信息与控制　xxxx 年　第 x 卷　第 x 期：1 ~ 13

DOI：10. 13976/j. cnki. xk. 2025. 3053　　CSTR: 32166. 14. xk. 2025. 3053

动态环境移动机器人 NMPC-IRRT*分层路径规划分层路径规划

陈　阳，黄宜庆

安徽工程大学电气工程学院，安徽 芜湖　241000

基金项目：国家自然科学基金区域创新发展联合基金(U22A2079)；安徽高校协同创新项目(GXXT-2020-069)

通信作者：黄宜庆，yiqhuang@ahpu.edu.cn　收稿/录用/修回：2025 − 10 − 13/2025 − 12 − 30/2025 − 03 − 13

摘要

针对移动机器人在动态复杂环境下的路径规划问题，提出了一种基于非线性模型预测的控制与启发式 RRT*的分层路径规划算法 (NMPC-IRRT*)。该算法分为上下两层，上层根据障碍物分布，利用启发式 RRT*算法，通过椭圆采样启发式策略生成高质量全局初始轨迹；下层利用非线性模型预测控制 (NMPC) 局部导航控制器，结合线性运动模型预测动态障碍物未来轨迹，实现对路径的实时跟踪与动态避障。仿真对比显示，在复杂动态环境下，该算法的平均运行时间为 51.47 s，较 DWA-IRRT*算法缩短了约 71.5%，平均路径长度缩短了 32.9%；当动态障碍物数量增加至 30 个的高密度场景时，规划成功率仍能保持在 90% 以上，且其轨迹曲率方差和角速度变化率均为对比算法中最低。该算法通过全局拓扑引导与局部预测优化的协同，解决了传统采样算法路径质量差及动态避障实时性不足的问题，实现了机器人在复杂未知动态场景下的自主导航。

关键词

启发式 RRT*

非线性模型预测控制

路径规划

动态避障

分层控制

中图法分类号：TP273

文献标志码：A

Hierarchical Path Planning for Mobile Robots in Dynamic Environments Using NMPC-IRRT*

CHEN Yang, HUANG Yiqing

College of Electrical Engineering, *Anhui Polytechnic University*, *Wuhu* 241000, *China*

Abstract

To address the path planning problem of mobile robots in dynamic complex environments, a hierarchical path planning algorithm based on nonlinear model predictive control and informed RRT* (NMPC-IRRT*) is proposed. The algorithm is divided into upper and lower layers: the upper layer utilizes the informed RRT* algorithm based on obstacle distribution to generate a high-quality global initial trajectory through an elliptical sampling heuristic strategy. The lower layer employs an NMPC local navigation controller, combined with a linear motion model to predict the future trajectories of dynamic obstacles, achieving real-time path tracking and dynamic obstacle avoidance. Simulation comparisons show that in complex dynamic environments, the average running time of the algorithm is 51.47 s, which is approximately 71.5% shorter than that of the DWA-IRRT* algorithm, and the average path length is reduced by 32.9%. In high-density scenarios where the number of dynamic obstacles increases to 30, the planning success rate

Keywords

informed RRT*;

path planning;

nonlinear model predictive control;

dynamic obstacle avoidance;

hierarchical control

remains above 90%, and its trajectory curvature variance and angular velocity change rate are the lowest among the compared algorithms. Through the synergy of global topological guidance and local predictive optimization, the algorithm addresses the problems of poor path quality and insufficient real-time performance in dynamic obstacle avoidance of traditional sampling algorithms, achieving autonomous navigation for robots in complex unknown dynamic scenes.

0 引言

随着自动化技术的快速发展，移动机器人已在物流、服务、救援、探索等领域得到广泛应用，其自主导航系统已成为提高各行业管理效率的关键[1-3]。作为自主导航的核心环节，路径规划与跟踪备受关注，其核心目的是使移动机器人能够准确快速地到达预定的目标位置，同时在整个运动过程中避开静态障碍物、动态障碍物以及危险区域在内的所有影响因素[4-5]。然而，在充斥着行人、车辆等未知动态障碍物的真实世界中，实现鲁棒、智能的导航仍然面临巨大挑战。

现阶段的路径规划方法可以分成两类，全局路径规划与局部路径规划。全局路径规划算法包含Dijkstra 算法[6-7]、A*算法[8-9] 等。近年来的研究趋势更倾向于将多种算法深度融合，以取长补短。A*算法严重依赖于启发函数，且在动态环境下性能不佳。成怡等[10] 将改进的 A*算法与 Morphin 搜索树算法相结合，提出了一种动态路径规划方法。在该方法中减少了路径规划时所需的关键节点数，之后再利用 Morphin 搜索树算法进行全局路径规划。但是 Morphin 算法的短视性使其容易陷入局部最优，固定的弧线集限制了其灵活性。全局路径规划方法中还有一类基于采样的全局路径规划方法，主要包括概率路线图算法（PRM）[11]、快速随机树算法（RRT）[12-13] 及其衍生算法[14-16] 等，这类算法在处理高维空间时能够快速地获取路径结果并表现出良好的效率和完备性。但是这些全局算法通常设定环境是静态的，难以在动态环境下直接应对突发的动态障碍物变化和环境变化。CUI 等[17] 优化采样策略和节点创建方法，展示了更优的性能。随着机器人集群、无人机编队等应用的兴起，多智能体协同路径规划受到广泛关注。这类研究不仅要为单个智能体规划路径，还需要解决它们之间的协同与冲突避免问题，计算复杂度更高，约束条件更多。这些融合方法在预测障碍物未来行为方面仍有局限，可能导致规划出的路径在面对多个高速动态障碍时显得过于保守或反应迟缓。

局部路径规划算法包括人工势场法（APF）[18-19]、动态窗口法（DWA）[20] 和向量场直方图法（VFH）[21] 等。与此同时，基于深度强化学习（DRL）[22-23] 的路径规划方法也取得了显著进展。这类方法让智能体在大量的模拟环境中进行试错学习，能够掌握在复杂动态场景下的隐式导航策略，展现出强大的环境适应能力。LIU 等[24] 采用逆向强化学习方法，创新性地将 Dijkstra 算法与逆向强化学习框架结合，成功绕开了对精确地图的依赖，为在数据稀疏区域进行路径规划提供了新的思路。但模型的训练过程涉及神经网络的前向和后向传播，并且在每个迭代周期内都需要为大量的历史数据重复运行 Dijkstra 算法来生成当前策略和计算梯度。DRL 方法通常需要大量的训练数据和计算资源，尤其是在需要严格满足动力学约束和安全距离的情况下，其安全性和可靠性仍面临挑战。最新的研究集中在如何提高模型预测控制（MPC）的计算效率，以及如何将 MPC 与全局路径进行更平滑的集成。KIM 等[25] 提出了一种创新的思路，利用 MPC 的前瞻性优化纯追踪算法。这种方法将 MPC 作为经典控制器的一个智能参数调节器，在保证行驶平顺性的同时，通过约束确保跟踪误差始终在安全边界内。但是需要建立运动学模型和纯追踪控制器的集成模型，这个模型比单纯的车辆动力学模型更复杂且可能是非线性的，增加了建模和后续线性化的难度。MPC 的性能好坏直接取决于其内部预测模型的准确性，若集成模型对轮胎侧偏特性等参数的估计不准确，MPC 的预测结果就会产生偏差。错误的预测结果会导致 MPC 计算出一个并非最优的距离，若将次优的参数再输入给追踪控制器，可能导致实际的跟踪效果不及一个基于规则的自适应前视距离算法。

尽管现有研究已在特定方面取得了进展，但在复杂动态环境下实现全局路径的平滑高效跟踪，同时对多类型障碍物进行实时避障，仍然是一个挑战。为了解决 RRT 算法难以应对复杂动态环境、搜索效率低的问题，本文提出了一种 NMPC-IRRT*（非线性模型预测控制启发式 RRT*）分层路径规划算法。在全局路径规划层面，利用启发式 RRT*算

法在考虑静态障碍物和动态障碍物初始位置的情况下，生成一条从起点到目标点的初始优化路径。在局部路径规划层面，算法通过预测移动机器人的未来状态，并依据机器人当前状态实时调整其线速度与角速度，从而生成符合动力学约束又能有效避障的有效路径。

1　机器人与环境建模

1.1　机器人运动学模型

为了对移动机器人的运动进行规划和控制，需要建立能够描述移动机器人位置、速度以及姿态之间变化关系的运动学模型[26]。

采用常见的差速驱动机器人模型，通过独立控制左右轮的速度来实现前进、后退和转向。假设移动机器人是一个刚体，在 2 维平面上运动，其在任意t时刻的状态可用一个 3 维向量$\boldsymbol{q}(t)$来表示：

$$\boldsymbol{q}(t)=\begin{bmatrix} x(t) \\ y(t) \\ \theta(t) \end{bmatrix} \tag{1}$$

其中，$x(t)$和$y(t)$分别表示在全局坐标系中的X轴和Y轴坐标；$\theta(t)$表示机器人的朝向角，即机器人前进方向与全局坐标系的X轴之间的夹角。

机器人的控制输入通常是其角速度ω和线速度v，本文采用离散时间运动学模型。假设控制输入v_k和ω_k在Δt采样时间间隔内保持恒定，并且当前时刻k的状态为$\boldsymbol{q}_k=\left[x_k,y_k,\theta_k\right]^{\mathrm{T}}$。则下一时刻$k+1$的状态$\boldsymbol{q}_{k+1}=\left[x_{k+1},y_{k+1},\theta_{k+1}\right]^{\mathrm{T}}$，移动机器人的离散时间运动学模型可表示为

$$\begin{aligned} x_{k+1}&=x_k+v_k\cos(\theta_k)\Delta t \\ y_{k+1}&=y_k+v_k\sin(\theta_k)\Delta t \\ \theta_{k+1}&=\theta_k+\omega_k\Delta t \end{aligned} \tag{2}$$

其中，k是离散时间标签，Δt是时间步长，θ_{k+1}通常需要使用$WrapToPi$函数将其归一化到$[-\pi,\pi]$区间。

1.2　动态障碍物模型

为模拟真实动态环境，本文对动态障碍物O_i设定了基于线性恒定速度的运动模型，并引入碰撞反弹机制。在未发生碰撞时，第i个动态障碍物在k时刻的位置$\boldsymbol{p}_{\mathrm{obs},i}(k)$更新公式为

$$\boldsymbol{p}_{\mathrm{obs},i}(k+1)=\boldsymbol{p}_{\mathrm{obs},i}(k)+\boldsymbol{v}_{\mathrm{obs},i}(k)\cdot\Delta t \tag{3}$$

其中，$\boldsymbol{v}_{\mathrm{obs},i}(k)$为障碍物速度矢量，$\Delta t$为采样时间。

当障碍物与环境边界或静态障碍物的距离达到最小阈值时，算法通过模拟物理弹性碰撞机制，更新速度矢量方向

$$\boldsymbol{v}_{\mathrm{obs},i}(k+1)=\boldsymbol{v}_{\mathrm{obs},i}(k)-2\left(\boldsymbol{v}_{\mathrm{obs},i}(k)\cdot\boldsymbol{n}\right)\boldsymbol{n} \tag{4}$$

其中，n为碰撞平面的法向量。该机制确保了动态障碍物在有限空间内的持续运动和轨迹的不确定性，以此验证算法的鲁棒性。

1.3　环境建模

本文采用二维栅格地图法进行环境建模，地图设计为 25 m×20 m，其中包含移动机器人必须避开的静态障碍物，如图 1 所示。使用一个单元格数组来储存所有静态障碍物，每一个单元格代表一个独立的静态障碍物。为了实现多边形区域的填充与判定，障碍物顶点序列的首尾通常设为同一点，从而确保其形状的闭合性。

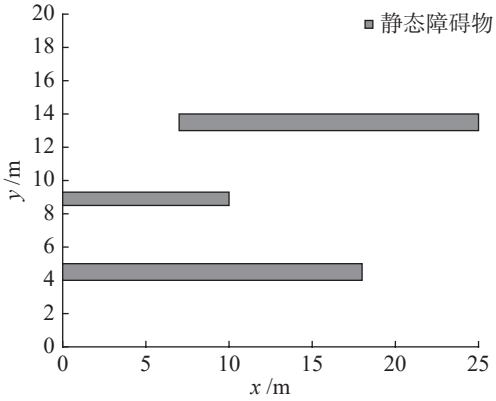


图 1　静态环境地图

Fig.1　Static environment map

动态障碍物具备与环境边界和静态障碍物进行碰撞反弹的交互特性，在环境中其位置随时间变化，是机器人必须实时规避的对象。算法确保动态障碍物的生成位置不与起点、终点、静态障碍物及其他动态障碍物的初始位置发生重叠。机器人的半径参数被用来确定机器人是否与静态或动态障碍物发生碰撞。在生成动态障碍物时，也会考虑机器人半径，避免动态障碍物初始位置过于靠近起点或终点。

本文为动态障碍物的行为设定了明确的动态变化依据。在基础运动模型方面，我们采用线性恒定速度(CV)模型来预测障碍物在短期内的轨迹。虽然 CV 模型无法精准描述障碍物的非线性变速行为，但得益于 NMPC 算法的滚动优化机制，在每个采样时刻利用最新的观测数据更新预测轨迹。这种高频的反馈校正特性能够有效抑制模型失配带来的误差，使机器人在面对变向或变速障碍物时，仍能通过快速重规划实现安全避障。在环境交互行为方面，动态障碍物被赋予了与环境进行物理碰撞后反弹的特性。这意味着当障碍物运动至边界或与静态

障碍物发生碰撞时，其运动状态会发生瞬时改变，模拟真实的物理反弹效果。这种非协作的且由事件驱动的交互机制增加了环境的不可预测性，从而能够严格考验算法在面对突发状况时的鲁棒性与实时响应能力。

2 全局路径规划和局部运动控制

2.1 启发式 RRT*算法

RRT*是 RRT(快速扩展随机树)算法的一种优化版本。RRT 算法可以在复杂高维空间中快速探索，但它通常无法找到最优路径。RRT*在 RRT 的基础上引入 2 个关键步骤，重连和选择最佳父节点，以确保渐近最优性，使得路径随着迭代次数趋于无穷大而收敛至最优。

为实现全局路径规划，启发式 RRT*算法在 RRT*的基础上引入了椭圆采样启发式策略，来高效寻找从起点到终点的无碰撞最优路径。椭圆采样启发式策略是启发式 RRT*算法的核心：一旦寻找到一个有效的路径，更高质量的路径必然存在于一个特定的子空间。这个子空间可以被定义为一个起点 x_{start} 和终点 x_{goal} 为焦点的椭球体。如图 2 所示，具体实现过程如下：

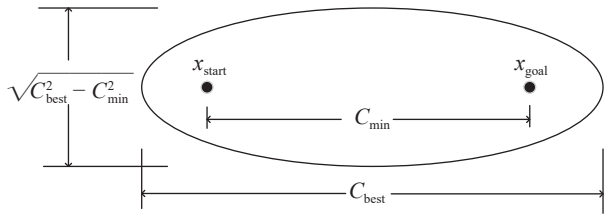


图 2 椭圆采样空间

Fig.2 Elliptical sampling space

启发式 RRT*具体实现过程如下：

步骤 1：初始化，使用标准的 RRT*算法在整个配置空间中进行探索，直到找到一条从起点到目标点的初始路径。计算这条初始路径的长度和起点到目标点的距离 C_{min}，并将初始路径长度设为当前最优路径长度 C_{best}。

步骤 2：根据当前的起点 x_{start}、终点 x_{goal} 和最优路径长度 C_{best}，定义一个椭圆采样区域。该椭圆的长轴长度为 C_{best}，短轴长度为 $\sqrt{C_{\text{best}}^2 + C_{\text{min}}^2}$。

步骤 3：在后续的迭代中，所有的随机采样点都只在这个椭圆区域内生成。为了在椭圆内均匀采样，可以先在一个单位圆内生成一个随机点，然后通过仿射变换将其映射到目标椭圆区域内。

步骤 4：在椭圆区域内进行采样，结合 RRT*的连接、重连和父节点优化机制，继续扩展随机树。每当找到一条比 C_{best} 更短的新路径时，就立即更新 C_{best} 的值。随着 C_{best} 值的减小，椭圆采样区域也会相应地收缩，从而进一步聚焦于更有潜力的搜索区域，加速算法的收敛。当达到预设的迭代次数或路径长度收敛到一定阈值时，算法终止并输出当前的最优路径。

启发式 RRT*算法在包含静态障碍物的环境中成功规划出一条无碰撞路径，图 3 展示了其搜索过程。其中，树状线条代表了启发式 RRT*算法在搜索过程中构建的扩展树，这些线条是从起点向四周随机采样并连接形成的，展示了算法对环境的探索过程。

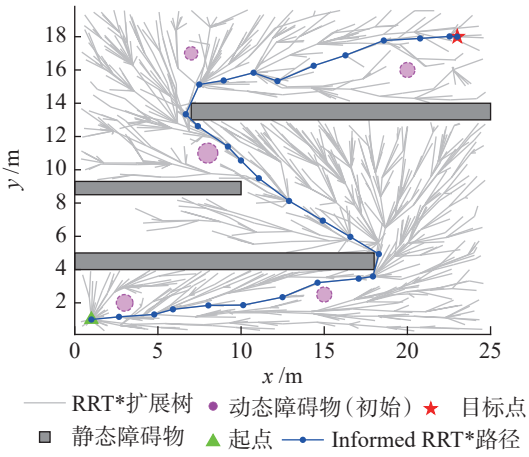


图 3 启发式 RRT*路径规划结果

Fig.3 Resulting path from informed RRT* planning

2.2 动态障碍物轨迹预测

实现 NMPC 控制器无碰撞路径规划的关键是对环境中动态障碍物未来位置的准确预测。采用简单高效的线性 CV 模型对障碍物的短期轨迹进行预测。在 NMPC(非线性模型预测控制)的短期预测时域内，假定每个动态障碍物都保持其当前的速度(包括大小和方向)进行匀速直线运动，CV 模型通过提供近似轨迹，实现了预测精度与计算效率之间的理想兼顾。

NMPC 本身需要在每个控制周期内求解一个复杂的优化问题，计算负担较重。CV 模型结构简单，预测过程无需迭代求解，极大地降低了整体算法的计算复杂度，有助于确保路径规划和控制的实时性。在 NMPC 的极短预测时域内，可以将许多动态障碍物的运动近似为匀速直线运动。由于控制器会高频重新规划，即使预测出现偏差，也可以在下

一个周期内快速纠正，因此在一定程度上缓解了模型误差带来的负面影响。

每一次迭代中，都会为 NMPC 控制器执行一次障碍物轨迹预测。在当前时刻k，会获取每一个动态障碍物的当前瞬时状态，包括当前位置以及速度，第i个动态障碍物的位置为$\boldsymbol{p}_{\mathrm{obs},i}(k+1)=\boldsymbol{p}_{\mathrm{obs},i}(k)+\boldsymbol{v}_{\mathrm{obs},i}(k)\cdot\Delta t$；其中障碍物当前位置为$\boldsymbol{P}_i(k)=\left[x_i(k),y_i(k)\right]^{\mathrm{T}}$；障碍物当前速度为$\boldsymbol{V}_i(t)=\left[v_{x_i}(t),v_{y_i}(t)\right]^{\mathrm{T}}$；$\Delta t$为采样时间。

NMPC 制器需要向前预测步数N_{p}和预测每个采样时间Δt，从而构建总时长$T_{\mathrm{pred}}=N_{\mathrm{p}}\cdot\Delta t$预测时域。算法会从当前时刻（$k=0$）开始，一直预测到时域末端（$k=N_{\mathrm{p}}$），对于预测时域中的每一个采样时间$\Delta t$，障碍物的未来位置是根据初始状态进行推算的。预测轨迹用作构建 NMPC 优化问题中非线性约束，确保机器人在启发式 RRT*的轨迹上与所有动态障碍物保持一个预定义的安全距离d_{safe}。

2.3　局部路径规划 NMPC 原理

NMPC 需要一个目标去跟随，由启发式 RRT*生成的全局路径，为 NMPC 的预测步数N_{p}生成一个详细的参考状态序列。首先找到当前机器人位置在全局路径上的投影点，然后沿着路径以机器人的最大速度向前采样，生成未来$N_{\mathrm{p}}+1$个离散采样点，这为优化器提供了明确的跟踪目标。

找到一个控制序列$\boldsymbol{U}_t$，使其轨迹跟踪成本最小。NMPC 控制器在每个采样时刻 t 求解以下优化问题，以找到最优的控制序列$\boldsymbol{U}_{t^*}$：

$$\boldsymbol{U}_{t^*}=\arg\min_{\boldsymbol{U}_t}J(\boldsymbol{x}_t,\boldsymbol{U}_t) \tag{5}$$

其中，$\boldsymbol{x}_t$是在时刻t的机器人当前状态，$\boldsymbol{x}_t=\left[x_{\mathrm{t}},y_{\mathrm{t}},\theta_{\mathrm{t}}\right]^{\mathrm{T}}$；$\boldsymbol{U}_t$是从时刻$t$开始，在预测时域$N_{\mathrm{p}}$内的控制输入序列$\boldsymbol{U}_t=\left\{u_0,u_1\cdots uN_{\mathrm{p}}-1\right\}$。

目标函数J是一个成本函数，通常由两部分组成：

$$J(\boldsymbol{x}_t,\boldsymbol{U}_t)=\sum_{k=0}^{N_{\mathrm{p}}-1}L(\boldsymbol{x}_{k+1|t},\boldsymbol{x}_{k|t}) \tag{6}$$

其中，L是单步的成本函数，包含两个部分，轨迹跟踪成本和控制能耗成本：

$$\begin{aligned}L(\boldsymbol{x}_{k+1|t},\boldsymbol{x}_{k|t})=&(\boldsymbol{x}_{k+1}-\boldsymbol{x}_{\mathrm{ref},k+1})^{\mathrm{T}}\boldsymbol{Q}(\boldsymbol{x}_{k+1}-\boldsymbol{x}_{\mathrm{ref},k+1})\\&+\boldsymbol{u}_k^{\mathrm{T}}\boldsymbol{R}\boldsymbol{u}_k\end{aligned} \tag{7}$$

轨迹跟踪成本用于衡量机器人预测状态与参考轨迹目标状态之间的偏差。控制能耗成本通过约束控制输入的大小，旨在抑制频繁变化的控制指令，从而提高机器人运动轨迹的平滑性与稳定性。

在目标函数中，为状态跟踪权重$\boldsymbol{Q}$中的位置项赋予了比姿态角项更高的权重值，以优先保证机器人能精确地跟随参考路径。为控制输入$\boldsymbol{u}$设置了较小的权重$\boldsymbol{R}$，允许控制器作出快速响应，在动态避障中表现出更高的灵活性。

控制输入饱和约束确保优化器产生的控制指令在机器人的可执行范围内：

$$\begin{aligned}v_{\min}\leqslant v_{k+t}\leqslant v_{\max}\\ \omega_{\min}\leqslant \omega_{k+t}\leqslant \omega_{\max}\end{aligned} \tag{8}$$

机器人在预测轨迹上的每一个点都必须与所有障碍物保持安全距离d_{safe}，不能发生任何碰撞。对于静态障碍物，机器人中心到该障碍物的最小距离必须大于安全距离

$$d_{\mathrm{static}}\geqslant r_{\mathrm{robot}}+d_{\mathrm{safe}} \tag{9}$$

对于动态障碍物，机器人预测位置$p_{\mathrm{robot}}(k)$与第j个动态障碍物的预测位置$p_{\mathrm{obs},j}(k)$之间的距离，必须大于它们半径之和加上安全距离。

$$\left|\left|p_{\mathrm{robot}}(k)-p_{\mathrm{obs},j}(k)\right|\right|_2\geqslant r_{\mathrm{robot}}+r_{\mathrm{obs},j}+d_{\mathrm{safe}} \tag{10}$$

算法通过预测动态障碍物的未来轨迹并将其作为严格约束进行求解，实现前瞻性的最优策略调整。在数值实现方面，本文将 NMPC 的有限时域优化问题转化为非线性规划问题。求解器基于 MATLAB 优化工具箱中的 fmincon 函数构建，并选用序列二次规划（SQP）算法进行迭代求解。SQP 算法通过将非线性约束问题近似为一系列二次规划子问题，在处理具有非线性动力学约束的优化问题时表现出优异的收敛性。为了提高求解效率满足实时性要求，采用热启动策略，即将上一时刻的最优控制序列作为当前时刻优化求解的初始猜测值。为了平衡计算精度与实时性，对优化器的终止条件进行了严格设定。最大迭代次数设为 50，函数容差$\varepsilon_{\mathrm{con}}$为$10^{-6}$。利用上一时刻的解进行热启动显著减少了每次迭代的搜索步数，使得单步平均求解时间远小于采样时间Δt，满足系统实时运行需求。

在 NMPC 框架中，预测时域的总时长为$T_{\mathrm{pred}}=N_{\mathrm{p}}\cdot\Delta t$必须覆盖机器人的最小安全刹车距离，即满足$T_{\mathrm{pred}}\cdot v_{\max}>\dfrac{v_{\max}^2}{2\mathrm{a}_{\max}}$，以确保控制器在检测到突发障碍时有足够的视距以便规划减速或绕行轨迹。本文采用 CV 模型预测障碍物轨迹。随着预测时间推移，线性预测与障碍物实际非线性运动的偏差会累积。T_{pred}不宜过长，避免将不确定性的障碍物位置作为硬约束，导致规划结果过于保守或无解。NMPC 的计算复杂度随N_{p}呈非线性增长，为保

证实时避障，必须满足单步求解时间 $T_{\text{cal}} < \Delta t$。$\Delta t$ 过小虽然能降低离散化误差，但会迫使 N_{p} 增大以维持相同的物理距离，从而增加计算负担。本文选取 $\Delta t = 0.1\text{s}$ 以平衡离散化精度与求解频率；选取 $N_{\text{p}} = 10$，使得预测视距 1.0s。这既保证了 $v_{\max}$ 下的有效避障反应时间，又确保 CV 模型在短时内的预测可信度。

2.4 停滞与摆脱模块设计

当机器人处于狭窄通道、多障碍物包围等复杂环境或受控制器局限性影响，导致机器人和目标点 x_{goal} 的距离大于设定阈值 $D_{\text{thresh_goal}}$ 的情况下，若线速度和角速度在持续时间内均小于预设的极小值，则判定机器人进入停滞状态并触发摆脱程序。

停滞状态检测模块旨在识别机器人是否陷入无法有效向目标前进的停滞僵局。该模块通过实时监测机器人的运动状态和机器人与目标的距离，在满足特定条件时自动启动摆脱机制。

在每个模拟时刻 t，控制器计算出线速度和角速度指令 $v_{\text{cmd}}(t)$ 和 $\omega_{\text{cmd}}(t)$。若此时机器人被判定为疑似停滞状态，则停滞状态连续计步器 $S_{\text{c}}(t) = S_{\text{c}}(t-1) + 1$，否则 $S_{\text{c}}(t) = 0$。

根据基础阈值 $S_{\text{max_orig}}$ 和仿真速度因子 f_{sim}，计算适配仿真场景的停滞状态最大连续步数阈值：

$$S_{\max} = \max(1, f_{\text{round}}(S_{\text{max_orig}}, f_{\text{sim}})) \tag{11}$$

计算机器人当前位置 $\boldsymbol{p}_{\text{robot}}(k)$ 与目标位置 $\boldsymbol{p}_G$ 的欧氏距离，衡量机器人与目标的远近，作为判断远离目标条件的依据：

$$\begin{aligned} d_{\text{goal}}(t) &= \|\boldsymbol{p}_{\text{robot}}(k) - \boldsymbol{p}_{\text{G}}\|_2 \\ &= \sqrt{(x_{robot}(t) - x_{\text{G}})^2 + (y_{\text{robot}}(\text{t}) - y_{\text{G}})^2} \end{aligned} \tag{12}$$

根据到达目标的距离阈值 $D_{\text{thresh_goal}}$ 和系数 K_{dist}，距离目标过远的判定标准 $D_{\text{far_goal}}$，用于区分机器人是否因离目标近而不触发停滞逻辑。其中判定标准计算公式：

$$D_{\text{far_goal}} = K_{\text{dist}} * D_{\text{thresh_goal}} \tag{13}$$

综合运动停滞和非动态避障两个条件,，判定机器人是否进入疑似停滞状态是后续停滞计数、最终判定的基础。

$$S_{\text{c}}(t) = \begin{cases} S_{\text{c}}(t-1) + 1, & P_{\text{stuck}}(\text{t})\text{为真} \\ 0, & P_{stuck}(t)\text{为假} \end{cases} \tag{14}$$

更新停滞状态连续计数器 $S_{\text{c}}(t)$，“疑似停滞”持续时计数累加，否则重置为 0，用于统计“疑似停滞”的连续步数。一旦检测到停滞状态，激活摆脱模块，根据当前摆脱阶段执行不同的动作序列。

当机器人启动摆脱程序时，它会首先执行一个动作序列来挣脱困境。第一步，它会以一个较小的负速度直线后退一段距离，从障碍物或角落中脱离，为自己下一步动作预留出所需的空间。然后在获得操作空间后，机器人会立即停止向后移动，并以一个较大的角速度按随机方向进行旋转，强行改变机器人的朝向。最后，在朝向一个新方向后，机器人会以中等速度直线前进，将自己从最初的被困位置脱离，并进入一个更有利于重新进行路径规划的新区域，为恢复正常导航做好准备。

移动机器人的摆脱策略常采用原地旋转以重构局部环境信息，或强制触发全局重规划。然而，对于受非完整约束限制的差速机器人，在狭窄空间中，如 U 形陷阱中，单纯的旋转往往无法改变运动学受限的局面。相比之下，本文提出的策略引入了后退摆脱机制，利用后退动作主动扩大机器人的前方转弯半径，从而为后续的重规划和避障预留了必要的几何空间。

3 仿真实验与结果分析

3.1 随机动态障碍物环境仿真

为了评估算法在随机动态环境下的性能，本文设计了一个包含固定多边形静态障碍物和多个随机生成非协作性动态障碍物的环境。要求移动机器人利用 NMPC-IRRT*算法不仅能够预测障碍物的短期线性轨迹，还必须能够鲁棒地处理由碰撞引起的瞬时改变。启发式 RRT*算法参数配置如表 1，状态跟踪权重 $\boldsymbol{Q} = \text{diag}\left(\begin{bmatrix} 10 & 10 & 0.5 \end{bmatrix}^{\text{T}}\right)$；控制输入权重 $\boldsymbol{R} = \text{diag}\left(\begin{bmatrix} 0.1 & 0.1 \end{bmatrix}^{\text{T}}\right)$。本文使用 MATLAB2021b 版本进行实验，在相同的两种障碍物环境下，将 NMPC-IRRT*算法与 APF 法、DWA-IRRT*法进行比较，其中启发式 RRT*最大迭代次数为 3 500。

表 1 控制参数

Tab.1 Control parameters

参数	数值	参数	数值
N_{p}	10	$\Delta t/\text{s}$	0.1
d_{safe}/m	0.1	f_{sim}	1.0
$v_{\max}/\left(\text{m}\cdot\text{s}^{-1}\right)$	5.0	K_{dist}	1.0
$v_{\min}/\left(\text{m}\cdot\text{s}^{-1}\right)$	0	$S_{\text{max_orig}}$	2.5
$\omega_{\max}/\left(\text{rad}\cdot\text{s}^{-1}\right)$	$\pi/2$	$r_{\text{robot}}/\text{m}$	0.3
$\omega_{\min}/\left(\text{rad}\cdot\text{s}^{-1}\right)$	$-\pi/2$	$D_{\text{thresh_goal}}/\text{m}$	1.5

地图大小设置为 25(m)×20(m)，移动机器人的起点和终点位置分别为 [1,1] 和 [23,18]，折线表示移动障碍物移动轨迹，连接采样点的规划线段表示启发式 RRT*搜索的最终路径。图 4 所示为在 MATLAB 环境中设置的类似仿真环境。

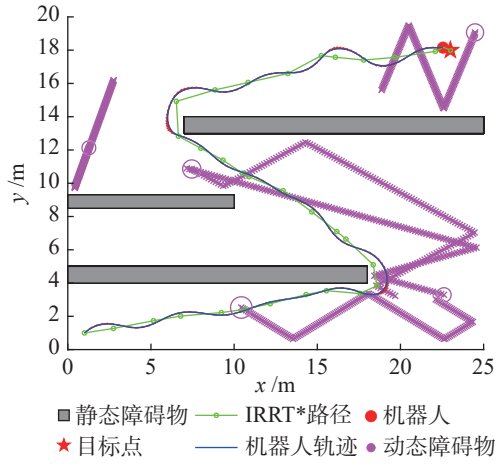


图 4　随机动态障碍物环境仿真

Fig.4　Simulation of random dynamic obstacle environments

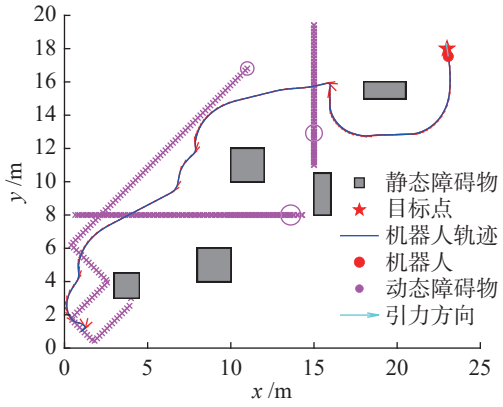
为验证算法的有效性，将本文提出的 NMPC-IRRT*与其他算法在稀疏障碍环境和复杂障碍环境地图中进行对比，地图中均存在移动障碍物。两种环境特性的主要区别在于障碍物的布局和由此产生的对机器人导航的约束程度。稀疏障碍环境包含少量分散的静态障碍物，拥有充足的路径规划自由空间，为机器人提供了充足的机动区域。复杂障碍物环境中，机器人的可通行区域被静态障碍物划分成狭窄且弯曲的 S 形通道，机器人必须在狭窄的空间内进行精确的导航和避障。

为客观地评估和比较不同路径规划算法在各种环境下的性能，本文引入了一个量化的环境复杂性指标(I_{EC})。复杂性指标由障碍物密度ρ、路径曲折度τ、可通行空间狭窄度ϑ加权组合而成。其中，障碍物密度ρ为地图中障碍物区域总面积与地图面积的比例；路径曲折度τ为起点到终点的无碰撞最短路径长度与两点间的欧几里得直线距离比值；可通行空间狭窄度ϑ是基于可通行空间的中轴线，通过计算最优路径L_{best}上每一点到最近障碍物的安全间隙，用这些间隙的倒数的平均值来表示狭窄度。加权求和$I_{EC}=\omega_1\rho+\omega_2\tau+\omega_3\vartheta$且$\omega_1+\omega_2+\omega_3=1$。实验中，稀疏障碍物环境$I_{EC}$为 0.23，复杂障碍物环境$I_{EC}$为 0.81。

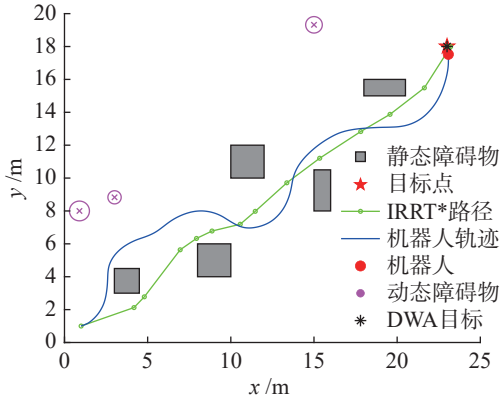
为消除随机性，确保对比的公平性，在实验对比时将动态障碍物固定起始位置和运动轨迹，固定动态障碍物的初始点以及初始运动方向，尽可能地让动态障碍物与移动机器人产生交集。本文将机器人看作质点，为平衡启发式 RRT*算法随机采样带来的不稳定性，分别对每种地图每种算法设置 50 次独立仿真实验，最终取平均值进行对比分析。

3.2　稀疏障碍物环境仿真

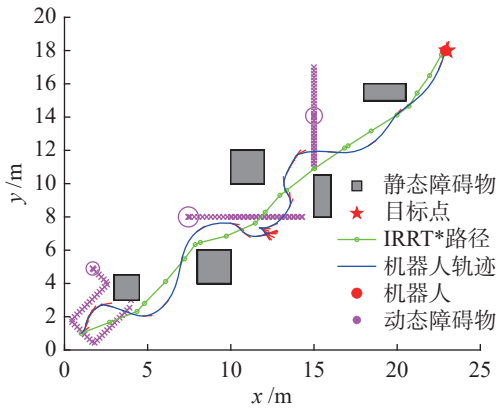
稀疏障碍物环境下各算法路径规划结果示意如图 5 所示。由图 (a) 可知，人工势场算法中当机器



(a) 人工势场算法



(b) DWA-IRRT*算法



(c) NMPC-IRRT*算法

图 5　稀疏障碍物环境下各算法路径规划结果

Fig.5　Path planning results of various algorithms in sparse obstacle environments

人在多个障碍物之间穿行时，路径不平滑，存在较为突出的曲折段，且机器人轨迹有向内凹陷后再绕出的变化趋势。由图 (b) DWA-IRRT*算法可以看出，机器人轨迹紧密跟随全局路径，但面对静态障碍物，它会保持一个很大的安全距离，导致路径被拉长。

图 (c)NMPC-IRRT*算法在其预测时域内考虑了动态障碍物的未来位置，使得机器人能够做出平滑且具有预见性的规避动作，生成的轨迹极为平滑，能高度一致地跟踪全局路径，轨迹调整自然流畅，未出现 DWA-IRRT*算法中可能出现的局部急转弯现象。

分别用传统人工势场(APF)算法、DWA-IRRT*算法、NMPC-IRRT*算法分别在稀疏障碍物地图中进行路径规划、迭代 50 次，平均路径长度、平均运行时间以及规划成功率的对比结果见表 2。

表 2　稀疏障碍物地图各算法性能对比

Tab.2　Performance comparison of algorithms for sparse obstacle maps

算法	平均路径长度 /m	平均运行时间 /s	规划成功率 /%a
APF	38.15	62.40	100
DWA-IRRT*	31.93	35.60	100
NMPC-IRRT*	23.3	31.82	100

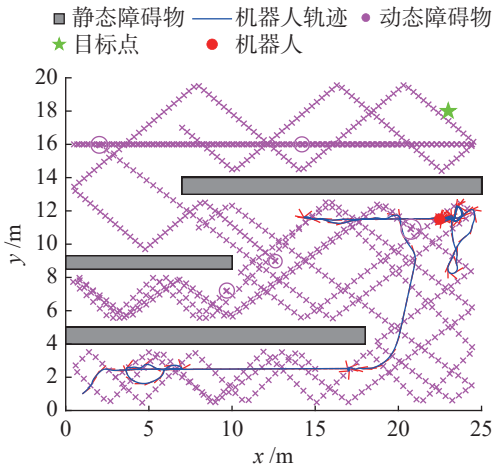
从表 2 中可以看出，虽然 3 种算法都能保证在稀疏障碍物地图下成功到达目标点，但是在平均路径长度上形成了鲜明的对比。与 APF 算法相比，DWA-IRRT*算法规划的路径平均缩短了约 16.30%。而 NMPC-IRRT*算法更具优势，其规划的路径平均比 APF 缩短了约 38.93%。这表明 NMPC-IRRT*算法在生成更短路径方面的优越性。

通过计算，在平均运行时间上，DWA-IRRT*算法较 APF 算法，缩短了约 42.95%；NMPC-IRRT*算法则进一步将平均运行时间缩短了约 49.01%。NMPC-IRRT*算法不仅能够规划出比 APF 算法短近 40% 的路径，还能将任务完成时间缩短近一半。

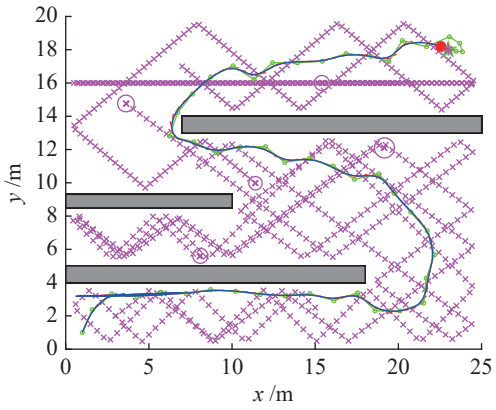
3.3　复杂障碍物环境仿真

复杂障碍物环境下各路径规划结果示意图如图 6 所示。静态障碍物将环境分割成了三个狭长且弯曲的 S 形通道，并增加了动态障碍物数量在狭窄的通道内移动。

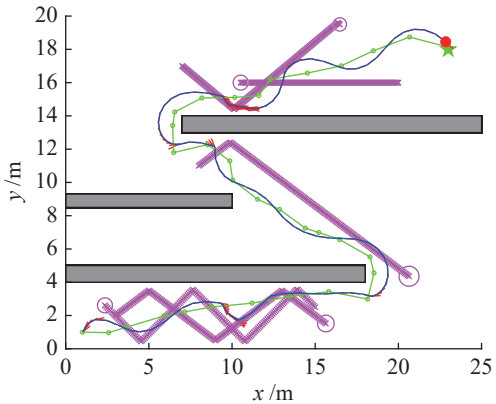
图 (a) 中 APF 算法下机器人轨迹严重震荡，由



(a) 人工势场算法



(b) DWA-IRRT*算法



(c) NMPC-IRRT*算法

图 6　复杂障碍物环境下各算法路径规划结果

Fig.6　Path planning results of various algorithms in complex obstacle environments

于引力与斥力在非目标位置达到平衡，机器人彻底丧失了移动能力，无法到达预定的目标点。DWA-IRRT*算法在动态障碍物靠近时，机器人会做出剧烈的、非最优的规避动作，这导致完成一次规避动作后，需要花费很长时间才能重新调整姿态，如图 (b) 所示。

图 (c)NMPC-IRRT*算法成功规划出了一条高质

量的全局路径。路径减少了不必要的曲折，为机器人提供了一条理论上最短且最高效的导航通道。即使在狭窄的通道内，它也能保持稳定的轨迹。在面对动态障碍物时，能够计算出最优且微小的调整，以最小的路径偏离和速度变化完成避障，然后立刻回到全局路径上。

从表 3 中可以看出，在最基本的任务完成能力上，APF 表现极差，完全不可使用。虽然 DWA-IRRT*的成功率较高，表明它在大多数情况下能够成功找到路径，但仍有一定概率规划失败或未能满足某些终止条件。在复杂的障碍物地图中，NMPC-IRRT*展现了其强大的鲁棒性。

表 3　复杂障碍物地图各算法数据对比

Tab.3　Data comparison of various algorithms on complex obstacle maps

算法	平均路径长度 /m	平均运行时间 /s	轨迹曲率方差 /m^{-2}	角速度变化率 /(rad/s^2)	规划成功率 /%
APF	—	—	314.112 4	2.525 5	0
DWA-IRRT*	78.39	180.9	101.944 0	1.492 2	87.3
NMPC-IRRT*	52.57	51.47	98.400 7	1.117 8	100

在路径规划的效率和质量上，NMPC-IRRT*占据优势，平均运行时间从 180.9 秒降低至 51.47 秒，效率提升超过两倍。这种显著的时间优势主要源于分层规划框架中的两个核心组件的高效协同，相比于传统的 RRT*算法需要在整个地图空间中持续盲目搜索，椭圆采样策略极大地减少了冗余的采样节点数量，加速了算法向最优路径的收敛，NMPC-IRRT*在生成高质量初始全局路径阶段的耗时相对较短。其次，NMPC 是一种具有预测性的优化控制策略。它通过预测未来一段时间内机器人和障碍物的状态，来求解一个最优控制序列，提前规划出一条平滑绕开动态障碍物并能迅速回归参考轨迹的路径，避免了不必要的时间损耗。

从轨迹的平滑度和执行稳定性来看，NMPC-IRRT* 依然是较合适的选择。其轨迹曲率方差和角速度变化率均为最低，这意味着它生成的路径拐弯更平缓，机器人在执行时姿态变化也更流畅、稳定。

在 50 次独立重复实验中，NMPC-IRRT*算法实现了 100% 的规划成功率，这得益于上层启发式 RRT 提供的全局拓扑引导，避免了算法陷入局部极小值。相比之下，DWA-IRRT*算法的成功率为 87.3%。在 S 形通道急转弯处，DWA 局部规划器因缺乏全局视野，容易在动态障碍物逼近时陷入震荡，导致超时。当动态障碍物与静态障碍物形成死角时，缺乏预测能力的 DWA 算法无法提前减速或变道，导致最终发生碰撞。人工势场法（APF）在复杂 S 形通道中极易在 U 形陷阱处产生极小值点，导致其在复杂场景下的成功率为 0%。上述实验对比验证了分层预测控制架构在提升复杂动态环境下鲁棒性方面的优势。

3.4　动态障碍物密度对算法性能的影响

为进一步验证算法在动态复杂环境下的鲁棒性与实时性，针对环境中动态障碍物数量增长的情况，本文在 3.3 节所述的复杂地图基础上，进行了不同动态障碍物密度下的测试。实验设置动态障碍物数量从 5 个逐渐增加至 30 个，每个密度等级下进行 50 次独立仿真，如图 7 所示。

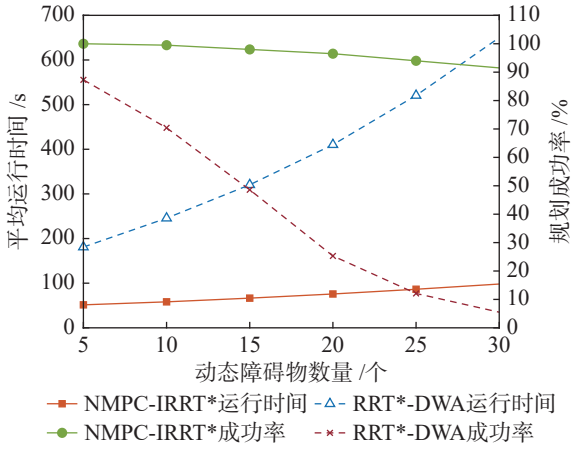


图 7　动态障碍物密度对算法性能的影响

Fig.7　Impact of dynamic obstacle density on algorithm performance

RRT*-DWA 算法在 5 个障碍物中，其平均运行时间已达 180.9 s。随着障碍物数量增多，该指标呈现指数级上升，在 30 个障碍物时机器人运行时间突破 600 s。在高密度环境下，基于局部避障的 DWA 易陷入局部极小值，导致机器人频繁触发停

滞检测机制并反复调用 RRT*进行全局重规划，从而消耗了大量计算资源和时间。相比之下，NMPC-IRRT*算法在基准测试中仅耗时 51.47 s。随着环境拥挤度增加，运行时间仅表现出平缓的线性增长。即使在 30 个障碍物的情况下，平均运行时间仍控制在 100 s 左右，这得益于 NMPC 的轨迹预测功能。算法能提前解算出未来一段时间内的无碰撞轨迹，通过平滑的速度调整主动规避潜在威胁，减少了因紧急避障导致的停顿和路径迂回。

DWA-IRRT*算法的成功率随障碍物密度增大出现断崖式下降。当障碍物数量超过 15 个时，环境过于狭窄且动态变化剧烈，DWA 难以搜索到有效速度指令，导致成功率迅速跌破 50%，在 30 个障碍物时仅为 5.5%，基本丧失了导航能力。NMPC-IRRT*算法随着障碍物增多，求解可行解的难度加大，但成功率始终保持在 90% 以上。这证明了全局路径引导与 NMPC 局部优化的分层架构能够有效应对高动态、高密度的复杂环境，保证机器人安全到达目标点。

3.5 摆脱性能分析

尽管 NMPC 在局部避障中表现优异，但在面对未知的陷阱型静态环境或动态障碍物封堵出口的情况时，常规摆脱策略往往采用原地旋转法以重构局部环境信息，或强制触发全局重规划。为了验证本文提出的摆脱模块的有效性，本节设计了一个 U 形陷阱，如图 8 所示。

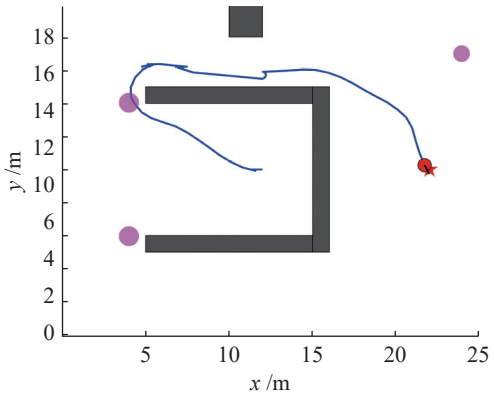


图 8 U 形陷阱场景下的停滞摆脱轨迹仿真

Fig.8 Simulation of stagnation recovery trajectory in U-shaped trap scenario

常规策略下，机器人受困于 U 形底部的斥力场与目标引力的平衡点。由于缺乏后退策略，机器人始终紧贴障碍物，受限于差速底盘的非完整约束，无法获得足够的转弯半径。面对出口处的动态封锁，机器人只能在狭小空间内反复原地打转与急停，最终因超时导致任务失败。在摆脱策略下，一旦检测到停滞状态，机器人会通过主动后退迅速拉开与障碍物的距离，进而为后续避障开拓出充足的操作空间。

实验对比了两种摆脱策略在 30 次独立仿真中的表现，如表 4 所示。原地旋转策略本质上是一种被动恢复机制，当移动机器人陷入 U 形陷阱，机器人同时受到目标点的强引力和墙壁的斥力，陷入了稳态的局部极小值。旋转能更新局部地图，但无法改变机器人的物理位置。机器人一旦尝试移动，又会立刻被斥力推回原点，反复挣扎直至超时，成功率为 0%。摆脱策略的成功得益于对空间的重新利用，生成的逃逸轨迹曲率方差显著降低，有效避免了在狭窄空间内因反复调整姿态而引发的碰撞风险。

表 4 U 形陷阱场景下摆脱策略性能对比

Tab.4 Performance comparison of escape strategies in U-Shaped trap scenarios

对比策略	轨迹平滑度/m^{-2}	平均耗时/s	摆脱成功率/%
摆脱策略	2.15	12.8	100
原地旋转	12.45	> 60	0

3.6 消融实验分析

为了进一步验证所提出的 NMPC-IRRT*分层规划框架中各个核心组件的有效性，本节设计了消融实验。，量化评估启发式 RRT*在全局路径规划中的优化作用，以及 NMPC 作为局部路径规划器的优越性。实验环境采用第 3.3 节中的复杂障碍物环境。

为验证上层启发式 RRT*引入的椭圆采样启发式策略的有效性，将其替换为标准的 RRT* 算法，构成一个变体算法，记为 NMPC-RRT*(非线性模型预测控制-RRT*)。该算法的局部规划器仍使用 NMPC，仅全局路径的生成方式有所不同。将 NMPC-RRT* 与完整的 NMPC-IRRT*算法进行对比。为验证下层局部规划器中 NMPC 相较于其他避障方法的优势，我们设计了另外两个变体算法。在该算法中，我们保留上层的启发式 RRT* 全局规划器，但将下层的 NMPC 局部规划器替换为广泛使用的动态窗口法和反应式规划器，分别记为 DWA-IRRT*(动态窗口启发式 RRT*)和 ZPNMPC-IRRT*(零预测启发式 RRT*)。将其与完整的 NMPC-IRRT* 算法进行对比，重点考察路径平滑度和动态避障能力，结果如图 9 所示。

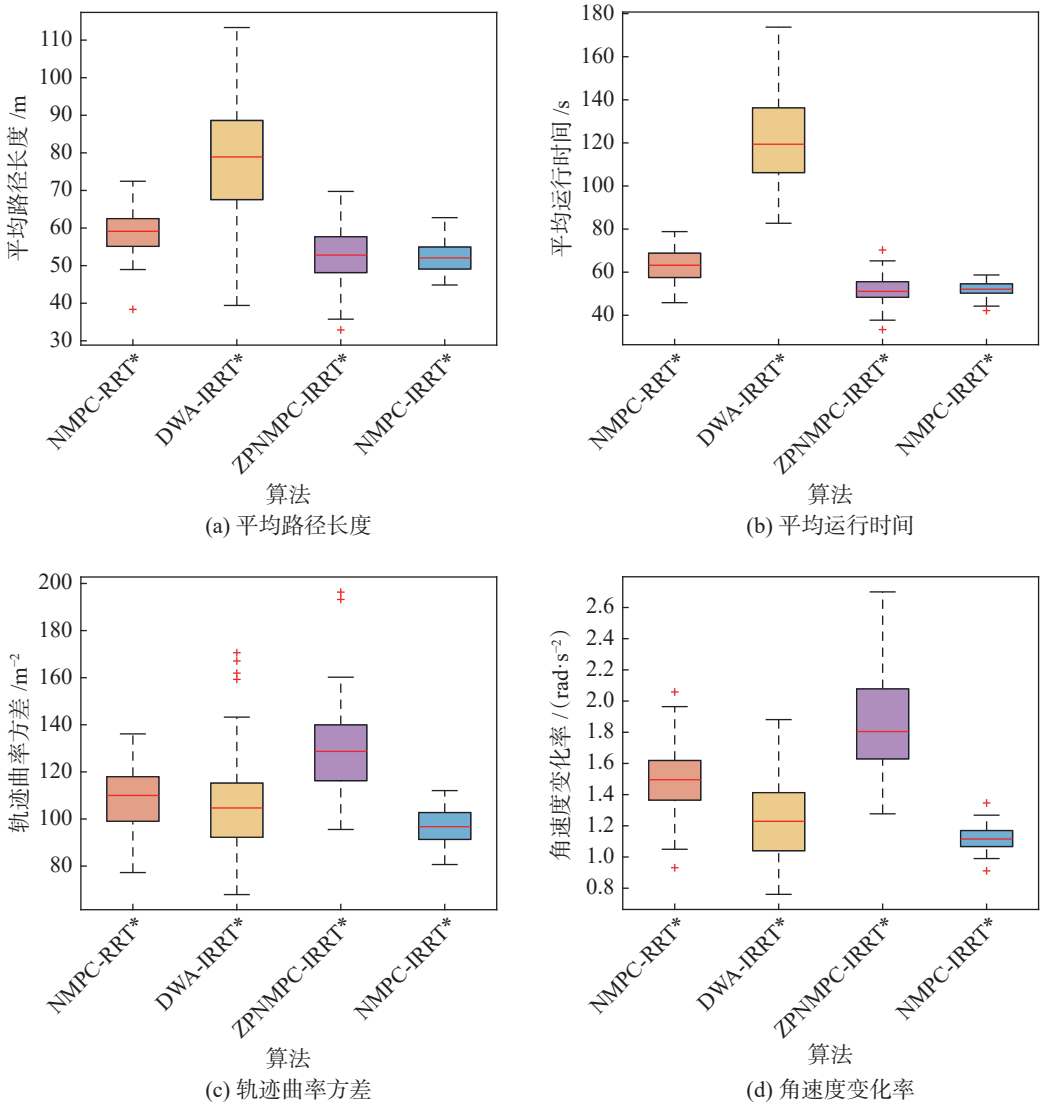


图 9　算法性能消融实验对比分析

Fig.9　Comparative analysis of algorithm performance ablation experiments

去除椭圆启发式采样后，算法的平均运行时间显著增加。这是因为标准的 RRT*在整个地图空间内进行盲目搜索，所以其收敛速度远慢于启发式 RRT*。虽然最终路径质量相近，但启发式 RRT*模块在提升规划效率上的优势显而易见。启发式 RRT*的椭圆启发式采样是提升算法整体效率和路径质量的关键。它通过聚焦于椭圆有效区域，快速生成高质量的全局路径，显著减少了无效搜索和运行时间。

在 NMPC 控制器去除轨迹预测能力后，DWA-IRRT*的路径长度和运行时间是所有算法中最差的，DWA 只根据当前时刻的障碍物信息做出规避，常常导致过度保守的、非最优的绕行，极大地浪费了时间和路程，平均路径长度显著增加。反应式控制器只能应对眼前的威胁，当与动态障碍物相遇时，往往会做出剧烈的、非最优的规避动作(如急转或长时间等待)，导致轨迹冗长甚至规划失败。而完整的 NMPC-IRRT*算法能够“预见”未来的碰撞风险，从而提前进行平滑、微小的轨迹调整，以近乎最优的方式完成避障。

4　结论

本文提出了一种适用于动态环境移动机器人的分层路径规划 NMPC-IRRT*算法。上层启发式 RRT*通过结合静态障碍物布局和动态障碍物的初始位置，采用椭圆采样启发式策略，生成高质量的初始全局路径。在下层，通过 NMPC 局部导航控制器，优先处理潜在碰撞，并利用线性运动模型对动态障碍物的未来轨迹进行预测，实现对动态障碍物威胁的实时响应。结果证明，这种全局信息引导的规

划与鲁棒、多方面的局部控制器的结合，为实际应用中自主机器人导航提供了一个有前景的解决方案。

本文采用计算高效的 CV 模型，但在面对具有强非线性意图(如追逐、急停)的障碍物时，预测精度存在局限。未来工作将考虑引入卡尔曼滤波或基于深度学习的轨迹预测模块，以平衡计算效率与长时预测精度。NMPC 需要在每个控制周期内求解非线性优化问题。随着环境中动态障碍物数量的增加，构建约束条件和求解优化的计算负荷呈非线性增长。在计算资源受限的低成本嵌入式硬件上，可能会面临实时性挑战。未来将结合长短期记忆网络(LSTM)或 Transformer 等深度学习方法，对动态障碍物的未来轨迹进行更精准的非线性预测，以替代现有的线性模型。

参考文献

[1] 孙哲, 马胜男, 解相朋, 等. 基于仿生算法的多式联运路径规划方法综述[J]. 控制与决策, 2025, 40(2): 375 − 386.
SUN Z, MA S N, XIE X P, et al. Bio-inspired optimization-based path planning algorithms in multimodal transportation: A survey[J]. Control and Decision, 2025, 40(2): 375 − 386.

[2] YANG J, LI S B, WANG Z, et al. Using deep learning to detect defects in manufacturing: A comprehensive survey and current challenges[J/OL]. Materials, 2020, 13(24)[2025-5-8]. https://doi.org/10.3390/ma13245755.

[3] 闫皎洁, 张锲石, 胡希平. 基于强化学习的路径规划技术综述[J]. 计算机工程, 2021, 47(10): 16 − 25.
YAN J J, ZHANG Q S, HU X P. Review of path planning techniques based on reinforcement learning[J]. Computer Engineering, 2021, 47(10): 16 − 25.

[4] TANG Y, ZAKARIA M A, YOUNAS M. Path planning trends for autonomous mobile robot navigation: A review[J/OL]. Sensors, 2025, 25(4)[2025-4-20]. https://doi.org/10.3390/s25041206.

[5] ZHANG S, LU C W, JIANG S, et al. An unmanned intelligent transportation scheduling system for open-pit mine vehicles based on 5G and big data[J/OL]. IEEE Access, 2020, 8[2025-6-11]. https://ieeexplore.ieee.org/document/9146106. DOI: 10.1109/ACCESS.2020.3011109.

[6] ZHOU X, YAN J, YAN M, et al. Path planning of rail-mounted logistics robots based on the improved Dijkstra algorithm[J/OL]. Applied Sciences, 2023, 13(17)[2025-4-13]. https://doi.org/10.3390/app13179955.

[7] BAI X S, WANG L, HU Y, et al. Optimal path planning method for IMU system-level calibration based on improved Dijkstra's algorithm[J]. IEEE Access, 2023, 11: 11364 − 11376.

[8] LI Y G, JIN R C, XU X R, et al. A mobile robot path planning algorithm based on improved A* algorithm and dynamic window approach[J]. IEEE Access, 2022, 10: 57736 − 57747.

[9] 项前, 梁光煜, 鲍劲松, 等. 安全高效的多四向穿梭车路径规划及实时避碰[J]. 计算机集成制造系统, 2025, 31(8): 2755 − 2771.
XIANG Q, LIANG G Y, BAO J S, et al. Safe and efficient multiple four-way shuttles path planning and real-time collision avoidance[J]. Computer Integrated Manufacturing Systems, 2025, 31(8): 2755 − 2771.

[10] 成怡, 肖宏图. 融合改进 A*算法和 Morphin 算法的移动机器人动态路径规划[J]. 智能系统学报, 2020, 15(3): 546 − 552.
CHENG Y, XIAO H T. Mobile-robot dynamic path planning based on improved A* and Morphin algorithms[J]. CAAI Transactions on Intelligent Systems, 2020, 15(3): 546 − 552.

[11] CHEN S, YANG G, CUI G, et al. Improved path planning and controller design based on PRM[J]. IEEE Access, 2025, 13: 44156 − 44168.

[12] ZHANG R, GUO H, ANDRIUKAITIS D, et al. Intelligent path planning by an improved RRT algorithm with dual grid map[J]. Alexandria Engineering Journal, 2024, 88: 91 − 104.

[13] CAO M L, ZHOU X J, JU Y P. Robot motion planning based on improved RRT algorithm and RBF neural network sliding[J]. IEEE Access, 2023, 11: 121295 − 121305.

[14] KANG M L, CHEN Q H, FAN Z M, et al. A RRT based path planning scheme for multi-DOF robots in unstructured environments[J/OL]. Computers and Electronics in Agriculture, 2024, 218[2025-7-21]. https://doi.org/10.1016/j.compag.2024.108707.

[15] 刘乐, 李斌, 方一鸣, 等. AGTB-RRT*: 一种基于近似测地线避障策略的目标双向 3D RRT*路径规划算法[J]. 控制与决策, 2024, 39(11): 3763 − 3771.
LIU L, LI B, FANG Y M, et al. AGTB-RRT*: An approximately geodesic target-biased bidirectional 3D RRT* path planning

algorithm[J]. Control and Decision, 2024, 39(11): 3763 − 3771.

[16] WANG Z, LI P, WANG Z, et al. APG-RRT: Sampling-based path planning method for small autonomous vehicle in closed scenarios[J]. IEEE Access, 2024, 12: 25731 − 25739.

[17] CUI X N, WANG C Q, XIONG Y, et al. More quickly-RRT*: Improved quick rapidly-exploring random tree star algorithm based on optimized sampling point with better initial solution and convergence rate[J/OL]. Engineering Applications of Artificial Intelligence, 2024, 133[2025-9-8]. https://doi.org/10.1016/j.engappai.2024.108246.

[18] 栾添添, 王皓, 孙明晓, 等. 基于动态变采样区域 RRT 的无人车路径规划[J]. 控制与决策, 2023, 38(6): 1721 − 1729.
LUAN T T, WANG H, SUN M X, et al. Unmanned vehicle path planning based on dynamic variable sampling region RRT[J]. Control and Decision, 2023, 38(6): 1721 − 1729.

[19] YANG J, ZHANG H, NING P. Path planning and trajectory optimization based on improved APF and multi-target[J]. IEEE Access, 2023, 11: 139121 − 139132.

[20] JAYAWEERA H M, HANNOUN S. A dynamic artificial potential field (D-APF) UAV path planning technique for following ground moving targets[J]. IEEE Access, 2020, 8: 192760 − 192776.

[21] HU L, CAI C. ARIME-DWA: Robot path planning utilizing advanced RIME optimization and dynamic windows approach[J]. IEEE Access, 2024, 12: 191480 − 191497.

[22] BAI Z, PANG H, HE Z, et al. Path planning of autonomous mobile robot in comprehensive unknown environment using deep reinforcement learning[J]. IEEE Internet of Things Journal, 2024, 11(12): 22153 − 22166.

[23] GANESAN M, KANDHASAMY S, CHOKKALINGAM B, et al. A comprehensive review on deep learning-based motion planning and end-to-end learning for self-driving vehicle[J]. IEEE Access, 2024, 12: 66031 − 66067.

[24] LIU S, JIANG H, CHEN S P, et al. Integrating Dijkstra's algorithm into deep inverse reinforcement learning for food delivery route planning[J/OL]. Transportation Research Part E: Logistics and Transportation Review, 2020, 142[2025-03-01]. https://doi.org/10.1016/j.tre.2020.102070.

[25] KIM Y, PAE D S, JANG S H, et al. Reinforcement learning for autonomous vehicle using MPC in highway situation[C]//2022 International Conference on Electronics, Information, and Communication. Piscataway, USA: IEEE, 2022: 1 − 4.

[26] FENG N, GUO X, YU X, et al. Human-Robot sharing operation in co-transporting for nonholonomic mobile robot[J]. IEEE Transactions on Cognitive and Developmental Systems, 2025, 17(5): 1087 − 1097.

作者简介

陈　阳(2001—)，女，硕士。研究领域为无人机路径规划等。

黄宜庆(1983—)，男，博士，教授。研究领域为多智能体协同，线性变参数控制，切换控制等。