卡尔曼滤波算法的程序实现和推导过程

--- 蒋海林（ QQ: 280586940） ---

卡尔曼滤波算法由匈牙利裔美国数学家鲁道夫·卡尔曼（ Rudolf Emil Kalman ）创立，这个数学家特么牛逼， 1930 年出生，现在还能走能跳，吃啥啥麻麻香，但他的卡尔曼滤波算法已经广泛应用在航空航天，导弹发射，卫星在轨运行等很多高大上的应用中。

让我们一边膜拜一边上菜吧， 下面就是卡尔曼滤波算法的经典程序， 说是经典，因为能正常运行的程序都长得差不多，在此向原作者致敬。看得懂的，帮我纠正文中的错误；不太懂的，也不要急，让我慢慢道来。

最后希望广大朋友转载时，能够保留我的联系方式，一则方便后续讨论共同进步，二则支持奉献支持正能量。

```
void Kalman_Filter(float Gyro,float Accel)///                角速度，加速度
{
    ///  陀螺仪积分角度（先验估计）
        Angle_Final = Angle_Final + (Gyro - Q_bias) * dt;
    ///  先验估计误差协方差的微分
        Pdot[0] = Q_angle - PP[0][1] - PP[1][0];
        Pdot[1] = - PP[1][1];
        Pdot[2] = - PP[1][1];
        Pdot[3] = Q_gyro;
    ///  先验估计误差协方差的积分
        PP[0][0] += Pdot[0] * dt;
        PP[0][1] += Pdot[1] * dt;
        PP[1][0] += Pdot[2] * dt;
        PP[1][1] += Pdot[3] * dt;
    ///  计算角度偏差
        Angle_err = Accel - Angle_Final;
    ///  卡尔曼增益计算
        PCt_0 = C_0 * PP[0][0];
        PCt_1 = C_0 * PP[1][0];

        E = R_angle + C_0 * PCt_0;

        K_0 = PCt_0 / E;
        K_1 = PCt_1 / E;
    ///  后验估计误差协方差计算
        t_0 = PCt_0;
        t_1 = C_0 * PP[0][1];

        PP[0][0] -= K_0 * t_0;
        PP[0][1] -= K_0 * t_1;
        PP[1][0] -= K_1 * t_0;
        PP[1][1] -= K_1 * t_1;

        Angle_Final += K_0 * Angle_err;          ///  后验估计最优角度值
        Q_bias      += K_1 * Angle_err;          ///  更新最优估计值的偏差
        Gyro_Final   = Gyro - Q_bias;            ///  更新最优角速度值
```

}

我们先把卡尔曼滤波的 5 个方程贴上来：

X(k|k-1)=A X(k-1|k-1)+B U(k) …… (1)// 先验估计

P(k|k-1)=A P(k-1|k-1) A ’+Q …… (2)// 协方差矩阵的预测

Kg(k)= P(k|k-1) H ’/ (H P(k|k-1) H ’ + R) …… (3)// 计算卡尔曼增益

X(k|k)= X(k|k-1)+Kg(k) (Z(k) - H X(k|k-1)) …… (4) 通过卡尔曼增益进行修正

P(k|k)= （I-Kg(k) H ） P(k|k-1) …… (5)// 更新协方差阵

这 5 个方程比较抽象，下面我们就来把这 5 个方程和上面的程序对应起来。

一，对于角度来说，我们认为此时的角度可以近似认为是上一时刻的角度值加上上一时刻陀螺仪测得的角速度乘以时间，因为 $d\theta = dt \times \omega$ ，角度微分等于时间的微分乘以角速度。

但是陀螺仪有个静态漂移 （而且还是变化的） ，静态漂移就是静止了没有角速度然后陀螺仪也会输出一个值，这个值肯定是没有意义的，计算时要把它减去。由此我们得到了当前角度的预测值 Angle_Final:

Angle_Final = Angle_Final + (Gyro - Q_bias) * dt;

其中等号左边 Angle_Final 为此时的角度，等号右边 Angle_Final 为上一时刻的角度， Gyro 为陀螺仪测的角速度的值， dt 是两次滤波之间的时间间隔。

#define dt 0.01

这是程序中的定义，同时零漂 Q_bias 也是一个变化的量。

但是就预测来说认为现在的漂移跟上一时刻是相同的即

Q_bias=Q_bias

将两个式子写成矩阵的形式

$$\begin{bmatrix} Angle \\ Q\_bias \end{bmatrix} = \begin{bmatrix} 1 & -dt \\ 0 & 1 \end{bmatrix} \begin{bmatrix} Angle^{\&} \\ Q\_bias^{\&} \end{bmatrix} + \begin{bmatrix} dt \\ 0 \end{bmatrix} Gyro$$

这个式子对应于卡尔曼滤波的第一个式子

X(k|k-1)=A X(k-1|k-1)+B U(k) …… (1)// 先验估计

X(k|k-1) 为 2 维列向量 $\begin{bmatrix} Angle \\ Q\_bias \end{bmatrix}$, A 为 2 维方阵 $\begin{bmatrix} 1 & -dt \\ 0 & 1 \end{bmatrix}$, X(k-1|k-1) 为 2 维列向量 $\begin{bmatrix} Angle^{\&} \\ Q\_bias^{\&} \end{bmatrix}$,

B 为 2 维列向量 $\begin{bmatrix} dt \\ 0 \end{bmatrix}$, U(k) 为 Gyro

二，这里是卡尔曼滤波的第二个式子

接着是预测方差阵的预测值，这里首先要给出两个值， 一个是漂移的噪声，一个是角度值的噪声 ，（所谓噪声就是 数据的方差值 ）

P(k|k-1)=A P(k-1|k-1) A ’+Q

这里的 Q 为向量 $\begin{bmatrix} Angle \\ Q\_bias \end{bmatrix}$ 的协方差矩阵，即 $\begin{bmatrix} cov(Angle,Angle) & cov(Q\_bias,Angle) \\ cov(Angle,Q\_bias) & cov(Q\_bias) \end{bmatrix} \cdot dt$

因为 漂移噪声 和角度噪声 是相互独立的，则 cov(Angle,Q_bias) =0； cov(Q_bias,Angle) =0；

又由性质可知 cov（x，x）=D（x）即方差，所以得到的矩阵如下：

$$Q = \begin{bmatrix} D(Angle) & 0 \\ 0 & D(Q\_bias) \end{bmatrix} \cdot dt$$

这里的两个方差值是开始就给定的常数，程序中定义如下：

#define Q_angle 0.001 //// 角度过程噪声的协方差

#define Q_gyro 0.003 //// 角速度过程噪声的协方差

接着是这一部分 A P(k-1|k-1) A ’，其中的 P（k-1|k-1） 为上一时刻的预测方差阵

卡尔曼滤波的目标就是要让这个预测方差阵最小。

其中 P(k-1|k-1) 设为 $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$，第一式已知 A 为 $\begin{bmatrix} 1 & -dt \\ 0 & 1 \end{bmatrix}$，则 A’为 $\begin{bmatrix} 1 & 0 \\ -dt & 1 \end{bmatrix}$

则计算 A P(k-1|k-1) A ’+Q（就是个矩阵乘法和加法，算算吧）结果如下 ：

$$\begin{bmatrix} a - c\ dt - b\ dt + d\ (dt)^2 + D(Angle)\ dt & b - d\ dt \\ c - d\ dt & d + D(Q\_bias)\ dt \end{bmatrix}$$

由于 $d\ (dt)^2$ 很小为了计算简便忽略不计。

于是得到 $\begin{bmatrix} a - c\ dt - b\ dt + D(Angle)\ dt & b - d\ dt \\ c - d\ dt & d + D(Q\_bias)\ dt \end{bmatrix}$

把整个式子完整的写出来：

$$\begin{bmatrix} a^{\#} & b^{\#} \\ c^{\#} & d^{\#} \end{bmatrix} = \begin{bmatrix} a - c\ dt - b\ dt + D(Angle)\ dt & b - d\ dt \\ c - d\ dt & d + D(Q\_bias)\ dt \end{bmatrix}$$

$$= \begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} -c\ dt - b\ dt + D(Angle)\ dt & -d\ dt \\ -d\ dt & D(Q\_bias)\ dt \end{bmatrix}$$

又由于 $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ 是 $\begin{bmatrix} a^{\#} & b^{\#} \\ c^{\#} & d^{\#} \end{bmatrix}$ 的上一状态的方差阵，所以可以写成程序可以实现的形式：

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} += \begin{bmatrix} (Q\_angle - c - b)\ dt & -d\ dt \\ -d\ dt & Q\_gyro\ dt \end{bmatrix}$$

此过程转化为如下程序：

```
Pdot[0]=Q_angle - PP[0][1] - PP[1][0];
    Pdot[1]= - PP[1][1];
    Pdot[2]= - PP[1][1];
    Pdot[3]=Q_gyro;

    PP[0][0] += Pdot[0] * dt;
    PP[0][1] += Pdot[1] * dt;
    PP[1][0] += Pdot[2] * dt;
```

PP[1][1] += Pdot[3] * dt;

三，这里是卡尔曼滤波的第三个式子

Kg(k)= P(k|k-1) H ’ / (H P(k|k-1) H ’ + R) …… (3)// 计算卡尔曼增益

即计算卡尔曼增益，这是个二维向量设为 $\begin{bmatrix} k_0 \\ k_1 \end{bmatrix}$，

H 叫做观测模型，是用来把真实的状态映射到观测空间中。真实的状态是不能被观测到的。因为测量值只有加速度计的测量值，所以 H 定义如下：

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

另外这里有一个常数 R，程序中的定义如下

#define R_angle 0.5 //// 测量噪声的协方差（即是测量偏差）

如果你把测量噪声方差设置的太高，滤波器将反应较慢，因为它会更加不相信新的测量值。但是如果设置过小，该值可能过冲，并且带来更多的噪声，因为更加相信加速计的测量值。

则第三个式子可写成 ：

$$k_g(k) = \frac{P(k|k-1)H^T}{HP(k|k-1)H^T + R}$$

$$\begin{bmatrix} K_0 \\ K_1 \end{bmatrix} = \frac{\begin{bmatrix} PP_{00} & PP_{01} \\ PP_{10} & PP_{11} \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix}}{\begin{bmatrix} 1 & 0 \end{bmatrix}\begin{bmatrix} PP_{00} & PP_{01} \\ PP_{10} & PP_{11} \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} + R\_angle} = \frac{\begin{bmatrix} PP_{00} \\ PP_{10} \end{bmatrix}}{PP_{00} + R\_angle}$$

分母即程序中的

PCt_0 = C_0 * PP[0][0];

PCt_1 = C_0 * PP[1][0];

分子即程序中的

E = R_angle + C_0 * PCt_0;

于是求出 $\begin{bmatrix} k_0 \\ k_1 \end{bmatrix}$

K_0 = PCt_0 / E;

K_1 = PCt_1 / E;

四，用误差还有卡尔曼增益来修正

X(k|k)= X(k|k-1)+Kg(k) (Z(k) - H X(k|k-1)) …… (4) 通过卡尔曼增益进行修正

这个矩阵带进去就行了 Z（k)=Accel..... 注意这个是加速度计算出来的角度

```
Angle_err = Accel - Angle;
```

对应程序如下

```
Angle    += K_0 * Angle_err;
Q_bias   += K_1 * Angle_err;
```

同时为了 PID 控制还有下次的使用把角速度算出来了

```
Gyro_x   = Gyro - Q_bias;
```

五，更新预测方差阵 ，为下一次调用进行准备 ：

P(k|k)= （I-Kg(k) H ）P(k|k-1) …… (5)// 更新预测方差阵

I 被称为单位矩阵，其定义为：

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

推导过程很简单，矩阵代进去算就行了 ：

$$\begin{bmatrix} PP_{00}^{\#} & PP_{01}^{\#} \\ PP_{10}^{\#} & PP_{11}^{\#} \end{bmatrix} = \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} K_0 \\ K_1 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} \right) \begin{bmatrix} PP_{00}^{\&} & PP_{01}^{\&} \\ PP_{10}^{\&} & PP_{11}^{\&} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} PP_{00}^{\&} & PP_{01}^{\&} \\ PP_{10}^{\&} & PP_{11}^{\&} \end{bmatrix} - \begin{bmatrix} K_0 \\ K_1 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} PP_{00}^{\&} & PP_{01}^{\&} \\ PP_{10}^{\&} & PP_{11}^{\&} \end{bmatrix}$$

$$= \begin{bmatrix} PP_{00}^{\&} & PP_{01}^{\&} \\ PP_{10}^{\&} & PP_{11}^{\&} \end{bmatrix} - \begin{bmatrix} K_0 \\ K_1 \end{bmatrix} \begin{bmatrix} PP_{00}^{\&} & PP_{01}^{\&} \end{bmatrix}$$

$$= \begin{bmatrix} PP_{00}^{\&} & PP_{01}^{\&} \\ PP_{10}^{\&} & PP_{11}^{\&} \end{bmatrix} - \begin{bmatrix} K_0 PP_{00}^{\&} & K_0 PP_{01}^{\&} \\ K_1 PP_{10}^{\&} & K_1 PP_{11}^{\&} \end{bmatrix}$$

又由于 $\begin{bmatrix} PP_{00}^{\&} & PP_{01}^{\&} \\ PP_{10}^{\&} & PP_{11}^{\&} \end{bmatrix}$ 是 $\begin{bmatrix} PP_{00}^{\#} & PP_{01}^{\#} \\ PP_{10}^{\#} & PP_{11}^{\#} \end{bmatrix}$ 的上一状态的方差阵 ，所以可以写成如下比较接近程序实现

方式的形式 ：

$$\begin{bmatrix} PP_{00} & PP_{01} \\ PP_{10} & PP_{11} \end{bmatrix} -= \begin{bmatrix} K_0 PP_{00} & K_0 PP_{01} \\ K_1 PP_{10} & K_1 PP_{11} \end{bmatrix}$$

转化成如下程序 ：

```
t_0 = PCt_0;
t_1 = C_0 * PP[0][1];
PP[0][0] -= K_0 * t_0;
PP[0][1] -= K_0 * t_1;
PP[1][0] -= K_1 * t_0;
PP[1][1] -= K_1 * t_1;
```

六，初始值

卡尔曼滤波算法的推导过程已经完成了，如果再回头去看一遍程序，相信大家会有一个全新的认识。卡尔曼算法从理论上讲是迭代的算法，也是最优算法，需要给很少的初始值：

```
#define   Q_angle 0.001          ////   角度过程噪声的协方差
#define   Q_gyro 0.003           ////   角速度过程噪声的协方差
#define   R_angle 0.5            ////   测量噪声的协方差（即是测量偏差）
#define   dt 0.01          ////   卡尔曼滤波采样频率
#define   C_0 1
void  Kalman_Filter_Init(        void )
{
        Q_bias      = 0;
        PP[0][0]            = 1;
        PP[0][1]            = 0;
        PP[1][0]            = 0;
        PP[1][1]            = 1;
        Angle_Final         = 0;
        Gyro_Final          = 0;
}
```

七，展望

如果卡尔曼滤波是稳定的，随着滤波的推进，卡尔曼滤波估计的精度应该越来越高，滤波误差方差阵也应趋于稳定值或有界值。但在实际应用中，随着测量值数目的增加，由于估计误差的均值和估计误差协方差可能越来越大，使滤波逐渐失去准确估计的作用，这种现象称为卡尔曼滤波发散。

针对卡尔曼滤波发散的原因，目前已经出现了几种有效抑制滤波发散的方法，常用的有衰减记忆滤波、限定记忆滤波、扩充状态滤波、有限下界滤波、平方根滤波、和自适应滤波等。这些方法本质上都是以牺牲滤波器的最优性为代价来抑制滤波发散，也就是说，多数都是次优滤波方法。

另外，最初提出的卡尔曼滤波基本理论只适用于状态方程和量测方程均为线性的随机线性高斯系统。但是大部分系统是非线性的，其中还有许多是强非线性的。非线性估计的核心就在于近似，给出非线性估计方法的不同就在于其近似处理的思想和实现手段不同。近似的本质就是对难以计算的非线性模型施加某种数学变换，变换成线性模型，然后用 Bayes 估计原理进行估计。进一步说，非线性变换到线性变换主要有两种实现手段，一种是 Taylor 多项式展开，一种是插值多项式展开。由此又衍生出离散非线性随机系统扩展卡尔曼滤波 (Extended kalman filter, 简称 EKF)，无迹卡尔曼滤波（ Unscented kalman filter, 简称 UKF），分开差分滤波器（ Divided Difference Filter ，简称 DDF