

无人平台轨迹规划算法研究

徐晓贞¹ 何 慧² 姜发圣¹ 肖文文¹

1.贵州航天天马机电科技有限公司 贵州 遵义 563000

2.江西理工大学 矿业工程学院 江西 赣州 341000

【摘要】：为解决 A*算法在无人平台轨迹规划中生成路径呈“锯齿状”、拐点转向困难等问题，本文研究基于 A*算法的改进及轨迹平滑处理方法。先阐述 A*算法的代价评估体系与启发式函数，再对比 Floyd 几何平滑与 B 样条曲线拟合的原理及效果。通过 30x30 网格地图仿真验证，从点数、长度、平滑度分析三种路径性能。结果表明 Floyd 算法能减少 53.5%路径点、缩短 2.56%路径长度，更适配无人平台战术机动需求，为无人平台高效轨迹规划提供技术支撑。

【关键词】：无人平台；轨迹规划；A*算法；Floyd 算法；B 样条曲线

DOI:10.12417/2705-0998.25.24.013

引言

无人平台在侦察、作战等场景的应用日益广泛，轨迹规划的合理性直接决定其机动效率与生存能力。A*算法因兼顾搜索完整性与方向性，成为离散栅格地图路径规划的经典方案，但离散节点生成的折线路径难以满足无人平台运动学约束。本文以解决路径平滑性与可行性矛盾为目标，引入 Floyd 算法与 B 样条曲线两种平滑方案，通过理论分析与仿真实验对比其性能，筛选适配无人平台高机动性、强隐蔽性需求的方案，为无人平台轨迹规划优化提供理论与实践依据。

1 基于 A*算法的改进与轨迹平滑处理

A* (A-star) 算法是一种经典的图搜索算法，广泛用于路径规划问题。它结合了广度优先搜索和贪心搜索的优点，通过启发式函数实现高效的路径查找。

A*算法需要三种主要的代价值来评估路径的优劣：

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

其中： $f(n)$ ：综合代价值，用于评价节点的优先级； $g(n)$ ：从起点到当前节点 n 的实际代价（已走过的路径长度）。 $g(n)$ 保证了路径的完整性； $h(n)$ ：当前节点 到目标节点的启发式估计代价（未走的路径长度预测值）。 $h(n)$ 提供方向性，引导搜索朝向目标节点。对于启发式函数 $h(n)$ 必须是可接受的，即始终不超过真实代价。常用的启发式函数：

曼哈顿距离（适用于格子地图，无对角移动）：

$$|x_1 - x_2| + |y_1 - y_2| \quad (2)$$

欧几里得距离（适用于连续空间）：

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (3)$$

对角距离（适用于允许对角线移动）：

$$\max(|x_1 - x_2|, |y_1 - y_2|) \quad (4)$$

A*算法作为一种经典的图搜索算法，其在离散的栅格地图中能够找到从起点到终点的最优或次优路径^[1]。然而，由于其搜索过程基于离散的网格节点，导致生成的路径存在一些缺点，路径由一系列网格中心点连接而成，呈现“锯齿状”或“阶梯状”的折线，无人平台（特别是阿克曼转向的车辆）无法执行直角转向或锐角转向，原始路径会导致平台在拐点处停车转向，严重影响机动效率。从连续运动的角度看，折线路径并非真正最短路径，平滑的曲线通常更短且更高效。而且路径拐点常常紧贴障碍物边缘，增加了碰撞风险。本文轨迹规划研究核心目标就是将 A*输出的离散折线路径，转化为一条连续、光滑、曲率连续且符合车辆运动学约束的可行轨迹。

2 平滑处理的核心原理

路径平滑的本质是一个优化过程，在满足安全性（无碰撞）和可行性（符合动力学）的前提下，对原始路径进行“再加工”。其核心原理可分为几何学原理和优化原理两大类。其中几何学原理方案是 Floyd 路径平滑算法，它基于一个简单的几何学思想：两点之间，直线最短。通过循环检测，尝试用一条直线段连接路径中不相邻的两个点，如果这条直线段不会与障碍物发生碰撞，则可以安全地删除这两点之间的所有中间点^[2]。另外一种优化原理是 B 样条曲线平滑法，B 样条曲线平滑法基于数值分析和曲线拟合理论，其目标是生成一条高阶连续的光滑曲线，它的基本原理是将 A*算法输出的路径点视为控制点，利用 B 样条基函数拟合出一条光滑曲线。该曲线不一定会完全通过所有控制点，但会被限制在控制点构成的凸包内，具有良好的局部控制和光滑特性。本文轨迹规划研究思想就是将 Floyd 算法和 B 样条曲线算法进行对比分析，选取综合性能好的算法，对 A*算法输出的路径进行光滑拟合，最终得到一条既短又平滑、且符合车辆运动学的高质量轨迹。

3 Floyd 算法

Floyd 算法的核心思想是动态规划。基本思路是对于图中的任意两个顶点 i 和 j ，考虑是否存在一个顶点 k ，使得从 i 到 j 的路径如果经过 k ，会比已知的路径更短^[3]。如果是，就更新 i 到 j 的最短距离。算法通过一个三重循环，逐步尝试将每一个顶点 k 作为中间节点，来更新所有顶点对 (i,j) 之间的最短距离。随着中间节点集合的不断扩大，最终得到所有点对之间的最短路径。用一个二维数组 $dist$ 来存储最短距离， $dist[i][j]$ 表示从顶点 i 到顶点 j 的当前已知最短距离。算法通过以下核心状态转移方程来更新 $dist$ 数组：

状态转移方程：

$$dist[i][j] = \min(dist[i][j], dist[i][k] + dist[k][j]) \quad (5)$$

其中： k 是当前正在被考虑作为中间节点的顶点； i 是源顶点； j 是目标顶点。

3.1 算法步骤

(1) 创建一个大小为 $n \times n$ (n 为顶点数) 的距离矩阵 $dist$ ；对于每个 i 和 j ：如果 $i=j$ ，则 $dist[i][j]=0$ ，即自己到自己的距离为 0；如果存在从 i 到 j 的边，则 $dist[i][j]$ 为该边的权重。否则， $dist[i][j]=\infty$ 。

(2) 对中间节点 k 从 0 到 $n-1$ 进行循环；对源节点 i 从 0 到 $n-1$ 进行循环；对目标节点 j 从 0 到 $n-1$ 进行循环；如果 $dist[i][k]$ 和 $dist[k][j]$ 都不是无穷大，则应用状态转移方程^[4]。

(3) 如果 $dist[i][k]+dist[k][j]<dist[i][j]$ ，则

$$dist[i][j]=dist[i][k]+dist[k][j]；$$

循环结束后， $dist$ 矩阵中存储的就是所有顶点对之间的最短距离。

3.2 B 样条曲线的原理

一条 P 次的 B 样条曲线由三要素定义：

(1) 控制点 (p_0, p_1, \dots, p_n) ，这些点“拉拽”着曲线的形状。在平滑中，这些通常是通过计算得到的，用于逼近原始路径点。

(2) 节点向量 $U=(u_0, u_1, \dots, u_m)$ 这是一个非递减的实数序列，决定了参数 u 如何映射到曲线上，它是 B 样条的核心。

(3) 次数 P ：决定了曲线的光滑度（连续性为 C^{P-1} ）。

B 样条曲线的参数公式为：

$$C(u) = \sum_{i=0}^n N_{i,p}(u) P_i \quad (6)$$

其中： $C(u)$ 是曲线上对应于参数 u 的点； P_i 是第 i 个控制

点（在二维平滑中， $P_i=(x_i, y_i)$ ）； $N_{i,p}(u)$ 是定义在节点向量 U 上的 P 次 B 样条基函数；参数 u 在范围 (u_p, u_{m-p}) 内变化，曲线定义在这个区间上。对于均匀节点向量，通常 $u \in [0, 1]$ 。

B 样条基函数 $N_{i,p}(u)$ 的递推定义（Cox-deBoor 公式），这是 B 样条的数学核心，是一个递归过程。这个公式揭示了 B 样条的递归和局部支撑特性。

第 0 次 ($P=0$) 基函数：

$$N_{i,0}(u) = \begin{cases} 1 & \text{如果 } u_i \leq u < u_{i+1} \\ 0 & \text{其他} \end{cases} \quad (7)$$

更高次 ($P>0$) 基函数：

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \quad (8)$$

3.3 B 样条路径平滑算法

通常不直接使用原始路径点作为 B 样条的控制点，因为那样曲线不会通过它们。相反，通常执行一个“拟合”过程。给定一组数据点 (Q_0, Q_1, \dots, Q_m) ，找到一条 P 次 B 样条曲线 $C(u)$ ，使其尽可能地逼近这些数据点。

(1) 参数化

为每个数据点 Q_k 分配一个参数值 \hat{u}_k ，常用累加弦长参数化法，其中 L 是总路径长度。这样 $\hat{u}_m = 1$ ，这考虑了数据点的分布密度。

$$\begin{cases} \hat{u}_0 = 0 \\ \hat{u}_k = \hat{u}_{k-1} + \frac{\|Q_k - Q_{k-1}\|}{L}, \quad k = 1, 2, \dots, m \end{cases} \quad (9)$$

(2) 确定节点向量 U

希望曲线有 $n+1$ 个控制点 ($n < m$)，以实现平滑和降噪，节点向量的数量是 $m+1=n+p+2$ 。使用平均法来确定内部节点。内部节点数量为 $n-p$ 。

计算内部节点索引：

$$d = \frac{m+1}{n-p+1} \quad (10)$$

对于 $j = 1, 2, \dots, n-p-1$ ，令 $i = |jd|$ ， $\alpha = jd - i_s$ 。则内部节点：

$$u_{p+j} = (1-\alpha) \cdot \hat{u}_{i-1} + \alpha \cdot \hat{u}_i \quad (11)$$

(3) 构建线性方程组并求解控制点 P

目标是 minimized 误差：

$$\min \sum_{k=0}^m \|Q_k - C(\hat{u}_k)\|^2 \quad (12)$$

将 $C(\hat{u}_k) = \sum_{i=0}^n N_{i,p}(\hat{u}_k)P_i$ 代入，得到一个线性最小二乘问题：

$$\begin{bmatrix} N_{0,p}(\hat{u}_0) & N_{1,p}(\hat{u}_0) & \cdots & N_{n,p}(\hat{u}_0) \\ N_{0,p}(\hat{u}_1) & N_{1,p}(\hat{u}_1) & \cdots & N_{n,p}(\hat{u}_1) \\ \vdots & \vdots & \ddots & \vdots \\ N_{0,p}(\hat{u}_m) & N_{1,p}(\hat{u}_m) & \cdots & N_{n,p}(\hat{u}_m) \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ \vdots \\ P_n \end{bmatrix} \approx \begin{bmatrix} Q_0 \\ Q_1 \\ \vdots \\ Q_m \end{bmatrix} \quad (13)$$

记系数矩阵为 A ，大小为 $(m+1) \times (n+1)$ ，控制点矩阵为 P ，数据点矩阵为 Q ，则方程为：

$$AP \approx Q \quad (14)$$

这个方程是超定的（因为数据点 $m+1 >$ 控制点 $n+1$ ），求解其正规方程来得到最小二乘解：

$$((A^T A)P = A^T Q) \quad (15)$$

解这个线性方程组，即可得到最优的控制点 P 。

(4) 生成平滑曲线

得到控制点 P 和节点向量 U 后，就完全定义了 B 样条曲线 $C(u)$ 。要得到平滑路径，只需在参数域 $[u_p, u_{m-p}]$ 内等间距地取一系列 u 值，代入公式 $C(u) = \sum N_{i,p}(u)P_i$ 计算即可。

总之， B 样条平滑是一个拟合（逼近）过程，而非插值过程。

曲线定义：

$$C(u) = \sum N_{i,p}(u)P_i \quad (16)$$

基函数递归：

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \quad (17)$$

通过解正规方程 $((A^T A)P = A^T Q)$ ，来找到最优的控制点 P ，

其中矩阵 A 由基函数在数据点参数 \hat{u}_k 处的值构成，这种方法能有效滤除噪声，生成一条高度光滑、且忠实于原始路径整体形状的新路径。

4 轨迹规划仿真验证

本仿真程序旨在实现从起点到终点的路径规划，使用 A^* 算法进行全局路径搜索，并分别使用 Floyd 算法和 B 样条曲线对原始路径进行平滑处理，最后通过可视化对比展示三种路径（原始 A^* 路径、Floyd 平滑路径、 B 样条平滑路径）的效果^[5]。

程序设计的初始环境和参数设置为：地图尺寸为 30×30 的网格地图，起点坐标 $(2,2)$ ，终点坐标 $(28,28)$ ，障碍物设置包括

圆形、矩形和对角线形状的障碍物，白色为可通行区域，黑色为障碍物区域。

程序运行结果：原始 A^* 算法的路径如图 1 所示，Floyd 平滑后的路径如图 2 所示， B 样条平滑后的路径如图 3 所示，三种路径对比如所示，性能分析报告如表 1 所示。

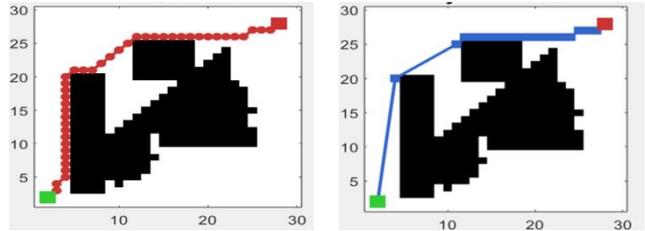


图 1 原始 A^* 算法路径 图 2 Floyd 平滑后的路径

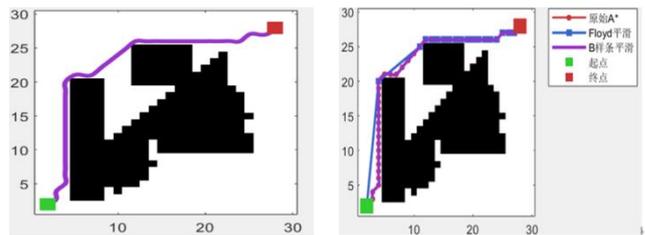


图 3 B 样条平滑后的路径 图 4 三种路径对比

表 1 性能分析报告

路径类型	点数	长度	平滑度
原始 A^* 路径	43	46.14	低
Floyd 平滑	20	44.96	中
B 样条平滑	129	46.51	高

综上，可以从点数、长度和平滑度三个方面进行比较分析。从路径点数来看，原始 A^* 路径有 43 个点，路径点较多，导致路径转折较多。Floyd 平滑后路径点减少到 20 个，减少了约 53.5%，说明 Floyd 平滑有效地减少了不必要的中间点，使路径更加简洁。 B 样条平滑后路径点增加到 129 个，这是因为 B 样条平滑通过插值增加了路径点的密度，使得路径更加连续光滑。从路径长度来看，原始 A^* 路径长度为 46.14，Floyd 平滑后路径长度略微减少到 44.96，减少了约 2.56%，这是因为 Floyd 平滑通过拉直路径减少了总长度。 B 样条平滑后路径长度增加到 46.51，增加了约 0.80%，这是因为 B 样条平滑为了保持光滑性，增加了路径长度。从平滑度来看，原始 A^* 路径平滑度低，因为它是基于网格搜索，路径由直线段组成，转折处尖锐。Floyd 平滑后路径平滑度为中，它通过删除共线点减少了转折，但路径仍然由直线段组成，在转折处可能不够光滑。 B 样条平滑后路径平滑度高，它利用样条曲线拟合，生成连续曲率的路径。

5 结语

针对无人战车平台高机动性、强隐蔽性的作战需求，轨迹规划算法需优先保障实时响应能力与战术敏捷性。Floyd 平滑法通过几何简化，能高效生成转折点最少、路径长度更优的航

线。此种路径不仅极大降低了平台的规划计算与姿态调整频次，确保了在动态战场中的快速反应，其简洁的直线运动特征也更利于利用地形进行隐蔽机动，有效缩短在开阔地带的暴露时间，从而显著提升战场生存能力。因此，从满足核心战术指标的角度，Floyd 平滑法是更为适宜的选择。

参考文献：

- [1] 王建园,符洋,陈金宝,等.基于改进 A-star 的无人机路径规划算法[J].空天技术,2025,(02):78-87.
- [2] 闫晓东,常天庆,郭理彬.基于改进遗传算法的无人战车跃进位置决策[J].兵工自动化,2023,42(03):77-83.
- [3] 谭光兴,黄磊昌,李明泽.基于改进混合 A*算法在动态环境中的快速路径规划[J].现代电子技术,2025,48(19):136-142.
- [4] 刘永涛,晁兴雨,朱屹晨,等.融合改进 A*和 DWA 算法的无人车路径规划研究[J].汽车技术,2025,(12):19-28.
- [5] 刘健,沈芸亦,邱锦,等.复杂地形约束下的多目标路径规划 A*算法研究[J].计算机应用与软件,2025,42(08):297-305+381.