信息与控制　xxxx 年　第 x 卷　第 x 期：1 ~ 11

DOI：10.13976/j.cnki.xk.2025.1833　CSTR: 32166.14.xk.2025.1833

面向多智能体合作的估计－控制－调度协同设计

李　鹏[1,2]，张吟龙[1,2]，梁　炜[1,2]，郑　萌[1,2]

1. 中国科学院沈阳自动化研究所机器人与智能系统国家重点实验室，辽宁 沈阳　110016；
2. 中国科学院大学，北京　100049

基金项目：国家自然科学基金项目(62273332)；中国科学院青年创新促进会会员项目(2022201)；广东省基础与应用基础研究基金项目(2023A1515011363)；辽宁省自然科学基金项目(2023JH26/10300028，2024JH3/10200029，2023-MSLH-219)

通信作者：张吟龙，zhangyinlong@sia.cn　收稿/录用/修回：2025－06－12/2025－09－26/2025－09－08

摘要

为了提高多智能体完全合作任务中无线网络化控制系统(WNCS)在资源受限环境下的控制性能，本文提出一种基于深度强化学习(DRL)的估计—控制—调度协同设计方法，通过将状态估计、控制策略和资源调度紧密结合，以优化多智能体的决策控制与资源调度。本方法采用深度强化学习策略，通过循环神经网络来学习 WNCS 中观测量和状态量的时序依赖关系，有效增强了本方法在复杂工业环境中的适应性，同时降低了对精确系统动力学模型的依赖。在 CoppeliaSim 仿真平台上进行的多智能体协作搬运实验表明，相较于现有解耦设计方法，本方法将协作搬运任务完成率提升了 3.8%，任务完成时间减少了 7.6%。

关键词

多智能体
完全合作型任务
无线网络化控制系统
协同设计
深度强化学习

中图法分类号：TP273

文献标志码：A

Estimation-Control-Scheduling Co-Design for Multi-Agent Cooperation

LI Peng[1,2], ZHANG Yinlong[1,2], LIANG Wei[1,2], ZHENG Meng[1,2]

1. *State Key Laboratory of Robotics and Intelligent Systems, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang* 110016*, China*;
2. *University of Chinese Academy of Sciences, Beijing* 100049*, China*

Abstract

To improve the control performance of wireless networked control system (WNCS) for fully cooperative multi-agent tasks in environments with limited radio resources, a Deep Reinforcement Learning (DRL)-based estimation-control-scheduling co-design method is proposed, which tightly integrates state estimation, control strategies, and resource scheduling to optimize multi-agent decision control and resource scheduling. The proposed method adopts a DRL strategy with recurrent neural networks to capture the temporal dependencies between observations and states in WNCS, thereby enhancing its adaptability in complex industrial environments while reducing reliance on accurate system dynamics models. Experimental results from multi-agent cooperative transportation tasks conducted on the CoppeliaSim simulation platform demonstrate that, compared to existing decoupled design methods, the proposed approach improves the cooperative transportation task success rate by 3.8% and reduces task completion time by 7.6%.

Keywords

multi-agent;
fully cooperative task;
wireless networked control system;
co-design;
DRL (deep reinforcement learning)

0 引言

多智能体系统是由多个智能体通过完全合作、完全竞争或竞争合作混合等方式来完成复杂任务的系统[1-2]。每个智能体都具有一定的自主性、交互能力和决策能力，能够根据自身的感知信息和环境状态作出独立的行动决策。完全合作型任务在许多工业领域中具有广泛的应用，如协同搬运、装配等。在这些任务中，多智能体系统必须协调各智能体的行动以实现共同的目标[3-4]。然而，传统的多智能体系统存在灵活性不足、安装部署成本高等问题。针对上述问题，在 WNCS 中，边缘处理器通过计算卸载、动态资源分配和资源供给等机制实现智能体的传输调度，从而优先保障对控制性能影响最大的控制回路的运行[5]，这种方式显著提升了多智能体协同的灵活性与可扩展性。尤其在复杂和动态环境下，WNCS 展现了极强的鲁棒性，在完全合作型任务中展现了广阔的应用前景。

无线网络可将智能体复杂的控制、感知等功能卸载到边缘处理器，使得生产过程更加高效、智能[6]。然而，无线网络资源受限且工业环境动态变化[7]，使得 WNCS 与多智能体系统的结合面临重大挑战，直接影响传输的可靠性和控制的实时性。因此，实现包含多个智能体的 WNCS 性能优化需对控制与通信协同设计，考虑它们之间的紧密交互[8]，以提高控制性能。

针对 WNCS 中存在的随机通信噪声、数据丢失等问题，张强[9] 提出了一种分布式自适应控制方法，并分析了相应闭环系统的鲁棒性质。GATSIS 等[10] 提出了一种基于信道感知调度和功率分配的设计方法，通过观察到的信道序列动态调整传输调度。在满足系统控制性能的前提下，最大限度地降低了总功耗。KNORN 等[11] 考虑了基于能量采集的无线闭环控制系统，通过马尔可夫模型刻画信道衰落和能量收集过程，提出分离定理下最优线性二次高斯（LQG）控制和动态规划能量分配策略。LIU 等[12] 提出了一种用于工业物联网的无编码控制方法，通过优化功率分配，实现了在慢衰落和快衰落信道下的超低延迟通信和稳定控制。HUANG 等[13] 针对工业物联网中半双工 WNCS 的传输调度优化问题，通过分析上行和下行信道的传输可靠性，提出了最小化长期平均成本函数的最优调度策略。但以上的传统方法都依赖于准确的系统动力学模型。然而，机器人与自动导引车（AGV）所面对的工业控制场景通常十分复杂，例如，在典型的协作运输和装配任务中，AGV 通过无线网络进行调度以装载和运输大型货物[14]。当 AGV 之间频繁发生通信错误时，协作任务可能会失败，这一问题限制了传统方法在非线性控制中的适用性。

相较于传统方法需要精确的系统动力学建模，基于强化学习的方法能够在没有精确建模的情况下，通过奖励机制引导智能体探索最优策略[15-16]，在设备有限的简单场景中，传统强化学习方法已展现出良好的控制效果。然而，在环境复杂性高、任务动态性强的完全合作型任务中，通常具有较大的状态空间和动作空间，强化学习方法难以有效收敛[17]。由于深度强化学习（DRL）方法能够利用深度神经网络进行函数逼近，自动提取高维数据的关键特征从而显著提升学习效率，因此有学者提出基于 DRL 的 WNCS 协同设计方法，以解决上述问题。EISEN 等[18] 提出了基于 DRL 的通信控制协同设计框架，但模型中的调度器独立于估计器和控制器进行设计。LIMA 等[19] 在多智能体系统中采用 DRL 方法进行联合控制和资源分配，但在协同设计时忽略了状态估计器。这些方法中调度器的解耦设计或状态估计器的忽略将导致次优的结果，从而降低系统的整体控制性能。ZHAO 等[20] 提出了基于 DRL 的无模型估计—控制—调度联合设计框架，但只考虑了网络中的一个智能体，无法适用于具有多个智能体的 WNCS。JIANG 等[21] 提出了一种联合序列调度和轨迹规划方法，允许单个移动充电器在有静态障碍的无线传感器网络中进行移动充电，却同样不适用于多智能体系统。因此，针对缺乏系统动力学模型的复杂工业环境，如何对具有多个智能体的 WNCS 通信和控制进行协同设计仍存在巨大挑战。

针对以上挑战，本文提出了一种基于 DRL 的 WNCS 估计—控制—调度协同设计方法，主要贡献总结如下：1) 针对多智能体协同控制与无线网络资源受限的问题，提出了一种融合多智能体系统与 WNCS 的创新框架，综合考虑了状态估计、控制策略和资源调度，以在资源受限的情况下最大限度地提高系统的控制性能。2) 针对工业环境的动态性，本文将具有观测数据丢失的 WNCS 最优控制问题建模为部分可观测马尔可夫决策过程，并开发了一种用于完全合作型任务的 DRL 策略，通过循环神经网络处理具有时间相关性的历史信息，而无需精确的系统动力学模型。

1　模型描述

1.1　WNCS 模型

本文考虑一个包含 N 个智能体的 WNCS（如 AGV 协作搬运），每个智能体由一个传感器进行观测，通过基于 OFDMA 的共享无线网络（例如 5G 或 WIFI 6）连接到远程边缘处理器。如图 1 所示，在时间步 t，智能体 $i=1,2,\cdots,N$ 在被调度后，将观测数据 $\boldsymbol{o}_{i,t}$ 通过共享无线网络发送到无线控制器；边缘处理器根据观测数据 $\boldsymbol{o}_{i,t}$，生成控制动作 $\boldsymbol{a}_{i,t}^{\mathrm{C}}$，随后将其发送给智能体 i 执行动作。

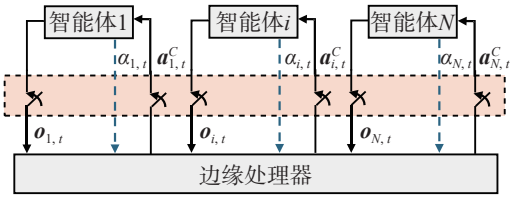


图 1　包含 N 个智能体的 WNCS

Fig.1　A WNCS with N agents

由于无线网络资源受限和无线信道衰落，WNCS 存在丢包和时延的问题。因此，本文将无线网络丢包的模型描述为：上行信道 i 在时间步 t 的丢包事件由伯努利变量 $\beta_{i,t}\sim B(p_{i,t}^{\mathrm{UL}})$ 表示，其中 $\beta_{i,t}=1$ 表示成功传输，$\beta_{i,t}=0$ 表示出现丢包。由于无线信道质量波动具有随机性，且单次传输结果仅有成功或失败 2 种可能，因此丢包事件可采用伯努利随机变量来描述[22]。同样，下行数据丢包事件由伯努利变量 $\alpha_{i,t}\sim Bernoulli(p_{i,t}^{\mathrm{DL}})$ 表示。

本文用信息年龄（AoI）来刻画 WNCS 中信息的新鲜度[23]。智能体观测数据的 AoI 表示为当前时间步 t 与各智能体最近一次成功传输的数据的生成时间步 $G_{i,t}$ 之间的差值。智能体 i 的观测数据在时间步 t 的 AoI 更新过程为

$$n_{i,t}^{\mathrm{AoI}}=\begin{cases}n_{i,t}^{\mathrm{AoI}}+1, & \text{数据传输失败}\\ t-G_{i,t}, & \text{数据传输成功}\end{cases}\tag{1}$$

由于传输开销可以忽略不计，本文假设智能体通过理想无线信道向边缘处理器发送 1 比特确认信息 $\alpha_{i,t}$。

1.2　协同设计目标

在完全合作型任务场景中，智能体 i 在时间步 t 的效益 $f_i\left(\boldsymbol{o}_{i,t},\boldsymbol{a}_{i,t}^{\mathrm{C}}\right)$ 取决于观测数据 $\boldsymbol{o}_{i,t}$ 和控制动作 $\boldsymbol{a}_{i,t}^{\mathrm{C}}$，表征其动作对任务的即时贡献。定义为

$$\bar{R}_{i,t}=f_i\left(\boldsymbol{o}_{i,t},\boldsymbol{a}_{i,t}^{\mathrm{C}}\right)\tag{2}$$

由于多个智能体共享一个资源受限的无线网络，系统需综合考虑所有智能体的任务完成情况和整体性能，并利用调度策略来动态分配网络资源。对于完全合作型任务，WNCS 的目标是获得最优策略 π^*，从而协调多个智能体的行为，同时避免冲突（$\varphi=0$），以最大化所有智能体的总效益，表示为

$$\pi^*\in\arg\max_{\pi}\lim_{T\to\infty}\sum_{t=1}^{T}\sum_{i=1}^{N}\gamma^{(t-1)}\cdot\bar{R}_{i,t}\tag{3a}$$

$$\text{s.t.}\ \sum_{i=1}^{N}\left(c_{i,t}^{\mathrm{UL}}+c_{i,t}^{\mathrm{DL}}\right)\leqslant C_t\tag{3b}$$

$$\frac{1}{T}\sum_{t=1}^{T}\gamma^{(t-1)}\cdot\bar{R}_{i,t}\geqslant\bar{R}_{\min}\tag{3c}$$

$$\sum_{t=1}^{T}\sum_{i=1}^{N}\varphi=0\tag{3d}$$

其中，$\gamma\in(0,1)$ 是折扣因子，当 γ 接近 0 时，代表智能体更加重视短期效益；当 γ 接近 1 时，智能体更加重视长期累计效益。C_t 为系统在时间步 t 时正交频分多址（OFDMA）网络可用的资源块的数量。$c_{i,t}^{\mathrm{UL}}$ 和 $c_{i,t}^{\mathrm{DL}}$ 分别为智能体 i 在时间步 t 上行和下行传输所用资源块的数量。$\bar{R}_{\min}$ 是在完全合作型任务场景中智能体的最小效益阈值，避免智能体为追求高效益而冒险采取危险动作（如相互碰撞）。

1.3　部分可观测马尔可夫决策过程

对于本文的决策问题（见式 (3)），由于无线通信信道存在丢包问题，观测数据并不总是能被无线控制器接收到。当一个智能体不能完全观察到系统的状态，并且底层的转移概率分布是未知的，智能体需要记忆过去的观测和行为来作出最优的决策[24]。因此，本文将该决策问题考虑为部分可观测的马尔可夫决策过程（POMDP），并采用 DRL 方法来解决。包含 N 个智能体的 WNCS 的状态空间、观测空间、动作空间和奖励函数定义如下：

状态空间 $\boldsymbol{S}=\{\boldsymbol{S}^{\mathrm{ag}},\boldsymbol{S}^{\mathrm{env}},\boldsymbol{S}^{\mathrm{ch}}\}$，表示智能体环境信息的集合，综合考虑智能体状态、工厂环境动态和信道条件。其中 $\boldsymbol{S}^{\mathrm{ag}}$ 包括智能体状态（例如位置），$\boldsymbol{S}^{\mathrm{env}}$ 表示环境动态，$\boldsymbol{S}^{\mathrm{ch}}$ 表示信道状态，包括 C、n^{AoI}、p^{UL} 和 p^{DL}。

观测空间 $\boldsymbol{O}=\{\boldsymbol{O}^{\mathrm{ag}},\boldsymbol{O}^{\mathrm{env}},\boldsymbol{O}^{\mathrm{ch}}\}$，表示智能体可感知的环境信息集合。其中 $\boldsymbol{O}^{\mathrm{ag}}$、$\boldsymbol{O}^{\mathrm{env}}$ 和 $\boldsymbol{O}^{\mathrm{ch}}$ 分别表示观测到的智能体状态、环境动态和信道状态。

动作空间 $\boldsymbol{A}=\boldsymbol{A}^1\times\boldsymbol{A}^2\times\cdots\times\boldsymbol{A}^N$，表示智能体的联合动作空间，是所有智能体与环境之间的交互行为。其中智能体 i 的 $\boldsymbol{A}^i$ 由控制动作（如智能体移动）和调度动作（如调度表）组成。

奖励函数R_t：环境根据智能体执行的动作反馈即时奖励，而智能体根据奖励调整行为策略。协同设计目标是最大化系统总效益，因此奖励函数应该与效益函数正相关。对于任何给定的动作$\boldsymbol{a}_t \in \boldsymbol{A}$，即时奖励计算如下：

$$R_t = \sum_{i=1}^{N} \overline{R}_{i,t} \tag{4}$$

在时间步 t 内，边缘处理器可能无法获得所有的智能体的观测数据。原则上，可以通过整个观测历史来推断该信息，但这将导致内存存储的巨大开销。传统基于贝叶斯滤波的信念状态方法虽能压缩历史信息为状态空间的概率分布，但其更新过程依赖环境动态。为此，本文在算法中使用循环神经网络，通过隐状态学习时序依赖关系，利用具有时间相关性的历史信息来重建当前状态解决部分可观测问题[25]。

2 方法

2.1 协同设计方法

本文基于 DRL 方法对估计器、控制器和调度器进行协同设计。部分可观测性使得系统的决策不能只依赖于当前观测数据，智能体需要记住历史数据，以便推断实际状态并选择最优行为[26]。因此，本文在进行估计、控制和调度决策时综合考虑了系统当前状态和历史状态。

本文将多智能体系统框架与 WNCS 框架相结合，提出了基于 DRL 的估计—控制—调度协同设计方法，如图 2 所示。边缘处理器由估计器、控制器和调度器组成。估计器通过历史观测数据重建当前系统状态，解决丢包导致的信息缺失问题，为控制决策提供可靠输入；控制器基于估计状态生成多智能体的控制动作；调度器通过动态分配网络资源，优先调度关键智能体的观测数据上传，优化网络资源受限情况下的整体系统性能。在时间步 t，边缘处理器通过共享无线网络收集从智能体与环境交互中获取的观测数据 $\boldsymbol{o}_t$。随后，估计器针对数据包丢失情况输出系统状态估计值 $\overline{\boldsymbol{o}}_t$，控制器则基于 $\overline{\boldsymbol{o}}_t$ 生成控制输入 $\boldsymbol{a}_t^{\mathrm{C}}$。紧接着，估计器会根据 $\boldsymbol{a}_t^{\mathrm{C}}$ 和 $\boldsymbol{o}_t$ 预测下一时刻的状态 $\hat{\boldsymbol{o}}_{t+1}$，并将其传递给调度器。调度器据此生成针对时间步 t+1 的调度表动作 a_{t+1}^{Tx}。最终，a_{t+1}^{Tx} 与 $\boldsymbol{a}_t^{\mathrm{C}}$ 将被同时传输至各智能体，这些智能体会立即执行对应的控制动作，并依照 a_{t+1}^{Tx} 的调度安排在时间步 t+1 发送观测数据 $\boldsymbol{o}_{t+1}$。

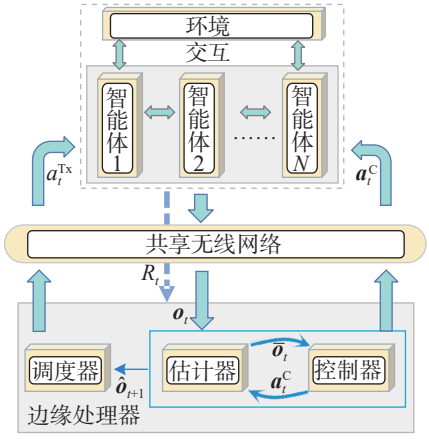


图 2 估计—控制—调度协同设计框架

Fig.2 Estimation-control-scheduling co-design framework

在本方法的协同设计中，估计器、控制器和调度器的损失函数在训练中通过信息依赖形成耦合关系：估计误差影响控制器的策略更新，还作为调度决策的参考指标；控制器输出参与状态预测，影响调度器的 Q 值计算；调度器的资源分配又决定观测数据的完整性，反过来影响估计精度，从而实现三者的协同优化。

2.2 基于 DL 的状态估计

无线网络的资源限制造成了 WNCS 中数据传输的丢包和时延。在理想情况下，智能体的观测数据可以直接反映其当前状态。然而，在实际环境中，系统的真实状态可能是无法观测到的，并且由于无线网络资源受限导致的丢包，观测数据往往是不完整的。

对于智能体 i，如果它在时间步 t 被调度且观测数据成功传输（$a_{i,t}^{\mathrm{Tx}} \cdot \beta_{i,t} = 1$），就可以直接使用观测数据 $\boldsymbol{o}_{i,t}$；否则，需要引入一个估计网络 $\pi_i^{\mathrm{E}}\left(\boldsymbol{h}_{i,t}^{\mathrm{E}};\boldsymbol{\theta}_i^{\mathrm{E}}\right)$，将观测数据 $\boldsymbol{o}_{i,t}$ 和历史数据 $\boldsymbol{h}_{i,t}^{\mathrm{E}}$ 作为输入，得到当前状态的估计值 $\overline{\boldsymbol{o}}_{i,t}$。通过最小化损失函数 $L_t(\boldsymbol{\theta}_i^{\mathrm{E}})$ 来训练参数 $\boldsymbol{\theta}_i^{\mathrm{E}}$，其中 $\boldsymbol{h}_{i,t}^{\mathrm{E}}$ 是在时间步 t 之前长为 M 的历史数据，由状态估计值和实际控制动作组成：

$$\boldsymbol{h}_{i,t}^{\mathrm{E}} \triangleq \begin{cases} \left[\overline{\boldsymbol{o}}_{i,t-L}, \boldsymbol{a}_{i,t-L}^{\mathrm{C}}, \ldots, \overline{\boldsymbol{o}}_{i,t-1}, \boldsymbol{a}_{i,t-1}^{\mathrm{C}}\right], & t > L \\ \left[0,0,\ldots,\overline{\boldsymbol{o}}_{i,1}, \boldsymbol{a}_{i,1}^{\mathrm{C}}, \ldots, \overline{\boldsymbol{o}}_{i,t-1}, \boldsymbol{a}_{i,t-1}^{\mathrm{C}}\right], & M \geqslant t > 1 \\ [0,0,\ldots,0,0], & t \leqslant 1 \end{cases} \tag{5}$$

估计网络的设计目标是最小化智能体观测数据和估计状态之间的差异，因此，损失函数定义为

$$L_t(\boldsymbol{\theta}_i^{\mathrm{E}}) = \left\| \boldsymbol{o}_{i,t} - \pi_i^{\mathrm{E}}\left(\boldsymbol{h}_{i,t}^{\mathrm{E}};\boldsymbol{\theta}_i^{\mathrm{E}}\right) \right\|^2 \tag{6}$$

本文使用长短期记忆（LSTM）网络来实现估计网络，相比 Transformer 等模型，LSTM 具有参数量

少、结构轻量的优势，其序列化处理结构更适用于工业控制场景的短序列数据建模。图 3 展示了基于 LSTM 的估计器结构(FC 表示全链接层)，并通过学习率为α^{E}的梯度下降法更新参数$\boldsymbol{\theta}_i^{\mathrm{E}}$：

$$\boldsymbol{\theta}_i^{\mathrm{E}} \leftarrow \boldsymbol{\theta}_i^{\mathrm{E}} - \alpha^{\mathrm{E}} \cdot \nabla_{\theta_i^{\mathrm{E}}} L_t(\boldsymbol{\theta}_i^{\mathrm{E}}) \tag{7}$$

通过估计网络$\pi_i^{\mathrm{E}}\left(\boldsymbol{h}_{i,t}^{\mathrm{E}};\boldsymbol{\theta}_i^{\mathrm{E}}\right)$，能够得到状态估计值为

$$\overline{\boldsymbol{o}}_{i,t} = \begin{cases} \boldsymbol{o}_{i,t}, & a_{i,t}^{\mathrm{Tx}} \cdot \beta_{i,t} = 1 \\ \pi_i^{\mathrm{E}}\left(\boldsymbol{h}_{i,t}^{\mathrm{E}};\boldsymbol{\theta}_i^{\mathrm{E}}\right), & a_{i,t}^{\mathrm{Tx}} \cdot \beta_{i,t} \neq 1 \end{cases} \tag{8}$$

得到的状态估计值$\overline{\boldsymbol{o}}_{i,t}$将作为控制器输入，用于生成控制动作$\boldsymbol{a}_{i,t}^{\mathrm{C}}$，确保智能体在观测数据缺失时仍能可靠运行。

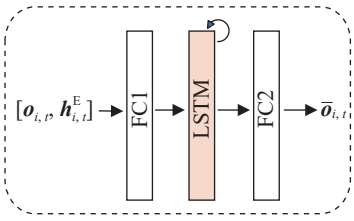


图 3　基于 LSTM 的估计器结构

Fig.3　LSTM-based estimator architecture

2.3　基于 DRL 的无模型控制

本文采用了基于“演员—评论家”结构的多智能体策略梯度方法(即 counterfactual multi-agent，COMA 算法)[27]。由于在完全合作型任务中，奖励函数是全局共享的，因此 COMA 算法采用了中心化的评论家网络，如图 4 所示(GRU 代表门控循环单元)，通过反事实基线来解决信用分配的问题。

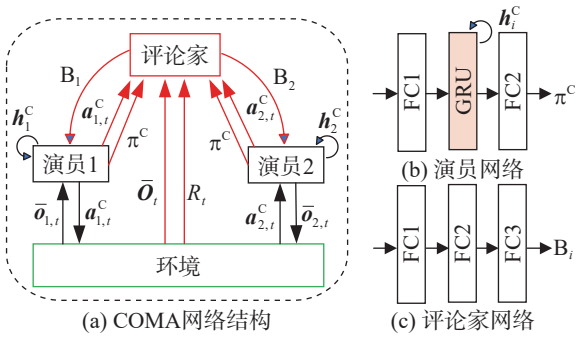


图 4　基于 COMA 的控制器结构

Fig.4　COMA-based controller architecture

控制策略函数由π_i^{C}给出，并通过向量$\boldsymbol{\theta}_i^{\mathrm{C}}$参数化。给定状态输入$\overline{\boldsymbol{o}}_{i,t}$和由估计状态和控制输入组成的控制历史信息$\boldsymbol{h}_{i,t}^{\mathrm{C}}$，控制策略表示为

$$\pi_i^{\mathrm{C}}\left(\boldsymbol{a}_{i,t}^{\mathrm{C}} \middle| \overline{\boldsymbol{o}}_{i,t}, \boldsymbol{h}_{i,t}^{\mathrm{C}}; \boldsymbol{\theta}_i^{\mathrm{C}}\right) \tag{9}$$

而对于中心化的评论家，则给定全局状态$\overline{\boldsymbol{O}}_t = [\overline{\boldsymbol{o}}_{1,t}, \cdots, \overline{\boldsymbol{o}}_{N,t}]$和联合控制动作$\boldsymbol{A}_t^{\mathrm{C}} = [\boldsymbol{a}_{1,t}^{\mathrm{C}}, \cdots, \boldsymbol{a}_{N,t}^{\mathrm{C}}]$，评论家网络输出状态动作价值(Q 值)。该网络通过改进的 TD(λ) 目标进行更新，能在蒙特卡洛造成的方差和自举造成的偏差之间找到较好的平衡。评论家网络的损失函数定义为

$$L_t(\boldsymbol{\psi}^{\mathrm{C}}) = \left(y_t^{(\lambda)} - \mathrm{Q}_{\psi^{\mathrm{C}}}(\overline{\boldsymbol{O}}_t, \boldsymbol{A}_t^{\mathrm{C}}; \boldsymbol{\psi}^{\mathrm{C}})\right)^2 \tag{10}$$

其中，$y_t^{(\lambda)}$是 TD(λ) 的目标值，是多步奖励与目标评论家网络的状态价值估计的加权和。评论家网络参数$\boldsymbol{\psi}^{\mathrm{C}}$和目标网络参数$\boldsymbol{\psi}'^{\mathrm{C}}$更新如下：

$$\boldsymbol{\psi}^{\mathrm{C}} \leftarrow \boldsymbol{\psi}^{\mathrm{C}} - \alpha^{\mathrm{C}} \cdot \nabla_{\psi^{\mathrm{C}}} L_t(\boldsymbol{\psi}^{\mathrm{C}}) \tag{11}$$

$$\boldsymbol{\psi}'^{\mathrm{C}} \leftarrow \tau\boldsymbol{\psi}^{\mathrm{C}} + (1-\tau)\boldsymbol{\psi}'^{\mathrm{C}} \tag{12}$$

其中，$0 < \tau \leqslant 1$是软更新率，α^{C}是评论家网络的学习率。

在传统的多智能体策略梯度方法中，演员网络的梯度更新依赖于全局奖励的时间差分(TD)误差，但由于所有智能体共享同一奖励信号，单个智能体的行为对全局奖励的具体贡献难以准确衡量(即信用分配问题)。COMA 通过引入一个中心化的评论家，利用反事实基线计算差异奖励。具体而言，反事实基线通过比较当前联合动作的 Q 值与该智能体采取其他可能动作时的期望 Q 值来计算奖励差值，从而精确衡量该智能体对团队奖励的个体贡献。

除智能体 i 外所有其他智能体的联合动作定义为$\boldsymbol{A}_{-i,t}^{\mathrm{C}} = [\boldsymbol{a}_{1,t}^{\mathrm{C}}, \cdots, \boldsymbol{a}_{i-1,t}^{\mathrm{C}}, \boldsymbol{a}_{i+1,t}^{\mathrm{C}}, \cdots, \boldsymbol{a}_{N,t}^{\mathrm{C}}]$，对于每个智能体 i，通过计算一个优势函数将当前的 Q 值与其反事实基线进行比较：

$$\begin{aligned} \mathrm{B}_i =& \mathrm{Q}_{\psi^{\mathrm{C}}}\left(\overline{\boldsymbol{O}}_t, \boldsymbol{A}_t^{\mathrm{C}}\right) - \sum_{a_{i,t}^{\mathrm{C}}} \mathrm{m}_i^{\mathrm{C}}\left(\boldsymbol{a}_{i,t}^{\mathrm{C}} \middle| \overline{\boldsymbol{O}}_t, \boldsymbol{h}_{i,t}^{\mathrm{C}}\right) \\ & \cdot \mathrm{Q}_{\psi^{\mathrm{C}}}\left(\overline{\boldsymbol{O}}_t, \left(\boldsymbol{A}_{-i,t}^{\mathrm{C}}, \boldsymbol{a}_{i,t}'^{\mathrm{C}}\right)\right) \end{aligned} \tag{13}$$

这种方法使得 COMA 能够精确量化每个智能体的动作对团队奖励的影响，从而缓解策略冲突。则带反事实基线的近似策略梯度为

$$\boldsymbol{g}_K = E_\pi\left[\sum_{i=1}^{N} \mathrm{B}_i \cdot \nabla_{\boldsymbol{\theta}_K^{\mathrm{c}}} \ln \pi_i^{\mathrm{c}}\left(\boldsymbol{a}_{i,t}^{\mathrm{c}} \middle| \boldsymbol{h}_{i,t}^{\mathrm{c}}; \boldsymbol{\theta}_K^{\mathrm{c}}\right)\right] \tag{14}$$

其中 K 是迭代次数，$\boldsymbol{\theta}_K^{\mathrm{C}}$是迭代 K 次时的策略网络参数，$\boldsymbol{\theta}_K^{\mathrm{C}} = \{\boldsymbol{\theta}_{1,K}^{\mathrm{C}}, \cdots, \boldsymbol{\theta}_{N,K}^{\mathrm{C}}\}$。随后，通过学习率为$\alpha^{\mathrm{A}}$的梯度上升来更新智能体 i 的策略参数：

$$\boldsymbol{\theta}_K^{\mathrm{C}} \leftarrow \boldsymbol{\theta}_K^{\mathrm{C}} + \alpha^{\mathrm{A}} \cdot \boldsymbol{g}_K \tag{15}$$

需要注意的是，所有的演员网络共享一套网络参数。

2.4　调度器设计

在 WNCS 中，调度问题通常涉及在资源受限的

环境下进行实时决策，以优化任务分配和资源利用率。尽管深度 Q 网络(DQN)通常用于解决具有离散动作的决策问题，但传统的 DQN 算法不能有效地解决 POMDP 问题。因此，本文采用基于深度递归 Q 网络(DRQN)的调度器，如图 5 所示，DRQN 通过引入循环神经网络来处理历史信息，使智能体在决策时能补偿部分丢失的状态信息。

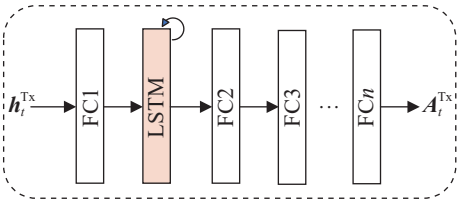


图 5 基于 DRQN 的调度器结构

Fig.5 DRQN-based scheduler architecture

在得到当前控制动作 $\boldsymbol{a}_{i,t}^{\mathrm{C}}$ 后，未来状态可以通过历史数据 $\boldsymbol{h}_{i,t}^{\mathrm{E}}$、当前控制动作 $\boldsymbol{a}_{i,t}^{\mathrm{C}}$ 和观测值 $\boldsymbol{o}_{i,t}$ 进行预测。未来状态由式 (16) 给出：

$$\hat{\boldsymbol{o}}_{i,t+1}=\pi_i^{\mathrm{E}}\left(\left(\boldsymbol{h}_{i,t}^{\mathrm{E}},\boldsymbol{a}_{i,t}^{\mathrm{C}},\boldsymbol{o}_{i,t}\right);\boldsymbol{\theta}_i^{\mathrm{E}}\right) \tag{16}$$

调度动作 $\boldsymbol{A}_t^{\mathrm{Tx}}=[a_{1,t}^{\mathrm{Tx}},\cdots,a_{N,t}^{\mathrm{Tx}}]$ 定义为 N 个 1 维调度信号 $a_{i,t}^{\mathrm{Tx}}\in\{0,1\}$ 组成的向量。实际上，当前时间步的调度动作是在接收传感器数据包之前进行的。当前的调度动作可通过 ε-贪心策略得到，即以概率 ε 选择随机动作 $\boldsymbol{A}_t^{\mathrm{Tx}}$，否则通过最大化 Q 值来选择调度动作：

$$\boldsymbol{A}_t^{\mathrm{Tx}}=\begin{cases}\text{随机选择}, & \text{概率为}\varepsilon\\ \underset{\boldsymbol{A}}{\operatorname{argmax}}\mathrm{Q}\left(\hat{\boldsymbol{O}}_{t+1},\boldsymbol{h}_t^{\mathrm{Tx}},\boldsymbol{A};\boldsymbol{\theta}_t^{\mathrm{Tx}}\right), & \text{概率为}1-\varepsilon\end{cases} \tag{17}$$

其中，$\boldsymbol{h}_t^{\mathrm{Tx}}$ 是调度器历史，包括各智能体的观测值及其调度动作。$\hat{\boldsymbol{O}}_{t+1}=[\hat{\boldsymbol{o}}_{1,t+1},\cdots,\hat{\boldsymbol{o}}_{N,t+1}]$ 是未来状态组成的矩阵。在训练迭代过程中，探索参数 ε 逐渐减小，最终收敛到 0.01，以确保策略稳定性。则 DRQN 的损失函数可以定义为

$$L_t(\boldsymbol{\theta}^{\mathrm{Tx}})=\left(y_t-\mathrm{Q}_{\boldsymbol{\theta}^{\mathrm{Tx}}}(\hat{\boldsymbol{O}}_{t+1},\boldsymbol{h}_t^{\mathrm{Tx}},\boldsymbol{A}_{t+1}^{\mathrm{Tx}};\boldsymbol{\theta}^{\mathrm{Tx}})\right)^2 \tag{18}$$

其中，TD 目标 y_t 通过目标网络计算得到。DRQN 网络参数 $\boldsymbol{\theta}^{\mathrm{Tx}}$ 及其目标网络参数 $\boldsymbol{\theta}'^{\mathrm{Tx}}$ 更新如下：

$$\boldsymbol{\theta}^{\mathrm{Tx}}\leftarrow\boldsymbol{\theta}^{\mathrm{Tx}}-\alpha^{\mathrm{Tx}}\cdot\nabla_{\boldsymbol{\theta}^{\mathrm{Tx}}}L_t(\boldsymbol{\theta}^{\mathrm{Tx}}) \tag{19}$$

$$\boldsymbol{\theta}'^{\mathrm{Tx}}\leftarrow\tau\boldsymbol{\theta}^{\mathrm{Tx}}+(1-\tau)\boldsymbol{\theta}'^{\mathrm{Tx}} \tag{20}$$

DRQN 的 LSTM 层需要学习时序依赖关系，其训练必须基于连续的样本序列。然而，传统 DQN 的经验回放通过随机采样独立的经验元组破坏了序列相关性，不适用于 DRQN[28]。因此，DRQN 采用序列化采样策略进行训练：从经验回放池中随机抽取包含 P 个回合(episode)的小批量数据，每个片段从完整回合的随机位置截取，包含 N_s 个连续时间步的经验元组(状态—动作—奖励序列)。

2.5 算法伪码

详细的协同设计方法训练过程的伪代码在算法 1 中给出。本文采用固定长度的滑动窗口机制构建历史缓存区，用以存放历史数据 $\boldsymbol{h}_{i,t}^{\mathrm{E}}$、$\boldsymbol{h}_{i,t}^{\mathrm{C}}$ 和 $\boldsymbol{h}_t^{\mathrm{Tx}}$。在经验回放池 D 中，时间步 t 的经验元组表示为 $\left(\overline{\boldsymbol{O}}_t,\boldsymbol{A}_t^{\mathrm{Tx}},\boldsymbol{A}_t^{\mathrm{C}},R_t,\overline{\boldsymbol{O}}_{t+1}\right)$。

算法1 基于DRL的协同设计算法

输入： 智能体、网络及环境设置
输出： 训练完成的DRL网络

1. 初始化各网络参数 $\boldsymbol{\theta}^{\mathrm{E}}$、$\boldsymbol{\theta}^{\mathrm{C}}$、$\boldsymbol{\psi}^{\mathrm{C}}$、$\boldsymbol{\theta}^{\mathrm{Tx}}$
2. 初始化各目标网络参数 $\boldsymbol{\psi}'^{\mathrm{C}}\leftarrow\boldsymbol{\psi}^{\mathrm{C}}$，$\boldsymbol{\theta}'^{\mathrm{Tx}}\leftarrow\boldsymbol{\theta}^{\mathrm{Tx}}$
3. 初始化经验回放池 D
4. **for** all episodes **do**
5. t=0，随机初始化调度动作 $\boldsymbol{A}_t^{\mathrm{Tx}}$
6. **for** an episode **do**
7. 根据式(8)更新估计状态 $\overline{\boldsymbol{O}}_t$
8. 根据式(9)更新控制动作 $\boldsymbol{A}_t^{\mathrm{C}}$
9. 根据式(4)更新奖励 R_t
10. 根据式(17)更新时间步t+1的调度动作 $\boldsymbol{A}_{t+1}^{\mathrm{Tx}}$
11. 更新历史数据 $\boldsymbol{h}_t^{\mathrm{E}}$、$\boldsymbol{h}_t^{\mathrm{C}}$ 和 $\boldsymbol{h}_t^{\mathrm{Tx}}$
12. 将 $\left(\overline{\boldsymbol{O}}_t,\boldsymbol{A}_t^{\mathrm{Tx}},\boldsymbol{A}_t^{\mathrm{C}},\mathcal{R}_t,\overline{\boldsymbol{O}}_{t+1}\right)$ 存到经验池 D
13. 分别根据式(7)(11)(12)(15)(19)(20)更新网络参数 $\boldsymbol{\theta}^{\mathrm{E}}$、$\boldsymbol{\theta}^{\mathrm{C}}$、$\boldsymbol{\psi}^{\mathrm{C}}$、$\boldsymbol{\theta}^{\mathrm{Tx}}$、$\boldsymbol{\psi}'^{\mathrm{C}}$ 和 $\boldsymbol{\theta}'^{\mathrm{Tx}}$
14. $t=t$+1
15. **end for**
16. **end for**

3 仿真结果与分析

3.1 仿真设置

本文考虑一个多 AGV 完全合作搬运场景。在一个 30×30 网格地图内，3 辆 AGV 到达货物存放区，形成队列，从而协作搬运大型货物。大型物体运输是一项典型的任务[29]，场景中的 AGV 之间必须避免相互碰撞。图 6 展示了 AGV 协作搬运场景示意图。

每个 AGV 的观测空间由货物位置、目标区域位置以及到其他 AGV 的相对距离组成。动作空间包含 5 个离散动作，前进、后退、左移、右移、停止[30]。

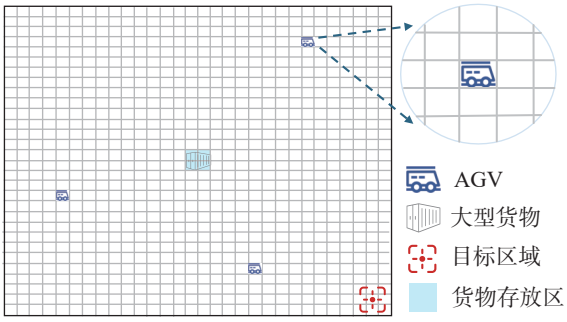


图 6　AGV 协作搬运场景

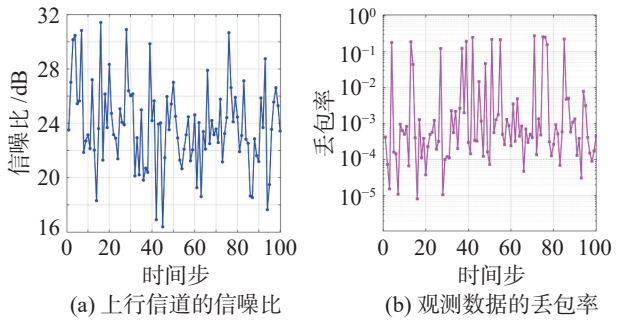
Fig.6　AGV collaborative transportation scenario

由于 IEEE 802.11ax 标准通过 OFDMA 多用户接入和集中调度机制能够有效降低时延抖动，满足 AGV 的实时控制需求[31]，因此本文考虑 AGV 通过 IEEE 802.11ax 网络与边缘处理器进行通信。IEEE 802.11ax 网络采用 20 MHz 工作带宽和 SISO 单天线传输模式，使用室内瑞利衰落信道模型进行仿真[32]，阴影衰落标准差为 3dB，调制与编码策略(MCS)动态自适应，最大重传 4 次。上行数据包大小为 16 384 B，下行数据包大小为 32 B，无线接入点(AP)部署在场景中心，AP 与 AGV 天线数为 1×1。为了衡量网络的性能，本文评估了一个随机的 AGV 和 AP 之间的信噪比以及 AGV 上行信道观测数据包的丢包率。如图 7 所示，当信噪比低于 20 dB 时，丢包率会大幅增加。

在协作搬运任务中，每个 AGV 的效益函数由多个因素决定，包括任务成功率、碰撞风险和任务完成时间，本文的 AGV 效益函数定义为

$$\overline{R}_{i,t} = r^{\mathrm{S}}_{i,t} - r^{\mathrm{P}}_{i,t} - r^{\mathrm{T}}_{i,t} \tag{21}$$

其中，$r^{\mathrm{S}}_{i,t}$ 代表完成一次运输任务所获得的奖励；$r^{\mathrm{P}}_{i,t}$ 代表基于碰撞风险的惩罚项，通过“社交半径”内的重叠面积来量化 AGV 之间的潜在碰撞风险，“社交半径”是每个 AGV 的安全交互范围，综合考虑了



(a) 上行信道的信噪比　(b) 观测数据的丢包率

图 7　一个 AGV 的上行信道的信噪比和丢包率

Fig.7　Signal-to-Noise Ratio and Packet Error Rate of the uplink from an AGV

AGV 的物理半径和移动步长及传感器噪声导致的距离估计偏差。具体而言，对于每个 AGV，检测所有满足间距 $d \leqslant 2R_{\mathrm{social}}$ 的 AGV，当 AGV 的“社交半径”内存在重叠区域时，估计器通过历史数据平滑噪声影响，从而在噪声干扰下仍能可靠评估碰撞风险，重叠面积越大，表明碰撞风险越高，惩罚越重，如图 8 所示；引入 $r^{\mathrm{T}}_{i,t}$ 则用于惩罚 AGV 的移动行为。

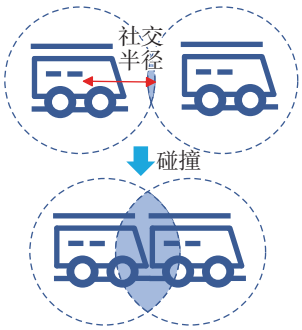


图 8　AGV 的“社交半径”

Fig.8　AGVs’ social radius

其他训练参数，比如奖励、学习率和折扣因子等，在表 1 中给出。其中，贪心参数 ε 采用线性衰减，在训练过程中从初始值 0.5 线性递减至 0.01，以此平衡探索与利用行为。

表 2　网络训练参数

Tab.2　Parameters for network training

参数/超参数名	参数值
任务完成奖励 r^{S}	50
碰撞惩罚 r^{P}	$\propto S_{\text{重叠面积}}$
移动惩罚 r^{T}	1
回合长度 T	100个时间步
历史数据长度 M	5
最大时间步	1.6×10^{4}个时间步
批量大小 $P\times N_s$	32
贪心参数 ε	0.01 ~ 0.5
折扣因子 γ	0.99
LSTM学习率 α^{E}	5×10^{-3}
演员网络学习率 α^{A}	5×10^{-3}
评论家网络学习率 α^{C}	5×10^{-3}
DRQN学习率 α^{Tx}	5×10^{-3}

3.2 仿真结果分析

为了验证所提出方法的有效性，本文通过100次独立试验将本方法与其他基于DRL的方法进行了对比，实验结果如表2所示。实验结果表明，在无线网络资源受限的环境中，无协同设计(简称NCoMASs)的方法、缺少估计器(简称MCoMASs[18])和单独设计调度器(简称CACoMASs[19])的传统解耦设计方法，由于估计器、控制器和调度器独立优化，各模块间的协同性不足，难以实现全局最优，从而导致WNCS整体控制性能下降，其平均任务完成率分别为77.8%、92.3%和91.6%，平均任务完成时间分别为90、79和81个时间步。与解耦设计方法相比，本文方法平均任务完成率提升至96.1%(相对提升3.8%～4.5%)，平均任务完成时间缩短至73个时间步(相对减少7.6%～9.9%)，体现了本文方法在WNCS中的性能优势。特别地，任务完成率的提升并未影响实时性，这得益于协同设计对估计、控制和调度的联合优化。

表3 本文方法与其他方法的对比

Tab.3 Comparison of the proposed method with other methods

方法	任务完成率/%	任务完成时间步
MCoMASs	92.3	79
CACoMASs	91.6	81
NCoMASs	77.8	90
本文方法	96.1	73

图9展示了在5个随机实验中，不同方法的平均任务完成率随训练回合数的变化关系。实验结果表明，各算法在训练过程中均能在600回合左右达到稳定状态。在资源受限的环境中，本文方法的任务完成率显著优于其他方法，其平均任务完成率在训练后期稳定在较高水平(约96%)。

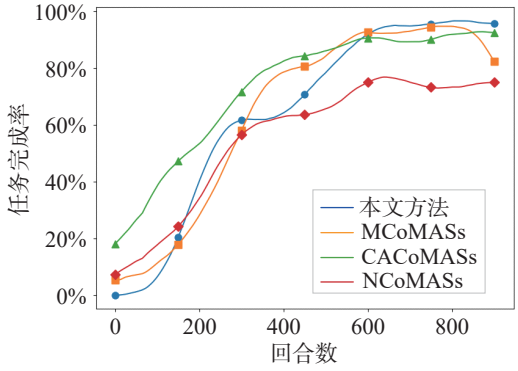


图9 任务完成率对比

Fig.9 Comparison on the task success rate

图10展示了不同方法的平均奖励随训练回合数的变化关系。折线代表5个不同随机实验的平均值，而阴影部分代表奖励的正负标准差。实验结果表明，在资源受限的环境中，本文方法的回合奖励值显著高于其他方法，应用本文方法的智能体在训练过程中的平均奖励单调上升，且标准差稳定在±6%内，阴影区域较窄，未观察到发散行为，表明策略稳定收敛。相比之下，其他方法的平均奖励值较低，且阴影区域较宽，稳定性较差。

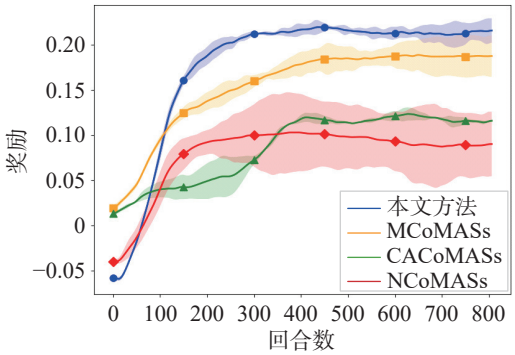


图10 回合奖励对比

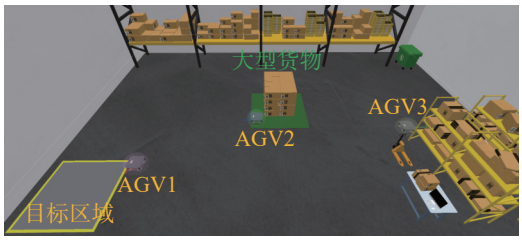
Fig.10 Comparison on episode reward

为使实验结果更具有现实意义，本文在CoppeliaSim仿真平台上进行了物理仿真实验。CoppeliaSim是一个功能强大的机器人仿真平台，支持多机器人系统的建模与仿真，能够模拟真实环境中的物理特性(如碰撞、传感器噪声等)[33]。

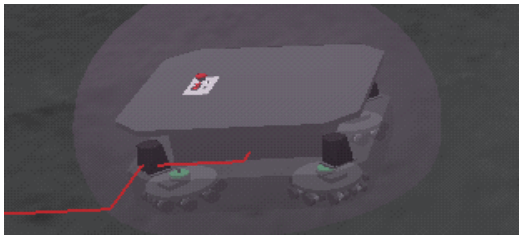
本文基于3.1节中的仿真设置搭建了一个900 m²的室内场景，场景中设置了货物存放区、目标区域以及障碍物，模拟真实工厂中的布局。在图11(a)中显示了协作搬运大型货物任务的图形界面，图11(b)中显示了AGV的社交半径。实验以全向移动AGV[34]为仿真控制对象，其尺寸参数如下：长0.64 m、宽0.64 m、高0.3 m。实验所搬运的货物尺寸为：长2 m、宽1.5 m、高1.5 m。

在实验过程中，CoppeliaSim仿真环境实时更新AGV的状态信息，并通过应用程序编程接口(API)将观测数据传输至Python端的边缘处理器。边缘处理器基于训练好的模型，对接收到的状态信息进行处理，并计算最优控制指令。随后，这些指令通过接口下发至AGV，控制AGV完成协同搬运大型货物的任务。

从图12(a)和图12(b)中可以看出AGV从随机位置出发，逐步抵达货物位置，形成队列并成功将大型货物搬运至指定地点。值得注意的是，图12(a)



(a) AGV协作搬运场景

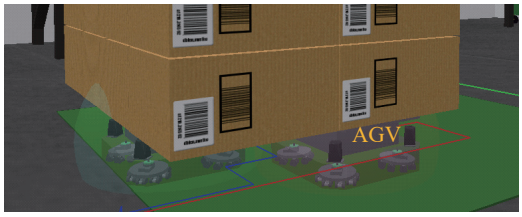


(b) AGV的社交半径

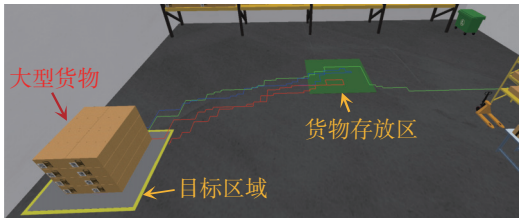
图 11　CoppeliaSim 中的仿真环境图

Fig.11　Simulation environment in CoppeliaSim

中 AGV3 在接近 AGV1 时主动调整了运动方向，成功避让已到达货物存放区的 AGV1，并与其协同形成队列。这一行为体现了本方法在动态环境中的灵活性和协作能力。



(a) AGV形成队列



(b) 成功将货物搬运到目标区域

图 12　协作搬运示意图

Fig.12　Illustration on collaborative transportation

本文对图 12(b) 中 AGV 的轨迹进行了量化分析，如表 3 所示。其中移动成本、绕行比率和运行时间均基于成功完成任务的事件进行计算[35]，且每一步的运行时间固定为 50 ms。实验结果表明，本方法在完成任务时使用了最少的移动步数，同时在移动成本和绕行比率等指标上均优于对比方法，展现了其在运动规划和任务执行效率上的显著优势。由于协同设计带来的高效资源调度策略和基于社交半径的冲突避免机制，显著减少了 AGV 的停顿次数，从而降低了移动成本。绕行比率的优化则降低了能耗并缓解了通道拥堵，是系统整体效率提升的间接体现。

表 4　运动轨迹对比

Tab.4　Comparisons on motion trajectories

方法	移动成本	绕行比率 /%
MCoMASs	1.19	18.6
CACoMASs	1.23	19.4
本文方法	1.04	15.9

4　结论

本文提出了一种基于 DRL 的 WNCS 估计—控制—调度协同设计方法，并提出了一个针对完全合作型任务的优化问题，以在无线网络资源受限的情况下最大限度地提高系统的控制性能。该方法使用了循环神经网络，利用具有时间相关性的历史信息来补偿丢包，从而解决部分可观测问题。本文设计了不同方法的对比试验，以评估本方法在基于 IEEE 802.11ax 网络的 AGV 协作搬运大型货物的场景中的有效性。仿真实验结果表明，本文方法提高了 AGV 的任务完成率，并有效减少了 AGV 的任务完成时间，并且调度器能在任务完成率、时间成本与资源利用率间达成接近帕累托前沿的有效折中。同时相比于其他方法 AGV 的移动成本降低了 0.15，绕行比率减少了 2.7%。尽管本文方法在 3 个 AGV 场景表现良好，需注意的是，其计算复杂度随智能体数量呈多项式增长，可能会影响本文方法在大规模场景中的控制性能。

后续将研究分布式架构以支持更大规模场景，并对分布式框架下算法的收敛性与性能边界开展深入的理论分析，为本文方法在复杂工业场景中的实际应用提供基础理论支撑。

参考文献

[1]　GRONAUER S, DIEPOLD K. Multi-agent deep reinforcement learning: A survey[J]. Artificial Intelligence Review, 2022, 55(2): 895 − 943.

[2]　任彦, 岳美霞, 解东, 等. 基于有限时间干扰观测器的多智能体系统的协同控制[J]. 信息与控制, 2021, 50(3): 343 − 349.

REN Y, YUE M X, XIE D, et al. Cooperative control of multi-agent systems based on finite-time disturbance observer[J]. Information and Control, 2021, 50(3): 343 − 349.

[3] PAPADOPOULOS G, KONTOGIANNIS A, PAPADOPOULOU F, et al. An extended benchmarking of multi-agent reinforcement learning algorithms in complex fully cooperative tasks[EB/OL]. (2025-07-04)[2025-07-21]. https://arxiv.org/abs/2502.04773. DOI: 10.48550/arXiv.2502.04773.

[4] LI Z, ZHANG H, LI X, et al. Distributed task scheduling for MEC-assisted virtual reality: A fully-cooperative multiagent perspective[J]. IEEE Transactions on Vehicular Technology, 2024, 73(7): 10572 − 10586.

[5] LI T, ZHU K, LUONG N C, et al. Applications of multi-agent reinforcement learning in future Internet: A comprehensive survey[J]. IEEE Communications Surveys & Tutorials, 2022, 24(2): 1240 − 1279.

[6] SARDAR A A, RAO A S, ALPCAN T, et al. Network resource allocation for Industry 4.0 with delay and safety constraints[J]. IEEE Transactions on Cognitive Communications and Networking, 2024, 10(1): 223 − 237.

[7] 裘莹, 张敬宣, 柯杰, 等. 工业无线网络实时传输调度算法研究综述[J]. 自动化学报, 2024, 50(11): 2102 − 2127.
QIU Y, ZHANG J X, KE J, et al. A survey of real-time transmission scheduling algorithms for industrial wireless network[J]. Acta Automatica Sinica, 2024, 50(11): 2102 − 2127.

[8] WANG Y, WU S, LEI C, et al. A review on wireless networked control system: The communication perspective[J]. IEEE Internet of Things Journal, 2024, 11(5): 7499 − 7524.

[9] 张强. 不确定环境下多自主体系统的分布式估计与控制[J]. 中国科学: 数学, 2013, 43(6): 529 − 540.
ZHANG Q. Distributed estimation and control of multi-agent systems in uncertain environment[J]. Scientia Sinica Mathematica, 2013, 43(6): 529 − 540.

[10] GATSIS K, PAJIC M, RIBEIRO A, et al. Opportunistic control over shared wireless channels[J]. IEEE Transactions on Automatic Control, 2015, 60(12): 3140 − 3155.

[11] KNORN S, DEY S. Optimal energy allocation for linear control with packet loss under energy harvesting constraints[J]. Automatica, 2017, 77: 259 − 267.

[12] LIU W, POPOVSKI P, LI Y, et al. Wireless networked control systems with coding-free data transmission for industrial IoT[J]. IEEE Internet of Things Journal, 2020, 7(3): 1788 − 1801.

[13] HUANG K, LIU W, LI Y, et al. Optimal downlink-uplink scheduling of wireless networked control for industrial IoT[J]. IEEE Internet of Things Journal, 2020, 7(3): 1756 − 1772.

[14] ZHOU Y, FENG Z, SONG Z, et al. Integrated sensing, communication, and control driven multi-AGV closed-loop control[J]. IEEE Transactions on Vehicular Technology, 2025, 74(7): 10853 − 10868.

[15] BAEK J, KADDOUM G. Heterogeneous task offloading and resource allocations via deep recurrent reinforcement learning in partial observable multifog networks[J]. IEEE Internet of Things Journal, 2021, 8(2): 1041 − 1056.

[16] 张宏达, 李德才, 何玉庆. 基于不完备信息预测的多智能体分布式协同[J]. 信息与控制, 2024, 53(1): 86 − 97.
ZHANG H D, LI D C, HE Y Q. Multi-agent distributed cooperation based on incomplete information prediction[J]. Information and Control, 2024, 53(1): 86 − 97.

[17] ZHANG H, ZHAO H, LIU R, et al. Collaborative task offloading optimization for satellite mobile edge computing using multi-agent deep reinforcement learning[J]. IEEE Transactions on Vehicular Technology, 2024, 73(10): 15483 − 15498.

[18] EISEN M, SHUKLA S, CAVALCANTI D, et al. Communication-control co-design in wireless edge industrial systems[C]//IEEE 18th International Conference on Factory Communication Systems. Piscataway, USA: IEEE, 2022: 1 − 8.

[19] LIMA V, EISEN M, GATSIS K, et al. Model-free design of control systems over wireless fading channels[J/OL]. Signal Processing, 2022; 197 [2025-07-21]. https://www.sciencedirect.com/science/article/pii/S0165168422000871. DOI: 10.1016/j.sigpro.2202.108540.

[20] ZHAO Z, LIU W, QUEVEDO D E, et al. Deep learning for wireless-networked systems: A joint estimation-control-scheduling approach[J]. IEEE Internet of Things Journal, 2024, 11(3): 4535 − 4550.

[21] JIANG C, CHEN W, WANG Z, et al. Deep reinforcement learning-based joint sequence scheduling and trajectory planning in wireless rechargeable sensor networks[J]. IEEE Sensors Journal, 2024, 24(8): 13699 − 13711.

[22] GASMI E, SID M A, HACHANA O. Nonlinear event-based state estimation using particle filter under packet loss[J]. ISA Transactions, 2024, 144: 176 − 187.

［23］LI S, YANG H C, HU F Y. Joint transmission mode selection and scheduling for AoI minimization in NOMA-capable WP-IoT networks: A deep transfer learning solution［J］. IEEE Transactions on Communications, 2025, 73(8): 5805 − 5816.

［24］WANG X, LAILI Y, ZHANG L, et al. Hybrid task scheduling in cloud manufacturing with sparse-reward deep reinforcement learning［J］. IEEE Transactions on Automation Science and Engineering, 2025, 22: 1878 − 1892.

［25］XU C, ZHANG X Y, YANG H H, et al. Optimal status updates for minimizing age of correlated information in IoT networks with energy harvesting sensors［J］. IEEE Transactions on Mobile Computing, 2024, 23(6): 6848 − 6864.

［26］LIN Y, XIAO L Q, TAO Y Y, et al. Multi-agent computing-energy-efficiency optimization in vehicular edge computing: Non-cooperative versus cooperative solutions［J］. IEEE Transactions on Wireless Communications, 2025, 24(7): 5461 − 5476.

［27］FOERSTER J N, FARQUHAR G, AFOURAS T, et al. Counterfactual multi-agent policy gradients［C］//Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence. New Orleans, USA: AAAI Press, 2018: 2974 − 2982.

［28］SAFAVINEJAD R, CHANG H H, LIU L J. Deep reinforcement learning for dynamic spectrum access: Convergence analysis and system design［J］. IEEE Transactions on Wireless Communications, 2024, 23(12): 18888 − 18902.

［29］NEDJAH N, FERREIRA G B, MOURELLE L M. Swarm robotics for collaborative object transport using a pushing strategy［J/OL］. Expert Systems with Applications, 2025; 271［2025-07-21］. https://www.sciencedirect.com/science/article/pii/S0957417425002325. DOI: 10.1016/j.eswa.2025.126610.

［30］SAGAR K V, JERALD J. Real-time automated guided vehicles scheduling with Markov decision process and double Q-learning algorithm［J］. Materials Today: Proceedings, 2022, 64(1): 279 − 284.

［31］HAN M Q, SUN X H, ZHAN W, et al. Multi-agent reinforcement learning based uplink OFDMA for IEEE 802.11ax networks［J］. IEEE Transactions on Wireless Communications, 2024, 23(8): 8868 − 8882.

［32］SUBHRAMOY M, DEBASHRI R, EISEN M, et al. L-NORM: Learning and network orchestration at the edge for robot connectivity and mobility in factory floor environments［J］. IEEE Transactions on Mobile Computing, 2024, 23(4): 2898 − 2914.

［33］LIU J, ANAVATTI S, GARRATT M, et al. A hierarchical mission planning system for multi-uncrewed ground vehicles using fast cost evaluation and ant colony optimisation［J/OL］. Information Sciences, 2024, 679［2025-07-01］. https://www.sciencedirect. com/science/article/ pii/S0020025524009435. DOI: 10.1016/j.ins.2024.121029.

［34］GALICKI M, BANASZKIEWICZ M, WEGRZYN M. Minimal-energy finite-time control of omni-directional mobile robots subject to actuators faults［J］. Nonlinear Dynamics, 2025, 113: 10061 − 10087.

［35］WANG B, LIU Z, LI Q, et al. Mobile robot path planning in dynamic environments through globally guided reinforcement learning［J］. IEEE Robotics and Automation Letters, 2020, 5(4): 6932 − 6939.

作者简介

李　鹏(2001—)，男，硕士生。研究领域为无线网络化控制系统。

张吟龙(1988—)，男，博士，副研究员。研究领域为多模态智能感知，机器视觉。

梁　炜(1974—)，女，博士，研究员。研究领域为状态估计及预测控制。