基于航海雷达的无人艇路径规划算法研究

胡智焕¹, 杨子恒¹, 刘笑成¹, 张卫东^{1*}, 罗荣²

1. 上海交通大学自动化系, 上海 200240;

2. 海军研究院, 北京 102442

* E-mail: wdzhang@sjtu.edu.cn

收稿日期: 2020-05-24; 接受日期: 2020-08-17; 网络版发表日期: 2021-01-04

国家自然科学基金(批准号: 61473183, U1509211, 61627810)和国家重点研发计划(编号: 2017YFE0128500)资助项目

摘要 海上航行的安全性对于无人船是至关重要的, 这涉及如何在动态海洋环境下规划出可行路径并通过反馈控制引导船舶沿该路径行驶等问题. 本文提出了一种适用于高速船舶的实时路径规划算法, 该算法利用航海雷达实时检测的障碍物位置信息, 可快速计算出一条能有效避开障碍物且光滑可行的路径. 在该方法中, 首先通过引入Frenet坐标系, 将一条路径分解为横向位移和纵向位移, 并分别用多项式曲线描述这两个方向上的位移. 进一步考虑到船体运动的约束以及航行的舒适性, 可计算得到一组可行路径, 即路径线束. 最后从路径线束中选取一条最优且无碰撞风险的路径. 通过实验, 验证了无人船避开静态障碍物的规划算法, 表明了该算法的可行性与实时性.

关键词 路径规划, 无人艇, 避障

1 引言

无人艇是一种安全可靠且无需人为操作的水面舰艇, 其涉及的自主导航技术可使得海洋运输更加安全、高效与便捷. 对于无人艇来说, 运动规划系统是其安全航行的基础. 近几年来, 运动规划的理论和应用等方面已开展了许多研究工作.

总的来说, 运动规划问题可以划分为三个方面^[1,2]. 首先, 航线规划需要考虑气象、管制、时间等因素, 计算出一条合理的路线. 其次, 无人艇的行为规划算法在遵循《国际海上避碰规则》(COLREGS)^[3]的同时, 确定其局部的航行方案. 最后, 如图1所示, 路径规划模

块计算出一条连续的运动轨迹, 从而实现避碰、速度保持和停止等功能. 本文侧重于研究无人艇的路径规划算法, 并不讨论航线规划或行为规划, 因而不考虑《国际海上避碰规则》.

近年来, 新的路径规划算法不断出现和发展, 其难点在于如何获得一条满足环境约束的轨迹. 该轨迹可以定义为一个时变函数 $\pi(t): [0, \Delta T] \rightarrow \Gamma$, 其中 Γ 代表了船体的构型空间, ΔT 表示规划周期. 为了计算最优路径, 最直接的方法是建立一个非凸优化模型, 并运用优化求解器得到最优解. 然而, 由于非凸问题的可行域可能存在多个局部最优点, 使得求解全局最优解的算法复杂度非常高^[4]. 为了在合理时间内得到可行的

引用格式: 胡智焕, 杨子恒, 刘笑成, 等. 基于航海雷达的无人艇路径规划算法研究. 中国科学: 技术科学, 2021, 51: 1401–1409

Hu Z H, Yang Z H, Liu X C, et al. Radar-based maritime path planning with static obstacles in a Frenet frame (in Chinese). Sci Sin Tech, 2021, 51: 1401–1409, doi: 10.1360/SST-2020-0212

图1 (网络版彩图)无人艇运动规划问题的基本框架, 其包括了全局规划、行为规划和路径规划

Figure 1 (Color online) Motion planning architecture for ASVs. It consists of global planning, behavioral layers and local path planning.

轨迹, 研究人员通常采用启发式算法来获得次优解. 典型的启发式算法, 如几何方法, 将障碍物和船只简化为凸体, 同时考虑运动学约束, 从而计算出可行轨迹的上下边界. 尤其对于二维平面的情况($\Gamma \subseteq \mathbb{R}^2$), 障碍物可以简化为一个正多边形, 大大提高计算效率. 在几何方法中具有代表性的有广义沃罗诺伊图法(generalized Voronoi diagrams)^[5]、速度障碍物法(velocity obstacles method)^[6]、人工势场法(artificial potential fields)^[7]和动态虚拟船舶法(dynamic virtual ship)^[8,9]. 这些算法已经被用于无人艇的海上测试, 且考虑了《国际海上避碰规则》. 然而由于速度障碍物法仅仅考虑障碍物的速度, 当遭遇复杂环境, 如避让多个障碍物时, 该算法容易陷入局部最优解或者无解, 因此不适合交通相对复杂的海上环境.

为了使得无人艇从A点到达B点, 另一种思路是使用占据栅格地图(occupancy grid map)表示A和B之间的状态空间, 并用图搜索法计算得到合理的路径. 具有代表性的算法如戴克斯特拉算法(Dijkstra algorithm)、A*算法、D*算法及其衍生算法, 已经被成功用于轮式机器人^[10]、无人车^[11]和无人艇^[12]的研究中. 为了找到一条由初始状态到达目标状态的最优路径, 该类方法通常将图中的顶点和边分别表示无人艇的运动状态和状态之间的转换. 例如, 在文献[13,14]中, 一种改进的A*算法采用了T型邻域搜索, 并且计算该邻域时候考虑了船体的物理约束. 对于速度相对较低的情形(如自动泊船^[15]或者穿梭游轮), 也可采用基于沃罗诺伊(Voronoi)势场的混合A*算法计算其最优路径. 然而, 图搜

索算法的复杂度也相对较高, 也不适合实时性要求高的高速无人艇应用.

曲线插值法, 采用参数化的几何曲线如线段圆弧、多项式曲线或者样条曲线等, 来拟合一系列航点, 从而进行轨迹规划. 运用前进速度和转弯速率的线性组合, 动态窗口法(dynamic window)^[16]可快速地计算出可行路径. 进一步, 改进的动态窗口法^[17]先根据加速度的分段线性组合构建出一条航行路径, 然后通过视线法和模型预测控制器对期望轨迹进行跟踪. 然而, 线段或者圆弧的组合不能保证路径足够光滑, 从而降低乘客舒适性.

本文对高速无人艇的路径规划问题进行了研究, 提出了一种基于样条曲线的轨迹生成方法. 生成的轨迹满足无人艇运动学和动力学约束, 保证该轨迹对推进器来说是可行的. 在优化指标中引入了加加速度的积分项, 用于提高乘客的舒适性. 同时满足算法实时性的要求, 保证50 Hz的更新速率, 从而适用于高速船舶. 将航海雷达的目标跟踪结果作为障碍物信息输入, 该算法也可有效避开水上障碍物.

2 理论模型

2.1 无人艇运动数学模型

为了描述船体的运动, 控制模块引入大地坐标系和附体坐标系, 如图2所示. 三自由度模型可用于描述船舶的平面运动:

$$\dot{\boldsymbol{\eta}}(t) = \mathbf{T}(\boldsymbol{\eta}(t)) \cdot \mathbf{v}(t), \quad (1)$$

$$\mathbf{M} \cdot \dot{\mathbf{v}}(t) + \mathbf{D} \cdot \mathbf{v}(t) = \boldsymbol{\tau}(t) + \boldsymbol{\tau}_e, \quad (2)$$

$$\mathbf{T}(\boldsymbol{\eta}(t)) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (3)$$

其中 $\boldsymbol{\eta}(t) = [x, y, \psi]^T$ 表示大地坐标系下的位置, $\mathbf{v}(t) = [u, v, r]^T$ 表示附体坐标系下的速度矢量. 我们假设质量矩阵 $\mathbf{M}$ 和阻尼矩阵 $\mathbf{D}$ 都是常数.

经过离散化, 可获得离散的状态空间模型:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A} \cdot \mathbf{x}(k) + \mathbf{B} \cdot \boldsymbol{\tau}(k), \\ \mathbf{A} &= \mathbf{I} + \Delta t \cdot \mathbf{A}_c, \quad \mathbf{B} = \Delta t \cdot \mathbf{B}_c, \\ \mathbf{A}_c &= \begin{bmatrix} 0 & \mathbf{T}(\boldsymbol{\eta}(t)) \\ 0 & -\mathbf{M}^{-1}\mathbf{D} \end{bmatrix}, \quad \mathbf{B}_c = \begin{bmatrix} 0 \\ \mathbf{M}^{-1} \end{bmatrix}, \end{aligned} \quad (4)$$

图2 (网络版彩图)大地坐标系和附体坐标系

Figure 2 (Color online) Global coordinate system and body-fixed coordinate system.

其中 $\mathbf{x}(t) = [x, y, \psi, u, v, r]^T$ 表示状态向量, $\tau(t) = [\tau_x, \tau_y, \tau_N]^T$ 代表推进器产生的推力, $\mathbf{I}$ 表示单位矩阵, Δt 表示控制器的刷新时间. 以上离散化的状态空间模型可作为卡尔曼滤波的依据^[18]. 同时, 采用推力分配算法将控制指令有效、合理地分配至各个推进器, 从而提高推力的控制精度.

2.2 Frenet坐标系

无人船的轨迹跟踪和规划算法都要求船体在规定的时间内, 达到期望的状态(如位置、首向 ψ 、速度 v 、曲率 κ 和加速度 a). 我们称这样一条随时间变化的状态曲线为参照线, 并且用样条曲线或者欧拉螺线等函数去定义参照线. 然而, 轨迹规划和轨迹跟踪的差异在于, 当无人船需要避开障碍物时, 无法按照预先给定的参照线航行, 从而需要重新规划一条合理的轨迹.

本文引入了Frenet坐标系, 如图3所示, 用于描述横向 $d(t)$ 和纵向 $s(t)$ 位移. 具体来说, $d(t)$ 表示船体当前位置与参照线的垂直偏差, $s(t)$ 代表船体在参照线上所行驶过的弧长. 因此, 生成的轨迹点 $\mathbf{x}(s(t), d(t))$ 就能用Frenet坐标系下的参数表示:

$$\mathbf{x}(s(t), d(t)) = \mathbf{r}(s(t)) + d(t) \cdot \mathbf{n}_r(s(t)), \quad (5)$$

其中向量 $\mathbf{n}_r, \mathbf{t}_r$ 分别表示参照线上投影点的法向量和切向量, $\mathbf{r}(s(t))$ 表示该投影点的笛卡儿坐标, 而 $\mathbf{n}_x, \mathbf{t}_x$ 代表生成轨迹点的法向量和切向量.

图3 (网络版彩图)基于参照线的Frenet坐标系

Figure 3 (Color online) Trajectory generation based on the reference line in the Frenet frame.

2.3 坐标转换

对于路径规划问题, Frenet坐标系比笛卡儿坐标系更直观、方便. 同时, 无人船运动轨迹的实时规划也需要两个坐标系之间的转换. 我们定义关于时间的偏导数 $(\dot{\cdot}) := \frac{\partial}{\partial t}(\cdot)$ 和关于弧长的偏导数 $(\cdot)' := \frac{\partial}{\partial s}(\cdot)$, 并给出Frenet坐标系下的运动状态 $[s, \dot{s}, \ddot{s}; d, \dot{d}, \ddot{d}]$ 或 $[s, \dot{s}, \ddot{s}; d, d', d'']$, 以及笛卡儿坐标系下的状态 $[x, \psi_x, \kappa_x, v_x, a_x]$. 本文假设船体的位置不会出现偏离参照线很远的情况, 即 $1 - \kappa_r d > 0$ 以及 $|\Delta\psi| < \pi/2$, 其中 $\Delta\psi := \psi_x - \psi_r$.

应用Frenet-Serret公式^[19],

$$\frac{dn_r}{ds} = -\kappa_r t_r, \quad \frac{dn_r}{dt} = -\dot{s} \kappa_r t_r, \quad (6)$$

$$\frac{dt_r}{ds} = \kappa_r n_r, \quad \frac{dt_r}{dt} = \dot{s} \kappa_r n_r, \quad (7)$$

其中 κ_r 代表参照线的曲率.

$$\mathbf{t}_r(s) = (\cos\psi_r(s), \sin\psi_r(s))^T,$$

$$\mathbf{n}_r(s) = (-\sin\psi_r(s), \cos\psi_r(s))^T.$$

将式(5)与Frenet-Serret公式结合, 可得

$$d = [\mathbf{x} - \mathbf{r}(s)]^T \cdot \mathbf{n}_r, \quad (8)$$

$$\dot{d} = v_x \sin\Delta\psi. \quad (9)$$

将位移 x 对时间求导可得到速度 v_x :

$$v_x = \|\dot{x}\|_2 = \sqrt{(1 - \kappa_r d)^2 \dot{s}^2 + \dot{d}^2},$$

其中 $\|\cdot\|_2$ 代表欧几里德范数(Euclidean norm). 利用速度 v_x , 可得

$$\begin{aligned} d' &= \frac{\dot{d}}{\dot{s}} = \frac{v_x \sin \Delta \psi}{\dot{s}} = \sin \Delta \psi \cdot \sqrt{(1 - \kappa_r d)^2 + \dot{d}^2} \\ \Rightarrow d' &= (1 - \kappa_r d) \tan \Delta \psi. \end{aligned} \quad (10)$$

将 $(x - r(s))^T t_r = 0$ 对时间求导, 可得

$$\dot{s} = \frac{v_x \cos \Delta \psi}{1 - \kappa_r d}. \quad (11)$$

生成轨迹 x 的弧长 s_x , 定义其曲率 $\kappa_x = d\psi_x / ds_x$, 可得

$$\Delta \psi' = \frac{d(\psi_x - \psi_r)}{ds} = \frac{1 - \kappa_r d}{\cos \Delta \psi} \kappa_x - \kappa_r.$$

将式(11)对弧长 s 求导可得

$$d'' = -(\kappa_r d)' \tan \Delta \psi + \frac{1 - \kappa_r d}{\cos^2(\Delta \psi)} \Delta \psi'. \quad (12)$$

轨迹点的加速度

$$\begin{aligned} a_x = \dot{v}_x &= \dot{s} \frac{1 - \kappa_r d}{\cos \Delta \psi} + \frac{\dot{s}^2}{\cos \Delta \psi} \left[d' \Delta \psi' - (\kappa_r d)' \right], \\ \ddot{s} &= \frac{a_x \cos \Delta \psi - \dot{s}^2 \left[d' \Delta \psi' - (\kappa_r d)' \right]}{1 - \kappa_r d}. \end{aligned} \quad (13)$$

横向偏移 d 的时间导数:

$$\dot{d} = \frac{ds}{dt} \frac{d}{ds} d = \dot{s} d', \quad (14)$$

$$\ddot{d} = d'' \dot{s}^2 + d' \ddot{s}. \quad (15)$$

2.3.1 从笛卡儿变换到Frenet坐标系

对于 t_0 时刻的船体运动 $[x, \psi_x, \kappa_x, v_x, a_x](t_0)$, 需要将其投影到参考线上并计算出对应的状态 $[s_0, \dot{s}_0, \ddot{s}_0; d_0, \dot{d}_0, \ddot{d}_0]$ 或者 $[s_0, \dot{s}_0, \ddot{s}_0; d_0, d'_0, d''_0]$. 例如, 当船体位于参考线的起点时, 可知Frenet坐标系中的位移 $s_0 = 0, \dot{s}_0 = 0$. 首先通过简单的数值方法确定投影点的弧长 s_0 :

$$s_0 = \arg \min \|x - r(s)\|_2.$$

由于参照线的函数已知, 结合 s_0 可计算出高阶的状态,

包括曲率 κ_r, κ_r' 和首向角 ψ_r . 进一步通过式(8)~(15)和 $\Delta \psi = \psi_x - \psi_r$, 可获得剩下的变量, 如

$$\dot{s}_0, \ddot{s}_0, d_0, \dot{d}_0, \ddot{d}_0, d'_0, d''_0.$$

2.3.2 从Frenet变换到笛卡儿坐标系

同样利用式(8)~(15), 将Frenet坐标系下的状态线束转换为笛卡儿坐标系下的状态 $[x, \psi_x, \kappa_x, v_x, a_x]$.

2.4 最优轨迹生成

为了充分搜索无人艇的构型空间, 规划系统将生成候选轨迹集合, 即轨迹线束, 并从中选取最优的一条轨迹. 基于参照线的候选轨迹集合, 综合航行舒适性、船体动力学约束以及避障条件而计算得到.

首先引入加加速度量化描述航行舒适性, 且通过定理1, 我们能够确定生成轨迹的函数形式.

定理1: 假设 t_0 时刻的初始状态 $P_0 = [p_0, \dot{p}_0, \ddot{p}_0]$, $t_1 = t_0 + \Delta T$ 时刻的状态 $P_1 = [\dot{p}_1, \ddot{p}_1]$, 用齐次多项式连接两点的状态可使以下的损失函数最小.

$$C = k_f J_t + k_g g(\Delta T) + k_h h(p_1),$$

$$J_t(p(t)) := \int_{t_0}^{t_1} \ddot{p}^2(\tau) d\tau,$$

其中, $g(\cdot)$ 和 $h(\cdot)$ 是任意函数, 且 $k_f, k_g, k_h > 0$.

上述理论的证明可参见文献[20]. 其中 $\ddot{p}(t)$ 可定义为加加速度, $J_t(p(t))$ 表示 ΔT 时间段内的加加速度的积分, 从而描述乘客的舒适性. 文献[21]证明了, 在Frenet坐标系中, 高速航行的系统, 其横向 $d(t)$ 和纵向 $s(t)$ 的轨迹可分开考虑. 在单独规划出各个方向上的轨迹, 合成为候选轨迹.

2.4.1 横向运动规划

对于初始横向状态 $D_0 = [d_0, \dot{d}_0, \ddot{d}_0]$, 定义惩罚项 $g(\Delta T) = \Delta T$ 和 $h(d_1) = d_1^2$, 则损失函数为

$$C_{\text{lat}} = k_f J_t(d(t)) + k_t \Delta T + k_d d_1^2, \quad (16)$$

该损失函数能够加快横向误差的收敛以及限制过大的横向偏移. 同时, 理想的运动应该沿着参照线或者平行于参照线切线, 使得终止时刻 $t_0 + \Delta T$ 的横向状态量 $\dot{d}_1 = \ddot{d}_1 = 0$. 由定理1可知, 五次多项式是损失函数(16)

的最优解:

$$d(t) = a_5 t^5 + a_4 t^4 + a_3 t^3 + a_2 t^2 + a_1 t + a_0.$$

横向的候选曲线集合包含了用不同终止状态 d_1 和 ΔT 生成的航行轨迹, 经过碰撞安全性分析后, 可保留无碰撞风险的曲线集合 Π_{lat} .

$$[d_1, \dot{d}_1, \ddot{d}_1, \Delta T]_{ik} = [d_i, 0, 0, \Delta T_k].$$

2.4.2 纵向运动规划

针对无人艇系统的匀速航行的工作模式 $\dot{s}_d = \text{const}$, 四次多项式

$$s(t) = a_4 t^4 + a_3 t^3 + a_2 t^2 + a_1 t + a_0$$

可使得以下损失函数达到最优解:

$$C_{lon} = k_f J_f(s(t)) + k_t \Delta T + k_s (s_1 - \dot{s}_d)^2,$$

其中纵向初始时刻 t_0 的状态为 $S_0 = [s_0, \dot{s}_0, \ddot{s}_0]$, 而终止时刻 $t_1 = t_0 + \Delta T$ 的状态为 $S_1 = [s_1, \dot{s}_1]$. 对于匀速运动, 加速度即为零 $\ddot{s}_1 = 0$. 通过微调终止时刻的速度 $\dot{s}_1$:

$$[\dot{s}_1, \ddot{s}_1, \Delta T]_{jk} = [\dot{s}_d + \Delta \dot{s}_j, 0, \Delta T_k],$$

可得纵向的候选曲线集合 Π_{lon} .

2.4.3 合成的轨迹

将横向和纵向轨迹集合作笛卡儿积 $\Pi_{lat} \times \Pi_{lon}$, 且

通过简单的代数运算获得加权损失函数:

$$C_{tot} = k_{lat} C_{lat} + k_{lon} C_{lon},$$

其中 $k_{lat}, k_{lon} > 0$, 可以得到最终的候选轨迹集合, 如图4所示.

2.5 算法框架

本文提出的路径规划算法, 将笛卡儿坐标系下的运动作为输入, 并输出预期的运动状态, 如图5所示. 对于规划算法内部, 首先将船体运动状态被投影到

图4 (网络版彩图)最优轨迹生成

Figure 4 (Color online) Optimal trajectory generation.

图5 (网络版彩图)算法框架。“Cart2Frenet”表示从笛卡儿到Frenet坐标系的转换, “Frenet2Cart”则表示逆过程

Figure 5 (Color online) Algorithm overview. “Cart2Frenet” means the coordinate transformation from Cartesian to Frenet, and “Frenet2Cart” means the coordinate transformation from Frenet to Cartesian.

Frenet坐标系,可计算出候选轨迹集合;进而排除不满足动力学约束的物理量(如线速度过大、加速度过大或者曲率过大)以及有碰撞风险的轨迹;最后找出损失函数 C_{tot} 最小的一条轨迹,即可规划出一条合理的路径,从而得期望的速度和首向角.将其作为比例-积分-微分(简称PID)控制器的输入,可计算出所需推力.

3 分析与讨论

为了验证路径规划算法,本文选用了一艘双体船作为实验平台(图6).航海雷达用于探测障碍物的位置,其天线转速为 24 r min^{-1} ;全球定位系统传感器(GPS)可以提供水平定位精度为 0.5 m ,首向精度为 0.75° 的导航定位功能,其采样频率为 10 Hz ;两个固定式螺旋桨分别位于两个浮体的后方,每个螺旋桨最大功率为 3.5 kW ;控制箱以及 24 V 直流锂电池位于船舱内.表1列出了双体船的几何参数.

图6 (网络版彩图)试验模型. (a) 试验无人艇; (b) 避障试验
Figure 6 (Color online) Sketch of the experimental system. (a) ASV; (b) collision avoidance test.

3.1 算法实现

无人艇的感知、规划和控制模块均由C++程序实现,包括了航海雷达的目标跟踪、路径规划、运动状态估计以及反馈控制器等算法.该程序基于多线程的框架开发,并运行在一台Intel i5, CPU主频为 2.9 GHz 的电脑上,且操作系统是Ubuntu 18.04.

3.2 路径跟踪实验

对于无人艇来说,路径跟踪是验证其控制系统性能的典型应用.在路径跟踪试验中,由用户设定若干个目标点;在每两个目标点之间,无人艇需要沿两点连线运动,并依次通过所有的目标点.如图7所示,任意给定了4个目标点,比较无人艇的实际位置和期望路径可以发现,跟踪控制系统使得船体实际位置逐渐收敛于期望路径.由图8可知,无人艇的实际速度也趋向于期望的前进速度.

3.3 避障实验

该实验验证规划算法在遇到静态障碍物时的避障

表1 双体船几何参数

Table 1 Vehicle specifications

参数		数值
长度 (m)		3.10
宽度 (m)		1.60
重量 (kg)		244.0
重心	X (m)	0.68
	Y (m)	0.00

图7 (网络版彩图)轨迹跟踪试验结果中的无人艇平面轨迹
Figure 7 (Color online) Planar trajectory for the path following test.

图 8 (网络版彩图) 轨迹跟踪试验结果中的状态变化。从上到下依次为前进速度、首向角和转弯速率

Figure 8 (Color online) Time series of vehicle motion in the path following test. The forward speed, heading and rate of turning are presented in each figure.

能力。在试验中, 给定的参照线是一条直线(图9), 船体一开始沿着直线运动。当检测到前方的障碍物时, 重新规划路径并避开障碍物, 平稳地完成避障。并在绕过障碍物之后, 重新返回参照线, 船的前行速度几乎保持不变(图10)。

4 结论

本文介绍了一种基于航海雷达的路径规划算法。该算法将平面轨迹规划问题简化为两个一维规划问题, 并采用高阶多项式曲线对无人艇轨迹进行建模; 同时生成的轨迹考虑了船体动力学约束、航行舒适性以及避障条件。路径跟踪和避障试验证明, 规划算法在避开静态障碍物的前提下, 能够实现平顺稳定的航行。

图 9 (网络版彩图) 避障试验结果中的无人艇平面轨迹。三角代表障碍物

Figure 9 (Color online) Planar trajectory in collision avoidance test. The triangle indicates the obstacle.

图 10 (网络版彩图) 避障试验结果中的无人艇状态。从上到下分别为前进速度和首向角

Figure 10 (Color online) Time series of vehicle motion in collision avoidance test. The forward speed and heading are presented in each figure.

参考文献

- Gonzalez D, Perez J, Milanes V, et al. A review of motion planning techniques for automated vehicles. *IEEE Trans Intell Transp Syst*, 2016, 17: 1135–1145
- Ma X F, Han W, Xie Y L, et al. Analysis and research on mission planning system of USV (in Chinese). *Ship Sci Tech*, 2019, 41: 54–57 [马向峰, 韩玮, 谢杨柳, 等. 水面无人艇任务规划系统分析. *舰船科学技术*, 2019, 41: 54–57]
- Cockcroft A N, Lameijer J N F. A Guide to the Collision Avoidance Rules. Oxford: Butterworth-Heinemann, 2012
- Reif J, Wang H. The complexity of the two dimensional curvature-constrained shortest-path problem. In: *Proceedings of the 3rd Workshop Algorithmic Found Robot*. Rice University Houston, 1998. 49–57
- Takahashi O, Schilling R J. Motion planning in a plane using generalized Voronoi diagrams. *IEEE Trans Robot Automat*, 1989, 5: 143–150
- Kuwata Y, Wolf M T, Zrazhitsky D, et al. Safe maritime autonomous navigation with COLREGS using velocity obstacles. *IEEE J Ocean Eng*, 2013, 39: 110–119

- 7 Lyu H, Yin Y. COLREGS-constrained real-time path planning for autonomous ships using modified artificial potential fields. *J Navigation*, 2019, 72: 588–608
- 8 Zhang G, Zhang C, Zhang X, et al. ESO-based path following control for underactuated vehicles with the safety prediction obstacle avoidance mechanism. *Ocean Eng*, 2019, 188: 106259
- 9 Zhang G, Huang C, Li J, et al. Constrained coordinated path-following control for underactuated surface vessels with the disturbance rejection mechanism. *Ocean Eng*, 2020, 196: 106725
- 10 Howard T M, Kelly A. Optimal rough terrain trajectory generation for wheeled mobile robots. *Int J Robotics Res*, 2007, 26: 141–166
- 11 Dolgov D, Thrun S, Montemerlo M, et al. Practical search techniques in path planning for autonomous driving. *Ann Arbor*, 2008, 1001: 18–80
- 12 Blaich M, Rosenfelder M, Schuster M, et al. Fast grid based collision avoidance for vessels using A* search algorithm. In: *Proceedings of the 17th International Conference on Methods & Models in Automation & Robotics. Międzyzdroje*, 2012. 385–390
- 13 Schuster M, Blaich M, Reuter J. Collision avoidance for vessels using a low-cost radar sensor. *IFAC Proc Volumes*, 2014, 47: 9673–9678
- 14 Blaich M, Rosenfelder M, Schuster M, et al. Extended grid based collision avoidance considering COLREGs for vessels. *IFAC Proc Volumes*, 2012, 45: 416–421
- 15 Li Y, Jia Z H, Zhang W B, et al. Layered trajectory planning and experiment for the autoberthing of unmanned surface vehicles (in Chinese). *J Harbin Eng Univer*, 2019, 40: 1043–1050 [李晔, 贾知浩, 张伟斌, 等. 面向无人艇自主靠泊的分层轨迹规划与试验. 哈尔滨工程大学学报, 2019, 40: 1043–1050]
- 16 Wilthil E F, Flåten A L, Brekke E F, et al. Radar-based maritime collision avoidance using dynamic window. In: *IEEE Aerospace Conference. Big Sky*, 2018: 1–9
- 17 Eriksen B H, Breivik M, Wilthil E F, et al. The branching-course model predictive control algorithm for maritime collision avoidance. *J Field Robotics*, 2019, 36: 1222–1249
- 18 Welch G, Bishop G. An introduction to the Kalman filter. *Proc ACM SIGGRAPH*, 2001
- 19 Kühnel W. *Differential Geometry*. American Mathematical Society, 2015. 77
- 20 Takahashi A, Hongo T, Ninomiya Y, et al. Local path planning and motion control for Agv in positioning. In: *IEEE/RSJ International Workshop on Intelligent Robots and Systems. Tsukuba*, 1989. 392–397
- 21 Werling M, Ziegler J, Kammel S, et al. Optimal trajectory generation for dynamic street scenarios in a frenet frame. In: *IEEE International Conference on Robotics and Automation. Anchorage*, 2010. 987–993

Radar-based maritime path planning with static obstacles in a Frenet frame

HU ZhiHuan¹, YANG ZiHeng¹, LIU XiaoCheng¹, ZHANG WeiDong¹ & LUO Rong²

¹ *Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China;*

² *Naval Research Academy, Beijing 102442, China*

Navigation safety at sea is important for autonomous surface vehicles, and involves the problems of motion planning in dynamic environments and their robust tracking through feedback control. We present a practical path planning method that generates smooth trajectories for a high-speed marine vehicle traveling in an unknown environment, where obstacles are detected in real time by marine radar. Our approach uses the polynomial curve to describe lateral and longitudinal trajectories in the Frenet frame, known as motion primitives, whose combination ensures that the planning area is properly covered. In addition, we can select a feasible, optimal, and collision-free trajectory from such a set of motion primitives that is generated by considering the vehicle dynamics and comfort. The effectiveness of the proposed algorithm is demonstrated in an experiment with static and dynamic obstacles.

path planning, autonomous surface vehicles, collision avoidance

doi: [10.1360/SST-2020-0212](https://doi.org/10.1360/SST-2020-0212)