

第 3 课 扩展 Kalman 滤波及多传感器融合

作者：范泽宣

你一直在制作的钢铁侠系统遇到了一个棘手的问题，你有多个返回相同状态变量的传感器，你经常听说 kalman 滤波融合，可是怎么利用 kalman 进行多传感器的融合呢，之前你了解的 kalman 知识中并没有涉及。不要着急，这节课将会详细介绍给你。

为了保证在任意高度时钢铁侠系统传感器返回数据具有鲁棒性，你选择了 GPS 和气压计两款传感器用来返回观察到了高度信息。

所以对于 $z_k = Cx_{k-1} + v_k$ 的感测方程，有如下的观测模型：

$$\begin{bmatrix} gps_k \\ \text{气压}_k \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} gps_{k-1} \\ \text{气压}_{k-1} \end{bmatrix}$$

再返回到第一节的内容，依然按照线性 kalman 滤波的方程建立算法程序（对照第一节课程的 5 个方程后以下程序），进行测试后得到了很好的传感器融合的效果。

所以线性传感器的多个融合可以通过这个简单的方法，这就是滤波融合的关键，是不是很简单，关键程序如下(假设 $R = \begin{bmatrix} r1 & 0 \\ 0 & r2 \end{bmatrix}$ ，

每个传感器各接收 500 个观测值):

```
xhat_signal[0] = (z_signal[0][0] + z_signal[1][0]) / 2;
p = 1;
for (k = 1; k < 1000; k++)
{
    xhat_signal[k] = xhat_signal[k-1];
    p += q;
    a = p + r1;
    d = p + r2;
    e = p / (a*d-p*p);
    g1 = e*d - e*p;
    g2 = e*a - e*p;
    p = (1 - (g1+g2)) * p;
    xhat_signal[k] = xhat_signal[k] + g1*(z_signal[0][k]-xhat_signal[k]) +
    g2*(z_signal[1][k]-xhat_signal[k]);
}
```

怎么样，再对照下第一课的公式内容理解下：

Predict:

$$\mathbf{X}_k = \mathbf{a} \mathbf{X}_{k-1}$$

$$\mathbf{P}_k = \mathbf{a}^* \mathbf{P}_{k-1} \mathbf{a}$$

Update:

$$g_k = p_{k-1} / (p_{k-1} + r)$$

$$\hat{x}_k = \hat{x}_{k-1} + g_k (z_k - \hat{x}_{k-1})$$

$$p_k = (1 - g_k) p_{k-1}$$

现在你可以很好的解决线性多传感器融合的问题了吧，传感器融合是 kalman 滤波应用的一个重要领域。在网上搜索 kalman 滤波会有一个经典的案例—加速度计和角速度计融合，很多人不能很好的理解，因为涉及到 EKF 的知识，在本节课的后面会给大家重点讲解一下，首先让我们继续学习下什么是 EKF。

在上一节课中，你和 Dsy 一同推导了动态系统的估计方法，但是实际系统中有诸多的非线性部分。怎么建立非线性系统的滤波算法，前辈们使用简单的思路将线性 kalman 扩展到非线性的领域，这种常规的方法被成为 EKF（扩展 kalman 滤波）。

扩展 kalman 滤波的原理很简单，就是利用泰勒公式将非线性的部分展开成高阶，并去掉高阶项，利用这个线性化技巧将其转化为一个近似的线性滤波问题。

EKF 的优点是不必预先计算标称轨迹，但是它只能在滤波误差及预测误差较小时才能使用。

正如介绍的一样，EKF 是基于线性 kalman 的，同样将第一节介绍的公式改进如下：

Model:

$$\begin{aligned} x_k &= x_{k-1} + w_k \\ z_k &= h(x_{k-1}) + v_k \end{aligned}$$

Predict:

$$\hat{x}_k = \hat{x}_{k-1}$$

$$p_k = p_{k-1} - q$$

Update:

$$g_k = p_k c_k (c_k p_k c_k + r)^{-1}$$

$$\hat{x}_k = \hat{x}_{k-1} + g_k (z_k - h(\hat{x}_k))$$

$$p_k = (1 - g_k c_k) p_{k-1}$$

$h(\hat{x}_k)$ 和 $h(x_{k-1})$ 是非线性部分，为了方便大家理解，这里去掉了状态转移矩阵。

包含状态转移矩阵的 EKF 又是怎样的呢，为了说明 EKF 的问题，我来举个简单的标量系统进行分析。系统的状态方程如果是：

$$X(k) = 0.5X(k-1) + 5 \cos(1.2X) + W(k)$$

观测方程为：

$$Y(k) = \frac{X^3(k)}{10} + V(k)$$

首先要将状态方程一阶线性化，求解状态转移矩阵 $\phi(k)$ 。

$$\phi(k) = \frac{\partial f}{\partial X} = 0.5 - 6 \sin X$$

然后一阶线性化观测矩阵 $H(k)$ ：

$$C(k) = \frac{\partial h}{\partial X} = 0.3X^2(k)$$

同样，如果状态方程和感测矩阵不是一维的，就要用雅克比矩阵来线性化，其实原理是一样的，雅克比矩阵只不过是线性代数中的一种多元函数求导的方法。

总结下 EKF 的处理过程可以归纳为以下几步：

第一步：一阶线性化状态方程，求解状态转移矩阵 $\phi(k)$ ；

第二步：一阶线性化观测矩阵，求解观测矩阵 $C(k)$ ；

第三步：求协方差矩阵预测 $P(k|k-1)$ ：

$$p_k = \phi(k) p_{k-1} \phi(k)^T - q$$

第四步：求 Kalman 滤波增益：

$$g_k = p_k c_k (c_k p_k c_k + r)^{-1}$$

第五步：求状态更新：

$$\hat{x}_k = \hat{x}_{k-1} + g_k (z_k - h(\hat{x}_k))$$

第六步：协方差更新：

$$p_k = (1 - g_k c_k) p_{k-1}$$

以上六步为 EKF 的一个计算周期，也是最经典的 EKF 处理过程。

将公式直接转换成程序就实现了 EKF 的应用，你依然想了解其中的数学推演，这部分内容其实很简单，最关键的推演核心在于通过已知的概率密度函数 $p(X, kT)$ ，获得一个实际的估计算法，EKF 的方案就是通过泰勒级数展开公式线性化获得当前状态向量的估计。Dsy 告诉你，**EKF 总结成一句话就是通过把你的非线性控制模型在雅克比空间的小范围内线性化，然后再利用线性卡尔曼滤波的方法进行滤波。**所以这部分数学方法的推演其实和线性 kalman 的内容很相似，看一下前几节课的内容自行推导吧，很简单。

既然已经学习了 EKF 和多传感器融合的知识，我们来看一个 EKF 传感器数据融合的例子。这部分我就不开脑洞编例子了，直接拿斯坦福大学 Gordon 的一个 6 DOF IMU Sensor Fusion Euler Angles（6 维度的传感器融合）例子。

首先是模型的状态空间如下：

Extended Kalman Filter – IMU Euler Angles

- state space (or better: its mean)

$$x = \begin{pmatrix} \theta_x \\ \theta_y \\ \theta_z \\ \omega_x \\ \omega_y \\ \omega_z \\ b_x \\ b_y \\ b_z \end{pmatrix} \begin{array}{l} \text{Euler angles} \\ \text{angular velocity} \\ \text{gyro bias} \end{array}$$

此次建立系统状态模型，如下：

- linear process model (assumes velocity & bias stay constant)

$$x = \begin{pmatrix} \theta_x \\ \theta_y \\ \theta_z \\ \omega_x \\ \omega_y \\ \omega_z \\ b_x \\ b_y \\ b_z \end{pmatrix} \quad x_{t+1} = Ax_t = \begin{pmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 & -\Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & -\Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & -\Delta t \\ & & & 1 & & & & & \\ & & & & 1 & & & & \\ & & & & & 1 & & & \\ & & & & & & 1 & & \\ & & & & & & & 1 & \\ & & & & & & & & 1 \end{pmatrix} x_t = \begin{pmatrix} \theta_x^{(t)} + \Delta t(\omega_x^{(t)} - b_x^{(t)}) \\ \theta_y^{(t)} + \Delta t(\omega_y^{(t)} - b_y^{(t)}) \\ \theta_z^{(t)} + \Delta t(\omega_z^{(t)} - b_z^{(t)}) \\ \omega_x^{(t)} \\ \omega_y^{(t)} \\ \omega_z^{(t)} \\ b_x^{(t)} \\ b_y^{(t)} \\ b_z^{(t)} \end{pmatrix}$$

接下来确定误差协方差矩阵:

- linear process model – covariance matrix

$$x = \begin{pmatrix} \theta_x \\ \theta_y \\ \theta_z \\ \omega_x \\ \omega_y \\ \omega_z \\ b_x \\ b_y \\ b_z \end{pmatrix} \quad Q_t = \begin{pmatrix} \Delta t^2 & & & & & & & & \\ & \Delta t^2 & & & & & & & \\ & & \Delta t^2 & & & & & & \\ & & & 0.01\Delta t & & & & & \\ & & & & 0.01\Delta t & & & & \\ & & & & & 0.01\Delta t & & & \\ & & & & & & 0.03\Delta t & & \\ & & & & & & & 0.03\Delta t & \\ & & & & & & & & 0.03\Delta t \end{pmatrix}$$

integration error from Taylor truncation

estimated error of angular velocity = const. approximation

estimated error of bias = const. approximation

接下来让我们确立下观测模型:

$$z = \begin{pmatrix} \tilde{\omega}_x \\ \tilde{\omega}_y \\ \tilde{\omega}_z \\ \bar{a}_x \\ \bar{a}_y \\ \bar{a}_z \end{pmatrix} \begin{array}{l} \text{raw gyro} \\ \text{measurements} \\ \\ \text{normalized} \\ \text{accelerometer} \\ \text{vector} \end{array}$$

接下来就是非线性部分了:

- nonlinear measurement model ☹

$$z = \begin{pmatrix} \tilde{\omega}_x \\ \tilde{\omega}_y \\ \tilde{\omega}_z \\ \bar{a}_x \\ \bar{a}_y \\ \bar{a}_z \end{pmatrix} = H(x) = \begin{pmatrix} \omega_x + b_x \\ \omega_y + b_y \\ \omega_z + b_z \\ -\sin(-\theta_z)\cos(-\theta_x) \\ \cos(-\theta_z)\cos(-\theta_x) \\ \cos(-\theta_x) \end{pmatrix}$$

通过观察我们发下，其实本模型中只有观察模型是非线性的。所以之前教给大家的 EKF 处理步骤的第一步就没有啦，直接从第二步开始，求解 $C(k)$:

$$J_H^{(x)} = \begin{pmatrix} \frac{\partial z}{\partial x_1} & \frac{\partial z}{\partial x_2} & \frac{\partial z}{\partial x_3} & \dots & \frac{\partial z}{\partial x_9} \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ \sin(-\theta_z)\sin(-\theta_x) & 0 & -\cos(-\theta_z)\cos(-\theta_x) & 0 & 0 & 0 & 0 & 0 & 0 \\ -\cos(-\theta_z)\sin(-\theta_x) & 0 & -\sin(-\theta_z)\cos(-\theta_x) & 0 & 0 & 0 & 0 & 0 & 0 \\ \cos(-\theta_x) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{matrix} \frac{\partial z_1}{\partial x} \\ \vdots \\ \frac{\partial z_6}{\partial x} \end{matrix}$$

这里的 $J_H^{(x)}$ 就是我们的 $C(k)$ ，其余的部分再按照后续的 EKF 过程进行求解就好了，有了这部分的理论支持，是不是看相似的滤波融合的例子都 so easy 了呢，哈哈，继续加油，下节课 Dsy 还会来教你 EKF 的应用，告诉你他正在做的导弹制导中的 EKF，国防领域的核心算法，是不是很期待，消化了这节课就去看下节课吧，继续加油。

本文参考: Advanced IMU Sensor Fusion with Kalman Filtering(Gordon Wetzstein—Stanford University)