



浙江工业大学

本科毕业设计论文

题目： 基于无人机飞控平台的算法研究

作者姓名 沈菲菲

指导教师 赵云波教授

专业班级 自动化 1304

学 院 信息工程学院

提交日期 2017 年 6 月 1 日

浙江工业大学本科毕业设计论文

基于无人机飞控平台的算法研究

作者姓名：沈菲菲

指导教师：赵云波教授

浙江工业大学信息工程学院

2017年6月

**Dissertation Submitted to Zhejiang University of Technology
for the Degree of Bachelor**

**Algorithm Research Based on UAV Flight Control
Platform**

Student: Shen Feifei

Advisor: Professor Zhao Yunbo

**College of Information Engineering
Zhejiang University of Technology
June 2017**

本科毕业设计的任务书
(用盖过红章的任务书替换)

基于无人机飞控平台的算法研究

摘 要

随着科技的进步与市场需求的发展，无人机从最初的军事用途已扩展到商业层面，大家对于无人机的认知和需求都在逐渐提高，未来民用无人机必将保持较快的发展趋势。如何控制无人机的飞行已成为研究无人机的一大热点，其中航迹规划技术作为无人机自主航行的关键技术更是当下研究人员需要研究解决的问题。

本文的研究内容是基于无人机飞控平台的算法研究，针对无人机航迹规划方面，利用人工势场法和 A*算法来实现对无人机飞行路径的规划，并在 MATLAB 上进行实验的仿真验证。通过比较不同算法之间在航迹规划上的性能差异以及算法本身的优缺点，来综合改进现有的算法，以达到更好的控制效果。论文的主要工作如下：

1. 综述论文研究背景与意义，以及无人机研究的国内外现状，介绍了无人机飞行控制的主要技术问题。

2. 介绍人工势场法的基本原理，并根据算法的基本原理在 MATLAB 上进行仿真验证。针对传统人工势场法在应用于无人机航迹规划中所存在的问题，提出了改进的方法来优化算法，解决了算法无法适应复杂环境而陷入局部停滞不前的问题，并总结了在算法中如何选取参数的方法。

3. 针对 A*算法，根据算法的基本原理在 MATLAB 上实现仿真验证，并在算法实现中加入了伪彩色图，使规划的路径能够更直观地显示出来。

4. 比较两种算法的优缺点，并对所做的工作进行总结。

关键词：无人机，航迹规划，人工势场法，A*算法

Algorithm Research Based on UAV Flight Control Platform

ABSTRACT

With the progress of science and technology and the development of market demand, UAV from the initial military use has been extended to the commercial level. Everyone for the UAV's awareness and demand are gradually improved. The future civilian UAV will remain faster trends. How to control the UAV flight has become a hot spot in the study of UAVs, in which the track planning technology as a UAV autonomous navigation of the key technology is now the researchers need to study and solve the problem.

The research content of this paper is the algorithm research which is based on UAV flight control platform, mainly for UAV flight planning aspects. The artificial potential field method and A* algorithm are used to achieve the UAV flight path planning, and experiment with the simulation in the MATLAB. By comparing the performance differences between the different algorithms in the track planning, the existing algorithms can be improved to achieve better control effect. The main work of the paper is as follows:

1. Summarize the research background and significance of the paper, as well as the current situation of UAV research at home and abroad. Introduce the main technical problems of UAV flight control.

2. Introduce the basic principle of artificial potential field method, and according to the basic principle of the algorithm, it has been implemented on MATLAB for experimental verification. Aiming at the problems existing in the planning of UAV trajectory, an improved method is proposed to solve the problem that the algorithm can not adapt to the complicated environment and fall into the local stagnation state. And how to select a parameter in the algorithm is also solved.

3. According to the basic principle of the algorithm, the A* algorithm has achieved simulation verification in MATLAB. And a pseudo-color map has been added to the algorithm, so that the planning path can be dynamically displayed.

4. Compare the advantages and disadvantages of the two algorithms, and summarize the work I've done.

Key Words: UAV, track planning, artificial potential field method, A* algorithm

目 录

摘要.....	I
ABSTRACT.....	II
第 1 章 绪论.....	1
1.1 无人机的发展历史和研究意义.....	1
1.2 无人机国内外研究现状.....	1
1.2.1 国外无人机研究现状.....	1
1.2.2 国内无人机研究现状.....	2
1.3 飞行控制的主要技术问题.....	3
1.3.1 无人机平台技术.....	4
1.3.2 自主控制技术.....	4
1.3.3 网络化通信技术.....	4
1.4 论文的组织结构.....	5
第 2 章 航迹规划的约束条件.....	6
2.1 航迹规划的自身约束条件.....	6
2.1.1 最大航程.....	6
2.1.2 最小步长.....	6
2.1.3 最大转弯角.....	6
2.2 航迹规划的外部约束条件.....	7
2.2.1 航迹高度.....	7
2.2.2 实时性要求.....	8
第 3 章 基于人工势场法的算法研究.....	9
3.1 基本原理.....	9
3.2 人工势场法的改进.....	10
3.3 增益参数的选取.....	11
3.4 算法的实现步骤.....	12
3.5 仿真结果分析.....	15
第 4 章 基于 A*算法的算法研究.....	19
4.1 基本原理.....	19
4.2 算法的实现步骤.....	20
4.3 仿真结果分析.....	22
第 5 章 总结与展望.....	25
5.1 总结.....	25
5.2 展望.....	25
参考文献.....	27
附录 1: 人工势场法算法程序.....	29

附录 2: A*算法程序.....	34
致谢.....	41

第 1 章 绪 论

1.1 无人机的的发展历史和研究意义

无人机（Unmanned Aerial Vehicle，缩写为 UAV）又称“空中机器人”，是一种由动力驱动、机上无人驾驶、可重复使用的航空器的简称^[1]。与一般的有人机相比，无人机相对具有更多的优势，比如无人机更加安全，避免了人员伤亡的风险，并且持续时间长，使用方便，能替代有人机进入比较恶劣的环境进行作业和执行枯燥重复性的任务。

无人机起源于英国，英国人研制了世界上第一架无人机。20 世纪 70 年代以来，其发展和应用开始得到广泛重视，特别是近几年高技术局部战争中无人机表现出了诸多独特的优势，展现了巨大的作战潜能，促使世界各国对无人机技术的重视达到了前所未有的高度^[2]。起初无人机只用于军事领域，主要用于战场侦察，在战争中常常出现，多次发挥重要的作用，大大减少了有人机和飞行员的损伤。无人机系统的发展经历了靶机—无人侦察机—一次性使用攻击无人机和察/打一体化无人机—无人作战飞机的发展历程，其使命由早期的靶机、情报/监视/侦察、电子软/硬杀伤发展到察/打一体至智能化无人作战飞机^[3]。随着科技的进步与市场需求的发展，无人机的应用已不再仅仅局限于军用方面，而是在民用方面也有了很大的发展，使无人机本身的用途得到了很大的扩展。比如，使用装有高清摄像头的无人机进行空中拍摄，对发生灾难的地区进行航拍以便更好地了解灾区的情况；也可在无线遥控的情况下，从空中进行影视剧的拍摄；同时无人机也可用于对电网进行自主巡航并实时传回所拍摄到的画面，让在监控室监查的工作人员可在电脑屏幕上同步收看传回的画面，并进行相应的操作，这不仅提高了对电网巡检的工作效率，也避免了人为检查线路时的潜在危险等等。

无人机按照结构功能可分为固定翼无人机、扑翼飞行器、小型软式飞船、旋翼无人机。不同类型和用途的无人机在作战过程中承担着相应的任务，而不同的无人机之间飞行性能的差异还是很大的，如何提高无人机之间的协同控制效率和视频导航能力，实现资源的有效利用还是具有挑战性的，因此对无人机系统的研究还是有一定意义的。

1.2 无人机国内外研究现状

1.2.1 国外无人机研究现状

目前在全世界范围内已掀起了无人机的研制热潮，全球共有 57 个国家研制和发展

无人机，种类达 1000 多种，其中已成为无人机产品的有 400 多种^[4]。这些无人机的研究主要以美国和以色列为代表，他们的发展程度是最高最快的，技术应用方面也是范围最广最为成熟的，其次是英国、法国、德国等欧洲国家。其中美国的研究成就拥有着主导地位，引领着世界的研究热潮，其无人机研究水平是居于世界前列的。

针对多无人机的协同侦察搜索、目标的跟踪定位和攻击以及编队控制、碰撞避免等特定的情况，国外已经开展了大量的无人机飞控系统自主协同控制方面的研究。此外，美国也比较重视创新意识，他们在无人机技术方面追求不断地突破，积极研究新型的无人机，而不局限于传统无人机。例如隐身无人机、变体无人机、太阳能无人机、倾转旋翼无人机等都是他们所研究的新型无人机类型。

当然除了美国，还有一些国家对无人机的研究也是具有一定的水平。比如以色列这个国家，在研究中、小型战术无人机方面已达到了一定的成就，其水平可与美国相提并论，对于无人机的开发、制造和使用方面具有很多有用的经验。以色列在两次中东战争中创新地使用了无人机，树立了用无人机支援地面作战的典范^[5]。欧洲一些发达国家也不愿落后，这些年来通过共同合作研发等方式已取得了一定的成就，正在逐渐缩小与美国的差距。由法国达索公司领衔，意大利、西班牙、瑞典、希腊、瑞士等 5 个国家共同参与研制的“神经元”（Neuron）无人隐形作战飞机首架验证机制造完成，这款无人机配备两个武器舱，并装备有光电目标跟踪系统，性能极其先进，已经在法国南部的伊斯特尔空军基地首飞成功^[6]。如今，已不仅仅只有欧美国家在研究无人机技术了，像日本、韩国、伊朗等国家也在积极地开展着无人机系统的研究工作。

为了使无人机可以自主执行任务，各研究机构致力于把图像传感器加到无人机系统上，使其具有自主导航和定位的功能。在文献[7]中设计了用于状态估计的图像传感器，利用视觉传感器来估计无人机的速度和相对位置，同时也阐述了两种不同的基于视觉的稳定四旋翼无人机飞行控制方法。一种是基于单应性估计技术和光流计算，另一种是利用光流方法调节无人机的速度。

1.2.2 国内无人机研究现状

相对国外而言，国内的无人机产业起步稍晚，因受历史条件、技术水平等各方面的影响，我国的无人机技术在总体发展水平上与美国等强国相比还是有很大差距的，但通过不断地努力，这种差距正在缩小。我国对无人机的研究已经向前迈进了一大步，无人机飞行控制的性能在不断地提高，形成了相对比较完善的无人机产业链。国内对无人机

的研究，主要是解决了飞行控制和平台自主能力不强的问题，像周锐等提出了无人战术飞机分层式智能控制结构，张新国等探讨了不确定环境下的自主飞行控制面临的挑战，唐强等探讨了多无人机自主飞行控制中的飞行规划与重规划、自主飞行控制结构和自主着陆等问题^[8]。针对各种问题，研究人员也提出了许多对应的算法进行控制。文献[9]在研究多无人机编队与集群控制问题时，设计了一种分布式控制算法。通过人工势场法和回转力方法的结合运用，使得无人机在障碍空间中避免碰撞。文献[10]提出了分布式多无人机目标状态融合估计算法。这种算法以多无人机协同跟踪地面目标为背景，结合目标状态融合估计，实现了多无人机跟踪航迹优化问题。

随着我国科技的不断进步和国民经济的发展需求，无人机广泛应用于受灾救援、电力巡线、环境监测、影视拍摄等多个领域。无人机的应用已不仅仅是军事方面，在民用范畴上的需求和应用也是非常的有发展前景。在日常生活工作中我们已能深切的感受到无人机带来的优越性。预计到 2025 年，国内无人机航拍市场规模约为 300 亿，农林植保约为 200 亿，安防市场约为 150 亿，电力巡检约为 50 亿，总规模将达 750 亿，但相应的监管法律法规缺失，导致无人机安全问题频发，持续得不到改善，会极大的限制民用无人机行业的发展^[11]。就国内而言还没有足够完善的法律法规来管控对无人机的使用，无人机与其他飞行器之间的航空安全问题，对地面上的人和房屋建筑的安全责任等等，诸如此类问题都急需我们进行深入研究，尽早提出相应的法律法规来进行管理，这样才有利于无人机行业的快速发展。

总的来说，无人机的发展还是具有较好的前景。研制或提出更多新颖算法，解决当前未完成的难题，对未来发展前景、趋势做出更多可靠、可行的预测，将成为未来世界无人机领域不可忽视的力量^[12]。随着科技的不断进步，信息化时代的到来，无人机将发挥越来越重要的作用，我们的生活将和无人机有着深深地联系。研究无人机飞控技术是一项具有挑战性的技术难题，如何优化算法来提高无人机系统的性能是我们不断研究和实现的目标。

1.3 飞行控制的主要技术问题

本次研究主要针对无人机飞控平台上的算法，对于无人机的飞行控制，其中有一些关键性的技术问题，主要包括：无人机平台技术、自主控制技术和网络化通信技术等。

1.3.1 无人机平台技术

无人机平台技术包括了高效气动力技术、隐身技术和气动载荷设计技术等等。高效气动力技术是能够让无人机的性能达到良好水平的重要技术支持,可以使无人机满足相应的设计要求,实现无人机保持高续航、高升力的能力。隐身技术分为外形隐身技术、等离子体隐身技术和主动隐身技术,其目的都是运用新的技术来降低无人机的可探测性。气动载荷设计技术是以轻量化布局为目的,结合无人机的飞行控制方式来开展的,有利于提高无人机的综合性能。

1.3.2 自主控制技术

无人机自主控制技术包括了规划与协同技术、自主决策技术和执行任务技术等。规划与协同技术主要可分为两个方面:一个是飞行的航迹规划,另一个是飞行器之间的协同控制。无人机在根据不同的外部环境约束条件下,能够自主地形成飞行轨迹,这种航迹规划能力是无人机自主控制系统所必须具备的基本能力。航迹规划对算法时间有着严格的限制条件,不仅要考虑环境的约束还要考虑执行任务之间的关系。航迹规划的方法包括最优式规划方法和启发式规划方法两大类,其中包括遗传算法、人工神经网络算法、蚁群算法、粒子群优化算法等等。无论是单机飞行还是多机编队协同控制,都需要解决如何有效地避免碰撞,规划航行轨迹,完成任务目标。当处在更加复杂的环境下时,无人机还需要有比较强的自主决策技术,从而能够更好地适应不同的状况。

1.3.3 网络化通信技术

无人机作为作战部队的力量倍增器,未来将会对通信系统的能力有持续不断增加的要求^[13]。通讯系统是无人机与外界保持联系的一个神经网络,实现了无人机飞行时与地面空间站之间的信息交互。提供有效可靠的通信是提高基于信息系统的体系作战能力的基础^[14]。针对无人机集群作战、协同作战以及网络化作战的应用需求,应突破无线宽带分布式动态多址接入、实时鲁棒的宽带传输、数据链网络顽存等关键技术,构建无人机集群数据链自适应网络体系,为实现实时、宽带、安全的无人机集群数据链提供技术支撑^[4]。通过网络编码技术结合路由技术的方式,选择其中编码机率最大的一条可行路径进行数据的传输,从而在不外加任何网络时延的基础上提高了网络质量,实现了网络通信的大容量数据传输。

1.4 论文的组织结构

本文章节内容安排如下：

第一章从总体上介绍了无人机的发展历史和研究意义，以及国内外无人机的研究现状，同时也介绍了飞行控制的主要技术，最后介绍了论文的组织结构和章节安排。

第二章介绍了无人机航迹规划的几个约束条件，主要包括无人机在飞行过程中因自身限制而所受的最大航程，最小步长和最小转弯半径等内在约束条件以及因环境因素而所受的航迹高度，实时性要求等外部约束条件。

第三章详细介绍了人工势场法的基本原理，并实现了算法的仿真实验。针对算法中的不足之处，提出相应的改进办法，并在 MATLAB 上进行实现，验证改进方法的优越性。同时对算法中的参数选定进行总结，提出了相应的选取原则。

第四章介绍了 A*算法的基本原理，并实现了算法的仿真实验。总结了 A*算法与人工势场法的优缺点。

第五章对所做的工作进一步总结，并对研究内容中的不足之处做出展望。

第 2 章 航迹规划的约束条件

无人机航迹规划是根据任务需要，在给定的规划空间里，在满足地形限制、敌方雷达、防空火力威胁、气象分布、自身机动特性等一系列约束条件下，找到一条从起始点到目标点的最满意或最优的路径，使得所花代价最小，安全系数最高，并保证任务能圆满完成^[15]。航迹评价作为航迹规划问题中的一个重要组成部分，其评价的标准直接影响到了所规划的无人机航迹是否可行。在进行航迹评价之前，需要先确定影响航迹性能的外部条件和无人机自身的约束条件，然后根据这些约束条件去规划路径。

2.1 航迹规划的自身约束条件

为了确保算法所规划出的航迹路线是可行的，能够符合无人机的气动性能和机动性能，我们需要将无人机本身的约束条件考虑在内。无人机在巡航飞行过程中主要受最大航程、最小步长、最小转弯半径等条件约束^[16]。

2.1.1 最大航程

最大航程是指在无人机飞行过程中总的航路里程不能太大而超过一定范围，因为无人机在飞行时所携带的能源是有限的，无人机不可能一直飞下去。最大航程这个条件对于无人机从起始点飞到目标点的过程中所经历的可行航迹点都将具有约束作用。该约束可表示为 $\sum_{i=1}^{n-1} |X_i X_{i+1}| \leq L_{\max}$ ，其中 $|X_i X_{i+1}|$ 为相邻两个航迹点之间的间隔长度。

2.1.2 最小步长

最小步长是指无人机因为自身存在的惯性，在改变航行方向之前需要先直飞的最小距离。这个约束与导航系统也有关系，因为不停地改变无人机的飞行方向会使得导航误差加大，同时无人机的能源消耗也会加大，所以需要有这个最小距离进行缓冲。该约束可表示为 $|X_i X_{i+1}| \geq l_{\min}$

2.1.3 最大转弯角

由于无人机本身的机动性能限制，无人机要想改变航行方向也只能在最大转弯角内

改变。该约束可描述为按照无人机飞行的当前航迹点的方向，将满足最大转弯角约束的邻近航迹点作为候选航迹点。如图 2-1 所示， s_i 为无人机所在的当前航迹点， s_{i-1} 为上一个航迹点位置，则根据当前航迹点飞行的不同方向，结合最大转弯角的约束，可以确定不同的候选航迹点集合 $\{s_j\}$ 。

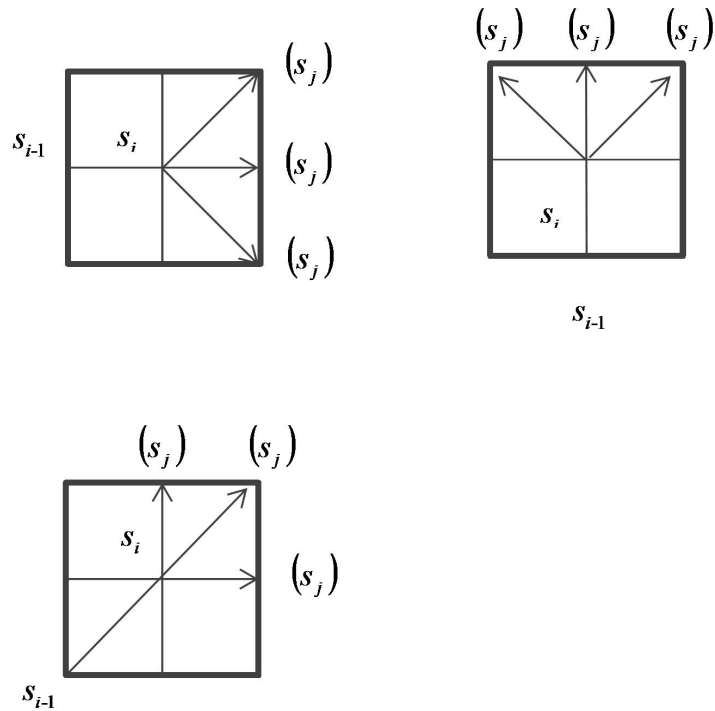


图 2-1 候选航迹点示意图

2.2 航迹规划的外部约束条件

航迹规划的外部约束条件指的是无人机在飞行中会受到环境及任务本身的要求所带来的约束，包括航迹高度，实时性要求等。

2.2.1 航迹高度

在航迹规划中，无人机容易受到地面障碍物的影响，包括房屋建筑，山峰树木，甚至电线电缆。这些障碍物都是具有一定高度的，导致无人机在飞行过程中不能离地面高

度过低，否则很容易发生与地面物体碰撞的可能。所以在无人机飞行中应保持必要的高度，但也不能飞得太高，因为在飞行高度提升的同时总的航路里程也会随之加大，这就使得有可能会不满足最大航程这一约束条件。因此在无人机航迹规划的时候需要我们协调好飞行高度。

2.2.2 实时性要求

当任务区域内的障碍物位置对于无人机而言是已知的时候，可以选用合适的算法进行航迹规划，得到一条最优的航迹路线。但在实际情况中环境比较复杂，障碍物的位置可能是在变动的，或者飞行途中任务发生变化时，原先规划好的路径可能就不适用了，需要重新规划出一条可行的路径。这就要求无人机有航迹在线重规划的能力，能够实时获取环境信息，并根据这些信息选用合适的算法来实现航迹的重新规划。

由此可见，无人机航迹规划的约束条件十分复杂，为简化问题的研究，本文主要是建立在以下假设上进行的：假设障碍物都是静态的，无人机在任务区域内以恒定的速度从起始点飞到目标点，不考虑任务区域的高度信息，即将无人机航迹规划问题简化为在二维平面上的路径规划问题。

第3章 基于人工势场法的算法研究

3.1 基本原理

人工势场法的基本思想是通过构造虚拟的目标引力场和障碍物斥力场，在两个势场的共同作用下，沿着势场合力的方向来寻找一条无碰撞的路径。以目标点所在的位置为引力中心来构造引力势场，势场力的大小随无人机与目标点距离的减少而单调递减，且方向指向靠近目标点方向。以障碍物所在位置为斥力中心构造斥力势场，其产生的斥力大小会随着无人机与障碍物之间距离的减少而迅速增大，且方向指向远离障碍物方向。然后按各个障碍物和目标点所产生的人工势场力的总和来确定无人机前进的方向，从而实现无碰撞路径规划。

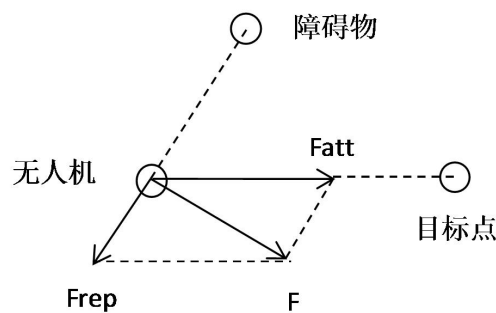


图 3-1 无人机受力图

假设 q 是处于二维平面中的一点，则该点的引力势函数可定义为：

$$U_{att}(q) = \frac{1}{2} k \rho_G^2(q) \quad (3-1)$$

其中 $\rho_G(q)$ 是 q 到目标点的相对距离， k 是引力增益系数。相应的 q 点受到 q_G 点的吸引力为该点引力势函数的负梯度，可表示为：

$$F_{att}(q) = k \rho_G(q) \quad (3-2)$$

当无人机接近目标点的过程中这个变量的值逐渐收敛到零。

q 点的斥力势函数可定义为：

$$U_{rep}(q) = \begin{cases} \frac{1}{2}m \left[\frac{1}{\rho(q)} - \frac{1}{\rho_0} \right]^2, & \rho(q) \leq \rho_0 \\ 0, & \rho(q) > \rho_0 \end{cases} \quad (3-3)$$

其中 $\rho(q)$ 是 q 到障碍物的距离, m 是斥力增益系数, ρ_0 是障碍物的影响距离, 即当离开一定距离的时候, 障碍物对无人机没有斥力影响, 其值的大小可以根据目标点和障碍物分布的具体情况而定。 q 点受到障碍物的斥力为该点斥力势函数的负梯度:

$$F_{rep}(q) = \begin{cases} m \left[\frac{1}{\rho(q)} - \frac{1}{\rho_0} \right] \frac{1}{\rho(q)^2}, & \rho(q) \leq \rho_0 \\ 0, & \rho(q) > \rho_0 \end{cases} \quad (3-4)$$

无人机所受到的总的势场函数为:

$$U = U_{att}(q) + U_{rep}(q) \quad (3-5)$$

总的合力为:

$$F = F_{att}(q) + F_{rep}(q) \quad (3-6)$$

这个力决定了无人机的运动方向。但在实际情况中, 当用这个力来控制无人机飞行时, 如果目标点与障碍物靠得比较近, 也就是说在障碍物的影响范围内时, 无人机将可能到达不了目标点。因为当无人机向目标点靠近的同时, 也越来越靠近障碍物, 这样会使得吸引力减小的同时斥力增加, 无人机受到的合力可能达到零而停滞不前。这也是用人工势场法来规划无人机航迹路线时存在的不足之处。

3.2 人工势场法的改进

根据上述的分析, 利用人工势场法在进行无人机航迹规划的时候, 如果障碍物设置在目标点附近, 当无人机在引力势场的作用下朝目标点方向飞行的同时也在向障碍物靠近, 而如果此时无人机受到的斥力合力与引力的大小相等, 方向相反, 那么无人机将永远不能到达目标点。为了使无人机能打破这种局面, 提出可以使无人机在所受合力为零的情况下, 略微加大引力的大小从而打破当前受力平衡的状态, 使无人机又受到引力而朝目标点移动。

改进的人工势场法在斥力与引力相等的情况下, 在原来的基础上添加了参数 a , 合力表达式为:

$$F = F'_{att}(q) + F_{rep}(q) = a - k\rho_i(q) + \sum_{i=1}^n \left\{ m \left[\frac{1}{\rho_i(q)} - \frac{1}{\rho_0} \right] \frac{1}{\rho_i(q)^2} \right\} \quad (3-7)$$

其中 n 表示障碍物个数。

3.3 增益参数的选取

当无人机所受的引力和斥力总和分布在一条直线上，并且方向正好相反时会出现局部极小点，即在这种情况下所受合力为零，无人机将停滞不前或产生局部震荡，此时无人机所受的引力和斥力之和为：

$$F_0(q) = F_{att}(q) + F_{rep}(q) = \|F_{att}\|a_0 + \|F_{rep}\|a_1 \quad (3-8)$$

由于 $a_0 = -a_1$ ，所以上式可改写为：

$$\|F_0\| = \|F_{att}\| - \|F_{rep}\| \quad (3-9)$$

为了避免合力为零，必须使得总的势场力大于零，因此

$$\|F_{att}\| - \|F_{rep}\| > 0 \quad (3-10)$$

将式(3-2)和(3-4)代入上式中，可得

$$k\rho_G(q) - m\left(\frac{1}{\rho(q)} - \frac{1}{\rho_0}\right)\frac{1}{\rho(q)^2} > 0 \quad (3-11)$$

由于 $k, m > 0$ ，所以有

$$\frac{k}{m} > \left(\frac{1}{\rho(q)} - \frac{1}{\rho_0}\right)\frac{1}{\rho(q)^2\rho_G(q)} = \left(\frac{1}{\rho(q)} - \frac{1}{\rho_0}\right)(\rho(q) - s)^{-2}\left(\frac{1}{\rho(q)} - \frac{s}{\rho(q)^2}\right) \quad (3-12)$$

式中 $s = \rho(q) - \rho_G(q)$ 表示障碍物和目标点之间的距离。

令 $t = \frac{1}{\rho(q)}$ ，设 $\alpha = \frac{1}{\rho(q)} - \frac{s}{\rho(q)^2} = -st^2 + t$ ，则有

$$\begin{cases} \alpha \leq 0 & s < \rho(q) < l \\ \alpha > 0 & l > \rho(q) > \rho_0 \end{cases} \quad (3-13)$$

综上所述，有：

$$\frac{k}{m} > \left(\frac{1}{\rho(q)} - \frac{1}{\rho_0}\right)(\rho(q) - s)^{-2}\alpha(\rho(q)) \quad (3-14)$$

令

$$k_{\max} = \max\left\{\left(\frac{1}{\rho(q)} - \frac{1}{\rho_0}\right)(\rho(q) - s)^{-2}\alpha(\rho(q))\right\}, l < \rho(q) < \rho_0$$

则

$$k_{\max} \leq \max \left\{ \left(\frac{1}{\rho(q)} - \frac{1}{\rho_0} \right) \right\} \max \{ (\rho(q) - s)^{-2} \} \max \left\{ \frac{1}{\rho(q)} - \frac{s}{\rho(q)^2} \right\}, l < \rho(q) < \rho_0 \quad (3-15)$$

其中

$$\max \left(\frac{1}{\rho(q)} - \frac{1}{\rho_0} \right) = \frac{1}{l} - \frac{1}{\rho_0}$$

$$\max (\rho(q) - s)^{-2} = (l - s)^{-2}$$

令 $t = \frac{1}{\rho(q)}$, 则有

$$\max \left(\frac{1}{\rho(q)} - \frac{s}{\rho(q)^2} \right) = \max (-st^2 + t), \quad \frac{1}{\rho_0} < t < \frac{1}{l}$$

当 $t = \frac{1}{2s}$ 时, 取得最大值, 则

$$\max \left(\frac{1}{\rho(q)} - \frac{s}{\rho(q)^2} \right) = \begin{cases} \frac{\rho_0 - s}{\rho_0^2} & \frac{1}{\rho_0} > \frac{1}{2s} \\ \frac{1}{4s} & \frac{1}{\rho_0} < \frac{1}{2s} \end{cases} \quad (3-16)$$

即

$$\frac{k}{m} > \begin{cases} \left(\frac{1}{l} - \frac{1}{\rho_0} \right) (l - s)^{-2} \frac{\rho_0 - s}{\rho_0^2} & \frac{1}{\rho_0} > \frac{1}{2s} \\ \left(\frac{1}{l} - \frac{1}{\rho_0} \right) (l - s)^{-2} \frac{1}{4s} & \frac{1}{\rho_0} < \frac{1}{2s} \end{cases} \quad (3-17)$$

综上所述便可得到引力场和斥力场增益系数的大致选择范围, 在实际中可以根据情况选取合适的 k 和 m , 以尽量避免所受合力为零的情况出现。

3.4 算法的实现步骤

针对人工势场法, 在矩形任务区域内以圆形区域来描述静态障碍物, 自定义起始点位置和目标点位置, 各个路径航迹点之间以等步长前进。人工势场法所建立的航迹规划空间地图模型如图 3-2 所示, 红色小圆表示无人机起始点, 红色倒三角表示目标点, 蓝色小圆表示障碍物所在位置。

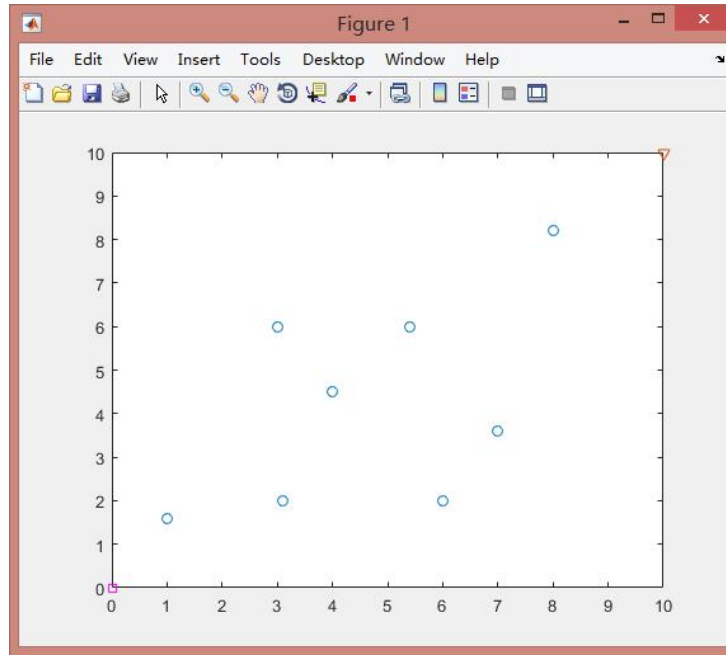


图 3-2 人工势场法航迹规划空间地图模型

在以上所建立的空间地图模型的基础上对无人机进行航迹规划，无人机从起始位置开始在受到引力和斥力的合力作用下朝目标点前进。引力会随着无人机靠近目标点而减小，相反障碍物的斥力会随着无人机与之距离的减小而增大，但这个斥力只有在一定的范围内才起作用，因此对于每一个障碍物而言，其斥力仅仅在其周围对无人机有导航作用。改进后的人工势场法具体实现步骤如下：

(1) 建立势场模型。首先确定引力场和斥力场的增益系数 k 和 m ，障碍物的影响距离 ρ_0 ，以及移动步长 l 。同时确定无人机的起始位置 S 和目标点位置 G ，同时按照具体情况设置障碍物的位置及障碍个数 n 。

(2) 根据建好的势场模型计算出无人机所受的引力和斥力情况，同时计算无人机的运动航向角 β_k 。判断所受的合力是否为零，若为零，则在原来合力的基础上增加一个引力 a ，否则继续下一步。

(3) 按照下式进一步计算无人机下一个航迹点的位置：

$$\begin{aligned}x_{k+1} &= x_k + l \cos \beta_k \\y_{k+1} &= y_k + l \sin \beta_k\end{aligned}$$

(4) 无人机移动到 (x_{k+1}, y_{k+1}) 后，更新当前的位置，即 $x_k = x_{k+1}$ ， $y_k = y_{k+1}$

(5) 判断无人机是否到达目标点，如果已经到达目标点则结束算法，停止航迹规划过程，否则就返回第二步继续执行。

改进后的人工势场法流程如下图 3-3 所示：

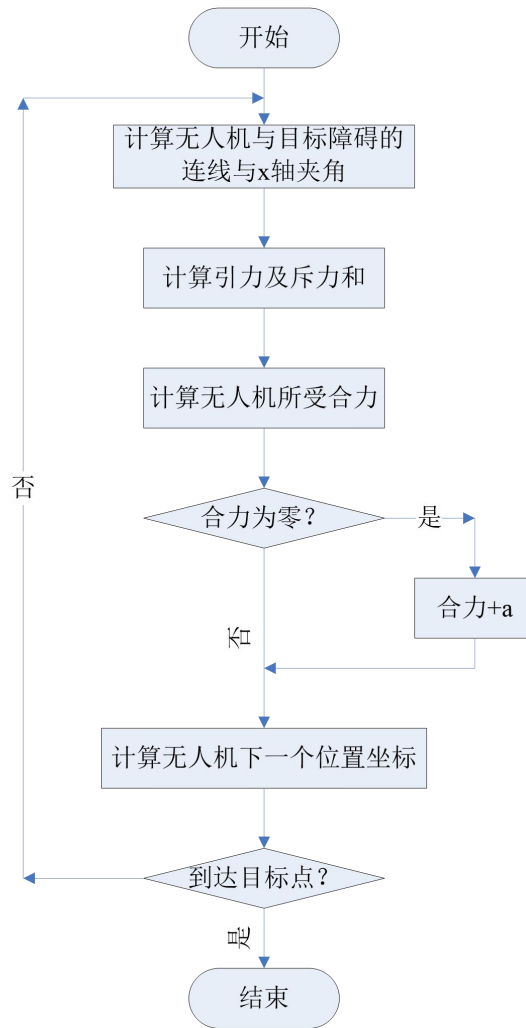


图 3-3 改进后的人工势场法算法流程

算法中所用到的主要函数如下表 3-1 所示：

表 3-1 人工势场法所用到的函数

函数名称	功能简介
algorithm_angle(X,Xsum,n)	计算角度函数
algorithm_Attraction(X,Xsum,k,Attract_angle)	计算引力函数
algorithm_Repulsion(X,Xsum,m,angle_re,n,Po)	计算斥力函数

函数具体的功能及各输入参数的意义表示如下：

algorithm_angle(X,Xsum,n)函数：计算角度函数，输入的参数分别表示当前航迹点的

坐标，目标和障碍物的坐标，障碍数目。该函数用于计算出当前航迹点与障碍物和目标点的连线与 x 轴之间的夹角度数。

algorithm_Attraction(X,Xsum,k,Attract_angle)函数：计算引力函数，输入的参数分别表示当前航迹点坐标，目标点坐标，引力增益系数，x 轴分量和引力方向之间的角度。该函数用于计算出目标点对当前航迹点所施加的引力在 x 轴和 y 轴上的分量大小。

algorithm_Repulsion(X,Xsum,m,angle_re,n,Po)函数：计算斥力函数，输入的参数分别表示当前航迹点坐标，障碍物坐标，斥力增益系数，x 轴分量和斥力方向之间的角度，障碍物个数，障碍物影响半径。该函数用于计算出障碍物对当前航迹点所施加的斥力在 x 轴和 y 轴上的分量大小。

3.5 仿真结果分析

本文在 PC 机上，使用 MATLAB R2015b 编写仿真程序。假设无人机飞行的地形区域为一个 10×10 的矩形区域内，起始位置在点 $S(0,0)$ ，目标位置为点 $G(10,10)$ 。各参数设定为：障碍影响距离 $\rho_0 = 10$ ，步长增益因子 $l = 0.2$ ，以及 8 个障碍物点具体设定位置如表 3-2 所示，并选取一个变量 K 用来记录无人机到达目标点时所经历的航迹点个数。因为相邻航迹点的步长相等，所以可以用 K 的值来评估这条路径的长短， K 值越大路径越长。根据式(3-17)可计算出引力增益系数和斥力增益系数之间的关系： $\frac{k}{m} > 1.5 \times 10^{-3}$ 。

表 3-2 障碍物位置分布

中心坐标	
(1, 1.6)	(6, 2)
(3.1, 2)	(5.4, 6)
(4, 4.5)	(7, 3.6)
(3, 6)	(8, 8.3)

选取不同的增益系数 k 和 m ，使得无人机在不同的情况下规划航迹路线，比较他们之间的路径长短。图 3-4 为引力增益系数 k 取 9，斥力增益系数 m 取 5 时的航迹规划图，从仿真结果可以看出路径长度 $K = 72$ 。图 3-5 和图 3-6 分别是 $k = 9$ ， $m = 10$ 以及 $k = 3$ ， $m = 5$ 时的航迹规划图，其路径长度 K 分别为 73 和 74。

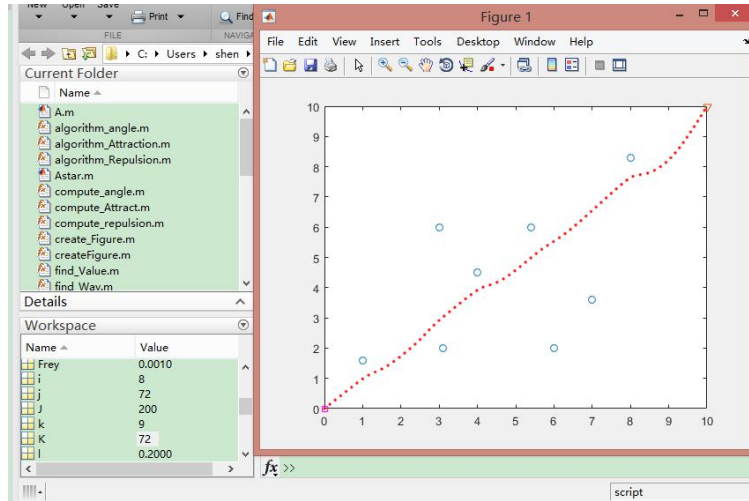


图 3-4 $k = 9$, $m = 5$ 时的航迹规划图

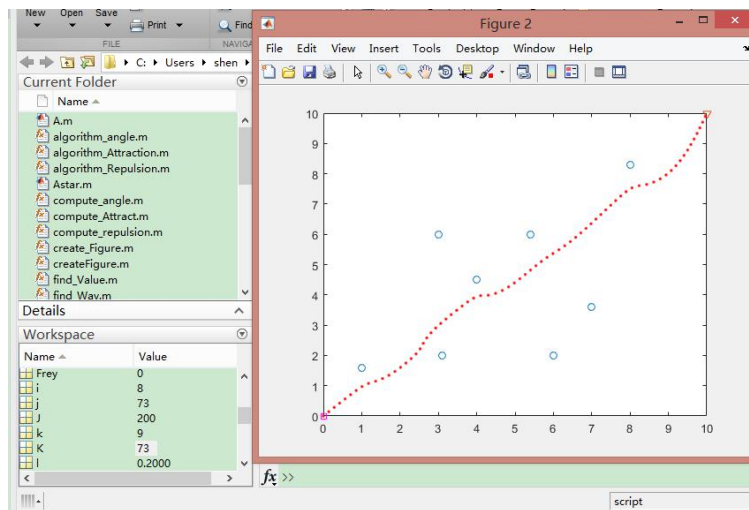


图 3-5 $k = 9$, $m = 10$ 时的航迹规划图

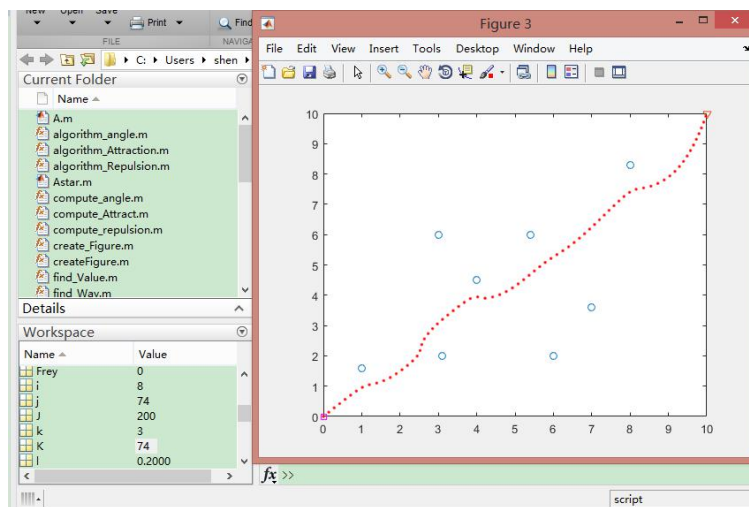


图 3-6 $k = 3$, $m = 5$ 时的航迹规划图

更进一步的可以对不同的 k , m 取值进行仿真, 得到对应的 K 值, 然后进行比较, 从中得出较佳的参数选取。当 $m=5$ 时, 取不同的 k 值, 得到对应的路径长度, 结果如表 3-3 所示; 当 $k=9$ 时, 取不同的 m 值, 得到对应的路径长度, 仿真结果如表 3-4 所示。

表 3-3 $m=5$ 时取不同的 k 值相应的 K

k	3	4	5	6	7	8	9	10	11	12	13	14	15	16
K	74	74	73	73	73	73	72	72	72	72	72	72	72	72

表 3-4 $k=9$ 时取不同的 m 值相应的 K

m	3	4	5	6	7	8	9	10	11	12	13	14	15	16
K	72	72	72	73	73	73	73	73	74	74	74	75	75	75

从表中可以看出, 当 $m=5, k \geq 9$ 时, K 值最小, 即路径是最短的; 当 $k=9, m \leq 5$ 时, K 值最小, 即路径最短。所以综上所述, 可以分别取引力增益系数 $k=9$, 斥力增益系数 $m=5$ 。

针对人工势场法在特殊情况下无法到达目标点的问题, 本文也对算法进行了优化, 并在仿真平台上进行了验证。仿真结果如下图 3-7 和图 3-8 所示。

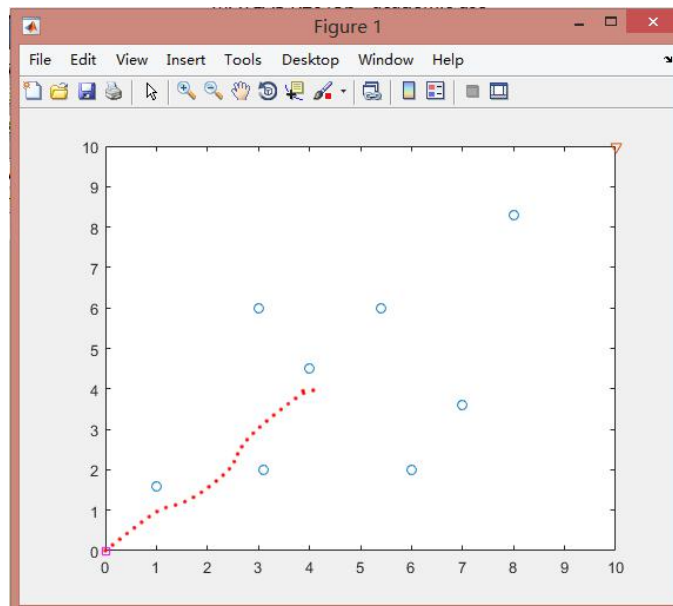


图 3-7 算法改进前的航迹规划图

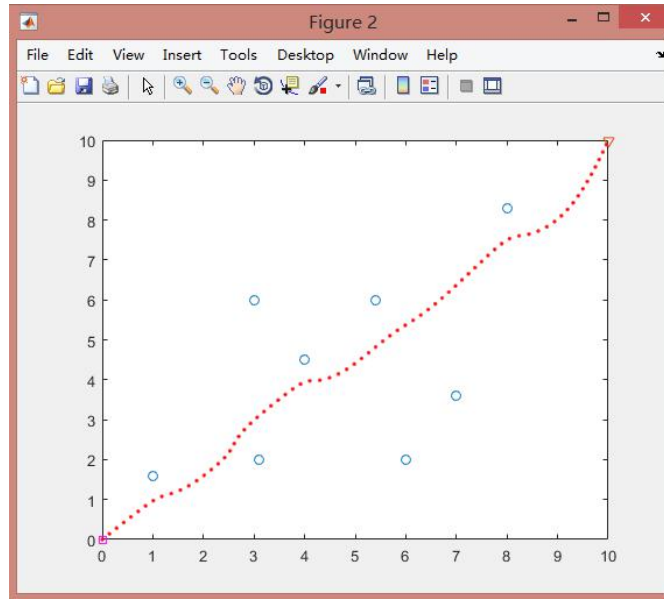


图 3-8 算法改进后的航迹规划图

从仿真结果可以看出，人工势场法能基本实现对无人机的航迹规划，使无人机从起始点到达目标点，并避免与障碍物碰撞。并且改进后的算法能很好的解决无人机因所受合力为零而停滞不前的问题。这种算法相较于其他算法而言，其规划速度较快，在数学描述上简洁，计算简单易于实现，得到的路径也是相对比较安全的，有良好的实时性，便于我们能够实时控制，适用于在目标点附近障碍物离得较远的情况下。同时也可以看出，当斥力增益 m 、引力增益 k 等参数选取不一样时会影响最终的航迹规划效果。

第 4 章 基于 A*算法的算法研究

4.1 基本原理

A*算法是一种启发式搜索算法，需要一个由点和边组成的网络空间，在此目标空间中引入启发信息，由定义好的代价函数确定节点拓展的规则，最终得到两点之间的最优路径^[17]。A*算法其实是将任务区域描述为一个栅格图，其中存在 N 个已知障碍物，无人机在飞行过程中需躲避这些障碍物，以确保飞行的安全。无人机从起始点 S 开始，从邻近的 8 个航迹点中选出代价估计最小的点作为下一个航迹点位置，重复选取直到到达目标位置 G。A*算法的代价估计可以看做是两部分代价的总和，即已经确定的代价和将要付出的代价，该代价估计函数可表示为：

$$f(n) = g(n) + h(n) \quad (4-1)$$

其中 $f(n)$ 是从起点经由航迹点 n 到达目标点的代价估计， $g(n)$ 是起始点到航迹点 n 的实际代价， $h(n)$ 是从航迹点 n 到目标点的最短路径的估计代价。

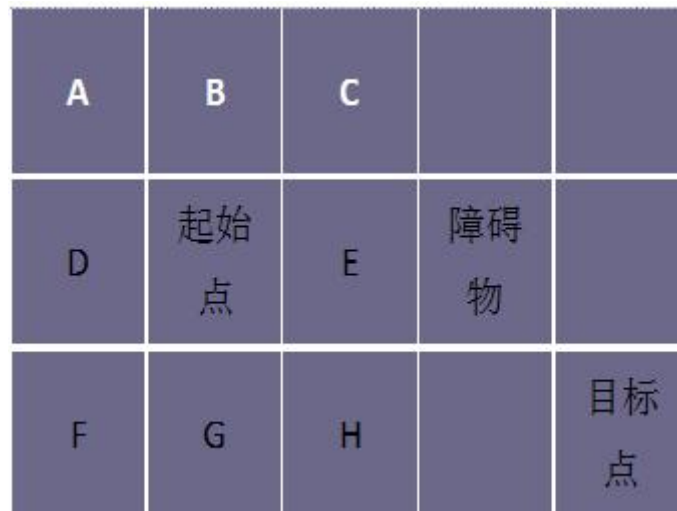


图 4-1 A*算法原理图

如图 4-1 中所示，首先假设每次移动到邻近的单元格时就表示无人机飞行了一个距离，所以从起始点到 A 点就移动了两个距离，即 $g(A) = 2$ ；从 A 点到目标点，忽略障碍物的阻拦，仅计算两点之间的距离，则移动了 6 个距离，即 $h(A) = 6$ 。所以

$f(n) = g(n) + h(n) = 8$ ，即从起始点经由航迹点 A 到目标点的代价估计为 8。同理可计算起始点相邻 8 个方向的估计代价。

把起始点相邻的所有航迹点的代价估计进行比较，选出其中代价最小的一个航迹点，把这个航迹点称为 P 节点，设置 P 节点为当前航迹点，并把起始点设置为 P 节点的父节点。然后再判断 P 节点邻近 8 个方向的有效点，选出其中 $f(n)$ 最小的航迹点，并把 P 节点设置为其父节点，依次类推从而形成一条父节点线，这条父节点线就是用 A* 算法所规划的无人机航迹路径，其路径可表示为

$$P = \{S, S(x_1, y_1), \dots, S(x_k, y_k), \dots, S(x_{K-1}, y_{K-1}), G\} \quad (4-2)$$

4.2 算法的实现步骤

针对 A* 算法，将任务区域分别进行 m 等分和 n 等分，则任务区域可描述为由 $(m+1)(n+1)$ 个节点构成的栅格图。在任务区域内会有 N 个已知障碍物存在，无人机需避免与这些障碍物发生碰撞，以确保在飞行过程中的安全问题。A* 算法所建立的航迹规划空间模型图如下图 4-2 所示，黄点为目标点，绿点为起始点，黑色部分为障碍物所在位置，其余白色部分为可飞行安全区域。

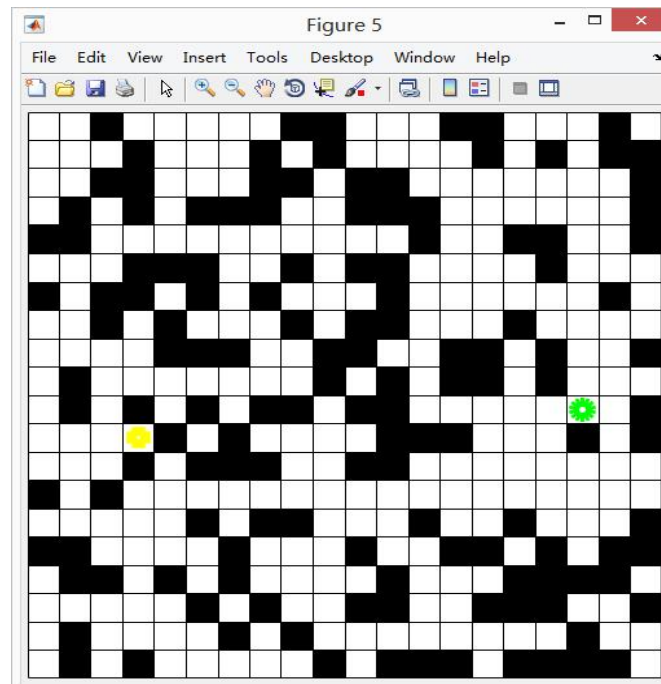


图 4-2 A* 算法航迹规划空间地图模型

A*算法在运用中需要用到两张表：Open 表和 Close 表。

其中 Open 表用于记录所有可考虑选择的航迹点；Close 表用于记录所有不再考虑的航迹点。A*算法的具体步骤如下：

(1) 将起始点记录为当前点 S，把它存入 Close 表中。

(2) 寻找航迹点 S 所有的邻近点，如果这个航迹点不可行或者已经在 Close 表中，则忽略它选择其他邻近点。若这个邻近点没有在 Open 表中也没有在 Close 表中，则计算出该邻近点的代价估计值 $f(n)$ ，并将它们放入 Open 表，同时设置它们的父节点为 S。

(3) 判断 Open 表是否已经为空，如果已经为空说明在到达目标点之前已经搜索完了所有可能的航迹点，航迹搜索失败，算法结束；否则搜索继续。

(4) 从 Open 表中找到一个 $f(n)$ 值最小的航迹点，作为下一个当前点 S。判断该点是否为目标点，如果是，则搜索成功，到达目标位置；否则搜索继续，返回步骤 (2)。

(5) 搜索结束后，从目标点开始，沿着每一个航迹点的父节点移动，直到回到起始点为止，这条路径就是所规划的无人机航迹路线。

A*算法流程如下图 4-3 所示：

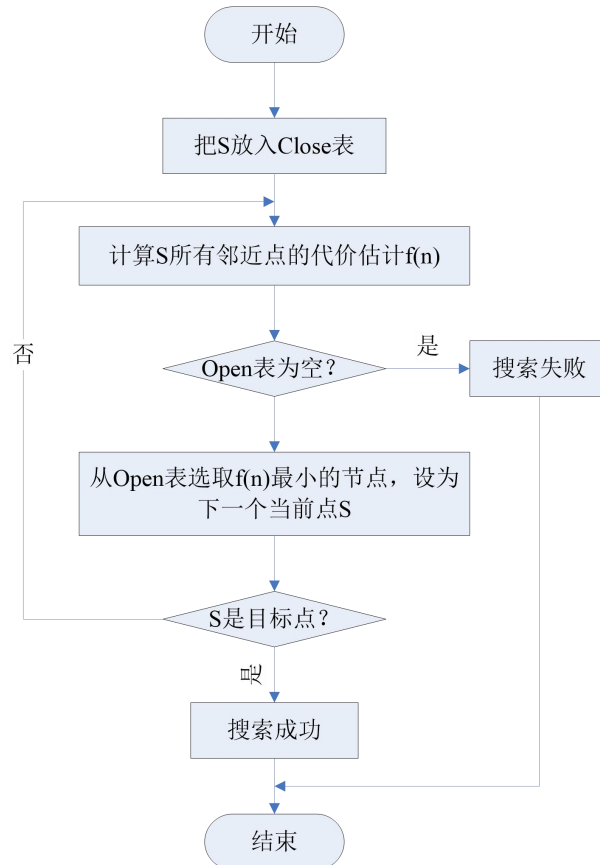


图 4-3 A*算法流程

算法中所用到的主要函数如下表 4-1 所示：

表 4-1 A*算法所用到的函数

函数名称	功能简介
initialization(n,obstacle)	初始化函数
create_Figure(view,cost,startpoint,goalpoint)	创建视图函数
find_Value(point,costpoint,view,goalpoint,method)	计算代价估计函数
find_Way(goalpoint,direction)	画出路径函数

函数具体的功能及各输入参数的意义表示如下：

initialization(n,obstacle)函数：初始化函数，其中 n 表示所设置的栅格地图模型中所划分的格数， $obstacle$ 表示障碍物在所建地图中所占的比例。该函数用于设置起始点和目标点的位置，以及整个地图界面的障碍物布局。

create_Figure(view,cost,startpoint,goalpoint)函数：创建视图函数，其中 $view$ 表示所设置的地图界面， $cost$ 表示所存储的各个节点的实际代价， $startpoint$ 表示起始点的位置， $goalpoint$ 表示目标点的位置。该函数用于画出起始点和目标点，以及障碍物分布，并应用伪彩色图以示区分。

find_Value(point,costpoint,view,goalpoint,method)函数：计算代价估计函数，其中 $point$ 表示当前点的坐标， $costpoint$ 表示之前花费的代价， $goalpoint$ 表示目标点坐标， $method$ 表示所用的方法。该函数用于计算当前点的邻近点的代价估计值。

find_Way(goalpoint,direction)函数：画出路径函数，其中 $goalpoint$ 表示目标点位置， $direction$ 表示各个父节点的来源方向。该函数用于跟随目标点位置到起始点位置，往回搜索父节点以还原出路径。

4.3 仿真结果分析

本文在 PC 机上，使用 MATLAB R2015b 平台进行仿真。搜索空间定义为一个 20×20 的二维栅格图，障碍物以 $rand$ 函数随机产生，以便检验算法对不同地形的可行性。起始点 S 设置在第 11 行第 18 列的位置，目标点 G 设置在第 12 行第 4 列的位置。如下图 4-4，图 4-5 和图 4-6 所示为不同障碍物模型下的航迹规划图，其中黑色部分表示障碍物

的位置,白色部分表示可飞行的区域,颜色表从蓝色到红色渐变表示了算法的搜索过程。

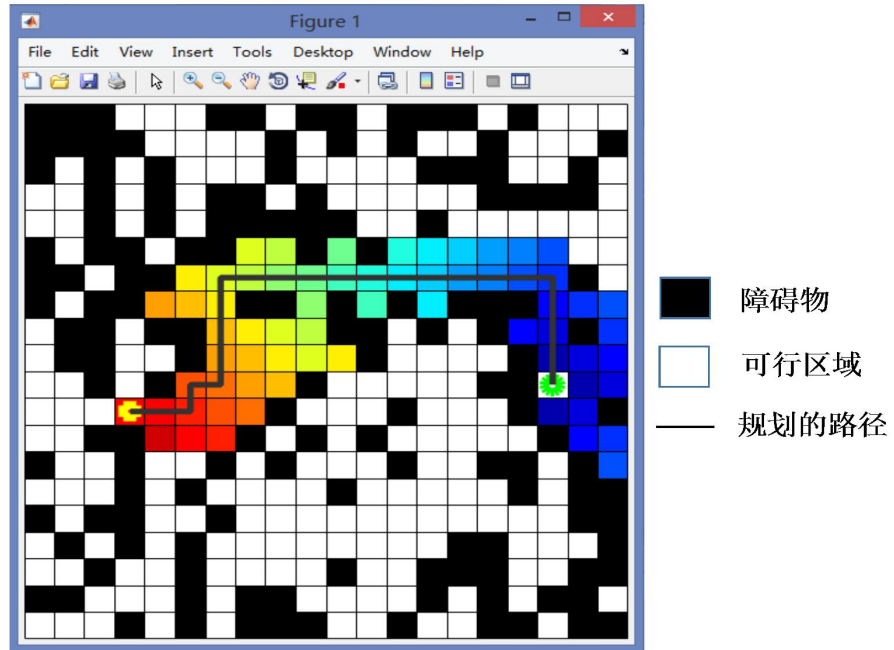


图 4-4 障碍物模型一下的航迹规划图

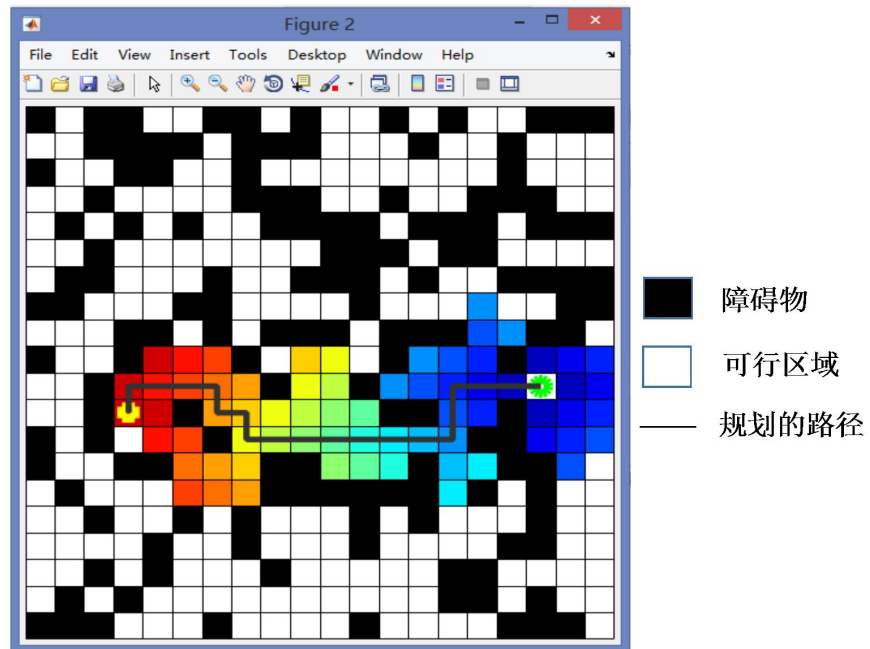


图 4-5 障碍物模型二下的航迹规划图

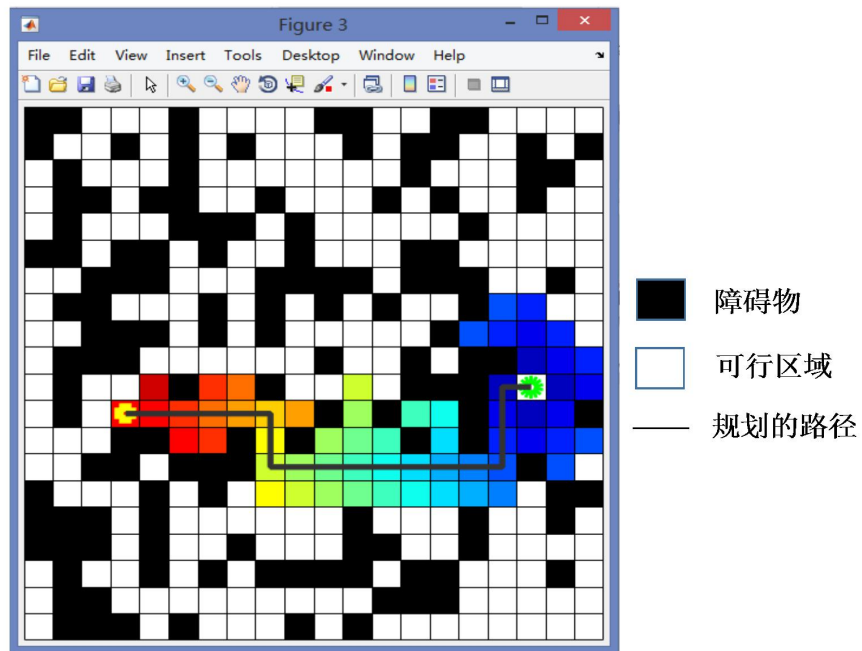


图 4-6 障碍物模型一下的航迹规划图

从仿真结果可以看出，A*算法是可行的。通过这种算法无人机可以很好的找到一条从起始点到目标点的路径，同时避免了与障碍物之间的碰撞。A*算法不仅是一种在静态障碍物中求解最短路径的直接搜索方法，也是处理一般搜索问题的常用有效算法。算法中所得到的距离估算值与实际值越接近，则航迹规划的速度就越快，得到的路径也越好。这种算法简单，易于工程实现，但算法的计算量相较于人工势场法来说是较大的。同时随着规划空间的增大算法的计算时间将逐渐增加，算法的信息存储量也将增大，这就不利于满足无人机需要快速反应以规避威胁的要求，在实时性方面没有人工势场法来得优异。

第 5 章 总结与展望

5.1 总结

随着科技的不断进步，信息化时代的到来，无人机将发挥越来越重要的作用，我们的生活将和无人机有着深深的联系。在这个信息时代，世界各国对无人机的研究、发展和应用都十分重视，加强国防建设和加快信息化建设中，高技术含量产物——无人机是重要的标志之一^[18]。近年来，对无人机的研究一直是个备受关注的研究热点。

本次论文研究是基于无人机飞控平台的算法研究，针对无人机航迹规划方面，利用人工势场法和 A*算法来实现对无人机飞行路径的规划，并在 MATLAB 上进行实验的仿真验证。通过比较算法在实现航迹规划上的性能不足，来综合改进现有的算法，以达到更好的控制效果。通过查找大量的相关资料并进行研究，论文主要做了以下工作：

(1) 分析了无人机研究的发展历史和意义，以及国内外的研究现状，同时分析了飞行控制的几个主要技术问题。

(2) 总结了无人机航迹规划的约束条件，包括最大航程、最小步长、最小转弯半径等无人机自身的约束和航迹高度、实时性要求等外部环境的约束。

(3) 对人工势场法进行研究，介绍了该算法的基本原理，并将其运用在航迹规划中，实现算法在 MATLAB 上的仿真并收集数据。根据人工势场法本身的限制和航迹规划的特点来对该算法做出改进，并在 MATLAB 上验证改进算法的可行性。针对算法中的参数如何选取的问题，本论文也做出了分析和总结。

(4) 对 A*算法进行研究，介绍了该算法的基本原理，并在 MATLAB 上实现了仿真验证，分析了 A*算法与人工势场法两种算法各自的优缺点。

5.2 展望

总的来说，无人机的发展还是具有较好的前景。研制或提出更多新颖算法，解决当前未完成的难题，对未来发展前景、趋势做出更多可靠、可行的预测，将成为未来世界无人机领域不可忽视的力量^[19]。研究无人机飞控技术是一项具有挑战性的技术难题，如何优化算法来提高无人机系统的性能是我们不断研究和实现的目标。

无人机航迹规划的课题研究所涉及的领域很广，由于时间上有限，本文只实现了算

法在 MATLAB 上的仿真，并针对仿真结果中遇到的问题做出改进。但实际无人机飞行环境更加复杂，还有很多需要解决的问题，有待我们做进一步的研究，这些内容包括：

（1）通过人工势场法和 A*算法规划的航迹路线中存在一些较大的转弯，这不利于无人机的飞行，在实际飞行情况中也是应尽量避免的，所以还需要对这些路径进行平滑处理。

（2）论文中将无人机所需避碰的障碍威胁简单的用圆形区域模型和栅格模型代替，这样处理有些过于粗糙，在实际的飞行环境中，肯定不会这么简单，还需要考虑更复杂的地形环境模型，对障碍的处理方法应进一步改进。

（3）本次论文将无人机航迹规划问题简化为在二维平面上的路径规划问题，没有考虑高度上的信息，所以后续可以考虑在三维地图模型上进行算法的研究。

参 考 文 献

- [1]龙威林. 无人机的发展与应用[J]. 产业与科技论坛,2014,(08):68-69.
- [2]金林敏,郑荣金,祁一民. 无人机在现代战争中的运用及发展[J]. 飞航导弹,2011,(09):29-32.
- [3]牛轶峰,沈林成,戴斌,徐昕,相晓嘉. 无人作战系统发展[J]. 国防科技,2009,(05):1-11.
- [4]陶于金,李沛峰. 无人机系统发展与关键技术综述[J]. 航空制造技术,2014,(20):34-39.
- [5]Paul G.Fahlstrom Introduction To UAV Systems Second Edition[M], Beijing: Publishing House of Electronics Industry, 2003, 1-34.
- [6]景岩. 无人机发展综述[J]. 才智,2013,(16):185.
- [7]路易斯路道夫加西亚卡里略, 四旋翼无人机控制-基于视觉的悬停与导航. 国防工业出版社,2016.
- [8]朱华勇,牛轶峰,沈林成,张国忠. 无人机系统自主控制技术研究现状与发展趋势[J]. 国防科技大学学报,2010,(03):115-120.
- [9]王正,朱兴动. 无人机全局渐近稳定自动编队飞行控制研究[J]. 系统仿真学报,2009,(07):2014-2017.
- [10]孙海波,周锐,邹丽,丁全心. 通信和测量受限条件下异构多 UAV 分布式协同目标跟踪方法[J]. 航空学报,2011,(02):299-310.
- [11]黄爱凤,邓克绪. 民用无人机发展现状及关键技术[C].//第九届长三角科技论坛——航空航天科技创新与长三角经济转型发展分论坛论文集.2012:24-30.
- [12]高凯,张翼麟,王一琳. 新兴国家典型无人机系统发展现状[J]. 飞航导弹,2014,(08):21-24.
- [13]Unmanned Systems Integrated Roadmap FY2011-2036[S]. Washington DC:Officer of The Secretary of Defense, 2011.
- [14]张爱民,梁书剑,马志强,付健. 军事通信抗干扰技术进展综述[J]. 通信技术,2011,(08):16-17+20.
- [15]胡中华. 基于智能优化算法的无人机航迹规划若干关键技术研究[D].南京航空航天大学,2011.
- [16]张帅,李学仁,张鹏,李博. 基于改进 A* 算法的无人机航迹规划[J]. 飞行力学,2016,(03):39-43.

- [17]陆锋. 最短路径算法:分类体系与研究进展[J]. 测绘学报,2001,(03):269-275.
- [18]李滨,杨笑天,王宏宇,单俊鑫. 森林防火中无人机的应用现状及发展趋势[J]. 科技创新导报,2015,(05):252-253.
- [19]高凯,张翼麟,王一琳. 新兴国家典型无人机系统发展现状[J]. 飞航导弹,2014,(08):21-24.

附录 1：人工势场法算法程序

```

clear;
Xo=[0 0];%起始点位置
Xsum=[10 10;1 1.6;3.1 2;4 4.5;3 6;6 2;5.4 6;7 3.6;8 8.3]; %目标点和障碍物的位置
k=9;%引力增益系数
m=10;%斥力增益系数
Po=2.5;%障碍物影响距离
n=8;%障碍物个数
l=0.2;%步长
K=0;%记录无人机到达目标点所经过的航迹点个数
J=200;%循环次数
Xj=Xo;%初始化，将无人机的起始点坐标赋给 Xj

for j=1:J%主循环开始
    Point(j,1)=Xj(1);
    Point(j,2)=Xj(2);%Point 用于保存无人机飞过的每个点的坐标。
%调用计算角度函数
    Angle=algorithm_angle(Xj,Xsum,n);%Angle 是计算出来的无人机和目标点以及无人机和障碍物之间与 x 轴的夹角
%调用计算引力函数
    angle_at=Angle(1);%无人机和目标点之间的角度，目标对无人机产生引力
    [Fatx,Faty]=algorithm_Attraction(Xj,Xsum,k,angle_at);%Fatx,Faty 是引力在 x,y 两个方向上的分量
%调用计算斥力函数
    for i=1:n
        angle_re(i)=Angle(i+1);%无人机和障碍物之间的角度，障碍物对无人机产生斥力
    end
    [Frex,Frey]=algorithm_Repulsion(Xj,Xsum,m,angle_re,n,Po);%Frex,Frey 是斥力在 x,y 两个方向上的分量
%计算合力和方向
    Forcexj=Fatx+Frex;%x 方向的合力
    Forceyj=Faty+Frey;%y 方向的合力

```

```

composition_angle(j)=atan(Forceyj/Forcexj);%合力与 x 轴方向的夹角向量
%计算无人机的下一步位置
if Forceyj < 0 && Forcexj <0
    Xnext(1)=Xj(1)-1*cos(composition_angle(j));
    Xnext(2)=Xj(2)-1*sin(composition_angle(j));
elseif Forceyj < 0 && Forcexj >0
    Xnext(1)=Xj(1)-1*cos(composition_angle(j));
    Xnext(2)=Xj(2)+1*sin(composition_angle(j));
elseif Forceyj > 0 && Forcexj <0
    Xnext(1)=Xj(1)+1*cos(composition_angle(j));
    Xnext(2)=Xj(2)-1*sin(composition_angle(j));
else
    Xnext(1)=Xj(1)+1*cos(composition_angle(j));
    Xnext(2)=Xj(2)+1*sin(composition_angle(j));
end
%保存无人机的每一个位置
Xj=Xnext;
if ((Xj(1)-Xsum(1,1))>0)&&((Xj(2)-Xsum(1,2))>0)
    K=j;%记录循环次数
    break;
end
end%大循环结束
K=j;
%画出起始点, 目标点, 障碍物分布以及各个航迹点位置
X=Point(:,1);
Y=Point(:,2);
x=[1 3.1 4 3 6 5.4 7 8];
y=[1.6 2 4.5 6 2 6 3.6 8.3];
figure;plot(x,y,'o',Xsum(1,1),Xsum(1,2),'v',0,0,'ms',X,Y,'.r');

```

主程序中调用的几个函数如下:

%计算角度函数

function All_angle=algorithm_angle(X,Xsum,n)%输入的参数分别为当前点坐标, 目标和障碍物坐标, 障碍物个数

```

for i=1:n+1
    Xx(i)=Xsum(i,1)-X(1);
    Yy(i)=Xsum(i,2)-X(2);

```

```

D(i)=sqrt(Xx(i)^2+Yy(i)^2);
if Xx(i)>0
    Angle=asin(Yy(i)/D(i));
else
    Angle=pi-asin(Yy(i)/D(i));
end
if i==1
    angle=Angle;
else
    angle=pi+Angle;
end
All_angle(i)=angle;%把无人机与目标点和障碍物的角度保存在 Angle 向量里面
end
end

```

%计算引力函数

```

function [Fatx,Faty]=algorithm_Attraction(X,Xsum,k,Attract_angle)
Rat=(X(1)-Xsum(1,1))^2+(X(2)-Xsum(1,2))^2;
rat=sqrt(Rat);%航迹点和目标点的距离
Fatx=k*rat*cos(Attract_angle);
Faty=k*rat*sin(Attract_angle);
end

```

%计算斥力函数

```

function [Frex,Frey]=algorithm_Repulsion(X,Xsum,m,angle_re,n,Po)
for i=1:n
    Rre(i)=(X(1)-Xsum(i+1,1))^2+(X(2)-Xsum(i+1,2))^2;
    rre(i)=sqrt(Rre(i));%航迹点和障碍物的距离
    if rre(i)>Po%如果障碍物和航迹点的距离大于障碍物影响半径 Po, 则斥力为 0
        Fx(i)=0;
        Fy(i)=0;
    else
        F(i)=m*(1/rre(i)-1/Po)*1/(rre(i)^2);%计算斥力大小
        Fx(i)=F(i)*cos(angle_re(i));%斥力在 x 方向上的分量
        Fy(i)=F(i)*sin(angle_re(i)); %斥力在 y 方向上的分量
    end
end
end

```

```

Frex=sum(Fx);%叠加斥力的分量
Frey=sum(Fy);
end

```

改进后的算法程序:

```

clear;
Xo=[0 0];%起始点位置
Xsum=[10 10;1 1.6;3.1 2;4 4.5;3 6;6 2;5.4 6;7 3.6;8 8.3]; %目标点和障碍物的位置
k=9;%引力增益系数
m=10;%斥力增益系数
Po=2.5;%障碍物影响距离
n=8;%障碍物个数
l=0.2;%步长
K=0;%记录无人机到达目标点所经过的航迹点个数
J=200;%循环次数
Xj=Xo;%初始化,将无人机的起始点坐标赋给 Xj
a=1.5;
for j=1:J%主循环开始
    Point(j,1)=Xj(1);
    Point(j,2)=Xj(2);%Point 用于保存无人机飞过的每个点的坐标
%调用计算角度函数
    Angle=algorithm_angle(Xj,Xsum,n);%Angle 是计算出来的无人机和目标点以及无人机和障碍物之间与 x 轴的夹角
%调用计算引力函数
    angle_at=Angle(1);%无人机和目标点之间的角度,目标对无人机产生引力。
    [Fatx,Faty]=algorithm_Attraction(Xj,Xsum,k,Attract_angle);%Fatx,Faty 是引力在 x,y 两个方向上的分量
%调用计算斥力函数
    for i=1:n
        angle_re(i)=Angle(i+1);%无人机和障碍物之间的角度,障碍物对无人机产生斥力
    end
    [Frex,Frey]=algorithm_Repulsion(Xj,Xsum,m,angle_re,n,Po); %Frex,Frey 是斥力在 x,y 两个方向上的分量
%计算合力和方向

```



```
Forcexj=Fatx+Frex;%x方向的合力
Forceyj=Faty+Frey;%y方向的合力
if (Forcexj==0)&&(Forceyj==0)
    Forcexj=a*Fatx+Frex;%x方向的合力
    Forceyj=a*Faty+Frey;%y方向的合力
end
composition_angle(j)=atan(Forceyj/Forcexj);%合力与x轴方向的夹角向量
```

附录 2: A*算法程序

```

clear;
n=20;%栅格地图的大小为 20*20
obstacle=0.45;%障碍物在整个地图中所占的比例
%初始化界面
[view,startpoint,goalpoint,cost,direction]=initialization(n,obstacle);
% 初始化 Open 表和 Close 表及其代价
Open=[startpoint]; Open_cost=[0]; Open_heuristic=[Inf];
Close=[]; Close_cost=[];
move_direction={'R','L','D','U'};
% 跟踪如果没有结果,则退出迭代次数
count=1;
% 创建视图使得能更清楚地看出地图模型界面
View=create_Figure(view,cost,startpoint,goalpoint);
while ~max(ismember(Open,goalpoint))&&~isempty(Open) %当 Open 表不为空也不是
目标点的时候
    [t,i] = min(Open_cost+Open_heuristic); %从 Open 表中选择代价最小的点 t,i 是
其下标
    %调用计算代价估计函数
    [costs,heuristics,points] =
find_Value(Open(i),Open_cost(i),view,goalpoint,'euclidean');
    Close=[Close;Open(i)]; %将 t 插入 Close 表中
    Close_cost=[Close_cost;Open_cost(i)]; %将 t 的代价计入 Close_cost
    % 更新 Open 表,分为三种情况
    if (i>1&&i<length(Open)) %t 在 Open 表的中间,删除 t
        Open=[Open(1:i-1);Open(i+1:end)];
        Open_cost=[Open_cost(1:i-1);Open_cost(i+1:end)];
        Open_heuristic=[Open_heuristic(1:i-1);Open_heuristic(i+1:end)];
    elseif (i==1) %t 是 Open 表的第一个元素,删除 t
        Open=Open(2:end);
        Open_cost=Open_cost(2:end);
        Open_heuristic=Open_heuristic(2:end);
    else %t 是 Open 表的最后一个元素,删除 t

```

```

Open=Open(1:end-1);
Open_cost=Open_cost(1:end-1);
Open_heuristic=Open_heuristic(1:end-1);
end
for j=1:length(points) %对于扩展的四个方向的坐标
    if ~isinf(costs(j)) %如果此点的实际代价不为 Inf,也就是没有遇到障碍物
        if ~max([Close;Open]==points(j)) %如果此点不在 Open 表和 Close 表中
            direction(points(j))=move_direction(j); %将此点的方向存在对应的
            direction 中
            cost(points(j))=costs(j); %将实际代价值存入对应的 cost 中
            Open=[Open;points(j)]; %将此点加入 Open 表中
            Open_cost=[Open_cost;costs(j)]; %更新 Open 表实际代价
            Open_heuristic=[Open_heuristic;heuristics(j)]; %更新 Open 表
            估计代价
        elseif max(Open==points(j)) %如果此点在 Open 表中
            I=find(Open==points(j)); %找到此点在 Open 表中的位置
            if Open_cost(I)>costs(j) %如果在 Open 表中此点的实际代价比现在所
            得的大
                cost(Open(I))=costs(j); %将当前的代价存入 cost 中
                Open_cost(I)=costs(j); %更新 Open 表中的此点代价
                Open_heuristic(I)=heuristics(j); %更新 Open 表中的此点估计代
                价
                direction(Open(I))=move_direction(j); %更新此点的方向
            end
        else %如果此点在 Close 表中,说明已经扩展过此点
            I=find(Close==points(j)); %找到此点在 Close 表中的位置
            if Close_cost(I)>costs(j) %如果在 Close 表中此点的实际代价比现在
            所得的大
                cost(Close(I))=costs(j); %将当前的代价存入 cost 中
                Close_cost(I)=costs(j); %更新 Close 表中的此点代价
                direction(Close(I))=move_direction(j); %更新此点的方向
            end
        end
    end
end
end
end

```

```

if isempty(Open)
    break;
end %当 OPEN 表为空, 表示所有点都已经查询过
set(View, 'CData', [cost cost(:,end); cost(end,:) cost(end,end)]);
set(gca, 'CLim', [0 1.1*max(cost(find(cost<Inf)))]);
drawnow;
end
% 画出最终路径
if max(ismember(Open,goalpoint)) %当找到目标点时
    disp('Solution found!');
    L=find_Way(goalpoint,direction);
    plot(L(:,2)+0.5,L(:,1)+0.5,'Color',0.2*ones(3,1),'LineWidth',4);
    drawnow;
elseif isempty(Open)
    disp('No Solution!');
end

```

主程序中调用的几个函数如下:

%初始化函数

```

function [view, startpoint, goalpoint, cost, direction] =
initialization(n,obstacle)

```

%初始化界面和障碍物

```

view=ones(n,n);
view(ceil(n^2.*rand(floor(n*n*obstacle),1)))=Inf;

```

% 设置起点和目标点位置

```

startpoint=sub2ind([n,n],10,18); %行列坐标转换为线性坐标
goalpoint=sub2ind([n,n],9,4);

```

% 将初始坐标设置为 0, 以免成为障碍物

```

view(startpoint)=0;
view(goalpoint)=0;

```

% cost 用来存储各个航迹点的实际代价

```

cost=NaN*ones(n,n);
cost(startpoint)=0; %设置起始点的实际代价为 0

```

% 生成 n*n 的矩阵, 用来存储各个点的父节点方向

```

direction=cell(n,n);

```

% 起始点设置为"S", 目标点设置为"G"

```

direction{startpoint}='S';

```

```

    direction{goalpoint}='G';
% 障碍物的来源方向设置为 0
    direction(view==Inf)={0};
end

%创建视图函数
function View=create_Figure(view,cost,startpoint,goalpoint)
    if isempty(gcf) %gcbf 是当前返回图像的句柄，如果为空
        window=figure('Position',[450 150 500 500]); %创建窗口位置，让窗口起
        点左边距离左下角 450，下边距离左下角 150，宽度 500，高度 500
        X_Y=axes('Position',[0.01 0.01 0.98 0.98]); %创建坐标系
    else
        % 获得当前图像并清空
        gcf;
        cla;
    end
    n=length(view);
    % 设置障碍物是黑色的区域，其他的都是白色的，0 是黑色
    view(view < Inf)=0;
%绘制伪彩色图
    pcolor([1:n+1],[1:n+1],[view view(:,end); view(end,:) view(end,end)]);
    map=flipud(colormap('jet')); %flipud 用于反转矩阵 colormap 是产生 jet 类型
    的颜色表，范围从蓝色到红色
%将颜色表中的第一种颜色变为白色，将最后一种颜色变为黑色
    map(1,:)=zeros(3,1);
    map(end,:)=ones(3,1);
    % 应用色彩图
    colormap(flipud(map));
    hold on;
    View=pcolor(1:n+1,1:n+1,[cost cost(:,end); cost(end,:)
cost(end,end)]);
%ind2sub 用来将线性坐标转换为行列坐标，返回行和列的位置
    [starty,startx]=ind2sub([n,n],startpoint);
    [goaly,goalx]=ind2sub([n,n],goalpoint);
    % 画出起始点和目标点

plot(startx+0.5,starty+0.5,'go','MarkerSize',10,'LineWidth',6); %'go'中 g

```

表示 green, o 表示 Circle (圆形)

```
plot(goalx+0.5,goaly+0.5,'ys','MarkerSize',10,'LineWidth',6); %加 0.5
是为了把坐标移到方块中央, 'ys'中 y 表示 yellow, s 表示 Square (方形)
end
```

%计算代价估计函数

```
function [costs,heuristics,points] =
find_Value(point,costpoint,view,goalpoint,method)
n=length(view);
```

% 将线性索引转换为行列值

```
[current(1) current(2)]=ind2sub([n n],point); %获得当前点的坐标,
current(1)是列坐标, current(2)是行坐标
```

```
[goal(1) goal(2)]=ind2sub([n n],goalpoint); %获得目标点的坐标
```

% 存放代价值和位置的地方

```
costs=Inf*ones(4,1);
heuristics=Inf*ones(4,1);
p=ones(4,2);
```

% 如果可以向左查询, 那么就是从右边移动

```
newx=current(2)-1;
newy=current(1);
if newx>0 %如果没有到边界
p(1,:)= [newy newx]; %获得新的坐标
switch lower(method)
case 'euclidean' %欧几里得距离
```

heuristics(1)=abs(goal(2)-newx)+abs(goal(1)-newy); %heuristics(1)为估计函数计算的距离代价

```
end
```

```
costs(1)=costpoint+view(newy,newx); %costs(1)为经过此方向点的实际代价
```

```
end
```

% 如果可以向右查询, 那么就是从左边移动

```
newx=current(2)+1;
newy=current(1);
if newx<=n
p(2,:)= [newy newx];
switch lower(method)
```

```

    case 'euclidean'
        heuristics(2)=abs(goal(2)-newx)+abs(goal(1)-newy);
    end
    costs(2)=costpoint+view(newy,newx);
end
% 如果可以向上查询, 那么就是从下边移动
newx=current(2);
newy=current(1)-1;
if newy>0
    p(3,:)=[newy newx];
    switch lower(method)
    case 'euclidean'
        heuristics(3)=abs(goal(2)-newx)+abs(goal(1)-newy);
    end
    costs(3)=costpoint+view(newy,newx);
end
% 如果可以向下查询, 那么就是从上边移动
newx=current(2);
newy=current(1)+1;
if newy<=n
    p(4,:)=[newy newx];
    switch lower(method)
    case 'euclidean'
        heuristics(4)=abs(goal(2)-newx)+abs(goal(1)-newy);
    end
    costs(4)=costpoint+view(newy,newx);
end
% 将行列返回到线性索引
points=sub2ind([n n],p(:,1),p(:,2)); %points 为扩展的四个方向上的坐标
end

%画出路径函数
function L=find_Way(goalpoint,direction) %从目标点位置到起始点位置
    n=length(direction);
    point=goalpoint;
% 将线性索引转换为行列
    [Ly,Lx]=ind2sub([n,n],point);
% 存储初始位置

```

```
L=[Ly Lx];  
% 直到处于起始位置为止  
while ~strcmp(direction{point}, 'S') %当查询到的点不是's'时  
    switch direction{point}  
        case 'L' %如果获得该点的来源点方向为左时  
            Lx = Lx - 1;  
        case 'R'  
            Lx = Lx + 1;  
        case 'U'  
            Ly = Ly - 1;  
        case 'D'  
            Ly = Ly + 1;  
    end  
    L=[L; Ly Lx];  
% 将行列转换为线性索引  
    point=sub2ind([n n], Ly, Lx);  
end  
end
```


致 谢

大学四年的本科学习生活即将结束，在这期间我从一个懵懂无知的高中生成长为一个能勇敢走入社会，直面各种困难的应届毕业生，这都多亏了有老师和同学对我的帮助，让我能在遇到困难和挫折的时候，能一一克服这些障碍。在这里正好借这个机会对所有帮助过我的老师和同学表示深深的感谢。

首先，我要特别感谢我的母校浙江工业大学，是它提供了良好的平台让我们能有更多的资源进行学习，去追求我们的梦想。

其次，我要感谢我的导师赵云波教授。赵老师治学态度严谨，工作认真负责，待人和蔼可亲。在整个毕业设计的过程中，赵老师给予了我很多建设性的意见，悉心地指导我完成了论文的研究方向选择，研究方法的确定以及各项研究工作。借此机会，我要再一次向赵老师表达我的感谢之意，没有赵老师的指导，我无法顺利完成毕业设计工作及毕业论文的撰写。

再次，我要感谢姚俊毅学长对我的帮助。在做毕业设计的过程中，学长给予了我很多宝贵的建议，让我在这条研究的道路上少走了很多弯路。当我有什么不懂的地方去请教学长的时候，他总能很耐心的向我讲解。在此，向姚学长表示深深的谢意。

最后，我要感谢我的父母，是你们给了我学习的机会，是你们在本就不怎么宽裕的情况下给了我无忧无虑的生活，让我能够专心完成自己的学业，也是你们给了我精神上的支持，让我能够有足够的勇气来面对生活中的困难。你们是我坚强的后盾，你们的鼓励是我前进的动力。

大学的这四年，我有过欢声和笑语，也有过迷茫和沮丧，但重要的是我学会了如何调整自己的心态，积极自信地去面对一切。未来将有更多的困难在等着我，希望自己能够始终保持这种心态，去迎接人生中的挑战！