状态方程：

测量方程：

：系统输入量

1. B：系统参数矩阵

H：观测矩阵

：过程噪声矩阵，W~N(0,Q)，Q为过程噪声协方差矩阵

：观测噪声矩阵，V~N(0,R)，R为观测噪声协方差矩阵

## 一、根据状态方程计算先验估计

用陀螺仪计算的角度做估计值：

Q\_bias：陀螺仪静态漂移，是一个变化的量，但是认为预测时现在的漂移跟上一时刻是相同的。

上面两个方程写成矩阵的形式，即得到**状态方程**：

用加速度分解得到的角度做测量值：，写成矩阵形式，得到**测量方程：**

将状态方程中的A、B代入上面红色公式，在程序中算出，而预测时认为不变。

## 二、计算先验估计误差协方差矩阵

1.先分析过程噪声的协方差矩阵Q

协方差知识：衡量两个变量的总体误差

当相互独立时，

当时，协方差又称方差：

相互独立，所以

在程序中，

2.计算先验估计误差协方差矩阵，在程序中，代入上方红色公式：

这一项比较小，计算时忽略，然后就是在程序中代公式了，计算出的四个值。

## 三、计算卡尔曼增益

由测量方程知：，先算分母

整体

在程序中观测噪声，代公式计算出和。

## 四、更新后验估计

先算括号里的：

整体

在程序中算出和

## 五、更新后验估计误差协方差矩阵

为单位矩阵，代公式=

在程序中代公式更新的四个值。

## 程序设计

/\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*@brief 获取x轴角度简易卡尔曼滤波

\*@param Accel：加速度计获取的角度、Gyro：陀螺仪获取的角速度

\*@retval x轴角度

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

float dt=0.005; //每5ms进行一次滤波

float Kalman\_Filter\_x(float Accel,float Gyro)

{

// static float angle\_dot; 角度微分=角速度，本系统用不到

static float angle;

float Q\_angle=0.001; // 过程噪声的协方差

float Q\_gyro=0.003; //过程噪声的协方差

float R\_angle=0.5; //测量噪声的协方差,值越大表示越不信任测量值

char C\_0 = 1; //在此程序显得多余

static float Q\_bias, Angle\_err; //陀螺仪静态漂移

static float PCt\_0, PCt\_1, E; //E:卡尔曼增益分母

static float K\_0, K\_1, t\_0, t\_1;

static float Pdot[4] ={0,0,0,0};

static float PP[2][2] = { { 1, 0 },{ 0, 1 } }; //PP为协方差矩阵Pk-的先验估计值

/\*\*\*\*\*\*\*\*\*\*\*\*\* 角度先验估计 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*///Xk = A Xk-1 + B Uk-1

angle += (Gyro - Q\_bias) \* dt;

/\*\*\*\*\*\*\*\*\*\*\*\*\* 先验估计误差协方差 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*///Pk- =A Pk-1 AT + Q

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;

/\*\*\*\*\*\*\*\*\*\*\*\*\* 计算卡尔曼增益 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*///Kk = Pk- HT / (H Pk- HT + R)

PCt\_0 = C\_0 \* PP[0][0]; //显得多余

PCt\_1 = C\_0 \* PP[1][0]; //显得多余

E = R\_angle + C\_0 \* PCt\_0; //求卡尔曼增益分母，直接E = R\_angle + PP[0][0]就行

K\_0 = PCt\_0 / E; //计算卡尔曼增益，为两行一列的矩阵

K\_1 = PCt\_1 / E;

/\*\*\*\*\*\*\*\*\*\*\*\*\* 角度后验估计 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*///Xk = Xk- + Kk (Zk - H Xk-)

Angle\_err = Accel - angle; //(Zk - H Xk-)

angle += K\_0 \* Angle\_err; //后验估计

Q\_bias += K\_1 \* Angle\_err; //后验估计

// angle\_dot = Gyro - Q\_bias; //输出值(后验估计)的微分=角速度

/\*\*\*\*\*\*\*\*\*\*\*\*\* 更新协方差矩阵 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*///Pk = (I - Kk H) Pk-

t\_0 = PCt\_0;

t\_1 = C\_0 \* PP[0][1];

PP[0][0] -= K\_0 \* t\_0; //后验估计误差协方差

PP[0][1] -= K\_0 \* t\_1;

PP[1][0] -= K\_1 \* t\_0;

PP[1][1] -= K\_1 \* t\_1;

return angle;

}