Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Behavioral control task supervisor with memory based on reinforcement learning for human–multi-robot coordination systems*&#

Jie HUANG[1,2,3], Zhibin MO[1,2,3], Zhenyi ZHANG[1,2,3], Yutao CHEN[†‡1,2,3]

[1]*School of Electrical Engineering and Automation, Fuzhou University, Fuzhou 350108, China*

[2]*5G+ Industrial Internet Institute, Fuzhou University, Fuzhou 350108, China*

[3]*Key Laboratory of Industrial Automation Control Technology and Information Processing of Fujian Province, Fuzhou University, Fuzhou 350108, China*

[†]E-mail: yutao.chen@fzu.edu.cn

Received June 14, 2021; Revision accepted Feb. 15, 2022; Crosschecked May 11, 2022

Abstract: In this study, a novel reinforcement learning task supervisor (RLTS) with memory in a behavioral control framework is proposed for human–multi-robot coordination systems (HMRCSs). Existing HMRCSs suffer from high decision-making time cost and large task tracking errors caused by repeated human intervention, which restricts the autonomy of multi-robot systems (MRSs). Moreover, existing task supervisors in the null-space-based behavioral control (NSBC) framework need to formulate many priority-switching rules manually, which makes it difficult to realize an optimal behavioral priority adjustment strategy in the case of multiple robots and multiple tasks. The proposed RLTS with memory provides a detailed integration of the deep Q-network (DQN) and long short-term memory (LSTM) knowledge base within the NSBC framework, to achieve an optimal behavioral priority adjustment strategy in the presence of task conflict and to reduce the frequency of human intervention. Specifically, the proposed RLTS with memory begins by memorizing human intervention history when the robot systems are not confident in emergencies, and then reloads the history information when encountering the same situation that has been tackled by humans previously. Simulation results demonstrate the effectiveness of the proposed RLTS. Finally, an experiment using a group of mobile robots subject to external noise and disturbances validates the effectiveness of the proposed RLTS with memory in uncertain real-world environments.

Key words: Human–multi-robot coordination systems; Null-space-based behavioral control; Task supervisor; Reinforcement learning; Knowledge base

https://doi.org/10.1631/FITEE.2100280 **CLC number:** TP18

1 Introduction

Human–multi-robot coordination systems (HMRCSs) (Zheng et al., 2017; Lippi and Marino, 2018) have been used in homes (Lee and Kim, 2018), military detection (Gans and Rogers, 2021), rescue robots (Queralta et al., 2020), industrial processes (Robla-Gómez et al., 2017), and outer space exploration (Bluethmann et al., 2003). Coordination between humans and robots can improve control efficiency and robustness, and allows robots to

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (No. 61603094)

& A preliminary version was presented at the Chinese Intelligent Systems Conference, Fuzhou, China, Oct. 16–17, 2021

Electronic supplementary materials: The online version of this article (https://doi.org/10.1631/FITEE.2100280) contains supplementary materials, which are available to authorized users

ORCID: Jie HUANG, https://orcid.org/0000-0001-7346-5034; Yutao CHEN, https://orcid.org/0000-0001-6748-2866

successfully complete specific predetermined tasks while encountering emergencies such as partial failure. On this topic, researchers have proposed different methods to achieve efficient and practical human–multi-robot coordination. An automated advising agent system has been introduced to solve the problem of supervising and operating multiple robots simultaneously in a large-scale human–multi-robot coordination system (HMRCS) for search and rescue (Rosenfeld et al., 2017). A distributed control strategy based on robust adaptive control has been designed to realize efficient physical interaction between human and multiple manipulators (Lippi et al., 2019). An intelligent robot navigation system has been proposed to realize human–multi-robot coordination control, where multiple robots can implement tasks with priority with external disturbances such as human and robot motion (Bajcsy et al., 2019). A robot fault human information processing (RF-HIP) model has been designed (Honig and Oron-Gilad, 2018) to solve communication and perception problems among humans and robots, in which efficient decision-making and control are achieved under the condition of perception error and response failure in some robots.

Although these approaches achieve effective cooperation among humans and robots, they do not consider repeated human intervention in the task execution process when robots encounter similar emergencies or failures. This may require consistent human attention, increasing the burden of human monitoring and controlling and the probability of making mistakes. In addition, frequent human participation and repeated intervention can result in significant decision-making time cost and task tracking errors, which seriously affect task execution process and may even cause safety problems. To tackle these problems, researchers have recently proposed learning methods to memorize human intervention information, called human-in-loop (HIL) hybrid enhanced intelligence (Zheng et al., 2017). For example, an HIL hybrid enhanced intelligent closed-loop system has been built by introducing machine learning and human knowledge into robotic decision-making (Fu et al., 2019). A hierarchical emotional episodic memory method has been proposed in social human–robot collaboration (Lee and Kim, 2018), where robots are able to remember and manage human experiences and predict and prevent emergencies.

However, conflicts among humans and robots are inevitable in human–robot cooperation as robots are flexibly combined and designated to complete tasks with increasing complexity. Null-space-based behavioral control (NSBC) (Antonelli and Chiaverini, 2006) is a practical method for resolving task conflicts. NSBC ensures that tasks with higher priority are fully executed, while those with lower priority can be partially executed using null-space projection and system redundancy. To this end, one of the key issues in NSBC is to design an implicit centralized supervisor to manage multiple tasks that may be in conflict. Traditional supervisors include the finite state automaton (FSA) method (Baizid et al., 2017), fuzzy logic method (Moreno et al., 1993; Huang et al., 2019), and model predictive control (MPC) (Chen et al., 2020), which can realize real-time dynamic switching of task priority. However, FSA and fuzzy logic methods need to manually formulate priority switching rules. These methods are not intuitive when their task space and state space are large. In addition, the supervisor based on MPC has the disadvantages of requiring an accurate mathematical model and high cost of real-time computation.

The task supervisor design is even more troublesome in the case of human intervention in HMRCSs, in which questions like when and how humans intervene are not easy to answer. In early studies, a human drift diffusion model (DDM) has been proposed to account for human intervention in the NSBC framework (Huang et al., 2020). However, these works do not consider the repeated and frequent human intervention problems which may cause high decision-making time cost and significant task tracking errors. Motivated by these issues, we propose a novel reinforcement learning task supervisor (RLTS) with memory in the NSBC framework (Mo et al., 2022). A deep Q-network (DQN) and a long short-term memory (LSTM) knowledge base are employed to address the problems of dynamic task priority adjustment and repeated human participation. In particular, the proposed RLTS with memory first memorizes human intervention history from the situations when robot systems are not confident in emergencies, and then reloads the history information when encountering the situation that was previously tackled by humans. RLTS is first trained

off-line, and can obtain dynamic task priority adjustment strategies online depending on the environment. This overcomes the defects of the traditional FSA and MPC task supervisors. This paper is significantly improved on the basis of Mo et al. (2022) in the following areas: (1) An artificial potential field model is introduced to improve the state selection accuracy of the RLTS. So, the proposed RLTS with memory can accurately determine whether to trigger human intervention, and the robots have the ability to avoid dynamic obstacles. (2) In addition to numerical simulations, experiments are conducted using real mobile robots to demonstrate the effectiveness of the proposed RLTS.

2 Preliminaries and problem statement

2.1 Markov decision process

The Markov decision process (MDP) (Aviv and Pazgal, 2005; Zhang et al., 2021) can be used to describe the interaction between agents and the environment. An agent can implement an action in the environment, obtain a new state and reward, and evaluate the next action according to the state at the current time. MDP contains a five-tuple (S, A, T, R, γ), where S represents a set of states, A represents a set of actions, T: $S \times A \times S \to [0, 1]$ denotes the transition probability function with $T^a_{ss'}=P[S_{t+1} = s'|S_t = s, A_t = a]$, R: $S \times A \times S \to \mathbb{R}$ denotes the reward function with $R^a_s = E[R_{t+1}|S_t = s, A_t = a]$, and γ is the discounting factor for future reward and $\gamma \in [0, 1]$.

The state-value function $v^\pi(s)$ and the action-value function $q^\pi(s, a)$ following policy π can be expressed as

$$v^\pi(s)=E_\pi[G_t|S_t=s]=\sum_{s'\in S} T^a_{ss'}[R^a_s+\gamma v^\pi(s')], \quad (1)$$

$$\begin{aligned} q^\pi(s,a) &= E_\pi[G_t|S_t = s, A_t = a] \\ &= R^a_s+\gamma\sum_{s'\in S} T^a_{ss'}\sum_{a'\in A}\pi(a'|s')q^\pi(s',a'), \end{aligned} \quad (2)$$

where G_t represents the attenuation sum of all rewards from state S_t at time t to the final state.

2.2 Deep Q-networks

In DQNs, the agent tries to learn the optimal action value function $q^*(s, a)$ through value iteration update (Mnih et al., 2015; Wang et al., 2020). In the value iteration process, DQNs introduce a deep neural network $q_w(s, a)$ with parameter w to replace the Q-table in Q-learning (Watkins and Dayan, 1992). The parameter w is learned by randomly sampling a mini-batch n_m of transitions from an experience replay buffer and minimizing the squared temporal-difference (TD) error. The cost function can be calculated as

$$L(w) = \sum_{\mu=1}^{n_m}(G_\mu - q(s, a; w))^2, \quad (3)$$

where $G_\mu = r + \gamma\max_{a'}\hat{q}(s', a'; w^-)$ is the TD goal, also called the expected state-action reward, w^- is the neural network parameter of the target Q-network, s and s' are the current and next states respectively, and a and a' are the current selected action and next action respectively.

2.3 Null-space-based behavioral control

The NSBC approach can be designed in a three-level structure consisting of elementary behaviors, composite behaviors, and a task supervisor (Baizid et al., 2015, 2017). The speed output of each elementary behavior can be combined and superimposed according to the geometric rules of null-space projection to obtain reference speed signals of the robots.

2.3.1 Elementary behaviors

In NSBC, elementary behaviors are the atomic task functions to be controlled at the kinematic level. They can be expressed by a function that involves the degree of freedom of the system and variables to be controlled.

Define $\boldsymbol{\rho} \in \mathbb{R}^m$ as the task variable and $\boldsymbol{\delta} \in \mathbb{R}^n$ as the system configuration. $\boldsymbol{\rho}_i$ is the function related to $\boldsymbol{\delta}_i$, $i = 1, 2, \ldots, \kappa$, and κ is the number of robots in the HMRCS. Thus, the corresponding task function of each robot can be expressed as

$$\boldsymbol{\rho}_i = f(\boldsymbol{\delta}_i). \quad (4)$$

The corresponding differential relationship of Eq. (4) is

$$\dot{\boldsymbol{\rho}}_i = \frac{\partial f(\boldsymbol{\delta}_i)}{\partial \boldsymbol{\delta}_i}\boldsymbol{v}_i = \boldsymbol{J}_i(\boldsymbol{\delta}_i)\boldsymbol{v}_i, \quad (5)$$

where $\boldsymbol{J}_i(\boldsymbol{\delta}_i) \in \mathbb{R}^{m\times n}$ is the configuration-related task Jacobian matrix of the i^{th} robot and $\boldsymbol{v}_i \in \mathbb{R}^n$ is

the stacked vector of the i^{th} robot. n depends on the specific system and controllable degree of freedom; for example, for a mobile robot, $n=3$. The system configuration here is referred to the position and orientation. The reference velocity $\boldsymbol{v}_{\text{d}}$ can be calculated by converting the local linear mapping (5) into the least-square formula. The integration of reference velocity would incur a certain drift of the reconstructed position of the robot, which can be compensated for by the following closed-loop inverse kinematics (CLIK) algorithm (Antonelli and Chiaverini, 2006):

$$\boldsymbol{v}_{i,\text{d}}=\boldsymbol{J}_i^{\dagger}(\dot{\boldsymbol{\rho}}_{i,\text{d}}+\boldsymbol{\Lambda}_i(\boldsymbol{\rho}_{i,\text{d}}-\boldsymbol{\rho}_i)), \tag{6}$$

where $\boldsymbol{J}_i^{\dagger}=\boldsymbol{J}_i^{\text{T}}(\boldsymbol{J}_i\boldsymbol{J}_i^{\text{T}})^{-1}$ is the pseudo-inverse matrix of $\boldsymbol{J}_i$ and $\boldsymbol{\Lambda}_i$ is a positive-definite constant gain matrix. Let $\boldsymbol{\rho}_{i,\text{d}},\boldsymbol{\rho}_i$ be the desired and actual positions of the i^{th} robot respectively, and $\tilde{\boldsymbol{\rho}}_i=\boldsymbol{\rho}_{i,\text{d}}-\boldsymbol{\rho}_i$ be the task tracking error.

Remark 1 A behavior is also called a task or mission in behavioral control in this study.

2.3.2 Composite behaviors

Composite behaviors are combinations of multiple elementary behaviors and determined by the priority of tasks.

Let $\boldsymbol{\rho}_j\in\mathbb{R}^{m_j}$ be the j^{th} task function, where $j=1,2,\cdots,K$ and m_j denotes the space dimension of the j^{th} task. Define a time-related priority function $g(j,t):N_K\times[0,\infty]\to N_K$, $N_K=\{1,2,\ldots,K\}$, representing a mapping between the task function index and the priority index. Then, the composite behavior combination rules can be defined as follows:

1. $j=1$ is assumed as the top priority. $j_\alpha>j_\beta$ indicates that j_β owns a higher priority than j_α. The behavior with priority j_α cannot interfere with the behaviors with priority j_β, $\forall j_\alpha$, $j_\beta\in N_K$, $j_\alpha\neq j_\beta$. The behaviors with a lower priority are allowed to be executed in the null-space of all behaviors with a higher priority.

2. The behavior Jacobian matrices $\boldsymbol{J}_{g(j,t)}\in\mathbb{R}^{m_j\times n}$, $j=1,2,\cdots,K$, determine the mappings from the generalized velocities of the system to the behavior velocities.

3. The dimension of the lowest-level task m_k can be greater than $m_n-\sum_{j=1}^{K-1}m_j$. So, the dimension m_n of the behavior space is larger than the total dimension of all behaviors.

4. $g(j,t)$ depends on a task supervisor according to task requirements and the sensor feedback information.

The velocity output of composite behaviors at time t is given by the following recursive expression by distributing a given priority to multiple elementary behaviors. Therefore, the velocity output of composite behaviors at time t can be expressed as

$$\boldsymbol{v}_{\text{d}}(t)=\boldsymbol{v}_1(t)+\sum_{i_{\text{c}}=2}^{\zeta}\boldsymbol{N}_{i_{\text{c}}-1}(t)\boldsymbol{v}_{i_{\text{c}}}(t), \tag{7}$$

where $\boldsymbol{v}_{i_{\text{c}}}(t)$ is the speed output of elementary behaviors i_{c}, $i_{\text{c}}=2,3,\ldots,\zeta$ represents the behavior priority of the behaviors in the i^{th} robot, ζ is the quantity of all elementary behaviors, and $\boldsymbol{N}_{i_{\text{c}}}(t)=\boldsymbol{I}-\boldsymbol{J}_{i_{\text{c}}}^{\dagger}(t)\boldsymbol{J}_{i_{\text{c}}}(t)$ is a projection onto the null-space of the augmented Jacobian matrix $\boldsymbol{J}_{i_{\text{c}}}(t)$. $\boldsymbol{J}_{i_{\text{c}}}(t)$ can be expressed as

$$\boldsymbol{J}_{i_{\text{c}}}(t)=[\boldsymbol{J}_1^{\text{T}}(t),\boldsymbol{J}_2^{\text{T}}(t),\ldots,\boldsymbol{J}_{\zeta}^{\text{T}}(t)]^{\text{T}}. \tag{8}$$

Remark 2 Only those elementary behaviors in conflicts are supposed to be allocated priorities among all elementary behaviors, while inter-independent behaviors can be executed autonomously with available degree of freedom.

2.3.3 Behavioral control task supervisor

In the NSBC approach, the real-time switching between composite behaviors must be assigned by a so-called task supervisor (Baizid et al., 2015), which can dynamically manage and adjust the composite behaviors. It can be triggered flexibly according to task requirements and sensor information. The design of the behavioral control task supervisor module will be detailed later.

2.4 Problem description and assumptions

The goal of this study is to develop an optimal adjustment strategy of the behavioral priority and reduce the frequency of repeated human participation. First, we have the following commonly used assumptions:

Assumption 1 All robot tasks are local tasks. There is no information interaction between robots. Each robot in HMRCSs pursues its own maximum benefit.

Assumption 2 Robots are autonomously controlled by a robot controller. Human intervention will take over control only when the robot system is

not confident enough or fails to complete the tasks within a limited time period.

3 RLTS with memory

In this section, we describe the design of a novel RLTS with memory to control the HMRCS. It is designed to obtain an optimal behavioral priority adjustment strategy and reduce the frequency of repeated human participation in the HMRCS by elaborately integrating the NSBC approach, DQN, and a specific memory base.

3.1 Basic framework of the HIL hybrid enhanced intelligence system

The basic framework of the HIL hybrid enhanced intelligence was systematically summarized in Zheng et al. (2017). When autonomous unmanned systems are in an abnormal situation in which robots are unable to execute tasks in a limited time period or the host computer is not confident enough to complete those specific tasks, the autonomous unmanned systems ask for human assistance and automatically update operation information to the knowledge base after estimating the confidence and a cognitive state of the host computer. Accuracy and credibility can be enhanced after introducing human prediction and intervention. The frequency of human participation can be reduced due to the knowledge base.

3.2 HMRCS under the NSBC-based framework

Based on the aforementioned framework, a novel HMRCS under the NSBC-based framework is designed, as shown in Fig. 1. It consists of six key elements: an autonomous decision maker, an RLTS with memory, the typical NSBC scheme, the actual physical environment, a data processing station based on DDM, and an HIL decision maker. They are marked ①–⑥ in the block diagram. A detailed description of each element is given below:

1. The autonomous decision maker. As shown in element ①, this element is responsible for robot moving autonomously according to preset programs or control laws. In our application scenario, the preset or designated programs include the tracking task, obstacle avoidance task, and collision avoidance task.

2. The RLTS with memory. As shown in element ②, the role of the supervisor is to obtain an optimal behavioral priority adjustment strategy and dynamically adjust the task priority during the task execution process. It learns, records, and reloads the control input of human intervention. This module will be further described in Section 3.3.

3. The NSBC scheme. As shown in element ③,

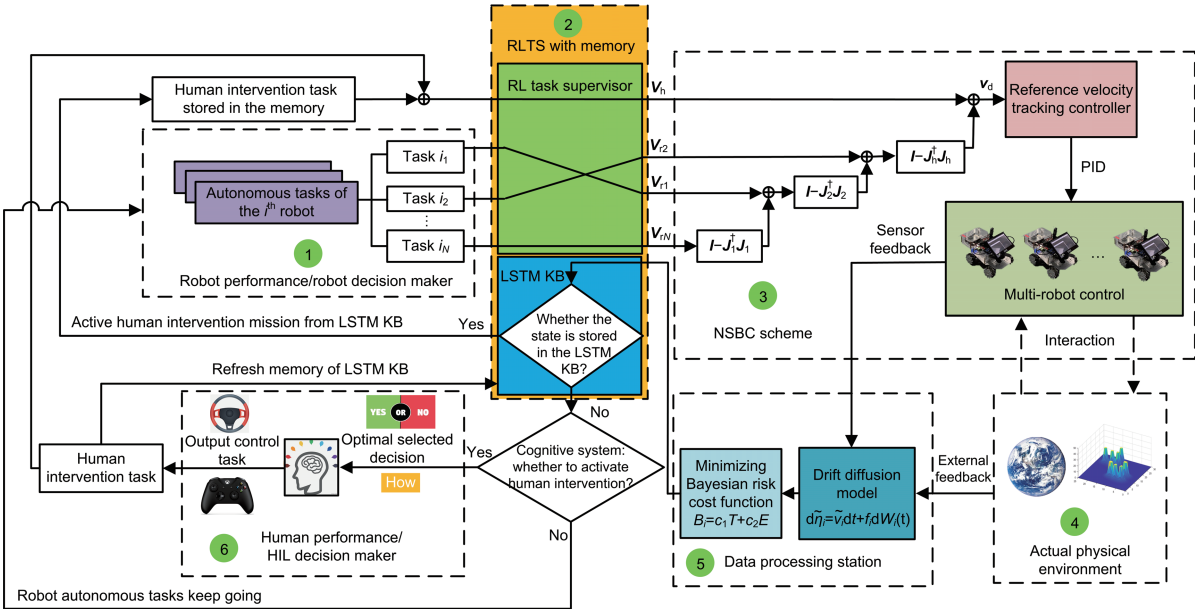

Fig. 1 The novel human–multi-robot coordination system (HMRCS) under the null-space-based behavioral control (NSBC) based framework (KB: knowledge base)

this element is responsible for controlling implementations. The desired speed control signal is obtained by fusing elementary task outputs based on the task priority from the RLTS. Then, a proportion integration differentiation (PID) controller embedded in the robots tracks the desired speed.

4. The actual physical environment. As shown in element ④, this element refers to the real environment including obstacles, terrains, and robots. Robots transform their states through actions or behavior execution, observe the environment through sensor feedback, and make their decisions for the next action.

5. Data processing station based on DDM. As shown in element ⑤, this element consists of an information collector, a filter, and a DDM (Bogacz et al., 2006). DDM is introduced to model the accuracy and reaction time of the value-based human choice in the intervention process. During task execution, the decision information is collected in real time, which helps determine the starting and ending time of human intervention. Once the accumulated information reaches a decision threshold, human intervention is activated. The decision threshold is determined by minimizing the Bayes risk brought by human intervention (Huang et al., 2020).

6. HIL decision maker. As shown in element ⑥, this element is responsible for supervising and taking over the multi-robot system (MRS) when necessary, which would generate human control input into the robots. The robot autonomous decision maker would take control again after eliminating the risk or failure. In this study, two human tasks are considered, including the monitoring task and the human intervention task. Other human behaviors such as planning and recording are omitted, but they can also be taken into account in the proposed framework.

3.3 Design of RLTS with memory

Based on the NSBC task supervisor, the design of our RLTS with memory is discussed in detail in this subsection. The proposed RLTS includes a reinforcement learning task allocation supervisor and an LSTM knowledge base. The supervisor is responsible mainly for dealing with task conflicts in the HMRCS task execution process, realizing effective human–robot coordination, and adjusting task priorities in real time. The LSTM knowledge base achieves mainly the memory and storage, and improves the independent decision-making ability and intelligence of the HMRCS.

3.3.1 Design of RLTS

In the RLTS with memory, the optimal mapping relationship between system states and composite tasks is obtained by off-line training. This ensures that each robot selects its optimal task priority order. To enhance the adaptability of the HMRCS in unknown environments, an artificial potential field model is introduced to help select states in reinforcement learning, to increase the accuracy of action or behavior selection, and to accelerate the training convergence. In this subsection, a DQN with a dueling structure is employed to accelerate the convergence of the neural network. The pseudo code of the proposed supervisor is given in Algorithm 1. Define E as a static environment, S_i as the set of states of the i^{th} robot, B as the set of behaviors, D as the experience replay buffer with capacity N_D, M as the total number of training episodes, and T_{step} as the time step of one episode.

The proposed supervisor satisfies the Markov

Algorithm 1 Reinforcement learning task supervisor (RLTS) for the i^{th} robot

Input: total number of training episodes M, ε-greedy policy decay coefficient γ_ε, time step T_{step} of an episode, and the target network update step length N_{C}
Initialize replay buffer D_i with capacity N_D
Initialize the initial value of ε-greedy policy as ε_0
Initialize action-value function $q_i(s_i, b_i; W_{i,q}, W_{i,\alpha}, W_{i,\beta}) = V_i(s_i; W_{i,q}, W_{i,\beta}) + A_{i,b}(s_i, b_i; W_{i,q}, W_{i,\alpha})$ with random initial weights $W_{i,q}, W_{i,\alpha}$, and $W_{i,\beta}$
for episode = 1:M **do**
  Initialize initial state s_0
  for t = 1:T_{step} **do**
    Select a random behavior $b_{i,t}$ with probability ε; otherwise, select $b_{i,t} = \underset{b}{\text{argmax}}\ q(s_{i,t}, b_{i,t}; W_{i,q}, W_{i,\alpha}, W_{i,\beta})$
    Execute behavior $b_{i,t}$ and observe reward $r_{i,t}$ and next state $s_{i,t+1}$
    Store this transition $(s_{i,t}, b_{i,t}, r_{i,t}, s_{i,t+1})$ in D_i
    Sample mini-batch n_m of transitions $(s_{i,t}, b_{i,t}, r_{i,t}, s_{i,t+1})$ from D_i with priority
    $y_{i,z} = r_{i,z} + \max\limits_{b_{i,z+1}} \hat{q}_i(s_{i,z+1}, b_{i,z+1}; W^-_{i,q}, W^-_{i,\alpha}, W^-_{i,\beta})$
    Perform a gradient descent step on $(y_{i,z} - q_i(s_{i,z}, b_{i,z}; W_{i,q}, W_{i,\alpha}, W_{i,\beta}))^2$ with respect to the network parameters $W_{i,q}, W_{i,\alpha}$, and $W_{i,\beta}$
    Set $s_{i,t+1} = s_{i,t}$
    Reset $\hat{q}_i = q_i$ at every N_{C} steps
  end for
end for

property, which means that robots interact with environment E at time t with state s_t, select a behavior b_t according to the ε-greedy policy, and obtain a reward r_t, and then the robots are transferred from the current state s_t to the next state s_{t+1}. The ε-greedy policy means that the robots behave with the maximum Q value with a probability $1-\varepsilon$ and randomly select a behavior with probability ε. In addition, the transition quadruple tuple (s_t, b_t, r_t, s_{t+1}) is stored in the experience replay buffer D. Then, at time $t+1$, the training continues until M episodes are finished. To make DQN explore optimistically in the early stage of the learning process and keep stable in the later stage, a high initial value of ε-greedy policy ε_0 and an appropriate decay coefficient γ_ε are set in RLTS.

In RLTS, the q-function is separated into two components: a value function $V(s; W_q, W_\beta)$ and an advantage function $A_b(s, b; W_q, W_\alpha)$, where $V(s; W_q, W_\beta)$ is estimated using a value function network with parameter W_β and $A_b(s, b; W_q, W_\alpha)$ is estimated using a state-dependent behavior advantage function network with parameter W_α. Then, the q-function is expressed by $q(s, b; W_q, W_\alpha, W_\beta) = V(s; W_q, W_\beta) + A_b(s, b; W_q, W_\alpha)$. After off-line training, RLTS can guide task priority selection on-line during task execution.

In addition, an artificial potential field value corresponding to a specific position of the robot is chosen as one of the states in RLTS. The states of reinforcement learning are selected as a joint matrix, which can reflect the position, potential field, and the distance from each obstacle to the robot, designed as

$$\boldsymbol{O}_i(t) = [\boldsymbol{P}_i(t), \boldsymbol{V}_i(t), \boldsymbol{D}_{i,\mathrm{o}}(t)], \tag{9}$$

where $\boldsymbol{O}_i(t)$ is the observation, and $\boldsymbol{P}_i(t)$, $\boldsymbol{V}_i(t)$, and $\boldsymbol{D}_{i,\mathrm{o}}(t)$ are the vectors of position, potential field, and distance from each obstacle to the i^{th} robot, respectively. To simplify the model, only the repulsive force field is considered. Thus, the potential field function of the j^{th} obstacle ($j = 1, 2, \ldots, \varphi$, and φ is the number of obstacles detected by the robot sensor) for the i^{th} robot can be defined as

$$U_{i,j} = \begin{cases} \frac{1}{2}\lambda\left(\frac{1}{d(\boldsymbol{P}_i, \boldsymbol{P}_{j,\mathrm{o}})} - \frac{1}{d_{j,\mathrm{o}}}\right)^2, & d(\boldsymbol{P}_i, \boldsymbol{P}_{j,\mathrm{o}}) \le d_{j,\mathrm{o}}, \\ 0, & d(\boldsymbol{P}_i, \boldsymbol{P}_{j,\mathrm{o}}) > d_{j,\mathrm{o}}, \end{cases} \tag{10}$$

where λ is a positive-definite constant gain, $d(\boldsymbol{P}_i, \boldsymbol{P}_{j,\mathrm{o}})$ represents the distance between the i^{th} robot and the j^{th} detected obstacle, and $d_{j,\mathrm{o}}$ denotes the influence radius of the j^{th} obstacle. The potential field of the HMRCS environment is shown in Fig. 2.

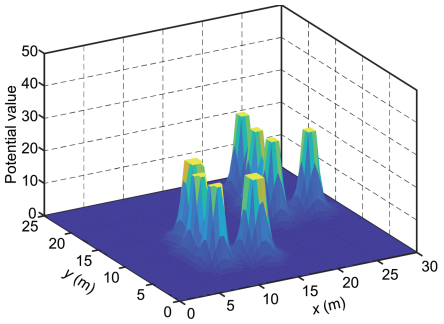

Fig. 2 Potential field of a specific environment

3.3.2 Design of the LSTM knowledge base

LSTM has been successfully applied to large-scale book retrieval knowledge bases (Zhou et al., 2018) and complex semantic recognition systems (Graves and Schmidhuber, 2005). In our application scenario, the LSTM knowledge base is designed to memorize human intervention information in the HMRCS, which includes the forget stage, selective memory stage, and output stage. The network can be realized as

$$\boldsymbol{\phi}_t = \sigma\left(\boldsymbol{\theta}_{\boldsymbol{l},\boldsymbol{\phi}}\boldsymbol{l}_t + \boldsymbol{\theta}_{\boldsymbol{h},\boldsymbol{\phi}}\boldsymbol{h}_{t-1} + \boldsymbol{\theta}_{\boldsymbol{c},\boldsymbol{\phi}}\boldsymbol{c}_{t-1} + \boldsymbol{b}_{\boldsymbol{\phi}}\right), \tag{11}$$

$$\boldsymbol{f}_t = \sigma\left(\boldsymbol{\theta}_{\boldsymbol{l},\boldsymbol{f}}\boldsymbol{l}_t + \boldsymbol{\theta}_{\boldsymbol{h},\boldsymbol{f}}\boldsymbol{h}_{t-1} + \boldsymbol{\theta}_{\boldsymbol{c},\boldsymbol{f}}\boldsymbol{c}_{t-1} + \boldsymbol{b}_{\boldsymbol{f}}\right), \tag{12}$$

$$\boldsymbol{c}_t = \boldsymbol{f}_t\boldsymbol{c}_{t-1} + \boldsymbol{\phi}_t \tanh\left(\boldsymbol{\theta}_{\boldsymbol{l},\boldsymbol{c}}\boldsymbol{l}_t + \boldsymbol{\theta}_{\boldsymbol{h},\boldsymbol{c}}\boldsymbol{h}_{t-1} + \boldsymbol{b}_{\boldsymbol{c}}\right), \tag{13}$$

$$\varkappa_t = \sigma\left(\boldsymbol{\theta}_{\boldsymbol{l},\varkappa}\boldsymbol{l}_t + \boldsymbol{\theta}_{\boldsymbol{h},\varkappa}\boldsymbol{h}_{t-1} + \boldsymbol{\theta}_{\boldsymbol{c},\varkappa}\boldsymbol{c}_t + \boldsymbol{b}_{\varkappa}\right), \tag{14}$$

$$\boldsymbol{h}_t = \varkappa_t \tanh\left(\boldsymbol{c}_t\right), \tag{15}$$

where $\boldsymbol{\phi}$, $\boldsymbol{f}$, $\varkappa$, and $\boldsymbol{c}$ represent the input gate, forget gate, output gate, and cell vectors respectively, $\boldsymbol{h}$ is the hidden vector, $\boldsymbol{l}$ is the data vector, $\sigma(\cdot)$ represents a logistic sigmoid function, $\tanh(\cdot)$ is a hyperbolic tangent function, and $\boldsymbol{\theta}$ represents the weight parameter matrix from a module to another. The LSTM knowledge base in the HMRCS is able to learn human intervention control information and experience history. In our HMRCS application scenario, the data can be defined as a time series array of five dimensions, which consist of the position, velocity, and potential field value of the i^{th} robot controlled by a person. The knowledge base needs to label processing history data according to different situations.

Therefore, the array can be expressed as

$$\boldsymbol{\Gamma} = (\boldsymbol{P}_{i,x}, \boldsymbol{P}_{i,y}, \boldsymbol{V}_{i,x}, \boldsymbol{V}_{i,y}, \boldsymbol{U}_i)^{\mathrm{T}}, \tag{16}$$

where $\boldsymbol{P}_{i,x}$, $\boldsymbol{P}_{i,y}$, $\boldsymbol{V}_{i,x}$, and $\boldsymbol{V}_{i,y}$ are vectors concerning the position and velocity in the x and y directions of the i^{th} robot, respectively, and $\boldsymbol{U}_i$ denotes a vector concerning the potential field value.

4 Simulation results

In this section, three mobile robots running on a flat plane are considered. The control goal of the simulation for the robots is to maintain formation and reach their target points in an unknown environment, without colliding with obstacles or the other robots. The parameters of the RLTS, environment configuration, and LSTM knowledge base are given in Table 1.

4.1 Task design

Three elementary tasks are briefly introduced: motion, obstacle avoidance, and human intervention tasks. The motion task is to control robots in approaching their target positions along a predetermined trajectory or according to robots' autonomous motion requirements. For mobile ground robots, based on the behavior design guidelines of the NSBC framework, the corresponding task function of the motion task is defined as

$$\boldsymbol{\rho}_{i,\mathrm{m}} = f_{\mathrm{m}}(\boldsymbol{p}_i), \quad \boldsymbol{\rho}_{i,\mathrm{m}} \in \mathbb{R}^{2\times 1}, \tag{17}$$

where $\boldsymbol{p}_i \in \mathbb{R}^{2\times 1}$ is the position of the i^{th} robot. The reference velocity of the motion task can be calculated as

$$\boldsymbol{v}_{i,\mathrm{m}} = \boldsymbol{J}_{i,\mathrm{m}}^{\dagger}(\dot{\boldsymbol{\rho}}_{i,\mathrm{md}} + \Lambda_{\mathrm{m}}\tilde{\boldsymbol{\rho}}_{i,\mathrm{m}}), \tag{18}$$

where $\boldsymbol{J}_{i,\mathrm{m}}^{\dagger}$ is the pseudo-inverse of $\boldsymbol{J}_{i,\mathrm{m}}$, $\boldsymbol{J}_{i,\mathrm{m}} = \boldsymbol{I}_2 \in \mathbb{R}^{2\times 2}$ denotes the Jacobian matrix of the motion behavior of the i^{th} robot, $\boldsymbol{\rho}_{i,\mathrm{md}}$ is the desired task function, Λ_{m} is the gain of the motion behavior, and $\tilde{\boldsymbol{\rho}}_{i,\mathrm{m}} = \boldsymbol{\rho}_{i,\mathrm{md}} - \boldsymbol{\rho}_{i,\mathrm{m}}$ is the task error of the motion behavior.

Then, the obstacle avoidance task drives the robots to avoid obstacles along their planned path. The task function is designed to keep a safe distance between the robots and the nearest obstacle, which is defined as

$$\rho_{i,\mathrm{o}} = \|\boldsymbol{p}_i - \boldsymbol{p}_{\mathrm{o}}\|, \rho_{i,\mathrm{o}} \in \mathbb{R}, \tag{19}$$

Table 1 Parameter values in the simulation

Parameter	Value
DDM threshold, ξ	4
Obstacle avoidance task gain, Λ_{o}	5
Motion task gain, Λ_{m}	0.4
Safe distance, d_{safe} (m)	1.8
Initial positions, $\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3$ (m)	(1, 8), (5, 6), (4, 1)
Target positions, $\boldsymbol{p}_4, \boldsymbol{p}_5, \boldsymbol{p}_6$ (m)	(27, 23), (31, 21), (30, 16)
Positions of obstacles	
$\boldsymbol{p}_{\mathrm{o}1}, \boldsymbol{p}_{\mathrm{o}2}$ (m)	(10.2, 12.3), (22, 20.5)
$\boldsymbol{p}_{\mathrm{o}3}, \boldsymbol{p}_{\mathrm{o}4}$ (m)	(8.8, 6.2), (21, 13.5)
$\boldsymbol{p}_{\mathrm{o}5}, \boldsymbol{p}_{\mathrm{o}6}$ (m)	(13.8, 6.4), (26, 14)
Positions of the newly detected obstacles, $\boldsymbol{p}_{\mathrm{o}7}$, $\boldsymbol{p}_{\mathrm{o}8}$ (m)	(8.8, 9.2), (21, 16.5)
Motion task functions	
$\boldsymbol{\rho}_{1,\mathrm{m}}$ (m)	(2+0.25t, 8+0.15t)
$\boldsymbol{\rho}_{2,\mathrm{m}}$ (m)	(6+0.25t, 6+0.15t)
$\boldsymbol{\rho}_{3,\mathrm{m}}$ (m)	(5+0.25t, 1+0.15t)
Sampling frequency, F (Hz)	20
Positive constant in the reward function, k_{r}	0.4
Total number of DQN training episodes, M	40 000
Target network update step length, N_{C}	50
Capacity of experience replay buffer, N_D	5000
Initial ε-greedy parameter, ε_0	0.8
ε-greedy decay coefficient, γ_ε	0.999 85
Numbers of hidden layers in the Q-network, h_1, h_2	128, 128
Number of output layers in the Q-network, h_3	3
Learning rate of LSTM, Ω	0.001
Sequence input of LSTM, l_{s}	6
LSTM learning rate drop period, T_{dp}	125
Learning rate drop factor of LSTM, F_{df}	0.2
Number of hidden units in the bidirectional LSTM, h_{b}	100

where $\boldsymbol{p}_{\mathrm{o}} \in \mathbb{R}^{2\times 1}$ is the position of the obstacle. The reference velocity of the obstacle avoidance behavior can be calculated as

$$\boldsymbol{v}_{i,\mathrm{o}} = \boldsymbol{J}_{i,\mathrm{o}}^{\dagger}\Lambda_{\mathrm{o}}\left(d_{\mathrm{safe}} - \|\boldsymbol{p}_i - \boldsymbol{p}_{\mathrm{o}}\|\right), \tag{20}$$

where $\boldsymbol{J}_{i,\mathrm{o}}^{\dagger}$ is the pseudo-inverse of $\boldsymbol{J}_{i,\mathrm{o}}$ of the i^{th} robot, $\boldsymbol{J}_{i,\mathrm{o}} = \frac{(\boldsymbol{p}_i - \boldsymbol{p}_{\mathrm{o}})^{\mathrm{T}}}{\|\boldsymbol{p}_i - \boldsymbol{p}_{\mathrm{o}}\|}$, Λ_{o} is the gain of the obstacle avoidance behavior, $d_{\mathrm{safe}} - \|\boldsymbol{p}_i - \boldsymbol{p}_{\mathrm{o}}\|$ is the task error of the obstacle avoidance behavior, and d_{safe} represents the safe distance between the robots and the detected obstacles.

The human intervention task is to describe the control of humans in the HMRCS in the NSBC framework. Human intervention is activated once the accumulated information reaches a decision

threshold, indicating that robots in the HMRCS might encounter problems or emergencies such as a local minimum and partial failure. The task output can be a time-dependent position or a speed signal. For mobile ground robots, the human intervention task is defined as

$$\boldsymbol{\rho}_{i,\mathrm{h}} = f_{\mathrm{h}}(\boldsymbol{p}_i), \quad \boldsymbol{\rho}_{i,\mathrm{h}} \in \mathbb{R}^{2\times 1}. \tag{21}$$

The reference velocity of the human intervention task can be calculated as

$$\boldsymbol{v}_{i,\mathrm{h}} = \boldsymbol{J}_{i,\mathrm{h}}^{\dagger}(\dot{\boldsymbol{\rho}}_{i,\mathrm{hd}} + \Lambda_{\mathrm{h}}\tilde{\boldsymbol{\rho}}_{i,\mathrm{h}}), \tag{22}$$

where $\boldsymbol{J}_{i,\mathrm{h}}^{\dagger}$ is the pseudo-inverse of $\boldsymbol{J}_{i,\mathrm{h}}$, $\boldsymbol{J}_{i,\mathrm{h}} \in \mathbb{R}^{2\times 2}$ denotes the Jacobian matrix of the human intervention task in the i^{th} robot, $\boldsymbol{\rho}_{i,\mathrm{hd}}$ is the desired human intervention function, Λ_{h} is the human intervention task gain, and $\tilde{\boldsymbol{\rho}}_{i,\mathrm{h}} = \boldsymbol{\rho}_{i,\mathrm{hd}} - \boldsymbol{\rho}_{i,\mathrm{h}}$ is the human intervention task error.

Finally, elementary tasks are fused according to Eq. (7), based on the task priority provided by RLTS.

4.2 Reward function in RLTS

The task tracking error of the i^{th} robot is defined as

$$\boldsymbol{e}_i(t) = \boldsymbol{\rho}_{i,\mathrm{md}} - \boldsymbol{p}_i(t). \tag{23}$$

The reward function $r_i(t)$ in the supervisor is designed as the sum of two parts:

$$r_i(t) = r_{i,1}(t) + r_{i,2}(t), \tag{24}$$

$$r_{i,1}(t) = -15\tanh(\boldsymbol{e}_i^{\mathrm{T}}(t)\boldsymbol{e}_i(t)\Delta t - k_{\mathrm{r}}), \tag{25}$$

$$r_{i,2}(t) = \begin{cases} -12, & \|\boldsymbol{p}_i - \boldsymbol{p}_{\mathrm{o}}\| < d_{\mathrm{safe}}, \\ 0, & \|\boldsymbol{p}_i - \boldsymbol{p}_{\mathrm{o}}\| \geq d_{\mathrm{safe}}, \end{cases} \tag{26}$$

where Δt denotes the sampling interval, and $k_{\mathrm{r}} \in (0, 0.5]$ is a small positive constant, which is used to improve the convergence of the neural network in the initial training phase.

4.3 Comparative analysis

In this subsection, a comparative simulation is conducted between the FSA-based task supervisor and our proposed RLTS with memory in the HMRCS framework. The important parameters used to realize our simulation are shown in Table 1, including the task and gain parameters of NSBC, configuration of robots and environmental obstacles, reinforcement learning parameters, and knowledge base parameters. In the simulation, three mobile robots move according to the preset task function trajectories.

Figs. 3 and 4 show the moving trajectories of the robots using the FSA-based task supervisor and our proposed RLTS with memory, respectively. When encountering obstacles on the way, the robotic system supervisor will assign task priorities and composite NSBC behaviors. Robots in the HMRCS sample their status information and collect real-time drift diffusion decision information during the task execution process. The real-time drift diffusion decision information collected using the FSA-based task supervisor and our proposed RLTS with memory is shown in Figs. 5 and 6, respectively. The decision information is also defined as the task tracking error. When robots detect new obstacles or fall into local minima, the human intervention task is triggered after the decision information reaches the decision threshold. During human intervention, human direct control input can help robots cross the dangerous area smoothly, and the control authority is handed over to the robot controller after they are far enough away from the obstacle or other hazardous areas.

4.3.1 FSA-based task supervisor analysis

Fig. 7 shows the distance between the robots and the detected obstacles, and Fig. 8 illustrates the switching mode of composite tasks using the FSA-based task supervisor. After analysis, human intervention would take over the control process when

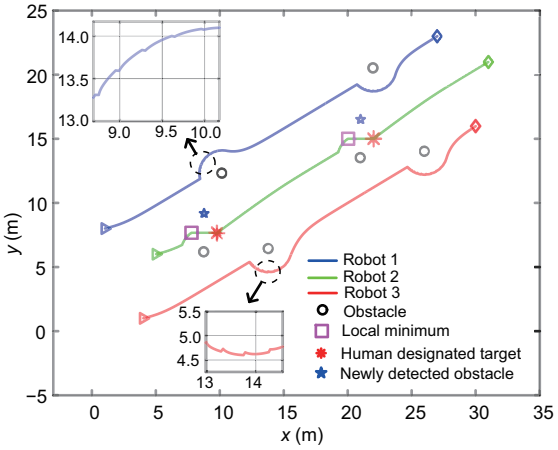

Fig. 3 Trajectories of the robots using the FSA-based task supervisor with two human intervention processes in the HMRCS

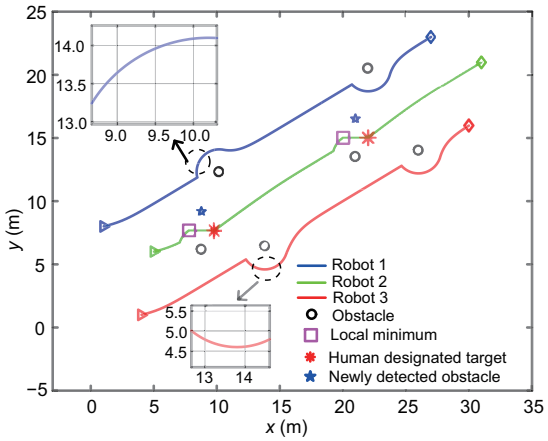

Fig. 4 Trajectories of the robots using the proposed RLTS with memory with one human intervention process and one reloading intervention process in the HMRCS

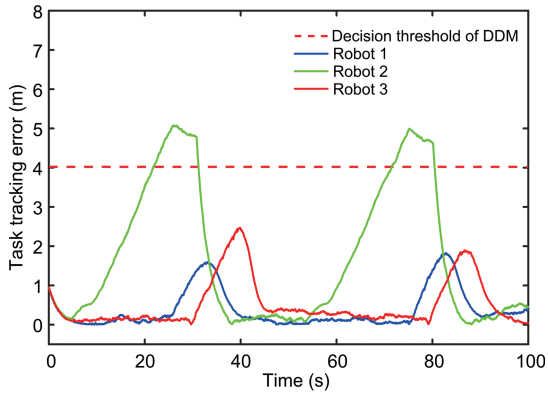

Fig. 5 Task tracking error of the robots using the FSA-based task supervisor with two human intervention processes in the HMRCS

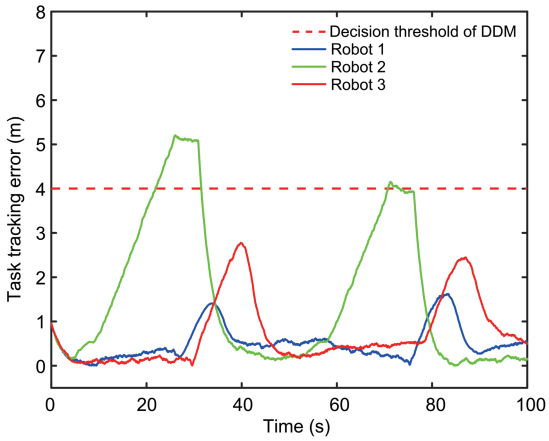

Fig. 6 Task tracking error of the robots using the proposed RLTS with memory with one human intervention process and one reloading intervention process in the HMRCS

robots encounter an emergency and reach the decision threshold with the FSA-based task supervisor.

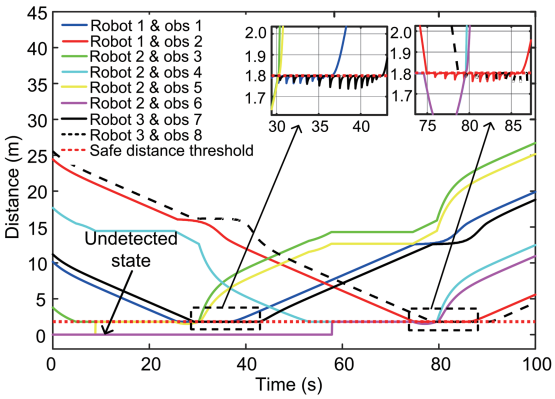

Fig. 7 Distance between the robots and the detected obstacles using the FSA-based task supervisor with two human intervention processes in the HMRCS

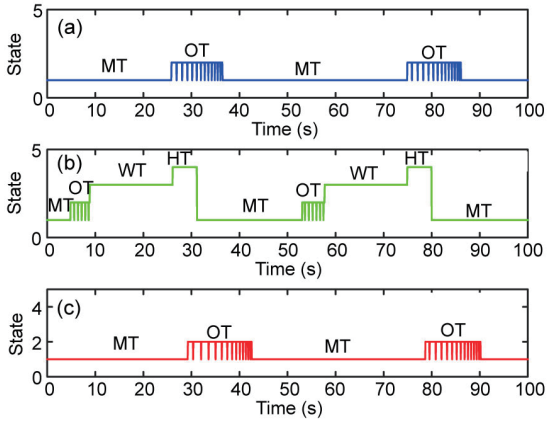

Fig. 8 Robot task mode using the FSA-based task supervisor in the HMRCS with two human intervention processes: (a) robot 1; (b) robot 2; (c) robot 3

MT: motion task; OT: obstacle avoidance task; WT: waiting for human intervention task; HT: human intervention task

In addition, the FSA-based supervisor lacks the ability to dynamically adjust the task priority, which means that the robots will constantly create safety concerns and speed mutation. When this occurs, a person takes over twice the FSA-based supervisor.

4.3.2 RLTS with memory analysis

Fig. 9 shows the distance between the robots and the detected obstacles, and Fig. 10 illustrates the switching mode of composite tasks using our proposed RLTS with memory. After analysis, human intervention would take over the control process when robots encounter the first local minimum and reach the decision threshold with the proposed RLTS with memory. The LSTM knowledge base updates the human controlled information and its training of the neural network simultaneously, and accurately and

efficiently reloads the control information when facing a similar situation. In addition, the HMRCS realizes an optimal behavioral priority adjustment strategy for human and robot tasks, making the moving trajectories of robots smoother and more accurate without speed mutation. Human intervention takes over once in the simulation under the RLTS with memory.

4.3.3 Performance comparison

After comparing simulation results of the two methods, their performances are summarized in Table 2. The moving trajectories of the robots when implementing the proposed RLTS with memory are smoother and more accurate, without bursting into the dangerous collision area, which demonstrates that the HMRCS realizes an optimal behavioral priority adjustment strategy for human and robot tasks. After integrating learning and memory abilities into the design of the supervisor, the HMRCS can memorize human intervention history when robot systems encounter emergencies and are not confident in decision making. Then the proposed RLTS with memory reloads the history information while encountering the same situation when tackled by humans. The frequency of repeated human intervention will be greatly reduced, and the intelligence of the HMRCS is considerably improved. At the end of the comparative analysis, the training loss curves of the RLTS with memory during 200 and 40 000 epochs are shown in Figs. 11 and 12, respectively. These training curves demonstrate that the proposed RLTS with memory has good convergence performance and practicability.

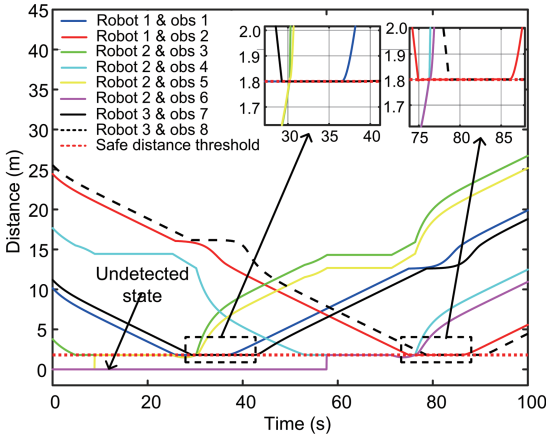

Fig. 9 Distance between the robots and the detected obstacles using the proposed RLTS with memory with one human intervention process and one reloading intervention process in the HMRCS

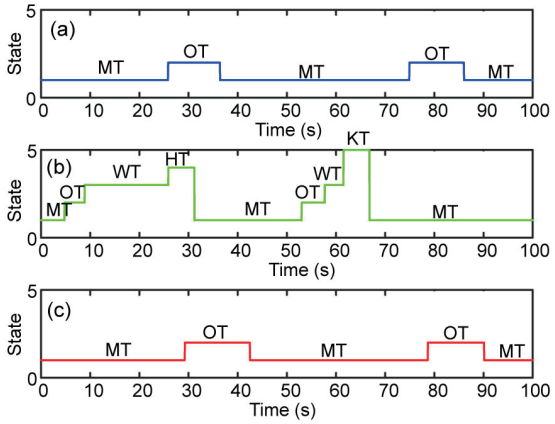

Fig. 10 Robot task mode using the proposed RLTS with memory in the HMRCS with one human intervention process and one reloading intervention process: (a) robot 1; (b) robot 2; (c) robot 3

MT: motion task; OT: obstacle avoidance task; WT: waiting for human intervention task; HT: human intervention task; KT: knowledge base reloading task

5 Experiments

In this section, we present an experiment using a group of mobile robots subject to external

Table 2 Performance comparison

Parameter	Value	
	FSA	RLTS
Frequency of task switch (Hz)	2.63	0.19
Number of task switches	28	4
Number of intervention tasks	2	1
Number of reloading tasks	0	1
Decision waiting time (s)	17.26	3.75

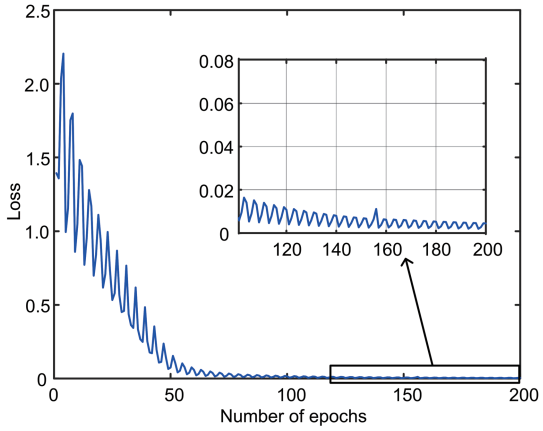

Fig. 11 LSTM network training loss of the RLTS with memory during 200 epochs (the training process is carried out after a human intervention task)

noise and disturbances, and the experiment was conducted in uncertain real-world environments. Two sets of verification were presented. The first was to verify that the RLTS can determine the optimal priority adjustment strategy to realize dynamic priority switching in real time. The second was to verify that the LSTM knowledge base can effectively reduce the frequency of human intervention. The experimental video of the proposed HMRCS is provided in the supplementary materials and can also be accessed from YouTube (https://youtu.be/gk8SRVTsp64) and Bilibili (https://www.bilibili.com/video/BV1GM4y1F7Lc).

5.1 Experimental setup

The experimental parameters of the HMRCS were set as follows: the decision threshold was 1.48 m. The obstacle avoidance task gain was 5. The motion task gain was 0.4. The safe distance was 1.8 m. The initial positions of robots 1, 2, and 3 were (0.5, 7.8), (0.8, 4.8), and (0.5, 1.8) m, respectively. The positions of obstacles were set as: $\boldsymbol{p}_{o1}$, (2.5, 7) m; $\boldsymbol{p}_{o2}$, (7.5, 9) m; $\boldsymbol{p}_{o3}$, (3.8, 6) m; $\boldsymbol{p}_{o4}$, (7.8, 6) m; $\boldsymbol{p}_{o5}$, (2.5, 3) m; $\boldsymbol{p}_{o6}$, (7.5, 1) m. The positions of the newly detected obstacles were set as: $\boldsymbol{p}_{o7}$, (4.2, 4) m; $\boldsymbol{p}_{o8}$, (8.2, 4) m. The motion task functions 1, 2, and 3 were set as (0.5+0.1t, 8), (1.5+0.1t, 5), and (0.5+0.1t, 2) m, respectively. The remaining parameters of the RLTS and LSTM were the same as those in Table 2.

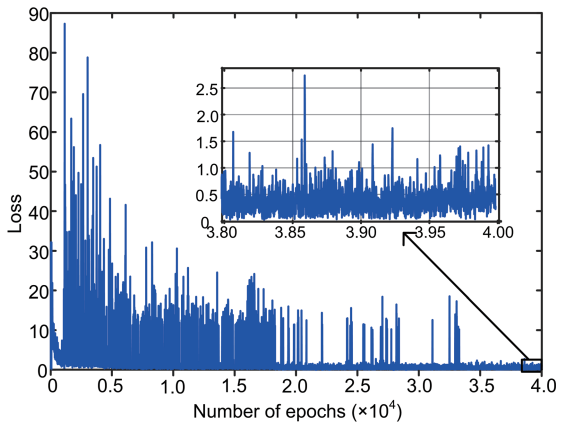

Fig. 12 Training loss of the RLTS with memory during 40 000 epochs

The experimental scheme of the HMRCS with multiple mobile robots is shown in Fig. 13. It is divided into four parts: an HIL decision-and-control center, a Linux-based computing control unit, a mobile robot chassis driving module, and an ultra-wideband (UWB) positioning system. Each robot was equipped with a Raspberry PI running on a Ubuntu system. The HIL decision-and-control center was responsible for monitoring the operation of the HMRCS and providing inputs to the human intervention task. The robots' position information and speed information were collected in real time by the UWB positioning system. The configuration of the physical experimental environment is shown in Fig. 14, including eight obstacles, three mobile robots, an HIL decision center, and four UWB positioning base stations and their terminals mounted on the robots. The configuration of a single mobile robot is shown in Fig. 15.

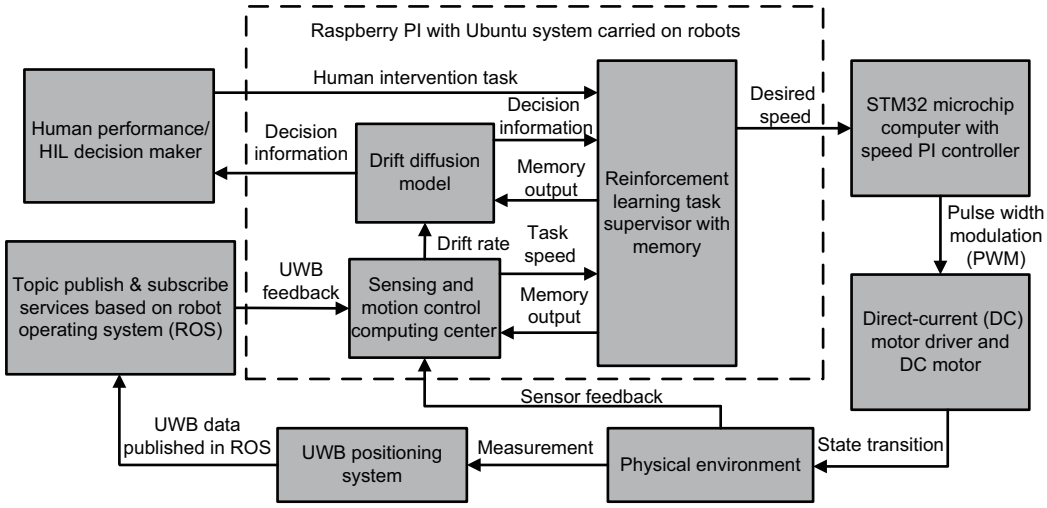

Fig. 13 Experimental scheme of the HMRCS

5.2 Experimental results

Snapshots of the experiment at 0, 36, 67, 78, 88, and 100 s are shown in Fig. 16. The task status, robot trajectories, and decision-making information during the experiment process are also presented in the snapshots. These results showed that the RLTS can dynamically adjust the behavioral priority. The supervisor can memorize human intervention history when the robots were not confident in decision making, and then reload the history information when encountering a situation that was previously tackled by a person.

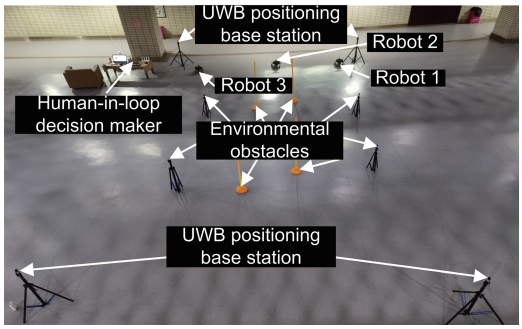

Fig. 14 Configuration of the human–multi-robot coordination experimental platform

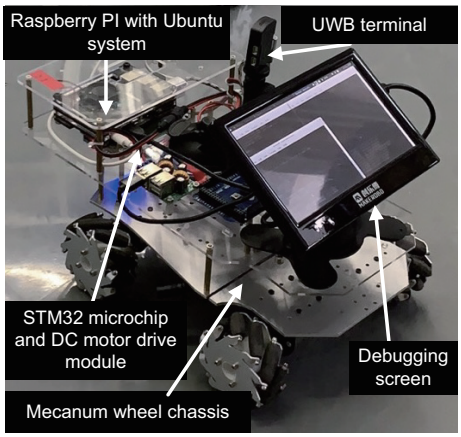

Fig. 15 Configuration of a single mobile robot in the HMRCS

6 Conclusions

In this study, we proposed an RLTS with memory by integrating DQN and LSTM knowledge base within an NSBC framework to address the problems of dynamic task priority adjustment and repeated human intervention in HMRCSs. Simulations and experiments were conducted in an uncertain real-world environment to demonstrate the effectiveness of the proposed RLTS. Results showed that RLTS can successfully memorize human intervention history and reload human control input when robots are not confident, greatly improving the robustness and flexibility of the HMRCS. Our future work will focus on multi-agent reinforcement learning in the HMRCS with more complex dynamics and more complicated environmental constraints.

Contributors

Jie HUANG and Zhibin MO designed the research. Zhibin MO and Zhenyi ZHANG processed the data and drafted the paper. Jie HUANG and Yutao CHEN helped

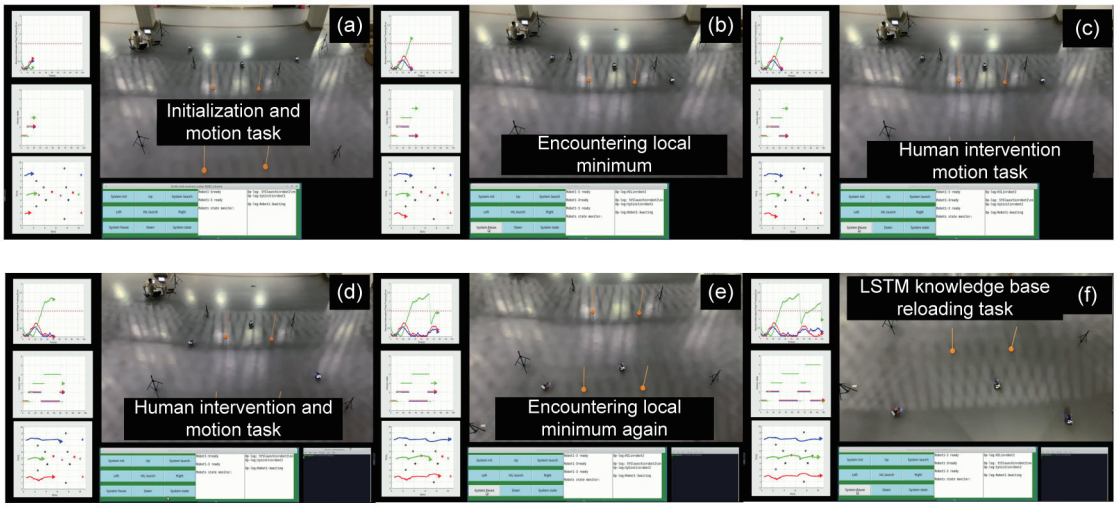

Fig. 16 Snapshots of the HMRCS experiment at 0 s (a), 36 s (b), 67 s (c), 78 s (d), 88 s (e), and 100 s (f)

organize the paper. Jie HUANG, Zhibin MO, and Yutao CHEN revised and finalized the paper.

Compliance with ethics guidelines

Jie HUANG, Zhibin MO, Zhenyi ZHANG, and Yutao CHEN declare that they have no conflict of interest.

References

Antonelli G, Chiaverini S, 2006. Kinematic control of platoons of autonomous vehicles. *IEEE Trans Rob*, 22(6):1285-1292. https://doi.org/10.1109/TRO.2006.886272

Aviv Y, Pazgal A, 2005. A partially observed Markov decision process for dynamic pricing. *Manag Sci*, 51(9):1400-1416. https://doi.org/10.1287/mnsc.1050.0393

Baizid K, Giglio G, Pierri F, et al., 2015. Experiments on behavioral coordinated control of an unmanned aerial vehicle manipulator system. IEEE Int Conf on Robotics and Automation, p.4680-4685. https://doi.org/10.1109/ICRA.2015.7139848

Baizid K, Giglio G, Pierri F, et al., 2017. Behavioral control of unmanned aerial vehicle manipulator systems. *Auton Robot*, 41(5):1203-1220. https://doi.org/10.1007/s10514-016-9590-0

Bajcsy A, Herbert SL, Fridovich-Keil D, et al., 2019. A scalable framework for real-time multi-robot, multi-human collision avoidance. Int Conf on Robotics and Automation, p.936-943. https://doi.org/10.1109/ICRA.2019.8794457

Bluethmann W, Ambrose R, Diftler M, et al., 2003. Robonaut: a robot designed to work with humans in space. *Auton Robot*, 14(2):179-197. https://doi.org/10.1023/A:1022231703061

Bogacz R, Brown E, Moehlis J, et al., 2006. The physics of optimal decision making: a formal analysis of models of performance in two-alternative forced-choice tasks. *Psychol Rev*, 113(4):700-765. https://doi.org/10.1037/0033-295X.113.4.700

Chen YT, Zhang ZY, Huang J, 2020. Dynamic task priority planning for null-space behavioral control of multi-agent systems. *IEEE Access*, 8:149643-149651. https://doi.org/10.1109/ACCESS.2020.3016347

Fu HJ, Chen SC, Lin YL, et al., 2019. Research and validation of human-in-the-loop hybrid-augmented intelligence in Sawyer. *Chin J Intell Sci Technol*, 1(3):280-286 (in Chinese). https://doi.org/10.11959/j.issn.2096-6652.201933

Gans NR, Rogers JGIII, 2021. Cooperative multirobot systems for military applications. *Curr Robot Rep*, 2(1):105-111. https://doi.org/10.1007/s43154-020-00039-w

Graves A, Schmidhuber J, 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neur Netw*, 18(5-6):602-610. https://doi.org/10.1016/j.neunet.2005.06.042

Honig S, Oron-Gilad T, 2018. Understanding and resolving failures in human-robot interaction: literature review and model development. *Front Psychol*, 9:861. https://doi.org/10.3389/fpsyg.2018.00861

Huang J, Zhou N, Cao M, 2019. Adaptive fuzzy behavioral control of second-order autonomous agents with prioritized missions: theory and experiments. *IEEE Trans Ind Electron*, 66(12):9612-9622. https://doi.org/10.1109/TIE.2019.2892669

Huang J, Wu WH, Zhang ZY, et al., 2020. A human decision-making behavior model for human-robot interaction in multi-robot systems. *IEEE Access*, 8:197853-197862. https://doi.org/10.1109/ACCESS.2020.3035348

Lee WH, Kim JH, 2018. Hierarchical emotional episodic memory for social human robot collaboration. *Auton Robot*, 42(5):1087-1102. https://doi.org/10.1007/s10514-017-9679-0

Lippi M, Marino A, 2018. Safety in human-multi robot collaborative scenarios: a trajectory scaling approach. *IFAC-PapersOnLine*, 51(22):190-196. https://doi.org/10.1016/j.ifacol.2018.11.540

Lippi M, Marino A, Chiaverini S, 2019. A distributed approach to human multi-robot physical interaction. IEEE Int Conf on Systems, Man and Cybernetics, p.728-734. https://doi.org/10.1109/SMC.2019.8914468

Mnih V, Kavukcuoglu K, Silver D, et al., 2015. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529-533. https://doi.org/10.1038/nature14236

Mo ZB, Zhang ZY, Chen YT, et al., 2022. A reinforcement learning mission supervisor with memory for human-multi-robot coordination systems. Proc Chinese Intelligent Systems Conf, p.708-716. https://doi.org/10.1007/978-981-16-6320-8_72

Moreno L, Moraleda E, Salichs MA, et al., 1993. Fuzzy supervisor for behavioral control of autonomous systems. Proc 19th Annual Conf of IEEE Industrial Electronics, p.258-261. https://doi.org/10.1109/IECON.1993.339071

Queralta JP, Taipalmaa J, Pullinen BC, et al., 2020. Collaborative multi-robot search and rescue: planning, coordination, perception, and active vision. *IEEE Access*, 8:191617-191643. https://doi.org/10.1109/ACCESS.2020.3030190

Robla-Gómez S, Becerra VM, Llata JR, et al., 2017. Working together: a review on safe human-robot collaboration in industrial environments. *IEEE Access*, 5:26754-26773. https://doi.org/10.1109/ACCESS.2017.2773127

Rosenfeld A, Agmon N, Maksimov O, et al., 2017. Intelligent agent supporting human–multi-robot team collaboration. *Artif Intell*, 252:211-231. https://doi.org/10.1016/j.artint.2017.08.005

Wang HN, Liu N, Zhang YY, et al., 2020. Deep reinforcement learning: a survey. *Front Inform Technol Electron Eng*, 21(12):1726-1744. https://doi.org/10.1631/FITEE.1900533

Watkins CJCH, Dayan P, 1992. *Q*-learning. *Mach Learn*, 8(3-4):279-292. https://doi.org/10.1007/BF00992698

Zhang KQ, Yang ZR, Başar T, 2021. Decentralized multi-agent reinforcement learning with networked agents: recent advances. *Front Inform Technol Electron Eng*, 22(6):802-814. https://doi.org/10.1631/FITEE.1900661

Zheng NN, Liu ZY, Ren PJ, et al., 2017. Hybrid-augmented intelligence: collaboration and cognition. *Front Inform Technol Electron Eng*, 18(2):153-179. https://doi.org/10.1631/FITEE.1700053

Zhou BT, Sun CJ, Lin L, et al., 2018. LSTM based question answering for large scale knowledge base. *Acta Sci Nat Univ Pek*, 54(2):286-292 (in Chinese). https://doi.org/10.13209/j.0479-8023.2017.155

List of supplementary materials

Video S1 Behavioral control based on reinforcement learning for human–multi-robot coordination systems