智能工程

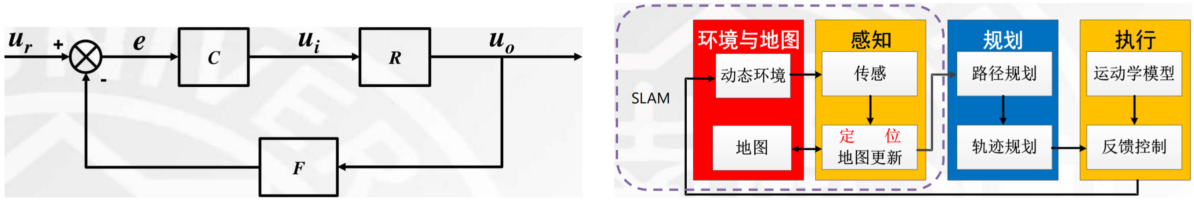
目 录

图 片

表 格

要 点

1 基础知识



(a) 课程内容 (b) 工作流程

图 1: 课程内容

	u_i	u_o	R	F	u_r	e	C
概念	系统输入	系统输出	系统模型	反馈单元	系统给定	系统误差	控制器
含义	对被控对象施加作用的手段	作业目标的可测系统状态	系统输入输出映射	系统输出映射变换	系统作业目标	作业目标与系统当前测量状态差值	系统误差与输入映射
内容	机器人运动学			机器人控制		机器人感知	机器人运动规划

表 1: 课程内容

课程内容

课程案例 移动机器人->轮式机器人->两轮差速机器人。

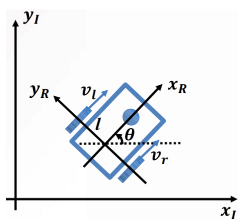


图 2: 两轮差速机器人模型

- 车轮半径r。
- 两轮转速φ_l, φ_r：$v_i = r\varphi_i$。
- 车轮到两轮中点距离l。

1. 正运动学模型2.3.4。
2. 运动控制器3.1。
3. 里程计模型4.3.1。

返回目录

2 机器人运动学

2.1 运动学模型

表征机器人驱动（输入）和空间位姿（输出）的关系。

机械臂与移动机器人的区别

- 机械臂本体坐标系固定，精度高；移动机器人本体坐标系随动，精度低。
- 非完整约束 [1] （不可积的微分约束）：移动机器人只知道码盘变化量无法获取位姿，状态取决于路径。
- 微分运动学（Differential Kinematics）：速度空间替代位置空间。

2.2 车轮

类型 [2]

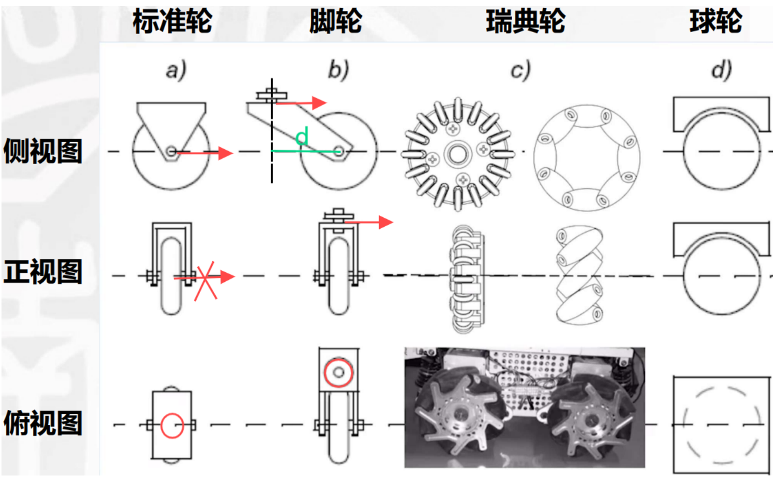


图 3: 车轮类型

返回目录

类型	自由度	约束	分类/特点
标准轮（Standard wheel）	2 沿轮平面滚动 沿垂直轴转动	1 沿轮轴滑动	标准固定轮：只能滚动 标准转向轮（舵轮）
脚轮（Castor wheel）	3 沿轮平面滚动 沿垂直轴转动 沿路轴运动	0	偏心距d：触地点到垂直旋转轴距离。 受扭矩压力，易损坏。
瑞典轮（Swedish wheel）	3 沿轮平面滚动（被动） 沿轮轴转动（主动） 沿垂直轴转动（被动）	0	麦克纳姆轮（Macanum wheel）：45°，至少4个共同使用。 连续切换轮：90°，至少3个共同使用。 对地面冲击大，噪音大，易损坏，成本高。
球轮（Spherical wheel）	3（全主动） 沿两个正交轮轴转动 沿垂直轴转动	0	成本高，可靠性差。

表 2: 车轮类型对比

选取

- 数量：至少三轮同时着地才能保证静态稳定性。四轮可以提升稳定性，但需要悬架。
- 大小：车轮越大通过性越好，但需要更大的扭矩。
- 多数形态都有非完整约束。

2.3 运动学建模

2.3.1 空间描述与状态表达

坐标系

- 惯性系I：作业目标、控制指令、传感器感知测量信息。
- 机器人系R：控制器误差输入、控制器控制指令。

- 笛卡尔系：右手法则。

位姿（POSE）

位置空间求导得到速度空间：

$$\xi_I = \begin{bmatrix} x_I \\ y_I \\ \theta_I \end{bmatrix}, \xi_R = \begin{bmatrix} x_R \\ y_R \\ \theta_R \end{bmatrix} \xRightarrow{\text{求导}} \dot{\xi}_I = \begin{bmatrix} \dot{x}_I \\ \dot{y}_I \\ \dot{\theta}_I \end{bmatrix}, \dot{\xi}_R = \begin{bmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\theta}_R \end{bmatrix}$$

惯性系旋转得到机器人系：

$$\dot{\xi}_R = R(\theta)\dot{\xi}_I$$

旋转阵$R(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$为单位正交阵，$R^T = R^{-1}$。

2.3.2 瞬心（*ICR*）法 [3]

瞬时旋转/曲率中心（ICR）
刚体上各点角速度相同。

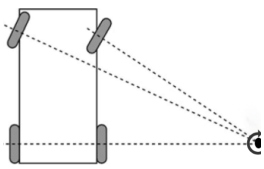


图 4: 瞬心

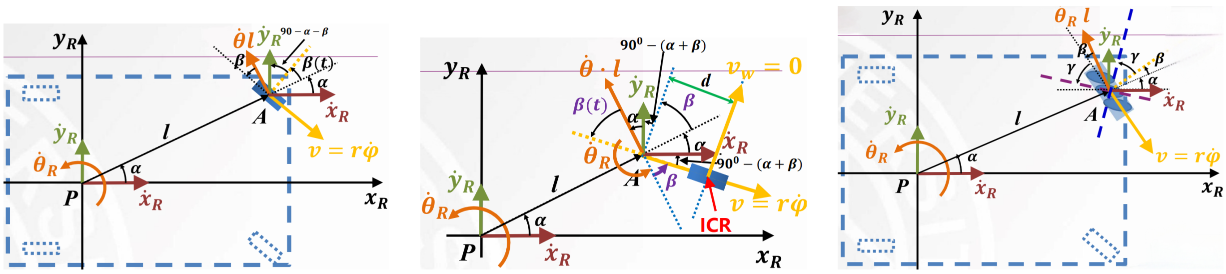
步骤

1. 坐标系变换。
2. 确定约束。
3. 确定瞬心：各轮轮轴到该点距离与速度成正比。
4. 求解$\dot{\xi}_R = \begin{bmatrix} \dot{x}_R & \dot{y}_R & \dot{\theta}_R \end{bmatrix}^T$。

2.3.3 约束方程法 [4]

要求　在水平面上运动，与地面点接触，不变形，安装在刚体表面，舵机转轴与地面垂直。

返回目录



(a) 标准轮　(b) 脚轮　(c) 瑞典轮

图 5: 车轮约束示意图

类型	约束	约束方程	主动轮	随动轮
标准轮	纯滚动	$\begin{bmatrix}\sin(\alpha+\beta(t)) & -\cos(\alpha+\beta(t)) & -l\cos\beta(t)\end{bmatrix} R\theta\dot{\xi}_I = r\dot{\varphi}$	√	x
	无滑动	$\begin{bmatrix}\cos(\alpha+\beta(t)) & \sin(\alpha+\beta(t)) & l\sin\beta(t)\end{bmatrix} R\theta\dot{\xi}_I = 0$	√	√
脚轮	纯滚动	$\begin{bmatrix}\sin(\alpha+\beta) & -\cos(\alpha+\beta) & -l\cos\beta\end{bmatrix} R\theta\dot{\xi}_I = r\dot{\varphi}$	√	x
	无滑动	$\begin{bmatrix}\cos(\alpha+\beta) & \sin(\alpha+\beta) & d+l\sin\beta\end{bmatrix} R\theta\dot{\xi}_I = -d\dot{\beta}$	√	x
瑞典轮	纯滚动	$\begin{bmatrix}\cos(\alpha+\beta+\gamma) & \sin(\alpha+\beta+\gamma) & l\sin(\beta+\gamma)\end{bmatrix} R\theta\dot{\xi}_I = r\dot{\varphi}\sin\gamma + r_{sw}\dot{\varphi}_{sw}$	√	x
	无滑动	$\begin{bmatrix}\sin(\alpha+\beta+\gamma) & -\cos(\alpha+\beta+\gamma) & -l\cos(\beta+\gamma)\end{bmatrix} R\theta\dot{\xi}_I = r\dot{\varphi}\cos\gamma$	√ 小轮	x

表 3: 车轮约束方程

约束方程

使用　根据各轮主/随动状态列运动约束方程，得到最多三个独立方程（对应$[x, y, \theta]$）。以下以N个标准轮（N_f个固定，N_s个转向）机器人为例：

- 滚动约束

$$J_1(\beta_s)R(\theta)\dot{\xi}_I - J_2\dot{\varphi} = 0$$

其中$J_1(\beta_s) = \begin{bmatrix} J_{1f(N_f\times 3)} \\ J_{1s}(\beta_s)_{(N_s\times 3)} \end{bmatrix}$, $\varphi(t) = \begin{bmatrix}\varphi_f(t) \\ \varphi_s(t)\end{bmatrix}$, $J_2 = \mathrm{diag}(r_1, \cdots, r_N)$为轮径对角阵。

- 滑动约束

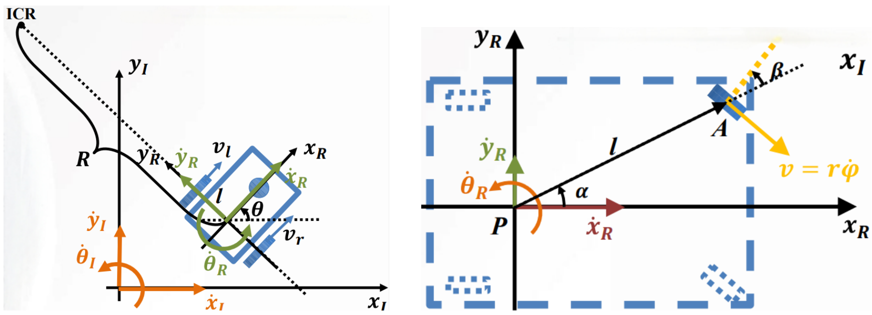
$$C_1(\beta_s)R(\theta)\dot{\xi}_I = 0$$

返回目录

其中$C_1(\beta_s)=\begin{bmatrix}C_{1f(N_f\times 3)}\\ C_{1s}(\beta_s)_{(N_s\times 3)}\end{bmatrix}$。

2.3.4 例子

以下以两轮差速机器人（见1）为例，$\alpha=\frac{\pi}{2},\beta=0$：



(a) 瞬心法　　(b) 约束方程法

图 6: 两轮差速机器人正运动学建模

瞬心法

两轮差速机器人的瞬心在两轮轮轴上，设其到机器人两轮中间的距离为R，有：

$$\dot{\theta}=\frac{\dot{x}_R}{R}=\frac{\dot{\varphi}_l r}{R-l}=\frac{\dot{\varphi}_r r}{R+l}$$

解得$R=\frac{\dot{\varphi}_r+\dot{\varphi}_l}{\dot{\varphi}_r-\dot{\varphi}_l}$，代回即可。

约束方程法

- 纯滚动：$\begin{bmatrix}\sin(\alpha+\beta) & -\cos(\alpha+\beta) & -l\cos\beta\end{bmatrix}\dot{\xi}_R=r\dot{\varphi}$。
- 无滑动：$\begin{bmatrix}\cos(\alpha+\beta) & \sin(\alpha+\beta) & l\sin\beta\end{bmatrix}\dot{\xi}_R=0$。

正运动学模型

$$\dot{\xi}_R=\begin{bmatrix}\dot{x}_R\\ \dot{y}_R\\ \dot{\theta}_R\end{bmatrix}=\frac{r}{2}\begin{bmatrix}\dot{\varphi}_l+\dot{\varphi}_r\\ 0\\ \frac{\dot{\varphi}_r-\dot{\varphi}_l}{l}\end{bmatrix}$$

返回目录

2.4 自由度

- 衡量机器人改变运行状态的能力。
- 需满足实际作业需求，考虑实现成本。
- 机器人设计基础、算法依据（一般同自由度机器人可采用相同控制规划算法）。
- 平面运动机器人自由度最大不能超过3。

分类 5

- 移动度（Degree of Mobility）δ_m：瞬时改变机器人运动状态的能力。

$$\delta_m = \dim[C_1(\beta_s)] = 3 - \operatorname{rank}[C_1(\beta_s)] \in [0, 3]$$

- 转向度（Degree of Steerability）δ_s：间接改变机器人运动状态的能力。

$$\delta_s = \operatorname{rank}[C_{1s}(\beta_s)] \in [0, 2]$$

- 机动度（Degree of Maneuverability）δ_M：改变机器人运动状态的能力。

$$\delta_M = \delta_m + \delta_s$$

  - 机动度相同，结构不一定相同。
  - $\delta_M = 2$，瞬心位于一条直线上；$\delta_M = 3$，瞬心可分布于空间任何一点。

例题 示意图中，圆表示全向轮，椭圆表示固定轮，可旋转椭圆表示舵轮。

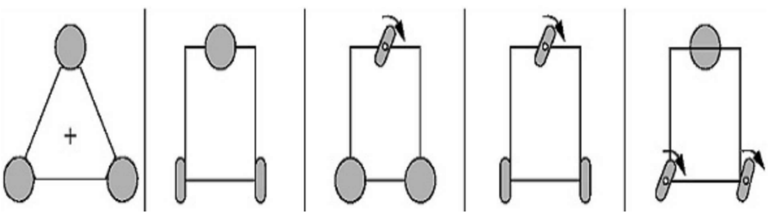


图 7: 自由度例题

	x	y	θ	δ_m	δ_s	δ_M
1	√	√	√	3	0	3
2	√	x	√	2	0	2
3	√	–	√	2	1	3
4	√	x	–	1	1	2
5	√	–	–	1	2	3

表 4: 自由度例题

√表示可直接控制，–表示可间接控制（不同形态下直接控制），x表示不可控制。

返回目录

实例（TYPE(移动度,转向度)）

- 全向机器人：
  - Type(3,0)：完整约束全方位移动机器人。
  - Type(2,1)：一个同心轮+两个瑞典轮。
  - Type(1,2)：多舵机全方位移动机器人。
- 非全向机器人：
  - Type(2,0)：差分移动机器人。
  - Type(1,1)：自动驾驶汽车（阿克曼转向）、自行车、叉车。

3 机器人运动控制

3.1 运动控制

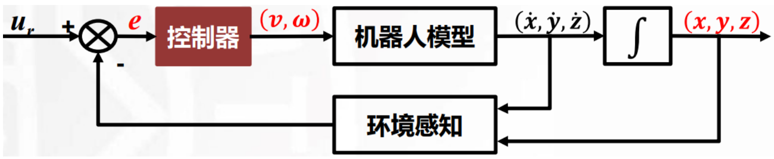


图 8: 运动控制器

误差（惯性系下给定与反馈）$\xRightarrow{\text{变换}}$输入（机器人系下控制输入）。

特点

- 大多存在滑动约束，是非完整系统，有侧向和姿态偏差。
- 非线性，控制器复杂，需根据可获得的反馈信号选取，且不能同时实现定点控制和跟踪控制。
- 不存在能完成控制目标的连续时不变（静态）反馈控制率。
- 受标定精度影响大，且由于执行器约束，控制输入要合理限幅。

分类

- 定点（镇定）控制（Regulation Control）：以指定姿态到达指定位置。
- 跟踪控制：

返回目录

- 轨迹跟踪控制（Trajectory Tracking Control）：跟随给定轨迹（速度+姿态）。
- 路径跟踪控制（Path Tracking Control）：跟随给定路径。

开环控制　将运动轨迹分割成直线和圆弧，存在以下问题：

- 直线和圆弧的曲率不一致，不连续。
- 速度、加速度约束。
- 难多解，不光滑，且无法自适应调整来面对环境变化。

控制器性能评价　取正定李雅普诺夫函数，其导数负定则系统渐进稳定。

两轮差速机器人运动控制　以下以两轮差速机器人（见1）为例。

$$\dot{\xi}_R = \begin{bmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\theta}_R \end{bmatrix} = \frac{r}{2}\begin{bmatrix} \dot{\varphi}_l + \dot{\varphi}_r \\ 0 \\ \frac{\dot{\varphi}_r - \dot{\varphi}_l}{l} \end{bmatrix} = \begin{bmatrix} v_1 \\ 0 \\ v_2 \end{bmatrix} \xRightarrow{\text{变换}} \dot{\xi}_I = \begin{bmatrix} \dot{x}_I \\ \dot{y}_I \\ \dot{\theta}_I \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

3.2 定点控制器 [6]

控制目标　机器人参考坐标系下误差$e = \begin{bmatrix} x & y & \theta \end{bmatrix}^T$，设计控制阵$K = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \end{bmatrix}$，其中$k_{ij} = k(t, e)$，得到控制输入$\begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} = Ke$，使$\lim_{t\to\infty} e(t) = 0$。

误差信号转换

惯性系下，实际状态$q = \begin{bmatrix} x & y & \theta \end{bmatrix}^T$与参考状态$q_r \begin{bmatrix} x_r & y_r & \theta_r \end{bmatrix}^T$之差为开环误差$\tilde{q} = \begin{bmatrix} \tilde{x} & \tilde{y} & \tilde{\theta} \end{bmatrix}^T = \begin{bmatrix} x - x_r & y - y_r & \theta - \theta_r \end{bmatrix}^T$。

1. 转换到机器人系

$$e = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = R(\theta)^T \tilde{q} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{\theta} \end{bmatrix} \xRightarrow{\text{闭环}} \dot{e} = \begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = R(\theta)\dot{\tilde{q}} + R(\dot{\theta})\tilde{q} = \begin{bmatrix} v_1 + v_2 e_2 \\ -v_2 e_1 \\ v_2 \end{bmatrix}$$

2. 转换到极坐标系

$$\begin{cases}\rho = \sqrt{\tilde{x}^2+\tilde{y}^2}\\ \beta = -\mathrm{atan2}(-\tilde{y},-\tilde{x})\\ \alpha = -\beta-\tilde{\theta}\end{cases} \xRightarrow{\text{闭环}} \begin{bmatrix}\dot{\rho}\\ \dot{\beta}\\ \dot{\alpha}\end{bmatrix} = \begin{bmatrix}-\cos\alpha & 0\\ \frac{\sin\alpha}{\rho} & 0\\ \frac{\sin\alpha}{\rho} & -1\end{bmatrix}\begin{bmatrix}v_1\\ v_2\end{bmatrix}$$

机器人系非线性控制器

设计控制器 $\begin{bmatrix}v_1\\ v_2\end{bmatrix} = \begin{bmatrix}-k_1e_1\\ -k_2e_2+e_2^2\sin(t)\end{bmatrix}$，代入得 $\begin{bmatrix}\dot{e}_1\\ \dot{e}_2\\ \dot{e}_3\end{bmatrix} = \begin{bmatrix}-k_1e_1+v_2e_2\\ -v_2e_1\\ -k_2e_2+e_2^2\sin(t)\end{bmatrix}$，其有误差时扰动，效果不佳。

极坐标系线性控制器

设计控制器 $\begin{cases}v_1 = k_\rho\rho\\ v_2 = k_\alpha\alpha+k_\beta\beta\end{cases}$，代入得

$$\begin{bmatrix}\dot{\rho}\\ \dot{\beta}\\ \dot{\alpha}\end{bmatrix} = \begin{bmatrix}-k_\rho\rho\cos\alpha\\ -k_\rho\sin\alpha\\ k_\rho\sin\alpha-k_\alpha\alpha-k_\beta\beta\end{bmatrix} \xRightarrow{\alpha\to 0} \begin{bmatrix}-k_\rho\rho\\ -k_\rho\alpha\\ k_\rho\alpha-k_\alpha\alpha-k_\beta\beta\end{bmatrix}$$

其中前两行非线性耦合，在$\alpha \to 0$时指数性稳定，非全局稳定。

极坐标系线性控制器

设计控制器 $\begin{cases}v_1 = k_\rho\rho\cos\alpha\\ v_2 = k_\alpha\alpha+\frac{k_\rho\sin\alpha\cos\alpha}{\alpha}(\alpha-k_\beta\beta)\end{cases}$，代入得

$$\begin{bmatrix}\dot{\rho}\\ \dot{\beta}\\ \dot{\alpha}\end{bmatrix} = \begin{bmatrix}-k_\rho\rho\cos^2\alpha\\ -k_\rho\cos\alpha\sin\alpha\\ k_\rho\cos\alpha\sin\alpha-k_\alpha\alpha-\frac{k_\rho\sin\alpha\cos\alpha}{\alpha}(\alpha-k_\beta\beta)\end{bmatrix} \xRightarrow{\alpha\to 0} \begin{bmatrix}-k_\rho\rho\\ -k_\rho\alpha\\ -k_\alpha\alpha+k_\rho k_\beta\beta\end{bmatrix}$$

其全局渐近稳定。

3.3 轨迹跟踪控制器 [7]

控制目标与误差变换

惯性系下，实际轨迹$q(t)=\begin{bmatrix}x(t) & y(t) & \theta(t)\end{bmatrix}^T$与参考轨迹$q_r\begin{bmatrix}x_r(t) & y_r(t) & \theta_r(t)\end{bmatrix}^T$之差为开环误差$\tilde{q}(t)=\begin{bmatrix}\tilde{x}(t) & \tilde{y}(t) & \tilde{\theta}(t)\end{bmatrix}^T=\begin{bmatrix}x(t)-x_r(t) & y(t)-y_r(t) & \theta(t)-\theta_r(t)\end{bmatrix}^T$，控制目标为$\lim_{n\to\infty}\tilde{q}(t)=0$。

辅助误差信号为：

$$e=\begin{bmatrix}\cos\theta & \sin\theta & 0\\ -\sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\tilde{x}\\ \tilde{y}\\ \tilde{\theta}\end{bmatrix}\xRightarrow{\text{闭环}}\dot{e}=\begin{bmatrix}v_1+v_2e_2-v_{1r}\cos e_3\\ -v_2e_1+v_{1r}\sin e_3\\ v_2-v_{2r}\end{bmatrix}$$

控制器

设计控制器$\begin{bmatrix}v_1\\ v_2\end{bmatrix}=\begin{bmatrix}-k_1e_1+v_{1r}\cos e_3\\ -v_{1r}\frac{\sin e_3}{e_3}e_2-k_2e_3+v_{2r}\end{bmatrix}$，代入得$\begin{bmatrix}\dot{e}_1\\ \dot{e}_2\\ \dot{e}_3\end{bmatrix}=\begin{bmatrix}-k_1e_1+v_2e_2\\ -v_2e_1+v_{1r}\sin e_3\\ -k_2e_3-v_{1r}\frac{\sin e_3}{e_3}e_2\end{bmatrix}$。

$e_3\to 0$时，控制器简化为$\begin{bmatrix}v_1\\ v_2\end{bmatrix}=\begin{bmatrix}-k_1e_1+v_{r1}\\ -v_{r1}e_2+v_{r2}\end{bmatrix}$，代入得$\begin{bmatrix}\dot{e}_1\\ \dot{e}_2\\ \dot{e}_3\end{bmatrix}=\begin{bmatrix}-k_1e_1+v_2e_2\\ -v_2e_1\\ -k_2e_3-v_{r1}e_2\end{bmatrix}$。

3.4 路径跟踪控制器 [8]

控制目标与误差变换

惯性系下，实际路径$q(s)=\begin{bmatrix}x(s) & y(s) & \theta(s)\end{bmatrix}^T$与参考路径$q_r\begin{bmatrix}x_r(s) & y_r(s) & \theta_r(s)\end{bmatrix}^T$之差为开环误差$\tilde{q}(s)=\begin{bmatrix}\tilde{x}(s) & \tilde{y}(s) & \tilde{\theta}(s)\end{bmatrix}^T=\begin{bmatrix}x(s)-x_r(s) & y(s)-y_r(s) & \theta(s)-\theta_r(s)\end{bmatrix}^T$，其中$s\in[0,1]$为路径参考变量，控制目标为$\lim_{n\to\infty}\tilde{q}(s)=0$。

作变换$\begin{cases}y_1 & =x+b\cos\theta\\ y_2 & =y+b\sin\theta\end{cases}$，进而得到闭环误差$\begin{bmatrix}\dot{y}_1\\ \dot{y}_2\end{bmatrix}=\begin{bmatrix}\cos\theta & -b\sin\theta\\ \sin\theta & b\cos\theta\end{bmatrix}\begin{bmatrix}v\\ \omega\end{bmatrix}=T(\theta)\begin{bmatrix}v\\ \omega\end{bmatrix}$。

返回目录

逆运算得到 $\begin{bmatrix} v \\ \omega \end{bmatrix} = T^{-1}(\theta)\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\frac{\sin\theta}{b} & \frac{\cos\theta}{b} \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$，故 $\begin{cases} \dot{y}_1 & = u_1 \\ \dot{y}_2 & = u_2 \\ \dot{\theta} & = \frac{u_2\cos\theta - u_1\sin\theta}{b} \end{cases}$。

控制器

设计控制器 $\begin{cases} u_1 = \dot{y}_{1d} + k_1(y_{1d} - y_1) \\ u_2 = \dot{y}_{2d} + k_2(y_{2d} - y_2) \end{cases}$，有 $\begin{cases} \dot{\tilde{y}}_1 = -k_1\tilde{y}_1 \\ \dot{\tilde{y}}_2 = -k_2\tilde{y}_2 \end{cases}$，系统指数性收敛。

4 机器人感知

4.1 传感器 [9]

分类	传感器	感受	源	分类	传感器	感受	源
触觉	接触开关，碰撞器	EC	P	测距	反射率传感器	EC	A
	光学屏障	EC	A		超声波传感器	EC	A
	非接触式接近传感器	EC	A		激光测距仪	EC	A
轮/电机	电刷编码器	PC	P		光学三角测量	EC	A
	电位计	PC	P		结构光	EC	A
	同步器，旋转变压器	PC	P	运动	多普勒雷达（Rader）	EC	A
	光电编码器	PC	P		多普勒声波	EC	A
	磁编码器	PC	P		激光雷达（Laser）	EC	A
	电感编码器	PC	P		里程计（Odometer）	PC	P
	电容编码器	PC	P		惯导系统（IMU）	PC	P
方向	罗盘（Compass）	EC	P		加速度传感器	PC	P
	陀螺仪（Gyroscope）	PC	P	信标	GPS	EC	A
	倾角仪	EC	A/P		有源光学或射频信标	EC	A
视觉	相机（Camera）	EC	P		有源超声波信标	EC	A
	视觉测距套件	EC	P		有源光学或射频信标	EC	A
	目标跟踪套件	EC	P		反射信标	EC	A

表 5: 传感器分类

返回目录

分类

- PC（Proprioceptive，本体感受）/EC（Exteroceptive，外感受）。
- A（Active，有源）/P（Passive，无源）。

特性

- 测量范围：测量上下界之差。
- 动态范围：测量范围上下界比率，常用对数表示（单位dB）。
- 分辨率：最小可测量变化量，一般为为动态范围下界。
- 线性度：输入输出信号的映射关系。

4.2 光电传感器

4.2.1 概述

把被测量变化转换成光信号变化，再转换成电信号。由辐射源、光学通路、光电器件组成。

特性

- 不受电磁干扰影响。
- 非接触测量。
- 频谱宽，高精度，高分辨率，高可靠性，发应快。

4.2.2 编码器 [10]

测量系统相对运动角度，具有高精度、高分辨率和高可靠性。按结构可分为接触式、光电式和电磁式，后两种为非接触式编码。

增量式旋转编码器

- 不能直接输出数字编码，需要数字电路。

- 原理：遮光周期性变化，莫尔条纹明暗交替，电压$U_0 = U_m \cos(\frac{2\pi}{W}x)$周期性变化，形成脉冲，根据脉冲数量可推算旋转角度。位置数据是相对的，掉电后需复位。
- 辨向：为判断光栅移动方向，使用D触发器整合两个光栅的信息。
  - D触发器：时钟信号有效时，$Q = D$。
  - 边缘D触发器：时钟信号处于有效边沿时，$Q = D$。

绝对式光电编码器

- 能直接输出某种码制的数码，掉电后无需复位。
- 格雷码（余3循环码）：任意相邻数只有一位二进制数不同，可以由二进制码按位异或（第一位保留）获得，属于可靠性编码，求反方便。

4.3 里程计

4.3.1 里程计模型

例子 以下以两轮差速机器人（见1）为例。

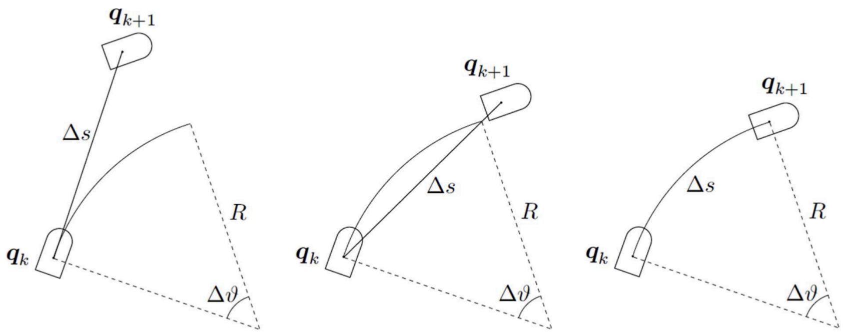
$$\dot{\xi}_R = \begin{bmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\theta}_R \end{bmatrix} = \frac{r}{2}\begin{bmatrix} \dot{\varphi}_l + \dot{\varphi}_r \\ 0 \\ \frac{\dot{\varphi}_r - \dot{\varphi}_l}{l} \end{bmatrix} = \begin{bmatrix} v \\ 0 \\ \omega \end{bmatrix} \xrightarrow{\text{转换到惯性系}} \begin{cases} \dot{x} &= v\cos\theta \\ \dot{y} &= v\sin\theta \\ \dot{\theta} &= \omega \end{cases}$$

码盘读数为：

$$\begin{cases} \Delta s = \frac{r}{2}(\Delta\phi_R + \Delta\phi_L) \\ \Delta\theta = \frac{r}{2d}(\Delta\phi_R - \Delta\phi_L) \end{cases} \xrightarrow{\text{小时间变化}} \begin{cases} \Delta s &= v_k T_s \\ \Delta\theta &= \omega_k T_s \end{cases}$$

建模方法 [11]

返回目录



(a) 欧拉法 (b) RK2法 (c) 解析积分法

图 9: 里程计建模方法

- 欧拉法

$$\begin{cases} x_{k+1} = x_k + v_k T_s \cos\theta_k \\ y_{k+1} = y_k + v_k T_s \sin\theta_k \\ \theta_{k+1} = \theta_k + \omega_k T_s \end{cases}$$

- RK2（二阶Runge-Kutta）法

$$\begin{cases} x_{k+1} = x_k + v_k T_s \cos(\theta_k + \frac{\omega_k T_s}{2}) \\ y_{k+1} = y_k + v_k T_s \sin(\theta_k + \frac{\omega_k T_s}{2}) \\ \theta_{k+1} = \theta_k + \omega_k T_s \end{cases}$$

- 解析积分法

$$\begin{cases} x_{k+1} = x_k + \frac{v_k}{\omega_k}(\sin\theta_{k+1} - \sin\theta_k) \\ y_{k+1} = y_k - \frac{v_k}{\omega_k}(\cos\theta_{k+1} - \cos\theta_k) \\ \theta_{k+1} = \theta_k + \omega_k T_s \end{cases} \xRightarrow{\omega_k=0} \begin{cases} x_{k+1} = x_k + v_k T_s \\ y_{k+1} = y_k \end{cases}$$

4.3.2 里程计误差

误差来源

- 数值积分误差。
- 运动学参数误差：速度不恒定，半径误差。
- 打滑。

误差传播（以RK2法为例）[12]

位姿更新为：

$$p' = f(x, y, \theta, \Delta\phi_R, \Delta\phi_L) = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \frac{r(\Delta\phi_R+\Delta\phi_L)}{2}\cos(\theta_k + \frac{r[\Delta\phi_R-\Delta\phi_L]}{4d}) \\ \frac{r(\Delta\phi_R+\Delta\phi_L)}{2}\sin(\theta_k + \frac{r[\Delta\phi_R-\Delta\phi_L]}{4d}) \\ \frac{r(\Delta\phi_R-\Delta\phi_L)}{2d} \end{bmatrix}$$

返回目录

其中$\Delta\phi_R, \Delta\phi_L$是控制输入量，有误差协方差矩阵迭代公式：

$$\sum_{p'} = \underbrace{\nabla_p f \cdot \sum_p \cdot \nabla_p f^T}_{\text{位姿}} + \underbrace{\nabla_{r|l} f \cdot \sum_\Delta \cdot \nabla_{r|l} f^T}_{\text{控制输入量}}$$

初始化姿态协方差矩阵$\sum_{p'}$（可零初始化），其更新量为：

$$\nabla_p f = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} & \frac{\partial f}{\partial \theta} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\Delta s \sin(\theta_k + \frac{\Delta\theta}{2}) \\ 0 & 1 & \Delta s \cos(\theta_k + \frac{\Delta\theta}{2}) \\ 0 & 0 & 1 \end{bmatrix}$$

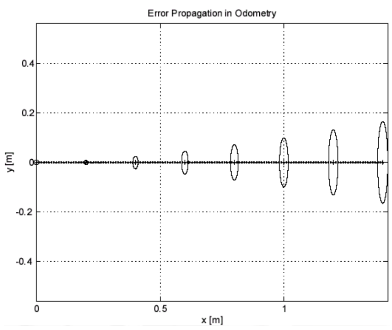
假设$\Delta\phi_R, \Delta\phi_L$的误差相互独立，有控制输入量协方差矩阵：

$$\sum_\Delta = covar(\Delta\phi_R, \Delta\phi_L) = \begin{bmatrix} k_r \|\Delta\phi_R\| & 0 \\ 0 & k_l \|\Delta\phi_L\| \end{bmatrix}$$

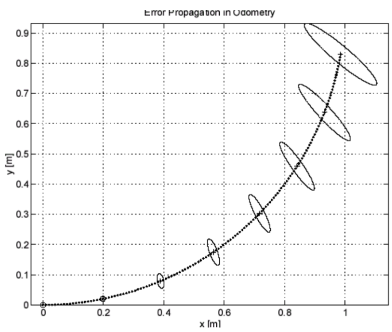
其更新量为：

$$\nabla_{rl} f = \begin{bmatrix} \frac{\partial f}{\partial \Delta\phi_R} & \frac{\partial f}{\partial \Delta\phi_L} \end{bmatrix} = \begin{bmatrix} \frac{r}{2}\cos(\theta_k + \frac{\Delta\theta}{2}) - \frac{r}{4d}\Delta s \sin(\theta_k + \frac{\Delta\theta}{2}) & \frac{r}{2}\cos(\theta_k + \frac{\Delta\theta}{2}) + \frac{r}{4d}\Delta s \sin(\theta_k + \frac{\Delta\theta}{2}) \\ \frac{r}{2}\sin(\theta_k + \frac{\Delta\theta}{2}) + \frac{r}{4d}\Delta s \cos(\theta_k + \frac{\Delta\theta}{2}) & \frac{r}{2}\sin(\theta_k + \frac{\Delta\theta}{2}) - \frac{r}{4d}\Delta s \cos(\theta_k + \frac{\Delta\theta}{2}) \\ \frac{r}{2d} & -\frac{r}{2d} \end{bmatrix}$$

误差转化展示 原理见9.1。



(a) 直线运动



(b) 曲线运动

图 10: 里程计误差转化展示

直线运动时误差方向与运动方向垂直，曲线运动时则不垂直。

返回目录

4.4 激光传感器

- 组成：激光器，激光检测器，测量电路。
- 特点：无接触远距离测量，速度快，精度高，量程大，抗干扰能力强。
- 激光测距：到达时间法（Time of Flight，TOF）：时间精度 $=\frac{\text{测量精度}}{c(3\times10^8)}$。
- 位移测量：对参考信号和测量信号进行相位测量。

5 机器人点云处理

5.1 直线提取

5.1.1 最小二乘法（*Least Squares Method*）

在求解拟合直线时，最小化拟合误差平方和，目标式为：

$$\min\sum_{i=1}^{n}d_i^2=\min\sum_{i=1}^{n}[y_i-f(x_i)]^2$$

其中$f(x)=ax+b$是拟合直线，(x,y)是待拟合的点坐标。

求解方法

1. 求偏导：求目标式关于a,b的偏导，得到如下极值条件：

$$\begin{cases}\frac{\partial\Pi}{\partial a}=2\sum_{i=1}^{n}[y_i-(a+bx_i)]=0\\ \frac{\partial\Pi}{\partial b}=2\sum_{i=1}^{n}x_i[y_i-(a+bx_i)]=0\end{cases}$$

2. 矩阵形式 [13] ：将拟合直线$f(x)=ax+b$增广为矩阵形式$Y=X\beta$，在误差$d=Y-X\beta\to 0$时，有$Y=X\beta$，由于X不一定是方阵，有：

$$Y=X\beta\Rightarrow X^TY=X^TX\beta\Rightarrow\beta=(X^TX)^{-1}X^TY$$

返回目录

5.1.2 *Split-and-Merge*（分割与合并） [14]

1. 分裂（Split）：以全集作为初始点集。对当前点集拟合直线（采用端点拟合），计算到最远点的距离，距离大于阈值则在该最远点处将点集分裂为两个子集，并对分裂后的两个子集进行迭代。
2. 合并（Merge）：检查相邻线段是否满足合并要求（合并后是否有过远点），若满足要求，则合并并拟合新的直线。

5.1.3 *Line-Regression*（线性回归）

1. 滑动拟合：选取窗口，在其内采用最小二乘法拟合直线，之后滑动窗口拟合新的直线。
2. 合并：检查相邻线段是否满足合并的角度和距离要求，满足则合并并拟合新的直线，直到所有线段不可再合并。

5.1.4 *RANSAC*（*Random Sample Consensus*，随机抽样一致性算法） [15]

- 外点（outliers）：异常值。
- 内点（inliers）：符合模型的数据点。

1. 根据内点比例w和找到一个完全由内点组成的样本的希望概率p计算迭代次数：

$$k = \frac{\log(1-p)}{\log(1-w^2)}$$

2. 从所有数据点中随机选择最小数量（直线2点，平面3点）的数据点子集，确定唯一的模型参数。
3. 计算剩余数据点与该模型的误差，小于设定阈值的为内点。
4. 重复迭代次数次采样，选取包含最多内点的模型。

5.1.5 *Hough-Transform*（霍夫变换）

图像空间中的一个点对应Hough空间中的一条线。激光定位任务中，常用极坐标$\rho = x\cos\theta + y\sin\theta$表示。在定距下，误差呈正态分布；而在变距下，误差增长与距离正相关。

1. 计算数据范围（Hough空间参数分辨率），并初始化累加器。

返回目录

2. 遍历边缘点，计算可能的参数组合，并在对应位置进行投票。
3. 在累加器中寻找峰值（可能不唯一），获得相应Hough空间参数。
4. 转换回图像空间，确定直线。

最小二乘直线拟合

点(ρ_i, θ_i)到拟合直线(r, α)的距离近似为d_i：

$$\rho_i \cos(\theta_i - \alpha) - r = d_i$$

使其加权平方和最小，得到最优拟合直线。
误差传播为：

$$C_x = \begin{bmatrix} \mathrm{diag}(\sigma_\rho^2) & 0 \\ 0 & \mathrm{diag}(\sigma_\theta^2) \end{bmatrix}_{2n\times 2n} \qquad F_{\rho\theta} = \begin{bmatrix} \frac{\partial\alpha}{\partial\rho_1} & \frac{\partial\alpha}{\partial\rho_2} & \cdots & \frac{\partial\alpha}{\partial\theta_{n-1}} & \frac{\partial\alpha}{\partial\theta_n} \\ \frac{\partial r}{\partial\rho_1} & \frac{\partial r}{\partial\rho_2} & \cdots & \frac{\partial r}{\partial\theta_{n-1}} & \frac{\partial r}{\partial\theta_n} \end{bmatrix}$$

$$C_{\alpha r} = F_{\rho\theta} C_x F_{\rho\theta}^T$$

5.1.6 性能对比

算法	复杂度	假阳性率FPR	精度	多直线检测	补充
Split-and-Merge	$n \log n$	低	低	适用	
Line-Regression	nn_f	低	低	适用	有序点云
RANSAC	Snk	高	高	需调整	容忍外点
Hough-Transform	$Snn_C + Sn_Rn_C$	高	高	适用	

表 6: 直线特征提取算法性能对比

5.2 定位与匹配

5.2.1 基于*SVD*的定位算法 [16]

条件与目标

在二维平面上，有基于世界坐标系的点云$P = \{p_1, p_2, \ldots, p_n\}$和基于激光坐标系的点云$Q = \{q_1, q_2, \ldots, q_n\}$，它们按下标顺序匹配，求解变换（旋转阵R和平移量t）。

返回目录

以误差平方加权和的形式建模，得到目标式：

$$(R, t) = \operatorname{argmin} \sum_{i=1}^{n} w_i ||(Rp_i + t) - q_i||^2$$

其中w_i表示匹配点对(p_i, q_i)的权重，可取为距离的倒数$w_i = \frac{1}{\sigma_i(\rho_i)}$。

加权平均

为求极值，对目标式求关于t的偏导：

$$2t(\sum_{i=1}^{n} w_i) + 2R(\sum_{i=1}^{n} w_i p_i) - 2\sum_{i=1}^{n} w_i q_i = 0 \xRightarrow[\sum_{i=1}^{n} w_i]{\text{等号两边同除}} t + R\frac{(\sum_{i=1}^{n} w_i p_i)}{\sum_{i=1}^{n} w_i} - \frac{\sum_{i=1}^{n} w_i q_i}{\sum_{i=1}^{n} w_i} = 0$$

取两个点云的加权中心点$\hat{p} = \frac{\sum_{i=1}^{n} w_i p_i}{\sum_{i=1}^{n} w_i}, \hat{q} = \frac{\sum_{i=1}^{n} w_i q_i}{\sum_{i=1}^{n} w_i}$，上式可化简为$t = \hat{q} - R\hat{p}$。该式描述了平移量和旋转阵的关系，将其带回目标式，得到单变量最值问题：

$$R = \operatorname{argmin} \sum_{i=1}^{n} w_i ||R(p_i - \hat{p}) - (q_i - \hat{q})||^2$$

去中心化

取去中心化$x_i = p_i - \hat{p}, y_i = q_i - \hat{q}$，目标式可化简为：

$$R = \operatorname{argmin} \sum_{i=1}^{n} w_i ||Rx_i - y_i||^2$$

将平方项变成矩阵相乘的形式：

$$||Rx_i - y_i||^2 = (Rx_i - y_i)^T(Rx_i - y_i) = x_i^T(R^TR)x_i - y_i^TRx_i - x_i^TR^Ty_i + y_i^Ty_i$$

- $R^{-1} = R^T \Rightarrow R^TR = E$。
- $x_i^TR^Ty_i$是一个1×1标量，转置后不变，故$x_i^TR^Ty_i = y_i^TRx_i$。

所以:

$$||Rx_i - y_i||^2 = x_i^Tx_i - 2y_i^TRx_i + y_i^Ty_i$$

其中仅有负号项与R相关，其它项都是定值，目标可改写为：

$$R = \operatorname{argmax} \sum_{i=1}^{n} w_i y_i^T R x_i$$

SVD分解

将其写成对角阵的迹的形式：

$$
\begin{aligned}
\sum_{i=1}^{n} w_i y_i^T R x_i &= tr(diag(w_1 y_1^T R x_1, w_2 y_2^T R x_2, \dots, w_n y_n^T R x_n)) \\
&= tr(diag(w_1, w_2, \dots, w_n) \begin{bmatrix} y_1^T & y_2^T & \vdots & y_n^T \end{bmatrix}^T R \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}) \\
&= tr(WY^T RX)
\end{aligned}
$$

因为矩阵的迹满足$tr(AB) = tr(BA)$，所以$tr(WY^T RX) = tr(RXWY^T)$。令$S = XWY^T$，基于SVD原理（见附录9.2），$S = U\Sigma V^T$，其中U, V是单位正交阵，Σ为对角阵。所以$tr(RXWY^T) = tr(RU\Sigma V^T) = tr(\Sigma V^T RU)$，后三者都是单位正交阵，它们的积$M$也是单位正交阵。目标改写为：

$$R = argmaxtr(\Sigma M)$$

求解变换 单位正交阵的最大迹在单位阵下取得的，有$M = E$，即$R = VU^T$。代回确定t。

5.2.2 基于*ICP*（*Iterative Closest Point*，迭代最近点）的点云匹配算法 [17]

1. 条件与目标：求解二维平面上检测点云$P = \{p_1, p_2, \dots, p_{N_p}\}$和目标点云$X = \{x_1, x_2, \dots, x_{N_x}\}$的匹配。
2. 计算最近点集：采取采样方法获得目标点云，并采用点集匹配方法为检测点云数据点匹配最近的目标点云数据点。
   - 均匀采样。
   - 随机采样。
   - 基于特征的采样。
   - 法向量空间采样。
   - Closest point Matching（CMP）。
   - Normal Shooting Matching。
   - Point-to-Plane Matching。
   - Projection Matching。
3. 变换：使用基于SVD的定位算法求解齐次变换矩阵并应用于检测点云。
4. 目标函数计算：统计对齐误差，如果达到阈值则停止迭代，否则重复上述操作。

6 机器人定位

6.1 定位与导航

导航 不能碰障碍物，掌握目标的方向。

- 基于行为：如沿墙边前进。
- 基于地图：已知地图，需要定位。

定位

- 问题
  - 全局定位：未知初始位置，根据地图进行定位。
  - 位置跟踪：已知初始位置，跟踪位置变化。
  - 绑架问题。
- 方法：基于机载传感器、基于额外传感器和路标、里程计。
- 分类（不确定度分布）：连续单峰（卡尔曼滤波）、连续多峰、离散多峰（粒子滤波）、拓扑。

6.2 贝叶斯定位

原理 新信息出现后的概率 = 概率 × 新信息带来的调整：

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)}$$

思想 使用低精度传感器（如里程计）跟踪运动状态，不确定度不断提高，定期使用高精度传感器（如激光雷达），修正估计。

特点

返回目录

- 连续型
  - 精度受传感器数据限制。
  - 通常是单一假设位姿估计，发散时丢失。
  - 表示紧凑，计算资源需求合理。
- 离散型
  - 精度受离散化分辨率限制。
  - 通常是多假设位姿估计，发散时收敛到另一单元，永不丢失。
  - 需大量内存和计算资源。

6.3 基于卡尔曼滤波的定位

6.3.1 卡尔曼滤波

推导 两次独立测量的概率均服从正态分布$p_1(q) = N(\hat{q}_1, \sigma_1^2), p_2(q) = N(\hat{q}_2, \sigma_2^2)$，它们整合得到的最终分布也服从正态分布：

$$
\begin{aligned}
p(q) = p_1(q) \cdot p_2(q) &= \frac{1}{\sigma_1\sqrt{2\pi}} \exp(-\frac{(q-\hat{q}_1)^2}{2\sigma_1^2}) \cdot \frac{1}{\sigma_2\sqrt{2\pi}} \exp(-\frac{(q-\hat{q}_2)^2}{2\sigma_2^2}) \\
&= \frac{1}{2\sigma_1\sigma_2\pi} \exp[-\frac{(q-\hat{q}_1)^2}{2\sigma_1^2} - \frac{(q-\hat{q}_2)^2}{2\sigma_2^2}] \\
&= \frac{1}{2\sigma_1\sigma_2\pi} \exp\{-\frac{1}{2}[\frac{q^2(\sigma_1^2+\sigma_2^2) - 2q(\hat{q}_1\sigma_2^2 + \hat{q}_2\sigma_1^2) + (\hat{q}_1^2\sigma_2^2 + \hat{q}_2^2\sigma_1^2)}{\sigma_1^2\sigma_2^2}]\} \\
&= \frac{1}{2\sigma_1\sigma_2\pi} \exp\{-\frac{1}{2}[\frac{q^2 - \frac{2q(\hat{q}_1\sigma_2^2+\hat{q}_2\sigma_1^2)}{\sigma_1^2+\sigma_2^2} + \frac{(\hat{q}_1^2\sigma_2^2+\hat{q}_2^2\sigma_1^2)}{\sigma_1^2+\sigma_2^2}}{\frac{\sigma_1^2\sigma_2^2}{\sigma_1^2+\sigma_2^2}}]\} \\
&= \underbrace{\frac{1}{2\sigma_1\sigma_2\pi} \exp[-\frac{1}{2}\frac{\frac{\hat{q}_1^2\sigma_2^2+\hat{q}_2^2\sigma_1^2}{\sigma_1^2+\sigma_2^2} - (\frac{\hat{q}_1\sigma_2^2+\hat{q}_2\sigma_1^2}{\sigma_1^2+\sigma_2^2})^2}{\frac{\sigma_1^2\sigma_2^2}{\sigma_1^2+\sigma_2^2}}]}_{\text{常数}} \underbrace{\exp[-\frac{1}{2}\frac{(q - \frac{\hat{q}_1\sigma_2^2+\hat{q}_2\sigma_1^2}{\sigma_1^2+\sigma_2^2})^2}{\frac{\sigma_1^2\sigma_2^2}{\sigma_1^2+\sigma_2^2}}]}_{\text{正态分布}} \\
&= N(\hat{q}, \sigma^2)
\end{aligned}
$$

有：

$$
\hat{q} = \underbrace{\frac{\hat{q}_1\sigma_2^2 + \hat{q}_2\sigma_1^2}{\sigma_1^2+\sigma_2^2}}_{\text{方差小的项权重大}} \Leftrightarrow \hat{q} = \hat{q}_1 + \frac{\sigma_1^2}{\sigma_1^2+\sigma_2^2}\underbrace{(\hat{q}_2 - \hat{q}_1)}_{\text{校正量}}
$$

$$
\sigma^2 = \frac{\sigma_1^2\sigma_2^2}{\sigma_1^2+\sigma_2^2} \Leftrightarrow \sigma^2 = \underbrace{\sigma_1^2 - \frac{\sigma_1^4}{\sigma_1^2+\sigma_2^2}}_{<\sigma_1^2,\sigma_2^2}
$$

返回目录

以$P=\sigma_1^2, Q=\sigma_2^2, R=\sigma^2$，记卡尔曼增益$K=P(P+Q)^{-1}$和创新协方差$\Sigma_{IN}=P+Q$：

$$\begin{aligned}\hat{q} &= \hat{q}_1 + P(P+Q)^{-1}(\hat{q}_2-\hat{q}_1) = \hat{q}_1 + K(\hat{q}_2-\hat{q}_1)\\ R &= P - P(P+Q)^{-1}P = P - K\cdot\Sigma_{IN}\cdot K^T\end{aligned}$$

有模型：

$$\begin{aligned}&\text{过程（推算）方程：} x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}\\ &\text{测量方程：} z_k = Hx_k + v_k\end{aligned}$$

其中$x_k\in R^n$为系统状态，$z_k\in R^m$为测量输出，$u_k\in R^l$为系统输入。$w_k\in R^n$为过程噪声（Process Noise）$p(w)=N(0,Q)$，$v_k\in R^m$为测量噪声（Measurement Noise，白噪声）$p(v)=N(0,R)$。

有先/后验估计$\hat{x}_k^-,\hat{x}_k$，先/后验估计误差$e_k^-=x_k-\hat{x}_k^-, e_k=x_k-\hat{x}_k$，先/后验估计方差$P_k^-=E[e_k^-e_k^{-T}], P_k=E[e_ke_k^T]$。其中，优化目标是$\mathrm{argmin}_K P_k$。

先后验转化关系为：

$$\hat{x}_k = \hat{x}_k^- + \underbrace{K}_{\text{卡尔曼增益}}\underbrace{(z_k-H\hat{x}_k^-)}_{\text{新息}}$$

P_k可进行转化：

$$\begin{aligned}P_k &= E[e_ke_k^T] = E[(x_k-\hat{x}_k)(x_k-\hat{x}_k)^T](\text{误差展开})\\ &= E[\{x_k-[\hat{x}_k^-+K(z_k-H\hat{x}_k^-)]\}\{x_k-[\hat{x}_k^-+K(z_k-H\hat{x}_k^-)]\}^T](\text{代入先后验转化关系})\\ &= E\{[(x_k-\hat{x}_k^-)-K(Hx_k+v_k-H\hat{x}_k^-)][(x_k-\hat{x}_k^-)-K(Hx_k+v_k-H\hat{x}_k^-)]^T\}(\text{代入测量方程})\\ &= E\{[(I-KH)e_k^- - Kv_k][(I-KH)e_k^- - Kv_k]^T\}(\text{误差重构})\\ &= (I-KH)\underbrace{E[e_k^-e_k^{-T}]}_{P_n^-}(I-KH)^T + K\underbrace{E[v_kv_k^T]}_{R}K^T - K\underbrace{E[v_ke_k^{-T}]}_{\text{不相干，0}}(I-KH)^T - (I-KH)\underbrace{E[e_k^-v_k^T]}_{\text{不相干，0}}K^T\\ &= (I-KH)P_k^-(I-KH)^T + KRK^T\end{aligned}$$

令偏导为0：

$$\frac{\partial P_k}{\partial K} = -2P_k^-H^T + 2KHP_k^-H^T + 2KR = 0$$

有$K=P_k^-H^T(HP_k^-H^T+R)^{-1}$，在方差趋于0时，$\lim_{R\to 0}K=H^{-1}, \lim_{P_k^-\to 0}K=0$。

步骤 [18]

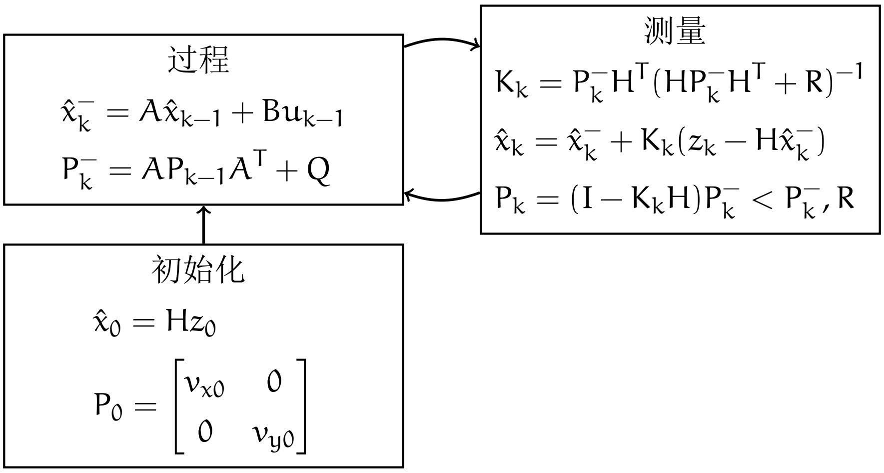


图 11: 卡尔曼滤波框图

6.3.2 基于卡尔曼滤波的定位算法 [19]

步骤 Prediction → Observation → Estimation

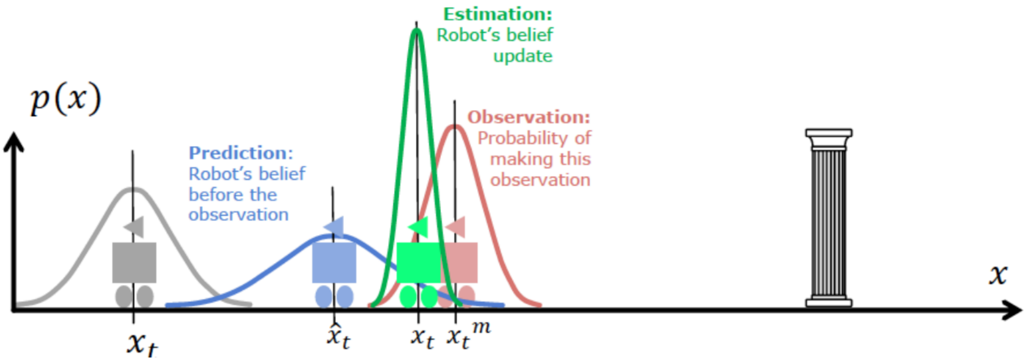


图 12: 基于卡尔曼滤波的定位算法示意图

1. 先验估计：预测模型，如里程计模型。
2. 先验估计误差：误差传导模型。
3. 观测：如激光定位，采用SVD确定位姿，采用ICP配准。
4. 观测误差：SVD匹配误差。
5. 基于观测的后验估计：卡尔曼滤波。

返回目录

状态	状态转移A	测量	测量矩阵H	性能
$\begin{bmatrix} x_k \\ y_k \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} u_k \\ v_k \end{bmatrix}$	$\begin{bmatrix} u_k \\ v_k \end{bmatrix}$	动态误差大 静止误差小 适合缓慢移动
$\begin{bmatrix} x \\ y \\ \frac{dx}{dt} \\ \frac{dy}{dt} \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$		$\begin{bmatrix} H_x & 0 & 0 & 0 \\ 0 & H_y & 0 & 0 \end{bmatrix}$ 未测量速度	动态误差小 静止误差大 低维估计高维

表 7: 卡尔曼滤波算法描述维度

描述维度

特点

- 严重依赖匹配精度，匹配失败则定位失败，且无法判断和恢复。
- 收敛速度受初始状态误差和协方差阵精度影响较大。
- 无法全局定位，无法应对绑架问题，只能位置跟踪。
- 不确定度需为单峰高斯分布。

6.4 蒙特卡洛定位

6.4.1 蒙特卡洛方法（*Monte Carlo Method*）

不断抽样，逐渐逼近。通用，但收敛速度慢，不精确。

分布函数拟合　用样本（粒子）分布处理各种分布。

- 已知概率分布求粒子分布：采样排除法，概率高的地方采样密集。
- 未知概率分布求粒子分布：重要性评估，迭代，根据预估分布（以均匀分布开始）撒粒子，再根据结果调整重要性评估（归一化）。

6.4.2 蒙特卡洛定位算法（粒子滤波算法，*MCL*）[20]

多信息融合，提高真值处重要性。

返回目录

重要性采样

目标分布：

$$p(x_i|z_1,z_2,\cdots,z_n)=\frac{\prod_k p(z_k|x_i)p(x_i)}{p(z_1,z_2,\cdots,z_n)}$$

建议分布：

$$p(x_i|z_i)=\frac{p(z_i|x_i)p(x_i)}{p(z_i)}$$

有重要性权重：

$$w_i=\frac{p(x_i|z_1,z_2,\cdots,z_n)}{p(x_i|z_i)}=\frac{p(z_i)\prod_{k\neq i}p(z_k|x_i)}{p(z_1,z_2,\cdots,z_n)}\propto\prod_{k\neq i}p(z_k|x_i)$$

步骤 在一次迭代（$X_{t-1}\to X_t$）中，经历以下过程：

1. 预测：基于运动模型（建议分布）对粒子进行更新和采样：

$$x_t^{(i)}=f(x_{t-1}^{(i)},u_t)$$

2. 校正：使用传感器模型计算粒子权重：

$$w_t^{(i)}=p(z_t|x_t^{(i)})$$

3. 重采样：根据归一化权重进行重采样（采用轮盘赌），权重越大概率越高。

激光雷达观测误差模型 m指地图。

- 误差
  - 激光雷达测量误差$p_{hit}(z_t|x_t,m)$。
  - 没有检测到障碍物$p_{max}(z_t|x_t,m)$：常值分布，激光雷达最大测距距离。

$$p_{max}(z_t|x_t,m)=\begin{cases}1 & z_t^k=z_{max}\\ 0 & \text{otherwise}\end{cases}$$

  - 随机错误$p_{rand}(z_t|x_t,m)$：均匀分布，错误的距离值。

$$p_{rand}(z_t|x_t,m)=\begin{cases}\frac{1}{z_{max}} & 0\leqslant z_t^k\leqslant z_{max}\\ 0 & \text{otherwise}\end{cases}$$

    - 激光雷达观测误差模型$p(z_t|x_t, m)$：上述误差的线性组合。

$$p(z_t|x_t, m) = \alpha_{hit} p_{hit} + \alpha_{max} p_{max} + \alpha_{rand} p_{rand}, \quad \alpha_{hit} + \alpha_{max} + \alpha_{rand} = 1$$

- 物理模型：$p_{hit}(z_t|x_t, m)$包含随机噪声，一般为高斯分布。

$$p_{hit}(z_t|x_t, m) = \begin{cases} N(z_t^{k*}, \sigma_{hit}^2) & 0 \leqslant z_t^k \leqslant z_{max} \\ 0 & otherwise \end{cases}$$

- 可能域（likelihood）：将激光束投影到地图中，$p_{hit}(z_t|x_t, m)$以距投影最近的点的距离为均值，激光测距偏差为方差。
    - 优点：在线计算量小，平滑性好，收敛性强，更符合实际情况。
    - 缺点：没有明确的物理意义，仅适用于静态环境（如路标工厂）。

特点

- 优点：计算简单，可解决大范围全局定位问题，适用于多种分布。
- 缺点：粒子数过大时占用内存大，计算效率低；大地图下，只用少量粒子可能发散；无法应对绑架问题。

6.4.3 自适应蒙特卡洛定位算法（AMCL） [21]

改进 引入短、长期指数滤波器衰减率$\alpha_{slow} \ll \alpha_{fast}$，计算短、长期重要性指数似然评价估计$w_{slow}, w_{fast}$，二者计算公式格式一致，正常时$w_{slow} < w_{fast}$。

解决问题

- 绑架问题：发生绑架时，w_{avg}会突然下降，导致$w_{slow} > w_{fast}$，将按照概率$\max(0, 1 - \frac{w_{fast}}{w_{slow}})$向粒子集中注入随机粒子。
- 粒子数问题：使用KLD（Kullback-Leibler Divergence，库尔贝克-莱布勒散度，计算概率分布间差异）采样，在收敛过程中减少粒子。

算法 标红部分是AMCL相较MCL的改进。

算法 1：AMCL

参数：$\alpha_{slow}, \alpha_{fast}$
初始化：初始化$\overline{X}_t = X_t$为空，初始化w_{slow}, w_{fast}
对于 $m = 1, \ldots, M$ 执行
  $x_t^{[m]} = f(u_t, x_{t-1}^{[m]})$
  $w_t^{[m]} = p(z_t, x_t^{[m]}, m)$
  $\overline{X}_t = \overline{X}_t + \langle x_t^{[m]}, w_t^{[m]} \rangle$
  $w_{avg} = w_{avg} + \frac{1}{M} w_t^{[m]}$ ▷ 平均权重
$w_{slow} = w_{slow} + \alpha_{slow}(w_{avg} - w_{slow})$ ▷ 慢速权重
$w_{fast} = w_{fast} + \alpha_{fast}(w_{avg} - w_{fast})$ ▷ 快速权重
对于 $m = 1, \ldots, M$ 执行
  以概率$\max(0, 1 - \frac{w_{fast}}{w_{slow}})$向$X_t$添加随机姿态 ▷ 应对绑架问题
  否则，从$\overline{X}_t$中按概率$w_t^{[m]}$采样$x_t^{[m]}$，并将其添加到X_t ▷ 重采样
返回X_t

7 机器人建图

7.1 地图

功能

- 支持机器人进行定位、导航、规划。
- 容易加入新的信息进行更新。
- 便于计算机储存和处理。

表示法

- 点云地图
  - 优点：可以完全表示环境三维信息，无需预定义尺寸。
  - 缺点：存储要求高，存在盲区和空洞，需判断占用和联通状态，不具有通用性。

返回目录

- 栅格地图：以允许误差大小确定栅格大小，用$0-1$的值表示栅格被占用的概率（不同激光数据中占用的概率）。
  - 计算
    - 概率p：$p(m_i|s_n)=1-E_rE_\alpha$，其中：

$$E_r=1-k_r(\frac{2(\rho_i-r_i)}{\Delta r})^2, E_\alpha=1-k_\alpha(\frac{2(\theta_i-\alpha)}{\Delta\alpha})^2$$

    - 几率l：几率 $=\frac{概率}{1-概率}$，几率与概率非线性正相关。

      利用贝叶斯公式得到每次测量（相互独立）的递推关系：

$$\begin{aligned}p(m_i|s_1,s_2,\cdots,s_n)&=\frac{p(s_n|m_i,s_1,s_2,\cdots,s_{n-1})p(m_i|s_1,s_2,\cdots,s_{n-1})}{p(s_n|s_1,s_2,\cdots,s_{n-1})}\\&=\frac{p(s_n|m_i)p(m_i|s_1,s_2,\cdots,s_{n-1})}{p(s_n)}\\&=\frac{p(m_i|s_n)p(s_n)p(m_i|s_1,s_2,\cdots,s_{n-1})}{p(m_i)p(s_n)}\\&=\frac{p(m_i|s_n)p(m_i|s_1,s_2,\cdots,s_{n-1})}{p(m_i)}\end{aligned}$$

      有对数几率更新公式：

$$\begin{aligned}l_{i,n}=\log\frac{p(m_i|s_1,s_2,\cdots,s_n)}{p(\bar{m}_i|s_1,s_2,\cdots,s_n)}&=\log\frac{p(m_i|s_n)p(\bar{m}_i)p(m_i|s_1,s_2,\cdots,s_{n-1})}{p(\bar{m}_i|s_n)p(m_i)p(\bar{m}_i|s_1,s_2,\cdots,s_{n-1})}\\&=\log\frac{p(m_i|s_n)}{p(\bar{m}_i|s_n)}+\underbrace{\log\frac{p(\bar{m}_i)}{p(m_i)}}_{l_0与测量无关}+\log\frac{p(m_i|s_1,s_2,\cdots,s_{n-1})}{p(\bar{m}_i|s_1,s_2,\cdots,s_{n-1})}\\&=\log\frac{p(m_i|s_n)}{p(\bar{m}_i|s_n)}+\underbrace{l_0}_{占用情况未知时，令p(m_i)=0.5,l_0=0}+l_{i,n-1}\end{aligned}$$

  - 特点
    - 优点：可以详细描述环境信息，易于定位和路径规划，无需预定义尺寸。
    - 缺点：存储要求高，存在盲区和空洞，需判断占用和联通状态，不具有通用性。
  - 拓展
    - 2.5维占用栅格地图（扩展高度图）：为每个栅格附加高度信息（障碍物最高高度）。

    * 3维占用栅格地图（Voxel Map，体素地图）：将空间分解为正方体，判断占用情况。
    * 多分辨率地图：障碍质密处栅格小，无障碍处栅格大，空间占用效率高，计算复杂度高，存储存在稀疏特征。
- 特征（语义）地图：连续多边形地图，空间占用效率高，定位精度高。
- 拓扑地图：节点和连线的拓扑结构图，便于导航和路径规划，难以精确定位。

7.2 SLAM（Simultaneous Localization and Mapping，同步定位与建图）

问题描述

- 定位（Localization）：根据观测序列、运动序列和地图，确定位姿。误差源于全局定位的初始误差和局部定位的观测误差。

$$E[X^t|Z^t, U^{t-1}, m]$$

- 建图（Mapping）：根据观测序列和位姿，构建地图。误差源于观测噪声。

$$E[m|Z^t, X^t]$$

- SLAM：根据观测序列、运动序列，确定位姿并构建地图。

$$E[X^t, m|Z^t, U^{t-1}]$$

7.3 基于滤波的SLAM算法

GMAPPING包。

7.3.1 *EKF-SLAM（扩展卡尔曼滤波SLAM）* [22]

第一个SLAM算法，面向特征地图，采用maximum likelihood进行数据关联。

返回目录

状态 机器人位姿$X_R = \begin{bmatrix} x & y & \theta \end{bmatrix}^T$，地图特征$M_i = \begin{bmatrix} x_i & y_i \end{bmatrix}^T$。

$$x_t = \begin{bmatrix} X_R \\ M_1 \\ M_2 \\ \vdots \\ M_n \end{bmatrix}_{(3+2n)\times 1} \qquad \Sigma_t = \begin{bmatrix} \Sigma_{X_R} & \Sigma_{X_R m_1} & \Sigma_{X_R m_2} & \cdots & \Sigma_{X_R m_n} \\ \Sigma_{m_1 X_R} & \Sigma_{m_1} & \Sigma_{m_1 m_2} & \cdots & \Sigma_{m_1 m_n} \\ \Sigma_{m_2 X_R} & \Sigma_{m_2 m_1} & \Sigma_{m_2} & \cdots & \Sigma_{m_2 m_n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \Sigma_{m_n X_R} & \Sigma_{m_n m_1} & \Sigma_{m_n m_2} & \cdots & \Sigma_{m_n} \end{bmatrix}_{(3+2n)\times(3+2n)}$$

高斯协方差矩阵表征了高度互相关性。

步骤

1. 初始状态：$X_0 = 0, \Sigma_0 = diag(\infty)$。
2. 基于运动模型进行状态估计：地图保持不变，与其相关项为0。

$$\bar{X}_t = f'(X_{t-1}, U_{t-1})$$

$$\bar{\Sigma}_{X_R} = \nabla f \cdot \Sigma_{R_{t-1}} \cdot \nabla f^T + R_t, \quad \bar{\Sigma}_t = \nabla f' \cdot \Sigma_{R_{t-1}} \cdot \nabla f'^T + F_x^T R_t F_x, \quad \nabla f' = \begin{bmatrix} \nabla f & 0 \\ 0 & I \end{bmatrix}$$

3. 基于观测模型进行预估观测：每个路标观测独立，交叉项为0。

$$\hat{Z}_t = \begin{bmatrix} Z_{1t} \\ Z_{2t} \\ \vdots \\ Z_{nt} \end{bmatrix}, \quad Q_t = \begin{bmatrix} Q_{1t} & 0 & \cdots & 0 \\ 0 & Q_{2t} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & Q_{nt} \end{bmatrix}$$

4. 建立状态估计和预估观测的数据关联。

$$K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1}$$

5. 状态更新。

$$X_t = \bar{X}_t + K_t[\hat{Z}_t - h(\bar{X}_t)], \quad \Sigma_t = [I - K_t H_t]\bar{\Sigma}_t$$

6. 获得实际观测：全局地图 = 全局观测位置 + 机器人测量数据。

   机器人系的激光观测数据$z_t^i = [r_t^i, \phi_t^i]$，投影到世界坐标系：

$$\begin{pmatrix} \bar{x}_m \\ \bar{y}_m \end{pmatrix} = \begin{pmatrix} \bar{x}_R \\ \bar{y}_R \end{pmatrix} + \begin{pmatrix} r_t^i \cos(\phi_t^i + \bar{\theta}_R) \\ r_t^i \sin(\phi_t^i + \bar{\theta}_R) \end{pmatrix}$$

返回目录

转换得到：

$$\delta = \begin{pmatrix} \delta_x \\ \delta_y \end{pmatrix} = \begin{pmatrix} \bar{x}_m - \bar{x}_R \\ \bar{y}_m - \bar{y}_R \end{pmatrix}, \quad q = \delta^T \delta, \quad \hat{z}_t^i = \begin{pmatrix} \sqrt{q} \\ \operatorname{atan2}(\delta_y, \delta_x) - \bar{\theta}_R \end{pmatrix}$$

有单路标雅可比：

$$\begin{aligned} H_i^j &=^{low} H_i^j F_{x,j} \\ &= \frac{1}{q} \begin{pmatrix} -\sqrt{q}\delta_x & -\sqrt{q}\delta_y & 0 & \sqrt{q}\delta_x & \sqrt{q}\delta_y \\ \delta_y & -\delta_x & -q & -\delta_y & \delta_x \end{pmatrix} \begin{pmatrix} I_3 & 0_{2j-2} & 0 & 0_{2N-2j} \\ 0 & 0 & I_2 & 0 \end{pmatrix} \end{aligned}$$

7. 在地图中加入新的路标：新路标与原路标不独立，因为前者由不确定度决定，而不确定度有后者计算。

$$\begin{aligned} m_{n+1} &= f(X_R, Z_{n+1}) \\ \nabla f_R &= \frac{\partial f(X_R, Z_{n+1})}{\partial X_R}, \quad \nabla f_Z = \frac{\partial f(X_R, Z_{n+1})}{\partial Z_{n+1}} \\ \Sigma_{m_{n+1}} &= \nabla f_R \cdot \Sigma_{X_R} \cdot \nabla f_R^T + \nabla f_Z \cdot Q_{Z_{n+1}} \cdot \nabla f_Z^T \\ \Sigma_{m_{n+1} m_i} &= \nabla f_R \cdot \Sigma_{X_R m_i} \\ \Sigma_{m_{n+1} X_R} &= \nabla f_R \cdot \Sigma_{X_R} \end{aligned}$$

得到新的状态（标红部分为扩充内容）：

$$X_t = \begin{bmatrix} X_R \\ M_1 \\ M_2 \\ \vdots \\ M_n \\ \color{red}{M_{n+1}} \end{bmatrix}, \quad \Sigma_t = \begin{bmatrix} \Sigma_{X_R} & \Sigma_{X_R m_1} & \cdots & \Sigma_{X_R m_n} & \color{red}{\Sigma_{X_R m_{n+1}}} \\ \Sigma_{m_1 X_R} & \Sigma_{m_1} & \cdots & \Sigma_{m_1 m_n} & \color{red}{\Sigma_{m_1 m_{n+1}}} \\ \Sigma_{m_2 X_R} & \Sigma_{m_2 m_1} & \cdots & \Sigma_{m_2 m_n} & \color{red}{\Sigma_{m_2 m_{n+1}}} \\ \vdots & \vdots & \ddots & \vdots & \color{red}{\vdots} \\ \Sigma_{m_n X_R} & \Sigma_{m_n m_1} & \cdots & \Sigma_{m_n} & \color{red}{\Sigma_{m_n m_{n+1}}} \\ \color{red}{\Sigma_{m_{n+1} X_R}} & \color{red}{\Sigma_{m_{n+1} m_1}} & \color{red}{\cdots} & \color{red}{\Sigma_{m_{n+1} m_n}} & \color{red}{\Sigma_{m_{n+1}}} \end{bmatrix}$$

8. 重复2-7步。

特点

- 计算复杂度高。
- 要求高斯分布，只能处理单峰假设。
- 依赖数据关联正确性，错误匹配会使算法发散。
- 仅适用于有全连接人工路标的点云地图或特征地图，无法使用栅格地图。

返回目录

7.3.2 FastSLAM（粒子滤波SLAM） 23

集成粒子滤波器与EKF，使用粒子分布表示位姿。

粒子降维 每个粒子相当于对机器人路径的一个假设，构建独立的地图，协方差矩阵只有对角线。

$$x = (\underbrace{x_1, \cdots, x_t}_{3t\text{，撒粒子获得}}, \underbrace{l_{1,x}, l_{1,y}, \cdots, l_{N,x}, l_{N,y}}_{2N\text{，计算获得}})^T$$

基于饶布莱克维尔定理进行因式分解：

$$\begin{aligned} p(x_{0:t}, \underbrace{l_{1:M}}_{\text{地图（路标）}} | z_{1:t}, u_{1:t}) &= \underbrace{p(x_{0:t}|z_{1:t}, u_{1:t})}_{\text{根据观测定位}} \underbrace{p(l_{1:M}|x_{0:t}, z_{1:t}, u_{1:t})}_{\text{根据观测建图}} \\ &= \underbrace{p(x_{0:t}|z_{1:t}, u_{1:t})}_{\text{粒子滤波}} \underbrace{\Pi_{i=1}^{M} p(l_i|x_{0:t}, z_{1:t}, u_{1:t})}_{\text{M个独立2维EKF}} \end{aligned}$$

其中粒子滤波部分不需要修正过去位姿信息，只需要维护当前位姿。

步骤

1. 运动更新。
2. 计算新特征的EKF。
3. 地图更新。
4. 基于观测计算权重：仅与当前观测相关。

$$\begin{aligned} w^{[k]} &= \frac{\overbrace{p(x_{1:t}^{[k]}|z_{1:t}, u_{1:t})}^{\text{目标分布}}}{\underbrace{p(x_{1:t}^{[k]}|z_{1:t-1}, u_{1:t})}_{\text{建议分布}}} \\ &= \frac{\eta p(z_t|x_{1:t}^{[k]}, z_{1:t-1}) p(x_t^{[k]}|x_{t-1}, u_t) p(x_{1:t-1}^{[k]}|z_{1:t-1}, u_{1:t-1})}{p(x_t^{[k]}|x_{t-1}, u_t) p(x_{1:t-1}^{[k]}|z_{1:t-1}, u_{1:t-1})} \\ &= \eta p(z_t|x_{1:t}^{[k]}, z_{1:t-1}) \end{aligned}$$

5. 插入新特征。
6. 基于权重重采样。

返回目录

数据关联 计算观测特征和地图特征的匹配概率，准的粒子概率高，不会因关联错误而全盘错误。

- 选择概率大的进行匹配。
- 按照概率进行随机匹配。

特点

- 降维高效。
- 随机采样使数据关联更鲁棒，利用多假设分析忽略位姿误差。
- 粒子数量选取：尽量多，过多可能占用过大内存，过少可能导致错误。

7.4 基于优化的SLAM算法

图优化SLAM中，位姿通过约束相互连接，表征不确定性，寻求最优的节点构型使约束误差最小。g2o包，主流。

构建图（前端）

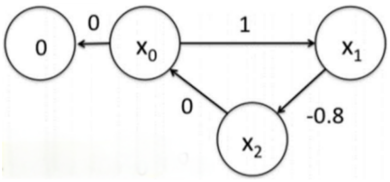
- 节点（Node）：机器人位姿，包含位姿$x_{1:n}(x, y, \theta)$、特征或路标$m_{1:k}(x, y)$信息。
  - 定位（Pose）：$x^T = (x_1^T, \cdots, x_n^T) \in R^{3n}$。
  - SLAM（Pose feature）：$x^T = (x_1^T, \cdots, x_n^T, m_1^T, \cdots, m_K^T) \in R^{3n+2K}$。
- 边（Edge）：位姿间的空间约束，齐次坐标。
  - 里程计测量$(X_i^{-1}X_{i+1})$和观测$(X_i^{-1}X_j)$。
  - 信息矩阵：节点间空间约束的不确定性，优化的权重。协方差的逆，值越大相关性越高，越应重视。

图优化（后端）

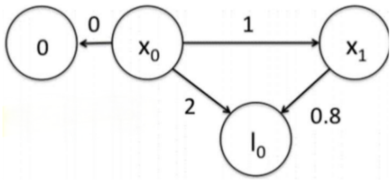
- 定位：$e_{ij}(x_i, x_j) = t2v[z_{ij}^{-1}(x_i^{-1}x_j)] \overset{e_{ij}(x_i,x_j)=0}{\Longrightarrow} z_{ij} = (x_i^{-1}x_{i+1})$。
- SLAM：$e_{ij}(z_i, x_j) = \hat{z}_{ij} - z_{ij} = R_j^T(z_i - t_j) - z_{ij} \overset{e_{ij}(x_i,x_j)=0}{\Longrightarrow} z_{ij} = R_i^T(z_i - t_j)$。

返回目录

例题 24



(a) 定位



(b) SLAM

图 13: 图优化例题

1. 定位：起点在0处，向前移动到达远点，编码器测得位移1m，向后移动回到起点（闭环检测），编码器测得位移0.8m。

   有目标函数：

$$c=\sum_{i=1}^{4} f_i^2=(x_0-0)^2+(x_1-x_0-1)^2+[x_2-x_1-(-0.8)]^2+(x_2-x_0-0)^2$$

   求偏导得到方程组，解得：

$$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix}=\begin{bmatrix} 3 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}^{-1}\begin{bmatrix} -1.0 \\ 1.8 \\ -0.8 \end{bmatrix}=\begin{bmatrix} 0 \\ 0.93 \\ 0.07 \end{bmatrix}$$

2. SLAM：起点在0处，观测到前方2m处有路标。向前移动到路标前0.8m处，编码器测得位移1m。

   有目标函数：

$$c=\sum_{i=1}^{4} f_i^2=(x_0-0)^2+(x_1-x_0-1)^2+(l_0-x_0-2)^2+(l_0-x_1-0.8)^2$$

   求偏导得到方程组，解得：

$$\begin{bmatrix} x_0 \\ x_1 \\ l_0 \end{bmatrix}=\begin{bmatrix} 3 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}^{-1}\begin{bmatrix} -3.0 \\ 0.2 \\ 2.8 \end{bmatrix}=\begin{bmatrix} 0 \\ 1.07 \\ 1.93 \end{bmatrix}$$

优化求解 略

特点

- 仅将实际发生关联的约束构建连接并进行优化，且不是每次观测都需要更新（多分辨率），效率高。
- 优化思想而非概率思想，易理解和计算。

7.5 LOAM

基于激光和里程计。

- 运动畸变矫正：点云时间戳对齐，利用IMU信息（无IMU时假设匀速）。
- 特征点提取：平面点、边缘点。
- 双odom：高频低精度和低频高精度结合。

8 机器人运动规划

8.1 运动规划（Motion Planning）

需求

- 安全性：规避障碍物。
- 可达性：从起点到终点的连通域。
- 光滑性：平稳舒适地运行。
- 可执行性：动力学约束和执行器约束。

分层规划

- 路径规划：只考虑几何约束，不考虑机器人和执行器约束。
  - 基于图搜索：算法完备，有限时间内返回解或返回无解。
  - 基于采样：概率完备。
- 轨迹规划：在给定路径上考虑机器人和执行器约束，生成随时间变化的运动序列。分在线和离线两种。

返回目录

8.2 基于图搜索的路径规划

8.2.1 静态路径规划

构建图

- 可视图（visibilty graph）：连线障碍物顶点和起点终点，找最短路径。可实现无碰撞最短路径距离规划，但太靠近障碍物。
- 维诺图（Voronoi graph）：路径距周围两个最近障碍物等距（L1/L2）。最大化与障碍物的距离，安全但较长。
- Bug算法图：起点和终点连线，在障碍物处沿边缘行走，更新起点迭代。
- 精确单元分解地图。
- 栅格地图：关联格间无障碍。
- 拓扑地图。
- 搜索树。

算法 [25]

算法	原理	完备性	最优性	补充
深度优先（DFS）	分支遍历	完备	不最优	无方向性，效率低下
宽/广度优先（BFS，泛洪搜索）	层次遍历	完备	不最优	无方向性，效率低下
Dijkstra	计算节点到起点的代价，优先遍历队列中代价最小的节点	完备	最优	地图信息利用不充分
A*	路径评价由移动开销G(n)和启发函数代价估计H(n)组合得到	完备	最优（需满足可采纳性和一致性）	高效，多解，不一定平滑

表 8: 图搜索算法对比

启发函数：

返回目录

- 可选曼哈顿距离、欧几里得距离、切比雪夫距离，是算法效率的重要因素。
- 最优性性质：
  - 可采纳性：低估代价。
  - 一致性：符合三角不等式。

算法补充

- 权重A*：为启发函数添加权重$\epsilon \geqslant 1$，限制扩展节点，提高效率，但不满足可采纳性，只能得到局部最优。
- ARA*：结合任意时间法和权重A*，快速得到次优，再优化。

8.2.2 动态路径规划

动态环境特点

- 运行过程中地图可能改变，有动态障碍物。
- 一般性环境，更符合实际。
- 需降低复杂度，提高实时性。

算法

- D*（动态A*）：由目标点向当前位置Dijkstra，利用变化前的地图信息进行动态规划。
- Fucussed D*：将D*的Dijkstra改为A*。
- LPA*：增量式搜索。
- D*-Lite：LPA*的动态形式。
- AD*：结合ARA*和D*-Lite。

8.3 基于采样的路径规划

概述

- 采样：在障碍物处密集采样。

返回目录

- 距离评估。
- 碰撞检测：凸障碍已比较完善，非凸障碍尚未完善。

PRM（概率路图） 略

RRT（快速探索随机树）

- RRT：迭代撒点，选择离目标近的点，按其方向步进（不一定到达），更新起点；最后到达终点附近，无法准确到达。完备而高效，但只能得到可行解而非最优解。
- RRT*：在路径的大圈中进行局部优化，重构节点关系，使局部最优。
- Informed RRT*：在起点终点连线为主轴的椭圆中进行优化。

8.4 面向碰撞的局部路径规划

- DWA（Dynamic Window Approach）：评价函数对方向角、空隙、速度进行加权。计算复杂度低，但可能不是全局最优。
- TEB（Timed Elastic Band）：连接起点终点为可变形路径，将约束视作引起形变的外力。

8.5 路径平滑

- 约束：初始、终止状态约束。曲线光滑。
- 曲线形态：圆弧、贝塞尔曲线。

8.6 轨迹规划

- 目标：在执行器约束下，沿给定光滑曲线给出可执行的运动命令序列。
- 约束：初始、终止状态约束，执行器约束，运动学约束。

返回目录

9 附录

9.1 误差转化展示

将误差传播协方差矩阵$\sum_p$转化成椭圆展示。

计算 取$\sum_p$左上二阶子阵$\sum_{p_{xy}} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{xy} & \sigma_{yy} \end{bmatrix}$，计算特征值$\lambda_1, \lambda_2$和特征向量$\vec{p}_1, \vec{p}_2$，其分别表示长短轴的大小和方向，圆心是$(x_k, y_k)$（直接对$\sum_p$求取特征根和特征向量，再取前两个，结果与其不同）。

意义

- 长短轴表示误差在不同方向上的大小，可通过开根号、乘系数等方法调节。
- 方向角表示误差传播的主要方向，体现了误差传播的各向异性。
- 椭圆大小反映了系统的误差范围，可按置信度缩放。

返回里程计12。

9.2 奇异值分解（Singular Value Decomposition，SVD）

根据$A_{m\times n}$计算$(A^TA)_{n\times n}$ 和$(AA^T)_{m\times m}$两个对称矩阵，求二者的特征值和特征向量：

$$A^TAv_i = \lambda_i v_i, i = 1, 2, \ldots, n \quad AA^Tu_i = \mu_i u_i, i = 1, 2, \ldots, m$$

其中，λ_i, μ_i是特征值，v_i, u_i是其对应的特征向量，分别组成$V = [v_1, v_2, \ldots, v_n]$和$U = [u_1, u_2, \ldots, u_m]$。

奇异值σ_i是λ_i, μ_i的平方根，将其按降序排列，并构造对角矩阵$\Sigma = diag(\sigma_1, \sigma_2, \ldots, \sigma_r)$，得到SVD分解$A = U\Sigma V^T$。

返回里程计5.2.1。