



武汉大学测绘学院
SCHOOL OF GEODESY AND GEOMATICS

《惯性导航原理》实验报告

实验 1: IMU误差标定

姓名: 明佳浩

学号: 2023302143030

年级: 2023级

班级: 智能导航二班

教师: 朱锋、张万威、孙骁

二〇二五年十月

目录

一、 实验概述	错误! 未定义书签。
二、 实验目的	3
三、 实验仪器	3
四、 实验原理	4
1. 加速度计的六位置法标定原理	4
2. 陀螺仪标定原理	7
3. 误差补偿原理	9
五、 实验步骤和方法	10
六、 注意事项	10
七、 程序设计	11
1. 数据读取	11
2. 加速度计标定程序设计	15
3. 陀螺仪标定程序设计	16
4. 误差程序设计	18
八、 实验结果	19
1. 加速度计标定结果	19
2. 陀螺仪标定结果	21
九、 实验总结	23

一、实验概述

惯性导航的解算本质是一个连续的积分过程，这一特性决定了惯性传感器中任何微小的初始误差，都会随着时间的推移被不断累积和放大，最终导致导航结果（如位置、姿态）产生不可接受的巨大偏差。因此，通过系统性的测试与标定，精确辨识并补偿这些误差源，是保证惯性导航系统长期可靠运行的根本前提。

本实验旨在通过加速度六位置法和陀螺仪角位置法对仪器零偏、比例因子等误差进行标定并补偿误差。

二、实验目的

- (1) 掌握三轴加速度计的静态六位置标定方法，标定加速度计的零偏、比例因子误差和交轴耦合误差。
- (2) 掌握角位置法标定陀螺的零偏和比例因子误差标定方法。
- (3) 掌握高精度转台的基本使用方法。
- (4) 对加速度计和陀螺仪误差标定后进行误差补偿。

三、实验仪器

- (1) 高精度三轴转台(图1)。
- (2) 待标定IMU，型号：星网宇达XW-7681，具体参数见表1。



图 1 高精度三轴转台

表 1 IMU 的主要误差参数

IMU型号	陀螺		加速度计	
	零偏 (deg/h, 1σ)	ARW (deg/ \sqrt{h})	零偏 (mGal, 1σ)	VRW (m/s/ \sqrt{h})
XW-GI7681	0.05(bias)	0.01	30	0.03
iMAR-FSAS	0.75(bias)	0.1	1000	0.03
SPAN-CPT	20(bias) 1(bias instability)	0.07	50000(bias) 750(bias instability)	0.1

四、实验原理

1. 加速度计的六位置法标定原理

加速度计的静态六位置标定法简便、可靠，是常用的实验室标定方法，可以标定出加速度计的零偏、比例因子和交轴耦合。标定要求加速度计的3个轴线分别朝上和朝下静止一段时间，记录加速度计的原始数据。

三轴加速度计的测量模型如下：

$$\tilde{f}^b = f^b + b_a + S_a f^b + N_a f^b + w_a \quad (1)$$

其中，

$\tilde{f}^b = [\tilde{f}_x \quad \tilde{f}_y \quad \tilde{f}_z]^T$ 是加速度计测量值向量，即实际输出；

$f^b = [f_x \quad f_y \quad f_z]^T$ 为加速计感知的真实比力向量，即理论输出；

$b_a = [b_{a,x} \quad b_{a,y} \quad b_{a,z}]^T$ 为加速度计零偏向量；

$w_a = [w_{a,x} \quad w_{a,y} \quad w_{a,z}]^T$ 为加速计的随机噪声向量；

S_a 和 N_a 分别为加速度计的比例因子误差和交轴耦合误差矩阵，记为

$$S_a = \begin{bmatrix} s_{a,x} & 0 & 0 \\ 0 & s_{a,y} & 0 \\ 0 & 0 & s_{a,z} \end{bmatrix}, N_a = \begin{bmatrix} 0 & \gamma_{a,yx} & \gamma_{a,zx} \\ \gamma_{a,xy} & 0 & \gamma_{a,zy} \\ \gamma_{a,xz} & \gamma_{a,yz} & 0 \end{bmatrix}$$

其中， $s_{a,x}$ 为 x 轴加速度计的比例因子误差， $\gamma_{a,yx}$ 为 x 轴敏感到 y 轴比例的交轴耦合误差，其他几项定义类似。

在实际的标定过程中，通过对加速度计中的一段静态数据求平均来降低随机噪声的干扰幅度，使其影响明显小于待标定的零偏参数。在这种情况下，可以忽略式(1)中的噪声项，并改写成以下矩阵形式：

$$\begin{bmatrix} \tilde{f}_x \\ \tilde{f}_y \\ \tilde{f}_z \end{bmatrix} = \begin{bmatrix} 1+s_{a,x} & \gamma_{a,yx} & \gamma_{a,zx} \\ \gamma_{a,xy} & 1+s_{a,y} & \gamma_{a,zy} \\ \gamma_{a,xz} & \gamma_{a,yz} & 1+s_{a,z} \end{bmatrix} \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} + \begin{bmatrix} b_{a,x} \\ b_{a,y} \\ b_{a,z} \end{bmatrix} \quad (2)$$

将上式进一步改写成如下齐次形式：

$$\begin{bmatrix} \tilde{f}_x \\ \tilde{f}_y \\ \tilde{f}_z \end{bmatrix} = \begin{bmatrix} 1+s_{a,x} & \gamma_{a,yx} & \gamma_{a,zx} & b_{a,x} \\ \gamma_{a,xy} & 1+s_{a,y} & \gamma_{a,zy} & b_{a,y} \\ \gamma_{a,xz} & \gamma_{a,yz} & 1+s_{a,z} & b_{a,z} \end{bmatrix} \begin{bmatrix} f_x \\ f_y \\ f_z \\ 1 \end{bmatrix} \quad (3)$$

上式等式右边的矩阵包含了12个待标定的加速度计误差参数，记为 M 。为了求解 M ，我们将加速度计按图2的方式，分别让x、y、z轴加速度计朝上和朝下静置，共采集六个位置的静态数据。

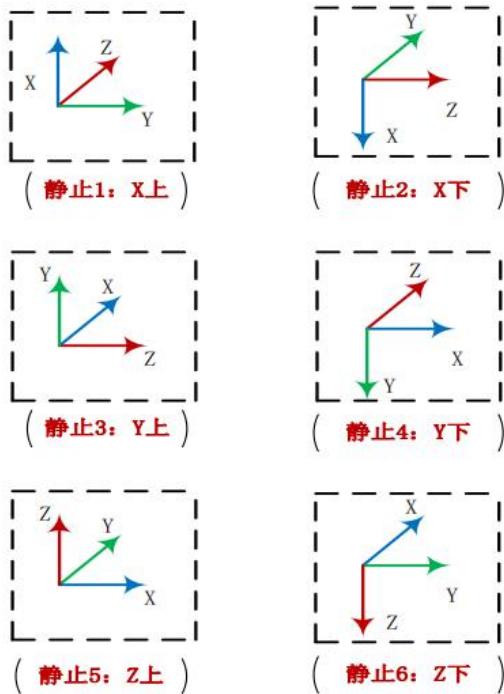


图 2 加速度计标定六位置法示意图，每个位置静置一段时间(如5min，根据加速度计噪声)

对应这六个位置，IMU感知的理论比力向量为：

$$f_1 = \begin{bmatrix} g \\ 0 \\ 0 \end{bmatrix}, f_2 = \begin{bmatrix} -g \\ 0 \\ 0 \end{bmatrix}, f_3 = \begin{bmatrix} 0 \\ g \\ 0 \end{bmatrix}, f_4 = \begin{bmatrix} 0 \\ -g \\ 0 \end{bmatrix}, f_5 = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}, f_6 = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \quad (4)$$

其中，比力向量的下标1-6分别表示加速度计x轴朝上，x轴朝下，y轴朝上，y轴朝下，z轴朝上，z轴朝下。

按公式(3)写出每个位置对应的加速度计测量模型，可合并为如下分块矩阵：

$$\begin{bmatrix} \tilde{f}_1 & \tilde{f}_2 & \tilde{f}_3 & \tilde{f}_4 & \tilde{f}_5 & \tilde{f}_6 \end{bmatrix} = \begin{bmatrix} 1+s_{a,x} & \gamma_{a,yx} & \gamma_{a,zx} & b_{a,x} \\ \gamma_{a,xy} & 1+s_{a,y} & \gamma_{a,zy} & b_{a,y} \\ \gamma_{a,xz} & \gamma_{a,yz} & 1+s_{a,z} & b_{a,z} \end{bmatrix} \begin{bmatrix} f_1 & f_2 & f_3 & f_4 & f_5 & f_6 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (5)$$

上式记为

$$L = MA \quad (6)$$

其中, $L = [\tilde{f}_1 \ \tilde{f}_2 \ \tilde{f}_3 \ \tilde{f}_4 \ \tilde{f}_5 \ \tilde{f}_6]$; $A = [f_1 \ f_2 \ f_3 \ f_4 \ f_5 \ f_6]$;

$$M = \begin{bmatrix} 1 + s_{a,x} & \gamma_{a,yx} & \gamma_{a,zx} & b_{a,x} \\ \gamma_{a,xy} & 1 + s_{a,y} & \gamma_{a,zy} & b_{a,y} \\ \gamma_{a,xz} & \gamma_{a,yz} & 1 + s_{a,z} & b_{a,z} \end{bmatrix}$$

从上式中可以看出, M矩阵中包含12个待估参数, 总计有18个点测量方差, 存在冗余观测, 因此可以使用最小二乘求解待估参数:

$$\hat{M} = LA^T (AA^T)^{-1} \quad (7)$$

注意, 由于A不算方阵, 不可直接求逆, 而 AA^T 为对称方阵, 可以进行求逆, 因此采用上述求解形式。

2. 陀螺仪标定原理

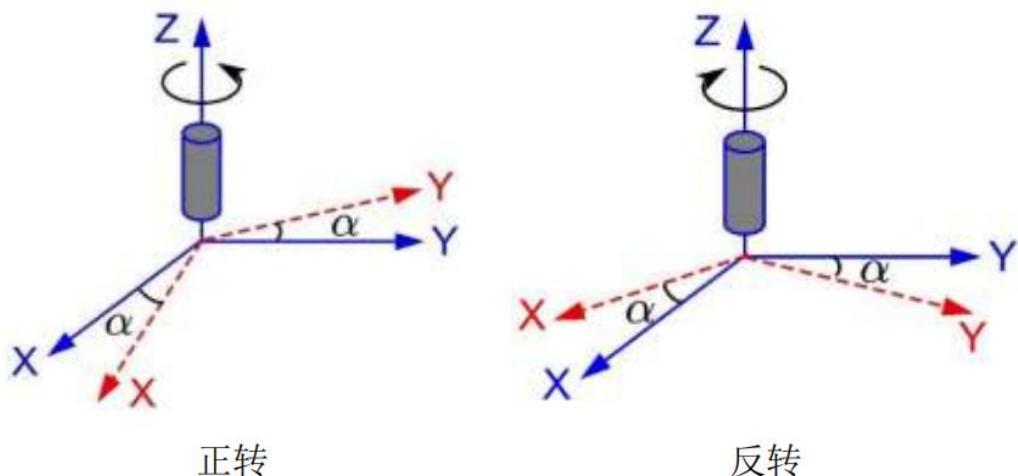


图3 z轴陀螺敏感轴朝上, 转台分别绕 z 轴正方向转动和绕 z 轴正方向反转

陀螺零偏和比例因子误差的两位置法标定原理如图3所示, 图中以z轴陀螺的转台角位置法标定为例进行说明, 控制转台让z轴陀螺敏感轴与铅锤线重合, 朝上摆放。首先, 控制陀螺绕z轴正方向转指定角度 α (一般为 360°), 然后控制转台绕z轴负方向转过同样的角度。在该情况下, z轴陀螺感知的角速度为转台角速度和地球自转角速度

在垂向的投影分量。因此，这两种情况下的测量模型如下(忽略陀螺白噪声)：

$$\begin{aligned}\tilde{\alpha}_1 &= b_{g,z}t + (1+s_{g,z})\alpha + (w_e \sin \varphi)t \\ \tilde{\alpha}_2 &= b_{g,z}t - (1+s_{g,z})\alpha + (w_e \sin \varphi)t\end{aligned}\quad (8)$$

其中， $\tilde{\alpha}_1$ 和 $\tilde{\alpha}_2$ 分别为正转和反转时 z 轴陀螺积分后的角度变化量； $b_{g,z}$ 和 $s_{g,z}$ 分别为 z 轴的陀螺零偏和比例因子误差； w_e 为地球自转角速度大小； φ 为当地纬度； t 为截取的陀螺积分时长(应该覆盖该次转台转到的完整时段)。

根据公式(8)易得陀螺的零偏和比例因子误差：

$$\begin{aligned}b_{g,z} &= \frac{\tilde{\alpha}_1 + \tilde{\alpha}_2}{2t} - (w_e \sin \varphi) \\ s_{g,z} &= \frac{\tilde{\alpha}_1 - \tilde{\alpha}_2}{2} - 1\end{aligned}\quad (9)$$

注意到上述公式采取的测量值为角度，而不是角速度，因此该方法被称为角位置法。之所以使用角度进行误差标定而不是角速度进行标定，是因为在实际情况下，难以控制角速度恒定，角位置标定对转台的要求低，更加经济实惠和方便使用。

需要注意的是，由于无法像加速度计六位置法标定那样对陀螺的观测值进行长时间平均来减小噪声的影响，采取角位置法进行标定时，其精度可能会受到陀螺随机噪声的影响。尽管根据陀螺仪输出进行积分计算转角在本质上可以起到一定的平均滤波水平，却往往不足以将陀螺噪声幅度降至低于其零偏的水平。

因此，在实际情况下，可以考虑加速计静态六位置法标定中收集的静态陀螺数据进行平均处理，此时陀螺感知的真实输入只有地球自转角速度，测量模型如下：

$$\begin{aligned}\hat{w}_{up} &= b_g + (1+s_g)w_e \sin \varphi \\ \hat{w}_{down} &= b_g - (1+s_g)w_e \sin \varphi\end{aligned}\quad (10)$$

其中， \hat{w}_{up} 和 \hat{w}_{down} 分别为陀螺敏感轴朝上和朝下时的角速度观测的均值，因此，容易求出陀螺零偏和比例因子

$$\begin{aligned} b_g &= \frac{\hat{w}_{up} + \hat{w}_{down}}{2} \\ s_g &= \frac{\hat{w}_{up} - \hat{w}_{down}}{2w_e \sin \varphi} - 1 \end{aligned} \quad (11)$$

然而，由于上式比例因子的系数为地球自转参数，该数值相对较小，往往无法充分展现比例因子的影响，因此，通过上式求得的比例因子误差精度相对较低。同时，尽管通过该模型同样可以求得交轴耦合矩阵，但是由于陀螺转动过程中水平轴陀螺感知到的地球自转分量是交变的，相对求解比较复杂，因此这里不做考虑。

综上，为了实现相对较高的标定精度，本次实验我们综合静态两位置法标定零偏和角位置法标定比例因子。

3. 误差补偿原理

误差参数标定后，可对IMU原始数据进行误差补偿，得到更高精度的比力和角速率观测值，补偿模型如下：

$$m_c = \begin{bmatrix} 1+s_x & \gamma_{yx} & \gamma_{zx} \\ \gamma_{xy} & 1+s_y & \gamma_{zy} \\ \gamma_{xz} & \gamma_{yz} & 1+s_z \end{bmatrix}^{-1} (\tilde{m} - b) \quad (12)$$

其中， m_c 表示补偿后的三轴陀螺或加速度计测量值向量， \tilde{m} 表示对应的原始测量值向量， b 为对应的零偏向量，矩阵内部为对应的比例因子和交轴耦合系数。

五、实验步骤和方法

- (1) 根据实验前设计好的动作对转台进行归零。
- (2) 控制转台依次采集六个位置的IMU静态数据。
- (3) 控制转台转动采集陀螺转动数据。
- (4) 参考示例文档，利用转换软件，对IMU原始数据进行解码和格式转换。
- (5) 编写程序计算加速度计和陀螺的零偏、比例因子等误差。
- (6) 标定误差后进行测量值误差补偿，对比分析误差影响。

六、注意事项

- (1) 如何使用IMU增量格式的原始数据

高精度惯导的输出往往并不是直接输出比力和角速度测量值，而是输出两个采样时刻之间的比力和角速度积分值：

$$\Delta\theta(t_k) = \int_{t_{k-1}}^{t_k} w_{ib}^b(t) dt, \Delta\nu(t_k) = \int_{t_{k-1}}^{t_k} f^b(t) dt \quad (13)$$

要将陀螺和加速度计的增量数据转换为角速度和比力，可以直接用角度增量和速度增量除以采样间隔(或乘以采样率)得到。对于陀螺的标定，由于采用角位置法本来就要对角速度积分，因此直接将角增量累加即可，但对于静态标定，可将陀螺输出的角增量累加后除以平均时长，即可得到平均角速度。

- (2) 静态平均时长的选取

加速度计六位置法标定和陀螺静态标定，都涉及对静态数据求平均来克服仪器噪声的影响，一般要求平均后噪声的幅度明显小于零偏噪声，因此平均时长的选取至关重要。

以陀螺为例，平均后，噪声序列的均方根(RMS)与平均时长T的关系为

$$RMS_{\text{平均}} = E\{\bar{w}^2(t)\} = \frac{ARW}{\sqrt{T}} \quad (14)$$

假设陀螺零偏大小为 b_g ，要使得平均后噪声幅度小于 $\frac{1}{3}b_g$ ，则所需平均时长为

$$\begin{aligned} RMS_{\text{平均}} &= \frac{ARW}{\sqrt{T}} \leq \frac{1}{3}b_g \\ T &\geq \left(\frac{3ARW}{b_g}\right)^2 \end{aligned} \quad (15)$$

(3) 标定数据的截取

标定陀螺比例因子误差的正反转过程中，截取的数据期间应该覆盖完整的转动过程，即启动之前到停稳之后，多截取几秒的静止数据对结果影响不大。但是考虑到公式(8)，正反转截取的区间长度应该严格相等(即t相等)，才能消除零偏项推导出公式(9)中的比例因子。

对于加速度计标定，通过观测波形来判断需要截取的静态数据段即可，一般不要求时间区间严格相等。

(4) 标定所需信息

转台实验室的纬度 $\varphi = 30.531651244^\circ$ (北纬)，椭球高 $h = 28.2134\text{m}$ ，重力加速计 $g = 9.7936174\text{m/s}^2$ ，地球自转角速度 $w_e = 7.292115 \times 10^{-5}\text{rad/s}$ 。

七、程序设计

1. 数据读取

参考图4数据加速度计格式转换说明，惯导输出数据中*号前面六组数据按顺序分别是zyx轴对应的加速度计和陀螺仪数据，并且由于

这些数据通常是数字信号等，因此先要乘以比例因子转为真实物理量；又由于采取的是增量输出形式，因此还需要乘采样率转为比力和角速度。

acc_z	acc_y	acc_x
64143	-64	241
9.7874450684	0.0097656250	0.0367736816

图4 加速度计数据格式转换

首先在程序中定义了原始数据结构体，用以存储数据。

```

1. // 原始惯导观测数据
2. struct RawInerData
3. {
4.     GPSTime gpstime;
5.     double freq;        // 采样频率(Hz)
6.     double acc[3];      // 加速度(m/s^2) zyx
7.     double gyro[3];     // 角速度(rad/s) zyx
8.     double delta_angle[3]; // 角增量(rad) zyx
9.     double delta_v[3];   // 速度增量 zyx
10.    RawInerData() {...}
11. };

```

然后设计单行数据处理函数，读取到惯导数据则进行格式转换，其他异常数据就返回。

```

1. /*
2. @brief 读取单行数据
3. */
4. bool readRawInerDataInLine(ifstream& file, RawInerData& data, int type, double freq)
5. [
6.     string line;
7.     // 文件结束或读取失败/空行/非RAWIMUSA数据
8.     if (!getline(file, line) || line.empty() || line.find("%RAWIMUSA") == string::npos)
9.     {
10.         return false;
11.     }
12.     // 移除*后面的内容

```

```
13.     size_t checksum_pos = line.find('*');
14.     string data_line = (checksum_pos != string::npos) ? line.substr(0, checksum_pos) : line;
15.     // 分割数据
16.     vector<string> tokens;
17.     stringstream ss(data_line);
18.     string token;
19.     while (getline(ss, token, ',')) { tokens.push_back(token); }
20.
21.     // 解析数据
22.     data.gpstime.week = stoi(tokens[1]);
23.     data.gpstime.sow = stod(tokens[2].substr(0, 9));
24.     data.frq = frq;
25.     data.delta_v[0] = stod(tokens[5]) * ACC_SCALE[type];
26.     data.delta_v[1] = -stod(tokens[6]) * ACC_SCALE[type];
27.     data.delta_v[2] = stod(tokens[7]) * ACC_SCALE[type];
28.     data.delta_angle[0] = stod(tokens[8]) * GYRO_SCALE[type];
29.     data.delta_angle[1] = -stod(tokens[9]) * GYRO_SCALE[type];
30.     data.delta_angle[2] = stod(tokens[10]) * GYRO_SCALE[type];
31.
32.     return true;
33. }
```

考虑到惯导采样率采样率较高，在实际使用是可能会按一定时间间隔取平均处理，因此在文件读取主函数需要对指定时间间隔内的惯导数据平均(角增量取累加)，具体设计如下：

```
1. /*
2. @brief 读取所有原始惯导观测数据，并按T时间间隔内取平均加速度/角速度等数据
3. @param T 时间间隔(秒) 0:总平均; > 0:该时间间隔内平均; -1或1/frq:单次数据
4. */
5. bool readAllRawInerDataInT(const char *filename, vector<RawInerData> &data, int type, double frq, double T)
6. {
7.     ifstream file(filename);
8.     if (!file.is_open())
9.     {
10.         cerr << "Error: Cannot open file " << filename << endl;
11.         return false;
12.     }
13.
14.     if (T == -1)
15.         T = 1 / frq;
16.     RawInerData accumulator;           // 累加数据
17.     int sample_count = 0;             // 已累加的样本数
18.     bool is_first_valid_epoch = true; // 文件第一个有效历元
19.
20.     RawInerData current_data;
21.     while (!file.eof())
22.     {
```

```
23.
24.     bool flag = readRawInerDataInLine(file, current_data, type, frq);
25.     if (!flag)
26.         continue;
27.     // 第一个有效历元
28.     if (is_first_valid_epoch && T > 0)
29.     {
30.         for (int i = 0; i < 3; ++i)
31.         {
32.             current_data.acc[i] = current_data.delta_v[i] * frq;
33.             current_data.gyro[i] = current_data.delta_angle[i] * frq;
34.         }
35.         data.push_back(current_data);
36.         is_first_valid_epoch = false;
37.         continue;
38.     }
39.
40.     // 累加
41.     accumulator.gpstime = current_data.gpstime;
42.     for (int i = 0; i < 3; i++)
43.     {
44.         accumulator.delta_v[i] += current_data.delta_v[i];
45.         accumulator.delta_angle[i] += current_data.delta_angle[i];
46.     }
47.     sample_count++;
48.
49.     // 达到时间间隔
50.     if (sample_count / frq >= T && T > 0)
51.     {
52.         accumulator.frq = frq;
53.         double total_time = sample_count / frq;
54.         for (int i = 0; i < 3; i++)
55.         {
56.             accumulator.acc[i] = accumulator.delta_v[i] / total_time;
57.             accumulator.gyro[i] = accumulator.delta_angle[i] / total_time;
58.         }
59.         data.push_back(accumulator);
60.         // 清零
61.         accumulator = RawInerData();
62.         sample_count = 0;
63.     }
64. }
65.
66. // 全平均或最后一组
67. if (sample_count > 0)
68. {
69.     accumulator.frq = frq;
70.     double total_time = sample_count / frq;
71.     for (int i = 0; i < 3; i++)
72.     {
73.         accumulator.acc[i] = accumulator.delta_v[i] / total_time;
74.         accumulator.gyro[i] = accumulator.delta_angle[i] / total_time;
75.     }
76.     data.push_back(accumulator);
```

```
77.    }
78.
79.    file.close();
80.    if (data.empty())
81.    {
82.        cerr << "Error: No valid data found in file " << filename << endl;
83.        return false;
84.    }
85.    return true;
86. }
```

2. 加速度计标定程序设计

由于待标定的零偏等误差主要是与仪器有关的误差，因此本实验中专门设计加速度计类以封装与加速度计仪器有关的属性与方法。

```
1. // 加速度计(与仪器相关)
2. class Accer
3. {
4. private:
5.     Vector3d bias;      // 零偏
6.     Matrix3d Sa, Na;   // 比例因子与交轴耦合矩阵
7. public:
8.     Accer()
9.     {
10.         bias = Vector3d::Zero();
11.         Sa = Matrix3d::Identity();
12.         Na = Matrix3d::Zero();
13.     }
14.     void demarcate(const array<RawInerData,6>& data);
15.     Vector3d compensate(const double acc_data[]);
16.     void showInfo();
17. };
```

其中，`showInfo()`函数主要负责在屏幕中打印私有属性，这里不做具体说明，`compensate()`函数设计见误差补偿模块。

`demarcate()`函数根据传入的六位置比力平均值对误差进行标定，具体见下：

```
1. /*
2. @brief 加速度计六位置标定
3. @param data:六个位置的平均观测数据(zyx,up->down)
4. */
5. void Accer::demarcate(const array<RawInerData, 6> &data)
6. {
7.     MatrixXd L = MatrixXd::Zero(3, 6);
8.     // 计算各个位置的平均值(xyz, up->down)
```

```
9.     for (int i = 0; i < 6; i++)
10.    {
11.        L(0, i) = data[i].acc[2];
12.        L(1, i) = data[i].acc[1];
13.        L(2, i) = data[i].acc[0];
14.    }
15.    // 构造方程组
16.    MatrixXd A = MatrixXd::Zero(4, 6);
17.    A(0, 0) = G_VAL;
18.    A(0, 1) = -G_VAL;
19.    A(1, 2) = G_VAL;
20.    A(1, 3) = -G_VAL;
21.    A(2, 4) = G_VAL;
22.    A(2, 5) = -G_VAL;
23.    A(3, 0) = 1; A(3, 1) = 1; A(3, 2) = 1; A(3, 3) = 1; A(3, 4) = 1; A(3, 5) = 1;
24.
25.    // 求解耦合后的零偏/比例因子/交轴耦合
26.    MatrixXd M = L * A.transpose() * (A * A.transpose()).inverse();
27.
28.    // 赋值
29.    bias(0) = M(0, 3), bias(1) = M(1, 3), bias(2) = M(2, 3);
30.    Sa(0, 0) = M(0, 0) - 1, Sa(1, 1) = M(1, 1) - 1, Sa(2, 2) = M(2, 2) - 1;
31.    Na(0, 1) = M(0, 1), Na(0, 2) = M(0, 2), Na(1, 0) = M(1, 0), Na(1, 2) = M(1, 2), Na(2,
   , 0) = M(2, 0), Na(2, 1) = M(2, 1);
32. }
```

3. 陀螺仪标定程序设计

与加速度计一样，本程序封装陀螺仪类对与其有关的属性和方法进行处理。

```
1. // 陀螺仪
2. class Gyro
3. {
4. private:
5.     Vector3d bias; // 角速度零偏
6.     Matrix3d Sg; // 角速度比例因子矩阵
7.
8. public:
9.     Gyro()
10.    {
11.        bias = Vector3d::Zero();
12.        Sg = Matrix3d::Identity();
13.    }
14.    void demarcate(const array<RawInerData, 6> &sta_data, const array<vector<RawInerData
   , 6> &dy_data);
15.    Vector3d compensate(const double gyro_data[]);
16.    void showInfo();
17. }
```

其他程序与加速度计中程序功能类似，值得注意的是，由于对陀螺仪标定首先需要通过静态两位置法标定陀螺零偏，然后再通过角位置法标定陀螺比例因子，因此，程序设计时既需要传入加速度计静态数据的平均值，又需要传入陀螺转动时采取的数据。同时，考虑到在实际操作中，难以控制同轴正负转动的时间区间相同，所以，这里不对读进来转动数据做平均，而选择在demarcate()程序中进行时间区间对齐处理，具体函数如下：

```
1. /*
2. @brief 陀螺仪标定
3. @param sta_data:静态两位置法数据(zyx,up->down)
4. @param dy_data:角位置法数据(zyx,+->-)
5. */
6. void Gyro::demarcate(const array<RawInerData, 6> &sta_data, const array<vector<RawInerData>, 6> &dy_data)
7. {
8.     // 静态两位置法计算零偏
9.     for (int i = 0; i < 3; i++)
10.    {
11.        double w_up = sta_data[2 * i].gyro[2 - i], w_down = sta_data[2 * i + 1].gyro[2 - i];
12.        bias(i) = (w_up + w_down) / 2.0;
13.    }
14.
15.    // 时间区间严格相等
16.    array<RawInerData, 6> prc_data;
17.    double frq = sta_data[0].frq;
18.    for (int i = 0; i < 3; i++)
19.    {
20.        double angle_bias = bias(i) / frq; // 角增量零偏改正
21.        int len1 = dy_data[2 * i].size(), len2 = dy_data[2 * i + 1].size();
22.        int num_samples = min(len1, len2);
23.        int start1 = (len1 > len2) ? (len1 - len2) / 2 : 0;
24.        int start2 = (len2 > len1) ? (len2 - len1) / 2 : 0;
25.
26.        // 在同步区间内进行累加
27.        for (int j = 0; j < num_samples; j++)
28.        {
29.            // 累加角增量
30.            for (int k = 0; k < 3; ++k)
31.            {
32.                prc_data[2 * i].delta_angle[k] += dy_data[2 * i][j + start1].delta_angle[k];
33.                prc_data[2 * i + 1].delta_angle[k] += dy_data[2 * i + 1][j + start2].delta_angle[k];
34.            }
35.        }
36.    }
37.}
```

```
38. // 角位置法计算比例因子
39. for (int i = 0; i < 3; i++)
40. {
41.     double angle1 = prc_data[2 * i].delta_angle[2 - i];
42.     double angle2 = prc_data[2 * i + 1].delta_angle[2 - i];
43.     Sg(i, i) = (angle1 - angle2) / (2 * 2 * PI) - 1;
44. }
45. }
```

4. 误差补偿程序设计

参考公式(12), 陀螺仪和加速度计误差补偿原理相同, 不同点在于本次实验未对陀螺仪进行交轴耦合标定, 因此不进行该项补偿。

```
1. /*
2. @brief 加速度计观测值改正
3. @param acc_data: 原始加速度观测值(zyx)
4. @return 改正后的加速度xyz
5. */
6. Vector3d Accer::compensate(const double acc_data[])
7. {
8.     Vector3d mg{acc_data[2], acc_data[1], acc_data[0]};
9.     Vector3d compg = (Matrix3d::Identity() + Sa + Na).inverse() * (mg - bias);
10.    return compg;
11. }
12.
13. /*
14. @brief 陀螺仪观测值改正
15. @param 原始角速度观测值(zyx)
16. @return 改正后的角速度xyz
17. */
18. Vector3d Gyro::compensate(const double gyro_data[])
19. {
20.     Vector3d mw{gyro_data[2], gyro_data[1], gyro_data[0]};
21.     Vector3d compw = (Matrix3d::Identity() + Sg).inverse() * (mw - bias);
22.     return compw;
23. }
```

八、实验结果

1. 加速度计标定结果

通过静态六位置法对加速度计进行标定，可以得到以下结果：

表2 加速度计零偏

x轴 (m/s^2)	y轴 (m/s^2)	z轴 (m/s^2)
-0.00972165	0.000617231	-0.0157585

表3 加速度计比例因子误差

x轴	y轴	z轴
0.000235919	0.000354155	0.000419929

表4 加速度计交轴耦合矩阵

0	0.00394708	-0.00799767
-0.00389479	0	0.000469406
0.00809381	-0.00039625	0

从上述结果可以看出，整体误差处于合理范围内，但是加速度计Z轴具有最大的零偏误差和比例因子误差，并且Z轴和X轴强耦合。

下面以X轴朝上的静态观测文件为例进行误差补偿分析

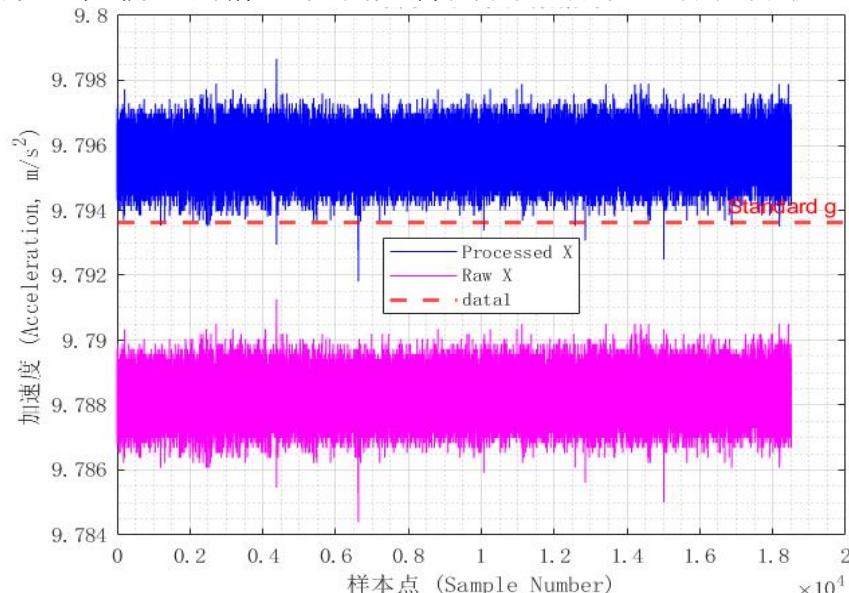


图4 XUp文件加速度计x轴数据

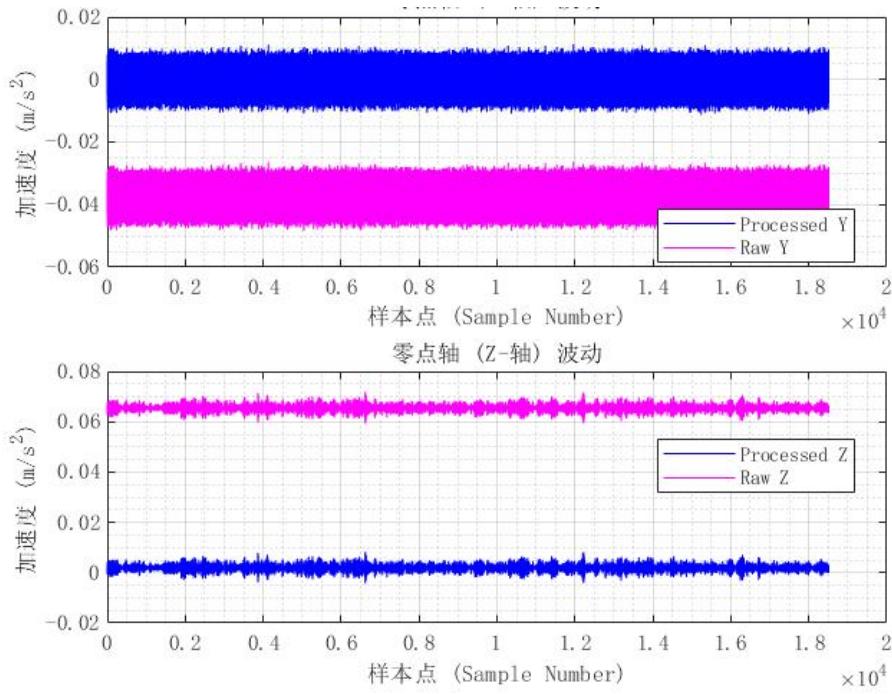


图5 XUp文件加速度计y轴和z轴数据

根据图(4)和图(5)可以看出，经过误差补偿后，加速度计未受力的方向改正值接近于0，重力方向相对于改正前更接近于理论重力值(改进了50左右)，但是仍然存在轻微的整体偏差。

表5 XUp文件重力轴数据分析

	均值(m / s^2)	标准差(m / s^2)	相对误差(m / s^2)
补偿后数据	9.795693	0.000826	0.0021
补偿前数据	9.788267	0.000847	0.0054

考虑到标定结果中x轴与z轴强耦合，并且z轴零偏等误差相对较大，下面对Z轴朝上的静态观测文件进行分析。

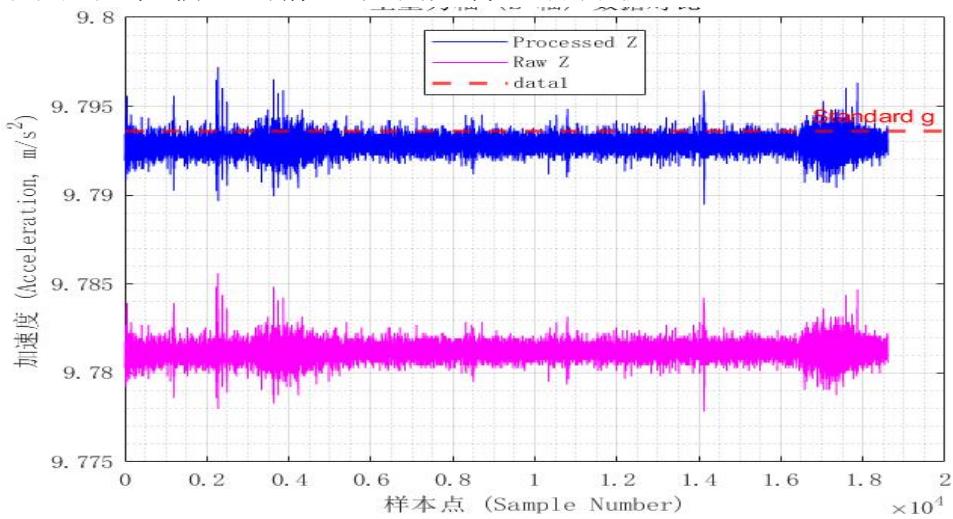


图6 ZUp文件加速度计z轴数据

由图(6)可以看出，尽管误差补偿后的观测值明显优于补偿前，但是整体仍然存在系统偏差，考虑到本次实验中我们仅对仪器零偏、比例因子和交轴耦合进行补偿，并未考虑到温度系数等其他因素的影响，推测该项偏差可能是由于仪器其他误差造成的。

2. 陀螺仪标定结果

先通过静态两位置法标定陀螺零偏，再通过角位置法标定比例因子标定陀螺比例因子，可以得到以下结果：

表6 陀螺仪零偏

x轴 (rad / s)	y轴 (rad / s)	z轴 (rad / s)
-3.43907e-06	2.39474e-07	-1.90242e-06

表7 陀螺仪比例因子误差

x轴	y轴	z轴
-0.000546012	-7.47405e-06	0.000217805

从上述结果可以看出，陀螺仪零偏误差较小，所以轴都非常稳定，几乎不存在零偏；比例因子误差转换为PPM分别为(-546,-7.47,217.8)，Y轴表现最佳，X轴误差相对较大，但是对整体的观测值影响也相对较小。

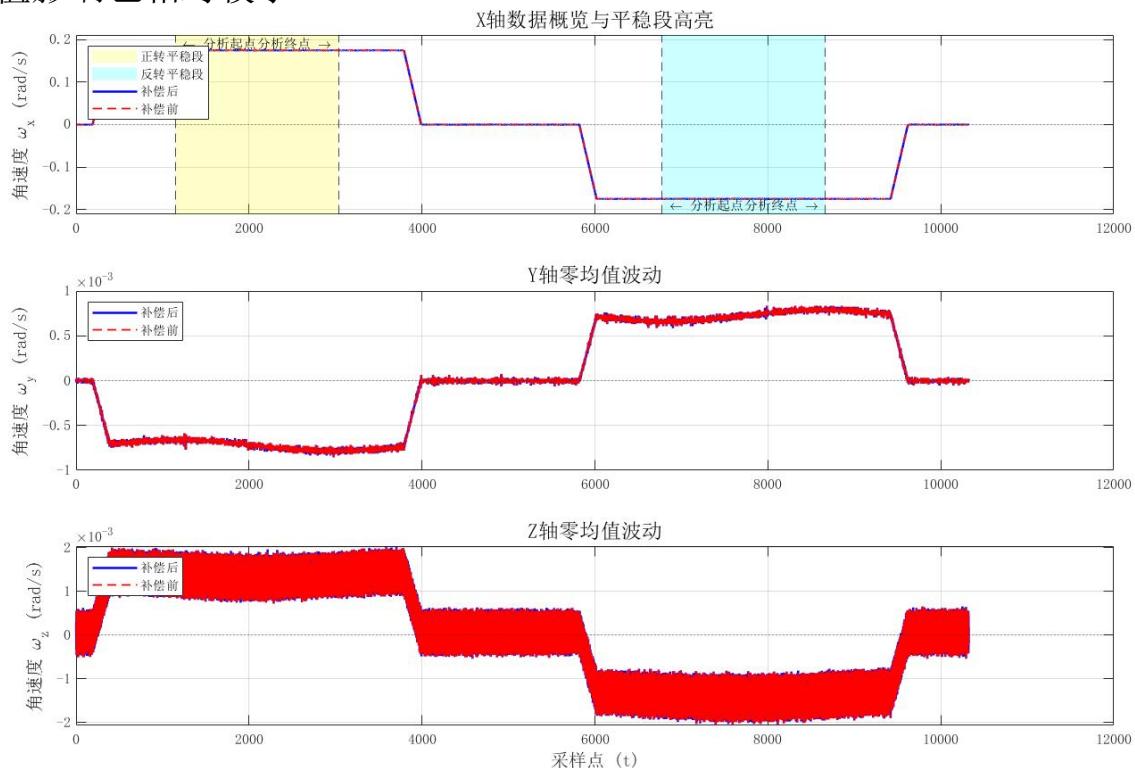


图7 陀螺仪补偿前后角速度数据

上图主要对陀螺仪绕X轴正转和反转的数据进行分析，可以发现，如前所述，仪器零偏和比例因子误差对观测值整体影响不大，未进行旋转的方向角速度理论值接近于0，但是波动幅度较大，原因在于本实验中并未进行交轴耦合改正，在陀螺旋转过程中旋转方向的角速度部分投影到非旋转方向，导致整体成分块稳定波动的形式；

旋转方向在宏观上体现不出观测误差，因此，对图中平稳段进行放大分析(图8)，可以看出，补偿后正反转均值更加接近于理论值(均值同理论值之间的误差更小)，但整体相差不大；并且补偿前后波动幅度并无明显变化并且数值较小，说明观测值随机噪声较小；结合加速度计和陀螺仪的数据可以看出，本次实验中陀螺仪的精度明显高于加速度计。

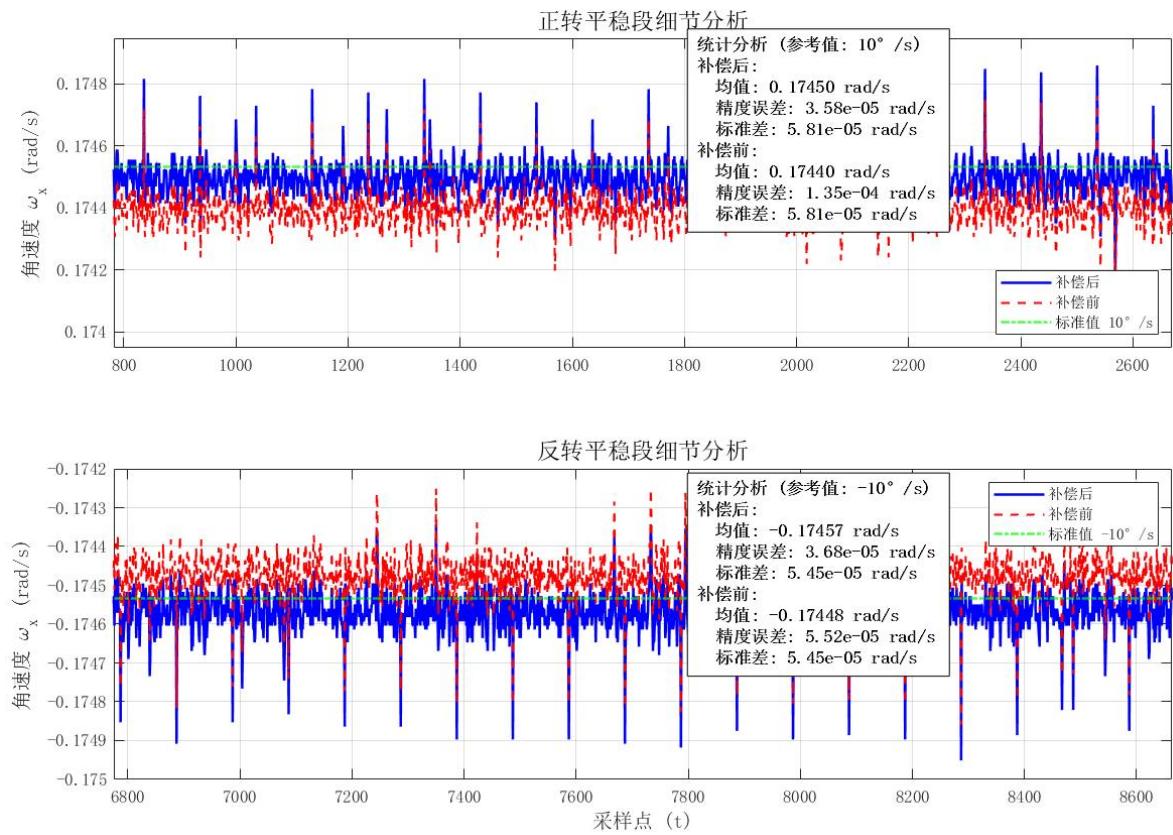


图8 陀螺仪补偿前后角速度平稳段分析

九、实验总结

本次实验的核心目标的掌握IMU中加速度计和陀螺仪主要误差参数的标定方法(静态位置法和角位置法)，理解IMU仪器的增量输出形式，对误差进行补偿分析。

从上述实验结果中可以看出，进行误差标定和补偿后，精度误差明显减小；但是随机误差并未出现明显的变化，说明改正的主要还是系统噪声，在后续导航算法中，需要对随机误差进行分析处理。并且尽管在实验中陀螺仪精度较高，我们并没有改正陀螺交轴耦合，但考虑到长时间导航的误差累计，该项误差也可能会造成相对较大的影响，因此，在现有实验条件下，可以考虑增加冗余观测数据(例如采取更多的位置组合)进行静态标定。

综合实验，我们可以认识到，任何IMU传感器都存在固有的系统性误差和随机误差。直接使用原始数据进行导航解算，误差会迅速累积，导致结果发散。通过精确的数学建模和严谨的标定实验，才可以有效地识别并补偿这些系统性误差，从而大幅提升IMU数据的质量。