

♡写在前面的话♡

Kalman Filter

问题描述

线性高斯系统

卡尔曼滤波公式

从极大验后估计MAP的角度解释

从极大后验概率的角度解释状态估计

用Kalman Filter解决状态估计

从贝叶斯估计的角度解释

从增益最优的角度来解释

与最小二乘之间的关系

Extend Kalman Filter

Error State Kalman Filter

Particle Filter

番外👁

经典状态估计中协方差的计算能准确反映机器人状态的完整性吗? ✓

Filter到底是不是直观意义上的过滤? ✓

filter & smoothing & 图优化

不同滤波器方法之间的简单比较

其他非高斯分布的解决方案

IMU ✓

1. IMU类型

2. IMU状态估计原理

3. IMU预计分

概率论 ✓

随机变量

极大似然估计&极大后验估计

Bayesian Estimation or Rules

Reference

♡写在前面的话♡

- 写本文的目的是想自己整理一个对我来说比较易懂的思路，来正确理解-推导KF以及其升级版的原理和公式，在去掉一些网上过长的前导知识铺垫的同时增加对简化的公式罗列的补充。
- 虽然之前也学过Kalman Filter，但是年代就远了之后很多公式就忘记了，也许当时总结的不够到位，理解的不够深刻。现在重新总结一下。
- 在测绘背景下，从状态估计的角度分析KF，这里引用高博知乎的回答：

任何传感器，激光也好，视觉也好，整个SLAM系统也好，要解决的问题只有一个：如何通过数据来估计自身状态。每种传感器的测量模型不一样，它们的精度也不一样。换句话说，状态估计问题，也就是“如何最好地使用传感器数据”。可以说，SLAM是状态估计的一个特例。

- 这里介绍的所有基于滤波的方法都是状态估计中的一种。任何一个需要估计的系统状态都可以写成运动方程和测量方程。其中运动方程即系统前后状态之间的转换关系，而测量方程是系统在某个状态下传感器的观测值。通常来说，运动方程和测量方程在实际过程中都会有噪声，因此如何能够从噪声这么多的观测数据中估算出正确的状态是个难事儿，毕竟世界很复杂。（这里的状态指的是位姿，并不是控制等领域中的那种状态）
- TODO

✓ 问题描述中的举例子

○ KF中另外两种角度解释

✓ EKF

✓ ESKF

Kalman Filter

1 如果将状态变量看作是一个服从[高斯分布] ([https://www.zhihu.com/search?q=高斯分布&search_source=Entity&hybrid_search_source=Entity&hybrid_search_extra={"sourceType":%3A"answer"%2C"sourceId":%3A1267117107}](https://www.zhihu.com/search?q=高斯分布&search_source=Entity&hybrid_search_source=Entity&hybrid_search_extra={)) 的随机变量。就可以用基于KF的方法来完成状态估计这个工作。在**卡尔曼滤波器 (KF)** 的通用算法中, 主要分为两个部分: 预测和更新。其中**预测**过程是说整个状态随时间变化, 依赖前一次的**状态**与**协方差**, 根据**运动模型**预测下一次的**状态**和**协方差** ([只依赖两个状态的原因] (#filter & smoothing & 图优化)); **更新**过程则是根据当前测量和预测的状态和协方差根据**测量方程**校正本次的**状态**和**协方差**。两者迭代往前走, 从而完成系统整个运动过程中的状态估计。

下面这个图很好的说明了KF的流程, 即状态在不断迭代预测-更新的过程。

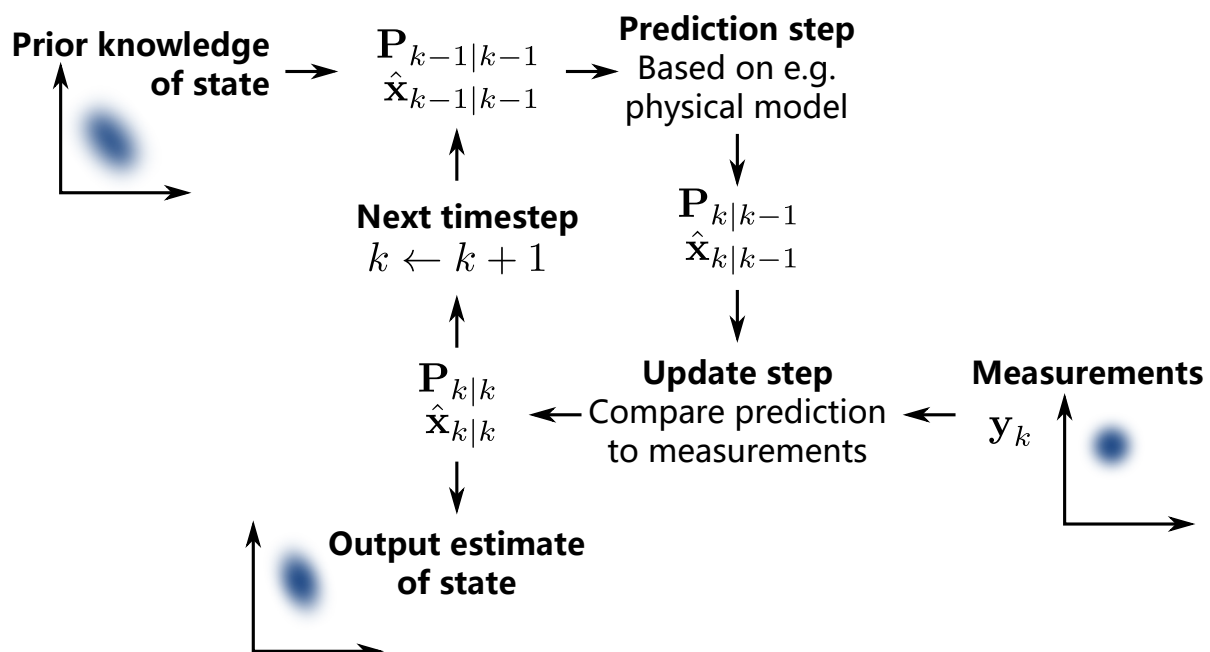


图1.Kalman Filter流程

问题描述

这里使用KF是为了解决状态估计问题, 那么这里将状态估计问题描述为:

机器人 (或某运动载体) 在各个时刻的状态为 $x_1, \dots, x_i, \dots, x_k, x_k \in \mathbb{R}^N$, k 为离散时间下标, 最大值为 K , 状态向量为一个 N 维的向量, 那么按照状态估计的基本条件 (有初值, 运动方程, 观测方程), 其对应的运动方程和观测方程可以抽象地用下面式子来表示:

$$x_k = f(x_{k-1}, u_k, w_k) \quad (1)$$

$$y_k = g(x_k, n_k) \quad (2)$$

- $f(\cdot)$ 为运动方程, 表示从前一刻状态 x_{k-1} 到当前时刻状态 x_k 之间的变换, 描述两个状态之间是如何转换的, 因此会涉及到状态转移方程 (状态转移矩阵)
- $u_k \in \mathbb{R}^N$ 为当前时刻的输入 (可以理解为运动向量这一中间态, 运动向量和运动状态向量之间需要控制矩阵来转换, 不是最终要估计的状态向量)
- $w_k \in \mathbb{R}^N$ 为输入数据的噪声

- $g(\cdot)$ 为观测方程，表示当前状态 x_k 与此时刻系统传感器观测的数据 $y_k \in \mathbb{R}^M$ 之间的转换，描述当前状态 x_k 与传感器观测到的 y_k 之间的关系，即观测方程（观测矩阵）是如何构建的
- $n_k \in \mathbb{R}^M$ 为系统传感器的观测噪声，该误差由硬件决定,维度为 M ，不一定和状态向量的维度相同
- 除了 u_k ，其他变量都是随机变量，噪声和初始状态一般假设是无关的，且各个时刻也都互不相关

带入具象化的栗子再描述一遍状态估计问题，方便理解其概念：

1. 当状态只关心位置时（或着说只能观测到坐标），状态 x 表示的就是坐标，一维空间： x ，二维空间： xy ，三维空间： xyz ，没有再高维了。

- 对应的运动方程可能是如下表示， Δx_k 就是运动向量，它是相邻两个状态中间过渡的量，

- $$x_k = x_{k-1} + \Delta x_k + w_k \quad (3)$$

- 对应的观测方程可能是如下，e.g.可能是一个LiDAR,观测到的量时角度 θ 和距离 r ,那么 y_k 就是这里的 θ 和 r 。 L_k 是它观测到的对象的位置，不是他的观测量。

- $$\begin{bmatrix} r \\ \theta \end{bmatrix}_k = \begin{bmatrix} \sqrt{\|x_k - L_k\|_2} \\ \tan^{-1} \frac{L_{k,y} - x_{k,y}}{L_{k,x} - x_{k,x}} \end{bmatrix} + n_k, \quad \text{where } L_k = [L_{k,x}, L_{k,y}] \quad (4)$$

2. 如果系统状态更丰富一些， x 表示的就除了位置 xyz 还有可能有速度 v ，加速度 a ，角速度 ω ，重力加速度 g 等. [here](#)。因此传感器除了LiDAR还要有能测量旋转的IMU。

- 对应的运动方程可能是：用5表示匀速运动，或者用6表示一般的动力学方程，符合牛顿第二定律。注意位置的一阶导数和二阶导数分别为速度和加速度 $\dot{x} = v, \ddot{x} = a$ 。 u_k 就是速度和加速度。

$$x_k = x_{k-1} + v \cdot \Delta t \quad (5)$$

- $$x_k = x_{k-1} + v \cdot \Delta t + \frac{1}{2} \cdot a \cdot \Delta t^2 \quad (6)$$

- 对应的观测方程也可能是如下表示，如果是相机，就是像素的位置或者逆深度，如果是激光雷达，就还和上述的一样，观测方程和对应传感器对应。

- $$p(u, v) = \frac{1}{Z_w} K(R \cdot P_w + t), \text{ 三维投影到二维的成像方程} \quad (7)$$

3. 不同的系统会有不同的运动模式，也会搭载不同的传感器，因此运动方程和观测方程会随之改动，因此原理推导时还得从抽象的概念开始。

这就是最一般的状态估计问题。根据 f, g 是否线性，分为线性/非线性系统。同时，对于噪声 w, n ，根据它们是否服从高斯分布，分为高斯/非高斯噪声系统。最一般的，也是最困难的问题，是非线性-非高斯(NLNG, Nonlinear-Non Gaussian)的状态估计。因此KF-EKF-PF分别对应线性高斯系统，非线性高斯系统和非线性非高斯系统。

线性高斯系统

因为高斯分布经过线性变换之后仍为高斯分布。而对于一个高斯分布，只要计算出它的一阶(期望)和二阶矩(平方的期望)，就可以描述它(高斯分布只有两个参数 $\mu, \Sigma = \delta^2$)，假设线性系统形式如下：

$$x_k = A_{k-1}x_{k-1} + v_k + w_k, k > 1, A_k \in \mathbb{R}^{N \times N} \text{为状态转移矩阵} \quad (8)$$

$$y_k = C_k x_k + n_k, k > 0, C_k \in \mathbb{R}^{M \times N} \text{为观测矩阵} \quad (9)$$

$$w_k \sim N(0, Q_k), Q_k \text{为运动模型噪声的协方差矩阵} \quad (10)$$

$$n_k \sim N(0, R_k), R_k \text{为观测模型噪声的协方差矩阵} \quad (11)$$

- 动力学模型是线性的，观测模型也是线性的（矩阵线性变换），状态噪声和观测噪声均为零均值的白噪声，这两种噪声，以及噪声与状态之间均互不相关。
- 必须要有一个初始状态为 $x_0 \sim \mathcal{N}(\check{x}_0, \check{P}_0)$ 。

- 其实应该将 $x_k = A_{k-1}x_{k-1} + v_k + w_k$ 写成 $x_k = A_{k-1}x_{k-1} + B_k u_k + w_k, B_k \in \mathbb{R}^{M \times N}$ 这种更一般的形式，不过前者对于推导KF已经够用了，因此这里就简化一下更方便。

卡尔曼滤波公式

直接上来就给出通用的KF的公式，明白整个文章最终要得到的是什么东西。🕒有的放矢🕒，KF的核心其实就是下面所示的5个公式，实现k时刻系统状态的估计。

$$\tilde{x}_k = A_{k-1}\hat{x}_{k-1} + v_k, \text{ 状态预测} \quad (12)$$

$$\tilde{P}_k = A_{k-1}\hat{P}_{k-1}A_{k-1}^T + Q_k, \text{ 方差预测} \quad (13)$$

$$K_k = \tilde{P}_k C_k^T (C_k \tilde{P}_k C_k^T + R_k)^{-1}, \text{ 卡尔曼增益} \quad (14)$$

$$\hat{x}_k = \tilde{x}_k + K_k (y_k - C_k \tilde{x}_k), \text{ 状态更新} \quad (15)$$

$$\hat{P}_k = (I - K_k C_k) \tilde{P}_k, \text{ 方差更新} \quad (16)$$

- 这里给出的公式尽量统一，主要是向STATE ESTIMATION FOR ROBOTICS这本书中做统一。其他参考资料的都差不多。
- \tilde{x}_k 表示k时刻状态的预测（又可理解为先验）， \hat{x}_k 表示k时刻状态的更新（又可理解为后验）， \tilde{P}_k, \hat{P}_k 同理。
- 从预测模型中可以看出，在没有额外观测时，系统误差是在不断增大的，因为每一个运动状态都有不确定性，不断累加就会增大其误差。为什么13是相加，可以从运动模型的方差计算公式入手理解，按照高斯分布的线性变换法则，方差就是相加的。
- 注意15中 $y_k - C_k \tilde{x}_k$ 就是整个更新过程中的更新量，是有实际物理意义的，表达的是观测值和期望预测值之间的差， K_k 可以看成这部分更新量的权重，当观测模型误差大时， K_k 会变小，使得系统更偏向运动模型预测得到的状态估值，如果观测模型误差小， K_k 会变大，系统更偏向于观测得到的值，会对原有的运动模型得到的状态估值给一个较大的修正量。

从极大后验估计MAP的角度解释

从极大后验概率的角度解释状态估计

既然要从MAP的角度来解释KF，那就要先从贝叶斯公式来描述一下状态估计问题（状态估计的通用形式之前已经介绍过了，这里更多从概率角度进行解释）。不考虑是什么系统，不考虑服从什么分布，状态估计就是要从给定的先验信息和所有时刻的运动v和观测y中推测出所有时刻的最优状态 \hat{x} ，那么状态估计就变成如下公式17所示：

$$\hat{x} = \arg \max_x p(x | v, y), \text{ MAP的目标} \quad (17)$$

$$= \arg \max_x \frac{p(y | x, v)p(x, v)}{p(y, v)}, \text{ 贝叶斯公式展开} \quad (18)$$

$$= \arg \max_x \frac{p(y | x, v)p(x | v)p(v)}{p(y | v)p(v)}, \text{ 进一步展开然后约掉} \quad (19)$$

$$= \arg \max_x \frac{p(y | x, v)p(x | v)}{p(y | v)}, \text{ 分母和} x \text{ 无关可视为常数，当} x \text{ 已知时，} y \text{ 与} v \text{ 无关} \quad (20)$$

$$= \arg \max_x p(y | x)p(x | v), \text{ 最终得到此式} \quad (21)$$

其中：

$$x = x_{0:K} = (x_0, \dots, x_K) \quad (22)$$

$$v = (\tilde{x}_0, v_{1:K}) = (\tilde{x}_0, v_1, \dots, v_K) \quad (23)$$

$$y = y_{0:K} = (y_0, \dots, y_K) \quad (24)$$

- 粗体和细体有时候分的不是那么严谨，容易造成理解困难。

根据线性高斯系统中的假设，对于所有时刻 $k = 0, \dots, K$ ，所有的噪声 ω_k, n_k 之间是无关的，因此可以对式21中的 $p(\mathbf{y} | \mathbf{x})$ 进行因子分解转换为所有条件概率相乘的形式，即变成：

$$p(\mathbf{y} | \mathbf{x}) = \prod_{k=0}^K p(y_k | x_k) \quad (25)$$

同时将式21中的 $p(\mathbf{x} | \mathbf{v})$ 分解为：

$$p(\mathbf{x} | \mathbf{v}) = p(x_0 | \check{x}_0) \prod_{k=1}^K p(x_k | x_{k-1}, v_k), \text{ 初始状态提出来方便后面连乘} \quad (26)$$

根据多元高斯向量的条件PDF以及条件高斯PDF的公式，可以将 $p(x_0 | \check{x}_0), p(x_k | x_{k-1}, v_k), p(y_k | x_k)$ 具体展开为如下所示：

$$p(x_0 | \check{x}_0) = \frac{1}{\sqrt{(2\pi)^N \det \check{\mathbf{P}}_0}} \times \exp \left(-\frac{1}{2} (x_0 - \check{x}_0)^T \check{\mathbf{P}}_0^{-1} (x_0 - \check{x}_0) \right) \quad (27)$$

$$p(x_k | x_{k-1}, v_k) = \frac{1}{\sqrt{(2\pi)^N \det \mathbf{Q}_k}} \quad (28)$$

$$\times \exp \left(-\frac{1}{2} (x_k - \mathbf{A}_{k-1}x_{k-1} - v_k)^T \mathbf{Q}_k^{-1} (x_k - \mathbf{A}_{k-1}x_{k-1} - v_k) \right) \quad (29)$$

$$p(y_k | x_k) = \frac{1}{\sqrt{(2\pi)^M \det \mathbf{R}_k}} \exp \left(-\frac{1}{2} (y_k - \mathbf{C}_k x_k)^T \mathbf{R}_k^{-1} (y_k - \mathbf{C}_k x_k) \right) \quad (30)$$

把上面这三个代入到 $\arg \max_{\mathbf{x}} p(\mathbf{y} | \mathbf{x}) p(\mathbf{x} | \mathbf{v})$ 21,然后两边取对数，将主要关注点从指数位置下沉到正常位置（即理解为求解概率最大变成求解目标函数最小），然后去掉前面的一些和状态无关的系数因子（机器人中的状态估计P35-36），就可以得到如下所示的最小化的目标函数。

$$\ln(p(y | x)p(x | v)) = \ln p(x_0 | \check{x}_0) + \sum_{k=1}^K \ln p(x_k | x_{k-1}, v_k) + \sum_{k=0}^K \ln p(y_k | x_k) \quad (31)$$

把具体的高斯展开结果带入上述等式，同时定义几个变量来简化目标函数，得到34：

$$J_{v,k}(x) = \begin{cases} \frac{1}{2} (x_0 - \check{x}_0)^T \check{\mathbf{P}}_0^{-1} (x_0 - \check{x}_0), & k = 0 \\ \frac{1}{2} (x_k - \mathbf{A}_{k-1}x_{k-1} - v_k)^T \mathbf{Q}_k^{-1} (x_k - \mathbf{A}_{k-1}x_{k-1} - v_k), & k = 1 \dots K \end{cases} \quad (32)$$

$$J_{y,k}(x) = \frac{1}{2} (y_k - \mathbf{C}_k x_k)^T \mathbf{R}_k^{-1} (y_k - \mathbf{C}_k x_k), \quad k = 0 \dots K \quad (33)$$

$$J(\mathbf{x}) = \sum_{k=0}^K (J_{v,k}(x) + J_{y,k}(x)), \text{ 最大化后验概率就等价于最小化该目标函数} \quad (34)$$

即MAP的问题就转换为了目标函数极值求解问题：

$$\hat{\mathbf{x}} = \arg \max_{\mathbf{x}} p(\mathbf{x} | \mathbf{v}, \mathbf{y}) \rightarrow \hat{\mathbf{x}} = \arg \min_{\mathbf{x}} J(\mathbf{x}) \quad (35)$$

把初始时刻状态 \check{x}_0 ,输入运动变量 \mathbf{v} ,和观测值 \mathbf{y} 写成列向量，状态量 \mathbf{x} 同理，将其余系数写成矩阵形式如下：

- 本节后面开始就不分粗细了，是在扣不动了...
- 空着的部分就是0。

$$\begin{bmatrix} \check{x}_0 \\ v_1 \\ \vdots \end{bmatrix} \quad \mathbf{v}$$

$$z = \begin{bmatrix} \vdots \\ v_K \\ y_0 \\ \vdots \\ y_K \end{bmatrix}, x = \begin{bmatrix} x_0 \\ \vdots \\ x_K \end{bmatrix}, \text{MAP即有了观测} z \text{来估算状态} x \quad (36)$$

$$\mathbf{H} = \begin{bmatrix} 1 & & & & \\ -\mathbf{A}_0 & 1 & & & \\ & \ddots & \ddots & & \\ & & -\mathbf{A}_{K-1} & 1 & \\ \hline \mathbf{C}_0 & & & & \\ & \mathbf{C}_1 & & & \\ & & \ddots & & \\ & & & \mathbf{C}_K \end{bmatrix}, \text{H矩阵由运动方程和观测方程组合而成} \quad (37)$$

$$\mathbf{W} = \left[\begin{array}{c|c} \begin{bmatrix} \check{\mathbf{P}}_0 & & \\ & \mathbf{Q}_1 & \\ & \ddots & \\ & & \mathbf{Q}_K \end{bmatrix} & \mathbf{0} \\ \hline \mathbf{0} & \begin{bmatrix} \mathbf{R}_0 & & \\ & \mathbf{R}_1 & \\ & \ddots & \\ & & \mathbf{R}_K \end{bmatrix} \end{array} \right], \text{W为对应的随机变量的方差} \quad (38)$$

整理一下，矩阵形式的目标函数就是：

$$J(\mathbf{x}) = \frac{1}{2}(\mathbf{z} - \mathbf{H}\mathbf{x})^T \mathbf{W}^{-1}(\mathbf{z} - \mathbf{H}\mathbf{x}) \quad (39)$$

对应的概率密度函数为：式40， η 为归一化因子，即之前省略掉的系统因子：

$$p(\mathbf{z} | \mathbf{x}) = \eta \exp \left(-\frac{1}{2}(\mathbf{z} - \mathbf{H}\mathbf{x})^T \mathbf{W}^{-1}(\mathbf{z} - \mathbf{H}\mathbf{x}) \right) \quad (40)$$

显然这个关于 \mathbf{x} 的二次型问题可以用求导的方式得到其解析解（也是最优解）：

$$\left. \frac{\partial J(\mathbf{x})}{\partial \mathbf{x}^T} \right|_{\hat{\mathbf{x}}} = -\mathbf{H}^T \mathbf{W}^{-1}(\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}) = \mathbf{0} \quad (41)$$

$$\Rightarrow (\mathbf{H}^T \mathbf{W}^{-1} \mathbf{H}) \hat{\mathbf{x}} = \mathbf{H}^T \mathbf{W}^{-1} \mathbf{z} \quad (42)$$

用Kalman Filter解决状态估计

接下来步入正题Kalman Filter的角度解释，首先看一张图，以图解的方式从MAP的角度阐述KF过程,如下：

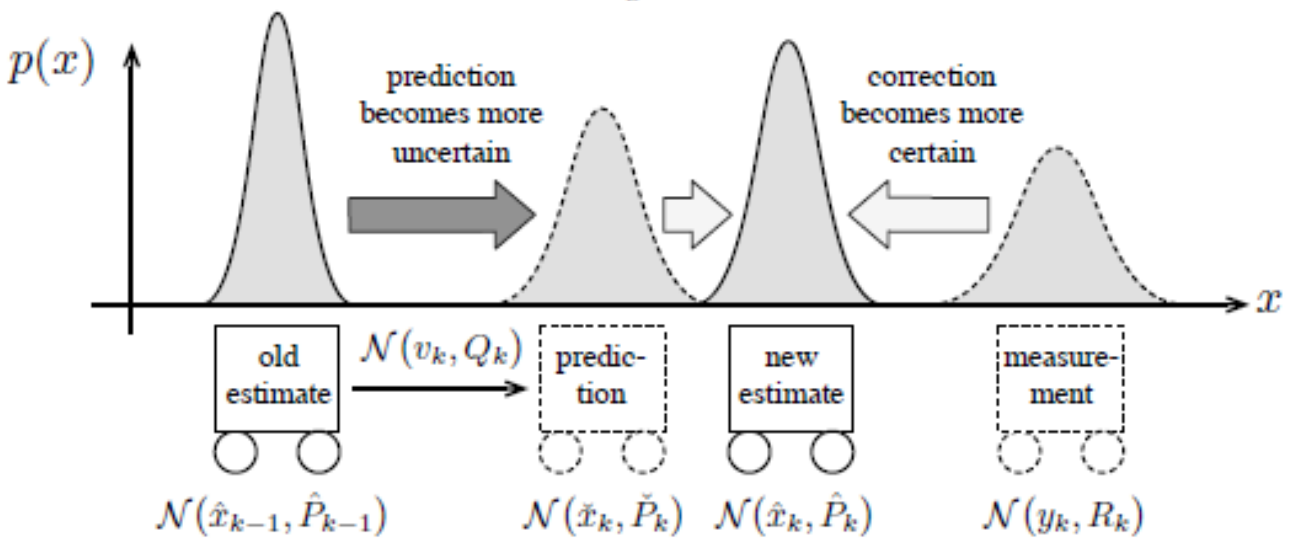


图2.从map解释KF

首先做一些假设。假设目前已经拿到 $k-1$ 时刻系统的状态估计值和对应的方差 $\hat{x}_{k-1}, \hat{P}_{k-1}$ ，这两个值是从系统一开始的0时刻不断推导得到的，同时也有了 k 时刻的运动向量 v_k 和观测 y_k ，状态估计的目的是为了估计得到 k 时刻的系统状态估值 \hat{x}_k, \hat{P}_k 。这里需要知道的是实际上状态是一直累加变化的，在线性高斯系统中完全符合马尔科夫性，因此基于马尔科夫假设，只用前一时刻的状态和方差，在当前观测和运动的基础上，估计当前时刻的状态（这就是KF的思想）。于是状态估计问题就可以看成是如下式子所示：（这段话的意思可以用式子之后的图来表示）

$$\{\hat{x}_{k-1}, \hat{P}_{k-1}, v_k, y_k\} \mapsto \{\hat{x}_k, \hat{P}_k\} \quad (43)$$

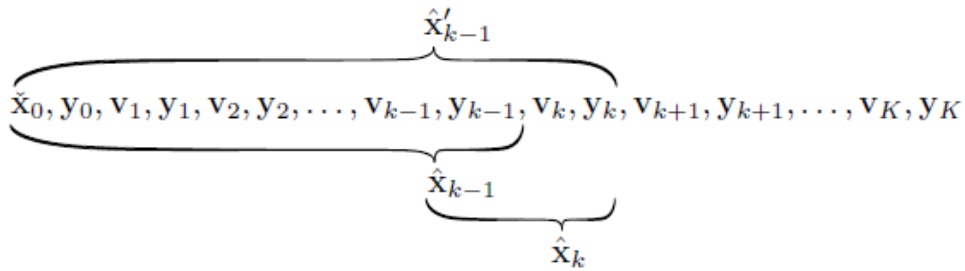


图3.递归滤波将过去所有的数据融合成一次估计

那么KF只用前一时刻的状态 \hat{x}_{k-1}' （它的意思是前一时刻的状态其实是由0-k时刻的全部状态所估计出来的，如图3所示。只是KF都不看 $k-1$ 时刻之前的状态了，同时KF无法预知未来，因此带有未来信息的状态 \hat{x}_{k-1}' 将会被消掉，作为一个递推算法，这是不合理的，因此为了和 \hat{x}_{k-1} 作区分多加了一个变量，可不用计较它的形式），估计当前时刻的状态 \hat{x}_k ，则状态向量 $\hat{x} = [\hat{x}_{k-1}', \hat{x}_k]^T$ 只有两维了，同时对应的矩阵形式中36, 37, 38也只剩相关的系数和权重了，如下所示 $z = H\hat{x}$ ：

$$z = \begin{bmatrix} \hat{x}_{k-1}' \\ v_k \\ y_k \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 0 \\ -A_{k-1} & 1 \\ 0 & C_k \end{bmatrix}, \quad W = \begin{bmatrix} \hat{P}_{k-1} & 0 \\ 0 & Q_k \\ 0 & 0 & R_k \end{bmatrix} \quad (44)$$

带入42那么此时MAP下状态估计问题的最优解就变成了：

$$\begin{bmatrix} \hat{P}_{k-1}^{-1} + A_{k-1}^T Q_k^{-1} A_{k-1} & -A_{k-1}^T Q_k^{-1} \\ -Q_k^{-1} A_{k-1} & Q_k^{-1} + C_k^T R_k^{-1} C_k \end{bmatrix} \begin{bmatrix} \hat{x}_{k-1}' \\ \hat{x}_k \end{bmatrix} = \begin{bmatrix} \hat{P}_{k-1}^{-1} \hat{x}_{k-1}' - A_{k-1}^T Q_k^{-1} v_k \\ Q_k^{-1} v_k + C_k^T R_k^{-1} y_k \end{bmatrix} \quad (45)$$

重点关注的是 \hat{x}_k ，因此可以采用边缘化（舒尔补）的方式将 \hat{x}_{k-1}' 从矩阵中消掉，进而只保留下包含 \hat{x}_k 的等式，消除的方式就是想方设法让要消除的变量对应的系数变成0，即45左乘如下矩阵：

$$\begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{Q}_k^{-1} \mathbf{A}_{k-1} \left(\hat{\mathbf{P}}_{k-1}^{-1} + \mathbf{A}_{k-1}^T \mathbf{Q}_k^{-1} \mathbf{A}_{k-1} \right)^{-1} & 1 \end{bmatrix} \quad (46)$$

45 就变成了如下的形式，从中可以得到 $\hat{\mathbf{x}}_k$ 的最优解：

$$\begin{aligned} & \begin{bmatrix} \hat{\mathbf{P}}_{k-1}^{-1} + \mathbf{A}_{k-1}^T \mathbf{Q}_k^{-1} \mathbf{A}_{k-1} & -\mathbf{A}_{k-1}^T \mathbf{Q}_k^{-1} \\ \mathbf{0} & \mathbf{Q}_k^{-1} - \mathbf{Q}_k^{-1} \mathbf{A}_{k-1} \left(\hat{\mathbf{P}}_{k-1}^{-1} + \mathbf{A}_{k-1}^T \mathbf{Q}_k^{-1} \mathbf{A}_{k-1} \right)^{-1} \\ & \times \mathbf{A}_{k-1}^T \mathbf{Q}_k^{-1} + \mathbf{C}_k^T \mathbf{R}_k^{-1} \mathbf{C}_k \end{bmatrix} \begin{bmatrix} \hat{\mathbf{x}}'_{k-1} \\ \hat{\mathbf{x}}_k \end{bmatrix} \\ & = \begin{bmatrix} \hat{\mathbf{P}}_{k-1}^{-1} \hat{\mathbf{x}}_{k-1} - \mathbf{A}_{k-1}^T \mathbf{Q}_k^{-1} \mathbf{v}_k \\ \mathbf{Q}_k^{-1} \mathbf{A}_{k-1} \left(\hat{\mathbf{P}}_{k-1}^{-1} + \mathbf{A}_{k-1}^T \mathbf{Q}_k^{-1} \mathbf{A}_{k-1} \right)^{-1} \left(\hat{\mathbf{P}}_{k-1}^{-1} \hat{\mathbf{x}}_{k-1} - \mathbf{A}_{k-1}^T \mathbf{Q}_k^{-1} \mathbf{v}_k \right) \\ + \mathbf{Q}_k^{-1} \mathbf{v}_k + \mathbf{C}_k^T \mathbf{R}_k^{-1} \mathbf{y}_k \end{bmatrix}. \end{aligned} \quad (47)$$

进一步化简（不看 $\hat{\mathbf{x}}'_{k-1}$ 的项）得：

$$\underbrace{\left(\mathbf{Q}_k^{-1} - \mathbf{Q}_k^{-1} \mathbf{A}_{k-1} \left(\hat{\mathbf{P}}_{k-1}^{-1} + \mathbf{A}_{k-1}^T \mathbf{Q}_k^{-1} \mathbf{A}_{k-1} \right)^{-1} \mathbf{A}_{k-1}^T \mathbf{Q}_k^{-1} + \mathbf{C}_k^T \mathbf{R}_k^{-1} \mathbf{C}_k \right)}_{\left(\mathbf{Q}_k + \mathbf{A}_{k-1} \hat{\mathbf{P}}_{k-1} \mathbf{A}_{k-1}^T \right)^{-1} \text{ by (SMW formula)}} \hat{\mathbf{x}}_k \quad (48)$$

$$= \mathbf{Q}_k^{-1} \mathbf{A}_{k-1} \left(\hat{\mathbf{P}}_{k-1}^{-1} + \mathbf{A}_{k-1}^T \mathbf{Q}_k^{-1} \mathbf{A}_{k-1} \right)^{-1} \left(\hat{\mathbf{P}}_{k-1}^{-1} \hat{\mathbf{x}}_{k-1} - \mathbf{A}_{k-1}^T \mathbf{Q}_k^{-1} \mathbf{v}_k \right) \quad (49)$$

$$+ \mathbf{Q}_k^{-1} \mathbf{v}_k + \mathbf{C}_k^T \mathbf{R}_k^{-1} \mathbf{y}_k \quad (50)$$

新定义一些变量 $\check{\mathbf{P}}_k$ (其实就是随机变量线性变化后方差的变换公式), $\hat{\mathbf{P}}_k$ 简化推导过程：

$$\check{\mathbf{P}}_k = \mathbf{Q}_k + \mathbf{A}_{k-1} \hat{\mathbf{P}}_{k-1} \mathbf{A}_{k-1}^T \quad (51)$$

$$\hat{\mathbf{P}}_k = \left(\check{\mathbf{P}}_k^{-1} + \mathbf{C}_k^T \mathbf{R}_k^{-1} \mathbf{C}_k \right)^{-1} \quad (52)$$

将 $\check{\mathbf{P}}_k, \hat{\mathbf{P}}_k$ 带入 48 进而可以得出（同时令 $\check{\mathbf{x}}_k$ 为状态的预测值）：

$$\hat{\mathbf{P}}_k^{-1} \hat{\mathbf{x}}_k = \mathbf{Q}_k^{-1} \mathbf{A}_{k-1} \left(\hat{\mathbf{P}}_{k-1}^{-1} + \mathbf{A}_{k-1}^T \mathbf{Q}_k^{-1} \mathbf{A}_{k-1} \right)^{-1} \quad (53)$$

$$\times \left(\hat{\mathbf{P}}_{k-1}^{-1} \hat{\mathbf{x}}_{k-1} - \mathbf{A}_{k-1}^T \mathbf{Q}_k^{-1} \mathbf{v}_k \right) + \mathbf{Q}_k^{-1} \mathbf{v}_k + \mathbf{C}_k^T \mathbf{R}_k^{-1} \mathbf{y}_k \quad (54)$$

$$= \underbrace{\mathbf{Q}_k^{-1} \mathbf{A}_{k-1} \left(\hat{\mathbf{P}}_{k-1}^{-1} + \mathbf{A}_{k-1}^T \mathbf{Q}_k^{-1} \mathbf{A}_{k-1} \right)^{-1} \hat{\mathbf{P}}_{k-1}^{-1} \hat{\mathbf{x}}_{k-1}}_{\check{\mathbf{P}}_k^{-1} \mathbf{A}_{k-1} \text{ 由下文推得}} \quad (55)$$

$$+ \underbrace{\left(\mathbf{Q}_k^{-1} - \mathbf{Q}_k^{-1} \mathbf{A}_{k-1} \left(\hat{\mathbf{P}}_{k-1}^{-1} + \mathbf{A}_{k-1}^T \mathbf{Q}_k^{-1} \mathbf{A}_{k-1} \right)^{-1} \mathbf{A}_{k-1}^T \mathbf{Q}_k^{-1} \right)}_{\check{\mathbf{P}}_k^{-1} \text{ by (SMW formula)}} \mathbf{v}_k \quad (56)$$

$$+ \mathbf{C}_k^T \mathbf{R}_k^{-1} \mathbf{y}_k \quad (57)$$

$$= \check{\mathbf{P}}_k^{-1} \underbrace{\left(\mathbf{A}_{k-1} \hat{\mathbf{x}}_{k-1} + \mathbf{v}_k \right)}_{\check{\mathbf{x}}_k} + \mathbf{C}_k^T \mathbf{R}_k^{-1} \mathbf{y}_k, \quad (58)$$

式 55 的推导如下：

$$\underbrace{\mathbf{Q}_k^{-1} \mathbf{A}_{k-1} \left(\hat{\mathbf{P}}_{k-1}^{-1} + \mathbf{A}_{k-1}^T \mathbf{Q}_k^{-1} \mathbf{A}_{k-1} \right)^{-1} \hat{\mathbf{P}}_{k-1}^{-1}}_{\text{apply SMW formula again}} \quad (59)$$

$$= \mathbf{Q}_k^{-1} \mathbf{A}_{k-1} \left(\hat{\mathbf{P}}_{k-1} - \hat{\mathbf{P}}_{k-1} \mathbf{A}_{k-1}^T \underbrace{\left(\mathbf{Q}_k + \mathbf{A}_{k-1} \hat{\mathbf{P}}_{k-1} \mathbf{A}_{k-1}^T \right)^{-1}}_{\check{\mathbf{P}}_k^{-1}} \mathbf{A}_{k-1} \hat{\mathbf{P}}_{k-1} \right) \hat{\mathbf{P}}_{k-1}^{-1} \quad (60)$$

$$= (\mathbf{Q}_k^{-1} - \underbrace{\mathbf{Q}_k^{-1} \mathbf{A}_{k-1} \hat{\mathbf{P}}_{k-1} \mathbf{A}_{k-1}^T}_{\check{\mathbf{P}}_k - \mathbf{Q}_k} \check{\mathbf{P}}_k^{-1}) \mathbf{A}_{k-1} \quad (61)$$

$$= \left(\mathbf{Q}_k^{-1} - \mathbf{Q}_k^{-1} + \check{\mathbf{P}}_k^{-1} \right) \mathbf{A}_{k-1} \quad (62)$$

$$= \check{\mathbf{P}}_k^{-1} \mathbf{A}_{k-1} \quad (63)$$

这样便可得到递归滤波器的形式，从k-1时刻递推出k时刻的系统状态。

$$\check{\mathbf{P}}_k = \mathbf{A}_{k-1} \hat{\mathbf{P}}_{k-1} \mathbf{A}_{k-1}^T + \mathbf{Q}_k, \quad 51 \quad (103)$$

$$\check{\mathbf{x}}_k = \mathbf{A}_{k-1} \hat{\mathbf{x}}_{k-1} + \mathbf{v}_k, \quad \text{其实就是运动方程} \quad (104)$$

$$\hat{\mathbf{P}}_k^{-1} = \check{\mathbf{P}}_k^{-1} + \mathbf{C}_k^T \mathbf{R}_k^{-1} \mathbf{C}_k, \quad 52 \quad (105)$$

$$\hat{\mathbf{P}}_k^{-1} \hat{\mathbf{x}}_k = \check{\mathbf{P}}_k^{-1} \check{\mathbf{x}}_k + \mathbf{C}_k^T \mathbf{R}_k^{-1} \mathbf{y}_k, \quad 58 \quad (106)$$

而KF的经典形式只要稍加变换就可得到：

令卡尔曼增益 \mathbf{K}_k 为：

$$\mathbf{K}_k = \hat{\mathbf{P}}_k \mathbf{C}_k^T \mathbf{R}_k^{-1} \quad (68)$$

在式66式左右各左乘 $\hat{\mathbf{P}}_k$ 得：

$$1 = \hat{\mathbf{P}}_k \left(\check{\mathbf{P}}_k^{-1} + \mathbf{C}_k^T \mathbf{R}_k^{-1} \mathbf{C}_k \right) \quad (69)$$

$$= \hat{\mathbf{P}}_k \check{\mathbf{P}}_k^{-1} + \mathbf{K}_k \mathbf{C}_k, \quad \text{上式展开并把} \mathbf{K}_k \text{带入} \quad (70)$$

$$\hat{\mathbf{P}}_k = (\mathbf{1} - \mathbf{K}_k \mathbf{C}_k) \check{\mathbf{P}}_k, \quad \text{等式两侧右乘} \mathbf{C}_k^T \mathbf{R}_k^{-1} \quad (71)$$

$$\underbrace{\hat{\mathbf{P}}_k \mathbf{C}_k^T \mathbf{R}_k^{-1}}_{\mathbf{K}_k} = (\mathbf{1} - \mathbf{K}_k \mathbf{C}_k) \check{\mathbf{P}}_k \mathbf{C}_k^T \mathbf{R}_k^{-1} \quad (72)$$

$$\mathbf{K}_k \left(\mathbf{1} + \mathbf{C}_k \check{\mathbf{P}}_k \mathbf{C}_k^T \mathbf{R}_k^{-1} \right) = \check{\mathbf{P}}_k \mathbf{C}_k^T \mathbf{R}_k^{-1}, \quad \text{等式两侧右乘} \mathbf{R}_k \quad (73)$$

$$\mathbf{K}_k = \check{\mathbf{P}}_k \mathbf{C}_k^T \left(\mathbf{C}_k \check{\mathbf{P}}_k \mathbf{C}_k^T + \mathbf{R}_k \right)^{-1}, \quad \text{即为卡尔曼增益} \quad (74)$$

综上所述，最终的卡尔曼滤波的核心公式就是：

$$\check{\mathbf{x}}_k = \mathbf{A}_{k-1} \hat{\mathbf{x}}_{k-1} + \mathbf{v}_k, \quad 104 \quad (107)$$

$$\check{\mathbf{P}}_k = \mathbf{A}_{k-1} \hat{\mathbf{P}}_{k-1} \mathbf{A}_{k-1}^T + \mathbf{Q}_k, \quad 103 \quad (108)$$

$$\mathbf{K}_k = \check{\mathbf{P}}_k \mathbf{C}_k^T \left(\mathbf{C}_k \check{\mathbf{P}}_k \mathbf{C}_k^T + \mathbf{R}_k \right)^{-1}, \quad 74 \quad (109)$$

$$\hat{\mathbf{x}}_k = \check{\mathbf{x}}_k + \mathbf{K}_k (\mathbf{y}_k - \mathbf{C}_k \check{\mathbf{x}}_k), \quad 106 \quad \text{等式两侧左乘} \hat{\mathbf{P}}_k \text{即得} \quad (110)$$

$$\hat{\mathbf{P}}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{C}_k) \check{\mathbf{P}}_k, \quad 71 \quad (111)$$

从MAP角度解释KF完结撒花。

从贝叶斯估计的角度解释

卡尔曼是线性系统的递推形式（recursive，也就是从 x_{k-1} 估计到 x_k ）的无偏最优估计（无偏最优指的是期望等于其真实值，具体内容鸽）。

由观测方程可得，用来估计当前的先验：

TODO

从增益最优的角度来解释

TODO

与最小二乘之间的关系

从现有数据（物体运动的时间和位置来估计其状态）用最小二乘的方式表示如下80，但是最小二乘法使用到了全部的数据，随着数据量的增大，计算量会暴增，显然不符合实际计算需求。因此提出将原始的方程82修改为递推的形式，就符合马尔科夫过程了。其中涉及到矩阵的变换，包括协方差和系数矩阵的变换。最重要的需要从最小二乘的矩阵中将状态量和观测量解耦出来，并对每种数据都加入噪声，重新定义观测矩阵与状态转移矩阵。

$$L = (Y_n - H_n \hat{x}_n)^T R_n (Y_n - H_n \hat{x}_n) \tag{80}$$

$$\hat{x}_n = \arg \min_{\hat{x}_n} J \tag{81}$$

$$\hat{x}_n = (H_n^T R_n H_n)^{-1} H_n^T R_n Y_n \tag{82}$$

最小二乘法求解最小目标函数的过程与卡尔曼滤波中求解协方差矩阵迹的最小值的过程非常相似，只不过最小二乘法是使用所有数据来进行优化，而卡尔曼滤波通过对迹求导来获得最优卡尔曼增益。可以通过构造特殊拟合函数的方式来让最小二乘法来递推运作，从而只传递状态噪声。然后定义状态转移过程，并加入过程噪声来描述状态转移误差。

相比使用贝叶斯公式与正态分布假设来推导卡尔曼滤波，最小二乘法推导的条件更为宽松（但是更难理解了，有点别扭，需要先从原来的方式理解KF，再从最小二乘角度理解才行）。使用贝叶斯公式来推导时需要假设所有的噪声服从正态分布，而最小二乘法推导仅需要满足高斯-马尔可夫定理（噪声零均值、同方差、不相关）即可。当然推导卡尔曼滤波的方法并不只这两种，但了解的越多理解越深刻，对学习也更有帮助。

- 从递推最小二乘法推导出卡尔曼滤波的过程并不严谨，直接使用先验估计来替换掉递推最小二乘法中的一些项。如果以后有更加严谨的方式我会补上证明。

Extend Kalman Filter

接下来几乎全文照抄高博知乎(高博YYDS)

然而实际情况是状态并不是服从高斯分布的，噪声也不是，而且运动方程和观测方程也不是线性的，高斯分布经过非线性变换是什么分布也估摸不清楚了，于是有了EKF，提出把一个非线性的东西进行线性化表示，取 \hat{x}_{k-1} ， \tilde{x} 处的近似值，那么公式1,2中表示的运动方程和观测方程就线性化成如下形式：

$$\begin{aligned} f(x_{k-1}, v_k, w_k) &\approx f(\hat{x}_{k-1}, v_k, 0) + \frac{\partial f}{\partial x_{k-1}}(x_{k-1} - \hat{x}_{k-1}) + \frac{\partial f}{\partial w_k} w_k \\ g(x_k, n_k) &\approx g(\tilde{x}_k, 0) + \frac{\partial g}{\partial x_k} n_k \end{aligned} \tag{83}$$

同时EKF认为状态和噪声都服从高斯分布（强假设），然后用线性化之后的矩阵去代替卡尔曼滤波器里的转移矩阵和观测矩阵即可。

$$\check{P}_k = F_{k-1} \hat{P}_{k-1} F_{k-1}^T + Q'_k$$

$$\begin{aligned}
\tilde{x}_k &= f(\hat{x}_{k-1}, v_k, 0) \\
K_k &= \tilde{P}_k G_k^T (G_k \tilde{P}_k G_k^T + R'_k)^{-1} \quad \text{其中} \quad F_{k-1} = \left. \frac{\partial f}{\partial x_{k-1}} \right|_{\hat{x}_{k-1}}, G_k = \left. \frac{\partial g}{\partial x_k} \right|_{\tilde{x}_k} \\
\hat{P}_k &= (I - K_k G_k) \tilde{P}_k \\
\hat{x}_k &= \tilde{x}_k + K_k (y_k - g(\tilde{x}_k, 0))
\end{aligned} \tag{84}$$

所以EKF面临的一个重要问题是，当一个高斯分布经过非线性变换后，如何用另一个高斯分布近似它？按照它现在的做法，存在以下的局限性：（注意是滤波器自己的局限性，还没谈在SLAM问题里的局限性）。

1. 即使是高斯分布，经过一个非线性变换后也不是高斯分布。EKF只计算均值与协方差，是在用高斯近似这个非线性变换后的结果。（实际中这个近似可能很差）。
2. 系统本身线性化过程中，丢掉了高阶项，在非高斯分布时，误差就很大了。
3. 线性化的工作点往往不是输入状态真实的均值，而是一个估计的均值。于是，在这个工作点下计算的 F, G ，也不是最好的。
4. 在估计非线性输出的均值时，EKF算的是 $\mu_y = f(\mu_x)$ 的形式。这个结果几乎不会是输出分布的真正期望值。协方差也是同理。

那么，怎么克服以上的缺点呢？途径很多：

- 为了克服第3条工作点的问题，以EKF估计的结果为工作点，重新计算一遍EKF，直到这个工作点变化够小。是为迭代EKF (Iterated EKF, IEKF)。
- 为了克服第4条，除了计算 $\mu_y = f(\mu_x)$ ，再计算其他几个精心挑选的采样点，然后用这几个点估计输出的高斯分布。是为Sigma Point KF (SPKF, 或UKF)。
- 如果说不要高斯分布假设，凭什么要用高斯去近似一个长得根本不高斯的分布呢？于是问题变为，丢掉高斯假设后，怎么描述输出函数的分布就成了一个问题。一种比较暴力的方式是：用足够多的采样点，来表达输出的分布。这种蒙特卡洛的方式，也就是粒子滤波(Particle Filter, PF)的思路。
- 如果再进一步，可以丢弃滤波器思路，说：为什么要用前一个时刻的值来估计下一个时刻呢？可以把所有状态看成变量，把运动方程和观测方程看成变量间的约束，构造误差函数，然后最小化这个误差的二次型。这样就会得到非线性优化的方法，在SLAM里就走向图优化那条路上去了。不过，非线性优化也需要对误差函数不断地求梯度，并根据梯度方向迭代，因而局部线性化是不可避免的。

Error State Kalman Filter

KF假设系统是线性高斯，EKF为了满足线性线性化假设，强行线性化的操作很多时候不符合实际情况，就有人想到如果将状态的误差看做是“状态”呢，每次对状态的误差按照KF的流程走一遍，然后对真实状态进行更新，然后再计算误差，一直迭代，于是有了ESKF。

无论是Joan大佬采用的基于四元数表示旋转的ESKF，还是weikun zhen、高博等采用的SO3表示的ESKF，其核心的基于KF思想的预测predict和更新update/correct都是一样的，主要的关注点是如何利用状态误差构建关于状态误差的状态方程以及状态和状态误差之间的更新。区别仅在于表示动力学方程中和旋转相关的 $\delta\theta$ 不同。

使用状态误差做KF的优点：

- 在旋转的处理上，ESKF的状态变量可以采用最小化的参数表达，也就是使用三维变量来表达旋转的增量。而传统KF需要用到四元数（4维）或者更高维的表达（旋转矩阵，9维），要不就得采用带有奇异性的表达方式（欧拉角）。
- ESKF总是在原点附近，离奇异点较远，并且也不会由于离工作点太远而导致线性化近似不够的问题。
- ESKF的状态量为小量，其二阶变量相对来说可以忽略。同时大多数雅可比矩阵在小量情况下变得非常简单，甚至可以用单位阵代替。
- 误差状态的运动学也相比原状态变量要来得更小，因为可以把大量更新部分放到原状态变量中。

考虑到IMU的特性，用什么类型的KF都会导致漂移，因此多传感器融合成为必要性，除了IMU之外的传感器的观测数据可以作为KF中更新模块的观测，纠正IMU的bias。

假设观测方程为: $v \sim \mathcal{N}(0, V)$ 为高斯分布的观测噪声

$$y = h(x_t) + v \quad (85)$$

按照KF的思想, 其对应的预测和更新为: 戴帽子的符号和KF表示的意义相同, \leftarrow 表示的就是更新, 赋值的意思, 数学上表示的意义不严谨, 权可认为与 "=" 意义相同。

$$\check{\delta x} \leftarrow F_x(x, u_m) \cdot \delta x, \text{ 误差预测} \quad (86)$$

$$\check{P} \leftarrow F_x P F_x^T + F_i Q F_i^T, \text{ 协方差更新} \quad (87)$$

$$K = \check{P} H^T (H \check{P} H^T + V)^{-1}, \text{ 卡尔曼增益} \quad (88)$$

$$\hat{\delta x} \leftarrow K(y - h(\hat{x}_t)), \text{ 误差更新} \quad (89)$$

$$\hat{P} = \check{P} + K H \check{P} = (I + K H) \check{P}, \text{ 协方差更新} \quad (90)$$

- 具体的公式这里就不贴了, 在Joan大神的论文中写的很清楚。
- 其中 $\delta x \sim \mathcal{N}(\hat{\delta x}, P)$, F_x, F_i 分别是状态方程对误差状态和噪声的一阶导。
- $H \equiv \frac{\partial h}{\partial x} \Big|_x = \frac{\partial h}{\partial x_i} \Big|_x \frac{\partial x_i}{\partial \delta x} \Big|_x = H_x X_{\delta x}$, 是根据链式法则求得的观测方程对状态误差的偏导数, 在KF中这里的H就是线性化的观测矩阵, 仿照EKF的思想实现线性化的假设, 这里也对状态误差做了线性化处理。
- 需要注意的是ESKF中是对误差状态建模的, 因此需要将误差状态更新inject到实际的状态向量中, 然后归零继续下一次线性化展开和KF过程。需要明白, **ESKF**还是按照**KF**的假设来的, 但处理的对象是状态误差, 虽然也有非线性化的操作, 但是和**EKF**直接对状态做**KF**是不同的。在图优化中也用了线性化的操作。

疑问:

- 将旋转表示在local系和global系之间的区别是? 分别在什么场景下使用呢?

Particle Filter

当然有一些系统不能有用高斯分布近似模拟噪声, 而且即使一开始的状态是高斯分布的, 但是经过非线性变换之后的分布不明, 因此需要使用一些像粒子滤波这样更复杂滤波器, 直接通过暴力的方式不断采样-算权重-重采样。越符合观测的粒子拥有越大的权重, 而权重越大就越容易在重采样时被采到。核心是采样数量, 以及权重如何设置。当然还有很多问题, 没做过也不知道深浅。

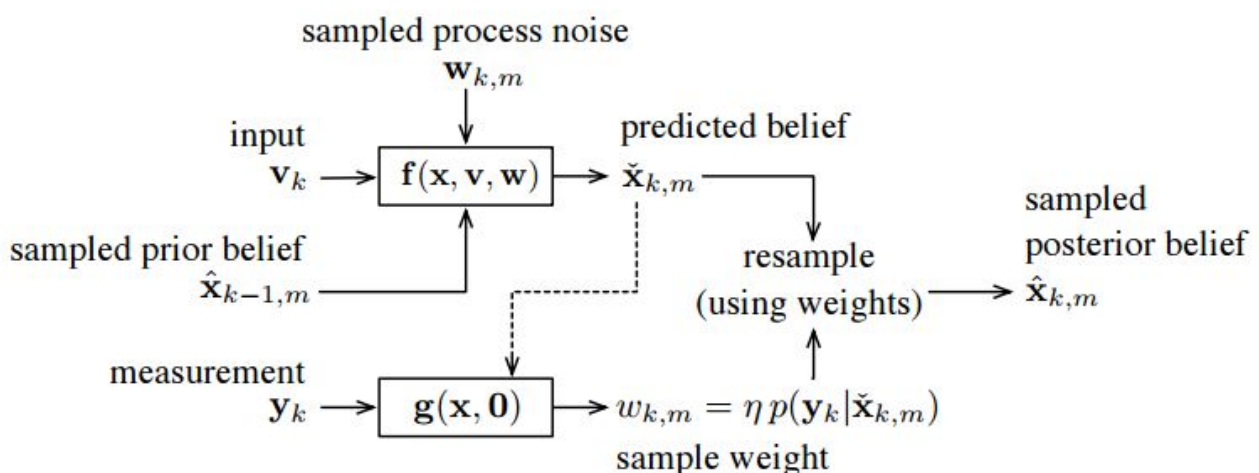


图4. Particle Filter示意图

- FastSLAM是基于PF的Vslam

经典状态估计中协方差的计算能准确反映机器人状态的完整性吗？

使用该方法计算所得的状态量协方差不能用以在退化和传感器失效等场景下机器人状态的完整性监控。[原文链接](#)。一点点引申：在“特征”离群值的判断与剔除算法机制的帮助下，可以在一定程度上规避垃圾观测数据和传感器失效引发的机器人状态估计错误问题，同时机器人状态完整性监控算法可为帮助机器人能准确地表征其状态估计量的准确值，以帮助机器人在传感器退化和传感器失效等场景下及时完成导航、定位策略的切换和多传感器观测模型的深度融合。另一方面，“特征”离群值的判断与剔除算法和机器人状态完整性监控算法也为全生命周期建图定位系统Life-long SLAM的稳定实现提供了基础。“特征”离群值的判断与剔除算法可以帮助克服Life-long对视角变化，天气变化，光照变化条件下，以及动态障碍物环境下，如何确保定位和建模的精度度的核心问题，而机器人状态完整性监控算法则辅助保证Life-long过程中约束构建的准确性，当Life-long过程中的环境发生变化之后，通过完整性监控可以避免匹配约束构建后的歧义，即是通过完整性监控获取得到的匹配约束准确度，从而对“较差”的约束进行其他处理。

Filter到底是不是直观意义上的过滤？

滤波器的本质是根据传感器的信号估计真实状态。静态的、离线的估计可以使用最小二乘或者批估计的方法，但是动态的、在线的估计（如机器人的位姿）就需要使用像卡尔曼这样复杂一点的滤波方法。如果把KF想成是一个能从带有噪声的观测数据/信号中得到真实的状态，那么可以认为是过滤掉了噪声，但是远比过滤要复杂的多，这里已经不是字面上的过滤这一个操作了，而是广义的过滤。

filter & smoothing & 图优化

- 在状态估计问题中，假设某物体的运动状态为 $x_1, x_2, \dots, x_{k-1}, x_k$ ，他们随时间不断变化的，如果认为当前状态 x_k 只和前面的 x_{k-1} 有关，那么就用KF,EKF,ESKF等基于filter的方法，也就是状态估计满足马尔科夫模型假设；如果认为当前状态 x_k 和前面的都有关，则用图优化的方式，采用batch的形式迭代进行最优状态的估计。
- 区别只在于使用的信息量，如果只用前一个或者前几个，那么就是KF-based的或者sliding window KF based的，当然马尔科夫模型假设太强了，实际不一定只和前面的几个有关，那么就会采用batch的形式一起图优化，理论上拿到的信息更多，结果应该更好才对。而实际使用中考虑到效率和实时性的要求，sliding window的方式更实用。
- smoothing则是说根据k时刻之后的状态，e.g. x_{k+1} 反向更新k时刻的状态 x_k ，反向的这个过程提升了当前状态估计结果的精度，毕竟前后都考虑到了，但是因为用了未来时刻的状态估计现在的状态，说明它做不到实时，还可能更耗时，只能是一种off-line线下操作。而这么做的目的就是为了得到更好的状态估计结果。

不同滤波器方法之间的简单比较

- 卡尔曼滤波是递归的线性高斯系统最优估计。
- EKF将NLNG系统在工作点附近近似为LG进行处理。
- PF去掉高斯假设，以粒子作为采样点来描述分布。

Gauss-Newton iterates over the entire trajectory, but runs offline and not in constant time

x_0 x_1 x_2 x_3 \dots x_{k-2} x_{k-1} x_k x_{k+1} x_{k+2} \dots x_K



Sliding-window filters iterate over several timesteps at once, run online and in constant time

x_0 x_1 x_2 x_3 \dots x_{k-2} x_{k-1} x_k x_{k+1} x_{k+2} \dots x_K



IEKF iterates at only one timestep at a time, but runs online and in constant time

x_0 x_1 x_2 x_3 \dots x_{k-2} x_{k-1} x_k x_{k+1} x_{k+2} \dots x_K



图5. 不同种类的滤波器之间的简单比较示意图

其他非高斯分布的解决方案

在视觉slam中存在大量非线性非高斯分布的情况，e.g.图像的点像素值不是高斯分布的，而其逆深度是符合高斯分布的，因此转换为逆深度进行求解。

IMU ✓

本文介绍的状态估计方法中，几乎每种都使用到了IMU这种传感器作为观测数据的输入，因此，有必要了解IMU到底是个啥，其工作原理十分有利于理解上述公式推导。这里只简单介绍和状态估计相关的，IMU本身是一个大的方向，我也没能力总结出来。imu

1. IMU类型

- 3轴IMU即只有3轴陀螺仪的IMU，其因为只有一个3轴陀螺仪gyroscopes，所以只能感知载体roll、pitch、yaw共3个自由度的姿态信息。
- 6轴IMU在3轴IMU的基础上加装了3轴加速度计accelerometers，因此在感知载体姿态的基础上，还能感知载体3个自由度上的加速度信息。
- 9轴IMU在6轴IMU的基础上加装了3轴磁强计magnetometer，由于3轴陀螺仪只能估计载体自身的相对位姿变化（通过加速度计也可获得载体的绝对roll和pitch），单凭3轴陀螺仪无法获取载体的全部姿态信息，而通过3轴磁强计就可以，本质上磁强计的感知原理类似于指南针。

有学者说，磁强计在小范围内还行，大范围容易受到地磁不一样导致的误差，所以目前6轴的在机器人领域用的更多一些；由于工业制造的问题，IMU安装到板子上之后，由于散热器震动，高温等原因会导致IMU给出的数值不稳定，这也是加装了IMU的状态估计方法在使用时尽量在线online标定的一个原因。

2. IMU状态估计原理

其实是符合牛顿第二定律 $f=ma$ ，利用动力学原理，从IMU测量的载体自身的旋转和加速度来得到当前的位置。其中就使用到了IMU积分的相关内容，加速度积分得到速度，速度积分得到位置，虽然IMU本身会输出一个直接积分得到的速度和位置，但是几乎无法使用，因此还是要用加速度和角速度自己积分。而为了提升积分的效率，有了预计分的思想。其中的数学也很硬核。

6轴IMU由两部分组成，加速度计和陀螺仪，这两个一起都是有偏置误差的，也就是说初始位置0位不准确，需要进行标定，同时测量量还是有噪声的。也就是说IMU输出的值是包含了偏置bias和噪声的，同时加速度计可以感应到地球重力，因此真实的载体的加速度需要减去重力加速度。

3. IMU预计分

IMU预计分涉及到的数学有点多，目前还没有来得及学习和整理。

- 百度百科的欧拉积分中有错误，但是一直不修改。

概率论

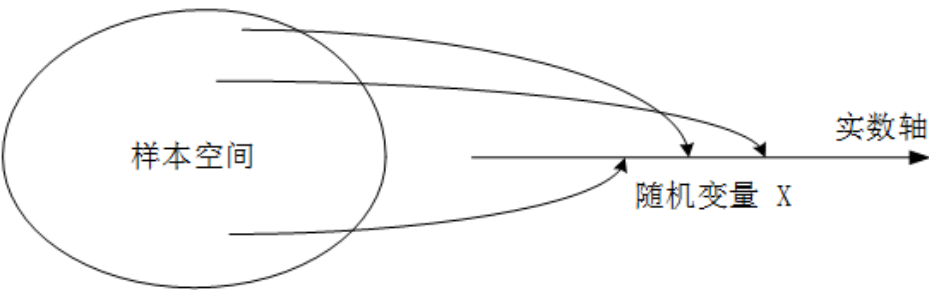
随机变量

有几个概念上的生疏导致理解困难。

1. 状态服从高斯分布是什么意思？是随机变量吗？噪声可以理解。

随机变量 *is a variable whose possible values are the outcomes of a random phenomenon*

随机变量的意思是将样本所在的样本空间到实数域的映射，状态是一个在某一空间（e.g. SE3, 欧式空间）真值周围随机变化的，为了描述其处于哪里，以及其对应的概率，需要映射到实数域来表示，因此完全可称状态为是随机变量。



2. 随机变量方差线性变换公式

$$D(x_1 + x_2) = D(x_1) + D(x_2) \tag{91}$$

$$D(Ax_1 + b) = A^2 D(x_1) + D(b), \text{ if } d \text{ is a random variable} \tag{92}$$

Emmmm...

简单罗列一些概率论中的常见术语，学的时候没用起来，用的时候早忘了...

1. 某一服从某一分布随机变量的概率密度函数 **probability density function, PDF**: 连续型如此，离散型就求和

$$F_X(x) = \int_{-\infty}^x p(\tau) d\tau \tag{93}$$

2. 多个类似随机变量的概率密度函数称为联合概率密度函数:

$$f_X(x) = \frac{\partial^n F}{\partial x_1 \dots \partial x_n} \Big|_x \tag{94}$$

$$F_X(x_1, x_2, \dots, x_n) = \int_D f_X(x_1, x_2, \dots, x_n) dx_1 \dots dx_n \tag{95}$$

3. 随机变量的两个统计意义上的度量 $\mu, \Sigma = \delta^2$:

$$E(X) = \int_{-\infty}^{\infty} xp(x)dx \quad (96)$$

$$Var(X) = E([X - E(X)]^2) \quad (97)$$

$$= \int_{-\infty}^{\infty} [x - E(x)]^2 p(x) dx = E(X^2) - E^2(X) \quad (98)$$

4. Sherman-Morrison-Woodbury

TODO

极大似然估计&极大后验估计

是解决同一个任务的两种不同方法，问题是利用已知的数据，推测模型的参数。极大似然估计MLE属于频率派，极大后验估计MAP属于贝叶斯派。

- MLE是为了最大化这个事情发生的概率。进一步可以推导向最小二乘。
- MAP是为了最大化在给定数据样本的情况下模型参数的后验概率，需要给定一个先验概率，然后不断修正。
- 为什么KF可以同时从MAP和贝叶斯推估的角度同时推导出来？是因为其假设为线性高斯的系统，两者推导结果一致，如果不是线性的，结果全然不同。

Bayesian Estimation or Rules

下图真的非常不错

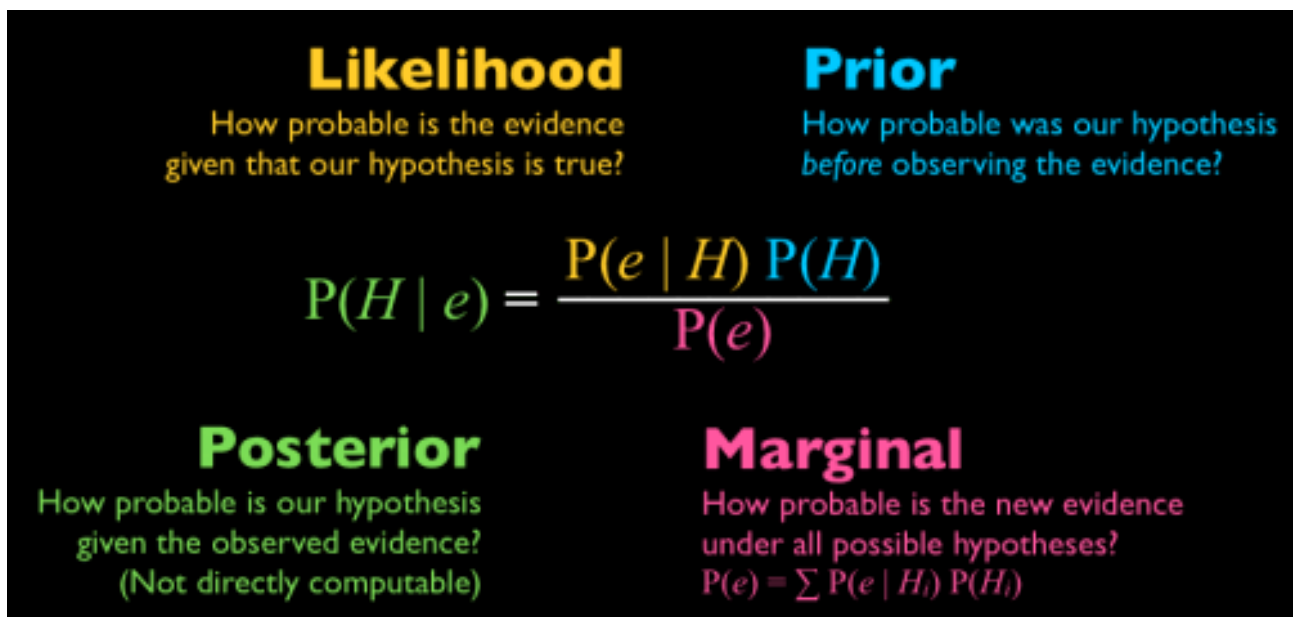


图4.贝叶斯公式

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)} \quad (99)$$

$$P(AB) = P(A|B)P(B) = P(B|A)P(A) \quad (100)$$

上面就是最基础的贝叶斯公式以及变体。如果把 $P(A)$ 用全概率公式写出来，就是将所有B事件发生的概率 $P(B)$ 下，对应A事件发生的条件概率 $P(A|B)$ 的总和，即贝叶斯公式就会变为：

- 连续型(积分)， $p_Y(y), p_X(x)$ 就是边缘概率密度， x, y 分别对应B, A

$$p(x|y) = \frac{p(x, y)}{p_Y(y)} = \frac{p(y|x)p_X(x)}{p_Y(y)} = \frac{p(y|x)p_X(x)}{\int p(y|x)p_X(x)dx} \quad (101)$$

- 离散型(求和)

$$P(A|B_i)P(B_i)$$

$$P(B_i|A) = \frac{P(A|B_i)P(B_i)}{\sum_{i=1}^n P(A|B_i)P(B_i)} \quad (102)$$

Reference

以下排名不分先后：

- [知乎上关于各种滤波器的讨论](#)
- [高博关于ESKF的简明推导](#)
- Quaternion based 的ESKF, Quaternion kinematics for the error-state Kalman filter
- weikun zhen CMU 2021年博士论文SO3 based的ESKF, Robust State Estimation and Mapping for Autonomous Inspection
- [LiDAR_ESKF](#) 坚信talk is cheap, show me the code 的真理。
- 最硬核的应该直接看原版的STATE ESTIMATION FOR ROBOTICS，高博翻译了中文版。
- [Kalman Filter's wikipedia](#).
- [IMU集合](#)
- [知乎推文1](#)