**上海海事大学**

**Python程序设计课程开题报告**

**题目 清舱机器人一体化控制系统**

**姓 名 何昂**

**学 号 202210310219**

**学科(专业) 人工智能专业**

**所在学院 信息工程学院**

**提交日期 2024年10月 5 日**

目 录

[第一章 引言 1](#_Toc178977023)

[1.1背景及研究意义 1](#_Toc178977024)

[1.2组员分工 3](#_Toc178977025)

[第二章 需要解决的问题 3](#_Toc178977026)

[第三章 项目路线 4](#_Toc178977027)

[参考文献 6](#_Toc178977028)

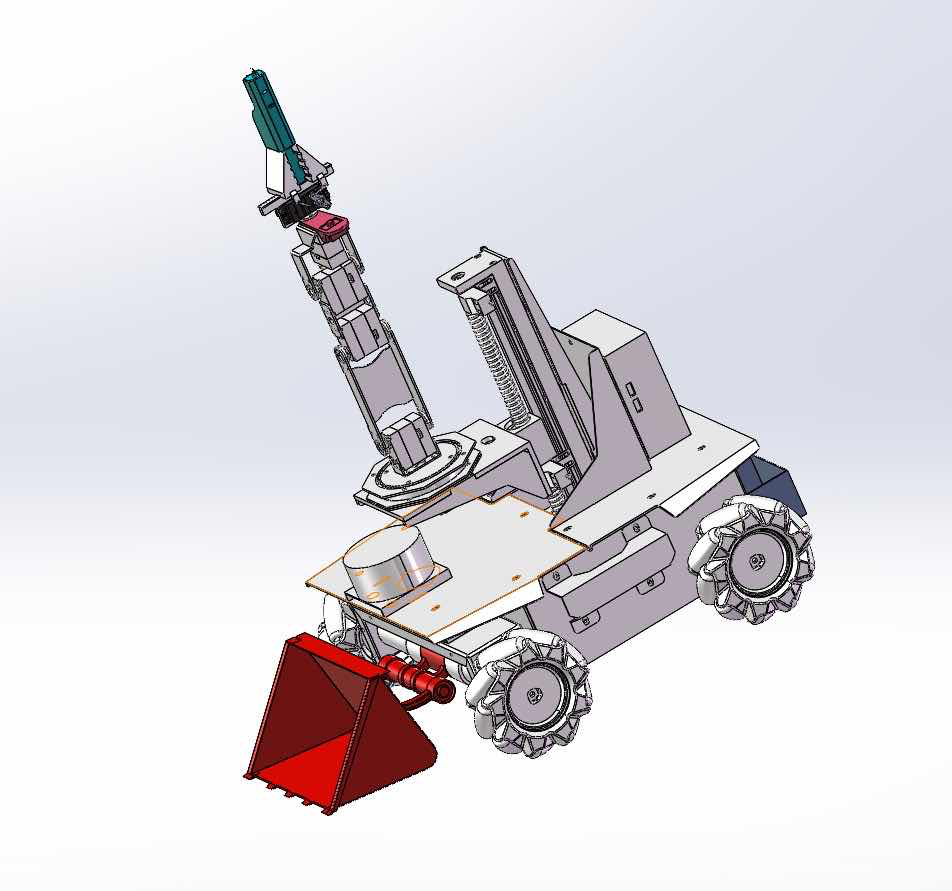
[附录 7](#_Toc178977029)

# 第一章 引言

## 1.1背景及研究意义

在海上货物运输装卸货的过程中，杂货船清舱是船员们的一大难题。在紧张的日程安排和繁重的任务表下，机组人员经常需要昼夜颠倒地工作来完成船舱清洁和卸货工作，有时甚至无法及时完成卸货和清舱，比如两端航程过近，前一段装载的是煤炭而下一段要装的是粮食（煤改粮），就要对船舱进行清洁，到达可以装载粮食的标准，而由于行程很近，导致不能及时的清洁，造成巨大的经济损失。此时，运用了人工智能技术的船舶清洗舱机器人就应运而生了，政府颁布了相关技术标准，包括《防治船舶污染海洋环境管理条例》《中华人民共和国船舶及其有关作业活动污染海洋环境防治管理规定》等，规定了船舶清仓机器人的设计、制造、安装等方面的要求，确保其符合安全、稳定、高效的标准。

在过去，船舶内部的清洁工作通常需要大量的人力投入，而且由于船舶内部空间狭小、环境复杂，清洁工作难度较大。因此，多功能清舱机器人的出现为船舶清洁带来了革命性的变化。这些机器人可以自动在船舱内部进行清洁，不仅提高了清洁效率，还能减少人力成本和提高工作安全性。随着航运业的发展和对船舶清洁卫生要求的提高，多功能清舱机器人的市场需求也在逐渐增加。



**图 1 船舶清洗舱机器人示意图**

以我们研发的该机器人为例，相较于普通的AGV底盘加装了铲斗，用于聚集散货，大部分货物卸下船后，剩下的散落的货物难以直接使用抓斗卸载，所以需要将货舱内的货物收集成堆货物，传统的方法需要人工操作铲车进行操作。然后，利用抓斗抓取并卸载货物。货船的船舱很大，需要大量的人力。然而，人工智能和机器人技术的引入可以有效解决这一问题。在大部分货物卸载完毕后，机器人可以将货物堆放起来，方便卸货。机器人通过灵活转向的底盘和自动避障功能还有紧急停止来实现这一目的，紧急停止功能是指当雷达识别到周围舱壁的距离，在即将碰撞前紧急停止，作为一个保险功能。首先自动避障功能和紧急停止功能确保了机器人不会撞到散货船的舱壁导致舱壁的结构损伤，全转向的底盘可以在舱底灵活的行进，按照预设的路线规划，将货物聚集在货舱中间，使得抓斗便于卸载货物，散货船货舱结构复杂、高度较高，并且有时会有有毒气体存在，例如装运煤炭时，煤炭可能会因为氧气不足散发一氧化碳，装运粮食时，可能会因为

****粮食受潮或发霉而散发氨气，这些对于船员来说非常危险。我们的多功能机器人可以代替船员执行这些危险的任务。总的来说，人工智能和机器人技术在船舶自动分拣和清洁领域的应用，可以让船员得到充分的休息，保障航行安全，同时减少这项重要任务所需的劳动力，让船员能够更加专注于其他任务。

**图 2 相关研究成果荣登海大官微 图3 在工博会现场演示给宋宝儒书记** **图4 获得中国国际创新创业大赛（互联网+）上海市银奖**

在机器人的使用过程中，使用linux系统中ros（机器人操作系统）**存在着众多需要协同的地方，每次开机或做出适应环境性的改变时都需要很多时间**，因此我们打算做一个**多功能集成的python系统**---清舱机器人一体化控制系统。机器人使用我们自创的EKF SLAM(增强卡尔曼滤波同步建图与定位算法)，在不同空间测试需要进行路径规划及精准度的测试和收集，需要ssh功能使得个人笔记本与机器人主控制板（nuc）进行连接等等一些功能将在第二章描述。

## 1.2组员分工

何昂：软件总体功能设计、UI界面设计、路径规划算法实现（A\*）、ssh功能实现、enu站心坐标系转换模块设计

黄宏伟：数据采集与收集模块、路径规划收集、卡尔曼滤波参数调优模块、路径规划算法实现（A\*）

宋欣宇：UI界面设计、数据采集与收集模块、路径规划收集、卡尔曼滤波参数调优模块

# 第二章 需要解决的问题

**1. 自主导航与路径规划需求：**在这个项目中，机器人需要在复杂的船舱环境中自主导航，避开障碍物，并根据预设的路径点完成任务，再优化路径时，使用A\*算法实现路径规划，Python通过对输入的路径点和障碍物进行处理，帮助机器人找到从起点到终点的最优路径，并实时显示路径规划的结果。

**2. SSH 远程控制需求：**用户需要远程管理和控制机器人，能够通过个人笔记本与机器人主控制板（nuc）建立连接，执行控制指令、上传程序、查看系统状态等。

**3. 数据采集与记录需求：**为了评估机器人的性能，需要实时采集路径规划数据、机器人运动数据，以及在不同环境下的精准度测试数据。Python程序实时输出路径规划的坐标数据，并通过数据记录模块记录机器人在环境中的运动轨迹和相关测试数据，便于后续分析和优化。

**4.实际坐标与站心坐标系转化**：在ROS操作系统中，机器人并非使用我们真实世界的坐标，而是使用enu（站心）坐标系，而每次修改程序要重新计算enu坐标显得比较麻烦，我们希望设置一个程序可以通过python代码直接运算的出现实坐标转换enu坐标。

**5.卡尔曼滤波调优：**所有SLAM算法都会因为外界变化有可能会造成地图的轻微变化，如激光遭到了反射或者地图中边缘点不够等等，因此每到一个新的环境中，卡尔曼滤波的参数都要进行调整，我们希望设计一个模块来更好地、更直观的来帮我们处理、解决误差。

因此，清舱机器人系统需要集成多种功能模块，如路径规划、定位、数据记录、SSH 连接、状态监控等，且能够通过一个统一的平台进行操作。

# 第三章 项目路线

**1. 自主导航与路径规划需求**

功能需求：机器人需要在复杂的船舱环境中自主导航，避开障碍物，并根据预设的路径点完成任务。

**实现方法：**

A 算法\*：使用Python的 heapq 库实现A算法，用于在包含障碍物的网格中寻找从起点到终点的最优路径。A算法基于启发式搜索，通过计算路径的代价函数，规划机器人行进路径。

实时路径显示：通过Python的 Tkinter 库创建图形界面，用户可以输入路径点，并动态地在界面中显示机器人行进路径及避障结果。

涉及的库：Tkinter（图形界面）、heapq（优先队列用于A\*算法）。

**2. SSH 远程控制需求**

功能需求：用户需要通过远程连接机器人主控板（nuc）来执行指令、上传程序、查看系统状态等。

**实现方法：**

paramiko 库：Python通过 paramiko 库实现SSH功能，使用户能够通过笔记本远程连接机器人主控板，并在远程终端上执行相关操作。用户可以通过系统界面一键登录和控制机器人，简化操作流程。

功能模块集成：远程控制功能集成在Python程序的图形界面中，用户可以通过单击按钮实现SSH连接及执行命令。

涉及的库：paramiko（SSH连接）。

**3. 数据采集与记录需求**

功能需求：需要实时采集机器人运行中的路径数据、运动轨迹及性能测试数据，以便后续分析和优化。

**实现方法：**

实时数据输出：通过Python的 ScrolledText 组件在图形界面中输出机器人路径的坐标点，用户可以实时查看路径规划及行进结果。

数据记录模块：程序在路径规划过程中记录相关数据（如路径点、障碍物位置、机器人运动轨迹），并保存这些数据用于性能评估及进一步分析。

涉及的库：Tkinter（图形界面与实时输出）、ScrolledText（滚动文本框）。

**4. 实际坐标与站心坐标系转换**

功能需求：在ROS系统中，机器人使用ENU（东-北-天）坐标系，而用户希望直接在现实坐标系与ENU坐标系之间进行转换，避免每次手动计算。

**实现方法：**

ENU转换模块：通过Python程序设计一个自动转换模块，用户输入实际的X、Y、Z坐标后，Python程序可以通过坐标转换公式，自动将其转换为ENU坐标，简化了操作流程。

GUI集成：该功能集成在图形界面中，用户可以通过输入框输入实际坐标，程序自动转换并显示结果。

涉及的库：numpy（用于数学计算）、Tkinter（图形界面）。

**5. 卡尔曼滤波调优需求**

功能需求：在不同环境下，SLAM算法的精度可能受到外部因素（如激光反射、边缘点等）的影响，因此需要对卡尔曼滤波的参数进行调整，以降低误差并提高SLAM算法的稳定性。

**实现方法：**

卡尔曼滤波调优模块：通过Python实现卡尔曼滤波的参数调优模块，用户可以在图形界面中通过滑块或输入框调整卡尔曼滤波的Q、R、P参数，以优化不同环境下的SLAM性能。

数据可视化：使用 matplotlib 库，将卡尔曼滤波的调整结果进行可视化展示，帮助用户更直观地了解参数调整对SLAM结果的影响。

涉及的库：numpy（数学计算）、matplotlib（数据可视化）、Tkinter（图形界面与滑块组件）。

在我们想要设计的清舱机器人系统中，A\*算法是路径规划的核心算法，用于帮助机器人在复杂的环境中找到从起点到终点的最优路径，并且避开障碍物。可以使用 heapq 库实现了优先队列管理，在路径规划中，优先扩展代价最小的节点，确保机器人行走的是最短路径。在用户界面部分，可以使用 Tkinter 库实现了完整的图形化界面，使得用户能够方便地输入路径点、添加障碍物并执行路径规划。用户可以通过该界面实时查看路径规划的结果，路径点、障碍物以及生成的最优路径会动态展示在界面中。Tkinter 的 Canvas 组件用于绘制路径，ScrolledText 组件用于显示路径的实时数据输出。为了增强用户体验，time 库用于模拟机器人在规划路径时的移动延迟，从而使得路径展示更加直观。除了路径规划，系统还集成了SSH远程控制功能，使用 paramiko 库让用户通过Python代码远程连接机器人主控板，并执行控制命令，这样用户无需物理接触设备即可远程操作机器人。此外，为了确保机器人能够适应不同的工作环境，系统还提供了自动坐标转换功能，通过 subprocess 调用外部脚本，Python将实际世界的坐标自动转换为ROS使用的ENU坐标系，大大简化了坐标转换过程。最后，为了应对环境变化对SLAM算法精度的影响，Python使用了卡尔曼滤波调优模块，用户可以通过界面调整滤波参数，提升SLAM算法在新环境中的表现，并通过 matplotlib 库实时可视化调整结果。综上所述，Python通过 Tkinter、heapq、paramiko、subprocess 等多个库的集成，不仅实现了机器人自主导航、远程控制、路径规划及数据记录，还为用户提供了简洁直观的操作界面，使得系统能够高效应对复杂的工作场景。



**图5 页面设计图**

# 参考文献

[1]Yue Wenlong Research on Point Cloud Processing Algorithm for Port Automatic Unloading System [D]. Shanghai Jiao Tong University, 2020. DOI: 10.27307/d.cnki. gsjtu.2019.002621(In Chinese).

[2] Checcucci Enrico, Piramide Federico, De Cillis Sabrina, et al. Health Information Technology Usability Evaluation Scale and User-Experience Questionnaire for 3D Intraoperative Cognitive Navigation System for Urological Procedures[J]. Medicina, 2023, 59(3):624-624.

[3] Yunze, L. Research on SLAM of Indoor Robot Based on Lidar. Master’s Thesis, South China University of Technology, Guangzhou, China, 2016.

[4] Zhao J, Liu S, Li J. Research and Implementation of Autonomous Navigation for Mobile Robots Based on SLAM Algorithm under ROS. Sensors. 2022; 22(11):4172. https: //doi.org /10.3390/s22114172

[5] Wenzhi, L. Research and Implementation of SLAM and Path Planning Algorithm Based on Lidar. Master’s Thesis, Harbin Institute of Technology, Harbin, China, 2018.

[6] Z. Yong et al., "An Autonomous Navigation Strategy Based on Improved Hector SLAM With Dynamic Weighted A\* Algorithm," IEEE Access, vol. 11, pp. 79553-79571, 2023, doi: 10.1109/ACCESS.2023.3299293.

[7] C. Tian, H. Liu, Z. Liu, H. Li and Y. Wang, "Research on Multi-Sensor Fusion SLAM Algorithm Based on Improved Gmapping," IEEE Access, vol. 11, pp. 13690-13703, 2023, doi: 10.1109/ACCESS.2023.3243633.

[8] Gao C. et al 2019 Research on a Panoramic Mobile Robot for Autonomous Navigation Proc. of the 3rd Int. Conf. on Mechatronics Engineering and Information Technology 87 204-207

[9] Zhang B, Li S, Qiu J, You G, Qu L. Application and Research on Improved Adaptive Monte Carlo Localization Algorithm for Automatic Guided Vehicle Fusion with QR Code Navigation. Applied Sciences. 2023; 13(21):11913. https://doi.org/10.3390/app132111913

[10]Yue Wenlong Research on Point Cloud Processing Algorithm for Port Automatic Unloading System [D]. Shanghai Jiao Tong University, 2020. DOI: 10.27307/d.cnki. gsjtu.2019.002621(In Chinese).

# 附录

使用的库：

**heapq**：用于实现A\*算法中的优先队列管理，帮助选择代价最小的节点进行扩展，从而实现最优路径的搜索。

**Tkinter**：用于创建图形用户界面（GUI），提供用户输入路径点、设置障碍物、执行路径规划、显示结果的操作界面。Canvas 组件用于绘制路径，ScrolledText 组件用于显示实时输出。

**PIL（Pillow）**：用于加载和展示机器人图片、团队LOGO等图片资源，使界面更加直观和美观。

**time**：用于模拟路径规划过程中的延迟，动态展示机器人行进路径，提供更加真实的路径展示效果。

**paramiko**：用于SSH远程控制，使用户可以通过Python远程连接机器人主控板（nuc），执行控制命令、上传程序、查看系统状态等。

**subprocess**：用于调用外部Python脚本，比如坐标转换模块（ENU转换器）和卡尔曼滤波调优模块等功能，通过调用外部程序实现更复杂的操作。

**numpy**：用于数学计算，尤其在坐标转换过程中，用于实际坐标和ENU坐标系之间的转换。

**matplotlib**：用于卡尔曼滤波调优时的结果可视化。用户可以在调整滤波参数后，通过该库动态显示优化后的SLAM性能表现。

开发环境：

系统 : Windows 10 专业版 64位 版本号 19045.4894

处理器 : Intel Core i7-12700H @ 2.7GHz核心数 14 (6P+8E)线程数 20 (12P+8E) 核心数 14 (6P+8E) 线程数 20 (12P+8E)频率 P-4339 MHzE-3292 MHz频率 P-4339 MHzE-3292 MHz电压 P-1.281 VE-1.248 V

显卡 : Intel Iris Xe GraphicsNVIDIA GeForce RTX 3060 Laptop 品牌 DELL 流处理器 768 显存 1G 品牌 DELL流处理器 3840 显存 6G (GDDR6 Samsung)

主板 : 0T9XYP 品牌 Alienware 芯片组 Intel Alder Lake-P PCH

硬盘 : PM9A1NVMeSamsung2048GB 容量:1.86TB

显示器 : Sharp LQ140M1LQ140M1 [DELL P/N: PPCRY]Sharp LQ140M1LQ140M1 [DELL P/N: PPCRY] 分辨率 1920\*1080 刷新率144Hz 屏幕尺寸 13.9英寸 分辨率 1920\*1080 刷新率-1Hz 屏幕尺寸 13.9英寸

内存 : 容量 32 GB通道 4频率 5188 MHz 时序 64-48-48-1121T

内存1 80AD00000000 DDR5-6400 (3200 MHz) 4GB

内存2 80AD00000000 DDR5-6400 (3200 MHz) 4GB

内存3 80AD00000000 DDR5-6400 (3200 MHz) 4GB

内存4 80AD00000000 DDR5-6400 (3200 MHz) 4GB内存580AD00000000 DDR5-6400 (3200 MHz) 4GB内存680AD00000000 DDR5-6400 (3200 MHz) 4GB内存780AD00000000 DDR5-6400 (3200 MHz) 4GB内存880AD00000000 DDR5-6400 (3200 MHz) 4GB

机器人系统（ssh模块连接的）：

操作系统：

Ubuntu 18.04

开发语言:

Python 3.6.5

C++

XML

Python开源包：

Pyhton3: sklearn、NumPy等等

Github开源包：

Hector-slam: <https://github.com/tu-darmstadt-ros-pkg/hector_slam>

Gmapping: <https://github.com/ros-perception/slam_gmapping>

机载计算机:Jeston nano

CPU:四核ARM [A57@1.43GHz](mailto:A57@1.43GHz)

GPU:128核Maxwell

显存:4GB 64位LPDDR4 25.6GB/S

储存:microSD

机器人路径规划功能简介（和python适配）：

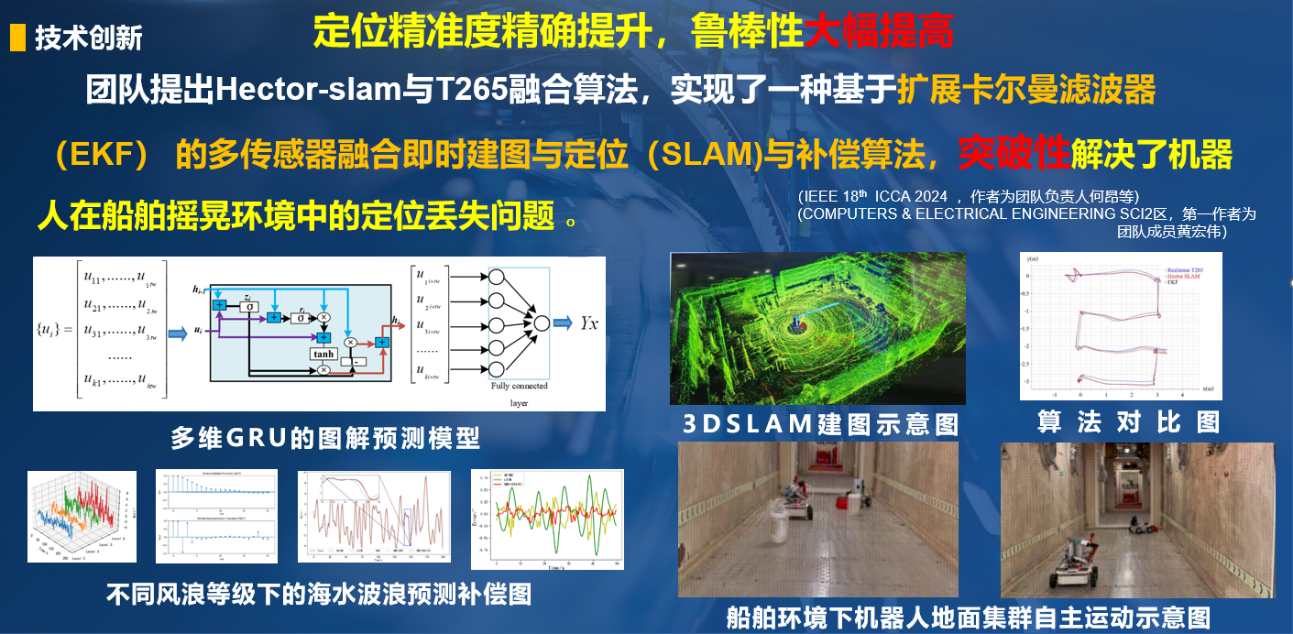


图6 机器人自主导航算法示意图

地图定位和建图模块在自动导航中起着重要作用。它使用激光雷达等传感器来采集机器人周围的环境信息，根据点云地图匹配和转换，并根据这些信息构建环境地图。SLAM算法有很多，这里选取hector-slam来进行介绍，这也是我用过的一个算法，hector-slam[9]用于同步建图与定位，使用了ROS中Navigation stack中gmapping功能包costmap来实现避障功能。这些算法将从传感器（镭神M10P）获取的激光雷达数据进行处理和滤波，将其中有用的特征提取出来，并用于定位和地图构建。对于地图定位，该模块会根据机器人当前的传感器数据与已建立的地图进行匹配，以估计当前机器人所处的位置。而对于地图建立，上述模块会将从传感器获得的数据进行处理，并将其整合到已有的地图中，以不断更新和完善地图，通过划分区域，船舶清舱机器人就可以根据预定的区域进行清舱和洗舱。