

清 华 大 学

综 合 论 文 训 练

题目：通用小型无人机地面控制站的
设计与实现

系 别：计算机科学与技术系

专 业：计算机科学与技术

姓 名：王 澎

指导教师：朱纪洪 教授

2018 年 6 月 25 日

关于学位论文使用授权的说明

本人完全了解清华大学有关保留、使用学位论文的规定，即：学校有权保留学位论文的复印件，允许该论文被查阅和借阅；学校可以公布该论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存该论文。

(涉密的学位论文在解密后应遵守此规定)

签 名： 王 澎 导师签名： 朱 纪 洪 日 期： 2018/06/25

中文摘要

无人机地面控制站作为无人机系统中的控制中枢与决策终端在整个无人机系统中的作用非常重要。目前一些广泛使用的与各型飞控配套的产品级地面站软件虽然种类较多且功能丰富，但是各自所适配的飞控硬件较为单一且源码结构较为复杂。通过设计并实现一款通用的小型无人机地面控制站可以满足实验室中希望在多型飞控架构上控制无人机实现特定任务及算法验证的需求。本项工作通过采用跨平台框架 Qt 搭配无线网络通信及部分开放的地图与 GIS，实现了一款多功能模块化的通用地面控制站。本项工作主要内容为：

1. 设计实现无人机地面控制站协议层。采用 UDP 网络通信协议，实现地面控制站的控制指令上行至无人机以及无人机的状态数据信息下行至地面控制站。
2. 设计实现无人机地面控制站数据层。在 Qt 窗体中嵌入网页形式的地图，实现无人机位置显示。使用 SQLite 数据库，保存无人机发送至地面站的数据。
3. 设计实现无人机地面控制站应用层。搭配地面站构建无人机仿真代理，与地面站进行控制指令与数据的交互。实现操纵台、虚拟仪表以及航迹显示功能。
4. 设计实现无人机航线规划相关算法。在调研了相关研究工作的基础上，实现了部分基础的无人机航线规划算法，并对算法的性能指标进行了对比评价。

关键词：无人机；地面控制站；Qt；航线规划

ABSTRACT

The UAV Ground Control Station (GCS), which takes the role as the control center and decision terminal, has long been one of the most significant part in the drone system. Although plenty of widely used product-level ground station softwares which are designed for various kinds of flight controller are of rich functions today, the choice of available flight controller for each kind of them is quite simple and their source codes are generally complicated. By designing and implementing a universal light UAV ground control station, it is possible to meet the needs of the laboratory for controlling specific tasks and verifying algorithms on a multi-type flight control architecture. In this work, a multi-functional, modular, universal ground control station was implemented using a cross-platform framework Qt with wireless network communications and various open maps and GIS with a main content of:

1. Design and implement the protocol layer of UAV Ground Control Station. By using the UDP network communication protocol, control commands from the ground control station are transmitted upstream to the drone and the state data of the drone is downlinked to the ground control station.
2. Design and implement the data layer of UAV Ground Control Station. In the Qt form embedded in the form of web pages in order to displaying drone position. Use the SQLite database to save the data that the drone sends to the ground station.
3. Design and implement the application layer of UAV Ground Control Station. Collaborate with the ground station to construct a drone simulation agent in order to interacting with the ground station by control instructions and data. Implement consoles, virtual meters, and track display capabilities.
4. Design and implement algorithms of UAV path planning. Based on the research of related works, some basic UAV path planning algorithms were implemented, and the performance indicators of the algorithm were compared and evaluated.

Keyword: UAV; ground control station; Qt; path plan

目录

第 1 章 引言	1
1.1 研究背景与意义	1
1.2 国内外相关研究及发展现状	2
1.3 本文研究的主要内容	4
1.3.1 无人机地面站的组成与功能	4
1.3.2 本文结构安排	6
第 2 章 无人机地面控制站协议层	7
2.1 无人机地面控制站通信功能设计	7
2.2 基于 UDP 的通信协议设计	8
2.2.1 OSI 架构与 UDP 通信协议	8
2.2.2 UDP 通信协议在本项目中的应用	8
2.3 本章小结	12
第 3 章 无人机地面控制站数据层	13
3.1 无人机地面控制站地图导航功能设计	13
3.1.1 百度地图开发介绍及控件特点	13
3.1.2 百度地图的加载	14
3.1.3 百度地图中引入功能性控件	15
3.1.4 Qt 中开发 Web 应用的方式	17
3.2 无人机地面控制站数据库模块	20
3.3 本章小结	22
第 4 章 无人机地面控制站应用层	23
4.1 无人机状态仿真及无人机代理	23
4.1.1 无人机状态仿真及所需参数	23
4.1.2 无人机位置仿真与坐标计算	26
4.1.3 无人机代理参数设计与实现	29
4.2 无人机地面控制站操纵台	30
4.3 无人机地面控制站虚拟仪表	31

4.4 无人机地面控制站航迹显示	32
4.5 本章小结	32
第 5 章 无人机航线规划	34
5.1 无人机航线规划总体介绍	34
5.2 相关研究工作	35
5.3 本项目采用的航线规划方法	38
5.3.1 A*算法.....	38
5.3.2 人工势场	39
5.3.3 遗传算法	40
5.3.4 PRM.....	41
5.3.5 RRT.....	42
5.4 实验结果分析	43
5.5 本章小结	45
第 6 章 总结与展望	47
插图索引	48
表格索引	50
参考文献	51
致 谢	56
声 明	57
附录 A 外文资料的书面翻译.....	58

第1章 引言

1.1 研究背景与意义

无人机作为一个代替人在空中执行某些特定任务的“飞行物”，最早出现在 19 世纪末期，被道格拉斯·阿尔德里奇和威廉·艾迪等人用来在高空中拍摄照片，从而在战争环境中被用作一种最为初级的侦查手段。随着其在军事领域中的需求日益增长，无人机系统在此后的一系列战争诸如越南战争，阿富汗战争中发挥了越来越重要的作用^[1]。

除了在军事活动中无人机可以起到侦察敌方活动、尾随敌方目标、诱骗敌方攻击武器以及凭借自身所负载武器执行一些攻击任务之外，随着无人机系统的逐步发展，其在民用领域也得到许多应用。目前无人机在农作物监控、牧群监控、渔业保护、电力传输设备检查、森林消防、地图绘制等方面应用较多^[2]。然而上述所描述的应用场景关注点集中于对传统的工农业实现人机互补。本文将关注点转移到新型服务行业中，考虑在城市密集环境下的无人机多任务点路径规划与自主飞行，从而可以为“无人超市-无人机外卖”这样的无人电商模式给出一些较为基本的技术验证与技术支持。除此之外，考虑城市密集环境下的无人机多任务点路径规划与自主飞行在军事领域中同样具有应用价值，如利用无人机的小型优势在城市环境中实现侦察任务等。

一个完整的无人机系统应当包括控制站（Control Station）、无人飞机本体、无人飞机所携带的任务载荷、无人飞机所携带的导航系统以及控制站与无人飞机或无人飞机之间的通信链路等。其中控制站在整个无人机系统中最为重要。作为整个无人机系统中的控制中枢与决策终端，控制站通常承担的任务包括但不限于：从发射到回收的全过程控制、状态监控、确保通信链路的正常工作、飞行路径规划以及任务的分配与完成等。

为了对于上述问题进行较为深入的研究，本项目首先在参考国内外一些开源无人机地面控制站的设计的基础上，采用跨平台框架设计实现了一款通用小型无人机地面控制站，一方面用于项目第二部分中的城市密集环境中的多任务点路径规划的研究，另一方面为今后逐步开发出一个模块化的多功能无人机地面控制站提供基础。

1.2 国内外相关研究及发展现状

目前无人机地面控制站在国内外都有一定程度的研究。

对于无人机在军事上的应用，通常为一个整体的部署完整的舱室，图 1.1 展示了美国通用原子公司为“捕食者”无人机设计实现的下一代地面站 Advanced Cockpit GCS^[3]。



图 1.1 美国通用原子公司 Advanced Cockpit GCS

而对于民用地面站，通常情况下部署于 PC 端或手机移动终端。图 1.2 展示了四款目前使用较为广泛了无人机地面站软件。按照从上到下从左到右的顺序依次是开源框架 QGroundControl、MissionPlanner 以及软件 Horizon 与 UgCS。可以看到上述无人机地面站的界面元素基本相同，除了有地图或者 GIS，还包含一些无人机当前的数据信息，如电量，GPS 信号，飞行高度，对地飞行速度，飞行方向，航迹等。并且可以通过操作界面实现对无人机飞行模式、航路点、滚转角、偏航角和俯仰角的控制。



图 1.2 几款常用的产品级无人机地面站软件

上述地面站系统通常是对于特定的飞控仪进行设计的，如 QGroundControl 是针对 Pixhawk 飞控设计的；MissionPlanner 是针对 APM 飞控开发的；Horizon 是针对 MP2028 自动驾驶仪研制的。且为了满足多种不同用户的需求，在功能实现上较为丰富导致源码结构较为复杂。在实验室环境下，一方面所使用的无人机使用的飞控具有不确定性，另一方面所需要的功能更为专一，故而亦有较多研究关注构建小型的通用无人机地面站的方法。

文献^[2, 4]从整体架构上分析了无人机地面站应当包含的模块以及在开发通用型地面站是应当遵守的以接口标准为前提的自顶向下设计思想。文献^[5-7]从整体上介绍了开发地面站系统的过程。虽然实现有所不同，但是普遍关注通过串口或无线网络建立通信，通过 Access 等数据库保护数据，通过 Google Earth 等地图或 GIS 实现可视化。在采用的编程框架的选择中，C#中的 WPF 技术^[8, 9]与跨平台的 Qt 框架^[10, 11]较为常用。除此之外，部分工作将关注点聚焦于地面站软件与操作手之间的人机交互上。通过利用 FlightGear 飞行模拟器^[12]、Direct3D 三维建模^[13]、虚拟现实技术^[14]以及手势控制等。特别的，在刚刚结束的 ICRA 2018 的会议中，其中有一场题目为 Human-Robot-Interaction with UAVs: Challenges and Frontiers 的 workshop 专门讨论无人机的人机交互问题。其中 senseFly 公司介绍了他们的全 3D 显示的地面站 eMotion，DJI 大疆介绍了其最新的支持使用者通过多种手势操

作无人机的 FlightAutonomy2.0 系统。可以预见到，未来人机交互将成为无人机地面站与无人机之间通信与控制的新的研究方向。

1.3 本文研究的主要内容

1.3.1 无人机地面站的组成与功能

无人机地面控制站的基本结构应如图 1.3 所示。

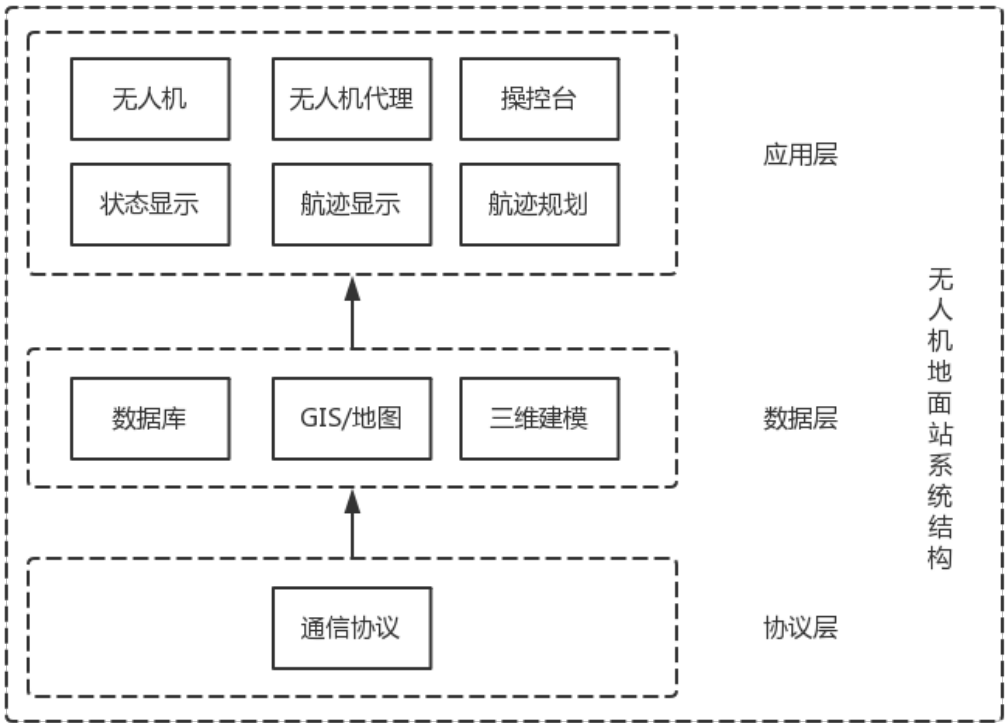


图 1.3 无人机地面站系统结构

可以看到其中这里将无人机地面站的整体结构分成了三个层次——协议层、数据层与传输层。其中最协议层位于最底层，包括无线网络通信与串口通信，为地面站与无人机进行数据连接。位于中间的是数据层，它为整个系统提供所需要的所有数据，包括需要用到的地理信息 GIS 或者地图，通过三维建模软件构建出的三维场景数据信息以及为无人机飞行时向地面站传送回的数据提供存储能力的数据库。数据层需要调用通信协议的方法来与无人机进行通信从而获得无人机

的数据。最上层为应用层。该层利用数据层的数据及接口以及底层协议层的通信方法，实现各种应用，包括但不限于无人机飞控、无人机代理、地面站操控台、状态显示、航迹显示以及航迹规划等。

在明确了无人机地面站的主要功能模块以及之间的关系后，再来整理一下无人机地面站所需要实现的功能。如图 1.4 所示，无人机地面站应当维持通信协议实现通信连接、实现地图导航、保持数据监控与显示、从无人机处获得图片或者视频音频等流媒体数据以及具备实现任务规划的能力。其中任务规划包含很多内容，这里我们最为关心的是航路规划。

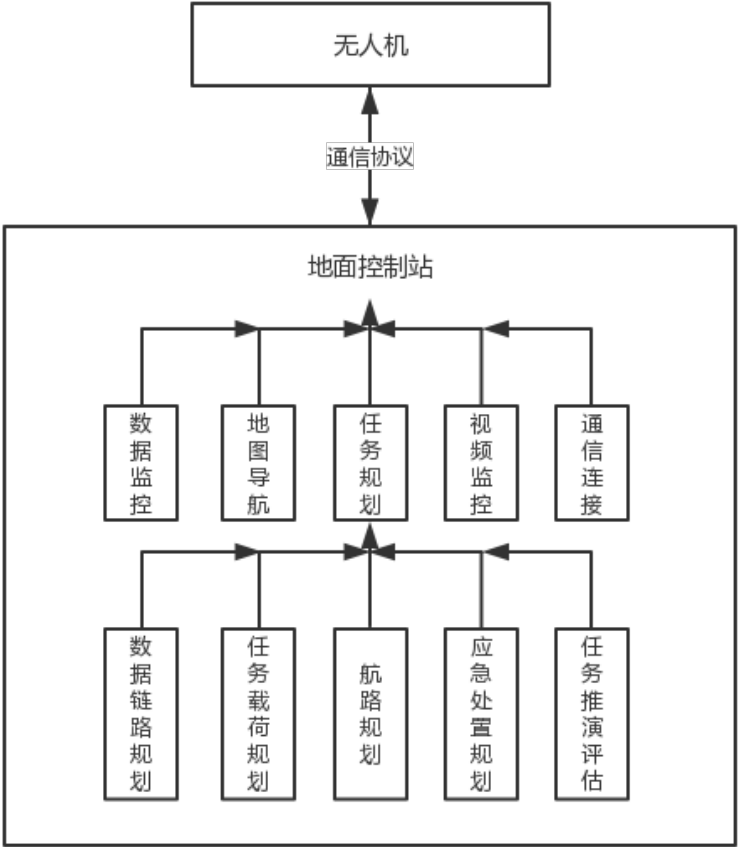


图 1.4 无人机地面站功能结构

基于上述对于无人机地面控制站组成结构以及功能结构的分析，在本项目中对于其中一部分主要内容进行了实现，从而可以获得一个通用小型无人机地面站。

1.3.2 本文结构安排

第一章分析无人机及无人机地面站的发展现状及当前主要研究方向。同时详细分析了地面站系统的组成结构以及应当具备的功能结构。

第二章具体介绍本项目中实现的无人机地面站的协议层。主要包括一个基于UDP的通信协议。

第三章具体介绍本项目中实现的无人机地面站的数据层。主要包括地图模块的加载以及数据库的构建与使用。

第四章具体介绍本项目中实现的无人机地面站的应用层。主要包括无人机代理的实现、位于地面站界面上的操控台、虚拟仪表以及航迹显示。

第五章单独介绍无人机航线规划算法。主要包括航线规划的问题引出、相关研究工作、本项目中所实现的5个基本算法以及它们的性能与评价。

第六章总结项目并讨论进一步开发的方向。

第2章 无人机地面控制站协议层

2.1 无人机地面控制站通信功能设计

无人机地面控制站由于自身所具有的数据接收、数据处理、数据显示、以及控制指令的发送等功能，与远端的一个或者多个无人机实体的数据通信能力是必需也是必要的。一个好的、稳定的通信协议的建立，有助于帮助无人机地面控制站与远端无人机之间进行高效可靠的数据传输。目前无人机地面控制站与远端无人机之间的通信实现远程通信与本地通信两种。远程通信通常需要数据发射站或卫星通信等这里我们不做讨论。本地通信是指无人机地面控制站同无线传输站之间的通信，或者无人机上远程无线通信设备与无人机的载荷之间的通信链路与信道。本地通信一般主要分为两种：一种是基于网络通信的方式，即远端无人机通过部分机载网络通信模块（通常集成在芯片中）从而可以提供 WiFi 功能或 4G 等蜂窝网络功能；另一种是基于串口通信的方式，即远端无人机依靠某些无线传输设备（如数传电台）与本地无人机地面控制站实现串口通信。对于无线网络通信，目前已经有很多应用采用了基于 UDP 的 WiFi^[15]或者 4G^[16]方式进行通信，同时在最新的 5G 技术上亦有技术尝试与验证^[17]。对于串口通信，在一些无人机地面站软件以及飞控系统中同样有所使用^[18-20]。

随着无线网络技术的发展，一些适合于搭载于小型无人机上的嵌入式微处理器对于无线 WiFi 也提供了越来越多的支持。如基于 ARM 的微型电脑主板树莓派 Raspberry Pi 在其最新的 Raspberry Pi 3 Model B 以及 Raspberry Pi 3 Model B+中已经自带了 WiFi 模块，其中 Raspberry Pi 3 Model B+支持双频 2.4GHz 和 5GHz 的 WiFi 和蓝牙 4.2。故而在本项目的设计中选择采用无线网络作为传输方式的实现。这里介绍本项目所采用并实现的一种基于 UDP 的无线通信协议，通过在无人机地面控制站与无人机或无人机代理之间实现这种简单的基于数据报的通信方式，可以完成项目预期的通信需求。事实上目前在较多大型无人机地面站中广泛使用 MAVLink 传输协议。MAVLink 是一种基于串口通信的高层开源通信协议。其可以在智能实体（如无人车、无人机等）与相应的控制站终端之间实现通信。目前在较多大型复杂开源无人机地面控制站系统，如 MissionPlanner 和 QGroundControl 中使用的即是此种通信协议。在进一步的研究工作中，通过研究

实现更为复杂的 MAVLink 协议有助于帮助本项目的无人机地面控制站系统与一些采用该协议进行通信的飞控平台如 APM 以及 Pixhawk 实现对接，从而为系统的模块化设计与可扩展性提供技术基础。

2.2 基于 UDP 的通信协议设计

2.2.1 OSI 架构与 UDP 通信协议

在介绍本项目中采用的基于 UDP 的无人机地面控制与远端无人机之间的通信协议之前，首先简要介绍 OSI 架构与 UDP 通信协议。OSI（Open System Interconnect）架构作为一套开放系统互连参考模型，给出了一种多个信息系统及实体之间的不同层次上的互连的框架。该模型共包含 7 个层次，其中与我们所关心的 UDP 传输协议有关系的是第 4 层传输层。该层所关心的是建立、维持与解除传送连接并关注错误恢复以及流量控制。实际上，在传输层中，包含有 TCP 与 UDP 两种常用的传输协议。但是由于 TCP 在设计过程中考虑的是可靠传输，所以在建立连接时需要多次确认以及数据传输的过程中需要通过确认与重传来保证所传输数据的可靠性，这就导致这种通信协议一方面较为复杂，另一方面对于一些对于准确性与数据可靠性要求不高的场合（如视频流的传输），TCP 显得过于繁琐。故而在一些对于实时性要求较高，同时对于数据的可靠性要求不是很高的场合，面向无连接的 UDP 传输协议更加符合要求。在本项目中，通过 UDP 传输协议，地面站可以向远端无人机发送控制指令，同时远端无人机可以给本地无人机地面站回传自身的地理位置坐标、姿态角、速度、高度等信息，这些信息一旦出现丢包，本地无人机地面控制站不需要要求远端无人机进行重传，只需要等待下一次的发送而来的数据报即可。同时我们需要注意到，考虑到本无人机地面站系统的可扩展性，即未来逐步从目前的“单机单站”向“多机单站”到“多机多站”的一个具有互操作性的复杂系统的衍生，UDP 传输协议通过设置源端口号以及目的端口号并可以实现单播、广播与组播的特性更加符合我们的要求。故而在本项目中选择基于 UDP 通信协议设计无人机地面控制站与远端无人机之间的数据通信链路。

2.2.2 UDP 通信协议在本项目中的应用

由于在本项目中所使用的框架为 Qt，为了简化开发过程，采用 Qt 所提供的网络通信模块，即 QtNetwork 模块中的 QUdpSocket 套接字来构建通信过程。Socket

套接字是最早由 UC Berkeley 在 UNIX4.2BSD 中所提供的一种对于网络连接的源端口与目的端口以及之间的传输层协议的抽象。如图 2-1 所示，套接字的作用是向下对于传输层的协议进行封装，向上为应用层存在的用户进程提供接口。

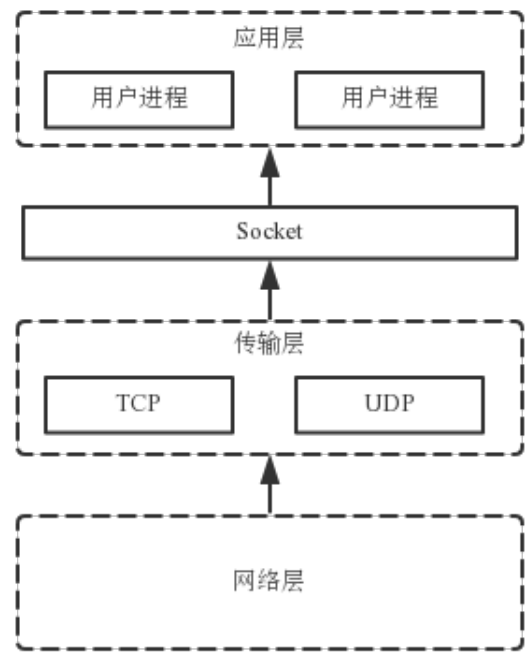


图 2.1 Socket 在 OSI 七层模型中的位置

在本项目中，实现了本地无人机地面控制站与一个无人机代理之间的 UDP 通信连接。无人机代理的作用是对无人机的状态进行一个简化的建模并在接收并解析从地面站传送来的数据报后给出在触发相应的操作指令之后无人机的姿态以及位置等数据信息应当如何变化，即相当于一个仿真。关于无人机代理的具体实现将在之后的章节中详细介绍。通过实现无人机代理与无人机地面控制站之间的通信连接可以验证所采用的通信协议的合理性与可靠性，从而为之后与无人机实体之间的通信连接提供基础。

在 Qt 中进行 UDP 通信较为简单。首先需要设定通信双方的端口。为了避免与系统内核所使用的一些端口形成冲突，通常在用户所运行使用的应用程序中使用大于 1000 的端口作为端口号。同时由于 UDP 报头使用两个字节来存放端口号，这就意味着 UDP 的端口号的取值范围在 0 到 65535 之间。在本项目中，本地无

人机地面控制站所使用的端口号默认为 3355，无人机代理所使用的端口默认为 5555，同时也支持在实际使用的过程中由用户主动修改默认端口号。需要注意的是，为了避免绑定端口时产生冲突，对于进行 UDP 通信的双方，如果双方处在同一个 IP 地址中（如运行在同一台电脑上的无人机地面控制站与无人机代理这样的两个实例），则在通信前需要将二者设置在不同的端口，在发送信息的过程中只需要指定本机 IP 以及目的端口即可；而如果双方在不同的主机上持有不同的 IP 地址，则通常的做法是通信双方约定一个端口，通过该端口进行通信。在设置好通信双方的端口之后，对于发送数据与接收数据分开进行讨论。

发送数据有如下图 2-2 所示的三种传输方式：单播、组播与广播。其中单播表示通信双方的 IP 地址都是唯一确定的，即每次发送数据报的时候只向某一个指定的地址与端口号发送；组播表示每次向某一分组中的所有地址及相应端口号发送数据报；广播则表示每次向与源在相同网络范围中的所有地址及相应端口号发送数据报。在 Qt 中，对于这三种传输方式都有较为直接的封装可以使用。在这里介绍通过单播与广播实现通信过程的基本方法。在实例化一个 QUdpSocket 之后，需要利用 QUdpSocket 的 bind 方法对其进行端口绑定，否则无法监听需要被接受的数据报。绑定成功后利用 QUdpSocket 所提供的 writeDatagram 方法可以实现数据报的发送。对于单播，写法为：

```
udpSocket->writeDatagram((char*)&uav_data,    sizeof(uav_data),    targetIp,
targetPort);
```

其中 uav_data 代表无人机的一些状态参数，targetIp 是目标地址，targetPort 是目标端口。对于广播，写法类似，只需要将上述 targetIp 替换为 QHostAddress::Broadcast 即可。

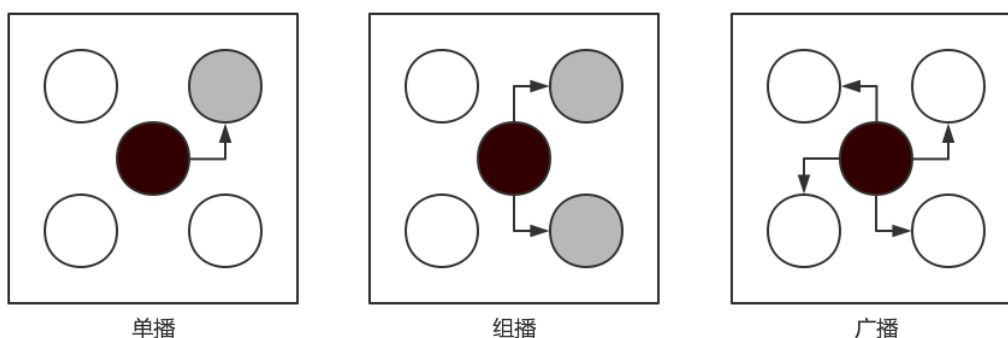


图 2.2 UDP 三种通信形式示意图

接收数据与上述发送数据有一定区别。由于 UDP 是一个无连接的协议，对于通信中接收数据的一方而言，其可以不需要知道究竟是谁给它发送了数据，即数据接收的过程可以不需要识别收到的数据报的源 IP 地址，故而接受过程可以被描述为 UDP 协议在接收端维护一个栈用来接收并缓存发送给当前端口的数据报。通过扫描监听该栈中是否有数据报，来进一步表示是否需要进行读取数据报的操作。在 `QUdpSocket` 中，当接受到数据报时，其会发射 `readyRead` 信号，只需要通过一个信号槽的连接机制将该信号连接到读取数据报的槽函数并利用套接字中所提供的 `readDatagram` 方法即可。写法为：

```
udpSocket->readDatagram((char*)&gcs_control, sizeof(gcs_control));
```

其中省略了对于接受到的数据报的源 IP 地址以及源端口的限制。

上述过程简要描述了在 Qt 中建立无人机地面控制站与无人机或无人机代理之间的 UDP 通信的过程。在本项目中，为了更加方便清晰地进行无人机地面控制站与无人机或无人机代理之间的通信，需要明确传输过程中二者之间传输的数据报中的数据段的内容。

在本项目的实现中，一方面，我们约定从地面站发送向无人机的数据报的数据段内容是一个包含控制指令以及经纬度坐标的结构体，其格式如下图所示。其中 `control_status` 为一个 `int`，从 0 到 5，对应于地面站的一些基本的控制指令，具体内容会在后续章节中详细介绍；`uav_lng` 与 `uav_lat` 均为 `double` 类型数据，代表一组经纬度坐标，其作用是在一些特殊的指令中，需要控制无人机飞往某一个特定的经纬度位置或者为无人机规划好路径后指引其按照某一经纬度坐标序列循路而飞。

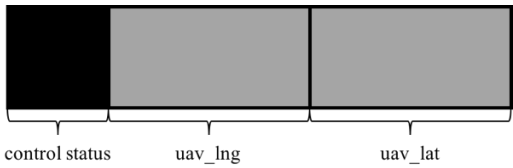


图 2.3 地面站向无人机所发指令的 UDP 数据报数据段内容

另一方面，我们约定从无人机回传到地面站的数据段内容为一个包含无人机状态信息的结构体，其具体格式如图 2-4 所示。其中所包含的数据内容依次为 `double` 型的经纬度坐标以及 `float` 型的方向角、速度、高度、俯仰角、偏航角以及横滚角。无人机或无人机代理通过传输这样的一系列数据给无人机地面控制站就

可以基本实现无人机的姿态展示与航迹显示了，从而满足本项目的基本目标。

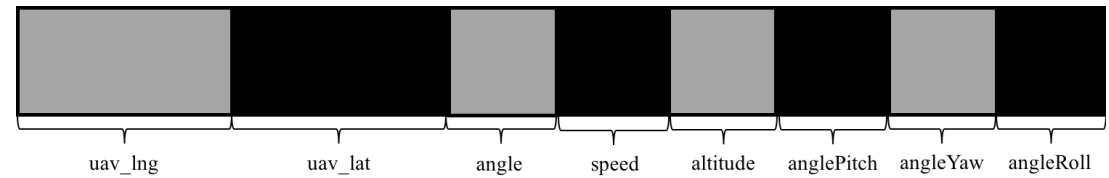


图 2.4 无人机向地面站所发信息的 UDP 数据报数据段内容

我们需要注意的是在网络通信的过程中，数据报中的数据段被强制转换为了一个 `QByteArray` 的字节数组。而对于一个包含各种数据类型的结构体而言，编译器通常会对其进行 4 字节地址对齐。虽然目前我们约定的两个结构体中的数据段最小单位为 4 字节，但是考虑到系统的可扩展性，为了给日后增加更多种类的数据（如 `char` 型数据）提供方便，此处利用 `#pragma pack(1)` 强制将结构体按照 1 字节地址对齐，便于在其被转换为字节数组进行传输之后对其进行解析。

2.3 本章小结

本章介绍了无人机地面控制站的协议层的实现过程。通过介绍 UDP 传输协议并结合套接字在 Qt 框架中实现无人机与地面站之间的数据通信。详细介绍了本项目中所使用的协议在从地面站上行到无人机以及无人机下行至地面站的通信过程中 UDP 数据报数据段的字段内容结构。按照本章节中的方式所设计的地面站通信协议可以满足本项目的基本目标。

第3章 无人机地面控制站数据层

3.1 无人机地面控制站地图导航功能设计

无人机地面控制站作为在本地显示远端无人机的各项飞行数据的本地应用，无论是部署于某些大型的定制设备上还是作为一个轻型的便携式集成操作单元，地图的功能对于其而言都是非常重要的。通过在初期对于国内外各型无人机地面控制站系统的调研可以看到，一个成熟完整的地面站系统应当以地图或者地理信息系统（Geographic Information System, GIS）作为其主界面，从而可以较为直观的显示远端无人机的位置信息，方便操作手在本地的地面控制站对于远端无人机进行进一步的操作，如通过操纵手柄及按键来控制无人机姿态、在地图或 GIS 系统上设置任务点以及为无人机规划航线等。

目前主流的 GIS 系统包括美国环境系统研究所公司（Esri）所开发并维护的 ArcGIS，美国 MapInfo 公司所开发的 MapInfo 等提供二维地理视图以及地理信息的 2D GIS；Skyline 公司所开发的 Skyline 以及开源的 Cesium 等支持三维场景建模的 3D GIS。其中 Cesium 是一个基于 JavaScript 语言的使用 WebGL 的地图引擎，支持 2D、2.5D 以及 3D 形式的地图显示。其可以解析并加载多种二维地图如百度地图、必应地图、ArcGIS 地图以及 MapBox 等，同时可以动态加载三维模型。除了这些较为复杂且功能强大的商业或者开源 GIS，我们日常所使用的百度地图、必应地图等也提供 JavaScript API 接口，供开发者使用。需要注意的是，Google Earth 在此前关闭了开发者 API，故而目前无法通过 Web 形式访问 Google Earth，只能通过在本机运行 Google Earth 软件并调用其 COM API 的形式进行进程通信来访问，这种方法较为复杂。同时由于本项目所使用的跨平台框架 Qt 在 5.5 版本之后引入了 Chrome 浏览器的内核 Chromium，其在性能与使用的便捷程度上都较之前版本有较大改善，故而考虑使用 Web 接口的方式将一个 GIS 模块或者地图模块嵌入到本项目中。综上，在综合考虑到开发的便捷性以及所能呈现的效果后，选择采用百度地图的 JavaScript API 来进行开发。

3.1.1 百度地图开发介绍及控件特点

百度地图为开发者提供 Android 端、iOS 端以及 Web 端的 API 服务接口。其中 Android 端与 iOS 端均支持地图、定位、鹰眼轨迹、导航以及全景这 5 个 SDK，

方便开发者利用百度地图构建行车定位与导航等功能。由于在本项目中对于地图及 GIS 的需求仅仅只是希望得到一个 2D 或 3D 的地理信息环境从而可以在本地无人机地面控制站通过解析远端无人机发送回来的包含其位置信息的数据并实时显示其位置信息及周边环境,故而我们采用百度地图所提供的 Web 端 JavaScript API。通过使用该 API 可以在网页中集成入具有多功能且可以交互的地图模块。

使用百度地图作为无人机位置显示的模块框架的优势如下所示:

(1) 百度地图 JavaScript API 支持 HTTP 及 HTTPS 并免费开放且无访问次数限制,只需要在开发者平台上申请密钥即可直接使用。

(2) 在控件支持方面,百度地图提供部分常用控件,如平移缩放控件和缩略地图控件等。同时也可以自主添加自定义控件,只需要初始化定义所生成的控件的浮动位置以及偏移量并且利用 JavaScript 中的 prototype 属性将自定义控件继承于控件抽象基类 Control 即可。

(3) 百度地图的呈现方式为图层叠加。可以在最底层的地图上叠加标注,从而实现在地图上添加覆盖物的功能。对于所添加的标注,提供点 (Marker)、文本 (Label)、折线 (Polyline)、多边形 (Polygon) 以及圆 (Circle) 等覆盖物。除此之外,通过利用 prototype 属性继承覆盖物抽象基类 Overlay,可以实现自定义覆盖物。覆盖物的使用在本项目中较为重要,一方面需求在于远端无人机需要以一个图标的形式实时显示在地图上从而方便操作手工作;另一方面需求在于当进行无人机自主航迹规划的时候,通常所采用的方法是在地图上绘制一些障碍威胁区 (通常代指敌方火力可以覆盖的区域) 或威胁点 (通常代指敌方的侦查探测源如雷达等) 并按照这些威胁区的空间关系采用一些算法计算出一条可以避开一系列障碍威胁区 (点) 或者最大程度的降低所受到的总威胁的可行航迹。通过百度地图 API 所提供的一系列覆盖物标注的方法可以节缩短项目开发过程同时符合本项目的需求。

(4) 百度地图提供在城市环境中的多种出行方式的路线规划。由于大部分无人机项目所预设的应用场景在尺度上较为宏观,所以对于城市环境一般不做特别考虑。但是在本项目中,我们希望通过在获取城市道路以及楼宇信息之后可以为无人机系统提供一种多任务点的航线规划功能,所以百度地图 API 中所提供的多种路线规划方式对于本项目的建立可以起到简化项目开发流程并为我们所设计的算法提供对照组的功能。

3.1.2 百度地图的加载

由于所使用的是 Web 端 API，所以构建地图实例的内容均在一个 html 文件中完成。首先需要完成 html 文件中的 head 段的编写从而将地图容器按照所设置的样式嵌入到浏览器页面中。为了界面的美观性以及设计的整体性，不妨让地图容器充满整个网页。在 head 段中加入百度地图 API 的脚本文件即可在后续编写 JS 脚本从而实现加载各类控件、算法逻辑以及与 Qt 主程序之间的通信。

此后只需要在 script 标签中实现 JS 脚本。可以通过实例化一个 BMap 对象来创建地图实例。进一步的，可以通过其余一些方法及属性完成地图的滚轮缩放、通过 IP 或者经纬度坐标设置地图初始中心等功能。具体不予赘述，详细可以参考百度地图 JavaScript API^[4]及百度地图开放平台^[5]中的实例。

3.1.3 百度地图中引入功能性控件

在地图中可以通过调用实例化后的地图中的 addControl 方法来向其中引入地图控件。除一些基本的控制地图的控件之外，根据本项目中的需求，需要在进行航迹规划之前在地图上通过交互式操作设置威胁区。由于在百度地图的开源库中提供一个鼠标绘制的工具条库，其支持在地图上通过点击鼠标交互性的绘制点、折线、多边形、矩形以及圆，故而在本项目中引入该 DrawingManager 库。在本项目中，添加地图类型、平移与缩放、小地图（缩略地图）、比例尺以及鼠标绘图工具实现项目需求。

同时百度地图也支持创建自定义功能性控件。通过定义控件的构造函数并继承抽象基类 Control，可以创建自定义功能控件。在创建控件的时候需要为方位（defaultAnchor）以及偏移（defaultOffset）设置初始化。当使用该控件的时候，与前面使用 API 中提供的控件方法相同，即创建一个已经定义好的自定义控件类的对象，并使用 addControl 方法。创建控件对象的同时会对其进行初始化，此时只需要为 PositionDisplay.prototype.initialize() 方法创建一个 DOM 元素并为此 DOM 元素绑定相应的事件，完成这些后将该控件 DOM 元素返回给 initialize() 方法。DOM 全称为 Document Object Model 即文档对象模型。其作用是将文档内的对象按照树形结构组织起来。在本项目所使用的百度地图中，最终所有的控件 DOM 元素都将被包含进地图容器所在的 DOM 元素中，从而构成一个以地图容器为根，各个控件为子节点的树状结构。

最终在 Qt 主界面中嵌入的地图及相应控件模块的结构如图 3.1 所示。可以看到其中被红色方框勾画出的区域为在地图控件模块中所引入的控件，从左到右从上到下依次对应的控件为平移缩放、鼠标绘图、地图类型、比例尺以及小地图（缩

略图)。

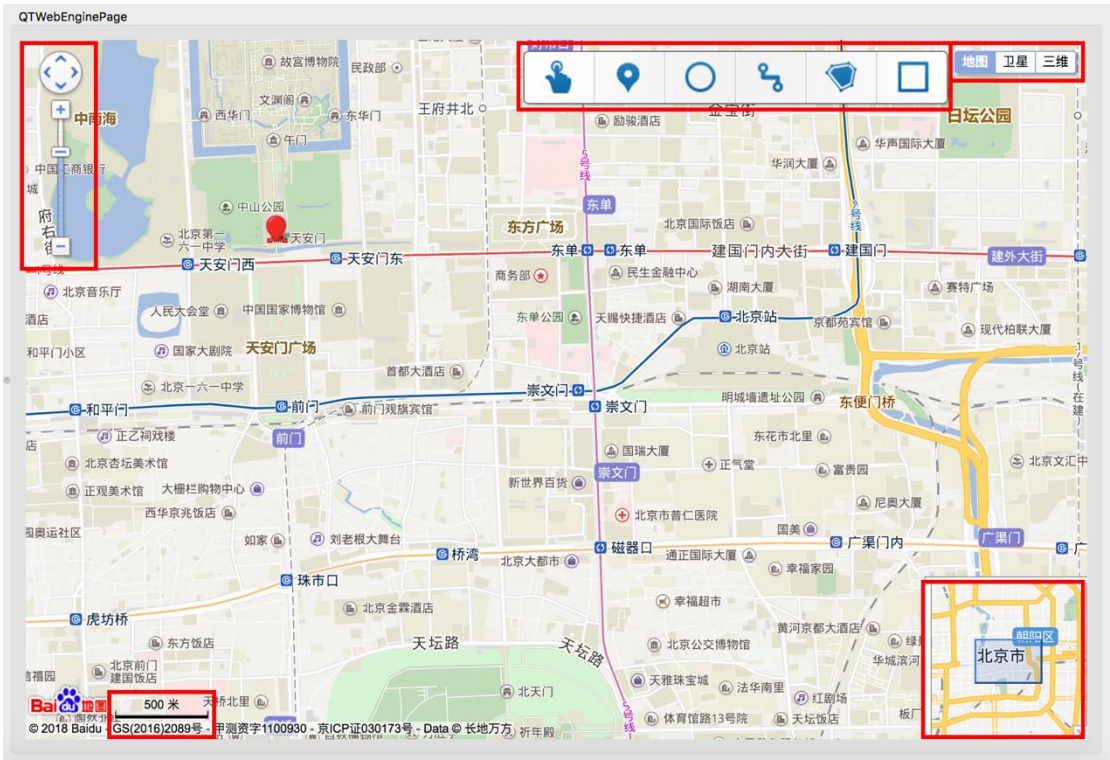


图 3.1 Qt 主程序中地图控件模块效果图

通过利用鼠标绘制类 `DrawingManager` 可以为地图添加覆盖物从而形成本项目中所需要的威胁区域的交互式构建，效果如图 3.2 所示。图中展示了添加多边形、矩形以及圆等覆盖物在普通地图（左）与卫星地图（右）上的效果。

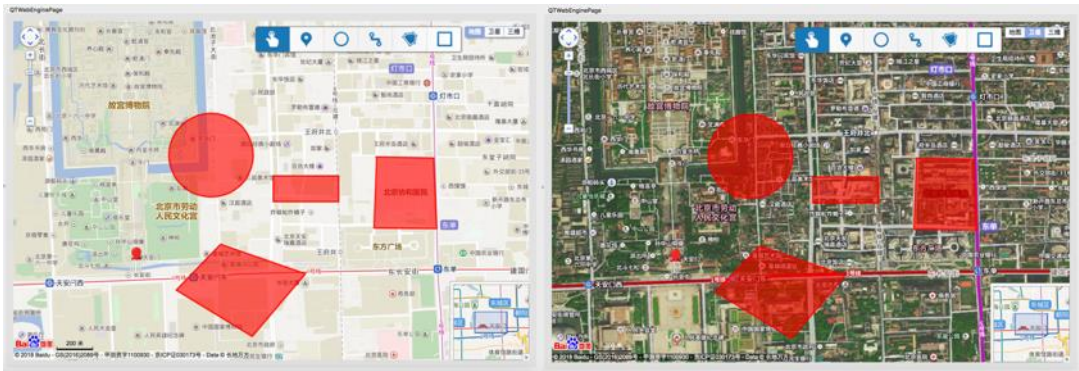


图 3.2 在地图控件中增加威胁区效果图

按照上述介绍将百度地图嵌入到本项目中，从而可以通过百度地图获取位置

坐标信息及周边地理环境信息,满足了本项目对于地图或者 GIS 系统的基本需求。同时也为项目添加多种 GIS 的可行性提供了技术基础。

3.1.4 Qt 中开发 Web 应用的方式

Qt 在 5.2 版本给出了一个新的网络浏览器模块 WebEngine 并且在 5.5 版本之后摒弃了此前所使用的 WebView 模块。在新的框架中, WebEngine 以 Chromium 为内核,通过这种将一个浏览器内核植入到 Qt 框架中的方式, Qt 可以在系统中没有浏览器的情况下依旧为用户提供与网页进行交互的功能。

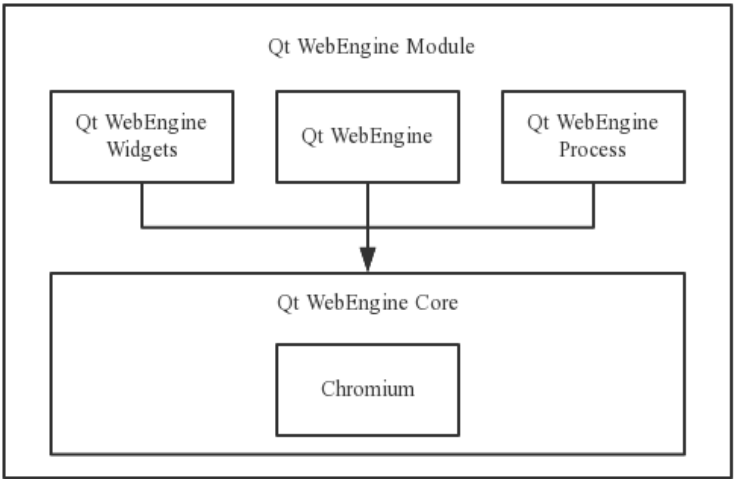


图 3.3 Qt WebEngine 模块架构

如图 3.3 所示, Qt 中 WebEngine 模块包括 Qt WebEngine、Qt WebEngine Widgets 以及 Qt WebEngine Process。其中 Qt WebEngine 模块用来在 Qt Quick 中构建 Web 应用, Qt WebEngine Widgets 用来在 C++风格的 Widgets 窗体中构建 Web 应用, 而 Qt WebEngine Process 是一个渲染网页以及处理网页中的 JS 脚本的独立进程, 在使用过程中作为整个进程中的一个子进程, 通常用户感知不到其存在。

在本项目中, 使用 Qt WebEngine Widgets 模块将百度地图的网页界面包装为一个 Widget 嵌入到主窗口中。图 3.4 给出了 QtWebEngine Widgets 模块内子模块之间的调用关系。可以看到, 在一个包含 Web 网页的窗体组件中, 最顶层的组件为一个用来加载网页的 WebEngine View。每一个 view 都需要包含一个用来表示网页内容的 page。通过一个 page 还可以获得这个网页组件访问网页的历史 history、该网页可以接受的响应 action 以及控制网页的 JS 脚本和 cookie 的 profile。在本项目中, 只使用其中的 view 以及 page 类即可满足项目需求。

首先继承 WebEngine Widgets 模块中的 QWebEnginePage 类，实例化该派生类可以得到一个用于浏览或者编辑网页的对象。这里为了调用保存在本地并已经加载入 qrc 的网页文档资源，如代表本项目中所需要的百度地图控件模块所在的 baiduMap.html，需要使用 QWebEnginePage 中的 acceptNavigationRequest()方法。该方法为纯虚函数，故而需要在继承 QWebEnginePage 类的时候实现该方法。该方法的作用是如果返回 true 则接受所传入的 url 进行导航即加载页面，否则该 WebEnginePage 对象不接受导航。为了调用存放在本地 qrc 中的网页文档，通过判断所传入 url 地址的 url scheme 是否为 qrc 来实现对于 qrc 中相应 url 的加载。之所以采用这样的方式是为了本项目的无人机地面站系统在之后逐步更新时可以与更多的本地 App 项目实现链接。url scheme 是一种本地应用程序之间实现链接跳转的方式，目前用户经常遇到的点击某个链接可以从一个应用程序（如微信）跳转到另一个应用程序（如知乎）就是通过 url scheme 技术所实现。这里不对 url scheme 技术进行详细介绍，详细说明可以参考 facebook 所提供的相关开发者文档^[6]。

将上述继承 WebEnginePage 得到的派生类实例化之后利用 setPage()方法加载入相应的 WebEngineView 窗体组件之后就可以完成网页文档的加载。此后可以通过 setUrl 的方式加载前面两小节中所介绍的百度地图 html 网页文档。。

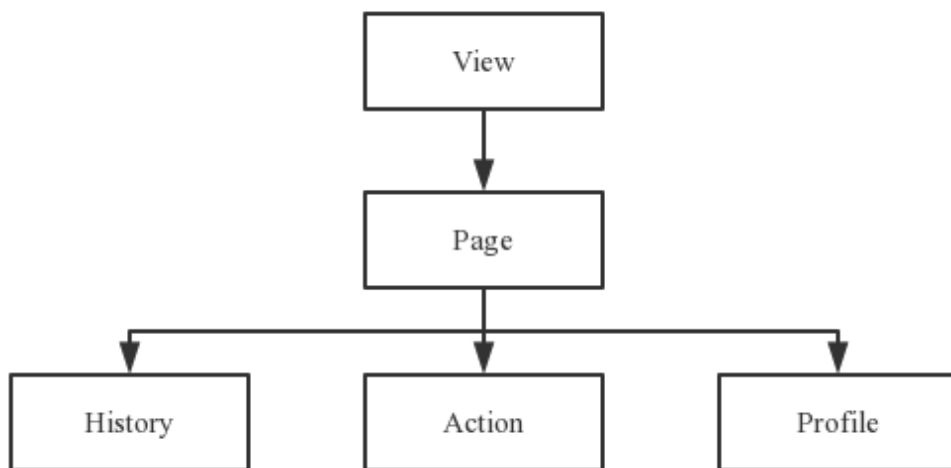


图 3.4 Qt WebEngine Widget 模块内子模块架构

在完成了网页文档的加载之后，一个重要的问题是如何与其进行通信。在 WebEngine 框架中，Qt 窗体与 HTML 页面内的 DOM 元素以及 JS 脚本之间的通信过程通过一个 QWebChannel 类实现。为了便于构建项目，定义了一个 Bridge 类

用来表示经过 QWebChannel 所传递的各个参数以及它们对应的信号槽。可以理解为 QWebChannel 相当于通道而 Bridge 类相当于在通道中被传送的物品。对于每一个构建出来的通道，需要调用 registerObject()方法来为其注册其需要进行传送的物品从而构建一个特定的可以传送某些物品的通道。在构建完成该通道之后，使用 setWebChannel()方法将其与 page 对象进行绑定从而为该 page 对象设置其所关联的通道从而实现窗体中与 HTML 网页之间的通信。

相应的，在 HTML 网页文档中，需要通过在其 script 段中使用 QWebChannel 类来接受从窗体传递过来的参数以及向窗体回传参数。通过在这里关联 Bridge 类中的向 HTML 页面发送数据的信号或者调用 Bridge 类中的接收 HTML 页面传递过来的参数的槽函数可以分别实现窗体向网页页面传递参数以及网页页面向窗体传递参数，具体方式如下所示：

```
new QWebChannel(qt.webChannelTransport, function(channel) {
    var content = channel.objects.content;
    content.sendCoordinate.connect(function(message1, message2) {
        // message have been sent by the paras
    });
    content.receiveCoordinate(lng, lat); // send paras back to widget
});
```

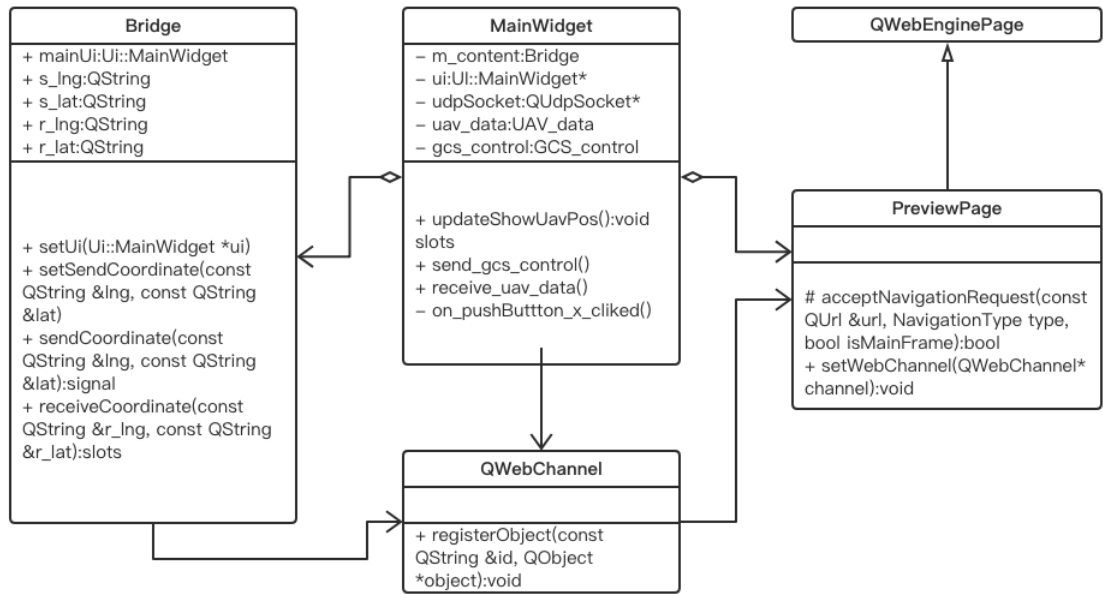


图 3.5 本项目实现在 Qt 窗体中嵌入一个 Web 页面并与之通信的 UML 类图

在详细介绍了网页的加载以及窗体与网页之间的通信的实现方法之后，图 3.5 给出了本项目中这一部分的相关类之间的 UML 类图。

上述内容介绍了本项目中实现 Qt 窗体与 HTML 页面之间进行双向参数传递的过程，从而可以实现从 Qt 窗体中向 HTML 页面传递远端无人机的位置坐标以及 HTML 页面在获得操作手通过交互性操作在其上构建的威胁区的参数后将其回传给 Qt 窗体从而可以为进一步进行航迹规划算法的实现提供所需参数，满足了项目的预期需求。

3.2 无人机地面控制站数据库模块

由于远端的无人机通常为小型无人机，其数据存储能力普遍不强，而相对应的本地地面控制站的数据存储能力显然更有优势。为了便于在远端无人机任务结束之后可以再次复盘之前在执行任务时的航迹、姿态动作等数据，地面控制站应当构建一个用于存储无人机飞行时各项被传输回地面控制站的数据的数据库。

表 3.1 UAVData 数据表中的字段定义

序号	字段名	类型	参数意义	说明
1	stamp	INT	时间戳	主键-非空
2	uavLat	REAL	纬度	
3	uavLng	REAL	经度	
4	anglePitch	REAL	俯仰角	
5	angleYaw	REAL	偏航角	
6	angleRoll	REAL	滚转角	
7	speed	REAL	对地速度	
8	altitude	REAL	海拔高度	
9	angle	REAL	飞行方向角	

对于本项目中使用的跨平台框架 Qt，其提供与一些常用的数据库之前的连接。通过在项目配置文件中添加 sql 模块就可以在程序中使用 QtSQL 模块中的类 QSql。Qt SQL 支持的数据库驱动包括：Oracle、MySQL、Borland InterBase、Access 以及 SQLite 等。为了方便起见，本项目目前采用轻量的跨平台单机型数据库 SQLite。使用 SQLite 的优点在于其所有的数据表及索引都存储在一个文件中，对

于操作者以及应用程序而言可以将其对待为一个文件，使用起来非常方便。

在本项目中，需要被存储入数据库的为远端无人机向地面控制站所发送回来的数据报内的数据段信息。根据前面所介绍的传输协议，我们可以得到需要在 SQLite 中所建立的 UAVData 数据表的各个字段及其所代表的含义，如表 3.1 所示。需要注意的是，为了构建起一个方便查询的数据表，应当为该表设置主键。这里采用时间戳作为主键，即收到的第 1 个 UDP 数据报是时间戳 stamp 记为 1，此后每收到一个数据报，该时间戳自增 1 并与数据报中的数据段一同被写入 UAVData 数据表中。

在 Qt 中，通过定义一个 QSqlDatabase 类的对象来实现与数据库的连接。对于已经通过 SQLite 建立好的数据库文件，可以通过读文件的方式读入并与 QSqlDatabase 类的对象连接从而完成数据库的连接与登录。接着调用 QSqlDatabase::open()方法完成数据的打开。具体实现方式如下所示：

```
QSqlDatabase db = QSqlDatabase::addDatabase("QSQLITE");  
db.setDatabaseName("dbFileName.db");  
db.open();
```

在打开了一个数据库之后，通过 QSqlQuery 类来创建数据库表。该类提供直接使用 SQL 语句进行数据库操作的接口。QSqlQuery 的 exec()方法接受一条 SQL 语句从而实现创建数据表。具体的实现方法如下所示：

```
QSqlQuery query;  
query.exec("create table uavdata  
          (stamp int primary key,  
           uavLat real,  
           uavLng real,  
           ...,  
           angle real)");
```

在完成了数据表的建立之后，可以通过使用 QSqlQuery 类中的 prepare()方法来为插入的多个字段设置占位符防止字段被错误转义为字符串，之后利用 bindValue()方法绑定记录中的各个字段的值，最终通过 exec()方法来向数据表中插入记录。具体的实现方式如下所示：

```
query.prepare("insert into uavdata values(?,?,?,?,?,?,?,?)");  
query.bindValue(0,r_stamp);  
...
```

```
query.bindValue(8,r_angle);
```

在完成了向数据表中插入记录之后，可以通过使用 SQL 语句 `select` 来获取数据表的相应字段的数据记录。具体方法为：

```
query.exec("select * from uavdata order by stamp");
```

执行完上述语句之后 `query` 指针指向结果集中的第一个位置，需要循环调用 `query` 的 `next()` 方法来逐一读出 `uavdata` 数据表中的每条记录。向 `query` 的 `value()` 方法传入一个字段编号即可获得对应数据记录的某一字段的值。由此实现了对于数据表 `uavdata` 按照时间戳 `stamp` 的升序读取数据记录，即按照时间先后顺序读取无人机发送给地面站的数据从而实现复盘功能。

3.3 本章小结

本章介绍了无人机地面控制站系统中的数据层的实现过程。之所以将这些功能模块结合在一起称为数据层，是因为这些模块为整个系统提供了数据支持。地图与 GIS 模块为地面站提供了无人机所在位置及周边地理环境信息；数据库模块为地面站提供了存储无人机发送来的数据并在需要的时候读取相应数据的能力。数据层通过使用协议层所提供的传输接口实现地面站与无人机的数据传输与共享，同时为上层的应用层提供所需要的数据以及对于数据的读取与存储能力。

在本章中首先详细介绍了百度地图 JavaScript API 在 Qt 中的使用方法（包括对于 API 的解读以及在 Qt 中嵌入 Web 应用的项目结构与类间关系）并给出了部分示例代码。之后介绍了 Qt 中使用数据库的方式并配合 SQLite 数据库给出部分在本项目中所使用到的对于数据库的处理方法的示例代码。

通过使用本章中所介绍的方法可以实现一个满足项目需求的简易数据层。从而与协议层一起共同为上层的应用层提供服务。

第4章 无人机地面控制站应用层

4.1 无人机状态仿真及无人机代理

本项目所实现的无人机地面控制站需要同远端的无人机进行数据通信从而可以实现远端无人机在本地地面站平台上显示姿态数据以及位置信息等功能。为了方便开发与调试本项目所实现的无人机地面控制站，配套构建了一个无人机仿真模型即无人机代理。该代理的作用是模拟仿真一个与地面站平台相连的远端无人机实体。在接收到地面站的控制指令之后，该代理会根据指令对于相关无人机参数进行修改并定时向地面站通过传输协议进行数据回传。本小节将首先分析该无人机代理应当包含的状态参数，其次介绍在构建无人机代理的过程中需要对于全球地理坐标系进行的一些近似处理，最后会给出本项目中所实现的一个简易的无人机代理。

4.1.1 无人机状态仿真及所需参数

无人机可以按照常规的结构布局分为固定翼与旋翼两种，此外还包括一些采用非常规结构布局的无人机，如扑翼飞机等。对无人机进行较为严格的状态仿真的前提之一是需要了解无人机的气动布局以及由此带来的对于无人机的空气动力学建模的影响。对于普通的固定翼飞机而言，主要包括为飞机提供上升动力的升力在机翼水平方向上的分量即升致阻力以及飞机自身在飞行过程中其他部分所受到的阻力即寄生阻力。这些阻力同空气密度、无人机速度、无人机翼展面积及其截面面积等都有相关性。而旋翼飞机除了需要考虑上述两种阻力之外还需要考虑旋翼上的剖面阻力^[21]。

可以看到，对于实际的无人机进行建模是一件较为复杂的事情。由于在本项目中所关心的重点在于构建无人机地面控制站，所以对于无人机的仿真建模采取一种简化的方式，即忽略这些在实际无人机飞行过程中产生的阻力，将无人机近似为一个在空域中移动的不受外界因素干扰的具有姿态的物体。该物体拥有速度 v 、高度 h 以及方向角 α 这三个属性。其中速度表示对地速度，高度表示海拔高度，方向角表示无人机飞行方向与正北方向之间的顺时针夹角并约定数据范围从0度到360度。进一步的，对于一个具有姿态的无人机而言，其机身坐标系同大地坐标系之间的关系可以通过自身的三个姿态角来表示。即俯仰角 θ (Pitch)、滚转角

γ (Roll) 以及偏航角 φ (Yaw)。图 4.1 给出一个直观示例^[22]。

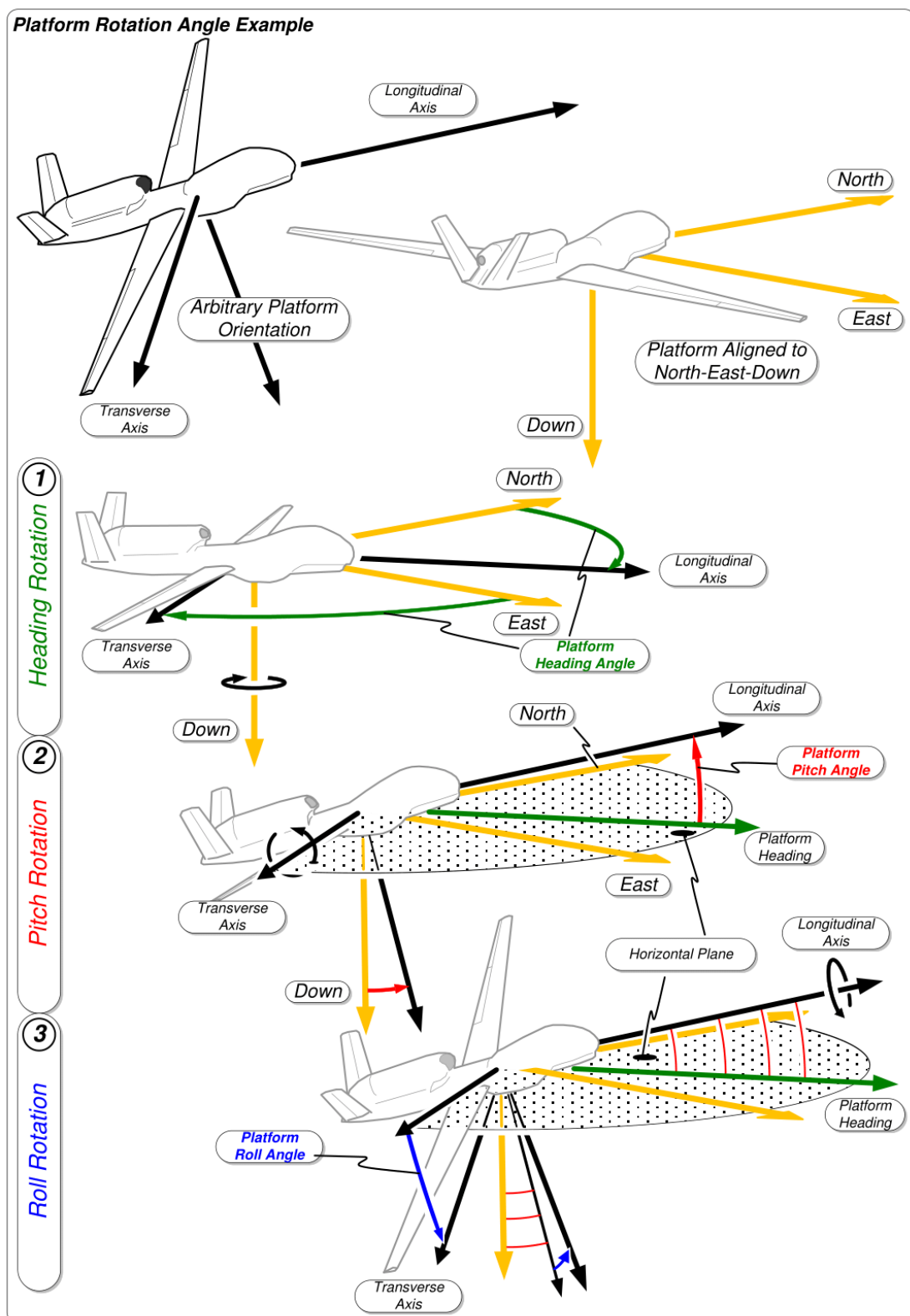


图 4.1 无人机姿态角同大地坐标系之间的关系

图 4.1 所示来源于美国国家地理空间情报局（The National Geospatial-Intelligence Agency, NGA）下属的运动图像标准委员会（Motion Imagery Standards Board, MISB）所制定的 0601 标准（MISB Standard 0601）。该标准详细介绍了其无人机系统的数据链路本地集（Datalink Local Set）^[22]。

该数据链路本地集以“捕食者”无人机作战平台为例介绍了地面控制中心与远端无人机之间进行数据链路通信时无人机平台所传递回来的自身各项数据的格式。其目的是通过这种约定好的标准的形式可以丰富无人机平台所获得的动态图像的信息，即将无人机平台的自身的各项数据规定为标准化的元数据（metadata）从而实现与所获得的运动图像的同步传输^[14]。元数据中包含了对于无人机平台的姿态角的定义。如图 4.1 所示，对于无人机平台，其自身的坐标系由一个从尾翼指向机头的纵向坐标轴、一个从左翼指向右翼横穿机身的横向坐标轴以及一个与之前两个坐标轴所形成的平面呈垂直关系的指向地心方向的垂直方向坐标轴共同构成。为了方便表示，按照三维右手笛卡尔坐标系的形式，将上述 3 个坐标轴依次定义为 x_b 、 y_b 和 z_b 且其正方向分别依次指向无人机机身的前方、右方以及下方，和无人机中心原点 o 共同组成机体坐标系 $ox_by_bz_b$ 。

MISB 0601 标准通过规定方向角（heading rotation）、俯仰角（pitch rotation）以及滚转角（roll rotation）从而可以描述无人机的某一瞬时时间的姿态。其中当上述三个角度均为 0 度的时候，无人机机翼被纬度线横穿且平行于地面向正北方向飞行。方向角表示无人机的飞行方向（即 x_b 轴正方向）在正北与正东方向所形成的平面上的投影与正北方向之间的顺时针夹角，即图 4.1 中第 1 部分用绿色箭头标注出来的 Platform Heading Angle 角度。俯仰角表示无人机当前飞行方向与地面正北-正东坐标平面之间的夹角，亦即 x_b 轴正方向与 x_b 轴投影在地平面上的无人机飞行方向矢量之间的夹角，即图 4.1 中第 2 部分用红色箭头标注出来的 Platform Pitch Angle 角度。滚转角表示无人机当前翼展方向与地面正北-正东坐标平面之间的夹角，亦即 y_b 轴正方向与地面水平面之间的夹角，即图 4.1 中第 3 部分用蓝色箭头标注出来的 Platform Roll Angle 角度。

在表示飞机姿态的过程中偏航角 φ （Yaw）的定义较为模糊。在一些情况下其被用作与上文所述的方向角（heading rotation）等同的定义。而在一些存在领航僚机或者事先预定好了航线的情况下，偏航角被用作表示与无人机自身飞行方向同领航僚机相对于无人机自身的位置矢量之间的夹角。在本项目中，我们对于偏航角的定义分两种情况进行考虑。当没有加载航线的时候，即当前飞行的远端无人机没有预定航线可以参考的时候，此时无人机的偏航角表示无人机方向角变化的

角速度，即

$$\varphi = \frac{\partial \alpha}{\partial t} \quad (4-1)$$

当无人机地面控制站为远端无人机进行了航迹规划后，无人机的偏航角 φ 的意义就更改为表示预定航线方向矢量与无人机当前飞行方向之间的顺时针夹角。即若预定航线方向矢量到无人机当前飞行方向之间的顺时针角度取值为 0 到 180 度（含），则偏航角为对应的顺时针角度值；否则偏航角取预定航线方向与无人机飞行方向之间的逆时针角度值（取值范围为 0 到 180 度（不含））。

上述六个参数是本项目中所关注并建模的无人机代理的动态参数。除了这些动态参数之外，还需要一些固定的初始化参数，如无人机的最大飞行速度，无人机的加速度，无人机的转弯角速度与角加速度等。这些参数的种类以及与无人机上述六个动态参数之间的关系将在之后的无人机代理实现部分详细介绍。

4.1.2 无人机位置仿真与坐标计算

在介绍无人机代理的详细实现过程之前，还需要对于地理坐标系进行简要的介绍，从而为无人机代理实现过程中所遇到的需要计算两个经纬度坐标之间的距离以及两个经纬度之间的方向角的情况提供理论基础。

WGS（World Geodetic System）是由美国国防部提出的一种地心坐标系，目前在国际上被大量采用。其最新版本 WGS-84 被用在 GPS 定位系统中。如图 4.2 所示，WGS 坐标系将地球近似认为是一个椭球体，即一个长半轴与短半轴分别为赤道半径以及子午面与地表相切所得切面半径的椭圆绕地轴 Z 轴旋转所得到的旋转椭球体。由此可以将任意一点按照其于地心坐标系 OXYZ 的 X 轴夹角与 OXY 平面的夹角即经纬度坐标以及高程信息表述为 $P(L, B, H)$ ，其中 L 为经度， B 为纬度， H 为高度^[14]。

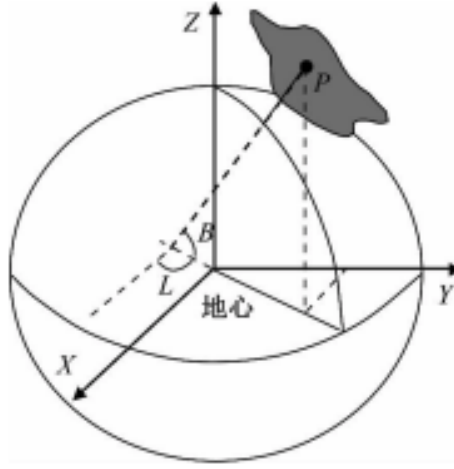


图 4.2 WGS 坐标系示意图

为了计算两个经纬度坐标之间的球面距离，若近似将上述椭球体看做一个正球体，则可以借助 haversine 公式^[15]。首先定义 haversine 函数：

$$hav(\theta) = \sin^2\left(\frac{\theta}{2}\right) = \frac{1 - \cos(\theta)}{2} \quad (4-2)$$

记两个坐标点之间的球面距离为 d ，球体半径为 r ，两者经度换算为弧度制分别为 λ_1 、 λ_2 ，两者纬度换算为弧度制分别为 φ_1 、 φ_2 ，则有如下所示的 haversine 公式：

$$hav\left(\frac{d}{r}\right) = hav(\varphi_2 - \varphi_1) + \cos(\varphi_1)\cos(\varphi_2)hav(\lambda_2 - \lambda_1) \quad (4-3)$$

haversine 公式给出了一种计算球体表面上的两个点之间的圆心角的方式。由于在本项目中对于上述计算两个点之间的距离的需求仅为计算从无人机代理向地面站所发送的前后两个数据报的数据段之间的位置距离以及二者之间的方向角，可以对上述 haversine 公式进行一些近似优化。对于一般的旋翼无人机而言，其最大运动飞行速度通常在 100km/h 内，如 DJI 的 Inspire2 型号无人机最高时速为 93km/h；而对于一般的固定翼无人机而言，其最大运动飞行速度通常在 300km/h 以内，如美国“捕食者”无人机最高时速为 240km/h。我们不妨认为本项目中所适配的无人机速度在 300km/h 量级即约 83m/s。由于我们在无人机代理处规定，其每隔一秒时间需要向无人机地面站发送数据包报告自身的位置以及姿态等信

息数据，那么在一秒时间内，其最多移动的距离约为 83m/s。考虑地球赤道半径为 6378137m，故而其位置改变的球心角约为：

$$83/6378137 \approx 0.000013 = 180^\circ \times 0.000013/\pi \approx 0.00075^\circ \quad (4-4)$$

显然圆心角值足够小，也就意味着经纬度的改变值足够小。故而对上述 haversine 公式进行如下的近似变换：

$$\sin^2\left(\frac{d}{2r}\right) = \sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos(\varphi_1)\cos(\varphi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right) \quad (4-5)$$

利用 θ 很小时 $\sin \theta \approx \theta$ 的性质对上式进行近似处理并化简：

$$\left(\frac{d}{2r}\right)^2 = \left(\frac{\varphi_2 - \varphi_1}{2}\right)^2 + \cos(\varphi_1)\cos(\varphi_2)\left(\frac{\lambda_2 - \lambda_1}{2}\right)^2 \quad (4-6)$$

$$\left(\frac{d}{r}\right)^2 = (\varphi_2 - \varphi_1)^2 + \cos(\varphi_1)\cos(\varphi_2)(\lambda_2 - \lambda_1)^2 \quad (4-7)$$

由于纬度改变很小，不妨认为 $\varphi_1 \approx \varphi_2$ ，故而上式可以进一步化简得到：

$$d = r \times \sqrt{(\varphi_2 - \varphi_1)^2 + [\cos(\varphi_1)(\lambda_2 - \lambda_1)]^2} \quad (4-8)$$

类似的可以得到计算上述两个经纬度坐标之间的方向角 α 的方式：

$$\alpha = \arcsin\left(\frac{r \times \cos(\varphi_1)(\lambda_2 - \lambda_1)}{d}\right) \quad (4-9)$$

通过上述过程给出了本项目计算在 WSG 地理坐标系中获得两个相邻真实经纬度坐标后计算它们之间距离的方式。即，当与真实的具有 GPS 或北斗数据模块的无人机相连的时候，应当采用上述方式进行经纬度坐标之间的距离与方向角的计算。对于无人机代理可以进一步采用墨卡托投影所得到的平面坐标系，即认为经纬度线等距垂直分布，即对于上述计算距离以及方位角的时候，认为 $\cos(\varphi_1)$ 项始终为 1。

4.1.3 无人机代理参数设计与实现

通过前述过程的介绍，给出无人机代理的具体参数表如下所示：

表 4.1 无人机代理参数表

参数变量名称	参数意义	单位	类型
uav_lat	纬度	°	double
uav_lng	经度	°	double
anglePitch	俯仰角	°	float
angleYaw	偏航角	°	float
angleRoll	滚转角	°	float
speed	对地速度	m/s	float
altitude	海拔高度	m	float
directionAngle	方向角	°	float
accelerateSpeedUp	正向加速度	m/s^2	float
accelerateSpeedDown	反向加速度	m/s^2	float
directionChangeAngle	方向角速度	°/s	float
rollChangeAngle	滚转角速度	°/s	float
pitchChangeAngle	俯仰角速度	°	float
rollChangeMaxAngle	最大滚转角度	°	float
pitchChangeMaxAngle	最大俯仰角度	°	float
climbRate	爬升速率	m/s	float
maxSpeed	最大飞行速度	m/s	float
maxAltitude	最高飞行高度	m	float

初始化时应设置无人机代理所在的经纬度坐标。同时可以对于表示无人机自身性能的参数（即上述表格中位于方向角之后的固定参数）进行设置。对于无人机代理的状态采用简化模型。认为无人机在加速过程中与减速过程中为一个匀加速运动过程，直至达到最高运行速度或者速度归零。认为无人机在转弯的时候其方向角以左转或右转方向角改变角速度匀速变化。认为无人机在上升或者下降的时候保持机头上仰或下沉角度，从而以一个与速度相关的形式匀加速改变无人机所处海拔高度直至达到最大海拔高度或者落在地平面上（注意，目前为了方便处

理暂时没有考虑地面高程信息，在今后的工作中将把地面高程信息引入）。

考虑到一般安装在真实无人机上的 GPS 模块大约每隔 1s 时间向地面站发送一次位置信息，故而在此处为了模仿真实无人机与地面站之间的通信方式来更新自身信息，无人机代理维护一个计时器类 QTimer 的对象 updateTimer。通过设置 updateTimer.start(1000)使得其每隔 1000ms 触发一次超时并向地面站发起一次 UDP 通信，从而使得地面站可以每隔大约 1s 就能更新无人机代理的状态信息。

4.2 无人机地面控制站操纵台

无人机地面控制站操纵台的形式多种多样，可以通过外接控制手柄或者地面控制站主界面上提供软按键等方式来实现。在本项目中，通过利用 Qt 中的 QPushButton 实现简单的地面控制站虚拟操控台，样式如图 4.3 所示

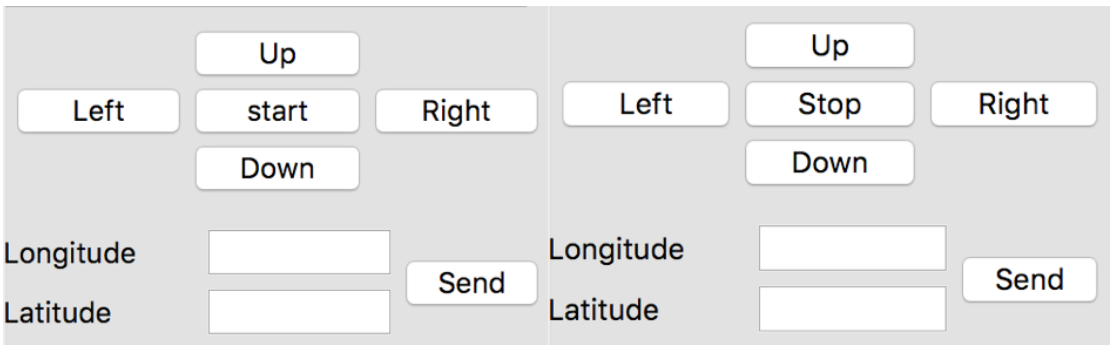


图 4.3 无人机地面控制站操纵台

其中 Start 按钮用来使能其余按钮并开始接收无人机代理发送来的 UDP 数据报。Up、Down、Left 以及 Right 分别表示控制远端无人机向前加速、向前减速、向右侧转弯以及向左侧转弯。通过绑定信号槽的方式连接到通信协议模块的发送 UDP 数据报模块。利用这种方式可以实现使用虚拟按钮的方式对无人机进行简单的控制。特别的，为了模仿 USB 手柄对于无人机的控制方式，这里对于按钮触发的信号考虑的是按钮按下（pressed）与松开（released）这两个信号。按键按下，则触发相应的槽函数，修改当前无人机状态标志位所按下按键对应的状态并向无人机代理发送 UDPSocket。按键释放，则触发相应的槽函数，修改当前无人机状态标志位为 5（表示保持当前速度和方向角匀速飞行）并向无人机代理发送 UDPSocket。

4.3 无人机地面控制站虚拟仪表

除了需要通过地图或者 GIS 控件来实时显示无人机位置状态信息，地面站系统中还应当有可以显示无人机各项飞行姿态与飞行状态的数据。本项目中利用一款开源的飞机仪表盘控件 QFlightInstrument^[16]实现对于无人机以及无人机代理传送回地面控制站之后的数据可视化。如图 4.4 所示，本项目中包含了速度计、飞行方向角罗盘、高度计、展示飞机姿态角的角度仪以及一个综合展示各项数据的综合姿态表。其中速度计、角度仪、高度计以及罗盘是构成飞机仪表盘中的基础部分，因其分布的位置通常被称为 Basic T。



图 4.4 无人机地面控制站虚拟仪表

4.4 无人机地面控制站航迹显示

为了便于操作手通过地面站对于远端无人机进行控制，还需要为本地无人机地面控制站增加航迹显示功能。在本项目中使用的百度地图可以通过将一组经纬度坐标点作为一个序列传入其中并更新此前所绘制的折线（Polyline）的方式来实现航迹显示的功能。具体方式为每次获得由 QWebChannel 向 Web 页面传输的经纬度坐标参数，由此建立坐标点并将其加入包含所有坐标点的队列中。对于该队列调用的最后一个元素调用绘制折线的 polyline 类的 setPath()方法即可实现航迹的绘制。图 4.5 给出了本项目所实现的航迹绘制的效果，可以看到基本符合预定需求。

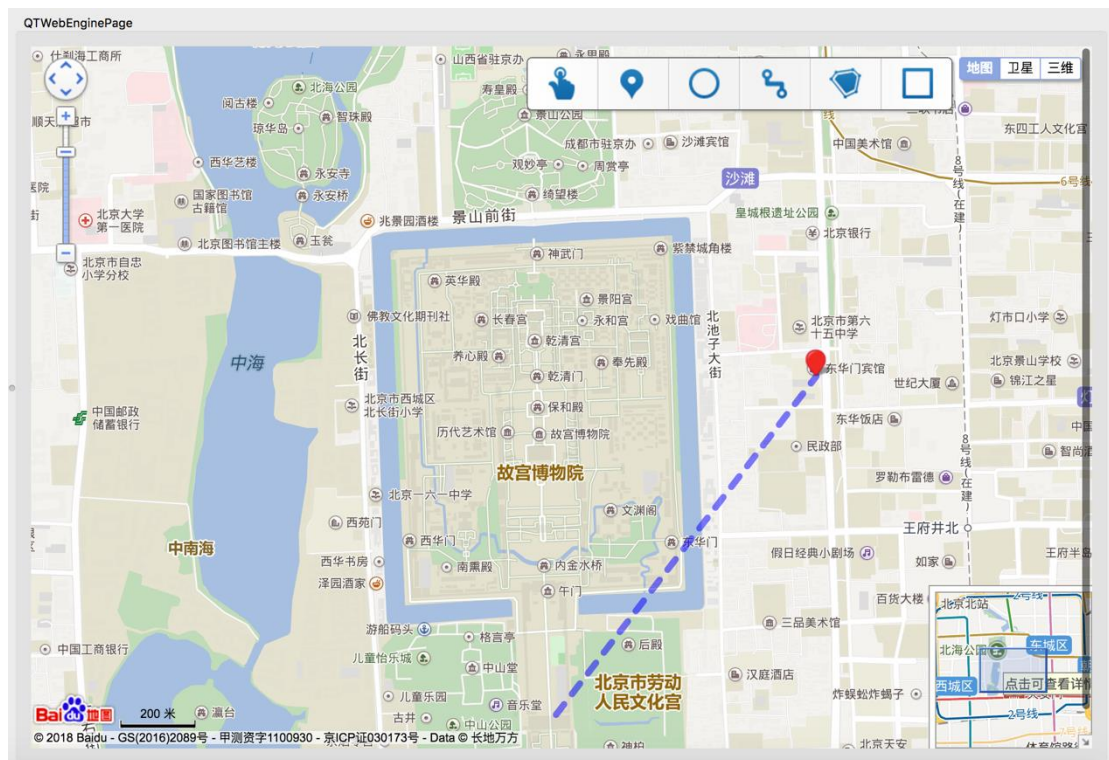


图 4.5 无人机航迹显示

4.5 本章小结

本章介绍了无人机地面控制站系统中的应用层的实现过程。为了测试验证无人机地面站软件，首先实现了一个模拟无人机状态的无人机代理。该无人机代理采用经纬度坐标表征无人机的位置，采用无人机的飞行速度、飞行高度、方向角

以及三个姿态角来表征无人机的状态。为了实现无人机代理计算无人机在某一状态下的位置坐标变化函数，研究了在地心坐标系中计算两个经纬度坐标之间的距离及方向角的方法以及近似优化。随后介绍了本项目中实现的无人机地面站操作台、无人机地面站虚拟仪表以及无人机航迹显示这三种依靠数据层的导航模块所提供的地理位置信息以及协议层所提供的传输协议所得到的应用。所实现的功能符合项目预期需求。

第5章 无人机航线规划

5.1 无人机航线规划总体介绍

无人机航线规划（也称为路径规划或航迹规划）是无人机地面控制站系统中一个较为重要的组成部分。当有任务需要无人机执行的时候，通常情况下首先在本地的地面站上为无人机根据已知的部分地理数据信息以及任务目标等进行任务规划，包括对无人机飞行的航线的规划以及对无人机所携带的载荷的控制情况进行规划等。这里我们暂不讨论对于载荷的控制与规划的问题，而主要讨论无人机航线的规划问题。

目前常见的一些无人机地面站软件中的路径规划模块较为简单，通常是按照逐次飞抵预先设置好的一系列航线点的方式，或者按照某些固定的飞行路线模式（如绕矩形飞、绕圆弧飞等）从而实现航线的规划，具体可以参考图 1.2 中所展示出的四款无人机地面控制站的规划界面及其效果。然而随着无人机所需要执行的任务越来越复杂，这种人为设定航迹点的较为直接的航线规划方式在可行性与实际效果上逐渐面临越来越多的挑战。由于无人机目前已经在军用以及民用上得到了较为广泛的应用，下面就分别给出无人机在军用以及民用上的应用场景从而说明通过一些算法来生成无人机的飞行航线是有意义的并且有直接需求的。

对于军用场景，随着现代化战争中各式无人装备的兴起，无人机作为无人装备中重要的组成部分在战场上担负大量侦查与部分精准打击等任务。在执行这些任务的时候，不仅仅需要控制无人机到达指定任务点，同时应当注意在飞行过程中对于敌方的雷达覆盖区域以及火力覆盖区域等威胁区进行最大程度的规避从而提升无人机的战场存活率。然而，由于无人机任务区地理环境信息较为复杂、已知的威胁区数量较多且拓扑关系复杂以及可能出现未知的突发威胁，传统的人为设置航迹点就会导致出现效率低下、不能充分规避威胁以及所规划航路在总长度上未必较优等问题。类似的，对于民用场景，考虑时下兴起的无人电商概念及模式，无人机可以充当商户与客户之间的递送链条中的商品载体。在这种情况下，我们考虑的问题是通过尽量短的路程以及较小的飞行总转向角（可以一定程度上降低无人机所携带载体的摆动）将商品顺利送到目的地。然而由于城市环境中楼宇建筑等障碍物数量众多位置关系复杂，当在线的无人机及任务数量较为庞

大的时候，为它们人为地设置飞行航迹点就会带来效率低下的问题。综上所述，对于这些越来越复杂的任务已经相应的任务应用场景，传统的人为设置航迹点的方式已经不再完全适用，需要通过一些智能算法来实现无人机的航线规划。

在本章中，我们将会讨论一些较为常用的无人机航线规划算法并在一些简单场景中进行测试。由于本项目侧重于实现一款无人机地面控制站，现阶段对于无人机航线规划算法的实际效果的要求并不是很高。我们会在“相关研究工作”这一小节中对于目前国内外在该方面的研究进行简要的总结，但一些最新的方法受限于数据与实验设备等暂时无法实现，留待下一步的研究工作中进一步完善。

5.2 相关研究工作

无人机航线规划作为机器人运动规划的其中一个课题，在近三十年来受到了广泛的关注并发展出了许多相关子课题以及相应的解决方法。

目前对于无人机航线规划的研究在对于无人机所遇到的威胁状况上有三种不同的认识。一部分研究更多的关注无人机自身的动力学等参数以及由此所带来的对于无人机的航线规划所引入的约束，此类研究通常不将研究重点放在无人机所需要面对的地理环境信息或者外界对于无人机的威胁而是专注于生成符合无人机实际飞行情况的可飞行路径。通过利用一些函数或曲线连接多个无人机航路点从而生成满足无人机实际飞行所需要满足的约束条件的路径。目前所采用的主要有 Dubins 曲线^[23,24]、B 样条曲线^[25]、PH 曲线^[26]以及 Sigmoid 函数^[27]等。一部分研究关注移动威胁，即任务不能在任务初始时就明确了解各种可能出现的威胁情况，随着无人机在执行任务飞行的过程中，其会随机遇到移动的未知威胁。解决该类问题主要采用的方法有基于卡尔曼滤波与优化控制 MPC 算法的方法^[28]和基于马尔科夫决策的方法^[29,30]。除了上述所介绍的研究之外，更多的研究将关注点放置于固定威胁与障碍物的情况下进行路径规划。

关于在固定威胁与障碍物的情况下进行机器人的运动规划（包括无人机的飞行路径规划、机器人在地面的运动路线规划以及机械手臂在复杂障碍物场景中的运动路线规划等），目前主要有三大类被广泛研究的方法，分别是：由 A* 算法所引出的一系列图搜索方法、由进化算法作为基础的一系列优化算法以及由 PRM 算法与 RRT 算法作为基础的一系列随机采样算法。这三大类方法各有优缺点，当然随着对于该问题的研究逐渐深入，近年来更多的研究工作已经不再局限于单独研究其中一类算法，而是将不同方向的算法进行结合从而得到一些在面对更加复

杂的情况的时候效果更好或者更加稳定的算法。

A*算法作为一种非常成熟的图搜索算法，在一些较早的图论中的最短路径问题中拥有很好的表现。在无人机航线规划问题中，由于 A*算法是一种离散的算法，通常需要将状态空间按照最小步长进行分割，或者采用一些图分割的方法如 Voronoi 图^[31, 32]或骨架提取^[33]等方法。之后就可以利用 A*算法启发式的遍历状态空间最终找到最优路径。目前较多采用 A*算法的方法都会对于得到的最优路径进行进一步的平滑处理从而使得所得到的结果更加符合无人机实际飞行是需要满足的一些动力学约束^[34, 35]。同时也有很多研究的关注点聚焦于对于 A*算法进行改进与优化，如 Basic Theta*，Phi*算法是对于 A*算法的修改，RSR 以及 JPS 算法是对于 A*算法的优化^[36]。除了在二维空间中，基于 A*算法的这一类图搜索算法在三维空间中同样可以适用^[37, 38]。但是上述所讨论的从 A*算法所发展而来的算法都是静态环境下对于无人机的路径进行规划。为了解决动态环境下的搜索，一方面 D*算法被提出用于解决该类问题^[39, 40]，另一方面一部分研究工作则关注于将 A*算法置于实时导航中，通过接受传感器所发回的数据信息进行即时路径重新规划^[41]。

如果不以全局搜索为出发点考虑无人机航线规划问题，一部分研究工作转而将该问题推向试图在整个状态空间中建立一个可以反映每个位置对于运动物体的适应性程度的函数映射。该函数映射被称为是人工势场 (Artificial Potential Field)^[42]。通过将无人机在状态空间中的路径规划问题转化为一个在线的控制决策问题，人工势场法可以解决 A*等搜索算法所带来的高时间成本的问题，但是其自身受到所建立的人工势场函数需要一些先验知识的影响，并不能保证所寻路径的评价指标较好，并且传统的人工势场法经常会面临陷入局部最优或者无法有效探寻状态空间的问题。因而目前较多工作通过人工势场法与一些如模拟退火^[43]等方法来实现跳出局部最优解，或者在出现局部最优解的地方构建一个虚拟障碍物^[44]来引导运动物体脱离局部最优位置。人工势场法的提出为运动物体在有障碍的情况下进行在线实时路径规划提供了一种不同的解决问题的视角，通过使用多种多样的势场函数、结合多种其他的优化算法以及采用先全局粗略规划后局部使用人工势场来细化平滑处理等方式可以得到效果较好的规划路线。

如果抛开希望对于该问题进行全局的理解与处理的出发点，运动物体路径规划问题事实上可以比较直接的对应于一个有约束的多目标优化问题。通过使用一系列的优化算法可以解决该问题。通过将每条可行路径进行编码并通过一些方式来获得新的路径并进行筛选，优化算法可以最终完备地得到一条至少是局部最优

的路径。目前通常使用的优化算法主要是各类群体进化算法及他们的改进变体，如遗传算法^[45-51]、粒子群算法^[52-55]、蚁群算法^[56-58]、人工蜂群算法^[59]、帝国竞争算法^[60]等等。目前很多工作并不单单使用单一一种优化算法，而是采用分层的思想，在静态层面使用 A*、稀疏 A* 等搜索方法构建粗略的全局航路而在动态层面使用各种进化算法来应对动态威胁的出现。

进一步的，为了能够在面对更加复杂的环境时更快的生成出可行路线，一部分研究工作转而考虑随机采样的路径规划方法。PRM (Probabilistic Roadmap)^[61]与 RRT (Rapidly-exploring random tree)^[62, 63]这两大类算法的提出标志着基于随机采样的算法在路径规划问题的研究中的兴起。随机采样通过对于配置空间的一个子集进行采样，并通过这些采样点生成一个轨迹。基于随机采样的算法适用于高维度的场景，但是缺点是不能直接生成符合无人机运动学约束的路径，需要对于直接得到的结果进行平滑操作来获得可实际飞行的路线。很多研究工作在上述两类基础的算法上发展出了新的随机采样路径规划算法。最早的一个对于 RRT 算法的改进是让 RRT 同时从初始起点以及目标终点生长最终让二者相连，即 RRT-connect 算法^[64]。为了可以直接改善生成轨迹的运动学效果，一些工作在建立搜索树或者构建路径图的时候，选择使用平滑曲线或者一些可用的运动基元^[65, 66]。此后一些进一步在最优性上对于该类算法进行改进的方法被提出，包括 RRT*^[67]，PRM*，RRG^[68]，Informed RRT*^[69]，BIT*^[70]以及 RABIT*^[71]等。

随着机器学习与深度学习的兴起于快速发展，一部分研究工作重新将目标转向利用强化学习来解决机器人在线路径决策与规划的问题。比较直接与基础的方法将该问题分拆为两层，即全局静态规划与局部动态规划。目前的工作采用 A* 与 Q 学习^[72]相结合或者 PRM 与强化学习^[73]相结合（即 PRM-RL）来解决动态或未知环境的无人机航线规划问题，可以取得较好的效果。其中 PRM-RL 方法是 Google Brain 在今年 ICRA 2018 上发表的工作，其可以实现在室内传感器嘈杂环境下长达 215 米的轨迹规划与飞行以及在一个 6.3 亿倍大于训练环境的真实环境中，完成了 1000 米以上的航空货物运输任务的飞行航线规划与实际飞行，同时并未破坏任务约束。在附录 A 中，对 PRM-RL 这篇文献进行了书面翻译。

由于机器人路径规划与机械手臂的运动规划在近年来获得了越来越多的关注，在这方面出现了一些非常优秀的开源算法库。如包含搜索相关算法的 SBPL (Search-Based Planning Lab) 以及包含随机采样相关算法的 OMPL (Open Motion Planning Library)^[74]。通过利用这些算法库可以实现各种路径规划算法并且可以获得三维可视化效果并部署于实际的机器人实体上。

5.3 本项目采用的航线规划方法

在本项目中由于航线规划只是其中一个模块且本项目的重点工作是完成一个无人机地面控制站系统，故而对于航线规划的部分暂时不进行较为深入的研究与实现，而是对于一些较为常见的航线规划算法进行实现从而为今后进一步的研究工作打下基础。在之后的 5 个小节中，我们会依次介绍基于搜索的 A* 算法，基于势能函数的人工势场算法，采用优化算法思想的进化算法家族中的遗传算法以及基于随机采样的 PRM 与 RRT 算法。这些算法基本涵盖了目前研究无人机航线规划的主流方法。本项目的算法实现部分参考该开源算法集^[75]。下面上述 5 个基本算法进行简要介绍。

5.3.1 A* 算法

A* 算法是一种较为传统并且常用的启发式搜索算法。无论是在图搜索中求解最短路还是在本章节中所讨论的路径规划问题中，A* 算法都是一种较为基本的方法。

A* 算法的主要思路是维护一个代表了所选择扩展的节点的优先级的优先队列。对于图上的所有节点，访问其的步骤为，首先访问到一个未被发现的节点并将其状态从未被发现更改为被发现但是未被扩展；其次访问到一个未被扩展的节点，将其临近的节点加入到被发现的列表中并将该扩展过的节点的状态修改为已被扩展，至此该节点的访问结束。多次迭代并且每次扩展评价指标最高的节点，直至抵达目标节点。

为了可以表征一个节点到目标节点的远近优先级，可以启发式的对于每个节点构建一个函数：

$$f(u) \equiv g(u) + h(u) \quad (5-1)$$

其中 $g(u)$ 表示节点在已扩展的节点所组成的图中距离初始节点的最短路径长度。 $h(u)$ 表示当前节点距离目标节点的启发式值且要求该值小于等于从当前节点到达最终目标节点的无碰撞距离。在最简单的情况下可以取 $h(u)$ 为当前位置节点到目标位置节点的欧氏距离。由于 $h(u)$ 必须保证在当前位置节点靠近目标位置节点的时候逐渐减小，故而每次迭代后选择的新的扩展的节点总会逐渐靠近目标节点，从而可以找到到达目标节点的最短可行路径。A* 算法是最优完备的，即如果存在一条最优路径则利用 A* 算法一定可以找到该最优路径。

在本项目中我们采用的邻接矩阵为：

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

其中 2 代表当前节点，1 代表可以从 2 扩展出的节点。不同的临界矩阵会带来不同的效果。如采用下述邻接矩阵可以减少搜索的次数，但是得到的路径在最优性上的表现较差：

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

其中 0 表示不能被扩展的点。而采用下述邻接矩阵得到的路径则会在最优性上具有优势但是相应的其在搜索空间的复杂程度上就更高一些：

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

5.3.2 人工势场

人工势场方法是一种考虑图中障碍物与目标点对于处于当前位置点物体的综合影响从而形成当前位置点的势能函数的方法。根据所得到的势能函数，可以利用梯度下降方法获得到达目标位置的较优路径。

人工势场法将目标点对于当前位置物体的吸引程度即引力势场理解为一个关于目标点到当前位置的距离的二次方的函数，即若记当前点到目标点的距离为 d ，引力势能系数为 ε ，则当前点的引力势能函数为：

$$U_1 = \frac{1}{2} \varepsilon d^2 \quad (5-2)$$

而类似于在物理中所学习到的势能与引力之间的关系，这里可以认为所谓的目标点对于当前位置物体的引力是当前位置的引力势能的梯度：

$$F_1 = -\nabla U_1 = \varepsilon d \quad (5-3)$$

类似的，对于环境中的障碍物，人工势场法认为其对于当前位置物体产生了排斥作用的斥力。与物理中所定义的斥力类似，该斥力与距离的二次方成反比，具体形式如下所示：

$$U_2 = \begin{cases} \frac{1}{2}\eta\left(\frac{1}{d}-\frac{1}{d_0}\right)^2 & d \leq d_0 \\ 0 & d > d_0 \end{cases} \quad (5-4)$$

其中 d 表示当前位置到障碍物的距离， d_0 表示障碍物的斥力所影响的范围， η 表示障碍物的斥力势能系数。可以看到通过限制障碍物斥力势能的影响范围可以避免过多障碍物对于同一位置产生过于复杂的斥力势能叠加且当距离达到一定程度时，由于斥力势能随着距离的增加而减少，故而对于较远处的位置的斥力影响近乎可以忽略。类似于上述从引力势能向引力转化的方式，通过计算斥力势能的梯度可以计算得到障碍物对于某一位置的斥力矢量：

$$F_2 = -\nabla U_2 = \begin{cases} \eta\left(\frac{1}{d}-\frac{1}{d_0}\right)\frac{1}{d^2}\nabla d & d \leq d_0 \\ 0 & d > d_0 \end{cases} \quad (5-5)$$

通过上述建立的引力与斥力的模型，可以采用引力斥力叠加的方式得到当前位置的所受到的合力方向从而按照预先所设定的步长向该方向移动；也可以采用计算总能量的方式计算引力势能与斥力势能的叠加从而最终得到该图中任意位置在人工势场中所对应的总能量函数并相应的按照梯度下降的方法来计算势能下降最快的方向，而该方向所形成的路线就是我们所求的一种最优化路径。

5.3.3 遗传算法

前述两种方法主要考虑的是直接对于障碍物或者优化目标进行建模从而利用搜索的方式或者建模确定每一个位置点在整体环境中所获得的评价。然而我们可以将无人机的航线规划问题从另一个角度看做一个多约束的优化问题。这些约束条件可能包括但不限于：无人机航线与障碍物的碰撞规避，无人机自身的动力学属性（姿态角、水平转弯角以及飞行高度等）等。而优化的目标通常包括无人机航线的总长度以及所受到的总威胁，即我们总是希望无人机可以在尽可能短的航

线中尽可能的避开威胁区。由此无人机航线规划便可以转变为一个优化问题。

优化算法囊括了很多算法,包括但不限于遗传算法、粒子群算法、蚁群算法、人工免疫网络以及这些算法的各种变种。这些算法的基本思想是相同的,即通过将解进行编码并构造适应性函数,通过对于初始的解集进行某种方式的部分保留与部分更新从而逐渐向最优解靠拢直至收敛。这里选取其中的遗传算法进行实现。

遗传算法的基本思想是,对于初始的种群(即初始的可行解集),按照适应度采用轮盘赌等方式进行筛选,之后采用交叉变异的方法更新当前种群,并再次根据适应度进行筛选。随着迭代次数的增加,所得到的种群逐渐向最优解趋近。

对于无人机航线规划问题,我们所面对的问题是如何将在一个图中寻找最短路的问题抽象化为一个最优化的实际问题。我们可以假设所有的可行解具有相同的点的数目。这样起点终点与这些编码的点所生成的序列就可以构成一个合格的可行解。为了让上述航线避免与障碍物或者威胁区进行碰撞,可以为由前后两个点所连接线段与障碍物进行碰撞所形成的碰撞事件定义一个很大的与路径在障碍物内部的长度成正比的惩罚函数。而适应度函数则定义为路径总长度。

5.3.4 PRM

在讨论过上述进化算法后,可以发现,进化算法往往需要耗费大量的时间进行运算,同时为了获得更好的结果,若采用较为精细复杂的编码方式则会耗费较多的计算资源。通过基于随机采样的方法,可以快速的在一定程度上获得近似最优解的路径规划方法,同时对于实时出现的动态未知障碍拥有较好的避障能力。

PRM (Probabilistic Roadmap, 随机路径图) 算法是一种基于随机采样的路径规划算法,其基本思想非常简单。

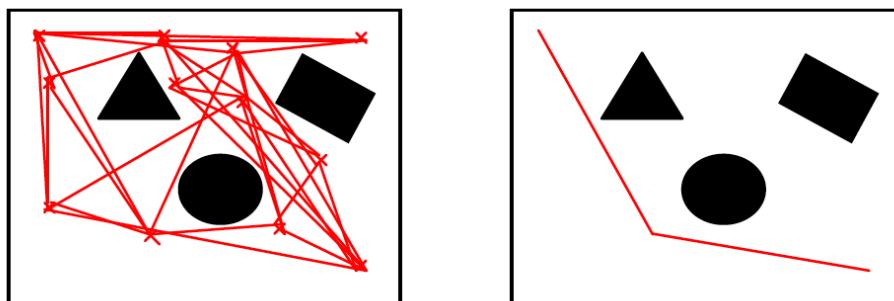


图 5.1 PRM 算法示意图

如图 5.1 所示,可以看到 PRM 的基本思路是在图中随机生成一些与障碍物不

发生碰撞的点。在本项目中采用最基本的一种构建图的方式，即对于所有获得的点，将它们两两连线，如果连线不与障碍物发生碰撞则将该连线作为一条边加入到图中。最终可以得到如图 5.1 左侧所示的一个图模型。对于该图模型只需要利用一些最短路算法即可得到在该图模型中对应的最短路。这里采用之前所介绍过的 A*算法进行实现。其中边的权值由其欧式距离确定，而节点启发式函数由其到达目标点的欧式距离所确定。最终得到的结果如图 5.1 右侧所示。

事实上，上述对于 PRM 的简化实现在追求更高的路线最优性的时候依旧会遇到很多麻烦。为了获得更优的路线，最直接的方式是在 PRM 建图的时候增加点的个数。但是由于增加了采样点的个数所带来的影响是 PRM 所构建的路径图的边数会呈 N 平方的方式发生增加，势必会导致所构建的图较为复杂。因而改进方法为对于每个采样点，其只与周围一定距离范围内的最近 K 个点相连。同时还需要考虑对于某些没有被覆盖到的区域进行二次撒点。此时为了表征每个节点附近是否还需要二次撒点，可以采用一些启发式函数。这些启发式函数应当满足与该节点最近的相邻点的距离成反比或与该节点周围一定距离内的相邻点的个数成反比或与该节点周围与其没有相连的节点中距离最近者的距离等。通过上述这些方式可以对本项目中目前所采用的简单的 PRM 算法进行改进。

5.3.5 RRT

与前述的 PRM 算法类似，RRT（Rapidly Exploring Random Tree，快速扩展随机树）也是一种基于随机采样的航线路径规划方法。

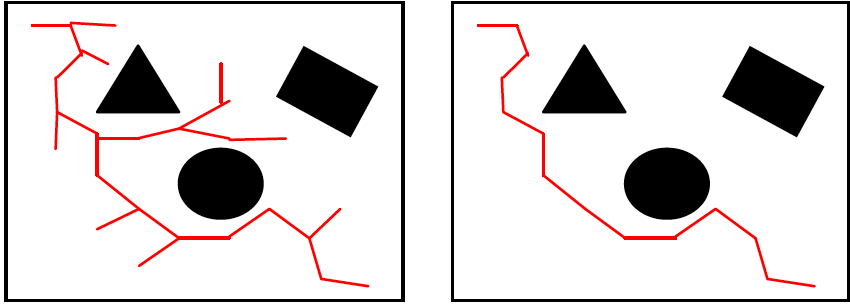


图 5.2 RRT 算法示意图

如图 5.2 所示，RRT 是一种生长算法，在状态空间中通过逐渐生长出一系列节点从而形成一棵树并通过这棵树来一定程度上伸展探索整个状态空间。RRT 的生长方法为，从初始节点开始，每次向某个随机方向生长，通过选择当前树中离

该生长方向最近的节点，将其向该随机生长方向向前扩展一个较小的确定距离，从而实现在状态空间中的搜索。在生长的过程中，一旦出现与障碍物发生碰撞的情况，则放弃扩展。为了使得 RRT 可以逐渐向目标位置逼近，可以采用每次以一定概率向目标位置方向进行扩展的方式。如果当前扩展节点到达目标位置附近的一个阈值范围内则算法终止。最终经过多次迭代，RRT 可以获得一条从初始起点位置到目标位置的路径，从而实现无人机的航线规划。

5.4 实验结果分析

通过对于上述 5 种算法的分析与实现，在两种不同的应用场景中对其进行了实验，结果如下所示。其中所用到的两种实验场景分别如图 5.3 中左右两幅图为例。这里采用的是位图的形式来表示地理环境信息。位图由像素点组成，其中白色点（灰度值为 255）为可以飞行的区域，而黑色点（灰度值为 0）为障碍物或威胁区，是不可飞行的区域。

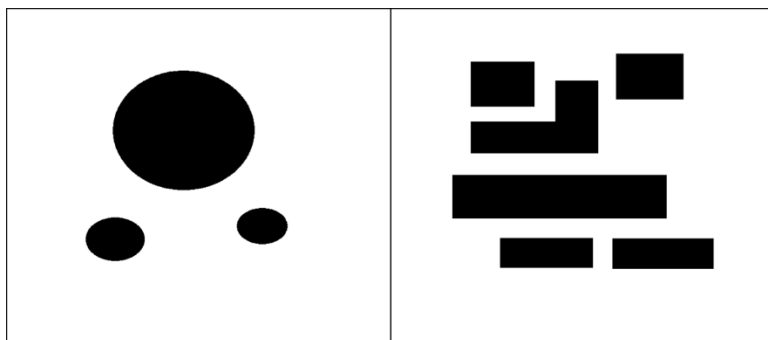


图 5.3 威胁区与障碍物情况

对于图 5.3 中的左图，其是对于无人机执行侦查任务时候所需要面对的威胁区的情况所进行的建模。无人机需要从初始起点绕过上述威胁区并最终达到目的地。而对于图 5.3 中的右图，其是对于无人机执行城市密集环境下的载荷递送等任务是城市建筑物等障碍物的情况所进行的建模。

利用前一小节所介绍的 5 中算法分别在这两种情况下进行测试，得到的结果如图 5.4 及图 5.5 所示。其中图 5.4 从左向右依次展示了在场景 1 中采用 A*、人工势场、遗传算法、PRM 算法以及 RRT 算法所得到的航线规划结果；图 5.5 从左向右依次展示了在场景 2 中采用 A*、人工势场、遗传算法、PRM 算法以及 RRT 算法所得到的航线规划结果。

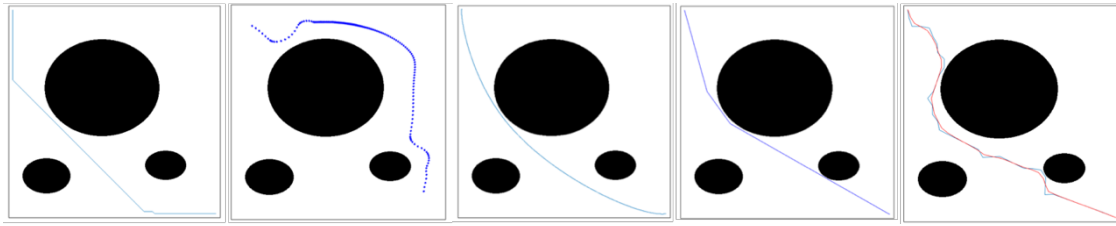


图 5.4 情况 1 威胁区最优路径各算法求解结果

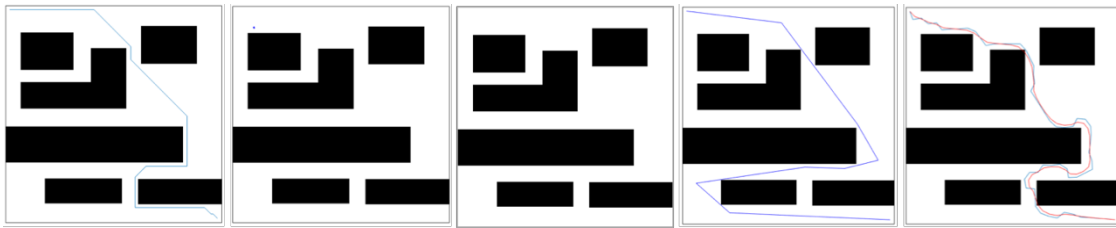


图 5.5 情况 2 障碍物最优路径各算法求解结果

为了进一步对于上述航线规划结果进行量化分析，在上述两个场景中通过多次实验对于前述 5 种算法的计算用时以及所得到的航线规划的总长度进行测量，得到的平均值如表 5.1 所示。其中用时单位为秒，路径长度单位为像素。

表 5.1 无人机航线规划算法用时及路径长度表

	A*	人工势场	GA	PRM	RRT
第 1 组用时	327.68	5.19	26.33	1.21	2.65
第 1 组路径长度	775.48	792.72	848	745.81	829.56
第 2 组用时	344.87	NaN	NaN	0.75	9.80
第 2 组路径长度	1044.77	NaN	NaN	1623.85	1217.96

可以看到上述 5 种算法总体差异较大。在图 5.4 的情形 1 中，我们可以看到，5 种算法基本都能给出类似的路径选择，只不过 A*、PRM 以及 RRT 所给出的路径曲线较为陡峭，而遗传算法与人工势场所给出的路径结果较为平滑。实际上对于上述路径存在折线点过多所导致的无人机实际情况下不易按照所规划出的航线飞行的问题，可以采用一些平滑的处理来解决。如采用选取相邻两段子路径各自中点并在垂点附近寻求第三个点构成三次样条曲线，利用 SQP 序列二次规划在满足三次样条每处的曲率半径不小于无人机的转弯半径的条件下对于最短距离求解最优解。这里按照这种思路对于 RRT 中所得到的最初结果（如蓝线所示）进

行了平滑，得到了平滑后的 RRT 路径（如红线所示）可以看到所得到的平滑路径较之前的多转向折线更加平滑，适合无人机飞行。

然而对于图 5.5 所示的情况 2，可以看到遗传算法与人工势场并不能寻找到合适的可行路径。对于人工势场而言，这是由于初始情况距离障碍物过于接近以及周围的障碍物数目太多导致远处的目标点对该初始位置的引力并不能克服其周围的障碍物对其的影响，导致其陷入了局部最优，即无论再向哪个方向前进，所受到的斥力都会大于该局部最优位置，这就导致了人工势场方法在这种情况下无法按照我们预期的设想完成任务。进一步的，在实验的过程中还发现，人工势场法可能由于受到斥力的影响而在向目标点靠拢的过程中被推出边界，亦无法完成任务。对于遗传算法而言，这是由于在类城市密集环境下，障碍物（建筑物）之间的复杂拓扑位置关系导致可行解在横向或纵向坐标轴上并不能按照单调递增的方式进行扩展，而遗传算法建立编码通常需要对于某种坐标轴按照单调递增的方式来确定编码集，这种复杂的路径为遗传算法带来了挑战。而对于 A*、PRM 以及 RRT，可以看到它们在这种类城市密集环境下依旧可以顺利完成任务。

进一步的，根据表 5.1 我们可以看到，在计算时间这个评价指标中，A*算法与遗传算法所耗费的时间远远大于其余三种算法。可以进一步遇见到，在更加复杂的障碍物情况下，如果我们希望可以快速甚至实时获得当前位置到某一目标位置的航线规划结果，单纯的 A*算法与遗传算法是不能胜任的。而相对比之下，我们可以看到，利用基于随机采样的算法，可以较快的得到可行路线并且该路线在路径长度的评价指标上并不会比其余三种方法逊色很多。对于算法的稳定性而言，A*算法一定能找到一条最优解；而基于随机采样的 PRM 与 RRT 虽然不能保证每次都能找到最优解，但是多得到的解也是可以接受的；而遗传算法与人工势场在一些问题上会发生无法得到可行解的问题，这也就以为着这两种算法的稳定较差。综合考虑，在无人机航线规划的问题上，基于随机采样的航线规划算法表现较好，通过对 PRM 以及 RRT 增加一些满足无人机自身属性的约束的平滑步骤，可以较快较好的给出无人机可用的航线规划路线。

5.5 本章小结

在本章中，我们首先介绍了无人机航线规划的问题。接着我们阐述了目前大多数无人机地面控制站所采用的通过人为设定航迹点的方式实现无人机的航线规划所可能遇到的问题以及我们的项目中融入各种航线规划算法的意义。进而我

们对于目前在研究无人机或机器人的运动规划问题上所使用的主要方法，包括基于图搜索的 A* 等算法，基于人工势场的一系列算法，基于进化算法的一系列算法，基于随机采样的一系列算法以及采用深度学习强化学习的算法。通过对于相关研究工作的调研，我们明确了在我们项目中可以采用的一系列无人机航线规划算法。接下来我们依次介绍 A* 算法、人工势场算法、遗传算法、PRM 算法以及 RRT 算法的原理以及在该问题中的应用方式。在这些介绍的基础上，我们简单实现了上述 5 种算法并对它们的性能进行了比较。由于时间精力有限，该部分暂时还未能整合接入到本项目的地面站控制站中，在下一阶段将主要实现该部分功能。

第6章 总结与展望

本文给出了一种实现一款轻量级的通用无人机地面站的方法。该方法将无人机地面站系统进行分层化与模块化解析从而有助于进一步的版本迭代与升级。

事实上，作为模块化设计的一部分，无人机地面站的三层结构中每一层都可以改进与增加设计。如协议层中 UDP 通信方式，其虽然实现简单但是当网络环境不好的时候容易产生堵塞的问题。同时目前所考虑的数据段内容比较简单，在真实操作的时候还应当增加考虑如舵机转速、电池电量等需要在实际使用该软件的额时候同样重要的参数。而参数的增加也会给 UDP 通信带来压力。故而需要一种更好的传输协议。数据层应当可以引入更多的数据源。譬如，对于导航模块亦即地图与 GIS 模块，可以增加更多可选模块。由于通常使用的 Google 地图以及必应地图均提供与百度地图非常相似的 JavaScript API，应当可以进一步支持对于所使用的地图源的选择。对于 GIS 软件而言，很多都支持地图瓦片导出为某些形式的文件，如 ArcGIS 支持导出为 Shapefile，而一些二三维建模软件如 Sketchup 支持导入这种格式的地理信息数据文件，故而是一个设想是数据层应当嵌入一个二三维建模软件，支持读取各种 GIS 所生成的地理信息数据文件从而实现模块化。对于数据库部分而言，目前项目中所使用的是非常简单轻量级的 SQLite 数据库。这种数据库对于目前所面对的“单站单机”架构是可以满足需求的。然而若希望系统可以支持介入更多的无人机或者多个“地面站-无人机”组协同完成任务，则需要考虑使用分布式数据库以及数据分发服务 DDS (Data Distribution Service)。而在应用层中则有更多内容可以拓展实现。如使用 FlightGear 来模拟无人机飞行状态或者在 Unity3D 中建模无人机及其周边环境后引入 VR 来提升地面站软件的人机交互能力。遗憾的是，精力与时间有限，这些内容只能留在项目的迭代与升级中实现了。

无人机地面控制站作为一个与实际应用紧密结合的课题其所包含的内容远不止于本工作目前所介绍的部分。希望在今后的工作中，可以进一步实现与真实无人机飞控的对接，从而可以让本项目所实现的无人机地面控制站真正控制实际的无人机进行飞行并且进一步在真实环境中测试验证本项目中目前所实现使用的航线规划算法。

插图索引

图 1.1	美国通用原子公司 Advanced Cockpit GCS.....	2
图 1.2	几款常用的产品级无人机地面站软件	3
图 1.3	无人机地面站系统结构	4
图 1.4	无人机地面站功能结构	5
图 2.1	Socket 在 OSI 七层模型中的位置	9
图 2.2	UDP 三种通信形式示意图	10
图 2.3	地面站向无人机所发指令的 UDP 数据报数据段内容	11
图 2.4	无人机向地面站所发信息的 UDP 数据报数据段内容	12
图 3.1	Qt 主程序中地图控件模块效果图	16
图 3.2	在地图控件中增加威胁区效果图	16
图 3.3	Qt WebEngine 模块架构.....	17
图 3.4	Qt WebEngine Widget 模块内子模块架构	18
图 3.5	本项目实现在 Qt 窗体中嵌入一个 Web 页面并与之通信的 UML 类图 ..	19
图 4.1	无人机姿态角同大地坐标系之间的关系	24
图 4.2	WGS 坐标系示意图	27
图 4.3	无人机地面控制站操纵台	30
图 4.4	无人机地面控制站虚拟仪表	31
图 4.5	无人机航迹显示	32
图 5.1	PRM 算法示意图.....	41
图 5.2	RRT 算法示意图.....	42
图 5.3	威胁区与障碍物情况	43

图 5.4	情况 1 威胁区最优路径各算法求解结果	44
图 5.5	情况 2 障碍物最优路径各算法求解结果	44

表格索引

表 3.1	UAVData 数据表中的字段定义.....	20
表 4.1	无人机代理参数表	29
表 5.1	无人机航线规划算法用时及路径长度表	44

参考文献

- [1] PAULGERINF AHLSTROM, THOMASJAMESGLEASON, 法尔斯特伦, et al. 无人机系统导论 [M]. 国防工业出版社, 2015.
- [2] 马聪颖, 高瑞周, 朱玉祜. 无人机地面控制站通用化软件架构 [J]. 航空计算技术, 2013, 43(3): 112-3.
- [3] General Atomics Aeronautical Systems, Inc. Advanced Cockpit GCS[EB/OL]. [2015-02-19]. <http://www.ga-asi.com/advanced-cockpit-gcs>.
- [4] KANG Y, YUAN M. Software design for mini-type ground control station of UAV; proceedings of the International Conference on Electronic Measurement & Instruments, F, 2009 [C].
- [5] 刘科. 无人机通用地面站软件的设计与实现 [D]; 南昌航空大学, 2013.
- [6] 刘龔. 无人机地面控制站的设计与开发 [D]; 南京航空航天大学, 2013.
- [7] 张治生. 无人机地面站系统设计与开发 [D]; 西北工业大学, 2007.
- [8] 王鑫, 邓宏彬. 基于 WPF 技术的无人机地面指挥控制站的设计与实现; proceedings of the 全国工业控制计算机技术年会, F, 2014 [C].
- [9] 杨青, 赵锋, 李阳. 基于 C#的无人机地面站软件设计 [J]. 电子质量, 2017, 5): 48-51.
- [10] 李帅, 范项媛. 基于 Qt 的无人机地面站软件系统的设计 [J]. 雷达科学与技术, 2017, 15(4): 410-4.
- [11] 贾杰, 刘智腾. 基于 Qt 的小型无人机地面站软件设计与实现 [J]. 计算机与现代化, 2015, 7): 94-7.
- [12] 于琰平. 基于 FlightGear 的四旋翼无人机三维可视仿真系统研究 [D]; 天津大学, 2010.
- [13] 韩景壮, 厉明, 周凌, et al. 无人机视景仿真平台设计与实现 [J]. 计算机测量与控制, 2014, 22(5): 1527-9.
- [14] 乔志华, 李一波, 康绍鹏, et al. 基于虚拟现实技术的无人机地面控制站系统设计; proceedings of the '2008 系统仿真技术及其应用学术会议论文集, F, 2008 [C].
- [15] 孙雨, 裴海龙. 小型无人直升机通信系统的设计与实现 [J]. 计算机测量与控制, 2011, 19(4): 954-7.
- [16] 谢相博, 徐光辉, 范凯鑫, et al. 基于 4G 的无人机远程巡逻系统 [J]. 通信技术, 2015, 48(11): 1305-9.
- [17] 大漠. 世界首例无人机 5G 技术测试在无锡完美收官 [J]. 航空制造技术, 2016, 18): 14-.

- [18] 方国军, 王海涛, 罗秋凤, et al. 某小型无人机飞控系统串行通信程序的设计 [J]. 计算机测量与控制, 2007, 15(12): 1850-1.
- [19] 马少瑛, 乌仁格日乐. 基于 CSerialPortEx 类的无人机地面站串口通信的实现 [J]. 电脑知识与技术, 2011, 07(22): 5312-4.
- [20] 何湘智, 王荣春, 周伟. 基于 VC~(++)和 TRTD 的无人机地面站软件开发 [J]. 现代电子技术, 2010, 33(22): 46-8.
- [21] REGAUSTIN, 奥斯汀, 陈自力, et al. 无人机系统:设计、开发与应用 [M]. 国防工业出版社, 2013.
- [22] Motion Imagery Standards Board (MISB). UAS Datalink Local Set[EB/OL]. [2018-02-22]. <http://www.gwg.nga.mil/misb/docs/standards/ST0601.12.pdf>.
- [23] 关震宇, 杨东晓, 李杰, et al. 基于 Dubins 路径的无人机避障规划算法 [J]. 北京理工大学学报, 2014, 34(6): 570-5.
- [24] 吴克风, 曹晓文, 周其忠, et al. 任意距离下 Dubins 最短路径研究 [J]. 战术导弹技术, 2017, 1): 76-84.
- [25] FOO J L, KNUTZON J, KALIVARAPU V, et al. Path Planning of Unmanned Aerial Vehicles using B-Splines and Particle Swarm Optimization [J]. Journal of Aerospace Computing Information & Communication, 2009, 6(4): 271-90.
- [26] 王怿, 祝小平, 周洲. 基于 PH 曲线的无人机路径规划算法 [J]. 计算机仿真, 2013, 30(3): 76-9.
- [27] 陈雯雯, 刘明, 雷建和, et al. 基于 Sigmoid 函数的四旋翼无人机轨迹规划算法 [J]. 控制工程, 2016, 23(6): 922-7.
- [28] 任佳, 高晓光, 张艳. 移动威胁情况下的无人机路径规划 [J]. 控制理论与应用, 2010, 27(5): 641-7.
- [29] AL-SABBAN W H, GONZALEZ L F, SMITH R N. Wind-energy based path planning for Unmanned Aerial Vehicles using Markov Decision Processes; proceedings of the IEEE International Conference on Robotics and Automation, F, 2012 [C].
- [30] RAGI S, CHONG E K P. UAV Path Planning in a Dynamic Environment via Partially Observable Markov Decision Process [J]. IEEE Transactions on Aerospace & Electronic Systems, 2013, 49(4): 2397-412.
- [31] BHATTACHARYA P, GAVRILOVA M L. Roadmap-Based Path Planning - Using the Voronoi Diagram for a Clearance-Based Shortest Path [J]. Robotics & Automation Magazine IEEE, 2008, 15(2): 58-66.
- [32] 单提晓, 蒋蓁. 一种基于分割法的无人机路径规划新方法 [J]. 弹箭与制导学报, 2014, 34(2): 185-8.

- [33] 袁操, 周德云, 张堃. 一种骨架提取的无人机航迹规划法 [J]. 电光与控制, 2012, 19(6): 5-7.
- [34] 符小卫, 高晓光. 一种无人机路径规划算法研究 [J]. 系统仿真学报, 2004, 16(1): 20-1.
- [35] 姚雨, 李庆, 陈曦. 优化的 A*算法在航迹规划上的应用 [J]. 微电子学与计算机, 2017, 34(7): 51-5.
- [36] DUCHOŇ F, BABINEC A, KAJAN M, et al. Path Planning with Modified a Star Algorithm for a Mobile Robot ☆ [J]. Procedia Engineering, 2014, 96(96): 59-69.
- [37] QI Z, SHAO Z, PING Y S, et al. An Improved Heuristic Algorithm for UAV Path Planning in 3D Environment; proceedings of the International Conference on Intelligent Human-Machine Systems and Cybernetics, F, 2010 [C].
- [38] REN T, ZHOU R, XIA J, et al. Three-dimensional path planning of UAV based on an improved A* algorithm*; proceedings of the Guidance, Navigation and Control Conference, F, 2017 [C].
- [39] STENTZ A. Optimal and Efficient Path Planning for Partially Known Environments; proceedings of the IEEE International Conference on Robotics and Automation, 1994 Proceedings, F, 2002 [C].
- [40] STENTZ A. The focussed D* algorithm for real-time replanning; proceedings of the International Joint Conference on Artificial Intelligence, F, 1995 [C].
- [41] MACALLISTER B, BUTZKE J, KUSHLEYEV A, et al. Path planning for non-circular micro aerial vehicles in constrained environments; proceedings of the IEEE International Conference on Robotics and Automation, F, 2013 [C].
- [42] KHATIB O. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots [M]. Springer New York, 1986.
- [43] 王强, 张安, 吴忠杰. 改进人工势场法与模拟退火算法的无人机航路规划 [J]. 火力与指挥控制, 2014, 8): 70-3.
- [44] MIN C L, MIN G P. Artificial potential field based path planning for mobile robots using a virtual obstacle concept; proceedings of the Ieee/asme International Conference on Advanced Intelligent Mechatronics, 2003 Aim 2003 Proceedings, F, 2003 [C].
- [45] GEMEINDER M, GERKE M. GA-based path planning for mobile robot systems employing an active search algorithm [J]. Applied Soft Computing, 2003, 3(2): 149-58.
- [46] MOHANTA J C, PARHI D R, PATEL S K. Path planning strategy for autonomous mobile robot navigation using Petri-GA optimisation [J]. Computers & Electrical Engineering, 2011, 37(6): 1058-70.
- [47] WANG Y, CHEN Z. Genetic Algorithms(GA) Based Flight Path Planning with Constraints [J]. Journal of Beijing University of Aeronautics & Astronautics, 1999, 25(3): 355-8.

- [48] OZ I, TOPCUOGLU H R, ERMIS M. A meta-heuristic based three-dimensional path planning environment for unmanned aerial vehicles [M]. Society for Computer Simulation International, 2013.
- [49] 周青, 张锐, 索晓杰, et al. 具有时间约束的无人机遗传算法航迹规划 [J]. 航空计算技术, 2016, 46(2): 93-6.
- [50] 李楠, 张建华. 基于改进遗传算法的无人机航路规划 [J]. 计算机仿真, 2016, 33(4): 91-4.
- [51] 马云红, 周德云. 一种无人机路径规划的混沌遗传算法 [J]. 西北工业大学学报, 2006, 24(4): 468-71.
- [52] MASEHIAN E, SEDIGHIZADEH D. A multi-objective PSO-based algorithm for robot path planning; proceedings of the IEEE International Conference on Industrial Technology, F, 2010 [C].
- [53] WANG Q, ZHANG A, QI L. Three-Dimensional Path Planning for UAV Based on Improved PSO Algorithm; proceedings of the Control and Decision Conference, F, 2014 [C].
- [54] 韩超, 王赢. 一种基于改进 PSO 的无人机航路规划方法 [J]. 舰船电子工程, 2014, 34(4): 49-53.
- [55] 方群, 徐青. 基于改进粒子群算法的无人机三维航迹规划 [J]. 西北工业大学学报, 2017, 35(1): 66-73.
- [56] 尚梦雨. 无人机实时蚁群算法路径规划 [J]. 自动化应用, 2016, 12): 61-3.
- [57] 程琪, 荆涛, 于志游. 利用三次样条改进蚁群算法的无人机航路规划 [J]. 计算机测量与控制, 2016, 24(8): 272-4.
- [58] FAN X, LUO X, YI S, et al. Optimal Path Planning for Mobile Robots Based on Intensified Ant Colony Optimization Algorithm; proceedings of the IEEE International Conference on Robotics, Intelligent Systems and Signal Processing, 2003 Proceedings, F, 2003 [C].
- [59] LI B, GONG L G, YANG W L. An Improved Artificial Bee Colony Algorithm Based on Balance-Evolution Strategy for Unmanned Combat Aerial Vehicle Path Planning [J]. The Scientific World Journal, 2014, 2014(1): 95-104.
- [60] 叶春, 高浩. 帝国竞争算法在无人机三维航线规划中的应用研究 [J]. 测控技术, 2017, 36(11): 98-101.
- [61] KAVRAKI L E, LATOMBE J C, OVERMARS M H. Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces; proceedings of the IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, F, 1996 [C].
- [62] LAVALLE S. Rapidly-exploring random trees : a new tool for path planning [J]. Research Report, 1998, 293--308.
- [63] LAVALLE S M, KUFFNER J J. Randomized kinodynamic planning; proceedings of the IEEE International Conference on Robotics and Automation, 1999 Proceedings, F, 2002 [C].

- [64] KUFFNER J J, LAVALLE S M. RRT-connect: An efficient approach to single-query path planning; proceedings of the IEEE International Conference on Robotics and Automation, 2000 Proceedings ICRA, F, 2002 [C].
- [65] FRAZZOLI E, DAHLEH M A, FERON E. Maneuver-based motion planning for nonlinear systems with symmetries [J]. IEEE Transactions on Robotics, 2005, 21(6): 1077-91.
- [66] NETO A A, MACHARET D G, CAMPOS M F M. Feasible RRT-based path planning using seventh order Bézier curves; proceedings of the Ieee/rsj International Conference on Intelligent Robots and Systems, F, 2010 [C].
- [67] KARAMAN S, WALTER M R, PEREZ A, et al. Anytime Motion Planning using the RRT*; proceedings of the IEEE International Conference on Robotics and Automation, F, 2011 [C].
- [68] KARAMAN S, FRAZZOLI E. Sampling-based algorithms for optimal motion planning [M]. Sage Publications, Inc., 2011.
- [69] GAMMELL J D, SRINIVASA S S, BARFOOT T D. Informed RRT*: Optimal Sampling-based Path Planning Focused via Direct Sampling of an Admissible Ellipsoidal Heuristic [J]. 2014, 7(4): 2997-3004.
- [70] GAMMELL J D, SRINIVASA S S, BARFOOT T D. Batch Informed Trees (BIT*): Sampling-based Optimal Planning via the Heuristically Guided Search of Implicit Random Geometric Graphs [J]. 2014, 2015(3067-74.
- [71] CHOUDHURY S, GAMMELL J D, BARFOOT T D, et al. Regionally accelerated batch informed trees (RABIT*): A framework to integrate local information into optimal path planning; proceedings of the IEEE International Conference on Robotics and Automation, F, 2016 [C].
- [72] PANG T, RUAN X, WANG E, et al. Based on A* and Q-Learning Search and Rescue Robot Navigation [J]. Telkomnika Indonesian Journal of Electrical Engineering, 2012, 10(7): 1889-96.
- [73] FAUST A, RAMIREZ O, FISER M, et al. PRM-RL: Long-range Robotic Navigation Tasks by Combining Reinforcement Learning and Sampling-based Planning [J]. 2017,
- [74] SUCAN I A, MOLL M, KAVRAKI L E. The Open Motion Planning Library [J]. IEEE Robotics & Automation Magazine, 2012, 19(4): 72-82.
- [75] RAHUL KALA. Codes for Robot Path Planning[EB/OL]. [2014-06-07]. <http://rkala.in/codes.php>.

致 谢

随着本篇文章步入尾声，也就意味着前前后后一个多学期的毕业设计工作即将告一段落。回望过去的半年时间，在我学习相关领域知识与完成毕业设计的过程中，身边的师长与亲人给了我许多的帮助与鼓励。

首先感谢我的导师朱纪洪教授。感谢朱老师给了我这个题目，让我可以对于无人机系统以及无人机地面站软件进行了一个系统的并且较为深入的研究，并在此基础上进行了软件开发。同时也感谢朱老师在我完成毕业设计的过程中对我的指导、鞭策与鼓励。其次感谢在我完成毕设过程中给予我研究方向以及技术选型方向上的指导与建议的诸多学长与前辈。他们是：华如科技技术总监张柯先生，航天三院三部副主任孙鑫先生、郭洪文先生以及张王成先生。当然也要感谢我的家人。在论文的一些细节的地方，父亲给了我很多建议。在去航天三院进行多次技术交流时，一直是母亲保障我的出行。

最后我还想感谢一下清华这个园子，因为在这个园子里，有许许多多优秀的人在激励着我不断前进。不只是单纯的“自强不息，厚德载物”，而是于点滴小事中告诉我如何成为一个更优秀的人。也许多年以后，我不会再记得那些刷过的夜里调不通的代码、看不完的课本以及写不完的大作业，但是我会永远记得这四年是一段像阳光一般明亮的时光。

声 明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签 名： 王 澎 日 期： 2018/06/25

附录 A 外文资料的书面翻译

PRM-RL：强化学习和基于采样的路径规划方法结合的远距离机器人导航任务

摘要——我们提出了一种解决远距离导航任务的分层方法：**PRM-RL**，它结合了基于采样的路径规划和强化学习（**RL**）。**RL** 代理可以学习短程、点对点的导航策略，在没有大规模拓扑结构知识的情况下可以捕获机器人的动态情况和任务约束。接着，基于采样的规划方式可以提供可连接机器人配置的路线图，**RL** 代理可以成功的由这些机器人配置导航。在控制机器人规划和支持远距离导航的问题上，我们使用了相同的 **RL** 代理。我们使用了基于概率路线图（**PRMs**）做基于采样的路径规划方法。这个 **RL** 代理是基于特征和深度神经网络的方法在连续状态和执行空间所建造的。我们同时通过仿真和真机评估测试来 **PRM-RL** 方法，在两个导航任务中使用非平凡的机器人动力学：在办公环境中的端到端的差动驱动的室内导航，以及在城市环境中，具有负载位移约束的空中货物运输。结果显示 基于采样的规划方法展示了其相对于单独使用 **RL** 代理和采用其他传统基于采样的规划方法的优越性。在室内导航任务中，**PRM-RL** 成功的完成了嘈杂传感器环境下长达 215 米的轨迹；在航空货物运输任务中，**PRM-RL** 在一个 6.3 亿倍大于训练环境的真实环境中，完成了 1000 米以上的飞行航线，同时并未破坏任务约束。

A.1 介绍

远距离导航任务需要机器人在满足任务约束的同时，安全移动相当长的距离。例如（图 A.1a），室内导航需要一个机器人使用噪音传感器数据在建筑物之间穿行躲避障碍。另一个例子，一个空中货物运输需要一个悬挂式的无人驾驶飞行器（**UAV**）长距离飞行躲避障碍，同时最小化负载的震荡（图 A.1b）。我们将远距离导航任务分成两个部分：远距离无碰撞路径规划和局部机器人控制。远距离无碰撞路径规划从起始位置开始，到远距离的目标重点躲避障碍，从而发现一个没有障碍的路径。局部机器人控制产生一个机器人执行任务的可行控制，同时满足

任务约束和保持在无障碍路径附近的要求。

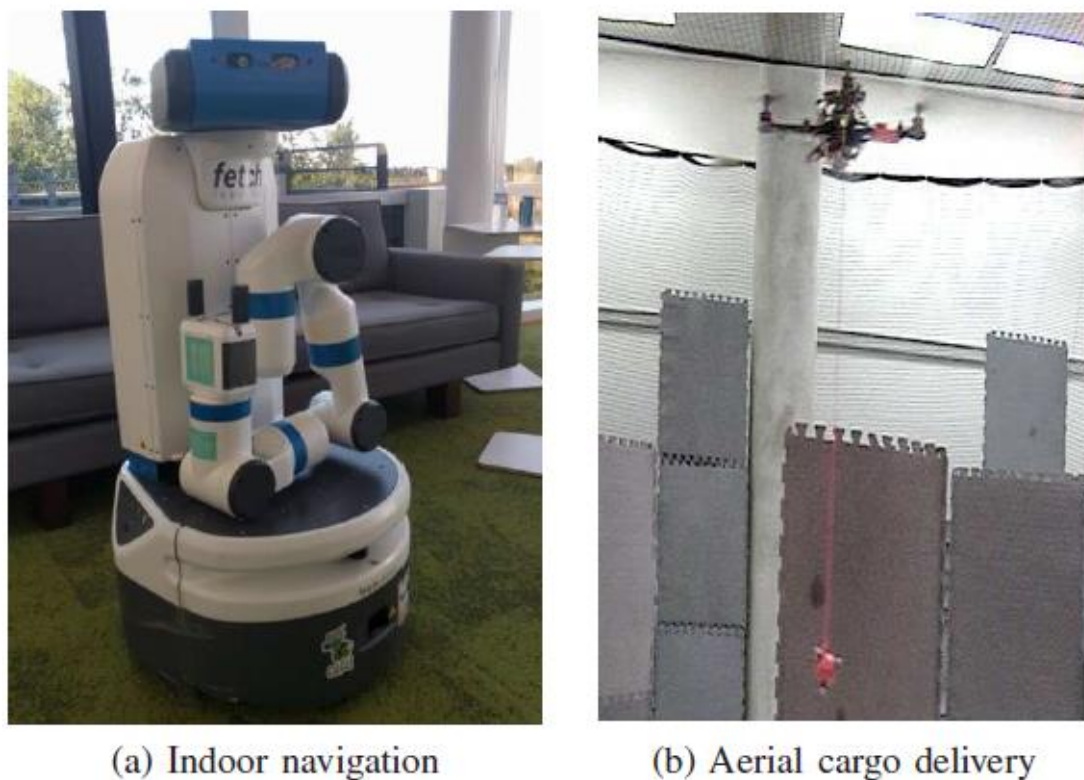


Fig. A.1. Case studies of long-range navigation tasks.

图 A.1 远距离导航任务所使用的情形

基于采样的规划方式，例如概率路线图（PRMs）和快速搜索随机树（RRTs），可以通过近似配置空间（C-space）的拓扑结构，也就是所有可能的机器人配置，快速解决该问题。这些方法通过采样 C-space 中的点建立一个图或者一个树，如果这些点之间存在局部的无障碍路径则将其连接。通常，这样的局部路径由一组测试或者一个低开销的局部规划方法产生。

如果不考虑无障碍的路径是如何产生的，执行过程会引入新的问题。一个机器人不能简单的遵循 C-space 的路径，但是必须 1) 满足任务的约束，2) 处理环境的变化，以及 3) 补偿传感器噪声、测量误差和未建模的系统动力学问题。强化学习方法提供了一个解决机器人控制问题的可行性方法。RL 代理商解决了复杂机器人控制问题，适应了新的环境，证明了其对噪音和误差的鲁棒性，同时甚至学习了更复杂的技能。尽管如此，RL 代理商在反馈稀疏的情况下是很难训练

的。这对将强化学习应用到导航问题同时是机遇，也是挑战。RL 代理已经被成功的运用于各类排列的导航任务和电子游戏。因此，他们对于处理任务约束是一个好的选择。相反地，复杂地图上的远距离导航任务只有稀疏的反馈，使得代理难以训练，亦或是仅在短距离上成功，因为其在局部最小值上的漏洞。例如，城市地图和办公室平面图经常有在宽阔障碍的另一边的需求，这会使得局部代理很迷惑；又或者是在箱型峡谷的另一端，这会使局部代理被困住。

我们提出了 PRM-RL 方法，一个解决远距离导航的任务的方法，它克服了 PRMs 和 RL 代理的局限，通过相互利用他们弥补另一个的短板。在 PRM-RL 方法中，一个 RL 代理被训练执行一个局部的点到点的任务，学习任务学术，系统动力学和传感器噪音，完全独立于远距离环境结构。接着，PRM-RL 利用 RL 代理决定连接性，建立一个路径图，而不是使用传统的在 C-space 中的无障碍直线插补法。PRM-RL 方法仅在 RL 代理可以持续的在两点之间执行局部的点到点的任务的时候，并且所有的配置点在产生的路径中是无障碍的，才会将两点连接。路线图，学习可以被 RL 代理导航的远距离环境结构。相比于基于纯 C-space 建立的线性的连接度的路线图，PRM-RL 路径可以遵循机器人动力学和任务限制。该线路图同时可以学习避免可能会造成 RL 代理失败的局部最小问题。最终建立的远距离导航规划方法结合了 PRM 的高效规划和 RL 代理的鲁棒性，同时可以避免局部最小的陷进，并缓冲环境的合理改变。

评估该方法时，我们主要集中在两个方面：室内导航（图 A.1a）和空中货物运输（图 A.1b）。室内导航问题需要一个微分驱动机器人来在建筑物内导航（图 A.2），同时仅使用它的 LIDAR 传感器（图 A.3）避免静态的障碍物。我们从目标建筑物的蓝图中建立了一个 PRMs，在最小的地图上使用有噪声的传感器和动力学训练 RL 代理，为了模仿真正的机器人的未经处理的、有噪声的传感器输入。我们在代理没有被训练过的环境中评估路径规划和 PRMs 的执行。空中货物运输问题需要一个有悬挂负载的四旋翼无人机运输货物，在最小震荡的情况下，同时保证货物的位移在一个给定的最小上界以下。合成的四旋翼无人机货物系统是一个非线性的、不稳定的系统。我们在一个模拟的城市环境下评估规划方法，并在两个实验环境中测试，评估其对任务和动态约束的保持持久性。我们发现在有静态障碍物的环境中，规划方法可以保持任务约束在一段长距离的轨迹上（总长超过 1km）。最终，两个问题在物理机器人真机的结果显示，PRM-RL 方法可以保证系统动力约束，同时提供可行的轨迹。

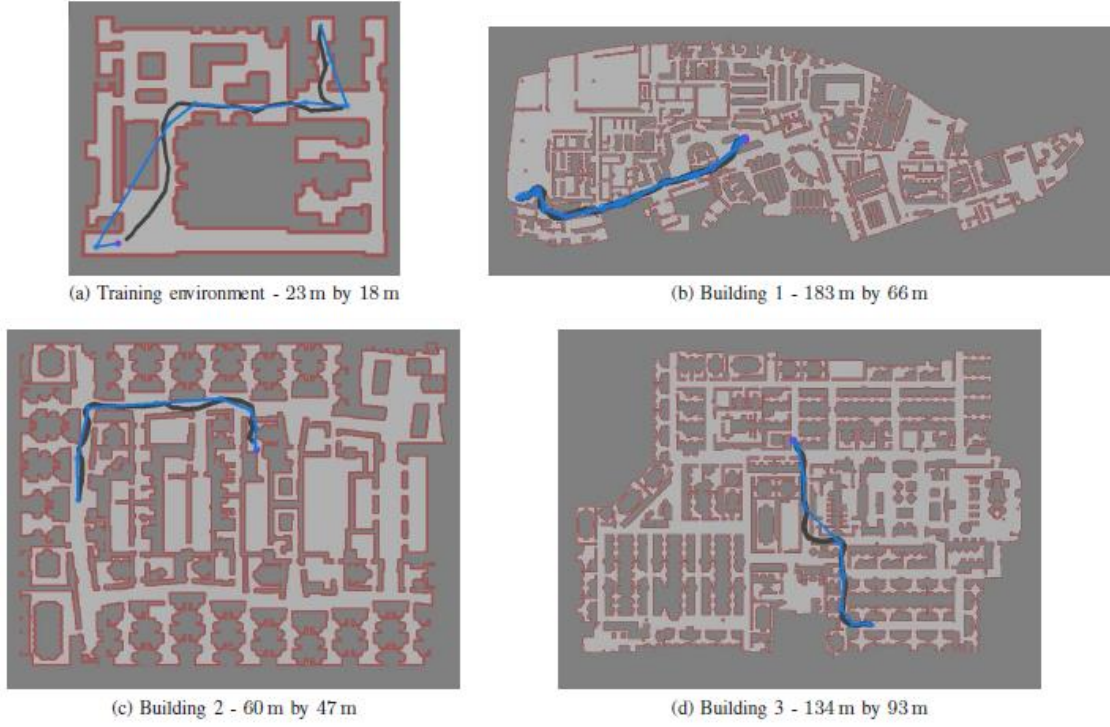


Fig. A.2. The environments used for the indoor navigation tasks are derived from real building plans. a) The smallest environment is used to train the RL agent. b)-d) The PRMs are built using agents trained in the training environment. Red regions are deemed too close to obstacles and cause episode termination when the robot enters them; white is free space from where the start and goals are selected. Line-connected PRM waypoints (blue) and agent executed trajectory with RL agent (black).

图 A.2 室内导航任务所使用的环境信息是从真实楼房内的路径规划问题所获得的。a) 最小的一个环境用来训练 RL 代理。b)-d) PRMs 算法通过在训练环境中训练代理来构建。红色区域被认为过于狭小的障碍区域因其会导致机器人卡死在其中。白色区域是可供机器人自由活动的空间且起始与终止位置皆在该区域中选择。蓝色的线将 PRM 轨迹点串起来。最终 RL 代理所执行的轨迹如黑色线所示。



Fig. A.3. Lidar observation that the differential drive robot uses for navigation. This observation corresponds to a hallway with a clear path ahead, and walls to the left and to the right. The white

rays mean there are not objects within 5m, and black rays mean that there is an obstacle near.

Notice the sensor noise in the center.

图 A.3 用于差速驱动机器人导航的激光雷达观测。该观测结果对应于前面有清晰路径的走廊以及左右两侧的墙。白色的光线意味着 5 米内没有物体。黑色的光线意味着附近有障碍物。注意中心的传感器噪音。

A.2 相关工作

PRMs 已经在从机器人到分子折叠等规划问题上得到了广泛的运用。它们也被正和岛强化学习中，通过使用 PRM 节点作为强化学习代理的状态空间，以减小整个状态空间维度。相比之下，我们的工作将强化学习作为 PRM 的局部规划方法在整个状态空间进行学习。在之前的工作中，对于空中货物运输任务，我们训练 RL 代理以跟踪 PRMs 通过使用直线局部规划方法产生的路径。在本次论文的工作中，RL 代理本身作为一个局部规划方法，减少了训练多个追踪代理的需求。PRMs 同样被修改成可以适用于移动障碍物、有噪声的传感器和局部误差的问题。安全的 PRM 使用概率碰撞检查与直线规划方法，与所有节点和边相关联，可以检测潜在碰撞。在我们的方法中，RL 局部规划使用蒙特卡洛路径，并在有噪声的传感器和动力学情况下做确定性的碰撞检查。我们仅在路径可以被连续的导航情况下添加边。

强化学习在解决未知系统动力的运动规划问题中很流行，并使得机器人可以学习原本很困难甚至不可能的任务。梯度深度决策策略（DDPG）是目前最优的算法，它可以再高维度的状态和动作空间中工作，同时可以基于未处理的传感器观测下学习控制机器人。连续动作拟合值迭代方法（CAFVI）是一个基于特征的连雪的状态和行为强化学习算法，它已经被用于解决多机器人任务、倒挂式飞行钟摆和障碍物躲避问题。在本文中，我们使用 DDPG 作为室内导航的局部规划方法，CAFVI 作为空中运输任务的局部规划方法。模式预测控制（MPC），行动过滤和层次策略估计方法，就像 RL，提供了遵从机器人动力学和任务约束的策略。尽管如此，他们在执行时间计算量上比 RL 更加费时，使他们在建立计算量需求大的路径图上不实用。

A.3 方法

PRM-RL 在三个状态下工作：RL 代理训练，路线图创建和路线图查询。图 4 显示了该方法的整体概述。

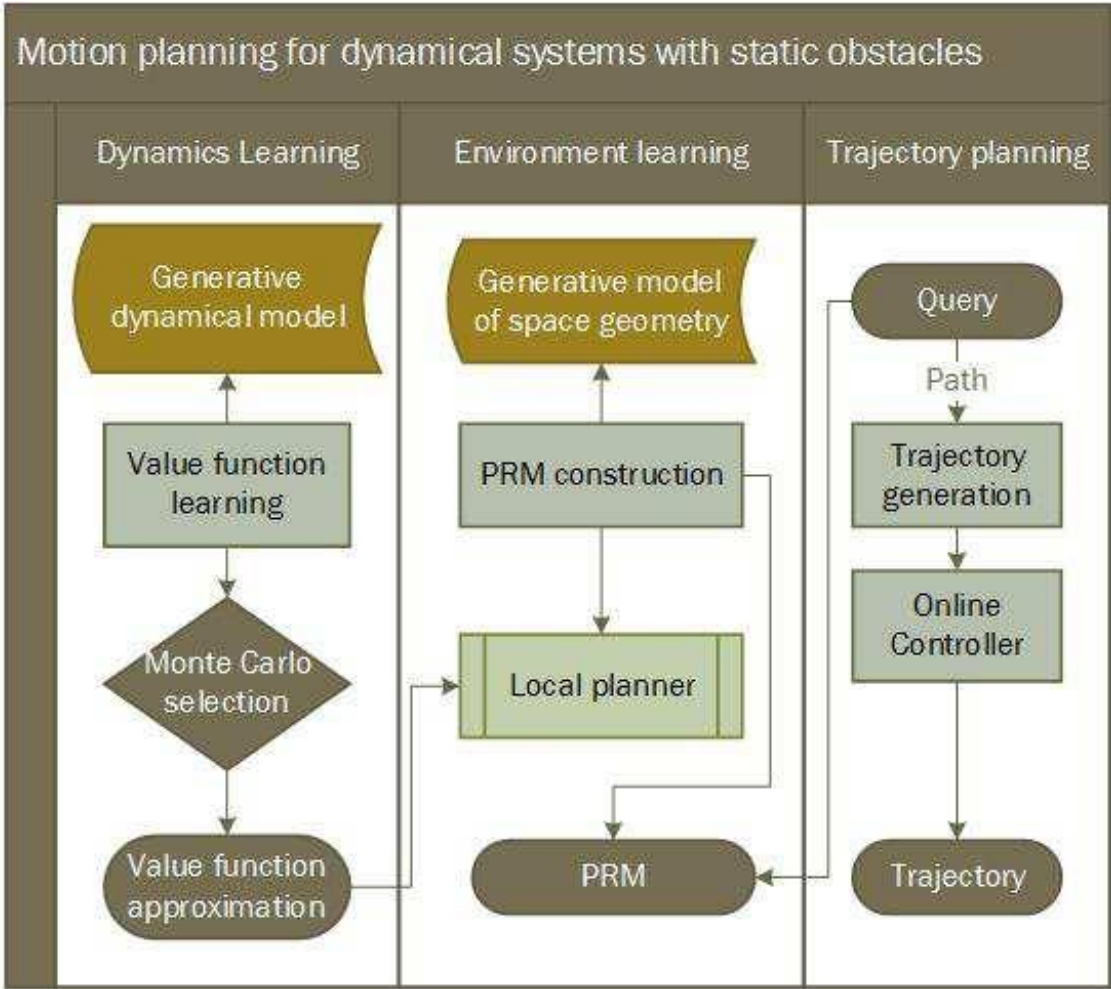


Fig. A.4. PRM-RL flow chart

图 A.4 PRM-RL 流程图

使得 PRM-RL 可移植到新环境的一个核心特征是它学习任务和系统动力学与执行环境完全隔离。在第一个阶段，我们训练 RL 代理在一个与执行环境类似的环境中执行一个任务，但环境有一个小的状态空间使得学习更容易处理。RL 代理训练状态是一个蒙特卡洛模拟执行过程：不论使用哪一种训练算法（DDPG, CAFVI 或是其他的），我们训练多个策略并选择最合适的一个作为 PRM-

RL 的下一个阶段使用。最佳策略，或价值函数，接着被传到路线图创建阶段。PRM 建立方法针对一个特定的执行环境学习一个线路图，使用算法 1 运用最好的 RL 代理作为局部规划方法。建立好的路线图可以被用作环境中的多个查询，只要执行时用的是同一个 RL 代理。当 RL 代理是依赖于机器人或任务的，它就是独立于环境的，因此它可以被用来针对多个不同环境建立路线图，就像我们在结果中展示的一样。

A.4 RL 代理训练

PRM-RL 使用一个开始状态 s 和一个目标状态 g 作为机器人状态空间 S 中的两个有效点。机器人状态空间 S 是一组所有可能的机器人控制空间的机器人观测值，因此是配置空间 $C\text{-space}$ 的超集。一个状态空间点 $s \in S$ 是有效的当且仅当它对于某些断言 $L(s)$ 满足任务约束，并且该点在 $C\text{-space}$ 上的投影 $p(s)$ 属于 $C\text{-free}$ ，一个 $C\text{-space}$ 上的仅包含无碰撞点的划分。当其中任意一个条件不符时，系统的安全性不能被保证，所以任务不能被执行。当系统状态足够接近目标状态，也就是在配置空间满足 $\|p(s) - p(g)\| \leq \epsilon$ ，任务被视为完成。我们的模板是寻找一个转换函数：

$$\dot{s} = f(s, a) \quad (A-1)$$

该函数可以使得系统最终完成任务。正式地表述，在马尔科夫决策过程设置中，我们的目标是寻找一个策略 $\pi: S \rightarrow A$ 使得对于初始状态 s ，存在一个 $n_0 > 0$ ，使得 $\forall n \leq n_0$ $s_n = \pi^n(s)$ 和 $L(s_n)$ 均成立，并且 $\|p(s_{n_0}) - p(g)\| \leq \epsilon$ 。

RL 代理被训练以执行点到点的导航任务，没有关于工作空间的拓扑结构的存储。强化学习寻找在离散时间 MDP 一个解决方案，该解决方案有连续的多维状态和行为空间，以一组状态、行为、转换和奖励反馈给出： (S, A, P, R) 。其中， $S \subset \mathbb{R}^{d_s}$ 是机器人的状态和观测空间。对于室内导航任务，我们使用了极坐标系下的机器人实际位置和目标点的相对位置 g ，以及所有可能的 LIDAR 观测结果（图 3），包括 64 个射线、 220° 的视野和 500 米的深度。总的来说，状态空间是： $s = (g, o) \in \mathbb{R}^{66}$ 。在空中货物运输任务中，状态空间是位置和速度向量以及货物的质心 $s = [s_p, s_v, \eta, \dot{\eta}] \in \mathbb{R}^{10}$ 的结合，其中 $s_p = [x \ y \ z]^T$ 是无人机的质心，其

线速度是 $s_v = [\dot{x} \ \dot{y} \ \dot{z}]^T$ ，悬挂货物的角位移是 $\eta = [\psi \ \phi]^T$ 。整个描述都在一个由四旋翼无人机质心位置为中心，角速度为 $\dot{\eta} = [\dot{\psi} \ \dot{\phi}]^T$ 球体坐标系中。

行为空间， $A \subset \mathbb{R}^{d_a}$ ，是机器人所有可执行的行为的集合。对于室内导航任务来说，行为空间是一个轮速度的二维向量 $a = (v_l, v_r) \in \mathbb{R}^2$ ，而对于空中运输货物来说，行为空间是一个应用于四旋翼质心 $a \in \mathbb{R}^3$ 的加速的向量。

转换概率 $P: S \times A \rightarrow \mathbb{R}$ 是一个状态和行动的的概率分布。就像很多强化学习系统一样，我们假设一个简化的黑盒模拟器的存在，它完全没有对完整非线性系统动力学的认知。室内导航任务的模拟器是一个 5Hz 的动力学模拟器。为了模拟不完美的真实世界传感，模拟器对观测添加高斯噪声 $N(0, 0.1)$ 。空中货物运输的模拟器是 50Hz 的，因为四旋翼内在的不稳定性，是一个[10]中四旋翼无人机负载模型的简化模型。

在训练过程中，代理观测一个标量的反馈 $R: S \rightarrow \mathbb{R}$ ，学习一个策略 $\pi(s) = a$ ，使得基于一个给定的观测状态 $s \in S$ ，返回一个行为 $a \in A$ ，使得代理可以执行以最大化长期反馈值：

$$\pi^*(s) = \operatorname{argmax}_{a \in A} E \left(\sum_i \gamma^i R(s_i) \right) \quad (\text{A-2})$$

我们奖励代理如果它完成了目标，并针对特定的任务给予特定的任务奖励。对于室内导航任务，我们奖励代理远离障碍物，而对于空中货物运输任务，我们奖励代理最小货物位移。

我们利用连续的 RL 算法的行为在一个小环境中训练代理。我们选择连续的 RL 行为因为，尽管其更难训练，它们提供更精确的控制，以及更快的策略评估时间。我们运用 DDPF 在一个小空间中（图 A.2a）训练室内导航任务。空中货物运输任务使用 CAFVI 训练，仅使用 1m 来训练达到目标。训练一次以后，代理可以被用于不同的环境中，通过观察世界并受到观测值和状态的回复规划轨迹，评估策略，并运用状态的推荐行为。注意代理在状态空间中规划，这比 C-space 的维度更高。

A.5 PRM 建立

基本的 PRM 算法，通过执行统一的机器人配置空间的机器人配置状态的随机采样，仅保持无障碍样本作为路线图的节点。PRMs 接着通过使用局部规划方法，尝试连接样本和其邻居节点。如果在两个节点中存在一个无障碍的路径，那么该边会被加入路线图中。

我们调整基本的 PRM 算法，改变节点连接的方式。因此，我们主要关注其对噪声的鲁棒性和对任务的持久性，我们仅仅连接两个配置如果 RL 代理可以连续的在两个点之间执行点到点的任务。因为状态空间 S ，是一个 C -space 的超集，我们采样其实节点和目标节点配置点的各种变化，而如果成功率超过阈值则添加边成功。注意，这代表 PRM 轨迹不能够保证无碰撞的路径。但这不是改算法的局限性，而是源于传感器的噪声。尽管如簇，在之后讨论结果的部分，我们评估了碰撞的可能性的最低界限。

算法 1 描述了 PRM-RL 如何添加边的过程。我们采样状态空间与起始节点和目标节点在配置空间中相关的多个点，并且尝试连接这两个点。仅当代理足够接近目标节点时，尝试才是成功的。计算整个轨迹的总长度，我们对所有步骤的距离进行求和，并添加剩余的距离到目标中。我们给边添加的长度是成功边的平均值。如果成功率高于预先设定的阈值时，算法建议添加边到路线图中。如果进行了太多不成功的尝试，算法终止。算法 1 的碰撞检测的个数是 $O(\max_{steps} * num_{attempts})$ ，因为对于每条边有多次尝试。每一次对轨迹的检查尝试可以并行于 $num_{attempts}$ 个处理器。

Algorithm 1 PRM-RL Add edge

Input: $s, g \in C_{space}$: Start and goal.
Input: $p_{success} \in [0, 1]$ Success threshold.
Input: $num_{attempts}$: Number of attempts.
Input: ϵ : Sufficient distance to the goal.
Input: max_{steps} : Maximum steps for trajectory.
Input: $L(s)$: Task predicate.
Input: π : RL agent's policy.
Input: D Generative model of system dynamics.
Output: $add_{edge}, success_{rate}, length$

```
1:  $success \leftarrow 0, length \leftarrow 0$ 
2:  $needed \leftarrow p_{success} * num_{attempts}$ 
3: for  $i = 1, \dots, num_{attempts}$  /* Run in parallel.*/ do
4:    $s_s \leftarrow s.SampleStateSpace()$  // Sample from the
5:    $s_g \leftarrow g.SampleStateSpace()$  // state space
6:    $success_{rate} \leftarrow 0, steps \leftarrow 0, s \leftarrow s_s$ 
7:    $length_{trial} \leftarrow 0$ 
8:   while  $L(s) \wedge steps < max_{steps} \wedge \|p(s) - p(s_g)\| >$   

    $\epsilon \wedge p(s) \in \text{C-free}$  do
9:      $s_p \leftarrow s, a \leftarrow \pi(s)$ 
10:     $s \leftarrow D.predictState(s, a)$ 
11:     $num_{steps} \leftarrow num_{steps} + 1$ 
12:     $length_{trial} \leftarrow length_{trial} + \|s - s_p\|$ 
13:  end while
14:  if  $\|p(s) - p(s_g)\| < \epsilon$  then
15:     $success \leftarrow success + 1$ 
16:  end if
17:  if  $needed > success \wedge i > needed$  then
18:    return False, 0, 0 // Not enough success, we can
    terminate.
19:  end if
20:   $length_{trial} \leftarrow length_{trial} + \|p(s) - p(g)\|$ 
21:   $length \leftarrow length + length_{trial}$ 
22: end for
23:  $length \leftarrow \frac{length}{success}, success_{rate} \leftarrow \frac{success}{i}$ 
24: return  $success_{rate} > p_{success}, success_{rate}, length$ 
```

为了产生远距离轨迹，我们在路径图上查询，返回一组点列。一个高级的规划方法将 RL 代理引入，来产生邻近点列的轨迹。当机器人在点列目标的范围内，一个高级的规划方法用点列中下一个点来替换目标。

A.6 结果

在本节中，我们将评估 PRM-RL 用于室内导路和空中货物运输的性能。

A.6.1 室内导路

我们按照图 2 所述的 4 张地图展开工作。我们使用 DDPG 算法训练 RL 代理在 14-17 米的路径上（如图 A.2a 所示的地图上）躲避障碍物。我们的设置如文献 22 所述。不同的是我们在表演者网络中设置了 2 个隐藏的层次，在批评者网络中设置了 2 个双隐的层次，以及 1 个用来表示状态的额外的隐藏层次，重量以 0.01 衰减。训练的批量是 124，其中亚当优化器的 α 是 0.9， β 是 0.999，小数点到 $1e-08$ ，表演者的学习速率是 $7.37e-05$ ，批评者的学习速率是 $1.14e-08$ 。目标网络每 13 个训练步骤更新 1 次，并配有 200K 的重放缓冲区。在训练和使用 RL 代理时，目标的公差是 0.5m。这使得 RL 代理能够在噪声和动态的环境下训练，即不依赖于外部的定位，因而是十分必要的。回顾一下，唯一需要输入给代理的指令是目标的位置和噪声的 LIADR 数据（图 A.3）。评估环境的大小是训练环境的 12-52 倍。

我们通过与使用直线规划器（PRM-SL）构建的 PRMs 进行比较来评估 PRM-RL。我们不与 RRT 进行比较，因为它们是一次性规划器，且在建设过程中花费过高。每个路线图根据从 C 自由空间中选择的 100 个询问点进行评估。我们研究 1) 建立路线图的成本；2) 规划轨迹的质量；3) 仿真中代理的实际性能；4) 实验结果。

1) 路线图构建评估：在构建 PRMs 的过程中，我们试验超过 20 次，连接边缘的成功率是 85%。PRM 尝试连接距节点 10m 以内的所有最近的点。我们为三种不同的节点密度构建路线图：每平方米 0.1、0.2 和 0.4 个样本。

表 A.1 总结了路线图的特点。我们检查路线图中的节点数量，边缘数量和为构建路线图而执行的冲突校对，且包括 100 个随机生成的询问点的成功几率。如我们预期的一样，在所有环境和路线图构建方法中，更高的采样密度会生成更大的地图和更成功的询问。地图中节点的数量并不取决于本地规划器，但边缘数量和冲突校对取决于本地规划器。因为我们对始终失败的边缘组提前终止冲突校对，所以碰撞校对的次数大约是使用算法 1 的 RL 本地规划器的 10-20 倍。SL 规划器不通过噪声传感器观察，所以需要单个试验来添加或删除边缘。我们观察到，使

用 RL 本地规划器所制作的路线图更密集地连接着 15% 和 50% 以上的边缘。这是预料之中的，因为 RL 代理能够绕过拐角和直线规划器不能识别的小障碍。

表 A.1 不同节点采样密度（每平方米 0.1, 0.2 和 0.4 个样本）的路线图构建总结。评价指标包括环境、方法、100 次评估的成功率、节点数量、边的数量以及碰撞检查的次数。

TABLE A.1

ROADMAP CONSTRUCTION SUMMARY DIFFERENT NODE SAMPLING DENSITIES (0.1, 0.2, AND 0.4 SAMPLES PER METER SQUARED). ENVIRONMENT , METHOD , SUCCESS RATE ON 100 QUERY EVALUATION , NUMBER OF NODES , NUMBER OF EDGES , NUMBER OF COLLISION CHECKS

Sampling density	Method	Query success rate (%)			Nodes			Edges			Collision Checks		
		0.1	0.2	0.4	0.1	0.2	0.4	0.1	0.2	0.4	0.1	0.2	0.4
Training	PRM-RL	0.32	0.36	0.50	16	32	63	33	166	663	13009	38292	223898
	PRM-SL	0.06	0.09	0.29	16	32	63	15	123	464	914	3649	17264
Building 1	PRM-RL	0.14	0.31	0.43	436	871	1741	3910	15632	59856	1476931	5755744	23303949
	PRM-SL	0.06	0.17	0.15	436	871	1741	3559	13937	52859	156641	622257	2393841
Building 2	PRM-RL	0.17	0.22	0.38	116	232	463	403	1602	6833	294942	1174655	5218619
	PRM-SL	0.06	0.11	0.18	116	232	463	276	1190	5365	18297	72850	312859
Building 3	PRM-RL	0.19	0.39	0.56	441	881	1761	2962	11850	45623	1152524	4492144	17947728
	PRM-SL	0.07	0.11	0.08	441	881	1761	1852	7570	30267	97088	375304	1493816

2) 预期的轨迹特征：现在我们看看 PRM-RL 在四种环境中对 100 个随机选择的询问点的预期性能。我们只关注密度为每平方米 0.4 个样本的路线图。表 A.2 总结了路点预期数量，路迹长度和路程。SL 本地策划器的结果更乐观，预期在规划器轨迹上 100% 成功。RL 代理则出于每个边的成功是独立的随机事件，计算了基于每边的联合概率的预期成功概率。因此，多个路点的预期成功率的下限为 0.85^{nw} ，其中 nw 是路点的数量。因此，2 号建筑轨迹成功率的下限应为 $0.85^6 = 37\%$ ，而 3 号建筑的下限为 $0.85^{12.65} = 13\%$ 。表 2 中的预期成功率高于 90%，这意味着在路线图建设期间，添加到路线图中的大部分边缘成功率为 95-100%。RL 本地规划器的实际成功率在下限之间。RL 代理不要求机器人在目标区域休息，所以机器人在切换路路点时会遇到一些惯性阻力，从而导致失败。也就是说，PRM-SL 的实际成功率明显较低。更重要的是，PRM-SL 规划器没有估计路径风险。我们还看到，PRM-RL 路径包含更多路点，但 3 号建筑除外。3 号建筑最大，通过它的路径需要更多的转弯。RL 代理可以在不添加路点的情况下执行其中的一些转弯。因为代理使用机器人可完成的现实评估标准，RL 代理的预期轨迹长度和持续时间较长。

表 A.2 100 个查询中预计的轨迹及实际导航的轨迹。评价指标包括：环境、方法、预期和实际成功率、路径中的点数、预期轨道长度（米）以及所用时间（秒）。

TABLE A.2
EXPECTED PATH AND TRAJECTORY CHARACTERISTICS OVER 100 QUERIES.
ENVIRONMENT , METHOD , ACTUAL AND EXPECTED SUCCESS PERCENT , NUMBER
OF WAYPOINTS IN THE PATH , EXPECTED TRAJECTORY LENGTH IN METERS , AND
DURATION IN SECONDS

Environment	Method	Success (%)		Number of waypoints		Trajectory length (m)		Duration (s)	
		Actual	Expected						
Training	PRM-RL	50	90	μ	σ	μ	σ	μ	σ
	PRM-SL	28	100	7.11	2.88	18.68	11.80	36.28	36.28
Building 1	PRM-RL	43	91	5.44	3.51	9.39	9.69	8.29	8.29
	PRM-SL	15	100	12.09	10.68	56.88	63.37	107.78	107.78
Building 2	PRM-RL	38	95	11.99	8.88	46.69	43.85	43.07	43.07
	PRM-SL	18	100	6.05	4.46	21.54	25.82	41.69	41.69
Building 3	PRM-RL	56	92	6.98	5.69	18.75	23.05	16.97	16.97
	PRM-SL	8	100	12.62	5.12	64.94	33.96	122.31	122.31
				15.58	8.02	59.03	35.47	54.00	54.00

表 A.3 成功与失败的导航轨迹特点。评价指标包括：环境、方法、预期和实际成功率、路径中的点数、预期轨道长度（米）以及所用时间（秒）。

TABLE A.3
CHARACTERISTICS OF THE SUCCESSFUL AND UNSUCCESSFUL TRAJECTORIES.
ENVIRONMENT , METHOD , ACTUAL AND EXPECTED SUCCESS PERCENT , NUMBER
OF WAYPOINTS IN THE PATH , EXPECTED TRAJECTORY LENGTH IN METERS , AND
DURATION IN SECONDS

Environment	Method	Success (%)	Number of waypoints				Trajectory length (m)				Duration (s)			
			Successful		All		Successful		All		Successful		All	
Training	PRM-RL	50	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
	PRM-SL	28	4.29	2.38	4.32	2.88	13.88	8.62	14.54	9.46	36.74	26.06	37.04	37.04
Building 1	PRM-RL	43	2.45	3.02	2.49	2.73	6.71	7.10	6.39	6.32	6.71	7.10	6.39	6.39
	PRM-SL	15	10.76	11.01	8.44	9.77	51.17	56.66	43.16	51.74	117.46	108.60	112.29	112.29
Building 2	PRM-RL	38	4.37	4.69	4.09	4.51	20.19	21.78	18.32	20.46	20.19	21.78	18.32	18.32
	PRM-SL	18	8.08	3.67	3.96	4.53	32.93	21.58	23.64	20.78	70.18	45.87	77.33	77.33
Building 3	PRM-RL	56	3.89	5.63	3.41	4.71	15.31	20.79	12.29	17.20	15.31	20.79	12.29	12.29
	PRM-SL	8	9.74	5.60	9.61	5.79	58.71	32.09	57.60	34.78	130.79	63.06	130.06	130.06
			5.66	5.33	5.21	4.46	22.46	19.67	21.30	17.50	22.46	19.67	21.30	21.30

3) 实际轨迹特征：为了评估实际的室内导路的任务性能，我们查看成功与失败询问的特征。表 A.3 总结了路点的数量，路迹长度和成功和失败路迹的持续时间的差异。RL 代理的成功率比直线规划器高。PRM-RL 在 3 好建筑中表现最佳，而 PRM-SL 在该环境中表现最差。最长的成功执行轨迹是 216 米长，需要 400 秒才能完成，并通过 45 个路点。

正如预期的那样，成功轨迹的路点比表 A.2 的预期路点少，这意味着较短的询问更有可能成功。我们也看到，在只有很少路点的情况下，不成功的查询失败。PRM-RL 的整体成功率较高，在最大的环境（建筑 3）中表现最佳，而 PRM-SL 表现最差。图 A.2 描述了测试环境中的一些 PRM-RL 轨迹。

4) 真实机器人实验：为测试我们的方法在真实机器人上的有效性和转移性，我们创建了一个简单的障碍环境，四个障碍物分布在 8m X 3m 的空间里。图 A.5 显示了由于传感器的噪声引起的单个查询方差。每个试验轨迹在 0.5m 的目标区域内达到，平均距离为 0.37m。

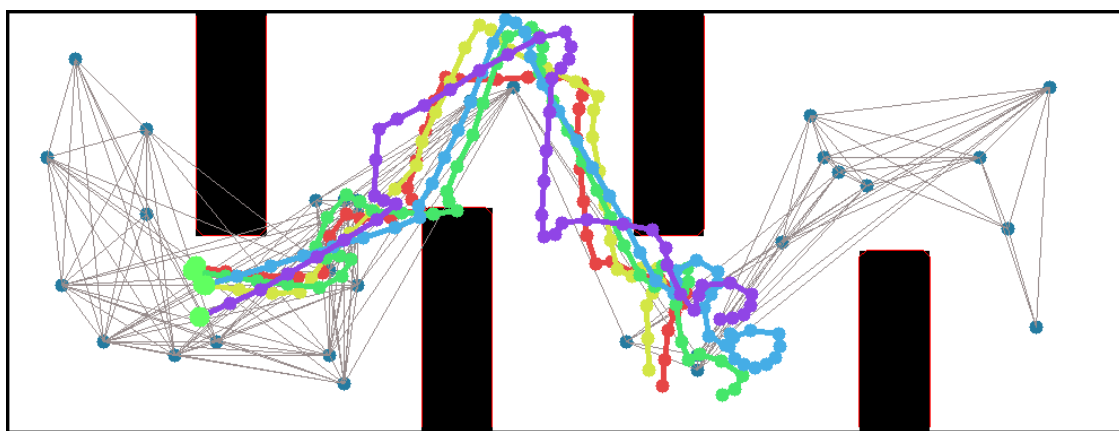


Fig. A.5. Trajectories for a single query, executed on the real differential drive robot over five trails (green, blue, red, purple, and yellow). The grey straight lines indicate graph edge connectivity of the PRM-RL. The trajectories are captured with a mocap system.

图 A.5 单次查询的轨迹，在真正的差分驱动机器人上按照图示五种路径（绿色，蓝色，红色，紫色和黄色）运动。灰色直线表示 PRM-RL 的图形边缘连通性。轨迹是通过 mocap 系统捕获的。

A.6.2 空中货物运输

一个城市的模型，也就是我们的模拟环境，允许我们对 PRM-RL 远距离测试，而实验环境允许我们在四旋翼无人机负载情况下验证我们的算法。城市模型（图 A.6a）是 450m-700m，高 200 米。它的边界框是四旋翼边界框的 2.5 亿倍，RL 代理训练环境的 6.3 亿倍大。限定负载位移小于 45° ，轨迹的评估是针对一个固定的起始点和一个随机选择目标点之间的 100 条查询请求的结果。我们测量负载位移、轨迹持续时间，并与 PRM-SL 相比较。空中货物运输任务是一个确定的问题，

我们建立的路线图需要一次尝试和 100% 的成功率。另外，RL 代理需要在沿途点上休息，使得不会因为惯性导致任务失败。

1) 模拟结果：图 A.6 展示了负载位移和轨迹持续时间的结果。xy 坐标系展示了城市的投影。红色区域为起点位置。数据点对的 xy 坐标为查询请求的在城市 xy 坐标系中的目标位置。图 A.6b 中数据点的 z 坐标是负载在整个轨迹中的最大位移，而图 A.6c 中则是完成轨迹所需要的时间。方块表示的点（绿色）为 PRM-SL 的结果，三角表示的点（蓝色）是 PRM-RL 的结果。最长的轨迹总长度超过 1 千米。最大负载位移在连续情况下持续低于离散规划方法展示的负载位移，同时保持在要求的 45° 以下。行为过滤保证了负载位移的限制，但是因为这是一个离散的行为规划方法有超过 170 万个行为，它在每一次局部规划方法执行时添加了 170 万个碰撞检测。这使得该方法在复杂度方面无法和 PRMs 共同使用。为了抵消负载唯一控制，PRM-RL 轨迹需要花费更长的时间完成任务。他们一般包括平均 5 倍多的沿途点。图 6c 展示了在起始点附近的去榆中，PRM-RL 和 PRM-SL 的轨迹有着类似的持久度。而随着目标距离的增加，差距变得越来越明显。

2) 实验结果：实验在物理机器人上评估了 PRM-RL 以证实了模拟结果。我们对比了在整个轨迹上的路径长度、持久度和负载位移。实验在 AscTec Hummingbird 四旋翼上进行，携带一个 62 厘米的悬挂负载，重 45g，在 MARHES 空中实验台上完成。该实验环境大小为 3 米-4 米-2 米，包含 3-5 个 2 米高的障碍物（见视频）。四旋翼和负载位置，通过 100Hz 的动力捕捉系统进行追踪，要求负载位移不超过 10° 。

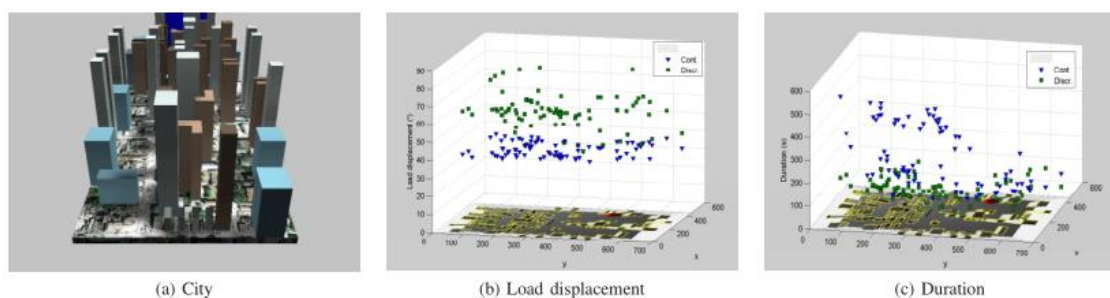
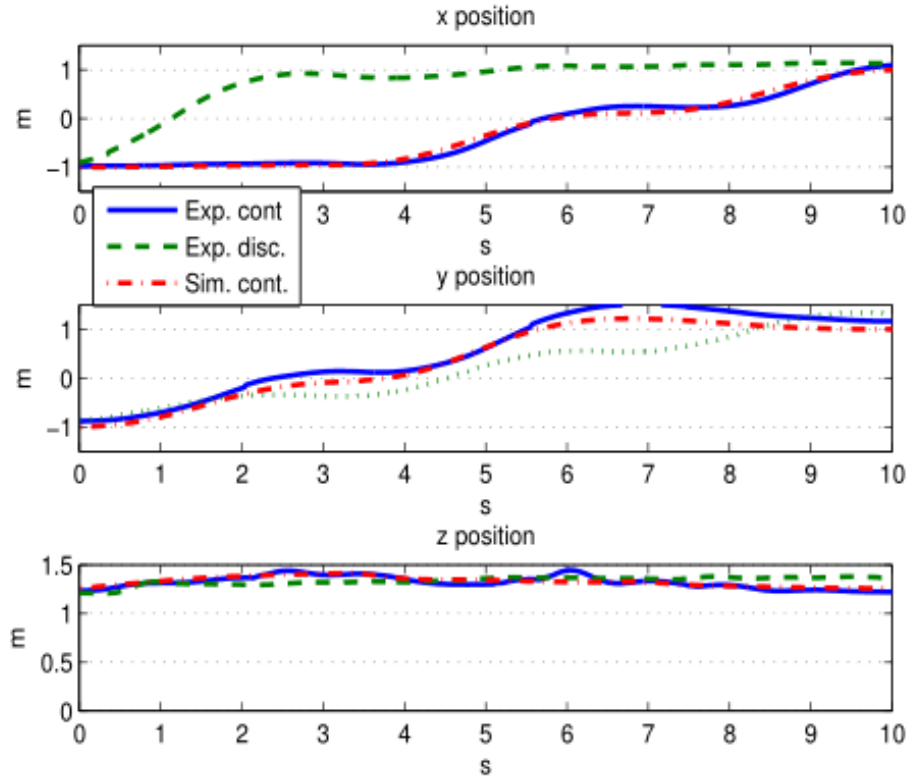
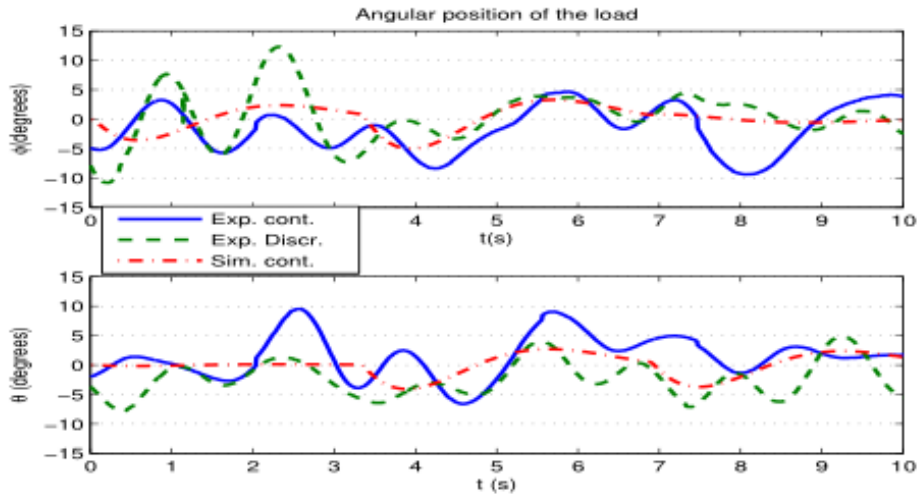


Fig. A.6. (a) Areal cargo delivery city environment. (b) Load displacement and (c) trajectory duration for PRM-RL (triangle) vs. PRM-SL (square) in the city projected on the xy-plane.

图 A.6 (a) 区域货物交付城市情况。(b) 载荷位移。(c) 投影在 xy 平面上的城市中 PRM-RL（三角形）与 PRM-SL（正方形）的轨迹持续时间。



(a)



(b)

Fig. A.7. Experimental PRM-RL trajectory with three edges, on the quadrotor demonstrating quadrotor (a) and load (b) trajectories. PRM-RL with continuous action RL from a hardware experiment is shown in comparison to the simulated trajectory. Also shown for comparison is RL with discrete actions following the PRM-SL path from a hardware experiment.

图 A.7 PRM-RL 的实验性轨迹应用于四旋翼飞机导航，其中(a)为四旋翼飞机，(b)为负载。显示了硬件实现的具有连续作用 RL 的 PRM-RL 与仿真轨迹的比较。此外，为了比较，还显示了具有硬件实现的在 PRM-SL 路径的基础上采用离散动作的 RL 的飞行情况。

图 A.7 展示了交通工具和负载轨迹在实验和模拟中高度一致，同时甚至在试验中满足负载位移不超过 10° 。另外，由于未建模的负载波动，模拟和实验存在 10° 差距，和我们之前的[10]中的结果一致。（注：文中此处以及此前多次引用了[10]这篇工作，即 A. Faust, I. Palunko, P. Cruz, R. Fierro, and L. Tapia. Automated aerial suspended cargo delivery through reinforcement learning. *Artificial Intelligence*, 247:381 - 398, 2017. Special Issue on {AI} and Robotics.。该篇文章是本文的团队此前的相关研究工作。）

接着，我们在同一环境下创建了 PRM-SL 轨迹。交通工具轨迹略有不同，因为 PRMs 在两个情况下有区别。PRM-SL 不能直接控制负载位移，而 PRM-RL 拒绝添加超允许最大位移的边。负载轨迹（图 A.7b）显示了负载位移有 2.5 秒超过了 10° 。视频 1 中包括了实验录像。

A.7 结论

我们设计了 PRM-RL，一个解决远距离导航任务的层次的规划方法，结合了基于采样的路径规划和 RL 代理，在大型试验中完成任务。对于两个研究案例的评估，一个是有噪声传感器反馈任务，另一个是在一个复杂的、不稳定的动力学情况下，我们展示了 PRM-RL 扩展了 RL 带和基于采样的规划方法的能力。室内导航任务成功的完成了 210 米长的轨迹，空中货物运输在训练空间的 6.3 亿倍大的空间中完成了 1 千米长的距离。两个任务都在机器人上进行了验证。

书面翻译对应的原文索引

- [1] Faust A, Ramirez O, Fiser M, et al. PRM-RL: Long-range Robotic Navigation Tasks by Combining Reinforcement Learning and Sampling-based Planning[J]. 2017.

综合论文训练记录表

学生姓名	王澎	学号	2014011514	班级	计 45
论文题目	通用小型无人机地面控制站的设计与实现				
主要内容以及进度安排	<p>主要内容:</p> <ol style="list-style-type: none"> 1. 对于目前传统的以及实验室现有的无人机地面控制站进行改进,采用WPF优化前端UI显示。 2. 在目前GCS的2D地图的基础上引入ArcGIS实现3D地图嵌入并支持自主导入地理信息数据进行测试实验。 3. 无人机航路规划模型的建立并在3维空间中测试传统2维地图上的搜索算法,包括: A*, Voronoi图, 遗传算法, 粒子群算法及蚁群算法。 4. 在3D GCS上尝试将A*算法结合进化算法等解决航路规划问题。 <p>进度安排</p> <ol style="list-style-type: none"> 1. (16周~1周) 文献调研与框架学习。 2. (1周~8周) 基于WPF采用C#开发前端界面,实现基本数据收发与3D地图显示。 3. (4周~6周) 在简单航路规划问题上测试现有2D地图上的相关算法。 4. (7周~9周) 优化GCS界面,尽量模块化并将各种组件进行封装最终集成为一个可定制的GCS系统。 5. (10周~13周) 在更复杂环境,如城市,测试与改进前述航路规划算法。 6. (14周~16周) 整合GCS系统,整理实验数据,完成论文。 <p>指导教师签字: 朱纪洪</p> <p>考核组组长签字: 刘石印</p> <p>2018年 1 月 16日</p>				
中期考核意见	<p>已完成相关文献的调研,搭建成了一个可定制GCS台1套. 对航路规划算法也进行了初步研究</p> <p>考核组组长签字: 刘石印</p> <p>2018 年 4 月 11 日</p>				

指导教师评语	<p>王彪同学设计实现了无人机地面控制站协议层、数据层、应用层，并通过航迹规划模型，验证了相关算法的有效性。论文内容丰富、逻辑完整，达到本科生综合论文训练水平。</p> <p>指导教师签字：朱和洪</p> <p>2018年6月15日</p>
评阅教师评语	<p>王彪同学的毕业论文设计论文摘要针对无人机地面站的协议层、数据层和应用层进行了设计分析，并进行了航迹规划方法，通过整合，完成系统功能，论文结构清晰，书写规范，达到毕业</p> <p>评阅教师签字：朱和洪</p> <p>论文符合要求。2018年6月15日</p>
答辩小组评语	<p>论文针对无人机地面控制站，对系统了其协议层、数据层和应用层的整合，并对相关规划算法进行实现。答辩过程表述清晰，回答问题正确。</p> <p>答辩小组组长签字：刘子平</p> <p>2018年6月15日</p>

总成绩：89

教学负责人签字：黄氏烈

2018年6月15日