第四章　安全约束下的蜂群无人机分布式协同控制方法研究

在蜂群无人机协同突防过程，需要在保证集群内部安全的情况下对规划出来的轨迹进行跟随。上一章提出了基于强化学习的路径规划算法，这一路径是将蜂群无人机看作一个整体去规划出来的，而没有考虑集群内部每一架无人机节点的运动轨迹。因此，有必要设计合理的协同控制方法，根据预期的轨迹实现对无人机集群的控制。此外，由于蜂群无人机集群规模大，集中式算法会带来较大的通信压力。因此需要设计分布式架构，通过优化蜂群无人机内部的信息传递来减轻系统整体的通信负担。

本章将围绕蜂群无人机在保证集群内部安全的约束下的分布式协同控制方法。首先，基于领导者-跟随者架构，设计基于图神经网络的多智能体强化学习算法，用于输出集群内每架无人机的动作。其次，设计合理奖励函数和超参数，并利用 Python 在小规模集群中进行训练。最后，将训练的结果迁移到较大规模的蜂群无人机集群中并进行再次训练和测试，以证明算法的可行性。

4.1　问题描述

上一章节设计的蜂群无人机路径规划算法能够规划出躲避敌方威胁区的突防路线，因此本章的目的是在已知路线的情况，控制蜂群无人机对预期路线进行协同控制以实现集群的突防路线跟随。本章针对蜂群无人机协同突防中的分布式协同控制问题，给出了如图20的算法总体框架。首先，针对蜂群无人机突防场景下控制下的安全约束问题，考虑大规模集群中内部避碰的技术难点，引入深度强化学习算法，通过学习达到集群跟随与内部安全的平衡。其次，针对无人机存在通信约束，考虑无人机通信距离和通信数量限制，以及邻居节点变化问题，引入图神经网络作为特征提取网络，以实现对邻居信息的充分利用。

本节主要针对在已知规划的路径的前提下，带有内部避碰问题的大规模无人机集群的协同控制问题开展研究。在本章中，假设无人机间距离较为松散，特别是无人机之间没有上下重叠的情况。对松散编队，可以认为无人机之间没有气动力学的干扰，因此不妨假设无人机均处于大致相同的高度，将环境简化为二维环境。

如图21所示，本章基于领导者-跟随者架构设计集群控制算法。其中，领导者有且仅有一个，上面部署路径规划算法并获得跟随规划好的路线；其余无人机均为跟随者，通过跟随领导者从而实现集群的整体运动。在通信方面，考虑通信范围和最大通信数量

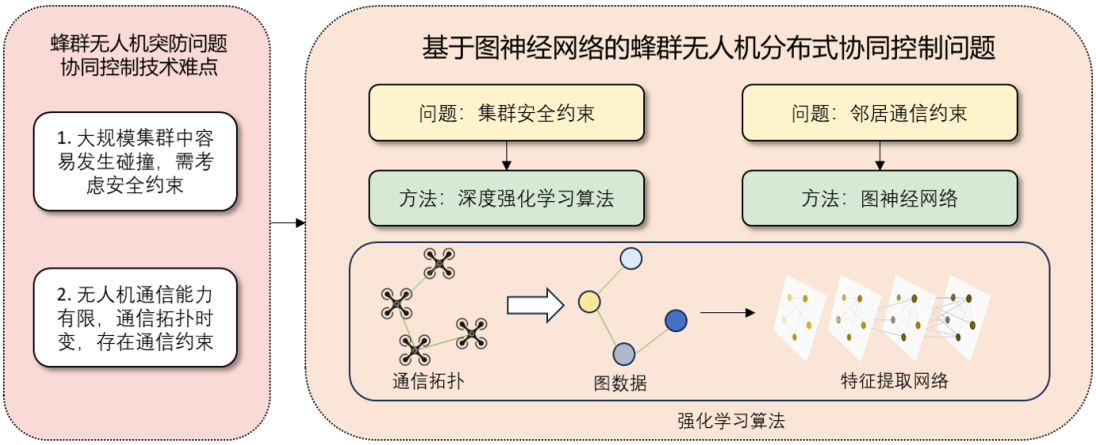


图 20　基于图神经网络的蜂群无人机分布式协同控制算法总体框架

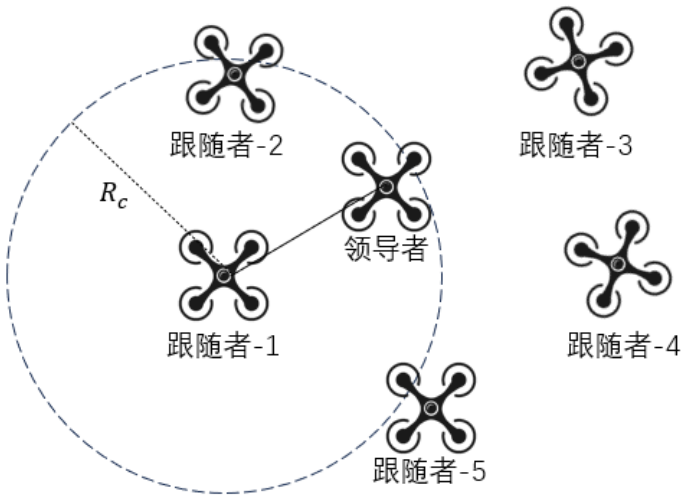


图 21　无人机蜂群

两种约束。在本章中，定义最大通信距离 R_c，每个无人机的最大通信数量为 n_c。换言之，集群中每个无人机只能与以自身为中心，半径为 R_c 的圆形区域内的邻居无人机通信，且最多只能获得并处理 n_c 个邻居无人机的信息。从反方面讲，当无人机 R_c 范围内没有任何友方无人机时，该无人机节点被认为失联。

因此，对于领导者，只需要使用单体的无人机控制器对规划的路径进行跟随，其不需要考虑其他跟随者无人机。由于领导者位置是集群运动的必要信息，本章假设领导者无人机时刻广播自己状态 s_0，每个邻居无人机均能获取到领导者无人机的状态。而对于每个跟随者无人机节点而言，其目的是不脱离无人机集群的同时与其他无人机保持安全距离 R_s。每个跟随者无人机都能够获取自身状态 s_i 和在通信约束下获取邻居节点状态 s_j。接下来，本章将设计基于图神经网络的多智能体强化学习算法，对每一个跟随者无人机给出控制输入 u_i 以实现集群飞行。

而对于强化学习算法来说，其输出是无人机的控制输入 u_i。由于无人机底层飞控的

存在，可以在多个控制输入层级中进行选择，如速度、加速度、角速度以及最底层的电机指令层。选择的控制输入越底层，控制的精度通常可以越高，响应时间越短；但另一方面，对强化学习所涉及的模型也变得越复杂，训练难度越大。考虑到松散编队对响应时间的需要和强化学习算法的训练难度，在本章中考虑无人机的加速度环作为外层集群控制器的输出，并作为底层飞控的控制输入。因此，在二维环境下，无人机模型被简化为一个二阶积分器系统，如公式4.1所示。

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{v_x} \\ \dot{v_y} \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ a_x \\ a_y \end{bmatrix}. \tag{4.1}$$

此时，每个无人机的状态为其二维位置和速度，控制输入为加速度。强化学习协同控制给出加速度控制指令后，无人机飞控在保持高度不变的情况下解算底层控制指令，以跟随外部控制器的加速度指令。

4.2 基于图神经网络的强化学习协同控制算法

4.2.1 DDPG 架构

考虑到强化学习协同控制算法的输出为加速度，即强化学习的动作空间应该是连续的二维空间，因此算法基于 Deep Deterministic Policy Gradient（DDPG）架构进行设计。

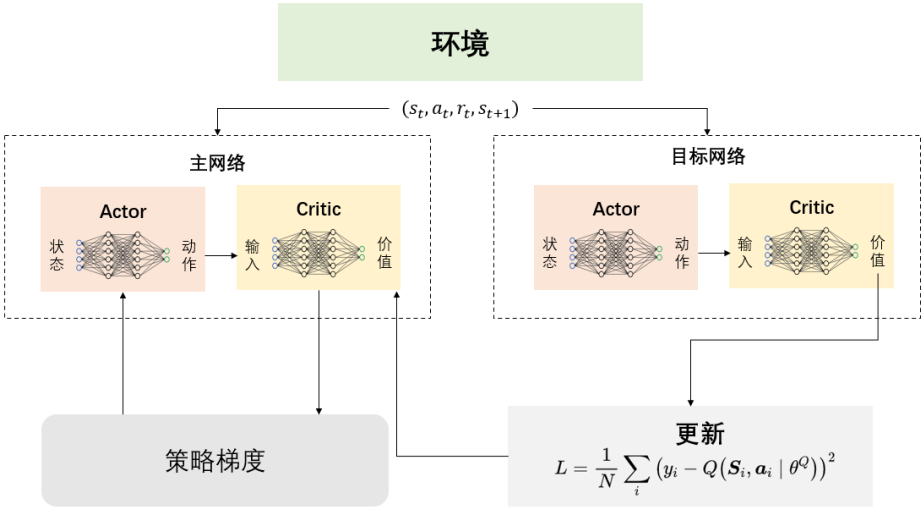


图 22 DDPG 架构图

DDPG 架构如图22所示。DDPG 主要包含四个网络，分别是两个 Actor 网络和两个 Critic 网络。Actor 网络是一个确定性的策略网络，其输入是智能体的状态，输出是一个

算法 3: DDPG 算法

随机初始化 Critic 网络 $Q\left(s,a \mid \theta^{Q}\right)$ 和 Actor 网络 $\mu\left(s \mid \theta^{\mu}\right)$ 的权重 θ^{Q} 和 θ^{μ}。
初始化目标网络 Q' 和 μ' 的权重：$\theta^{Q'} \leftarrow \theta^{Q}, \theta^{\mu'} \leftarrow \theta^{\mu}$。
初始化经验回放池 R。
for episode $= 1, M$ **do**
　初始化环境和动作探索过程。
　获得智能体的初始化观测是 s_1。
　for $t = 1, T$ **do**
　　根据当前 Actor 网络输出并添加探索噪声以选择动作：
　　$a_t = \mu\left(s_t \mid \theta^{\mu}\right) + \mathcal{N}_t$。
　　执行动作 a_t 并获得奖励 r_t 和新时刻状态 s_{t+1}。
　　将一组更新数据 (s_t, a_t, r_t, s_{t+1}) 添加到经验回放池 R 中。
　　if 经验回放池大小 > 批大小 N **then**
　　　从经验回放池中随机选择大小为 N 的 minibatch。
　　　定义 $y_i = r_i + \gamma Q'\left(s_{i+1}, \mu'\left(s_{i+1} \mid \theta^{\mu'}\right) \mid \theta^{Q'}\right)$
　　　定义 Critic 网络损失为 $L = \frac{1}{N}\sum_i \left(y_i - Q\left(s_i, a_i \mid \theta^{Q}\right)\right)^2$。
　　　利用梯度下降最小化 Critic 网络损失并更新 Critic 网络损失权重 θ^{Q}。
　　　更新动作网络权重：

$$\nabla_{\theta^{\mu}} J \approx \frac{1}{N}\sum_{j} \nabla_a Q\left(s,a \mid \theta^{Q}\right)\Big|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^{\mu}} \mu\left(s \mid \theta^{\mu}\right)\Big|_{s_i}$$

　　　更新目标网络：

$$\theta^{Q'} \leftarrow \tau\theta^{Q} + (1-\tau)\theta^{Q'}$$

$$\theta^{\mu'} \leftarrow \tau\theta^{\mu} + (1-\tau)\theta^{\mu'}$$

　　end
　end
end

具体的动作，适用于连续动作空间的问题。在本章中，该网络将输出无人机的加速度。Critic 网络是用于评价 Actor 网络的动作输出的，其输入为智能体的状态和 Actor 所输出的动作，输出为其对该状态下的该动作的评价值 $Q(s,a)$。Critic 网络是根据环境给出的奖励值函数进行更新的，而 Actor 网络是根据 Critic 网络输出来更新网络权重的。此外，这两种网络均还分为主网络和目标网络。主网络为主要的更新对象，而目标网络主要是用于在网络更新过程中保持目标值相对稳定。目标网络的权重会根据主网络权重以一定的折扣率持续更新，如公式4.2所示。

$$\begin{aligned} \theta^{Q'} &\leftarrow \tau\theta^{Q} + (1-\tau)\theta^{Q'} \\ \theta^{\mu'} &\leftarrow \tau\theta^{\mu} + (1-\tau)\theta^{\mu'} \end{aligned} \tag{4.2}$$

其中 Q 代表主网络中的 Critic 网络，μ 代表主网络中的 Actor 网络；Q' 代表目标网络中的 Critic 网络，μ_t 代表目标网络中的 Actor 网络；τ 是更新的折扣率。该算法流程如算法3所示。

4.2.2 特征提取网络设计

由于在蜂群无人机松散编队中，每个无人机的邻居节点并不固定，因此引入图神经网络以实现无人机对邻居信息的感知和处理。

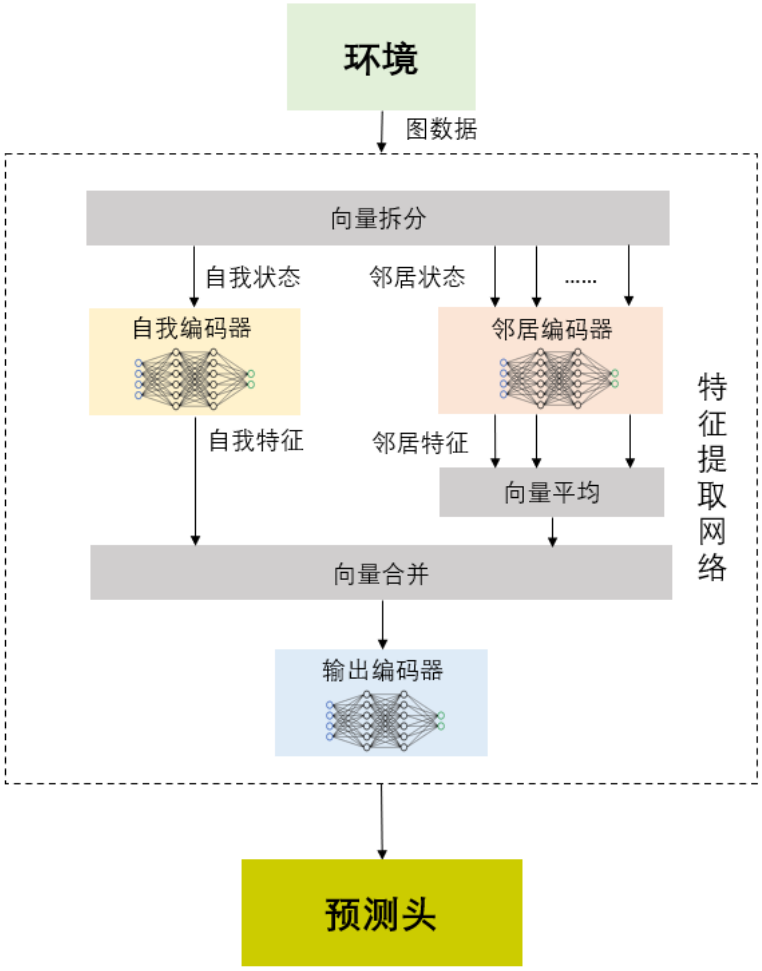


图 23 特征提取网络架构图

特征提取网络架构如图23所示，其基于一跳的图卷积网络进行了改进。由于图卷积网络在处理当前节点数据和邻居节点数据时均进行了统一的求和或平均处理，因此网络在结构上并未对当前节点和邻居节点加以直接区分，在同一层上所使用的网络参数是共享的。而在本章节的问题中，需要处理的自我节点数据除了包括自身与领导者的状态差，还包括初始时这一状态差值。这是为了减少邻居无人机的位置振动而设计，具体原因将在奖励函数设置部分介绍。而邻居节点不需要这一差值。因此，设计了两个不同

的编码器用于分别处理两种信息，具体网络设计如下。

首先，特征提取网络的输入是包含无人机节点状态和邻居无人机状态的图数据。当向量输入后，对于一般的图卷积网络而言，每个节点的数据均经过共享参数的第一层神经网络；而在本章中，将网络改进为两个编码器，即自我编码器（self-encoder）和邻居编码器（neighbor-encoder），数据向量首先进行拆分，分为自我状态和多个邻居状态。自我状态通过自我编码器形成自我特征，而多个邻居状态分别通过邻居编码器形成同样数量的邻居特征，这些状态通过的是同一个邻居编码器，结构和参数完全一致。随后，多个邻居特征通过向量平均的形式形成统一的固定维度的特征向量，并与自我特征进行向量合并，最后合并的向量通过输出编码器输出特征提取后的嵌入向量，嵌入向量将由预测头处理进行最终的输出。在本章中，Actor 网络的预测头的最终输出是两维的动作，而 Critic 网络的输出是对某状态下的动作输出的评价值。

4.2.3 算法设计

4.2.3.1 观测空间

在松散编队中，无人机的控制输入为水平方向的加速度。因此，神经网络不需要输入底层的无人机状态。在本章中，定义无人机 i 的状态为：

$$s_i = \begin{bmatrix} x_i \\ y_i \\ v_{x_i} \\ v_{y_i} \end{bmatrix} \tag{4.3}$$

网络输入为自我状态和领导者状态差值，以及这一差值的初始值。此外，在输入神经网络时，还同时包含了该节点邻居无人机的状态。邻居无人机状态被定义在当前节点的机体系下，这是因为对该无人机而言，邻居的相对位置和相对速度比绝对状态更加重要。因此节点 i 考虑的每个邻居状态 s_i^j 为：

$$s_i = \begin{bmatrix} x_j - x_i \\ y_j - y_i \\ v_{x_j} - v_{x_i} \\ v_{y_j} - v_{y_i} \end{bmatrix} \tag{4.4}$$

如果节点 i 的邻居数量为 N_i，则最终其观测输入为：$o_i = (s_i - s_0, s_i^{init} - s_0^{init}, s_i^{j_1}, \cdots, s_i^{j_{N_i}})$，

观测空间维数为 $4 \times (N_i + 2)$。也就是说，对于不同节点，或者同一节点的不同时刻，观测空间维数可以不同。

4.2.3.2 动作空间

神经网络的输出是无人机的水平加速度 a_x 和 a_y。因此动作空间维数固定为 2。考虑到无人机的能够响应的加速度和速度是有界的，在此定义无人机单方向加速度最大值为 a_{max}，单方向最大速度为 v_{max}。那么有：

$$a_i = \min(\max(a_i, -a_{\max}), a_{\max}) \tag{4.5}$$

$$v_i^{t+1} = \begin{cases} v_{\max} & \text{if } v_i^t + a_i^t * dt > v_{\max} \\ -v_{\max} & \text{if } v_i^t + a_i^t * dt < -v_{\max} \\ v_i^t + a_i^t * dt & \text{otherwise} \end{cases} \tag{4.6}$$

4.2.3.3 奖励函数

奖励函数是人为制定的强化学习奖励机制，用于指示无人机正确的行为。在本章中，强化学习协同控制器的目的有以下几点：

1. 集群。控制器应该使蜂群无人机聚集在领导者周围的一个区域范围内，每个无人机都不能离开集群。

2. 避碰。无人机与无人机之间应该保持安全距离，不能发生碰撞以影响集群安全。

3. 能量消耗最低。无人机应该尽量保持飞行的平稳，不应该频繁加速或减速造成过多能量消耗。

考虑到上述目的，本小节按无人机状态和无人机与邻居之间的相对位置来设计奖励函数。图24介绍了无人机周围三个重要的范围。红色区域半径为安全距离 R_s，当邻居无人机处于该范围时，容易发生碰撞；绿色区域半径为期望距离 R_e，与处于这一区域的邻居无人机发生碰撞的几率较低，且能够保持较好的通信；黄色区域半径为通信距离 R_c，处于这一区域的邻居无人机尽管仍能够正常通信，但一旦超出通信距离就会失去连接。

因此设计奖励函数如公式4.7所示。其中 d_{min} 是距离无人机最近的邻居节点距离，δx_i 和 δx_i^{init} 是节点 i 与领导者无人机的位置差以及位置查到初始值。r_i^f 是用于维持集

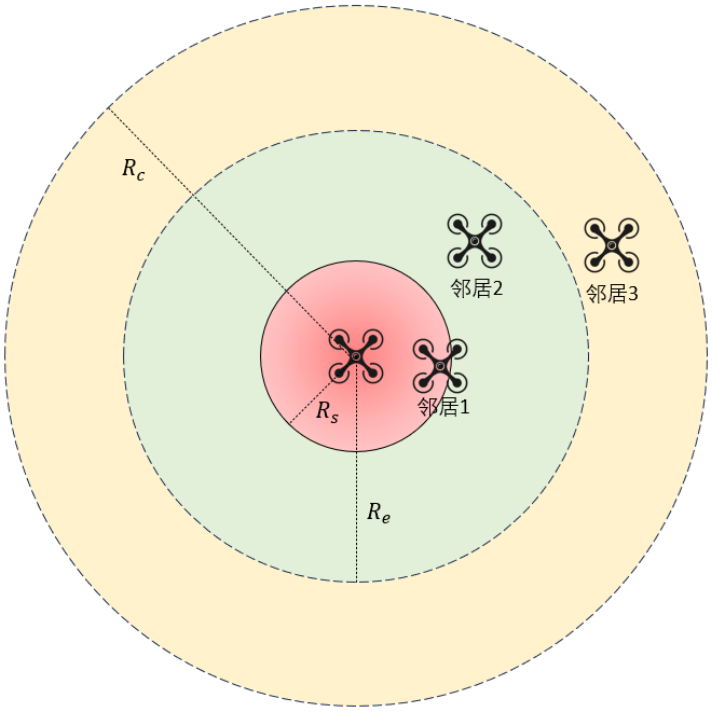


图 24 无人机范围示意图

群的奖励函数，r_i^s 是用于减少能量损耗的奖励函数，两者之和为节点 i 获得的奖励。r_1，r_2，r_3，k_1，k_2 均为常数。

$$r_i^f = \begin{cases} r_1 & \text{if } d_{min} < R_s \\ r_2 & \text{if } d_{min} > R_c \\ -k_1 * d_{min}^2 & \text{if } R_e < d_{min} \leq R_c \\ r_3 & \text{otherwise} \end{cases} \tag{4.7}$$

$$r_i^s = -k_2 \times \|\delta x_i - \delta x_i^{init}\|^2$$

$$r_i = r_i^f + r_i^s.$$

4.3 算法训练

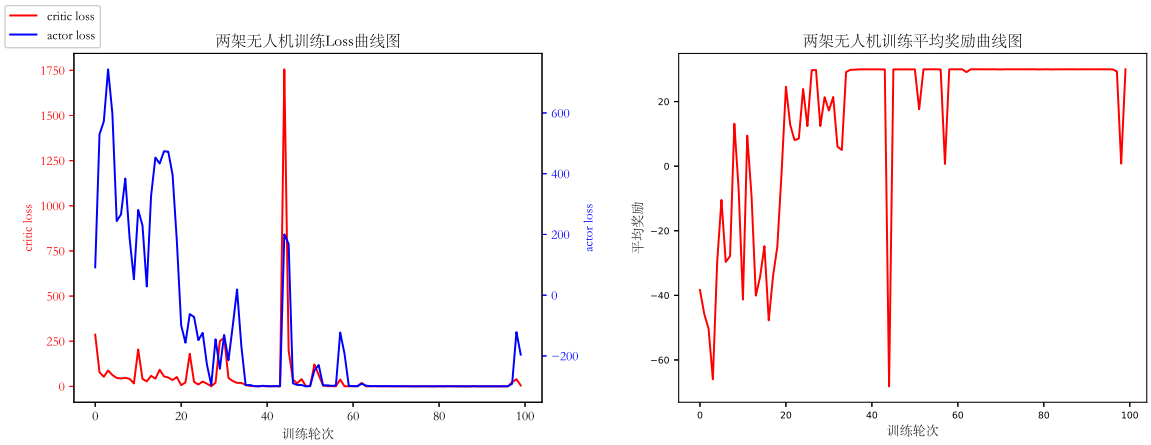
为了方便训练，本章采用课程学习的方法进行多次训练。在 1993 年，Jeffery Elman 提出课程学习用于训练神经网络。课程学习的核心思想是像人类学习一样，从简单问题开始学起并逐步扩展到复杂问题。在神经网络的训练中，课程学习表现为先从有限的、简单的数据集开始训练，并逐渐增加训练样本的难度。实验表明，课程学习能够加速模型收敛，还有可能提升模型的最终性能。而对强化学习来说，应用课程学习需要制定一系列从简单到复杂的任务，这些任务应该使用相同的强化学习算法和神经网络结构，以便从简单问题过度到复杂问题时，训练的成果能够直接迁移。

在本章中，蜂群无人机集群中无人机的数量是问题复杂与否的关键。对于少量无人机，集群飞行和避碰的任务更加简单。同时，在不同规模的蜂群无人机中，上述设计的算法和网络结构是通用的。因此，采用不同规模的无人机集群是一组可行的课程设计。在训练过程中，各参数设置如表2所示。

表 2 强化学习训练参数设置

参数	数值	描述
M	100	训练轮次
d_s^a	$4 \times (N_i + 2)$	Actor 网络输入层维数
d_o^a	2	Actor 网络输出层维数
d_e^a	16	Actor 网络嵌入层维数
$d_{h_1}^a$	32	Actor 网络第一隐藏层维数
$d_{h_2}^a$	64	Actor 网络第二隐藏层维数
$d_{h_3}^a$	16	Actor 网络第三隐藏层维数
d_s^c	$4 \times (N_i + 2) + 2$	Critic 网络输入层维数
d_o^c	1	Critic 网络输出层维数
d_e^c	16	Critic 网络输入层维数
$d_{h_1}^c$	32	Critic 网络第一隐藏层维数
$d_{h_2}^c$	64	Critic 网络第二隐藏层维数
$d_{h_3}^c$	16	Critic 网络第三隐藏层维数
r_1	-100	碰撞惩罚值
r_2	-50	失联惩罚值
r_3	30	期望范围奖励值
k_1	0.01	松散惩罚系数
k_2	0.01	能量惩罚系数
R_s	0.5m	安全距离
R_c	50m	通信距离
R_e	10m	期望距离
l_a	1^{-4}	Actor 网络学习率
l_c	1^{-3}	Critic 网络学习率
τ	1^{-2}	Target 网络软更新率
N_R	100	经验池大小
N_b	5	批大小
n_c	4	最大通信数量

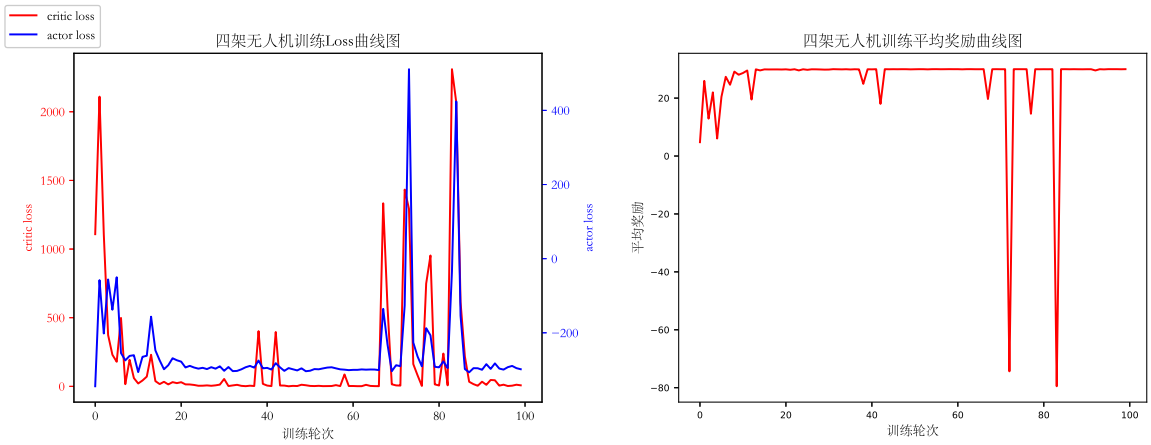
训练算法为基于 DDPG 架构和图神经网络的多智能体课程强化学习算法，算法将训练过程分为多个阶段，每个阶段的无人机数量不同，分别为 $\{N_1, N_2, \cdots, N_{n_c}\}$。经过试验，本章设置的课程学习分为了三个阶段，无人机数量分别为 2，4 和 32 架。领导者无人机初始化随机的探索任务，并根据 PID 控制器进行轨迹跟踪任务；跟随者无人机根据网络输出采取动作，并获得新的状态和奖励用于网络更新。值得注意的是，尽管算法是多智能体的 DDPG 算法，但与 Multi-Agent Deep Deterministic Policy Gradient（MADDPG）算法有所不同。MADDPG 算法的 Critic 网络的输入是所有智能体观测的拼接，即中心化训练、分布式执行。这样训练的网络对于无人机数量比较敏感，不容易推广到任意数量。而本章的算法是分布式的，对于节点数量变化仍能适用。



(a) 两架无人机训练 Loss 曲线图　　(b) 两架无人机训练平均奖励曲线图

图 25 两架无人机训练情况

两架无人机的训练 Loss 曲线如图25(a)所示。在两架无人机的训练过程中，一架无人机为领导者，只有一架跟随者无人机。唯一的跟随者需要学习的任务只有跟随领导者这一较为简单的任务，无需考虑其他邻居节点。初始时跟随者无人机不能较好地跟随，而处于自由探索阶段，整体上与领导者的距离较远。此时奖励机制会给予较大的负奖励，如图25(b)所示。在经过一段时间的训练之后，Actor 网络和 Critic 网络损失均下降并在大约 60 轮次后收敛，而无人机获得的平均奖励也升高并维持在 20 以上。值得注意的是，由于两架无人机发生碰撞的概率较低，此时强化学习控制器对碰撞情况的学习仍有不足，只是初步学习到了对领导者的跟随。Loss 函数在收敛后的突然增大和平均奖励突然减小代表着该次探索中发生了一定次数的碰撞。从图中可以看出，两架无人机在训练过程中碰撞次数较少，因此需要在更大规模的集群中进行进一步学习。

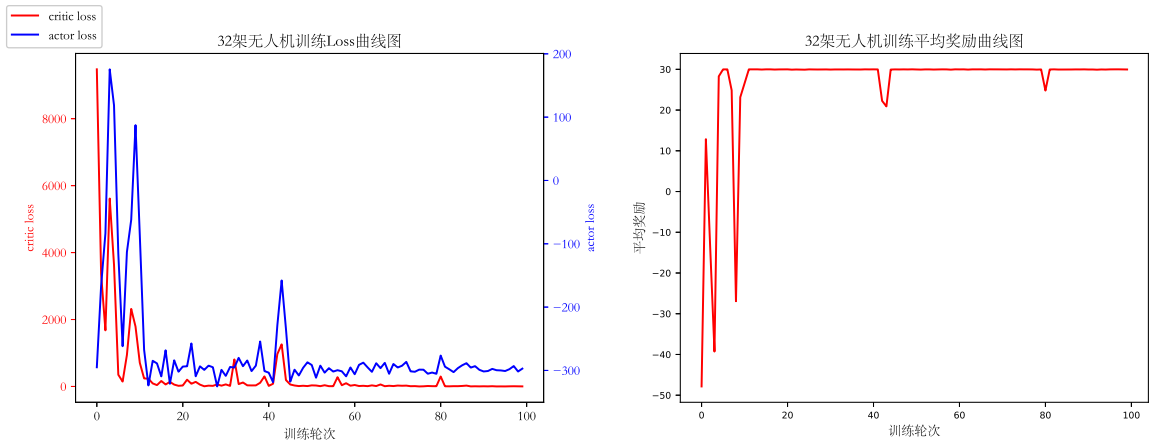


(a) 四架无人机训练 Loss 曲线图　　(b) 四架无人机训练平均奖励曲线图

图 26 四架无人机训练情况

在两架无人机情况进行训练后，保留网络权重参数进行四架无人机情况的训练。对

于不同的集群规模，基于图神经网络的特征提取网络结构完全一致，因此可以直接导入权重。四架无人机训练的 Loss 曲线图如图26(a)所示。从图中可以看出，在经过大约 20 训练轮次之后，神经网络达到了初步的收敛，能够一定程度上适应四架无人机集群的任务。但是，在训练轮次到达 80 左右时，网络的平均损失突然增大，这是因为每个训练轮次领导者的探索路线是不同的。在 80 训练轮次左右，领导者的任务路线中存在突然加速或较大转弯的情况，此时网络没有较好地实现避碰，编队内部出现了大量的碰撞情况。图26(b)也可以看出，在大约 20 次轮次后，奖励趋于稳定；而在 80 轮次附近，奖励出现多次降低，平均奖励达到-80 左右，能够说明集群内部发生了较多碰撞。最终，网络损失和奖励均趋于稳定。



(a) 32 架无人机训练 Loss 曲线图　　(b) 32 架无人机训练平均奖励曲线图

图 27 32 架无人机训练情况

最后，将 4 架无人机训练的网络应用到 32 架无人机进行更进一步的训练。在 32 架无人机的情况中，无人机之间的碰撞会更容易发生。值得注意的是，在 4 架无人机集群的训练过程中，每架无人机尽管对避碰有了一定程度的学习，但在其预期位置附近的振荡仍然存在。这是因为 4 架无人机实际上还比较松散，一定程度的振荡不容易造成碰撞。而在 32 架无人机的集群中，无人机在位置附近的振荡应该被避免，否则容易发生机间碰撞。

如图27(a)和图27(b)所示是 32 架无人机的训练过程 Loss 曲线图和平均奖励曲线图。从图中可以看出，在 32 架无人机的训练初期，集群内部仍然发生了较多的碰撞，平均奖励较低而 Loss 较大。在大约 20 轮次的训练之后，神经网络基本达到收敛，奖励也趋于稳定。最终的评价奖励要高于 4 架无人机训练最终阶段的平均奖励而更加接近 30，这是因为无人机在经过训练之后减少了振荡现象，为了减少能量损耗的惩罚 r_i^s 更加接

近于 0。如图28所示是集群跟随预期路径的轨迹图。从图中可以看出，经过训练的蜂群无人机能够正常跟随期望轨迹，且集群内部无碰撞。

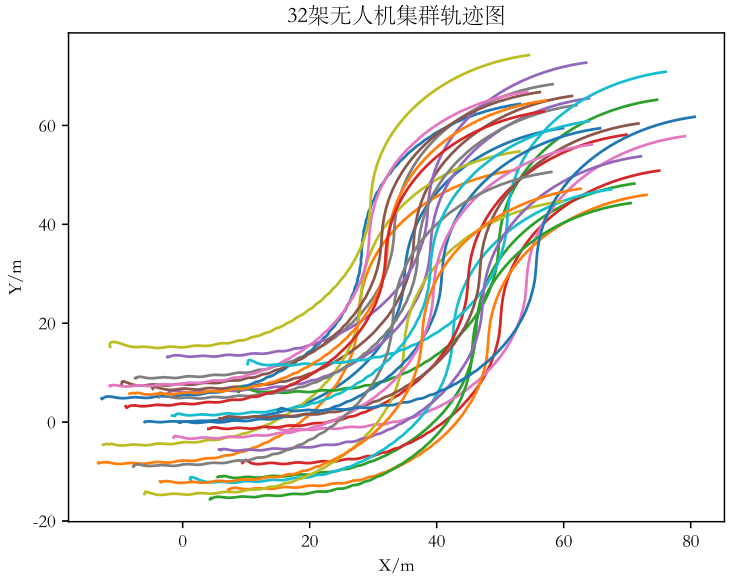


图 28　32 架无人机集群轨迹图

4.4　本章小结

本章针对安全约束下的蜂群无人机分布式协同控制方法进行了研究，首先提出了基于图神经网络和 DDPG 架构的强化学习协同控制算法，并利用课程学习机制设计了训练过程。其次，给出了训练中的强化学习参数，并按照给出的算法和参数进行了训练，并通过分析训练损失和平均奖励证明了训练方法的有效性。最终，针对给定的期望轨迹进行了测试，测试中 32 架蜂群无人机能够在保证集群内部安全的情况下正常跟随预期轨迹，证明了算法的可行性。