基于四元数解算陀螺仪姿态角算法的实现

关越魏[1]，何波贤[1]，于仁清[1]，丁广威[2]
（1. 92724部队，山东 青岛 266108；2. 92106部队，山东 青岛 266108）

摘　要：采用四元数将三轴陀螺仪以及三轴加速度计的数据进行互补融合，讨论了四元数解算的设计过程，给出了相应步骤的详细代码。实验证明，该算法有效抑制陀螺仪的误差发散，并提高了系统的测量精度。

关键词：四元数；姿态解算；叉乘；微处理器

DOI:10.16184/j.cnki.comprg.2015.09.013

1　引言

基于陀螺仪的姿态角测量目前大量应用于汽车、移动电子设备、无人机等领域，尤其是手机上的应用，比如GPS导航、感应控制、体感互动、拍照稳定等功能的实现。目前对陀螺仪的姿态角解算方法有3种，欧拉角法由于存在“奇点”现象而不能应用于全姿态跟踪，并难以在工程实践中广泛应用；且方向余弦法方程的计算量大，工作效率低；而四元数方法比较难理解，但是计算量简单并能够实时解算姿态角，因此得到广泛的应用。

基于四元数的姿态解算方法能够有效结合陀螺以及加速度计的误差特性，将运动场以及重力加速度两个互不相干的物理矢量进行互补融合。主要利用陀螺仪测量的角速度作为四元数的更新，以重力加速度作为四元数的观测，通过8位微处理器实时解算姿态角。

2　硬件结构

MPU-6050是集成三轴陀螺、三轴加速度计、温度传感器为一体的组合传感器，其中主控采用AVR单片机，传感器数据主要通过I2C接口形式传输到单片机上进行实时处理，系统结构框图如图1所示。

图1　硬件结构框图

3　基于四元数的姿态解算

四元数基本思路为：一个坐标系到另一个坐标系的变换可以通过绕一个定义在参考坐标系中的矢量μ的单次转动来实现，其中矢量μ既代表了旋转轴的长度，同时也代表旋转角，所以它的表达式为：

$$Q(q_0+q_1+q_2+q_3)=q_0+q_1i+q_2j+q_3k \tag{1}$$

并且四元数的旋转矩阵为：

$$C_b^R=\begin{bmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2-q_3q_0) & 2(q_1q_3+q_0q_2) \\ 2(q_1q_2+q_0q_3) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_2q_3+q_0q_1) & q_0^2-q_1^2-q_2^2+q_3^2 \end{bmatrix} \tag{2}$$

四元数的解算姿态角的流程如图2所示。

图2　四元数解算流程

3.1　四元数初始化

利用Atmega8单片机在采样周期内连续采集MPU6050传感器的10组数据，然后求和取平均值，作为四元数的输入，四元数的初始化主要是对误差积分以及全局四元数进行赋值，代码如下：

```
volatile float exInt, eyInt, ezInt;  // 误差积分
volatile float q0, q1, q2, q3; // 全局四元数
q0 = 1.0f , q1 = 0.0f , q2 = 0.0f , q3 = 0.0f ;  //初始化四
//元数
exInt = 0.0 ,eyInt = 0.0, ezInt = 0.0;  // 误差积分
```

3.2　修正角速度误差

三轴重力加速度计由于噪声比较大，而且在飞行过程中，受机体振动影响比三轴陀螺仪明显，短时间内的可靠性不高。陀螺仪噪声小，但是由于积分是离散的，长时间的积分会出现漂移的情况，因此需要将用重力加速度计求得的姿态来矫正陀螺仪积分姿态的漂移。

根据余弦矩阵和欧拉角的定义以及地理坐标系的重力向量，设经旋转至机体坐标系上的重力加速度为：

$$g_b=C_b^g\cdot g=\begin{bmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2-q_3q_0) & 2(q_1q_3+q_0q_2) \\ 2(q_1q_2+q_0q_3) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_2q_3+q_0q_1) & q_0^2-q_1^2-q_2^2+q_3^2 \end{bmatrix}\begin{bmatrix}0\\0\\1\end{bmatrix} \tag{3}$$

简化有：$g_b=\begin{pmatrix} g_{bx} \\ g_{by} \\ g_{bz} \end{pmatrix}=\begin{pmatrix} 2(q_1q_3-q_0q_2) \\ 2(q_2q_3+q_0q_1) \\ q_0^2-q_1^2-q_2^2+q_3^2 \end{pmatrix}$

然后把机体测得的加速度值$a_b=(a_{bx}\ a_{by}\ a_{bz})$作归一化处理。

加速度计测量的重力向量为$a_b=(a_{bx}\ a_{by}\ a_{bz})$；经陀螺积分后的姿态来推算出的重力向量是$v_x\ v_y\ v_z$；它们之间的误差向量就是陀螺积分后的姿态和加速度计测量的姿态之间的误差。并且根据向量间的误差，可以用向量积（也叫外积、叉乘）来表示，$e_x\ e_y\ e_z$就是两个重力向量的叉积。这个叉积向量仍旧是位于机体坐标系上的，而陀螺积分误差也是在机体坐标系上，而且叉积的大小与陀螺积分误差成正比，正好用来纠正陀螺。由于陀螺是对机体直接积分，所以对陀螺的纠正量会

作者简介：关越魏（1986-），男，助理讲师，研究方向：计算机仿真；何波贤，硕士，研究方向：计算机仿真。

收稿日期：2015-01-19

直接体现在对机体坐标系的纠正，所以有：

$$e = a_b \times g_b = \begin{vmatrix} x & y & z \\ a_{bx} & a_{by} & a_{bz} \\ g_{bx} & g_{by} & g_{bz} \end{vmatrix} \tag{4}$$

其中：x，y，z 是机体坐标系上的单位向量，经简化得：

$$e = \begin{pmatrix} e_x \\ e_y \\ e_z \end{pmatrix} = \begin{pmatrix} a_{by} \cdot g_{bz} - a_{bz} \cdot g_{by} \\ a_{bz} \cdot g_{bx} - a_{bx} \cdot g_{bz} \\ a_{bx} \cdot g_{by} - a_{by} \cdot g_{bx} \end{pmatrix}$$

这部分代码为：

```
// 测量正常化
 norm = sqrt(ax*ax + ay*ay + az*az);
 ax = ax / norm;
 ay = ay / norm;
 az = az / norm;
 // 估计方向的重力
 vx = 2*(q1*q3 - q0*q2);
 vy = 2*(q0*q1 + q2*q3);
 vz = q0*q0 - q1*q1 - q2*q2 + q3*q3;
 // 经叉乘积并求出误差
 ex = (ay*vz - az*vy);
 ey = (az*vx - ax*vz);
 ez = (ax*vy - ay*vx);
```

3.3 数据融合

把上述求解出来的叉乘误差当作 PI 修正陀螺零偏，通过调节 Kp，Ki 两个参数，可以控制加速度计修正陀螺仪积分姿态的速度，已到达数据融合的目的。经过实验 Ki=2.0f，Kp=0.005f 收敛的效果比较好，这部分代码如下：

```
// 积分误差比例积分增益
 exInt = exInt + ex*Ki;
 eyInt = eyInt + ey*Ki;
 ezInt = ezInt + ez*Ki;
 // 调整后的陀螺仪测量
 gx = gx + Kp*ex + exInt;
 gy = gy + Kp*ey + eyInt;
 gz = gz + Kp*ez + ezInt;
```

3.4 四元数更新

四元数的更新本质就是求解四元数的微分方程，根据四元数的微分方程定义，对应的矩阵形式为：

$$\begin{bmatrix} \dot{q}_0(t) \\ \dot{q}_1(t) \\ \dot{q}_2(t) \\ \dot{q}_3(t) \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} \begin{bmatrix} q_0(t) \\ q_1(t) \\ q_2(t) \\ q_3(t) \end{bmatrix} \tag{5}$$

对于四元数的微分方程求解方法一般有很多种，比如龙格库塔、牛顿法、毕卡算法等，为了满足精度以及单片机的处理速度的需求，采用一阶龙格库塔求解四元数的微分方程：

$$Q(t+h) = Q(t) + hk_1$$

$$k_1 = \frac{1}{2}\omega(t) * Q(t)$$

把一阶龙格库塔代入方程（5）最后得出四元数的微分方程为：

$$\begin{bmatrix} q_0(t+h) \\ q_1(t+h) \\ q_2(t+h) \\ q_3(t+h) \end{bmatrix} = \begin{bmatrix} q_0(t) \\ q_1(t) \\ q_2(t) \\ q_3(t) \end{bmatrix} \begin{bmatrix} 1 & -\frac{\omega_x}{2}h & -\frac{\omega_y}{2}h & -\frac{\omega_z}{2}h \\ \frac{\omega_x}{2}h & 1 & \frac{\omega_z}{2}h & -\frac{\omega_y}{2}h \\ \frac{\omega_y}{2}h & -\frac{\omega_z}{2}h & 1 & \frac{\omega_x}{2}h \\ \frac{\omega_z}{2}h & \frac{\omega_y}{2}h & -\frac{\omega_x}{2}h & 1 \end{bmatrix} \tag{6}$$

其中 ω_x ω_y ω_z 为三轴输出的实际角速度。

这部分代码为：

```
#define halfT 0.5f          // 采样周期的一半
//四元数更新
q0 = q0 + (-q1*gx - q2*gy - q3*gz)*halfT;
q1 = q1 + (q0*gx + q2*gz - q3*gy)*halfT;
q2 = q2 + (q0*gy - q1*gz + q3*gx)*halfT;
q3 = q3 + (q0*gz + q1*gy - q2*gx)*halfT;
```

3.5 规范化处理

使用 8 位微处理器实时计算四元数的更新，存在误差等因素，会导致在计算过程中会逐渐失去规范化特性，因此每一次四元数更新之后，必须对四元数进行规范化处理，其中规范化方程为：

$$q_i = \frac{q_i}{\sqrt{\hat{q}_0^2 + \hat{q}_1^2 + \hat{q}_2^2 + \hat{q}_3^2}} \tag{7}$$

其中 i=0,1,2,3 $\hat{q}_0$、$\hat{q}_1$、$\hat{q}_2$、$\hat{q}_3$ 为四元数更新所得值。

这部分代码如下：

```
// 四元数规范化
 norm = sqrt(q0*q0 + q1*q1 + q2*q2 + q3*q3);
 q0 = q0 / norm;
 q1 = q1 / norm;
 q2 = q2 / norm;
 q3 = q3 / norm;
```

3.6 欧拉角转换

最后根据四元数转换成欧拉角的方程进行转化：

俯仰角：$pitch = -\arcsin(2(q_1q_3 - q_0q_2))$ (8)

横滚角：$roll = \arctan\left(\frac{2(q_0q_1 + q_2q_3)}{1 - 2(q_1^2 + q_2^2)}\right)$ (9)

方位角：$yew = -\arctan\left(\frac{2(q_1q_2 + q_0q_3)}{1 - 2(q_1^2 + q_2^2)}\right)$ (10)

这部分代码如下：

```
yaw = -atan2(2 * q[1] * q[2] + 2 * q[0] * q[3], -2 * q[2]*q[2] - 2 * q[3] * q[3] + 1)* 180/M_PI;
pitch= -asin(-2 * q[1] * q[3] + 2 * q[0] * q[2])* 180/M_PI;
roll= atan2(2 * q[2] * q[3] + 2 * q[0] * q[1], -2 * q[1] * q[1] - 2 * q[2] * q[2] + 1)* 180/M_PI;
```

4 实验验证

实验主要利用三轴摇摆台进行测试四元数算法的精度，首先把三轴陀螺系统安装在三轴摇摆台的内框上，然后转动转台，转台上由 1024 光栅码盘计算出转台转过的角度，采集

的数据经过串口传到上位机，然后对三轴陀螺以及对应转轴的码盘转换出来的角度值进行对比。因为三轴加速度计只能融合俯仰角以及横滚角，所以实验仅对俯仰角、横滚角进行对比，如图 3、图 4 所示。

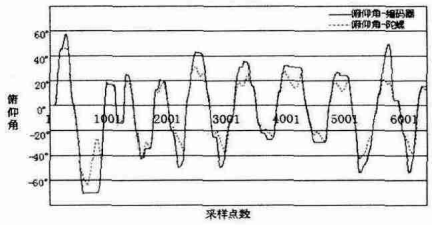

图 3　俯仰角

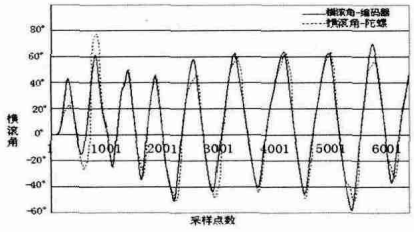

图 4　横滚角

通过上述数据对比，该四元数融合算法跟随性能比较好，能够有效地抑制因陀螺系统随机漂移误差，并能解算出陀螺的姿态角。

5　结语

基于四元数的解算方法，利用叉乘有效地把三轴陀螺以及三轴加速度计的数据进行融合，使得测量的俯仰角、横滚角逼近真角度，经过试验验证了该算法的有效性，且计算量少，在姿态控制领域有这良好的应用前景。

参考文献

[1] 张荣辉，贾宏光，陈涛. 基于四元数法的捷联式惯性导航系统的姿态解算. 光学精密工程，2008，(10)：1963-1968.

[2] 邓正隆. 惯性技术. 哈尔滨工业大学出版社，2006.

[3] 付梦印. Kalman 滤波理论及其在导航系统中的应用. 科学出版社，2010.

[4] 马艳红，胡军. 姿态四元数相关问题. 空间控制技术与应用，2008，(03)：55-60.

(上接第 11 页)

```
        {
    case DataControlRowType.Header://如果创建的是标题行
            TableCellCollection tcHeader = e.Row.Cells;
             //创建变量,以便通过该变量操作当前行表头的单
//元格集合;
             tcHeader.Clear();//清空单元格集合
            /* 第一行表头 */
            tcHeader.Add(new TableHeaderCell());
            //给单元格集合添加一个 TableHeaderCell 类型的
//表头单元格
             tcHeader [0].Attributes.Add ("bgcolor", "Medi-
umPurple");
            //设置此表头单元格的背景颜色为中度紫
            tcHeader[0].Attributes.Add("colspan", "5");
            //设置此表头单元格跨越 5 列
            tcHeader[0].Text = "全部信息</th></tr><tr>";
            //设置此表头单元格中显示的文字,并结束此行
            /* 第二行表头 */
            tcHeader.Add(new TableHeaderCell());
      tcHeader[1].Attributes.Add("bgcolor", "LightSteelBlue");
            tcHeader[1].Attributes.Add("colspan", "2");
            tcHeader[1].Text = "基本信息";
            tcHeader.Add(new TableHeaderCell());
            tcHeader [2].Attributes.Add ("bgcolor", "DarkSea-
Green");
            tcHeader[2].Attributes.Add("colspan", "2");
            tcHeader[2].Text = "联系方式";
            tcHeader.Add(new TableHeaderCell());
             tcHeader[3].Attributes.Add("bgcolor", "LightSteel-
Blue");
            tcHeader[3].Text = "政治面貌</th></tr><tr>";
            /* 第三行表头 */
            tcHeader.Add(new TableHeaderCell());
            tcHeader[4].Attributes.Add("bgcolor", "Khaki");
            tcHeader[4].Text = "编号";
            tcHeader.Add(new TableHeaderCell());
            tcHeader[5].Attributes.Add("bgcolor", "Khaki");
            tcHeader[5].Text = "姓名";
            tcHeader.Add(new TableHeaderCell());
            tcHeader[6].Attributes.Add("bgcolor", "Khaki");
            tcHeader[6].Text = "电话号码";
            tcHeader.Add(new TableHeaderCell());
            tcHeader[7].Attributes.Add("bgcolor", "Khaki");
            tcHeader[7].Text = "家庭住址";
            tcHeader.Add(new TableHeaderCell());
            tcHeader[8].Attributes.Add("bgcolor", "Khaki");
            tcHeader[8].Text = "政治面貌";
            break;
    }
```

4　结语

基于 GridView 控件实现显示/隐藏列及多重表头的方法在 Web 应用程序开发中具有实际意义和应用价值。文中给出的程序代码已在 Vistual Studio+SQL Server 开发环境中验证通过，并实现了上述功能。

参考文献

[1] 陈印. GridView 中数据行批量删除的实现 [J]. 四川职业技术学院学报，2014，24 (1).