基于 RRT 和 DWA 的欠驱动 UUV 路径规划

严浙平，黄俊儒，吴　迪
（哈尔滨工程大学 自动化学院，黑龙江 哈尔滨 150001）

摘　要　针对无人潜航器（UUV）在未知水下复杂环境的路径规划问题，设计了随机树以及动态窗口的融合算法。该算法基于快速扩展随机树（RRT）以及动态窗口（DWA）两层规划设计，第一层利用随机树算法快速规划出全局路径，在此基础上第二层加载全局路径，针对 UUV 模型的欠驱动和非线性，利用动态窗口算法完成局部路径规划，保证约束条件下 UUV 路径的安全性。通过融合参数 μ 修正内外框架的融合度，有效地弥补了全局路径算法的无法躲避动态障碍物的缺点以及局部路径算法全局能力低下的问题。最后，通过对比仿真验证了融合算法相比于随机树全局算法和动态窗口局部算法的优越性。

关键词　水下无人潜航器；快速扩展随机树；动态窗口；路径规划
中图分类号　TP301.6：U661　　**文献标识码**　A　　**文章编号**　2096-5753(2020)03-0258-07
DOI　10.19838/j.issn.2096-5753.2020.03.014

Underactuated UUV Path Planning Based on RRT and DWA

YAN Zheping，HUANG Junru，WU Di
（College of Automation，Harbin Engineering University，Harbin 150001，China）

Abstract　Aiming at the path planning of unmanned underwater vehicle（UUV）in unknown underwater complex environment，an RRT-DWA fusion algorithm was designed. The algorithm is based on two-layers planning design of rapid expansion random tree（RRT）and dynamic window（DWA）. The first layer uses the RRT algorithm to quickly plan the global path. Based on this and considering the underactuation and non-linearity of the UUV model，the second layer loads the global path，and the DWA algorithm is used to complete the local path planning to ensure the safety of the UUV path under the constraints. The fusion parameters of the inner and outer frames are corrected by the fusion parameters μ，which effectively makes up for the shortcomings of the global path algorithm that cannot avoid dynamic obstacles and improves the low efficiency of the local path algorithm. Finally，the superiority of the RRT-DWA fusion algorithm compared with the RRT global algorithm and the DWA local algorithm is verified through comparative simulation.

Key words　unmanned underwater vehicle；RRT；DWA；path planning

0　引言

水下无人潜航器（Unmanned Underwater Vehicle，UUV）的运动规划是 UUV 核心技术之一，本文重点讨论水平面规划。路径规划以功能和侧重点分为全局路径规划和局部路径规划，两者有区别但又可以相互转化。全局路径规划是在已知周围环境的前提下可以快速规划出一条最优路径，但全局路径规划有时不考虑动力学约束，并且水下环境复杂，环境信息获取并不完全，容错率较低；局部路径规划侧重于局部环境规避未知障碍物的能力，但其寻找到最优全局路径的效率低下。如何使 UUV

收稿日期：2020-04-13
作者简介：严浙平（1972-），男，博士，教授，主要从事水下无人潜航器相关研究。

快速安全地到达目标点是一个重要的研究方向。

全局路径规划有很多成熟的算法，早在 1958 年就有了 Dijsktra 算法，改进后又有了 A*和 D*算法，以及蚁群算法、贪心算法、粒子群算法等。丁帅以减少水下机器人能量消耗为前提，基于 RRT* 算法设计了能耗最低的路径规划算法[1]；孙波针对移动机器人的路径规划提出了粒子群优化算法[2]；邓超针对 UUV 三维空间移动提出了双种群粒子群算法[3]。局部路径规划一般配备了各类传感器以获取局部地图或者探测未知障碍物，其算法也有人工势场法以及动态窗口法。同时也有衍生的改进和结合的算法：黄辰考虑到机器人的尺寸，通过激光雷达获取局部地图，改进了 DWA 算法[4]；何壮壮在 ROS 环境下提出了针对两轮机器人的 D*和 DWA 结合算法等[5]。

针对欠驱动 UUV 的研究也比较成熟，张伟针对垂直面的路径跟踪目标设计了基于模型预测连续系统跟踪控制器，避免离散化，提高了控制器的泰勒展开阶次，提高了精度[6]。赵玉飞针对 UUV 近海底的路径规划问题，提出了改进粒子群算法[7]。该算法可以有效规避静态障碍物，但是无法规避动态障碍物，针对这一问题，本文提出了快速扩展随机树以及动态窗口的融合算法框架。该算法一方面继承了 RRT 算法的全局规划能力，一方面继承了 DWA 算法的实时避障能力，实现了实时避障的全局路径规划。通过修改外部框架的距离参数可以提高全局路径的精度，同时通过修改内部框架的评价函数参数可以使 UUV 应对不同的需求和地形，最后设计了融合参数 μ，通过修改 μ 来确定内外框架的融合度。

1　UUV 水平面数学模型

欠驱动 UUV 路径规划为水平面的规划，所以不考虑垂直方向运动，水平面模型由欠驱动 UUV 六自由度模型在水平面方向解耦得到。

水平面运动学模型：

$$\begin{aligned}\dot{x}&=u\cos\psi-v\sin\psi\\ \dot{y}&=u\sin\psi+v\cos\psi\\ \dot{\psi}&=r\end{aligned}\tag{1}$$

水平面动力学模型：

$$\begin{aligned}\dot{u}&=\frac{m_2}{m_1}vr-\frac{d_1}{m_1}u+\frac{1}{m_1}X_{\mathrm{PROP}}\\ \dot{v}&=-\frac{m_1}{m_2}ur-\frac{d_2}{m_2}v\\ \dot{r}&=\frac{m_1-m_2}{m_3}uv-\frac{d_3}{m_2}r+\frac{1}{m_3}\tau_r\end{aligned}\tag{2}$$

其中

$$\begin{aligned}&m_1=m-X_{\dot{u}},m_2=m-Y_{\dot{v}}\\ &m_3=J_z-N_{\dot{r}},d_1=X_u+X_{u|u|}|u|\\ &d_2=Y_v+Y_{v|v|}|v|,d_3=N_r+N_{r|r|}|r|\end{aligned}\tag{3}$$

动态窗口算法中 UUV 未来时域的状态由状态空间模型得到，由式（1）和式（2）转换为 UUV 的水平面状态空间方程：

$$\begin{cases}\dot{x}=f(x)+g(x)u\\ y=h(x)\end{cases}\tag{4}$$

$$x=[u\ v\ r\ x\ y\ \psi]^{\mathrm{T}}=[x_1\ x_2\ x_3\ x_4\ x_5\ x_6]^{\mathrm{T}}$$

$$f(x)=\begin{bmatrix}a_1x_2x_3+a_2x_1\\ b_1x_1x_3+b_2x_2\\ c_1x_1x_2+c_2x_3\\ x_1\cos x_6-x_2\sin x_6\\ x_1\sin x_6+x_2\cos x_6\\ x_3\end{bmatrix}$$

$$g(x)=[g_1(x)\ g_2(x)]=\begin{bmatrix}a_3&0&0&0&0&0\\ 0&0&c_3&0&0&0\end{bmatrix}$$

$$u=[u_1\ \ u_2]^{\mathrm{T}}=[X_{\mathrm{prop}}\ \ \tau_r]^{\mathrm{T}},y=[h_1\ \ h_2\ \ h_3]^{\mathrm{T}}=[x_4\ \ x_5\ \ x_6]^{\mathrm{T}}$$

$$a_1=\frac{m_2}{m_1},a_2=-\frac{d_1}{m_1},a_3=\frac{1}{m_1},b_1=-\frac{m_1}{m_2},b_2=-\frac{d_2}{m_2},$$

$$c_1=\frac{m_1-m_2}{m_3},c_2=-\frac{d_3}{m_3},c_3=\frac{1}{m_3}$$

式中：$x\in R^6,u\in R^2,y\in R^3$，状态空间方程的状态量为 UUV 的位置姿态，输出量为 UUV 的实际位置坐标；x，y 为 UUV 水平面坐标；u，v 为横、纵向速度；ψ，r 为艏向角和角速度；X_{prop}、τ_r 为纵向推力以及转向力矩。

2 路径规划算法

2.1 RRT 全局路径规划算法

在全局路径规划的算法中，快速扩展随机树（Rapidly-exploring Random Trees，RRT）算法由 LaValle 教授在 20 世纪末提出。RRT 算法相对于其他全局算法的优点在于可以更快更灵活地规划出路径，并且不要求障碍物形状的规则性。但是缺点也尤为明显：它的效率并不稳定，由于其迭代过程没有目标性，所以每次的路径选取并不相同。经过类似迷宫的狭窄区域很难找到出口，需要大量反复计算，浪费时间和资源。并且所需路径越平滑，计算量也越大。一些改进的 RRT 算法弥补了部分缺点，且可通过插值修正曲线。

初始化的地图只包含随机树的根节点也就是起始点 Q_{int} 和目标点 Q_{gol}。首先按照策略概率 p 生成一个随机点 Q_{rad}，p 的生成由一个随机数保证。随机树上离该随机点最近的节点 Q_{near}（初始化的情况下为起始点）向 Q_{rad} 矢量方向生成一个新节点 Q_{new}。Q_{near} 和 Q_{new} 的长度为距离参数 q_{dist}，修改此参数可以修改 RRT 算法的精度和计算量。之后对随机树上的节点 Q_{near} 以及 Q_{new} 的路线做障碍物碰撞检测，若发生碰撞则放弃此条路径，若没有发生碰撞则将节点 Q_{new} 加入到随机树节点中，Q_{near} 和 Q_{new} 加入到路径矩阵。重复以上步骤直到 Q_{new} 与 Q_{gol} 之间的距离小于设定值 Q_{set}，即说明到达目标点。最后依据贪心算法简化得到的路径，从起点 Q_{int} 起依次寻找与 Q_{gol} 节点满足碰撞检测函数的节点 q*，用此路径替代之前的路径，再以 q*为终点依次寻找得到一个最简路径。同时需要注意以下几点。

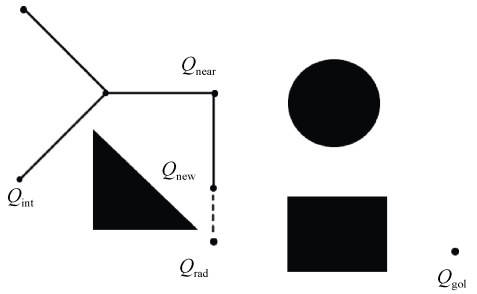

图 1 RRT 算法示意图

Fig. 1 RRT algorithm diagram

1）策略概率 p 依据目标点的距离给定，以一个随机数确保概率的有效性。距离目标点越近生成 Q_{rad} 的概率越高，否则随机树将聚集在一个区域，不能很好地向目标点方向伸展探索空间。p 选取为

$$p=\begin{cases} p_1 & (D_{\text{dist}}(Q_{\text{rad}},Q_{\text{gol}})<q_{lit}) \\ 1-p_1 & (D_{\text{dist}}(Q_{\text{rad}},Q_{\text{gol}})>q_{lit}) \end{cases} \quad (5)$$

2）为了保证算法的效率和可控性，需要设定每次搜索的循环次数，在给定次数之内完成循环，否则放弃此次搜索。以下为 RRT 算法的具体步骤：

①加载全局地图（获取障碍物信息，给定全局路径起点 Q_{int}，终点 Q_{gol}），设置距离参数 q_{dist}，范围参数 Q_{set}；

②设置循环；

③随机生成 Q_{rad}，设置全局终点附近生成 Q_{rad} 的概率 p；

④选取距离 Q_{rad} 最近的随机树节点 Q_{near} 向 Q_{rad} 方向伸展，伸展长度为距离参数 q_{dist}，伸展后得到新节点 Q_{new}；

⑤对 Q_{near} 与 Q_{new} 之间的路径进行碰撞检测，若此路径与障碍物存在交集则舍弃该节点，若不存在交集则将 Q_{new} 加入到随机树节点中；

⑥检测循环次数，若达到循环次数则回到③重新构建随机树，未超出次数则继续扩展随机树；

⑦检测 Q_{new} 与 Q_{gol} 之间的距离是否小于 Q_{set}，若小于说明到达目标点则跳出循环，并输出随机树节点以及路径，若不小于则回到③；

⑧通过碰撞检测简化输出的随机树节点和路径。从起点依次寻找与终点 Q_{gol} 满足未碰撞的节点，找到未碰撞的节点 q*后即舍弃该节点之后的节点，简化此节点到终点的路径，以此节点为终点重复以上过程继续向前寻找未碰撞节点得到最简路径。

2.2 DWA 局部路径规划算法

相对于全局路径规划来说，局部路径规划并不要求对周围环境百分之百的了解。全局路径规划为移动机器人在已知全局地图的前提下规划出从一个区域到另一个区域的具体路线，而局部路径规划的地图多为临时获取，由获得的局部地图规划出一

条临时路线，随着机器人的移动，局部地图实时更新，路线也实时更新。局部路径规划往往和各类传感器协同工作，由传感器获取周围环境的具体信息，进而得到局部地图进行规划[8]。局部路径规划的缺点是受参数以及位置环境物体影响，随机性较高，无法生成一个固定高效的全局路线[9-10]。

动态窗口算法（Dynamic Window Approach，DWA）同样是 20 世纪末提出，其使用的是在线规划的方法：根据机器人本身的运动学模型得到各类约束，利用运动学约束根据当前的状态得出下一拍的动态区域，区域内包括机器人下一拍能实现的全部运动轨迹和状态。常用的移动机器人运动学模型在相邻拍数的轨迹一般用直线代替，用圆弧代替相比于直线更精确一些，而 UUV 的受力十分复杂，需要 UUV 专用的模型。之后通过对区域内的所有轨迹进行评价分析，得到一条得分最高的即为最优轨迹。

DWA 算法的核心主要有 2 部分，第 1 部分为动态窗口的生成，依据以下几点。

1）速度与角速度约束，由 UUV 本身的性能得到最大与最小速度。

$$V_m=\{v\in[v_{\min},v_{\max}],\omega\in[\omega_{\min},\omega_{\max}]\} \tag{6}$$

2）加速度约束，UUV 的加速度受螺旋桨的推力以及水流情况的影响，所能达到的最小与最大加速度，在一拍时间内与最大最小速度和角速度共同组成了一个轨迹范围，生成了一拍时间内的动态窗口。

$$\begin{aligned}V_d=\{(v,\omega),&v\in[v_c-v_b\cdot\Delta t,v_c+v_a\cdot\Delta t],\\&\omega\in[\omega_c-\omega_b\cdot\Delta t,\omega_c+\omega_a\cdot\Delta t]\}\end{aligned} \tag{7}$$

3）碰撞检测，为了使 UUV 避免碰撞到障碍物，相应的距离需要小于对应的速度。

$$V_a=\{(v,\omega),v\le\sqrt{2\cdot D_{\mathrm{dist}(v,\omega)}\cdot v_b},\omega\le\sqrt{2\cdot D_{\mathrm{dist}(v,\omega)}\cdot\omega_b}\} \tag{8}$$

式中：$D_{\mathrm{dist}(v,\omega)}$ 为 UUV 与障碍物之间的轨迹距离，若在一定距离内小于对应速度则该条轨迹保留，否则去除该条轨迹。

第 2 部分为轨迹的评分策略函数 $G(v,\omega)$，根据生成的动态窗口得到若干条轨迹，分析每一条轨迹的效率和安全性，评分主要依据以下几点。

1）艏向 $E_{\mathrm{heading}(v,\omega)}$ 评分，UUV 轨迹的实际航向越接近目标方向，则该评分越高。

2）安全系数 E_{sdist} 评分，UUV 轨迹距离障碍物越远，则评分越高；若该条轨迹距离障碍物小于对应速度的安全距离则直接舍弃；全路路径的安全性首先由此函数检验。

3）效率 $E_{\mathrm{velocity}(v,\omega)}$ 评分，根据 UUV 轨迹的曲率、角加速度以及效率进行评分，越平滑速度越快，评分越高。

由于三类分属于不同的系统，为了防止某一参数占比过高导致不同路径的评分占比差距过大，将三类评价函数统一可以得到 $E_{\mathrm{evaluate}(i)}$。

$$E_{\mathrm{evaluate}(i)}=\frac{E_{\mathrm{evaluate}(i)}}{\sum_{i=1}^{n}E_{\mathrm{evaluate}(i)}} \tag{9}$$

加入可调整的占比得到最终的评分函数。α 为艏向评分占比，β 为安全系数评分占比，γ 为效率评分占比。

$$\begin{aligned}G(v,\omega)=\sigma(&\alpha\cdot N_{\mathrm{heading}(v,\omega)}+\beta\cdot N_{\mathrm{sdist}(v,\omega)}+\\&\gamma\cdot N_{\mathrm{velocity}(v,\omega)})\end{aligned} \tag{10}$$

3　基于 RRT 和 DWA 的改进路径规划算法

针对全局算法和局部算法的优缺点以及 UUV 性能与安全性的需求，提出了 RRT 和 DWA 改进的融合算法。该算法主要有两层规划设计，一层外部框架和一层内部框架。外部框架由 RRT 算法实现，内部框架将由 RRT 算法得到全局路径分段，再由 DWA 在线规划。融合算法的框架如图 2 所示。具体流程如下：

1）首先全局框架加载全局地图，当获取到起始点和目标点信息后规划出一条全局路径。传感器进入待机状态，传感器的范围即是局部框架动态轨迹窗口的范围。

2）将得到的全局路径发送给局部框架，利用 DWA 算法检验全局路径是否满足 UUV 的运动学规律，若不满足则重新生成全局路径。此外，当传感器得到的局部地图中有全局地图中未知的障碍物，并且此障碍物不在安全距离之外时则会发生碰撞，发出碰撞警告。

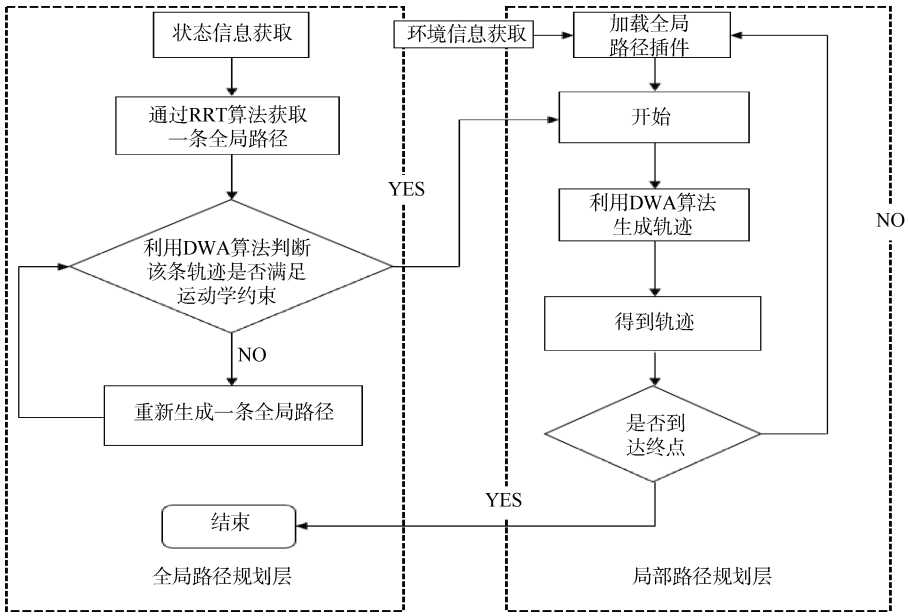

图 2　RRT-DWA 融合算法框架

Fig. 2　RRT-DWA fusion algorithm framework

3）局部框架加载全局路径后，由融合参数 μ 确定全局路径插值，μ 参数即为由 RRT 算法得到的最终全局路径的分段比例，根据路径的长短以及障碍物的密度进行修改，这里取为 0.05。再由 DWA 算法在线生成局部路径，直到到达目标点完成循环，得到修正后的路径。

4　仿真验证分析

针对以上设计的算法，构建 1 个 1 000 m^2 的仿真地图，图中黑色区域为障碍物；目标的起始点为原点，终点为地图右上角。首先给定 3 组参数验证单独使用动态窗口的全局路径规划性能。

图 3 为利用 DWA 算法得到的路径，改变 DWA 算法中评价函数中艏向评分占比，安全系数评分占比以及效率评分占比将得到不同的路径。图中蓝色和绿色两条轨迹的航向角评分分别为 0.5 和 0.45，占比较高，在（300，300）坐标附近时为保证航向角评分路径的生成朝向目标点，UUV 从障碍物上方行驶。下方红色轨迹的安全性评分为 0.6，占比更高，所以在同一位置为保证安全性的评分，远离障碍物，UUV 向下行驶。这就导致了单纯的 DWA 算法受很多因素影响，很难找到一条简洁的全局轨迹。

表 1　动态窗口仿真参数

Table 1　Dynamic window simulation parameters

初始状态	参数组 1	参数组 2	参数组 3
$u=0$	α=0.5	α=0.45	α=0.1
$v=0$	β=0.3	β=0.3	β=0.6
$r=0$ (°)/s，ψ=45°	γ=0.2	γ=0.25	γ=0.3

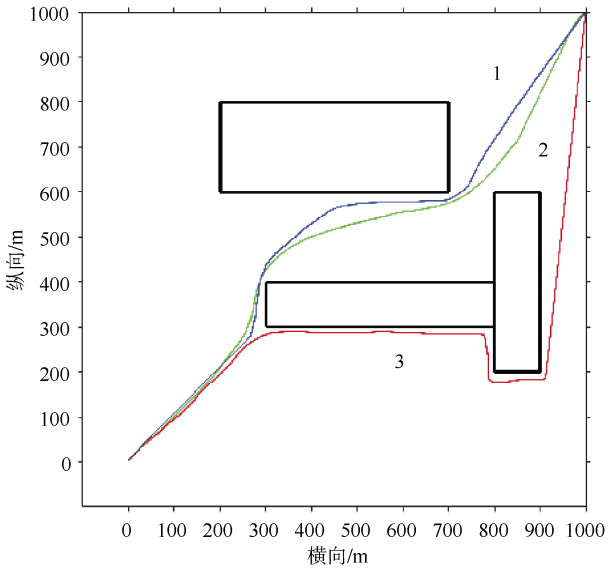

图 3　基于 DWA 算法的局部规划

Fig. 3　Local planning based on DWA algorithm

图 4 和图 5 为利用 RRT 算法得到的全局路径，两次生成的全局路径并不相同，说明该算法有较高的灵活性。由于 RRT 算法得到的全局路径未考虑 UUV 的运动学约束，在图 4 以及图 5 中部分路径与障碍物相对较近，小于 10 m 的安全距离，安全性得不到保证。

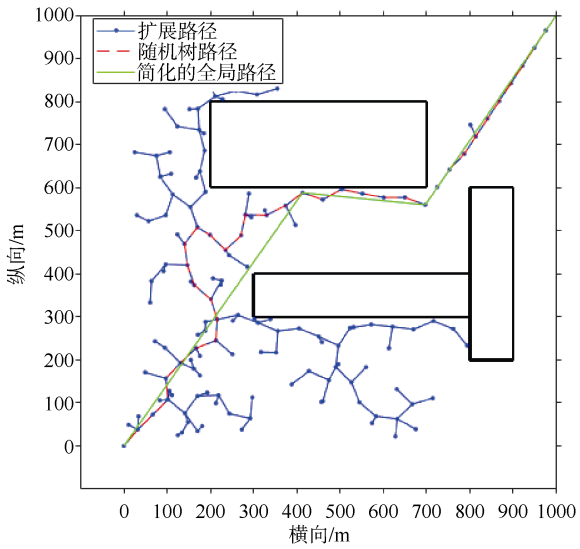

图 4　基于 RRT 算法的全局规划（1）

Fig. 4　Global planning based on RRT（1）

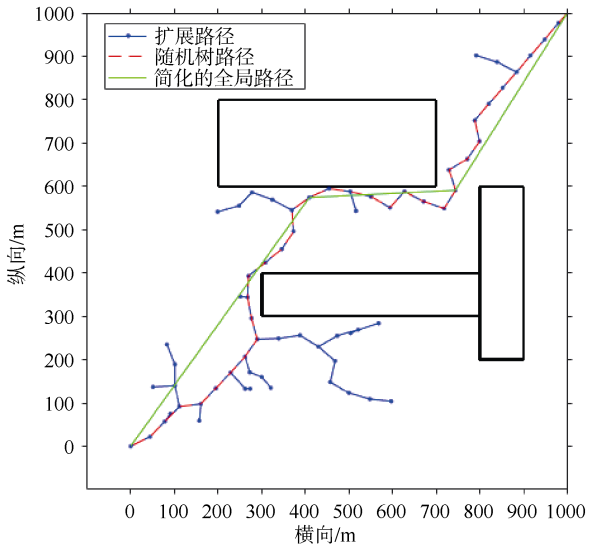

图 5　基于 RRT 算法的全局规划（2）

Fig. 5　Global planning based on RRT（2）

图 6 和图 7 为 RRT-DWA 融合算法生成的路径，其中绿色轨迹为图 4 和图 5 中利用 RRT 得到的全局路径，红色轨迹为 RRT-DWA 融合算法轨迹。由图可知，该条轨迹由于采用了 RRT 算法，其轨迹比较灵活，很容易找到一条简洁的全局路径。同时后加入的 DWA 算法使得该条路径满足 UUV 的运动学规律，可以时刻与障碍物保持距离，保证了安全，继承了 DWA 算法的躲避移动或未知障碍物的能力。

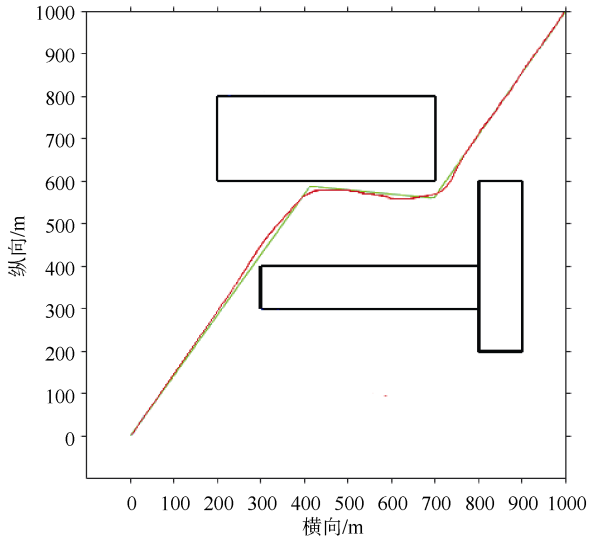

图 6　基于融合算法的全局规划（1）

Fig. 6　Global planning based on RRT-DWA（1）

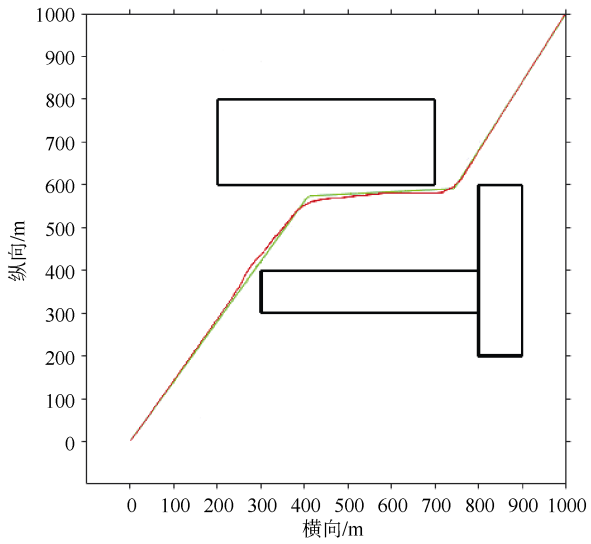

图 7　基于融合算法的全局规划（2）

Fig. 7　Global planning based on RRT-DWA（2）

表 2 中 RRT1 和 RRT-DWA1 分别对应图 4 和图 6 中一段范围内的最近障碍物的距离，RRT2、RRT-DWA2 对应图 5 和图 7 中障碍物距离。由表知 RRT 算法得到的轨迹无法实时保证安全距离；融合算法得到的轨迹可以始终保证 10 m 的安全距离，说明了融合算法具有良好的避障能力。

表 2 最近障碍物距离

Table 2 Nearest obstacle distance

全局路径	RRT1	RRT-DWA1	RRT2	RRT-DWA2
最近障碍物距离/m	8.9	18.6	9.7	16.3

5 结束语

为保证 UUV 水平面规划的效率以及安全性，以 UUV 水平面状态空间方程为研究对象，针对传统局部路径规划和全局路径规划导航方式的优缺点，提出了 RRT-DWA 融合算法。融合算法的外部框架不考虑动力学约束，仿真结果显示有较好的全局路径搜索能力和灵活性；内部框架结合 UUV 的运动模型、动力学模型以及传感器实时获取局部信息，通过评价函数可以实现障碍物的规避。仿真结果显示，融合算法始终可以保持 10 m 的安全距离，保证了安全，通过内部框架判断全局路径的可行性。同时应对不同的海底区域可以通过调整局部框架的评价函数参数来满足安全性或者效率的需求。综上，RRT-DWA 融合算法可以实现以上功能，满足 UUV 现实需求。

参考文献

[1] 丁帅，陈苗苗，王猛，等. 基于 RRT*算法的水下机器人全局路径规划方法[J]. 舰船科学技术，2019，41（17）：66-73.

[2] 孙波，陈卫东，席裕庚. 基于粒子群优化算法的移动机器人全局路径规划[J]. 控制与决策，2005（9）：1052-1055，1060.

[3] 严浙平，邓超，迟冬南，等.双种群粒子群算法及其在 UUV 路径规划中的应用[J]. 计算机工程与应用，2013，49（15）：1-5.

[4] 黄辰. 基于智能优化算法的移动机器人路径规划与定位方法研究[D]. 大连：大连交通大学，2018.

[5] 何壮壮，丁德锐. 基于 D-star 和 DWA 的改进机器人导航方法[J]. 电子测量技术，2019，42（12）：122-128.

[6] 张伟，郁晨曦，滕延斌，等. 基于模型预测控制的 UUV 路径跟踪控制研究[J]. 仪器仪表学报，2017，38（11）：2659-2666.

[7] 严浙平，邓超，赵玉飞，等. 无人水下航行器近海底空间路径规划方法[J]. 哈尔滨工程大学学报，2014，35（3）：307-312.

[8] 李宁. 面向家庭环境的移动机器人局部路径规划算法研究[D]. 哈尔滨：哈尔滨工业大学，2018.

[9] 田永永，李梁华. 基于速度方向判定的动态窗口法[J]. 农业装备与车辆工程，2018，56（8）：39-42.

[10] THANELLAS G A，MOULIANITIS V C，ASPRAGATHOS N A. A spatially wind aware quadcopter（UAV）path planning approach[J]. IFAC PapersOnLine，2019，52（8）：283-288.

（责任编辑：肖楚楚）