基于深度强化学习的无人机分布式集群网络规划研究

作者姓名 张靖宇

指导教师姓名、职称 张文博 副教授

申请学位类别 工学硕士

学校代码 10701　　学　号 20021210985
分 类 号 TP39　　密　级 公开

西安电子科技大学

硕士学位论文

基于深度强化学习的无人机分布式集群网络规划研究

作者姓名：张靖宇

一级学科：控制科学与工程

二级学科（研究方向）：模式识别与智能系统

学位类别：工学硕士

指导教师姓名、职称：张文博 副教授

学　　院：电子工程学院

提交日期：2023 年 4 月

Research on Distributed Cluster Network Planning of UAV Based on Deep Reinforcement Learning

A thesis submitted to

XIDIAN UNIVERSITY

in partial fulfillment of the requirements

for the degree of Master

in Control Science and Engineering

By

Zhang Jingyu

Supervisor: Zhang Wenbo Title: Associate Professor

April 2023

摘要

无线通信网络是实现无人机飞行自组网（Flying Ad-Hoc Network，FANET）的必要条件，通过无人机之间建立的低延时，高稳定性的无线通信链路，可以实现无人机之间的协同控制，区域网络覆盖等技术。近年来，随着多无人机集群控制技术的发展使得应用场景不断被拓宽，这对无人机间的无线网络链路规划带来了巨大挑战。其中基于分布式无人机的无线网络链路规划方法因其安全性高，稳定性强，部署成本低等特点而受到广泛关注。然而，在分布式组网下，无人机对环境的观测能力受限，为补偿这一缺点，在设计算法时需要综合考虑无人机动力学特征、通信网络框架、通信协议和无线电磁环境干扰等多个方面问题，使得算法设计困难，可拓展性不强。

本文围绕基于深度强化学习的分布式集群无人机网络规划方法展开研究，首先利用深度神经网络对高维特征具有较强拟合能力的特点，结合目前通信强化学习与离线强化学习方法，通过端到端的训练实现对分布式无人机组网下的特征拟合，同时生成通信协议，从而简化分布式网络规划算法的设计过程。其次，基于研究结论，进一步对通信网络框架进行设计，并根据网络框架对所设计的强化学习算法进行适应性改进。最后进行相应软硬件设计与实现。主要研究成果如下：

（1）针对目前分布式网络规划算法设计困难，只能针对少部分应用环境进行设计等特点，提出了一种基于深度强化学习的网络拓扑控制模型（Network Topology Control Model，NTC-Net）。通过参照通信强化学习方法设计NTC-Net的神经网络结构和对应的仿真环境。使用离线强化学习方法，通过集中式拓扑控制算法生成的数据集对NTC-Net进行端到端训练，使NTC-Net能够根据环境特征自动生成通信协议并进行无线通信链路决策，从而大大简化传统算法的设计流程，降低设计成本。实验结果表明，NTC-Net在无人机数量小于9时有一定的收敛效果，证明了该算法的可行性。

（2）针对FANET应用中的扩大网络覆盖范围、增强信号覆盖能力和优化网络性能的需求，在所提出的NTC-Net基础上，设计了一种无人机通信网络框架和ENTC-Net。首先，根据现有通信网络模型和NTC-Net的运行机制，设计了兼容互联和中继两种模式的通信网络框架。其次，在所设计的通信网络框架下，对NTC-Net的消息组合方式进行适应性改进，并对其训练方法进行设计，提出拓展网络拓扑控制模型（Extend Network Topology Control Model，ENTC-Net）。实验结果表明，在设计的通信网络框架下，ENTC-Net仍然保持较好的收敛效果，证明了框架与算法有效性。

（3）针对所设计的算法部署与实现问题，对相应的软硬件平台进行了设计。首先，从系统框架角度出发，对无人机、地面站和通信模块进行了设计。其次，基于UWB通信技术，结合所设计的通信网络和软硬件框架，对系统的软硬件进行了实现。通过

在系统上部署和测试NTC-Net和ENTC-Net，从硬件角度证明了本文所设计算法的可行性，并验证了所设计系统的有效性。

本文提出的算法，实现了集中式网络规划算法与分布式算法的端到端映射，为今后的无人机分布式网络规划算法设计提供了新的思路。

关 键 词：多无人机， 通信强化学习， 离线强化学习， 重要性采样， 拓扑控制

ABSTRACT

A wireless communication network is a necessary condition for realizing a Flying Ad-Hoc network (FANET). By establishing wireless communication links between UAVs with low delay and high stability, FANET can achieve cooperative control, regional network coverage, and other technologies. In recent years, the development of multi-UAV cluster control technology has broadened the application scenarios, which brings about specific challenges to the wireless network link planning among UAVs. Among the different methods, the wireless network link planning method based on distributed UAV has been widely concerned due to its high security, strong stability, and low deployment cost. However, this approach has limitations due to the UAV's limited ability to observe the environment. Thus, designing the algorithm requires comprehensive consideration of UAV dynamic characteristics, communication network framework, communication protocol, wireless electromagnetic environment interference, and other aspects. This makes the algorithm design difficult and challenging to scale.

This paper focuses on the planning method of a distributed cluster UAV network based on deep reinforcement learning. First, a deep neural network is used for its strong fitting ability on high-dimensional features. Combined with current communication reinforcement learning and offline reinforcement learning methods, end-to-end training can be used to achieve feature fitting under a distributed UAV network, and generate communication protocols. This simplifies the design process of the distributed network planning algorithm. Secondly, based on the research conclusion, the communication network framework is further designed, and the reinforcement learning algorithm is improved according to the network framework. Finally, the corresponding software and hardware design are implemented. The main research results are as follows:

1. A Network Topology Control Model (NTC-Net) based on deep reinforcement learning has been proposed due to the difficulty in designing current distributed network planning algorithms, which can only be designed for a limited number of application environments. The neural network structure and corresponding simulation environment of NTC-Net have been designed by referring to the communication reinforcement learning method. The off-line reinforcement learning method has been employed to conduct end-to-end training on

NTC-Net using the data set generated by the centralized topology control algorithm. Consequently, NTC-Net can automatically generate communication protocols and make wireless communication link decisions based on environmental characteristics, thereby significantly simplifying the design process of traditional algorithms and reducing design costs. Experimental results show that NTC-Net has certain convergence effect when the number of UAVs is less than 9, which verifies the feasibility of the algorithm.

2. Aiming at the requirements of expanding network coverage, enhancing signal coverage capability, and optimizing network performance in FANET applications, a UAV communication network framework, and ENTC-Net are designed based on the proposed NTC-Net. Firstly, according to the existing communication network model and the operation mechanism of NTC-Net, a communication network framework compatible with interconnection and relay modes is designed. Secondly, under the framework of the designed communication Network, the message combination mode of NTC-Net is improved, its training method is designed, and the Extend Network Topology Control Model (ENTC-Net) is proposed. The experimental results show that the convergence effect of ENTC-Net is still good under the designed communication network framework, which proves the effectiveness of the framework and algorithm.

3. In order to deploy and implement the designed algorithm, the corresponding software and hardware platform are designed. First, the UAV, ground station and communication module are designed from the perspective of system framework. Secondly, based on UWB communication technology, combined with the designed communication network and software and hardware framework, the software and hardware of the system are realized. By deploying and testing NTC-Net and ENTC-Net on the system, the feasibility of the algorithm designed in this paper is proved from the hardware point of view, and the effectiveness of the designed system is verified.

The algorithm proposed in this paper realizes the end-to-end mapping between the centralized network planning algorithm and the distributed algorithm, and provides a new idea for the design of distributed network planning algorithm of UAV in the future.

Keywords: Multi-UAV, Communication Reinforcement Learning, Off-line Reinforcement Learning, Importance Sampling, Topology Control

插图索引

表格索引

符号对照表

符号	符号名称
γ	折扣回报因子
U_t	智能体在t时刻的折扣回报
r_t	智能体在t时刻获得的奖励
a_t	智能体在t时刻的动作
s_t	智能体在t时刻的状态
π	智能体动作策略概率密度函数
μ	专家动作策略概率密度函数
Q_π	动作价值函数
V_π	状态价值函数
θ	策略网络参数
ω	动作价值网络参数
J^θ	以θ为参数的优化目标函数
$\boldsymbol{N}$	环境中智能体集合
N	环境中智能体数量
τ_i	智能体i的马尔科夫决策轨迹
τ	环境中所有智能体的马尔科夫决策轨迹
D_{KL}	KL散度
$D_{KL,\max}$	最大KL散度
h^i	智能体i产生的消息编码
h_t^i	智能体i在t时刻产生的消息编码
m^i	智能体i接收到的消息编码
m_t^i	智能体i在t时刻接收到的消息编码
o^i	智能体i的观测
o_t^i	智能体i在t时刻的观测
a^i	智能体i的动作
a_t^i	智能体i在t时刻的动作
$a_t^{i,*}$	针对智能体i在t时刻的专家决策
a_t^*	针对所有智能体在t时刻的专家决策
$\tilde{o}^i$	智能体i的等效观测
$\tilde{o}_t^i$	智能体i在t时刻的等效观测

$\tilde{O}^i$	智能体i的等效观测空间
$f_{i,j}^{g}(h)$	智能体i将消息编码h发送到智能体j的门控函数
$r_{\max}$	最大奖励
P_{a^*}	智能体策略分布中选择专家决策a^*的最小概率
ε	非专家策略的概率
$A_t^{i,j}$	智能体i在t时刻针对智能体j的动作的随机变量
A_t^i	智能体i在t时刻的动作的随机变量
P_p	多智能体策略与专家策略拟合度
$\bar{P}_p$	多智能体策略与专家策略平均拟合度
F_{sat}	速度限幅函数
$v_{\max}$	无人机飞行最大水平速度模长
T_{step}	神经网络在训练时的前向传播时间步
$\bar{T}_{step}$	神经网络在训练时的多次前向传播的平均时间步
y_{LP}	二阶巴特沃斯低通滤波模块
f_c	截止频率
f_s	采样频率

缩略语对照表

缩略语	英文全称	中文对照
AC	Actor-Critic	演员-评论家
ATOC	Attentional Communication Model	注意力交流模型
AWR	Advantage-Weighted Regression	优势加权回归
BCQ	Batch-Constrained Deep Q-learning	批量约束深度Q学习
BiCNet	Bidirectionally Coordinated Network	双向协调网络
CommNet	Communication Neural Net	通信神经网络
CTDE	Centralized Training Decentralized Execution	集中学习分散执行
DDPG	Deep Deterministic Policy Gradient	深度确定性策略梯度
DIAL	Differentiable Inter-Agent Learning	可微智能体间学习
Double-DQN	Double Deep Q-Network	双深度Q网络
DQN	Deep Q-Network	深度Q网络
ENTC-Net	Extend Network Topology Control Model	拓展网络拓扑控制模型
FANET	Flying Ad-Hoc Network	飞行自组网
FC	Full Connection	全连接网络
FlowComm	Flow Communication	流通信算法
GA-Comm	Graph Attention Communication	图注意力通信算法
GG	Gabriel Graph	加布里埃尔图
GNN	Graph Neural Network	图神经网络
GPS	Global Positioning System	全球定位系统
IC3Net	Individualized Controlled Continuous Communication Model	个性化控制连续通信模型
ICQ-MA	Multi-Agent Implicit Constraint Q-learning	多智能体隐式保守Q学习
IEEE	Institute of Electrical and Electronics Engineers	电气与电子工程师协会
IIR	Infinite Impulse Response	无限脉冲响应
IMU	Inertial Measurement Unit	惯性测量单元
IQL	Independent Q-Learning	独立Q学习

LSTM	Long Short-Term Memory	长短期记忆网络
MABCQ	Multi-Agent Batch-Constrained Deep Q-learning	多智能体批量约束深度Q学习
MAC	Media Access Control	介质访问控制
MARL	Multi-Agent Reinforcement Learning	多智能体强化学习
MCU	Micro Control Unit	微控制单元
MDP	Markov Decision Process	马尔可夫决策过程
MG	Markov Game	马尔可夫博弈
NE	Nash Equilibrium	纳什均衡
NTC-Net	Network Topology Control Model	网络拓扑控制模型
OOD	Out-Of-Distribution	超出分布范围
OSI	Open System Interconnection Reference Model	开放式系统互联通信参考模型
PPO	Proximal Policy Optimization	近端策略优化
QL	Q-Learning	Q学习
RIAL	Reinforced Inter-Agent Learning	强化智能体间学习
RNN	Recurrent Neural Network	循环神经网络
ROS	Robot Operating System	机器人操作系统
SARSA	State–Action–Reward–State–Action	状态-动作-奖励-状态-动作
TCA	Topology Control Algorithm	拓扑控制算法
TD	Temporal-Difference	时序差分
UWB	Ultra Wide Band	超宽带

目　录

第一章 绪论

1.1 研究背景及意义

随着无人机技术的发展，单体无人机控制技术逐渐趋于成熟，并逐渐向群体化发展。因集群无人机间相互协作的特点，相较单体无人机，集群无人机拥有更加广泛的应用领域。在未知环境探查方面，集群无人机间通过共享传感器信息[1]，在不同的空间位置同时进行侦查扫描[2]，大幅度提高了工作效率；在物品运输方面，多个无人机可以集中运输一个大于自身升力上限的载荷[3]，也可以同时分散运输多个载荷[4]，在全局角度上提高了无人机的整体载荷能力；在作战方面，多个无人机可以有不同的角色与分工，从而实现多功能一体化作战[5]，提升作战效率。而实现以上诸多应用的关键技术之一就是无人机间的网络通信，只有建立稳定，低延时的网络才能保证无人机间数据与控制命令的传输与接收。

目前对无人机的控制方式分为集中式，分散式和分布式三类。不同的控制方式决定了无人机不同的网络拓扑，其中分布式控制方式相较其他两种方式，无需部署中心节点对无人机群进行管理，有着去中心化，低成本，灵活可靠等优点，在目前受到了广泛研究。

而随着飞行自组网[6]（Flying Ad-Hoc Network，FANET）的提出，对无人机的组网有了新的要求。无人机因其高机动性，无法在群体中形成固定的链路拓扑。对于分布式无人机，尤其在大规模集群环境下，使用简单广播方式会占用大量信道资源。同时因为无人机观测能力受限，无法保证所建立的链路达到全局最优。如何设计无人机之间的通信算法与协议，保证无人机建立高稳定低延时的网络拓扑，对目前分布式无人机协同控制、提供网络覆盖[7]等应用的实现有着重要的意义。

然而目前无人机分布式网络规划算法的设计过程较为复杂，难以部署到实际应用。首先考虑到分布式无人机意味着每个无人机作为网络节点，对其他无人机及网络状态只存在局部观测。需要通过局部观测数据，通过对无人机的运动与控制特征建模，对无人机的通信规则和通信协议进行设计，从而来获得一个性能良好的算法[8-9]。其次考虑到无人机应用领域广，应用环境复杂，运动学特征多样，对不同的应用场景可能需要设计不同的网络规划算法，这将使得分布式组网方案应用成本上升。

随着深度强化学习的发展，尤其是通信强化学习领域，展现出了通过对分布式智能体的简单训练就能获得一定的自组织自管理的能力[10]，这一点正好符合目前分布式网络规划的研究重点。但同时因为通信强化学习算法起步较晚，且目前研究还停留在已经建立网络的基础上进行智能体间的通信。因此，通过通信强化学习实现自身通

信网络链路决策将是一个巨大的挑战。

综上所述，将深度强化学习技术与无人机自组网技术结合，可以极大程度的减低网络规划算法的设计难度与设计成本，并为进一步优化分布式网络规划算法性能提供了广阔空间。

1.2 国内外研究现状

1.2.1 无人机通信自组网拓扑控制算法

拓扑控制算法（Topology Control Algorithm，TCA）是一种控制无线通信网络拓扑结构的算法，也是实现无线网络规划的核心，被广泛应用于通信网络控制和智能体协同等领域。作为 FANET 的关键算法，TCA 保证了无人机之间稳定的通信链路和较低的通信延迟，为整个系统的稳定提供了必要性保证。

最早的拓扑控制算法使用了简单的拓扑生成机制，如相关邻居图[11]（Relative Neighborhood Graph，RNG）、 加布里埃尔图[12]（Gabriel Graph，GG），基于弹簧网格的控制算法[13]等。这些算法在链路拓扑优化方面体现出了良好的效果，算法要求使用节点的全局信息，故只能被用于集中式或分散式的控制。而随着无人机技术的发展与无人机应用领域的拓宽，集群数量与范围逐渐扩大，简单的拓扑生成算法无法满足应用需求，基于分布式的 TCA 开始受到人们的广泛关注。Leng 等人[14]基于 k-hop 聚类使用节点间的相对速度和距离选择通信网络簇头，并通过定时向外广播调整聚集成员，从而实现分布式拓扑控制。Leonov 等人模仿蚁群[15]和蜂群[16]的特点，提出了蚁群-蜂群临时路由（AC-BC Ad Hoc）和蜂群临时路由（Bee Ad Hoc），通过使用蚁群到实物源形成最短路径的原理和蜂群考虑最大花蜜量形成最短路径的原理，设计了一套路由协议，以此形成全局最优的路由拓扑。Farmani 等人[17]通过创建一个传感器管理和路径管理来记录无人机状态，通过估计无人机信息密度实现无人机的连接与聚类，提出了基于密度的有噪声应用的空间聚类（Density-Based Spatial Clustering of Applications with Noise，DBSCAN）算法。Lin 等人[18]通过对无人机位置进行网格化，并考虑空间最短距离设计了网格位置无中心最短路径（GPNC-SP）路由协议，针对快速移动的无人机，该方法实现了良好的网络拓扑规划效果。Arafat 等人[19]考虑了无人机在飞行过程中的定位噪声对拓扑控制的影响，在无人机对定位信息在网络中传播时，使用了元启发式优化算法来实现定位误差最小化。对于无人机节点变换导致网络需要重新规划的时间问题，Wang 等人[20]提出了一种动态自适应协议，极大程度的降低了链路拓扑变换时的路由更新时间。Yan 等人[21]尝试将无人机集群控制与强化学习结合，提出了有经验重放的连续演员-评论家算法（Continuous Actor–Critic with Experience Replay，CACER），并通过设计一套半实物仿真系统证明了算法的可行性，之后又基

于注意力的种群不变网络[22]（Attention-Based Population-Invariant Network，APINet）设计了可容纳更多无人机的强化学习算法，并测试该算法对于动态数量蜂群无人机控制的稳定性。但两种算法仅实现了领导者到跟随者的下行链路控制。

总的来说，目前无人机 TCA 的挑战主要集中在：（1）通信网络稳定性需要进一步提升。无人机作为高机动性的网络节点，其链路受到无线发射功率和环境干扰的制约，需要通过构建合理的链路拓扑保证网络连通。（2）模型维度不高。在设计算法时，需要先对多无人机系统的通信网络进行建模，为方便数学推导，一般仅针对要解决的主要问题进行建模。而对于其他问题，在建模过程中则进行简化或直接忽略。这使得根据模型设计的算法针对性较强，对不同环境适应能力差。（3）协议内容复杂，系统多样化，可拓展性不强。基于模型维度不高的问题，所设计的算法和协议针对不同环境有所不同，这使得算法与协议从特征上无法统一，导致不同的场景需要部署不同的系统，极大的提升了系统部署成本。

1.2.2 强化学习

强化学习是一种让智能体通过自主选择动作与环境互动的算法，通过执行动作并接收环境的反馈来调整自身策略，以实现最优价值策略的目标。早在20世纪60年代，强化学习的概念就已经被提出，而随着Q学习[23]（Q-Learning，QL）的提出，强化学习的概念也开始受到人们的关注。目前，随着神经科学、大数据的发展和计算机硬件算力的快速提升，深度神经网络开始受到广泛研究与应用。Mnih等人[24]将强化学习中的价值函数使用神经网络进行近似，提出了深度Q网络（Deep Q-Network，DQN）算法，并使用Atari系列游戏进行测试，通过其优越的性能，展现出了深度强化学习发展的可能性。

在DQN提出之后，Silver等人[25]通过推导证明了确定性策略梯度算法的存在。在此基础上，将价值梯度和策略梯度同时使用神经网络进行近似，结合演员-评论家（Actor-Critic，AC）算法[26]，Lillicrap等人[27]提出了深度确定性策略梯度（Deep Deterministic Policy Gradient，DDPG）算法，证明了基于策略的深度强化学习方法的可行性。

事实上，基于深度神经网络的强化学习方法还存在收敛性和收敛速度的一些问题。例如DQN中使用的时序差分（Temporal-Difference，TD）误差进行训练时，TD误差会在价值函数对未来价值估计过高时不断累积，从而出现过估计现象，使得在智能体通过最优价值选择动作的过程出现偏差。在此问题上，Hasselt等人[28]提出了双深度Q网络（Double Deep Q-Network，Double-DQN）方法，通过使用两个Q网络，一个用于动作选择，另一个用于TD误差估计来缓解DQN的过估计问题。同样的，在AC算法中，因为Critic是对于动作价值函数的拟合，同样存在因为动作价值高估而导致策略函数

出现决策偏差的过估计现象。而对于AC这一类基于策略的学习方法，可以采用基线（Baseline）技术降低策略梯度下降过程中的方差，使得估计更加准确，收敛更快。常用的在优势演员-评论家[29]（Advantage Actor-Critic，A2C）和REINFORCE算法中就使用了状态价值函数作为Baseline来降低对策略函数梯度的方差。其他的关于收敛性的研究还有Hessel等人[30] 证明了在训练过程中使用多步决策计算得到的梯度，比起单步决策计算得到的梯度，具有更好的收敛性。

除了上述降低梯度方差的方法外，还可以通过预先录制数据集的方法解决深度强化学习在收敛性方面的问题，以减少价值过估计或错误估计的情况。稀疏奖励环境是深度强化学习的一大挑战，智能体在该环境下难以通过自身决策获得不同奖励来更新参数，从而容易陷入局部最优甚至无法收敛。为解决此问题，Rajeswaran等人[31]通过对动作进行预先录制，并将录制的动作用于策略梯度计算，实现对仿真环境中机械手的训练，相对于未预先录制动作的训练，该方法取得了良好的收敛结果。深度强化学习收敛速度在机械臂操控[32]，自动驾驶[33]等涉及安全问题的领域也是难以接受的，在实际环境的训练过程中，智能体错误的决策可能会导致灾难性的后果。在此基础上，基于行为克隆的离线强化学习被提出。而离线强化学习面临的最大问题是，因为离线策略和实际策略差距过大而导致的超出分布范围（Out-of-Distribution，OOD）现象。针对此问题，Schulman等人[34]通过对策略梯度的重要性采样过程引入KL散度约束，提出了近端策略优化（Proximal Policy Optimization，PPO）算法，并证明了其良好的性能。Fujimoto等人[35]针对基于价值的离线学习进行了研究，提出了批量约束深度Q学习（Batch-Constrained Deep Q-learning，BCQ），证明在数据足够好的情况下，可以只通过数据就可以学到与行为策略一致甚至更好的效果。Peng等人[36]提出了一套优势加权回归（Advantage-Weighted Regression，AWR）方法，该方法将智能体的一种现实轨迹作为策略梯度输入，从而实现对智能体的训练，为实现基于策略的离线强化学习框架提供技术支撑。Kumar等人[37]提出了保守Q学习（Conservative Q-Learning，CQL）算法，该算法针对在离线学习中，智能体策略分布与数据集策略分布不同而导致价值函数出现的过估计现象，通过在训练中加入正则化项，使其变得更加保守，提高了算法的收敛速度。除此之外，也出现了针对智能体策略进行优化的保守离线模型的策略优化（Conservative Offline Model-Based Policy Optimization，COMBO）算法[38]，而Kostrikov等人[39]提出的隐式Q学习算法对状态价值函数进行了重新构造，大大简化了之前对价值或策略函数的正则化方法，并取得了良好的效果。

总的来说，目前强化学习的挑战主要集中在：（1）复杂任务下的奖励函数设计困难。在复杂任务下，智能体状态空间较大，维度较高，通过简单的奖励函数难以限制和引导智能体行为，容易导致训练过程出现对某个动作状态的过估计或错误估计，从而无法得到理想的训练效果。（2）数据集有限，训练过程中的收敛方向难以控制。

通过预先录制的数据集对智能体进行训练，可以在一定程度上解决奖励函数设计困难问题，但训练过程中智能体会不可避免地进入数据集中少有或没有的状态，使得训练无法进行。故对于如何制作数据集和设计训练过程的约束是目前研究的重点。

1.2.3 多智能体强化学习方法

多智能体强化学习（Multi-Agent Reinforcement Learning，MARL）方法是单智能体强化学习方法的延伸，相较于单智能体强化学习，多智能体强化学习除了存在智能体与环境的交互外，还增加了智能体与智能体之间的博弈，最常见的智能体间的博弈概念是纳什均衡[40]（Nash Equilibrium，NE）。最早随着Q-Learning的发展，Gupta等人[41]提出了独立Q学习（Independent Q-Learning，IQL）的框架，并将DQN、DDPG、AC等算法参照IQL的框架将其应用于多智能体环境中，证明了由单智能体方法直接拓展到多智能体方法的可行性。在之后由于深度强化学习出现，MARL也同单智能体一样开始了爆发式的发展[42]，较有代表性的工作是Tampuu等人[43]将DQN和IQL结合在一起，用于两个智能体进行乒乓游戏的博弈，并取得了良好的效果。同样的，对于DDPG算法，Lowe等人[44]提出了多智能体深度确定性策略梯度（Multi-Agent Deep Deterministic Policy Gradient，MADDPG）算法，该算法采用了AC框架，对每个智能体都有一个评论家，评论家通过获取全局信息对动作价值进行估计，并用于更新智能体的演员网络。

事实上，对于多智能体环境，智能体除了存在对环境状态的局部观测以外，还存在相互之间的交流，即通信行为。Foerster等人[45]首次提出了多智能体通信的概念，提出了强化智能体间学习（Reinforced Inter-Agent Learning，RIAL）算法和可微智能体间学习（Differentiable Inter-Agent Learning，DIAL）算法。两种算法在整个多智能体环境下通过使用长短期记忆网络（Long Short-Term Memory，LSTM）和智能体间参数共享，不仅可以减少神经网络参数的数量，还能借助LSTM的记忆能力使得不同智能体表现出不同的行为特征。通过使用集中学习分散执行（Centralized Training Decentralized Execution，CTDE）框架实现了智能体间的协作，证明智能体可以通过学习形成自己的通信协议。Sukhbaatar等人[46]提出的通信神经网络（Communication Neural Net，CommNet）结构则使用了连续量作为通信数据，并在通信时通过计算平均值来组合数据，相比RIAL和DIAL可以在环境中容纳更多的智能体。Peng等人[47]提出了双向协调网络（Bidirectionally Coordinated Network，BiCNet），采用了AC和CTDE框架，并使用智能体间参数共享，在星际争霸游戏中获得了良好的效果。

以上通信网络在通信过程中均使用广播方式，但实际应用中可能会涉及到通信距离，通信带宽等限制，并且智能体也不需要接收与自己无关的通信数据。为解决该问题，后续的研究中提出了门控机制。Jiang等人[48]通过在智能体通信过程中加入注意力

机制来决定是否发送信息，提出了注意力交流模型（Attentional Communication Model，ATOC），该算法在大型合作场景中可以使智能体选择自身是否为发起者，并由一个LSTM对信息进行组合，提升了智能体间的通信效率和训练效果。Singh等人[49]通过在每一个智能体上加入一个消息门，由智能体自己控制消息的发送，提出了个性化控制连续通信网络（Individualized Controlled Continuous Communication Mode，IC3Net）。该算法的消息门使用全连接网络（Full Connection，FC）实现，使用智能体中的LSTM网络的隐藏层数据作为通信消息。IC3Net不仅证明了网络在合作环境中的优秀性能，而且消息门控的存在还使得算法可以在合作-竞争的混合环境中得到应用。Kim等人[50]则考虑了通信信道带宽问题，提出了调度网络（SchedNet），其通过单独训练一个消息调度器来对通信消息的发送进行控制，使得智能体产生的离散消息的发送可以被控制。

对于一些更加复杂的通信情形，如在智能体之间建立通信管道的情况，图注意力通信[51]（Graph Attention Communication，GA-Comm）和流通信[52]（Flow Communication，FlowComm）算法通过建立一个连接图的方式控制智能体间进行通信，其中GA-Comm使用图注意力机制对无向图进行构建，并通过图神经网络（Graph Neural Network，GNN）对消息进行组合，而FlowComm则使用显式的方式建立有向图，可以使得对智能体的控制更加精细，收敛效果更好。同样的，Xiao等人[53]通过使用图注意力机制自适应地调整智能体与周围邻居权重，从而在通信受限的环境下实现了无人机的协同控制。但三者的连接图都由一个全局调度进行管理，因此无法完全实现智能体分布式工作。

对于离线强化学习同样可以拓展到MARL中。Yang等人[54]首次基于BCQ提出了多智能体隐式保守Q学习（Multi-Agent Implicit Constraint Q-learning，ICQ-MA）算法，并在多智能体环境下进行了拓展，同时提出了状态-动作-奖励-状态-动作（State–Action–Reward–State–Action，SARSA）形式的乘子，并证明了该乘子的收敛性。Jiang和Lu[55]考虑了多智能体下不同的状态转移概率可能会导致价值函数估计偏差，使得策略函数无法正确收敛。在此基础上，作者基于BCQ提出了多智能体批量约束深度Q学习（Multi-Agent Batch-Constrained Deep Q-Learning，MABCQ）算法，并实现了完全去中心化。

总的来说，目前MARL的挑战主要集中在：（1）多智能体间通信机制尚未有统一框架。目前通信强化学习还处在发展阶段，针对不同问题的研究所提出的神经网络结构、训练方法等都有所不同，导致智能体间的通信机制及其实现难度也有所不同。（2）多智能体下离线强化学习实现困难。多智能体环境相较于单智能体环境，其状态空间维度增加，这对离线强化学习训练所需要的数据集和梯度约束提出了更高的要求，故难以使用离线强化学习方法解决多智能体问题。

1.3 文主要工作和章节安排

综上所述，目前分布式集群拓扑控制方法仍然存在建模、协议设计复杂，部署成本高等缺点。本文基于对主流通信强化学习框架的总结，使用全局拓扑控制算法生成离线数据，设计无人机通信决策神经网络并对其进行训练，实现全局拓扑控制算法在分布式环境下的映射。同时设计相关软硬件系统，实现从算法训练到部署的全部流程，为算法的进一步研究与实现提供平台。本文的结构框图如图1.1所示：

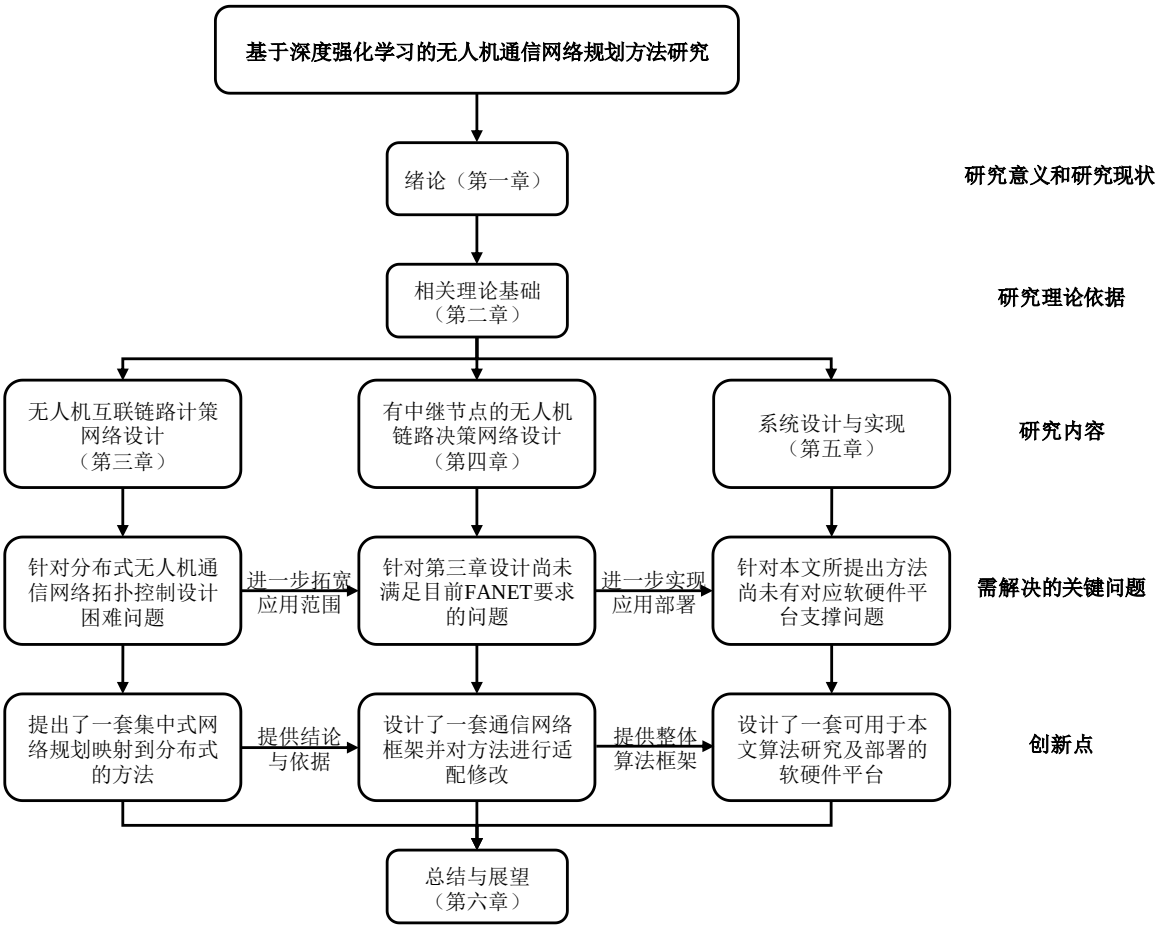


图1.1 论文结构框图

如图1.1，本文创新性地采用以下研究方法，为章节安排提供依据：

（1）多个子领域调查研究。本文主要基于深度强化学习的无人机分布式集群网络规划展开研究，为提供新的分布式无人机网络规划方法，第二章对现有深度强化学习的多个子领域进行调研。利用各子领域所解决问题的重点偏向不同的特点，对各子领域有效的方法进行总结，以此为依据在第三章和第四章进行算法设计。

（2）系统性实验研究。本文为面向集群无人机应用的研究，为保证算法有效性的证明充分，故在通过实验结论对算法性能进行测试分析的同时，也对支持算法运行

的系统框架进行逐步完善，从而进一步对完善后的系统框架进行整体实验，以此构成第三、四、五章的递进关系。在完善设计的同时，也得到了对算法有效性的更加充分的证明。

（3）仿真与实物耦合。本文主要以解决集群无人机分布式网络规划的实际问题为目标，需要考虑算法的最终实现问题，为保证实验的安全性，同时最大限度的减小算法训练与实际部署间的差异。在第三章和第四章中，算法与系统设计过程中始终保持仿真环境与实物之间联系。最终由第三、四章的仿真实验预测效果，并由第五章的实物系统展现算法效果，达到根据实物系统效果更加精准的优化算法性能的目的。

本文的章节安排如下：

第一章首先介绍了本文研究开展的背景，并对现阶段无人机分布式拓扑控制和强化学习研究进行总结，从而引出本文基于强化学习解决无人机网络拓扑规划问题的研究内容与章节安排。

第二章对相关的强化学习理论进行了介绍并进行规律性总结，为后续章节中的算法与系统设计提供理论和框架方面的论据。通过介绍强化学习的概念，引出深度强化学习的子领域。在子领域中，首先通过单智能体强化学习和多智能体强化学习对深度强化学习训练方法的规律进行总结；其次对现有有主流的通信强化学习神经网络结构和通信机制进行总结；最后对文中所使用的离线强化学习方法进行了介绍与梯度公式推导。

第三章研究无人机互联环境下的深度强化学习通信决策算法。首先，通过将无人机分布式拓扑控制模型抽象为通信强化学习模型，利用抽象出的模型提出章节解决的问题以及重点。其次，根据第二章总结的现有模型与理论，设计了一套离线通信强化学习算法，并介绍离线数据来源及其生成方法。再次，根据提出的问题与设计的神经网络结构，设计训练时的仿真环境。最后，在仿真环境中对算法进行训练实验，证明了所设计算法的有效性。

第四章基于第三章设计的算法，基于有无线中继节点的环境，对网络通信框架进行设计，并对通信决策算法进行适应性改进，实现网络规划算法的整体设计。首先，根据第三章算法的结论，考虑算法对中继节点的兼容，对通信网络框架进行设计。其次，根据所设计的框架，对第三章的算法在通信消息组合方式方面进行适应性改进。为提升改进后算法的训练效果，本章提出了一套使用低通滤波约束和引导神经网络收敛的训练方法。最后，根据本章设计，对第三章设计的仿真环境进行了拓展，并基于以上设计对改进算法进行仿真实验，验证改进效果并测试算法性能。

第五章基于对网络规划算法的整体设计，对系统的软硬件进行了设计与实现。首先基于算法及通信网络框架需求，对硬件及其连接关系进行了设计。其次，基于所设计的硬件，考虑软件复用，将软件框架分为上位机与下位机两部分进行设计。最后，

使用超宽带（Ultra Wide Band，UWB）技术作为通信手段，制作一套实物系统，验证了本文提出算法在实物平台上的可行性。

第六章对本文所研究的工作进行了总结，针对所设计的系统以及实验现象，分析当前不足并总结值得改进的地方，展望今后的研究方向。

第二章 相关理论基础

2.1 引言

强化学习作为一种高效的智能决策技术，在许多领域取得了显著成功。深度强化学习作为强化学习的一个重要分支，其通过深度神经网络拟合高维特征，使智能体能够在更复杂的环境中进行学习和决策。深度强化学习也因此受到广泛研究，并与多智能体强化学习、通信强化学习和离线强化学习等领域交叉产生了更多子领域。

本章将对深度强化学习相关理论进行规律性总结，同时为之后章节中的算法及其实验设计和相关公式推导提供理论支撑。首先，介绍强化学习的理论基础。其次，通过单智能体强化学习和多智能体强化学习对深度强化学习的理论基础进行介绍。再次，总结当前主流的通信强化学习神经网络结构和通信机制的规律，为基于通信强化学习的神经网络结构设计提供理论依据。最后，介绍基于重要性采样的离线强化学习理论并推导其梯度公式，为之后通信强化学习神经网络的训练公式推导及文章的实验设计提供支撑。

2.2 强化学习基础

强化学习是一个智能体与环境交互的过程，该过程通常被建模为马尔可夫决策过程（Markov Decision Process，MDP）。马尔可夫决策过程由元组$(\boldsymbol{S},\boldsymbol{A},\boldsymbol{P},\boldsymbol{R},\gamma)$表示，其中$\boldsymbol{S}$表示智能体的状态空间，$\boldsymbol{A}$表示智能体的动作空间，$\boldsymbol{P}$表示智能体在状态$s\in\boldsymbol{S}$下选择动作$a\in\boldsymbol{A}$而使得状态转移到$s'\in\boldsymbol{S}$的概率。$\boldsymbol{R}$表示智能体在状态$s\in\boldsymbol{S}$下选择动作$a\in\boldsymbol{A}$使状态转移到$s'\in\boldsymbol{S}$后获得的奖励。$\gamma\in[0,1]$表示折扣回报因子，其中智能体在$t$时刻的折扣回报如下：

$$U_t = R_t + \gamma U_{t+1} \tag{2-1}$$

其中，U_t表示在t时刻获得的回报，包括当前奖励R_t和未来的回报U_{t+1}与折扣回报因子γ的乘积。U_t，U_{t+1}和R_t均为随机变量。

在MDP中，一个处在s_t状态下的智能体通过自身决策π决定当前状态下的动作a_t。故有智能体动作策略概率密度函数（简称策略函数）$\pi(a_t|s_t)\in[0,1]$，存在随机变量$A_t\sim\pi(\cdot|s_t)$，对随机变量A_t进行蒙特卡洛抽样，获得样本a_t并执行该动作，此时智能体状态随机变量$S_t\sim\boldsymbol{P}(\cdot|s_t,a_t)$根据其概率分布转移到$s_{t+1}$状态，并获得该状态下的奖励$r_t=R(s_t,a_t,s_{t+1})$。整个过程如图2.1所示，其中$s'$表示从$s$通过动作$a$在环境中转移

得到的新状态。

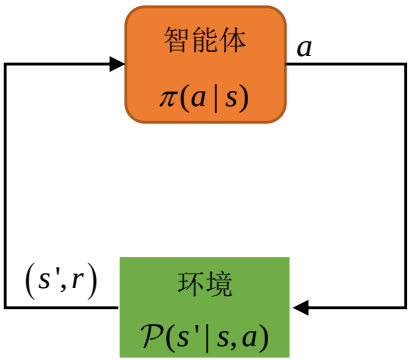


图2.1 强化学习状态转移关系示意图

为方便建模，根据 MDP，引入了动作价值函数Q_π与状态价值函数V_π。其中动作价值函数表示在状态s_t下选择动作a_t使智能体所能获得的当前回报的数学期望，即当前奖励和未来所有回报的均值，用公式表示如下：

$$Q_\pi(s_t,a_t)=\mathrm{E}\left[U_t|S_t=s_t,A_t=a_t\right] \tag{2-2}$$

强化学习的优化过程是通过特定的迭代方法寻找最优的策略π，从而使得智能体在不同状态选择最优动作并执行，保证自身获得全局最大回报，用公式表示如下：

$$Q^*(s_t,a_t)=\max_\pi Q_\pi(s_t,a_t) \tag{2-3}$$

状态价值函数则是在动作价值函数的基础上进一步对随机动作A求数学期望，得到智能体在当前状态下执行所有可能的动作而得到的回报的均值，以方便对策略函数提出优化目标，用公式表示如下：

$$V_\pi(s_t)=\mathrm{E}_A\left[Q_\pi(s_t,A)\right] \tag{2-4}$$

以上理论构成了强化学习的基础，也是之后强化学习及其分支领域的核心。

2.3 深度强化学习

强化学习的迭代过程是寻找最优策略函数的过程，而对策略函数和动作价值函数的建模除了基于模型以外，也可以使用深度神经网络对其进行拟合，从而产生了深度

强化学习方法。深度强化学习的出现使得强化学习领域得到了快速发展，并衍生出了多个子领域，以下将对文章相关的领域及理论进行详细介绍，所介绍领域关系如图 2.2 所示。

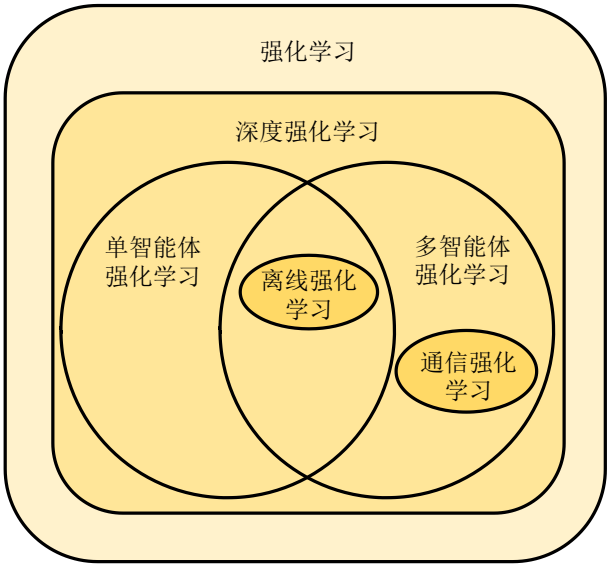


图2.2 强化学习领域关系示意图

2.3.1 单智能体强化学习基础

在单个智能体中，其动作决策可以通过策略函数提供的概率密度估计对动作进行蒙特卡洛抽样；也可以通过动作价值函数进行决策，即在任意时刻t根据动作价值函数找到动作a_t使得$Q_\pi(s_t,a_t)$达到最大值。因此，单智能体的强化学习优化方法也被分为基于价值的方法和基于策略的方法。

基于价值的方法可将动作价值函数表示为$Q_\pi(s,a;\omega)$，其中ω是神经网络参数。算法实现的目标是通过调整参数ω来实现对动作价值函数$Q_\pi(s_t,a_t)$的估计，该过程的公式表示如下：

$$\left(\mathcal{T}^\pi Q_\pi\right)(s_t,a_t)=Q_\pi(s_t,a_t)+E_S\left[r_t+\gamma Q_\pi(S,a_{t+1})-Q_\pi(s_t,a_t)\right] \tag{2-5}$$

其中$\mathcal{T}^\pi$是贝尔曼算子。但在实际情况中，通常无法获得对智能体在执行动作a_t后可能会转移的状态S，所以在实际训练中一般采用蒙特卡洛抽样来作为对下一状态的均值估计，并通过建立损失函数来实现价值函数对未来回报的估计。损失函数表示如下：

$$\mathcal{L}_{Value}(\omega)=\frac{1}{2}\left[Q_\pi(s_t,a_t;\omega)-y_t\right]^2 \tag{2-6}$$

其中 y_t 为t时刻下价值函数对回报的估计误差，用公式表示如下：

$$y_t = r_t + \gamma Q_\pi \left(s_{t+1}, a_{t+1}; \omega \right) \tag{2-7}$$

式(2-7)被称为 TD 误差，较有代表性的算法为 SARSA 算法。在式(2-7)中，若不再通过对随机动作 A进行蒙特卡洛抽样得到 a_{t+1}，而是直接选择概率最大的动作进行 TD 误差计算，那么可以推导得到工程中最常用的 DQN 算法。用公式表示如下：

$$y_t = r_t + \gamma \max_a Q_\pi \left(s_{t+1}, a; \omega \right) \tag{2-8}$$

基于策略的方法可将策略函数表示为 $\pi \left(a \mid s; \theta \right)$，其中 θ 为神经网络参数。算法实现的目标是通过调整参数 θ 来获得最优策略函数，通过最优策略函数对下一步动作进行选择，保证智能体获得最大回报，即获得最大的状态价值。故优化目标函数表示如下：

$$J^\theta \left(\theta \right) = \mathrm{E}_S \left[V_\pi \left(S; \theta \right) \right] \tag{2-9}$$

综合式(2-2)和(2-4)，对式(2-9)其求梯度得到公式如下：

$$\nabla J^\theta \left(\theta \right) = \mathrm{E}_{A \sim \pi(\cdot \mid S), S} \left[Q_\pi \left(A, S \right) \nabla \log \pi \left(A \mid S; \theta \right) \right] \tag{2-10}$$

但在实际训练中，获取期望的难度较大，考虑对 $\nabla J^\theta \left(\theta \right)$ 进行蒙特卡洛抽样可以获得对 A 和 S 的无偏估计，故在实际训练中一般采用的梯度上升计算公式如下：

$$g(a_t, \theta) = Q_\pi \left(a_t, s_t \right) \nabla \log \pi \left(a_t \mid s_t; \theta \right) \tag{2-11}$$

$$\theta_{t+1} = \theta_t + \beta g(a_t, \theta_t) \tag{2-12}$$

基于该方法，产生了较为代表性的 REINFORCE 算法和部分可观察马尔可夫决策过程（Partially Observable Markov Decision Process，POMDP）算法。而经典的 AC 算法是将动作价值的估计 $Q_\pi \left(s, a; \omega \right)$ 作为评论家（Critic），将策略函数 $\pi \left(a \mid s; \theta \right)$ 作为演员（Actor），其中 Q_π 通过式(2-6)和式(2-7)进行梯度下降更新，π 的更新则是将 Q_π 代入式(2-11)和式(2-12)进行。

2.3.2 多智能体强化学习基础

多智能体强化学习是对单智能体强化学习的拓展，除了单智能体与环境交互以外，还引入了智能体之间的相互通信与博弈。其在单智能体基础上，将MDP拓展为马尔可夫博弈（Markov Game，MG）。其元组表示为$\left(\boldsymbol{N},\boldsymbol{S},\{\boldsymbol{A}^i\}_{i\in\boldsymbol{N}},\boldsymbol{P},\{\boldsymbol{R}^i\}_{i\in\boldsymbol{N}},\{\boldsymbol{O}^i\}_{i\in\boldsymbol{N}},\gamma\right)$。其中$\boldsymbol{N}=\{1,\cdots,N\}$表示智能体的集合，$\{\boldsymbol{A}^i\}$和$\{\boldsymbol{R}^i\}$分别表示第$i$个智能体的动作空间和奖励空间，$\{\boldsymbol{O}^i\}$表示智能体$i$对于环境的局部观察，$\boldsymbol{S}$表示整个多智能体所处环境下的全部状态空间，$\boldsymbol{P}$表示所有智能体的状态转移概率。

MG下的状态转移过程如图2.3所示，在系统角度下可以认为是智能体与环境的交互过程，仍可用MDP描述。但是在环境中的单个智能体角度下，MDP不完全适用。一方面，智能体对环境的观测受到限制，只能通过自己的局部观测$o^i\in\boldsymbol{O}^i$来进行动作决策，从而限制了其对环境的全局认知。另一方面，智能体的状态转移不仅受到自身动作的影响，还受到其他智能体的动作影响。

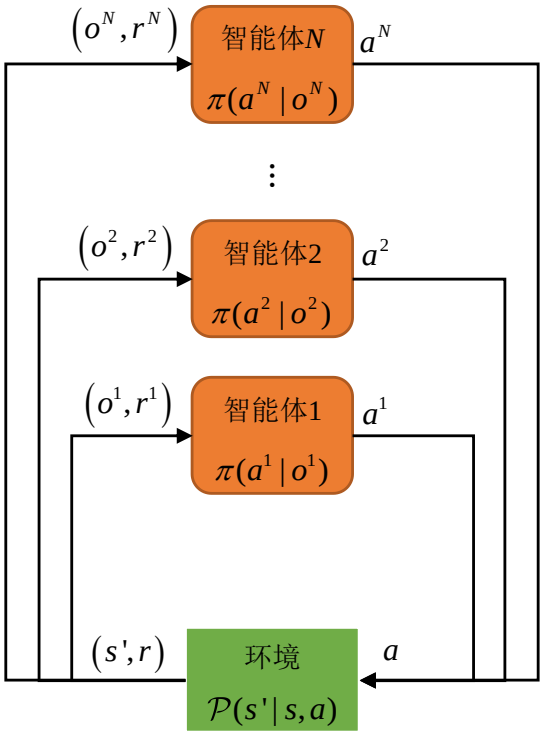


图2.3 马尔可夫博弈下的状态转移关系示意图

因此，在所有智能体同时进行动作决策和状态转移的情况下，需要引入纳什均衡定义多智能体的优化目标。纳什均衡的描述如下：

假设对于环境中所有智能体$\boldsymbol{N}$在任意状态$s\in\boldsymbol{S}$下，都存在最优的策略$\pi^*=\left(\pi^{1,*},\cdots,\pi^{N,*}\right)$，满足关系如下：

$$V^{i}_{\pi^{i,*},\pi^{-i,*}}(s) \geq V^{i}_{\pi^{i},\pi^{-i,*}}(s), \forall i \in \boldsymbol{N} \tag{2-13}$$

其中$-i$表示除i以外的其他智能体，智能体i在状态s下的状态价值$V^{i}_{\pi^{i},\pi^{-i}}(s)$表示如下：

$$V^{i}_{\pi^{i},\pi^{-i}}(s_t) = \mathrm{E}\left[U^{i}_{t} \mid A^{i}_{t} = a^{i}_{t}, S^{i}_{t} = s^{i}_{t}\right] \tag{2-14}$$

由式(2-13)和式(2-14)可知，在纳什均衡下，在其他智能体都选择最优策略的情况下，智能体自身也将选择最优策略，以达到最优的结果。

2.3.3 通信强化学习

通信强化学习是多智能体强化学习的分支，其通过控制智能体间进行数据交互，实现智能体间更高效的合作。参考目前现有的通信强化学习框架[10]，可将大部分框架其抽象为如图 2.4 所示结构。由图 2.4 可知，通信强化学习使得智能体之间可以通过发送消息进行通信，使得智能体之间的耦合性更强。换句话说，多智能体强化学习过程中加入通信后，智能体可以通过将自身状态和观测作为通信消息发送给其他智能体，接收到消息的智能体做出的决策将更加有利于全局合作。事实上，Singh 等人[56]通过实验证明了在竞争环境中智能体会选择相互间不通信。

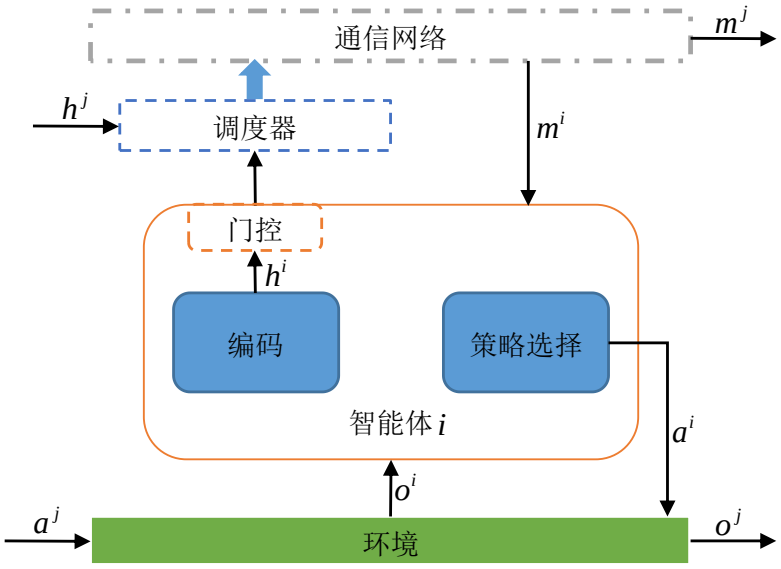


图2.4　通信强化学习整体框架示意图

从整体框架上看，以智能体i为例，在一个时间步内，智能体有两个输入，即对环境存在局部观测o^i和来自通信网络的消息m^i。智能体会产生两个输出，一个是自身策略选择出的动作a^i，它将被智能体在环境执行并使智能体发生状态转移；另一个是智能体隐藏状态h^i，它将被提交到通信网络并发送到其他智能体$j(j \neq i)$。

（1）通信机制

通信机制的设计是通信强化学习研究的重点之一。由图 2.4 所示，虚线框中的门控和调度器对于通信系统都具有对消息进行筛选的功能，故在实际应用中只会选其一或者都不用。如 IC3Net、ATOC、门控-演员评论家消息学习[57]（Gated Actor-Critic Message Learner，Gated-ACML）等通过学习自身门控来控制消息是否发送；而 SchedNet、FlowComm、GA-Comm 等则是通过一个全局的调度器来控制消息是否送入网络；较早的 RIAL、DIAL、CommNet、BiCNet 等没有使用二者，而是直接将消息送入通信网络。

（2）消息组合方式

在消息被送入通信网络后，网络会对消息进行组合，而消息的组合方式也是通信强化学习研究的另一个重点。在 CommNet，IC3Net 中采用了求消息均值的方式，用公式表示如下：

$$m^i = \frac{1}{J}\sum_{j\in \boldsymbol{J}} f_{i,j}^g\left(h^j\right) \tag{2-15}$$

其中 $\boldsymbol{J}$ 表示当前时间步下存活的智能体集合，J 表示存活的智能体个数，$f_{i,j}^g$ 表示相对智能体 i 下的智能体 j 的门控函数，即 $f_{i,j}^g$ 的作用是决策从智能体 j 的消息是否要由智能体 i 接收。对于上述两种方法，都有当 $i=j$ 时，$f_{i,i}^g = I$，表示智能体之后会接收到自己发出的消息，而 CommNet 因为不存在门控和调度，所以 $f_{i,j}^g \equiv I$。

对于 RIAL、DIAL 和 SchedNet 等算法使用了对消息进行向量维度方向的拼接，这一方法虽然会使得智能体对于接收消息 m^i 的维度随着智能体数量增加而呈指数增加，但也保证了数据的完整性，不会受到其他智能体消息的干扰。

而对于 ATOC、GA-Comm 和 BiCNet 等算法则是通过训练一个特殊的神经网络对消息进行组合。全局的消息组合神经网络的设计虽然使得实际应用部署变得困难，但是从文章的实验结果来看，该方式有着相对其他两种消息组合方式更加突出的效果。

（3）消息接收与决策

对于智能体来说，如何将接收到的消息进行组合并进行动作决策，是通信强化学习研究的最后一个重点。为解决这个问题，研究的重心放在了如何设计合理的网络结构上。在上面提到的大部分研究中，都设计了深度神经网络作为智能体的编码器，由 FC 或循环神经网络（Recurrent Neural Network，RNN）构成。若使用 RNN，智能体将获得额外的记忆能力来辅助决策，但也将导致算法在梯度下降过程中路径复杂，梯度难以被准确计算，从而影响训练收敛效果。

编码器产生的信息编码在大部分文献中被称为隐藏状态。而现有的研究中，在智

能体的观测 o^i、接收的消息 m^i 和自身的隐藏状态 h^i 的耦合方法上体现出了多样性，考虑与本文的相关性，这里仅对主流规律进行总结。如图 2.5 所示：

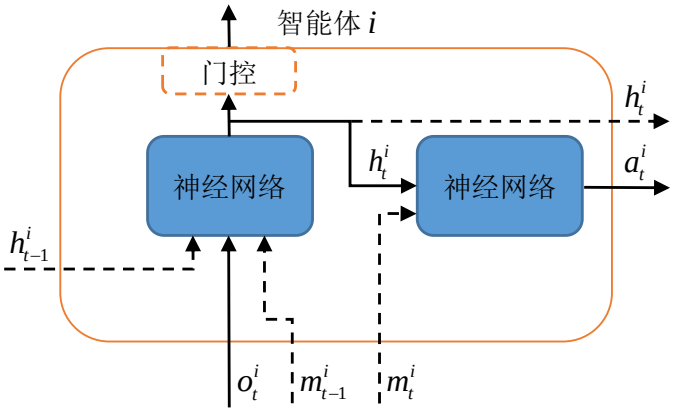


图2.5　主流的隐藏状态与消息耦合方法示意图

以智能体 i 为例，在 t 时刻，智能体获得观测 o_t^i 和来自其他智能体上一步的消息 m_{t-1}^i 后，使用神经网络作为编码器对输入进行编码，如果在编码时同时使用了上一步的隐藏状态 h_{t-1}^i，则此时神经网络被作为 RNN 使用。区别较大的是，在 RIAL、DIAL、CommNet 和 IC3Net 等算法中，观测 o_t^i、消息 m_{t-1}^i 和隐藏层数据 h_{t-1}^i 被一同作为 RNN 输入进行编码，而在 BiCNet、GA-Comm 和 SchedNet 等算法中，中消息 m_t^i 和隐藏层数据 h_t^i 被作为决策网络的输入。

2.3.4 离线强化学习

离线强化学习在目前的大部分研究集中在单智能体上，它是一种不需要通过环境交互，而使用数据集 $\boldsymbol{D}=\{\tau_1,\tau_2,...\}$ 进行训练的有监督学习方法，其中 $\tau=\{s_0,a_0,r_0,s_1,a_1,r_1,...\}$ 表示智能体 MDP 的一种现实。该方法可以使得智能体通过来自数据集的演示进行学习，从而减少神经网络初始化时对环境不断试错的过程，大大减少智能体训练时间。而离线强化学习最大的问题在于数据集是否能引导智能体行为收敛。当智能体的决策出现数据集中没有的情况时，可能会导致智能体决策进入 OOD 状态。此时如何减少后续智能体和数据集之间的偏差并实现对智能体的训练是离线强化学习研究的热点。

在离线强化学习中，由于使用数据集对智能体进行训练，为避免决策进入 OOD 状态，通常在式(2-5)的基础上，使用重要性采样来降低智能体与数据集策略函数分布不同而导致的偏离问题，其公式表示如下：

$$\left(\mathcal{T}^{\pi}Q\right)(s_t,a_t)=Q(s_t,a_t)+E_S\left[r_t+\gamma\rho(S,a_{t+1})Q(S,a_{t+1})-Q(s_t,a_t)\right] \quad (2\text{-}16)$$

其中$\rho(S,a_{t+1})$为重要性采样权重，其定义如下：

$$\rho(S,a_{t+1})=\frac{\pi(a_{t+1}|S)}{\mu(a_{t+1}|S)} \tag{2-17}$$

其中π为智能体自身的策略函数，μ为数据集提供的策略分布。为方便后文表述，利用神经网络拟合π时，神经网络参数表示为θ，即$\pi(a|s;\theta)$，同样拟合μ的神经网络参数表示为θ'，即$\mu(a|s;\theta')$。在实际计算过程中，一般通过求解能最大化状态价值的策略来实现，即对式(2-9)进行优化。

离线强化学习因为完全依靠数据集训练，故在对智能体决策好坏的评判标准除了智能体执行动作后获得的奖励以外，还存在其策略分布与数据集策略分布的偏差，一般以 KL 散度衡量。KL 散度计算公式如下：

$$D_{KL}(\pi\|\mu)=\sum_{x}\pi(x)\log\frac{\pi(x)}{\mu(x)} \tag{2-18}$$

现有的基于价值函数[35]或 AC 框架[58]-[59]的算法都采用了 KL 散度作为优化过程中的约束，用以保持智能体决策分布与数据集的相似性。文献[60]证明了基于策略的深度强化学习有更好的收敛效果，故本文将基于离线策略的方法来设计训练方法。其中 PPO 算法是一种简单的离线策略算法，其在考虑避免 OOD 现象的同时，也考虑了算法效率。该算法的学习过程可以表示如下：

$$\pi_{k+1}\leftarrow\arg\max_{\pi}E_{A\sim\pi(\cdot|s)}\left[Q^{\pi_k}(s,A)\right] \tag{2-19}$$

在式(2-16)和(2-19)的基础上，建立初始优化目标函数如下：

$$J^{\theta'}(\theta)\approx\sum_{(s_t,a_t)}\rho(s_t,a_t)Q^{\mu}(s_t,a_t) \tag{2-20}$$

加入 KL 散度约束项，得到优化目标函数如下：

$$J_{PPO}^{\theta'}(\theta)=J^{\theta'}(\theta)-\beta D_{KL}(\pi\|\mu) \tag{2-21}$$

其中β是一个自适应的参数，用于在训练过程中防止价值过估计。其调整规则为，当$D_{KL}(\pi\|\mu)>D_{KL,\max}$时，增大$\beta$，当$D_{KL}(\pi\|\mu)<D_{KL,\min}$时，减小$\beta$。

另一个离线策略方法是 AWR，与 PPO 不同，AWR 是对最优策略函数的直接拟合，使用拉格朗日乘子法对奖励加权回归（Reward-Weighted Regression，RWR）进行推导得到的。同样的，AWR 的学习过程表示如下：

$$V_k^{\boldsymbol{D}} \leftarrow \arg\min_V E_{S,A\sim\boldsymbol{D}}\left[\left\|U^{\boldsymbol{D}} - V(S)\right\|\right] \tag{2-22}$$

$$\pi_{k+1} \leftarrow \arg\max_\pi E_{S,A\sim\boldsymbol{D}}\left[\log \pi_k(A|S)\exp\left(\beta^{-1}\left(U^{\boldsymbol{D}} - V_k^{\boldsymbol{D}}(S)\right)\right)\right] \tag{2-23}$$

其中$U^{\boldsymbol{D}}$代表数据集中记录的回报奖励。算法首先对价值函数$V(s;\omega)$进行拟合，在用拟合得到的价值函数更新策略函数π。故同样可以得到其优化目标函数为：

$$J_{AWR}^{\theta}(\theta) = \sum_{(s_t,a_t)} \log\pi(a_t|s_t)\exp\left(\beta^{-1}\left(U_t - V(s_t;\omega)\right)\right) \tag{2-24}$$

根据目标优化函数进行梯度上升迭代，即可训练得到一个较优的策略函数π。

2.4 本章小结

本章对本文所使用的强化学习相关理论基础进行了介绍，并对现有的通信强化学习和离线强化学习方法进行了总结。从强化学习总体框架出发，根据公式推导逐步深入，总结深度强化学习训练过程使用的通用理论。并在之后对主流的通信强化学习方法所使用的神经网络结构和基于重要性采样的离线强化学习训练方法进行规律性总结，为后续章节的算法设计提供框架方面的论据。

第三章 无人机互联链路决策算法设计

3.1 引言

拓扑控制是网络规划算法中的核心，其为无人机网络提供了低能耗，高鲁棒性的链路。然而，传统的分布式拓扑控制方法通常设计复杂，计算复杂度较高，处理的信息维度较低，导致一种算法无法适应多种应用环境。

深度神经网络在非线性模型拟合方面体现出了极大的优势，因为其可以实现端到端的训练，简化了对复杂模型的建模过程，并在近几年取得了突出的效果。而深度强化学习为复杂环境中的神经网络训练提供了一套良好的框架，通过简单设计网络结构及其输入输出，以及对应的奖励函数即可解决复杂建模场景的诸多问题。本章节将针对现存在的无人机拓扑控制算法设计过程复杂的问题，根据第二章研究与总结的现有强化学习技术及其框架，设计一套可以将全局拓扑控制方法映射到分布式拓扑控制的强化学习算法，并设计对应的仿真环境用于算法的训练与验证。最后，通过实验与分析，评估所提出的深度强化学习算法在无人机拓扑控制方面的性能表现。

3.2 问题描述

分布式网络拓扑控制系统包括通信，计算和控制三大模块。其中通信部分用于建立智能体之间的物理和逻辑连接，并进行智能体间的数据传输和交互。计算部分用于对智能体和智能体间关系进行数学建模，包括考虑智能体属性，智能体相对位置，智能体间链路健康状态等。根据所建立的模型优化得到链接决策。控制器主要在时间和资源维度整合、协调智能体的通信和计算功能。通过考虑数据传输过程中时延导致的异步问题、通信链路带宽导致的数据传输大小受限问题和智能体自身算力导致的计算资源受限问题等来控制数据输入输出和对应功能触发，从而达到预期的性能。

分布式网络控制系统中三个模块之间的耦合十分紧密。控制模块需要由通信模块提供真实的网络环境反馈，由计算模块提供最新的链路决策；计算模块同样需要来自通信模块的反馈，同时通过通信模块接收其他智能体的数据，这些信息由控制模块提供，并控制计算模块进行计算；通信模块需要接收来自计算模块提供的协议数据并进行打包，根据计算模块决策，在控制模块协调下与其他智能体建立通信，并发送数据。这也导致整个模型设计与建模变得十分复杂，如何设计系统并优化其性能是学界一直探讨的问题。

无人机之间构建合理的通信拓扑，保证无人机之间良好的数据带宽是集群无人机稳定控制的必要条件。因此，在研究无人机通信拓扑控制时，一般考虑网络拓扑生成

和网络整体连通性问题。前者主要体现在无人机链接决策问题，后者主要体现在无人机决策形成的网络拓扑是否达到全局最优的问题。

FANET 最大的特点在于无人机作为网络节点的位置变化导致的链路质量和信号发射功率的改变。故在无人机网络拓扑控制策略中，需要无人机的实时位置，各个无人机作为网络节点的负载，剩余能量等参数，从而通过设计合理的网络拓扑算法，对节点间的连接链路进行决策。

基于MG，可以将链路通信决策过程表示为如图3.1所示。

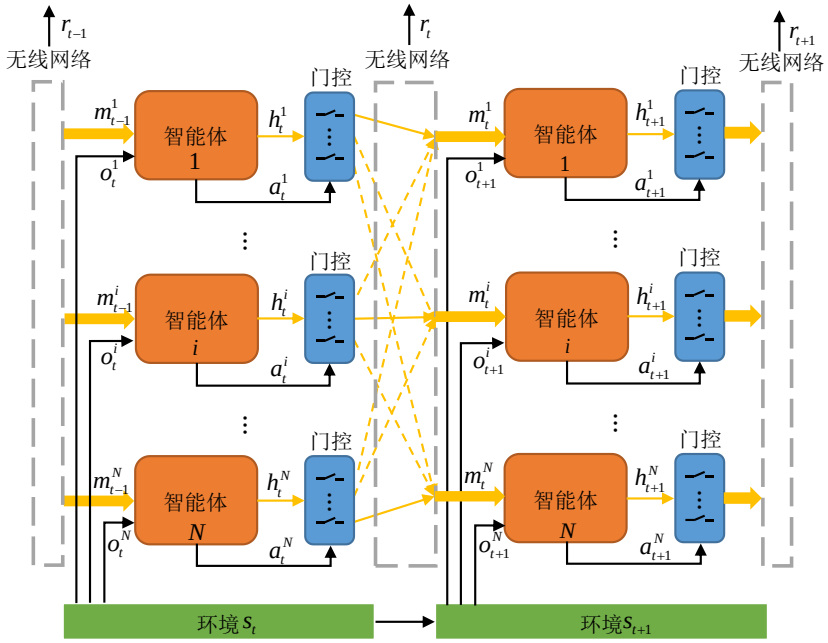


图3.1　链路通信决策过程示意图

由于需要实现分布式智能体的拓扑控制，所以智能体的动作空间应该是对于其他智能体的建立链路的选择，即对是否与其他智能体建立链接进行决策。一旦选择并建立链接，则会依靠所建立的链接将自己编码的消息 h^i 发送给链接对象。从链接对象角度来说，在接收了智能体 i 和其他智能体的消息后，通过自己对环境的观测，将产生新的编码和新的决策，从而继续影响环境中的其他智能体。

在现有的通信强化学习算法中，主要考虑智能体间进行通信来实现信息交流，从而实现协同作业。而本文主要考虑智能体间通信链路的建立，通过所有智能体的决策和动作执行来形成一个稳定和高效的链路拓扑。其不同点在于，现有研究大部分是智能体基于一个已经建立的、稳定的网络进行通信，本文则是考虑智能体在没有建立网络链接的环境中进行链路决策，从而建立网络，并通过所建立的网络链路进行通信。所以本文更偏向于研究如何设计并训练一个神经网络，使得该神经网络能够完成通信网络的构建，保证神经网络对于通信网络链路不稳定导致的消息未送达或错误送达时

的抗干扰能力。

从单个智能体来看，智能体在每一个时间步都需要获取对环境的观测o_t^i、自身产生的消息h_t^i和来自其他智能体的消息m_t^i，结合二者生成自己新的通信消息h_{t+1}^i，并通过自身决策动作a_{t+1}^i对消息进行发送。为实现此过程，需要对智能体神经网络结构进行合理设计，在实现上述数据通路的同时，保证神经网络生成的决策动作在无线干扰环境下的稳定性。

为方便结合现有研究对模型进行分析，可对上述模型进行变换。以智能体i为例，其接收到的消息m^i组合过程表示如下：

$$m^i = f^c\left(f_{i,1}^g\left(h^1\right),...,f_{i,N}^g\left(h^N\right)\right) \tag{3-1}$$

其中$f^c(\bullet)$表示对智能体接收到的消息的组合方法。在此基础上，智能体输入c^i表示如下：

$$c^i = m^i + Co^i \tag{3-2}$$

其中C是一个矩阵，用于将智能体观测o^i调整到和m^i相同的维度。此时，若将获得的输入c^i作为智能体新的等效观测$\tilde{o}^i \triangleq c^i, \tilde{o}^i \in \tilde{\boldsymbol{O}}^i$，即将上述的模型从通信强化学习框架转换到更为广泛的多智能体强化学习框架。而从单个智能体角度看，$\tilde{o}^i$也可认为是智能体i的状态，可以将智能体i的状态转移过程用 MDP 描述。

基于上述分析可知，等效观测$\tilde{o}^i$包含了o^i和m^i两个变量，其中m^i受到来自其他智能体的影响，故$\tilde{o}^i$的观测空间$\tilde{\boldsymbol{O}}^i$较大，难以设计奖励函数实现对神经网络训练的引导。另外，虽然离线强化学习可以解决奖励函数设计困难的问题，但对于离线数据集$\boldsymbol{D}$中的一种现实τ，其表示的智能体所处状态s是智能体的一种确定状态。而在本文所使用的模型中，$\tilde{o}^i$实际包含了对环境的观测o^i和来自其他智能体的消息m^i。根据 IQL 算法研究的结论[41]，单智能体模型可以直接应用在同质多智能体模型中，那么对于环境中的智能体i，$\tilde{o}^i$可以等效为离线强化学习框架下智能体的所处状态s。因此，对于离线强化学习框架下作为智能体决策输入的状态s，包含了已知的观测o^i和待优化的变量m^i，其中m^i的存在使得本文模型与现有的离线强化学习模型框架不符，无法使用现有方法对模型进行训练。

综上所述，本章需要解决的问题如下：

（1）使用分布式神经网络决策构建全局最优网络拓扑。

（2）保证神经网络在无线链路干扰环境下的稳定性。

（3）本文模型训练困难且不适用于现有离线强化学习算法。

3.3算法框架

3.3.1 系统设计

针对上一节描述的通信决策过程，对于整个系统来说，本文采用多智能体强化学习中较为普遍的 CTDE 框架，并考虑方便部署，使所有智能体共享网络参数。

对于智能体，涉及的输入包括对环境的观测，来自其他智能体的消息；输出为智能体的链路决策，该决策用于智能体与其他智能体连接并进行网络通信。在对深度强化学习神经网络结构进行设计前，需要确定无人机观测空间，消息向量和动作空间。

（1）观测空间

在分布式场景下，每架无人机对环境的观测能力是有限的，即分布式无人机i对当前t时刻的全局状态s_t，有局部观测o_t^i。o_t^i是由无人机自身传感器将s_t的局部信息转换而成的数据，例如通过全球定位系统（Global Positioning System，GPS）将无人机在空间中的绝对位置转换为经纬度和海拔高度数据传入无人机飞控，但每一架无人机i都无法观测到其他无人机$j\left(j\neq i\right)$的位置数据。故分布式无人机只能通过相互通信来获取其他无人机的观测数据。

为突出本文针对无人机网络规划的研究重心，仅对 FANET 中最重要的无人机空间状态参数作为观测，即无人机的位置，速度，加速度状态。其运动状态方程表示如下：

$$\begin{bmatrix}\dot{p}^e\\ \dot{v}^e\end{bmatrix}=\begin{bmatrix}v^e\\ 0_3\end{bmatrix}+\begin{bmatrix}0_3\\ I_3\end{bmatrix}a^e \tag{3-3}$$

其中$p^e\in\mathbb{R}^{3\times1}$表示地球坐标系下无人机的三轴位置，$v^e\in\mathbb{R}^{3\times1}$表示地球坐标系下无人机的三轴速度，$a^e\in\mathbb{R}^{3\times1}$表示地球坐标系下无人机三轴加速度。

无人机飞行控制单元包含了提供角速度观测的陀螺仪，提供加速度观测的惯性测量单元（Inertial Measurement Unit，IMU），提供飞行高度观测的气压计等传感器。对于室外无人机还会加装 GPS 等提供对三维空间位置和速度的观测。本章节假设无人机能够获取自身三维空间位置，则式(3-3)中状态是可观的。进一步的，考虑到若直接将上述无人机状态观测输入，将导致无人机的观测向量维度过高，不利于后续功能的拓展。根据文献[61]提供的对深度神经网络对微分方程和偏微分方程有着很强的拟合能力的证明。可以认为对神经网络输入必要的无人机状态数据，而不用输入整个状态向量，便可在神经网络的隐藏层中可以根据需要对无人机的状态维度进行展开，进行特征提取。

另外，由于所有智能体共享参数，为保证智能体能做出不同决策，参考文献[45]

的思路，在将自身位置作为观测输入的同时，也将自身的编码以独热（one-hot）的形式作为观测一同输入。

（2）消息向量

无人机间在建立连接后需要将自身状态和接收到的消息发送给目标无人机，而该消息内容由神经网络生成，并不需要用户定义，故这里只需对消息向量的类型、维度、消息组合方式和初始化方式进行设计。

关于消息向量的类型，文献[46]证明了连续数据相对于离散数据有更好的训练效果，同时考虑目前无人机无线通信方式都采用数字信号进行通信，故使用浮点数的消息向量可以获得更好的训练和部署效果。

关于消息向量的维度，考虑到目前无人机间分布式通信主要以 Zigbee、蓝牙和 UWB 等方式进行通信。因为大部分通信方法都存在带宽受限问题，所以需要对消息向量的维度进行限制，即规定每一架无人机只能输出固定大小的消息，其中包含若干个单精度浮点数。具体做法是，在网络中设计一对编解码器，在消息发送前对消息进行编码，并将编码消息转换为单精度浮点数加入数据包中进行发送。而在无人机接收到消息后，将消息重新转换为神经网络中使用的双精度浮点数，并使用解码器进行解码，再送入后续神经网络进行处理。关于在传输过程中的精度损失问题，文献[45]通过提出的 DIAL 证明了对数据进行精度方面的截取在数学上等价于对离散信号叠加了噪声，依然可以对梯度进行推导。

关于消息的组合方式，考虑到本章采用的是无人机互联的环境，无人机的消息可以直接点对点传输，在另一架无人机接收后将其保存入内存等待调用。故采用消息拼接方式可以最小化对信息的损失，在训练过程中达到较好的收敛效果。

关于消息的初始化方式，考虑到在实际应用中，一方面无人机在部署不同场景时的位置摆放是不同的，但可以在摆放时保证任意两架无人机之间通信条件良好。另一方面，在系统启动时无人机之间并不知道其他无人机的状态，无法进行拓扑决策。故在初始化时，所有无人机将零向量作为消息输入，生成消息后，用广播替代决策动作。即在初始化时，无人机会收到来自其他所有无人机的消息，并进行自身的第一次决策。

（3）动作空间

无人机需要通过动作决策来选择与环境中的无人机建立连接，其连接数可以大于或等于1。而由于强化学习的动作选择是基于策略函数输出的概率分布进行抽样，独热向量能准确的反映这一分布，保证网络的收敛。具体而言，动作选择过程如图3.2所示，动作空间的维度为2^{N-1}，在神经网络运行时对其进行采样，得到对应的编号，并将编号转换为二进制表示，无人机选择二进制对应的为1的位表示的无人机进行通信。

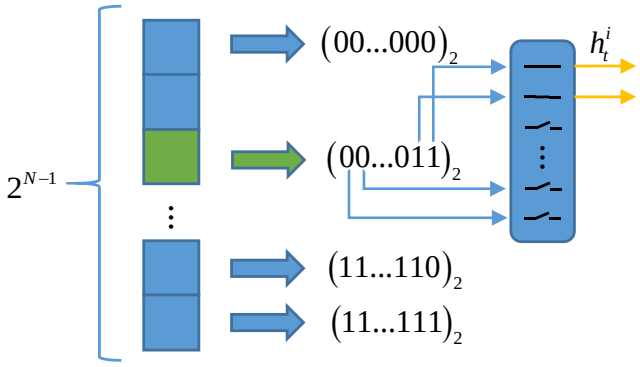


图3.2　基于One-Hot的动作选择过程示意图

综上所述，结合分布式拓扑控制模型，其计算模块即为无人机的神经网络，神经网络对数据进行计算处理后，通信模块接收来自神经网络的编码数据和动作决策，将数据按网络协议打包成数据帧，并根据动作决策将数据帧发送到目标无人机。而控制模块则用于协调这一过程，控制计算模块和通信模块的数据流和模块本身的触发。

3.3.2 神经网络结构

基于以上分析，结合2.3.3节中对现有通信强化学习主流网络框架的特征分析，设计网络拓扑控制模型（Network Topology Control Model，NTC-Net）的神经网络结构如图3.3所示。

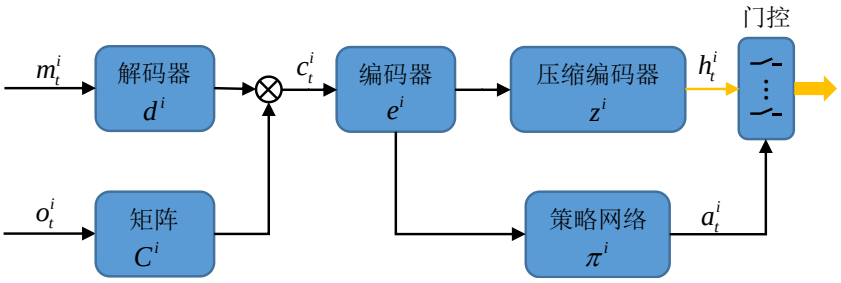


图3.3　神经网络结构图

在图 3.3 中，除编码器e^i为 LSTM 以外，其他神经网络模块均为 FC。

对于无人机i，在t时刻，控制模块会先将无人机飞控对于自身的观测状态o_t^i读取出来，同时从通信模块读取当前时刻搜集到的消息m_t^i。之后首先将读取到的消息送入解码器d^i进行解码，再将其与调整过维度后的观测状态$C^i o_t^i$进行加和得到混合数据c_t^i，其过程用公式表示如下：

$$c_t^i = C^i o_t^i + d^i\left(m^i\right) \tag{3-4}$$

将混合数据 c_t^i 送入编码器 e^i 进行编码后，一方面被送到决策网络 π_i 中获得决策概率分布并抽样得到动作 a_t^i，另一方面被送到压缩编码器 z^i 中进行压缩得到要发送的数据 h_t^i。最后控制器将压缩后的数据和链接决策送入通信模块，通信模块根据链接决策将消息发送给对应的无人机。

3.3.3 训练方法

根据 3.2 节的分析可知，由于消息 m^i 的存在，本文所使用的神经网络框架与现有离线学习框架存在差异，不能直接使用现有模型框架进行训练。

目前对于多智能体离线强化学习的训练框架较少，收敛性证明不足，并且文献[54]通过一个例子证明了基于价值的离线强化学习在多智能体环境下容易出现对状态价值的高估问题。因此本文考虑采用基于策略的学习框架。参考目前 CommNet、IC3Net 和 GA-Comm 算法使用的训练框架，有以下推导。

在 PPO 算法使用的式(2-20)中，对基于策略 μ 的动作价值估计函数 $Q^{\mu}\left(s,a\right)$ 不再适用，进而通过式(2-2)将式(2-20)简化为折扣回报表示如下：

$$J_U^{\theta'}\left(\theta\right)\approx\sum_{\left(s_t,a_t\right)}\rho\left(s_t,a_t\right)U_t \tag{3-5}$$

将上式代入式(2-21)中可得优化目标函数如下：

$$J^{\theta'}\left(\theta\right)=J_U^{\theta'}\left(\theta\right)-\beta D_{KL}\left(\pi\|\mu\right) \tag{3-6}$$

式(3-5)的参数更新可以使用 RINFORCE 框架实现。为进一步推导公式，参考 PPO 算法的推导过程，对式(2-10)进行重要性采样，并加入 KL 散度约束，可将式(3-6)以多智能体策略梯度的格式表示如下：

$$\nabla J^{\theta'}\left(\theta\right)=\sum_i\left[E_{A^i\sim\mu^i\left(\cdot|c_t^i;\theta'\right)}\left[U_t^i\rho\left(c_t^i,A^i\right)\nabla\log\pi\left(c_t^i,A^i\right)\right]-\beta\nabla D_{KL}\left(\pi^i\|\mu^i\right)\right] \tag{3-7}$$

在 NTC-Net 中，专家策略分布 μ 由集中式拓扑控制算法给出，所以 μ 在训练过程中是已知分布，训练时可以直接由上式计算梯度。

本章节主要验证所提出算法的可行性，为方便证明，在单个时间步下只使用一种专家策略，而不考虑多种可能的策略以及它们的时序关系。因此，设计专家策略分布为：只有动作 $a_t^{i,*}$ 的概率密度为 1，其他工作均为 0，即 $\mu\left(\tilde{o}_t^i,a_t^{i,*}\right)=1$。依据设计的分布可知，智能体根据专家策略进行的决策为最优决策，获得奖励为 $r_{\max}$，而非最优决策的奖励为 0。

在此假设下，根据式(2-18)对 D_{KL} 进行计算时，会有分母为 0 的情况出现。因此，本文引入了另一种形式的 KL 散度约束：在训练过程中，设置一个超参数 t_{cnt} 用于控制前向传播长度，通过在每一回合训练开始时重置一个计数器，当 $D_{KL}\left(\pi \| \mu\right) > D_{KL,\max}$ 时，计数器加一，反之，计数器归零。当计数器的计数达到 t_{cnt} 时，停止前向传播，开始反向传播更新神经网络参数，该过程如算法 3.1 所示。在此基础上，式(3-5)可以被表示如下：

$$\nabla J^{\theta'}\left(\theta\right) = \sum_{i} U_t^i \nabla \pi\left(c_t^i, a_t^{i,*}\right) \tag{3-8}$$

其中折扣回报表示如下：

$$U_t^i = \sum_{k=t} \gamma^{k-t} r_{\max} \tag{3-9}$$

值得注意的是，本文只对学习过程设置了一个常数奖励，并不意味着智能体无法根据环境对不同动作进行探索，而是保证智能体在环境噪声的影响下，能在训练过程中生成一个较为稳定的通信协议，从相互通信的角度来提升系统鲁棒性。

为了方便 $D_{KL,\max}$ 的计算，本文引入了一个新的超参数 $P_{a*} \in \left(0,1\right)$，该超参数的意义是，假设在智能体的策略分布 π 中，除专家策略 $a*$ 以外其他策略都服从均匀分布，此时 $a*$ 的最小概率密度为 P_{a*}。由于 $\mu\left(\tilde{o}_t^i, a_t^{i,'}\right) = 0$，为防止计算 KL 散度时出现分母为 0 的情况，这里引入一个极小值 ε 替代分母为 0 的项，故也将 ε 称为非专家策略的概率。其具体计算方式如下：

$$D_{KL,\max} = \left(1 - P_{a*}\right)\frac{N-1}{N}\log\left(\frac{1}{\varepsilon}\right) + P_{a*}\log\left(P_{a*}\right) \tag{3-10}$$

整个训练过程的伪代码如下：

算法 3.1： 神经网络训练

输入： 允许决策错误最大时间步 t_{cnt}，回报折扣 γ，奖励值 $r_{\max}$，允许最大时间步 T，最大KL散度 $D_{KL,\max}$，梯度优化函数 $f_{op}(x)$

初始化神经网络参数 θ
for $epoch = 1,2,...$ **do**
    重置决策错误计数器 $cnt = 0$，初始化轨迹 τ，初始化消息 m_{init}
    将局部观测 o_0 和初始消息 m_{init} 输入智能体神经网络，得到编码消息 h_{init}
    将 h_{init} 按广播形式拼接为 m_0
    for $t = 0,1,...,T$ **do**
        根据 m_t 和 o_t 输出决策动作 a_t 和消息 h_t
        根据 a_t 将消息拼接组合为 m_{t+1}
        生成专家决策 a_t^*
        if $D_{KL}(\pi \| \mu) > D_{KL,\max}$ **then**
            $cnt = cnt + 1$
            if $cnt == t_{cnt}$ **then**
                break
            end if
        else
            $cnt = 0$
        end if
        记录轨迹 $\tau \leftarrow \left(\pi\left(s_t, a_t^*\right)\right)$
    end for
    更新神经网络参数 $\theta \leftarrow \theta - f_{op}\left(\nabla J^{\theta'}(\theta)\right)$
end for

3.3.4 评价指标

本文基于离线强化学习方法对模型进行训练，因为训练方法中采用的奖励函数为常数，所以无法直接使用智能体获得的奖励来衡量训练效果，需要重新定义一个指标用于衡量智能体决策与专家决策的相关性，以用于评估算法性能，推导过程如下：

首先，需要获得在神经网络初始化状态下的动作决策与专家决策的相关性。因为每个智能体神经网络输出是对其他智能体是否建立链接的决策，即动作的样本空间为 $\Omega = \{$连接，不连接$\}$，所以在神经网络初始化时，其并未对观测数据 o^i 和通信消息 m^i 提取特征。可认为在任意 t 时刻，智能体 i 对是否连接其他智能体 j 的决策动作 $A_t^{i,j} \in \Omega$ 服从概率相等的 0-1 分布，即 $A_t^{i,j} \sim B(1,0.5)$。综上对所有智能体的动作决策与专家决策进行异或运算并求均值，该过程表示如下：

$$X_t = \frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{j=1, j\neq i}^{N} A_t^{i,j} \oplus a_t^{i,j,*} \tag{3-11}$$

由于智能体的神经网络处于初始状态，没有智能体间没有建立联系，故可认为随机变量 $A_t^{i,1},...,A_t^{i,N-1}$ 相互独立。而智能体的神经网络也没有对输入和输出建立联系，所以经过神经网络消息和动作均为随机输出。综上所述，经过神经网络决策后的动作 $A_t^1,...,A_t^N$ 相互独立，计算 X_t 的期望如下：

$$E[X_t] = \frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{j=1, j\neq i}^{N} E\left[A_t^{i,j} \oplus a_t^{i,j,*}\right] = 0.5 \tag{3-12}$$

在神经网络运行的每一个时间步都将根据策略函数输出的概率分布对动作进行抽样，从而得到样本 $x_0, x_1, ..., x_M$。由式(3-12)可知 $E[X_t]$ 数值较大，在计算过程中不可忽略，所以定义神经网络决策相对于专家决策的拟合度 P_p 表示如下：

$$P_p = \left(0.5 - \frac{1}{T_{step}} \sum_{t=0}^{T_{step}} x_t\right) \cdot 200\% \tag{3-13}$$

其中 T_{step} 为神经网络在训练时的前向传播时间步。对 T_{step} 的具体解释为：由于算法 3.1 中存在对神经网络前向传播的 KL 散度约束，当一次前向传播结束时，记录所使用的时间步即为 T_{step}。

在式(3-13)定义下的 $P_p \in [-1,1]$，但可以认为 $P_p < 0$ 时智能体决策与专家决策相悖，此时决策无意义。因此，只有当 $P_p \geq 0$ 时才将其作为衡量智能体决策与专家决策拟合度的指标。

除了使用式(3-13)的衡量指标外，由于 T_{step} 与 KL 散度约束相关，表示智能体策略分布对专家策略分布的拟合能力，故也作为另一个衡量指标。一般情况下，在多个回合间记录的 T_{step} 总体方差较大，故本文采用多次前向传播记录的 T_{step} 的均值 $\bar{T}_{step}$ 来衡量训练效果。

3.4 拓扑控制算法

智能体通过每一步生成的专家策略分布 μ 更新神经网络参数，从而在迭代过程中生成通信协议，达到分布式智能体对专家策略的模仿的目的。根据上文分析，分布 μ 实际上是指集中式拓扑控制算法在 t 时刻根据无人机 i 状态计算得到的最优动作 $a_t^{i,*}$，且有 $\mu(\tilde{o}_t^i, a_t^{i,*}) = 1$。而由式(3-8)可知，神经网络的训练完全依靠 $a_t^{i,*}$ 的选择，所以理论

上只要能找到一个合理的集中式拓扑控制算法，都可以由神经网络进行学习，从而获得支持分布式拓扑控制的神经网络。因此，本节将对文章中实验用到的集中式拓扑控制算法进行介绍。

首先，无人机作为网络节点，其无线发射功率受限，当无人机间直线距离超过一定值后，则无法进行正常通信。所以在生成全局拓扑时，无人机间的距离将作为硬约束。

其次，考虑网络连通性问题，即任意两个无人机节点的数据可以通过合理的路由进行交互传输。若存在两个节点无法连通，则认为当前拓扑也是没有意义的。其伪代码如算法 3.2 所示。根据距离约束生成拓扑阶段，就需要检查网络连通性，若网络存在无法连通的两个节点，则认为该环境下无法优化，在训练过程中直接结束对该场景的训练。

算法 3.2： 生成专家策略

输入： 当前时刻所有无人机的观测位置坐标 $o_t^i, i \in \boldsymbol{N}$

输出： 当前时刻专家策略 a_t^*

1：根据硬约束生成拓扑图 G
2：**if** G 存在两个节点不连通 **then**
3：　结束当前训练
4：**end if**
5：调用拓扑优化方法
6：**for** $i \in \boldsymbol{N}$ **do**
7：　从图 G 中读取无人机 i 应选择的动作 $a_t^{i,*}$ 记录入 a_t^*
8：**end for**

为达到更好的优化效果，本节采用了目前常用的 GG 算法作为集中式拓扑专家策略的生成器。其算法伪代码如算法 3.3 所示。

算法 3.3： 使用GG进行拓扑优化

输入： 需要优化的拓扑图 G

输出： 优化后的拓扑图 G

```
for u,v,w∈N and u≠v≠w do
    计算u和v的中点m
    if 图G中u和v之间不存在连接 then
        continue
    else if d(m,w)<d(m,u) then
        清除图G中u与v之间的边
    end if
end for
```

3.5 仿真环境设计

强化学习的训练过程是一个需要与环境交互从而获得奖励和更新状态的过程，在进行实验之前，需要有一个仿真环境提供强化学习训练，以提供强化学习神经网络需要拟合的环境特征。

考虑到强化学习与环境交互频繁，为减小计算机不必要的负载，提高计算速度，本文基于 C++语言对 Python 模块进行开发，使仿真环境能够被使用在基于 Python 的深度学习框架下。仿真环境框架如图 3.4 所示。

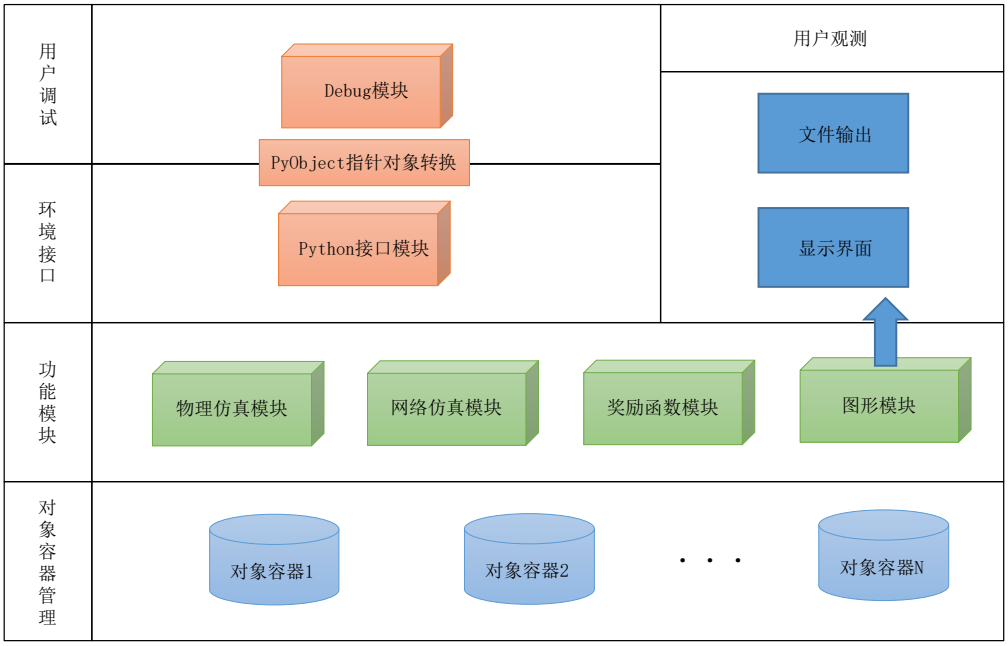


图3.4 仿真环境软件框架结构图

在图 3.4 中，除用户观测部分外，均为实体程序文件，其下层功能模块为上层调用，同一层级则相互独立，保证各个功能模块去耦，提高整体程序开发灵活性。

为保证开发效率，环境在编译时产生两个文件，一个是由图 3.4 中 Python 接口模块产生的，可以用于 Python 调用的动态链接库文件。另一个是由图 3.4 中 Debug 模块产生的，可以用于 gdb 调试的可执行文件。其中 Debug 模块通过 PyObject 指针转换函数调用 Python 接口模块来模拟 Python 对其调用的效果，从而使得用户可以对整个程序进行调试，及时发现并修复程序漏洞。

用户观测包括图形界面输出和文件输出，该输出由图形模块统一管理，由图形模块对仿真状态进行图形化生成，并通过用户调用对应函数接口将图形化输出结果实时显示在图形界面上，或输出为视频格式到文件中，用于观测训练过程。

3.5.1 对象容器管理

物体对象容器相当于仿真环境的内存管理模块，用于存储环境中各类物体的属性和状态，如环境中无人机的大小、最大通信带宽、图形形状等属性和位置、速度和连接链路等状态。为保证程序开发的可拓展性，容器管理器按照分布式网络控制系统结构预定义了对象容器中可访问的内存及函数，对于用户开发的其他自定义方法则只能通过对象指针方式传递。其接口结构如图 3.5 所示。

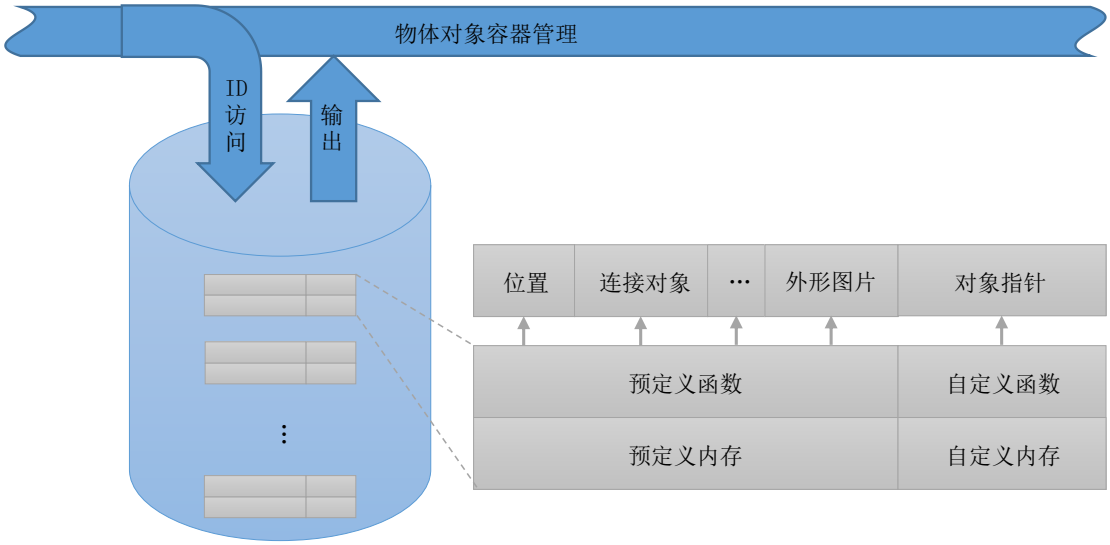


图3.5 对象容器接口结构图

用户通过继承管理器提供的类进行对物体对象类进行开发，开发好的类将由对象容器管理器根据用户需求对类进行实例化并存入同类的容器中。上层模块只能通过对应的类型和对象 ID 对物体对象进行访问，从而实现了对象容器管理器与模块的解耦。

同时，存储在容器中的对象将在每回合环境状态转移过程中被各个模块遍历访问，

例如物理仿真模块将访问每一个对象的位置，速度和大小等数据，网络仿真模块将访问对象所决策的与其他对象的网络连接等。从而实现物体对象状态的自动更新，优化开发流程。

3.5.2 功能模块

功能模块用于实现仿真环境的各项功能，主要包括物理仿真模块，网络仿真模块，奖励函数模块和图形模块。由于本文通过传统集中式拓扑控制算法引导强化学习决策，集中式拓扑控制算法很容易拓展到三维，故为方便验证训练效果，现阶段仅使用二维环境进行仿真。

（1）物理仿真模块

该模块主要用于仿真智能体的空间状态变化，由于本文研究算法中并未涉及无人机路径与运动控制问题，因此在该模块中仅基于无人机随机运动仿真作开发。

物理仿真模块基于式(3-3)对无人机状态进行仿真，不考虑无人机高度。为保证程序的计算速度，同时考虑实际应用要求强化学习神经网络对噪声具有一定抗干扰能力，故采用欧拉法对仿真环境中的无人机进行状态更新，更新过程表示如下：

$$\begin{bmatrix} p(k+1) \\ v(k+1) \end{bmatrix} = \begin{bmatrix} I_2 & I_2\Delta t \\ 0_2 & I_2 \end{bmatrix} \begin{bmatrix} p(k) \\ F_{sat}\left(v(k)\right) \end{bmatrix} + \begin{bmatrix} 0_2 \\ I_2\Delta t \end{bmatrix} a\left(k\right) + \begin{bmatrix} 0_2 \\ I_2\Delta t \end{bmatrix} w \tag{3-14}$$

其中 Δt 为状态更新的时间步，$p(k) \in \mathbb{R}^{2\times1}$ 表示第 k 时间步下的水平位置向量，$v(k) \in \mathbb{R}^{2\times1}$ 表示第 k 时间步下的水平速度向量，$a(k) \in \mathbb{R}^{2\times1}$ 表示第 k 时间步下的水平加速度向量，$w \sim N\left(0,\sigma\right)$ 为过程噪声随机变量，F_{sat} 为限幅函数，F_{sat} 的表达如下：

$$F_{sat}\left(v\right) = \begin{cases} v, |v| \le v_{\max} \\ \dfrac{v}{|v|} \cdot v_{\max}, |v| > v_{\max} \end{cases} \tag{3-15}$$

为防止神经网络对于无人机空间状态出现对运动轨迹的过拟合现象，在环境初始化时，除了对无人机位置，速度进行随机设置，并在状态更新过程中加入噪声以外，还需要对无人机和环境设置碰撞与势场，用以兼容大多数无人机应用场景，防止神经网络过拟合。因此在当前环境中，考虑程序计算复杂度，无人机加速度 $a\left(k\right)$ 定义如下：

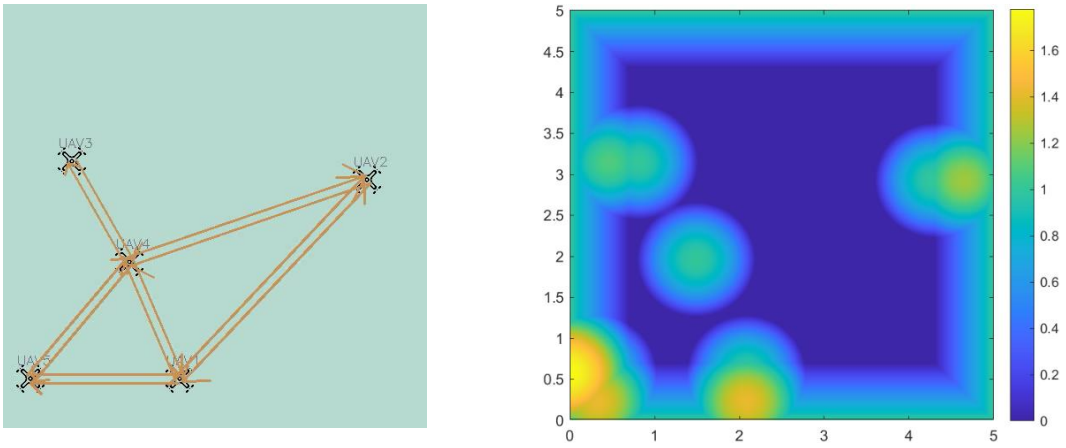
$$a\left(k\right) = f_g\left(p,k\right) - f_{res}\left(v\right) \tag{3-16}$$

其中 $f_g\left(p,k\right)$ 表示在 p 位置下的第 k 时间步的比力向量，f_{res} 表示在速度 v 下产生的阻

力。f_g 由两部分构成，一部分是环境中的势场 f_{evn}，另一部分是无人机自身的势场 f_{uav}，其表达式如下：

$$f_g(p,k)=f_{evn}(p)+f_{uav}(p,k) \tag{3-17}$$

f_{evn} 和 f_{uav} 在环境中可以由用户自定义。在本文中，在如图 3.6(a)所示的仿真状态下，设置的势场如图 3.6(b)所示。



(a) (b)

图3.6 仿真环境及其势场示意图

其中 $f_{evn}(p)$ 考虑无人机位置 p 在靠近环境边界一定范围时，与最近边界距离 d_p 存在一次函数关系，方向为逆梯度方向。而 $f_{uav}(p,k)$ 模仿圆柱形势场，同时减少运算量，当质点位置 p 靠近无人机一定范围时，与无人机距离 d_f 存在二次函数关系，最大值为无人机位置中心，方向为逆梯度方向。

在环境中，为运动物体配置了阻力模块，该模块可以简单地使用正比例函数模型表示如下：

$$f_{res}(v)=rv \tag{3-18}$$

其中 r 表示阻力系数。

势场的存在不一定能避免无人机与边界或相互之间产生碰撞，环境中设置碰撞盒为圆形，碰撞发生时，遵循动量定理。但在仿真过程中应通过合理配置限幅函数 F_{sat} 和比力函数 f_g 来避免碰撞，因为碰撞在实际环境中是不被允许的。

（2）网络仿真模块

网络仿真模块主要针对物理层与数据链路层的仿真，即考虑无线发射功率和通信信道干扰的情况。但二者存在一定的耦合关系，且信道干扰情况与智能体所处环境有

关，难以建模。所以在设计网络仿真模块时，对无线发射功率的仿真计算和信道干扰模型进行了化简。

在不同的实际环境中，通信受到环境干扰而无法完成的概率分布不同。考虑到神经网络训练过程是对环境特征提取与拟合的过程。为防止神经网络在训练过程中对环境特征出现过拟合情况，在网络仿真模块的设计中，假设在干扰环境下成功发送消息的概率为 p_m，即是否发送成功服从 0-1 分布 $X \sim B(1, p_m)$。实际上，针对不同场景，概率 p_m 也不一样，虽然无法得知 p_m 的具体分布，但有大量实验，特别是在计算机视觉的数据增强领域[62-64]的实验表明对于神经网络来说，只要一个量足够随机，就不会被提取为重点特征。考虑在程序中的实现难度和计算速度，不妨假设概率 p_m 的统计分布为高斯分布。因此在仿真环境中，设计一个随机数发生器 $Z(\mu_w, \sigma_w)$，使其能产生服从以 μ_w 为均值以 σ_w^2 为方差的高斯分布的随机数样本 z，并对随机变量 $X \sim B(1, z)$ 进行抽样，得到是否发送成功的结果，通过结果决定是否将通信数据存入目标智能体对象的内存中。由于 Z 服从高斯分布，若在采样时得到的样本 z 出现小于 0 或大于 1 的情况，程序会将样本废弃，重新采样，保证样本可用。

无线发射功率也是影响随机数发生器的均值与方差的重要一环。在仿真环境中，考虑无线信号在空间中的衰减，通过调整 $Z(\mu_w, \sigma_w)$ 的参数，即增大方差 σ_w^2，并粗略认为距离与均值 μ_w 成正比例关系，其表达式如下：

$$\mu_w(d_{i,j}) = m(d_{\max} - d_{i,j}) + \mu_{\min}, d_{i,j} \le d_{\max} \tag{3-19}$$

其中 $d_{\max}$ 表示允许通信的最大距离，m 为可调系数。为保证经过仿真环境训练的神经网络可以被部署，$d_{\max}$ 的设置一般小于实际的最大通信距离，同时 $Z(\mu_w, \sigma_w)$ 与 $d_{\max}$ 存在关系如下：

$$Z(\mu_w, \sigma_w) = \begin{cases} Z(\mu_w(d_{i,j}), \sigma_w), d_{i,j} \le d_{\max} \\ Z(0,0), d_{i,j} > d_{\max} \end{cases} \tag{3-20}$$

即在超过通信距离后，智能体 i 和智能体 j 之间的通信一定无法完成，以此作为智能体的硬约束，保证通信过程的稳定性。

（3）奖励函数模块

奖励函数模块通过判断智能体状态，生成评价指标、奖励值和专家策略，从而引导智能体训练。本文主要使用离线学习方法对模型进行训练，故在模块开发过程中仅加入了评价指标计算与专家策略生成功能。相关算法已在 3.3 和 3.4 节进行了推导与阐述。

（4）图形模块

图形模块用于将无人机运动状态与连接状态转换为图形方式进行显示，显示界面如图 3.6(a)所示，方便调试和训练时对智能体状态进行直观的观测。为不影响仿真环境效率，该模块由用户设置开关，并配置了实时显示与输出视频文件两种功能。实时显示一般用于模型验证，这使得整个算法会在显示的帧间被挂起；而视频输出一般用于对训练过程的观测，不会将算法挂起，保证训练效率。

3.5.3 环境接口与用户调试

为保证开发效率，环境在编译时会产生两个文件。一个是由图中 Python 接口模块产生的，可以用于 Python 调用的动态链接库；另一个是由图中 Debug 模块产生的，可以用于 gdb 调试的可执行文件。其中 Debug 模块通过 PyObject 指针转换函数，调用 Python 接口模块来模拟 Python 对其直接调用的效果，从而使得用户可以通过 gdb 实现对整个程序进行调试，及时发现并修复程序漏洞。需要注意的是，两个文件编译过程完全分离，采用不同优化等级，从而保证最终应用的 Python 接口模块的存储空间大小和执行效率。

3.6 实验结果与分析

基于 3.2 节对等效观测 $\tilde{o}^i$ 的分析，现有的 ICQ-MA 和 MABCQ 算法无法以带有变量 m^i 的 $\tilde{o}^i$ 作为输入，并结合文献[54]对 BCQ 算法在多智能体环境下不收敛的结论，同时考虑本文使用的折扣回报公式(3-9)中，状态价值与智能体当前时间步为正比例关系，使得价值估计函数失去意义，所以大部分使用价值函数作为基线的策略学习算法在本文环境下会出现退化，如式(2-24)被退化表示如下：

$$J^{\theta}(\theta)=\sum_{(s_t,a_t)}\log\pi(a_t\mid s_t) \tag{3-21}$$

因此本文提出的 NTC-Net 无法对现有算法形成对比，本节的实验重点为对 NTC-Net 可行性的验证和性能研究。

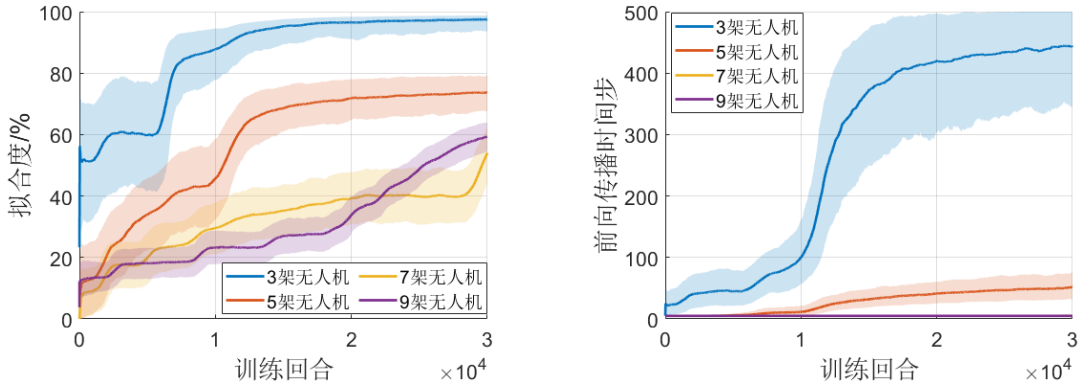
本节使用的超参数如表 3.1 所示。

表3.1 神经网络训练超参数

参数名称	数值
固定超参数	
训练回合数	30000
优化器	RMSprop
学习率	0.001
允许决策错误最大时间步 t_{cnt}	5
折扣回报 γ	0.9
奖励值 $r_{\max}$	0.001
允许最大时间步 T	500
非专家策略概率 ε	1e-6
P_{a*}	0.5
解码器激活函数	Sigmoid
可调超参数	
通信带宽（字节）	16-64
隐藏层维度	64-256

实验 1：不同无人机数量对本章算法框架的收敛性能的影响。

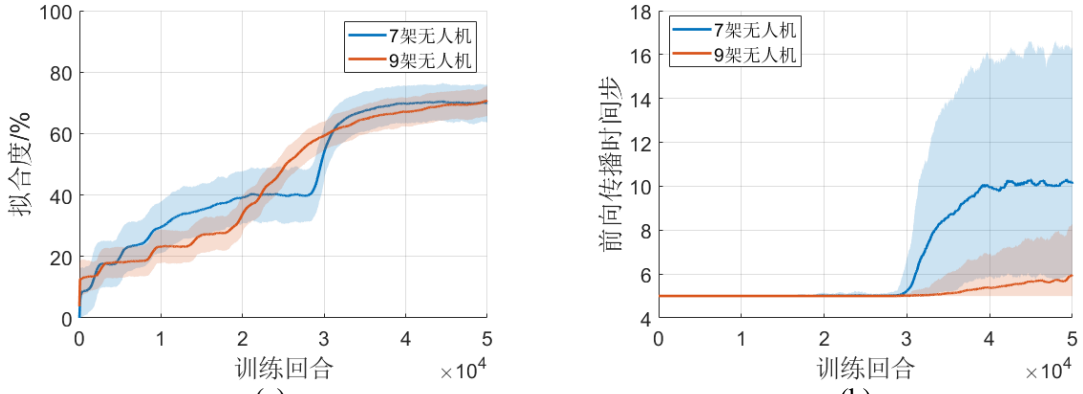
设置通信带宽为 16 字节，隐藏层维度为 128 进行训练，训练结果如图 3.7 所示。其中图 3.7(a)中的拟合度由式(3-13)计算得出。由图 3.7 可知，随着无人机数量增加，拟合度逐渐下降，前向传播时间步 T_{step} 随着无人机数量增加而减少，当无人机数量为 7 架时，1000 次前向传播的平均时间步 $\overline{T}_{step}$ 为 6.7 步，当无人机数量为 9 架时 $\overline{T}_{step}$ 为 5 步。事实上，拟合度并不能单一反应神经网络训练效果，因为拟合度是统计无人机每一次决策与专家策略的差距，并在运行结束后求均值得出的。所以虽然从图 3.7(a)中在第 30000 回合训练后的结果来看，9 架无人机的情况优于 7 架无人机的情况，但 9 架无人机的 $\overline{T}_{step}$ 明显小于 7 架无人机，故无法直接得出神经网络在 9 架无人机环境下训练效果优于 7 架无人机环境的结论。



(a) (b)

图3.7 不同数量无人机训练结果对比图

为进一步研究无人机数量在 7 架和 9 架时的最终收敛结果，在已经对神经网络训练了 30000 回合的基础上，对两种情况继续增加了 20000 回合训练，得到结果如图 3.8 所示。



(a) (b)

图3.8 拓充训练结果对比图

由图 3.8(a)可知，二者最终在 50000 回合训练后拟合度接近，但图 3.8(b)中 7 架无人机环境下的 1000 次前向传播的平均时间步 $\bar{T}_{step}$ 明显大于 9 架无人机环境下，即在式(3-13)中，前者以更高的前向传播时间步 T_{step} 计算得到的拟合度 P_p 和后者更低的前向传播时间步 T_{step} 计算得到的拟合度 P_p 得到的结果相近。那么可以认为 7 架无人机环境下的拟合能力优于 9 架无人机环境下的拟合能力。

从以上结果可知，本章所提出的神经网络框架具备基本的收敛能力，解决了离线强化学习在多智能体环境下难以收敛的问题。其中，在 3 架无人机的环境下，拟合结果表现较好，满足实际应用部署的要求。

实验 2：通信带宽与隐藏层维度对算法性能的影响。

为研究无人机数量与通信带宽和隐藏层维度的关系，本节进行了多组实验，考虑

到无人机数量为 3 架时已经具备了较好的收敛结果，故以下实验将基于 5 架、7 架和 9 架无人机的环境进行。相关参数与结果如表 3.2 所示。其中$\bar{T}_{step}$表示 1000 次前向传播后计算的均值。

表3.2 不同数量下的通信带宽与隐藏层维度训练效果对比

无人机数量（架）	通信带宽（字节）	隐藏层维度	拟合度P_p（%）	平均前向传播时间步$\bar{T}_{step}$
5	16	128	73.74	51.43
5	32	64	77.26	65.65
5	32	128	75.17	58.60
5	32	256	71.5	45.91
5	**64**	**128**	**78.26**	**77.51**
7	16	128	54.02	5.24
7	**32**	**64**	**69.89**	**15.19**
7	**32**	**128**	**75.43**	**15.16**
7	32	256	21.35	5.00
7	64	128	72.43	13.99
9	16	128	59.32	5.01
9	32	64	43.26	5.00
9	32	128	58.11	5.01
9	**32**	**256**	**73.58**	**6.65**
9	64	128	63.52	5.08

表 3.2 中对对应无人机数量下训练效果较好的参数组及其结果进行了加粗标注，由表 3.2 可知，训练效果并不随通信带宽和隐藏层维度的增加而增加，这一点在无人机数量为 5 架时较为明显。对于此现象的结论与具体分析如下。

通信带宽与神经网络隐藏层维度并不是越高越好，而是需要根据无人机数量适当进行调整。过低的通信带宽会导致无人机间传递的信息量不足，无法做出合适的决策。过低的隐藏层维度则会导致神经网络欠拟合，无法达到理想效果。若通信带宽过高，根据 3.2 节的分析，将会导致智能体等效观测$\tilde{o}^i$维度过高，同样也会出现欠拟合的情况。一方面，通信消息由隐藏层编码而来，故过高的隐藏层维度也会间接导致智能体等效观测$\tilde{o}^i$维度过高，从而出现欠拟合；另一方面，过高的隐藏层维度也会导致训练成本增加，需要进行更长时间的训练。故在实际应用中，应根据机群中无人机的数量来设置不同的通信带宽和隐藏层维度。

另外，因为每一架无人机可以用过自身的决策动作来控制与其他无人机的通信，而无人机错误的决策也会导致将消息发送至错误的目标无人机上。从 3.2 节化简得到的 MDP 模型的角度上看，这一错误将导致整个系统转移到另一个不可知的状态。且因为消息的数据内容随着网络参数的更新而更新，所以整个系统的状态空间维度很高。这使得拟合能力有限的神经网络无法准确提取特征，故在无人机较多的情况下，得到的拟合度不会很高。

3.7 本章小结

本章基于无人机互联环境对通信网络规划算法进行了初步设计，提出了NTC-Net，并对其有效性进行了验证。首先，基于第二章对现有算法的神经网络结构和训练方法的总结，针对本文所要解决的分布式网络拓扑控制问题，对算法框架进行了设计。其次对训练NTC-Net的神经网络所使用的仿真环境进行了设计。最后基于NTC-Net和仿真环境对模型进行了训练与仿真实验验证，并通过调整参数对NTC-Net性能进行测试，分析了参数调整与算法性能的关系，证明了算法的有效性，也发现了算法的不足。

第四章 有中继节点的无人机网络规划算法设计

4.1 引言

在FANET应用中，无人机之间的互联和与中继节点的连接成为了扩大组网范围的关键。中继节点在无人机网络中起到扩展通信范围、提高信号覆盖和优化网络性能的作用。因此，研究如何有效地利用中继节点来提高无人机网络性能显得尤为重要。

本章以NTC-Net为基础，针对无人机与中继节点的应用场景进行研究。首先对无人机通信网络框架进行设计，完善网络规划算法，为之后的算法改进和系统设计提供框架支撑。其次，在设计的网络框架基础上，对NTC-Net进行适应性改进，并根据改进的神经网络结构，对训练方法进行研究与设计，并对仿真环境进行对应拓展。最后，通过设计实验，在证明本章训练方法的有效性的同时，将其与第三章算法进行对比，并进行多组训练，评估本章算法性能及其存在的问题。通过以上研究，为之后的系统设计实现和未来更广泛的FANET应用奠定基础。

4.2 问题描述

在目前针对无人机网络规划的研究中，除了研究无人机之间通信用于集群控制外，也有将无人机作为路由节点实现通信网络对于用户的覆盖[65-66]。在提升网络覆盖的应用场景下，无人机除了和其他无人机进行通信外，也可能会和基站，有人机等中继节点进行通信，从而实现信息的远距离传播[67]，如图 4.1 所示。

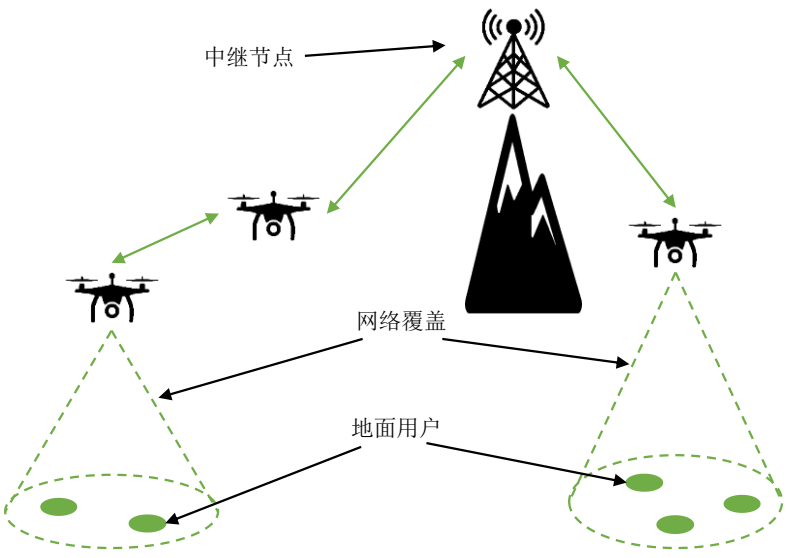


图4.1 利用无人机与中继节点提升网络覆盖场景示意图

考虑实际应用，中继节点一般为已部署的设备，可开发空间受限，因此在中继节点上部署的算法应尽可能轻量化，减少对节点的存储空间和计算资源的占用，以适应不同软件平台和不同计算能力的中继设备。这意味着占用资源较大的神经网络将无法在中继节点上进行部署。

另外，中继节点的主要功能为对数据的转发，即从一架无人机接收数据并将其发送到另一架无人机。为保证中继节点能高效工作，通过中继节点转发的数据应尽可能少，以减小对中继节点的通信带宽的占用，减小传输延迟，同时降低无人机之间的通信成本。

综上所述，本章需要解决的问题如下：

（1）深度强化学习算法与中继节点算法的兼容。

（2）无人机和中继节点之间的通信机制。

（3）神经网络生成的消息的压缩。

4.3无人机通信网络框架设计

算法的部署需要依托于当前技术与平台，故需要基于当前技术框架对算法的兼容性进行具体设计。对于拓扑控制与当前网络模型的关系一直存在争议，最近的一篇文献[68]对此做出了总结与解释，认为拓扑控制因为考虑了带宽、无线功率等硬件方面的约束，也考虑了整体网络延迟、节点负载等软件约束，应该介于网络层和数据链路层之间。考虑到设计难度，本章仅考虑固定中继节点的情况。

4.3.1 整体框架设计

考虑以上情况，基于第三章链路决策的框架，参考对当前拓扑控制与数据链路层的关系，同时考虑无人机集群的设计要求和算法功能的可拓展性，设计无线通信网络框架如下图 4.2 所示。

本文主要在数据链路层面上展开研究。图 4.2 中无线组网模块通过将环境观测与接收到的消息作为神经网络输入，得到编码数据与无线链路决策，从而与对应无人机建立通信，发送编码数据消息，并从全局角度形成最优链路拓扑。基于建立的拓扑，网络层的路由用于对其数据交互进行管理，通过寻找规划最优路径来实现对无人机之间上位机数据的发送与转发，使得无人机系统可在基于建立的链路拓扑上进行路由规划与数据和命令的交互。

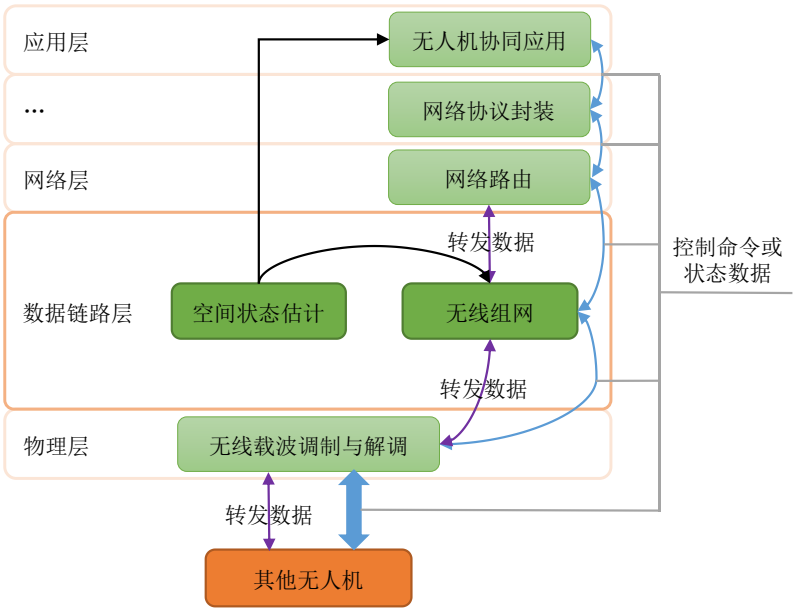


图4.2 具有决策能力节点的通信网络框架结构图

而对于无人机以外的没有决策能力的节点，设计其框架如图 4.3 所示。

图4.3 不具有决策能力节点的通信网络框架结构图

与无人机所用的框架不同，中继节点不存在与无人机相关的应用，故只需要对数据做转发，具体的转发方式与链路选择受到网络层以及其上层的控制。而其中的无线组网模块也与无人机所用模块不同，其模块只支持被动模式。

4.3.2 无线组网模块设计

无线组网模块作为本文研究中心，为更清晰地说明，给出模块内部结构及功能如图 4.4 所示。

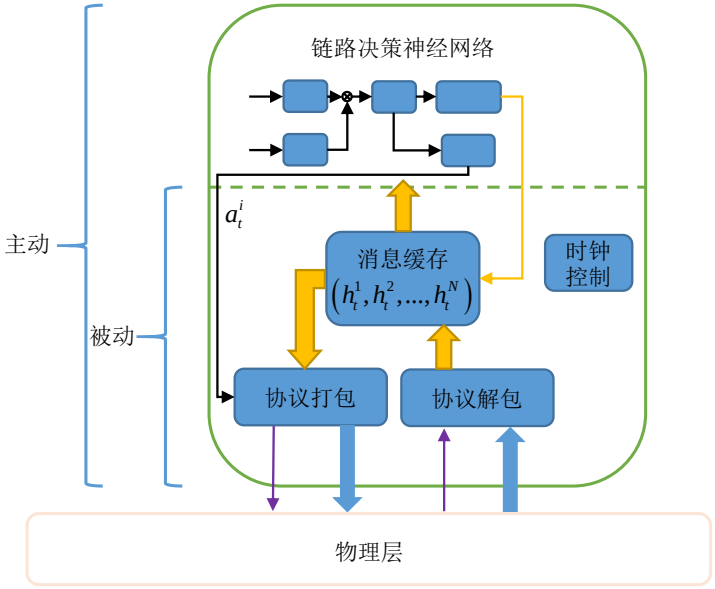


图4.4 无线组网模块内部结构及功能示意图

（1）链路决策神经网络

主体结构与第三章所述相同，其产生的消息编码会被事先保存在模块的消息缓存中，而其决策产生的动作会被载入到协议打包模块中，从而控制协议打包模块来阻止或允许数据的发送，对于网络层来说，协议打包模块实现的是直接对网络的通断进行控制。

（2）消息缓存

负责每个时间步内对输入缓存的消息进行保存。主动模式下，由时钟控制模块触发将保存的消息提交到神经网络后进行清空，而在被动模式下由时钟控制模块触发直接清空，而其中具体保存方式需要根据协议特征进行设计。

（3）协议打包

负责根据链路层协议对数据进行打包。在被动模式下，会将缓存中的所有数据根据协议进行打包，而在主动模式下，则只会将决策网络产生的消息数据根据协议进行打包。

（4）协议解包

负责将接收到的数据包解包并提交到网络层，同时取出其中的消息数据存入缓存中。若消息数量不为 1，则会根据消息组合方式判断数据是否能写入或覆盖缓存，以保护缓存中数据完整。

（5）时钟控制

负责与其他无人机进行时钟同步，以实现不同节点间功能的同步触发。无人机间的时钟同步发生在神经网络初始化阶段。此时单个节点的神经网络生成的第一条消息会通过广播的形式发送到每一个节点，在广播时会加入自身的系统时间，所有节点共

同采用编号最小的节点的时钟作为参考。

在主动模式下，也就是在无人机中，模块的发送受到来自网络层的控制，当网络层有要发送的数据，模块会根据决策动作判断是否要打包。若动作允许打包，则会对数据进行打包发送，若不允许，则直接返回网络层发送失败。

在被动模式下，模块有自动回复功能。当接收到来自其他节点的数据包时，模块会产生一个只包含消息数据的数据包作为回复，实现数据转发的功能。

4.3.3 通信协议设计

IEEE 802 是电气与电子工程师协会（Institute of Electrical and Electronics Engineers，IEEE）标准中关于局域网和城域网的技术标准，其限定在开放式系统互联通信参考模型（Open System Interconnection Reference Model，OSI）中的物理层和数据链路层。而无人机的组网大多使用局域网进行，如常用的 WiFi 基于 IEEE 802.11 标准，常用于无人机与地面站的连接；蓝牙、Zigbee 基于 IEEE 802.15 标准，常用于无人机之间的连接。故本节将根据 IEEE 802 数据链路层中的介质访问控制（Media Access Control，MAC）协议特征，对消息组合方式进行设计。

不同的 IEEE 802 的 MAC 协议在数据传输时有不同的控制策略，但其帧结构都包含了 MAC 头，载荷和 MAC 尾。一般来说，MAC 头用于标记帧的类型、帧长度、帧的源地址与目的地址等信息，而载荷则由用户自定义，MAC 尾一般用于对帧进行校验，保证数据帧在传输过程中的数据完整无误。其结构如图 4.5 所示。

MAC头	载荷	MAC尾
包含帧的属性以及地址信息等	用户自定义	一般用于校验

图4.5 IEEE 802通用MAC帧结构图

基于 4.3.2 节中的无线组网模块设计，当中继节点的无线组网模块接收到数据帧并成功校验后，会对载荷数据进行读取，并取出其中的消息数据，再将其他数据提交到上位层级，同时生成一个数据帧，将数据帧的相关属性和地址信息重新写入，并将缓存中的消息数据按格式写入载荷字段，最后发送该帧作为应答

与第三章使用的拼接的消息组合方式不同，在本章设计的框架下，中继节点的消息被保存在节点的缓存中，并在通信过程中将缓存数据发送出去。若继续采用拼接的方式，则在打包数据时消息将会占用大部分载荷空间。因此本章使用消息累加的方法对其占用载荷空间进行压缩，该过程的公式表示如下：

$$m^i = \sum_{j \in \boldsymbol{N}} f_{i,j}^{g}\left(h_m^j\right) \tag{4-1}$$

其中 h_m^j 表示缓存中的消息，若没有接收到来自智能体 j 的消息，则 $h_m^j = 0$。根据 4.3.2 节中设计的中继节点自动应答机制，设计 MAC 帧的载荷结构如图 4.6 所示。

载荷	
消息来源标记	消息
$\lceil N/8 \rceil \cdot 8$ 字节	$4n_{bd}$ 字节

图4.6　MAC帧载荷结构图

图 4.6 中 n_{bd} 代表消息 m^i 的维度，由于采用的是单精度浮点数表示，故其大小为 $4n_{bd}$ 字节。而消息来源标记使用的是标志位的方式，即一个无人机对应一个标志位。在实际通信时，会对两种情况进行处理。

（1）本地保存的消息是接收到的消息的子集

该情况下，将会使用接收消息对本地消息进行覆盖。

（2）所接收消息与本地消息交集为空

该情况下，将会对接收消息和本地消息进行相加合并，并对消息来源标记使用位或运算合并。

除此之外，节点将不对消息进行处理。

协议同时也考虑了节点间通信连接失败的问题，为降低因通信连接失败而导致的接收消息不全问题，在神经网络计算时间步间隔内，对无人机的网络层的路由提出了多次发送 HELLO 包的要求，以触发无人机对神经网络所生成消息的发送，其触发时序如图 4.7 所示。通过路由对自组网模块进行通信触发，提升至少有一次消息传递成功的概率。

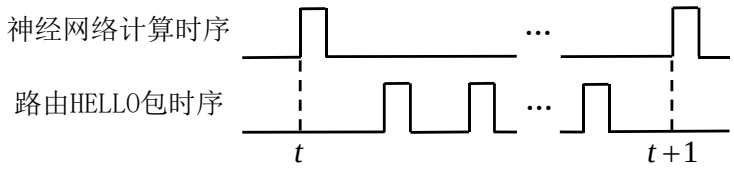


图4.7　HELLO包时序与神经网络计算时序关系示意图

4.4算法框架

4.4.1 神经网络结构

在本章的模型框架下，无人机间消息的组合方式相较第三章，由拼接方式改为加

和方式。这一点虽然增加了系统设计的复杂程度，但也实现了对传输消息的压缩。

对消息使用的加和的组合方式，将对消息本身的辨识度和神经网络对消息的辨识能力提出更高的要求。不同于消息拼接的组合方式，消息加和无法直观的分辨消息来源，当有无人机出现错误决策而将消息发送到错误无人机时，可能会导致接收到消息的无人机无法分辨其中是否含有误发送消息，从而做出错误决策，导致系统进一步发散。所以消息加和的方式实际上降低了系统的稳定性，为保持系统有较好的收敛效果，基于 NTC-Net 进行了神经网络结构和训练方法两方面的改进，提出了拓展网络拓扑控制神经网络（Extend Network Topology Control Model，ENTC-Net）。ENTC-Net 的神经网络结构如图 4.8 所示。对比 NTC-Net，ENTC-Net 的神经网络在解码器前级加入了低通滤波模块。

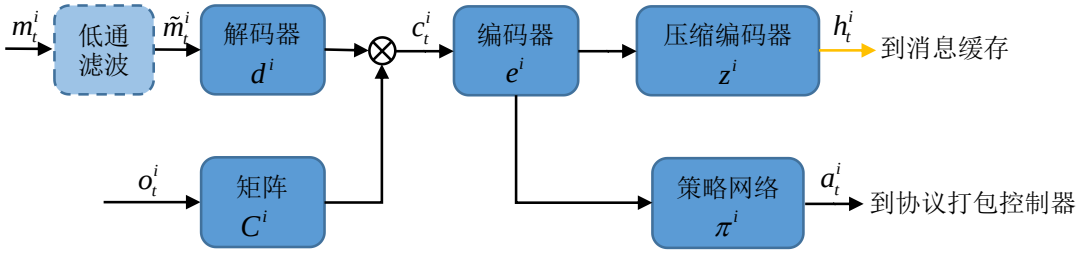


图4.8 神经网络结构图

4.4.2 低通模块

拓扑控制的要求之一就是链路稳定，即要求链路拓扑保持的时间越长越好，频繁的变换是不被接受的。另外，无人机运动状态是高阶连续的，所以其消息可以是连续的。故在对 ENTC-Net 进行训练时，通过加入低通模块来引导其形成对生成消息和决策缓慢变化的认知。

设计采用了无限脉冲响应（Infinite Impulse Response，IIR）滤波器作为低通滤波模块，相比有限脉冲响应（Finite Impulse Response，FIR）滤波器在更少参数下达到更好的滤波效果。IIR 滤波器的表达式如下：

$$y_{LS}(k)=\frac{\sum_{i=0} b_i x(k-i)}{1-\sum_{i=1} a_i y(k-i)} \tag{4-2}$$

为方便说明，本文以工程上常用的二阶巴特沃斯低通滤波器为例，对低通滤波模块进行进一步介绍。同时为使低通滤波模块的运算方便简洁，将其设计为循环神经网络的形式，其公式表示如下：

$$h_{LP}(k)=\begin{bmatrix}0&0&0&0\\1&0&0&0\\b_1&b_2&a_1&a_2\\0&0&1&0\end{bmatrix}h_{LP}(k-1)+s_{LP}(k)\begin{bmatrix}1\\0\\b_0\\0\end{bmatrix} \tag{4-3}$$

$$y_{LP}(k)=\begin{bmatrix}0&0&1&0\end{bmatrix}h_{LP}(k) \tag{4-4}$$

其中 s_{LP} 为输入序列，y_{LP} 为输出序列，$h_{LP}(k)$ 为滤波器隐藏层向量。b_0、b_1、b_2、a_1、a_2 为滤波器参数，其表达式如下：

$$b_0=\frac{\Omega_\alpha^2}{c} \tag{4-5}$$

$$b_1=2b_0 \tag{4-6}$$

$$b_2=b_0 \tag{4-7}$$

$$a_1=\frac{2\left(\Omega_\alpha^2-1\right)}{c} \tag{4-8}$$

$$a_2=\frac{1-\sqrt{2}\Omega_\alpha+\Omega_\alpha^2}{c} \tag{4-9}$$

其中，$\Omega_\alpha=\tan\left(\pi f_c/f_s\right)$，$c=1+\sqrt{2}\Omega_\alpha+\Omega_\alpha^2$。在初始化滤波器时，仅需确定截止频率 f_c 和采样频率 f_s 即可确定滤波器参数。

根据上式，定义低通滤波模块表达如下：

$$f_{LP}\left(s_{LP}(k)\right)\triangleq y_{LP}(k) \tag{4-10}$$

4.4.3 低通滤波作用于神经网络的收敛性证明

为说明低通滤波模块的加入对神经网络的影响，本节将设计一个简单的对比实验，提供低通滤波模块使用对连续数值输入和输出的 RNN 训练效果提升的充分性证明。

实验所用 RNN 神经网络结构如图 4.9 所示。

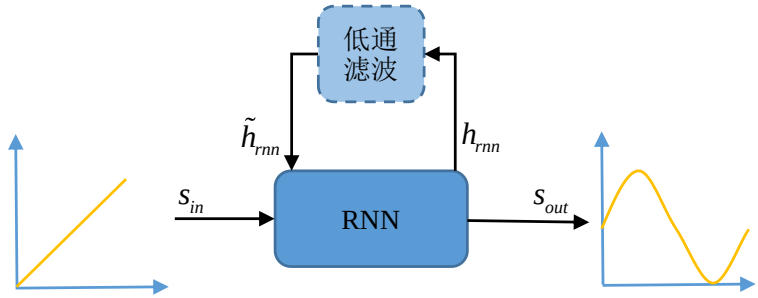


图4.9 实验内容示意图

如图 4.9，构建一个简单的 RNN 网络，使用线性自增的序列作为输入，对输入进行非线性映射得到输出序列。对于输出序列的选择，考虑到需要一个拟合难度高且变化缓慢的序列，可以满足本章对策略网络决策缓慢变化的要求。故选取一个具有周期性质的正弦函数作为输出序列。

实验组中，在 RNN 的隐藏层循环中加入了一个低通滤波模块，用于对上一时刻的隐藏层数据进行滤波，即 $\tilde{h}_{rnn} = f_{LP}\left(h_{rnn}\right)$。而对照组去除了低通滤波模块的应用，即 $\tilde{h}_{rnn} = h_{rnn}$。分别对上述两组神经网络进行 100 次实验，每次进行 10000 次训练，相关参数如表 4.1 所示，其损失随训练回合下降关系如图 4.10 所示。

表4.1 RNN训练超参数

参数名称	数值
输入序列	$s_{in}(k) = 0.01k$
输出序列	$s_{out}(k) = \sin\left(4\pi s_{in}(k)\right)$
序列长度	100
隐藏层维度	8
低通滤波采样频率 f_s	100
低通滤波截止频率 f_c	10
RNN 激活函数	tanh

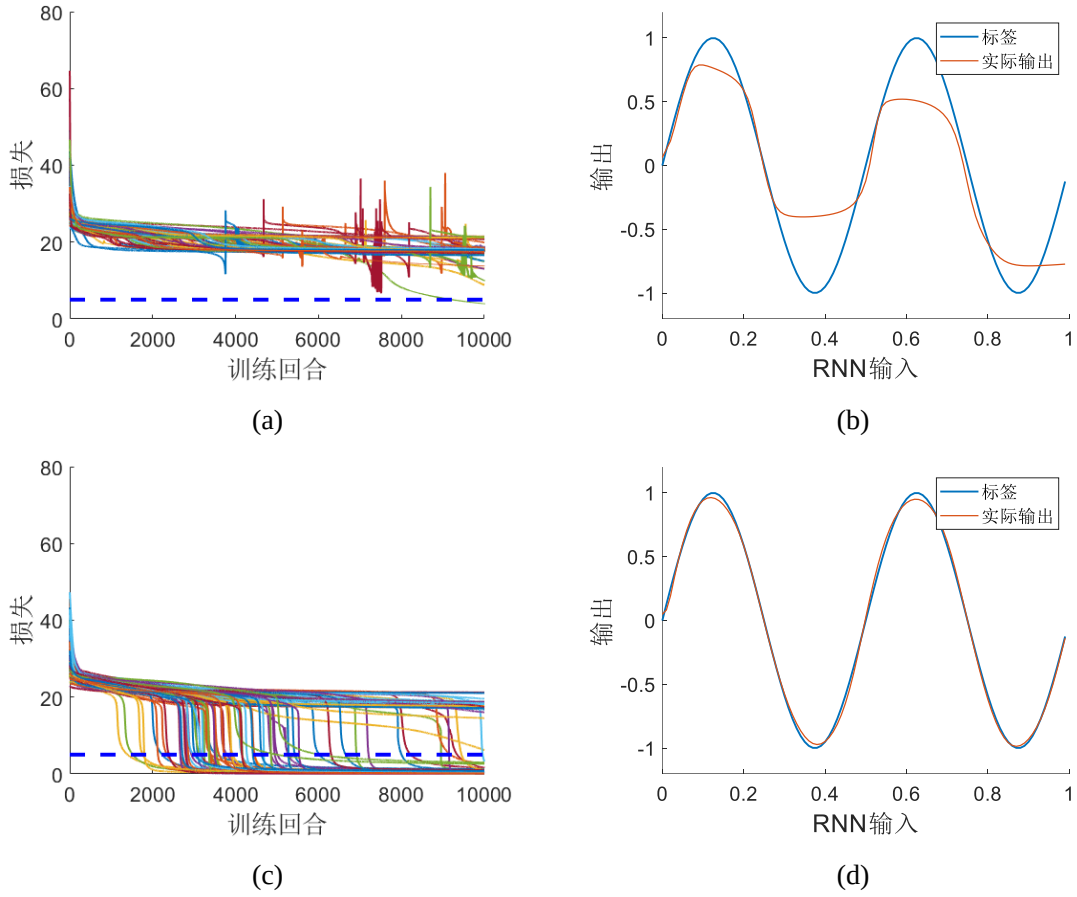


图4.10 有无低通滤波的RNN训练结果对比图

图 4.10(a)为对照组的 RNN 的训练损失随训练回合变化关系。图 4.10(b)为对照组在第 10000 回合时达到的最小损失 4.018 的实验输出的结果；图 4.10(c)为实验组的 RNN 的训练损失随训练回合变化关系。图 4.10(d)为实验组在第 10000 回合时达到的最小损失 0.0453 的实验输出的结果。

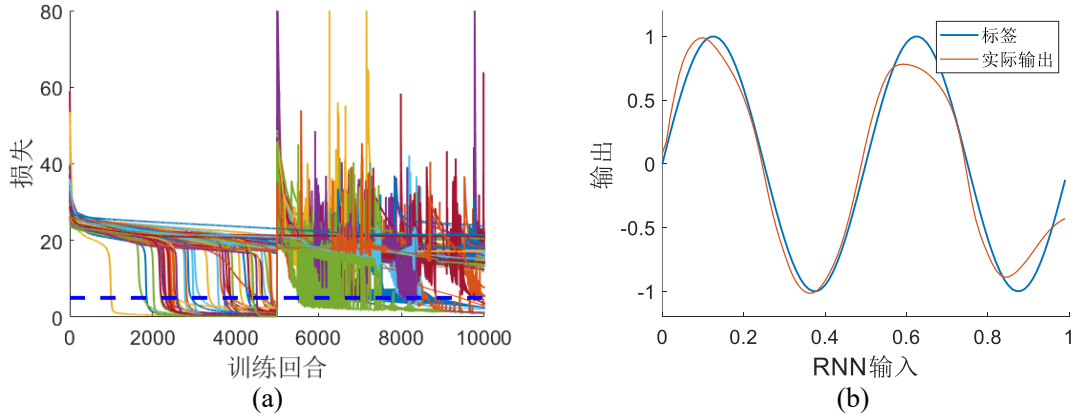


图4.11 5000回合后关闭低通滤波的训练结果对比图

为进一步证明低通模块的存在对 RNN 的收敛没有必要性关系，在对有低通滤波模块的 RNN 迭代到一半次数时关闭模块，即令 $\tilde{h}_{rnn}=h_{rnn}$，继续完成之后的训练。得到结果如图 4.11 所示。同时为方便判断，以最后一次迭代时的损失小于等于 5 作为收敛率计算依据。收敛结果对比如表 4.2 所示。

表4.2 RNN收敛结果对比

	最小损失	收敛数量
无低通滤波	4.0180	1
有低通滤波	0.0453	75
使用低通滤波预训练	0.7840	8

由以上实验现象可知，在没有使用低通滤波模块时，RNN 几乎不收敛。而加入低通模块后，RNN 大量收敛，且在中途关闭低通滤波模块对 RNN 隐藏层作用时，仍可以保证部分 RNN 收敛。可见，低通滤波模块的加入使得输出数值变化缓慢的 RNN 训练效果有明显提升。

从控制系统的角度来说，对 RNN 的隐藏层向量加入低通滤波模块，相当于对闭环控制系统的反馈路径加入了低通滤波，为整个系统增加了零点和极点。即对 RNN 的训练过程引入了约束，当 RNN 的输入输出为某个函数关系映射时，低通滤波模块可以引导训练过程提取出对应的映射关系特征。即使在训练过程使用模块并在之后将模块短路，依然可以得到相比原 RNN 较好的收敛效果。

4.4.4 训练方法

结合上述证明，将本章使用的环境表示为如下图 4.12 所示：

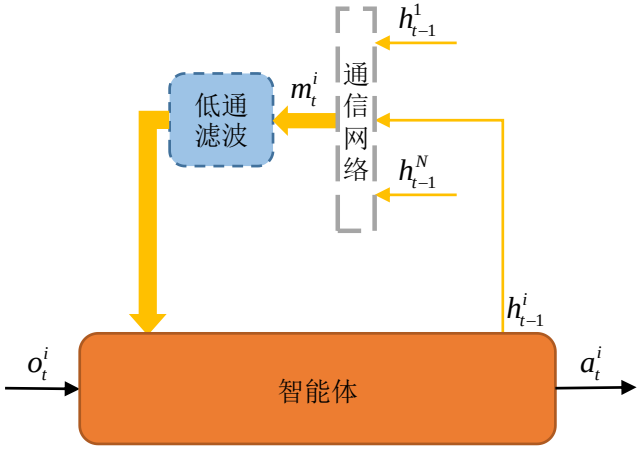


图4.12 网络消息传递与智能体关系示意图

由图 4.12 可知，因为智能体间消息传递类似于 RNN 中的隐藏层向量循环，并且存在高阶连续的状态观测 o_t^i 作为输入和变化缓慢的动作 a_t^i 作为输出，所以增加低通滤波模块对其训练过程进行约束，将获得更好的收敛效果。

ENTC-Net 的神经网络与实验所用 RNN 不同点在于前者结构更加复杂。首先，因为 ENTC-Net 中的多智能体相互通过通信建立连接，所以单个智能体神经网络的隐藏层数据是来自其他多个智能体神经网络隐藏层数据的组合。其次，低通模块是一个参数固定的模块，其截止频率固定，无法完全适应整个系统的带宽。因此，在设计训练方法时，需要在 4.4.3 节结论的基础上进行改进。

本章所设计的训练方法分为预训练和训练两个阶段，如图 4.13 所示。

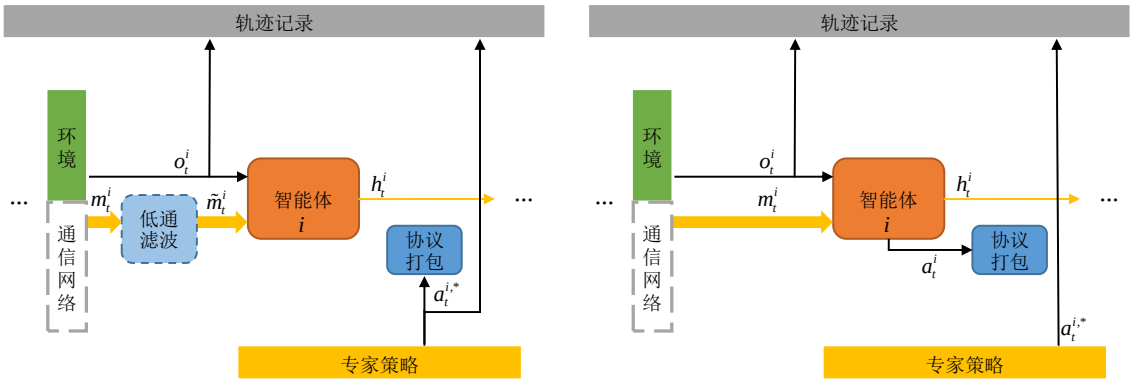


图4.13 神经网络前向传播过程示意图

（1）预训练

如图 4.13(a)所示，预训练时，网络链路决策直接由专家决策替代，保证智能体内的神经网络能够针对理想状态提取必要特征，形成通信协议。

（2）训练

如图 4.13(b)所示，正式训练时，将低通滤波模块短路，其链路决策使用智能体自身的决策，以训练神经网络进一步具备在有通信噪声环境下的鲁棒性。

综上所述，两个阶段的不同之处在于有无低通滤波器的接入和智能体的决策来源。

考虑到对于无人机的错误决策将导致其他无人机接收消息的改变，等价于其他无人机接收到的消息被噪声污染，而被噪声污染的消息会被带入神经网络反向传播的梯度中，导致训练效率降低。为降低噪声的影响，一方面在预训练时采用了专家策略替换智能体产生的策略，另一方面在整个训练过程中，会进行多次前向传播，再进行一次反向传播更新神经网络参数，用以降低噪声干扰。综上，所设计的训练算法如下：

算法 4.1： 神经网络训练

输入： 允许决策错误最大时间步 t_{cnt}，回报折扣 γ，奖励值 $r_{\max}$，允许最大时间步 T，最大KL散度 $D_{KL,\max}$，梯度优化函数 $f_{op}(x)$，最大批处理量 $b_{\max}$

初始化神经网络参数 θ
for $epoch = 1,2,...$ **do**
  if 预训练阶段 **then**
    接入低通滤波模块
  else
    短路低通滤波模块
  end if
  初始化批处理量计数 $b_{\max} = 0$
  while $b_{cnt} < b_{\max}$ **do**
    重置决策错误计数器 $cnt = 0$，初始化轨迹 τ，初始化消息 m_{init}
    将局部观测 o_0 和初始消息 m_{init} 输入智能体神经网络，得到编码消息 h_{init}
    将 h_{init} 按广播形式拼接为 m_0
    for $t = 0,1,...,T$ **do**
      根据 m_t 和 o_t 输出决策动作 a_t 和消息 h_t
      生成专家决策 a_t^*
      if 预训练阶段 **then**
        根据 a_t^* 生成全局网络链路图 G_{net}
      else
        根据 a_t 生成全局网络链路图 G_{net}
      end if
      根据 G_{net} 将消息组合为 m_{t+1}
      if $D_{KL}(\pi \| \mu) > D_{KL,\max}$ **then**
        $cnt = cnt + 1$
        if $cnt == t_{cnt}$ **then**
          break
        end if
      else
        $cnt = 0$
      end if
      记录轨迹 $\tau \leftarrow \left(\pi\left(s_t, a_t^*\right)\right)$
      $b_{cnt} = b_{cnt} + 1$
    end for
  end while
  更新神经网络参数 $\theta \leftarrow \theta - f_{op}\left(\nabla J^{\theta'}(\theta)\right)$
end for

4.5 仿真环境功能模块拓展

本节将基于 3.5 节开发的仿真环境，结合本章设计，对仿真模块的功能进行拓展，同时对本章所使用仿真环境相关设置进行说明。

首先对于对象管理模块中，需要加入中继节点对象。在实际环境中，有人机、地面基站、用户等都可以抽象为仿真环境中的中继对象。

（1）物理仿真模块

为保证训练过程中的随机性，其应用的物理仿真模块与无人机应用的模块相同，但设置其速度限幅函数 F_{sat} 中 $v_{\max}=0$，即设置神经网络单次前向传播过程中，中继节点位置固定，以降低 ENTC-Net 的训练难度。因为该对象一般不会和无人机处在同一高度层，所以在仿真过程中不会判断与无人机的物理碰撞和势场的影响。拓展后仿真环境界面如图 4.14 所示。

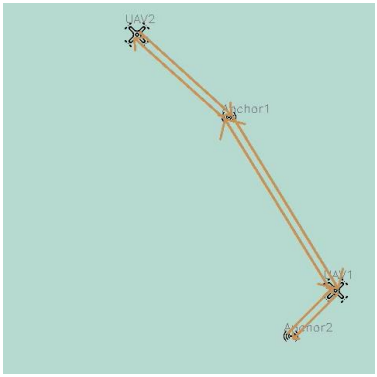


图4.14 仿真环境的中继节点拓展后显示界面

（2）网络仿真模块

在上一章介绍的网络仿真模块的基础上，根据本章内容，增加了对全局网络链路图 G_{net} 的计算。其中 G_{net} 是一个记录环境中所有节点的消息传播关系的邻接矩阵。在实际应用中，如图 4.7 所示，在神经网络进行前后两次决策的时间步内，所有节点会按决策的链路进行多次通信，其示意图如图 4.15 所示。

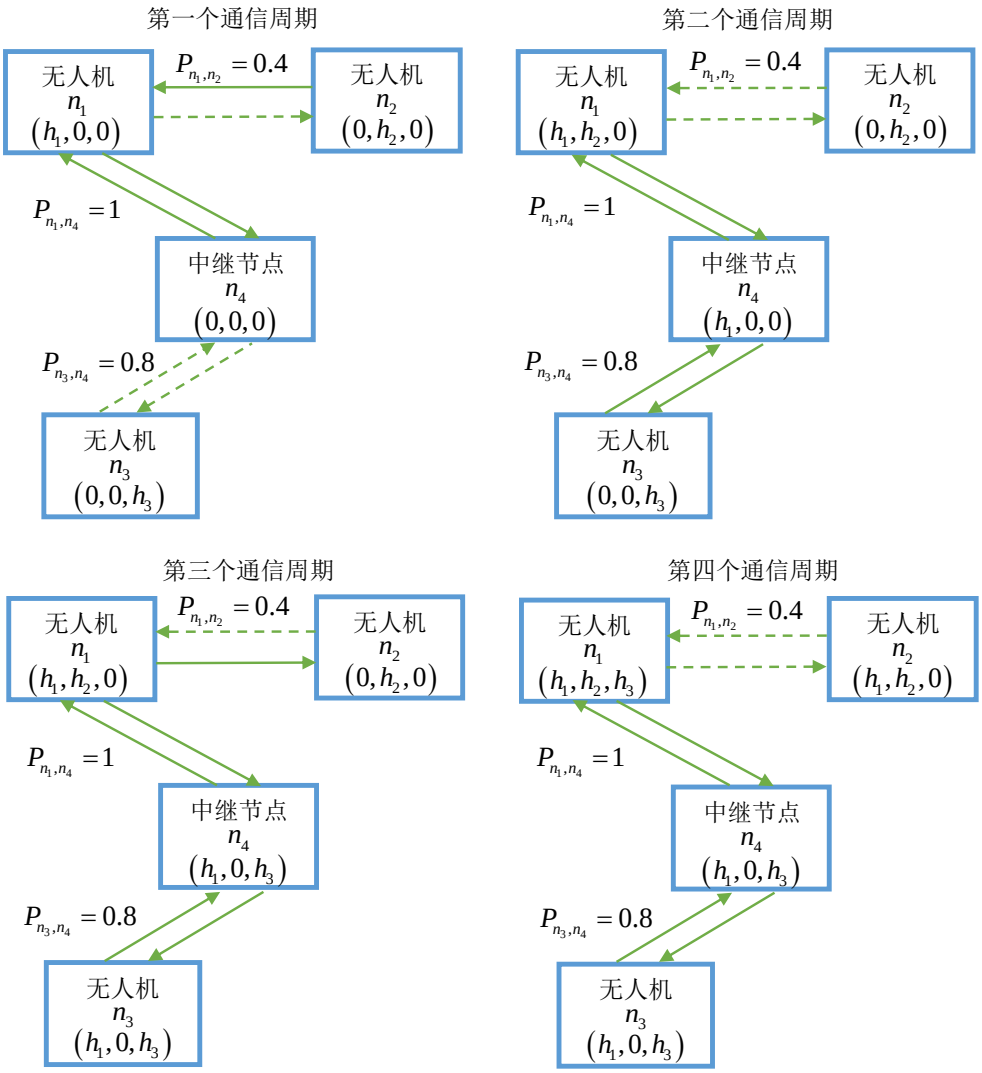


图4.15 网络通信仿真过程示意图

由图 4.15 可知，环境基于通信网络节点间仅存在按时间周期进行通信任务的假设，根据节点间数据传输成功概率 $P_{n_i,n_j}, i \neq j$ 来抽样并迭代计算节点中消息缓存的存储情况。图 4.15 中节点间的箭头表示神经网络决策允许的链路，实线箭头表示连接成功，那么它的消息将基于这个链路进行传输，反之，虚线箭头表示连接失败。节点中的括号内容表示节点的消息缓存状态。需要强调的是，为了避免无人机收到过多消息影响自身判断，只有中继节点会在通信时发送缓存中所有消息，而具有自主决策能力的无人机节点只会发送自身神经网络产生的消息。G_{net} 计算的伪代码如下：

算法 4.2： 计算消息传播图

输入： 通信周期数量T，节点数量N，随机数发生器均值μ方差σ，决策动作a_t

输出： 消息传播图G_{net}

1：初始化单位矩阵$G_{net}=I_{N\times N}$

2：根据a_t由网络仿真模块计算得P_{n_i,n_j}

3：**for** $cnt=1,2,...,T$ **do**

4：　**for** all $i,j\in\{1,2,...,N\}$ **do**

5：　　对随机变量$X\sim B\left(1,P_{n_i,n_j}\right)$抽样得到$x$

6：　　取G_{net}的j行i列元素$G_{net}^{i,j}$

7：　　$G_{net}^{i,j}=G_{net}^{i,j}$ and x

8：　**end for**

9：**end for**

4.6实验结果与分析

基于本章对于通信网络框架的设计和 NTC-Net 的适应性改进，本节将对 ENTC-Net 进行仿真实验，以测试算法的有效性以及性能。本节所使用的超参数与表 3.1 中大致相同，其他修改和增加的参数如表 4.3 所示。

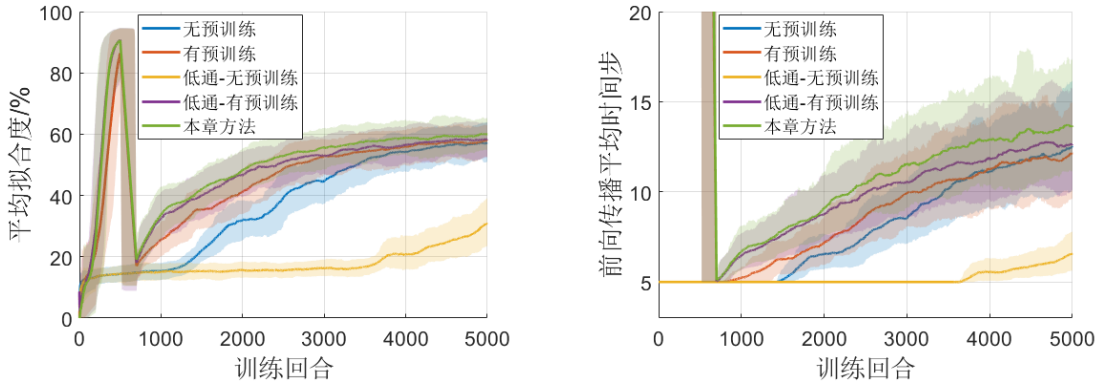
表4.3 神经网络训练超参数

参数名称	数值
预训练回合数	500
总训练回合数	5000
低通滤波模块采样频率f_s	10
低通滤波模块截止频率f_c	1
最大批处理量$b_{\max}$	100
允许最大时间步T	100
隐藏层维度	128

参照第三章结论，由于隐藏层维度的变化和通信带宽的变化存在一定耦合关系，但耦合关系不是本章研究重点，故本节将不再考虑隐藏层维度的变化。与第三章实验不同的是，本章的算法 4.1 在算法 3.1 的前向传播基础上增加了一个批处理循环，所以本节实验的拟合度P_p和前向传播时间步T_{step}都是一个批处理中的所有P_p和T_{step}的均值，即$\overline{P}_p$和$\overline{T}_{step}$。

实验 1：本章所设计的训练方法的有效性证明。

根据第三章的训练结果，选取了训练效果一般的 5 架无人机的环境，使用无人机互联的模型，并采用本章所设计的网络通信框架，对 ENTC-Net 进行了五组实验，结果如图 4.16 所示。五组实验分别为：（1）直接训练。（2）在全程屏蔽低通模块的情况下使用本章训练方法。（3）全程带低通滤波模块直接训练。（4）在全程使用低通模块的情况下使用本章训练方法。（5）使用本章的训练方法。



(a) (b)

图4.16 低通模块预训练效果对比图

由图 4.16 可知，预训练的作用在于引导神经网络前期提取必要特征以避开部分局部最优解，使得之后的训练过程中拟合度上升更快，拥有更好的训练结果。但低通滤波模块的存在对于网络训练过程不一定是完全有利的，因为低通滤波模块的截止频率固定，而无人机的速度是在一定范围内随机变化，所以低通滤波模块只能在预训练过程中引导神经网络提取一定的特征，在之后的训练过程中将其短路会取得更好的效果。

另一方面，对比第三章中图 3.7 同样在 5 架无人机下的结果，本章使用算法训练效果相对下降。故在使用算法时，应根据应用环境，对算法进行选择。

实验 2：环境中不同中继节点数量对算法性能的影响。

为研究环境中存在的中继节点对 ENTC-Net 性能的影响，实验在不同无人机数量下，在环境中的随机位置设置了 1-3 个中继节点，得到的实验结果如表 4.4 所示。本节实验默认在通信带宽为 16 字节的情况下进行。

表4.4 不同中继节点的训练结果对比

无人机数量	1 个节点		2 个节点		3 个节点	
	$\bar{P}_p$（%）	$\bar{T}_{step}$	$\bar{P}_p$（%）	$\bar{T}_{step}$	$\bar{P}_p$（%）	$\bar{T}_{step}$
2	84.01	29.14	59.73	18.27	42.04	9.59
3	64.38	24.53	45.45	13.30	32.30	6.59
4	38.45	6.70	29.08	5.13	15.26	5.00
5	28.84	5.00	21.37	5.00	10.86	5.00

由表 4.4 可知，在环境中加入中继节点后，模型依然具备收敛能力。虽然中继节点的位置在每次仿真环境复位时都会随机设置，且无人机事先不知道中继节点的位置，但无人机仍可根据仿真环境中提供的无线信号衰减模型，即根据式(3-20)来判断中继节点的位置，从而完成链路决策。但随着节点数量的增加，无人机除了需要对中继节点进行定位以外，还需要分辨中继节点，这使得对于中继节点定位的难度大大增加，使得无人机无法作出准确决策。而相对于无人机数量变化的角度，其对于中继节点的影响较小，但当无人机数量达到 5 架时，已无法收敛。

因此，在实际应用中，应尽可能地在训练过程中固定中继节点的位置，或将其位置限制在一定的范围内，以有效提升算法的收敛能力。相关效果演示将在 5.4.4 节中给出。

实验 3：设置无人机不同通信带宽对算法性能的影响。

因为 ENTC-Net 与 NTC-Net 最大的区别在于消息组合方式由拼接方式改为加和方式，故有必要对不同通信带宽对算法性能的影响进行研究。考虑到实验 2 中，当中继节点数量为 2 及以上时已无法达到较好的收敛效果，为保证实验数据对于超参数的灵敏度，本节实验设置在中继节点为 1 的环境下进行。与第三章实验相同，实验在通信带宽为 16 字节、32 字节和 64 字节的情况下进行，结果如表 4.5 所示。

表4.5 不同通信带宽的训练结果对比

无人机数量	16 字节		32 字节		64 字节	
	$\bar{P}_p$（%）	$\bar{T}_{step}$	$\bar{P}_p$（%）	$\bar{T}_{step}$	$\bar{P}_p$（%）	$\bar{T}_{step}$
2	84.01	29.14	87.99	31.17	88.66	32.86
3	64.38	24.53	70.40	32.57	70.89	34.05
4	54.93	11.16	62.80	15.75	63.57	16.64
5	28.84	5.00	35.97	5.32	38.50	5.55

由表 4.5 可知，增加通信带宽可以有效的增加最终的拟合度 $\bar{P}_p$ 和前向传播时间步

$\bar{T}_{step}$。相比第三章的实验结果，ENTC-Net 对通信带宽的增加表现得更加稳定，主要原因在于 NTC-Net 采用消息拼接的组合方式，通信带宽的增加与无人机数量存在倍数关系，这将大大增加通信协议的自由度，影响算法的收敛。而 ENTC-Net 则采用了消息加和的组合方式，通信带宽增加带来的正面影响大于负面影响，故在对算法进行部署时，可尽可能使用较高带宽。

4.7 本章小结

本章针对目前FANET的应用需求，基于NTC-Net，考虑中继节点的加入，对网络框架进行了设计，明确了有自主决策能力的无人机、地面站节点和无自主决策能力的中继节点的功能要求。基于所设计的网络框架，本章对NTC-Net进行了适应性改进，提出了ENTC-Net，同时提出了对通信消息加入低通滤波模块的训练方法。最后对ENTC-Net进行实验验证，一方面证明了本章所提出的训练方法显著提升了网络性能，另一方面通过调整参数对算法性能进行了测试，并与第三章结果进行对比。最终得出结论，NTC-Net与ENTC-Net各有优劣，应根据不同应用环境选择对应算法。

第五章 系统设计与实现

5.1 引言

在前几章的研究中，已经实现了无人机网络规划算法的设计，并在仿真环境中对其进行了训练与验证。然而，为将这些理论成果应用到实际场景中，还需要将算法部署到具体的无人机系统上，验证其在实际应用中的性能与可行性。因此，本章将设计一个软硬件系统，以支持NTC-Net和ENTC-Net的部署与应用，完成从模型训练到实际部署的全过程。

本章的研究将基于两方面展开。首先，根据整体系统框架，对无人机系统硬件进行设计，并对软件框架进行设计与编写。其次，基于UWB通信技术，将所设计的通信网络框架进行实现，设计对应的无线通信模块，并通过实际环境中的表现，验证算法和所设计系统的有效性。借助所设计的平台，将进一步提高算法的研究与开发效率，推动基于深度强化学习的无人机网络规划算法在实际场景中的广泛应用。

5.2 系统硬件设计

考虑本文使用的是深度强化学习算法，对硬件的计算能力有一定的要求。同时为方便无人机与训练环境以及地面站对接，无人机端采用机载计算机作为上位机与飞控下位机结合方式，由地面站与机载计算机连接进行控制命令与飞控状态交互，实现用户对无人机群的控制与状态观测。系统硬件结构及关系如图 5.1 所示。

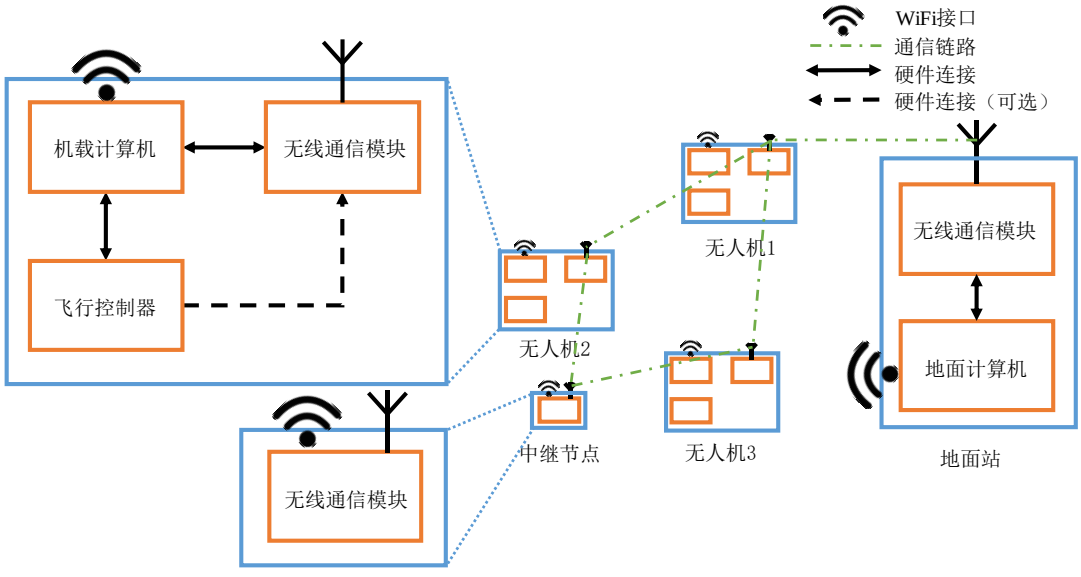


图5.1 系统硬件结构及其关系示意图

（1）无人机硬件

对于无人机的硬件，主要由无线通信模块和飞行控制器和机载计算机构成，

无线通信模块主要负责无线数据的收发工作。模块有单独的（Micro Control Unit，MCU）控制，当模块接收到空间中的无线信号时，将对信号进行解调、解码、校验等一系列工作，最终把符合设置的数据发送到机载计算机。同样的，机载计算机将数据和命令发送到无线通信模块后，模块负责对其进行封装并发送。对于有定位功能的通信模块，还需要与飞控进行连接，获取飞控的 IMU 数据，并由微控制单元 MCU 融合定位数据进行解算，从而获得无人机的飞行状态数据，并提交到机载计算机和飞控中。

飞行控制器主要负责无人机的六轴自由度控制，即三轴运动与三轴姿态。一般飞控除了其内部集成的捷联惯导模块，磁罗盘和气压计外，还包括外部的 GPS，光流和激光测距等传感器，用于实现对无人机位置以及速度的精确估计。

机载计算机主要有两项功能，一方面运行无人机协同应用，用户可以通过应用实现对飞控的命令发送，从而实现对无人机的协同控制；另一方面作为一个拓扑控制节点，实现无人机自组网功能。无人机协同应用与无人机自组网功能符合图 4.2 所描述的通信网络框架。

（2）地面站硬件

对于地面站的硬件系统，主要由无线通信模块和地面计算机构成。其中无线通信模块与无人机端的模块相同，而地面计算机同样包括用户应用与拓扑控制功能。不同的是，地面计算机使用的用户应用为地面站控制台，用于实现对所有无人机的状态观测和操控。

（3）中继节点硬件

对于中继节点，其硬件是一个可以独立运行的无线通信模块，此时模块会多装配一个 WiFi 模块作为调试接口。

系统中每个节点都带有一个 WiFi 接口，用于实现地面站对每个节点的控制和状态观测。需要注意的是，WiFi 接口仅用于辅助系统开发期间的调试工作，同时保证无人机飞行过程的安全，而不直接参与网络规划功能的实现。

综上所述，设计无人机硬件结构如图 5.2 所示。

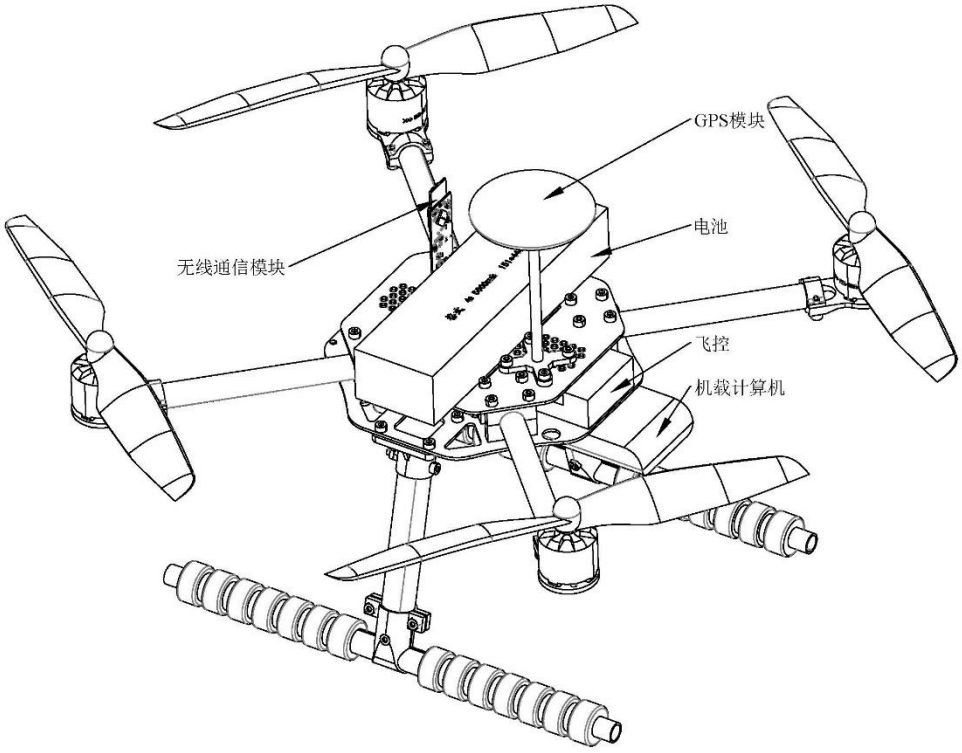


图5.2 无人机硬件结构示意图

5.3 系统软件设计

基于以上硬件设计，在软件方面，根据其功能划分为上位机软件和下位机软件。上位机软件基于 Linux 下的机器人操作系统（Robot Operating System，ROS）系统开发设计，运行于机载计算机和地面计算机中。下位机软件基于 RT-Thread 内核开发设计，运行于无线通信模块中。无人机的飞控使用了 PX4 开源飞控，不涉及本节设计范围内，故不做单独阐述。

5.3.1 上位机软件

由于上位机软件同时包含了机载计算机软件和地面计算机软件，且二者功能有重叠，为提高开发效率，降低维护难度，在开发时二者使用同一套框架，对相同模块进行复用，并基于各自功能进行拓展，其框架如图 5.3 所示。其中 ROS Core 是 ROS 系统的内核，并不属于软件中模块设计的一部分，但在设计中 ROS Core 起到了各个模块间通信和同步的作用，并通过传输控制协议（Transmission Control Protocol，TCP）与地面站计算机的 ROS Core 通信，实现地面站计算机对调试数据的记录。因此为说明必要的模块间的联系，将其画出。

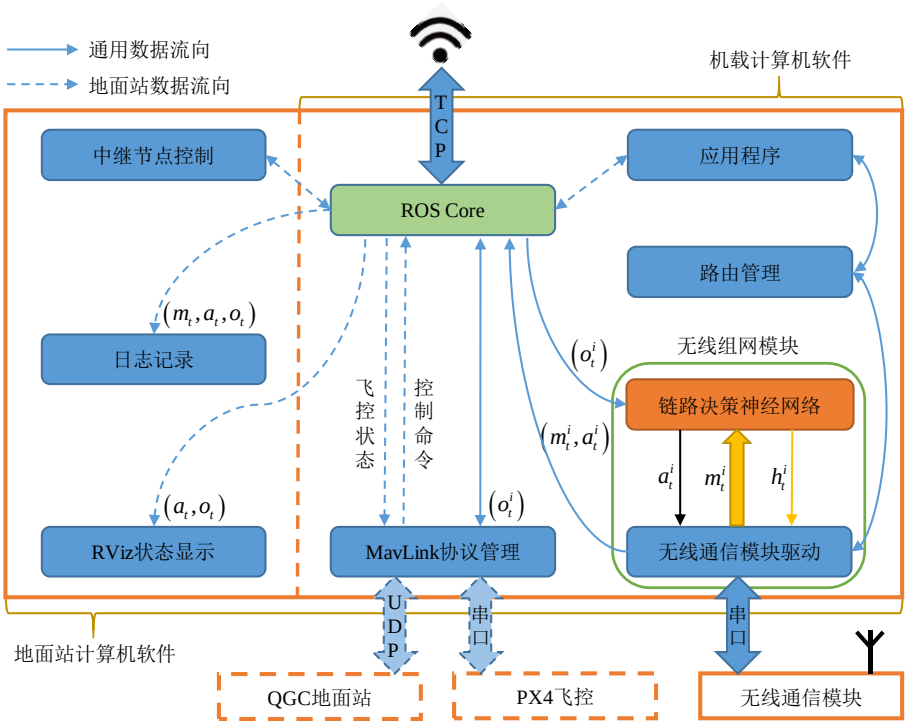


图5.3 上位机软件框架图

在图 5.3 中，机载计算机软件和地面计算机软件的区别可以解释为两点：第一，地面计算机相较机载计算机多了图中左侧模块，同时包含了虚线和实线在内的数据流向关系，可以认为地面计算机软件的功能是机载计算机软件的拓展。第二，机载计算机需要与飞控建立通信，故在对必要数据使用 Mavlink 协议进行处理后，通过串口与飞控建立通信；而地面计算机则需要对无人机进行控制，故连接对象替换为通过用户数据报协议（User Datagram Protocol，UDP）与 QGC 地面站，从而实现对飞控状态的观测与控制。

下面将对各个模块功能进行介绍，其中无线组网模块在 4.2 节中已进行过详细介绍，本节主要基于第四章所设计的通信网络框架进行了软件方面的实现，故在此不再介绍。

（1）中继节点控制模块

该模块通过与环境中的中继节点建立连接，并在地面计算机端提供命令行交互窗口，从而实现用户对中继节点的功能设置和对多个中继节点的管理。

（2）日志记录模块

该模块会记录环境中所有无人机飞行过程中的观测状态，同时记录神经网络的消息输入和动作决策输出，从而实现用户对飞行后的数据进行分析。同时日志记录也可为强化学习进一步的训练提供更加真实的数据集，提升系统鲁棒性。

（3）RViz 状态显示模块

该模块基于 ROS 内核下的 RViz 界面进行开发，用于实时显示无人机飞行轨迹与链路建立关系，方便用户观测与调试。

（4）MavLink 协议管理模块

该模块用于实现软件与 QGC 地面站或 PX4 飞控的耦合，其主要有两个功能。第一个功能是对地面站或飞控发出的 MavLink 数据帧进行解包，对要发送到地面站或飞控的数据进行打包。在软件的角度实现了与外部设备或软件的通信，在系统的角度实现了 QGC 地面站对 PX4 飞控的连接与控制。第二个功能是在机载计算机端保障无人机飞行安全。模块会自动向飞控发送心跳包，若出现特殊情况，如 WiFi 或无线模块断开连接，则停止发送心跳包，让飞控控制无人机降落。

（5）应用程序模块

该模块为用户自主开发的模块，可在模块内进行功能拓展。本文主要研究基于深度强化学习的网络规划，故对于该模块仅实现了地面站对无人机随机飞行的控制。

（6）路由管理模块

该模块功能对应 4.3.1 节中介绍的网络层的路由功能，考虑本文研究重心，并未对该模块进行深入开发，而是仅保留了 HELLO 包的定时发送，用于触发无线通信模块发送数据。

（7）无线通信模块驱动

该模块是一个驱动模块，对无线通信模块的串口通信协议进行了封装，仅提供了软件对无线通信模块的读写功能与部分参数设置功能。

5.3.2 下位机软件

下位机软件同时包含了机载无线通信和中继无线通信两种不同模式，同样考虑开发效率，二者将使用同一套框架，其结构如图 5.4 所示。

与上位机软件开发方式不同，下位机软件采用了字符终端触发的应用程序的开发方式，采用 RT-Thread 提供的 msh 字符终端来实现对应用程序的调用。下位机会通过检测串口是否被连接来控制 msh 连接到不同驱动接口上，实现 msh 的复用。

（1）机载无线通信模式

在该模式下，msh 通过串口与上位机实现交互，使得上位机可以通过 msh 对模块的开关、无线芯片发射功率和数据帧最大接收长度等参数进行设置。同时上位机也通过 msh 提交所要发送的数据和接收来自其他节点的数据。

（2）中继无线通信模式

在该模式下，无线组网模块按照图 4.4 所示的被动方式工作，路由管理模块被单独启动，其功能与图 5.3 中的路由管理模块功能相同。在此模式下，上位机可通过 WiFi 来实现对模块的参数设置。

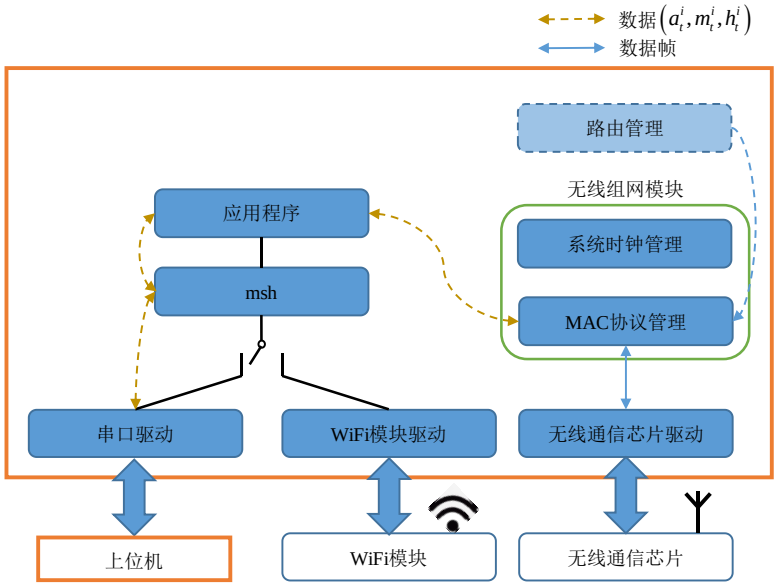


图5.4　下位机软件框架图

5.4 基于 UWB 的系统实现

UWB 是一种使用窄脉冲进行数据传输的无线载波通信技术，由于其具有功耗低，抗干扰能力强和频带宽的特点，被广泛应用于包括无人机通信在内的各种商业和军事领域。DW1000 是基于 UWB 技术的一款无线通信芯片，基于其生产的 UWB 模块 DWM1000 遵循 IEEE 802.15.4 协议，并集成了天线、射频电路、电源电路和时钟电路，功能强大且开发方便。本节将使用该模块作为无线通信模块对硬件展开设计并对本文所研究的系统与算法进行验证。

5.4.1 无线通信模块硬件设计

参照前述软硬件框架，设计无线通信模块如下图 5.5 所示。

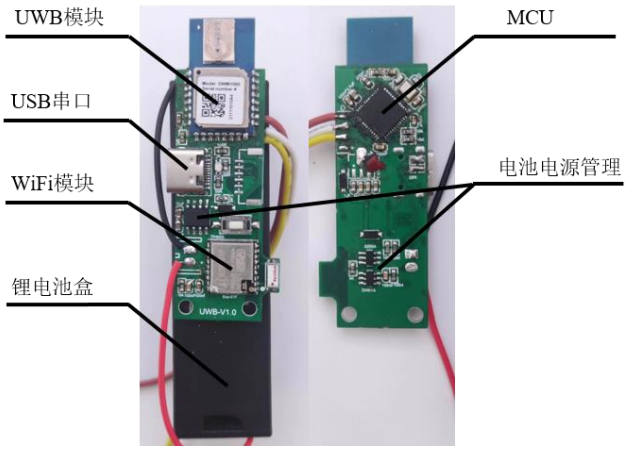


图5.5　无线通信模块硬件实物图

图 5.5 中所展示的为中继节点的硬件电路，对于无人机节点，仅需在中继节点的基础上，对 WiFi 模块、电池电源管理电路和锂电池盒不作焊接即可。模块所使用串口为 Type-C 接口，由 MCU 的 USB 外设通过 CDC 类虚拟串口实现，可使用数据线直接与机载计算机或地面计算机连接。

5.4.2 无线通信 MAC 协议设计

DWM1000 内部集成了 IEEE 802.15.4 的物理协议与 MAC 协议，可在其芯片内部实现对协议的自动收发与校验，减小 MCU 负担，增加通信效率，也为本文的框架验证提供了平台依据。其中 DWM1000 内所使用的 MAC 协议结构如图 5.6 所示。

MAC头							载荷	MAC尾
帧控制	序列号	目的PAN ID	目的地址	源PAN ID	源地址	安全头	载荷数据	校验位
2字节	1字节	0, 2字节	0, 2, 8字节	0, 2字节	0, 2, 8字节	0, 5, 6, 10, 14字节	可变长度	2字节

图5.6 DWM1000内部MAC协议结构图

图 5.6 中，帧控制位用于标记 MAC 帧的功能，包括信标模式、数据模式、应答模式和 MAC 命令模式。序列号用于对 MAC 帧进行计数，若数据帧发送成功则自动加一。目的 PAN ID 和目的地址表示要发送到的目的无人机的地址，同理源 PAN ID 和源地址表示自身的地址。安全头用于对载荷提供认证或加密功能，载荷数据是用户自定义的字段，校验位保存了对整个数据帧的校验码，用于保证数据传输的正确性。

PAN ID 与地址保证了数据帧的点对点发送。在相同频段下，一个信号源发出的电磁波会被一定空间范围内的节点所接收并解调。而 PAN ID 和地址的作用就是提供接收节点判断依据，数据帧内的 PAN ID 和目的地址与自身不同，则直接丢弃该帧。DWM1000 中在硬件上就集成了此功能，保证接收到的数据帧能被快速处理，减少冲撞概率。

根据 5.3.2 节中设计的下位机软件框架和 4.3.3 节中设计的通信协议框架，结合 DWM1000 的硬件特性，设计 MAC 帧协议如图 5.7 所示。

MAC头					载荷				MAC尾
帧类型	序列号	PAN ID	目的地址	源地址	保留	消息来源标记	消息	载荷数据	校验位
2字节	1字节	2字节	2字节	2字节	2字节	1字节	16字节	0-98字节	2字节
0x8861					0x0000				

图5.7 本文应用的MAC协议结构图

因为 DWM1000 对帧长度发送有限制，通过对帧类型的指定，可以将 MAC 头进行如图所示的化简，从而获得更大的载荷长度。由于本文实验所用无人机数量为 2 架，

故数据来源标记只需要一个字节，但仍在此之前设置了两个保留字节，一方面用于提供后面消息和载荷的字节对齐，提高程序效率；另一方面用于方便后续无人机数量的拓展，例如当采用第三章算法时，消息来源标记可以和保留字节合并，即不使用消息来源标记功能。另外，在初始化阶段，系统时间会写入载荷中，用于实现无线通信模块的时钟同步。

5.4.3 其他硬件

除对无线通信模块进行自主设计外，系统中其他硬件采用了现有模块或设备，如表 5.1 所示。

表5.1　系统设备模块配置

硬件名称	型号
机载计算机	MOREFINE J4125
飞行控制器	CUAV V5 nano
GPS 模块	Holybro M8N
WiFi 路由器	Redmi K40 热点

5.4.4 实验演示

实验选用了室外开阔场景进行测试，如图 5.8 所示。

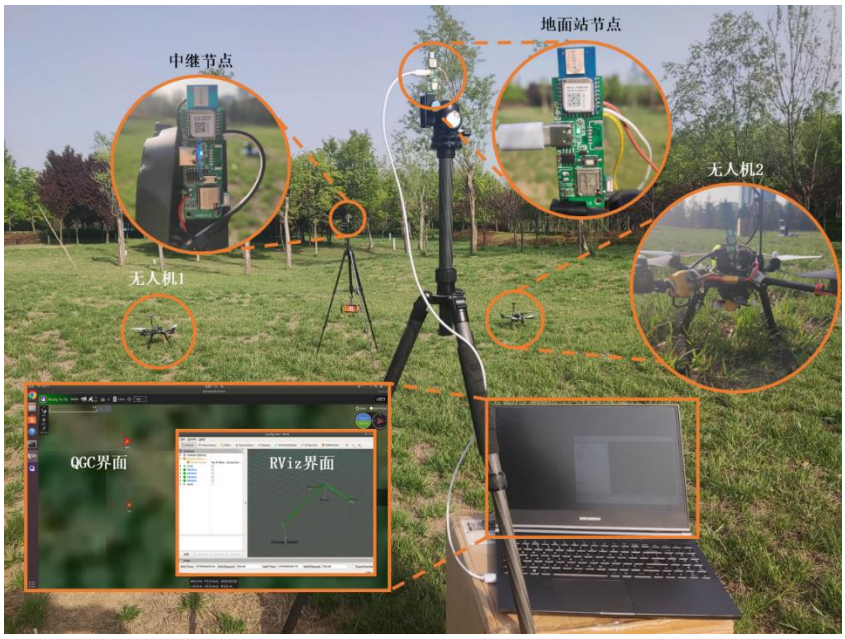


图5.8　测试现场

如图 5.8 所示，实验配置了作为主动模式运行的 2 个无人机节点、1 个地面站节点和作为被动模式运行的 1 个地面站节点。实验分为两组，分别基于第三章和第四章算法进行，测试算法在实际应用的可行性及其性能。

实验过程中，为保证安全，防止无人机意外相撞，在测试时将两架无人机高度分别控制在 2 米和 3 米，并控制水平随机运动。

由于飞控估计的状态中混有大量有色噪声，因此将来自飞控的状态观测作为神经网络观测输入会影响神经网络性能。所以在进行实际测试前，首先采集了 3 组飞行数据，并对记录的位置数据进行随机旋转和平移变换。将处理后的轨迹按 80%的概率替代仿真环境中的物理仿真模块输出轨迹，实现对神经网络进行短时间的训练，最后再进行部署。整个过程如图 5.9 所示。

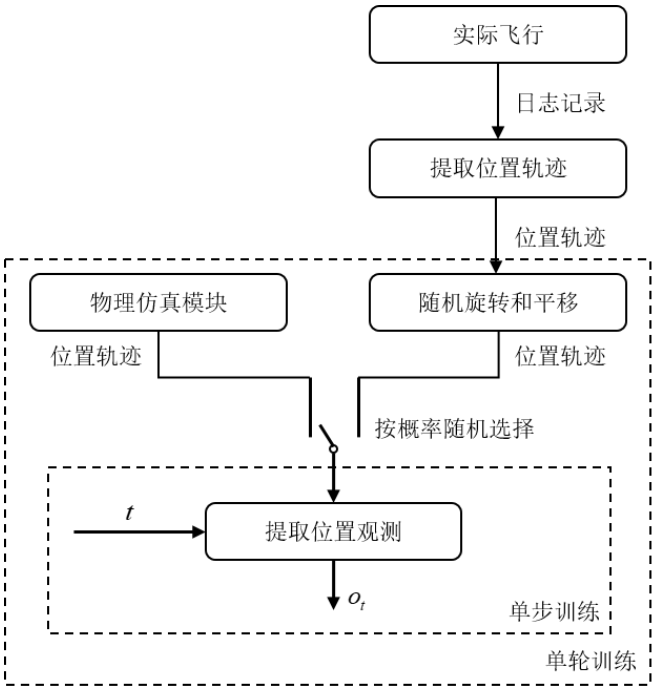


图5.9 部署前训练方法示意图

实验 1：无人机互联环境下的 NTC-Net 部署实验。

在本节实验中，中继节点并未被开启，仅使用了 2 个无人机节点和 1 个地面站节点，共 3 个主动节点。测试过程中上位机界面显示如图 5.10 所示。

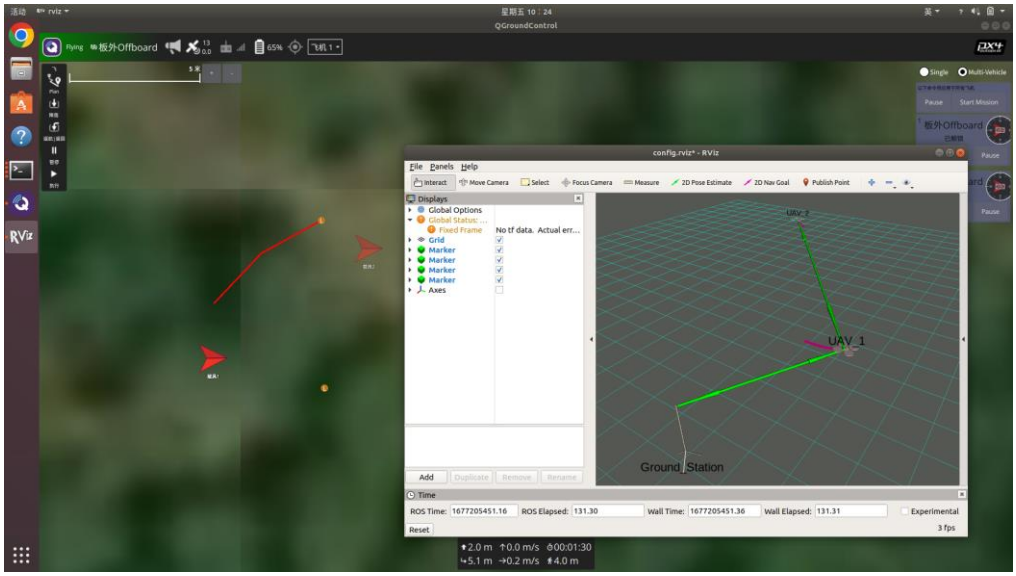


图5.10 无人机互联链路决策部署实验上位机界面

在整个飞行过程中，所有主动节点决策与全局专家决策的拟合度 P_p 如图 5.11 所示。

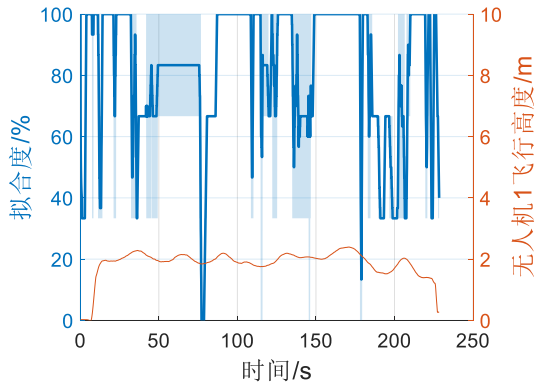


图5.11 无人机互联链路决策部署实验拟合度结果图

在图 5.11 中，因为所有节点由地面站计算机统一控制，所以使用无人机 1 的飞行高度来表示无人机的启停状态。对比拟合度 P_p 可知，在起飞前，由于无人机静止状态下训练数据集较少，在该状态下，拟合度相对较低。在系统启动的 78 秒左右出现了大量决策错误，但在之后得到恢复，表明算法具有一定稳定性。从整体来看，拟合度基本维持在较高水平，但阴影部分表明在决策过程中存在均匀震荡，这在训练过程中被允许，但对于实际链路建立是十分有害的。

实验 2：有中继节点的 ENTC-Net 部署实验。

本节实验中，在实验 1 的基础上开启中继节点，即在系统中存在 3 个主动节点和 1 个被动节点，无人机或地面可通过中继节点进行消息和数据转发。与 4.6 节实验不

同的是，本节实验在训练阶段对中继节点位置在一定范围内进行了限制，使其在每回合训练时中继节点的位置都在一个小范围内随机生成。在图 5.9 所示的部署前训练过程中，也只对无人机轨迹进行随机旋转与平移变换，而对中继节点的位置不作变换。测试过程中上位机界面显示如图 5.12 所示。

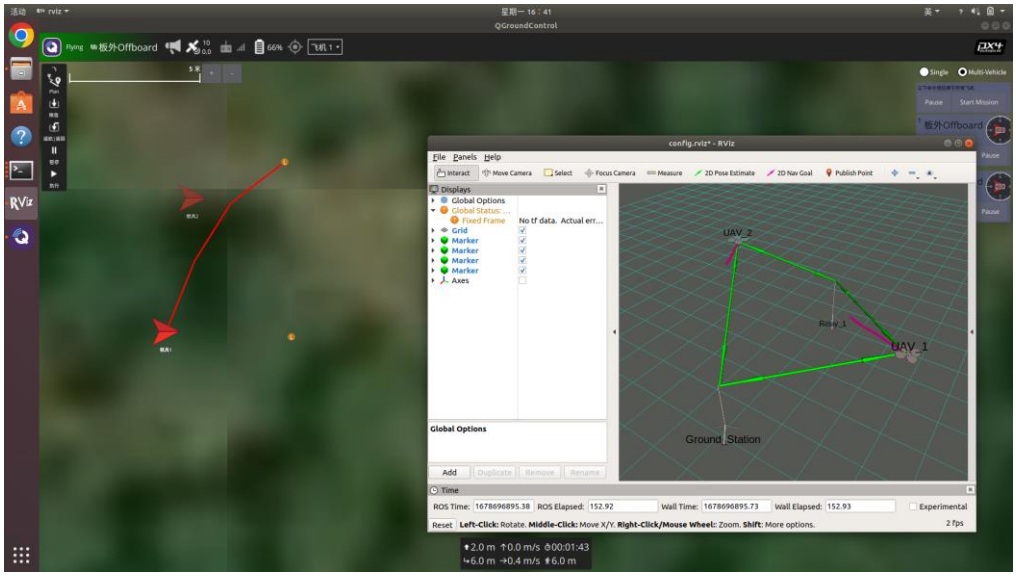


图5.12 有中继节点的无人机链路决策部署实验上位机界面

在整个飞行过程中，所有主动节点决策与全局专家决策的拟合度 P_p 如图 5.13 所示。

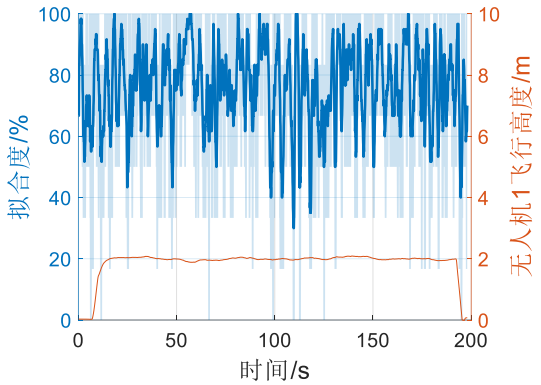


图5.13 有中继节点的无人机链路决策部署实验拟合度结果图

由图 5.13 可知，相比实验 1，链路决策过程的震荡更加严重，但整体依然具有较高的拟合度和一定的稳定性。

对比 4.6 节中的表 4.4 的内容，同样采用 3 架无人机 1 个中继节点的模型，其中地面站需要使用一个训练过程中的无人机模型。本节实验约束中继节点位置后得出的

整体拟合度为 76.03%，相比随机设置中继节点得出的 64.38%的拟合度有所提升，并且前者几乎能够稳定运行，而后者平均前向传播时间步为 24.53。所以在考虑部署时，应尽量确定中继节点位置，以获得较好的性能。

综上两个实验对本章所设计的软硬件进行了系统性验证，并得到了实际测量结果。现阶段的算法在具有主动决策能力的节点数量小于 3 时可以稳定运行。比较二者结果可知，在互联环境下链路拥有相对较好的稳定性，但应用场景局限，只能用于所有节点都具有决策和编码能力的环境下；有中继节点的环境下决策震荡较为明显，但能够适用于异构节点的环境。在实际部署过程中，应根据实际环境对算法进行选择。

5.5 本章小结

本章基于前两章对算法和网络通信框架的设计，将系统分为无人机硬件，地面站软件和无线通信模块软硬件三部分进行设计，并基于UWB通信技术对整体系统进行实现与实际飞行验证。实际飞行过程中整体系统运行稳定。通过所设计的系统测试了NTC-Net与ENTC-Net对全局决策的拟合性能，并进行对比，验证了应根据实际情况对算法进行选择的结论。

第六章 总结与展望

6.1 全文工作总结

本文主要围绕基于深度强化学习的分布式无人机网络规划算法开展研究。首先，针对目前分布式无人机网络拓扑规划与控制算法设计困难、对不同环境兼容性差等问题，提出了一套将集中式执行的网络拓扑控制算法通过通信强化学习框架映射到分布式执行的方法，提出 NTC-Net。其次，针对目前 FANET 的需求，对通信网络框架进行了设计，完善网络规划算法，并对方法进行适配改进，提出 ENTC-Net。最后，针对目前本文所提出方法尚未有对应软硬件平台支撑，设计了一套可用于本文算法研究及部署的软硬件平台。

论文取得的主要研究成果和创新内容如下：

（1）针对目前分布式无人机网络拓扑规划与控制算法设计困难、对不同环境兼容性差等问题，提出了一套通过集中式网络拓扑控制算法提供参考决策，自生成网络通信协议的强化学习模型 NTC-Net。为保证算法在无线链路受到干扰时的稳定性，本文对算法的仿真环境进行了设计，并加入了诸多干扰项，用于提升算法鲁棒性。实验结果表明，在无人机数量小于 9 时，通过配置合理的消息带宽和隐藏层维度，NTC-Net 可以得到一定的训练效果，保证算法收敛。

（2）针对目前 FANET 对网络覆盖范围和网络性能的需求，在 NTC-Net 的基础上进一步考虑无人机组网时有中继节点的情况，设计了一套通信网络框架，用于兼容无人机和中继节点间的通信，并根据设计的通信网络框架，对 NTC-Net 的消息组合方式进行适应性改进，提出了 ENTC-Net。同时，设计用于训练 ENTC-Net 的方法，使得对 ENTC-Net 的训练获得更好的收敛效果。实验结果表明，ENTC-Net 在有 2 个以下的中继节点环境下可以获得较好的训练效果，但在无中继节点环境下相比 NTC-Net 的效果有所下降，在部署应用时应根据实际情况进行选择。

（3）针对NTC-Net和ENTC-Net的部署和实现问题，对系统的硬件框架进行了设计，并根据所设计的通信网络框架，对系统软件框架进行了设计。根据所设计的系统框架，基于UWB通信技术，对无人机、地面站和无线通信模块的软硬件进行了实现。通过实际飞行对NTC-Net和ENTC-Net进行了部署测试，证明了系统的可行性，同时对两种算法性能进行测试，证明NTC-Net和固定中继节点下的ENTC-Net的有效性。

6.2 未来工作展望

本文针对目前分布式无人机通信网络规划算法设计困难的问题，基于深度强化学

习设计了一套完整的无人机通信网络拓扑控制系统，实现了系统从训练到最终部署的通路。然而本文的研究有待进一步深入，仍需进一步研究。关于后续的研究工作如下：

（1）算法收敛性能的进一步提高。目前 NTC-Net 和 ENTC-Net 仅在无人机和中继节点数量小于 3 时有良好的性能，可达到安全部署的要求。需要从训练过程中的梯度下降算法约束和专家策略分布两个方面入手改进，通过设计更加合理的梯度约束和提供单步下更加丰富的专家策略来减少神经网络得到的策略分布与专家策略分布的偏离程度，获得更好的训练效果，增加算法对无人机和中继节点的容量。

（2）算法运行过程中决策震荡的抑制。目前 NTC-Net 和 ENTC-Net 在实际运行过程中会出现链路策略频繁变化的情况，该情况将极大程度的影响链路稳定性，进而影响网络延时。应在训练过程中进一步增加约束来抑制算法出现不稳定的决策，增加算法产生策略与全局策略不同时的容忍度，引导神经网络向生成稳定决策的方向进行收敛。这一点可能会降低算法拟合度，但可以保证数据通过所建立的链路传输的稳定性。

（3）软件系统的进一步完善。目前软件系统还处在初期阶段，功能丰富但集成程度不高，不利于快速安装与上手。同时各个组件涉及不同编程语言，组件间通信协议多样，开发流程复杂，难以满足本文研究的进一步深入的要求。需要对整体程序框架、数据结构和通信协议进行统一，最终实现对所有组件的集成，从而提升软件系统的开发效率和算法的训练、调试和部署效率。

参考文献

[1] Xing L, Fan X, Dong Y, et al. Multi-UAV cooperative system for search and rescue based on YOLOv5[J]. International Journal of Disaster Risk Reduction, 2022, 76: 102972.

[2] Alsamhi S H, Shvetsov A V, Kumar S, et al. UAV computing-assisted search and rescue mission framework for disaster and harsh environment mitigation[J]. Drones, 2022, 6(7): 154.

[3] 鲜斌,王光怡,蔡佳明.多无人机吊挂负载运输系统的非线性鲁棒控制设计[J/OL].吉林大学学报(工学版),1998,1-13[2023-04-05].

[4] Xu Y, Guo R, Liu X, et al. Efficient face recognition via multi-UAV-edge collaboration in UAV delivery service[C]//2022 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom). IEEE, 2022: 676-683.

[5] 齐小刚,李博,范英盛等.多约束下多无人机的任务规划研究综述[J].智能系统学报,2020,15(02):204-217.

[6] Bekmezci I, Sahingoz O K, Temel Ş. Flying ad-hoc networks (FANETs): A survey[J]. Ad Hoc Networks, 2013, 11(3): 1254-1270.

[7] Mozaffari M, Saad W, Bennis M, et al. Efficient deployment of multiple unmanned aerial vehicles for optimal wireless coverage[J]. IEEE Communications Letters, 2016, 20(8): 1647-1650.

[8] Ding L, Han Q L, Guo G. Network-based leader-following consensus for distributed multi-agent systems[J]. Automatica, 2013, 49(7): 2281-2286.

[9] Xiong F, Zheng H, Ruan L, et al. Energy-saving data aggregation for multi-UAV system[J]. IEEE Transactions on Vehicular Technology, 2020, 69(8): 9002-9016.

[10] Zhu C, Dastani M, Wang S. A survey of multi-agent reinforcement learning with communication[J]. ArXiv Preprint ArXiv:2203.08975, 2022.

[11] Supowit K J. The relative neighborhood graph, with an application to minimum spanning trees[J]. Journal of the ACM (JACM), 1983, 30(3): 428-448.

[12] Gabriel K R, Sokal R R. A new statistical approach to geographic variation analysis[J]. Systematic Zoology, 1969, 18(3): 259-278.

[13] Shucker B, Bennett J K. Virtual spring mesh algorithms for control of distributed robotic macrosensors[J]. University of Colorado at Boulder, Technical Report CU-CS-996-05, 2005, 136.

[14] Leng S, Zhang Y, Chen H H, et al. A novel k-hop compound metric based clustering scheme for ad hoc wireless networks[J]. IEEE Transactions on Wireless Communications, 2009, 8(1): 367-375.

[15] Leonov A V. Modeling of bio-inspired algorithms AntHocNet and BeeAdHoc for flying ad hoc

networks (FANETS)[C]//2016 13th International Scientific-Technical Conference on Actual Problems of Electronics Instrument Engineering (APEIE). IEEE, 2016, 2: 90-99.

[16] Leonov A V. Application of bee colony algorithm for FANET routing[C]//2016 17th International conference of young specialists on micro/nanotechnologies and electron devices (EDM). IEEE, 2016: 124-132.

[17] Farmani N, Sun L, Pack D J. A scalable multitarget tracking system for cooperative unmanned aerial vehicles[J]. IEEE Transactions on Aerospace and Electronic Systems, 2017, 53(4): 1947-1961.

[18] Lin Q, Song H, Gui X, et al. A shortest path routing algorithm for unmanned aerial systems based on grid position[J]. Journal of Network and Computer Applications, 2018, 103: 215-224.

[19] Arafat M Y, Moh S. Localization and clustering based on swarm intelligence in UAV networks for emergency communications[J]. IEEE Internet of Things Journal, 2019, 6(5): 8958-8976.

[20] Wang B, Sun Y, Do-Duy T, et al. Adaptive D-hop connected dominating set in highly dynamic flying ad-hoc networks[J]. IEEE Transactions on Network Science and Engineering, 2021, 8(3): 2651-2664.

[21] Yan C, Xiang X, Wang C. Fixed-wing uavs flocking in continuous spaces: A deep reinforcement learning approach[J]. Robotics and Autonomous Systems, 2020, 131: 103594.

[22] Yan C, Low K H, Xiang X, et al. Attention-based population-invariant deep reinforcement learning for collision-free flocking with a scalable fixed-wing UAV swarm[C]//2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2022: 13730-13736.

[23] Watkins C J C H. Learning from delayed rewards[J]. 1989.

[24] Mnih V, Kavukcuoglu K, Silver D, et al. Playing atari with deep reinforcement learning[J]. ArXiv Preprint ArXiv:1312.5602, 2013.

[25] Silver D, Lever G, Heess N, et al. Deterministic policy gradient algorithms[C]//International Conference on Machine Learning. Pmlr, 2014: 387-395.

[26] Konda V, Tsitsiklis J. Actor-critic algorithms[J]. Advances in Neural Information Processing Systems, 1999, 12.

[27] Lillicrap T P, Hunt J J, Pritzel A, et al. Continuous control with deep reinforcement learning[J]. ArXiv Preprint ArXiv:1509.02971, 2015.

[28] Van Hasselt H, Guez A, Silver D. Deep reinforcement learning with double q-learning[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2016, 30(1).

[29] Mnih V, Badia A P, Mirza M, et al. Asynchronous methods for deep reinforcement learning[C]//International Conference on Machine Learning. PMLR, 2016: 1928-1937.

[30] Hessel M, Modayil J, Van Hasselt H, et al. Rainbow: Combining improvements in deep reinforcement learning[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2018, 32(1).

[31] Rajeswaran A, Kumar V, Gupta A, et al. Learning complex dexterous manipulation with deep

reinforcement learning and demonstrations[J]. ArXiv Preprint ArXiv:1709.10087, 2017.

[32] Rahmatizadeh R, Abolghasemi P, Behal A, et al. Learning real manipulation tasks from virtual demonstrations using LSTM[J]. ArXiv Preprint ArXiv:1603.03833, 2016.

[33] Bojarski M, Del Testa D, Dworakowski D, et al. End to end learning for self-driving cars[J]. ArXiv Preprint ArXiv:1604.07316, 2016.

[34] Schulman J, Wolski F, Dhariwal P, et al. Proximal policy optimization algorithms[J]. ArXiv Preprint ArXiv:1707.06347, 2017.

[35] Fujimoto S, Meger D, Precup D. Off-policy deep reinforcement learning without exploration[C]//International Conference on Machine Learning. PMLR, 2019: 2052-2062.

[36] Peng X B, Kumar A, Zhang G, et al. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning[J]. ArXiv Preprint ArXiv:1910.00177, 2019.

[37] Kumar A, Zhou A, Tucker G, et al. Conservative q-learning for offline reinforcement learning[J]. Advances in Neural Information Processing Systems, 2020, 33: 1179-1191.

[38] Yu T, Kumar A, Rafailov R, et al. Combo: Conservative offline model-based policy optimization[J]. Advances in Neural Information Processing Systems, 2021, 34: 28954-28967.

[39] Kostrikov I, Nair A, Levine S. Offline reinforcement learning with implicit q-learning[J]. ArXiv Preprint ArXiv:2110.06169, 2021.

[40] Filar J, Vrieze K. Competitive Markov decision processes[M]. Springer Science & Business Media, 2012.

[41] Gupta J K, Egorov M, Kochenderfer M. Cooperative multi-agent control using deep reinforcement learning[C]//Autonomous Agents and Multiagent Systems: AAMAS 2017 Workshops, Best Papers, São Paulo, Brazil, May 8-12, 2017, Revised Selected Papers 16. Springer International Publishing, 2017: 66-83.

[42] Zhang K, Yang Z, Başar T. Multi-agent reinforcement learning: A selective overview of theories and algorithms[J]. Handbook of Reinforcement Learning and Control, 2021: 321-384.

[43] Tampuu A, Matiisen T, Kodelja D, et al. Multiagent cooperation and competition with deep reinforcement learning[J]. PloS One, 2017, 12(4): e0172395.

[44] Lowe R, Wu Y I, Tamar A, et al. Multi-agent actor-critic for mixed cooperative-competitive environments[J]. Advances in Neural Information Processing Systems, 2017, 30.

[45] Foerster J, Assael I A, De Freitas N, et al. Learning to communicate with deep multi-agent reinforcement learning[J]. Advances in Neural Information Processing Systems, 2016, 29.

[46] Sukhbaatar S, Fergus R. Learning multiagent communication with backpropagation[J]. Advances in Neural Information Processing Systems, 2016, 29.

[47] Peng P, Wen Y, Yang Y, et al. Multiagent bidirectionally-coordinated nets: Emergence of human-level

coordination in learning to play starcraft combat games[J]. ArXiv Preprint ArXiv:1703.10069, 2017.

[48] Jiang J, Lu Z. Learning attentional communication for multi-agent cooperation[J]. Advances in Neural Information Processing Systems, 2018, 31.

[49] Singh A, Jain T, Sukhbaatar S. Learning when to communicate at scale in multiagent cooperative and competitive tasks[J]. ArXiv Preprint ArXiv:1812.09755, 2018.

[50] Kim D, Moon S, Hostallero D, et al. Learning to schedule communication in multi-agent reinforcement learning[J]. ArXiv Preprint ArXiv:1902.01554, 2019.

[51] Liu Y, Wang W, Hu Y, et al. Multi-agent game abstraction via graph attention neural network[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2020, 34(05): 7211-7218.

[52] Du Y, Liu B, Moens V, et al. Learning correlated communication topology in multi-agent reinforcement learning[C]//Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems. 2021: 456-464.

[53] Xiao J, Yuan G, He J, et al. Graph attention mechanism based reinforcement learning for multi-agent flocking control in communication-restricted environment[J]. Information Sciences, 2023, 620: 142-157.

[54] Yang Y, Ma X, Li C, et al. Believe what you see: Implicit constraint approach for offline multi-agent reinforcement learning[J]. Advances in Neural Information Processing Systems, 2021, 34: 10299-10312.

[55] Jiang J, Lu Z. Offline decentralized multi-agent reinforcement learning[J]. ArXiv Preprint ArXiv:2108.01832, 2021.

[56] Singh A, Jain T, Sukhbaatar S. Learning when to communicate at scale in multiagent cooperative and competitive tasks[J]. ArXiv Preprint ArXiv:1812.09755, 2018.

[57] Mao H, Zhang Z, Xiao Z, et al. Learning agent communication under limited bandwidth by message pruning[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2020, 34(04): 5142-5149.

[58] Abdolmaleki A, Springenberg J T, Tassa Y, et al. Maximum a posteriori policy optimisation[J]. ArXiv Preprint ArXiv:1806.06920, 2018.

[59] Wu Y, Tucker G, Nachum O. Behavior regularized offline reinforcement learning[J]. ArXiv Preprint ArXiv:1911.11361, 2019.

[60] Zhang K, Koppel A, Zhu H, et al. Global convergence of policy gradient methods to (almost) locally optimal policies[J]. SIAM Journal on Control and Optimization, 2020, 58(6): 3586-3612.

[61] Lagaris I E, Likas A, Fotiadis D I. Artificial neural networks for solving ordinary and partial differential equations[J]. IEEE Transactions on Neural Networks, 1998, 9(5): 987-1000.

[62] DeVries T, Taylor G W. Improved regularization of convolutional neural networks with cutout[J]. ArXiv Preprint ArXiv:1708.04552, 2017.

[63] Singh K K, Yu H, Sarmasi A, et al. Hide-and-seek: A data augmentation technique for weakly-supervised localization and beyond[J]. ArXiv Preprint ArXiv:1811.02545, 2018.

[64] Zhong Z, Zheng L, Kang G, et al. Random erasing data augmentation[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2020, 34(07): 13001-13008.

[65] You W, Dong C, Cheng X, et al. Joint optimization of area coverage and mobile-edge computing with clustering for FANETs[J]. IEEE Internet of Things Journal, 2020, 8(2): 695-707.

[66] Zhang C, Zhang L, Zhu L, et al. 3D deployment of multiple UAV-mounted base stations for UAV communications[J]. IEEE Transactions on Communications, 2021, 69(4): 2473-2488.

[67] Koushik A M, Hu F, Kumar S. Deep Q-learning-based node positioning for throughput-optimal communications in dynamic UAV swarm network[J]. IEEE Transactions on Cognitive Communications and Networking, 2019, 5(3): 554-566.

[68] Alam M M, Arafat M Y, Moh S, et al. Topology control algorithms in multi-unmanned aerial vehicle networks: An extensive survey[J]. Journal of Network and Computer Applications, 2022: 103495.