

卡尔曼滤波 C 代码分析

czqtqwan

2015-11-23

(部分内容来源于网络，转载请注明出处)

目录

一、卡尔曼滤波五个基本的公式.....	3
1.预测阶段.....	3
2.校正阶段.....	3
二、代码详解.....	3
2.1 定义变量.....	3
2.2 第一个公式.....	4
2.3 第二个公式.....	4
2.4 第三个公式.....	5
2.5 第四个公式.....	6
2.6 第五个公式.....	6
三、完整代码.....	7

卡尔曼滤波 c 代码分析

卡尔曼滤波的思想其实很简单，就是根据方差实现的一种最优估计方法。

一、卡尔曼滤波五个基本的公式

1. 预测阶段

1.1 $X(k|k-1)=AX(k-1|k-1)+BU(k)$先验估计

A, B 为矩阵系数, $X(k|k-1)$ 是根据上一状态预测的结果, $X(k-1|k-1)$ 是上一次的最优结果, $U(k)$ 为现在状态的控制量。

1.2 $P(k|k-1)=A P(k-1|k-1) A^T+Q$误差协方差

$P(k|k-1)$ 是 $X(k|k-1)$ 的协方差, $P(k-1|k-1)$ 是 $X(k-1|k-1)$ 的协方差, Q 是估计过程的误差协方差。

2. 校正阶段

2.1 $Kg(k)= P(k|k-1) H^T / (H P(k|k-1) H^T + R)$计算卡尔曼增益

H 也为系数矩阵, R 为测量值的噪声协方差。

2.2 $X(k|k)= X(k|k-1) + Kg(k)(Z(k) - H X(k|k-1))$修正估计

$Z(K)$ 是测量值。

2.3 $P(k|k)=(I-Kg(k) H) P(k|k-1)$更新误差协方差

I 是为 1 的矩阵, 对于单模型单测量 I 为 1。

二、代码详解

看代码的实现过程要更清晰有逻辑些, 以下是一个平衡车的卡尔曼滤波代码。

2.1 定义变量

先是一系列变量的定义, 和五个公式中的系数是对应的, 还有一些计算过程的中间变量:

```
float Q_angle=0.001; //陀螺仪噪声的协方差（估计过程的误差协方差）
float Q_gyro=0.003;  //陀螺仪漂移噪声的协方差（估计过程的误差协方差）
```

```
float R_angle=0.5;           //加速度计测量噪声的协方差
float dt=0.005;              //积分时间，dt 为滤波器采样时间(秒)
char C_0 = 1;                //H 矩阵的一个数
float Q_bias=0, Angle_err=0; //Q_bias 为陀螺仪漂移
float PCt_0=0, PCt_1=0, E=0; //中间变量
float K_0=0, K_1=0, t_0=0, t_1=0; //K 是卡尔曼增益，t 是中间变量
float Pdot[4] = {0,0,0,0};    //计算 P 矩阵的中间变量
float PP[2][2] = { { 1, 0 }, { 0, 1 } }; //公式中 P 矩阵，X 的协方差
```

2.2 第一个公式

以下代码对应第一个公式 $X(k|k-1)=AX(k-1|k-1)+BU(k)$

```
void Kalman_Filter(float Gyro,float Accel)//Gyro 陀螺仪的测量值，Accel 加速度计的角度计
算值
{
    Angle+=(Gyro - Q_bias)*dt;
    //角度测量模型方程，角度估计值=上一次的最优角度+(角速度-上一次的最优零漂)*dt
    //就漂移来说认为每次都是相同的 Q_bias=Q_bias;
```

由此得到矩阵：

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

状态估计矩阵

这里要估计的 X 有两个一个角度，一个是陀螺仪的零漂，把它们写成矩阵的形式：

$$X = \begin{bmatrix} Angle \\ Q_bias \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & -dt \\ 0 & 1 \end{bmatrix}$$

2.3 第二个公式

以下代码对应第二个公式 $P(k|k-1)=AP(k-1|k-1)A^T+Q$

公式中 Q 为向量 A 的协方差矩阵

$$\begin{bmatrix} \text{cov}(Angle, Angle) & \text{cov}(Q_bias, Angle) \\ \text{cov}(Angle, Q_bias) & \text{cov}(Q_bias, Q_bias) \end{bmatrix}$$

协方差矩阵

因为陀螺仪漂移噪声和角度噪声是相互独立的，所以 $\text{cov}(Angle, Q_bias)=0$ ， $\text{cov}(Q_bias, Angle)=0$ ，又因为 $\text{cov}(x, x)=D(x)$ ，方差值 Q_angle ， Q_gyro 程序开头定义已给出，所以：

$$Q = \begin{bmatrix} Q_Angle & 0 \\ 0 & Q_gyro \end{bmatrix}$$

```
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;
```

卡尔曼滤波的目标就是使 $P(k-1|k-1)$ 值最小，设：

$$P = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

代入公式 $P(k|k-1) = A P(k-1|k-1) A^T + Q$ 计算得：

$$P = \begin{bmatrix} a - c \times dt - b \times dt + d \times (dt)^2 & b - d \times dt \\ c - d \times dt & d \end{bmatrix} + Q$$

dt 的平方值太小，所以 $d \times (dt)^2$ 可以忽略掉：

$$P = \begin{bmatrix} a - c \times dt - b \times dt & b - d \times dt \\ c - d \times dt & d \end{bmatrix} + Q$$

现在来看代码，是不是和公式很像？代码中的 $Pdot$ 为矩阵计算得中间变量， PP 即是公式中的 P ，对应 a, b, c, d 。仔细的话，可以发现代码和公式有一点点不同，就是 Q_angle ， Q_gyro 都乘上了 dt ，但是 Q_angle ， Q_gyro 是固定值， dt 也是固定值，所以它们相乘也还是一个固定的值，只要 Q_angle ， Q_gyro 初始设置合适就没有影响了。

2.4 第三个公式

以下代码对应第三个公式 $Kg(k) = P(k|k-1) H^T / (H P(k|k-1) H^T + R)$

Kg 是一个二维向量，对应 $Angle$ 和 Q_bias 的卡尔曼增益：

$$Kg = \begin{bmatrix} K_0 \\ K_1 \end{bmatrix}$$

H 是一个系数矩阵：

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

为什么是这个值呢？这就需要知道 H 的定义了，卡尔曼滤波的推导公式中有一个量测方程：

$$z_k = Hx_k + v_k$$

Z 为观测变量,也就是程序中输入的角度测量值 Accel,可见 X 矩阵中的 Angle 和 Accel 是直接相关的,而 Q_bias 则和 Accel 无关,所以 $H = \begin{bmatrix} 1 & 0 \end{bmatrix}$ 。

R 即为加速度计测量出角度值的噪声。

变量都弄明白意思了,接下来就是计算了,这里还是把 PP 当成 a,b,c,d。

$$P \times H^T = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \times \begin{bmatrix} 1 & 0 \end{bmatrix}^T = \begin{bmatrix} a & c \end{bmatrix}, \text{ 对应 PCt_0 和 PCt_1}$$

$$H \times P \times H^T = \begin{bmatrix} 1 & 0 \end{bmatrix} \times \begin{bmatrix} a & b \\ c & d \end{bmatrix} \times \begin{bmatrix} 1 & 0 \end{bmatrix}^T = a$$

```
PCt_0 = C_0 * PP[0][0];           //矩阵乘法的中间变量
PCt_1 = C_0 * PP[1][0];           //C_0=1
E = R_angle + C_0 * PCt_0;        //分母
//卡尔曼增益,两个,一个是 Angle 的,一个是 Q_bias 的
K_0 = PCt_0 / E;
K_1 = PCt_1 / E;
```

2.5 第四个公式

以下对应第四个公式 $X(k|k) = X(k|k-1) + Kg(k)(Z(k) - H X(k|k-1))$

这部分比较简单,很容易理解,代入相应值即可。这个程序是可以算出最优角速度值的,不过一般用不到。

```
Angle_err = Accel - Angle;
Angle += K_0 * Angle_err;
Q_bias += K_1 * Angle_err;
//Gyro_x = Gyro - Q_bias;          //计算得最优角速度值
```

2.6 第五个公式

以下对应第五个公式 $P(k|k) = (I - Kg(k)H)P(k|k-1)$

I 是全 1 的矩阵:

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

所以,第五个公式可以转化成以下形式:

$$P(k|k) = (P(k|k-1) - Kg(k)H)P(k|k-1)$$

$$Kg(k) \times (H \times P(k|k-1)) = \begin{bmatrix} K_0 \\ K_1 \end{bmatrix} \times \left(\begin{bmatrix} 1 & 0 \end{bmatrix} \times \begin{bmatrix} a & b \\ c & d \end{bmatrix} \right) = \begin{bmatrix} K_0 \times a & K_0 \times b \\ K_1 \times a & K_1 \times b \end{bmatrix}$$

```
t_0 = PCt_0;           //矩阵计算中间变量,相当于 a
t_1 = C_0 * PP[0][1];   //矩阵计算中间变量,相当于 b
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;
}
```

到此，整个代码就分析完毕了，虽然这段代码不多，但是还是涉及到了许多的数学公式，有很多值得学习和借鉴的地方。

三、完整代码

```
float Angle=0.0;           //卡尔曼滤波器的输出值，最优估计的角度
//float Gyro_x=0.0;        //卡尔曼滤波器的输出值，最优估计的角速度
float Q_angle=0.001;       //陀螺仪噪声的协方差（估计过程的误差协方差）
float Q_gyro=0.003;        //陀螺仪漂移噪声的协方差（估计过程的误差协方差）
float R_angle=0.5;         //加速度计测量噪声的协方差
float dt=0.005;            //积分时间，dt 为滤波器采样时间(秒)
char C_0 = 1;              //H 矩阵的一个数
float Q_bias=0, Angle_err=0; //Q_bias 为陀螺仪漂移
float PCt_0=0, PCt_1=0, E=0; //中间变量
float K_0=0, K_1=0, t_0=0, t_1=0; //K 是卡尔曼增益，t 是中间变量
float Pdot[4] = {0,0,0,0};    //计算 P 矩阵的中间变量
float PP[2][2] = { { 1, 0 }, { 0, 1 } }; //公式中 P 矩阵，X 的协方差

void Kalman_Filter(float Gyro,float Accel)//Gyro 陀螺仪的测量值，Accel 加速度计的角度计算值
{
    Angle += (Gyro - Q_bias)*dt;
    //角度测量模型方程，角度估计值=上一次的最优角度+(角速度-上一次的最优零漂)*dt
    //就漂移来说认为每次都是相同的 Q_bias=Q_bias

    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;

    PCt_0 = C_0 * PP[0][0]; //矩阵乘法的中间变量
    PCt_1 = C_0 * PP[1][0]; //C_0=1
    E = R_angle + C_0 * PCt_0; //分母
    K_0 = PCt_0 / E; //卡尔曼增益，两个，一个是 Angle 的，一个是 Q_bias 的
    K_1 = PCt_1 / E;

    Angle_err = Accel - Angle;
    Angle += K_0 * Angle_err; //计算最优角度
    Q_bias += K_1 * Angle_err; //计算最优零漂
    //Gyro_x = Gyro - Q_bias; //计算得最优角速度

    t_0 = PCt_0; //矩阵计算中间变量，相当于 a
    t_1 = C_0 * PP[0][1]; //矩阵计算中间变量，相当于 b
}
```

```
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;  
}
```