in the response vector $\mathbf{r}$ is set to 1, and other components are set to 0, thus achieving the new response vector $\mathbf{r}'$. Then, the adaptive part N of the neuron associated with the sample label in the motor layer Z is also modified. For instance, supposing that in motor layer Z, the sample label corresponding to the sensory input is j, then the synaptic weights and the corresponding age n of the jth neuron in motor layer Z will also be modified. The update manner of its age is the same as Equation (6), while the update rule of the synaptic weights can be described as follows:

$$W_{yz}(j) = \frac{n_j - 1}{n_j} W_{yz}(j) + \mathbf{r}' \tag{8}$$

where W_{yz} denotes the synaptic weight between the jth neuron in the motor layer Z and the internal layer Y.

3) After the SNN is trained, then it is tested. During the testing, layer X gets the test samples, and the response value of each neuron in internal layer Y is computed, and the neuron with the highest response value is determined by the following Equation (9), then the response vector $\mathbf{r}$ of this fire neuron in layer Y is modified, and the modifying manner is the same as that in the above training stage.

$$r_y(i) = \frac{1}{1 - W_{xy}(i)}, \ j = \arg \max_{1 \le i \le m} r_y(i) \tag{9}$$

where $r_y(i)$ denotes the response value of the ith neuron in internal layer Y, and m represents the number of the neurons in layer Y.

Subsequently, the response of each neuron in motor layer Z is computed with the following Equation (10). Similarly, the neuron with the highest response value in layer Z is determined, and then the output Z(j) of this fire neuron represents the final predicting output of the SNN.

$$r_z(i) = r'_y \cdot W_{yz}(j), \ j = \arg \max_{1 \le i \le n} r_z(i) \tag{10}$$

where $r_z(i)$ denotes the response value of the ith neuron in motor layer Z, r'_y represents the new response vector of the layer Y, and n is the number of the neurons in motor layer Z.

3.3. **SNN model used in this work.** In this paper, a novel neural network model consisting of two basic SNNs is constructed, as graphically depicted in Figure 5. The working spaces of the two basic SNNs are different; thus the parameter settings in the two SNNs are also different.

SNN-1 always works in the movement of the mobile robot. Layer X_1 is assigned to 3 neurons, according to the dimension of the sensory input. Number of the neurons in layer Y_1 is changeable, in accordance with the actual requirements, in this work it is set to 10,

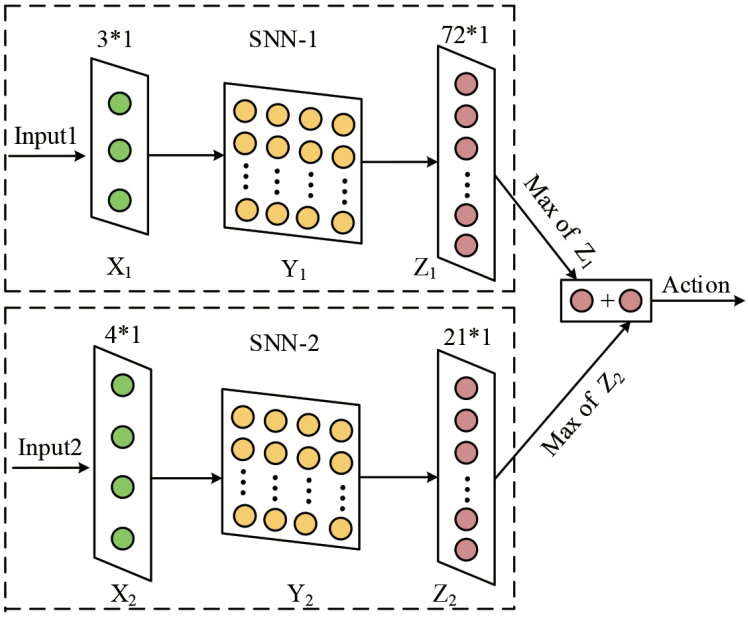


FIGURE 5. Schematic diagram of the new SNN model used in this work

since the performance is good after the test. Motor layer Z is assigned to 72 neurons, corresponding to the 72 movement directions of the robot. In the training of the SNN, inputs of the SNN-1 are the 2016 groups of the training samples. Concretely, the inputs of the SNN-1 can be depicted as follows:

$$Input_1 = \left(\frac{x_1}{14}, \frac{y_1}{14}, \frac{\omega_1}{2\pi}\right) \tag{11}$$

where coordinates (x_1, y_1) denotes one group of the training sample, and the angle ω_1 represents the angle between the positive direction of the coordinate axis X and the line from the origin of the coordinate system to the coordinates (x_1, y_1).

SNN-2 works in risk regions where the distance between the agent and the obstacle is not greater than 1. In this work, the safe threshold is set to 1. It is noteworthy that it is changeable according to the actual requirement. Layer X_2 is assigned to 4 neurons, similarly, the number of the neurons in layer Y_2 is also changeable, and in this work, it is set to 10. Training samples of the SNN-2 come from the 36288 groups of the training data. Concretely, the input of the SNN-2 can be depicted as follows:

$$Input_2 = \left(x_2, y_2, \frac{\theta_1}{2\pi}, \frac{\theta_2}{2\pi}\right) \tag{12}$$

where (x_2, y_2) denotes the coordinates of the obstacle, θ_1 and θ_2 denote the angle between the positive direction of the coordinate axis X and the line from the origin of the coordinate system to the obstacle coordinates (x_2, y_2) and the target coordinates (x_1, y_1), respectively.

In the following two sections, the simulated experiment and physical experiment will be performed to demonstrate the potential of the proposed methodology in this work.

4. Simulation Experiment.

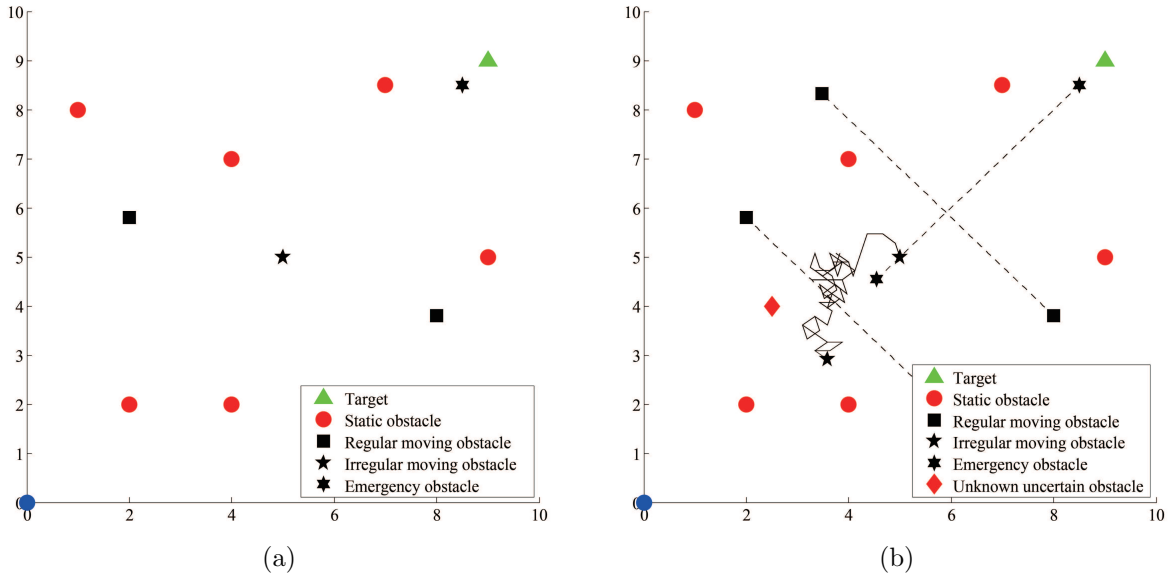
4.1. Setting of the simulation environment.

The simulation environment is constructed in $10 * 10$ space, as shown in Figure 6(a). There are 6 static obstacles, 2 obstacles in regular motion, 1 obstacle in irregular motion and 1 emergency obstacle. Figure 6(b) shows the final scene of this simulation environment. At one moment, an unknown uncertain obstacle appears at position $(2.5, 4)$, the emergency obstacle starts to move, its moving direction is 225° and the step length is set to 0.2.

4.2. Behavior decision based on the real-time decision system.

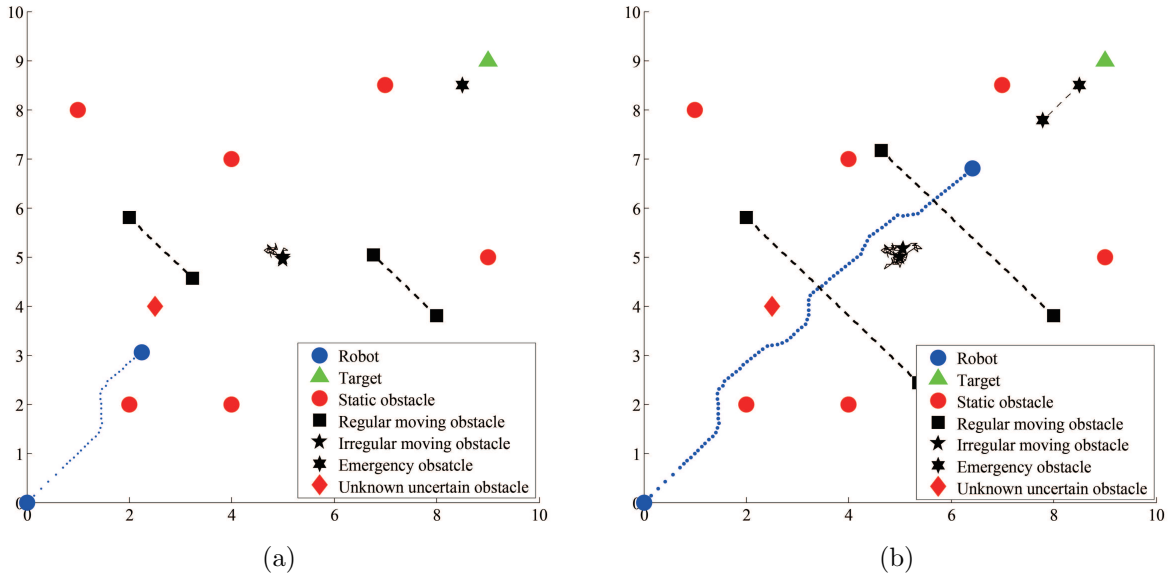
The initial location of the agent is located at $(0, 0)$ and the target locates at $(9, 9)$. The agent's movement step is set to 0.1, the first safe distance μ is set to 2, and the second safe distance ϵ is set to 1. If the distance between the agent and the target is less than 0.4, it is considered that the agent approaches the target and the movement stops, the behavior decision of the agent finishes. In addition, when the agent moves out of the scene boundary, or it collides with the obstacle, or its movement step is greater than 500, the experiment is considered to be failure, and the agent stops the current movement.

During the experiment, at moments t_1 and t_2, the state of the agent and obstacles are recorded. At the moment t_1, the unknown uncertain obstacle suddenly appears, and the agent moves to the 36th step (as shown in Figure 7(a)). At this moment, the distance between the agent and the unknown uncertain obstacle is not greater than the second safe distance ϵ, namely, the agent state belongs to the event D_1, and the agent stops immediately and re-plans in accordance with the environment status. Figure 7(b) provides the environment status at the moment t_2. At this moment, the agent moves to the 96th step, an emergency obstacle enters the first safe distance of the agent, then the agent is under the state of the event E_1, and the real-time decision system predicts the position of



(a) (b)

FIGURE 6. Schematic diagram of the simulation environment at the (a) initial scene and (b) the final scene



(a) (b)

FIGURE 7. Scheme of the simulation environment state at the (a) the first moment t_1 and (b) the second moment t_2

the emergency obstacle according to the agent's environment, and accelerates the agent to avoid it, and Figure 7(b) shows the avoidance process of the agent. Moreover, at the initial stage of the agent motion, the distance between the agent and the nearest obstacle is greater than the first safe distance μ, and the agent has a short acceleration stage, as shown in Figure 8.

Figure 9 provides the distance between the agent and the nearest obstacle in this experiment, and each valley floor corresponds to each obstacle the agent avoids during its movement. When the robot moves about 22 steps, the valley floor corresponds to the agent avoiding the static obstacle. When the robot moves about 50 steps, the valley floor corresponds to the agent avoiding the unknown uncertain obstacle. Similarly, the valley

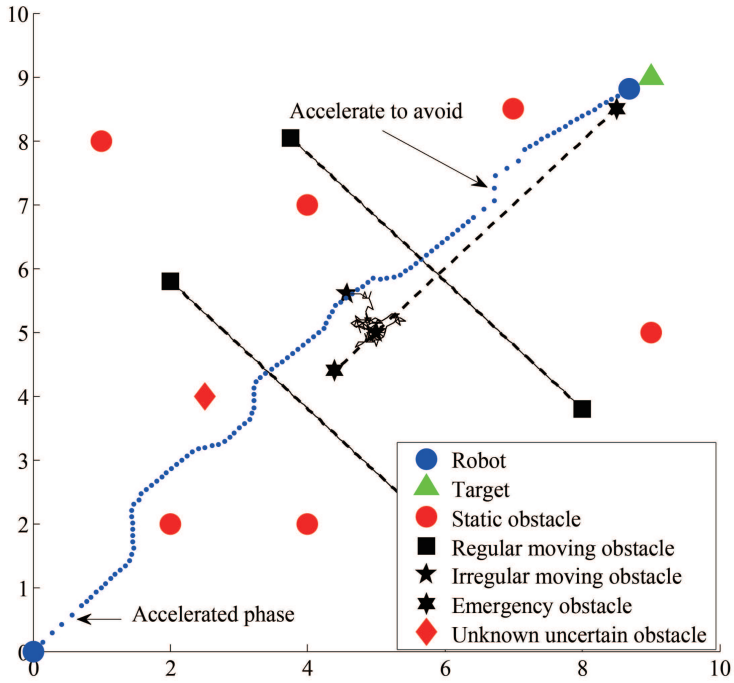


FIGURE 8. The complete moving trajectory of the agent in the simulation environment

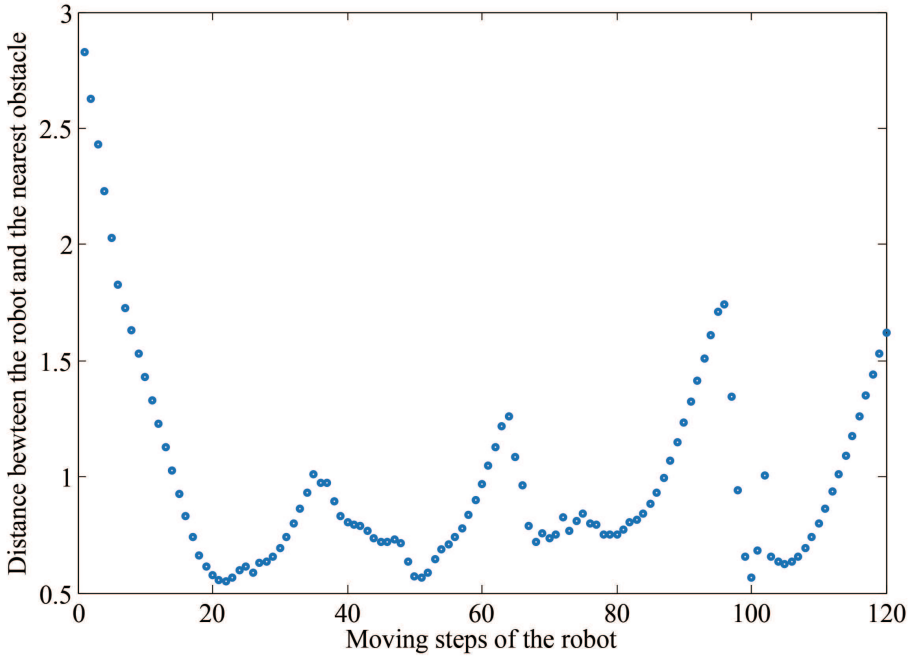


FIGURE 9. The distance between the agent and the nearest obstacle changing with the movement steps of the agent

floor between 67 and 82 steps corresponds to the agent avoiding the obstacle in irregular motion and the second obstacle in regular motion. Finally, when the agent moves about 100 steps, the distance between the agent and the obstacle varies greatly as shown in Figure 9, and it corresponds to the agent accelerating to avoid the emergency obstacle.

4.3. **Comparative experiment.** To analyze the effect of the behavior decision of the agent in dynamic complex environment, in this section, the movement steps of the agent

are used to be the evaluation criteria to evaluate the performance. Since the dynamic window algorithm (DWA) and ant colony optimization (ACO) are often used in the path planning of the mobile robot, we compare the experiment results of the path planned by these three algorithms, namely, the minimal, average and maximal movement steps with these three algorithms in 50 runs are used to illustrate their performance, as shown in Table 2.

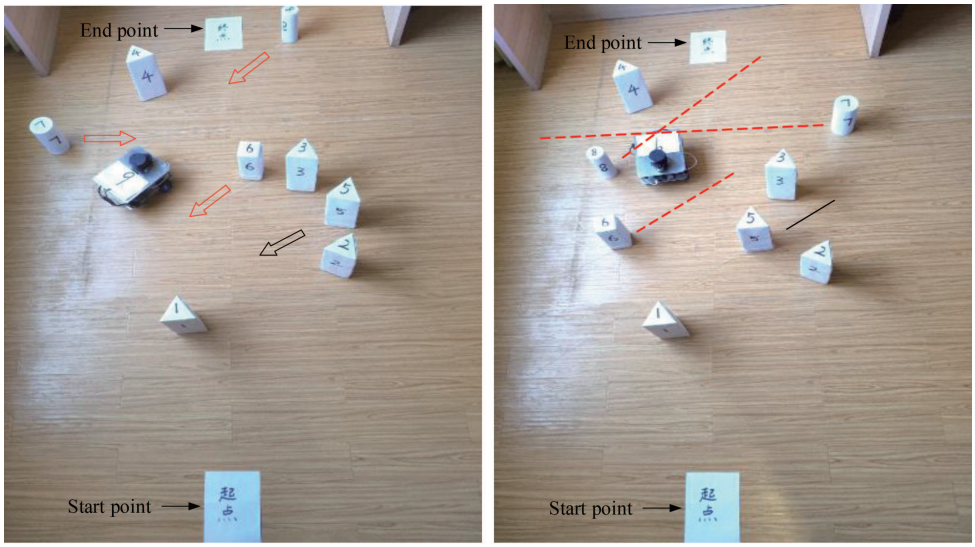
TABLE 2. Experiment results in the 50 runs with the three algorithms

Algorithm	Minimum steps	Maximum steps	Average steps
DWA	140	153	147
ACO	133	151	140
Our method	128	137	131

Table 2 shows that in the 50 experiments, the behavior decision algorithm proposed in this work achieves the best decision effect, which results from the fact that in the dynamic complex scenes, when unexpected emergency events occur, the real-time decision system with the delibetare/reactive framework can react quickly to the changes in the environment and make faster corresponding response; thus the agent can avoid the obstacles quickly and flexibly, leading to the less movement steps.

5. Physical Experiment.

5.1. Experiment environment construction. The real experimental environment is a smooth indoor space with nine obstacles, as shown in Figure 10(a), labels 1-4 denote the static obstacles, label 5 denotes the unknown uncertain obstacle suddenly appearing, labels 6-7 denote the obstacles in regular motion, label 8 denotes the emergency obstacle moving towards the robot quickly, and label 9 denotes the obstacle in irregular motion. In Figure 10(a), the red arrows represent the movement directions of the obstacles 6, 7 and 8, and the black arrow denotes the movement direction of the obstacle 5. In Figure 10(b), the red dotted lines display the trajectories of the obstacles 6, 7 and 8, while the black line displays the trajectory of obstacle 5.

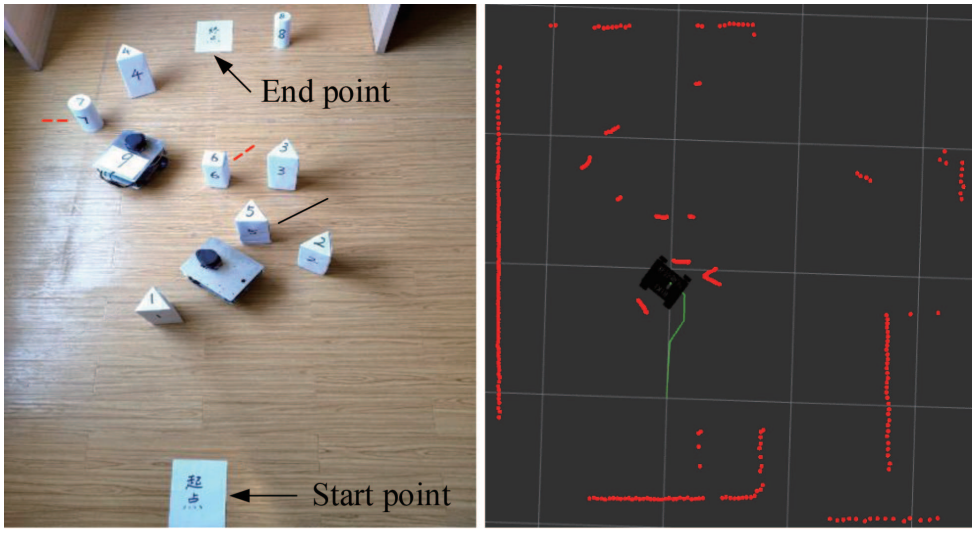


(a) (b)

FIGURE 10. (color online) The experiment environment status at (a) the initial moment and (b) the final moment

5.2. **Performance demonstration of the behavioral decision-making.** The initial linear velocity and the angular velocity of the robot are set to 0.1 m/s and 0.2 rad/s, respectively. During the whole experiment, the mobile phone is used to capture the robot's environmental state at 4 moments when the robot avoids the obstacles, and the display interfaces of the corresponding 4 moments are recorded in the Rviz of the laptop.

At the first moment, the obstacle 5 suddenly appears in the detection field of the robot from behind the obstacle 2. At this moment, the distance between the robot and the obstacle 5 is less than the second safety distance ϵ, so the robot makes an emergent braking and re-plans the path to avoid the obstacle 5. Figure 11(a) displays the experiment status at the first moment. As depicted graphically in Figure 11(a), the robot moves towards its left front direction to avoid the obstacle suddenly appearing. The red dotted lines in Figure 11(a) display the moving trajectories of the obstacles 6 and 7 at this moment. Figure 11(b) provides the display interface in the Rviz at this first moment. In Figure 11(b), the black object resembling a rectangle is the Rikirobot in Figure 11(a), and the green line connecting to the robot represents its moving trajectory, and the red circle points denote the environment information detected by the lidar. When the robot starts moving from the start point, the distance between the robot and the nearest obstacle (label 1) is greater than the second safety distance ϵ, namely, during the robot moving from the start point to the position at the first moment, the robot accelerates in a short distance.

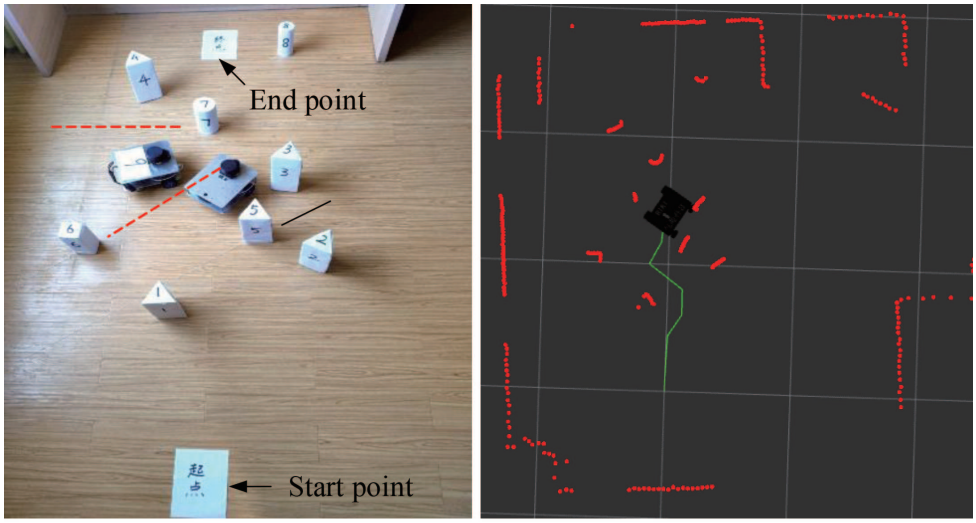


(a) (b)

FIGURE 11. (color online) The experiment environment status at (a) the first moment and (b) the corresponding display interface in the Rviz of the laptop at this moment

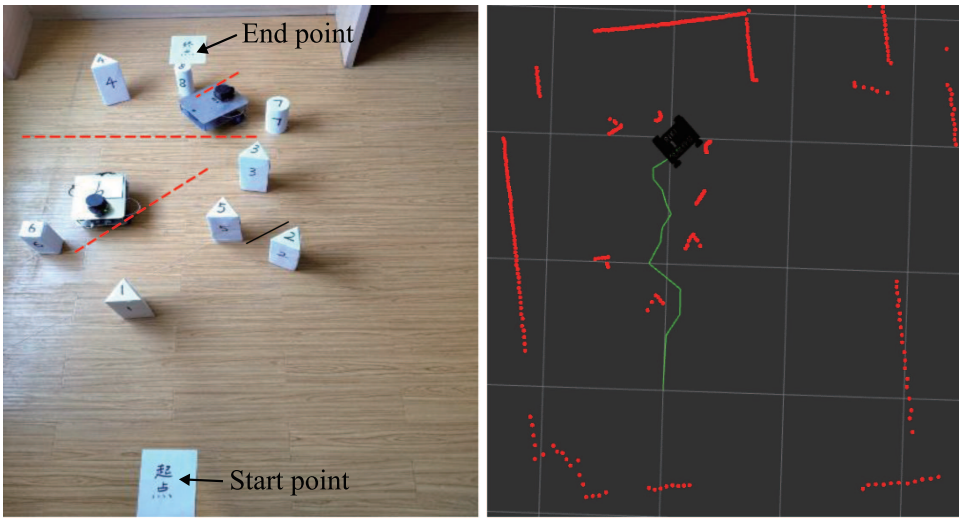
At the second moment, the robot locates between the dynamic obstacle 9 in irregular motion and the static obstacle 3, and the robot first avoids the dynamic obstacle 9 that moves towards it. When the robot moves near the static obstacle 3, it begins to avoid it. At the second moment, the physical environment status and its corresponding interface in Rviz are displayed in Figure 12(a) and Figure 12(b), respectively. The moving trajectories of the obstacles 6 and 7 are shown in Figure 12(a), and at this moment, the obstacle 6 stops.

Furthermore, Figure 13(a) and Figure 13(b) show the physical environment status and its corresponding interface in Rviz at the third moment. At this moment, the robot mainly



(a) (b)

FIGURE 12. (color online) The experiment environment status at (a) the second moment and (b) the corresponding display interface in the Rviz of the laptop at this moment

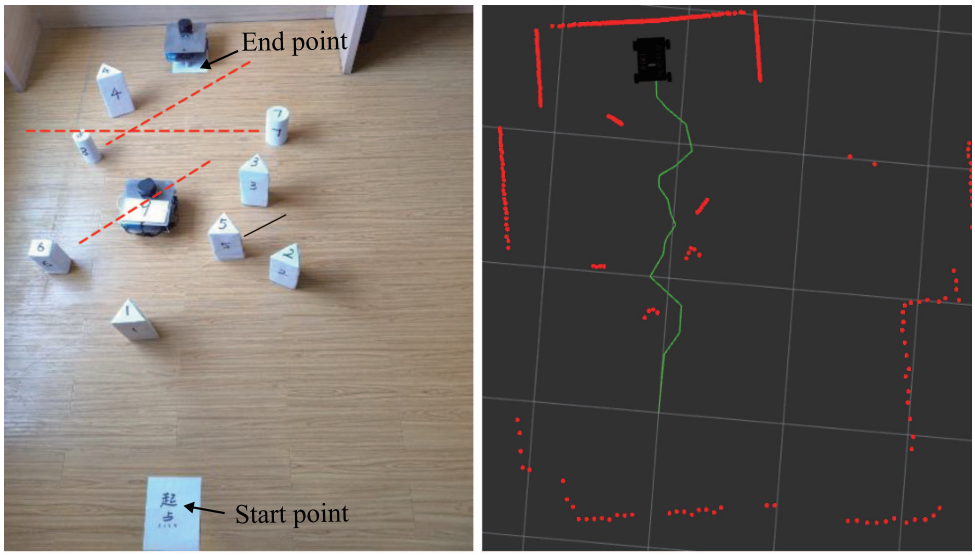


(a) (b)

FIGURE 13. (color online) The experiment environment status at (a) the third moment and (b) the corresponding display interface in the Rviz of the laptop at this moment

avoids the emergent obstacle 8 which moves towards it with high speed. The real-time decision system predicts the location of the obstacle 8 at the next moment according to its speed and movement direction, then the movement speed and direction of the robot are determined, so that the robot can accelerate to avoid the obstacle 8. At this moment, except the obstacles 8 and 9, all other obstacles stop. The moving trajectories of the obstacles and the robot are shown in Figure 13(a) and Figure 13(b).

Finally, Figure 14(a) and Figure 14(b) display the physical environment status and its corresponding interface in Rviz at the fourth moment. At this moment, the robot safely reaches the destination; meanwhile, all the obstacles and the mobile robot stop. Figure



(a) (b)

FIGURE 14. (color online) The experiment environment status at (a) the fourth moment and (b) the corresponding display interface in the Rviz of the laptop at this moment

14(a) shows the moving trajectories of the obstacles 5, 6, 7 and 8, while Figure 14(b) displays the total moving trajectory of the robot.

The above whole experiment process illustrates that the proposed real-time decision-making model based on the deliberate/reactive structure can make perfect moving decision for the mobile robot, even in the dynamic complex environment with the unexpected emergent events. The inherent reason lies in the fact that in the dynamic complex environment, whatever the static obstacles, dynamic obstacles in regular motion, dynamic obstacles in irregular motion, or the unexpected emergent obstacles suddenly appearing, the real time decision system with the delibetare/reactive framework can react quickly to the changes in the environment and make faster response, causing the robot to avoid the obstacles quickly and flexibly.

6. **Conclusions and Future Work.** This paper proposes a real-time behavioral decision-making model for mobile robots in dynamic complex environment. The real-time decision model is based on the SNN and uses a deliberate/reactive hybrid control architecture, to detect the dynamics of obstacles in the environment, determine the type of obstacles, and decide the movement behavior of the robot based on the detected information. Experiment results in the simulation environment and physical environment demonstrate the potential of the proposed method.

Although the real-time decision model has achieved good performance in robot behavioral decision, there are still some aspects needed to be further studied in the next research.

1) The scene information is achieved only through the lidar equipped on the mobile robot, which can be equipped with visual sensors and other type sensors. Hence, investigating the multi-sensor fusion technology to achieve more precise scene information in practical applications requires further research.

2) In this work, the real-time decision system is designed for only a single robot. How to extend the decision system and make it suitable for multi-robot system is another research direction in the future.

REFERENCES

[1] E. D. Lambert, R. Romano and D. Watling, Optimal smooth paths based on clothoids for car-like vehicles in the presence of obstacles, *International Journal of Control, Automation and Systems*, vol.19, pp.2163-2182, 2021.

[2] M. M. Rahman, K. Ishii and N. Noguchi, Optimum harvesting area of convex and concave polygon field for path planning of robot combine harvester, *Intelligent Service Robotics*, vol.12, pp.167-179, 2019.

[3] M. Labbadi and M. Cherkaoui, Robust adaptive global time-varying sliding-mode control for finite-time tracker design of quadrotor drone subjected to Gaussian random parametric uncertainties and disturbances, *International Journal of Control, Automation and Systems*, vol.19, pp.2213-2223, 2021.

[4] M. S. Qureshi, P. Swarnkar and S. Gupta, A supervisory on-line tuned fuzzy logic based sliding mode control for robotics: An application to surgical robots, *Robotics and Autonomous Systems*, vol.109, pp.68-85, 2018.

[5] S. Wan, Z. Gu and Q. Ni, Cognitive computing and wireless communications on the edge for health-care service robots, *Computer Communications*, vol.149, pp.99-106, 2020.

[6] R. Perula-Martinez, A. Castro-Gonzlez, M. Malfaz, F. Alonso-Martin and M. A. Salichs, Bioinspired decision-making for a socially interactive robot, *Cognitive Systems Research*, vol.54, pp.287-301, 2019.

[7] N. Kousi, D. Dimosthenopoulos, A. S. Matthaiakis, G. Michalos and S. Makris, AI based combined scheduling and motion planning in flexible robotic assembly lines, *Procedia CIRP*, vol.86, pp.74-79, 2019.

[8] X. Chen and J. Huang, Combining particle filter algorithm with bio-inspired anemotaxis behavior: A smoke plume tracking method and its robotic experiment validation, *Measurement*, vol.154, DOI: 10.1016/j.measurement.2020.107482, 2020.

[9] S. Yu, Y. Guo, L. Meng, T. Qu and H. Chen, MPC for path following problems of wheeled mobile robots, *IFAC-PapersOnLine*, vol.51, pp.247-252, 2018.

[10] E. R. M. Aleluya, A. D. Zamayla and S. L. M. Tamula, Decision-making system of soccer-playing robots using finite state machine based on skill hierarchy and path planning through Bezier polynomials, *Procedia Computer Science*, vol.135, pp.230-237, 2018.

[11] B. Sebastian and P. Ben-Tzvi, Support vector machine based real-time terrain estimation for tracked robots, *Mechatronics*, vol.62, pp.1-10, 2019.

[12] M. R. V. Sanchez, S. Mishima, M. Fujiwara, G. Ai, M. Jouaiti, Y. Kobryn, S. Rimbert, L. Bougrain, P. Hénaff and H. Wagatsuma, Methodological design for integration of human EEG data with behavioral analyses into human-human/robot interactions in a real-world context, *ICIC Express Letters*, vol.14, no.7, pp.693-701, 2020.

[13] B. Cai, S. Huang, D. Liu and G. Dissanayake, Rescheduling policies for large-scale task allocation of autonomous straddle carriers under uncertainty at automated container terminals, *Robotics and Autonomous Systems*, vol.62, pp.506-514, 2014.

[14] T. Peynot, R. Fitch, R. McAllister and A. Alempijevic, *Resilient Navigation through Online Probabilistic Modality Reconfiguration*, Springer-Verlag, Berlin, 2013.

[15] L. Nardi and C. Stachniss, User preferred behaviors for robot navigation exploiting previous experiences, *Robotics and Autonomous Systems*, vol.97, pp.204-216, 2017.

[16] 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*, vol.51, no.30, pp.210-215, 2018.

[17] D. Wang, H. Wang and L. Liu, Unknown environment exploration of multi-robot system with the FORDPSO, *Swarm and Evolutionary Computation*, vol.26, pp.157-174, 2016.

[18] W. Gao, Q. Tang, B. Ye, Y. Yang and J. Yao, An enhanced heuristic ant colony optimization for mobile robot path planning, *Soft Computing*, vol.24, pp.6139-6150, 2020.

[19] B. K. Patle, D. R. Parhi, A. Jagadeesh and S. K. Kashyap, Application of probability to enhance the performance of fuzzy based mobile robot navigation, *Applied Soft Computing*, vol.75, pp.265-283, 2019.

[20] A. Raza and B. R. Fernandez, A multi-tier immuno-inspired framework for heterogeneous mobile robotic systems, *Applied Soft Computing*, vol.71, pp.333-352, 2018.

[21] 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*, vol.43, pp.7869-7891, 2018.

[22] D. Wang, Y. Duan and J. Weng, Motivated optimal developmental learning for sequential tasks without using rigid time-discounts, *IEEE Trans. Neural Networks and Learning Systems*, vol.29, no.10, pp.4917-4931, 2018.
[23] D. Wang, W. Si, Y. Luo, H. Wang and T. Ma, Goal-directed autonomous navigation of mobile robot based on the principle of neuromodulation, *Network: Computation in Neural Systems*, vol.30, nos.1-4, pp.79-106, 2019.

Author Biography

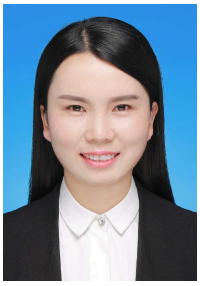
Jian Xu obtained a bachelor's degree. Now he is a senior economist and the general manager of State Grid Luoyang Electric Power Supply Company. His research interests include economic management, electrical engineering, and power security.

Liang Wang obtained a master's degree. Now he is a senior engineer in the Communication and Information Center of State Grid Luoyang Electric Power Supply Company. His research interests include computer application technology, and digital technology application.

Qilong Kou obtained a master's degree. Now he is a senior engineer in the Communication and Information Center of State Grid Luoyang Electric Power Supply Company. His research interests include computer science technology, and artificial intelligence application.

Tao Fang obtained a master's degree. Now he is a senior engineer in the Operation and Maintenance Department of State Grid Luoyang Electric Power Supply Company. His research interests include power system automation, power transmission and substation technology.

Dan You obtained a bachelor's degree. Now she is a senior economist in the Internet Department of State Grid Luoyang Electric Power Supply Company. Her research interests include economic management, and digital technology application.



Leiyue Zhou obtained a master's degree. Now she is an engineer in the Communication and Information Center of State Grid Luoyang Electric Power Supply Company. Her research interests include computer application technology and network security.

Yadong Zhang obtained his master's degree from Zhengzhou University in 2020. Now he works in the Patent Examination Cooperation (Henan) Center of the Patent Office, CNIPA. His research interests include artificial intelligence and machine learning.