

2024-2025 学年第二学期  
武汉大学测绘学院导航工程专业

## 卫星导航原理实验报告

课程名称： 卫星导航原理

专业班级： 智能导航一班

姓 名： 秦旗峰

学 号： 2023302143029

实验名称： MW 与 GF 组合观测值

实验时间： 2025.05

# 目录

1 实验目的.....	3
2 文件读取及组合观测值计算.....	3
2.1 文件格式.....	3
2.2 GF 与 MW 组合观测值计算 .....	4
2.2.1 GF 组合 .....	4
2.2.2 MW 组合 .....	5
2.2.3 组合观测值历元间求差分.....	5
3 编程解算.....	5
3.1 常量定义.....	5
3.2 函数定义.....	5
3.3 主函数.....	6
3.4 数据可视化.....	8
4 结果分析.....	8
5 实验总结与反思.....	10
参考.....	10

# 1 实验目的

加深对 RINEX 观测值文件内容和结构的理解，编程读取观测值文件得到 GPS 和 BDS 的相位和伪距观测值进行 GF 和 MW 组合观测值计算，并绘制时序图，进行周跳探测。

## 2 文件读取及组合观测值计算

### 2.1 文件格式

**RINEX3.03 观测值文件：**观测值文件保存了观测时刻卫星的伪距、载波相位和多普勒等观测值。文件由文件头和数据体组成。文件头中较为重要的信息是观测值存储顺序：

1.	G	16	C1C	C2W	C2X	C5X	D1C	D2W	D2X	D5X	L1C	L2W	L2X	L5X	S1C	SYS / # / OBS TYPES
2.			S2W	S2X	S5X											SYS / # / OBS TYPES
3.	R	16	C1C	C1P	C2C	C2P	D1C	D1P	D2C	D2P	L1C	L1P	L2C	L2P	S1C	SYS / # / OBS TYPES
4.			S1P	S2C	S2P											SYS / # / OBS TYPES
5.	J	12	C1C	C2X	C5X	D1C	D2X	D5X	L1C	L2X	L5X	S1C	S2X	S5X		SYS / # / OBS TYPES
6.	C	12	C2I	C6I	C7I	D2I	D6I	D7I	L2I	L6I	L7I	S2I	S6I	S7I		SYS / # / OBS TYPES
7.	E	16	C1X	C5X	C7X	C8X	D1X	D5X	D7X	D8X	L1X	L5X	L7X	L8X	S1X	SYS / # / OBS TYPES
8.			S5X	S7X	S8X											SYS / # / OBS TYPES
9.	S	4	C1C	D1C	L1C	S1C										SYS / # / OBS TYPES

以“SYS/#/OBS TYPES”结尾，所占列数 80 列，一行最多存放一类卫星的 13 个不同观测值，剩下观测值另起一行填写。在本实验中，我们需要读取北斗卫星的 C2I、C6I、L2I 和 L6I，其中 C2I、C6I 为伪距观测值，分别属于 B1 和 B3 频段，L2I 和 L6I 为载波相位观测值，分别属于 B1 和 B3 频段；还需要读取 GPS 卫星的 C1C、C2W、L1C 和 L2W，C1C 和 L1C 分别为 L1 频段的伪距和相位观测值，C2W 和 L2W 是 L2 频段的伪距和相位观测值。需要注意的是载波相位观测值以周(cycle)为单位，计算时需要进行单位转换。

RINEX3.03 观测值文件数据体结构较以往 RINEX 格式有了很大改变，一个历元的观测值以时间和观测卫星数开头：

1. > 2020 01 01 00 00 0.0000000 0 40

一颗卫星的观测值数据独占一行，长度不限。一个观测值数据占 16 位，分为观测值、LLI 和信号强度。值得注意的是，一颗卫星在一个历元中可能缺少某一种

观测值，因此不能按照空格读取，需要按照文件头给出的顺序，按字符串占位读取。

## 2.2 GF 与 MW 组合观测值计算

对于一颗双频点卫星，假设在某一时刻接收到其的频率为 $f_1$ 和 $f_2$ ，伪距观测值为 $P_1$ 和 $P_2$ ，以米为单位的载波相位观测值为 $L_1$ 和 $L_2$ ，具体表达式为：

$$P_i = \rho_r^s + dt_r - dt^s + d_{r,f_i} - d_{f_i}^s + d\rho_r^s + I_{r,f_i}^s + T_r^s + e_{r,f_i}^s \quad (1)$$

其中 $\rho_r^s$ 为真实卫地距， $dt_r$ 和 $dt^s$ 分别为接收机钟差和卫星钟差， $d_{r,f_i}$ 和 $d_{f_i}^s$ 分别为接收机端硬件延迟和卫星端硬件延迟， $d\rho_r^s$ 为卫星轨道误差， $I_{r,f_i}^s$ 是电离层延迟误差， $T_r^s$ 为对流层延迟误差， $e_{r,f_i}^s$ 为其他误差( $i = 1,2$ )。

$$L_i = \rho_r^s + dt_r - dt^s + \lambda_{f_i} \cdot (N_{r,f_i}^s + \delta_{r,f_i} - \delta_{f_i}^s) + d\rho_r^s - I_{r,f_i}^s + T_r^s + \varepsilon_{r,f_i}^s \quad (2)$$

其中 $\rho_r^s$ 为真实卫地距， $dt_r$ 和 $dt^s$ 分别为接收机钟差和卫星钟差， $\lambda_{f_i}$ 为载波波长， $\delta_{r,f_i}$ 和 $\delta_{f_i}^s$ 分别为接收机端硬件延迟和卫星端硬件延迟， $N_{r,f_i}^s$ 为整周未知数， $d\rho_r^s$ 为卫星轨道误差， $I_{r,f_i}^s$ 是电离层延迟误差， $T_r^s$ 为对流层延迟误差， $\varepsilon_{r,f_i}^s$ 为其他误差( $i = 1,2$ )。

### 2.2.1 GF 组合

GF 组合，即消几何距离的组合观测值(geometry-free)，又称为“电离层残差组合”。组合观测值 $L_{gf}$ 和 $P_{gf}$ 计算如下：

$$L_{gf} = L_1 - L_2 = N_{r,f_1}^s \lambda_{f_1} - N_{r,f_1}^s \lambda_{f_1} + A \frac{f_1^2 - f_2^2}{f_1^2 f_2^2} \quad (3)$$

$$P_{gf} = P_1 - P_2 = -A \frac{f_1^2 - f_2^2}{f_1^2 f_2^2} \quad (4)$$

其中， $A$ 为电离层延迟系数， $A = 40.3 \cdot TEC$ 。

由于载波相位观测值噪声比伪距观测值噪声更小，因此常用式(3)作为 GF 组合观测值计算。

GF 组合观测值与接收机至卫星的几何距离无关，消除了轨道误差、接收机钟差、卫星钟差和对流层延迟误差的影响，但是其整周模糊度失去了整数特性。由于 GF 组合观测值包含电离层的残差，常用于电离层 TEC 的计算以及数据编辑和周跳检测。

### 2.2.2 MW 组合

MW 组合即 Melbourne-Wubben 组合，由不同类型不同频率观测值间的线性组合。

$$L_{MW} = \frac{1}{f_1 - f_2} (f_1 L_1 - f_2 L_2) - \frac{1}{f_1 + f_2} (f_1 P_1 - f_2 P_2) = \lambda_{MW} (N_{r,f_1}^s - N_{r,f_2}^s) \quad (5)$$

其中， $\lambda_{MW} = \frac{c}{f_1 - f_2} = 0.86m$ ，为 MW 组合观测值的波长。

MW 组合观测值消除了电离层、对流层、接收机和卫星钟差以及测站卫星几何距离等误差影响，仅受观测噪声和多路径效应影响；该组合观测值的计算结果只包含宽巷模糊度参数，常用于整周模糊度的确定和周跳探测。但由于 MW 组合观测值计算过程引入了伪距观测值，观测噪声较大。

### 2.2.3 组合观测值历元间求差分

理想情况下，在短时间内，整周模糊度和电子总含量 TEC 保持稳定，相邻历元间组合观测值求差分能够快速捕获整周模糊度变化和电离层波动。

## 3 编程解算

### 3.1 常量定义

由于解算时需要 GPS 的 L1 和 L2 频段频率以及 BDS 的 B1 和 B3 频率，宏定义如下常量方便后续解算：

```
1. #define GPS_L1 1575.42e6 // L1 频率
2. #define GPS_L2 1227.60e6 // L2 频率
3. #define BDS_B1 1561.098e6 // B1 频率
4. #define BDS_B3 1268.52e6 // B3 频率
5. #define c_LIGHT 299792458.0 // 光速
```

### 3.2 函数定义

本次实验主要定义了两个函数，一个用于提取观测值索引，一个用于按照索引提取观测值。

```
void ObsIndex(const vector<string>& types, char sys, int& idxP1, int& idxP2, int& idxL1, int& idxL2)
double extractObs(const string& line, int index)
```

具体定义如下：

```
// 解析观测类型并提取关键索引
void ObsIndex(const vector<string>& types/*观测值类型*/, char sys, int& idxP1, int& idxP2, int& idxL1, int& idxL2)
{
    idxP1 = idxP2 = idxL1 = idxL2 = -1; //初始化索引
    for (int j = 0; j < (int)types.size(); ++j) {
        const string& t = types[j];
        if (sys == 'G') {
            if (t == "C1C") idxP1 = j;
            if (t == "C2W") idxP2 = j;
            if (t == "L1C") idxL1 = j;
            if (t == "L2W") idxL2 = j;
        }
        else if (sys == 'C') {
            if (t == "C2I") idxP1 = j;
            if (t == "C6I") idxP2 = j;
            if (t == "L2I") idxL1 = j;
            if (t == "L6I") idxL2 = j;
        }
    }
}
```

图 3-1 提取索引函数

ObsIndex 函数的作用是解析观测值类型并提取关键索引。传递参数中 types 为该卫星系统的所有观测值类型,通过主函数读取文件头得到;sys 为系统类型,主要用于区分北斗和 GPS 系统;后四个参数分别为两个伪距和两个载波相位观测值的索引。通过遍历 types,找到对应观测值的索引,方便观测值后续截取。

```
double extractObs(const string& line, int index) //提取观测值
{
    int start = index * 16 + 3; //观测值起始位置
    if (start + 14 > (int)line.size()) return 0.0;
    string obs = line.substr(start, 14);
    return atof(obs.c_str());
}
```

图 3-2 按照索引提取观测值函数

extractObs 函数用于从字符串中截取观测值,形参 line 是读取得到的一行观测值,index 是所需观测值的索引。在 RINEX3.03 观测值文件中,一个观测值占 16 位,其中 14 位是我们需要读取的,第 15 位为 LLI,第 16 位为信号强度。前三位为卫星的 PRN,所以 start 从 3 开始,加上索引的 16 倍,每次读 14 位便可以获取到特定索引的观测值。

### 3.3 主函数

主函数负责读取文件、提取观测值和进行组合观测值计算。读取文件又分为读取文件头和数据体,读取文件头时读取北斗和 GPS 观测值类型并将其保存在其分别保存在 obsTypesG 和 obsTypesC 中。按行读取数据体时,如果不是北斗卫星和 GPS 跳过该观测值,如果是北斗卫星或 GPS 卫星则按照索引提取相应观测值进行解算。

```

int main() {
    ifstream ifs("JFNG00CHN_R_20200010000_01D_30S_MO.20o");
    ofstream out("result.txt"); // 输出结果文件
    string line;
    vector<string> obsTypesG, obsTypesC;
    while (getline(ifs, line))
    {
        if (line.find("SYS / # / OBS TYPES") != string::npos)
        {
            char sys = line[0];
            if (sys != 'G' && sys != 'C') continue;
            int n = stoi(line.substr(3, 3));
            vector<string> tempTypes; //临时存储观测值类型
            int count = 0;
            do {
                for (int i = 0; i < 13 && count < n; ++i, ++count) //一行最多读13个
                {
                    string t = line.substr(7 + i * 4, 3); //提取观测值类型
                    tempTypes.push_back(t);
                }
                if (count < n) getline(ifs, line);
            } while (count < n);
            if (sys == 'G') obsTypesG = tempTypes;
            if (sys == 'C') obsTypesC = tempTypes;
        }
        else if (line.find("END OF HEADER") != string::npos) {
            break;
        }
    }
}

```

图 3-3 读取文件头并提取观测值类型

```

int idxP1G, idxP2G, idxL1G, idxL2G, idxP1C, idxP2C, idxL1C, idxL2C;
ObsIndex(obsTypesG, 'G', idxP1G, idxP2G, idxL1G, idxL2G);
ObsIndex(obsTypesC, 'C', idxP1C, idxP2C, idxL1C, idxL2C);

```

图 3-4 提取索引

```

while (getline(ifs, line)) {
    if (line.empty() || line[0] != '>') continue; //跳过非观测数据行
    int numSats = stoi(line.substr(32, 3)); //卫星数量
    int hour = stoi(line.substr(13, 2)); //提取时
    int min = stoi(line.substr(16, 2)); //提取分
    double sec = stof(line.substr(19, 9)); //提取秒
    for (int i = 0; i < numSats; ++i)
    {
        getline(ifs, line); //读取一行卫星观测数据
        if (line.size() < 3) continue;
        string prn = line.substr(0, 3);
        char sys = prn[0];
        if (sys != 'G' && sys != 'C') continue;
        int idxP1 = -1, idxP2 = -1, idxL1 = -1, idxL2 = -1;
        double f1 = 0, f2 = 0;
        vector<string>* obsTypes = nullptr; //观测值类型指针
        //根据系统选择频率和索引
        if (sys == 'G')
        {
            idxP1 = idxP1G; idxP2 = idxP2G;
            idxL1 = idxL1G; idxL2 = idxL2G;
            f1 = GPS_L1; f2 = GPS_L2;
            obsTypes = &obsTypesG;
        }
        else if (sys == 'C')
        {
            idxP1 = idxP1C; idxP2 = idxP2C;
            idxL1 = idxL1C; idxL2 = idxL2C;
            f1 = BDS_B1; f2 = BDS_B3;
            obsTypes = &obsTypesC;
        }
    }
}

```

图 3-5 读取数据记录索引

```

if (idxP1 < 0 || idxP2 < 0 || idxL1 < 0 || idxL2 < 0) continue;
//按照索引提取观测值
double L1 = extractObs(line, idxL1);
double L2 = extractObs(line, idxL2);
double P1 = extractObs(line, idxP1);
double P2 = extractObs(line, idxP2);
if (L1 == 0 || L2 == 0 || P1 == 0 || P2 == 0) continue;
//计算MW和GF
double lambda1 = c_LIGHT / f1;
double lambda2 = c_LIGHT / f2;
double L1_m = L1 * lambda1;
double L2_m = L2 * lambda2;
double mw = (f1*L1_m - f2*L2_m)/(f1-f2) - (f1 * P1 + f2 * P2)/ (f1 + f2);
double gf = L1_m - L2_m;
//保存到文件
out << hour << " " << min << " " << setprecision(1) << fixed << sec << " ";
out << prn << " " << setprecision(3) << fixed << mw << " " << gf << endl;
cout << prn << fixed << setprecision(3) << " MW: " << mw << " GF: " << gf << endl;

```

图 3-6 按索引提取观测值并解算

### 3.4 数据可视化

历元 MATLAB 对提取到的数据进行可视化处理，代码较为繁琐，在附件中给出。

## 4 结果分析

在理想情况下，一颗卫星的 GF 和 MW 组合观测值应为常数，而在实际情况中，往往会因为周跳、电离层波动和高度角变化引起组合观测值波动。

结果分析过程中，北斗的 GEO 卫星我选取了 C01 和 C02; IGSO 选择了 C06 和 C08; MEO 选择了 C11 和 C12。而 GPS 卫星，我则选取了 G02、G05、G13、G15 和 G20。

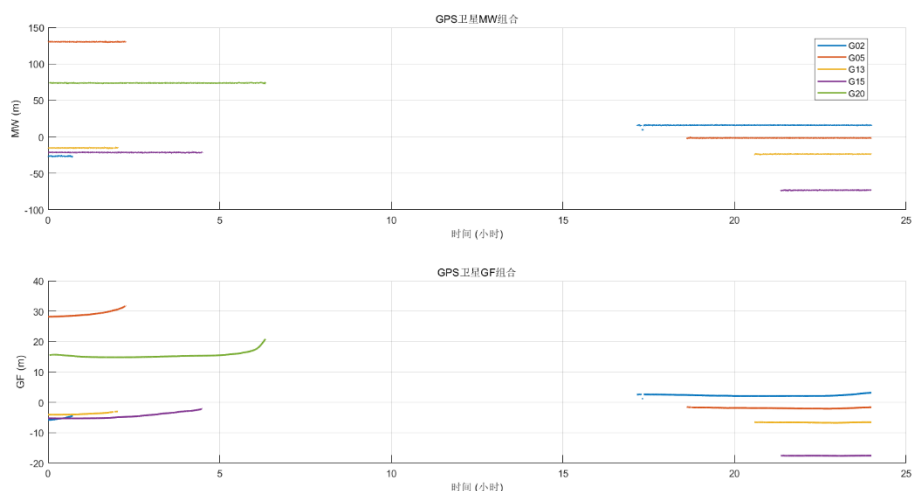


图 4-1 GPS 卫星 MW 和 GF 组合观测值



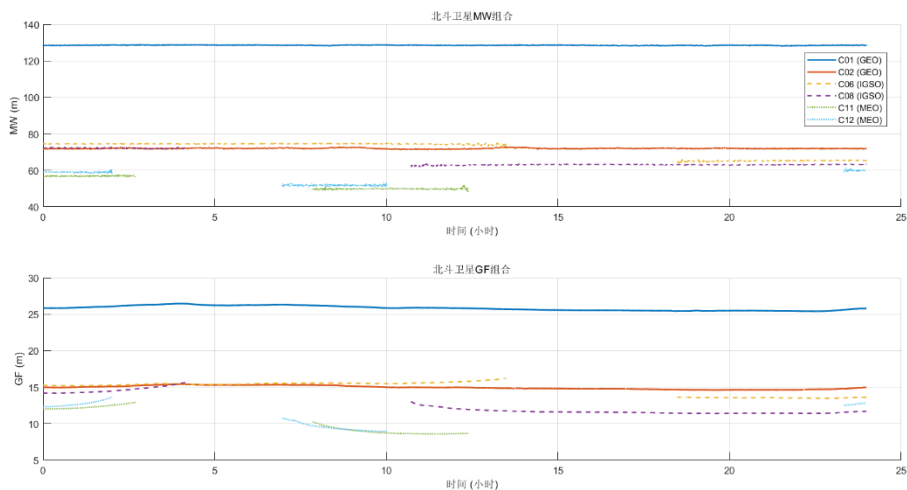


图 4-2 BDS 卫星 MW 和 GF 组合观测值

由于 MEO 和 IGSO 并不总在测站观测范围内，因此存在部分时间段缺少观测值情况，而 GEO 总是相对测站静止，因此观测值总是存在。

通过时序图分析可得，在信号出现和消失前后观测值会出现较大变化，主要是由于高度角过低引起的信号噪声的增加以及电离层延迟误差增加，从而影响 MW 和 GF 组合观测值。

为了更好地进行周跳探测，需要绘制 MW 和 GF 差分时序图，

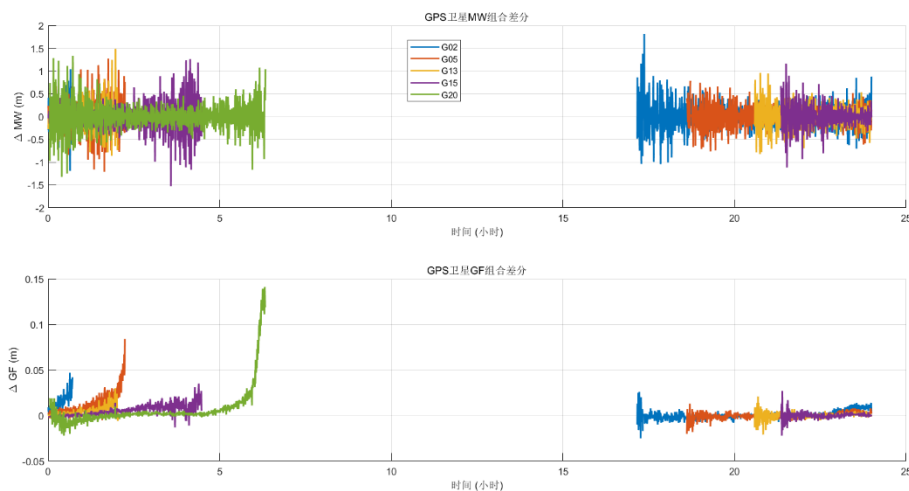


图 4-3 GPS 卫星 MW 和 GF 差分观测值

从图 4-3 我们可以更加直观地看到卫星处于低高度角时误差增大的情况。由 GPS 卫星的 MW 组合差分图可以看到，未出现 MW 差分值很大的情况，可能并未出现周跳，因此影响 GF 观测值质量下降的因素为低高度角时电离层延迟增大，引起粗差。

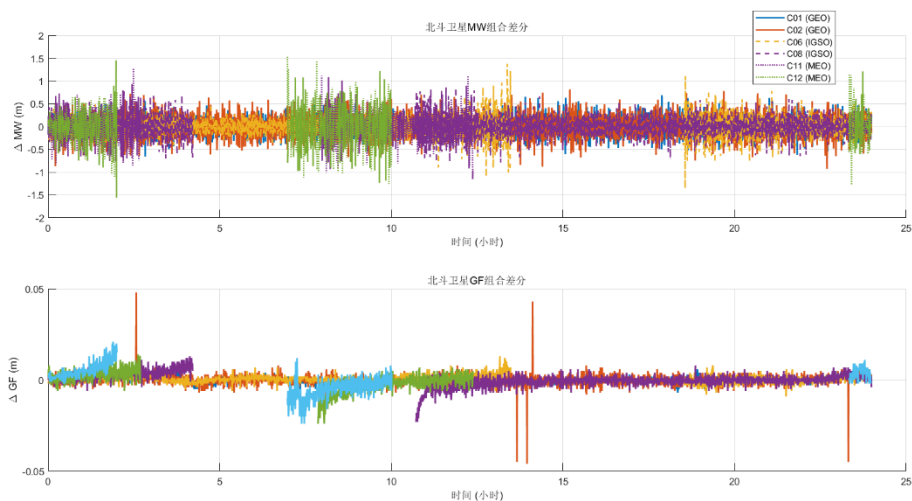


图 4-4 BDS 卫星 MW 和 GF 差分观测值

图 4-4 展示了 BDS 卫星的 MW 和 GF 差分观测值。C02 卫星 GF 差分出现 5 次跳变，且跳变误差近乎相同，有理由认为 C02 号卫星出现了 5 次周跳。其余卫星也存在低高度角误差增大的情况，与 GPS 卫星趋势相同。

## 5 实验总结与反思

本次实验完成了从 RINEX3.03 观测值文件中提取观测值进行 GF 和 MW 组合观测值计算的任务，进一步加深了我们对 RINEX 格式文件的熟悉程度和对一些常见组合观测值的理解。GF 和 MW 组合常常被用作周跳和粗差探测，以及电离层波动检测。本次实验为我们今后进行卫星导航研究打下了很好的基础。

本次实验的难点在于读取文件并提取观测值。由于观测值文件不是按照空格对观测值进行分类，而是按照一定索引进行排列，与我们之前接触的文件有较大不同，需要我们去进一步学习相关操作，提高了我们的编程能力。

本次实验过程中我发现了自身的许多不足之处，包括对文件格式不够了解，对算法流程不够清晰等。通过查阅相关文件与博客，同时也向老师询问，很好的解决了这些问题，也提高了自身的能力。

## 参考

[https://blog.csdn.net/qq\\_32109917/article/details/](https://blog.csdn.net/qq_32109917/article/details/)

<https://download.csdn.net/download/tuoyakan9097/11168584>