BIT

从两个角度看经典卡尔曼滤波

汇报人：王子恺

时间：2020/7/29

北京理工大学
BEIJING INSTITUTE OF TECHNOLOGY

德以明理 学以精工

目录 | CONTENTS

1 背景与前言
2 卡尔曼滤波的直觉性思考
3 卡尔曼滤波与最小二乘法
4 工程应用实例
5 内容总结

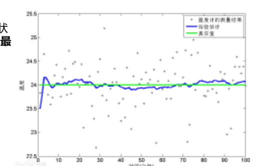
---

1 背景与前言

背景与前言

卡尔曼滤波（Kalman filtering）是一种利用线性系统状态方程，通过系统输入输出观测数据，**对系统状态进行最优估计**的算法。

对系统状态的直接观测通常存在一定的**不确定性**，反映在数据上则是观测数据上存在着服从某种分布的随机噪声。最优估计的过程削弱了噪声的影响，所以最优估计可以看作是“**滤波**”的过程。

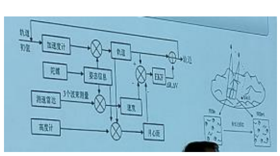


背景与前言

什么领域可以应用卡尔曼滤波?

理论上凡是涉及**估计**与**预测**的领域都可以尝试使用。

包括但不限于航空航天、自动驾驶、机器人、SLAM、图像处理等



---

2 卡尔曼滤波的直觉性思考

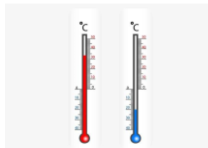
卡尔曼滤波的直觉性思考

误差与权重的思想：

卡尔曼滤波在处理和融合多源信息时，核心思路是**加权**

经典例子：两个温度计测量房间温度时，信哪个温度计多一点

在这样的思想下，我们每写一个方程，每做一次外推，都要衡量一下这项行动的**不确定度**，以便于融合时分配权重。



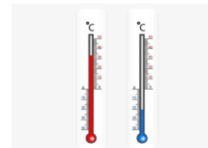
卡尔曼滤波的直觉性思考

误差与权重的思想：

我们通常用**方差**、**协方差矩阵**来描述‘不确定度’

需要强调的一点是，经典卡尔曼滤波假定误差是服从零均值**高斯分布**的

$$\mathcal{N}(x,\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



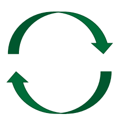
卡尔曼滤波的直觉性思考

卡尔曼滤波的运行过程可以看作是“**预测**”、“**更新**”的循环

预测：

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1}$$

$$P_k^- = AP_{k-1}A^T + Q$$



卡尔曼滤波的直觉性思考

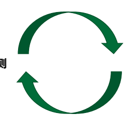
卡尔曼滤波的运行过程可以看作是“**预测**”、“**更新**”的循环

$$K_k = \frac{P_k^- H^T}{HP_k^- H^T + R}$$

更新：

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$$

$$P_k = (I - K_k H)P_k$$



卡尔曼滤波的直觉性思考

案例假设

假如，你做了一个可以在树林中行走的小机器人，这个机器人需要准确的知道它自己所处的位置，因为它需要导航。

状态： $\vec{x}_k = (\vec{p}, \vec{v})$

协方差： $P_k = \begin{bmatrix} \Sigma_{pp} & \Sigma_{pv} \\ \Sigma_{vp} & \Sigma_{vv} \end{bmatrix}$

卡尔曼滤波的直觉性思考

我们可以建立机器人的**运动模型**

机器人有一个GPS**传感器**，它精确到10米左右，测量间隔为10s。（实际中通常为0.1s）

人为施加**控制量**，例如加大油门，可以产生加速度a，为简化案例，设其为已知量

卡尔曼滤波的直觉性思考

在GPS的测量结果**尚未到达**时，我们需要用运动学方程来**预测**下一时刻机器人的状态量（位置与速度）

运动学方程：
$$p_k = p_{k-1} + \Delta t v_{k-1} + \frac{1}{2} a \Delta t^2$$
$$v_k = v_{k-1} + a \Delta t$$

$\xrightarrow{\text{矩阵}}$

$$\hat{\mathbf{x}}_k = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \hat{\mathbf{x}}_{k-1} + \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix} a = \mathbf{F}_k \hat{\mathbf{x}}_{k-1} + \mathbf{B}_k \vec{\mathbf{u}}_k$$

协方差矩阵：
$$\hat{\mathbf{x}}_k = \mathbf{F}_k \hat{\mathbf{x}}_{k-1} + \mathbf{B}_k \vec{\mathbf{u}}_k \xrightarrow[]{\substack{Cov(x)=\Sigma \\ Cov(\mathbf{A}x)=\mathbf{A}\Sigma\mathbf{A}^T}} \mathbf{P}_k = \mathbf{F}_\mathbf{k} \mathbf{P}_{k-1} \mathbf{F}_k^T + \mathbf{Q}_k$$

这就是之前提到的五大公式其二：
$$\begin{cases} \hat{x}_{\bar{k}} = A\hat{x}_{k-1} + Bu_{k-1} \\ P_k^- = AP_{k-1}A^T + Q \end{cases}$$

卡尔曼滤波的直觉性思考

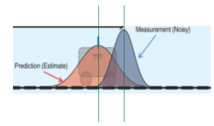
当GPS的测量值**到达**时：

测量值： $\vec{\mu}_{\text{expected}} = \mathbf{H}_k \hat{\mathbf{x}}_k + v_k$

协方差矩阵： $\Sigma_{\text{expected}} = \mathbf{H}_k \mathbf{P}_k \mathbf{H}_k^T + R_k$

现在，我们有了**两个**对系统状态的**估计值**，一个由运动学模型**预测**得出，一个由传感器**测量**得出。

基于我们之前的假设：模型噪声和测量噪声服从均值为0的**高斯分布**



这意味着，我们的估计值和测量值，也都可以看作是服从高斯分布的**多维随机变量**

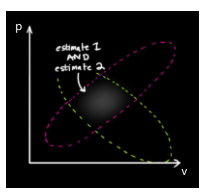
卡尔曼滤波的直觉性思考

将本案例中的预测值与测量值，作为**二维随机变量**大致画在平面图中

光斑越亮的位置，代表可能性越大

两个分布的重叠部分，即两个斑点都很亮的区域，是真实值最可能出现的地方。

求两个概率的重叠部分，其实就是做**乘法**



卡尔曼滤波的直觉性思考

求重叠即做乘法：
$$\underbrace{\mathcal{N}(x,\mu_0,\sigma_0)}_{\text{模型预测}} \cdot \underbrace{\mathcal{N}(x,\mu_1,\sigma_1)}_{\text{传感器测量}} = \underbrace{\mathcal{N}(x,\mu',\sigma')}_{\text{最终估计}}$$

代数运算+归一化：
$$\mu' = \mu_0 + \frac{\sigma_0^2(\mu_1 - \mu_0)}{\sigma_0^2 + \sigma_1^2}$$
$$\sigma'^2 = \sigma_0^2 - \frac{\sigma_0^4}{\sigma_0^2 + \sigma_1^2}$$

化简：
$$\mu' = \mu_0 + \mathbf{k}(\mu_1 - \mu_0)$$
$$\sigma'^2 = \sigma_0^2 - \mathbf{k}\sigma_0^2$$

其中： $\mathbf{k} = \frac{\sigma_0^2}{\sigma_0^2 + \sigma_1^2}$ 为权重

卡尔曼滤波的直觉性思考

所以我们在融合模型预测值和传感器测量值时，其实一共只需要右边这三个式子：

$$\begin{cases} \mathbf{k} = \frac{\sigma_0^2}{\sigma_0^2 + \sigma_1^2} & \text{为权重系数} \\ \mu' = \mu_0 + \mathbf{k}(\mu_1 - \mu_0) & \text{为最终估计} \\ \sigma'^2 = \sigma_0^2 - \mathbf{k}\sigma_0^2 & \text{为方差评价} \end{cases}$$

写成**矩阵形式**：
$$\mathbf{K} = \Sigma_0(\Sigma_0 + \Sigma_1)^{-1}$$
$$\vec{\mu}' = \vec{\mu_0} + \mathbf{K}\left(\vec{\mu_1} - \vec{\mu_0}\right)$$
$$\Sigma' = \Sigma_0 - \mathbf{K}\Sigma_0$$

$\xrightarrow{\text{代入协方差阵}}$

$$\begin{cases} \mathbf{K}' = \mathbf{P}_k \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \\ \hat{\mathbf{x}}_k' = \hat{\mathbf{x}}_k + \mathbf{K}'\left(\vec{\mathbf{z}_k} - \mathbf{H}_k \hat{\mathbf{x}}_k\right) \\ \mathbf{P}_k' = \mathbf{P}_k - \mathbf{K}'\mathbf{H}_k \mathbf{P}_k \end{cases}$$

卡尔曼滤波的直觉性思考

至此，我们完成了对机器人小车的一次状态估计。

顺便还推导出了经典卡尔曼滤波的五大公式

只要继续迭代下去，就可以获得对系统状态的最优估计

模型预测：
$$\begin{cases} \hat{x}_{\bar{k}} = A\hat{x}_{k-1} + Bu_{k-1} \\ P_k^- = AP_{k-1}A^T + Q \end{cases}$$

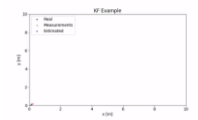
加权融合：
$$\begin{cases} \mathbf{K}' = \mathbf{P}_k \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \\ \hat{\mathbf{x}}_k' = \hat{\mathbf{x}}_k + \mathbf{K}'\left(\vec{\mathbf{z}_k} - \mathbf{H}_k \hat{\mathbf{x}}_k\right) \\ \mathbf{P}_k' = \mathbf{P}_k - \mathbf{K}'\mathbf{H}_k \mathbf{P}_k \end{cases}$$

卡尔曼滤波的直觉性思考

值得一提的是，所谓的**卡尔曼增益K**，作为融合时的权重系数，反映的是我们对测量值的**信任程度。**

$$\mathbf{K}' = \mathbf{P}_k \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k \mathbf{H}_k^T + \mathbf{R}_k)^{-1}$$

正确使用卡尔曼滤波，需要对**过程噪声**和**测量噪声**的协方差做出较为准确的判断。



在**正确使用**的卡尔曼滤波器中，卡尔曼增益会**不断减小。**这意味着测量值对最终估计所起的作用会越来越小，估计模型会**越来越准确**

3 卡尔曼滤波与最小二乘法

卡尔曼滤波与最小二乘法

- 经常有文献会提到，卡尔曼滤波是是一种**线性，无偏，且误差方差最小**的随机系统最优估计算法。但从刚才的直觉性理解中，我们似乎没有看到这类性质的清晰体现。我们还需要从**最优估计**的角度去理解卡尔曼滤波。
- 学习卡尔曼滤波估计，要知道最优估计理论里面的相关知识，更要了解最基础最根本的最优估计理论，那就是最小二乘法。

卡尔曼滤波与最小二乘法

普通最小二乘法：

在一个线性系统中，系统状态估计方程为： $Z = Hx + v$

设$\hat{x}$ 为x的最优估计，可定义误差为： $e = Z - H\hat{x}$

定义代价函数： $J = e_1^2 + e_2^2 + \cdots + e_n^2$

将代价函数写成向量：

$$\begin{aligned} J &= e^T e = (Z - H\hat{x})^T (Z - H\hat{x}) \\ &= Z^T Z - \hat{x}^T H^T Z - Z^T H\hat{x} + \hat{x}^T H^T H\hat{x} \end{aligned}$$

卡尔曼滤波与最小二乘法

对矩阵求偏导，令导数为0：

$$\begin{aligned} \frac{\partial J}{\partial \hat{x}} &= -H^T Z - H^T Z + (H^T H + H^T H)\hat{x} \\ &= -2H^T Z + 2H^T H\hat{x} = 0 \end{aligned}$$

解得 $\hat{x}$ 为： $\hat{x} = (H^T H)^{-1} H^T Z$ 即为**普通最小二乘的估计值**

可以看出，估计值是观测值的线性函数，即线性估计

估计误差： $\tilde{x} = x - \hat{x} = -(H^T H)^{-1} H^T v$

当测量误差的均值为0时，估计误差的数学期望等于零，**最小二乘估计是无偏估计**

卡尔曼滤波与最小二乘法

- 在普通最小二乘中，假设了每一次测量的权重相同，但事实上这样并不合理。假如一个估计值偏离真实值很远，那么它对估计结果的影响就应该被削弱，反之影响应该加强。**加权最小二乘**就是做这样的事。

我们给之前的代价函数增加一个权重矩阵： $J_w(\hat{x}) = (Z - H\hat{x})^T W (Z - H\hat{x})$

展开求偏导，如法炮制得： $\hat{x}_{wls} = (H^T R^{-1} H)^{-1} H^T R^{-1} Z$

其中R为噪声方差阵： $R = E[vv^T]$

卡尔曼滤波与最小二乘法

- 但是，加权最小二乘仍然有个问题，就是前面计算出最优估计量后，当有新的数据进来时候，又要重新计算最优估计量。数据量比较大的时候，计算量会变得特别大。
  所以引出了递推最小二乘，当有新的数据进来，不会再把以前的数据重新计算一遍，而是用当前的观测值对以前的最优估计量进行矫正更新。

第k次测量： $Z_k = H_k X + v_k$

假设第k次的最优估计量为前一次最优估计量与测量误差的线性组合： $\hat{x}_k = \hat{x}_{k-1} + K_k(Z_k - H_k \hat{x}_{k-1})$

卡尔曼滤波与最小二乘法

估计误差： $\tilde{x} = x_k - \hat{x}_k = x_k - \hat{x}_{k-1} - K_k(Z_k - H_k \hat{x}_{k-1})$

又由加权最小二乘： $\hat{x}_{wls} = (H^T R^{-1} H)^{-1} H^T R^{-1} Z$

代入估计误差方差阵P_k：

$$
\begin{aligned}
P_k &= Var(\tilde{x}) = E[\tilde{x}\tilde{x}^T] \\
&= E[-(H^T R^{-1} H)^{-1} H^T R^{-1} v][-(H^T R^{-1} H)^{-1} H^T R^{-1} v]^T \\
&= E[-(H^T R^{-1} H)^{-1} H^T R^{-1} v v^T R^{-1} H (H^T R^{-1} H)^{-1}] \\
&= (H^T R^{-1} H)^{-1} H^T R^{-1} R R^{-1} H (H^T R^{-1} H)^{-1} \\
&= (H^T R^{-1} H)^{-1} H^T R^{-1} H (H^T R^{-1} H)^{-1} \\
&= (H^T R^{-1} H)^{-1}
\end{aligned}
$$

卡尔曼滤波与最小二乘法

最优估计： $\hat{x}_{wls} = P_k H_k^T R_k^{-1} Z_k$

将P_k也写成递推形式：

$$
\begin{aligned}
P_k &= E[(x_k - \hat{x})(x_k - \hat{x})^T] \\
&= E[x_k - \hat{x}_{k-1} - K_k(Z_k - H_k \hat{x}_{k-1})] \\
&\quad \cdot [x_k - \hat{x}_{k-1} - K_k(Z_k - H_k \hat{x}_{k-1})]^T \\
&= (1 - K_k H_k) P_{k-1} (1 - K_k H_k)^T + K_k R_k K_k^T
\end{aligned}
$$

代价函数为P_k的迹： $J = tr(P_k)$

卡尔曼滤波与最小二乘法

J对K_k求偏导，令其为0得： $K_k = P_{k-1} H_k^T (H_k P_{k-1} H_k^T + R_k)^{-1}$

代回P_k： $P_k = P_{k-1} - K_k H_k P_{k-1} = (1 - K_k H_k) P_{k-1}$

递推最小二乘到此推导完毕。主要方程是如下四个：

$$
\begin{cases}
Z_k = H_k X + v_k \\
\hat{x}_k = \hat{x}_{k-1} + K_k(Z_k - H_k \hat{x}_{k-1}) \\
K_k = P_{k-1} H_k^T (H_k P_{k-1} H_k^T + R_k)^{-1} \\
P_k = P_{k-1} - K_k H_k P_{k-1} = (1 - K_k H_k) P_{k-1}
\end{cases}
$$

卡尔曼滤波与最小二乘法

递推最小二乘法：

$$
\begin{cases}
K_k = P_{k-1} H_k^T (H_k P_{k-1} H_k^T + R_k)^{-1} \\
\hat{x}_k = \hat{x}_{k-1} + K_k(Z_k - H_k \hat{x}_{k-1}) \\
P_k = P_{k-1} - K_k H_k P_{k-1} = (1 - K_k H_k) P_{k-1}
\end{cases}
$$

卡尔曼滤波更新方程组：

$$
\begin{cases}
\mathbf{K}' = \mathbf{P}_k \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \\
\hat{\mathbf{x}}'_k = \hat{\mathbf{x}}_k + \mathbf{K}'(\vec{\mathbf{z}}_k - \mathbf{H}_k \hat{\mathbf{x}}_k) \\
\mathbf{P}'_k = \mathbf{P}_k - \mathbf{K}' \mathbf{H}_k \mathbf{P}_k
\end{cases}
$$

两者具有极其相似的形式

卡尔曼滤波与最小二乘法

对比发现，卡尔曼滤波比递推最小二乘法只多了“**模型预测**”这一部分

而模型预测方程实际就是**离散系统的状态空间表达式**

我们可以将卡尔曼滤波理解为线性系统的离散**状态空间模型**和**递推最小二乘**的组合应用。因此卡尔曼滤波继承了最小二乘法的一部分特征。

模型预测：

$$
\begin{cases}
\hat{\boldsymbol{x}}_{\bar{k}} = \boldsymbol{A}\hat{\boldsymbol{x}}_{k-1} + \boldsymbol{B}\boldsymbol{u}_{k-1} \\
P_k^- = A P_{k-1} A^T + Q
\end{cases}
$$

加权融合：
(递推最小二乘)

$$
\begin{cases}
\mathbf{K}' = \mathbf{P}_k \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \\
\hat{\mathbf{x}}'_k = \hat{\mathbf{x}}_k + \mathbf{K}'(\vec{\mathbf{z}}_k - \mathbf{H}_k \hat{\mathbf{x}}_k) \\
\mathbf{P}'_k = \mathbf{P}_k - \mathbf{K}' \mathbf{H}_k \mathbf{P}_k
\end{cases}
$$

4 工程应用实例

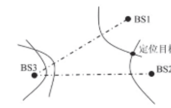
工程应用实例

在时间同步中的应用

在典型的目标定位任务中，往往涉及测距。

典型测距公式 $\Delta d = c*(T1 - T2)$ 看似很简单

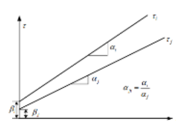
但由于每个基站都有自己独立的时钟源，每个基站的时钟源所用的晶振都存在不同的频率偏差，况且各基站时钟启动时间不同，在执行定位任务前，必须进行各基站间的时钟同步。



工程应用实例

我们使用线性模型来描述时钟间的差异。晶振的频率偏差虽然会随温度呈现非线性变化，但在每一个较短的时间内，我们可以认为**线性模型是可靠的。**

模型： $\tau_i(t) = \alpha_i t + \beta_i$

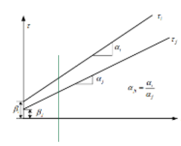


所谓的时钟同步，抽象出来就是建立**相同横坐标**下，两条直线上**点的映射关系。**

工程应用实例

我们知道，想建立两条直线的映射关系，至少要知道直线之间的**相对斜率**，以及至少**一对点**的映射

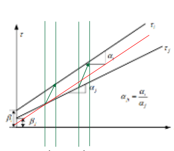
由于各基站分布在空间中，具有一定的距离，我们无法在一个时间点同时获得两个基站的本地时间。所以相对斜率，点的映射，都是**无法直接准确测量。**



工程应用实例

但我们仍然有委曲求全的办法，通过基站间发送**时钟校对包**，分别记录本地时间，来获得含有延时的时间测量数据

我们可以使用二维卡尔曼滤波，估计相对斜率，并估计出一对点的映射，实现时间同步



工程应用实例

预测：

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1}$$

```
x0 = dt*x1 + x0;
x0 = fmod(x0, CLOCK_PERIOD_SEC);
```

$$P_k^- = AP_{k-1}A^T + Q$$

```
p0 = (dt*dt*processNoiseVar) * 0.5 + dt*(dt*p3 + p2)
       + dt*p1 + p0;
p_tmp=ONE_OVER_ROOT_TWO*dt*processNoiseVar
          + dt*p3;
p1 = p_tmp + p1;
p2 = p_tmp + p2;
p3 = processNoiseVar + p3;
```

工程应用实例

更新:

$$\mathbf{K}' = \mathbf{P}_k \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k \mathbf{H}_k^T + \mathbf{R}_k)^{-1}$$

```
k0 = p0 / (measNoiseVar + p0);
k1 = p2 / (measNoiseVar+ p0);
```

$$\hat{\mathbf{x}}'_k = \hat{\mathbf{x}}_k + \mathbf{K}'(\vec{\mathbf{z}_k} - \mathbf{H}_k \hat{\mathbf{x}}_k)$$

```
x0 = k0*measuredError + x0;
x1 = k1*measuredError + x1;
```

$$\mathbf{P}'_k = \mathbf{P}_k - \mathbf{K}'\mathbf{H}_k \mathbf{P}_k$$

```
p3 = p3 - k1*p1;
p2 = p2 - k1*p0;
p1 = p1 - k0*p1;
p0 = p0 - k0*p0;
```


工程应用实例

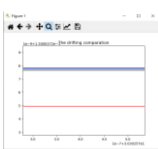
设置初值: x0 = 0.0; x1 = 1.0; p0 = p1 = p2 = p3 = 0.0;

设置噪声: measNoiseVar = 3e-20; processNoiseVar = 5e-20;

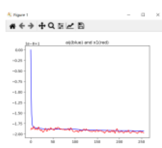
计数初值: counter = 0; outlier_counter = 0;

工程应用实例

实际运行情况:



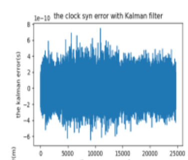
估计斜率（局部）
(平均误差2.599e-09)

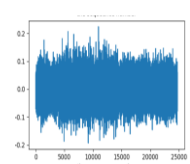


估计相对斜率（整体）

工程应用实例

实际运行情况:





估计映射点的误差
（大时间跨度）

5 内容总结

内容总结

- 卡尔曼滤波是一种**状态估计算法**，它利用**加权**的思想，**融合**状态空间模型的预测值与传感器的测量值，对系统状态做出最优估计。当系统是线性且过程噪声与测量噪声服从高斯分布时，经典卡尔曼滤波有着非常优秀的表现
- 卡尔曼滤波十分强调对“不确定度”的判断与量化，这有助于我们分配融合时的权重
- 卡尔曼滤波可以看作是递推最小二乘（Recursive Least-Square, RLS)的一类特殊应用，因而也具有递推最小二乘的部分性质，例如：**线性，无偏，且误差方差最小**

谢谢大家
请各位老师同学批评指正