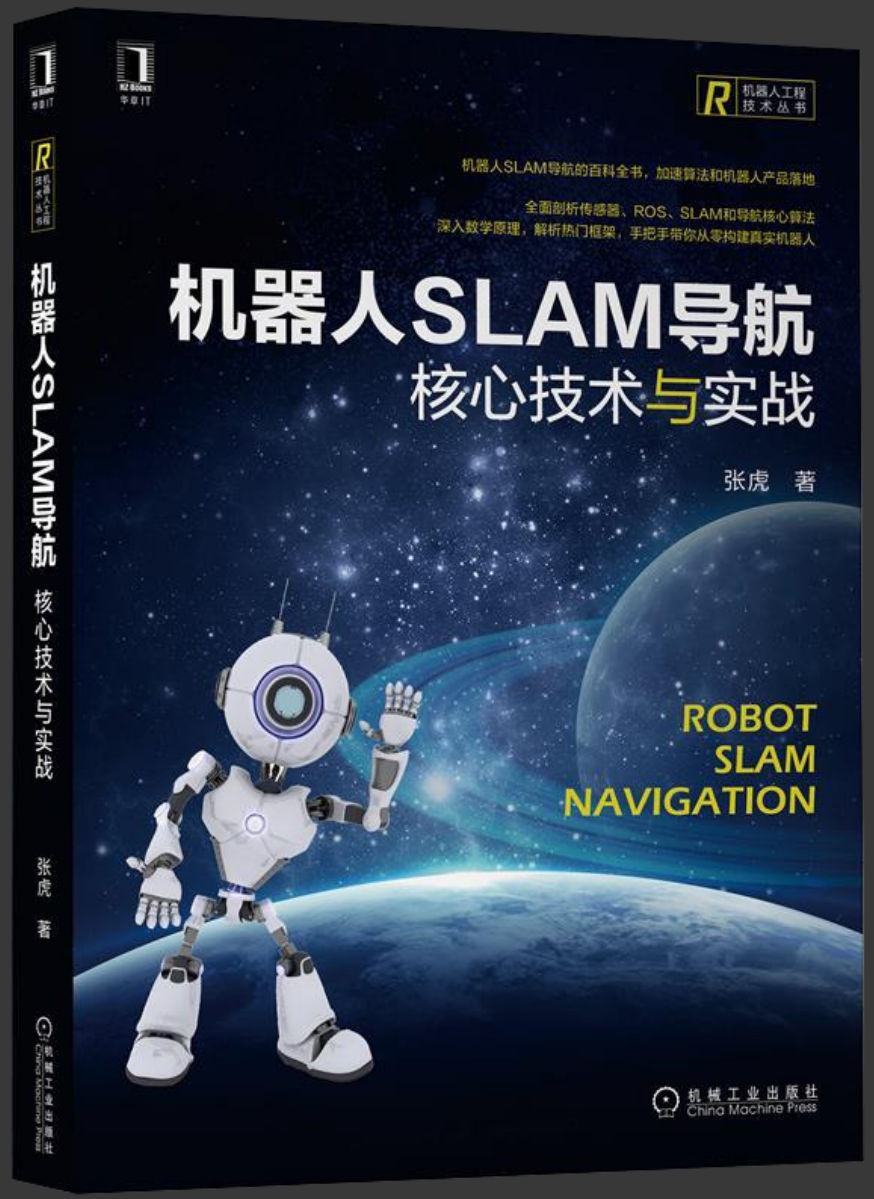


第1季

第7章：SLAM中的数学基础

主讲人：张虎

（小虎哥哥爱学习）

- 先导课
- 第1季：快速梳理知识要点与学习方法 √
- 第2季：详细推导数学公式与代码解析
- 第3季：代码实操以及真实机器人调试
- 答疑课

------（永久免费●系列课程●长期更新）------

本书内容安排

一、编程基础篇

第1章：ROS入门必备知识

第2章：C++编程范式

第3章：OpenCV图像处理

二、硬件基础篇

第4章：机器人传感器

第5章：机器人主机

第6章：机器人底盘

三、SLAM篇

第7章：SLAM中的数学基础

第8章：激光SLAM系统

第9章：视觉SLAM系统

第10章：其他SLAM系统

四、自主导航篇

第11章：自主导航中的数学基础

第12章：典型自主导航系统

第13章：机器人SLAM导航综合实战

搞SLAM为什么还要学编程知识和硬件知识？

SLAM，你真的了解吗？

- SLAM并不仅仅指代某种特定算法（求解方法）
- SLAM是一种问题的总称
- 你可能是一个SLAMer，也可能是码农、数学家、焊板子的、机械狗、嵌入式、AI调包侠、...

SLAM，你真的了解吗？

问题	求解方法（也叫算法）
1+1=?	• 计算器 • 算盘 • 笔算 • 心算 • 扳手指 • ...
SLAM （同时定位与建图）	• Gmapping • Cartographer • LOAM • ORB-SLAM • Rtabmap • ...

注意：不要以偏概全，某种算法（比如ORB-SLAM）并不能代表SLAM的全貌，你所做的是一个很细分的方向而已。

SLAM的价值是什么？

萌新：SLAM是不是不行了，搞SLAM还有前途吗？

大佬：

？

① SLAM是人类生产生活的客观需求，SLAM是一种问题。

② 只要SLAM问题一天不解决，那么搞SLAM就一定有前途。

③ 至少目前来看，SLAM问题还没有出现完美的解决方案。

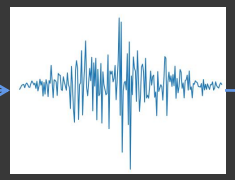
SLAM的价值是什么？

躯体使智能得以延展 → AI的最终归宿是机器人 → 机器人的完全自主化 → 自主移动技术 → SLAM

- 如果人工智能技术仅仅停留在虚拟的网络和数据之中，那么其挖掘并利用新知识的能力将很难扩展开来。
- 换句话说，人工智能需要真实的躯体。

SLAM的价值是什么？

躯体使智能得以延展 → AI的最终归宿是机器人 → 机器人的完全自主化 → 自主移动技术 → SLAM



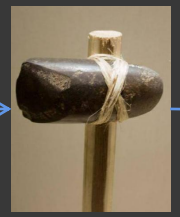
语言

文字

纸

互联网

人工智能

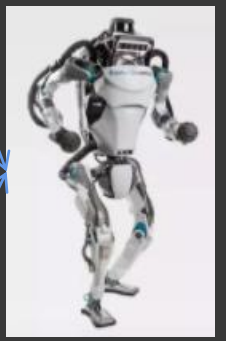


石器

铁器

汽车

飞机

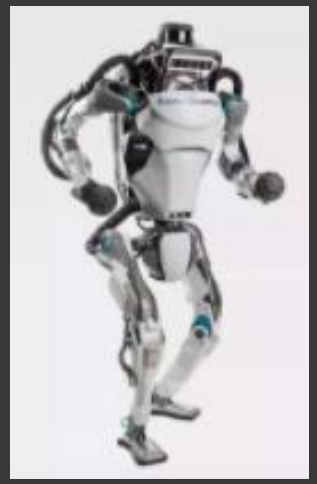


机器人

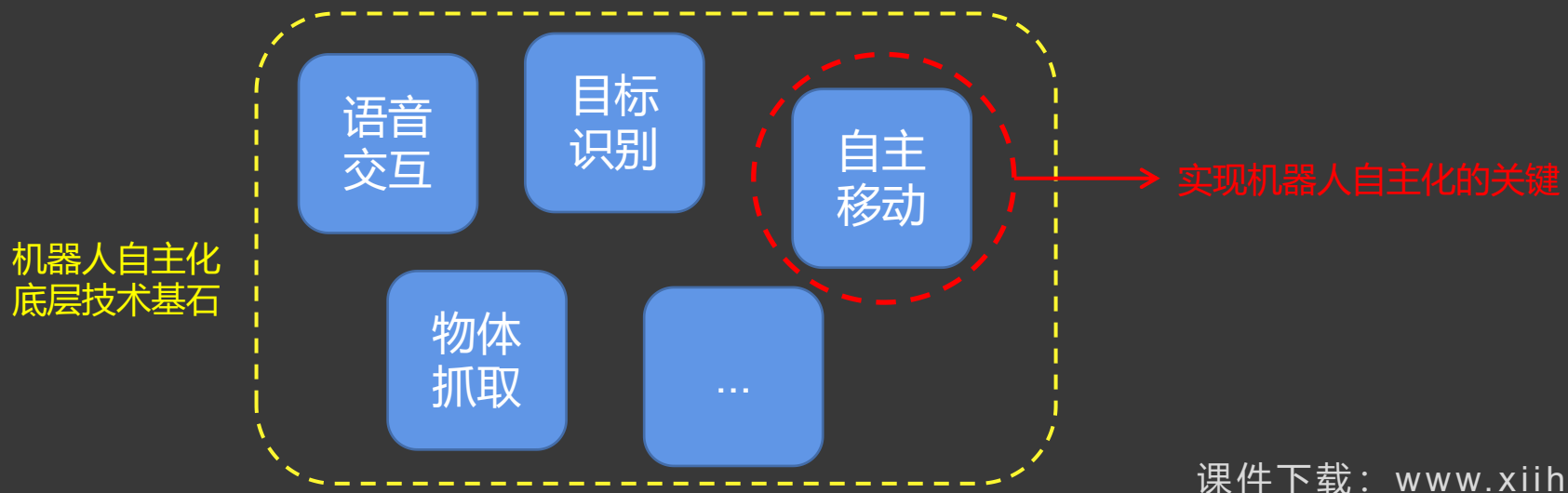
- 人形机器人
- 自动驾驶汽车
- 无人机
- 智能叉车
- ...

SLAM的价值是什么？

躯体使智能得以延展 → AI的最终归宿是机器人 → 机器人的完全自主化 → 自主移动技术 → SLAM



- ✓ 让机器人实现完全自主化也就成了人类一直以来的梦想。
- ✓ 没有外界指令的干预
- ✓ 机器人能通过传感器和执行机构与环境自动发生交互
- ✓ 完成特定的任务



SLAM的价值是什么？

躯体使智能得以延展 → AI的最终归宿是机器人 → 机器人的完全自主化 → 自主移动技术 → SLAM

为什么需要自主移动？

机械臂

VS

机械臂+移动底盘

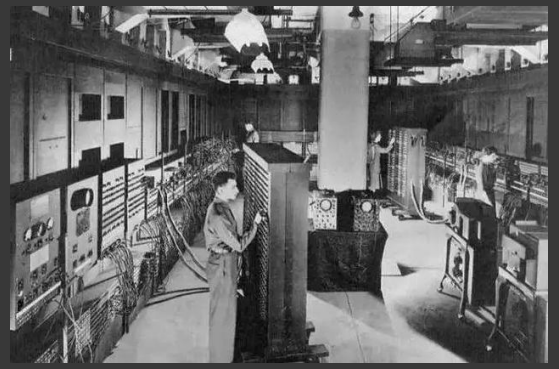
智能音响

VS

智能音响+移动底盘

SLAM的价值是什么？

躯体使智能得以延展 → AI的最终归宿是机器人 → 机器人的完全自主化 → 自主移动技术 → SLAM



早期计算机

智能手机

自主移动机器人

- 轮式底盘
- 双足人形机器人
- 四足机器狗
- 无人机
- 无人船
- ...

* 机器人是人工智能技术应用能力的有效延展，而能自主移动的机器人更是极大地拓展了人工智能技术的应用范围。

SLAM的价值是什么？

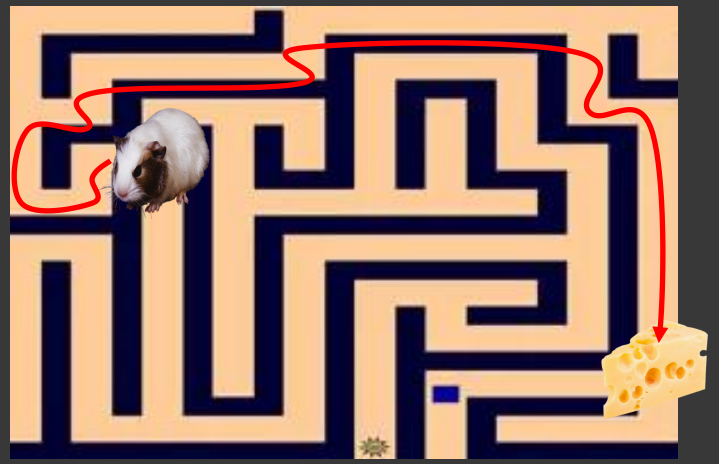
躯体使智能得以延展 → AI的最终归宿是机器人 → 机器人的完全自主化 → 自主移动技术 → SLAM

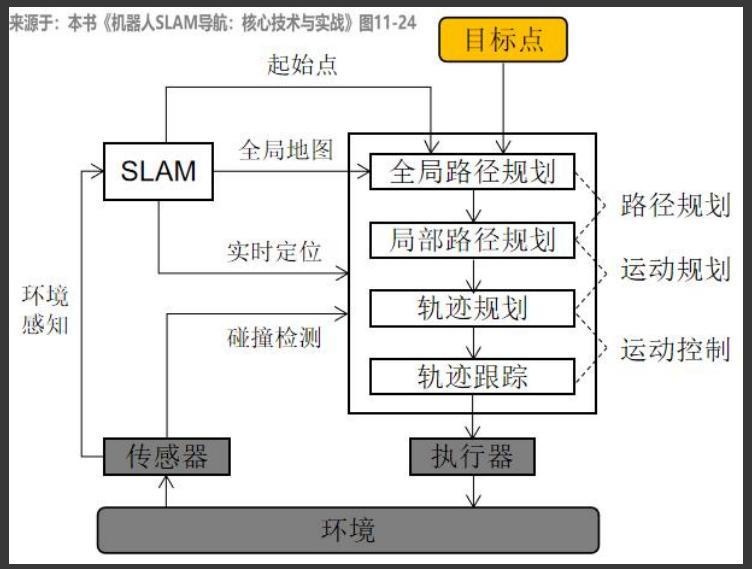
从工程上看，自主移动实质上就是解决从地点A到地点B的问题

① 我在哪?

② 我将到何处去?

③ 我该如何去





目前SLAM的概念已经泛化了，对定位、导航、自主移动等应用的笼统习惯指代。SLAM问题的求解方法也早已脱离滤波或优化等传统方法，现在的新求解方法与传统的SLAM理论框架可能有较大区别。

如果不用SLAM这个名词，你也可以用ABC、123之类的名词来指代。所以，大家不用再纠结于SLAM这个名词本身的定义，名词的定义往往只是约定俗成而已。

SLAM的应用领域？

- 机器人
- 自动驾驶
- 其他

自主移动是一种根本性需求


火星探测


农业采摘


服务机器人


深海勘探


电力巡检


扫地机器人


工厂物料运送AGV


送餐


岩洞勘测机器人


割草

SLAM的应用领域？

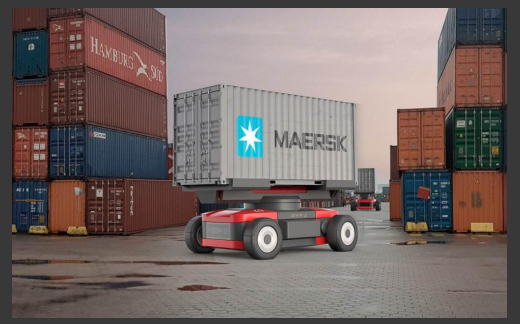
- 机器人
- 自动驾驶
- 其他

自主移动是一种根本性需求

无人驾驶

物流配送

无人清扫车



港口集装箱搬运

无人货运卡车

景区无人驾驶观光车

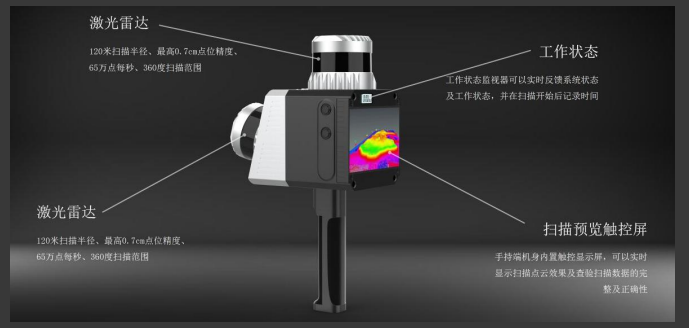
无人驾驶公交车

SLAM的应用领域？

- 机器人
- 自动驾驶
- 其他

自主移动是一种根本性需求

测绘无人机



建筑物手持SLAM建图仪

AR/VR中的应用

内容概要

7.1 SLAM发展简史

- SLAM的起源
- 数据关联、收敛和一致性
- SLAM的基本理论
- SLAM的学习方法

SLAM并不是某种具体的算法，而是一类问题的总称。

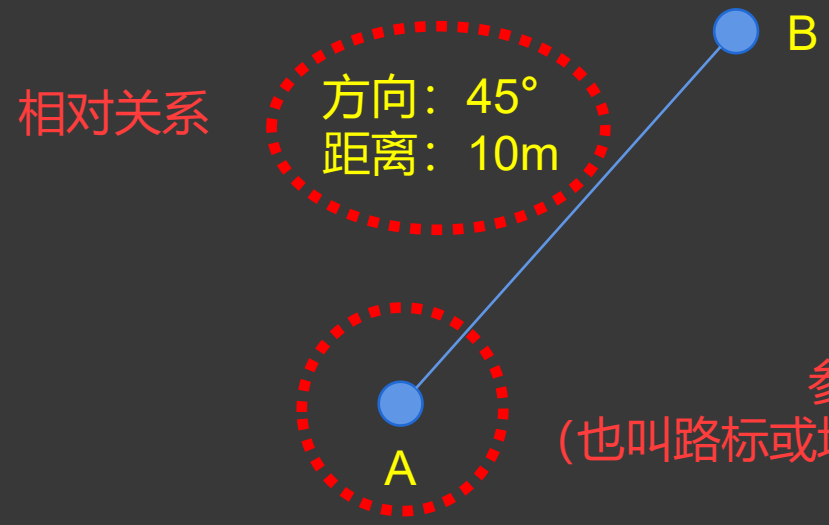
问题	求解方法（也叫算法）
1+1=?	• 计算器 • 算盘 • 笔算 • 心算 • 扳手指 • …
SLAM （同时定位与建图）	• Gmapping • Cartographer • LOAM • ORB-SLAM • Rtabmap • …

7.1 SLAM发展简史

- SLAM的起源
- 数据关联、收敛和一致性
- SLAM的基本理论
- SLAM的学习方法

SLAM的起源 →
① 定位问题 →
参考物
相对关系 →
直接 →
欧氏坐标
极坐标
球坐标
四叉树、八叉树
…
间接 →
拓扑描述
模糊描述
…

② 建图问题
③ SLAM问题
④ 导航问题
⑤ SLAM问题的泛化



相对关系
方向：45°
距离：10m

参考物
（也叫路标或地标，landmarks）
参考点、面、体
轮廓
特征
特殊标记
…

7.1 SLAM发展简史

- SLAM的起源
- 数据关联、收敛和一致性
- SLAM的基本理论
- SLAM的学习方法

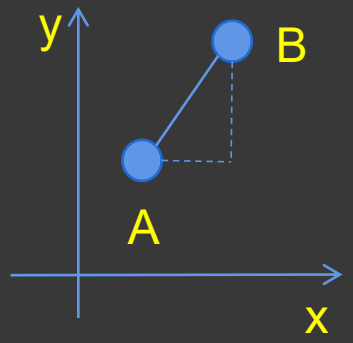
SLAM的起源 →
① 定位问题 →
② 建图问题
③ SLAM问题
④ 导航问题
⑤ SLAM问题的泛化

定位问题 →
参考物
相对关系 →
直接 →
间接 →

直接 →
欧氏坐标
极坐标
球坐标
四叉树、八叉树
…

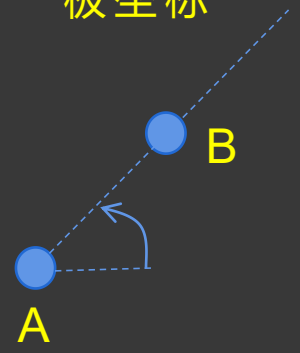
间接 →
拓扑描述
模糊描述
…

欧氏坐标



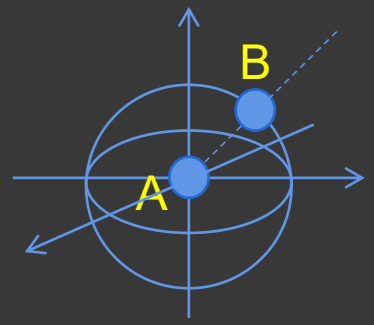
$$A=(1,2)$$
$$\overrightarrow{AB}=(2,3)$$
$$B=A+\overrightarrow{AB}=(3,5)$$

极坐标



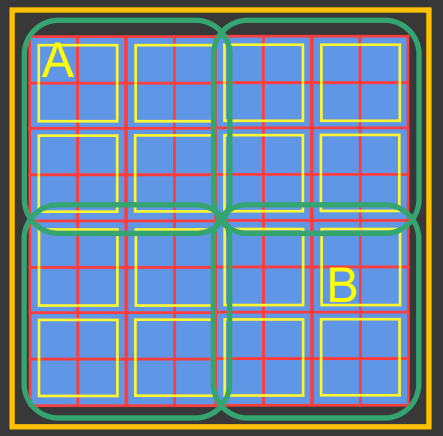
$$A=(0,0°)$$
$$B=(10,45°)$$

球坐标



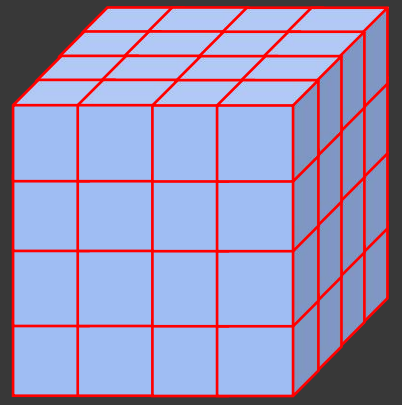
$$A=(0,0°,0°)$$
$$B=(10,30°,45°)$$

四叉树



$$A=(1,1,1,1,...)$$
$$B=(1,4,2,3,...)$$

八叉树



$$A=(1,1,1,1,...)$$
$$B=(1,8,5,7,...)$$

7.1 SLAM发展简史

- SLAM的起源
- 数据关联、收敛和一致性
- SLAM的基本理论
- SLAM的学习方法

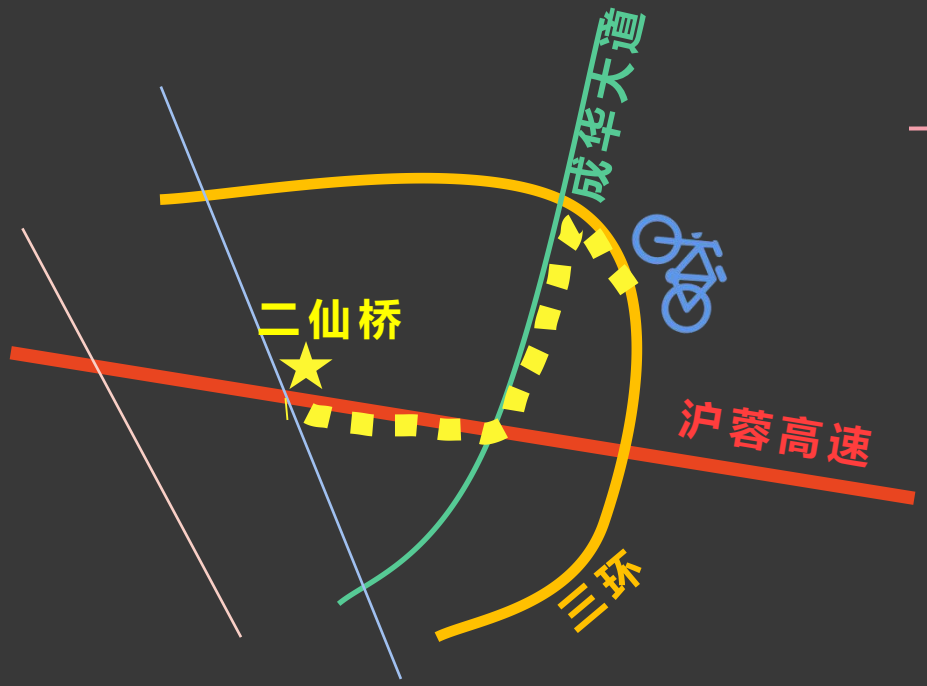
① 定位问题
② 建图问题
③ SLAM问题
④ 导航问题
⑤ SLAM问题的泛化

参考物
相对关系

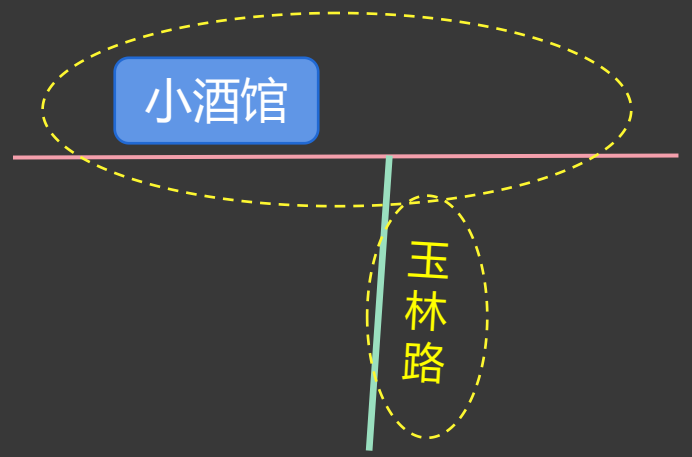
直接
- 欧氏坐标
- 极坐标
- 球坐标
- 四叉树、八叉树
- …

间接
- 拓扑描述
- 模糊描述
- …

拓扑描述



模糊描述



【成都旅游打卡】
走到玉林路的尽头
坐在小酒馆的门口

成都天府国际机场

7.1 SLAM发展简史

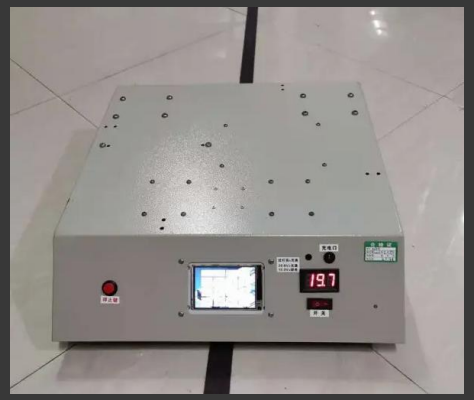
- SLAM的起源
- 数据关联、收敛和一致性
- SLAM的基本理论
- SLAM的学习方法

SLAM的起源 →
① 定位问题 →
② 建图问题
③ SLAM问题
④ 导航问题
⑤ SLAM问题的泛化

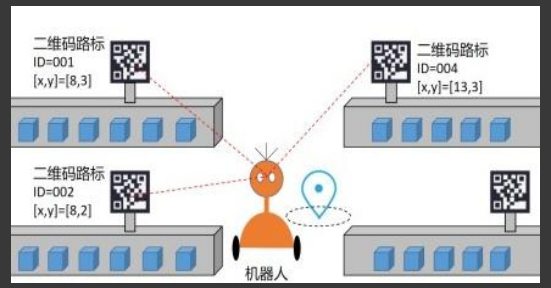
定位问题 →
参考物
相对关系 →

相对关系 →
直接 →
欧氏坐标
极坐标
球坐标
四叉树、八叉树
…

间接 →
拓扑描述
模糊描述
…

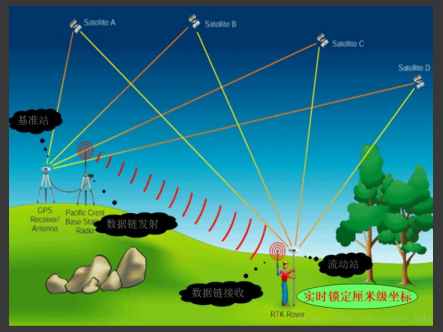


磁条参考物



二维码参考物

无线电参考物



卫星参考物

定位的本质都是提前部署参考物

7.1 SLAM发展简史

- SLAM的起源
  - ① 定位问题
  - ② 建图问题
  - ③ SLAM问题
  - ④ 导航问题
  - ⑤ SLAM问题的泛化
- 数据关联、收敛和一致性
- SLAM的基本理论
- SLAM的学习方法

人体三维结构=切片1+切片2+切片3+...

建图的本质都是基于已知位置切片的拼接

7.1 SLAM发展简史

- SLAM的起源
- 数据关联、收敛和一致性
- SLAM的基本理论
- SLAM的学习方法

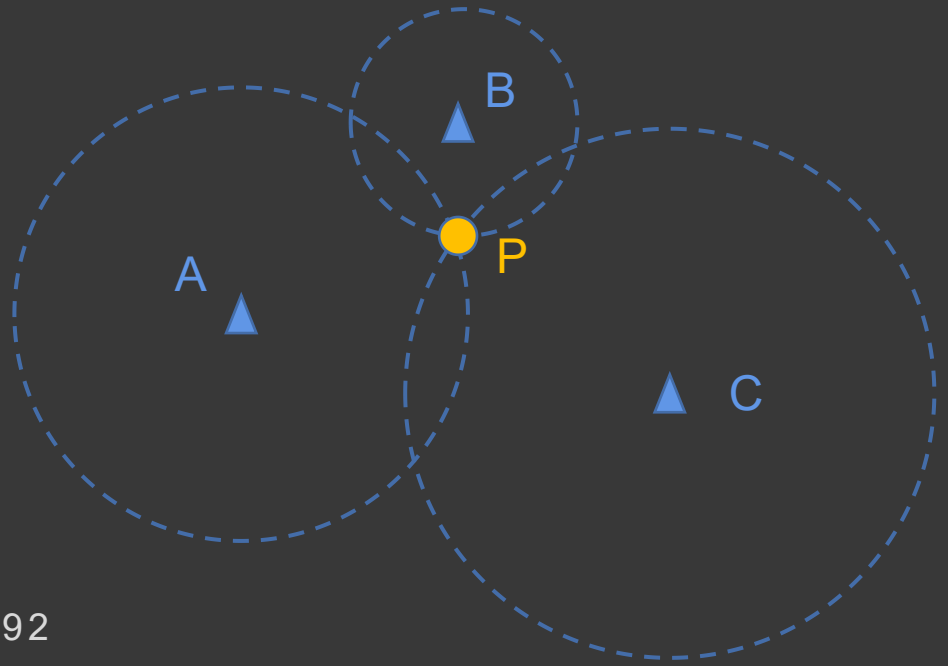
① 定位问题
② 建图问题
③ SLAM问题 → **同时解决定位和建图两个问题**
④ 导航问题
⑤ SLAM问题的泛化

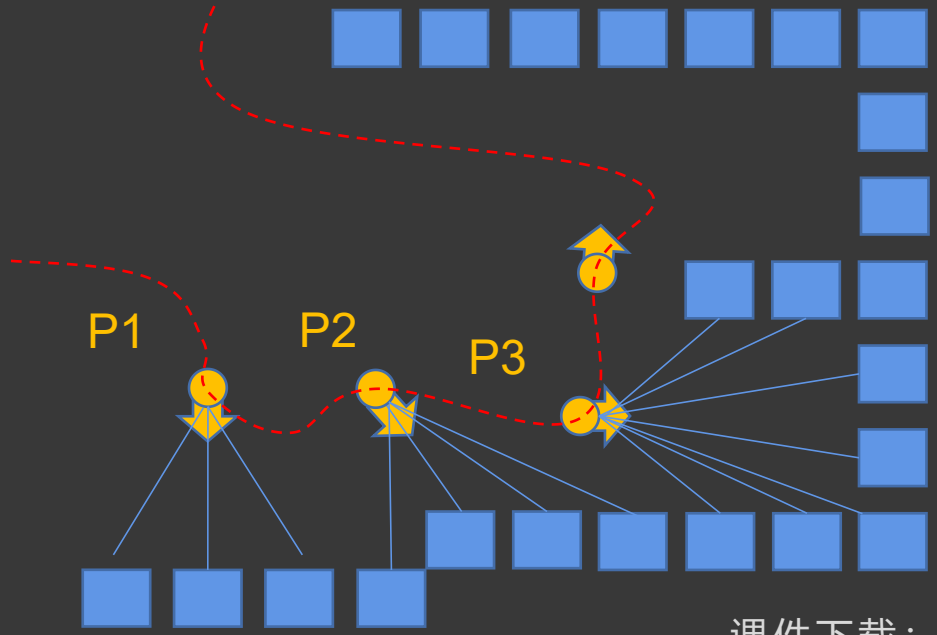
定位问题：

- 已知全局地图坐标信息
- 利用观测求解机器人在地图中的坐标

建图问题：

- 已知机器人的实时坐标
- 利用观测求解被观测物体的坐标





7.1 SLAM发展简史

- SLAM的起源
- 数据关联、收敛和一致性
- SLAM的基本理论
- SLAM的学习方法

① 定位问题
② 建图问题
③ SLAM问题 → **同时解决定位和建图两个问题**
④ 导航问题
⑤ SLAM问题的泛化

定位问题 VS 建图问题

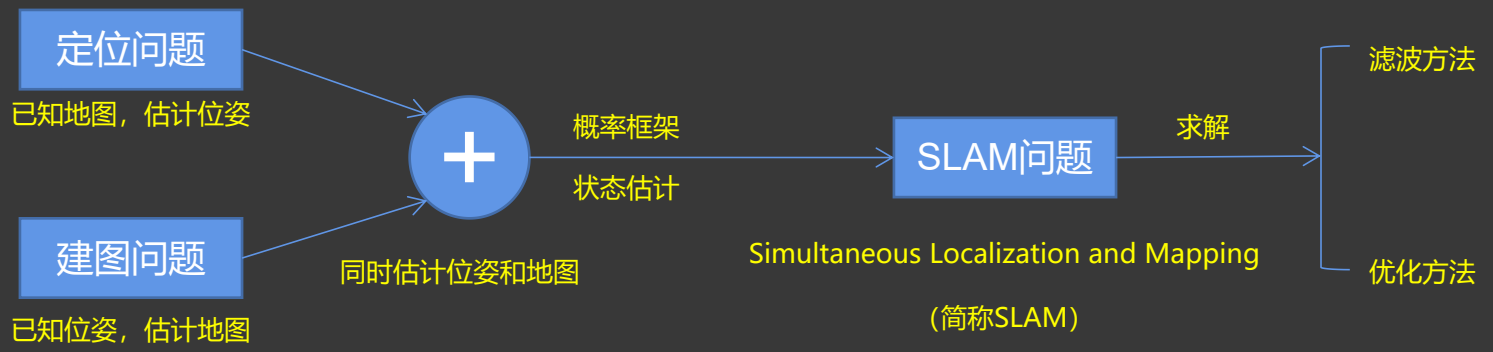
7.1 SLAM发展简史

- SLAM的起源
- 数据关联、收敛和一致性
- SLAM的基本理论
- SLAM的学习方法

① 定位问题
② 建图问题
③ SLAM问题 → **同时解决定位和建图两个问题**
④ 导航问题
⑤ SLAM问题的泛化

谁都会提出几个问题，关键是要能给出解决问题的思路才行

（PS:要么解决问题，要么解决提出问题的人）

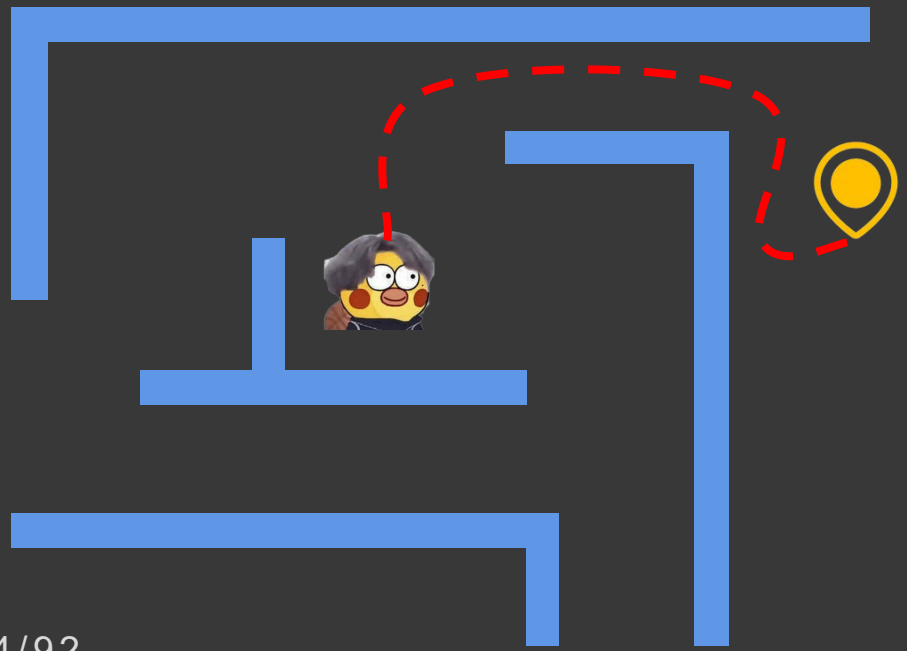


7.1 SLAM发展简史

- SLAM的起源
- 数据关联、收敛和一致性
- SLAM的基本理论
- SLAM的学习方法

① 定位问题
② 建图问题
③ SLAM问题
④ 导航问题
⑤ SLAM问题的泛化

从工程上看，自主移动实质上就是解决从地点A到地点B的问题
① 我在哪？
② 我将到何处去？
③ 我该如何去



我在哪？->取决于何种参考系

我将到何处去？->取决于目的地在参考系下采用的描述方式

我该如何去？->取决于路线和运动方式

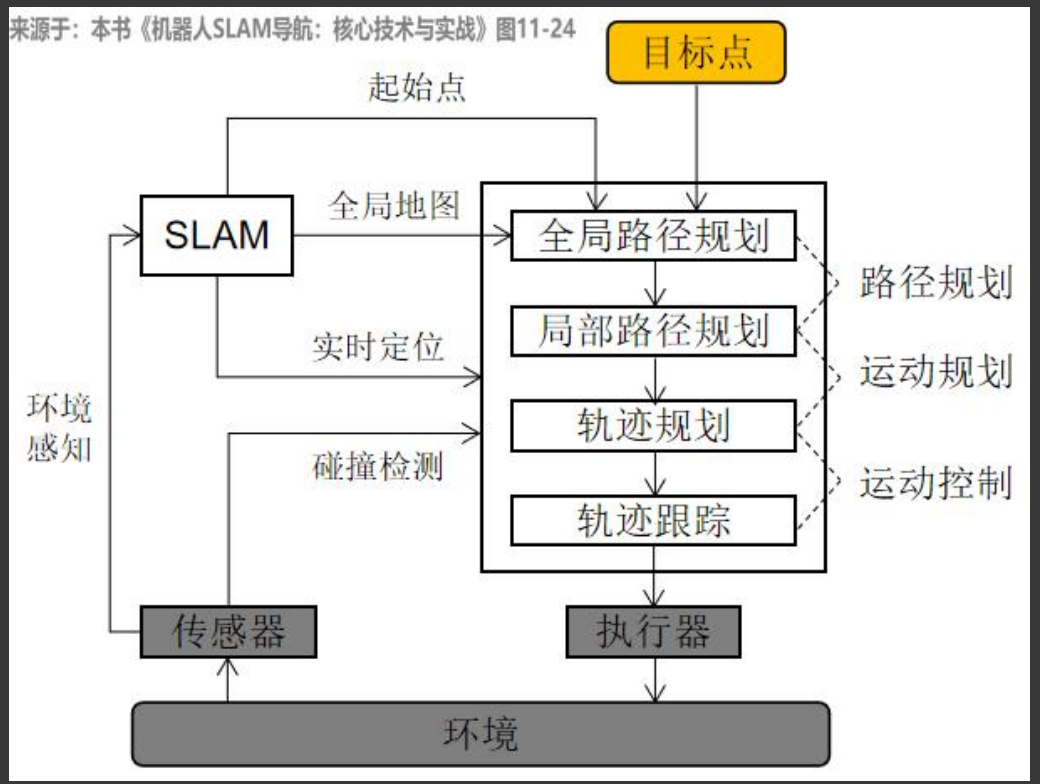
7.1 SLAM发展简史

- SLAM的起源
- 数据关联、收敛和一致性
- SLAM的基本理论
- SLAM的学习方法

① 定位问题
② 建图问题
③ SLAM问题
④ 导航问题
⑤ SLAM问题的泛化

从工程上看，自主移动实质上就是解决从地点A到地点B的问题
① 我在哪？
② 我将到何处去？
③ 我该如何去

虽然其它领域（比如VR/AR/测绘/三维建模）也讨论SLAM问题，但自动驾驶/机器人导航对SLAM问题更迫切



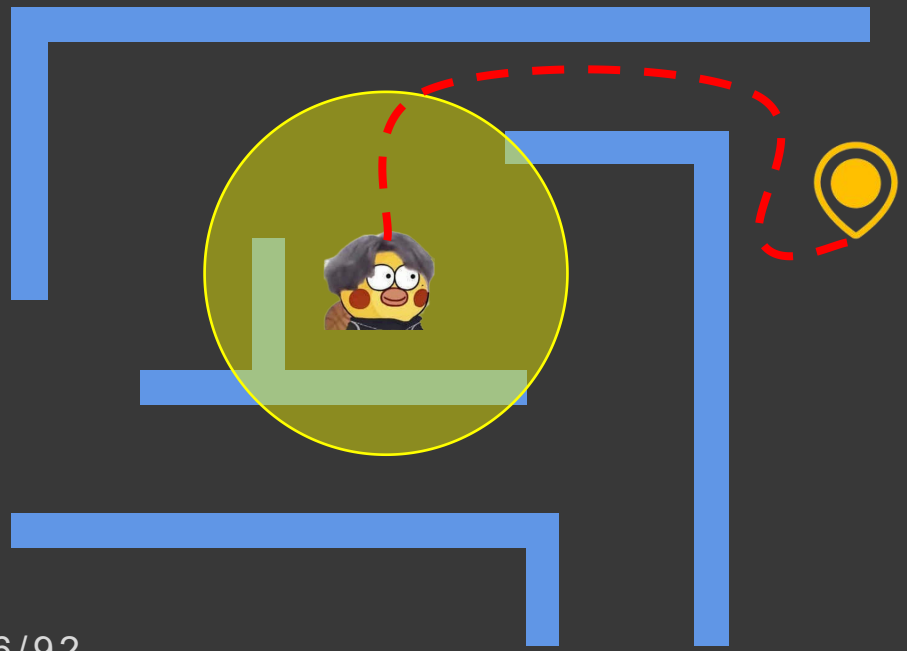
7.1 SLAM发展简史

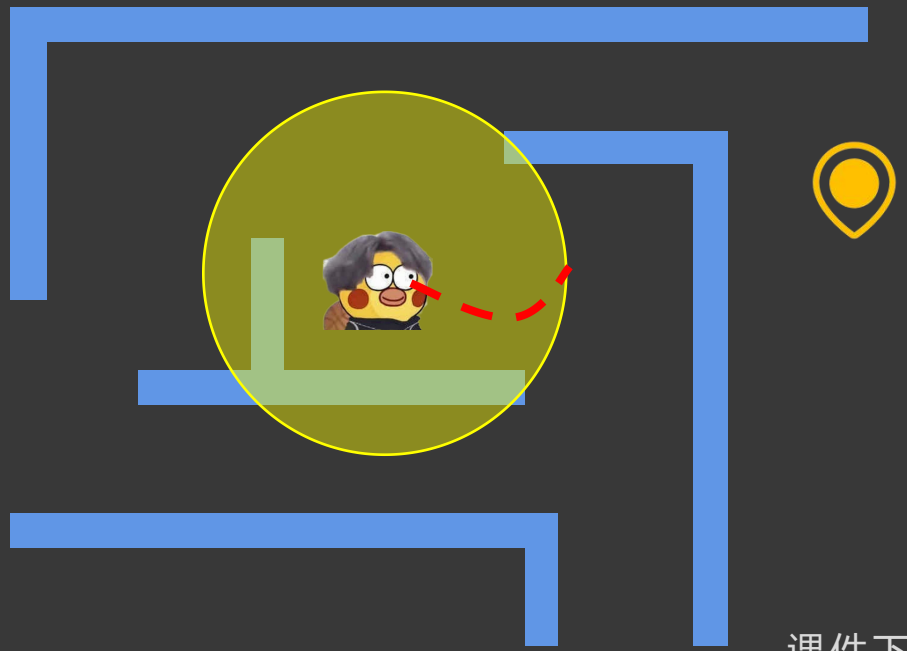
- SLAM的起源
- 数据关联、收敛和一致性
- SLAM的基本理论
- SLAM的学习方法

① 定位问题
② 建图问题
③ SLAM问题
④ 导航问题
⑤ SLAM问题的泛化

从工程上看，自主移动实质上就是解决从地点A到地点B的问题
① 我在哪？
② 我将到何处去？
③ 我该如何去

SLAM的作用：全局地图、全局定位
- 全局定位+全局避障+局部避障
- 全局定位+局部避障



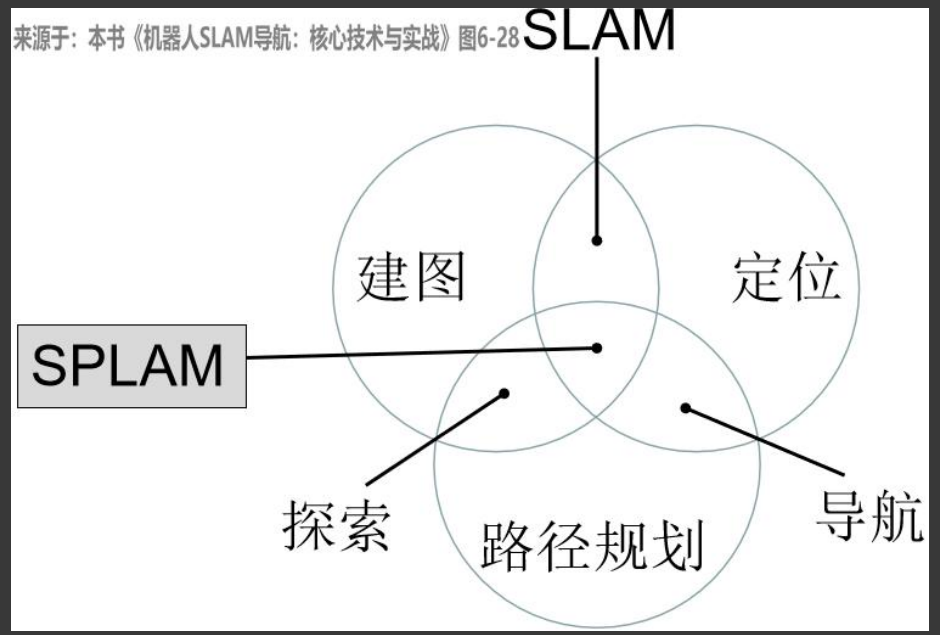


7.1 SLAM发展简史

- SLAM的起源
- 数据关联、收敛和一致性
- SLAM的基本理论
- SLAM的学习方法

① 定位问题
② 建图问题
③ SLAM问题
④ 导航问题
⑤ SLAM问题的泛化 → 没有SLAM难道就不能自动驾驶和导航吗？

- GPS信号覆盖盲区
- GPS定位延迟
- 高精度定位
- 全局决策
- 未知环境探索



目前SLAM的概念已经泛化了，对定位、导航、自主移动等应用的笼统习惯指代。SLAM问题的求解方法也早已脱离滤波或优化等传统方法，现在的新求解方法与传统的SLAM理论框架可能有较大区别。

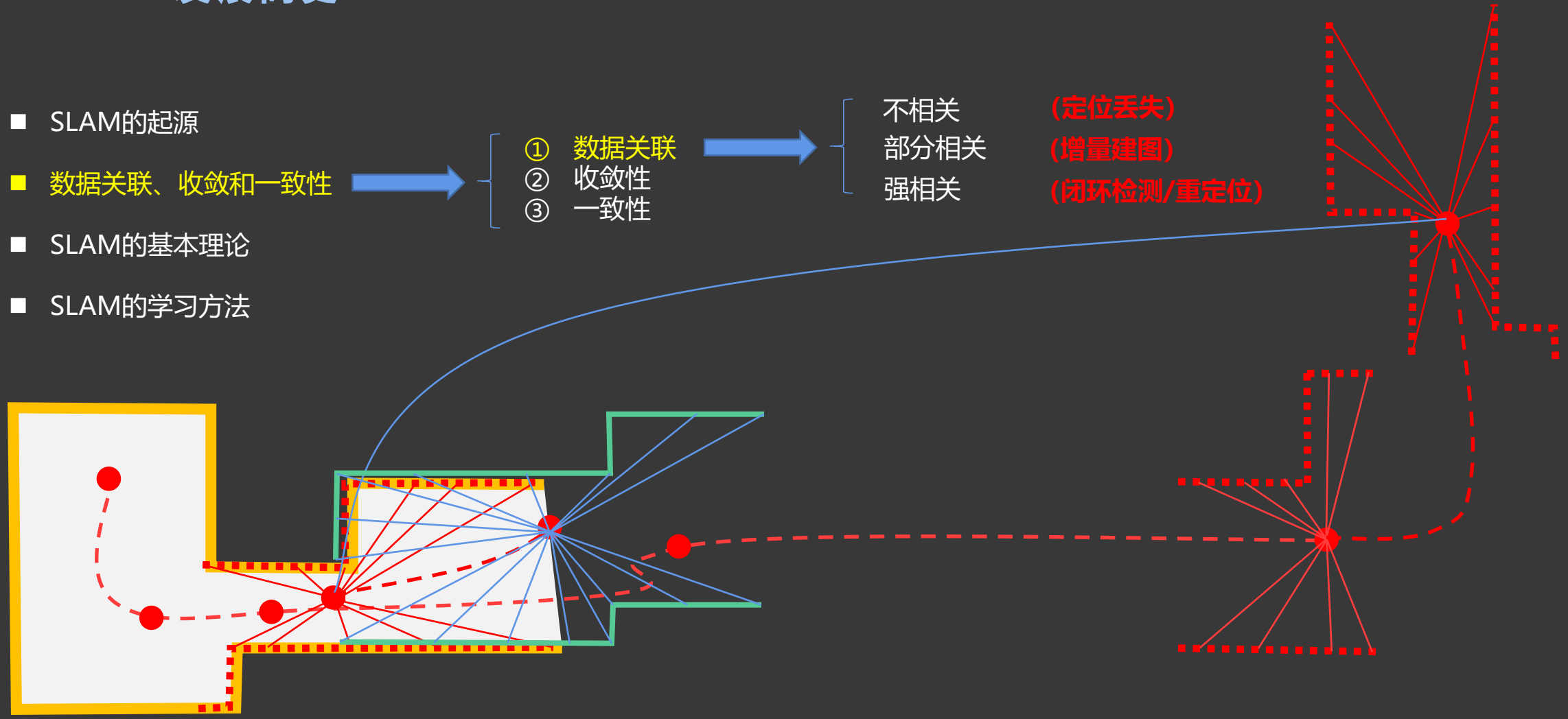
如果不用SLAM这个名词，你也可以用ABC、123之类的名词来指代。所以，大家不用再纠结于SLAM这个名词本身的定义，名词的定义往往只是约定俗成而已。

7.1 SLAM发展简史

- SLAM的起源
- 数据关联、收敛和一致性
- SLAM的基本理论
- SLAM的学习方法

① 数据关联
② 收敛性
③ 一致性

不相关 （定位丢失）
部分相关 （增量建图）
强相关 （闭环检测/重定位）



数据关联的数据形式是多样的，除了基于激光轮廓，还可以基于激光特征点、图像特征、机器学习特征等。

（其实目前的人工智能算法，也就相当于一个数据关联模型）

7.1 SLAM发展简史

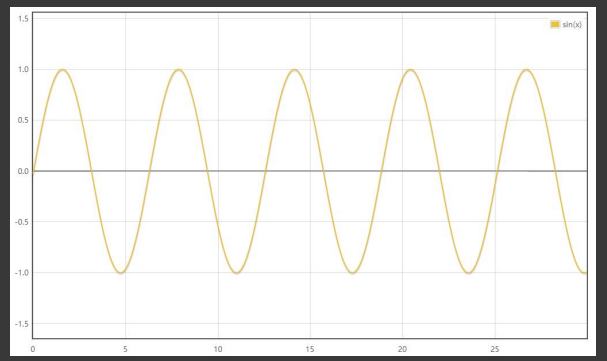
- SLAM的起源
- 数据关联、收敛和一致性
- SLAM的基本理论
- SLAM的学习方法

数据关联、收敛和一致性 →
① 数据关联
② 收敛性
③ 一致性

数学分析
基本概念之一 →
数列收敛
级数收敛（绝对收敛、条件收敛）
函数收敛（定点收敛、无穷收敛、路径收敛）
函数序列收敛（逐点收敛、一致收敛）

$$\{x_n\} = \{0.1 \quad 0.01 \quad 0.001 \quad 0.0001 \quad \ldots\}$$

$$\lim_{n\to\infty} x_n = 0$$



$$\lim_{x\to\infty} \sin(x) = ?$$

本章讨论的SLAM问题主要是基于概率框架建模，所以这里重点讨论随机变量的收敛性。

7.1 SLAM发展简史

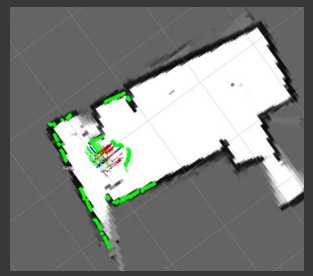
- SLAM的起源
- 数据关联、收敛和一致性
- SLAM的基本理论
- SLAM的学习方法

① 数据关联
② 收敛性
③ 一致性

数学分析
基本概念之一

数列收敛
级数收敛（绝对收敛、条件收敛）
函数收敛（定点收敛、无穷收敛、路径收敛）
函数序列收敛（逐点收敛、一致收敛）

不确定性是普遍存在的，而概率理论的目的是试图从不确定性事物中发现一些确定性规律



SLAM数学模型
（数据关联过程）

$$\lim_{n\to\infty}\left(\frac{T_1+T_2+\ldots+T_n}{n}\right)\to T \qquad \lim_{n\to\infty}\left(\frac{S_1+S_2+\ldots+S_n}{n}\right)\xrightarrow{?}S_{n+1}$$

$$\lim_{n\to\infty} f(z_1,z_2,z_3,\ldots,z_n)\xrightarrow{?}<pose,map>$$

切入问题的角度（数学建模方式）对能否发现问题中的有用规律至关重要，
收敛性一定程度可以作为评价指标，但也不绝对。

7.1 SLAM发展简史

- SLAM的起源
- 数据关联、收敛和一致性
- SLAM的基本理论
- SLAM的学习方法

① 数据关联
② 收敛性
③ 一致性

数学分析
基本概念之一

数列收敛
级数收敛（绝对收敛、条件收敛）
函数收敛（定点收敛、无穷收敛、路径收敛）
函数序列收敛（逐点收敛、一致收敛）

随机变量：X

随机变量序列：$\{X_n\}=\{X_1 \quad X_2 \quad X_3 \quad \dots \quad X_n\}$

强

依概率1收敛（几乎处处收敛）：$\forall \varepsilon>0, P\left(\lim_{n\to\infty}\left|X_n-X\right|<\varepsilon\right)=1$

依概率收敛（测度收敛）：$\forall \varepsilon>0, \lim_{n\to\infty}\left(P\left(\left|X_n-X\right|<\varepsilon\right)\right)\to 1$

依分布收敛：$\lim_{n\to\infty} F_{X_n}\to F_X$

依范数收敛：$q\in N^+, \lim_{n\to\infty}\left(E\left\|X_n-X\right|^q\right)\to 0$

弱

7.1 SLAM发展简史

- SLAM的起源
- 数据关联、收敛和一致性
- SLAM的基本理论
- SLAM的学习方法

① 数据关联
② 收敛性
③ 一致性

数学分析
基本概念之一

数列收敛
级数收敛（绝对收敛、条件收敛）
函数收敛（定点收敛、无穷收敛、路径收敛）
函数序列收敛（逐点收敛、一致收敛）

随机变量：X

随机变量序列：$\{X_n\}=\{X_1 \quad X_2 \quad X_3 \quad \dots \quad X_n\}$ （独立同分布）

弱大数定律：$E(X)\neq\infty,\forall\varepsilon>0,\lim_{n\to\infty}P\left(\left|\frac{X_1+X_2+\dots+X_n}{n}-E(X)\right|<\varepsilon\right)\to 1$ （依概率收敛）

强大数定律：$E(X)\neq\infty,D(X)\neq\infty,\forall\varepsilon>0,P\left(\lim_{n\to\infty}\left|\frac{X_1+X_2+\dots+X_n}{n}-E(X)\right|<\varepsilon\right)=1$ （依概率1收敛）

中心极限定理：$\lim_{n\to\infty}(X_1+X_2+\dots+X_n)\sim N(n\cdot E(X),n\cdot D(X))$

大数定律是将**统计平均**用来表示**数学期望**的依据

中心极限定理可以解释**正态分布**为何普遍存在

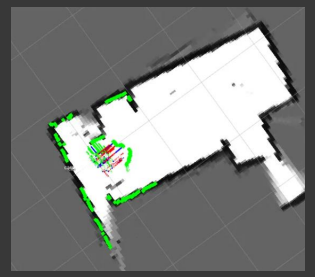
7.1 SLAM发展简史

- SLAM的起源
- 数据关联、收敛和一致性
- SLAM的基本理论
- SLAM的学习方法

数据关联、收敛和一致性：
① 数据关联
② 收敛性
③ 一致性

统计（真实的试验） —— 大数定律、中心极限定理 —— **概率（假想的理论）**

用大量观测数据
估计感兴趣的未知参数的取值



SLAM数学模型
（数据关联过程）

$$\lim_{n\to\infty} f(z_1, z_2, z_3, \ldots, z_n) \xrightarrow{?} < pose, map >$$

收敛性如何？
- 收敛 or 发散
- 一致性（依概率1收敛、依概率收敛、依分布收敛、依范数收敛）
- 收敛速度

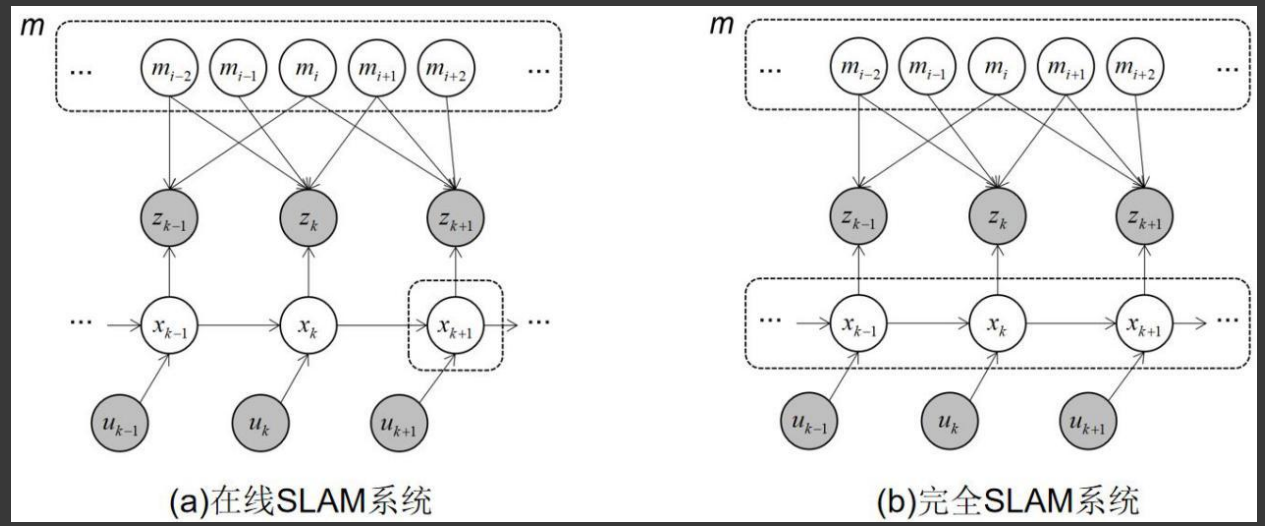
7.1 SLAM发展简史

- SLAM的起源
- 数据关联、收敛和一致性
- SLAM的基本理论
- SLAM的学习方法

求解SLAM问题的方法，称为SLAM算法

① SLAM求解方法（算法）
② SLAM求解方法（算法）的分类
③ SLAM学习路线

		在线SLAM系统	完全SLAM系统
贝叶斯网络表示	EKF滤波	EKF-SLAM	
	粒子滤波		Fast-SLAM Gmapping
最小二乘表示	直接法		Graph-SLAM
	优化法		Cartographer LOAM ORB-SLAM
机器学习表示	端到端法	DeepVO	
	改进模块法	NeRF-SLAM NICE-SLAM	CNN-SLAM
	生物启发法	RatSLAM NeuroSLAM	



(a)在线SLAM系统　　(b)完全SLAM系统

7.1 SLAM发展简史

- SLAM的起源
- 数据关联、收敛和一致性
- SLAM的基本理论
- SLAM的学习方法

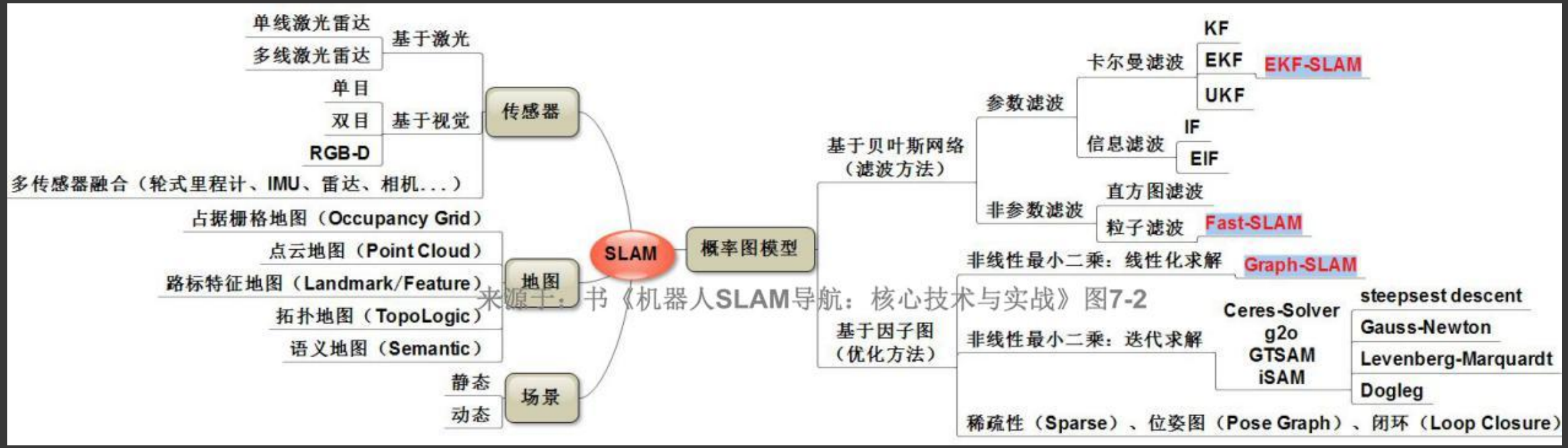
SLAM的基本理论 →
① SLAM求解方法（算法）
② SLAM求解方法（算法）的分类
③ SLAM学习路线

SLAM求解方法（算法）的分类 →
- 按求解结果分类
  - 在线SLAM
  - 完全SLAM
- 按问题表示分类
  - 基于贝叶斯的SLAM
  - 基于最小二乘的SLAM
  - 基于机器学习的SLAM
- 按传感器分类
  - 激光SLAM
  - 视觉SLAM
  - 多传感器融合SLAM
- 按地图分类
  - 二维SLAM
  - 三维SLAM
  - 语义SLAM
- 按场景分类
  - 静态SLAM
  - 动态SLAM

7.1 SLAM发展简史

- SLAM的起源
- 数据关联、收敛和一致性
- SLAM的基本理论
- SLAM的学习方法

① SLAM求解方法（算法）
② SLAM求解方法（算法）的分类
③ SLAM学习路线



来源于：书《机器人SLAM导航：核心技术与实战》图7-2

7.1 SLAM发展简史

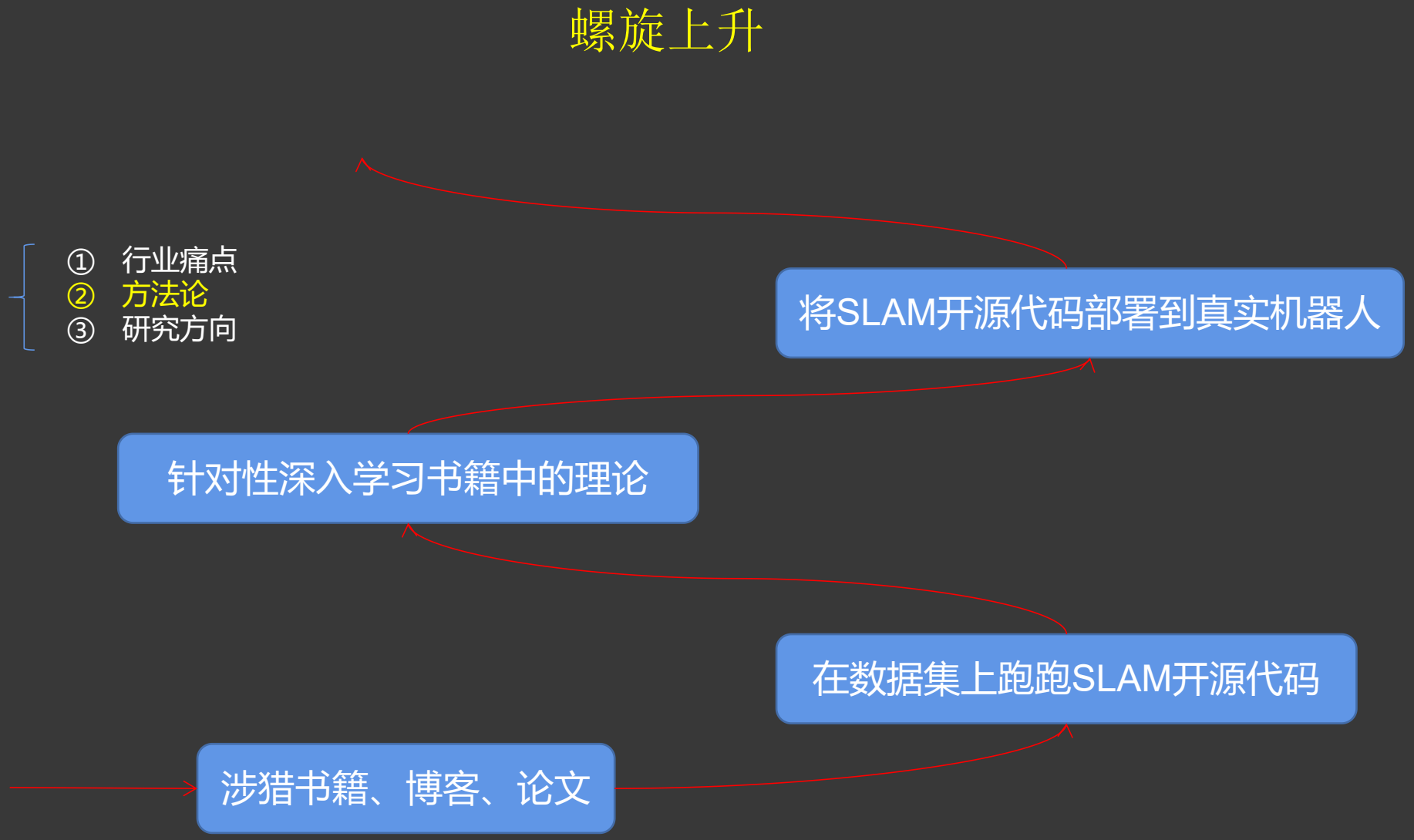
- SLAM的起源
- 数据关联、收敛和一致性
- SLAM的基本理论
- SLAM的学习方法 → ① 行业痛点 ② 方法论 ③ 研究方向

SLAM是一个软硬件相结合，理论加实战的浩大工程性问题

企业的目的是要将SLAM技术真正落地到产品，
而不是等你慢慢研究数学理论或简单调几个参数，
每个从业者都应该建立起大局观以避免重复造轮子，
换句话说我们缺乏SLAM全栈人才，这也正是我写这本书的初衷。

7.1 SLAM发展简史

- SLAM的起源
- 数据关联、收敛和一致性
- SLAM的基本理论
- SLAM的学习方法 → ① 行业痛点 ② 方法论 ③ 研究方向



7.1 SLAM发展简史

- SLAM的起源
- 数据关联、收敛和一致性
- SLAM的基本理论
- SLAM的学习方法 → ① 行业痛点 ② 方法论 ③ 研究方向

巨大的信息量是获取研究思路的前提

① 发现新应用领域：水下SLAM应用、手动编辑SLAM地图、SLAM云计算化

② 软件工程的优化：去ROS化、裸机级SLAM、跨平台兼容

③ 改进SLAM某些功能模块：特征提取新方法、闭环检测可靠性、持久化建图机制

④ 多传感器融合：传感器标定、时间戳同步、去干扰数据、数据关联

⑤ AI+SLAM：端到端SLAM、语义SLAM、特征工程

⑥ 克服异常场景：玻璃障碍物、传感器盲区、光照变化、上下坡问题

内容概要

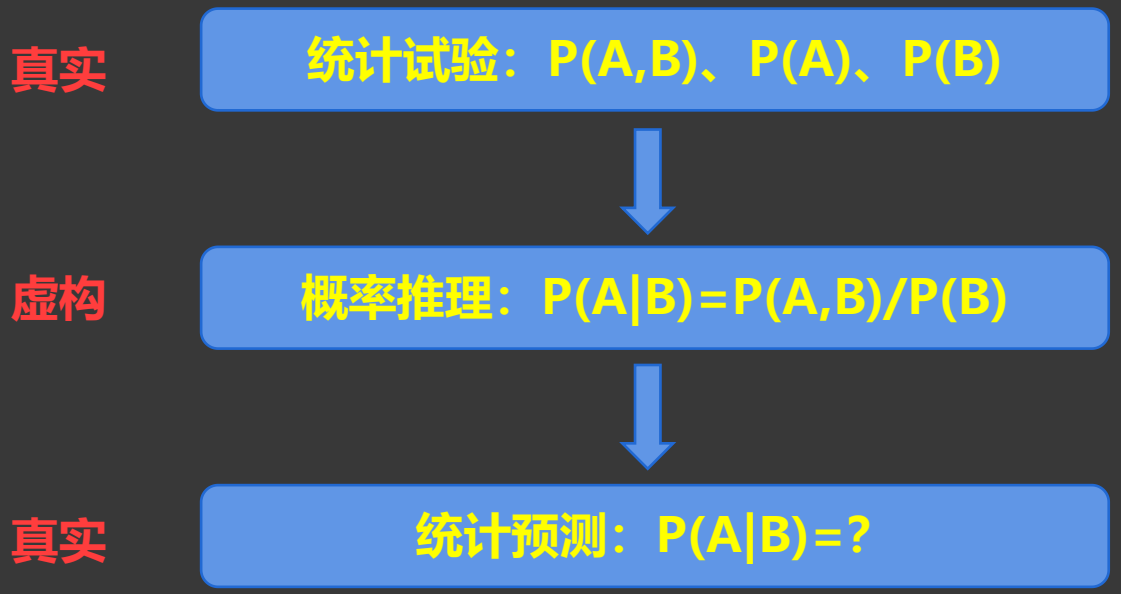
7.2 SLAM中的概率理论

概率理论，是什么？

不确定性是普遍存在的，而**概率理论**的目的是试图从**不确定性事物**中发现一些**确定性规律**

概率理论**不一定具有明确物理意义**，很多时候只是借用这套理论体系以便于问题的研究

概率理论是归纳演绎的方法，第1步：**统计试验**，第2步：**概率推理**，第3步：**统计预测**



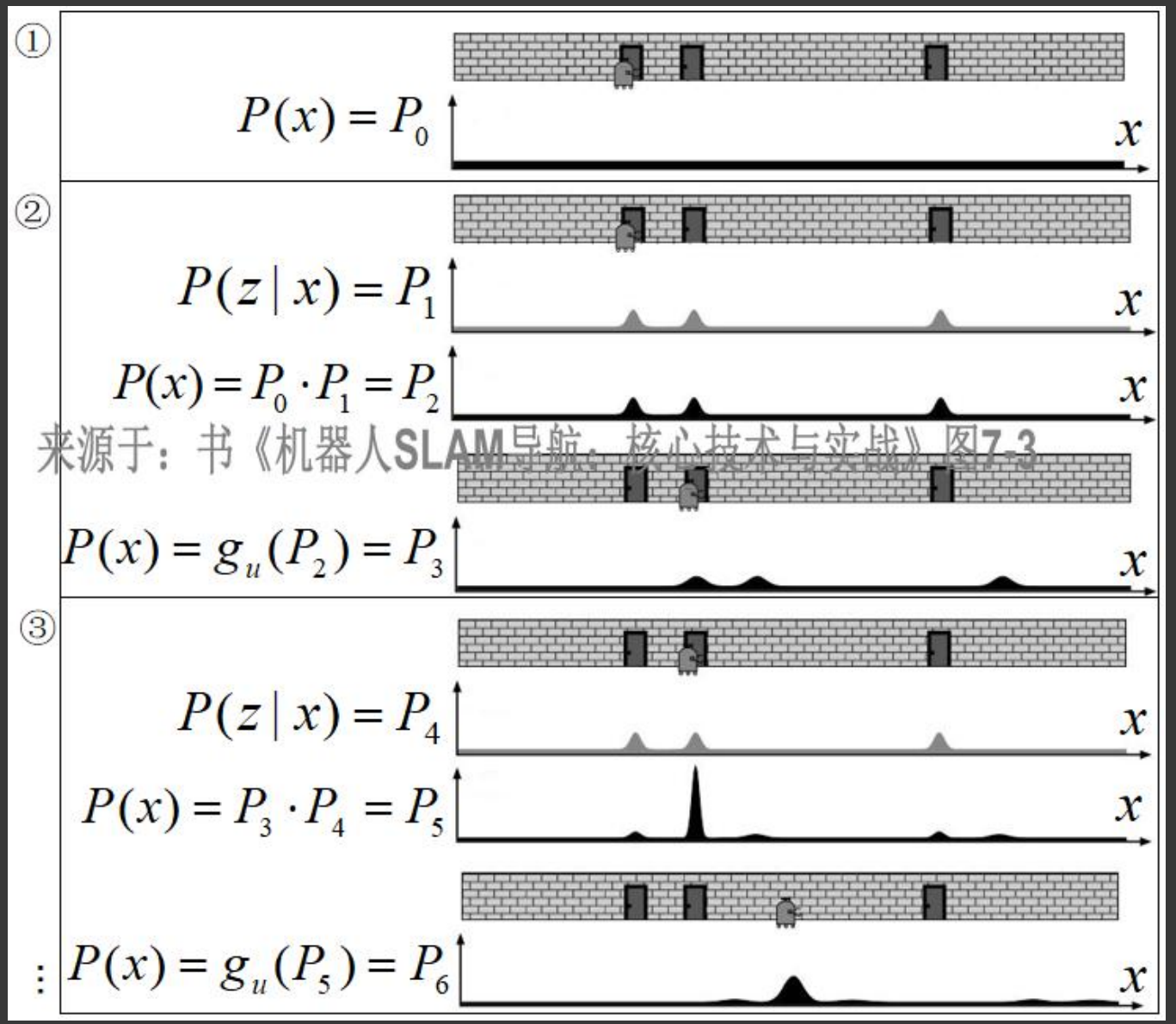
研究SLAM问题必须要用概率理论吗？

- 不一定，只是目前来看概率理论对解决SLAM问题有显著效果。
- 现在用神经网络、机器学习大模型、仿生等理论也取得了很多效果。

7.2 SLAM中的概率理论

机器人问题中为什么存在不确定性，原因包括传感器测量噪声、电机控制偏差、计算机软件计算精度近似等

利用概率描述机器人中的不确定性，这样机器人中的不确定性就可以在概率理论框架下被计算和推演，即所谓的概率机器人学



来源于：书《机器人SLAM导航：核心技术与实战》图7-3

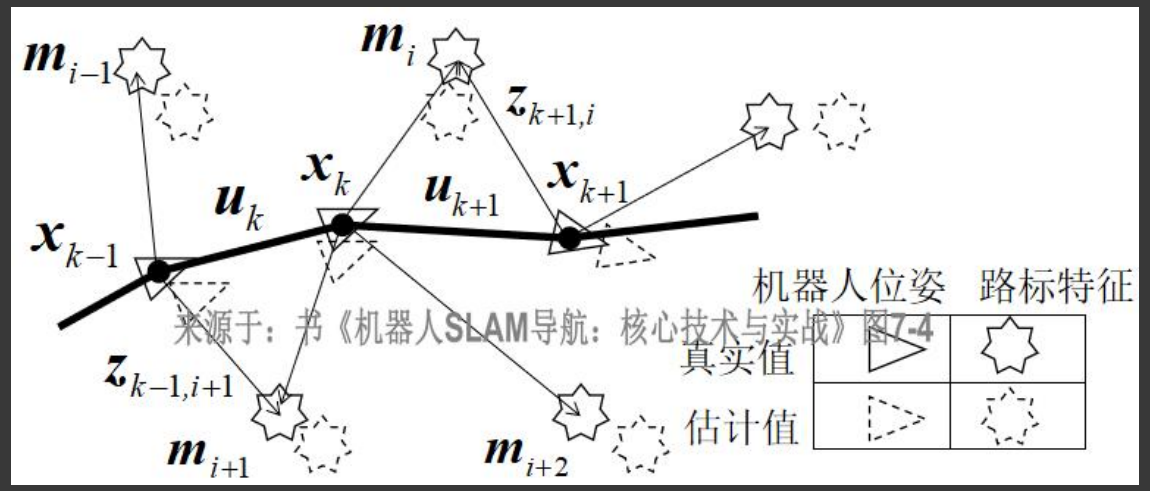
7.2 SLAM中的概率理论

- 状态估计问题
- 概率运动模型
- 概率观测模型
- 概率图模型

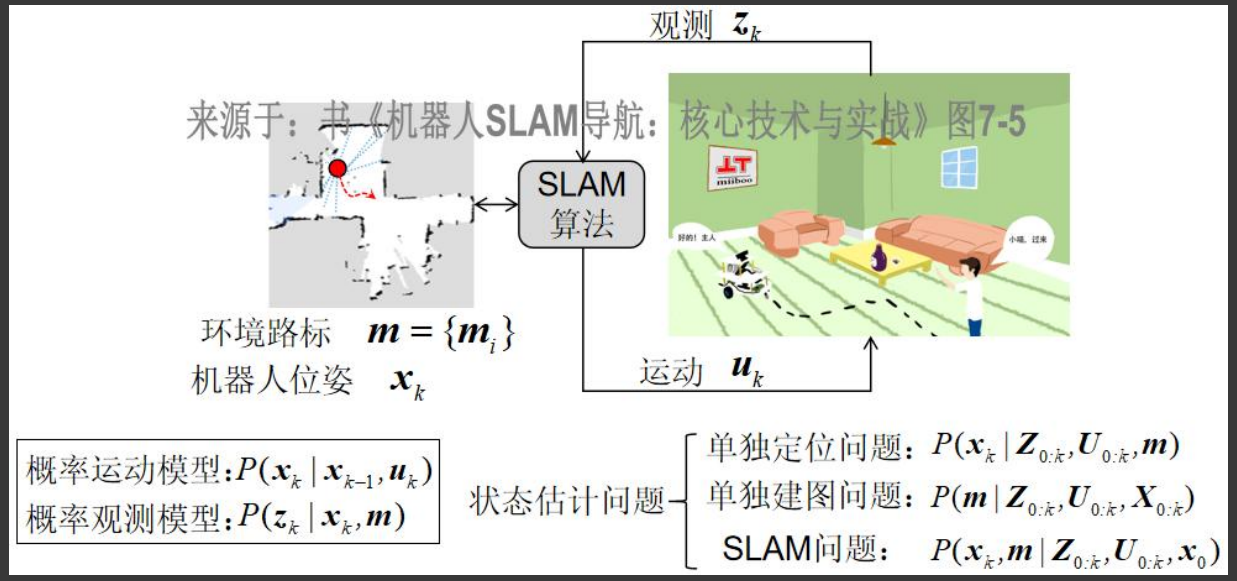
状态（感兴趣的物理量）
- 直接状态（比如尺子测量的身高）
- 间接状态（比如曹冲称象的重量）

估计（猜测）
- 猜测方法（概率理论）
- 猜测依据（观测数据）

不能直接准确测量的物理量
就需要进行状态估计



机器人SLAM中的状态估计问题

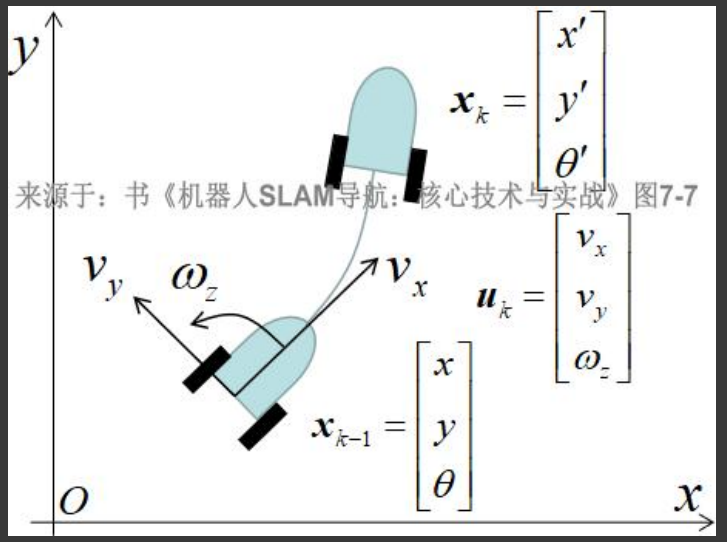


机器人SLAM中的状态估计问题的概率描述

7.2 SLAM中的概率理论

- 状态估计问题
- 概率运动模型
- 概率观测模型
- 概率图模型

概率运动模型 → 速度运动模型 → 里程计运动模型



来源于：书《机器人SLAM导航：核心技术与实战》图7-7

状态转移方程：

$$x_k = \begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = g(x_{k-1}, u_k) = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega_z \end{bmatrix} \Delta t$$

误差描述：

$$u_k \sim N(\bar{u}_k, \Sigma_{u_k})$$
$$x_{k-1} \sim N(\bar{x}_{k-1}, \Sigma_{x_{k-1}})$$
$$x_k \sim N(\bar{x}_k, \Sigma_{x_k})$$

在没有先验信息的情况，通常将误差假设为高斯分布

$$\Sigma_{u_k} = \begin{bmatrix} \sigma_{v_x}^2 & 0 & 0 \\ 0 & \sigma_{v_y}^2 & 0 \\ 0 & 0 & \sigma_{\omega_z}^2 \end{bmatrix}$$

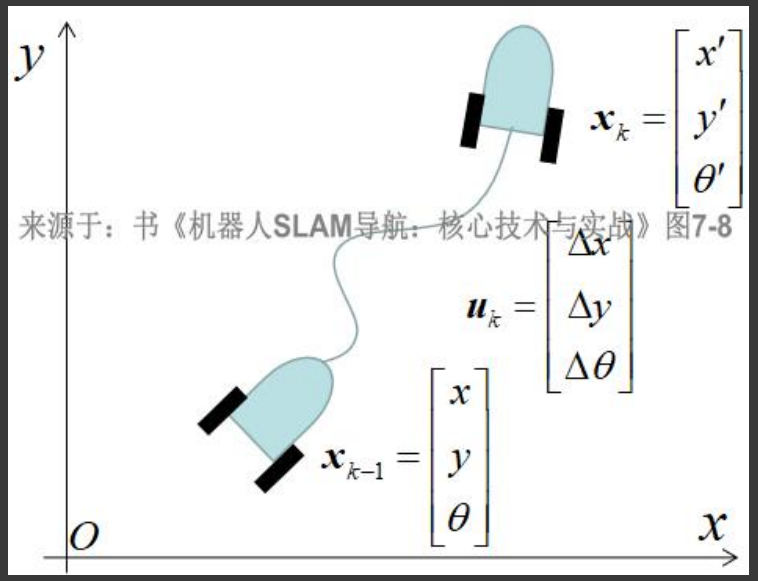
$$\Sigma_{x_k} = \begin{bmatrix} \frac{\partial g}{\partial x_{k-1}} & \frac{\partial g}{\partial u_k} \end{bmatrix} \begin{bmatrix} \Sigma_{x_{k-1}} & 0_{3\times3} \\ 0_{3\times3} & \Sigma_{u_k} \end{bmatrix} \begin{bmatrix} \frac{\partial g}{\partial x_{k-1}} & \frac{\partial g}{\partial u_k} \end{bmatrix}^{\mathrm{T}}$$

（其中，$\begin{bmatrix} \frac{\partial g}{\partial x_{k-1}} & \frac{\partial g}{\partial u_k} \end{bmatrix}$ 为函数g的雅克比矩阵）

7.2 SLAM中的概率理论

- 状态估计问题
- 概率运动模型
- 概率观测模型
- 概率图模型

概率运动模型 → 速度运动模型 / 里程计运动模型



来源于：书《机器人SLAM导航：核心技术与实战》图7-8

状态转移方程：

$$\boldsymbol{x}_k = \begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = g(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k) = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta \theta \end{bmatrix}$$

误差描述：

$$\boldsymbol{u}_k \sim N(\bar{\boldsymbol{u}}_k, \Sigma_{\boldsymbol{u}_k})$$
$$\boldsymbol{x}_{k-1} \sim N(\bar{\boldsymbol{x}}_{k-1}, \Sigma_{\boldsymbol{x}_{k-1}})$$
$$\boldsymbol{x}_k \sim N(\bar{\boldsymbol{x}}_k, \Sigma_{\boldsymbol{x}_k})$$

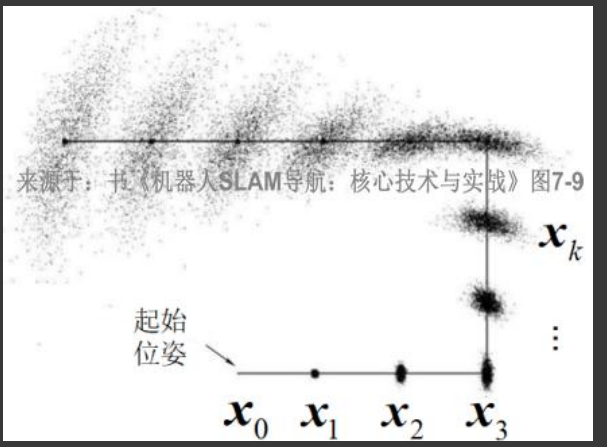
在没有先验信息的情况，通常将误差假设为高斯分布

$$\boldsymbol{\Sigma}_{\boldsymbol{u}_k} = \begin{bmatrix} \sigma_{\Delta x}^2 & 0 & 0 \\ 0 & \sigma_{\Delta y}^2 & 0 \\ 0 & 0 & \sigma_{\Delta\theta}^2 \end{bmatrix}$$

$$\sigma_{\Delta x}^2 = \xi_{\Delta xy} + \alpha_1\sqrt{(\Delta x)^2 + (\Delta y)^2} + \alpha_2|\Delta\theta|$$
$$\sigma_{\Delta y}^2 = \xi_{\Delta xy} + \alpha_1\sqrt{(\Delta x)^2 + (\Delta y)^2} + \alpha_2|\Delta\theta|$$
$$\sigma_{\Delta\theta}^2 = \xi_{\Delta\theta} + \alpha_3\sqrt{(\Delta x)^2 + (\Delta y)^2} + \alpha_4|\Delta\theta|$$

$$\boldsymbol{\Sigma}_{\boldsymbol{x}_k} = \begin{bmatrix} \frac{\partial g}{\partial \boldsymbol{x}_{k-1}} & \frac{\partial g}{\partial \boldsymbol{u}_k} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{x}_{k-1}} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{\Sigma}_{\boldsymbol{u}_k} \end{bmatrix} \begin{bmatrix} \frac{\partial g}{\partial \boldsymbol{x}_{k-1}} & \frac{\partial g}{\partial \boldsymbol{u}_k} \end{bmatrix}^{\mathrm{T}}$$

（其中，$\begin{bmatrix} \frac{\partial g}{\partial \boldsymbol{x}_{k-1}} & \frac{\partial g}{\partial \boldsymbol{u}_k} \end{bmatrix}$ 为函数g的雅克比矩阵）



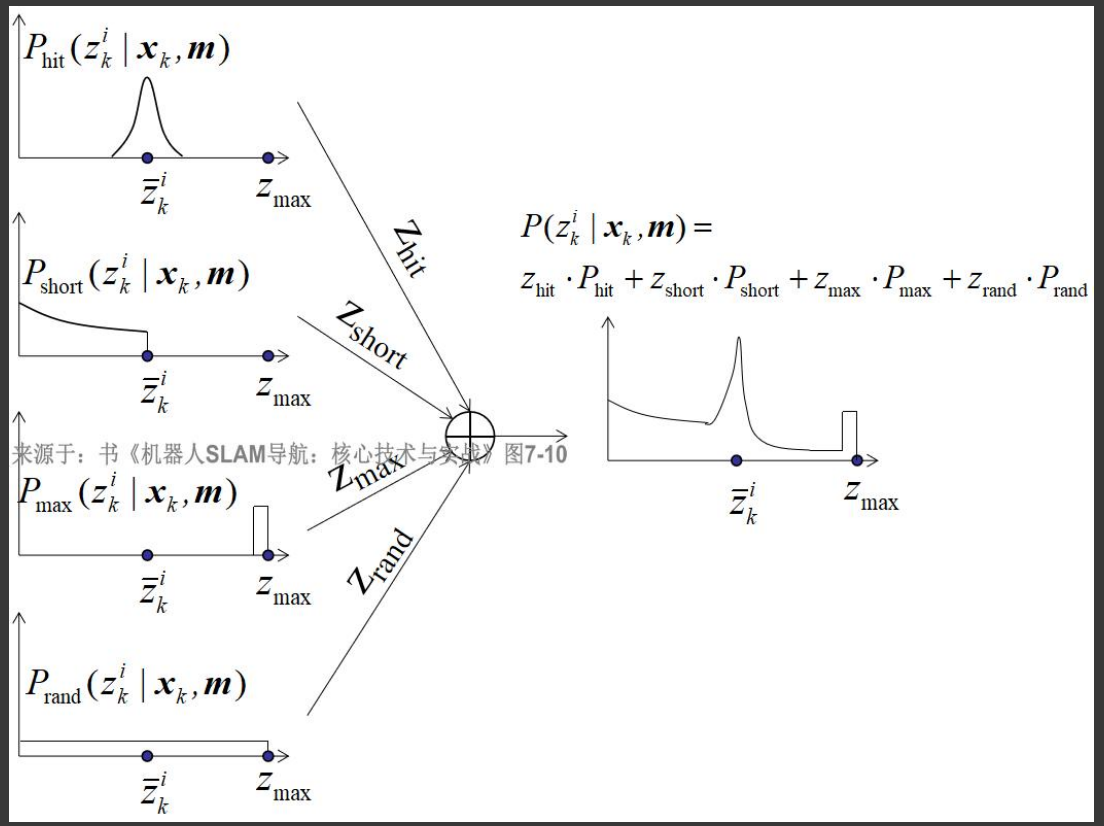
来源于：书《机器人SLAM导航：核心技术与实战》图7-9

机器人位姿不确定性随运动的关系

7.2 SLAM中的概率理论

- 状态估计问题
- 概率运动模型
- 概率观测模型
  - 激光
    - 波束模型
    - 似然域模型
  - 视觉（在第9章中讨论）
- 概率图模型

$$P(z_k^i \mid \boldsymbol{x}_k, \boldsymbol{m}) = z_{\text{hit}} \cdot P_{\text{hit}} + z_{\text{short}} \cdot P_{\text{short}} + z_{\max} \cdot P_{\max} + z_{\text{rand}} \cdot P_{\text{rand}}$$



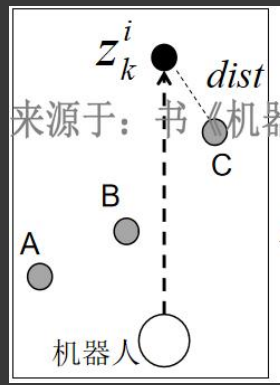
数学模型的构建，是基于大量实践数据归纳总结出来的经验

7.2 SLAM中的概率理论

- 状态估计问题
- 概率运动模型
- 概率观测模型
  - 激光
    - 波束模型
    - 似然域模型

$$P(z_k^i \mid \boldsymbol{x}_k, \boldsymbol{m}) = z_{\text{hit}} \cdot P_{\text{hit}} + z_{\text{max}} \cdot P_{\text{max}} + z_{\text{rand}} \cdot P_{\text{rand}}$$

  - 视觉（在第9章中讨论）
- 概率图模型



来源于：书《机器人SLAM导航：核心技术与实战》图7-11

$$P_{\text{hit}}(z_k^i \mid \boldsymbol{x}_k, \boldsymbol{m}) = \frac{1}{\sigma_{\text{hit}}\sqrt{2\pi}} \mathrm{e}^{-\frac{dist^2}{2\sigma_{\text{hit}}^2}}$$

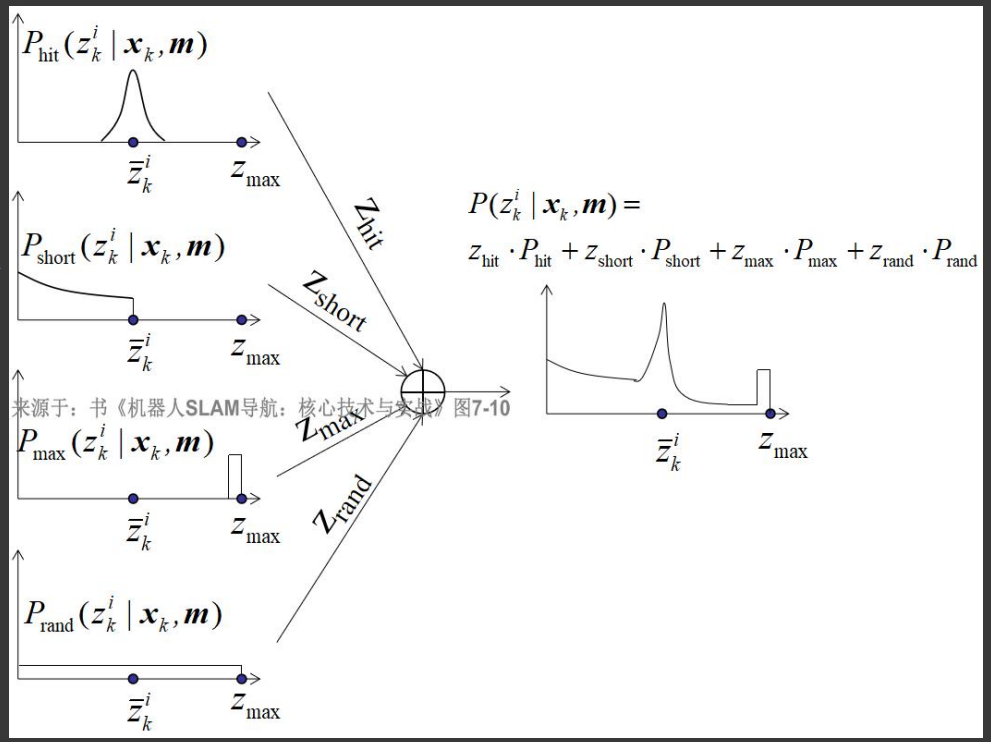
数学模型的构建，是基于大量实践数据归纳总结出来的经验

7.2 SLAM中的概率理论

- 状态估计问题
- 概率运动模型
- 概率观测模型
- 概率图模型

概率观测模型 → 激光、视觉（在第9章中讨论）

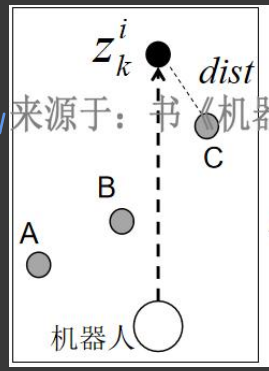
激光 → 波束模型、似然域模型



来源于：书《机器人SLAM导航：核心技术与实战》图7-10

波束模型缺乏光滑性，导致构建出来的地图具有很多毛刺。

似然域模型较为平滑，构建出来的地图不易出现毛刺。

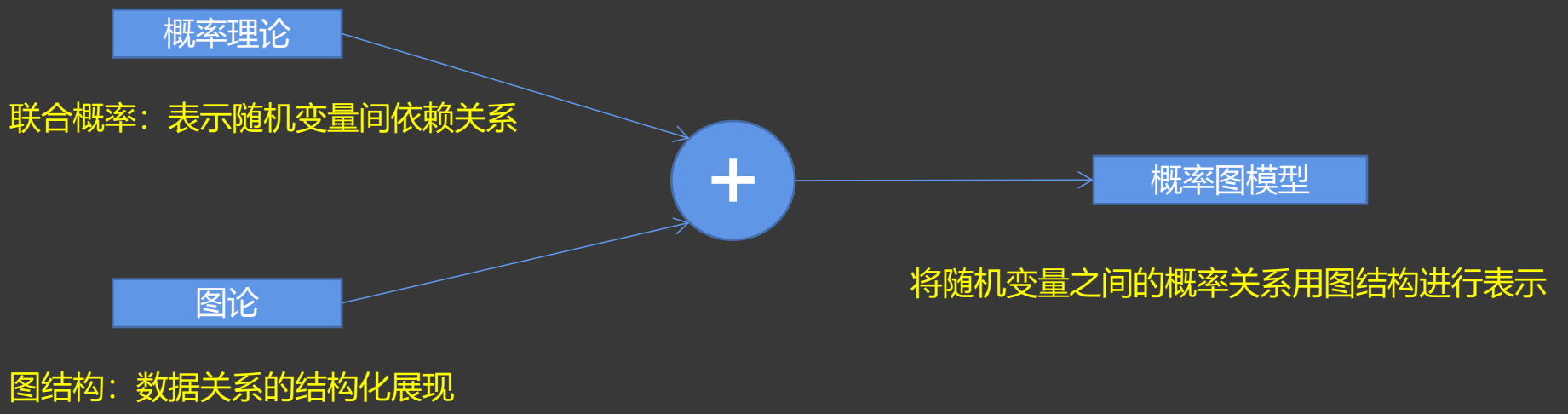


来源于：书《机器人SLAM导航：核心技术与实战》图7-11

$$P_{hit}(z_k^i \mid x_k, m) = \frac{1}{\sigma_{hit}\sqrt{2\pi}} e^{-\frac{dist^2}{2\sigma_{hit}^2}}$$

7.2 SLAM中的概率理论

- 状态估计问题
- 概率运动模型
- 概率观测模型
- 概率图模型 → 当有了概率运动模型和概率观测模型后，怎么描述**运动模型**与**观测模型**之间的概率关系呢？



7.2 SLAM中的概率理论

- 状态估计问题
- 概率运动模型
- 概率观测模型
- 概率图模型

连续随机变量 VS 离散随机变量

（比如观测温度） （比如掷骰子）

随机变量集合：$X=\{A,B,C,D\}$

边缘分布：（概率加法）

$$P(A)=\sum_{D}\sum_{C}\sum_{B}P(A,B,C,D)$$

联合分布：（概率乘法）

$$\begin{aligned}P(A,B,C,D)&=P(A,B,C)P(D\mid A,B,C)\\&=P(A,B)P(C\mid A,B)P(D\mid A,B,C)\\&=P(A)P(B\mid A)P(C\mid A,B)P(D\mid A,B,C)\end{aligned}$$

概率分布的展开形式，可以简化吗？

利用图论结构化的数据表示方式，可以让随机变量之间的依赖关系变得更为直观。

贝叶斯准则：

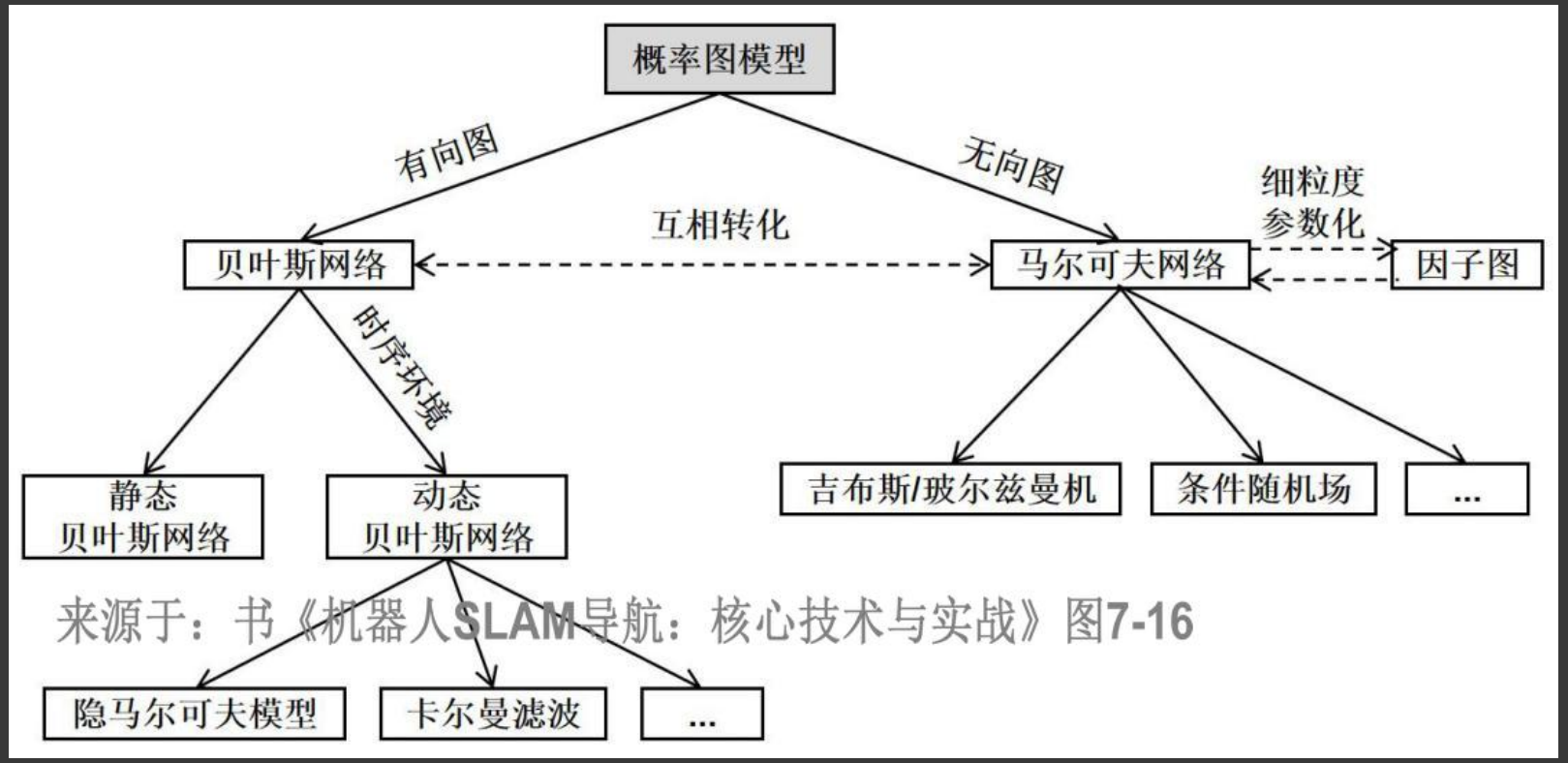
$$P(B\mid A)=\frac{P(A\mid B)P(B)}{P(A)}$$

一般形式

$$\begin{aligned}P(B\mid A,C)&=\frac{P(B,A,C)}{P(A,C)}\\&=\frac{P(A\mid B,C)P(B,C)}{P(A,C)}=\frac{P(A\mid B,C)P(B\mid C)P(C)}{P(A\mid C)P(C)}=\frac{P(A\mid B,C)P(B\mid C)}{P(A\mid C)}\end{aligned}$$

7.2 SLAM中的概率理论

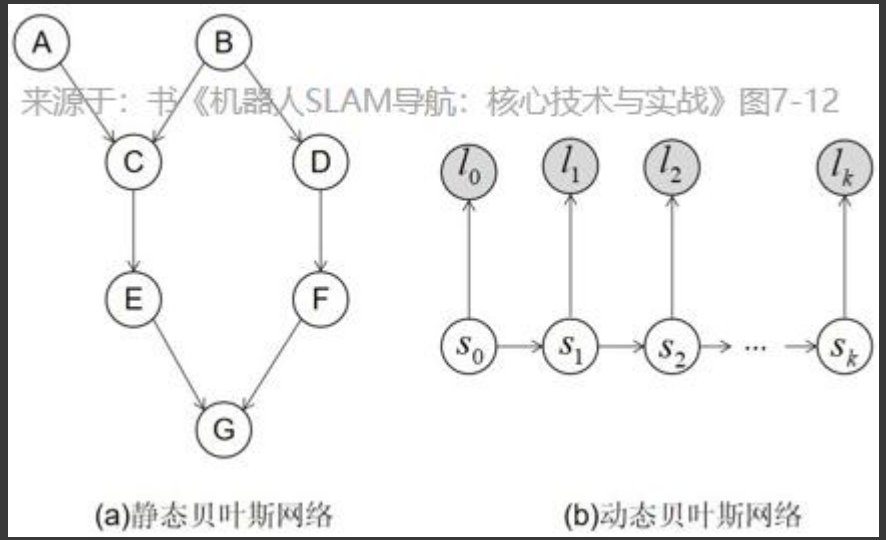
- 状态估计问题
- 概率运动模型
- 概率观测模型
- 概率图模型



来源于：书《机器人SLAM导航：核心技术与实战》图7-16

7.2 SLAM中的概率理论

- 状态估计问题
- 概率运动模型
- 概率观测模型
- 概率图模型
  - 贝叶斯网络
  - 马尔可夫网络
  - 因子图



箭头表示随机变量之间依赖的因果关系

条件独立性

联合分布**化简**形式：

$$P(A,B,C,D,E,F,G)=P(A)P(B)P(C\mid A,B)P(D\mid B)P(E\mid C)P(F\mid D)P(G\mid E,F)$$

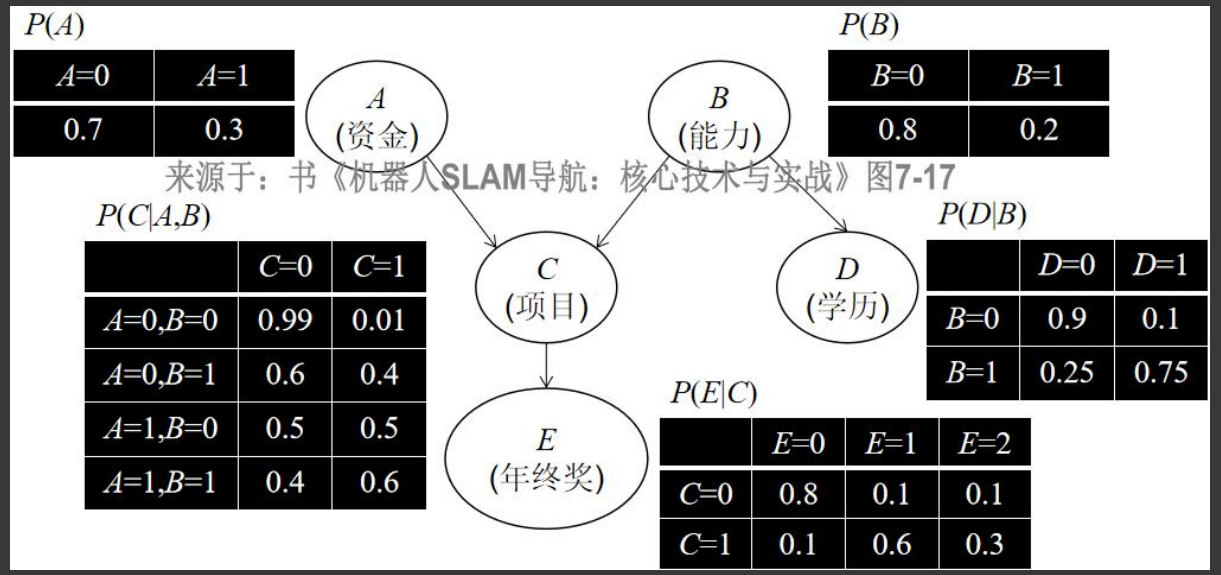
联合分布**化简**一般形式：

$$P(X_1,X_2,...,X_n)=\prod_{i=1}^{n}P(X_i\mid Pa_{X_i})$$

7.2 SLAM中的概率理论

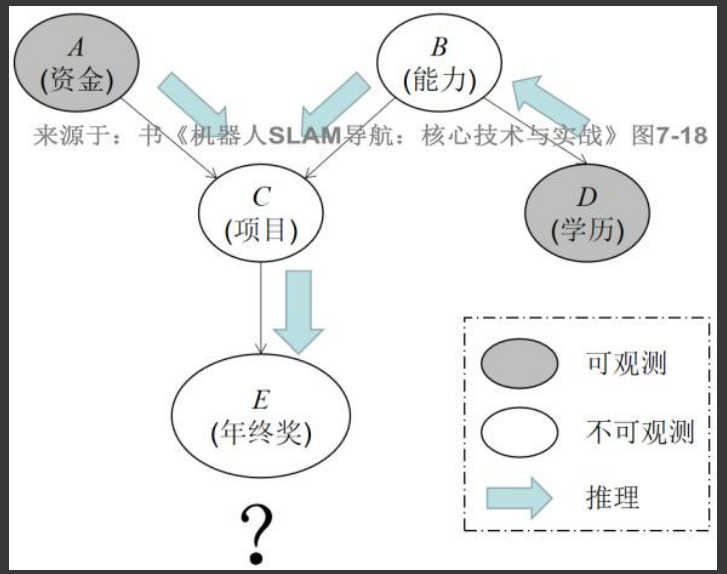
- 状态估计问题
- 概率运动模型
- 概率观测模型
- 概率图模型
  - 贝叶斯网络
  - 马尔可夫网络
  - 因子图

利用**贝叶斯网络**进行SLAM问题的**表示**及**推理**过程



静态贝叶斯网络【表示】

【学习】
- 模型结构
- 模型参数

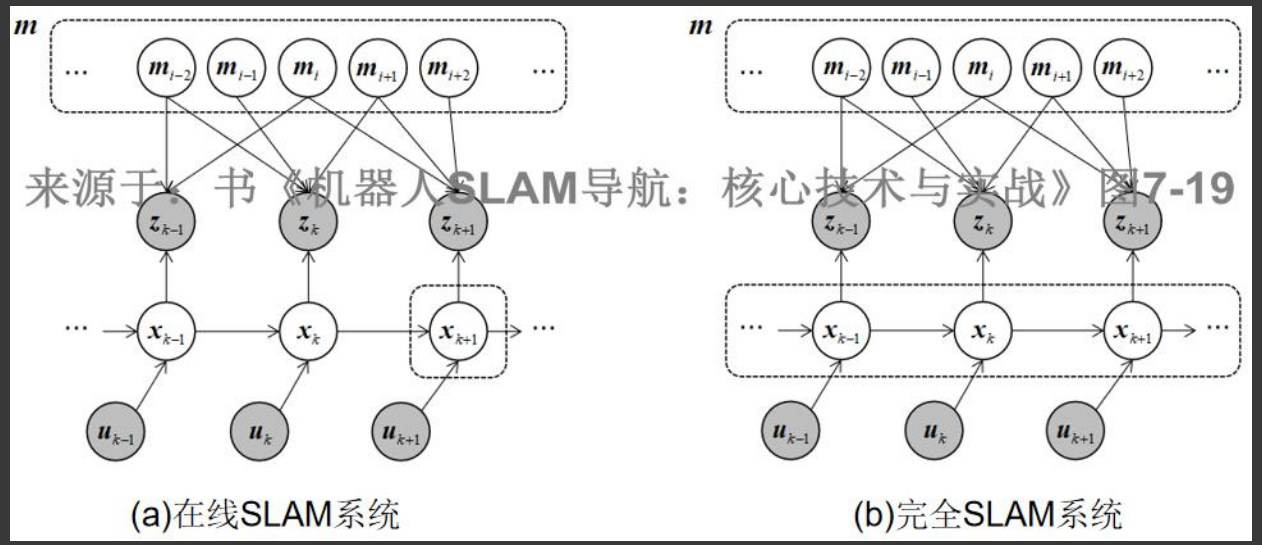


静态贝叶斯网络【推理】
- 给定观测
- 先验信息（模型结构、模型参数）
- 预测结果

7.2 SLAM中的概率理论

利用**贝叶斯网络**进行SLAM问题的**表示**及**推理**过程

- 状态估计问题
- 概率运动模型
- 概率观测模型
- 概率图模型 → 贝叶斯网络 / 马尔可夫网络 / 因子图



来源于：书《机器人SLAM导航：核心技术与实战》图7-19

动态贝叶斯网络【对SLAM问题的表示】

【学习】
- 模型结构
- 模型参数

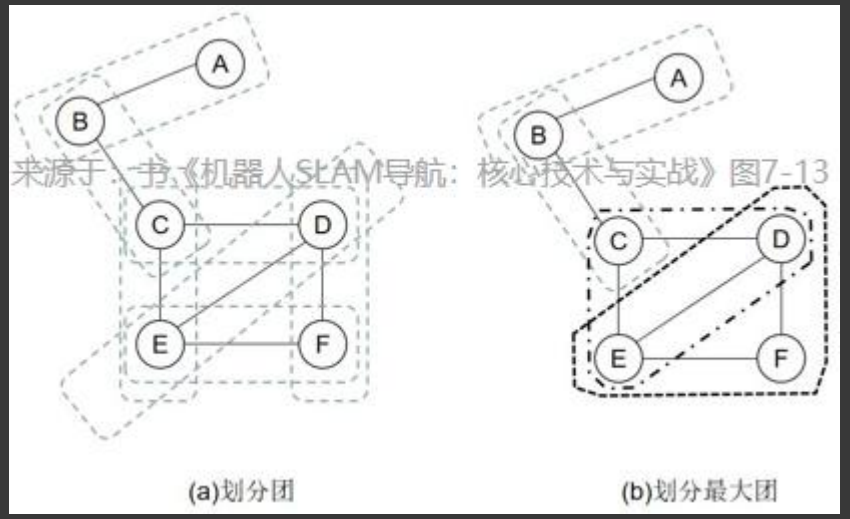
动态贝叶斯网络【对SLAM问题的推理】
- 给定观测
- 先验信息（模型结构、模型参数）
- 预测结果

在线SLAM：$P(\boldsymbol{x}_k, \boldsymbol{m} \mid \boldsymbol{Z}_{0:k}, \boldsymbol{U}_{0:k}, \boldsymbol{x}_0)$

完全SLAM：$P(\boldsymbol{X}_{0:k}, \boldsymbol{m} \mid \boldsymbol{Z}_{0:k}, \boldsymbol{U}_{0:k}, \boldsymbol{x}_0)$

7.2 SLAM中的概率理论

- 状态估计问题
- 概率运动模型
- 概率观测模型
- 概率图模型
  - 贝叶斯网络
  - 马尔可夫网络
  - 因子图



连线表示随机变量之间隐式相关性，可表示循环依赖

团中的节点必须两两之间都有连接，团的势能函数是一个抽象函数，需要在实际问题中被具体化为特定形式

不能再往节点中加入节点的团，就是最大团

条件独立性

联合分布**化简**形式：

$$P(A,B,C,D,E,F)=\frac{1}{Z}\psi_{Q_1}(A,B)\psi_{Q_2}(B,C)\psi_{Q_3}(C,D)\psi_{Q_4}(C,E)\psi_{Q_5}(D,E)\psi_{Q_6}(D,F)\psi_{Q_7}(E,F)$$

联合分布最大团**化简**形式：

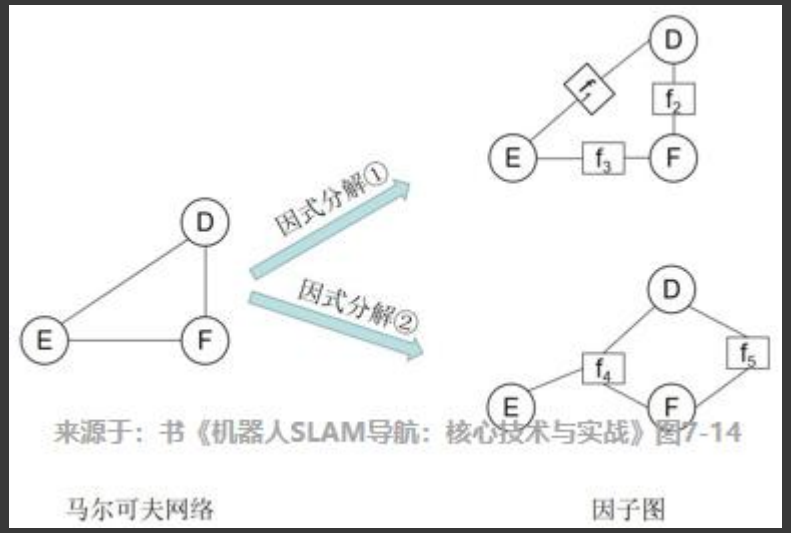
$$P(A,B,C,D,E,F)=\frac{1}{Z}\psi_{Q_1}(A,B)\psi_{Q_2}(B,C)\psi_{Q_8}(C,D,E)\psi_{Q_9}(D,E,F)$$

联合分布最大团**化简**一般形式：

$$P(X_1,X_2,...,X_n)=\frac{1}{Z}\prod_{i=1}^{m}\psi_{Q_i}(Q_i)$$

7.2 SLAM中的概率理论

- 状态估计问题
- 概率运动模型
- 概率观测模型
- 概率图模型
  - 贝叶斯网络
  - 马尔可夫网络
  - 因子图



马尔可夫网络中团的势能函数只能**笼统**表示团内节点之间的相关性

将团的势能函数进行因式分解，分解出来的因子项对随机变量之间的关系描述更**具体**

因式分解不唯一，所以马尔可夫网络到因子图的转换也不唯一

联合分布因子项**化简**形式1：$\psi_{Q_9}(D,E,F)=f_1(D,E)\bullet f_2(D,F)\bullet f_3(E,F)$

联合分布因子项**化简**形式2：$\psi_{Q_9}(D,E,F)=f_4(D,E,F)\bullet f_5(D,F)$

7.2 SLAM中的概率理论

- 状态估计问题
- 概率运动模型
- 概率观测模型
- 概率图模型 → 贝叶斯网络 / 马尔可夫网络 / 因子图

利用因子图进行SLAM问题的表示及推理过程

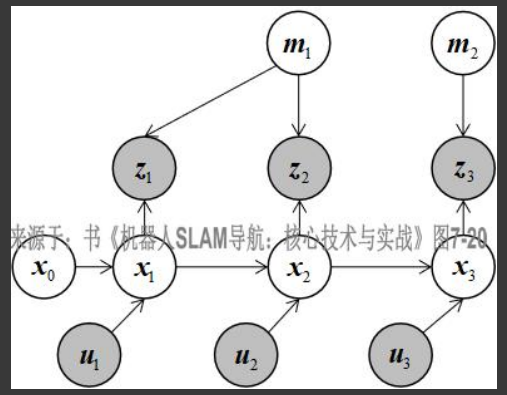
设：

机器人位姿状态：$\boldsymbol{X}=\{\boldsymbol{x}_0,\boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{x}_3\}$

路标：$\boldsymbol{m}=\{\boldsymbol{m}_1,\boldsymbol{m}_2\}$

运动量：$\boldsymbol{U}=\{\boldsymbol{u}_1,\boldsymbol{u}_2,\boldsymbol{u}_3\}$

观测量：$\boldsymbol{Z}=\{\boldsymbol{z}_1,\boldsymbol{z}_2,\boldsymbol{z}_3\}$



贝叶斯网络表示的SLAM问题

SLAM问题表示（联合分布）：

$$P(\boldsymbol{X},\boldsymbol{m},\boldsymbol{Z},\boldsymbol{U})=P(\boldsymbol{m}_1)P(\boldsymbol{m}_2)P(\boldsymbol{u}_0)P(\boldsymbol{u}_1)P(\boldsymbol{u}_2)\ \}\text{先验，可忽略}$$
$$\times P(\boldsymbol{x}_0)P(\boldsymbol{x}_1\mid\boldsymbol{x}_0,\boldsymbol{u}_1)P(\boldsymbol{x}_2\mid\boldsymbol{x}_1,\boldsymbol{u}_2)P(\boldsymbol{x}_3\mid\boldsymbol{x}_2,\boldsymbol{u}_3)\ \}\text{运动}$$
$$\times P(\boldsymbol{z}_1\mid\boldsymbol{x}_1,\boldsymbol{m}_1)P(\boldsymbol{z}_2\mid\boldsymbol{x}_2,\boldsymbol{m}_1)P(\boldsymbol{z}_3\mid\boldsymbol{x}_3,\boldsymbol{m}_2)\ \}\text{观测}$$

SLAM问题推理（状态估计）：

（利用给定观测状态估计未知状态，比如最大后验估计）设：

可直接量测信息：$\boldsymbol{V}=\{\boldsymbol{Z},\boldsymbol{U}\}$

不可直接量测信息（待估计量）：$\boldsymbol{S}=\{\boldsymbol{X},\boldsymbol{m}\}$

$$\boldsymbol{S}_{\text{MAP}}=\arg\max_{\boldsymbol{S}}P(\boldsymbol{S}\mid\boldsymbol{V})$$

将贝叶斯网络箭头的条件概率替换成因子图的最小二乘约束边

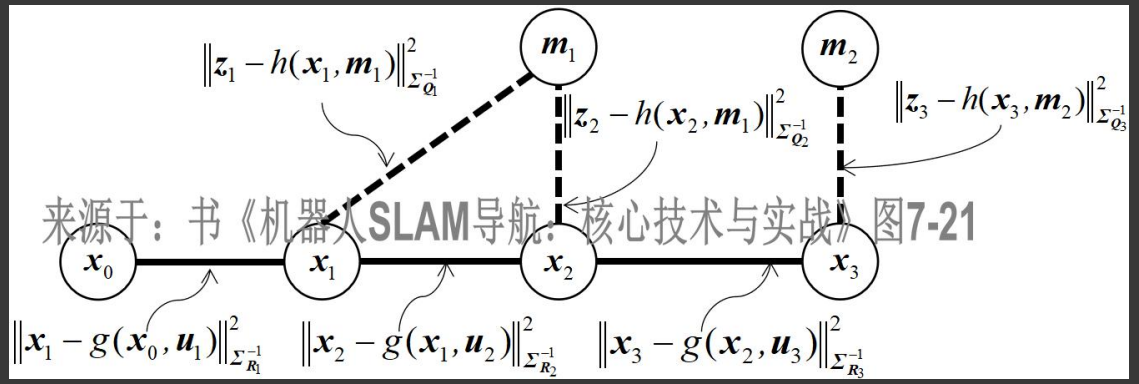
$$P(\boldsymbol{S}\mid\boldsymbol{V})$$
$$\propto P(\boldsymbol{x}_0)P(\boldsymbol{x}_1\mid\boldsymbol{x}_0,\boldsymbol{u}_1)P(\boldsymbol{x}_2\mid\boldsymbol{x}_1,\boldsymbol{u}_2)P(\boldsymbol{x}_3\mid\boldsymbol{x}_2,\boldsymbol{u}_3)\}\text{运动}$$
$$\times P(\boldsymbol{z}_1\mid\boldsymbol{x}_1,\boldsymbol{m}_1)P(\boldsymbol{z}_2\mid\boldsymbol{x}_2,\boldsymbol{m}_1)P(\boldsymbol{z}_3\mid\boldsymbol{x}_3,\boldsymbol{m}_2)\}\text{观测}$$

在获得直接量测信息$\boldsymbol{U}$和$\boldsymbol{Z}$后，利用条件概率分布关系对不可量测信息$\boldsymbol{X}$和$\boldsymbol{m}$进行推理，这就是SLAM问题的状态估计过程。

$$P(\boldsymbol{S}\mid\boldsymbol{V})\propto\psi(\boldsymbol{S})=\prod_i\psi_i(\boldsymbol{X},\boldsymbol{m},\boldsymbol{Z},\boldsymbol{U})$$

由于估计过程关心的是$\boldsymbol{X}$和$\boldsymbol{m}$，所以可以用运动和观测这两个约束来构建因子函数，并将不关心的$\boldsymbol{U}$和$\boldsymbol{Z}$隐藏到因子函数约束之中。

最小二乘约束形式因子图表示的SLAM问题



最小二乘估计

$$\boldsymbol{S}_{\text{MAP}}=\arg\max_{\boldsymbol{S}}\psi(\boldsymbol{S})$$
$$\propto\arg\max_{\boldsymbol{S}}(\sum_{i1}\ln\psi_{i1}(\boldsymbol{x}_k,\boldsymbol{x}_{k-1})+\sum_{i2}\ln\psi_{i2}(\boldsymbol{x}_k,\boldsymbol{m}_j))$$
$$\propto\arg\max_{\boldsymbol{S}}(-\frac{1}{2}\sum_{i1}(\boldsymbol{x}_k-g(\boldsymbol{x}_{k-1},\boldsymbol{u}_k))^{\mathrm{T}}\boldsymbol{\Sigma}_{\boldsymbol{R}_k}^{-1}(\boldsymbol{x}_k-g(\boldsymbol{x}_{k-1},\boldsymbol{u}_k))$$
$$-\frac{1}{2}\sum_{i2}(\boldsymbol{z}_k-h(\boldsymbol{x}_k,\boldsymbol{m}_j))^{\mathrm{T}}\boldsymbol{\Sigma}_{\boldsymbol{Q}_k}^{-1}(\boldsymbol{z}_k-h(\boldsymbol{x}_k,\boldsymbol{m}_j)))$$
$$\propto\arg\min_{\boldsymbol{S}}(\sum_{i1}\|\boldsymbol{x}_k-g(\boldsymbol{x}_{k-1},\boldsymbol{u}_k)\|^2_{\boldsymbol{\Sigma}_{\boldsymbol{R}_k}^{-1}}+\sum_{i2}\|\boldsymbol{z}_k-h(\boldsymbol{x}_k,\boldsymbol{m}_j)\|^2_{\boldsymbol{\Sigma}_{\boldsymbol{Q}_k}^{-1}})$$

- 假设运动噪声和观测噪声都是0均值高斯分布
- 用运动误差和观测误差分别来构造运动约束和观测约束这2种因子项
- 误差项采用分布的协方差矩阵加权求和

* 关于SLAM问题表示从贝叶斯网络转换到因子图的具体数学推导，将在第2季课程中详细讨论

内容概要

7.3 估计理论

不管是用**贝叶斯网络**还是**因子图**，一旦SLAM问题用概率图模型得到表示后，

接下来就是利用**可观测量**推理**不可观测量**，也就是说SLAM问题的求解过程是一个**状态估计**。

7.3 估计理论

- 估计量的性质
- 估计量的构建
- 各估计量比较

所谓**估计**，就是研究某问题时感兴趣的参数不能够通过精确测量得知，只能通过一组观测样本值来猜测参数的可能取值。

有时候估计量比较**直观**，比如估计室内的温度，感兴趣的参数θ就是温度；

有时候估计量并**不直观**，比如估计机械零件的精度，感兴趣的参数θ是误差分布函数f的特征量。

估计既然是一种猜测行为，可以毫无根据的瞎猜，也可以依据严密的逻辑策略进行科学的猜测。

这就涉及到如何评价估计量的好坏程度，借助**估计量的性质**可以对估计好坏程度进行评价。

①一致性：

强

$\hat{\theta} \xrightarrow[Z=\{z_k\}|k\to\infty]{} \theta$ 依概率1收敛（几乎处处收敛）

$\hat{\theta} \xrightarrow[Z=\{z_k\}|k\to\infty]{P} \theta$ 依概率收敛（测度收敛）

$\hat{\theta} \xrightarrow[Z=\{z_k\}|k\to\infty]{d} \theta$ 依分布收敛

$\hat{\theta} \xrightarrow[Z=\{z_k\}|k\to\infty]{q} \theta$ 依范数收敛

弱

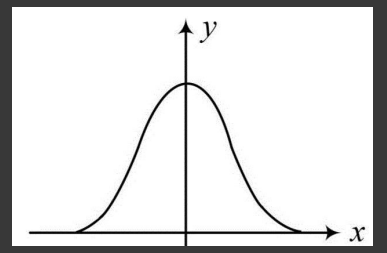
②偏差性：

$E[\hat{\theta}^k]$ k=1，一阶原点矩（期望）

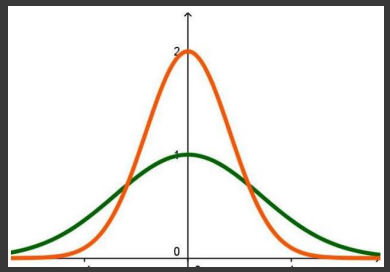
$E[(\hat{\theta}-E[\hat{\theta}])^k]$ k=2，二阶中心矩（方差）

无偏估计？

$E[\hat{\theta}]=\theta$



最小方差估计？



无偏估计只是保证估计量在期望上是正确的，估计量本身还是有不确定性，方差描述了估计量的不确定性，方差越小估计不确定性越小

*最小方差的存在性，可以用CRLB来判别

7.3 估计理论

好的估计量应该是**最小方差无偏估计**（Minimum Variance Unbiased Estimation，MVUE），
然而这会非常困难，因此需要寻找**近似方法**

- 估计量的性质
- 估计量的构建
- 各估计量比较

最大似然估计（MLE）：

$$\hat{\theta}_{\text{MLE}} = \arg\max_{\theta} l(\theta \mid Z) = \arg\max_{\theta} \sum_{k} \ln(P(z_k \mid \theta))$$

最小二乘估计（LSE）：

$$\hat{\theta}_{\text{LSE}} = \arg\min_{\theta} J(\theta) = \arg\min_{\theta} \sum_{k} (z_k - h(x_k, \theta))^2$$

贝叶斯估计（Bayes）：

$$\hat{\theta}_{\text{Bayes}} = \arg\min_{\hat{\theta}} \gamma = \arg\min_{\hat{\theta}} E_{P(Z,\theta)}[C(e)] = \arg\min_{\hat{\theta}} \iint C(\theta - \hat{\theta}) P(Z,\theta) dZ d\theta = \arg\min_{\hat{\theta}} \int \left[\int C(\theta - \hat{\theta}) P(\theta \mid Z) d\theta \right] P(Z) dZ$$

最小均方误差估计（MMSE）：

$$\hat{\theta}_{\text{MMSE}} = \arg\min_{\hat{\theta}} E_{P(Z,\theta)}[C(e)] = \arg\min_{\hat{\theta}} \int \left[\int (\theta - \hat{\theta})^2 P(\theta \mid Z) d\theta \right] P(Z) dZ$$

化简

$$\hat{\theta}_{\text{MMSE}} = \int \theta P(\theta \mid Z) d\theta = E_{P(\theta \mid Z)}[\theta]$$

最大后验估计（MAP）：

$$\hat{\theta}_{\text{MAP}} = \arg\min_{\hat{\theta}} E_{P(Z,\theta)}[C(e)] = \arg\min_{\hat{\theta}} \int \left[\int_{-\infty}^{\hat{\theta}-\delta} 1 \cdot P(\theta \mid Z) d\theta + \int_{\hat{\theta}+\delta}^{\infty} 1 \cdot P(\theta \mid Z) d\theta \right] P(Z) dZ = \arg\min_{\hat{\theta}} \int \left[1 - \int_{\hat{\theta}-\delta}^{\hat{\theta}+\delta} P(\theta \mid Z) d\theta \right] P(Z) dZ \propto \arg\max_{\hat{\theta}} \int_{\hat{\theta}-\delta}^{\hat{\theta}+\delta} P(\theta \mid Z) d\theta$$

化简

$$\hat{\theta}_{\text{MAP}} = \arg\max_{\theta} P(\theta \mid Z) = \arg\max_{\theta} \frac{P(Z \mid \theta) P(\theta)}{P(Z)} \propto \arg\max_{\theta} \underbrace{P(Z \mid \theta)}_{\text{似然}} \cdot \underbrace{P(\theta)}_{\text{先验}}$$

均方误差与方差的区别和联系？
（在无偏估计下，均方误差等于方差）

$$\begin{aligned} & MSE[\hat{\theta}] \\ &= E[(\hat{\theta} - \theta)^2] \\ &= E[\hat{\theta}^2 - 2\hat{\theta}\theta + \theta^2] \\ &= E[\hat{\theta}^2] - 2\theta E[\hat{\theta}] + \theta^2 \\ &= (E[\hat{\theta}^2] - (E[\hat{\theta}])^2) + (E[\hat{\theta}] - \theta)^2 \\ &= D[\hat{\theta}] + (E[\hat{\theta}] - \theta)^2 \end{aligned}$$

* 关于公式的具体数学推导，将在第2季课程中详细讨论

7.3 估计理论

实际情况，观测到的样本数量不可能无穷多，因此**一致性**只是理论上需要满足的条件。在样本数量有限时，讨论估计值与实际值之间的偏差将更有意义，也就是**偏差性**。所以估计策略的目标是让估计量的偏差量尽量小，最小方差无偏估计无疑是一个理想的估计量。然而，最小方差无偏估计实现非常困难，因此需要寻找**近似策略**

- 估计量的性质
- 估计量的构建
- 各估计量比较
  - 从策略角度对比
  - 从模型角度对比
  - 等价转换关系

最大似然估计假设概率分布模型是知道的，只是概率分布模型中的参数θ未知。尝试θ的不同取值，看模型是否能输出与观测结果一样的数据。当某个θ取值能让模型输出与观测结果一样的数据的概率最大，那么这个θ取值就是最适合的估计参数。

$$\begin{aligned}\hat{\theta}_{\mathrm{MLE}} &= \arg\max_{\theta} l(\theta \mid Z) \\ &= \arg\max_{\theta} \sum_{k} \ln(P(z_k \mid \theta))\end{aligned}$$

最小二乘估计是计算观测样本点与模型实际点之间的平方误差，求使得该平方误差最小的参数值θ，求出来的这个参数值θ就是估计参数

$$\begin{aligned}\hat{\theta}_{\mathrm{LSE}} &= \arg\min_{\theta} J(\theta) \\ &= \arg\min_{\theta} \sum_{k} (z_k - h(x_k, \theta))^2\end{aligned}$$

贝叶斯估计首先也是构建关于误差的代价函数，然后最小化代价函数在概率密度下的期望

$$\begin{aligned}\hat{\theta}_{\mathrm{Bayes}} &= \arg\min_{\hat{\theta}} \gamma \\ &= \arg\min_{\hat{\theta}} E_{P(Z,\theta)}[C(e)] \\ &= \arg\min_{\hat{\theta}} \iint C(\theta - \hat{\theta}) P(Z, \theta) dZ d\theta \\ &= \arg\min_{\hat{\theta}} \int \left[\int C(\theta - \hat{\theta}) P(\theta \mid Z) d\theta \right] P(Z) dZ\end{aligned}$$

- 可以看出，贝叶斯估计中的C(e)是关于误差e的抽象函数，也就是说贝叶斯估计是一个通用的形式，具体的估计量形式与函数C(e)的具体实现有关。最小均方误差估计和最大后验估计，就是贝叶斯估计的常见形式。
- **最小均方误差估计**就是用θ在后验分布上的概率**均值**作为估计取值。
- **最大后验估计**就是用θ在后验分布上最大值（也叫的**众数**）处对应的值作为估计取值。

7.3 估计理论

- 估计量的性质
- 估计量的构建
- 各估计量比较
  - 从策略角度对比
  - 从模型角度对比
  - 等价转换关系

	（概率）模型	模型待估参数
最大似然估计	已知	未知确定量
最小二乘估计	未知	未知确定量
（贝叶斯估计）	已知	未知随机量
最小均方误差估计	已知	未知随机量
最大后验估计	已知	未知随机量

来源于：书《机器人SLAM导航：核心技术与实战》表7-1

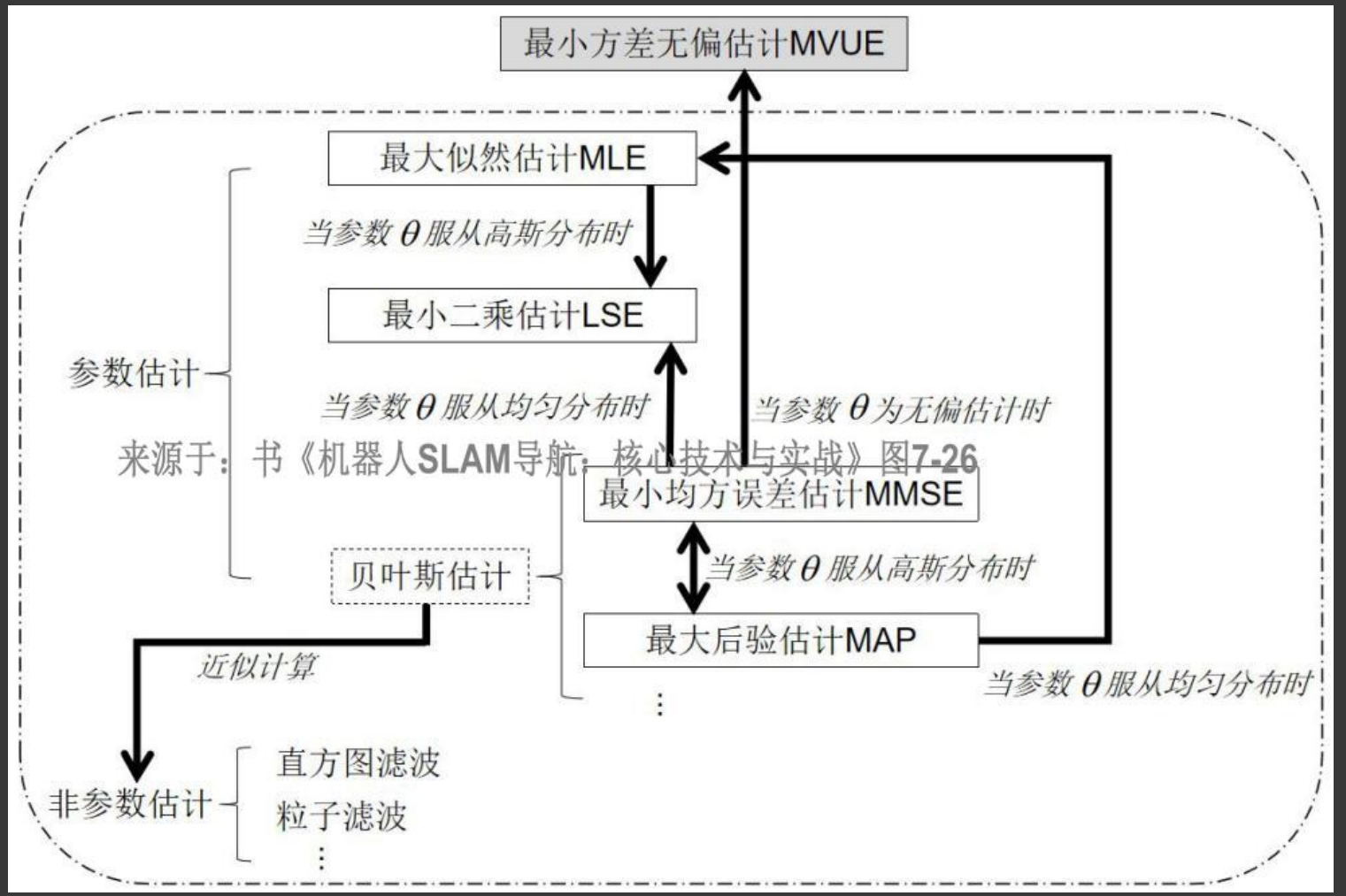
（概率）模型已知 VS （概率）模型未知

模型待估参数确定 VS 模型待估参数随机

经典估计 VS 贝叶斯估计

7.3 估计理论

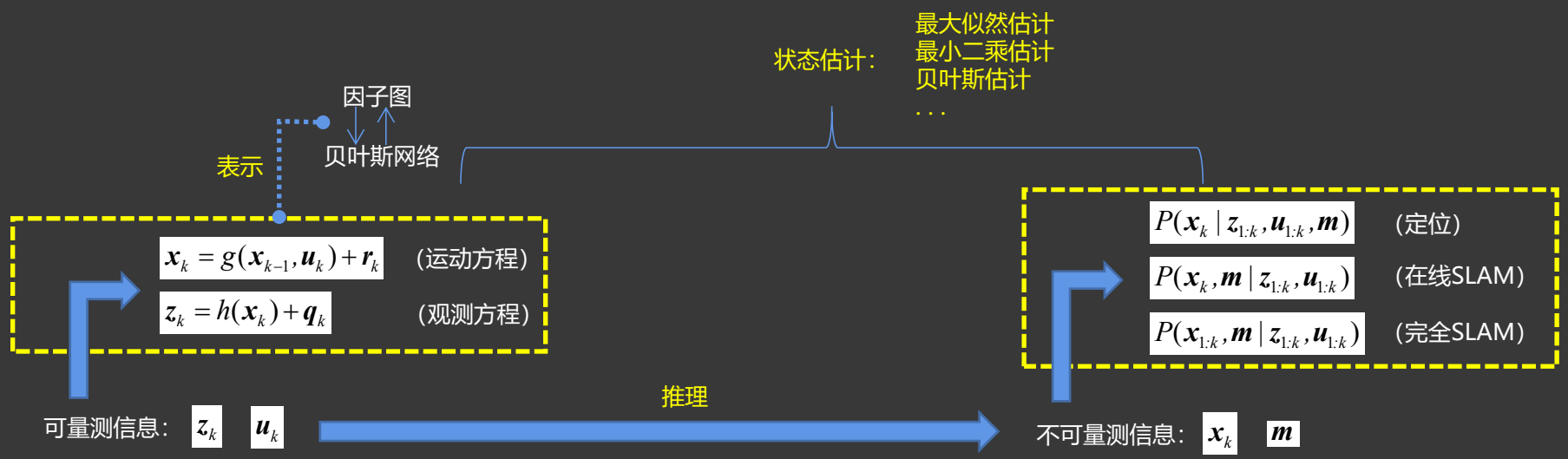
- 估计量的性质
- 估计量的构建
- 各估计量比较
  - 从策略角度对比
  - 从模型角度对比
  - 等价转换关系



* 关于公式的具体数学推导，将在第2季课程中详细讨论

内容概要

7.4 基于贝叶斯网络的状态估计



7.4 基于贝叶斯网络的状态估计

- 贝叶斯估计 ➡ 以贝叶斯网络（表示）、定位(待估问题)、贝叶斯估计（求解方法）这种情况为例来讨论：
- 参数化实现
- 非参数化实现

置信度，其实是对**后验概率分布**的符号化简写：

$$bel(\boldsymbol{x}_k)=P(\boldsymbol{x}_k\mid \boldsymbol{z}_{1:k},\boldsymbol{u}_{1:k})$$

贝叶斯准则

$$=\frac{P(\boldsymbol{z}_k\mid \boldsymbol{x}_k,\boldsymbol{z}_{1:k-1},\boldsymbol{u}_{1:k})P(\boldsymbol{x}_k\mid \boldsymbol{z}_{1:k-1},\boldsymbol{u}_{1:k})}{P(\boldsymbol{z}_k\mid \boldsymbol{z}_{1:k-1},\boldsymbol{u}_{1:k})}$$

$$=\eta\cdot P(\boldsymbol{z}_k\mid \boldsymbol{x}_k,\boldsymbol{z}_{1:k-1},\boldsymbol{u}_{1:k})P(\boldsymbol{x}_k\mid \boldsymbol{z}_{1:k-1},\boldsymbol{u}_{1:k})$$

贝叶斯网络中的条件独立性：$P(\boldsymbol{z}_k\mid \boldsymbol{x}_k,\boldsymbol{z}_{1:k-1},\boldsymbol{u}_{1:k})=P(\boldsymbol{z}_k\mid \boldsymbol{x}_k)$

$$bel(\boldsymbol{x}_k)=\eta\cdot P(\boldsymbol{z}_k\mid \boldsymbol{x}_k)\cdot\overline{bel(\boldsymbol{x}_k)}$$

全概率展开：

$$\overline{bel(\boldsymbol{x}_k)}=P(\boldsymbol{x}_k\mid \boldsymbol{z}_{1:k-1},\boldsymbol{u}_{1:k})$$

$$=\int P(\boldsymbol{x}_k\mid \boldsymbol{x}_{k-1},\boldsymbol{z}_{1:k-1},\boldsymbol{u}_{1:k})P(\boldsymbol{x}_{k-1}\mid \boldsymbol{z}_{1:k-1},\boldsymbol{u}_{1:k})d\boldsymbol{x}_{k-1}$$

贝叶斯网络中的条件独立性：$P(\boldsymbol{x}_k\mid \boldsymbol{x}_{k-1},\boldsymbol{z}_{1:k-1},\boldsymbol{u}_{1:k})=P(\boldsymbol{x}_k\mid \boldsymbol{x}_{k-1},\boldsymbol{u}_k)$

常识：$P(\boldsymbol{x}_{k-1}\mid \boldsymbol{z}_{1:k-1},\boldsymbol{u}_{1:k})=P(\boldsymbol{x}_{k-1}\mid \boldsymbol{z}_{1:k-1},\boldsymbol{u}_{1:k-1})$

$$\overline{bel(\boldsymbol{x}_k)}=P(\boldsymbol{x}_k\mid \boldsymbol{z}_{1:k-1},\boldsymbol{u}_{1:k})$$

$$=\int P(\boldsymbol{x}_k\mid \boldsymbol{x}_{k-1},\boldsymbol{z}_{1:k-1},\boldsymbol{u}_{1:k})P(\boldsymbol{x}_{k-1}\mid \boldsymbol{z}_{1:k-1},\boldsymbol{u}_{1:k})d\boldsymbol{x}_{k-1}$$

$$=\int P(\boldsymbol{x}_k\mid \boldsymbol{x}_{k-1},\boldsymbol{u}_k)P(\boldsymbol{x}_{k-1}\mid \boldsymbol{z}_{1:k-1},\boldsymbol{u}_{1:k-1})d\boldsymbol{x}_{k-1}$$

$$=\int P(\boldsymbol{x}_k\mid \boldsymbol{x}_{k-1},\boldsymbol{u}_k)\cdot bel(\boldsymbol{x}_{k-1})d\boldsymbol{x}_{k-1}$$

递归贝叶斯滤波：

$$\begin{cases}\overline{bel(\boldsymbol{x}_k)}=\int P(\boldsymbol{x}_k\mid \boldsymbol{x}_{k-1},\boldsymbol{u}_k)\cdot bel(\boldsymbol{x}_{k-1})d\boldsymbol{x}_{k-1} & \text{运动预测}\\ bel(\boldsymbol{x}_k)=\eta\cdot P(\boldsymbol{z}_k\mid \boldsymbol{x}_k)\overline{bel(\boldsymbol{x}_k)} & \text{观测更新}\end{cases}$$

思考：

- 递归过程为何是求待估量的概率分布（即置信度）？
- 为何不直接求出待估量的具体取值呢？

* 关于公式的具体数学推导，将在第2季课程中详细讨论

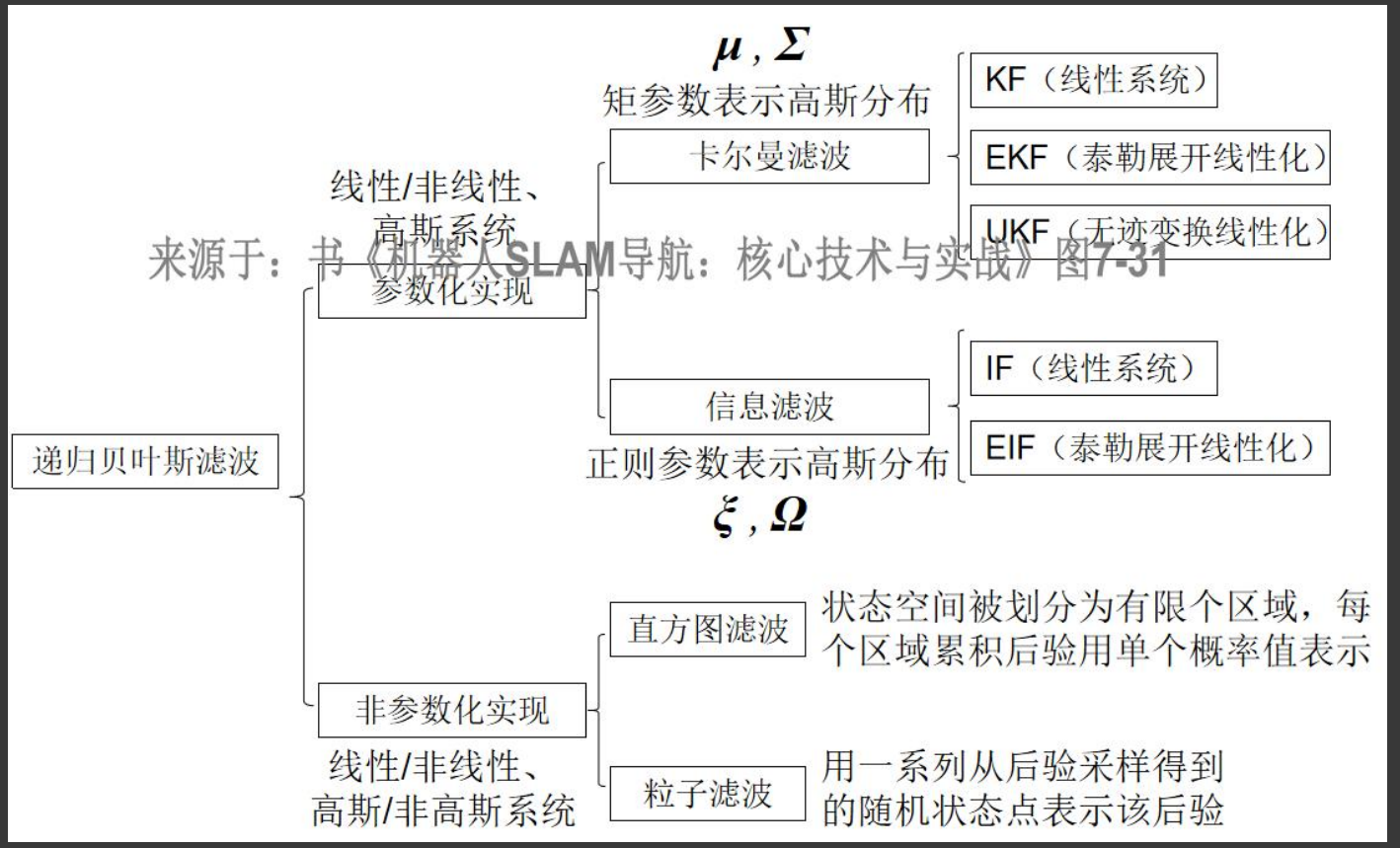
7.4 基于贝叶斯网络的状态估计

- 贝叶斯估计 ➡ 以贝叶斯网络（表示）、定位(待估问题)、贝叶斯估计（求解方法）这种情况为例来讨论：
- 参数化实现
- 非参数化实现

由于后验概率分布 $P(x_k \mid z_{1:k}, u_{1:k})$ 没有给定确切的形式，也就是说递归贝叶斯滤波是一种通用框架。在给定不同形式的 $P(x_k \mid z_{1:k}, u_{1:k})$ 分布后，递归贝叶斯滤波也就对应不同形式的算法实现。

	高斯分布	非高斯分布
线性系统	线性高斯系统：KF、IF	线性非高斯系统
非线性系统	非线性高斯系统：EKF、UKF、EIF	非线性非高斯系统：HF、PF

来源于：书《机器人SLAM导航：核心技术与实战》表7-2



来源于：书《机器人SLAM导航：核心技术与实战》图7-31

7.4 基于贝叶斯网络的状态估计

高斯分布可以用矩参数（均值和方差）进行表示，那么多维高斯分布为：

$$P(\boldsymbol{x})=\frac{1}{\sqrt{(2\pi)^n\det(\boldsymbol{\Sigma})}}\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$

- 贝叶斯估计
- 参数化实现
- 非参数化实现

参数化实现：
- 卡尔曼滤波
  - 线性卡尔曼滤波（线性高斯系统）
  - 扩展卡尔曼滤波（非线性高斯系统）
  - 无迹卡尔曼滤波（非线性高斯系统）
- 信息滤波
  - 线性信息滤波（线性高斯系统）
  - 扩展信息滤波（非线性高斯系统）

运动方程：$\boldsymbol{x}_k=\boldsymbol{A}_k\boldsymbol{x}_{k-1}+\boldsymbol{B}_k\boldsymbol{u}_k+\boldsymbol{r}_k$ → 概率运动模型：

$$P(\boldsymbol{x}_k\mid\boldsymbol{x}_{k-1},\boldsymbol{u}_k)=\frac{1}{\sqrt{(2\pi)^n\det(\boldsymbol{R}_k)}}\exp\left(-\frac{1}{2}(\boldsymbol{x}_k-(\boldsymbol{A}_k\boldsymbol{x}_{k-1}+\boldsymbol{B}_k\boldsymbol{u}_k))^{\mathrm{T}}\boldsymbol{R}_k^{-1}(\boldsymbol{x}_k-(\boldsymbol{A}_k\boldsymbol{x}_{k-1}+\boldsymbol{B}_k\boldsymbol{u}_k))\right)$$

观测方程：$\boldsymbol{z}_k=\boldsymbol{C}_k\boldsymbol{x}_k+\boldsymbol{q}_k$ → 概率观测模型：

$$P(\boldsymbol{z}_k\mid\boldsymbol{x}_k)=\frac{1}{\sqrt{(2\pi)^n\det(\boldsymbol{Q}_k)}}\exp\left(-\frac{1}{2}(\boldsymbol{z}_k-\boldsymbol{C}_k\boldsymbol{x}_k)^{\mathrm{T}}\boldsymbol{Q}_k^{-1}(\boldsymbol{z}_k-\boldsymbol{C}_k\boldsymbol{x}_k)\right)$$

待估状态量置信度的高斯分布形式：

$$bel(\boldsymbol{x}_k)=\frac{1}{\sqrt{(2\pi)^n\det(\boldsymbol{\Sigma}_k)}}\exp\left(-\frac{1}{2}(\boldsymbol{x}_k-\boldsymbol{\mu}_k)^{\mathrm{T}}\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_k-\boldsymbol{\mu}_k)\right)$$

* 卡尔曼滤波其实就是利用下面的5个核心公式，递归计算状态 $\boldsymbol{x}_k$ 置信度的参数 $\boldsymbol{\mu}_k$ 和 $\boldsymbol{\Sigma}_k$

$$\begin{cases}①\ \overline{\boldsymbol{\mu}_k}=\boldsymbol{A}_k\boldsymbol{\mu}_{k-1}+\boldsymbol{B}_k\boldsymbol{u}_k\\ ②\ \overline{\boldsymbol{\Sigma}_k}=\boldsymbol{A}_k\boldsymbol{\Sigma}_{k-1}\boldsymbol{A}_k^{\mathrm{T}}+\boldsymbol{R}_k\end{cases}\}\text{运动预测}$$

$$③\ \boldsymbol{K}_k=\overline{\boldsymbol{\Sigma}_k}\boldsymbol{C}_k^{\mathrm{T}}(\boldsymbol{C}_k\overline{\boldsymbol{\Sigma}_k}\boldsymbol{C}_k^{\mathrm{T}}+\boldsymbol{Q}_k)^{-1}\ \}\text{卡尔曼增益}$$

$$\begin{cases}④\ \boldsymbol{\mu}_k=\overline{\boldsymbol{\mu}_k}+\boldsymbol{K}_k(\boldsymbol{z}_k-\boldsymbol{C}_k\overline{\boldsymbol{\mu}_k})\\ ⑤\ \boldsymbol{\Sigma}_k=(\boldsymbol{I}-\boldsymbol{K}_k\boldsymbol{C}_k)\overline{\boldsymbol{\Sigma}_k}\end{cases}\}\text{观测更新}$$

线性卡尔曼滤波（KF）

高斯分布矩参数化

高斯分布矩参数化

$$\begin{cases}\overline{bel(\boldsymbol{x}_k)}=\int P(\boldsymbol{x}_k\mid\boldsymbol{x}_{k-1},\boldsymbol{u}_k)\cdot bel(\boldsymbol{x}_{k-1})d\boldsymbol{x}_{k-1}\ \}\text{运动预测}\\ bel(\boldsymbol{x}_k)=\eta\cdot P(\boldsymbol{z}_k\mid\boldsymbol{x}_k)\overline{bel(\boldsymbol{x}_k)}\ \}\text{观测更新}\end{cases}$$

递归贝叶斯滤波

* 卡尔曼滤波是递归贝叶斯滤波在高斯分布矩参数化下的具体实现形式

7.4 基于贝叶斯网络的状态估计

高斯分布可以用矩参数（均值和方差）进行表示，那么多维高斯分布为：

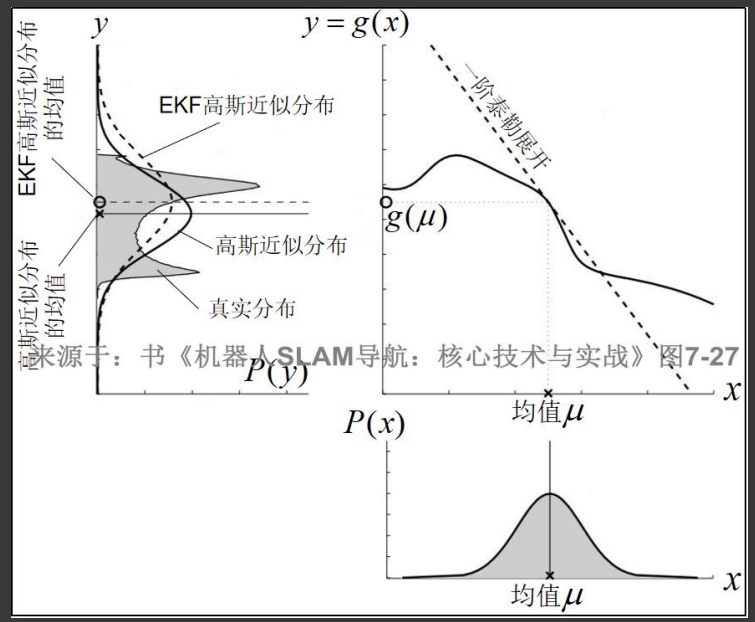
$$P(\boldsymbol{x})=\frac{1}{\sqrt{(2\pi)^n\det(\boldsymbol{\Sigma})}}\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$

- 贝叶斯估计
- 参数化实现
- 非参数化实现

卡尔曼滤波
- 线性卡尔曼滤波（线性高斯系统）
- 扩展卡尔曼滤波（非线性高斯系统）
- 无迹卡尔曼滤波（非线性高斯系统）

信息滤波
- 线性信息滤波（线性高斯系统）
- 扩展信息滤波（非线性高斯系统）

实际机器人运动方程和观测方程往往是非线性的，也就是说高斯随机变量经过非线性函数g和h变换后都将变成非高斯随机变量，因此计算过程需要先线性化近似处理



来源于：书《机器人SLAM导航：核心技术与实战》图7-27

一阶泰勒展开进行线性化

$$\begin{cases}①\ \overline{\boldsymbol{\mu}_k}=g(\boldsymbol{\mu}_{k-1},\boldsymbol{u}_k)\\ ②\ \overline{\boldsymbol{\Sigma}_k}=\boldsymbol{G}_k\boldsymbol{\Sigma}_{k-1}\boldsymbol{G}_k^{\mathrm{T}}+\boldsymbol{R}_k\end{cases}\text{运动预测}$$

$$③\ \boldsymbol{K}_k=\overline{\boldsymbol{\Sigma}_k}\boldsymbol{H}_k^{\mathrm{T}}(\boldsymbol{H}_k\overline{\boldsymbol{\Sigma}_k}\boldsymbol{H}_k^{\mathrm{T}}+\boldsymbol{Q}_k)^{-1}\ \text{卡尔曼增益}$$

$$\begin{cases}④\ \boldsymbol{\mu}_k=\overline{\boldsymbol{\mu}_k}+\boldsymbol{K}_k(\boldsymbol{z}_k-h(\overline{\boldsymbol{\mu}_k}))\\ ⑤\ \boldsymbol{\Sigma}_k=(\boldsymbol{I}-\boldsymbol{K}_k\boldsymbol{H}_k)\overline{\boldsymbol{\Sigma}_k}\end{cases}\text{观测更新}$$

扩展卡尔曼滤波（EKF）

$$\begin{aligned}g(\boldsymbol{x}_{k-1},\boldsymbol{u}_k)&\approx g(\boldsymbol{x}_{k-1},\boldsymbol{u}_k)|_{\boldsymbol{x}_{k-1}=\boldsymbol{\mu}_{k-1}}+g'(\boldsymbol{x}_{k-1},\boldsymbol{u}_k)|_{\boldsymbol{x}_{k-1}=\boldsymbol{\mu}_{k-1}}\bullet(\boldsymbol{x}_{k-1}-\boldsymbol{\mu}_{k-1})\\&=g(\boldsymbol{\mu}_{k-1},\boldsymbol{u}_k)+g'(\boldsymbol{\mu}_{k-1},\boldsymbol{u}_k)\bullet(\boldsymbol{x}_{k-1}-\boldsymbol{\mu}_{k-1})\\&=g(\boldsymbol{\mu}_{k-1},\boldsymbol{u}_k)+\boldsymbol{G}_k\bullet(\boldsymbol{x}_{k-1}-\boldsymbol{\mu}_{k-1})\end{aligned}$$

运动方程：$\boldsymbol{x}_k=g(\boldsymbol{x}_{k-1},\boldsymbol{u}_k)+\boldsymbol{r}_k$ → 一阶泰勒展开进行线性化 → 概率运动模型：

$$P(\boldsymbol{x}_k\mid\boldsymbol{x}_{k-1},\boldsymbol{u}_k)=\frac{1}{\sqrt{(2\pi)^n\det(\boldsymbol{R}_k)}}\exp\left(\begin{array}{l}-\frac{1}{2}(\boldsymbol{x}_k-(g(\boldsymbol{\mu}_{k-1},\boldsymbol{u}_k)+\boldsymbol{G}_k\bullet(\boldsymbol{x}_{k-1}-\boldsymbol{\mu}_{k-1})))^{\mathrm{T}}\\ \bullet\boldsymbol{R}_k^{-1}(\boldsymbol{x}_k-(g(\boldsymbol{\mu}_{k-1},\boldsymbol{u}_k)+\boldsymbol{G}_k\bullet(\boldsymbol{x}_{k-1}-\boldsymbol{\mu}_{k-1})))\end{array}\right)$$

观测方程：$\boldsymbol{z}_k=h(\boldsymbol{x}_k)+\boldsymbol{q}_k$ → 一阶泰勒展开进行线性化 → 概率观测模型：

$$\begin{aligned}h(\boldsymbol{x}_k)&\approx h(\boldsymbol{x}_k)|_{\boldsymbol{x}_k=\overline{\boldsymbol{\mu}_k}}+h'(\boldsymbol{x}_k)|_{\boldsymbol{x}_k=\overline{\boldsymbol{\mu}_k}}\bullet(\boldsymbol{x}_k-\overline{\boldsymbol{\mu}_k})\\&=h(\overline{\boldsymbol{\mu}_k})+h'(\overline{\boldsymbol{\mu}_k})\bullet(\boldsymbol{x}_k-\overline{\boldsymbol{\mu}_k})\\&=h(\overline{\boldsymbol{\mu}_k})+\boldsymbol{H}_k\bullet(\boldsymbol{x}_k-\overline{\boldsymbol{\mu}_k})\end{aligned}$$

$$P(\boldsymbol{z}_k\mid\boldsymbol{x}_k)=\frac{1}{\sqrt{(2\pi)^n\det(\boldsymbol{Q}_k)}}\exp\left(\begin{array}{l}-\frac{1}{2}(\boldsymbol{z}_k-(h(\overline{\boldsymbol{\mu}_k})+\boldsymbol{H}_k\bullet(\boldsymbol{x}_k-\overline{\boldsymbol{\mu}_k})))^{\mathrm{T}}\\ \bullet\boldsymbol{Q}_k^{-1}(\boldsymbol{z}_k-(h(\overline{\boldsymbol{\mu}_k})+\boldsymbol{H}_k\bullet(\boldsymbol{x}_k-\overline{\boldsymbol{\mu}_k})))\end{array}\right)$$

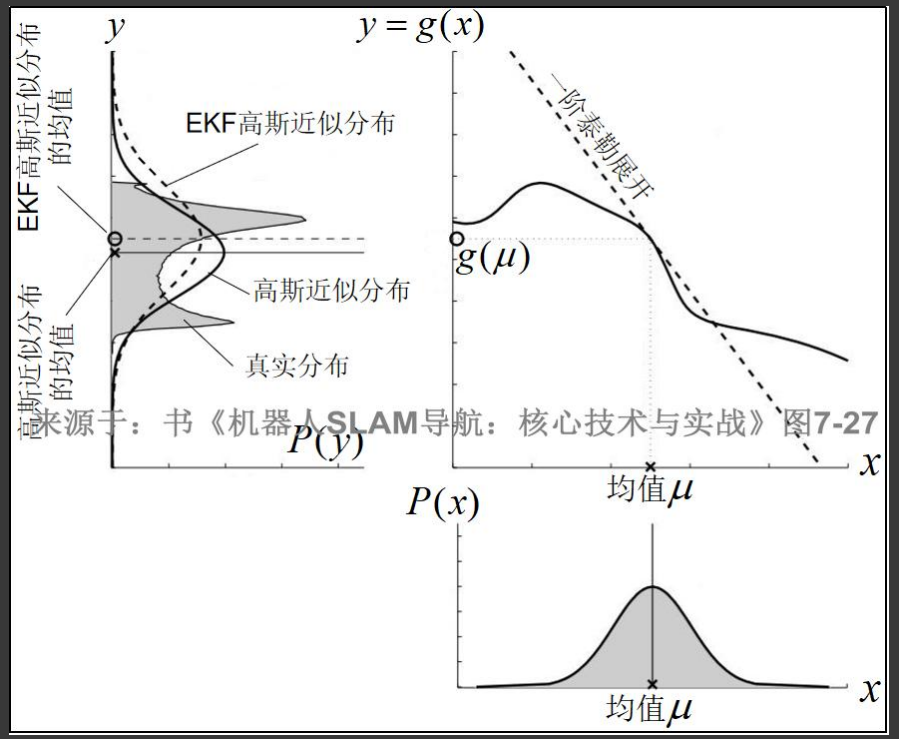
7.4 基于贝叶斯网络的状态估计

高斯分布可以用矩参数（均值和方差）进行表示，那么多维高斯分布为：

$$P(\boldsymbol{x})=\frac{1}{\sqrt{(2\pi)^n\det(\boldsymbol{\Sigma})}}\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$

- 贝叶斯估计
- 参数化实现
- 非参数化实现
  - 卡尔曼滤波
    - 线性卡尔曼滤波（线性高斯系统）
    - 扩展卡尔曼滤波（非线性高斯系统）
    - 无迹卡尔曼滤波（非线性高斯系统）
  - 信息滤波
    - 线性信息滤波（线性高斯系统）
    - 扩展信息滤波（非线性高斯系统）

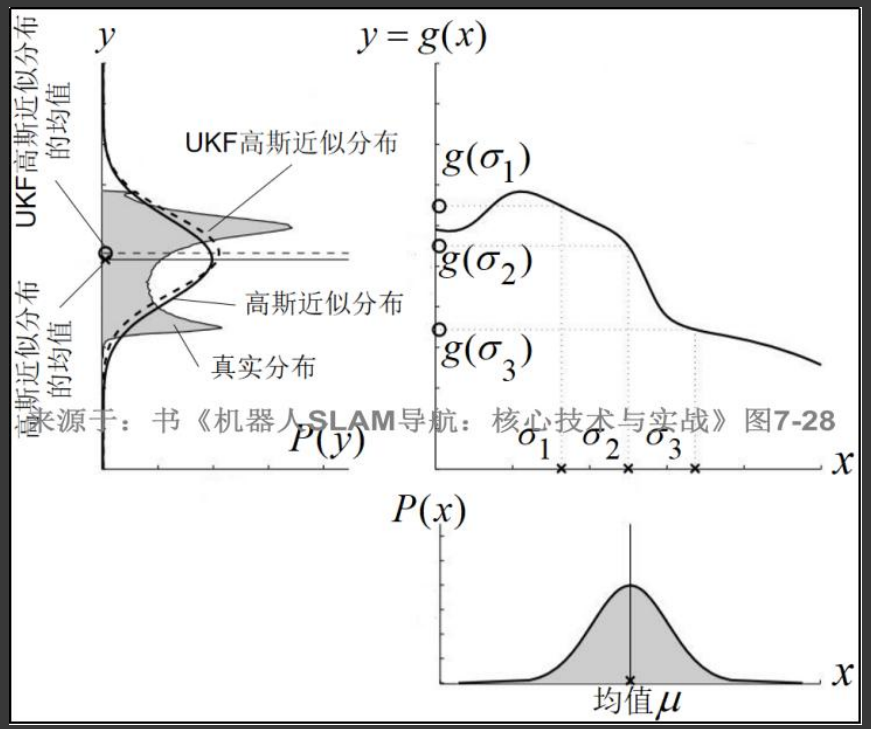
- 如果x的不确定性和函数g的非线性度都小，那么扩展卡尔曼滤波（EKF）中线性化近似效果是好的。
- 但是，当x的不确定性和函数g的非线性度较大时，扩展卡尔曼滤波（EKF）的近似效果就不好了，此时可以用无迹卡尔曼滤波（UKF）来处理。
- 除了EKF和UKF方法，还有很多其他处理非线性系统线性近似的方法，这里就不再过大展开了



来源于：书《机器人SLAM导航：核心技术与实战》图7-27

一阶泰勒展开进行线性化

VS



来源于：书《机器人SLAM导航：核心技术与实战》图7-28

无迹变换进行线性化

7.4 基于贝叶斯网络的状态估计

高斯分布可以用矩参数（均值和方差）进行表示，那么多维高斯分布为：

$$P(\boldsymbol{x})=\frac{1}{\sqrt{(2\pi)^n\det(\boldsymbol{\Sigma})}}\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$

- 贝叶斯估计
- 参数化实现
  - 卡尔曼滤波
    - 线性卡尔曼滤波（线性高斯系统）
    - 扩展卡尔曼滤波（非线性高斯系统）
    - 无迹卡尔曼滤波（非线性高斯系统）
  - 信息滤波
    - 线性信息滤波（线性高斯系统）
    - 扩展信息滤波（非线性高斯系统）
- 非参数化实现

矩参数（均值和方差）可以转换成正则参数（信息矩阵和信息向量）：

$$\begin{cases}\boldsymbol{\Omega}=\boldsymbol{\Sigma}^{-1}\\ \boldsymbol{\xi}=\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\end{cases}$$

那么正则参数形式的多维高斯分布为：

$$\begin{aligned}P(\boldsymbol{x})&=\frac{1}{\sqrt{(2\pi)^n\det(\boldsymbol{\Sigma})}}\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)\\&=\frac{1}{\sqrt{(2\pi)^n\det(\boldsymbol{\Sigma})}}\exp\left(-\frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{x}+\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}-\frac{1}{2}\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\right)\\&=\underbrace{\frac{1}{\sqrt{(2\pi)^n\det(\boldsymbol{\Sigma})}}\exp\left(-\frac{1}{2}\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\right)}_{\text{常数项}}\cdot\exp\left(-\frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\underbrace{\boldsymbol{\Sigma}^{-1}}_{\boldsymbol{\Omega}}\boldsymbol{x}+\boldsymbol{x}^{\mathrm{T}}\underbrace{\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}}_{\boldsymbol{\xi}}\right)\\&=\eta\cdot\exp\left(-\frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\Omega}\boldsymbol{x}+\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\xi}\right)\end{aligned}$$

矩参数和正则参数只是表示高斯分布的2种不同方式，高斯分布的本质是一样的。
因此，线性信息滤波的推导与线性卡尔曼滤波是类似的

$$\begin{cases}①\ \overline{\boldsymbol{\Omega}_k}=(\boldsymbol{A}_k\boldsymbol{\Omega}_{k-1}^{-1}\boldsymbol{A}_k^{\mathrm{T}}+\boldsymbol{R}_k)^{-1}\\ ②\ \overline{\boldsymbol{\xi}_k}=\overline{\boldsymbol{\Omega}_k}(\boldsymbol{A}_k\boldsymbol{\Omega}_{k-1}^{-1}\boldsymbol{\xi}_{k-1}+\boldsymbol{B}_k\boldsymbol{u}_k)\end{cases}\text{运动预测}$$

$$\begin{cases}③\ \boldsymbol{\Omega}_k=\overline{\boldsymbol{\Omega}_k}+\boldsymbol{C}_k^{\mathrm{T}}\boldsymbol{Q}_k^{-1}\boldsymbol{C}_k\\ ④\ \boldsymbol{\xi}_k=\overline{\boldsymbol{\xi}_k}+\boldsymbol{C}_k^{\mathrm{T}}\boldsymbol{Q}_k^{-1}\boldsymbol{z}_k\end{cases}\text{观测更新}$$

线性信息滤波
(IF)

- 可以发现线性卡尔曼滤波和线性信息滤波的性能是对偶的。
- 线性卡尔曼滤波中的运动预测是增量的，而线性信息滤波中的运动预测需要求矩阵逆运算，所以线性卡尔曼滤波在运动预测上计算效率更高。
- 不过，线性卡尔曼滤波中的观测更新需要求矩阵逆运算，而线性信息滤波中的观测更新是增量的，所以线性信息滤波在观测更新上计算效率更高。

7.4 基于贝叶斯网络的状态估计

高斯分布可以用矩参数（均值和方差）进行表示，那么多维高斯分布为：

$$P(\boldsymbol{x})=\frac{1}{\sqrt{(2\pi)^n\det(\boldsymbol{\Sigma})}}\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$

- 贝叶斯估计
- 参数化实现
- 非参数化实现

参数化实现：
- 卡尔曼滤波
  - 线性卡尔曼滤波（线性高斯系统）
  - 扩展卡尔曼滤波（非线性高斯系统）
  - 无迹卡尔曼滤波（非线性高斯系统）
- 信息滤波
  - 线性信息滤波（线性高斯系统）
  - 扩展信息滤波（非线性高斯系统）

矩参数和正则参数只是表示高斯分布的2种不同方式，高斯分布的本质是一样的。
对于非线性高斯系统，扩展信息滤波与扩展卡尔曼滤波是类似的

$$\begin{cases}\boldsymbol{\mu}_{k-1}=\boldsymbol{\Omega}_{k-1}^{-1}\boldsymbol{\xi}_{k-1}\\ \overline{\boldsymbol{\mu}_k}=g(\boldsymbol{\mu}_{k-1},\boldsymbol{u}_k)\\ ①\ \overline{\boldsymbol{\Omega}_k}=(\boldsymbol{G}_k\boldsymbol{\Omega}_{k-1}^{-1}\boldsymbol{G}_k^{\mathrm{T}}+\boldsymbol{R}_k)^{-1}\\ ②\ \overline{\boldsymbol{\xi}_k}=\overline{\boldsymbol{\Omega}_k}\bullet\overline{\boldsymbol{\mu}_k}\end{cases}\text{运动预测}$$

$$\begin{cases}③\ \boldsymbol{\Omega}_k=\overline{\boldsymbol{\Omega}_k}+\boldsymbol{H}_k^{\mathrm{T}}\boldsymbol{Q}_k^{-1}\boldsymbol{H}_k\\ ④\ \boldsymbol{\xi}_k=\overline{\boldsymbol{\xi}_k}+\boldsymbol{H}_k^{\mathrm{T}}\boldsymbol{Q}_k^{-1}(\boldsymbol{z}_k-h(\overline{\boldsymbol{\mu}_k})+\boldsymbol{H}_k\overline{\boldsymbol{\mu}_k})\end{cases}\text{观测更新}$$

扩展信息滤波
（EIF）

对实际非线性系统进行
线性化近似处理

矩参数（均值和方差）可以转换成正则参数（信息矩阵和信息向量）：

$$\begin{cases}\boldsymbol{\Omega}=\boldsymbol{\Sigma}^{-1}\\ \boldsymbol{\xi}=\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\end{cases}$$

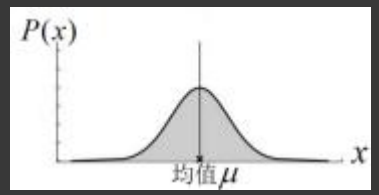
那么正则参数形式的多维高斯分布为：

$$\begin{aligned}P(\boldsymbol{x})&=\frac{1}{\sqrt{(2\pi)^n\det(\boldsymbol{\Sigma})}}\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)\\&=\frac{1}{\sqrt{(2\pi)^n\det(\boldsymbol{\Sigma})}}\exp\left(-\frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{x}+\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}-\frac{1}{2}\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\right)\\&=\underbrace{\frac{1}{\sqrt{(2\pi)^n\det(\boldsymbol{\Sigma})}}\exp\left(-\frac{1}{2}\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\right)}_{\text{常数项}}\cdot\exp\left(-\frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\underbrace{\boldsymbol{\Sigma}^{-1}}_{\boldsymbol{\Omega}}\boldsymbol{x}+\boldsymbol{x}^{\mathrm{T}}\underbrace{\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}}_{\boldsymbol{\xi}}\right)\\&=\eta\cdot\exp\left(-\frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\Omega}\boldsymbol{x}+\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\xi}\right)\end{aligned}$$

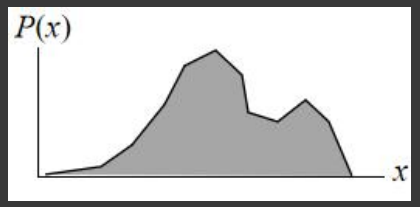
7.4 基于贝叶斯网络的状态估计

- 贝叶斯估计
- 参数化实现
- 非参数化实现

实际问题往往除了是**非线性系统**外，通常还是**非高斯系统**。



高斯分布



非高斯分布

$$P(\boldsymbol{x})=\frac{1}{\sqrt{(2\pi)^n \det(\boldsymbol{\Sigma})}}\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$

（概率密度函数可解析表示）

$$P(\boldsymbol{x})=\alpha_0+\alpha_1 x+\alpha_2 x^2+\ldots+\alpha_n x^n+\ldots$$

（概率密度函数不可解析表示，用无限级数逼近）

概率运动模型：$P(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1},\boldsymbol{u}_k)=$ 可解析

概率观测模型：$P(\boldsymbol{z}_k \mid \boldsymbol{x}_k)=$ 可解析

待估计量的置信度：$bel(\boldsymbol{x}_k)=P(\boldsymbol{x}_k \mid \boldsymbol{z}_{1:k},\boldsymbol{u}_{1:k})=$ 可解析

$P(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1},\boldsymbol{u}_k)=$ 不可解析

$P(\boldsymbol{z}_k \mid \boldsymbol{x}_k)=$ 不可解析

$bel(\boldsymbol{x}_k)=P(\boldsymbol{x}_k \mid \boldsymbol{z}_{1:k},\boldsymbol{u}_{1:k})=$ 不可解析

？

参数化实现
（对概率分布进行参数化表示）

非参数化实现
（对概率分布进行非参数化样本模拟）

7.4 基于贝叶斯网络的状态估计

- 贝叶斯估计
- 参数化实现
- 非参数化实现
  - 直方图滤波
  - 粒子滤波

利用区域划分将待估状态量从连续空间**离散化**到离散状态空间

①离散化的概率运动模型 $P(\boldsymbol{x}_k^i \mid \boldsymbol{x}_{k-1}^j, \boldsymbol{u}_k) \approx \eta_1 \cdot P(c(\boldsymbol{x}_k^i) \mid c(\boldsymbol{x}_{k-1}^j), \boldsymbol{u}_k)$

②离散化的概率观测模型 $P(\boldsymbol{z}_k \mid \boldsymbol{x}_k^i) \approx \eta_2 \cdot P(\boldsymbol{z}_k \mid c(\boldsymbol{x}_k^i))$

递归贝叶斯滤波

$$\begin{cases} \overline{bel(\boldsymbol{x}_k)} = \int P(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}, \boldsymbol{u}_k) \cdot bel(\boldsymbol{x}_{k-1}) d\boldsymbol{x}_{k-1} \\ bel(\boldsymbol{x}_k) = \eta \cdot P(\boldsymbol{z}_k \mid \boldsymbol{x}_k) \overline{bel(\boldsymbol{x}_k)} \end{cases}$$ 运动预测 / 观测更新

离散化

③离散递归贝叶斯滤波

（遍历待估状态的所有离散区域）

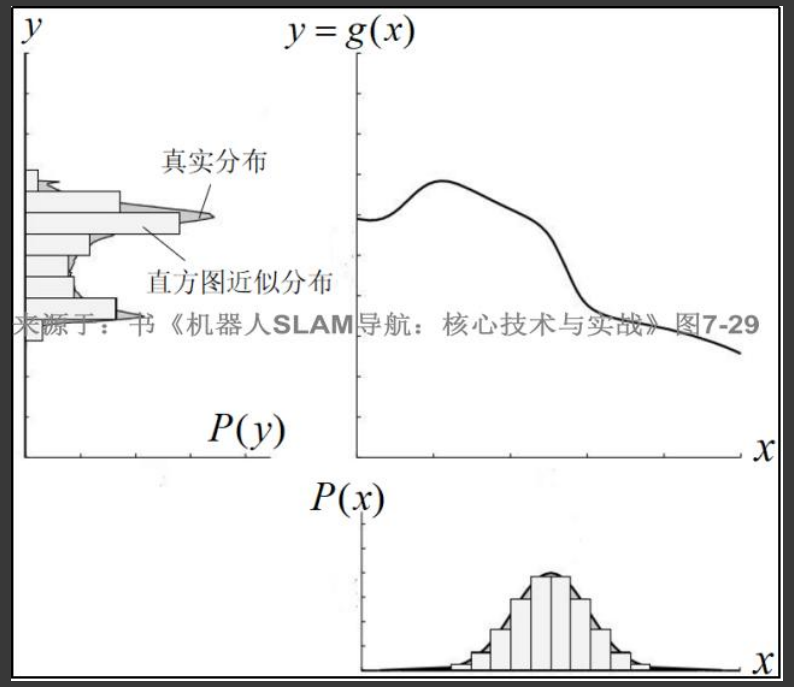
$$\begin{cases} \overline{bel(\boldsymbol{x}_k^i)} = \sum_j P(\boldsymbol{x}_k^i \mid \boldsymbol{x}_{k-1}^j, \boldsymbol{u}_k) \cdot bel(\boldsymbol{x}_{k-1}^j) \\ bel(\boldsymbol{x}_k^i) = \eta \cdot P(\boldsymbol{z}_k \mid \boldsymbol{x}_k^i) \cdot \overline{bel(\boldsymbol{x}_k^i)} \end{cases}$$ 运动预测 / 观测更新

7.4 基于贝叶斯网络的状态估计

- 贝叶斯估计
- 参数化实现
- 非参数化实现
  - 直方图滤波
  - 粒子滤波

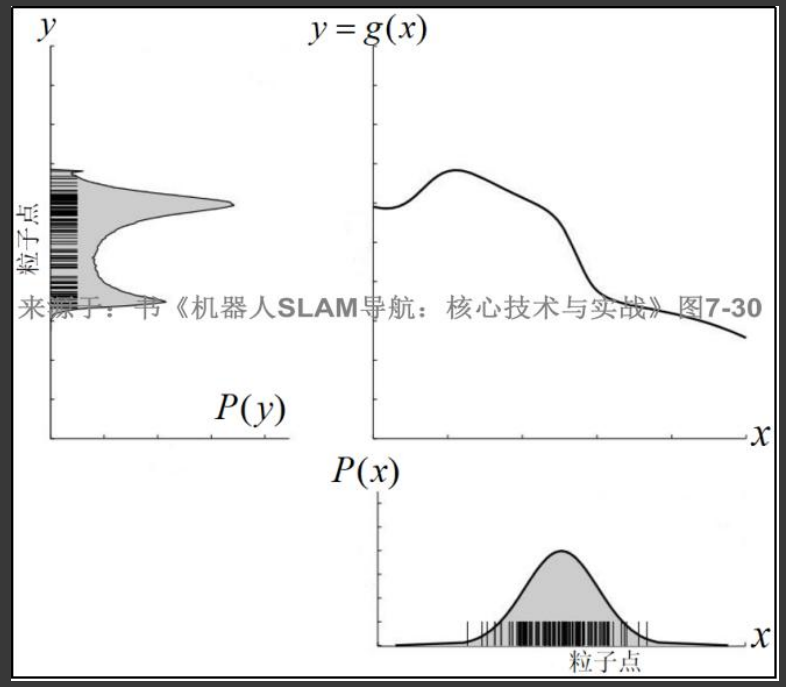
* 直方图滤波，可以认为是用**区域划分**方法对待估量的状态空间进行离散化，用每个划分区域上质心的概率代表整个区域的概率大小

* 粒子滤波，可以认为是用**随机粒子点**对待估量的状态空间进行离散化，用粒子点的稠密度代表该取值空间处的概率大小



来源于：书《机器人SLAM导航：核心技术与实战》图7-29

直方图近似表示概率分布

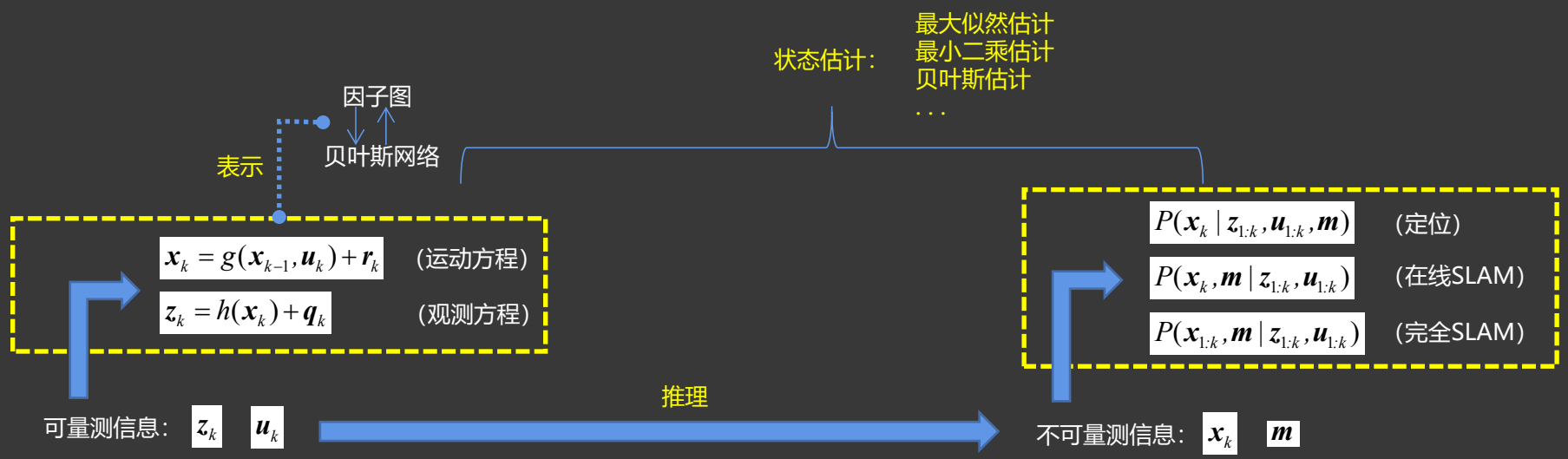


来源于：书《机器人SLAM导航：核心技术与实战》图7-30

粒子近似表示概率分布

内容概要

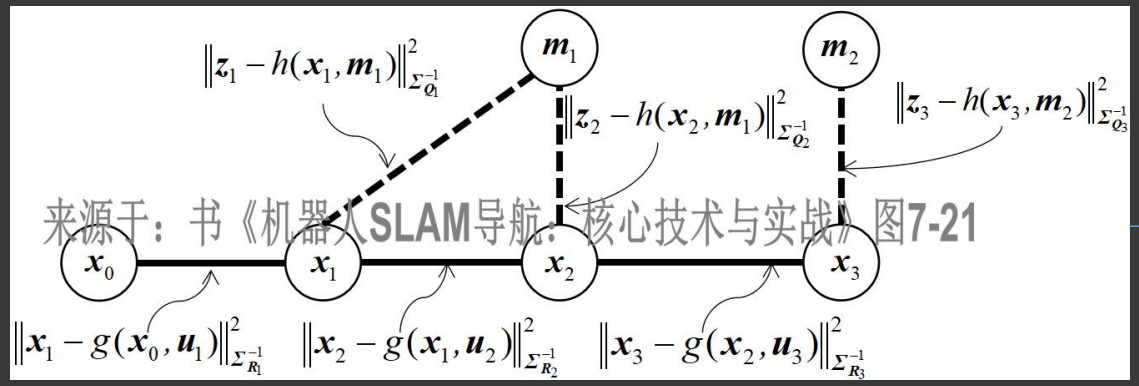
7.5 基于因子图的状态估计



7.5 基于因子图的状态估计

- 非线性最小二乘估计 ➡ 以因子图（表示）、完全SLAM(待估问题)、最小二乘估计（求解方法）这种情况为例来讨论：
- 直接求解方法
- 优化方法
- 各种优化方法对比
- 常用优化工具

最小二乘约束形式因子图表示的SLAM问题



来源于：书《机器人SLAM导航：核心技术与实战》图7-21

最小二乘估计

（实际一般为非线性最小二乘估计）

$$
\begin{aligned}
S_{\mathrm{MAP}} &= \arg\max_{S} \psi(S) \\
&\propto \arg\max_{S}\Big(\sum_{i1} \ln \psi_{i1}(x_k, x_{k-1}) + \sum_{i2} \ln \psi_{i2}(x_k, m_j)\Big) \\
&\propto \arg\max_{S}\Big(-\frac{1}{2}\sum_{i1}(x_k - g(x_{k-1}, u_k))^{\mathrm{T}} \Sigma^{-1}_{R_k}(x_k - g(x_{k-1}, u_k)) \\
&\qquad -\frac{1}{2}\sum_{i2}(z_k - h(x_k, m_j))^{\mathrm{T}} \Sigma^{-1}_{Q_k}(z_k - h(x_k, m_j))\Big) \\
&\propto \arg\min_{S}\Big(\sum_{i1}\|x_k - g(x_{k-1}, u_k)\|^2_{\Sigma^{-1}_{R_k}} + \sum_{i2}\|z_k - h(x_k, m_j)\|^2_{\Sigma^{-1}_{Q_k}}\Big)
\end{aligned}
$$

思考：
这里的非线性是指什么？

非线性最小二乘估计的通用形式

（严格说是，非线性加权最小二乘）

$$\min_{x}\sum_{i}\|y_i - f(x_i)\|^2_{\Sigma^{-1}_i}$$

* 关于公式的具体数学推导，将在第2季课程中详细讨论

7.5 基于因子图的状态估计

- 非线性最小二乘估计
- 直接求解方法
- 优化方法
- 各种优化方法对比
- 常用优化工具

非线性最小二乘问题

$$\min_x \sum_i \|y_i - f(x_i)\|^2_{\Sigma_i^{-1}}$$

泰勒展开进行线性化近似

$$f(x) = \frac{f(x_0)}{0!} + \frac{f'(x_0)}{1!}(x-x_0) + \frac{f''(x_0)}{2!}(x-x_0)^2 + \ldots + \frac{f^{(n)}(x_0)}{n!}(x-x_0)^n + R_n(x)$$

$$\begin{aligned} f(x_i) &\approx f(x_i)|_{x_i=x0_i} + f'(x_i)|_{x_i=x0_i} \bullet (x_i - x0_i) \\ &= f(x0_i) + f'(x0_i) \bullet (x_i - x0_i) \\ &= f(x0_i) + F_i \bullet (x_i - x0_i) \end{aligned}$$

（一般取一阶泰勒展开）

$$\begin{aligned} \min_x \sum_i \|y_i - f(x_i)\|^2_{\Sigma_i^{-1}} &\propto \min_x \sum_i \|y_i - (f(x0_i) + F_i \bullet (x_i - x0_i))\|^2_{\Sigma_i^{-1}} \\ &= \min_x \sum_i \|(y_i - f(x0_i) + F_i \bullet x0_i) - F_i \bullet x_i\|^2_{\Sigma_i^{-1}} \\ &= \min_x \sum_i \left\| \underbrace{\Sigma_i^{-1/2}(y_i - f(x0_i) + F_i \bullet x0_i)}_{b_i} - \underbrace{\Sigma_i^{-1/2} F_i}_{A_i} \bullet x_i \right\|^2 \\ &= \min_x \sum_i \|b_i - A_i \bullet x_i\|^2 \end{aligned}$$

理想情况
最小值可取到0

解线性方程

$$\begin{cases} A_1 \bullet x_1 = b_1 \\ A_2 \bullet x_2 = b_2 \\ \ldots \\ A_i \bullet x_i = b_i \\ \ldots \end{cases}, \text{令} A = \begin{bmatrix} A_1 & 0 & \ldots & 0 & \ldots \\ 0 & A_2 & \ldots & 0 & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & A_i & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \\ \ldots \\ x_i \\ \ldots \end{bmatrix}, b = \begin{bmatrix} b_1 \\ b_2 \\ \ldots \\ b_i \\ \ldots \end{bmatrix} \Rightarrow Ax = b$$

Cholesky分解

$$\begin{aligned} Ax = b &\Leftrightarrow (A^{\mathrm{T}}A)x = A^{\mathrm{T}}b \\ &\Leftrightarrow (R^{\mathrm{T}}R)x = A^{\mathrm{T}}b \\ &\Leftrightarrow R^{\mathrm{T}}(Rx) = A^{\mathrm{T}}b \\ &\Leftrightarrow R^{\mathrm{T}}y = A^{\mathrm{T}}b \end{aligned}$$

其中：$A^{\mathrm{T}}A = R^{\mathrm{T}}R, Rx = y$

QR分解

$$\begin{aligned} Ax = b &\Leftrightarrow (QR)x = b \\ &\Leftrightarrow Q^{-1}QRx = Q^{-1}b \\ &\Leftrightarrow Rx = Q^{-1}b \\ &\Leftrightarrow Rx = Q^{\mathrm{T}}b \end{aligned}$$

其中：
$A = QR$
Q是正交矩阵$(Q^{-1} = Q^{\mathrm{T}})$
R是上三角矩阵

7.5 基于因子图的状态估计

由于实际SLAM问题的非线性最小二乘中，很难找到合适的线性化方法，初值也比较难确定，并且代价函数的误差往往不能最小化到0值，所以上面介绍的直接求解方法很难利用在实际问题中。实际工程中，通常采用优化方法进行迭代求解。

- 非线性最小二乘估计
- 直接求解方法
- 优化方法
- 各种优化方法对比
- 常用优化工具

非线性最小二乘问题

$$\psi(\boldsymbol{x})=\sum_i \|\boldsymbol{y}_i-f(\boldsymbol{x}_i)\|^2_{\boldsymbol{\Sigma}_i^{-1}}$$

实际情况
最小值取不到0
初值难确定
线性化困难

迭代策略

梯度下降算法

$$\Delta\boldsymbol{x}_{\text{grad}}=-\alpha\cdot\nabla\psi(\boldsymbol{x}^{(k)})$$
$$\boldsymbol{x}^{(k+1)}=\boldsymbol{x}^{(k)}+\Delta\boldsymbol{x}_{\text{grad}}$$

最速下降算法

$$\alpha_k=\arg\min_{\alpha\geq 0}\psi(\boldsymbol{x}^{(k)}-\alpha\cdot\nabla\psi(\boldsymbol{x}^{(k)}))$$
$$\Delta\boldsymbol{x}_{\text{SD}}=-\alpha_k\cdot\nabla\psi(\boldsymbol{x}^{(k)})$$
$$\boldsymbol{x}^{(k+1)}=\boldsymbol{x}^{(k)}+\Delta\boldsymbol{x}_{\text{SD}}$$

高斯-牛顿算法

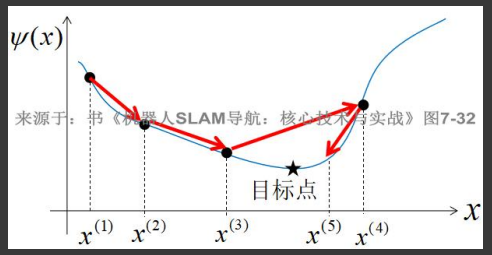
$$\Delta\boldsymbol{x}_{\text{GN}}=-(\boldsymbol{J}_e^{\text{T}}\boldsymbol{J}_e)^{-1}\bullet\boldsymbol{J}_e^{\text{T}}\bullet\boldsymbol{e}$$
$$\boldsymbol{x}^{(k+1)}=\boldsymbol{x}^{(k)}+\Delta\boldsymbol{x}_{\text{GN}}$$

列文伯格-马夸尔特算法

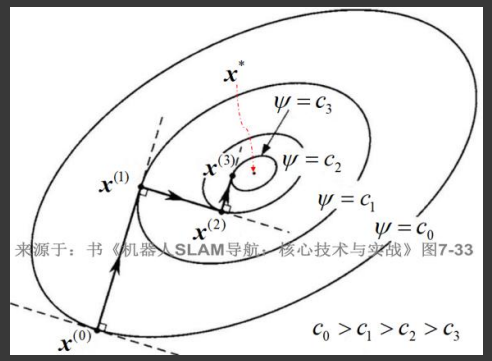
$$\Delta\boldsymbol{x}_{\text{LM}},\mu_{k+1}=LM(\boldsymbol{x}^{(k)},\mu_k)$$
$$\boldsymbol{x}^{(k+1)}=\boldsymbol{x}^{(k)}+\Delta\boldsymbol{x}_{\text{LM}}$$

狗腿算法

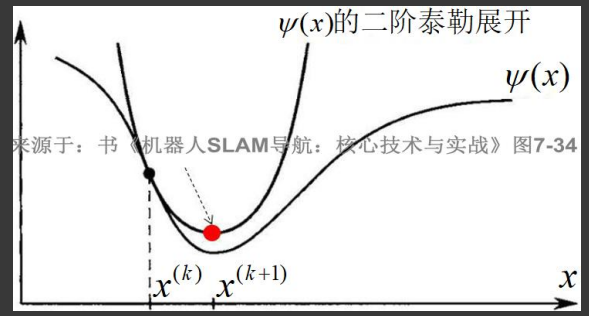
$$\Delta\boldsymbol{x}_{\text{DL}},d_{k+1}=Dogleg(\boldsymbol{x}^{(k)},d_k)$$
$$\boldsymbol{x}^{(k+1)}=\boldsymbol{x}^{(k)}+\Delta\boldsymbol{x}_{\text{DL}}$$



梯度下降



最速下降



牛顿迭代

简化

高斯-牛顿迭代

* 关于公式的具体数学推导，将在第2季课程中详细讨论

7.5 基于因子图的状态估计

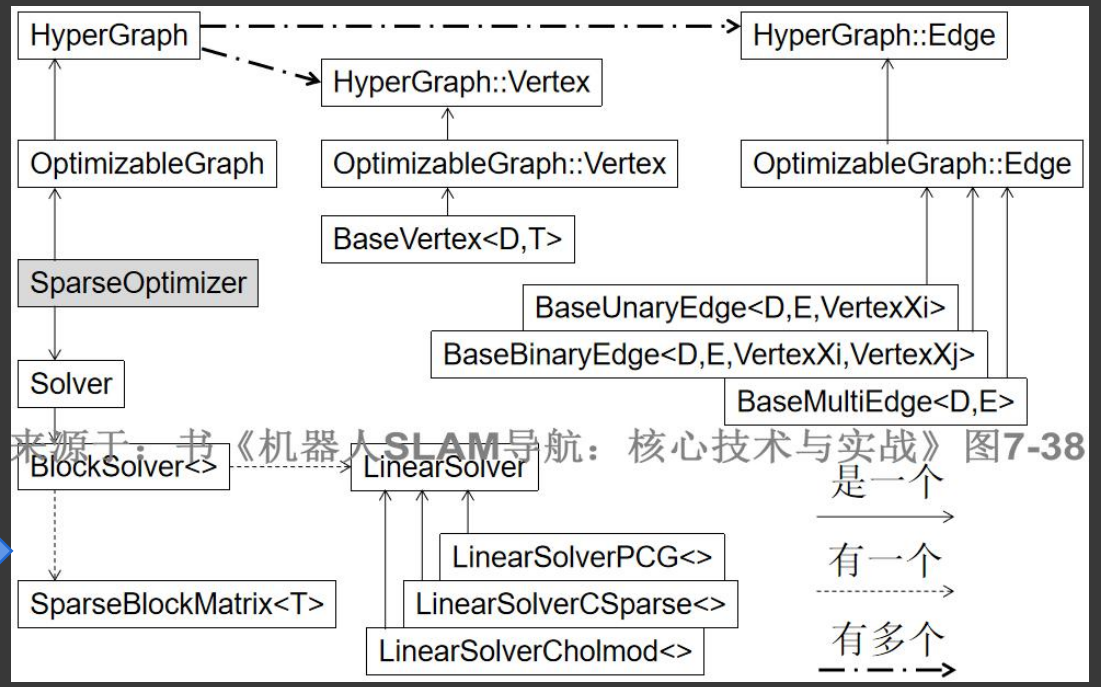
- 非线性最小二乘估计
- 直接求解方法
- 优化方法
- 各种优化方法对比
- 常用优化工具

- 可以发现，每一种方法都是前一种的改进，下降速度越来越好，但是每一步迭代的计算代价越来越高。
- 因此，在实际应用中要结合实际问题合理选择。比如Dogleg算法虽然每步迭代下降效果是最好的，但是每步迭代付出的计算代价也最昂贵，站在整个迭代过程看，迭代效率并不一定比最普通的梯度下降法好。

7.5 基于因子图的状态估计

- 非线性最小二乘估计
- 直接求解方法
- 优化方法
- 各种优化方法对比
- 常用优化工具

常用优化工具：
- 图优化工具g2o（轻量级）
- 非线性优化工具Ceres-Solver（复杂但性能强）
- 增量优化工具GTSAM（精度稍低但支持增量优化）



来源于：书《机器人SLAM导航：核心技术与实战》图7-38

g2o框架的C++类

内容概要

7.6 典型SLAM算法

		在线SLAM系统	完全SLAM系统
贝叶斯网络表示	EKF滤波	EKF-SLAM	
	粒子滤波		Fast-SLAM Gmapping
最小二乘表示	直接法		Graph-SLAM
	优化法		Cartographer LOAM ORB-SLAM
机器学习表示	端到端法	DeepVO	
	改进模块法	NeRF-SLAM NICE-SLAM	CNN-SLAM
	生物启发法	RatSLAM NeuroSLAM	

7.6 典型SLAM算法

- EKF-SLAM
- Fast-SLAM
- Graph-SLAM
- 现今主流SLAM算法

$$P(\underbrace{x_k, m_{1:L}}_{s_k} \mid z_{1:k}, u_{1:k}) \sim N(\mu_k, \Sigma_k)$$

其中，

$$s_k = [x_k, m_1, m_2, ..., m_L]^T$$

$$\mu_k = [\mu_{x_k}, \mu_{m_1}, \mu_{m_2}, ..., \mu_{m_L}]^T$$

$$\Sigma_k = \begin{bmatrix} \Sigma_{x_k} & \Sigma_{x_k m_1} & \cdots & \Sigma_{x_k m_L} \\ \Sigma_{m_1 x_k} & \Sigma_{m_1} & \cdots & \Sigma_{m_1 m_L} \\ \cdots & \cdots & \cdots & \cdots \\ \Sigma_{m_L x_k} & \Sigma_{m_L m_1} & \cdots & \Sigma_{m_L} \end{bmatrix} = \begin{bmatrix} \Sigma_{xx} & \Sigma_{xm} \\ \Sigma_{mx} & \Sigma_{mm} \end{bmatrix}$$

$$s_k = g(s_{k-1}, u_k) + r_k \text{或} \begin{bmatrix} x_k \\ m_1 \\ \cdots \\ m_L \end{bmatrix} = \begin{bmatrix} g'(x_{k-1}, u_k) + r'_k \\ m_1 \\ \cdots \\ m_L \end{bmatrix}$$

$$z_k = h(s_k) + q_k$$

其中 r_k 为运动过程携带的高斯噪声，协方差矩阵记为 $R_k = \begin{bmatrix} R'_k & 0 \\ 0 & 0 \end{bmatrix}$；$q_k$ 为观测过程携带的高斯噪声，协方差矩阵记为 Q_k。

在线SLAM系统求解

$$\begin{cases} ①\overline{\mu_k} = g(\mu_{k-1}, u_k) \\ ②\overline{\Sigma_k} = G_k \Sigma_{k-1} G_k^T + R_k \end{cases} \text{运动预测}$$

$$③K_k = \overline{\Sigma_k} H_k^T (H_k \overline{\Sigma_k} H_k^T + Q_k)^{-1} \text{卡尔曼增益}$$

$$\begin{cases} ④\mu_k = \overline{\mu_k} + K_k(z_k - h(\overline{\mu_k})) \\ ⑤\Sigma_k = (I - K_k H_k)\overline{\Sigma_k} \end{cases} \text{观测更新}$$

卡尔曼滤波

以EKF-SLAM为代表的**在线SLAM系统**，其EKF滤波方法是一种增量求解方法，只需要输入新量测信息就能完成对SLAM状态量的求解，这种增量求解方法具有很大优势。但是EKF-SLAM也有两个明显的缺点，一方面，其将所估计后验分布假设成高斯分布，非线性高斯系统必须被线性化近似成线性高斯系统，这种简单粗暴的**线性化近似**容易引起较大误差；另一方面，状态量的协方差矩阵的复杂度随路标数量是二次方增长，也就是说其**无法建立大规模地图**。为了解决线性化和构建大规模地图的问题，大家开始放弃EKF-SLAM这种在线SLAM增量求解的好处，转向**完全SLAM系统**的研究。

7.6 典型SLAM算法

- EKF-SLAM
- Fast-SLAM
- Graph-SLAM
- 现今主流SLAM算法

完全SLAM系统求解

$$P(x_{1:k}, m_{1:L} \mid z_{1:k}, u_{1:k})$$

$$P(x_{1:k}, m_{1:L} \mid z_{1:k}, u_{1:k}) \propto \underbrace{P(x_{1:k} \mid z_{1:k}, u_{1:k})}_{\text{定位问题（采用}PF\text{实现）}} \bullet \underbrace{P(m_{1:L} \mid x_{1:k}, z_{1:k})}_{\text{建图问题（采用}EKF\text{实现）}}$$

7.6 典型SLAM算法

- EKF-SLAM
- Fast-SLAM
- Graph-SLAM
- 现今主流SLAM算法

完全SLAM系统求解

$$P(x_{1:k}, m_{1:L} \mid z_{1:k}, u_{1:k})$$

$$P(x_{1:k}, m_{1:L} \mid z_{1:k}, u_{1:k}) \propto \min \sum_{i=1}^{m} \left\| e_i \right\|^2$$

直接法解线性方程

最小二乘问题

7.6 典型SLAM算法

- EKF-SLAM
- Fast-SLAM
- Graph-SLAM
- 现今主流SLAM算法

来源于：书《机器人SLAM导航：核心技术与实战》图7-39

第8章：激光SLAM系统

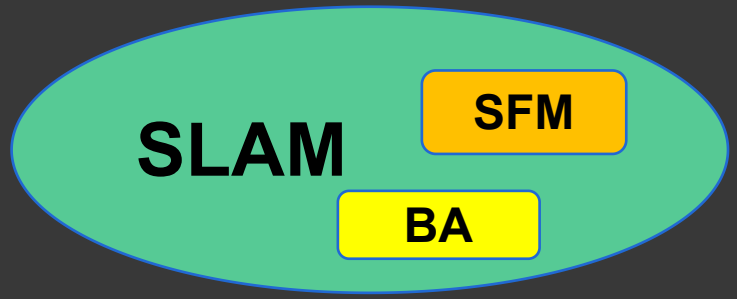
第9章：视觉SLAM系统

第10章：其他SLAM系统

内容概要

7.7 SFM、BA和SLAM比较

名称	描述
SFM	SFM（structure from motion，运动重建）是计算机视觉领域的概念，通过收集到的无序图片进行离线三维重建，可以理解为多个图片的拼接。SFM一般使用在比较大的范围，比如构建一个城市的百度地图。SFM具有两个鲜明的特点，一个特点是输入算法的图片是**无序**的，另一个特点是重建过程**离线**进行对实时性没有要求。
BA	BA（bundle adjustment，光束平差法）是摄影测量领域的概念，通过估计运动相机的位姿来反推场景中路标点的位姿，场景路标点又可以反过来作用于相机位姿估计，这其实就是一个局部优化问题。相机位姿与路标的通过相机观测模型进行约束关联，其实就是用到了重投影误差这个约束。与SFM相比，输入BA算法的图片是需要**有顺序**的。
SLAM	SLAM是移动机器人领域的概念，利用机器人的运动模型和观测模型数据实时估计机器人位姿和路标。与BA相比，SLAM除了有观测数据还有运动数据，SLAM有回环检测能通过全局优化消除累积误差，并且SLAM计算要求尽可能的**实时性**。从这里可以看出，SLAM其实涵盖了SFM和BA所研究的问题，是一个更为复杂和困难的问题。



- 例程源码下载： https://github.com/xiihoo/Books_Robot_SLAM_Navigation
- 课件PPT下载： www.xiihoo.com

敬请关注,长期更新...

下集预告