



西北工业大学

本科毕业设计论文

题 目 基于强化学习的轮式机器人决策算法研究

专业名称 软件工程

学生姓名 韩 萧 阳

指导教师 史 豪 炳

毕业时间 2019.07

摘要

强化学习是最近倍受关注的学习模型。强化学习通过智能体与环境的交互进行学习，以改变智能体的动作响应以获得适应环境最优的行为策略。深度强化学习通过应用深度神经网络模型，提升了强化学习对于高维复杂问题上的性能，已经广泛应用于机器人领域。然而在真实环境中，对于轮式机器人的复杂控制任务和决策系统，深度强化学习往往需要大量的额外工作以保证数据有效性和样本利用效率。

本文对轮式机器人决策系统和强化学习算法做了简要的介绍，接着对于基于值函数和策略搜索的深度强化学习，描述了它们的基本原理。同时详细阐述了基于值函数的 DQN 和基于演员 - 评论家模型的 DDPG 的实现和训练过程。此外，本文提出了应用多智能体异步训练的方法，以解决深度强化学习在实际应用中面临的轮式机器人时延问题和采样速度问题。

本文基于机器人操作系统 ROS 和仿真平台 Gazebo 搭建了轮式机器人物理仿真环境，并在该环境内进行了 DQN、DDPG 以及多智能体异步 DDPG 的训练和测试，验证了它们的性能和效果。在真实环境下，通过使用上述算法的轮式机器人智能体与人工设计的确定型有限状态机的敌方轮式机器人进行对战，探究了上述算法在真实环境中的适应能力。

关键词： 强化学习，ROS，轮式机器人，决策系统

ABSTRACT

As a recently-recognized learning model, reinforcement learning learns through the interaction between the agent and the environment to change the action response of the agent to obtain the optimal policy. Deep neural networks provide rich representations that can enable reinforcement learning algorithms to perform effectively. Reinforcement learning methods have been applied to range of robotic tasks. However, for complex control tasks and decision system of wheeled robots in the real world environment, deep reinforcement learning typically requires significant additional work to ensure data validity and sample utilization efficiency.

In this paper, we give a brief introduction to wheeled robot decision system and reinforcement learning algorithm. We describe the basic principles of deep reinforcement learning based on value function and policy search. In the meanwhile, we elaborate the realization and training process of Deep Q-learning and Deep Deterministic Policy Gradient. In addition, we propose a method of applying multi-agent asynchronous training to solve the problem of wheeled robot delay and sampling efficiency in practical applications.

Based on the robot operating system ROS the the simulation platform Gazebo, we build a wheeled robot physics simulation environment, in which we train DQN, DDPG and multi-agent asynchronous DDPG and verify their performance and effects. We explore the adaptability of the above algorithm by using the wheeled robot agent to fight against enemy with human design DFA in the real environment.

Key Words: reinforcement learning, ROS, wheeled robot, decision system

目 录

第一章 绪论	1
1.1 研究背景	1
1.1.1 轮式机器人决策系统研究现状	1
1.1.2 强化学习研究现状	2
1.2 研究内容	3
1.3 章节安排	4
第二章 轮式机器人决策系统	5
2.1 问题定义	5
2.2 决策系统架构	6
2.2.1 架构概述	6
2.2.2 模组通信	7
2.2.3 模组功能	7
2.3 决策系统设计	8
2.3.1 确定性有限状态机	10
第三章 强化学习算法研究与分析	12
3.1 强化学习基本原理	12
3.1.1 马尔可夫决策过程	12
3.2 强化学习算法	14
3.2.1 值函数	15
3.2.2 策略搜索	16
3.2.3 深度强化学习	17
3.3 深度强化学习算法	18
3.3.1 Deep Q-learning	19
3.3.2 Deep Deterministic Policy Gradient	20
3.3.3 多智能体异步 DDPG	22
第四章 实验数据采集与分析	24
4.1 仿真平台	24
4.1.1 ROS 简介	24
4.1.2 RVIZ 与 Gazebo	24
4.2 真实环境	25
4.3 奖励函数定义	26
4.4 代码设计	27
4.4.1 Deep Q-learning	28
4.4.2 Deep Deterministic Policy Gradient	28
4.4.3 多智能体异步 DDPG	29

4.5 实验结果.....	30
4.5.1 仿真实验结果	30
4.5.2 真实环境下对抗结果.....	30
第五章 全文总结	32
5.1 全文总结.....	32
5.2 对未来工作的展望	32
参考文献	33
致谢	35
毕业设计小结.....	36

第一章 绪论

1.1 研究背景

机器人技术是机械、电子、控制、计算机、人工智能等多学科交叉的领域。进入 21 世纪以来，国内外对机器人技术的发展越来越重视，机器人技术被认为是对未来新兴产业发展具有重要意义的高新技术之一 [1]。机器人的研发、制造与应用是衡量一个国家科技创新和高端制造业水平的重要标志。

机器人技术的研究和应用已经从传统的工业领域快速扩展到其他领域，如医疗健康、家政服务、外形探索、勘测勘探等。无论是传统的工业领域还是其他领域，对机器性能要求的不断提高，使机器人必须面对更极端的环境、完成更复杂的任务。

许多国家加大对机器人技术的研究投入，并将其作为未来新兴产业寄予厚望，是未来高技术、新兴产业竞争的制高点，对于国家经济发展和国防建设具有重要意义。

近年来，我国在国家自然科学基金、863 计划以及国家科技重大专项等规划中对机器人技术给予了极高的关注度。国际上，美国启动了“美国国家机器人计划” [2]。欧盟在第七框架计划 (FP7) 中规划了“认知系统与机器人技术”研究。日本制定了机器人技术长期发展战略。韩国制定了“智能机器人基本计划”。

1.1.1 轮式机器人决策系统研究现状

轮式移动机器人主要有智能轮椅、导游机器人、野外侦察机器人、大型智能车辆等。其定位、运动规划、自主控制、服务作业等技术和方法也得到广泛研究。随着人工智能、计算机网络技术、传感器技术等新技术的飞速发展，以及工业程度的不断提高，轮式机器人能够更好的服务社会。

目前对于轮式机器人的研究工作，主要集中在路径规划方法上。路径规划是轮式机器人研究领域的关键技术之一，旨在规划一条从起点到目标点的无碰撞路径，同时优化性能指标如距离、时间或者能耗，其中距离是最常采用的方法 [3]。

本文着重于在可靠路径规划算法的基础上，即在一台拥有可靠定位、避障和路径规划的机器人上，思考和探索一种能够适应复杂环境做出自主决策的轮式机器人决策系统。目前，模糊逻辑 [4]、决策树 [5]、状态机、遗传算法 [6]、神经网络 [7] 等都是较为成功有效的轮式机器人决策方法。但这些方法通常需要假设完整的环境信息，然而，在大量的实际应用中需要智能体具

有适应不确定环境的能力。因此，如何提高机器人路径规划的自学能力和自适应性成为当前研究的关键技术。

强化学习 (Reinforcement Learning, RL) 方法通过智能体与位置环境交互，并尝试动作选择使累积回报最大，该方法通常运用马尔可夫决策过程 (Markov Decision Processes, MDP) 进行环境建模。马尔可夫决策过程模型主要针对理想情况下的单智能体系统，智能体环境的不确定性也可由部分可观测马尔可夫决策过程 (Partially Observable Markov Decision Processes, POMDP) 进行描述。强化学习算法不需要给定任何状态下的指导信号，只通过智能体与环境交互进行学习并优化控制参数，在先验信息较少的复杂优化决策问题中具有广阔的应用前景。

1.1.2 强化学习研究现状

人工智能的一个首要目标就是生成能够与环境交互，通过尝试与错误学习以优化自身行为的全自主的智能体。创造一个能够有效学习的人工智能系统一直以来是一个长期的挑战，从能够感知和与周边环境进行交互的机器人到与自然语言与多媒体交互的基于软件的智能体。强化学习是一种经验驱动的全自主学习的数学方法框架 [8]。

强化学习的目标是需要学习一种策略，即当智能体 agent 处于一种状态 state，做出一个动作 action 的决策。如果我们将动作看作对状态的标签，强化学习就可以类比监督学习，这样策略就相当于一个分类器或者回归器。主要的区别在于强化学习的数据往往需要通过尝试、和环境进行交互获得。算法则根据环境给予的反馈来调整策略。

强化学习的任务通常使用马尔可夫决策过程描述。智能体 agent 处于一个环境中，每个状态 state 为 agent 对环境的感知。当智能体 agent 执行一个动作后，环境会按照概率转移到另一个状态；同时，环境会根据奖励函数给予智能体 agent 一个反馈，通常是奖励 reward。综合而言，强化学习主要包含四个要素：状态 state、动作 action、转移概率 P 以及奖励函数 reward。

在过去，人工智能通过强化学习达成了许多成就。然而，先前的方法缺乏可泛化性并且只能在定义在相当低维空间的问题有效。随着深度神经网络的广泛使用，函数逼近和特征学习这两大法宝使我们不断克服这些问题。

深度学习的优势在机器学习的许多领域都有重大作用，显著地提升了在经典任务上的表现，例如目标检测，文本识别和语言翻译 [9]。深度学习最重要的特征就是深度神经网络可以自动地从图像、文本和声音等高维数据中抽象出简洁的低维特征。通过在深度神经网络中设计启发式的偏差，尤其是层级表示，机器学习使用者在解决维度灾难方面做出了有效的进步 [10]。随着使用

深度学习算法的强化学习算法，深度强化学习算法的使用，深度学习也同样地加速了强化学习的进步。

深度学习使强化学能够泛化应用到先前极为棘手的决策问题，比如高维状态 - 动作空间。在最近的强化学工作中有三项工作的成功极为突出。

首先是深度强化学习的革命的临门一脚，能够以人类水平从图像像素级别学习游玩雅达利 2600 个电子游戏的智能体 [11]。通过为强化学中函数逼近技术的不稳定性提供解决方案，这项工作首次令人信服地证明了强化学习智能体可以仅基于奖励信号在原始的高位观察结果上进行训练。

第二个突出的成功是开发了混合深度强化学系统 AlphaGo [12]，它在击败了围棋领域的人类世界冠军。与主导围棋系统的人工设计的下棋策略不同，AlphaGo 是由使用监督学习和强化学训练的神经网络组成，并结合传统的启发式搜索算法，即蒙特卡洛搜索算法。

最新的令人惊讶的工作是由 OpenAI 基于 Dota 2 应用场景开发的通用 AI 系统 Open Five，它通过学习团队合作、长期规划和隐藏信息，开始捕捉到真实世界复杂性和连续性，并在 5V5 的 Dota 2 游戏中击败了人类顶尖的职业选手。OpenAI Five 表明，当前的深度强化学可以实现大规模的长期规划。

深度强化学算法已经应用与广泛的问题，例如机器人技术，其中一些机器人可以直接从现实世界的摄像机输入学习控制策略 [13] [14]，而取代了人工设计的控制区或者从机器人状态的低维特征中学习。为了向更强大的智能体迈进。深度强化学已被用于创建可以进行元学习的智能体 [15] [16]，允许模型推广到他们从未见过的复杂视觉环境。

虽然电子游戏是一个有趣的挑战，但是学习如何玩雅达利或者 Dota 电子游戏并不是深度强化学的最终目标。深度强化学背后的驱动力之一是创建能够学习如何适应现实世界的智能系统。从调度管理到装载物品，深度强化学可以增加能够被自主学习的物理任务的数量。但是深度强化学不知预测，因为强化学是通过反复验证来解决优化问题的一般方法。从设计最先进的机器翻译模型到构建新的优化功能，深度强化学已经被用于处理各种机器学习任务。并且，与深度学习在机器人学习的许多分支中的应用相同，在未来深度强化学将是构建通用人工智能系统的重要组成部分。

1.2 研究内容

本文将使用轮式机器人中应用最广泛的操作系统 ROS 作为研究平台。ROS 因具有完备的跨平台消息传递机制和进程处理能力而广泛应用于机器人研究中。Gazebo 是 ROS 平台上一种功能强大的仿真环境模拟平台，本文使用 Gazebo 作为仿真和模拟实验的主要平台。Tensorflow 作为应用最广泛的开源

深度学习框架之一，能够使开发者迅速构建深度学神经网络模型，研究深度学习算法，本文采用 Tensorflow 作为搭建深度神经网络的基础框架。OpenAI gym 是一个应用于开发搭建和比较研究强化学习的开源工具组件，它提供了简单易用的强化学习环境搭建框架接口和模板。本文使用 OpenAI gym 强化学习环境接口来封装 ROS 中轮式机器人感知、通信和控制等功能，使环境与全自主轮式机器人决策系统解耦，从而能够更加方便地在仿真环境与真实环境中切换。

本文的主要研究对象是全自主的轮式机器人，通过在轮式机器人上应用深度强化学习模型，期望使其具有在复杂真实环境场景下自主与敌方机器人作战的能力，即能够自主巡航、索敌、追踪和攻击。本文的主要工作有如下几个方面：

1. 深入分析目前主流的强化学习模型，解析其理论本质。
2. 搭建仿真平台，模拟在复杂条件下全自主轮式机器人的决策过程。
3. 探索深度强化学习、逆强化学习和模仿学习等多种模型与方法，以在轮式机器人上实现决策功能。
4. 在现有理论基础上，针对轮式机器人决策问题，对网络结构、奖励函数和训练方法上做出改进。

1.3 章节安排

本文的章节安排如下：

第1章为绪论，介绍了轮式机器人决策系统的概况，研究意义，给出了本文的设计方法，最后介绍了本文的研究目的，内容及结构。

第2章概述轮式机器人决策系统概述。定义了本文探究的问题边界，概括介绍了轮式机器人控制系统架构，定义了轮式机器人决策系统和接口。

第3章详细分析了强化学习算法研究与分析，阐述了本文实验中主要使用的三种深度强化学习方法，DQN、DDPG 以及多智能体异步 DDPG。

第4章详细分析了实验数据采集与分析，分析了在仿真平台与真实环境下的训练特点与挑战，展示了实验数据与结果。

第5章总结全文，并展望未来的研究工作。

第二章 轮式机器人决策系统

2.1 问题定义

本文研究的轮式机器人决策问题是基于 ICRA DJI 机甲大师人工智能挑战赛 (ICRA DJI RoboMaster AI Challenge) 的。该问题的定义为：在给定的室内有边界的场地内，由敌我双方各使用两台搭载有传感器与弹丸发射机构的轮式机器人在场地内进行对战。挑战赛比赛场地，是一个长为 8 米、宽为 5 米的长方形区域，主要包含启动区、补给区、防御加成区、障碍块区和保护围挡区，如图 2.1 所示。

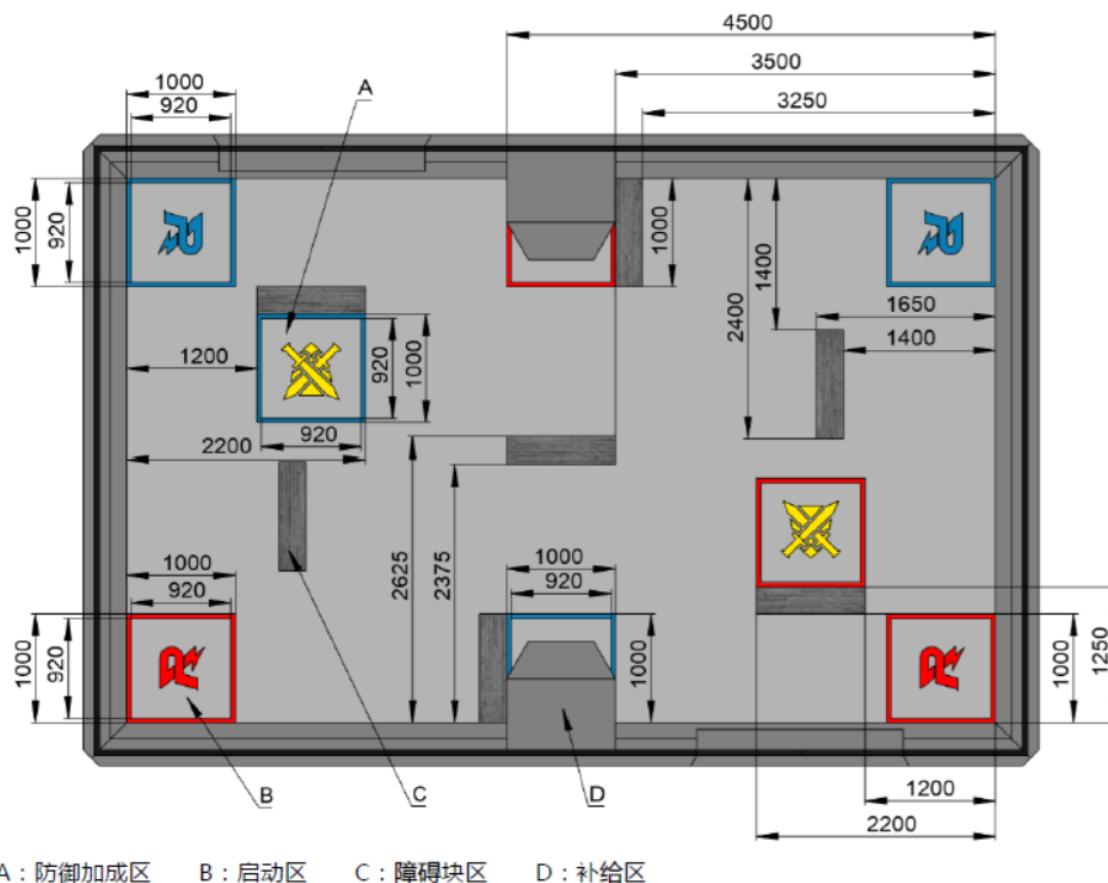


图 2.1: 挑战赛场地

敌我双方各一台轮式机器人分别从各自图 2.1 的 B 启动区启动，在挑战赛场地内进行全自主的对抗。在挑战赛场地内，设置诸多不可移动的灰色障碍物如图 2.1 的 C 障碍块区所示。同时设置了 A 防御加成区，轮式机器人在这一区域中停留超过 5s 后即可获得伤害减半的 buff 持续时间 30s。在 D 补给区己方轮式机器人可以获得弹药补给。

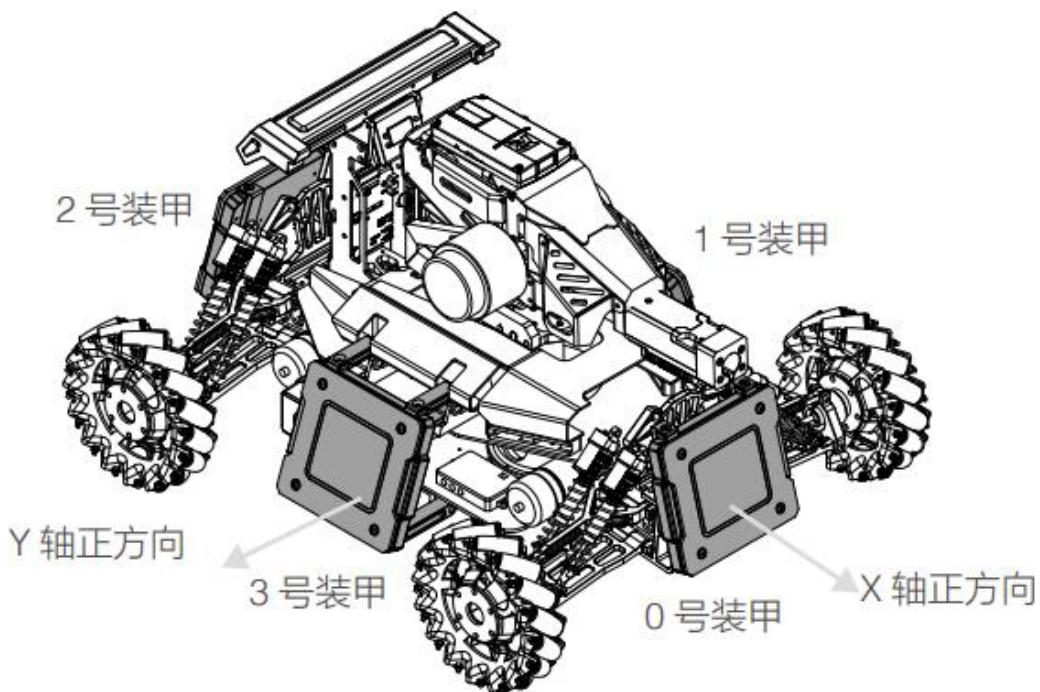


图 2.2: 轮式机器人与装甲模块

轮式机器人可以搭载激光雷达、摄像头、UWB、超声波等多种传感器。轮式机器人四周各装有装甲模块。装甲模块装有红蓝两种色彩的 LED 用以区别敌我，同时装有压力传感器以检测车身是否被击中，并计算剩余血量。装甲模块如图2.2所示。

2.2 决策系统架构

2.2.1 架构概述

在应用与实践中，我们总结形成一套基于 ROS 的轮式机器人的控制软件系统。该架构包括该架构包括驱动模组、感知模组、规划模组、控制模组和决策模组。

驱动模组通过选取适合的驱动软件包调用摄像头、雷达、声呐、UWB 等传感器数据以获取轮式机器人实时环境信息；通过解析步兵战车通信协议，构造串口驱动类，实现对轮式机器人的控制。感知模组通过雷达、声呐、UWB 等传感器信息实现地图构建、实时定位；通过轻量级深度神经网络：SSD-MobileNets 达到了在 Jetson TX2 开发组件上的实时目标定位与追踪。规划模组主要通过优化 Navigation 功能包，实现了轻量级的路径规划器，使轮式

机器人在有限的计算资源下，完成了全自主巡航功能。控制模组使用了离散式增量 PID 控制，其结合经典控制理论与 SIMULINK 仿真技术，实现了对目标物的低超调高速自动追踪。

2.2.2 模组通信

我们使用 ROS 的 Publish/Subscribe 机制实现各个模组之间的通信。各个模组的数据流如图2.3所示。各个 ROS 消息结点的消息传递关系如图2.5所示。

驱动模组负责将轮式机器人状态、比赛进度、电机里程计数据、摄像头图像、激光雷达云点阵、UWB 定位等数据推送给感知模组。感知模组使用目标检测、物体识别与实时定位算法，确定当前己方轮式机器人与敌方轮式机器人在场地中的位置等状态信息。决策系统根据状态信息，将决策己方轮式机器人做出的反应动作，即目标位置与姿态，并将其推送给规划模组以经行路径规划。规划模组根据目标位置与姿态使用导航算法得出底盘与云台运动的速度等数据。这些数据由控制模组处理后，通过驱动模组中的串口驱动发送给下位机执行。

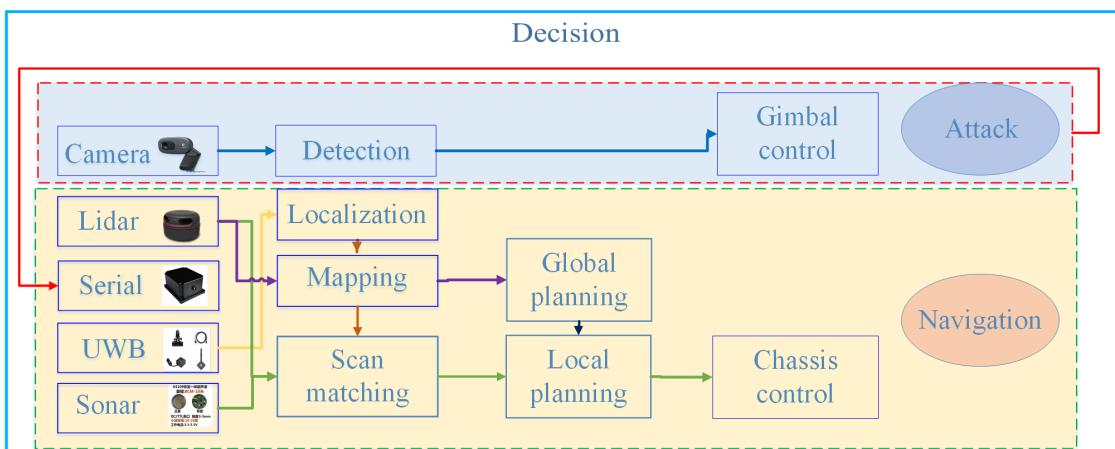


图 2.3: 模组通信数据流

2.2.3 模组功能

总体而言，决策系统主要依赖改软件系统的两大功能：导航和攻击，如图2.4所示。导航功能主要负责在挑战赛环境下自主避障、路径规划。攻击功能主要负责实时检测、追踪和打击敌方目标。决策系统建立在可靠的导航与攻击功能上，实际上负责协调导航与攻击功能模块的交互。决策系统可以视为模组功能沟通与协调的结点，通过接收感知信息以合理地调用各个模组功能。

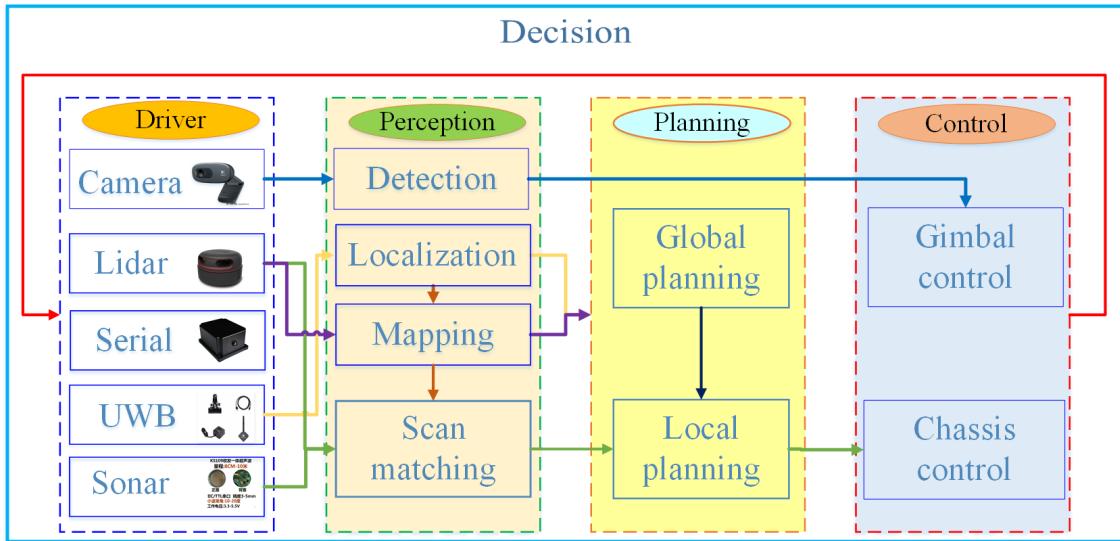


图 2.4: 模组功能

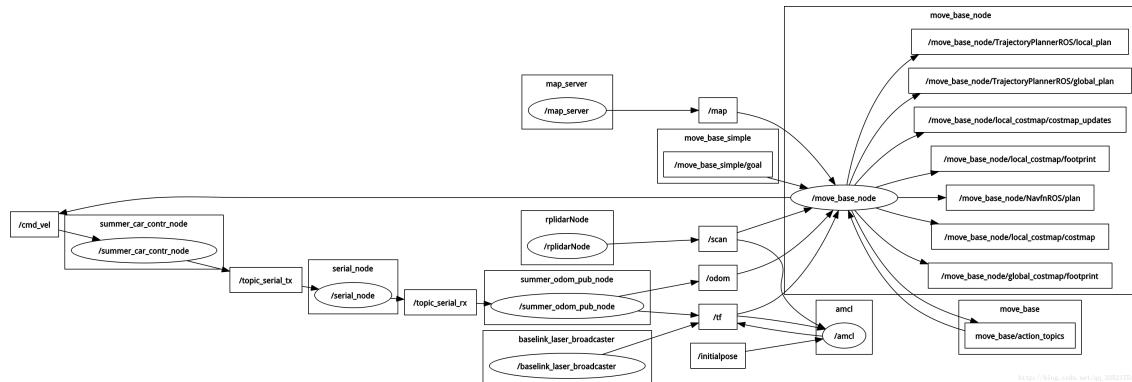


图 2.5: ROS 消息结点关系

2.3 决策系统设计

在 ROS 平台上，决策系统实际上是一个消息处理结点，它通过处理轮式机器人的感知信息，向轮式机器人发布控制命令，已协调调用各个模组的功能，从而能够实现轮式机器人全自主的巡航、索敌、追踪与打击，如图2.6 所示。

在决策系统设计时，考虑到可扩展性、在虚拟环境与真实环境下的适应性、与深度强化学习算法的兼容性，我们使用 OpenAI gym 标准接口封装了轮式机器人感知、通信与控制功能，构建一个环境适配器类 env，从而实现了智能体 agent 与环境 env 的交互。智能体 agent 通过状态 state、动作 action 和奖励 reward 与环境 env 进行交互，从而进行训练，如图2.7 所示。

按照 OpenAI gym 标准，我们将复杂的轮式机器人感知与控制抽象为智能体 agent 和环境 env 之间交互的三个函数：

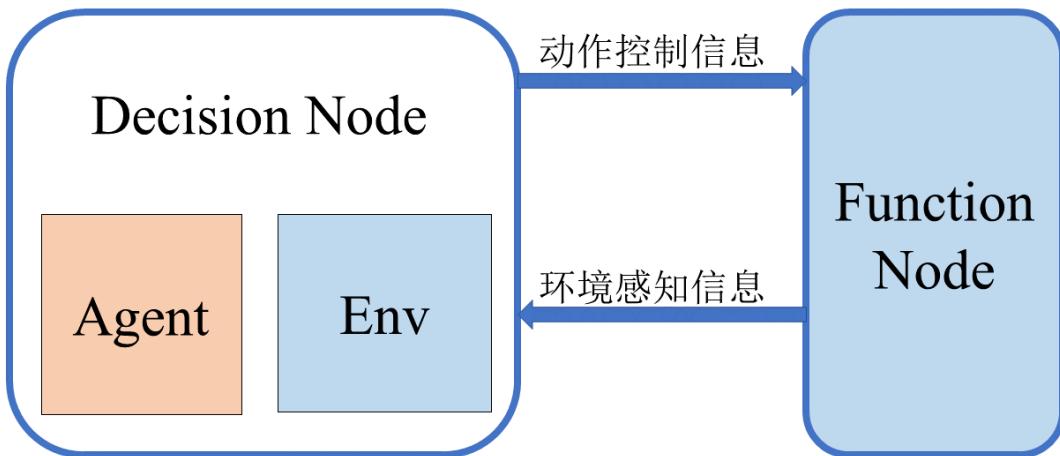


图 2.6: 决策结点消息交互接口

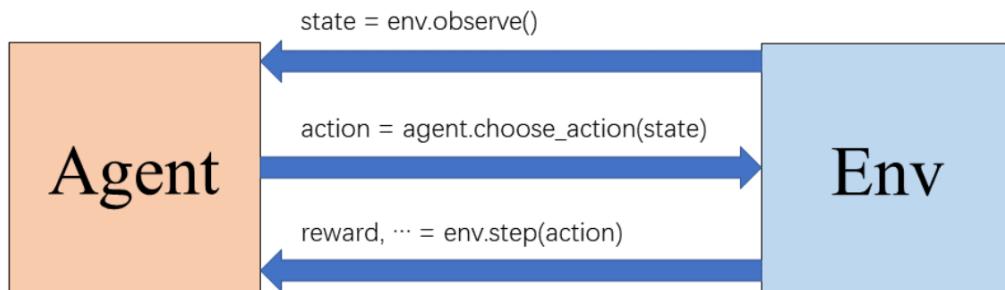


图 2.7: 轮式机器人智能体与环境交互接口

1. 感知环境，`env.observe`。同步消息机制。输入为空，输出为轮式机器人智能体当前在环境中感知的状态。
2. 动作决策，`agent.choose action`。同步消息机制。输入为轮式机器人智能体当前状态，输出为轮式机器人智能体自主做出的动作响应决策。
3. 执行动作，`env.step`。同步消息机制。输入为轮式机器人智能体做出的动作响应，输出为环境反馈给轮式机器人智能体的奖励或惩罚和下一步。

应用适配器模式的思想，将智能体 `agent` 做出动作决策与环境 `env` 交互解耦。环境 `env` 隐藏了决策系统结点与 ROS 目标识别、定位、移动控制等其他消息结点的交互细节，使智能体 `agent` 只关心与环境 `env` 的交互。

环境 `env` 无需关心智能体 `agent` 的实现细节，通过继承 `env`，可以是智能

体在仿真轮式机器人环境与真是轮式机器人环境之间切换，从而提高了工作效率，减少了实验好耗时。

同样的，智能体 agent 无需关心环境 env 与轮式机器人的交互细节，通常来说，智能体 agent 只需要实现输入一个状态向量或矩阵 s 时，输出一个动作响应向量或矩阵 a ，就能够与环境 env 进行交互。因此，智能体 agent 可以在不同的强化学习算法、不同的神经网络模型之间灵活切换，甚至可以替换为非强化学习模型，如确定性有限状态机和行为树。

2.3.1 确定性有限状态机

决策系统上我们使用人工设计的确定性有限状态机作为本文实验的基线，通过对所有可能出现的情况与相对应的行为来构建有限状态机，其中状态图如图2.8所示，活动图如图2.9 所示，我们成功地对轮式机器人的行为策略进行规划，并在实际实践中取得良好的效果。

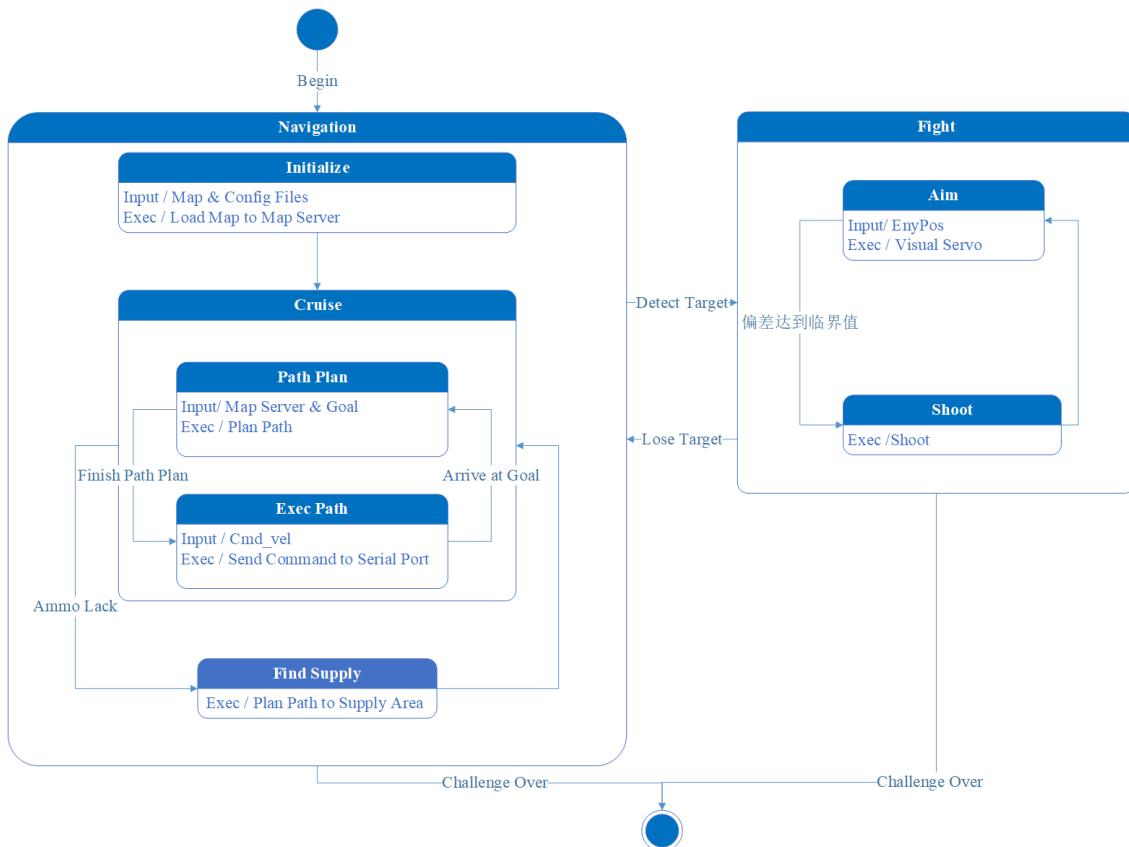


图 2.8: 轮式机器人确定性有限状态机

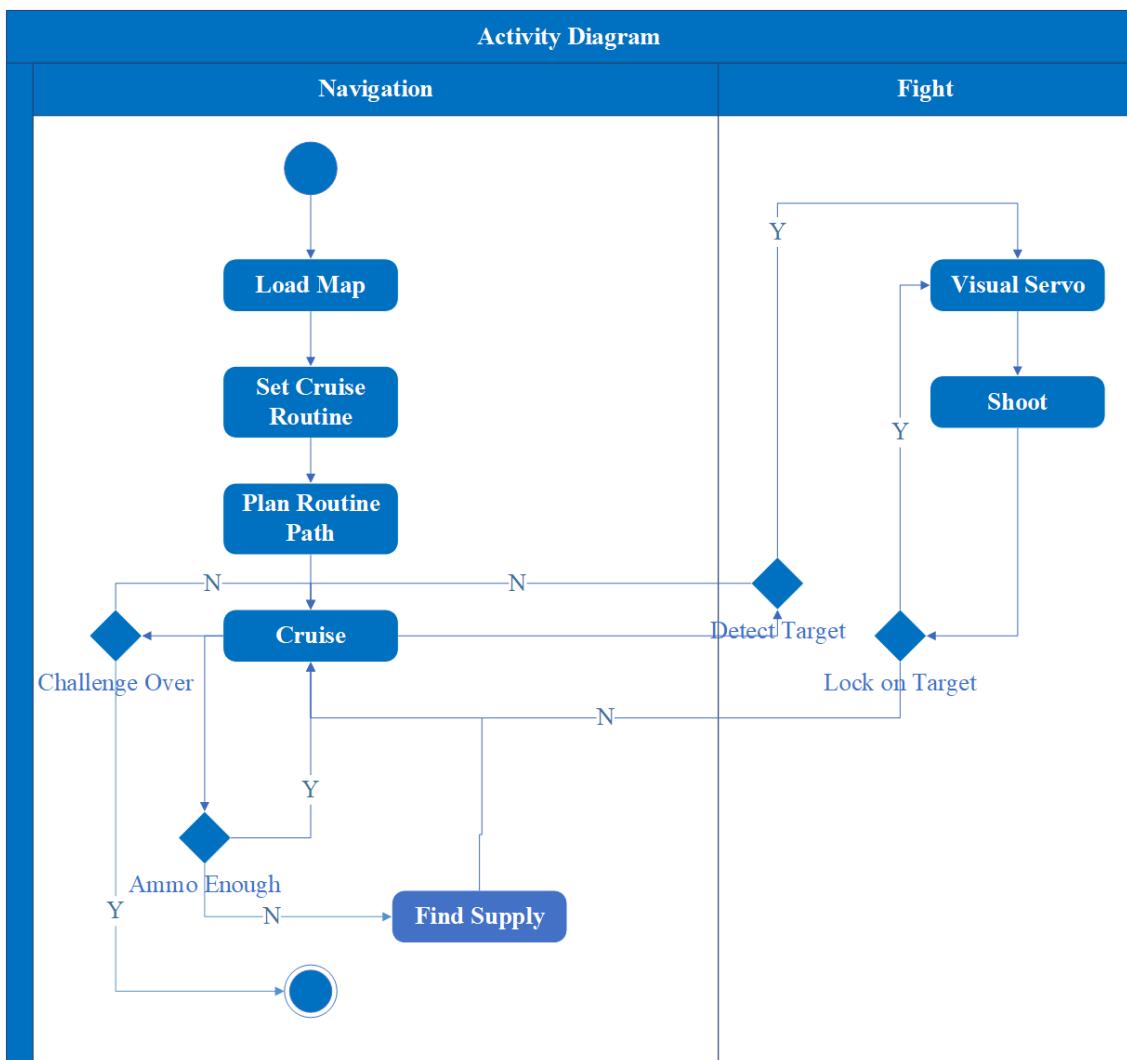


图 2.9: 轮式机器人活动图

第三章 强化学习算法研究与分析

3.1 强化学习基本原理

强化学习的本质是通过与环境的交互进行学习。强化学习智能体与环境进行交互，并在观察到其行为的结果时，通过学习来改变自己的行为方式以响应所获得的奖励或惩罚。这种试错学习方法的范式来源于行为主义心理学，是强化学习的主要基础之一 [17]。另一个影响了强化学习的关键思想史最优控制，它提供了支撑该领域的数学形式。

在强化学习过程中，有机器学习算法控制的自主智能体在 t 时刻从环境中观察到状态 s_t 。智能体通过在状态 s_t 中执行动作 a_t 来与环境交互。当智能体执行动作时，环境和智能体将会根据当前的状态和所选择的动作转移到新的状态 s_{t+1} 。状态是对环境的充分统计数据，因此包括智能体采取最佳动作的所有必要信息，比如轮式机器人在环境中的位置。

最佳的动作顺序取决于环境提供的奖励。每次环境转换到新的状态时，它还会向智能体提供一个标量奖励 r_{t+1} 作为反馈。智能体的目标是学习一种最大化预期累加折扣的回报的策略。给定一个状态，策略返回要执行的动作，最优的策略就是能够最大化环境预期回报的策略。在这方面，强化学习旨在解决与最优控制相同的问题。然而，不像最优控制模型，强化学习中挑战的是智能体在不能获得状态转移的模型的情况下，通过反复试验来了解环境中行为的后果。与环境的每次交互都会产生信息，智能体会使用这些信息来更新其知识。这种感知、动作、学习的循环如图3.1所示。

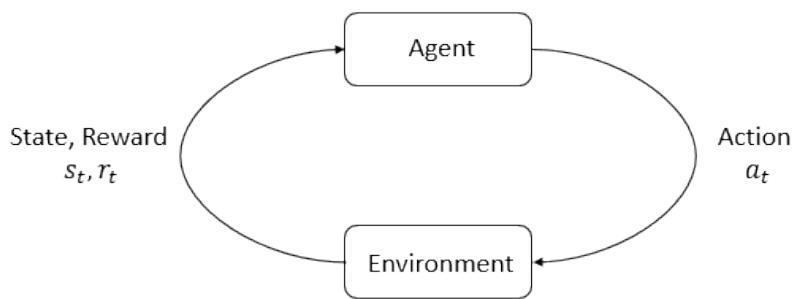


图 3.1: 强化学习基本模型

3.1.1 马尔可夫决策过程

强化学习过程可以用马尔可夫决策过程 (MDP, Markov Decision Process) 描述，它包含了以下定义：

- \mathcal{S} , 状态集合, 包含一个初始状态分布 $p(s_0)$ 。
- \mathcal{A} , 动作集合。
- $\mathcal{T}(s_{t+1}|s_t, a_t)$, 状态转移函数, 从 t 时刻的状态 - 动作对映射到 $t + 1$ 时的状态分布。
- $\mathcal{R}(s_t, a_t, s_{t+1})$, 瞬时奖励。
- $\gamma \in [0, 1]$, 折扣参数, 较低的折扣参数就会使模型更加重视瞬时奖励, 较高的折扣参数则会使模型更加重视长期奖励。

通常来说, 策略 π 定义了在特定时间特定状态下的行为方式, 是一个从状态到每个可能的动作的概率的映射: $\pi : \mathcal{S} \rightarrow p(\mathcal{A} = a|\mathcal{S})$ 。 $\pi(a_t|s_t)$ 就是指在状态 s_t 时选择执行动作 a_t 的概率。

在马尔可夫决策过程中, 一个智能体从初始状态 $s_0 \in \mathcal{S}$ 开始, 在 t 时刻, 观察到状态 s_t , 根据策略 $\pi(a|s_t)$ 选择一个动作 $a_t \in \mathcal{A}$ 执行, 同时根据状态转移函数 $\mathcal{T}(s|s_t, a_t)$, 改变了智能体在环境中的状态到 $s_{t+1} \in \mathcal{S}$, 并使智能体获得奖励 $r = \mathcal{R}(s_t, a_t, s_{t+1})$ 。这一过程, 如图3.2 所示。

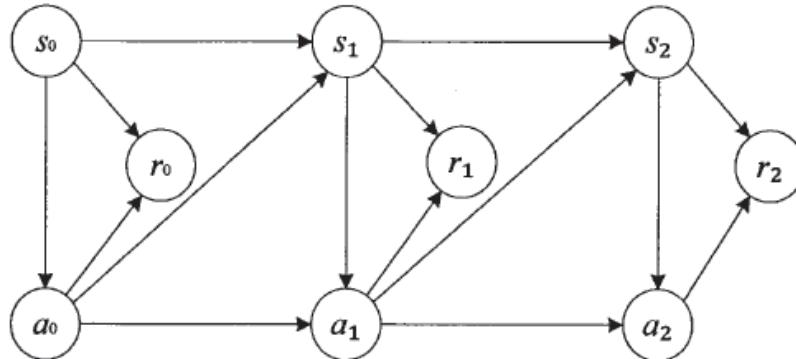


图 3.2: MDP 动态过程

对于情节性任务 (Episodic Task), 指状态会在时间 T 后结束并重置, 以自然结束智能体 (agent) 和环境 (environment) 的交互, 即所有任务可以被分解成一系列情节 (episode), 而情节的数目是有限的。那么在每一个情节中, 状态、动作和奖励的序列就组成一个轨迹 (trajectory)。一种策略中每一个轨迹所对应的瞬时奖励的折扣累加就是这个轨迹的回报 (return)。

$$R = \sum_{k=0}^{T-1} \gamma^k r_{t+k}. \quad (3.1)$$

对于连续性任务 (Continuing Task)，指任务不会自然结束，会一直持续到 $T = \infty$ 。在这种情况下，我们使用 $\gamma < 1$ 以保证累加奖励趋向于无穷。对于这种情况，我们不能计算一个完整轨迹的返回值，而是使用轨迹的一个有限子集。

强化学习的目标就是找到一个最优策略 π^* 以最大化回报的期望。

$$\pi^* = \operatorname{argmax}_{\pi} \mathbb{E}[R|\pi]. \quad (3.2)$$

对于基于马尔可夫决策过程理论的强化学习来说，有一个重要的概念，那就是只有当前的状态会影响下一个状态，换而言之，在给定当前状态的情况下，未来状态与过去的状态是条件无关的。这意味着在状态 s_t 做出的任何动作决策只依赖于前一状态 s_{t-1} [18]，而非 s_0, s_1, \dots, s_{t-1} 。即使这一假设被大多数强化学习方法采纳，但是这在本次实验的真实环境下是不现实的。因为这要求智能体的状态是能够被完全观察到的。

实际上当处于真实环境下，轮式机器人智能体只能观察到真实世界的部分状态。因此使用与马尔可夫决策过程的一般情况，部分可观测马尔可夫决策过程 (POMDP)。在部分可观测马尔可夫决策过程中，智能体接收到一个观察结果 (observation) $o_t \in \Omega$ 。观察结果是一个依赖于当前状态和先前动作的概率分布： $p(o_{t+1}|s_{t+1}, a_t)$ 。

在给定先前置信状态，采取的动作和当前的观察状态下，部分可观测马尔可夫决策过程通常维护一个对当前状态的置信度。深度学习中更常见的方法是利用递归神经网络 (RNN, Recurrent Neural Network)。它与前馈神经网络不同，是一个动态系统。当真实状态无法获知只能估计时，这种方法通过动态系统和状态空间模型解决了部分可观测马尔可夫决策过程。

3.2 强化学习算法

目前强化学习面临着许多挑战：

- 强化学习必须通过与环境的反复实验来推断最优策略。智能体收到的唯一学习信号是奖励。
- 智能体的观察取决于其行为并且很可能包含强时间相关性。
- 智能体必须处理长时间的依赖关系。通常，一个动作的后果在环境状态进行多次转移后才会显现。

以轮式机器人自主决策作战为例。如果目标位置已经确定，我们可能能够估计剩余的距离，并将其作为奖励信号，但我们不太可能确切地知道轮式

机器人需要采取什么样的动作序列才能达到目标。由于轮式机器人必须在做出行进目标决策的同时导航，它的决策影响了它所能感知到的状态空间。最后，在导航了几个交叉路口后，轮式机器人可能会发现自己处于一条死路。从学习动作的后果到平衡探索与实践之间存在一系列问题，但这最终都可以来强化学习框架内解决。

根据是否基于模型，可以将强化学习算法分为基于模型学习 (model-based learning) 和模型无关学习 (model-free learning) 两种。

我们定义智能体学习和优化的策略称为目标策略，把智能体与环境进行实际交互行为的策略称为行为策略。根据智能体学习到的目标策略与智能体与实际环境交互的行为策略是否相同，我们将智能体分为同步策略学习 (on-policy learning)，又称在线学习和异步策略学习 (off-policy learning)，又称离线学习两种。

解决强化学习问题的方法主要分为两种：基于值函数的强化学习和基于策略搜索的强化学习。同时也有两种方法的混合，基于演员 - 评论家 (actor-critic) 模型的强化学习。

3.2.1 值函数

值函数方法基于估计所处状态的回报期望。状态值函数 $V^\pi(s)$ 描述了处于状态 s 时，后续采取策略 π 的回报期望：

$$V^\pi(s) = \mathbb{E}[R|s, \pi]. \quad (3.3)$$

最优策略 π^* 对应的值函数 $V^*(s)$ 就是相应的最优值函数：

$$V^*(s) = \max_{\pi} V^\pi(s) \forall s \in \mathcal{S}. \quad (3.4)$$

如果我们已经得到最优的值函数 $V^*(s)$ ，最优策略 π^* 就可以通过回溯法得到。当智能体处于状态 s_t 时，在所有当前可以选择的动作中选择一个动作 a 最大化 $\mathbb{E}_{s_{t+1} \sim \mathcal{T}(s_{t+1}|s_t, a)}[V^*(s_{t+1})]$ 。

在强化学习的过程中，状态转移函数 \mathcal{T} 是不可得的。因此，我们构建状态动作值函数 $Q^\pi(s, a)$ ，即 Q 函数：

$$Q^\pi(s, a) = \mathbb{E}[R|s, a, \pi] \quad (3.5)$$

在给定 Q 函数的情况下，最优策略可以通过在每一个状态下贪心选择动

作 $a = \operatorname{argmax}_a Q^\pi(s, a)$ 来找到。在这种情况下，我们也可以定义 $V^\pi(s)$ ，即 $V^\pi(s) = \max_a Q^\pi(s, a)$ 。

为了实际学习 Q^π ，我们利用马尔可夫属性并将函数定义为 Bellman 方程，其具有以下递归形式：

$$Q^\pi(s_t, a_t) = \mathbb{E}[r_{t+1} + \gamma Q^\pi(s_{t+1}, \pi(s_{t+1}))]. \quad (3.6)$$

这意味着可以通过自助法 (Bootstrapping) 来优化 Q^π ，即我们可以使用对 Q^π 的估计的当前值来改进我们的估计。这是 Q-learning [19] 和 SARSA [20] 等算法的基础。

对于 off-policy 的 Q-learning 算法有：

$$Q^\pi(s_t, a_t) \leftarrow Q^\pi(s_t, a_t) + \alpha(r_t + \gamma \max_a Q^\pi(s_{t+1}, a)). \quad (3.7)$$

对于 on-policy 的 SARSA 算法有：

$$Q^\pi(s_t, a_t) \leftarrow Q^\pi(s_t, a_t) + \alpha(r_t + \gamma Q^\pi(s_{t+1}, a_{t+1})). \quad (3.8)$$

为了从任意 Q^π 中找到最优 Q^* ，我们使用广义策略迭代，其中策略迭代包括策略评估和策略改进。策略评估改进了值函数的估计，这可以通过最小化根据当前策略采取的路径产生的 TD 误差来实现。随着估计的改进，通过基于更新的值函数贪婪地选择动作，自然可以改善策略。通用策略迭代允许交错步骤，而不是单独地执行这些步骤以进行收敛，从而可以更快地进行。

3.2.2 策略搜索

策略搜索方法不需要维护值函数模型，而是直接搜索最优策略。通常选择参数化策略，之后使用基于梯度或无梯度的更新优化这些参数来最大化回报期望。使用神经网络对策略进行编码已经成功适用于无梯度和基于梯度的训练方法。无梯度优化可以有效地覆盖低维参数空间，尽管它们在应用于大型网络方面取得了一些成功 [21]，但基于梯度的训练仍然是大多数深度强化学习的首选方法。当策略拥有大量参数时，基于梯度的训练的样本效率更高。

在直接构造策略时，通常输出概率分布的参数；对于连续动作，这可以是高斯分布的均值和标准偏差，而对于离散动作，这可以是多项分布的个体概率。这样做的结果是一个可以直接采样动作的随机策略。使用无梯度方法，优化的策略则需要在预定义的模型类中进行启发式搜索。

策略梯度 (Policy Gradients): 对于如何改进参数化策略，梯度可以提供强有力的学习信号。然而，为了计算回报期望，我们需要对当前策略参数化引入的合理轨迹进行平均。这种平均需要确定性近似或通过采样的随机近似 [22]。确定性近似只能应用于基于模型的算法，其中可以获得状态转移函数的模型。在更常见的模型无关的 R 强化学习算法中，通常使用蒙特卡罗方法估计回报期望。对于基于梯度的学习，这种蒙特卡罗近似提出了挑战，因为梯度不能通过随机函数的这些样本。因此，我们转向梯度的估计量，在强化学习中称为 REINFORCE 规则 [23]。REINFORCE 规则可以用于计算随机变量 X 的函数 f 对于参数 θ 的期望梯度：

$$\nabla_{\theta} \mathbb{E}_X[f(X; \theta)] = \mathbb{E}_X[f(X; \theta) \nabla_{\theta} \log p(X)]. \quad (3.9)$$

由于该计算依赖于轨迹的经验回归，因此得到的梯度具有高方差。通过引入噪声较小的无偏估计，可以减少方差。执行此操作的一般方法是减去基线，这意味着通过优势而不是纯回报来加权更新。

演员 - 评论家 (Actor-critic): 可以将值函数与策略的 2 显示表示相结合，从而产生演员 - 评论家方法，如图3.3所示。“演员”(策略)通过使用“评论家”(值函数)的反馈来学习。在这样做时，这些方法通过值函数方法的偏差引入来权衡策略梯度的方差减少 [23] [24]。

演员 - 评论家方法使用值函数作为策略梯度的基线，因此演员 - 评论家方法与其他基线方法之间唯一的根本区别在于演员 - 评论家方法利用了一个优化过的值函数。

3.2.3 深度强化学习

深度强化学习的许多成功都是基于将强化学习的先前工作扩展到高维问题。这归功于低维特征学习和神经网络的强大函数逼近特性。通过特征学习，深度强化学习可以有效地处理维度灾难。例如，卷积神经网络 (CNN, Convolutional Neural Network) 可以用作强化学习智能体的组件，允许它们直接从原始的高维视觉输入中学习。通常，深度强化学习通过训练神经网络以近似最优策略 π^* ，或者最优点函数 V^* , Q^* 。

尽管当前深度强化学习成功使用无梯度的方法，但是绝大多数当前的工作依赖于梯度的反向传播算法。当可用时，梯度能够提供强大的学习信号。实际上，这些梯度是基于近似，通过采样或其他方式估计的，因此我们必须设计具有有效的归纳偏差的算法，以使它们易于处理。

反向传播的另一个好处是将回报期望的优化是为随机函数的优化。这个函数可以包含多个部分，包括模型、策略和值函数，通过各种方式组合。单个部

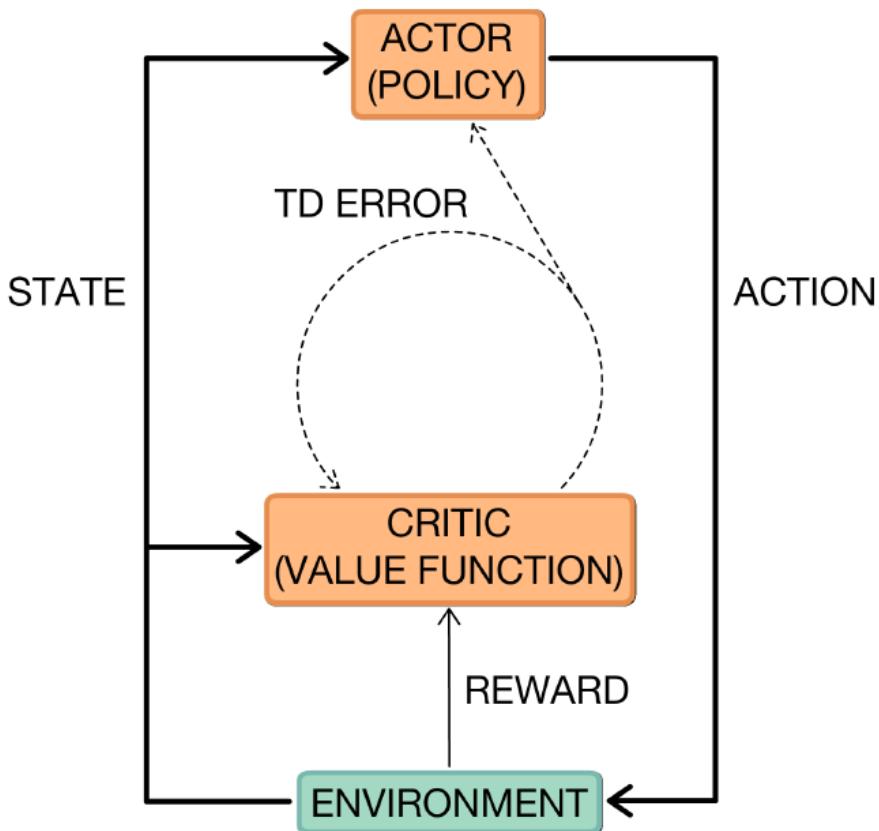


图 3.3: 演员 -评论家 (Actor-critic) 方法

分，例如值函数，可能不会直接优化回报期望，而是可以体现强化学习领域的有用信息。例如，使用可区分的模型和策略，可以在整个过程中进行前向传播和后向传播；另一方面，不确定性可能在很长一段时间内累积，此时使用值函数来总结统计数据可能是恰当的。我们之前已经提到特征学习和函数逼近是深度强化学习成功的关键，但也可以说深度学习领域激发了对强化学习的新思维方式。

3.3 深度强化学习算法

深度强化学习已经准备好在 AI 领域掀起一场革命。深度强化学习向建立对于真实世界具有高水平理解能力的全自主系统迈出了坚实的一步。当前，深度学习使强化学习有能力处理先前极为棘手的问题。深度强化学习也同样应用于机器人学领域，能够直接从真实世界摄像头的图片输入学习机器人控制策略。在本文中，我们首先介绍强化学习的通常概念，之后进一步阐述基于值和基于策略的方法。本文将覆盖深度强化学习的核心算法，包括 DQN [11]、DDPG [26]、异步 DDPG [27]。同时，我们强调使用深度神经网络的优势，着

重于关注于强化学习的理解。

3.3.1 Deep Q-learning

Deep Q-learning 是第一个将深度学习模型与强化学习结合在一起从而成功地直接从高维的输入学习控制策略的算法。在 Q-learning 中使用 Q 值表来存储状态 - 动作对相应的 Q 值。然而当状态和动作空间是高维连续时，使用 Q 值表则不切实际。解决方法就是利用值函数近似，通过函数来近似 Q 值分布。借助深度神经网络来表示这个 Q 值函数就是 DQN 的核心思想。

通常 DQN 的实现中，会把收集的状态、动作、执行动作后的状态和奖励等信息存在内存中，训练的时候多次使用，称为回放记忆 (replay memory)。注意到每个状态 - 动作对的 Q 值都要拟合，一个函数拟合自己可能引入额外的噪声，所以通常使用一个延迟更新的函数 Q' 来求新的 Q 值，称为目标值网络 (target network)，如图3.4所示。如算法1所示。

DQN 的改进包括 Double DQN [28]、Dueling DQN [29] 和 Prioritized Replay [30]。Double DQN 是在引入了 target network 后，改进 Q 值的计算方法，目的是减少因为 Max Q 值计算带来的计算偏差，或者称为过度估计问题。考虑到 Q 值和状态，动作都相关，但我们实际上更注重动作带来的奖励，Dueling DQN 对网络结构做了改进。Prioritized Replay 探讨在回访记忆采样的优先级问题。

DQN 以端到端的训练方式开创了深度强化学习对于高维输入的控制策略问题的解决方案。通过回放记忆解决相关性及非静态分布问题，使用目标值网络解决稳定性问题。但其在处理长时间的动作序列问题上仍然存在问题。

Algorithm 1 Deep Q-learning

- 1: 初始化容量为 N 的回访记忆 \mathcal{D}
 - 2: 随机初始化动作值函数 Q
 - 3: **for** episode = 1, M **do**
 - 4: 初始化状态 s_1
 - 5: **for** $t = 1, T$ **do**
 - 6: 以概率 ϵ 选择一个随机动作 a_t
 - 7: 否则选择动作 $a_t = \max_a Q^*(s_t, a; \theta)$
 - 8: 执行动作 a_t 得到奖励 r_t 并观察到环境 s_{t+1}
 - 9: 保存 (s_t, a_t, r_t, s_{t+1}) 到回访记忆 \mathcal{D}
 - 10: 从回访记忆中采样一个 minibatch(s_t, a_t, r_t, s_{t+1})
 - 11: 计算 $y_j = \begin{cases} r_j, & s_j \text{ 为终止状态} \\ r_j + \gamma \max_{a'} Q(s_{j+1}, a'; \theta), & s_{j+1} \text{ 不为终止状态.} \end{cases}$
 - 12: 计算 $(y_j - Q(s_j, a_j; \theta))^2$ 的梯度下降并反向传播
 - 13: **end for**
 - 14: **end for**
-

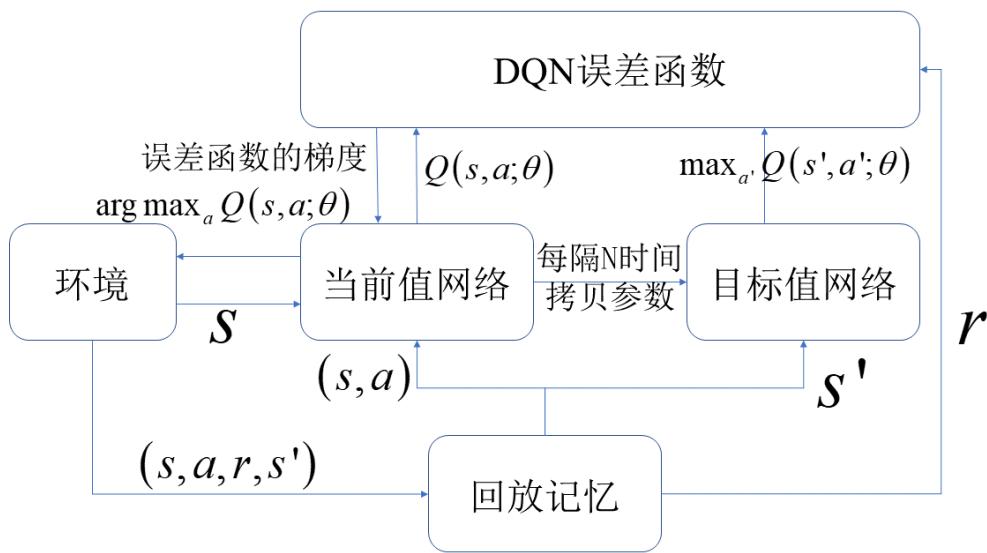


图 3.4: DQN 算法流程图

3.3.2 Deep Deterministic Policy Gradient

深度确定性策略梯度 (DDPG, Deep Deterministic Policy Gradient) 是利用 DQN 扩展 Q-learning 算法的思路，对确定性策略梯度方法进行改造，基于前述演员 - 评论家框架的算法。该算法可用于解决连续动作空间上的深度强化学习问题。

DDPG 中第一个 D 是深度神经网络，当概率策略的方差趋近于 0 的时候，就是确定性策略，其运用了 Actor-Critic 框架，把 DQN 和策略梯度混合了起来，显著提高样本利用率。如果动作空间也是连续的，那么就无法直接取到最大的 Q 值，那么我们再用个深度学习网络，称为演员 Actor，演员的任务就是选 Q 值大的动作（确定性策略），演员的梯度来自值函数的估计网络（评论家），这个做法的最大优势是，能够离线地更新策略，即像 DQN 一样，从回放记忆采样出数据来训练，如图3.5所示。区别于 DQN，DQN 每隔一定的迭代次数后，将参数复制给实现网络；而 DDPG 中目标网络的参数每次迭代都以微小量逼近实现网络的参数。

在 DDPG 中，分别使用参数为 θ^μ 和 θ^Q 的深度神经网络来表示确定性策略 $a = \pi(s|\theta^\mu)$ 和动作值函数 $Q(s, a|\theta^Q)$ 。其中，策略网络用来更新策略，对应演员；智网络用来逼近状态动作对的值函数，并提供梯度信息，对应评论家。

目标函数被定义为带折扣的回报期望。

$$J(\theta^\mu) = \mathbb{E}_{\theta^\mu} \left[\sum_{t=0}^T \gamma^t r_{t+1} \right]. \quad (3.10)$$

通过随机梯度法对目标函数进行端到端的优化。目标函数关于 θ^μ 的梯度等价于 Q 值函数关于 θ^μ 的期望梯度：

$$\frac{\partial J(\theta^\mu)}{\partial \theta^\mu} = \mathbb{E}_s \left[\frac{\partial Q(s, a | \theta^Q)}{\partial \theta^\mu} \right]. \quad (3.11)$$

根据确定性策略 $a = \pi(s | \theta^\mu)$, 可得

$$\frac{\partial J(\theta^\mu)}{\partial \theta^\mu} = \mathbb{E}_s \left[\frac{\partial Q(s, a | \theta^Q)}{\partial a} \frac{\partial \pi(s | \theta^\mu)}{\partial \theta^\mu} \right]. \quad (3.12)$$

沿着提升 Q 值的方向更新策略网络的参数。

通过 DQN 中更新网络的方式来更新评论家网络, 梯度信息为:

$$\frac{\partial L(\theta^Q)}{\partial \theta^Q} = \mathbb{E}_{s, a, r, s' \sim \mathcal{D}} [(r + \gamma Q'(s', \pi(s' | \theta^{\mu'}) | \theta^{Q'}) - Q(s, a | \theta^Q)) \frac{\partial Q(s, a | \theta^Q)}{\partial \theta^Q}] \quad (3.13)$$

如算法2所示。

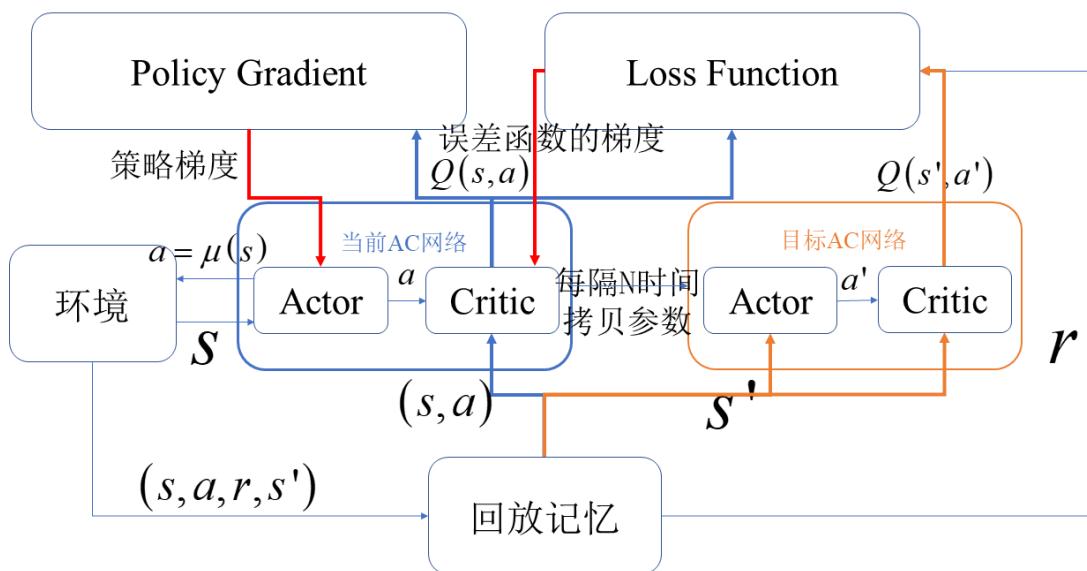


图 3.5: DDPG 算法流程图

Algorithm 2 Deep Deterministic Policy Gradient

- 1: 以随机参数 θ^Q 和 θ^μ 初始化评论家网络 $Q(s, a|\theta^Q)$ 和演员策略 $\mu(s|\theta^\mu)$
- 2: 初始化容量为 N 的回访记忆 \mathcal{D}
- 3: **for** episode = 1, M **do**
- 4: 初始化状态 s_1
- 5: 初始化一个计算过程 \mathcal{N}
- 6: **for** $t = 1, T$ **do**
- 7: 选择动作 $a_t = \mu(s_t|\theta^\mu) + \mathcal{N}_t$
- 8: 执行动作 a_t 得到奖励 r_t 并观察到环境 s_{t+1}
- 9: 保存 (s_t, a_t, r_t, s_{t+1}) 到回放记忆 \mathcal{D}
- 10: 从回放记忆中采样一个 minibatch(s_t, a_t, r_t, s_{t+1})
- 11: 计算 $y_j = \begin{cases} r_j, & s_j \text{ 为终止状态} \\ r_j + \gamma Q(s_{j+1}, \mu'(s_{j+1}|\theta^{\mu'}); \theta^{Q'}) & s_{j+1} \text{ 不为终止状态.} \end{cases}$
- 12: 减小损失以更新评论家: $L = \frac{1}{N} \sum_j ((y_j - Q(s_j, a_j; \theta^Q))^2)$
- 13: 使 用 策 略 梯 度 以 更 新 演 员: $\nabla_{\theta^\mu} J \approx \frac{1}{N} \sum_i \nabla_a Q(s, a|\theta^Q)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu} \mu(s|\theta^\mu)$
- 14: **end for**
- 15: **end for**

DDPG 不仅在一系列连续动作空间的任务中表现稳定, 而且求得最优解所需要的时间步也远远少于 DQN。与基于值函数的深度强化学习方法相比, 基于演员 - 评论家框架的深度策略梯度方法优化策略效率更高, 求解速度更快。

3.3.3 多智能体异步 DDPG

与虚拟环境相比, 在真实环境中, 深度强化学习在轮式机器人上的训练面临以下挑战:

- 在虚拟环境下训练深度强化学习时, 可以通过读取一帧状态后暂停模拟器, 训练网络后再启动模拟器执行动作。而在真实环境下的轮式机器人控制是一个要求强实时性的问题。当读取一帧状态还未选择动作时, 状态可能已经发生了变换, 这就不满足马尔可夫决策过程。
- 在虚拟环境下可以对环境变化的模拟进行加速, 而真实环境下的轮式机器人不可能做到, 这就造成了采样速度非常缓慢。
- 在虚拟环境下可以不用考虑轮式机器人的安全问题, 在真实环境下则需要对轮式机器人的动作空间做出一系列的限制。

频繁的使轮式机器人暂停以排除网络训练时延是不现实的:

1. 首先, 这对于轮式机器人的控制电路、电机与制动装置造成大量的损耗。

2. 其次，若想在训练时延期间，保持轮式机器人状态不变，则同时需要关停敌我双方所有轮式机器人，则需要建立敌我双方的通信机制。这在挑战赛规则中是不允许的，同时也丧失了决策系统的存在意义。
3. 最后，频繁的暂停一定会拖慢采集训练样本的速度。

因此我们采取多智能体异步训练，即多个轮式机器人作为收集线程负责收集样本，一个训练线程负责训练网络，如算法3所示。在收集样本运行的每一个 episode 中，初始化时从训练线程同步策略网络，每一步结束后，将状态、动作、奖励等存入共享的回放记忆中。这样使得在真实环境中的轮式机器人能够不受网络训练时延的困扰，同时成倍的加快训练数据采样的速度。除此之外，通过设置不同的动作参数，如不同贪心策略系数 ϵ 和随机噪声 \mathcal{N} 可以使模型更具稳健性。

Algorithm 3 多智能体异步 DDPG

```

1: // 训练线程
2: 以随机参数  $\theta^Q$  和  $\theta^\mu$  初始化评论家网络  $Q(s, a|\theta^Q)$  和演员策略  $\mu(s|\theta^\mu)$ 
3: 初始化容量为  $N$  的回访记忆  $\mathcal{D}$ 
4: for iteration = 1,  $I$  do
5:   从回放记忆中采样一个 minibatch( $s_t, a_t, r_t, s_{t+1}$ )
6:   计算  $y_j = \begin{cases} r_j, & s_j \text{ 为终止状态} \\ r_j + \gamma Q(s_{j+1}, \mu'(s_{j+1}|\theta^{\mu'}); \theta^{Q'}), & s_{j+1} \text{ 不为终止状态.} \end{cases}$ 
7:   减小损失以更新评论家:  $L = \frac{1}{N} \sum_j ((y_j - Q(s_j, a_j; \theta^Q))^2)$ 
8:   使用策略梯度以更新演员:  $\nabla_{\theta^\mu} J \approx \frac{1}{N} \sum_i \nabla_a Q(s, a|\theta^Q)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu} \mu(s|\theta^\mu)$ 
9: end for
10: // 收集线程  $n, n = 1...N$ 
11: 以随机参数  $\theta_n^\mu$  初始化评演员策略  $\mu(s|\theta_n^\mu)$ 
12: for episode = 1,  $M$  do
13:   同步策略网络权重  $\theta_n^\mu \leftarrow \theta_m u$ 
14:   初始化状态  $s_1$ 
15:   初始化一个计算过程  $\mathcal{N}$ 
16:   for  $t = 1, T$  do
17:     选择动作  $a_t = \mu(s_t|\theta_n^\mu) + \mathcal{N}_t$ 
18:     执行动作  $a_t$  得到奖励  $r_t$  并观察到环境  $s_{t+1}$ 
19:     保存  $(s_t, a_t, r_t, s_{t+1})$  到回放记忆  $\mathcal{D}$ 
20:   end for
21: end for

```

第四章 实验数据采集与分析

4.1 仿真平台

4.1.1 ROS 简介

ROS(Robot Operating System, 机器人操作系统)是为机器人软件开发的一种计算机操作系统架构，源自于斯坦福AI机器人项目。ROS能够为跨平台、跨语言的异构计算机集群提供结构化通信。ROS包括标准的操作系统环境，包括底层设备控制与管理、进程间消息传递机制和软件包管理等功能。ROS操作系统能够使开发者非常方便的构建多语言的机器人软件功能。如2.2.1所述，我们在ROS系统的基础上构建驱动、感知、规划、控制和决策功能模组。

4.1.2 RVIZ 与 Gazebo

RVIZ是ROS自带的图形化工具。在本文中，我们将轮式机器人的雷达、里程计等传感器数据以及定位、路径规划等感知和规划功能模组中间结果在RVIZ界面中进行可视化，如图4.1所示。Gazebo是基于ROS的一款功能强大的3D机器人仿真模拟器。我们使用Gazebo进行仿真建模，准确有效地模拟复现轮式机器人在挑战赛场地内的状态和行为，如图4.2所示。Gazebo通过高性能的物理引擎，对轮式机器人进行物理建模，动力学仿真，模拟真实环境下的传感器和噪声，使我们的强化学习算法获得逼近于真实环境的训练。

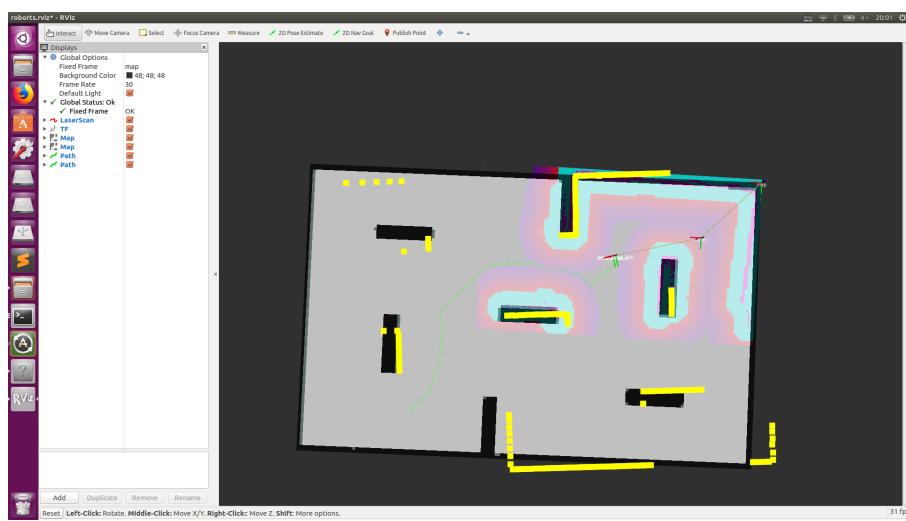


图 4.1: RVIZ 界面

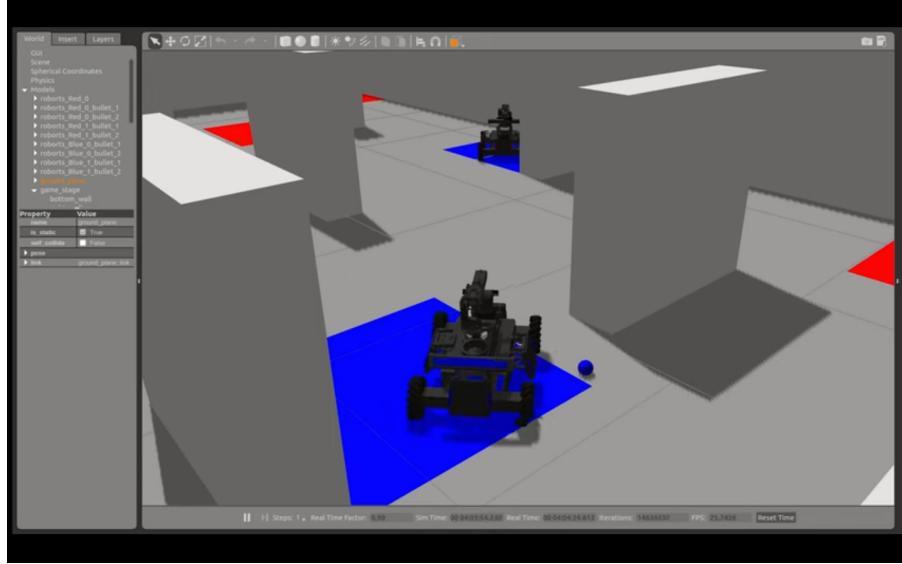


图 4.2: Gazebo 物理仿真

4.2 真实环境

我们在真实环境中使用的轮式机器人如图4.3所示。在真实环境中的深度强化学习训练具有比在虚拟环境下训练更具挑战性的问题：

1. 必须选择适当的策略或者值函数表示，以便实现对物理硬件实用的训练时间。
2. 必须提供实例演示来初始化策略并减轻训练期间的安全问题。
3. 必须考虑到轮式机器人在真实世界的感知、通信和控制时延。

针对安全问题，我们在进行真实环境下的实验时，进行一下设置以保证训练过程安全有效：

1. 训练时，靶车即敌方机器人由人类操作员操作，以保证轮式机器人与场地人员安全。进行测试时，靶车由如图2.8所示的确定性有限状态机控制作为实验基线。
2. 训练时，轮式机器人在硬件上设置安全阈值和异常中断阈值，如表??。核查控制代码，使其控制论机器人的过程中，各安全项目的理论值在安全阈值以内。创建一个 ROS 结点，轮询安全项目数值，如果有超过异常中断阈值，则记录异常数值并终止轮式机器人运动。
3. 训练时，使用远程桌面 VNC 控制轮式机器人。除操作员以外，无关人员禁止进入训练场地。

针对时延问题，我们使用多智能体异步训练方法以求轮式机器人智能体能够克服物理时延带来的问题。



图 4.3: 轮式机器人实物

4.3 奖励函数定义

通过使用 OpenAI gym 标准的接口，实现 ROS 各个功能模组与决策系统的交互。通过串口向上位机发送轮式机器人的基本信息，轮式机器人智能体能够获得如比赛时间 t 、血量 h 、功率、里程计、速度、发射机构状态和遭受伤害的方向 d 等状态信息。通过传感器、定位算法 (AMCL)，轮式机器人智能体能够感知到自身位置与底盘偏航角 (x, y, φ) 。通过目标识别算法 (SSD-mobilenet)，轮式机器人可以感知到敌人的位置 (x_e, y_e) 。

轮式机器人智能体所做出的动作包含下一步的目标位置与底盘偏航角 (x', y', φ') ，或者相应的速度 (v_x, v_y, ω) ，或加速度 (a_x, a_y, b) ，和发射机构云台的开火角度 (θ, ψ) 。即云台俯仰角 pitch、云台偏航角 yaw。

考虑到与现有通信协议与控制算法整合，云台发射机构交由自动打击系统单独控制，打击策略为发现目标即开火，同时矫正云台姿态，智能体负责底盘运动的决策。最终定义状态 s 和动作 a :

$$s = (x, y, \varphi, x_e, y_e, h, t, d), \quad (4.1)$$

$$a = (x', y', \varphi'). \quad (4.2)$$

奖励函数为：

$$r = \begin{cases} -1, & \text{每受到一点伤害} \\ 1, & \text{敌方轮式机器人出现在视野内} \\ 10, & \text{获得增益 buff.} \end{cases} \quad (4.3)$$

4.4 代码设计

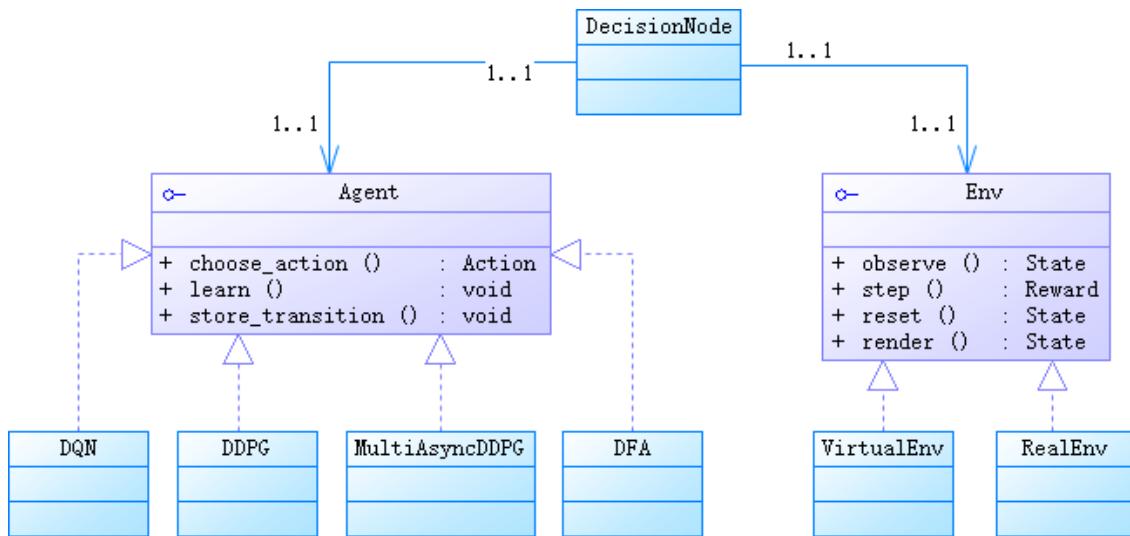


图 4.4: 轮式机器人决策系统类图

智能体类 agent 与环境类 env 结构与接口，如图4.4所示。我们按照 OpenAI gym 接口规范标准，将智能体和与智能体直接交互的环境封装为接口。

环境类 env 隐藏了与轮式机器人感知、控制和通信等细节，从而使智能体类 agent 只需要关心与环境类 env 的交互，即接收一个状态 s ，做出相应的动作决策 a 。同时，智能体类 agent 隐藏了自主决策算法的训练和执行细节，环境类只需要关心执行智能体类输出的动作决策 a 。这样的结构允许我们在仿真环境到真实环境下的平滑过渡，同时也支持不同网络结构的深度强化学习算法的切换，甚至是使用非强化学习算法，如确定性有限状态机或遗传算法等启发式搜索算法。

针对深度强化学习，对于每一个情节 (episode)，我们设置最大步长数。训练时，轮式机器人智能体达到步长后即终止本次情节，重置环境，进入下一个情节的训练。训练过程如图4.5所示。

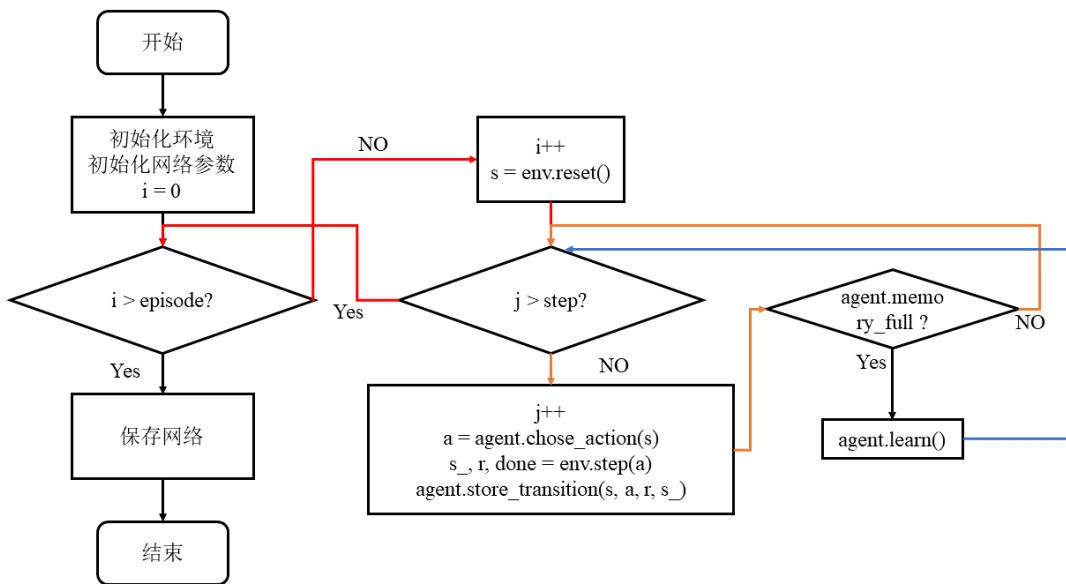


图 4.5: 训练过程流程图

4.4.1 Deep Q-learning

使用 Tensorflow 构建 DQN 网络，网络结构如图4.6所示。其中当前网络和目标网络均为三层全连接层神经网络，每层的神经元数分别为 64, 128, 256，隐藏层的激活函数为 Relu，输出层的激活函数为 Tanh。

4.4.2 Deep Deterministic Policy Gradient

使用 Tensorflow 构建 DDPG 网络，网络结构如图4.7所示。其中当前网络和目标网络同构。网络中演员为三层全连接层神经网络，每层的神经元数分别为 64, 128, 256，隐藏层的激活函数为 Relu，输出层的激活函数为 Tanh；

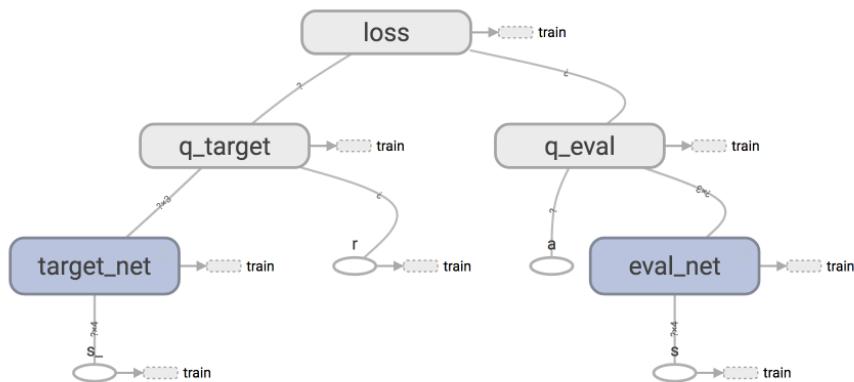


图 4.6: DQN 网络结构图

评论家网络为三层全连接神经网络，每层的神经元数分别为 64, 128, 1，输出标量评价值，即 Q 值的估计。

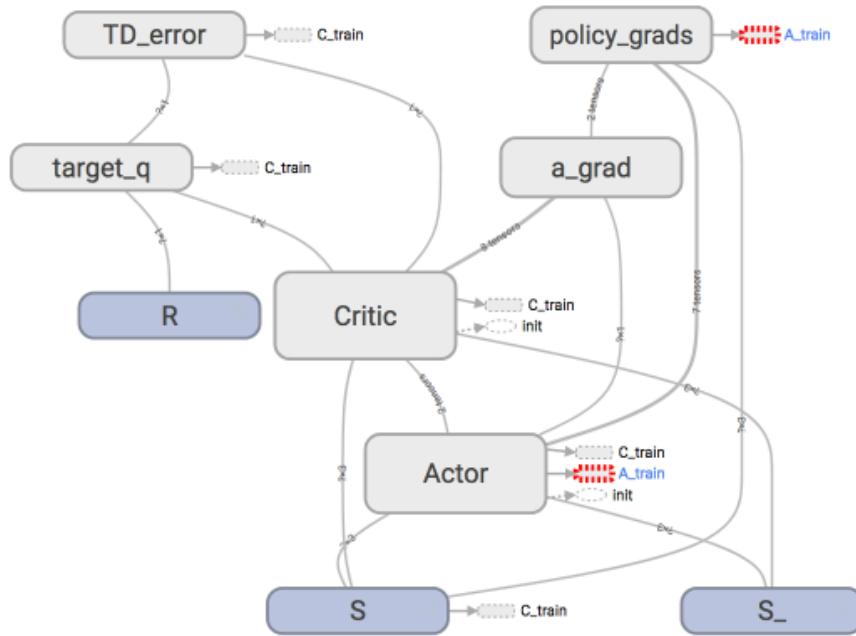


图 4.7: DDPG 网络结构图

4.4.3 多智能体异步 DDPG

使用 Tensorflow 构建多智能体异步 DDPG，其网络结构图与 DDPG 保持一致。在 Gazebo 仿真环境中，一次使用四台轮式机器人智能体进行对抗训练，如图4.8所示。

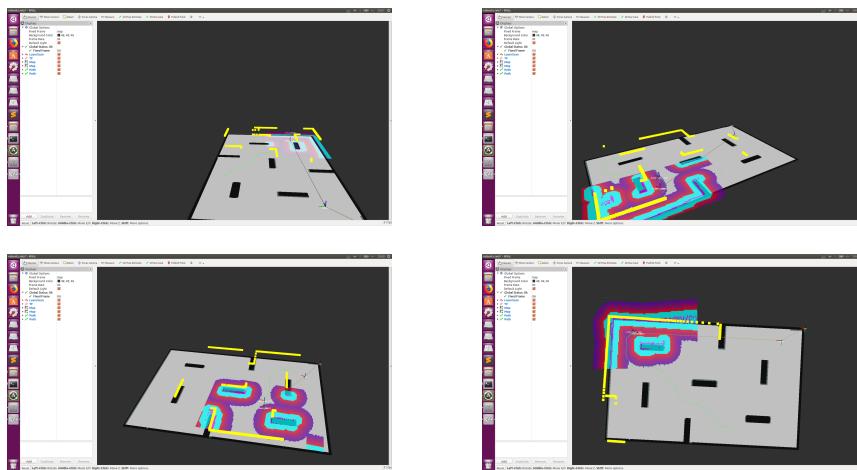


图 4.8: 多智能体异步 DDPG 训练

4.5 实验结果

4.5.1 仿真实验结果

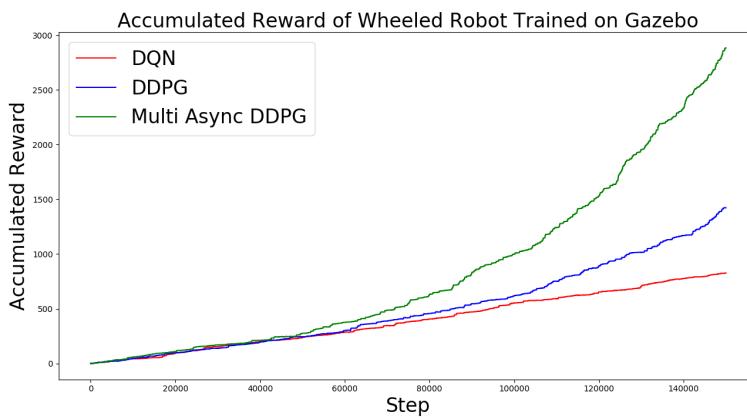
在仿真环境下，我们使用上述定义的参数，应用 DQN，DDPG，多智能体异步 DDPG，在 Gazebo 仿真平台上训练全自主轮式机器人。使用的超参数有：

- 情节数，episode: 500。
- 最长步数，step: 300。
- 折扣参数， γ : 0.90。
- 学习率，learning rate: 1e-3。
- 回放记忆长度，reply memory: 30000。
- 批处理长度，batch size: 50。

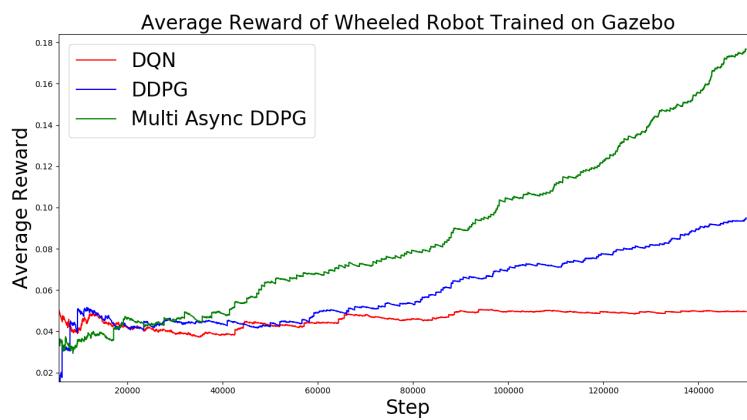
应用三种不同算法在仿真环境下的表现如图4.9所示。其中图4.9(a)表示仿真环境下轮式机器人在训练中共获得的奖励累加值，图4.9(b)表示仿真环境下轮式机器人的步长平均奖励，即奖励累加值除以当前进行过的步长，可以是作为奖励期望的估计。由奖励设置式4.3 可推知，步长平均奖励近似于敌方出现在轮式机器人视野的概率，我们可将之称为锁定率。在图中我们可以看出，在 DQN、DDPG 以及多智能体异步 DDPG 三种深度强化学习算法中，多智能体异步 DDPG 算法收敛速度较快，其模型锁定率能够达到 18% 以上，而 DDPG 和 DQN 分别收敛在 9% 和 4% 左右。

4.5.2 真实环境下对抗结果

在真实环境下，我们使用确定性有限状态机作为基线与三种不同算法训练的轮式机器人智能体各进行 8 次实际对战，在 120s 的比赛时间内的比赛结果如图4.10所示。图中黑色直线为失败线，分布在黑色直线上的点即为血量被敌方轮式机器人伤害殆尽而结束比赛；棕色线为胜利线，分布在棕色线上的点即为在保佑一定血量的情况下使敌方轮式机器人血量消耗殆尽；橙色线为优势线，即在比赛时间结束时，敌我双方都没有击败对方，此时处于优势线右侧的点表示在比赛结束时，我方轮式机器人智能体保有血量高于敌方。由图中结果可以发现多智能体异步 DDPG 算法和 DDPG 算法在真实环境下与敌方对抗时各有三次取得优势，实际上的表现相近，而 DQN 则很难取得优势。这说明了多智能体异步 DDPG 算法实际上是 DDPG 算法的快速收敛版。通过多智能体异步的方法部分克服了时延问题和采样速度较慢的问题。



(a) 仿真环境下轮式机器人的累积奖励



(b) 仿真环境下轮式机器人的步长平均奖励

图 4.9: 仿真环境下轮式机器人的训练表现

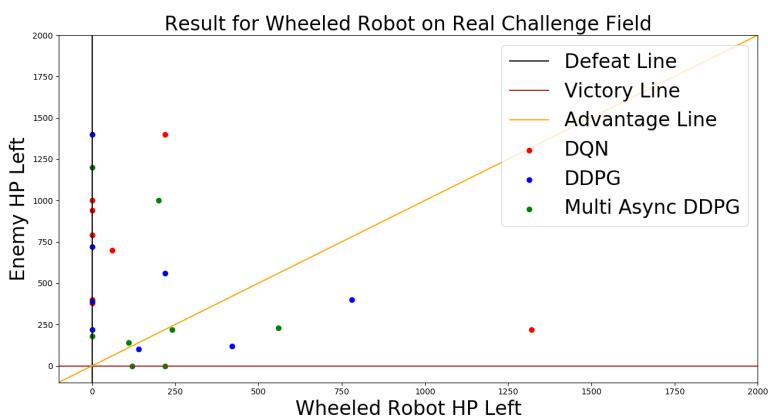


图 4.10: 真实环境下轮式机器人对抗结果

第五章 全文总结

5.1 全文总结

深度强化学习已经准备好在 AI 领域掀起一场革命。深度强化学习向建立对于真实世界具有高水平理解能力的全自主系统迈出了坚实的一步。当前，深度学习使强化学习有能力处理先前极为棘手的问题。本文通过应用深度强化学习在轮式机器人上，以实现轮式机器人的全自主战斗决策。主要做的工作如下：

1. 基于 ROS，构建了轮式机器人的决策系统。通过将智能体 agent 部分与环境 env 部分解耦，实现了决策系统的模块化。
2. 简述了强化学习的基本原理，介绍了基于值函数与策略搜索的强化学习基本算法。
3. 基于 Gazebo，搭建了轮式机器人仿真平台。
4. 提出了多智能体异步的训练方法，解决了真实环境下的时延问题和采样速度问题。
5. 实现并检验了 DQN、DDPG 和多智能体异步 DDPG 在仿真环境和真实环境下的训练和表现。

5.2 对未来工作的展望

根据本文的分析，可以发现深度强化学习已经广泛应用于机器人控制任务，从简单运动到复杂操作和全自主移动机器人控制。然而，强化学习在真实世界中的实际应用通常需要超出学习算法本身的大量额外工程：

1. 机器人无法感知到真实世界的全貌，即机器人只能部分感知当前状态。
2. 真实世界往往是奖励稀疏的，如何确定奖励函数是一个棘手的问题。
3. 机器人在真实环境下存在着感知、通信和控制时延，这还会使得机器人无法。

在未来的工作中，应该着重与解决这些问题。建立基于部分部分可观测马尔可夫决策过程的模型和算法；应该设计基于状态 - 动作对概率分布的奖励函数，使连续的状态 - 动作空间都能收到奖励或者惩罚信号；使用并发和分布式的方法，克服机器人在真实世界中训练的时延。

参考文献

- [1] 徐扬生. 智能机器人引领高新技术发展 [J]. 企业科协, 2010 (9): 28-31.
- [2] Christensen H I, Batzinger T, Bekris K, et al. A roadmap for us robotics: from internet to robotics[J]. Computing Community Consortium, 2009, 44.
- [3] Salichs M A, Moreno L. Navigation of mobile robots: open questions[J]. Robotica, 2000, 18(3): 227-234.
- [4] Beom H R, Cho H S. A sensor-based navigation for a mobile robot using fuzzy logic and reinforcement learning[J]. IEEE transactions on Systems, Man, and Cybernetics, 1995, 25(3): 464-477.
- [5] Pandey A, Sonkar R K, Pandey K K, et al. Path planning navigation of mobile robot with obstacles avoidance using fuzzy logic controller[C]//2014 IEEE 8th International Conference on Intelligent Systems and Control (ISCO). IEEE, 2014: 39-41.
- [6] Wang C, Soh Y C, Wang H, et al. A hierarchical genetic algorithm for path planning in a static environment with obstacles[C]//IEEE CCECE2002. Canadian Conference on Electrical and Computer Engineering. Conference Proceedings (Cat. No. 02CH37373). IEEE, 2002, 3: 1652-1657.
- [7] Pandey A, Sonkar R K, Pandey K K, et al. Path planning navigation of mobile robot with obstacles avoidance using fuzzy logic controller[C]//2014 IEEE 8th International Conference on Intelligent Systems and Control (ISCO). IEEE, 2014: 39-41.
- [8] Sutton R S, Barto A G. Introduction to reinforcement learning[M]. Cambridge: MIT press, 1998.
- [9] LeCun Y, Yoshua Bengio, and Geoffrey Hinton[J]. Deep learning. nature, 2015, 521(7553): 436-444.
- [10] Bengio Y, Courville A, Vincent P. Representation learning: A review and new perspectives[J]. IEEE transactions on pattern analysis and machine intelligence, 2013, 35(8): 1798-1828.
- [11] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540): 529.
- [12] Silver D, Huang A, Maddison C J, et al. Mastering the game of Go with deep neural networks and tree search[J]. nature, 2016, 529(7587): 484.
- [13] Levine S, Finn C, Darrell T, et al. End-to-end training of deep visuomotor policies[J]. The Journal of Machine Learning Research, 2016, 17(1): 1334-1373.
- [14] Levine S, Pastor P, Krizhevsky A, et al. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection[J]. The International Journal of Robotics Research, 2018, 37(4-5): 421-436.
- [15] Duan Y, Schulman J, Chen X, et al. RL²: Fast Reinforcement Learning via Slow Reinforcement Learning[J]. arXiv preprint arXiv:1611.02779, 2016.
- [16] Wang J X, Kurth-Nelson Z, Kumaran D, et al. Prefrontal cortex as a meta-reinforcement learning system[J]. Nature neuroscience, 2018, 21(6): 860.
- [17] Vinyals O, Ewalds T, Bartunov S, et al. Starcraft ii: A new challenge for reinforcement learning[J]. arXiv preprint arXiv:1708.04782, 2017.

- [18] Kaelbling L P, Littman M L, Cassandra A R. Planning and acting in partially observable stochastic domains[J]. Artificial intelligence, 1998, 101(1-2): 99-134.
- [19] Watkins C J C H, Dayan P. Q-learning[J]. Machine learning, 1992, 8(3-4): 279-292.
- [20] Rummery G A, Niranjan M. On-line Q-learning using connectionist systems[M]. Cambridge, England: University of Cambridge, Department of Engineering, 1994.
- [21] Koutník J, Cuccu G, Schmidhuber J, et al. Evolving large-scale neural networks for vision-based reinforcement learning[C]//Proceedings of the 15th annual conference on Genetic and evolutionary computation. ACM, 2013: 1061-1068.
- [22] Deisenroth M P, Neumann G, Peters J. A survey on policy search for robotics[J]. Foundations and Trends® in Robotics, 2013, 2(1–2): 1-142.
- [23] Williams R J. Simple statistical gradient-following algorithms for connectionist reinforcement learning[J]. Machine learning, 1992, 8(3-4): 229-256.
- [24] Konda V R, Tsitsiklis J N. On actor-critic algorithms[J]. SIAM journal on Control and Optimization, 2003, 42(4): 1143-1166.
- [25] Schulman J, Moritz P, Levine S, et al. High-dimensional continuous control using generalized advantage estimation[J]. arXiv preprint arXiv:1506.02438, 2015.
- [26] Lillicrap T P, Hunt J J, Pritzel A, et al. Continuous control with deep reinforcement learning[J]. arXiv preprint arXiv:1509.02971, 2015.
- [27] Gu S, Holly E, Lillicrap T, et al. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates[C]//2017 IEEE international conference on robotics and automation (ICRA). IEEE, 2017: 3389-3396.
- [28] Van Hasselt H, Guez A, Silver D. Deep reinforcement learning with double q-learning[C]//Thirtieth AAAI Conference on Artificial Intelligence. 2016.
- [29] Wang Z, Schaul T, Hessel M, et al. Dueling network architectures for deep reinforcement learning[J]. arXiv preprint arXiv:1511.06581, 2015.
- [30] Schaul T, Quan J, Antonoglou I, et al. Prioritized experience replay[J]. arXiv preprint arXiv:1511.05952, 2015.

致 谢

首先要感谢我的毕业设计指导老师史豪斌副教授。论文是在史老师的悉心指导下完成的。在论文的进展中，史老师提供了实验平台和学习资料，在学业和论文写作上提出了许多宝贵意见。史老师严谨的治学态度以及丰富的实践经验将是我以后学习和工作的动力和楷模。在此谨向我的指导老师史豪斌副教授表示衷心的感谢。

同时要感谢西北工业大学竞技机器人基地的同学和队友们，是我们一起共同的努力，让我们能在 ICRA 的平台上进行机器人与人工智能的竞技和挑战。

毕业设计小结

这次毕业设计是对我四年本科学习的一个总结，涉及了机器人感知、控制与通信，深度神经网络与深度强化学习，这对我来说是一个全面的考验。虽然在过去的比赛中经常接触和使用仿真环境 Gazebo 以及机器人操作系统 ROS，但是对它们的理解并不算透彻。在这次毕业设计的过程中，我又系统地学习了 ROS，深刻理解了 ROS 消息机制。对于深度神经网络方面，我在这次的毕业设计实践中感受数学工具在开辟新算法和思路时的重要性以及扎实的打好数学基础对于计算机相关专业学习的重要性。

通过本次毕业设计，我对深度强化学习在轮式机器人上的应用有了更深入的了解，对今后研究生阶段的学习和奋斗的目标也更加明确。