多智能体分布式围捕控制算法研究

摘要

近年来，多智能体系统的分布式编队控制问题已成为国内外各研究者的关注点，这是由于其在社会、工业和国防等领域有着广泛的应用前景本项目使用 MAPPO 的强化学习方法编写集群算法，并通过软硬件结合的方式，首先在 Gazebo 环境搭建仿真平台，验证集群算法的可靠性和实用性，然后在此基础上搭建了基于 Omini 全向三轮小车的硬件实验平台，进一步在现实应用场景中验证算法的可行性。在集群围捕算法方面，本项目使用了三种强化学习框架：MAPPO、MADDPG 和 MATD3。其中，MAPPO 是 PPO 算法在多智能体任务中的变种，具有高效的学习效率。MADDPG 算法，作为 DDPG 算法的扩展，适用于合作、竞争和混合合作竞争的多智能体环境。MATD3 算法对 DDPG 算法进行了优化，引入了双重网络和延迟更新机制，取得了在连续控制任务上的良好表现。硬件实验平台方面，我们使用基于 ROS2 的 Omini 三轮全向 wheeltec 小车，尝试了 marvelmind 声波定位系统、amcl 定位算法等定位方案，利用 rviz2 可视化了小车集群围捕。本项目的创新点在于：一、采用分布式的编队控制系统，有效的解决集中式控制系统编队控制计算复杂、鲁棒性较差的缺点，可以组建成规模更大、拓扑结构更加复杂的多智能体编队。二、借助复杂网络的理论，通过借鉴成熟的图论理论来研究编队信息流、编队构型、控制律设计。

关键词： 多智能体强化学习，围捕控制，Omini 小车，ROS2

ABSTRACT

In recent years, the distributed formation control problem of multi-agent systems has become a focal point for researchers both domestically and internationally. This is due to its wide-ranging applications in areas such as society, industry, and defense. The formation control problem refers to the control technology where multiple agents, in the process of reaching their destination, maintain a specific formation while adapting to environmental constraints. In this project, a cluster algorithm is developed using the reinforcement learning approach of MAPPO. Through a combination of hardware and software, the algorithm's reliability and practicality are first verified in a simulated environment built on the Gazebo platform. Subsequently, a hardware experimental platform is established to further validate the algorithm's feasibility in real-world scenarios. In the context of cluster capture algorithms, three reinforcement learning frameworks are employed. Firstly, the Actor-Critic framework is utilized, where the algorithm integrates the considerations of both Actor and Critic components to achieve interactive decision-making and value evaluation based on environmental feedback. Secondly, the MAPPO algorithm is applied as a variant of the PPO algorithm for multi-agent tasks, demonstrating

efficient learning capabilities. Thirdly, the MADDPG algorithm, an extension of the DDPG algorithm, is employed for multi-agent environments involving cooperation, competition, and mixed cooperation-competition scenarios. Finally, the MATD3 algorithm is introduced to optimize the DDPG algorithm, incorporating dual networks and a delayed update mechanism, resulting in excellent performance in continuous control tasks. In terms of the hardware experimental platform, the project utilizes the ROS2-based Omnidirectional three-wheel Wheeltec robot. Various localization solutions, including the marvelmind ultrasonic positioning system and the amcl localization algorithm, are explored. The visualization of the robot cluster capture is achieved using rviz2.The project's innovations lie in two main aspects. Firstly, it adopts a distributed formation control system, effectively addressing the drawbacks of centralized control systems, such as computational complexity and poor robustness. This approach allows for the formation of larger-scale and more complex topological multi-agent formations. Secondly, leveraging the theoretical framework of complex networks, the project draws on mature graph theory to study formation information flow, formation configuration, and control law design.

KEY WORDS： Keywords: Multi-agen Reinforcement Learning, Encirclement Control, Omnidirectional Robot (Omini Car), ROS2

1 绪论

近年来，多智能体系统的分布式编队控制问题已成为国内外各研究者的关注点，这是由于其在社会、工业和国防等领域有着广泛的应用前景。所谓编队控制问题，就是指多个智能体在到达目的地的过程中，保持某种特定队形，同时适应环境约束的控制技术。多智能体系统的分布式编队控制问题主要包含几个方面：编队控制，即如何控制一个编队的集体运动；队形变换，即如何切换编队的队形；抗干扰性：即如何增强编队控制的鲁棒性，实现稳定控制。通过对多智能体编队控制算法的研究与实现，实现对多智能体的编队控制，并表现出良好的抗干扰性能，使得控制更稳定。进而使得对诸如无人机群，机器人群等实际中的多智能体的集体性控制成为可能，为实际多智能体的编队控制建立基础。同时搭建实验平台，在软硬件层面实现集群控制实验。

强化学习是一种通过与环境的交互，不断优化智能体决策的人工智能方法。智能体在与环境的交互过程中会获得不同的奖励，该奖励会引导智能体提高自己的决策方案 π ，以获得更大的奖励。如图 1-1 所示，智能体在 t 时刻状态为 S_t ，根据自己的决策方案执行动作 a_t ，执行动作后智能体进入下一个状态 S_{t+1} ，同时获得奖励值 r_{t+1} ，实现与环境的交互过程。智能体根据交互获得的状态数据、动作数据和奖励数据进行多次迭代优化，直到得到最优的策略 π^* 。

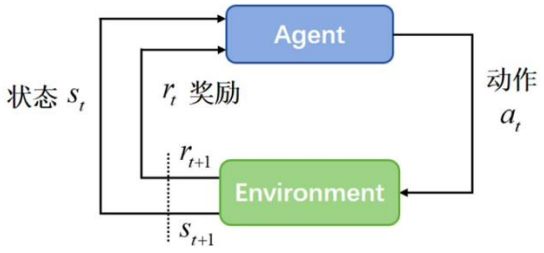


图 1-1 强化学习基本框架

在多智能体强化学习领域中，由于每个智能体均在环境中不断学习改善自身策略，从每个智能体的角度看，不清楚其他智能体的动作策略，而这些智能体的动作会对环境带来变化，导致环境变得不稳定，因此多智能体强化学习的难点在于不仅需要了解每个智能体与环境的交互过程，还需要考虑到其他智能体对环境的影响，使得问题变得较为复杂。

在多智能体领域中，每个智能体与环境的交互过程被定义为马尔可夫博弈(Markov Game)，也被称为随机博弈(Stochastic Game)。假设环境中存在 N 个智能体，马尔可夫博弈可用多元组 $\langle S,A,R,P,\gamma \rangle$ 表示。其中，S 为环境状态空间，用 $s \in S$ 表示环境状态；$A = A_1 \times A_2 \times ... \times A_N$ 为智能体的联合动作空间，用 $a = a_1 \times a_2 \times ... \times a_N$ 表示所有智能体联合动作；$R = \{R_1, R_2, ..., R_N\}$ 为智能体奖励函数集合；状态转移概率 $P: S \times A \times S$ 为指定智能体 i 在联合状态 s 及联合动作 a 时，环境状态改变到下一个环境状态 s' 的概率分布；γ 为累积奖励衰减因子。

智能体的策略 $\pi: S \to A$ 是从状态到动作的映射，在任意时刻 t，智能体 i 根据当前状态 $s_t \in S_i$ 和策略 $\pi(a|s)$ 选择动作 $a_t \in A_i$，并根据状态转移概率 P 达到下一时刻状态 $s_{t+1} \in S_i$，同时获得奖励 $r_{i,t} \in R_i$。引入累计折扣奖励如式 1-1 所示，多智能体强化学习算法目的是寻找最优策略 π^* 使每个智能体的累计折扣奖励最大化。

$$R_{i,t} = r_{i,t+1} + \gamma r_{i,t+2} + \gamma^2 r_{i,t+3} + ... = \sum_{k=0}^{\infty} \gamma^k r_{i,t+k+1} \tag{1-1}$$

其中 $\gamma \in [0,1]$ 为折扣因子，如果 γ 接近 0，则说明智能体更关心当下奖励；如果 γ 接近 1，则说明智能体更看重未来奖励，会分配更多权重给未来奖励。

另外，价值可以用于衡量当前状态的好坏，用 $V(s)$ 表示。价值是关于累计折扣奖励的期望函数，当前状态的价值不仅需要考虑下一时刻的奖励，而需要综合考虑后续每一个时刻的奖励。智能体 i 在策略 π 下状态 s 的价值可表示为式 1-2 所示。

$$V_{\pi,i}(s) = \mathbb{E}_{\pi,i}\left(r_{i,t+1} + \gamma r_{i,t+2} + \gamma^2 r_{i,t+3} + ... | S_t = s\right) = \mathbb{E}_{\pi,i}\left(R_{i,t} | S_t = s\right) \tag{1-2}$$

全向小车是一种具有全向运动能力的机器人底盘，Omnidirectional 三轮全向小车是其中的一种设计。这种小车通常由三个轮子组成，每个轮子都可以独立运动，使得小车能够在任意方向上实现平移和旋转。其结构呈平衡的三角形，轮子分布在底盘的 120 度角上。每个轮子具有特殊构造，允许在垂直于轮子表面的方向上滑动。通过精确控制三个轮子的运动速度和方向，小车可以实现全向平移和旋转，无需改变朝向。控制涉及对每个轮子的速度和方向进行精确调节，通常使用微控制器或嵌入式计算机结合传感器。

图 1-2 Omini 三轮全向小车

在本研究中，我们构建了硬件实验平台，以进一步验证我们提出的算法在实际场景中的可行性。我们选用了基于 ROS2 的 Omnidirectional 三轮全向 wheeltec 小车作为我们的硬件平台。这款小车的设计使其具备出色的机动性，为我们的实验提供了良好的机器人底盘。在定位方面，我们探索了多种解决方案，其中包括 marvelmind 声波定位系统以及 amcl 定位算法。通过这些方案，我们旨在实现对小车在实际环境中的精准定位，为后续的实验提供可靠的基础。同时，我们借助 rviz2 进行可视化，实现了小车集群围捕的实时展示，为我们的实验过程提供了直观的观察手段。这一硬件实验平台的建立将为我们深入研究多智能体编队控制算法提供有力支持，使得我们能够更加全面地评估算法在实际应用场景中的性能表现。

2 研究内容和方法

我们的研究任务主要分为 Gazebo 仿真平台搭建、MAPPO 强化学习算法研究、Omini 三轮小车部署三部分，核心是把 MAPPO 算法框架部署到 Omini 小车上，并进行实验验证强化学习算法。

我们的研究流程如图 2-1 所示。

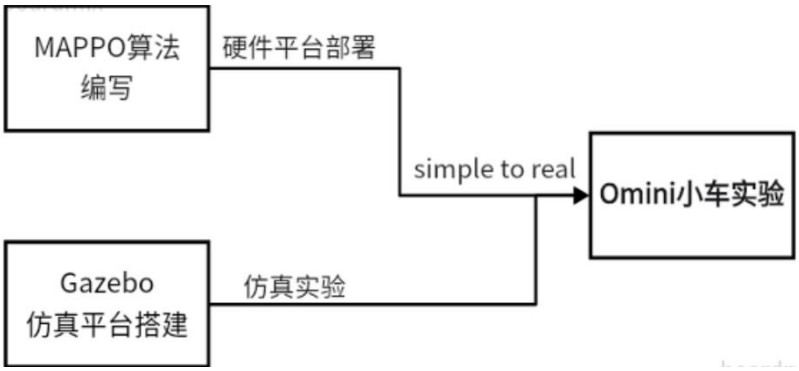


图 2-1 研究流程图

2.1 强化学习框架与原理

2.1.1 Actor-Critic 框架

Actor-Critic 算法，是一种基于模型的强化学习算法，相较于其他算法，它有着更快的收敛速度和更好的训练稳定性，在强化学习领域有着广泛的运用，其框架原理如图 2-2 所示。它综合考虑了 Actor 和 Critic 两个组件，两者的交互过程由环境决定，Actor 负责给出动作（即决策），而 Critic 则负责评估 Actor 给出的动作的价值，并根据价值反馈的信息进行调节，使 Actor 可以得到最大的奖励。与其他的模型学习方法相比，Actor-Critic 算法可以直接从环境中获取反馈信息，不需要额外的监督信号。

在 Actor-Critic 框架中，策略函数和价值函数分别用 Actor 和 Critic 两个单独的网络来近似，策略函数作为 Actor 来做动作选择，价值函数作为 Critic 来对策略函数进行评估，并根据 Critic 的输出来更新价值网络和策略网络。

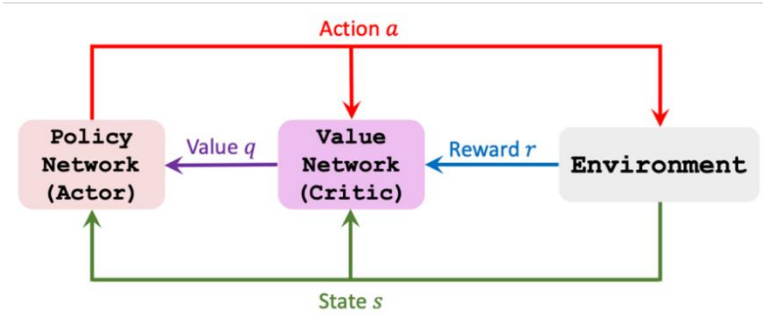


图 2-2 Actor-Critic 算法框架原理

2.1.2 MAPPO 算法

MAPPO （Multi-agent Proximal Policy Optimization）是近端策略优化（Proximal Policy Optimization, PPO）应用于多智能体任务的变种，该算法采用 Actor-Critic 架构，在有限算力的条件下具有较高的学习效率，仅需进行少量超参数搜索工作就可达到主流 MARL 算法的性能，在多种合作任务领域表现出了较高的水准。

PPO 算法是基于 PG 算法和 TRPO 算法的衍生，具有它们的优势，在能够处理连续动作空间的同时具有更好的收敛性。相比于 PG 算法对每个样本数据进行一次梯度更新，PPO 提出了新的目标函数（如式 2-1 所示），能够进行小批量更新。

$$J_{PPO}(\theta)=\sum_{t=1}^{T}\frac{\pi_{\theta}(a_t\mid s_t)}{\pi_{\theta old}(a_t\mid s_t)}\hat{A}_t-\lambda \mathrm{KL}\left[\pi_{\theta old}\mid\pi_{\theta}\right] \tag{2-1}$$

其中，KL 散度用于衡量两个分布之间的不同程度，若新旧策略差异过大，则 KL 散度较大，因此需要通过约束 KL 散度来使得策略更新不会过于发散。

另外，当 PPO 在进行小批量更新时，需要控制更新的幅度，定义重要性采样权重来衡量策略更新的幅度，如式 2-2 所示。

$$r_t(\theta)=\frac{\pi_{\theta}(a_t\mid s_t)}{\pi_{\theta old}(a_t\mid s_t)} \tag{2-2}$$

$r_t(\theta)$ 表示新旧策略变化的比例，如果 $r_t(\theta)$ 越接近 1，说明策略更新幅度越小。PPO 通过引入剪裁操作，严格限制策略更新的比例。如图 2-3 所示，剪裁系数为 ε，当 $\hat{A}_t > 0$ 时，若 $r_t(\theta) > 1+\varepsilon$，则 $L^{CLIP}(\theta) = (1+\varepsilon)\hat{A}_t$，否则 $L^{CLIP}(\theta) = r_t(\theta)\hat{A}_t$；当 $\hat{A}_t < 0$ 时，若 $r_t(\theta) < 1-\varepsilon$，则 $L^{CLIP}(\theta) = r_t(\theta)\hat{A}_t$，否则 $L^{CLIP}(\theta) = (1-\varepsilon)\hat{A}_t$。

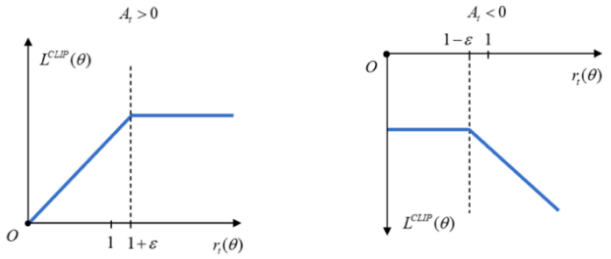


图 2-3 PPO 剪裁原理示意图

MAPPO 算法是 PPO 算法多智能体形式的扩展，具体方式是训练两个独立的网络，分别为参数为 θ 的策略网络 π_θ 和参数为 ϕ 的价值网络 V_ϕ，它们对所有同构的智能体共享网络权重。策略网络 π_θ 将智能体 i 的局部观测映射到连续动作空间的分类分布；价值网络 V_ϕ 将全局状态映射到状态价值的估计。

策略网络的损失函数 $L^{CLIP}(\theta)$ 使用了 PPO 中的剪裁思想，以限制策略更新的幅度；价值网络的损失函数 $L^{CLIP}(\phi)$ 则是进行了批归一化，有利于提高训练的是收敛性。损失函数的表达式如式 2-3 所示。

$$
\begin{aligned}
L^{CLIP}(\theta) &= \frac{1}{BN}\sum_{i=1}^{B}\sum_{k=1}^{N} min\left(c_{\theta,i}^{(k)}A_i^{(k)}, clip\left(c_{\theta,i}^{(k)}, 1-\varepsilon, 1+\varepsilon\right)A_i^{(k)}\right) \\
L^{CLIP}(\phi) &= \frac{1}{BN}\sum_{i=1}^{B}\sum_{k=1}^{N} max\Bigg[\left(V_\phi\left(s_i^{(k)}\right)-\hat{R}_i\right)^2, \\
&\qquad \left(clip\left(V_\phi\left(s_i^{(k)}\right), V_{\phi old}\left(s_i^{(k)}\right)-\varepsilon, V_{\phi old}\left(s_i^{(k)}\right)+\varepsilon\right)-\hat{R}_i\right)^2\Bigg]
\end{aligned}
\tag{2-3}
$$

其中，$c_{\theta,i}^{(k)}$ 是重要性采样权重，$A_i^{(k)}$ 是优势函数，ε 是剪裁系数。$V_\phi\left(s_i^{(k)}\right)$ 和 $\hat{R}_i$ 分别为归一化过后的价值和归一化过后的奖励，B 是批处理数，N 是智能体数量。

2.1.3 MADDPG 算法

MADDPG（Multi-Agent Deep-Deterministic Policy Gradient）算法继承于单智能体的深度确定性策略梯度 DDPG（Deep-Deterministic Policy Gradient）算法，是 Actor-Critic 框架下的一种在线式深度强化学习算法，采用“集中式训练、分布式执行”的方式，能够适用于环境不稳定的情况，可应用在合作、竞争及混合合作竞争的多智能体环境。

Actor-Critic 框架分两个网络。Actor 网络基于智能体的当前状态输出策略，为智能体选择最优动作，并通过 PG 算法不断优化策略网络。Critic 网络对智能体每个状态-动作组的理论价值即 Q 值进行评估，并根据智能体的实际收益，通过 TD 调整网络参数，提高评估的准确性。

Actor-Critic 结构通过使用两个深度神经网络分别实现了输出动作空间的连续化和输入状态空间的连续化，完美地将低维、离散空间上的强化学习拓展到高维、连续空间。DDPG 在 Actor-Critic 的框架上，借用了 DQN 算法的在线-目标双网络结构，即包括四个网络，分别是 Actor 网络 μ_θ，Critic 网络 Q_w，Target Actor 网络 $\mu'_{\theta'}$ 和 Target Critic 网络 $Q'_{w'}$，实现了由在线训练向离线训练的拓展，省却了如重要性采样等一系列复杂操作，同时也提高了数据的利用率。

在更新网络的时候，Actor 网络产生动作，以最小限度减少进程损失进行训练；Critic 网络进行 Q 值估计，以非政策方式最小化均方贝尔曼误差（MSBE）。两者的损失函数如式 2-4 所示，其中 y_j 为智能体 j 的目标 Q 值，γ 为下一奖励值的折扣系数。

$$\begin{aligned} L(\theta) &= -\frac{1}{m}\sum_{j=1}^{m} Q_w\left(s,\mu_\theta(s)\right) \\ L(w) &= \frac{1}{m}\sum_{j=1}^{m}\left[y_j - Q_w(s,a)\right]^2,\ \text{其中} y_j = r_j + \gamma Q'_{w'}\left(s',\mu'_{\theta'}(s')\right) \end{aligned} \tag{2-4}$$

另外，对于目标网络的更新，DDPG 算法中采用软更新方式，也可以称为指数平均移动（Exponential Moving Average, EMA）。即引入一个学习率 τ，将旧的目标网络参数和新的对应网络参数做加权平均，然后赋值给目标网络，如式 2-5 所示。由于目标网络中的参数是通过软更新的方式来缓慢更新的，因此它的输出会更加稳定，利用目标网络来计算目标值自然也会更加稳定，从而进一步保证 Critic 网络的学习过程更加平稳。

$$\begin{aligned} \theta'^{\mu'} &= \tau\theta^{\mu} + (1-\tau)\theta'^{\mu'} \\ w'^{Q'} &= \tau w^{Q} + (1-\tau)w'^{Q'} \end{aligned} \tag{2-5}$$

2.1.4 MATD3 算法

TD3 （Twin Delayed Deep Deterministic policy gradient）算法是对 DDPG 算法的优化，为使其能适用于多智能体场景，从而发展出 MATD3。TD3 算法也是 Actor-Critic 框架下的一种确定性深度强化学习算法，它结合了深度确定性策略梯度（DDPG）算法和双重 Q 学习（Double Q-learning）算法，在许多连续控制任务上都取得了不错的表现。

TD3 算法在 DDPG 算法的基础上，引入了双重网络，来解决网络过估计的问题，因此 TD3 算法中包括六个网络，分别是 Actor 网络 μ_θ，Critic1 网络 Q_{w_1}，Critic2 网络 Q_{w_2}，Target Actor 网络 $\mu'_{\theta'}$，Target Critic1 网络 $Q'_{w_1'}$，Target Critic2 网络 $Q'_{w_2'}$，在计算智能体的目标 Q 值时，会利用两套 Critic 网络中的较小值来估计下一个状态动作的价值，如式 2-6 所示。

$$y_j = r_j + \gamma min_{i=1,2} Q'_{w_i'}\left(s', \mu'_{\theta'}(s')\right) \tag{2-6}$$

另外，TD3 算法采用了目标策略平滑正则化，具体来说就是利用目标动作周围的区域来计算目标值，从而有利于平滑估计值。在计算目标值时，可以通过向目标动作中添加少量服从正态分布的随机噪声的方法，来近似动作的期望。因此，目标值的计算式可修改为式 2-7 示。

$$\begin{aligned} y_j &= r_j + \gamma min_{i=1,2} Q'_{w_i'}\left(s', \mu'_{\theta'}(s') + \varepsilon\right) \\ \varepsilon &\sim clip\left(N(0,\sigma), -c, c\right) \end{aligned} \tag{2-7}$$

TD3 算法还引入了延迟更新机制，指的是 Actor 网络的延迟更新，即 Critic 网络更新多次之后再对 Actor 网络进行更新。由于 Actor 网络是通过最大化累积期望回报来更新的，它需要利用 Critic 网络来进行评估，如果 Critic 网络非常不稳定，那么 Actor 网络自然也会出现震荡。因此，可以让 Critic 网络的更新频率高于 Actor 网络，即等待 Critic 网络更加稳定之后再来帮助 Actor 网络更新。

2.2 模型与场景建立

在本研究中，智能体被定义为无限二维平面上以一定策略进行移动的物体。围捕智能体是以围捕逃逸智能体为任务的智能体，其目标是将逃逸智能体包围，形成包围圈。逃逸智能体或目标是以逃离围捕智能体的追逐为其固定策略的智能体，其目标是离所有围捕智能体尽量远。本研究的模型建立分为智能体运动学模型建立和场景模型建立两个部分。

2.2.1 智能体运动学模型建立

智能体运动学模型的建立又分为围捕智能体运动学模型和逃逸智能体模型的建立。围捕智能体的运动学模型建立如式 2-8 图 2-4 所示，其控制量包含线速度和角速度，线速度可以分解为 x 方向分速度和 y 方向分速度。逃逸智能体的运动学模型建立如图 2-5 所示，使用人工势场法确定其速度方向，以期望保持和每个围捕智能体的距离不会过近。设逃逸目标为 M，围捕智能体的集合为 I，智能体和逃逸目标的位置分别为 $\overrightarrow{p_i}$ 和 $\overrightarrow{p_M}$，$d_{i,M}$ 表示二者的距离，合力矢量方向如式 2-9 所示。

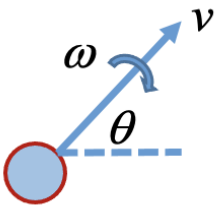


图 2-4 围捕智能体的运动学模型

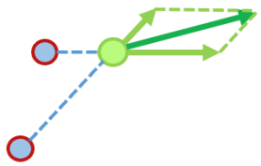


图 2-5 逃逸智能体的运动学模型

$$\begin{cases} \dot{x} = v\cos\theta \\ \dot{y} = v\sin\theta \\ \dot{\theta} = \omega \end{cases} \tag{2-8}$$

$$\vec{f}=\sum_{i\in I}\left(\frac{\overrightarrow{p_M}-\overrightarrow{p_i}}{{d_{i,M}}^2}\right) \tag{2-9}$$

2.2.2 围捕场景模型建立

围捕场景的建立包括智能体数量、智能体碰撞半径和智能体之间通信状况。逃逸智能体数量设为 1，围捕智能体数量可以任意设置，但不宜超过 5 个，这是由于多智能体算法在智能体数量增多时会产生不稳定或不收敛的现象。智能体碰撞半径设置为一个固定值，将智能体形状视为平面实心圆，两个智能体在平面中对应的圆形不能重叠，否则视为智能体的碰撞，为了保持一定安全距离，我们要求智能体圆心之间的距离不能小于一定值，这个值超过了两倍的智能体碰撞半径，以此保持任意智能体之间不会产生相撞可能。围捕智能体之间可以通信，获得其他智能体的观测信息。

2.3 围捕算法设计

2.3.1 动作空间设计

如图 2-6 所示，算法中围捕智能体的动作空间输出两个控制分量，径向控制分量 a_r 和切向控制分量 a_τ，两者共同表征在给定状态下，算法计算得出的智能体下一步应该采取的行动策略。注意算法给出的策略速度仅对围捕智能体生效，逃逸智能体仍然遵循其固定移动策略。

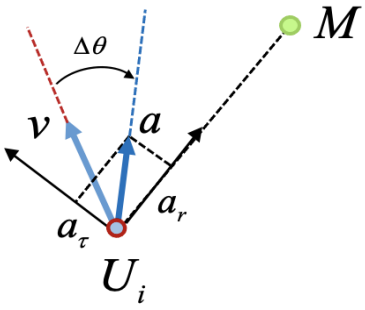


图 2-6 动作空间设计示意图

2.3.2 状态空间设计

算法中围捕智能体的状态空间能够表征智能体的运动策略、围捕的整体队形效果和智能体的围捕任务完成情况。如图 2-7 和式 2-10 所示，状态空间包括三个部分：围捕智能体 i 对于逃逸智能体 M 的特征 o^i_{loc}，所有围捕智能体所形成的群体对于逃逸智能体的状态特征 o^i_{ext}，以及围捕智能体对其他围捕智能体的特征 $o^{i,j}$。其中 o^i_{loc} 包括其速度矢量与相对位置矢量夹角 $Q_{i,M}$，夹角速度 $\dot{Q}_{i,M}$，与目标距离 d_i，距离变化速度 $\dot{d}_i$，自身速度 v_i 和加速度 $\dot{v}_i$ 六项。智能体的邻居定义为两智能体相对目标的方位角之间不存在其他智能体。o^i_{ext} 包括分别与两个邻居相对目标组成夹角的差值 $\Delta\alpha$，所有智能体与目标的剩余距离平均值 $\bar{d}$ 两项。$o^{i,j}$ 包括智能体速度和指向

其他智能体的位矢的夹角 $Q_{i,j}$ 、 $Q_{j,i}$ ，两者相对目标的夹角 $\alpha_{i,j}$ ，两者距离 $d_{i,j}$ ，两者与目标的距离差值 $\Delta d_{i,j}$ 五项。

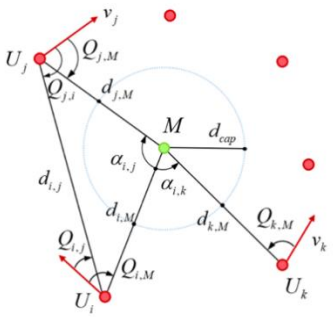


图 2-7 状态空间设计示意图

$$\begin{cases} o_{loc}^{i} = \left\{ Q_{i,M}, \dot{Q}_{i,M}, d_i, \dot{d}_i, v_i, \dot{v}_i \right\} \\ o_{ext}^{i} = \left\{ \Delta\alpha, \bar{d} \right\} \\ o^{i,j} = \left\{ d_{i,j}, Q_{i,j}, Q_{j,i}, \Delta d_{i,j}, \alpha_{i,j} \right\} \end{cases} \tag{2-10}$$

2.3.3 网络结构设计

算法的网络结构包括 Actor 网络和 Critic 网络，其中 Actor 网络将智能体的局部观测映射到连续动作空间的分类分布，Critic 网络将全局状态映射到状态价值的估计，具体设计如图 2-8 所示。

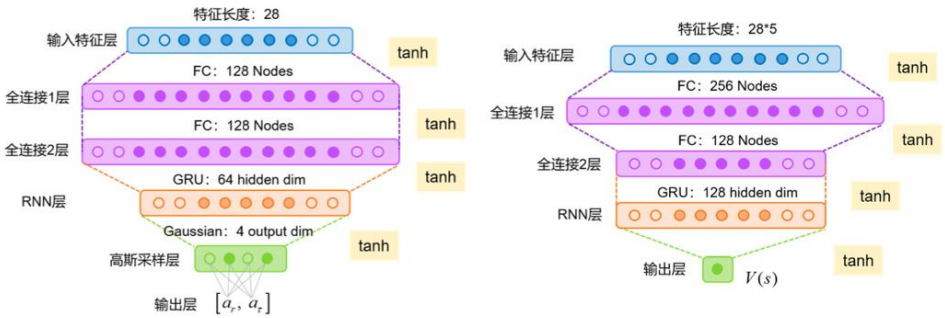


图 2-8 Actor-Critic 网络结构设计

2.3.4 奖励函数设计

本研究设计分段奖励函数来评估围捕智能体当前的状态，并且引导智能体完成围捕任务。在围捕过程中，每个智能体在每个仿真时间内会收获步长奖励 r_{step} ；若智能体 U_i 与邻居形成了合围态势，则收获协同奖励 r_{coop} ；若所有智能体均达成围捕态势，则收获围捕奖励 r_{cap} 。 U_i 的奖励函数定义如式 2-11 所示。

$$R_i = \begin{cases} r_{cap}, & U_k \text{ is done for all } k \in I \\ r_{coop}, & \text{only } U_i \text{ is done} \\ r_{step}, & \text{otherwise} \end{cases} \tag{2-11}$$

当U_i的剩余围捕半径趋近于 0，同时与左右邻居间的围捕角均趋近于期望围捕角时， 称U_i与邻居形成合围态势，其具体数学表达如式 2-12 所示。

$$|d_i| < \delta d \text{ and } |\alpha_{i,j} - \alpha_{exp}| < \delta\alpha \text{ and } |\alpha_{i,k} - \alpha_{exp}| < \delta\alpha \tag{2-12}$$

另外，步长奖励由两部分子奖励加权构成，其表达式如式 2-13 所示。

$$\begin{aligned} r_{step} &= \omega_1 r_f + \omega_2 r_d \\ \omega_1 &+ \omega_2 = 1 \end{aligned} \tag{2-13}$$

r_f为追逐围捕项，可以同时引导智能体完成对目标的追逐和围捕。设每个围捕智能体与目标的位置矢量为$\overrightarrow{p_{i,M}}$，将所有智能体的位矢求和，得到合力$\vec{F}$。如图 2-9 所示，当围捕智能体距离目标较远时，$\vec{F}$的模长较大，可以作为惩罚引导智能体缩小与目标的距离；当围捕智能体距离目标较近时，只有当智能体均匀地分布在目标周围才能使得合力为零，此时引导目标完成包围动作。其表达式如式 2-14 所示，其中k_1为调整系数。

$$r_f = e^{-k_1 \left\| \sum_i \left(\overrightarrow{p_M} - \overrightarrow{p_i} \right) \right\|} - 1 \tag{2-14}$$

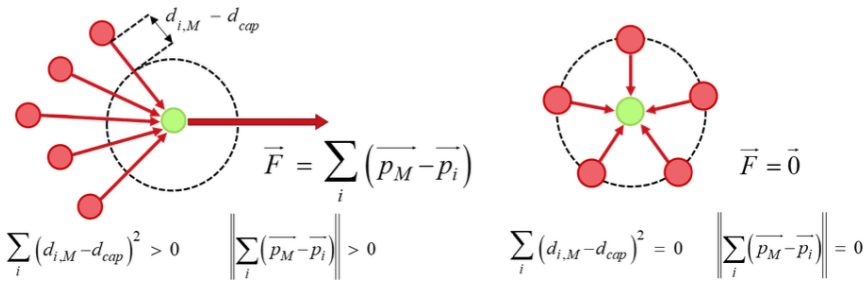


图 2-9 围捕态势示意图

r_d为围捕半径项，引导智能体与目标的距离保持在期望的围捕半径d_{cap}上，其表达式如式 2-15 所示，其中k_2为调整系数。

$$r_d = e^{-k_2 \sum_i \left(d_{i,M} - d_{cap} \right)^2} - 1 \tag{2-15}$$

3 硬件实验

3.1 Gazebo 仿真平台搭建

Gazebo 是一个开源的三维机器人仿真器，旨在模拟机器人在复杂环境中的行为。它采用先进的物理引擎，准确模拟机器人在现实世界中的运动和交互。支持模拟各种传感器，如摄像头、激光雷达，使开发者能够测试感知算法。与 ROS（机器人操作系统）紧密集成，支持 ROS1 和 ROS2，为机器人开发提供强大工具。作为开源项目，用户可以根据需求进行定制和扩展。Gazebo 还支持多个机器人在同一环境中同时运行，为研究多智能体系统提供了可能性。

我们在 Gazebo 仿真平台中采用的是 Lead-Follower（领导者-跟随者）策略，通过键盘节点控制小车运动。

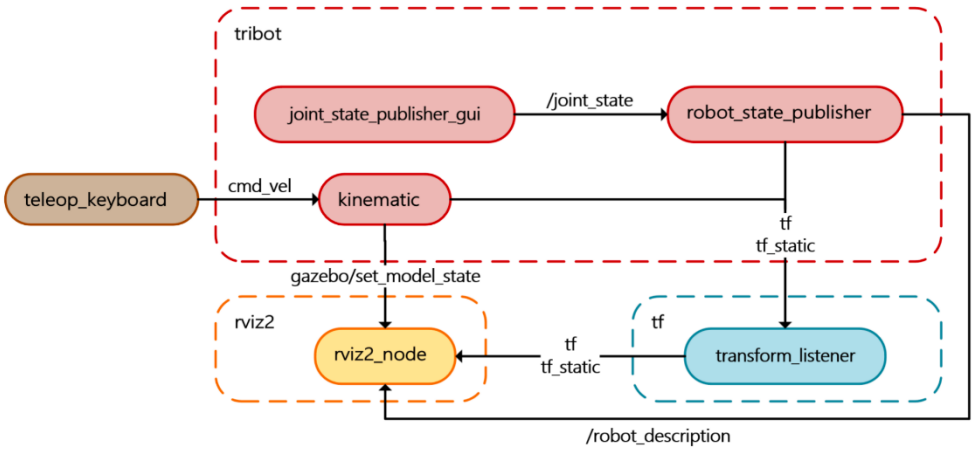


图 3-1 键盘节点控制领导者小车

在 Gazebo 仿真平台中，我们的成果是实现了生成多无人车集群，用键盘节点控制 leader 运动，两个 follwer 小车跟随 leader 实现简单跟随，并且无人车群各个车的质心位置可以实时演示。我们用 PlotJuggler 实现了可视化。下图是可视化效果：

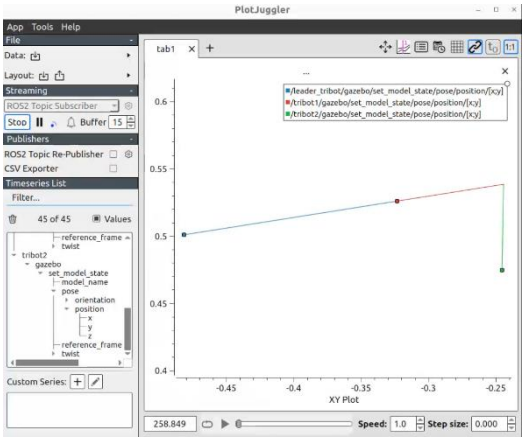


图 3-2 plotjuggler 可视化初始状态

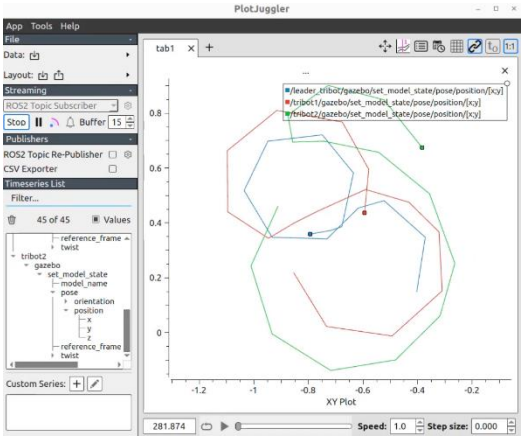


图 3-3 plotjuggler 可视化最终状态

3.2 Marvelmind 定位系统

Marvelmind 定位系统是一项先进的室内定位技术，为机器人、物流系统和其他自动化应用提供了高精度的实时定位和导航功能。该系统的核心原理基于超声波和无线通信技术，通过在空间中布置固定的标签或节点，以及在移动设备（如机器人或车辆）上安装的接收器，实现了精准的定位。在 Marvelmind 系统中，节点通过发射超声波信号，接收器捕获这些信号并计算出设备相对于节点的准确位置。这种方法不仅在室内环境中表现出色，而且具有卓越的高精度。这使得 Marvelmind 系统成为各种领域的理想选择，特别是在需要在有限空间内进行精确定位的场景下。系统的实时性是 Marvelmind 的另一个显著特点，设备能够即时调整其位置，使其在动态环境中表现出色。这种能力对于需要频繁调整路径或适应变化的应用非常关键。

在该项目中，我们完成了 marvelmind 声波定位系统的硬件烧录、初始化环境配置，利用 3D 打印完成了固定信标在实际空间中的放置。测试了环境重部署时间等成本的测试（重部署时间较短，可接受），并初步测试了在 5*3*2 场景中的单一信标和多个信标的定位效果。如下图所示。

图 3-4 marvelmind 部署实验场景

(a)单一信标定位

(b)多个信标定位

图 3-5 marvelmind 信标定位效果

Marvelmind 定位系统在某些情况下可能会遇到精度不足和跳变现象的问题。这可能源于多种原因，其中之一是信号干扰。存在其他无线信号源，如 Wi-Fi、蓝牙或其他 2.4GHz 频段的设备，可能导致定位系统的信号受到干扰，进而影响定位精度。此外，多径效应也是一个潜在问题，当信号通过多个路径传播到接收器时，可能引起信号不同部分以不同路径和时间到达接收器，从而导致跳变或不稳定的定位数据。反射和遮挡物体也可能影响信号的传播路径，进而影响系统的测量精度。

由于声波定位对环境的纯净性的极高要求，所以 marvelmind 的定位精度和信标频率不能满足硬件实验最低要求，我们最终采用了 ROS2 中的 ACML 定位算法（见图 3-7）。

3.3 Omini 全向小车实验

在硬件实验平台搭建中，采用了 omini 小车作为研究对象，该小车具备全向移动的能力，通过三轮的布局实现灵活的定位和导航。搭建的实验平台旨在构建多智能体编队系统，通过硬件配置和软件算法的结合，使小车能够协同工作、交流信息，并实现复杂任务的协同执行。

Omini 小车搭载了 MPU6050 传感器。MPU6050 是一款六轴（三轴加速度计和三轴陀螺仪）传感器，适用于测量和跟踪物体的运动。MPU6050 能够帮助 Omini 小车感知加速度和角速度，从而更好地理解自己的运动状态。MPU6050 内部有数字运动处理器，可以通过 I2C 接口与微控制器通信。通过读取 MPU6050 的数据，Omini 小车可以知道自己是在加速、减速，还是在旋转，从而更智能地应对不同的行驶情况。这个传感器在机器人领域被广泛应用，因为它提供了对运动的高度敏感性和精确性。在 Omini 小车中，MPU6050 的数据有助于实现平稳的运动控制和精确的导航。

为了简化开发过程，采用了虚拟机作为开发环境，选择了合适的虚拟机镜像，并进行了相应的装机操作。在这个过程中，解决了一些异常问题，如软件认证、虚拟机空间不足等，通过执行命令和调整虚拟硬盘空间等操作，确保了开发环境的稳定和可用性。

针对小车的开发，首先对 leader 小车进行了设置，包括开启和联网。通过远程登录小车终端，实现了对小车的远程操作和管理。同时，对小车的 ROS 文件进行了替换和重新编译，以适配实际硬件配置。在此过程中，解决了一些可能出现的报错情况，确保了文件替换和编译的顺利进行。

3.3.1 多个小车之间通信

在机器人系统中，小车之间的直接通信对于协同工作和协同感知等功能至关重要。为了实现这一目标，我们选择使用 ROS 2 作为通信框架，并通过局域网确保小车能够直接交换信息。我们确保每个小车在局域网中拥有唯一的 IP 地址。这可以通过手动设置 IP 地址或者使用 DHCP 自动获取 IP 地址来实现。每个小车的 IP 地址应在同一子网内，以保证它们在同一网络中。

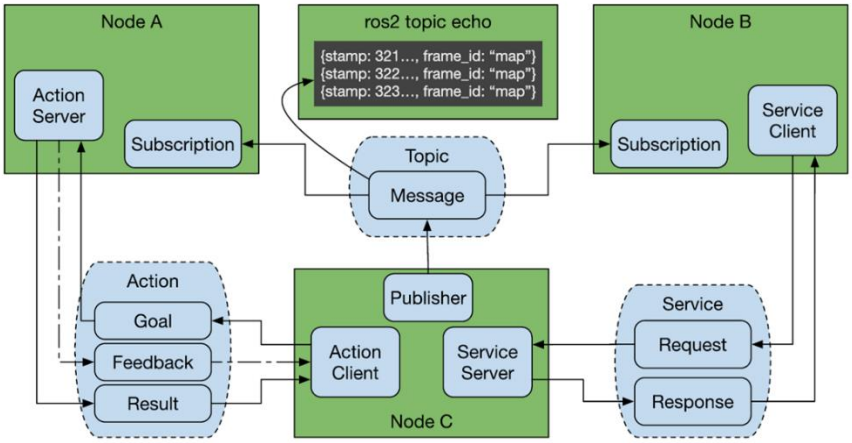


图 3-6 ROS2 通信示意图[14]

在 ROS 2 中，可以使用 DOMAIN（域） 的概念来实现多个 Omni 三轮小车之间的通信隔离。DOMAIN 是 ROS 2 中用于定义通信域的机制，它允许在同一网络中创建多个 ROS 2 系统，每个系统都有独立的域 ID。这种机制可以用于解决多个小车之间通信时可能产生的话题冲突和干扰问题。

通过在每个小车上设置不同的域 ID。在启动 ROS 2 节点时，可以通过设置 ROS_DOMAIN_ID 环境变量或者在 launch 文件中配置域 ID。例如，给第一个小车设置域 ID 1，给第二个小车设置域 ID 2。然后在每个小车上启动 ROS 2 节点，确保它们处于相同的网络中。使用不同的域 ID 来隔离它们的通信。在 ROS 2 中，话题（topics）和服务（services）等通信机制将受到域 ID 的影响。在相同域 ID 下的节点可以正常通信，而不同域 ID 的节点之间则相互隔离。

通过这种方式，每个 Omni 三轮小车可以在自己的域中通信，而不会与其他小车产生通信干扰。这对于在同一网络中运行多个小车系统的场景非常有用，尤其是当它们共享相同的通信通道时。

3.3.2 ACML 定位

在小车定位中使用了 AMCL 算法，该算法是一种基于蒙特卡罗方法的定位算法，用于估计机器人在环境中的位置。蒙特卡罗定位 (Monte Carlo Localization, MCL): MCL 是一种概率定位方法，通过随机采样粒子来估计机器人的位置。每个粒子表示机器人可能的状态，通过加权重采样的方式，根据传感器信息和运动模型对粒子进行更新，从而估计机器人的当前位置。

蒙特卡洛定位适用于局部定位和全局定位两类问题，尽管它相对的年轻，但是已经成为定位领域中的主流算法，如下所示为蒙特卡洛定位算法：

1: **Algorithm MCL**($\mathcal{X}_{t-1}, u_t, z_t, m$):
2: $\bar{\mathcal{X}}_t = \mathcal{X}_t = \emptyset$
3: *for* $m = 1$ *to* M *do*
4: $x_t^{[m]} = \textbf{sample\_motion\_model}(u_t, x_{t-1}^{[m]})$
5: $w_t^{[m]} = \textbf{measurement\_model}(z_t, x_t^{[m]}, m)$
6: $\bar{\mathcal{X}}_t = \bar{\mathcal{X}}_t + \langle x_t^{[m]}, w_t^{[m]} \rangle$
7: *endfor*
8: *for* $m = 1$ *to* M *do*
9: *draw* i *with probability* $\propto w_t^{[i]}$
10: *add* $x_t^{[i]}$ *to* $\mathcal{X}_t$
11: *endfor*
12: *return* $\mathcal{X}_t$

图 3-7 ACML 算法原理

通过把合适的概率运动和感知模型代入到粒子滤波算法中得到，使用 M 个粒子的集合表示置信度 bel(x_t)，初始置信度由先验分布随机产生的 M 个这样的粒子得到。算法第 4 行使用运动模型

采样，以当前置信度为起点使用粒子，第 5 行使用测量模型以确定粒子的重要性权值通过增加粒子总数 M 能提高定位的近似精度。MCL 以目前的形式解决了全局定位问题，但无法从机器人绑架或全局定位失败中恢复过来。当机器人位置被获取时，其他地方的不正确粒子会逐渐消失。在某种程度上，粒子只能“幸存”在一个单一的姿势附近，如果这个姿势恰好不正确，算法就无法恢复。而这个问题可通过相当简单的探索算法解决，其思想是增加随机粒子到粒子集合

在 ROS 中，AMCL 是一个常用的机器人定位软件包，与 ROS Navigation Stack 结合使用。通过订阅机器人传感器数据和运动信息，AMCL 实时更新机器人的位置估计。通过 ROS 参数配置，可以调整 AMCL 算法的各种参数以适应不同的环境和机器人特性。

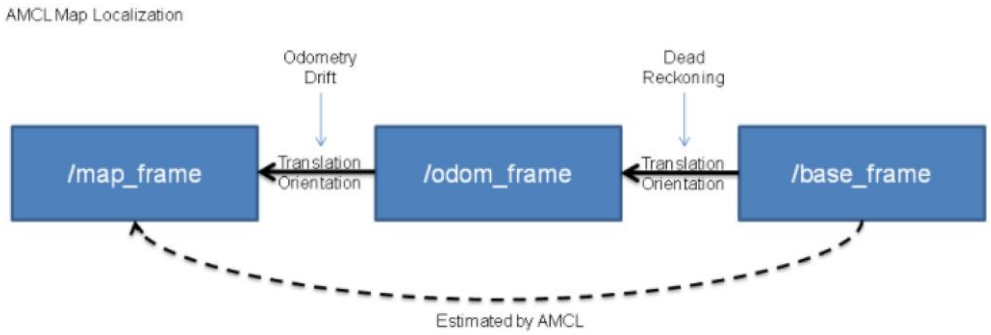


图 3-8 ACML 定位示意图

3.3.3 强化学习环境搭建与算法部署

在 omini 小车上部署强化学习环境与算法是为了使小车能够通过与环境的交互学到最优策略，同时也是本项目的主要创新点之一。使用 Anaconda 进行环境管理有助于简化依赖项管理和版本控制，确保整个系统的稳定性。首先，我们构建了一个强化学习环境，定义了小车的状态空间、动作空间和奖励函数。状态空间包括小车的位置、速度、传感器数据等，而动作空间定义了小车可以采取的动作，如前进、后退、左转、右转等。奖励函数被设计用于评估小车在特定状态下采取特定动作的好坏，直接影响到强化学习算法的训练效果。

在环境交互接口方面，我们实现了小车与强化学习环境的通信，包括状态的获取、动作的执行和奖励的反馈。这涉及与小车的硬件和传感器进行集成，确保环境模拟与真实世界的一致性。选择了适当的强化学习算法，考虑了任务复杂度和计算资源的因素。模型训练阶段中，小车通过不断与环境交互，优化策略以最大化累积奖励。训练完成后，将训练好的强化学习模型部署到小车上，使其能够在实际环境中实时决策和执行动作。实时控制与决策阶段涉及小车在运行时通过学到的策略进行实时决策，并通过控制指令将动作映射到小车的运动控制上。最后，对模型进行评估与调优，确保在实际应用中能够取得良好的性能。

整个部署过程需要考虑小车硬件、通信、算法的协同工作，以实现在复杂环境中的智能决策与行动。

4 研究结果

4.1 强化学习结果

本研究分别使用了 MAPPO、MADDPG 和 MATD3 三种算法来对智能体围捕进行训练，且为保证三种算法具有可对比性，使各算法的公共参数保持一致。

实验设置 3v1 的围捕场景，目标的最大速度为 $v_M = 0.3m/s$，可以得到如图 4-1 所示的群体奖励函数的训练曲线。可以发现，虽然 MAPPO 算法最初的学习效果较差，但在某一节点后开始迅速收敛，并在一个群体奖励较高的范围内振荡，可以认为最后智能体处于一个比较稳定的围捕状态。反观 MADDPG 和 MATD3 算法，由于这两个算法比较相似，因此训练曲线也相差不大，群体奖励都在振荡中上升，但在相同的训练轮数内，最终达到的群体奖励并不算高，智能体只能完成围捕的动作，但无法稳定保持在围捕状态。另外，MATD3 算法是从 MADDPG 算法优化而来的，所以 MATD3 算法总体上群体奖励稍高，训练效果相比之下也更好一点。

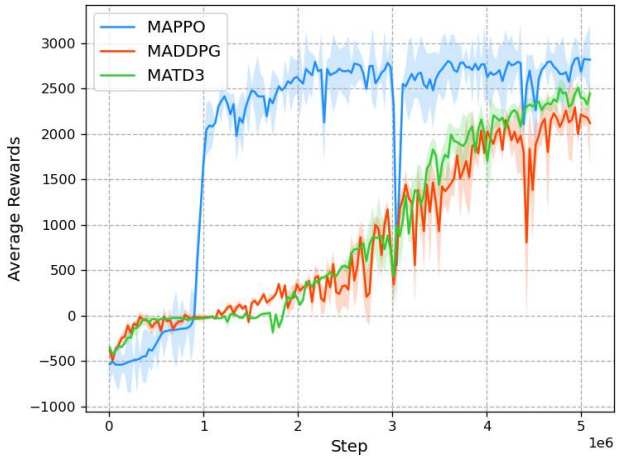
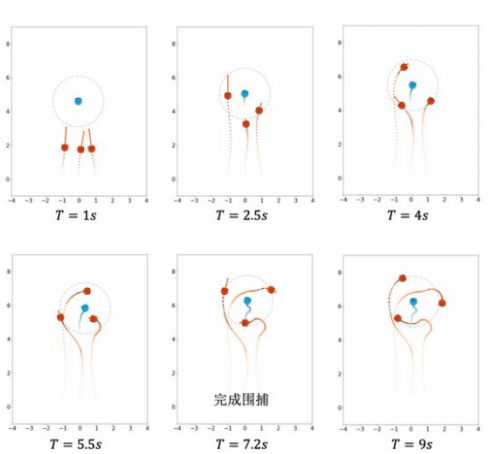
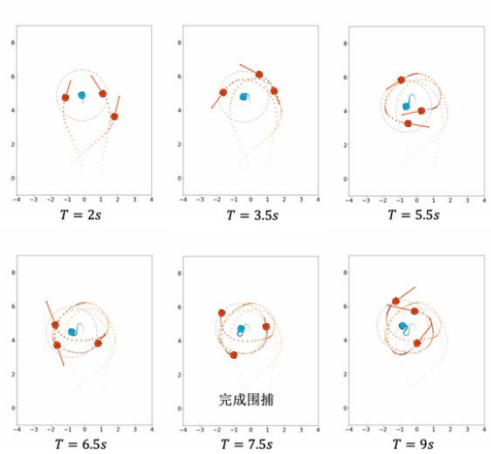


图 4-1 集群整体的奖励函数

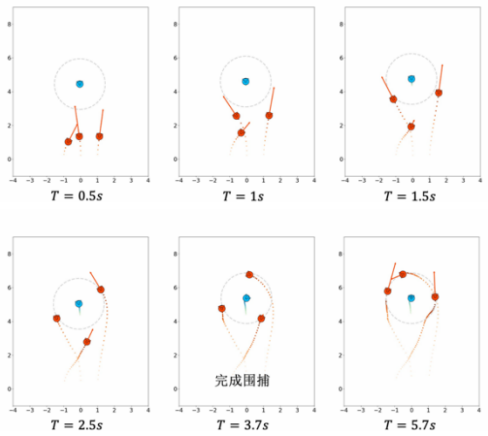
为进一步对比三种算法，给出如图 4-2 所示的三种算法围捕过程轨迹图。从图中可以发现，虽然 MAPPO 算法的围捕时间并不是最快的，但在这个算法下的智能体在围捕成功后，依然保持着围捕状态，围捕智能体会沿着围捕半径环绕在逃逸智能体周围。而其它两个算法在完成围捕后，并不会保持在围捕状态，而是会继续朝着原有方向前进并重新对逃逸智能体进行围捕。在速度方面，可以发现 MATD3 算法有着明显的优势，围捕智能体快速并直接地对逃逸智能体完成了围捕，几乎没有出现绕圈或大幅度调转方向的情况。



(a) 基于 MAPPO 的 3v1 围捕过程



(b) 基于 MADDPG 的 3v1 围捕过程



(c) 基于 MATD3 的 3v1 围捕过程

图 4-2 三种算法围捕过程轨迹图

4.2 硬件实验成果

4.2.1 1v1 直线跟车

在 1v1 直线跟车的场景中，我们建立了一个涉及领导者（leader）和跟随者（follower）的系统，其中领导者通过键盘控制进行直线运动，而跟随者以一定速度匀速跟随领导者。

通过 ROS 2 通信框架，我们成功在两台 omni 小车上实现了 1v1 直线跟车的智能行驶系统。这一系统的核心特点是采用 AMCL（Adaptive Monte Carlo Localization）作为定位算法，确保小车在未知环境中实现精准自主定位。

通过利用小车的惯性进行定位，成功实现了 1v1 直线跟车系统，无需激光雷达的支持。在这一系统中，依然采用了 ROS 2 通信框架，并通过局域网确保两台 omni 小车之间能够直接进行信息交换。

1v1 直线跟车的任务要求领导者小车通过键盘控制实现直线运动，而跟随者小车以 0.1m/s 的匀速跟随。AMCL 的定位算法保证了小车在运动中能够准确感知自身位置，从而实现精准跟随。

通过局域网的搭建，确保了小车之间直接的信息传递，避免了通信延迟，使得 1v1 直线跟车系统更加稳定和可靠。整个系统的搭建涉及到 ROS 2 的配置、AMCL 参数的调整以及局域网的设置，保证了系统的协同工作和高效运行。

图 4-3　1v1 直线跟车实验场景

4.2.2 强化学习部署并进行实验

在这个协同围捕任务中，三台 omini 小车通过强化学习算法的协同作用，成功地实现了对同一坐标目标的围捕。整个过程体现了集群的智能决策与协同行为。

首先，小车通过搭建的强化学习环境学到了有效的策略。这包括了在特定状态下采取何种动作才能最大化奖励，从而使得小车能够更加智能地应对各种情境。这个学习过程在模型训练阶段得以完成，确保小车具备了在实际环境中做出适应性决策的能力在围捕任务中，小车们通过实时的信息交流和协同决策，追踪目标并采取合适的动作。每台小车根据自身的状态信息和学到的策略，以及其他小车的位置信息，共同制定出最优的行动方案。这种协同决策保证了围捕行动的高效性和协调性。

围捕的过程本质上是一个协同追逐与包围的策略。小车们根据目标的移动情况，灵活地调整位置，相互之间通过 ROS 2 通信框架进行实时信息传递，以确保最优的围捕效果。整个过程中，三台小车协同工作，通过智能决策实现了对目标的成功围捕。

这个任务的效果体现在围捕目标的成功率和围捕时间的优化上。通过强化学习算法的不断优化，小车们逐渐学到了更为高效的围捕策略，使得围捕过程更迅速、更智能。最终，这三台 omini 小车成功地展现了在协同智能框架下的出色团队协作和目标围捕效果。

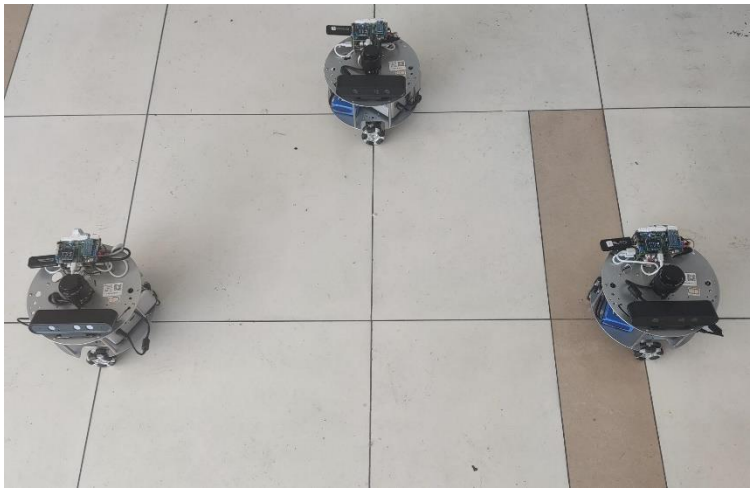


图 4-4 三台小车围捕实验场景

5 总结与展望

本项目深入研究了强化学习中的 Actor-Critic 框架以及 MAPPO、MADDPG、MATD3 等算法在多智能体围捕任务中的应用。在 Actor-Critic 框架中，Actor 和 Critic 两个组件相互协作，Actor 负责决策，Critic 负责评估动作的价值，通过环境反馈进行调节，实现对环境政策和行为的改进。MAPPO 算法在多智能体任务中表现出较高的学习效率，通过 Actor-Critic 架构实现对环境和状态政策的学习。MADDPG 算法继承了 DDPG 算法，通过集中式训练、分布式执行适用于不稳定环境，可应用于多种多智能体场景。MATD3 算法是对 TD3 算法的优化，适用于多智能体环境，通过双重 Q 学习、目标策略平滑等机制在连续控制任务上取得良好表现。

在围捕任务中，本研究通过对状态空间、动作空间、网络结构和奖励函数的设计，实现了三种算法对智能体围捕的训练。MAPPO 算法在稳定性和效率上表现出色，MADDPG 和 MATD3 算法虽然相似，但在群体奖励和围捕效果上略显不足。MATD3 算法在速度和稳定性上有明显优势，快速而直接地完成围捕，避免了绕圈和大幅度调转方向的情况。

在机器人系统的仿真和硬件实验上，本项目主要借助 Gazebo 仿真平台和各种先进技术。在仿真方面，Gazebo 是我们的核心工具，提供了开源的三维机器人仿真环境，用于模拟机器人在复杂场景中的行为。我们采用 Lead-Follower 策略，在仿真中成功实现了多无人车集群，通过键盘节点控制领导者，而两个 follower 小车则实现了简单的跟随行为。硬件方面，Marvelmind 定位系统和 Omini 全向小车成为关键组成部分。Marvelmind 通过超声波和无线通信实现高精度室内定位，而 Omini 小车具备全向移动的能力。我们搭建了强化学习环境，通过 Anaconda 进行环境管理，使小车能够在与环境的交互中学到最优策略。整个研究项目中，我们还解决了小车之间通信、定位算法、强化学习部署等关键问题。在硬件实验中，通过 ROS2 通信框架和 ACML 定位算法，我们成功实现了 1v1 直线跟车和协同围捕任务，展示了小车在复杂环境中的智能决策与协同行动。

展望未来,可以进一步优化算法,考虑更复杂的多智能体场景,例如从二维空间扩展到三维。同时，结合深度学习和强化学习的发展趋势，探索新的算法和模型结构，提高算法在实际应用中的适用性和鲁棒性。此外，对于围捕任务的实验设置和评价指标也可以进行更深入的研究，以提高实验结果的可解释性和可比性。硬件实验平台的搭建以及对不同定位方案的尝试为算法在实际应用中的验证提供了有益的经验，未来可以继续在实际场景中进行更深入的验证和优化。

6 参考文献

[1] 夏家伟，朱旭芳，张建强，罗亚松，刘忠. 基于多智能体强化学习的无人艇协同围捕方法研究[J/OL]. 控制与决策.

[2] Dong X, Yu B, Shi Z, et al. Time-varying formation control for unmanned aerial vehicles: Theories and applications[J]. IEEE Transactions on Control Systems Technology, 2014, 23(1): 340-348.

[3] Antonelli G, Arrichiello F, Chiaverini S. The entrapment/escorting mission for a multirobot system: Theory and experiments[J]. 2007 IEEE/ASME International Conference

[4] on Advanced Intelligent Mechatronics, 2007: 1-6. Hafez A, Iskandarani M, Givigi S, et al. UAVs in formation and dynamic encirclement via model predictive control[J]. IFAC Proceedings Volumes, 2014, 47(3): 1241-1246.

[5] Sarwal A, Agrawal D, Chaudhary S. Surveillance in an open environment by cooper ative tracking amongst sensor enabled robots[C]//2007 International Conference on Information Acquisition. 2007: 345-349.

[6] Ryan Lowe，Yi Wu，Aviv Tamar，Jean Harb，Pieter Abbeel，Igor Mordatch. Multi-Agent

Actor-Critic for Mixed Cooperative-Competitive Environmentsar. Xiv:1706.02275v4 [cs.LG] 14 Mar 2020.
[7] 杨立炜，付丽霞，李萍. 多智能体系统编队控制发展综述. 研究与设计，2020年12月
[8] 胡鸿翔，梁锦，温广辉，等.多智能体系统的群集行为研 究综述 ［J］. 南京信息工程大学学报（自然科学版），2018，10
[9] Vu Phi Tran, Matthew Garratt and Ian R.Petersen. Time-Varying Formation Control of a Collaborative Multi-Agent System Using Negative-Imaginary Systems Theory . arXiv:1811.06206v1 [cs.MA] 15 Nov 2018
[10]Yan, Z., Jouandeau, N., Cherif, A. A. (2013). A survey and analysis of multi-robot coordination. International Journal of Advanced Robotic Systems, 10(12), 399.
[11]孙长银，穆朝絮. 多智能体深度强化学习的若干关键科学问题［J］. 自动化学报，2020, 46(7)：1301-1312
[12]宗群，王丹丹，邵士凯，等. 多无人机协同编队飞行控制研究现状及发展［J］. 哈尔滨工业大 学学报，2017，49(3):1-14.
[13]Nvidia Jetson nano 安装Archiconda: https://blog.csdn.net/qq_41451125/article/details/116262611
[14]使用路由器搭建局域网并链接小车:https://blog.csdn.net/sru_alo/article/details/102483225
[15]ROS2入门21讲图文教程,通信接口: https://zhuanlan.zhihu.com/p/558154968