

上海交通大学硕士学位论文

上海交通大学学位论文 L^AT_EX 模板示例文档

硕 士 研 究 生：某 某

学 号：0010900990

导 师：某某教授

申 请 学 位：工学硕士

学 科：某某专业

所 在 单 位：某某系

答 辩 日 期：2014 年 12 月 17 日

授予学位单位：上海交通大学

Dissertation Submitted to Shanghai Jiao Tong University
for the Degree of Master

**A SAMPLE DOCUMENT FOR
L^AT_EX-BASED SJTU THESIS TEMPLATE**

Candidate:	Mo Mo
Student ID:	0010900990
Supervisor:	Prof. Mou Mou
Academic Degree Applied for:	Master of Engineering
Speciality:	A Very Important Major
Affiliation:	Depart of XXX
Date of Defence:	Dec. 17th, 2014
Degree-Conferring-Institution:	Shanghai Jiao Tong University

上海交通大学 学位论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：某某

日期：某某年某某月某某日

上海交通大学

学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定,同意学校保留并向国家有关部门或机构送交论文的复印件和电子版,允许论文被查阅和借阅。本人授权上海交通大学可以将本学位论文的全部或部分内容编入有关数据库进行检索,可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保 密 ☐, 在 _____ 年解密后适用本授权书。

本学位论文属于

不保密 ☐。

(请在以上方框内打“√”)

学位论文作者签名:

指导教师签名:

日期: 年 月 日 日期: 年 月 日

上海交通大学学位论文 L^AT_EX 模板示例文档

摘 要

横洋科技股份有限公司是由上海交通大学博士团队创立的一家海上智能系统研发企业，致力于运用人工智能和大数据技术重构海上无人系统，服务军民两用，为实现国家海洋强国梦贡献智慧力量，为打造“中国智造”的世界名片谱写辉煌。

摘要页的下方注明本文的关键词（4~6个）。

关键词： 上海交大, 饮水思源, 爱国荣校

A SAMPLE DOCUMENT FOR L^AT_EX-BASED SJTU THESIS TEMPLATE

ABSTRACT

Shanghai Jiao Tong University (SJTU) is a key university in China. SJTU was founded in 1896. It is one of the oldest universities in China. The University has nurtured large numbers of outstanding figures include JIANG Zemin, DING Guangen, QIAN Xuesen, Wu Wenjun, WANG An, etc.

SJTU has beautiful campuses, Bao Zhaolong Library, Various laboratories. It has been actively involved in international academic exchange programs. It is the center of CERNET in east China region, through computer networks, SJTU has faster and closer connection with the world.

KEY WORDS: SJTU, master thesis, XeTeX/LaTeX template

目 录

第一章 船体运动模型	1
1.1 坐标系	1
1.1.1 北东地坐标系	1
1.1.2 笛卡尔坐标系	1
1.1.3 随体坐标系	1
1.1.4 运动方程	1
1.2 海洋环境载荷	1
1.3 三自由度操纵性方程	3
1.4 三自由度动力定位方程	4
1.5 单自由度首向控制方程	4
1.6 四自由度横摇补偿操纵性方程	4
1.7 参数辨识	4
1.7.1 质量	5
第二章 控制器	7
2.1 推力分配	7
2.1.1 推进器简介	7
2.1.2 推力分配模型	8
2.1.3 求解器	11
2.1.4 欠驱动	13
2.1.5 全驱动船	17
2.2 PID 控制器	18
2.2.1 动力定位	19
2.3 轨迹跟踪	19
2.3.1 视线跟踪法	19
2.4 参数辨识	19
第三章 观测器	21
3.1 移动平均滤波	21
3.2 卡尔曼滤波	21

3.2.1	概述	21
3.2.2	模型	21
3.2.3	算法	22
3.2.4	经验	24
3.3	GPS 天线修正	24
第四章	决策与规划	25
4.1	路径规划	25
4.1.1	航道保持	25
4.1.2	低速泊船	32
4.2	航线规划	38
4.3	行为规划	38
第五章	环境感知	39
5.1	航海雷达	39
5.1.1	SDK	39
5.1.2	目标跟踪	39
5.2	计算机视觉	41
5.3	激光雷达	41
第六章	传感器	43
6.1	全球定位系统	43
6.1.1	GPS 定位定向天线 Hemisphere V102	43
6.1.2	墨卡托投影	44
6.2	惯性测量单元	45
6.3	航海雷达	45
6.4	红外热像仪	45
6.5	风速风向仪	45
第七章	系统架构	47
7.1	软件系统架构	47
7.2	电气系统架构	47
7.2.1	SDK	47

第八章 数学与引用文献的标注	49
8.1 数学	49
8.1.1 数字和单位	49
8.1.2 数学符号和公式	49
8.1.3 定理环境	50
8.2 引用文献的标注	51
全文总结	53
附录 A Maxwell Equations	55
附录 B 绘制流程图	57
参考文献	59
致 谢	63

插图索引

图 1-1 笛卡尔坐标系	2
图 1-2 在随体坐标系中, 船体六自由度的速度 u, v, w, p, q, r	2
图 1-3 转动惯量的测量原理	5
图 2-1 常见船舶推进器示意图	7
图 2-2 螺旋桨角度的定义	8
图 2-3 固定式双尾推螺旋桨的推力曲线	16
图 2-4 固定式双尾推螺旋桨的推力分配	17
图 2-5 基于视线的轨迹跟踪算法	20
图 3-1 卡尔曼滤波的示意图 (来自维基百科).	23
图 4-1 运动规划的整体框架	25
图 4-2 Frenet 坐标系下的轨迹生成	27
图 4-3 Frenet 坐标系下的最优轨迹线束	30
图 4-4 Frenet 坐标系下的最优轨迹生成算法逻辑图	32
图 4-5 Reeds-Shepp 曲线	33
图 4-6 混合 A* 搜索算法中的可行操作模式	35
图 4-7 混合 A* 搜索算法	35
图 4-8 轨迹平滑处理	36
图 5-1 雷达线束的坐标投影	40
图 6-1 UTM 区域的重叠	45
图 B-1 绘制流程图效果	57

表格索引

表 1-1 描述船体运动的常用符号	3
表 2-1 Mosek 中变量约束 (RT 代表实时变化量)	11
表 2-2 Mosek 中线性约束 (RT 代表实时变化量)	12
表 2-3 Mosek 中目标函数 (RT 代表实时变化量)	12
表 2-4 OSQP 中目标函数 (RT 代表实时变化量)	13
表 2-5 OSQP 中线性约束 (RT 代表实时变化量)	14
表 4-1 符号解释	27
表 6-1 Hemisphere V102 常用指令	43
表 6-2 防水接头接线方法	44

算法索引

主要符号对照表

ϵ	介电常数
μ	磁导率
ϵ	介电常数
μ	磁导率
ϵ	介电常数
μ	磁导率
ϵ	介电常数
μ	磁导率
ϵ	介电常数
μ	磁导率
ϵ	介电常数
μ	磁导率
ϵ	介电常数
μ	磁导率
ϵ	介电常数
μ	磁导率
ϵ	介电常数
μ	磁导率
ϵ	介电常数
μ	磁导率
ϵ	介电常数
μ	磁导率
ϵ	介电常数
μ	磁导率
ϵ	介电常数
μ	磁导率
ϵ	介电常数
μ	磁导率

ϵ 介电常数

μ 磁导率

ϵ 介电常数

μ 磁导率

ϵ 介电常数

μ 磁导率

ϵ 介电常数

μ 磁导率

ϵ 介电常数

μ 磁导率

ϵ 介电常数

μ 磁导率

ϵ 介电常数

μ 磁导率

ϵ 介电常数

μ 磁导率

ϵ 介电常数

μ 磁导率

ϵ 介电常数

μ 磁导率

ϵ 介电常数

μ 磁导率

ϵ 介电常数

μ 磁导率

第一章 船体运动模型

本章介绍船体运动模型，主要分为单自由首要控制模型、3 自由度操纵性模型、3 自由度动力定位模型、4 自由度带有横摇补偿的操纵性模型。

1.1 坐标系

1.1.1 北东地坐标系

北东地 (North-East-Down: NED) 坐标系 $\{\mathbf{n}\} = (x_n, y_n, z_n)$ 是常用的导航坐标系，通常定义在地球表面的切平面上，其中 x 轴指向地球北， y 轴指向地球东， z 轴垂直于地球表面并指向下。通常对于海上船舶，经纬度的变化并不大，我们可以假定 $\{\mathbf{n}\}$ 是惯性参考系。

北东地坐标系在程序里也称 Marine coordinate, 与之对应的是 UTM 坐标系 (UTM-x 指向正东, UTM-y 指向正北)。

1.1.2 笛卡尔坐标系

笛卡尔 (Cartesian) 坐标系如图1-1在数学中是一种正交坐标系，由法国数学家勒内·笛卡尔引入而有此名。二维的直角坐标系是由两条相互垂直、相交于原点的数线构成的。在平面内，任何一点的坐标是根据数轴上对应的点的坐标设定的。在平面内，任何一点与坐标的对应关系，类似于数轴上点与坐标的对应关系。

1.1.3 随体坐标系

随体坐标系 $\{\mathbf{b}\}$ 始终跟随船体的运动，通常我们随体坐标系的原点 o_b 置于船体中轴线上，并位于船尾甲板处 (船体上比较容易接触到某一点)。一般我们用六个变量来描述船体的运动，主要分为平动 (x, y, z) 和转动 (ϕ, θ, ψ) ，具体的描述和符号见图1-2和表1-1。图中，CoG 在随体坐标系中相对于 o_b 的坐标可表示为 $\mathbf{r}_g = (x_g, y_g, z_g)^T$

1.1.4 运动方程

1.2 海洋环境载荷

风、浪、流

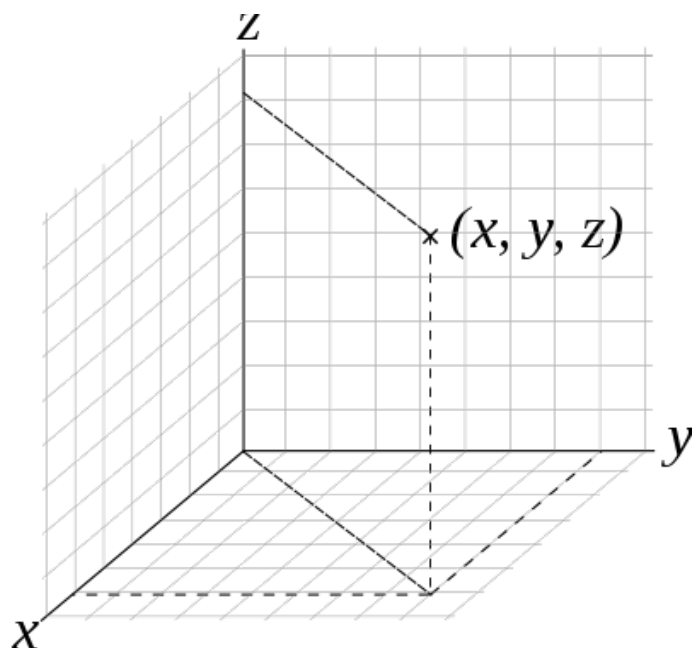


图 1-1 笛卡尔坐标系

Figure 1-1 Cartesian coordinate system

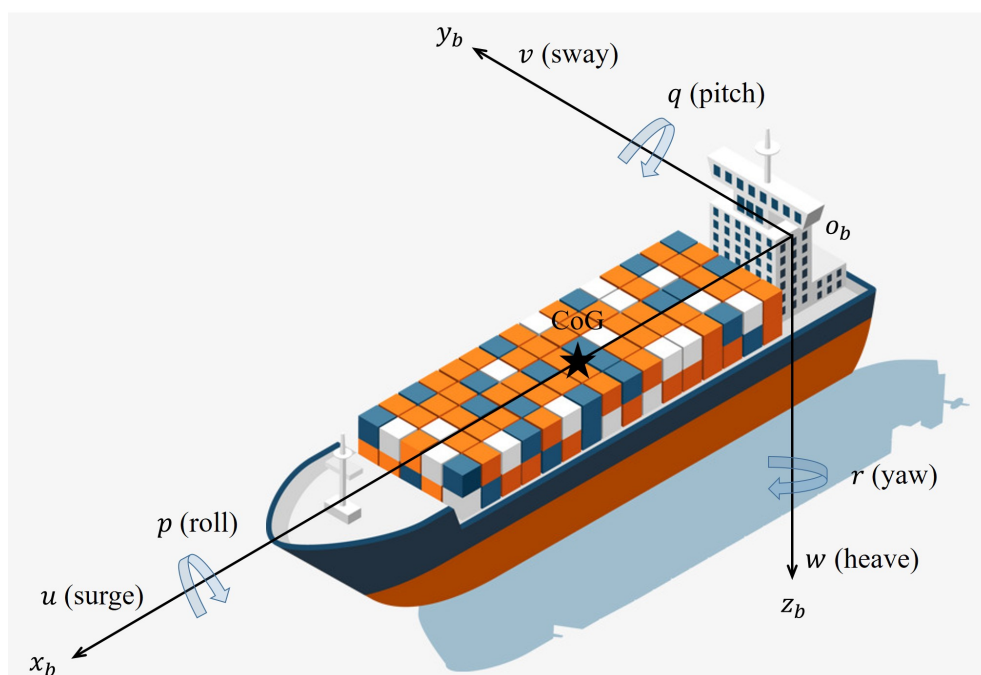


图 1-2 在随体坐标系 $\{\mathbf{b}\}$ 中, 船体六自由度的速度 u, v, w, p, q, r

Figure 1-2 The 6DoF velocities u, v, w, p, q, r in the body-fixed reference frame $\{\mathbf{b}\} = (x_b, y_b, z_b)$

表 1-1 描述船体运动的常用符号

自由度 (DOF)	力/力矩	线/角速度	位置/欧拉角
1 surge	X	u	x
2 sway	Y	v	y
3 heave	Z	w	z
4 roll	K	p	ϕ
5 pitch	M	q	θ
6 yaw	N	r	ψ

1.3 三自由度操纵性方程

三自由度操纵性方程只考虑低频的 surge, sway 和 yaw 方向上的运动, 并假定船体以恒定的速度前行 $u = U$, 忽略海流的作用,

$$(\mathbf{M}_{RB} + \mathbf{M}_A)\dot{\mathbf{v}} + (\mathbf{C}_{RB}(\mathbf{v}) + \mathbf{C}_A(\mathbf{v}) + \mathbf{D})\mathbf{v} = \boldsymbol{\tau} + \boldsymbol{\tau}_{wind} + \boldsymbol{\tau}_{wave} \quad (1-1)$$

$$\dot{\boldsymbol{\eta}} = \mathbf{J}(\boldsymbol{\eta})\mathbf{v} = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{v} \quad (1-2)$$

其中, $\mathbf{v} = (u, v, r)^T$ 是随体坐标系 $\{\mathbf{b}\}$ 下的速度, $\boldsymbol{\eta} = (N, E, \psi)^T$ 是北东地坐标系 $\{\mathbf{n}\}$ 下的位置, $\boldsymbol{\tau}, \boldsymbol{\tau}_{wind}, \boldsymbol{\tau}_{wave}$ 分别为作用在船体上的控制力、风力和流力, 均相对于随体坐标系。

$$\mathbf{M}_{RB} = \begin{pmatrix} m & 0 & 0 \\ 0 & m & mx_g \\ 0 & mx_g & I_z \end{pmatrix}, \quad \mathbf{M}_A = \begin{pmatrix} -X_{\dot{u}} & 0 & 0 \\ 0 & -Y_{\dot{v}} & -Y_{\dot{r}} \\ 0 & -Y_{\dot{v}} & -N_{\dot{r}} \end{pmatrix} \quad (1-3)$$

相对于随体坐标系原点 o_b 的附加质量 \mathbf{M}_A 可以通过势流计算软件得到; 在无人船程序中, 我们采用原点位于重心的随体坐标系 $\{\mathbf{b}_{CoG}\}$, 可使得质量矩阵为对角矩阵。同时推力、定位系统获得的位置、规控模块均相对于 $\{\mathbf{b}_{CoG}\}$ 。势流阻尼 $\mathbf{C}_A(\mathbf{v})$ 可通过势流计算软件得到, 有时可以忽略,

$$\mathbf{C}_A(\mathbf{v}) = \begin{pmatrix} 0 & 0 & Y_{\dot{v}}v + Y_{\dot{r}}r \\ 0 & 0 & -X_{\dot{u}}u \\ -Y_{\dot{v}}v - Y_{\dot{r}}r & X_{\dot{u}}u & 0 \end{pmatrix} \quad (1-4)$$

$$\mathbf{C}_{RB}(\mathbf{v}) = \begin{pmatrix} 0 & 0 & -m(x_g r + v) \\ 0 & 0 & mu \\ m(x_g r + v) & -mu & 0 \end{pmatrix} \quad (1-5)$$

线性阻尼矩阵 \mathbf{D} 可用模型试验辨识或者经验公式得到。线性阻尼通常在船舶低速航行的时候占主导，而高速的时候，非线性阻尼也需要考虑，参见^[1]第 136 页。

$$\mathbf{D} = \begin{pmatrix} -X_u & 0 & 0 \\ 0 & -Y_v & -Y_r \\ 0 & -N_v & -N_r \end{pmatrix} \quad (1-6)$$

1.4 三自由度动力定位方程

动力定位系统假定船体的航速很小，通常只考虑线性阻尼项，

$$(\mathbf{M}_{RB} + \mathbf{M}_A)\dot{\mathbf{v}} + \mathbf{D}\mathbf{v} = \boldsymbol{\tau} + \boldsymbol{\tau}_{wind} + \boldsymbol{\tau}_{wave} + \boldsymbol{\tau}_{current} \quad (1-7)$$

$$\dot{\boldsymbol{\eta}} = \mathbf{J}(\boldsymbol{\eta})\mathbf{v} \quad (1-8)$$

其中， \mathbf{M}_{RB} , \mathbf{M}_A , \mathbf{D} , $\mathbf{J}(\boldsymbol{\eta})$ 和 3DoF 操纵性方程是一样的，通常我们假定风力和流是缓变的，并根据经验公式估算得到。

1.5 单自由度首向控制方程

1.6 四自由度横摇补偿操纵性方程

1.7 参数辨识

假定一个弹簧振子的质量为 m , 弹簧刚度为 k , 则

$$\begin{aligned} F &= k\Delta x \\ T &= 2\pi\sqrt{\frac{m}{k}} \end{aligned} \quad (1-9)$$

进而假设船体的质量为 m , 其转动惯量为 I , 则船体受的恢复力矩为

$$\begin{aligned} \frac{mg}{2} \tan(\alpha) &\approx \frac{mg}{2} \alpha \\ &= \frac{mg}{2} \frac{\Delta\theta d}{L} \end{aligned} \quad (1-10)$$

$$M = \frac{mgd^2}{L} \Delta\theta \quad (1-11)$$

运用单摆原理，可得周期为

$$T = 2\pi \sqrt{\frac{I}{\frac{mgd^2}{L}}} = 2\pi \sqrt{\frac{IL}{mgd^2}} \quad (1-12)$$

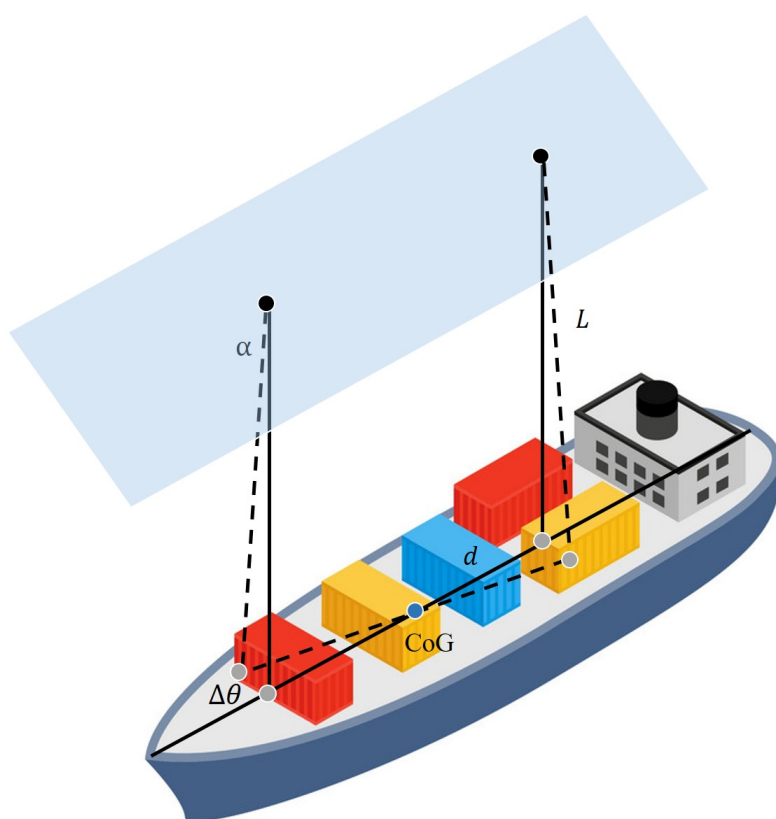


图 1-3 转动惯量的测量原理

Figure 1-3 Sketch of Inertial Measurement

1.7.1 质量

重心坐标一般相对于 o_b

第二章 控制器

船舶系统按不同的推进器配置可分为欠驱动 (Underactuated systems) 和全驱动 (fully-actuated systems) 系统。通常所说的动力定位船舶或者平台即为全驱动系统, 这类系统的输入量不少于控制量。而欠驱动船舶是输入比要控制的量少的一类典型系统。本章介绍船舶控制器, 主要包括 PID 控制器, 轨迹跟踪算法和推力分配算法。

2.1 推力分配

2.1.1 推进器简介

船用推进器主要包括隧道式螺旋桨、全回转螺旋桨、固定式螺旋桨加舵以及泵推。其简图如2-1



图 2-1 常见船舶推进器示意图

Figure 2-1 Sketch of common thrusters of vessel

对于每种推进器, 其推力可根据船级社 DNVGL 的规范^[2] 估算得到。

$$\tau_e = \beta_T \tau_n \quad (2-1)$$

其中, τ_e 称为有效推力, τ_n 是名义推力, β_T 是推力损失系数。

$$\tau_n = \eta_1 \eta_2 (D \times P)^{\frac{2}{3}} \quad (2-2)$$

其中, D 是以米 (m) 为单位的螺旋桨直径, P 是以千瓦 (kW) 为单位的推进器功率。 η_1 和 η_2 的取值参见 DNVGL 的规范。

对于舵，其推力

$$\begin{aligned}
 \tau_{surge} &= \tau_e(1 - C_x\alpha^2) \\
 \tau_{sway} &= \tau_e C_y \alpha \\
 C_x &= 0.02 C_y \\
 C_y &= 0.0126 k_1 k_2 \frac{A_r}{D^2}
 \end{aligned} \tag{2-3}$$

其中 α 是以度 ($^\circ$) 为单位的舵角，一般舵角不会超过 30° 。 A_r 是舵可转动部分的面积。 k_1 和 k_2 的取值参见规范。

2.1.2 推力分配模型

推力分配算法将广义力作为输入，通过建模并优化得到每个推进器上的转速和推力。

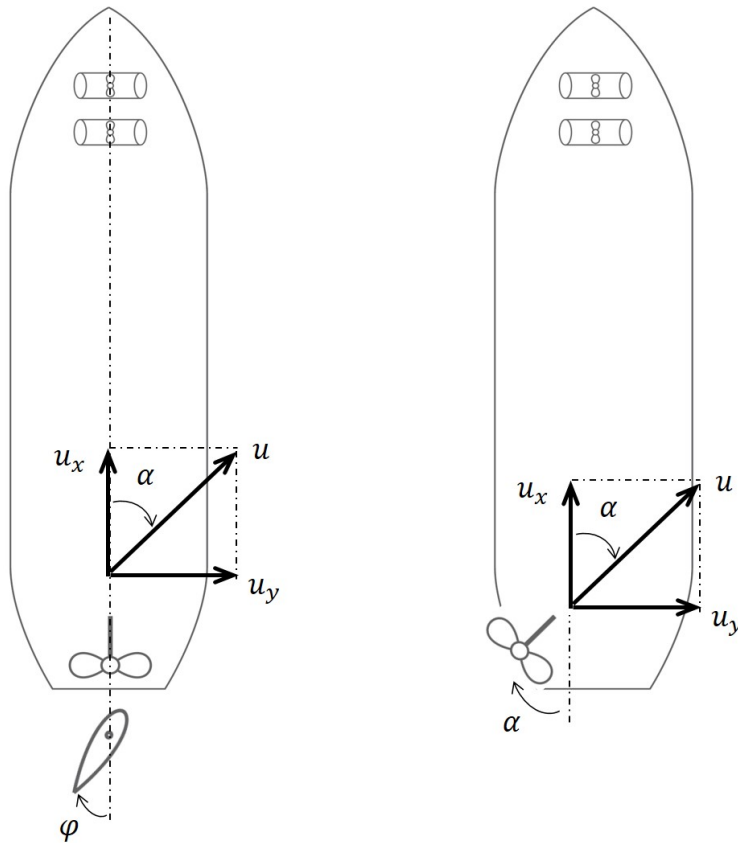


图 2-2 螺旋桨角度的定义

Figure 2-2 Sketch of azimuth of each thruster

$$\boldsymbol{\tau}(t) = \mathbf{B}(\boldsymbol{\alpha}) \cdot \mathbf{u} \quad (2-4)$$

其中, $\mathbf{u} = [u_1, u_2, \dots, u_m]^T \in \mathbb{R}^m$ 代表每个推进器产生的推力, $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \dots, \alpha_m]^T \in \mathbb{R}^m$ 代表每个推进器的角度, 其定义参见图2-2。假设在随体坐标系中, 推进器相对于重心的位置为 (l_{xi}, l_{yi}) , 则矩阵 $\mathbf{B}(\boldsymbol{\alpha}) \in \mathbb{R}^{3 \times m}$ 可得

$$\mathbf{B}(\boldsymbol{\alpha}) = \begin{bmatrix} \cos \alpha_1 & \cos \alpha_2 & \cdots & \cos \alpha_m \\ \sin \alpha_1 & \sin \alpha_2 & \cdots & \sin \alpha_m \\ l_{x1} \sin \alpha_1 - l_{y1} \cos \alpha_1 & l_{x2} \sin \alpha_2 - l_{y2} \cos \alpha_2 & \cdots & l_{xm} \sin \alpha_m - l_{ym} \cos \alpha_m \end{bmatrix} \quad (2-5)$$

我们假设推力和螺旋桨转速 n 呈二次关系

$$u_i = K_i n_i |n_i| \quad (2-6)$$

其中, K_i 可以从推力测量实验 (详见2.4节) 得到。

我们将推力分配问题转化为带有线性约束的二次规划问题 (Quadratic Programming)^[3], 目前我们采用 Mosek 或 osqp 求解器来计算优化解。

$$\begin{aligned} \min J(\Delta \boldsymbol{\alpha}, \Delta \mathbf{u}, s) &= \sum_{i=1}^m \left(\frac{dW_i}{du_i}(u_{0,i}) \Delta u_i + \frac{d^2 W_i}{du_i^2}(u_{0,i}) \Delta u_i^2 \right) + \Delta \boldsymbol{\alpha}^T \boldsymbol{\Omega} \Delta \boldsymbol{\alpha} + s^T \mathbf{Q} s \\ &\quad + \frac{d}{d\boldsymbol{\alpha}} \left(\frac{\rho}{\varepsilon + \det(\mathbf{B}(\boldsymbol{\alpha}) \mathbf{B}^T(\boldsymbol{\alpha}))} \right)_{\boldsymbol{\alpha}=\boldsymbol{\alpha}_0} \cdot \Delta \boldsymbol{\alpha} \\ &= \mathbf{g}_{\Delta u}^T \cdot \Delta \mathbf{u} + \Delta \mathbf{u}^T \mathbf{Q}_{\Delta u} \Delta \mathbf{u} + s^T \mathbf{Q} s + \Delta \boldsymbol{\alpha}^T \boldsymbol{\Omega} \Delta \boldsymbol{\alpha} + \mathbf{d}_{\rho}^T \Delta \boldsymbol{\alpha} \end{aligned} \quad (2-7)$$

subject to

$$\begin{aligned} s + \mathbf{B}(\boldsymbol{\alpha}_0) \cdot \Delta \mathbf{u} + \frac{\partial}{\partial \boldsymbol{\alpha}} (\mathbf{B}(\boldsymbol{\alpha}) \mathbf{u})_{\boldsymbol{\alpha}=\boldsymbol{\alpha}_0} \cdot \Delta \boldsymbol{\alpha} &= \boldsymbol{\tau} - \mathbf{B}(\boldsymbol{\alpha}_0) \cdot \mathbf{u}_0 \\ \underline{\Delta \mathbf{u}} \leq \Delta \mathbf{u} \leq \overline{\Delta \mathbf{u}} \\ \underline{\Delta \boldsymbol{\alpha}} \leq \Delta \boldsymbol{\alpha} \leq \overline{\Delta \boldsymbol{\alpha}} \end{aligned} \quad (2-8)$$

这里, \leq 代表向量不等式, W 代表推进器的功率, 通常假设 $W = u^{\frac{3}{2}}$ 。 $u_{0,i}$ 代表上一时刻第 i 个推进器产生的推力, s 代表估计推力与预期推力的偏差。我们用

$\mathbf{z} = [\Delta \mathbf{u}, \Delta \boldsymbol{\alpha}, s]^T \in \mathbb{R}^{2m+3}$, 上述方程简化可得

$$\begin{aligned} \min \mathbf{z}^T \mathbf{H} \mathbf{z} + \mathbf{g}^T \mathbf{z} \\ \text{s.t. } \mathbf{P} \mathbf{z} \leq \mathbf{h} \\ \mathbf{c} \mathbf{z} = \mathbf{b} \end{aligned} \quad (2-9)$$

$$\begin{aligned} \mathbf{c} &= \begin{bmatrix} \mathbf{B}(\boldsymbol{\alpha}_0) & \frac{\partial}{\partial \boldsymbol{\alpha}} (\mathbf{B}(\boldsymbol{\alpha}) \mathbf{u})|_{\mathbf{u}=\mathbf{u}_0, \boldsymbol{\alpha}=\boldsymbol{\alpha}_0} & \mathbf{I} \end{bmatrix} \\ \mathbf{b} &= \boldsymbol{\tau} - \mathbf{B}(\boldsymbol{\alpha}_0) \cdot \mathbf{u}_0 \\ \mathbf{H} &= \begin{bmatrix} \mathbf{Q}_{\Delta \mathbf{u}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Omega} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Q} \end{bmatrix} \\ \mathbf{g} &= \begin{bmatrix} \mathbf{g}_{\Delta \mathbf{u}} \\ \mathbf{d}_\rho \\ \mathbf{0} \end{bmatrix} \\ \mathbf{P} &= \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ -\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} & \mathbf{0} \end{bmatrix} \\ \mathbf{h} &= \begin{bmatrix} \overline{\Delta \mathbf{u}} \\ -\underline{\Delta \mathbf{u}} \\ \overline{\Delta \boldsymbol{\alpha}} \\ -\underline{\Delta \boldsymbol{\alpha}} \end{bmatrix} \end{aligned} \quad (2-10)$$

$$\begin{aligned} \mathbf{g}_{\Delta \mathbf{u}} &= \begin{bmatrix} \frac{3}{2} u_{0,1}^{\frac{1}{2}} \\ \frac{3}{2} u_{0,2}^{\frac{1}{2}} \\ \vdots \\ \frac{3}{2} u_{0,m}^{\frac{1}{2}} \end{bmatrix} \\ \mathbf{Q}_{\Delta \mathbf{u}} &= \begin{bmatrix} \frac{3}{4} u_{0,1}^{-\frac{1}{2}} & & & \\ & \frac{3}{4} u_{0,2}^{-\frac{1}{2}} & & \\ & & \ddots & \\ & & & \frac{3}{4} u_{0,m}^{-\frac{1}{2}} \end{bmatrix} \end{aligned} \quad (2-11)$$

每个推进器的推力变化和角度变化的上下限是实时变化的，针对每种推进器，其约束条件将在下面几小节讨论。

2.1.3 求解器

目前我们采用 Mosek 或 osqp 求解器来计算优化解，由于 Mosek 是商业求解器，之后会逐渐采用开源的求解器来替代 Mosek。

2.1.3.1 Mosek

本节介绍使用 Mosek 求解器的一些细节。我们用 m 表示推进器的数量， n 表示推力的维度 ($n = 3$)，则优化问题 (2-9) 中的参数 $\mathbf{z} \in \mathbb{R}^{2m+n}$ ， $\mathbf{b} \in \mathbb{R}^n$ ， $\mathbf{c} \in \mathbb{R}^{n \times (2m+n)}$

- 变量 \mathbf{z} 的约束 (对应 $\mathbf{Pz} \leq \mathbf{h}$)，如表 (2-1)
- 线性约束 (对应 $\mathbf{cz} = \mathbf{b}$)，如表 (2-2)
- 目标函数 (对应 $\mathbf{z}^T \mathbf{Hz} + \mathbf{g}^T \mathbf{z}$)，如表 (2-3)

表 2-1 Mosek 中变量约束 (RT 代表实时变化量)

Type	Name	Index		
		0:(m-1)	m:(2m-1)	2m:(2m+n-1)
MSKboundeye	bxx[]	MSK_BK_RA	MSK_BK_RA	MSK_BK_FR
double	blx[]	RT	RT	-MSK_INFI
double	bux[]	RT	RT	MSK_INFI

2.1.3.2 osqp

osqp 是由牛津大学和斯坦福大学合作开发的一款二次规划求解器。目前 0.6.0 的版本可以求解线性约束下的二次规划问题。

$$\begin{aligned}
 \min \quad & \frac{1}{2} \mathbf{x}^T \mathbf{P} \mathbf{x} + \mathbf{q}^T \mathbf{x} \\
 \text{s.t.} \quad & \mathbf{l} \leq \mathbf{A} \mathbf{x} \leq \mathbf{u}
 \end{aligned} \tag{2-12}$$

表 2-2 Mosek 中线性约束 (RT 代表实时变化量)

Type	Name	Index							
		0	1	2	...	2m-1	2m	2m+1	2m+2
MSKint32t	aptrb[]	0	3	6	...	6m-3	6m	6m+1	6m+2
MSKint32t	aptre[]	3	6	9	...	6m	6m+1	6m+2	6m+3

Type	Name	Index												
		0	1	2	3	4	5	...	6m-3	6m-2	6m-1	6m	6m+1	6m+2
MSKint32t	asub[]	0	1	2	0	1	2	...	0	1	2	0	1	2
double	aval[]	RT	RT	RT	RT	RT	RT	...	RT	RT	RT	1	1	1

Type	Name	Index		
		0	1	2
MSKboundkeye	bkc[]	MSK_BK_FX	MSK_BK_FX	MSK_BK_FX
double	blc[]	RT	RT	RT
double	buc[]	blc[0]	blc[1]	blc[2]

表 2-3 Mosek 中目标函数 (RT 代表实时变化量)

Type	Name	Index									
		0	1	...	m-1	m	...	2m-1	2m	2m+1	2m+2
MSKint32t	qsubi[]	0	1	...	m-1	m	...	2m-1	2m	2m+1	2m+2
MSKint32t	qsubj[]	0	1	...	m-1	m	...	2m-1	2m	2m+1	2m+2
double	qval[]	RT	RT	...	RT	$\pmb{\Omega}[0][0]$...	$\pmb{\Omega}[m-1][m-1]$	$\pmb{Q}[0][0]$	$\pmb{Q}[1][1]$	$\pmb{Q}[2][2]$

$$\begin{aligned}
A &= \begin{bmatrix} B(\alpha_0) & \frac{\partial}{\partial \alpha}(B(\alpha)u)|_{u=u_0, \alpha=\alpha_0} & I_n \\ I_m & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I_m & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_n \end{bmatrix} \\
l &= \begin{bmatrix} b \\ \frac{\Delta u}{\Delta \alpha} \\ -\infty \end{bmatrix} \\
u &= \begin{bmatrix} b \\ \frac{\Delta u}{\Delta \alpha} \\ \infty \end{bmatrix}
\end{aligned} \tag{2-13}$$

- 目标函数 (对应 $\mathbf{z}^T \mathbf{H} \mathbf{z} + \mathbf{g}^T \mathbf{z}$), 且 $P_{nnz} = 2m + n$, 如表 (2-4)
- 线性约束, $A_{nnz} = 2(mn + m + n)$, 且 $u, l \in \mathbb{R}^{2m+2n}$, 如表 (2-5)

表 2-4 OSQP 中目标函数 (RT 代表实时变化量)

Type	Name	Index										
		0	1	...	m-1	m	...	2m-1	2m	2m+1	2m+2	2m+3
c_float	P_x[]	$\frac{3}{4}u_{0,1}^{-\frac{1}{2}}$	$\frac{3}{4}u_{0,2}^{-\frac{1}{2}}$...	$\frac{3}{4}u_{0,m}^{-\frac{1}{2}}$	$\boldsymbol{\Omega}[0][0]$...	$\boldsymbol{\Omega}[m-1][m-1]$	$\boldsymbol{Q}[0]$	$\boldsymbol{Q}[1]$	$\boldsymbol{Q}[2]$	N/A
c_int	P_i[]	0	1	...	m-1	m	...	2m-1	2m	2m+1	2m+2	N/A
c_int	P_p[]	0	1	...	m-1	m	...	2m-1	2m	2m+1	2m+2	2m+3

2.1.4 欠驱动

目前大部分船舶是欠驱动的, 常见的欠驱动船舶的推进器配置有:(1) 双固定式尾推; (2) 主推配合船舵

2.1.4.1 双固定式尾推

- 转向可通过两个螺旋桨差速实现
- 通常用伺服电机或者步进电机带动螺旋桨
- 对于步进电机, 为了防止过快的正反转切换, 我们人为限制其切换速度。

表 2-5 OSQP 中线性约束 (RT 代表实时变化量)

Type		c_float	c_int
Name		A_x[]	A_i[]
	0	$\cos \alpha_1$	0
	1	$\sin \alpha_1$	1
	2	$l_{x1} \sin \alpha_1 - l_{y1} \cos \alpha_1$	2
	3	1	3

	4m-4	$\cos \alpha_m$	0
	4m-3	$\sin \alpha_m$	1
	4m-2	$l_{xm} \sin \alpha_m - l_{ym} \cos \alpha_m$	2
Index	4m-1	1	m+2
	4m	RT	0
	4m+1	RT	1
	4m+2	RT	2
	4m+3	1	m+3

	8m-4	RT	2m-1
	8m-3	RT	2m
	8m-2	RT	2m+1
	8m-1	1	2m+2
	8m	1	0
	8m+1	1	2m+3
	8m+2	1	1
	8m+3	1	2m+4
	8m+4	1	2
	8m+5	1	2m+5

推力曲线如2-3所示，正向的推力常数为 k^+ ，负向的为 k^- 。我们用 v_n 表示螺旋桨转速的最大变化率， Δt 表示采样时间， t_{p2n} 表示螺旋桨从正到负或者从负到正的切换时间

$$\begin{aligned}\Delta n_{max} &= v_n \Delta t \\ \Delta n_{max}^{p2n} &= \frac{2\Delta n_{max} \Delta t}{t_{p2n}}\end{aligned}\quad (2-14)$$

为了考虑正反转切换时间，我们采用了两种不同的转速限制：通常转速变化的极值为 Δn_{max} ；而在 $-\Delta n_{max}$ 和 Δn_{max} 之间，转速变化的极值为 Δn_{max}^{p2n} ，进而我们可以得到不同转速下，优化问题的约束条件。

1. $n_0 \geq \Delta n_{max}$
 - $\underline{\Delta\alpha} = \overline{\Delta\alpha} = 0$
 - $\underline{\Delta u} = \min(u_{max} - k^+ n_0^2, k^+(n_0 + \Delta n_{max})^2 - k^+ n_0^2)$
 - $\underline{\Delta u} = k^+(n_0 - \Delta n_{max})^2 - k^+ n_0^2$
2. $\Delta n_{max}^{p2n} \leq n_0 < \Delta n_{max}$
 - $\underline{\Delta\alpha} = \overline{\Delta\alpha} = 0$
 - $\underline{\Delta u} = k^+(n_0 + \Delta n_{max}^{p2n})^2 - k^+ n_0^2$
 - $\underline{\Delta u} = k^+(n_0 - \Delta n_{max}^{p2n})^2 - k^+ n_0^2$
3. $0 < n_0 < \Delta n_{max}^{p2n}$
 - if $u_d < 0$
 - $\underline{\Delta\alpha} = \overline{\Delta\alpha} = \pi$
 - $\underline{\Delta u} = \underline{\Delta u} = k^-(n_0 - \Delta n_{max}^{p2n})^2 - k^+ n_0^2$
 - elseif $u_d \geq 0$
 - $\underline{\Delta\alpha} = \overline{\Delta\alpha} = 0$
 - $\underline{\Delta u} = \underline{\Delta u} = k^+(n_0 + \Delta n_{max}^{p2n})^2 - k^+ n_0^2$
4. $-\Delta n_{max}^{p2n} < n_0 < 0$
 - if $u_d > 0$
 - $\underline{\Delta\alpha} = \overline{\Delta\alpha} = -\pi$
 - $\underline{\Delta u} = \underline{\Delta u} = k^+(n_0 + \Delta n_{max}^{p2n})^2 - k^- n_0^2$
 - elseif $u_d \leq 0$
 - $\underline{\Delta\alpha} = \overline{\Delta\alpha} = 0$
 - $\underline{\Delta u} = \underline{\Delta u} = k^-(n_0 - \Delta n_{max}^{p2n})^2 - k^- n_0^2$
5. $-\Delta n_{max} < n_0 \leq -\Delta n_{max}^{p2n}$
 - $\underline{\Delta\alpha} = \overline{\Delta\alpha} = 0$

- $\overline{\Delta u} = k^-(n_0 - \Delta n_{max}^{p2n})^2 - k^-n_0^2$
 - $\underline{\Delta u} = k^-(n_0 + \Delta n_{max}^{p2n})^2 - k^-n_0^2$
6. $n_0 \leq -\Delta n_{max}$
- $\overline{\Delta \alpha} = \underline{\Delta \alpha} = 0$
 - $\overline{\Delta u} = \min(u_{min} - k^-n_0^2, k^-(n_0 - \Delta n_{max})^2 - k^-n_0^2)$
 - $\underline{\Delta u} = k^-(n_0 + \Delta n_{max})^2 - k^-n_0^2$

其中, u_d 表示每个推进器上期望的推力, 如图2-4所示,

$$\begin{aligned} u_1 &= \frac{\tau_x}{2} + \frac{\tau_M}{2l} \\ u_2 &= \frac{\tau_x}{2} - \frac{\tau_M}{2l} \end{aligned} \quad (2-15)$$

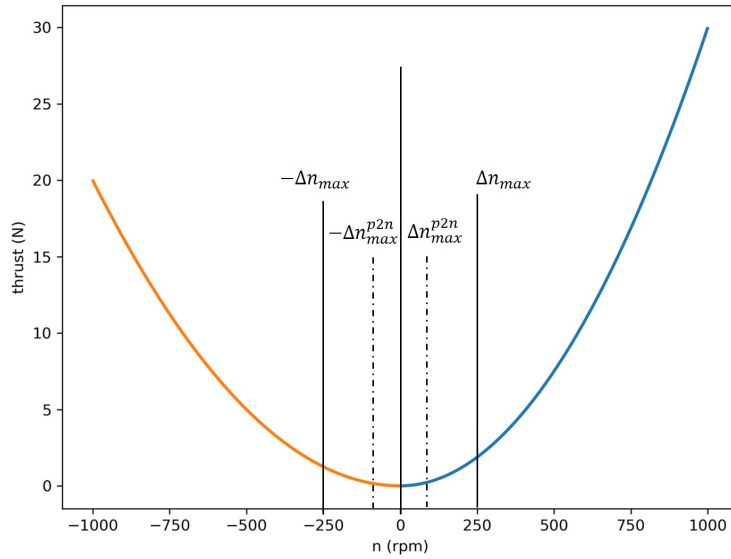


图 2-3 固定式双尾推螺旋桨的推力曲线

Figure 2-3 Thrust(N) vs. rotational speed (rpm) of Twin fixed thruster

2.1.4.2 主推配合船舵

如图2-2所示, 船舵的舵角 φ 和其产生推力方向 α 不一致, 他们的关系参见公式 (2-3)。

$$\begin{aligned} u_x &= kn^2(1 - 0.02C_y\varphi^2) \\ u_y &= kn^2C_y\varphi \end{aligned} \quad (2-16)$$

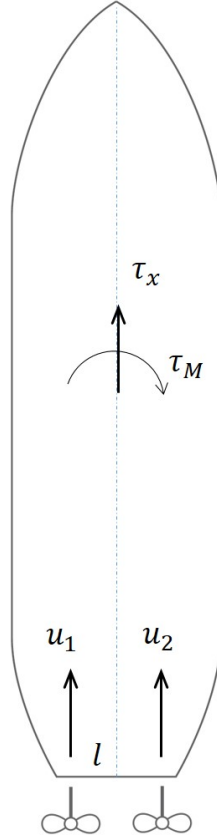


图 2-4 固定式双尾推螺旋桨的推力分配

Figure 2-4 Thrust allocation of Twin fixed thruster

由此可得

$$\tan \alpha = \frac{C_y \varphi}{1 - 0.02 C_y \varphi^2}, \quad \alpha \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] \quad (2-17)$$

$$u = kn^2 \sqrt{(0.02 C_y \varphi^2)^2 + (C_y^2 - 0.04 C_y) \varphi^2 + 1}$$

利用 $\tan \alpha$ 的单调性，我们能够计算 α 的约束条件；同时， u 的极值也可以分别计算 kn^2 和 $(0.02 C_y \varphi^2)^2 + (C_y^2 - 0.04 C_y) \varphi^2 + 1$ 的极值得到。

2.1.5 全驱动船

北东地 (North-East-Down: NED) 坐标系 $\{\mathbf{n}\} = (x_n, y_n, z_n)$ 是常用的导航坐标系，通常定义在地球表面的切平面上，其中 x 轴指向地球北， y 轴指向地球东， z 轴垂直于地球表面并指向下。通常对于海上船舶，经纬度的变化并不大，我们可以假定 $\{\mathbf{n}\}$ 是惯性参考系。

2.1.5.1 隧道式艏侧推

1. $n_0 \geq \Delta n_{max}$
 - $\underline{\Delta\alpha} = \overline{\Delta\alpha} = 0$
 - $\overline{\Delta u} = \min(u_{max} - k^+ n_0^2, k^+(n_0 + \Delta n_{max})^2 - k^+ n_0^2)$
 - $\underline{\Delta u} = k^+(n_0 - \Delta n_{max})^2 - k^+ n_0^2$
2. $0 < n_0 < \Delta n_{max}$
 - if $u_d < 0$
 - $\underline{\Delta\alpha} = \overline{\Delta\alpha} = -\pi$
 - $\overline{\Delta u} = \underline{\Delta u} = k^-(n_0 - \Delta n_{max})^2 - k^+ n_0^2$
 - elseif $u_d \geq 0$
 - $\underline{\Delta\alpha} = \overline{\Delta\alpha} = 0$
 - $\overline{\Delta u} = k^+(n_0 + \Delta n_{max})^2 - k^+ n_0^2$
 - $\underline{\Delta u} = -k^+ n_0^2$
3. $-\Delta n_{max} < n_0 < 0$
 - if $u_d > 0$
 - $\underline{\Delta\alpha} = \overline{\Delta\alpha} = \pi$
 - $\overline{\Delta u} = \underline{\Delta u} = k^+(n_0 + \Delta n_{max})^2 - k^- n_0^2$
 - elseif $u_d \leq 0$
 - $\underline{\Delta\alpha} = \overline{\Delta\alpha} = 0$
 - $\overline{\Delta u} = k^-(n_0 - \Delta n_{max})^2 - k^- n_0^2$
 - $\underline{\Delta u} = -k^- n_0^2$
4. $n_0 \leq -\Delta n_{max}$
 - $\underline{\Delta\alpha} = \overline{\Delta\alpha} = 0$
 - $\overline{\Delta u} = \min(u_{min} - k^- n_0^2, k^-(n_0 - \Delta n_{max})^2 - k^- n_0^2)$
 - $\underline{\Delta u} = k^-(n_0 + \Delta n_{max})^2 - k^- n_0^2$

2.2 PID 控制器

PID 控制器将位置和速度误差作为输入，计算得到预期的广义力，并通过推力分配算法给出每个推进器的转速和角度。对于一般的 PID 控制器，我们定义 $e(t) = SP - PV$ 为误差，则

$$u(t) = u_{bias} + K_p e(t) + \frac{K_I}{\tau_I} \int_0^t e(t) dt + K_d \dot{e}(t) \quad (2-18)$$

其中, SP 代表设定点 (setpoint), PV 代表实际点。对于离散情况下, 积分项通常为

$$\sum_{i=n_t-N}^{n_t} e_i(t) \Delta t \quad (2-19)$$

其中, n_t 代表当前时刻, N 表示积分长度。一般而言, 对于速度控制, 我们通常采用 PD 控制器, 可以减少超调和震荡; 对于位置控制, 可以用 PI 控制器减少静差。

2.2.1 动力定位

2.3 轨迹跟踪

2.3.1 视线跟踪法

视线跟踪法 (line of sight)^[4] 是一个路径规划的算法, 最早用于航空航天和导弹制导领域, 其主要思路是根据船的位置计算得到预期的首向角, 从而使得船体沿着预期的路径运动。如果我们对船体的前进速度 u 作反馈控制, 即

$$\lim_{t \rightarrow \infty} [u(t) - u_d(t)] = 0 \quad (2-20)$$

其中, u_d 是预期的前进速度。对于 LOS, 我们定义 $\eta = [s, d]^T$, 则

$$\begin{aligned} \mathbf{R}(\theta_k) &= \begin{bmatrix} \cos \theta_k & \sin \theta_k \\ -\sin \theta_k & \cos \theta_k \end{bmatrix} \\ \eta &= \mathbf{R}(\theta_k) \begin{bmatrix} x_0 - x_{wp1} \\ y_0 - y_{wp1} \end{bmatrix} \end{aligned} \quad (2-21)$$

$$\begin{aligned} \theta_{los} &= \theta_r + \theta_k \\ \theta_k &= \arctan \frac{y_{wp2} - y_{wp1}}{x_{wp2} - x_{wp1}} \\ \theta_r &= \arcsin \frac{-d}{nL_{pp}} \end{aligned} \quad (2-22)$$

2.4 参数辨识

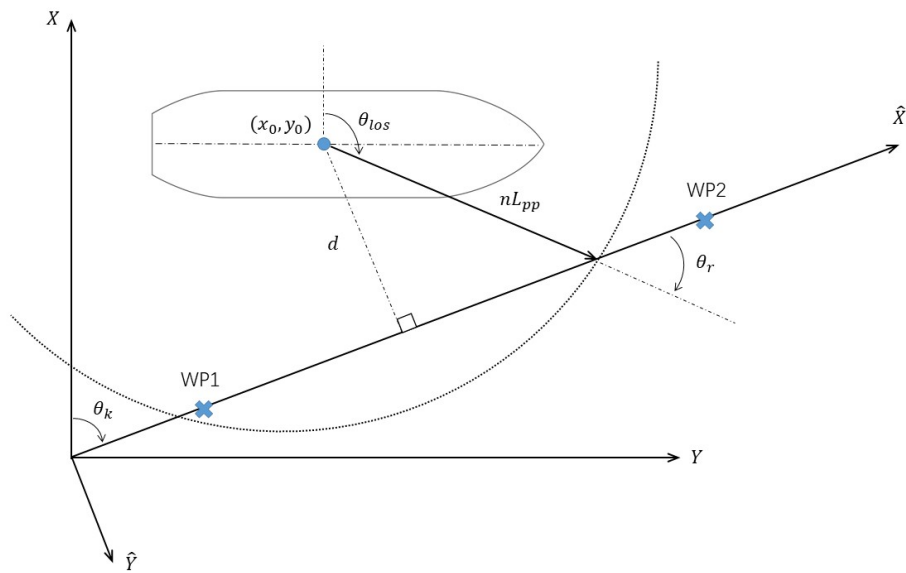


图 2-5 基于视线的轨迹跟踪算法

Figure 2-5 Trajectory tracking using line of sight(LOS)

第三章 观测器

3.1 移动平均滤波

简单移动平均（英语：simple moving average, SMA）是某变数之前 n 个数值的未作加权算术平均

$$SMA = \frac{p_1 + p_2 + \cdots + p_n}{n} \quad (3-1)$$

当计算连续的数值，一个新的数值加入，同时一个旧数值剔除，所以无需每次都重新逐个数值加起来：

$$SMA_{t1,n} = SMA_{t0,n} - \frac{p_1}{n} + \frac{p_{n+1}}{n} \quad (3-2)$$

然而，移动平均滤波的结果存在时滞，当 n 较大时，滤波器产生的结果将不可靠。

3.2 卡尔曼滤波

3.2.1 概述

卡尔曼滤波（Kalman filter）是一种高效率的递归滤波器（自回归滤波器），它能够从一系列的不完全及包含噪声的测量中，估计动态系统的状态。卡尔曼滤波会根据各测量量在不同时间下的值，考虑各时间下的联合分布，再产生对未知变数的估计，因此会比只以单一测量量为基础的估计方式要准。

卡尔曼滤波建立在线性代数和隐马尔可夫模型（hidden Markov model）上。其基本动态系统可以用一个马尔可夫链表示，该马尔可夫链建立在一个被高斯噪声（即正态分布的噪声）干扰的线性算子上的。系统的状态可以用一个元素为实数的向量表示。随着离散时间的每一个增加，这个线性算子就会作用在当前状态上，产生一个新的状态，并也会带入一些噪声，同时系统的一些已知的控制器的控制信息也会被加入。同时，另一个受噪声干扰的线性算子产生出这些隐含状态的可见输出。

3.2.2 模型

如图3-1所示，为了从一系列有噪声的观察数据中用卡尔曼滤波器估计出被观察过程的内部状态，必须把这个过程在卡尔曼滤波的框架下建立模型。也就是说对于每一步 k ，定义矩阵 F_k, H_k, Q_k, R_k ，有时也需要定义 B_k ，如下。

卡尔曼滤波模型假设 k 时刻的真实状态是从 $(k-1)$ 时刻的状态演化而来，符合下式：

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{B}_k \mathbf{u}_k + \mathbf{w}_k \quad (3-3)$$

其中

- \mathbf{F}_k 是作用在 \mathbf{x}_{k-1} 上的状态变换模型（/矩阵/矢量）。
- \mathbf{B}_k 是作用在控制器向量 \mathbf{u}_k 上的输入-控制模型。
- \mathbf{w}_k 是过程噪声，并假定其符合均值为零，协方差矩阵为 \mathbf{Q}_k 的多元正态分布。
- $\mathbf{w}_k \sim N(0, \mathbf{Q}_k)$

时刻 k ，对真实状态 \mathbf{x}_k 的一个测量 \mathbf{z}_k 满足下式：

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \quad (3-4)$$

其中 \mathbf{H}_k 是观测模型，它把真实状态空间映射成观测空间， \mathbf{v}_k 是观测噪声，其均值为零，协方差矩阵为 \mathbf{R}_k ，且服从正态分布 $\mathbf{v}_k \sim N(0, \mathbf{R}_k)$ 。初始状态以及每一时刻的噪声 $\mathbf{x}_0, \mathbf{w}_1, \dots, \mathbf{w}_k, \mathbf{v}_1, \dots, \mathbf{v}_k$ 都认为是互相独立的。

实际上，很多真实世界的动态系统都并不确切的符合这个模型；但是由于卡尔曼滤波器被设计在有噪声的情况下工作，一个近似的符合已经可以使这个滤波器非常有用。

3.2.3 算法

卡尔曼滤波是一种递归的估计，即只要获知上一时刻状态的估计值以及当前状态的观测值就可以计算出当前状态的估计值，因此不需要记录观测或者估计的历史信息。卡尔曼滤波器与大多数滤波器不同之处，在于它是一种纯粹的时域滤波器，它不需要像低通滤波器等频域滤波器那样，需要在频域设计再转换到时域实现。

卡尔曼滤波器的状态由以下两个变量表示：

- $\hat{\mathbf{x}}_{k|k}$ ，在时刻 k 的状态的估计；
- $\mathbf{P}_{k|k}$ ，后验估计误差协方差矩阵，度量估计值的精确程度。

卡尔曼滤波器的操作包括两个阶段：**预测与更新**。在预测阶段，滤波器使用上一状态的估计，做出对当前状态的估计。在更新阶段，滤波器利用对当前状态的观测值优化在预测阶段获得的预测值，以获得一个更精确的新估计值。

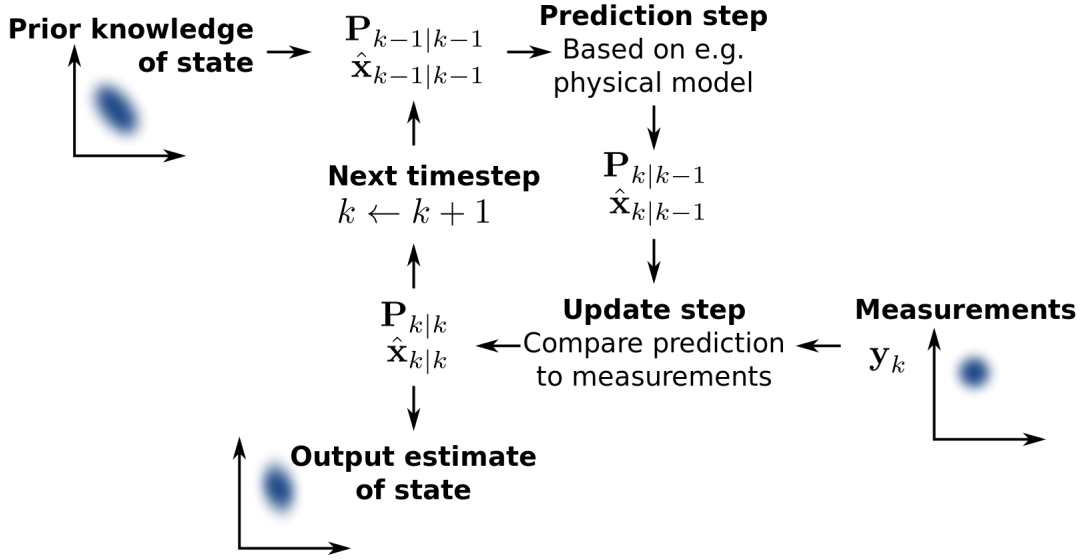


图 3-1 卡尔曼滤波的示意图

Figure 3-1 The Kalman filter keeps track of the estimated state of the system and the variance or uncertainty of the estimate. The estimate is updated using a state transition model and measurements. $\hat{\mathbf{x}}_{k|k-1}$ denotes the estimate of the system's state at time step k before the k -th measurement \mathbf{y}_k has been taken into account; $\mathbf{P}_{k|k-1}$ is the corresponding uncertainty.

3.2.3.1 预测

$$\begin{aligned}\hat{\mathbf{x}}_{k|k-1} &= \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_k \mathbf{u}_k \text{ (预测状态)} \\ \mathbf{P}_{k|k-1} &= \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{Q}_k \text{ (预测估计协方差矩阵)}\end{aligned}\quad (3-5)$$

3.2.3.2 更新

首先要算出以下三个量：

$$\begin{aligned}\tilde{\mathbf{y}}_k &= \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1} \text{ (测量余量, measurement residual)} \\ \mathbf{S}_k &= \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k \text{ (测量余量协方差)} \\ \mathbf{K}_k &= \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1} \text{ (最优卡尔曼增益)}\end{aligned}\quad (3-6)$$

然后用它们来更新滤波器变量 \mathbf{x} 与 \mathbf{P} ：

$$\begin{aligned}\hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k \text{ (更新的状态估计)} \\ \mathbf{P}_{k|k} &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1} \text{ (更新的协方差估计)}\end{aligned}\quad (3-7)$$

3.2.3.3 不变

如果模型准确，而且 $\hat{\mathbf{x}}_{0|0}$ 与 $\mathbf{P}_{0|0}$ 的值准确的反映了最初状态的分布，那么以下不变量就保持不变：所有估计的误差均值为零。

- $\mathbb{E}[\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}] = \mathbb{E}[\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}] = 0$
- $\mathbb{E}[\tilde{\mathbf{y}}_k] = 0$
- $\mathbf{P}_{k|k} = cov(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})$
- $\mathbf{P}_{k|k-1} = cov(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})$
- $\mathbf{S}_k = cov(\tilde{\mathbf{y}}_k)$

其中， $\mathbb{E}[a]$ 表示 a 的期望值， $cov(\mathbf{a}) = \mathbb{E}[\mathbf{a}\mathbf{a}^T]$

3.2.4 经验

卡尔曼滤波器中， \mathbf{Q} 代表 Process Noise Covariance Matrix, \mathbf{R} 代表 Measurement Noise Covariance Matrix; \mathbf{R} 越小，estimation 的结果更”信任”测量值，反之， \mathbf{Q} 越小，estimation 的结果更”信任”预测值。

3.3 GPS 天线修正

由于 GPS 天线安装的位置不在船体重心处，我们需要对 GPS 测量结果作修正处理，才能得到重心处的运动。

第四章 决策与规划

运动规划（英语：Motion Planning）是一个过程，在考虑运动约束的同时寻找从起始状态到目标状态的移动步骤，其整体框架如图4-1所示

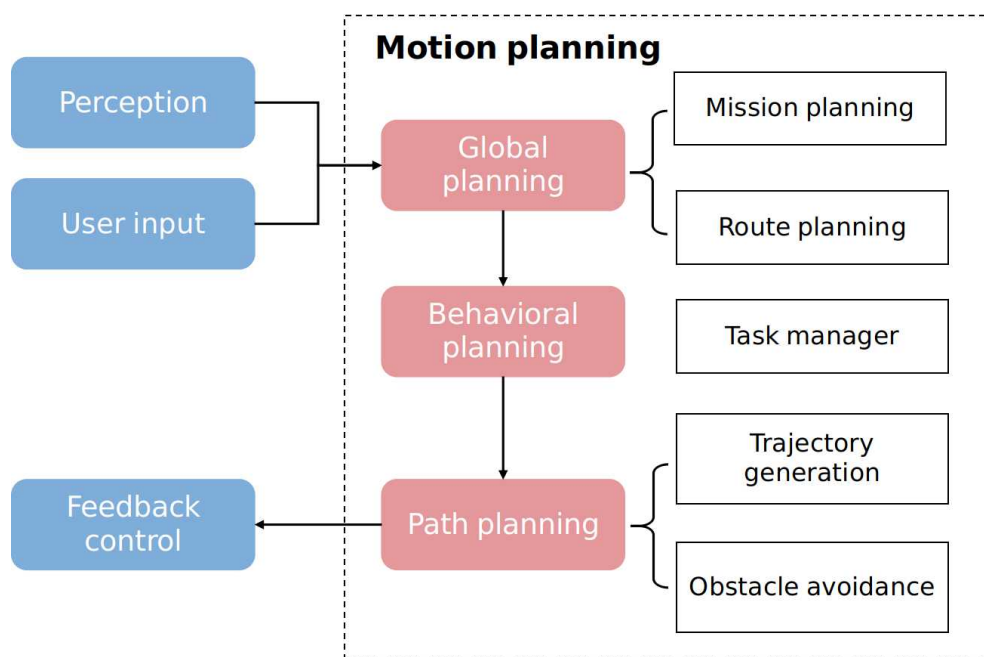


图 4-1 运动规划的整体框架

Figure 4-1 Sketch of motion planning

4.1 路径规划

4.1.1 航道保持

航道保持算法是无人船默认的局部路径规划算法，其适用于中高速航行的船舶，同时可实现多种规划目标（例如速度保持、超越、跟踪和停止等）。

4.1.1.1 Jerk

定理 4.1 给定初始时刻 t_0 的初始状态 $P_0 = [p_0, \dot{p}_0, \ddot{p}_0]$ 以及结束时刻 $t_1 = t_0 + T$ 的状态 $P_1 = [\dot{p}_1, \ddot{p}_1]$, 五次多项式是下列惩罚函数的最优解,

$$C = k_j J_t + k_t g(T) + k_p h(p_1) \quad (4-1)$$

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

其中, g 和 h 是任意函数, $k_j, k_t, k_p > 0$, J_t (Jerk) 通常用于描述加加速度的积分, 其与交通工具的舒适程度有关。

4.1.1.2 Frenet 坐标系

Frenet 坐标系被广泛用于轨迹跟踪和轨迹生成算法中^[5]。在 Frenet 坐标系中 (见图4-2), 我们以参考轨迹点为原点建立参考坐标系 (t_r, n_r) , 生成的轨迹点 $\mathbf{x}(s(t), d(t))$ 可表示为

$$\mathbf{x}(s(t), d(t)) = \mathbf{r}(s(t)) + d(t) \cdot \mathbf{n}_r(s(t)) \quad (4-2)$$

表4-1列出了常用的符号, 同时我们定义 $(\dot{\cdot}) := \frac{\partial}{\partial t}(\cdot)$ 和 $(\cdot)' := \frac{\partial}{\partial s}(\cdot)$

4.1.1.3 横向轨迹规划

• 中高速

给定横向的初始状态 $D_0 = [d_0, \dot{d}_0, \ddot{d}_0]$, 我们假设 $\dot{d}_1 = \ddot{d}_1 = 0$ (这也是我们期望的最终状态), 并假设 $g(T) = T$, $h(d_1) = d_1^2$, 由此可得

$$C_d = k_j J_t(d(t)) + k_t T + k_d d_1^2 \quad (4-3)$$

C_d 中最后一项用于惩罚最终状态 $d \neq 0$ 的情况。由定理4.1可得, $d(t)$ 是关于 t 的五次多项式 (quintic polynomials)。通过调整最终状态的 d_i 和 T_j

$$[d_1, \dot{d}_1, \ddot{d}_1, T]_{ij} = [d_i, 0, 0, T_j] \quad (4-4)$$

注意, 船体的状态是实时变化的, 因此 D_0 始终表示船体当前时刻的状态, 相应的 D_1 表示规划的最终状态, 也是实时变化的。

• 低速

表 4-1 符号解释

符号	物理含义
t	时间
T	时间间隔
\mathbf{r}	移动参考坐标系原点 (即参考轨迹点) 的笛卡尔坐标
s	参考轨迹点的弧坐标
d	生成轨迹点相对于参考轨迹点的横向偏移量
\mathbf{t}_r	移动参考坐标系的切向量
\mathbf{n}_r	移动参考坐标系的法向量
\mathbf{t}_x	生成轨迹点的切向量
\mathbf{n}_x	生成轨迹点的法向量
\mathbf{x}	生成轨迹的笛卡尔坐标
θ_r	参考轨迹点的方向角
κ_r	参考轨迹点的曲率
θ_x	生成轨迹点的方向角
κ_x	生成轨迹点的曲率
v_x	生成轨迹点的速度大小
a_x	生成轨迹点的加速度大小

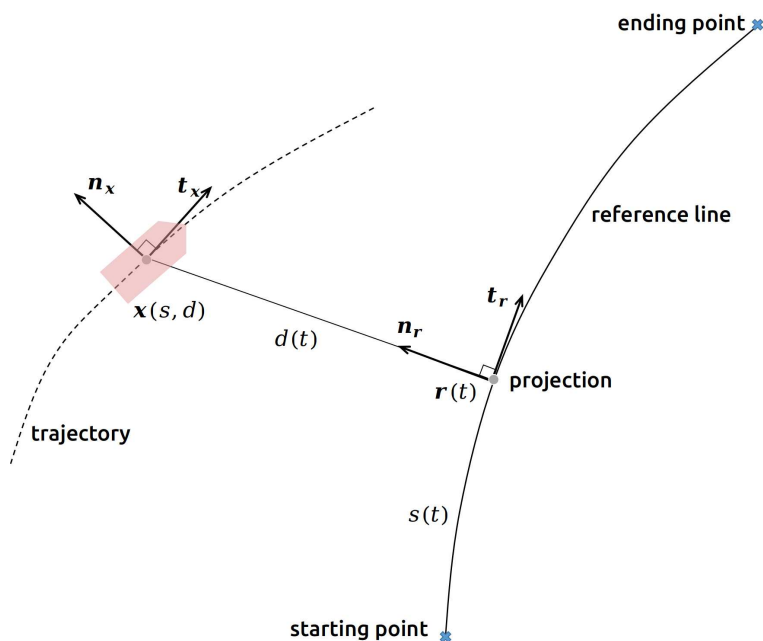


图 4-2 Frenet 坐标系下的轨迹生成

Figure 4-2 Trajectory generation in a Frenet frame

对于中高速的情况下, $s(t)$ 和 $d(t)$ 可以被分开计算; 但对于低速的情况, 对 $s(t)$ 和 $d(t)$ 独立处理的方法违背了刚体运动的 non-holonomic 的性质, 因此横向轨迹的计算需要考虑纵向轨迹

$$\mathbf{x}(s(t), d(t)) = \mathbf{r}(s(t)) + d(s(t)) \cdot \mathbf{n}_r(s(t)) \quad (4-5)$$

考虑低速的情形, 我们修改了惩罚函数

$$\begin{aligned} C_d &= k_j J_s(d(s)) + k_t S + k_d d_1^2 \\ J_s(d(s)) &:= \int_{s_0}^{s_1} d'''^2(\sigma) d\sigma \end{aligned} \quad (4-6)$$

该惩罚函数的最优值是 $d(s)$ 关于 s 的五次多项式, 且该多项式的初始点 $D_0 = [d_0, d'_0, d''_0]$ 和终止点为

$$[d_1, d'_1, d''_1, T]_{ij} = [d_i, 0, 0, T_j] \quad (4-7)$$

4.1.1.4 纵向轨迹规划

对于纵向的轨迹规划, 主要分为跟踪、超越、速度保持和停止。

• 速度保持

未感知到障碍物的时候, 我们希望船舶匀速航行。结合 s_1 的 Transversality condition^[6] 和定理4.1, 可知四次多项式 (quartic polynomials) 能够使得以下惩罚函数最小,

$$C_v = k_j J_t(s(t)) + k_t T + k_s [\dot{s}_1 - \dot{s}_d]^2 \quad (4-8)$$

其中, 我们给定 t_0 时刻 $S_0 = [s_0, \dot{s}_0, \ddot{s}_0]$, $t_1 = t_0 + T$ 时刻的状态 $S_1 = [\dot{s}_1, \ddot{s}_1]$ 。这就意味着, s 可以用 t 的四次多项式表示, 而多项式的初始条件分别为 $S_0 = [s_0, \dot{s}_0, \ddot{s}_0]$ 和以下终止条件

$$[\dot{s}_1, \ddot{s}_1, T]_{ij} = [\dot{s}_d + \Delta\dot{s}_i, 0, T_j] \quad (4-9)$$

当我们调整 $\Delta\dot{s}_i$ 和 T_j 时, 可得不同终止条件下的轨迹。

• 跟踪

对于跟踪、超越和停止的情况, 都存在目标的位置 $s_{target}(t)$, 给定了 $S_0 = [s_0, \dot{s}_0, \ddot{s}_0]$ 和以下的终止条件

$$[s_1, \dot{s}_1, \ddot{s}_1, T]_{ij} = [s_{target}(T_j) + \Delta s_i, \dot{s}_{target}(T_j), \ddot{s}_{target}(T_j), T_j] \quad (4-10)$$

可得 $s(t)$ 关于 t 的五次多项式是以下惩罚函数 (cost function) 的最优解,

$$C_t = k_j J_t + k_t T + k_s (s_1 - s_d)^2 \quad (4-11)$$

当我们调整终止条件中的 Δs_i 和 T_j , 可得多条纵向轨迹。

对于跟踪, 也就是与前方的船舶保持一定的距离, 这个距离也需要满足海事规范,

$$s_{target}(t) := s_{lv}(t) - (D_0 + \tau \dot{s}_{lv}(t)) \quad (4-12)$$

其中, D_0 和 τ 是常数, s_{lv} 和 \dot{s}_{lv} 分别是领导船只沿轨迹线的位置和速度。

• 超越和停止

$$s_{target}(t) = \frac{1}{2}[s_a(t) + s_b(t)] \quad (4-13)$$

其中 $s_a(t)$ 和 $s_b(t)$ 分别是周围两条船的位置。对于停止的情况, $s_{target} = s_{stop}$, $\dot{s}_{target} = 0$, $\ddot{s}_{target} = 0$

4.1.1.5 轨迹结合

通过调整终止条件, 我们可得横向和纵向的多项式线束 (Lattice), Π_{lat} 和 Π_{lon} , 从而生成轨迹线束 $\Pi_{lat} \times \Pi_{lon}$, 如图4-3所示。在选取最优轨迹的过程中, 我们考虑联合惩罚函数 $C_{tot} = k_{lat} C_{lat} + k_{lon} C_{lon}$, 同时保证生成轨迹与障碍物有一定的安全距离。最终保证避障的条件下, 选取最优轨迹。

4.1.1.6 坐标转换

计算轨迹线束时, 需要 Frenet 坐标系和 Cartesian 坐标系的相互转换。Frenet 坐标系下的船体状态为 $[s, \dot{s}, \ddot{s}; d, \dot{d}, \ddot{d}/d, d', d'']$ 和 $[\mathbf{x}, \theta_x, \kappa_x, v_x, a_x]$, 首先我们推导一些通用的公式。

我们定义 $\mathbf{t}_r(s) := [\cos(\theta_r(s)) \ \sin(\theta_r(s))]^T$, $\mathbf{n}_r(s) := [-\sin(\theta_r(s)) \ \cos(\theta_r(s))]^T$ 和 $\Delta\theta := \theta_x - \theta_r$, 且确保 $|\Delta\theta| < \frac{\pi}{2}$ 和 $1 - \kappa_r d > 0$ 。由 Frenet-Serret 公式可得

$$\begin{aligned} \frac{d\mathbf{n}_r}{ds} &= -\kappa_r \mathbf{t}_r, & \frac{d\mathbf{n}_r}{dt} &= -\dot{s} \kappa_r \mathbf{t}_r \\ \frac{d\mathbf{t}_r}{ds} &= \kappa_r \mathbf{n}_r, & \frac{d\mathbf{t}_r}{dt} &= \dot{s} \kappa_r \mathbf{n}_r \end{aligned} \quad (4-14)$$

结合 Frenet-Serret 公式, 由方程4-2可得

$$d = [\mathbf{x} - \mathbf{r}(s)]^T \cdot \mathbf{n}_r \quad (4-15)$$

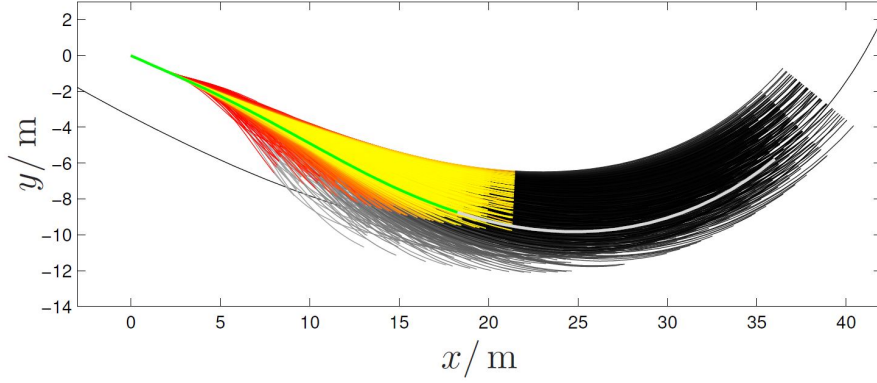


图 4-3 Frenet 坐标系下的最优轨迹线束, 其预测时间为 3 秒, 红色到黄色代表递增的惩罚值 (cost), 当障碍物不存在时, “绿-灰”的轨迹即为最优轨迹, 使得船体回归参考轨迹。

Figure 4-3 Resulting trajectory set in global coordinates for velocity keeping: The color map visualizes the increasing costs of both the reactive layer with 3.0 s lookahead from red to yellow and the alternatives for the long-term objectives from gray to black. As there are no obstacles within the 3.0 s horizon, the optimal trajectory of the free problem is chosen (green, light gray), which leads the vehicle back to the center line and to the desired speed.

对上式作时间的导数, 可得

$$\begin{aligned}
 \dot{d} &= [\dot{\mathbf{x}} - \dot{\mathbf{r}}(s)]^T \mathbf{n}_r + [\mathbf{x} - \mathbf{r}(s)]^T \dot{\mathbf{n}}_r \\
 &= v_x \mathbf{t}_x^T \mathbf{n}_r - \dot{s} \mathbf{t}_r^T \mathbf{n}_r - \dot{s} \kappa_r [\mathbf{x} - \mathbf{r}(s)]^T \mathbf{t}_r \\
 &= v_x \sin(\Delta\theta)
 \end{aligned} \tag{4-16}$$

同时, 我们得到

$$\begin{aligned}
 v_x &= \|\dot{\mathbf{x}}\|_2 \\
 &= \|\dot{\mathbf{r}}(s) + \dot{d} \mathbf{n}_r + d \dot{\mathbf{n}}_r\|_2 \\
 &= \|\dot{s} \mathbf{t}_r + \dot{d} \mathbf{n}_r - \dot{s} \kappa_r d \mathbf{t}_r\|_2 \\
 &= \left\| \begin{bmatrix} \mathbf{t}_r & \mathbf{n}_r \end{bmatrix} \begin{bmatrix} 1 - \kappa_r d & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{s} \\ \dot{d} \end{bmatrix} \right\|_2 \\
 &= \sqrt{(1 - \kappa_r d)^2 \dot{s}^2 + \dot{d}^2}
 \end{aligned} \tag{4-17}$$

且

$$\begin{aligned}
 d' &= \frac{dt}{ds} \frac{d}{dt} d = \frac{\dot{d}}{\dot{s}} = \frac{1}{\dot{s}} v_x \sin(\Delta\theta) \\
 &= \sqrt{(1 - \kappa_r d)^2 + d'^2} \sin(\Delta\theta) \\
 &\Rightarrow \\
 d' &= (1 - \kappa_r d) \tan(\Delta\theta)
 \end{aligned} \tag{4-18}$$

已知 $(\mathbf{x} - \mathbf{r}(s))^T \mathbf{t}_r = 0$, 对其作时间导数可得 $\frac{v_x}{\dot{s}} \cos(\Delta\theta) - 1 + \kappa_r d = 0$, 变化得到船体的速度大小

$$v_x = \dot{s} \frac{1 - \kappa_r d}{\cos(\Delta\theta)} \tag{4-19}$$

我们用 s_x 表示轨迹 \mathbf{x} 上的弧长, 根据曲率的定义 $\kappa_x = \frac{d\theta_x}{ds_x}$ 和 $\kappa_r = \frac{d\theta_r}{ds}$, 可得

$$\frac{d}{ds} = \frac{ds_x}{dt} \frac{dt}{ds} \frac{d}{ds_x} = \frac{v_x}{\dot{s}} \frac{d}{ds_x} = \frac{1 - \kappa_r d}{\cos(\Delta\theta)} \frac{d}{ds_x} \tag{4-20}$$

$$\begin{aligned}
 \Delta\theta' &= \frac{d(\theta_x - \theta_r)}{ds} = \frac{d}{ds} \theta_x - \kappa_r \\
 &= \frac{1 - \kappa_r d}{\cos(\Delta\theta)} \kappa_x - \kappa_r
 \end{aligned} \tag{4-21}$$

由公式4-18, 我们可得 d 对 s 的二阶导数

$$d'' = -(\kappa_r d)' \tan(\Delta\theta) + \frac{1 - \kappa_r d}{\cos^2(\Delta\theta)} \Delta\theta' \tag{4-22}$$

将速度 v_x 对时间求导可得

$$a_x := \dot{v}_x = \ddot{s} \frac{1 - \kappa_r d}{\cos(\Delta\theta)} + \frac{\dot{s}^2}{\cos(\Delta\theta)} [d' \Delta\theta' - (\kappa_r d)'] \tag{4-23}$$

同时, s 对时间 t 的二阶导数

$$\ddot{s} = \frac{a_x \cos(\Delta\theta) - \dot{s}^2 [d' \Delta\theta' - (\kappa_r d)']}{1 - \kappa_r d} \tag{4-24}$$

在中高速的情况下 (d 与 s 无关),

$$\begin{aligned}
 \dot{d} &= \frac{d}{dt} d = \frac{ds}{dt} \frac{d}{ds} d = \dot{s} d' \\
 \ddot{d} &= d'' \dot{s}^2 + d' \ddot{s}
 \end{aligned} \tag{4-25}$$

坐标系之间的转换将会用于 Frenet Lattice 的实时计算中 (如图4-4), 下面我们分别讨论两个坐标系之间的转换过程

• 从 Cartesian 到 Frenet

给定船体当前时刻的状态为 $[\mathbf{x}, \theta_x, \kappa_x, v_x, a_x](t_0)$, 我们希望得到在 Frenet 坐标系下, 当前时刻的状态 $[s_0, \dot{s}_0, \ddot{s}_0, d_0, \dot{d}_0, \ddot{d}_0]$ 或者 $[s_0, \dot{s}_0, \ddot{s}_0, d_0, d'_0, d''_0]$ 。

$$[\mathbf{x}, \theta_x, \kappa_x, v_x, a_x] \rightarrow [s, \dot{s}, \ddot{s}; d, \dot{d}, \ddot{d}/d', d''] \quad (4-26)$$

首先, 采用一些高效的数值方法得到船体在参考轨迹上的投影点 s

$$s = \arg \min_{\sigma} \|\mathbf{x} - \mathbf{r}(\sigma)\| \quad (4-27)$$

得到 s 之后, 根据参考轨迹的多项式公式, 可得 κ_r, κ'_r 和 θ_r , 从而计算 $\Delta\theta = \theta_x - \theta_r$, 剩下的变量也可以根据以上公式推导得到。

• 从 Frenet 到 Cartesian

当我们得到生成轨迹 $[s, \dot{s}, \ddot{s}; d, \dot{d}, \ddot{d}/d', d'']$ 的时候, 需要得出 Cartesian 坐标系中, 生成轨迹的状态

$$[s, \dot{s}, \ddot{s}; d, \dot{d}, \ddot{d}/d', d''] \rightarrow [\mathbf{x}, \theta_x, \kappa_x, v_x, a_x] \quad (4-28)$$

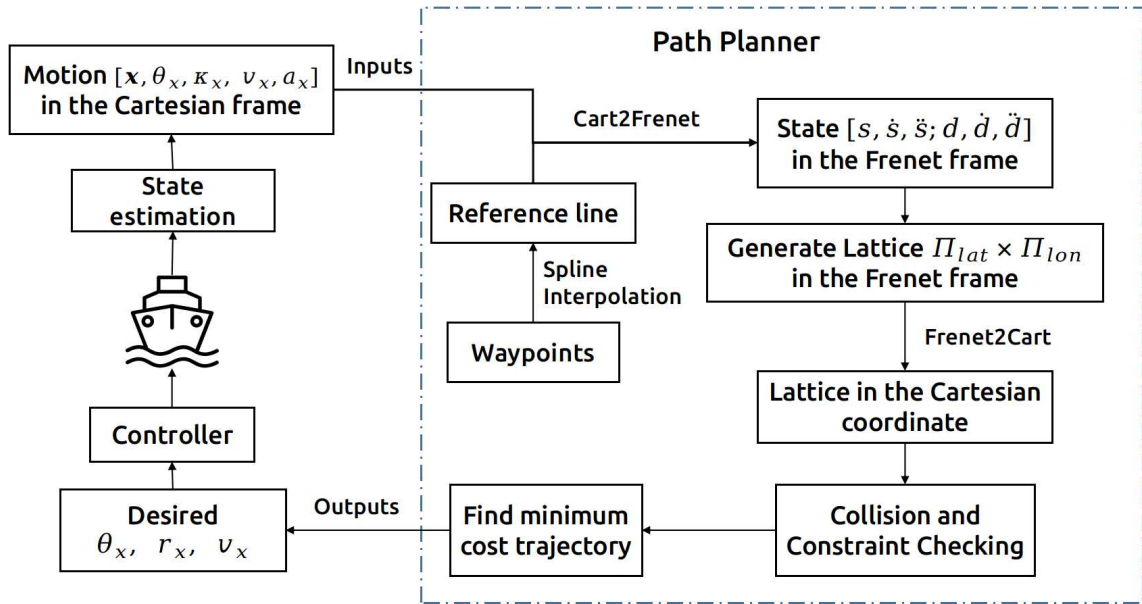


图 4-4 Frenet 坐标系下的最优轨迹生成算法逻辑图

Figure 4-4 Flowchart for trajectory generator in the Frenet frame

4.1.2 低速泊船

自动停车是一种自动驾驶汽车的系统, 它将车辆从行车道移入停车位, 以执行平行, 垂直或角度停车。适用于低速。

4.1.2.1 Reeds-Shepp 曲线

对于欠驱动的机器人 (nonholonomic system), 其运动模型为

$$\begin{aligned} x(t) &= x(0) + \int_0^t \varepsilon(\tau) \cos \phi(\tau) d\tau \\ y(t) &= y(0) + \int_0^t \varepsilon(\tau) \sin \phi(\tau) d\tau \\ \phi(t) &= \phi(0) + \int_0^t \eta(\tau) d\tau \end{aligned} \quad (4-29)$$

其中, $\gamma(t) = [x(t), y(t), \phi(t)]$ 为机器人的可行轨迹。

如何找到一条最短的路径把船从一个地方开到另外一个地方。答案就是图4-8所示的 Reeds-Shepp 曲线^[7]。假设车辆能以固定的半径转向, 且车辆能够前进和后退, 那么 Reeds-Shepp 曲线就是车辆在上述条件下从起点到终点的最短路径。该曲线不仅能保证车辆能够到达终点, 而且能保证车辆的角度能在终点到达预期角度。

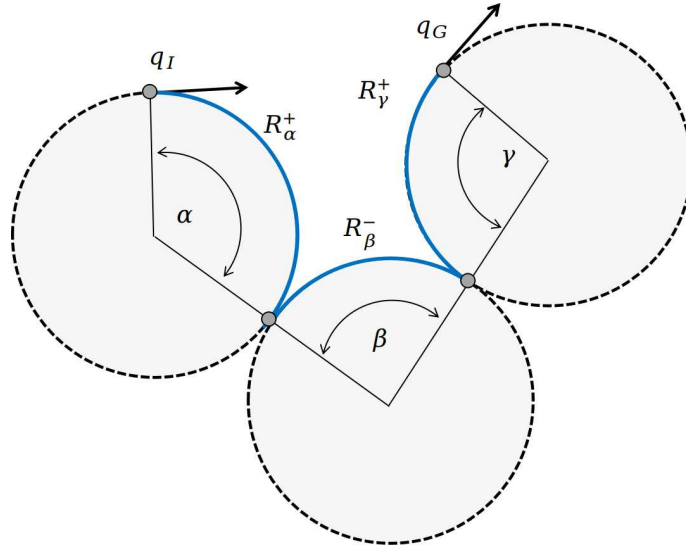


图 4-5 Reeds-Shepp 曲线

Figure 4-5 An example of Reeds-Shepp curve

4.1.2.2 碰撞检测

我们用矩形包络船体, 且用点、线段和最小边框 (bounding box) 表示周围的物体, 从而通过检测点、线段和最小边框是否与船体矩形重合来检测碰撞。

4.1.2.3 A* 算法

A* 搜索算法是一种在图形平面上，有多个节点的路径，求出最低通过成本的算法。用于游戏中的 NPC 的移动计算，或网络游戏的 BOT 的移动计算上。该算法综合了最良优先搜索和 Dijkstra 算法的优点：在进行启发式搜索提高算法效率的同时，可以保证找到一条最优路径（基于评估函数）。在此算法中，如果以 $g(n)$ 表示从起点到任意顶点 n 的实际距离， $h(n)$ 表示任意顶点 n 到目标顶点的估算距离（根据所采用的评估函数的不同而变化），那么 A* 算法的估算函数为：

$$f(n) = g(n) + h(n) \quad (4-30)$$

这个公式遵循以下特性：

- 如果 $g(n)$ 为 0，即只计算任意顶点 n 到目标的评估函数 $h(n)$ ，而不计算起点到顶点 n 的距离，则算法转化为使用贪心策略的最良优先搜索，速度最快，但可能得不出最优解；
- 如果 $h(n)$ 不大于顶点 n 到目标顶点的实际距离，则一定可以求出最优解，而且 $h(n)$ 越小，需要计算的节点越多，算法效率越低，常见的评估函数有——欧几里得距离、曼哈顿距离、切比雪夫距离；
- 如果 $h(n)$ 为 0，即只需求出起点到任意顶点 n 的最短路径 $g(n)$ ，而不计算任何评估函数 $h(n)$ ，则转化为单源最短路径问题，即 Dijkstra 算法，此时需要计算最多的顶点；

4.1.2.4 混合 A* 搜索

混合 A* 搜索算法 (Hybrid A* search algorithm)，如图4-6所示，如图4-7所示，

$$\Delta\theta_{max} = \kappa_{max} L$$

$$\begin{aligned} \Delta x &= \frac{L}{\Delta\theta_{max}} (\sin \Delta\theta_{max} \cos \theta - (1 - \cos \Delta\theta_{max}) \sin \theta) \\ \Delta y &= \frac{L}{\Delta\theta_{max}} (\sin \Delta\theta_{max} \sin \theta + (1 - \cos \Delta\theta_{max}) \cos \theta) \end{aligned} \quad (4-31)$$

4.1.2.5 轨迹平滑

由 A* 算法生成的轨迹比较粗糙，需要通过平滑处理获得更好的轨迹。我们假定轨迹点为 $\mathbf{x}_i = [x_i, y_i]$ ，如图4-8 所示，其中起点、终点和方向转换点保持不变

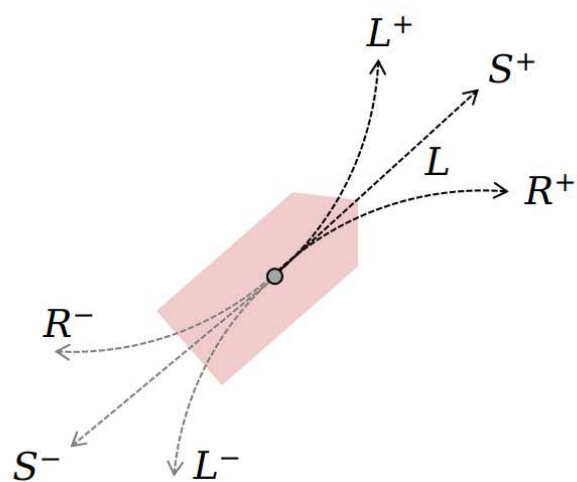


图 4-6 混合 A* 搜索算法中的可行操作模式

Figure 4-6 Maneuver set of hybrid A* search algorithm

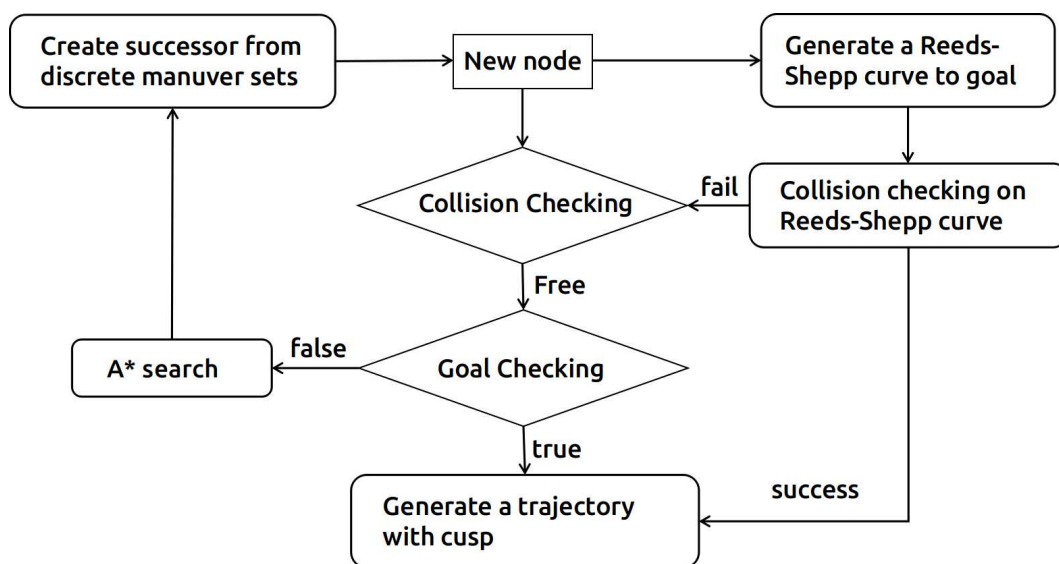


图 4-7 混合 A* 搜索算法

Figure 4-7 Hybrid A* search algorithm

$\Delta \mathbf{x}_s = 0, \Delta \mathbf{x}_{sw} = 0, \Delta \mathbf{x}_e = 0$, 剩下的轨迹点采用以下惩罚函数

$$F(\mathbf{x}) = \omega_s \sum_{i=1}^{e-1} \|\Delta \mathbf{x}_{i+1} - \Delta \mathbf{x}_i\|_2^2 + \omega_o \sum_{i=1}^{e-1} \sum_{j=1}^{N_i} (\|\mathbf{x}_i - \mathbf{o}_{ij}\|_2 - d_{max})^2 + \omega_k \sum_{i=1}^{e-1} \left(\frac{\Delta \Phi_i}{\|\Delta \mathbf{x}_i\|_2} - \kappa_{max} \right)^2 \quad (4-32)$$

其中, \mathbf{o}_{ij} 为轨迹点 \mathbf{x}_i 在半径 d_{max} 之内的障碍物位置, 两点之间的位移 $\Delta \mathbf{x}_i = \mathbf{x}_i - \mathbf{x}_{i-1}$ 和夹角

$$\begin{aligned} \Delta \Phi_i &= \left| \arctan \frac{\Delta y_{i+1}}{\Delta x_{i+1}} - \arctan \frac{\Delta y_i}{\Delta x_i} \right| \\ &= \arccos \frac{\Delta \mathbf{x}_i^T \Delta \mathbf{x}_{i+1}}{\|\Delta \mathbf{x}_i\|_2 \|\Delta \mathbf{x}_{i+1}\|_2} \end{aligned} \quad (4-33)$$

F_o 用于远离障碍物

$$\frac{\partial F_o}{\partial \mathbf{x}_i} = 2\omega_o \sum_{j=1}^{N_i} (\|\mathbf{x}_i - \mathbf{o}_{ij}\|_2 - d_{max}) \frac{\mathbf{x}_i - \mathbf{o}_{ij}}{\|\mathbf{x}_i - \mathbf{o}_{ij}\|_2} \quad (4-34)$$

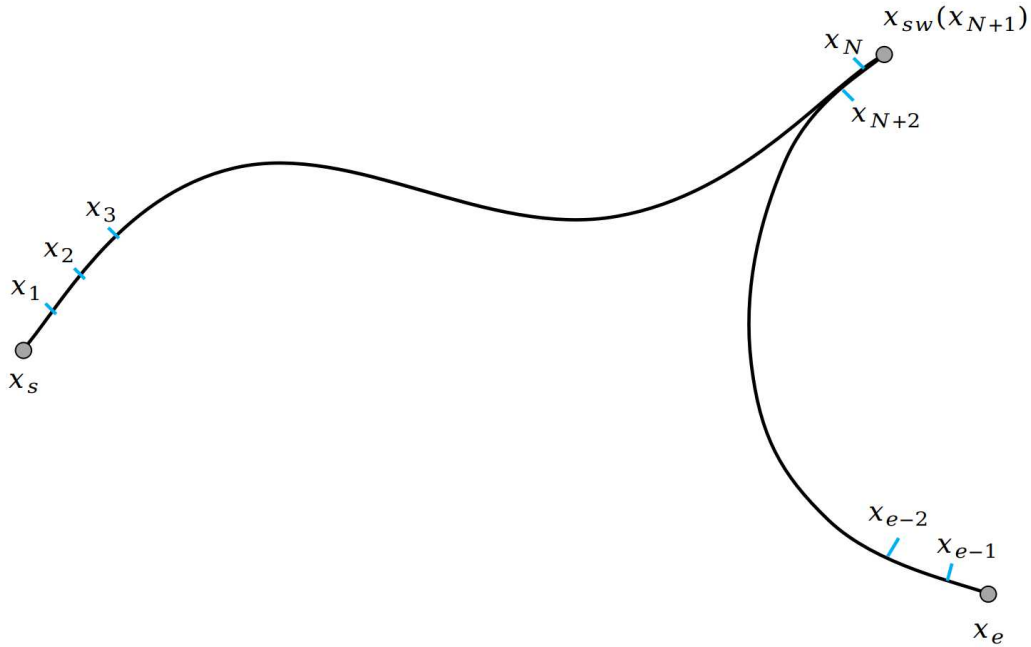


图 4-8 轨迹平滑处理

Figure 4-8 Trajectory Smoothing

对于 F_s ，表示轨迹的光滑性；

$$\begin{aligned}\hat{\mathbf{x}}_i &= \Delta \mathbf{x}_{i-1} - 2\Delta \mathbf{x}_i + \Delta \mathbf{x}_{i+1} \\ \frac{\partial F_s}{\partial \mathbf{x}_i} &= 2\omega_s (\hat{\mathbf{x}}_{i-1} - 2\hat{\mathbf{x}}_i + \hat{\mathbf{x}}_{i+1})\end{aligned}\quad (4-35)$$

其中，起点 $\hat{\mathbf{x}}_{i=s} = 0$ 和终点 $\hat{\mathbf{x}}_{i=e} = 0$

F_κ 用于惩罚过大的曲率，我们用 $\kappa_i = \frac{\Delta \Phi_i}{\|\Delta \mathbf{x}_i\|_2}$ 表示轨迹点的曲率

$$\begin{aligned}\frac{\partial \kappa_i}{\partial \mathbf{x}_i} &= \frac{1}{\|\Delta \mathbf{x}_i\|_2} \frac{\partial \Delta \Phi_i}{\partial \cos(\Delta \Phi_i)} \frac{\partial \cos(\Delta \Phi_i)}{\partial \mathbf{x}_i} - \frac{\Delta \Phi_i}{\|\Delta \mathbf{x}_i\|_2^2} \frac{\Delta \mathbf{x}_i}{\|\Delta \mathbf{x}_i\|_2} \\ \frac{\partial \kappa_i}{\partial \mathbf{x}_{i-1}} &= \frac{1}{\|\Delta \mathbf{x}_i\|_2} \frac{\partial \Delta \Phi_i}{\partial \cos(\Delta \Phi_i)} \frac{\partial \cos(\Delta \Phi_i)}{\partial \mathbf{x}_{i-1}} + \frac{\Delta \Phi_i}{\|\Delta \mathbf{x}_i\|_2^2} \frac{\Delta \mathbf{x}_i}{\|\Delta \mathbf{x}_i\|_2} \\ \frac{\partial \kappa_i}{\partial \mathbf{x}_{i+1}} &= \frac{1}{\|\Delta \mathbf{x}_i\|_2} \frac{\partial \Delta \Phi_i}{\partial \cos(\Delta \Phi_i)} \frac{\partial \cos(\Delta \Phi_i)}{\partial \mathbf{x}_{i+1}}\end{aligned}\quad (4-36)$$

其中

$$\frac{\partial \Delta \Phi_i}{\partial \cos(\Delta \Phi_i)} = \frac{-1}{(1 - \cos^2(\Delta \Phi_i))^{1/2}} \quad (4-37)$$

我们引入正交补 (orthogonal complements)

$$\mathbf{a} \perp \mathbf{b} = \mathbf{a} - \frac{\mathbf{a}^T \mathbf{b}}{|\mathbf{b}|} \frac{\mathbf{b}}{|\mathbf{b}|} \quad (4-38)$$

用于表示

$$\mathbf{P}_1 = \frac{\Delta \mathbf{x}_i \perp (-\Delta \mathbf{x}_{i+1})}{\|\Delta \mathbf{x}_i\|_2 \|\Delta \mathbf{x}_{i+1}\|_2}; \quad \mathbf{P}_2 = \frac{(-\Delta \mathbf{x}_{i+1}) \perp \Delta \mathbf{x}_i}{\|\Delta \mathbf{x}_i\|_2 \|\Delta \mathbf{x}_{i+1}\|_2} \quad (4-39)$$

从而

$$\frac{\partial \cos(\Delta \Phi_i)}{\partial \mathbf{x}_i} = -\mathbf{P}_1 - \mathbf{P}_2; \quad \frac{\partial \cos(\Delta \Phi_i)}{\partial \mathbf{x}_{i-1}} = \mathbf{P}_2; \quad \frac{\partial \cos(\Delta \Phi_i)}{\partial \mathbf{x}_{i+1}} = \mathbf{P}_1. \quad (4-40)$$

$$\Delta \hat{\Phi}_i = \frac{\Delta \Phi_i}{\|\mathbf{x}_i - \mathbf{x}_{i-1}\|_2}; \quad \hat{P}_1^i = \frac{P_1^i}{\|\mathbf{x}_i - \mathbf{x}_{i-1}\|_2 \sin \Delta \Phi_i}; \quad \hat{P}_2^i = \frac{P_2^i}{\|\mathbf{x}_i - \mathbf{x}_{i-1}\|_2 \sin \Delta \Phi_i}. \quad (4-41)$$

对于起点 ($i = s$) 和终点 ($i = e$), $\Delta\hat{\Phi}_i = 0, \hat{P}_1^i = \hat{P}_2^i = 0$ 。

$$\begin{aligned} \frac{\partial F_k}{\partial \mathbf{x}_i} = 2\omega_k \left[(\Delta\hat{\Phi}_{i-1} - \kappa_{max}) (-\hat{P}_1^{i-1}) + \right. \\ (\Delta\hat{\Phi}_i - \kappa_{max}) \left(\hat{P}_1^i + \hat{P}_2^i - \Delta\hat{\Phi}_i \frac{\mathbf{x}_i - \mathbf{x}_{i-1}}{\|\mathbf{x}_i - \mathbf{x}_{i-1}\|_2^2} \right) + \\ \left. (\Delta\hat{\Phi}_{i+1} - \kappa_{max}) \left(-\hat{P}_1^{i+1} + \Delta\hat{\Phi}_{i+1} \frac{\mathbf{x}_{i+1} - \mathbf{x}_i}{\|\mathbf{x}_{i+1} - \mathbf{x}_i\|_2^2} \right) \right] \end{aligned} \quad (4-42)$$

当我们得到了当前轨迹的梯度 $\nabla F(\mathbf{x})$, 使得下降方向 $\Delta\mathbf{x} = -\nabla F(\mathbf{x})$, 需要选择一个 t , 使得 $F(\mathbf{x} + t\Delta\mathbf{x}) < F(\mathbf{x})$, 这时可采用回溯线搜索 (backtracking line search) 计算得到该 t 。给定 $\alpha \in (0, 1/2), \beta \in (0, 1)$, 初始 $t = 1$, 重复 $t := \beta t$, 直到

$$F(\mathbf{x} + t\Delta\mathbf{x}) < F(\mathbf{x}) + \alpha t \nabla F(\mathbf{x})^T \Delta\mathbf{x} \quad (4-43)$$

4.2 航线规划

A* 算法

4.3 行为规划

第五章 环境感知

5.1 航海雷达

5.1.1 SDK

本节介绍一些常用航海雷达 (e.g. SIMRAD 4G) 的 SDK, 以及一些使用注意事项

5.1.2 目标跟踪

5.1.2.1 坐标投影

如图5-1所示, 我们用 θ 表示船体的首向角, θ_s 表示线束的方位角, $\overline{GR} = (x_r, y_r)$ 表示航海雷达在船上的位置。在随体坐标系中, $\overline{GS} = (x_r + \rho \cos \theta_s, y_r + \rho \sin \theta_s)$; 则在全局坐标系下, 线束的位置可表示如下

$$\begin{bmatrix} x_s \\ y_s \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x_r + \rho \cos \theta_s \\ y_r + \rho \sin \theta_s \end{bmatrix} + \begin{bmatrix} x_G \\ y_G \end{bmatrix} \quad (5-1)$$

5.1.2.2 分类 Clustering

5.1.2.3 包络 Miniball

5.1.2.4 识别 Identification

当我们得到发现的目标时 (Detected targets), 需要识别这些目标, 我们用以下符号表示

- V_{j0} : the previous velocity vector of j th tracking target
- V_{ji} : The predicted velocity vector of j th tracking target when applying the new position of i th detected target
- R_D^i : the radius of i th detected target
- R_T^j : the radius of j th tracking target
- V_T : the speed threshold of targets

对于跟踪中的目标 j , 他们是 acquiring 或 acquired, 我们可以

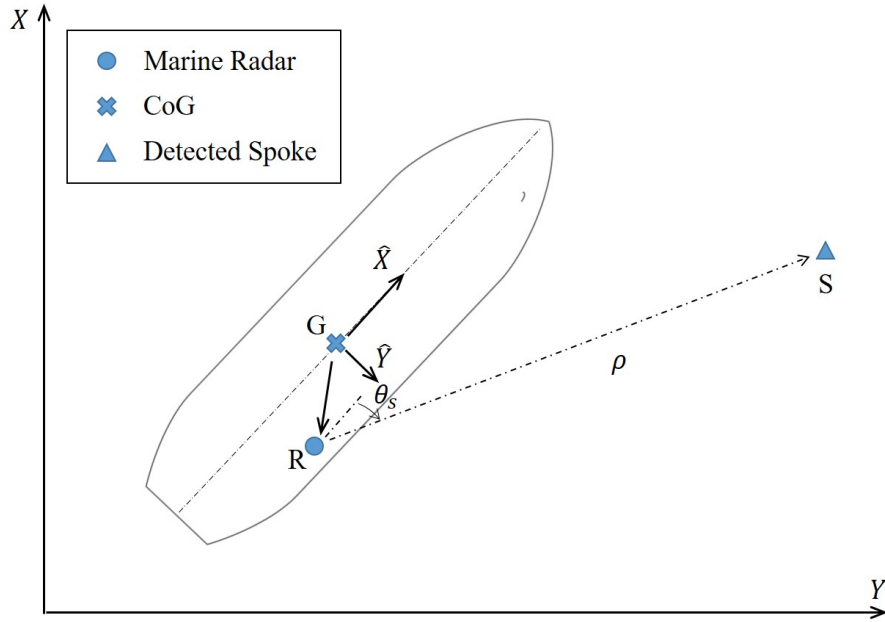


图 5-1 雷达线束的坐标投影

Figure 5-1 The coordinate projection of spoke data from marine radar

当 $|V_{j0}| \leq V_T$

$$\text{Loss} = k_R \frac{|R_D^i{}^2 - R_T^j{}^2|}{R_T^j{}^2} + k_V \frac{|V_{j0} - V_{ji}|}{|V_{j0}|} \quad (5-2)$$

For loop i in detected targets:

If $|V_{ji}| > V_{max}$, continue;

If $|V_{j0} - V_{ji}| > a_{max}\Delta T$, continue;

find the min Loss;

当 $|V_{j0}| > V_T$

$$\text{Loss} = k_R \frac{|R_D^i{}^2 - R_T^j{}^2|}{R_T^j{}^2} + k_V \frac{|V_{j0} - V_{ji}|}{|V_{j0}|} + k_A \frac{|\angle(V_{j0}, V_{ji})|}{\pi} \quad (5-3)$$

For loop i in detected targets:

If $|V_{ji}| > V_{max}$, continue;

If $|V_{j0} - V_{ji}| > a_{max}\Delta T$, continue;

If $|\angle(V_{j0}, V_{ji})| > \alpha_{max}\Delta T$, continue;

find the min Loss;

The identification for j tracking target, return the matched index in the detected targets, or return -1 if un-matched.

5.1.2.5 $\alpha - \beta$ 滤波

$\alpha - \beta$ 滤波器是一种线性状态观测器，可用于观测刚体运动的位置和速度。当我们假设在一段时间内 ΔT 物体的速度是恒定不变的，则

$$\begin{aligned}\hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_{k-1} + \Delta T \hat{\mathbf{v}}_{k-1} \\ \hat{\mathbf{v}}_k &= \hat{\mathbf{v}}_{k-1}\end{aligned}\tag{5-4}$$

5.2 计算机视觉

5.3 激光雷达

第六章 传感器

6.1 全球定位系统

全球定位系统 (Global Positioning System, GPS) 是一种以空中卫星为基础的高精度无线电导航的定位系统, 它在全球任何地方以及近地空间都能够提供准确的地理位置、刚体速度及精确的时间信息。对于室外定位, 一般采用 GPS 或者北斗定位系统。

6.1.1 GPS 定位定向天线 Hemisphere V102

Hemisphere V102 提供定向、定位、起伏、横滚和俯仰信息, 可利用 SBAS 差分 GPS 定位, 水平精度为 0.5 m; 侧向精度为 0.75°, 俯仰/横滚精度为 1.5°。使用时推荐采样频率 10 Hz, 波特率为 115200。在串口助手里面 (cutecom 或者 sscom) 可输入指令 (见表6-1), 修改 V102 输出的内容。由于 V102 不支持北斗功能, 与北斗相关的命令是无效的。每次设置完 GPS 之后, 记得 \$JSAVE 保存配置

表 6-1 Hemisphere V102 常用指令

命令	功能
\$JSHOW	查询 V102 的设置
\$JDIFF	查询当前差分模式
\$JDIFF, INCLUDE, SBAS	设置 SBAS 参与差分定位
\$JBAUD, X	设置当前串口波特率为 X, 推荐 X=115200
\$JASC, GPGGA, X	设置 NMEA GPGGA 消息的输出频率为 X Hz; 若 X=0, 不输出 GPGGA 的数据; 推荐 X=10
\$JASC, GPGSV, X	设置 NMEA GPGSV 消息的输出频率为 X Hz
\$JASC, GPHDT, X	设置 NMEA GPHDT 消息的输出频率为 X Hz
\$JASC, GPHPR, X	设置 NMEA GPGSV 消息的输出频率为 X Hz
\$JASC, GPROT, X	首向速度 (deg/min)
\$JASC, PSAT, ATTSTAT, X	打开副天线卫星信息, 同时包含基线长等信息
\$JASC, PVCT, X	输出主副天线的东向, 北向, 及天向的差值
\$JATT, GYROAID, YES	打开陀螺
\$JATT, GYROAID	查询陀螺当前状态
\$JATT, TILTAID, YES	打开倾斜传感器
\$JATT, TILTAID	查询当前倾斜传感器的状态

表 6-2 防水接头接线方法

Pin	颜色	Pin(*)	功能
1	白	1	Port C, RS-232 female DB9 pin 2, device out
2	绿	3	Port C, RS-232 female DB9 pin 3, device in
5	红	5	Power input
7	黄	2	Signal Ground
8	棕	7	Port A, RS-232 female DB9 pin 3, device in
9	蓝	6	Port A, RS-232 female DB9 pin 2, device out
10	黑	4	Power Ground
11	Drain	8	CH_GND

注: Pin 指的是 V102 上的 12 芯防水接头上的数字, Pin(*) 指的是控制盒上 8 芯防水接头上的数字; 在控制盒中, Signal Ground 一分为二, 黄色和灰色线共用。

6.1.2 墨卡托投影

墨卡托投影, 是正轴等角圆柱投影。由荷兰地图学家墨卡托 (G.Mercator) 于 1569 年创立。假想一个与地轴方向一致的圆柱切或割于地球, 按等角条件, 将经纬网投影到圆柱面上, 将圆柱面展为平面后, 即得本投影。墨卡托投影在切圆柱投影与割圆柱投影中, 最早也是最常用的是切圆柱投影。首先, 我们来了解一些常用的导航坐标系

- 地理坐标系 (Geographic coordinates) 地理坐标系一般是指由经度、纬度和相对高度组成的坐标系, 能够标示地球上的任何一个位置。
- 通用横轴墨卡托投影 (Universal Transverse Mercator, 简称 UTM)^[8] 是一种国际标准化的地图投影法。它使用笛卡尔坐标系, 标记南纬 80° 至北纬 84° 之间的所有位置。本坐标系采用 WGS84 系统作为坐标基础。
- 通用极立体坐标系 (Universal polar stereographic coordinate system, 简称 UPS) 作为 UTM 的补充, 标记南纬 80° 以南和北纬 84° 以北的所有位置, 以及与 UTM 有 0.5° 的重叠。

值得注意的是, 跨越 UTM 区域的时候会出现坐标不连续的问题。这里我们在 UTM 区域之间引入 100 公里 (100 km) 的重叠区 (如图6-1), 从而解决坐标不连续的问题。同时, 在路径规划算法中, 用户输入的坐标点需要经过计算, 确保 Waypoints 或者 DP setpoint 在同一个 UTM 区域且不超过 100km。

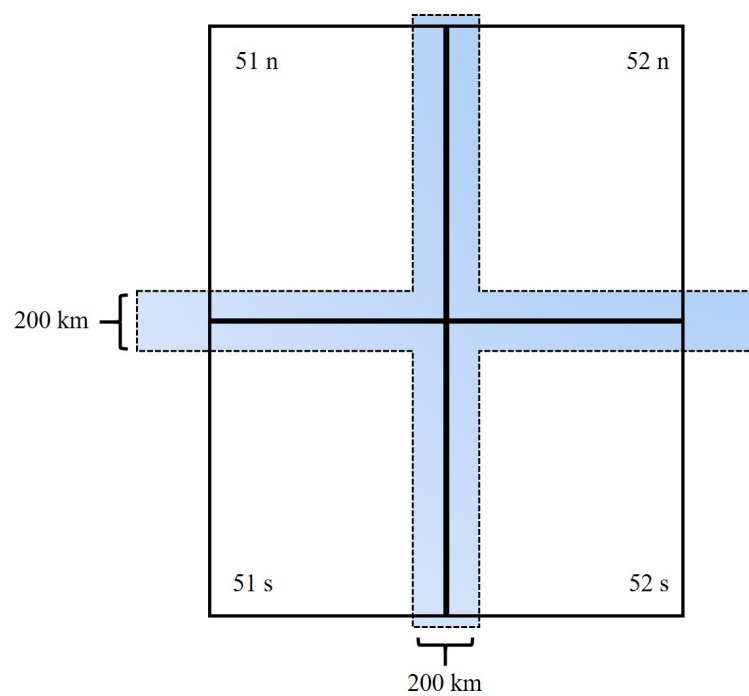


图 6-1 UTM 区域的重叠

Figure 6-1 The overlap between UTM zones

6.2 惯性测量单元

惯性测量单元 (Inertial measurement unit, 简称 IMU 是测量物体三轴姿态角 (或角速率) 以及加速度的装置。

6.3 航海雷达

6.4 红外热像仪

6.5 风速风向仪

第七章 系统架构

7.1 软件系统架构

7.2 电气系统架构

7.2.1 SDK

第八章 数学与引用文献的标注

8.1 数学

8.1.1 数字和单位

宏包 siunitx 提供了更好的数字和单位支持：

- 12 345.678 90
- $1 \pm 2i$
- 0.3×10^{45}
- $1.654 \times 2.34 \times 3.430$
- $\text{kg} \cdot \text{m} \cdot \text{s}^{-1}$
- $\mu\text{m} \mu\text{m}$
- $\Omega \Omega$
- 10 和 20
- 10, 20 和 30
- 0.13 mm, 0.67 mm 和 0.80 mm
- $10 \sim 20$
- $10^\circ\text{C} \sim 20^\circ\text{C}$

8.1.2 数学符号和公式

微分符号 d 应使用正体，本模板提供了 \dif 命令。除此之外，模板还提供了一些命令方便使用：

- 圆周率 π : \uppi
- 自然对数的底 e: \upe
- 虚数单位 i, j: \upi \upj

公式应另起一行居中排版。公式后应注明编号，按章顺序编排，编号右端对齐。

$$e^{i\pi} + 1 = 0, \quad (8-1)$$

$$\frac{d^2 u}{dt^2} = \int f(x) dx. \quad (8-2)$$

公式末尾是需要添加标点符号的，至于用逗号还是句号，取决于公式下面一

句是接着公式说的，还是另起一句。

$$\frac{2h}{\pi} \int_0^{\infty} \frac{\sin(\omega\delta)}{\omega} \cos(\omega x) d\omega = \begin{cases} h, & |x| < \delta, \\ \frac{h}{2}, & x = \pm\delta, \\ 0, & |x| > \delta. \end{cases} \quad (8-3)$$

公式较长时最好在等号“=”处转行。

$$\begin{aligned} & I(X_3; X_4) - I(X_3; X_4 | X_1) - I(X_3; X_4 | X_2) \\ &= [I(X_3; X_4) - I(X_3; X_4 | X_1)] - I(X_3; X_4 | \tilde{X}_2) \end{aligned} \quad (8-4)$$

$$= I(X_1; X_3; X_4) - I(X_3; X_4 | \tilde{X}_2). \quad (8-5)$$

如果在等号处转行难以实现，也可在+、-、×、÷运算符号处转行，转行时运算符号仅书写于转行式前，不重复书写。

$$\begin{aligned} \frac{1}{2} \Delta(f_{ij} f^{ij}) = & 2 \left(\sum_{i < j} \chi_{ij} (\sigma_i - \sigma_j)^2 + f^{ij} \nabla_j \nabla_i (\Delta f) \right. \\ & \left. + \nabla_k f_{ij} \nabla^k f^{ij} + f^{ij} f^k [2 \nabla_i R_{jk} - \nabla_k R_{ij}] \right). \end{aligned} \quad (8-6)$$

8.1.3 定理环境

示例文件中使用 ntheorem 宏包配置了定理、引理和证明等环境。用户也可以使用 amsthm 宏包。

这里举一个“定理”和“证明”的例子。

定理 8.1 (留数定理) 假设 U 是复平面上的一个单连通开子集, a_1, \dots, a_n 是复平面上有限个点, f 是定义在 $U \setminus \{a_1, \dots, a_n\}$ 上的全纯函数, 如果 γ 是一条把 a_1, \dots, a_n 包围起来的可求长曲线, 但不经过任何一个 a_k , 并且其起点与终点重合, 那么:

$$\oint_{\gamma} f(z) dz = 2\pi i \sum_{k=1}^n I(\gamma, a_k) \text{Res}(f, a_k). \quad (8-7)$$

如果 γ 是若尔当曲线, 那么 $I(\gamma, a_k) = 1$, 因此:

$$\oint_{\gamma} f(z) dz = 2\pi i \sum_{k=1}^n \text{Res}(f, a_k). \quad (8-8)$$

在这里, $\text{Res}(f, a_k)$ 表示 f 在点 a_k 的留数, $I(\gamma, a_k)$ 表示 γ 关于点 a_k 的卷绕数。卷绕数是一个整数, 它描述了曲线 γ 绕过点 a_k 的次数。如果 γ 依逆时针方向绕着 a_k 移动, 卷绕数就是一个正数, 如果 γ 根本不绕过 a_k , 卷绕数就是零。

定理 8.1 的证明。

证明 首先, 由……

其次, ……

所以……

□

8.2 引用文献的标注

按照教务处的要求, 参考文献外观应符合国标 GB/T 7714 的要求。模版使用 **BIBL^AT_EX** 配合 `biblatex-gb7714-2015` 样式包¹ 控制参考文献的输出样式, 后端采用 `biber` 管理文献。

请注意 `biblatex-gb7714-2015` 宏包 2016 年 9 月才加入 CTAN, 如果你使用的 **T_EX** 系统版本较旧, 可能没有包含 `biblatex-gb7714-2015` 宏包, 需要手动安装。**BIBL^AT_EX** 与 `biblatex-gb7714-2015` 目前在活跃地更新, 为避免一些兼容性问题, 推荐使用较新的版本。

正文中引用参考文献时, 使用 `\cite{key1,key2,key3...}` 可以产生“上标引用的参考文献”, 如 **Meta_CN**, [9, 10]。使用 `\parencite{key1,key2,key3...}` 则可以产生水平引用的参考文献, 例如 [11-13]。请看下面的例子, 将会穿插使用水平的和上标的参考文献: 关于书的 **Meta_CN**, [11, 13], 关于期刊的^[9, 14], 会议论文 [10, 15, 16], 硕士学位论文 [12, 17], 博士学位论文^[18-20], 标准文件 [13], 技术报告^[21], 电子文献 [22, 23], 用户手册 [24]。

当需要将参考文献条目加入到文献表中但又不在正文中引用, 可以使用 `\nocite{key1,key2,key3...}`。使用 `\nocite{*}` 可以将参考文献数据库中的所有条目加入到文献表中。

¹<https://www.ctan.org/pkg/biblatex-gb7714-2015>

全文总结

无人艇是智能化、信息化与传统船舶技术的创新集成。作为能在海上独立执行任务的载体，麻雀虽小、五脏俱全。由于海洋环境的特殊性，无人艇的技术难度不亚于无人机、无人车，这也导致海洋无人装备发展滞后于空天、陆地。

无人船艇自主航行，应具备 4 项基本能力：一是能够通过各类传感器完整地感知水面、水下态势；二是对各类感知目标进行分类，分析判断环境威胁；三是根据任务要求，规划最优路径和航行计划，并具备根据环境变化实时调整计划的能力；四是精准控制航向航迹，自主或遥控完成规定任务。

这些能力的形成，需要依托传统的船舶导航、通信、动力、作战系统，加上智能感知、规划决策、自动控制等方面的最新成果，构成一个完整的从观察到思考，再到执行和反馈的完整链路。

该技术文档是我们团队在无人船研发过程中的技术总结，包括各类传感器、通信、感知、规划、导航、控制、操作界面等系统。

附录 A Maxwell Equations

选择二维情况，有如下的偏振矢量：

$$\mathbf{E} = E_z(r, \theta) \hat{\mathbf{z}}, \quad (\text{A-1a})$$

$$\mathbf{H} = H_r(r, \theta) \hat{\mathbf{r}} + H_\theta(r, \theta) \hat{\boldsymbol{\theta}}. \quad (\text{A-1b})$$

对上式求旋度：

$$\nabla \times \mathbf{E} = \frac{1}{r} \frac{\partial E_z}{\partial \theta} \hat{\mathbf{r}} - \frac{\partial E_z}{\partial r} \hat{\boldsymbol{\theta}}, \quad (\text{A-2a})$$

$$\nabla \times \mathbf{H} = \left[\frac{1}{r} \frac{\partial}{\partial r} (r H_\theta) - \frac{1}{r} \frac{\partial H_r}{\partial \theta} \right] \hat{\mathbf{z}}. \quad (\text{A-2b})$$

因为在柱坐标系下， $\bar{\mu}$ 是对角的，所以 Maxwell 方程组中电场 \mathbf{E} 的旋度：

$$\nabla \times \mathbf{E} = i\omega \mathbf{B}, \quad (\text{A-3a})$$

$$\frac{1}{r} \frac{\partial E_z}{\partial \theta} \hat{\mathbf{r}} - \frac{\partial E_z}{\partial r} \hat{\boldsymbol{\theta}} = i\omega \mu_r H_r \hat{\mathbf{r}} + i\omega \mu_\theta H_\theta \hat{\boldsymbol{\theta}}. \quad (\text{A-3b})$$

所以 \mathbf{H} 的各个分量可以写为：

$$H_r = \frac{1}{i\omega \mu_r} \frac{1}{r} \frac{\partial E_z}{\partial \theta}, \quad (\text{A-4a})$$

$$H_\theta = -\frac{1}{i\omega \mu_\theta} \frac{\partial E_z}{\partial r}. \quad (\text{A-4b})$$

同样地，在柱坐标系下， $\bar{\epsilon}$ 是对角的，所以 Maxwell 方程组中磁场 \mathbf{H} 的旋度：

$$\nabla \times \mathbf{H} = -i\omega \mathbf{D}, \quad (\text{A-5a})$$

$$\left[\frac{1}{r} \frac{\partial}{\partial r} (r H_\theta) - \frac{1}{r} \frac{\partial H_r}{\partial \theta} \right] \hat{\mathbf{z}} = -i\omega \bar{\epsilon} \mathbf{E} = -i\omega \epsilon_z E_z \hat{\mathbf{z}}, \quad (\text{A-5b})$$

$$\frac{1}{r} \frac{\partial}{\partial r} (r H_\theta) - \frac{1}{r} \frac{\partial H_r}{\partial \theta} = -i\omega \epsilon_z E_z. \quad (\text{A-5c})$$

由此我们可以得到关于 E_z 的波函数方程：

$$\frac{1}{\mu_\theta \epsilon_z} \frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial E_z}{\partial r} \right) + \frac{1}{\mu_r \epsilon_z} \frac{1}{r^2} \frac{\partial^2 E_z}{\partial \theta^2} + \omega^2 E_z = 0. \quad (\text{A-6})$$

附录 B 绘制流程图

图 B-1 是一张流程图示意。使用 tikz 环境, 搭配四种预定义节点 (startstop、process、decision 和 io), 可以容易地绘制出流程图。

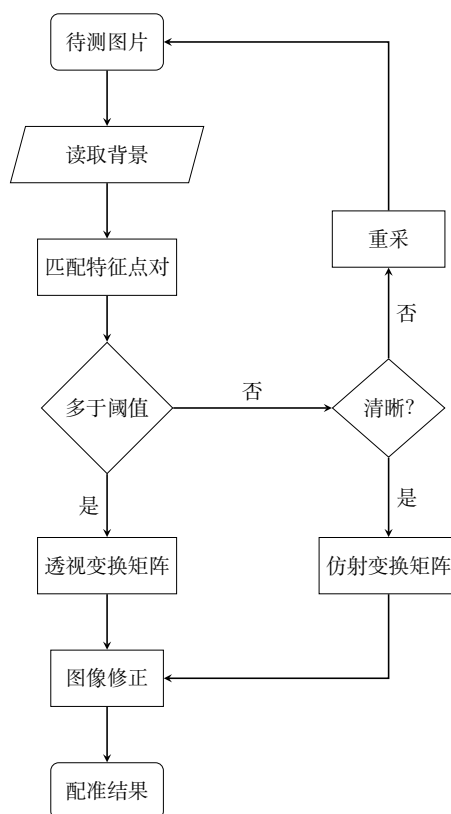


图 B-1 绘制流程图效果

Figure B-1 Flow chart

参考文献

- [1] FOSSEN T I. Handbook of marine craft hydrodynamics and motion control[M]. [S.l.]: John Wiley & Sons, 2011.
- [2] AS D. Assessment of station keeping capability of dynamic positioning vessels[J]., 2016.
- [3] JOHANSEN T A, FOSSEN T I, BERGE S P. Constrained nonlinear control allocation with singularity avoidance using sequential quadratic programming[J]. IEEE Transactions on Control Systems Technology, 2004, 12(1): 211-216.
- [4] FOSSEN T I, BREIVIK M, SKJETNE R. Line-of-sight path following of under-actuated marine craft[J]. IFAC proceedings volumes, 2003, 36(21): 211-216.
- [5] WERLING M, ZIEGLER J, KAMMEL S, et al. Optimal trajectory generation for dynamic street scenarios in a frenet frame[C]//2010 IEEE International Conference on Robotics and Automation. [S.l. : s.n.], 2010: 987-993.
- [6] TAKAHASHI A, HONGO T, NINOMIYA Y, et al. Local Path Planning And Motion Control For Agv In Positioning[C]//Proceedings. IEEE/RSJ International Workshop on Intelligent Robots and Systems '. (IROS '89) 'The Autonomous Mobile Robots and Its Applications. [S.l. : s.n.], 1989: 392-397. DOI: 10.1109/IROS.1989.637936.
- [7] REEDS J A, SHEPP L A. Optimal paths for a car that goes both forwards and backwards.[J/OL]. Pacific J. Math., 1990, 145(2): 367-393. <https://projecteuclid.org:443/euclid.pjm/1102645450>.
- [8] HAGER J W, BEHENSKY J F, DREW B W. The Universal Grids: Universal Transverse Mercator (UTM) and Universal Polar Stereographic (UPS). Edition 1[R]. [S.l.]: DEFENSE MAPPING AGENCY HYDROGRAPHIC/TOPOGRAPHIC CENTER WASHINGTON DC, 1989.
- [9] CHEN H, CHAN C T. Acoustic cloaking in three dimensions using acoustic metamaterials[J]. Applied Physics Letters, 2007, 91: 183518.

- [10] KIM S, WOO N, YEOM H Y, et al. Design and Implementation of Dynamic Process Management for Grid-enabled MPICH[C]//The 10th European PVM/MPI Users' Group Conference. Venice, Italy: [s.n.], 2003.
- [11] JOANNOPOULOS J D, JOHNSON S G, WINN J N. Photonic Crystals: Molding the Flow of Light[M]. [S.l.]: Princeton University Press, 2008.
- [12] 猪八戒. 论流体食物的持久保存[D]. 北京: 广寒宫大学, 2005.
- [13] IEEE Std 1363-2000. IEEE Standard Specifications for Public-Key Cryptography[M]. New York: IEEE, 2000.
- [14] CHEN H, WU B I, ZHANG B, et al. Electromagnetic Wave Interactions with a Metamaterial Cloak[J]. Physical Review Letters, 2007, 99(6): 63903.
- [15] KOCHER C, JAFFE J, JUN B. Differential Power Analysis[C]//WIENER M. Lecture Notes in Computer Science: Advances in Cryptology (CRYPTO '99): vol. 1666. [S.l.]: Springer-Verlag, 1999: 388-397.
- [16] 王重阳, 黄药师, 欧阳峰, 等. 武林高手论文集[C]//第 N 次华山论剑. 西安, 中国: 中国古籍出版社, 2006.
- [17] JEYAKUMAR A R. Metamori: A library for Incremental File Checkpointing[D]. Blacksburg: Virginia Tech, 2004.
- [18] 沙和尚. 论流沙河的综合治理[D]. 北京: 清华大学, 2005.
- [19] ZADOK E. FiST: A System for Stackable File System Code Generation[D]. USA: Computer Science Department, Columbia University, 2001.
- [20] 白云芬. 信用风险传染模型和信用衍生品的定价[D]. 上海: 上海交通大学, 2008.
- [21] WOO A, BAILEY D, YARROW M, et al. The NAS Parallel Benchmarks 2.0[R/OL]. The Pennsylvania State University CiteSeer Archives. 1995. <http://www.nasa.org/>.
- [22] 萧钰. 出版业信息化迈入快车道[EB/OL]. (2001-12-19) [2002-04-15]. <http://www.creader.com/news/20011219/200112190019.html>.
- [23] CHRISTINE M. Plant physiology: plant biology in the Genome Era[J/OL]. Science, 1998, 281: 331-332 [1998-09-23]. <http://www.sciencemag.org/cgi/collection/anatmorp>.

- [24] R Core Team. R: A Language and Environment for Statistical Computing[M/OL]. Vienna, Austria: [s.n.], 2012. <http://www.R-project.org/>.

致 谢

感谢那位最先制作出博士学位论文 L^AT_EX 模板的交大物理系同学!

感谢 William Wang 同学对模板移植做出的巨大贡献!

感谢 @weijianwen 学长一直以来的开发和维护工作!

感谢 @sjtug 以及 @dyweb 对 0.9.5 之后版本的开发和维护工作!

感谢所有为模板贡献过代码的同学们, 以及所有测试和使用模板的各位同学!

感谢 L^AT_EX 和 SJTUT_{HESIS}, 帮我节省了不少时间。