



上海交通大学  
SHANGHAI JIAO TONG UNIVERSITY

# 铱星信号处理及定位

汇报人 周胜杰



# 目 录

---



01-铱星星座介绍

02-信号处理流程

03-定位原理

04-代码简介

01

# 铱星星座介绍



# LEO卫星系统

- 随着各个低地球轨道（low-earth orbit, LEO）卫星系统的建立健全，LEO卫星系统被应用于导航、通讯等各个领域，吸引了广泛的关注



## 信号强

LEO卫星轨道高度较低，LEO卫星传播信号达到接收机过程中传播距离更短，信号的能量损耗少，信号强度高，不易受到干扰与欺骗。



## 分布变化快

LEO卫星运行速度快，具有更短的轨道周期，分布多样且变化快。



## 数量和频点资源丰富

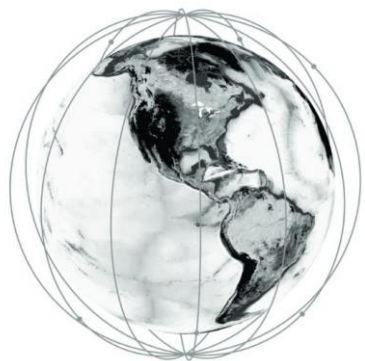
LEO卫星系统较多，且部分星座为超级卫星星座（satellite mega-constellations），现有的LEO卫星数量众多，卫星信号占据的频段丰富。随着LEO系统的不断发展和完善，LEO卫星系统资源的丰富性将进一步被拓展。

- LEO卫星星座可以被作为常规GNSS系统的辅助，用于增强系统的鲁棒性或者增加载波相位定位系统的整周模糊度求解速度或精密单点定位的收敛速度等服务

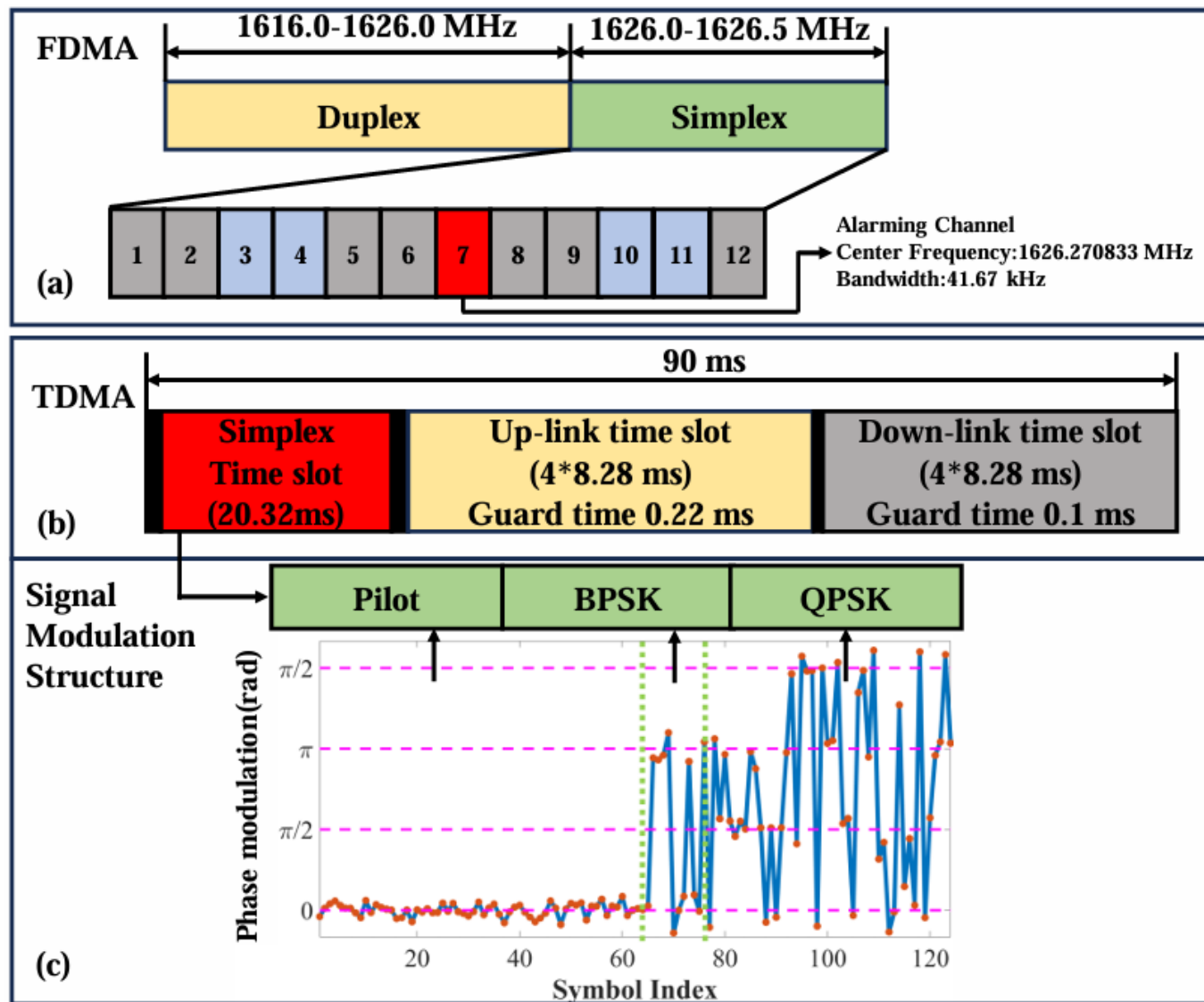
# 铱星卫星系统

铱星星座信息表

类型	值
星座构型	近极圆轨道
轨道高度	约780km
FDMA结构中用于定位的部分	单工第七信道 (振铃)
TDMA结构中用于定位的部分	每一帧的单工时隙 (位于每一帧信号前20.32ms)
信号类型	突发信号, 每一帧时长90ms
用于定位的信号调制方式	Pilot+BPSK+QPSK
码率	25000 Symbol/s



铱星星座构型示意图



铱星信号结构示意图

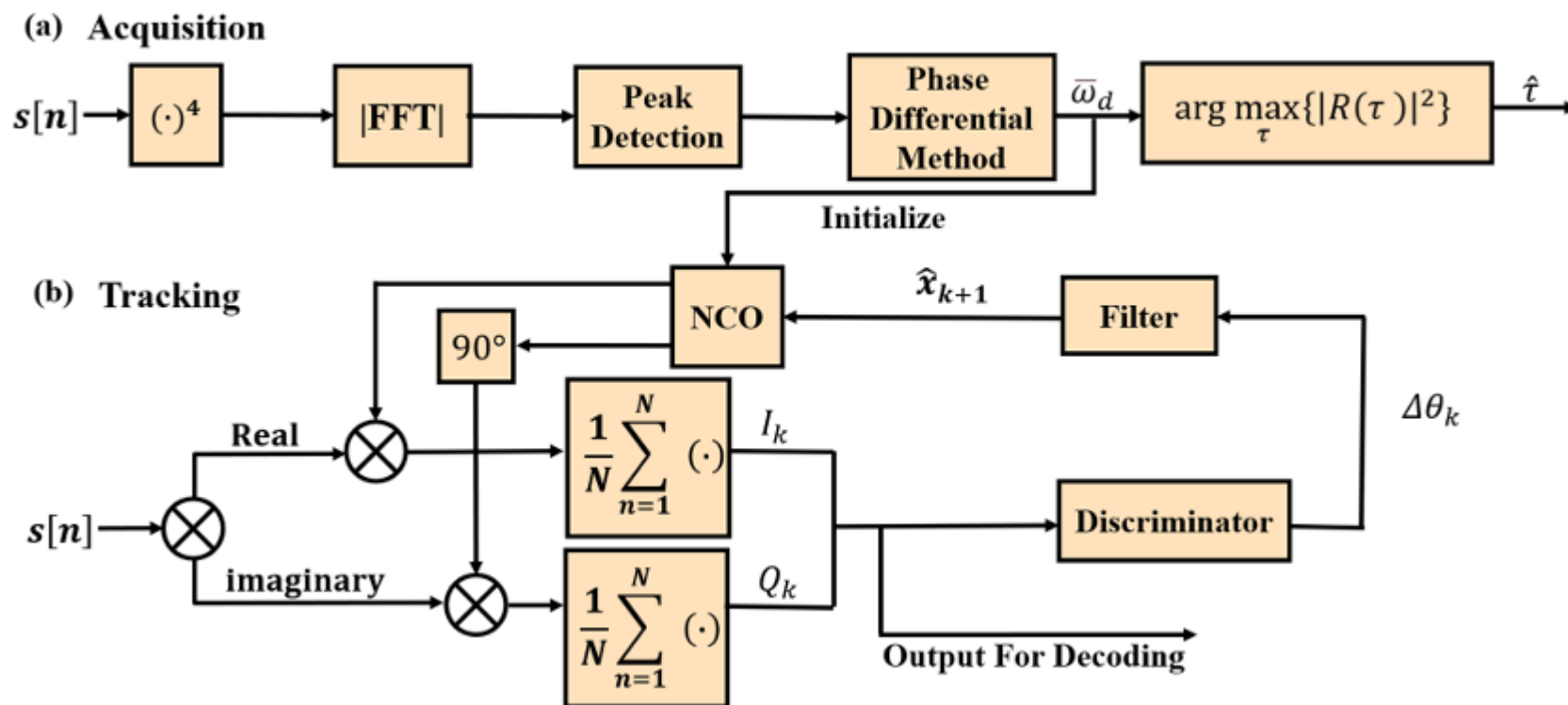
02

## 信号处理流程



# 铱星信号处理

- 对于基带处理，其目的为得到信号多普勒频移以及信号传输延迟的估计。实际操作时可以将两者的估计进行解耦。信号处理的流程图如下

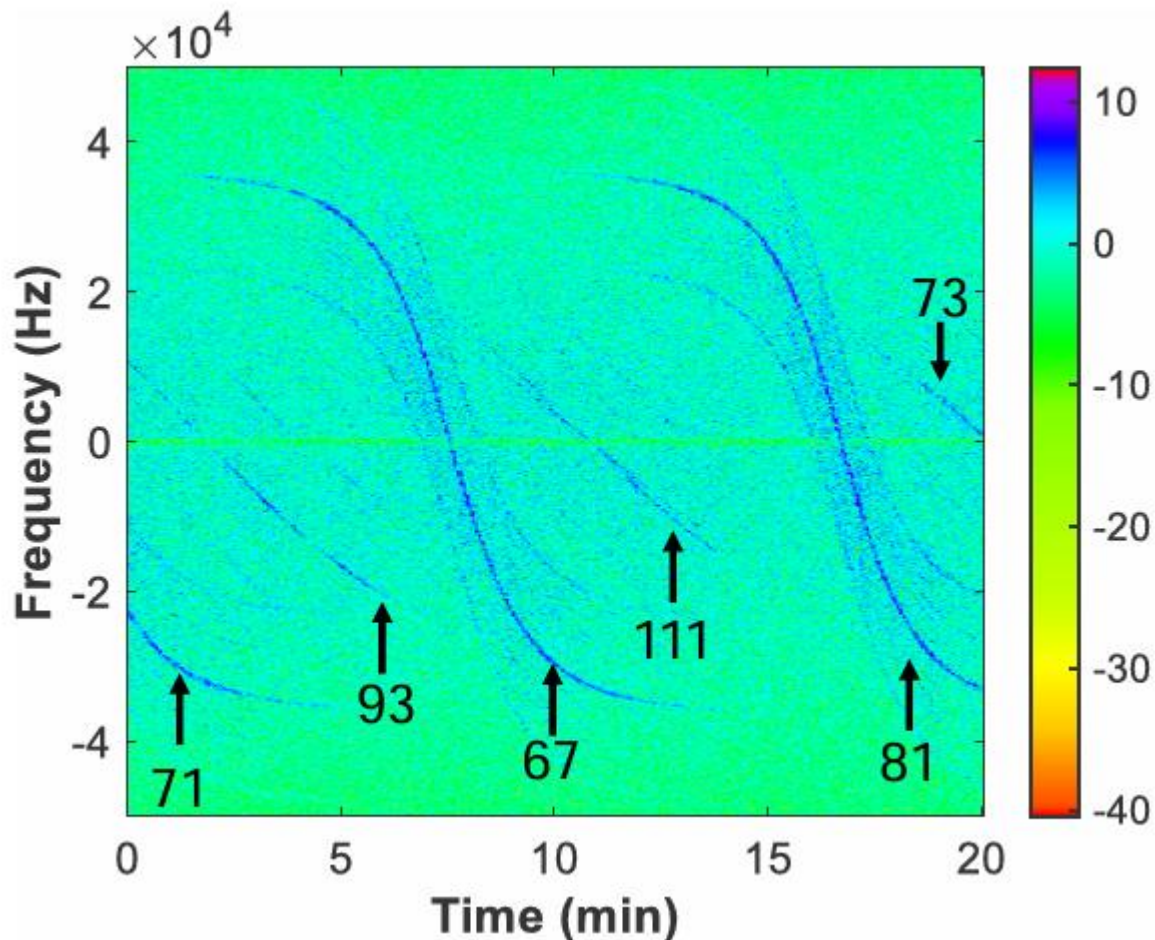


铱星信号处理流程示意图



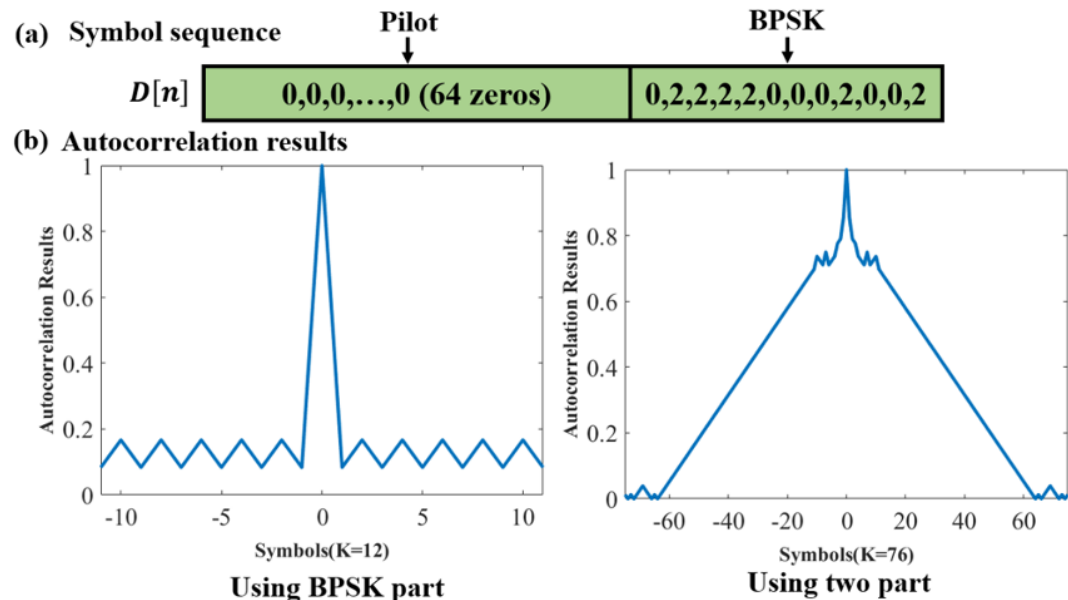
# 铱星信号捕获

- 捕获阶段，首先对信号进行四次方去除调制（等于把信号频率变成四倍），对信号进行FFT得到粗捕获频率（由于粗估计精度要求不高，不进行四次方结果误差可接受）



铱星信号频谱瀑布图（按垂直于x轴方向截取即为信号的FFT结果）

- 由于铱星信号的Pilot和BPSK部分固定，且存在较好的自相关特性，可以利用该特性将接收到信号与本地信号进行相关。铱星信号的自相关性示意图如下



铱星信号自相关特性示意图



# 铱星信号捕获公式表达

- 为描述准确，将捕获部分进行公式化表达。信号可建模为：

$$s[n] = \sqrt{C} m \left[ n - \frac{\tau[n]}{T_s} \right] \exp \left\{ j \left( \omega_{IF} + \omega_d[n] + \frac{\dot{\omega}_d[n] n T_s}{2} \right) n T_s + j \varphi[n] \right\} + v_s[n]$$

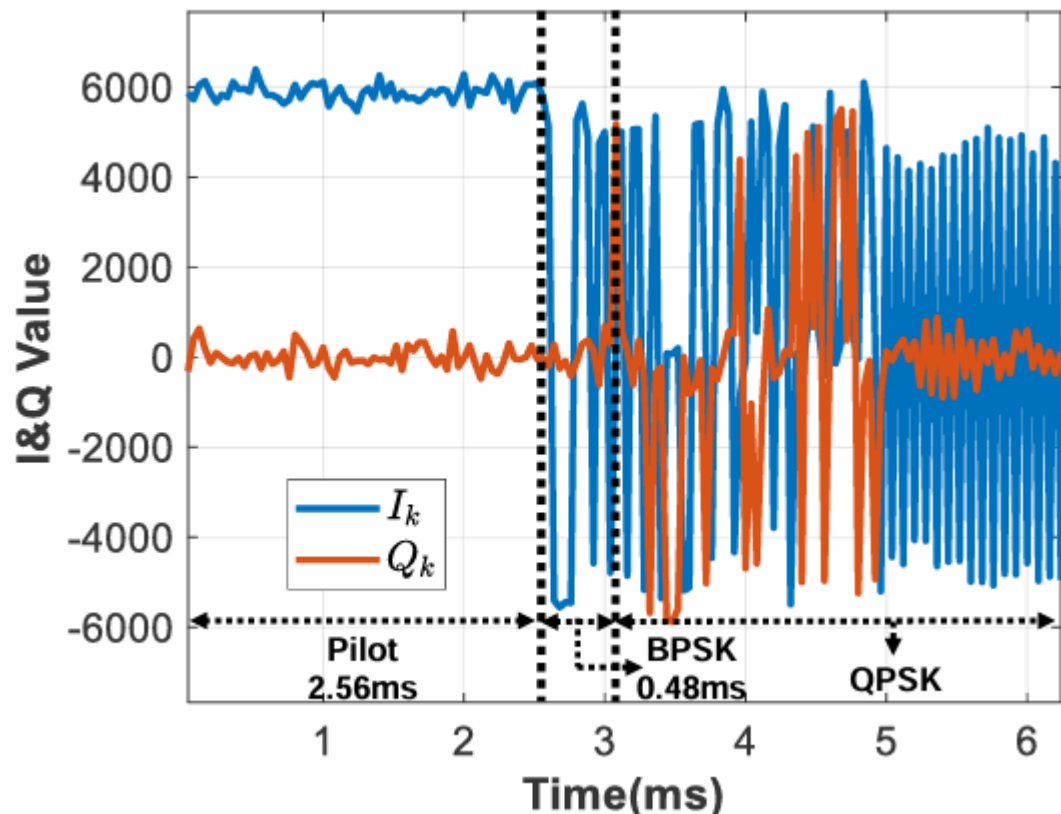
- 其中， $n$ 表示第 $n$ 个采样点， $C$ 表示信号功率， $T_s$ 表示采样间隔。 $\tau$ 表示信号从发射端到接收端的时延， $\omega_{IF}$ 表示中频信号频率， $\omega_d$ 表示多普勒频移， $\dot{\omega}_d$ 表示多普勒频移变化率， $\varphi$ 表示信号初始相位， $V_s$ 表示信号噪声。
- $m$ 表示信号中的相位调制，可表示为： $m[n] = \exp \left\{ j \left( \frac{\pi}{2} D[n] \right) \right\}$  其中symbol电文为  $D[n] \in \begin{cases} \{0\}(\text{Pilot}) \\ \{0, 2\}(\text{BPSK}) \\ \{0, 1, 2, 3\}(\text{QPSK}) \end{cases}$
- 则捕获过程可表示为： $(\hat{\tau}, \hat{\omega}_d) = \arg \max_{\tau, \omega_d} \left\{ \left| R(\tau, \omega_d) \right|^2 \right\}$
- 将频率估计和时延估计解耦后可表达为： $R(\tau) = \frac{1}{KN} \sum_{n=0}^{KN-1} s[n] m[n] \exp \{ -j \bar{\omega}_d n T_s \}$  其中 $K$ 表示运用了多少个symbol， $N$ 对应一个Symbol的采样点数量  $\bar{\omega}_d$ 是FFT粗捕获的结果

# 铱星信号跟踪

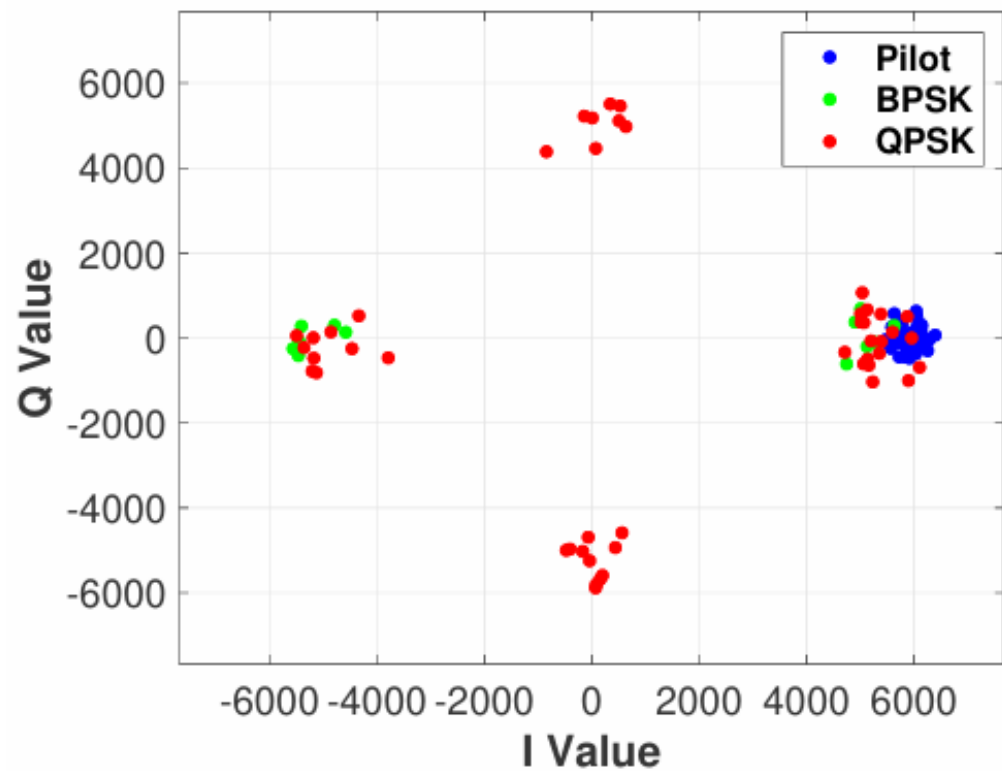
- 采用三阶锁相环对信号的相位、频率、频率变化率进行跟踪。每个粒元处理时间为一个symbol的持续时间（0.04ms），其状态量可表示为： $\mathbf{x}_k = [\varphi \quad \omega_d \quad \dot{\omega}_d]^T_k$
- 将接收到信号和本地信号相关（对于相关的详细原理描述请参考附带的《GPS原理与接收机设计》），有：
$$R_k = \frac{1}{N} \sum_{n=(k-1)N+1}^{kN} s[n] \exp \left\{ -j \left( \hat{\varphi}_k + \hat{\omega}_{dk} n T_s + \dot{\omega}_{dk} \frac{(n T_s)^2}{2} \right) \right\}$$
- $R_k$ 为复数，可以将其实数和虚数部分分为两支路表示，即： $R_k = I_k + jQ_k$
- 对相关后的结果使用鉴相器：
$$\Delta\theta_k = \begin{cases} \text{atan2}(Q_k, I_k) & (\text{Pilot}) \\ \text{atan}(Q_k/I_k) & (\text{BPSK \& QPSK}) \end{cases}$$
- 得到鉴相结果后使用比例积分滤波器即可得到下一粒元时刻的估计量： $\hat{\mathbf{x}}_{k+1} = \mathbf{A}\hat{\mathbf{x}}_k + \mathbf{A}\mathbf{L}\Delta\theta_k$   
其中 $\mathbf{A}$ 为状态转移矩阵， $\mathbf{L}$ 为滤波增益矩阵。

# 铱星信号跟踪结果展示

- 对于某一帧铱星信号跟踪结果如下。左图为IQ两支路随粒元变化情况，右图为其在复平面的分布情况。完成捕获跟踪即可得到相应的TOA和FOA观测量，进而进行定位



铱星信号IQ支路随时间变化图



铱星信号跟踪结果复平面分布图

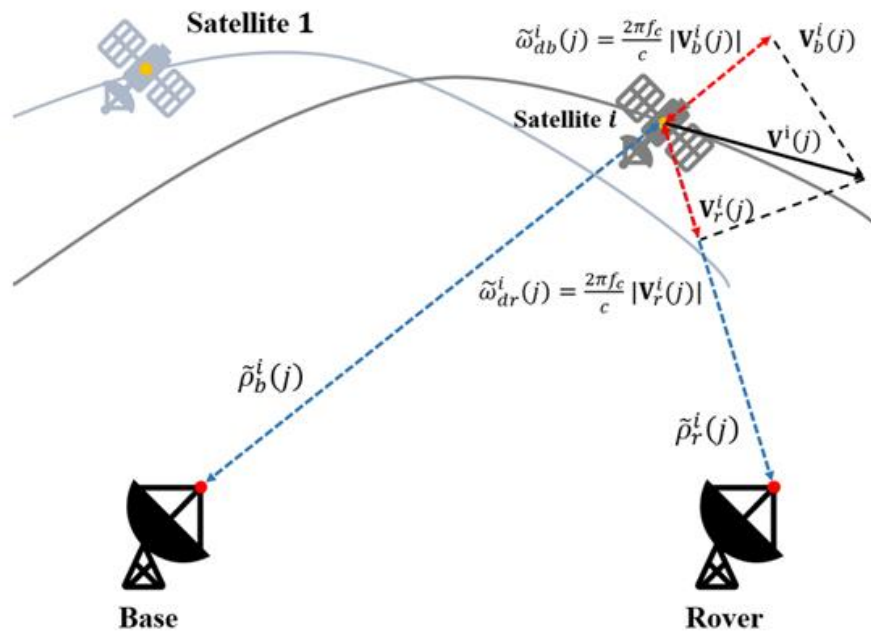
# 03

## 定位原理

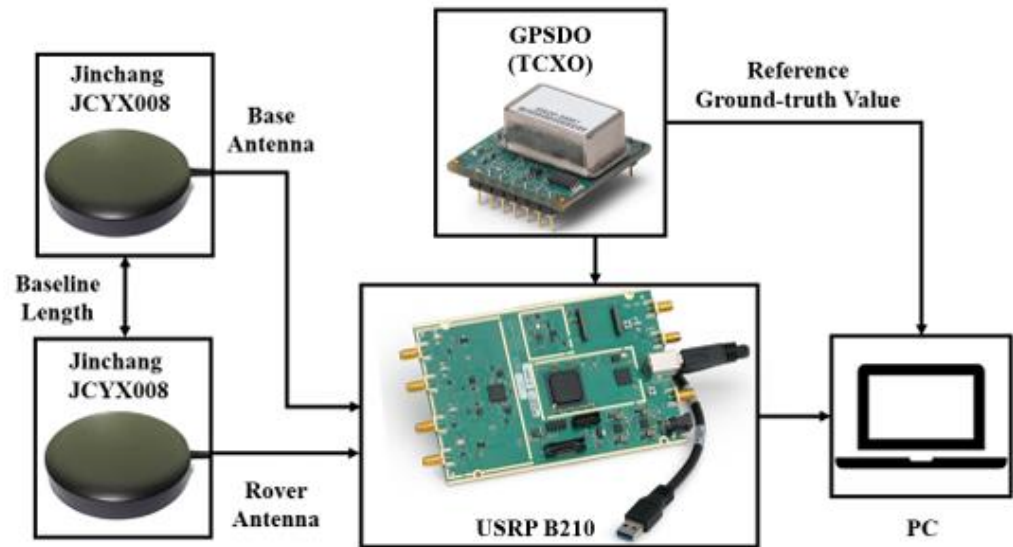


# 定位

- 本小节会介绍TOA、TDOA、FOA、FDOA四种定位方法。由于涉及差分定位，在接收信号时使用两个天线。实验时接收信号的示意图如左下图所示，右下图为实验设备



实验接收信号示意图



实验设备分布图

# 定位

□ 使用最小二乘方法进行定位估计。TOA定位和TDOA需要估计的状态为： $\mathbf{Y} = [\mathbf{P}_r^T, c\delta t_r]^T$

其中  $\mathbf{P}_r = [x_r, y_r, z_r]^T$  为接收机在ECEF坐标系中的坐标， $c\delta t_r$  为钟差造成的影响。

□ FOA和FDOA定位需要估计的状态为： $\mathbf{Y} = [\mathbf{P}_r^T, c\delta \dot{t}_r]^T$

其中  $\mathbf{P}_r = [x_r, y_r, z_r]^T$  为接收机在ECEF坐标系中的坐标， $c\delta \dot{t}_r$  为钟漂造成的影响。

□ 定位的目的为使得状态的估计值  $\hat{\mathbf{Y}}$  和实际值的差值  $\Delta \mathbf{Y} = \mathbf{Y} - \hat{\mathbf{Y}}$  的绝对值最小。该差值和观测量残差有关，定义接收机位置真值和TLE估计的卫星之间的理论观测量为  $\mathbf{Z}$ ，实验估计得到的观测量为  $\hat{\mathbf{Z}}$ ，则观测量残差为  $\Delta \mathbf{Z} = \mathbf{Z} - \hat{\mathbf{Z}}$ 。则观测量残差和定位状态差值的关系为： $\Delta \mathbf{Z} = \mathbf{G} \Delta \mathbf{Y} + \nu_z$ ，其中  $\mathbf{G}$  为雅可比矩阵，则  $\Delta \hat{\mathbf{Y}} = (\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T \Delta \mathbf{Z}$

□ 以  $|\Delta \mathbf{Y}| < 10^{-6}$  为收敛条件，反复迭代即可得到最终的定位结果

□ 不同的定位方法的  $\Delta \mathbf{Z}$  和  $\mathbf{G}$  不同，以不同的下标进行区分，并逐一进行介绍

# TOA定位

TOA观测量为信号从发射端到接收端的时延。在信号捕获阶段已经得到了部分时延，进一步计算得到完整的时延。

如左下图所示，信号在接收端的时间有： $\hat{t}_r(j) = \tilde{t}_r(j) + \hat{\tau}(j)$  其中  $\tilde{t}_r(j)$  是参考的时间基准， $\hat{\tau}(j)$  为捕获得到的部分时延， $j$  指第  $j$  帧信号得到的观测量

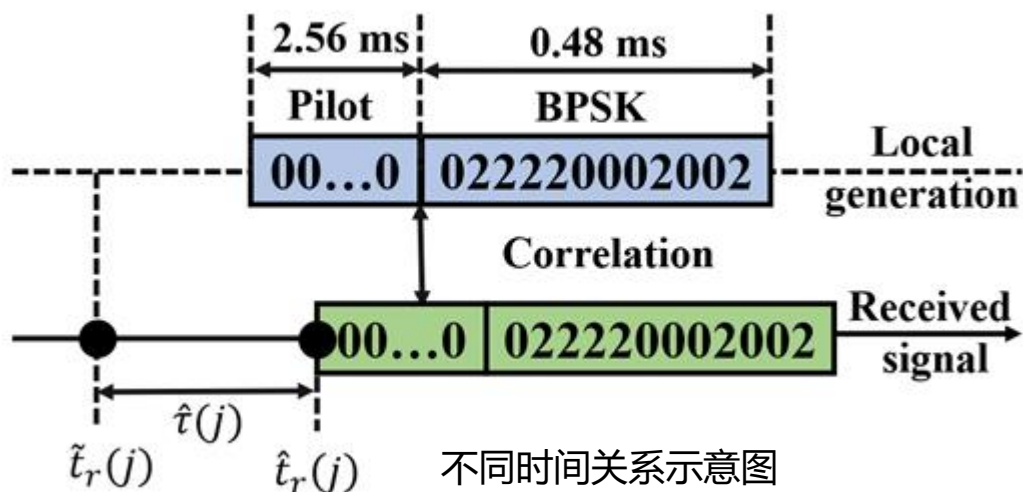
根据接收端时间和接收机位置真值，能得到发射端时间的粗略估计： $\tilde{t}_s(j) = \hat{t}_r(j) - \frac{\tilde{\rho}(j)}{c}$

$c$  为光速， $\tilde{\rho}(j)$  为根据接收机位置真值和TLE星历估计的卫星和接收机之间的距离  $\tilde{\rho}(j) = |\mathbf{P}^i(j) - \mathbf{P}_r(j)|$  其中  $\mathbf{P}_r(j)$  为接收机位置， $\mathbf{P}^i(j)$  为卫星位置。

由于不同帧之间发射时间存在固定间隔，即：

$$t_s(j) = t_s(1) + (j-1)90ms$$

可用多星多粒元的观测量对发射世家进行联合估计





# TOA定位

□ 由于不同帧之间发射时间存在固定间隔，

即： $t_s(j) = t_s(1) + (j-1)90ms$

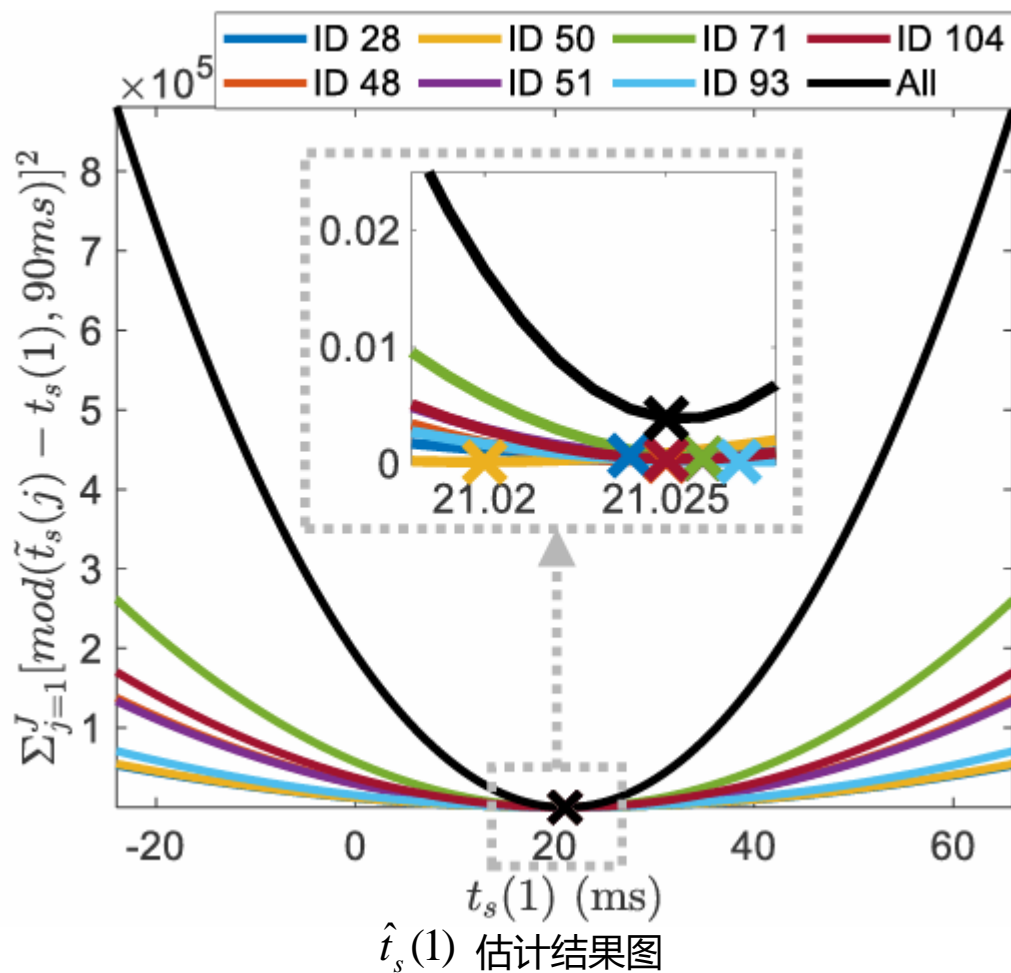
□ 可用多星多粒元的观测对某次发射进行联合估计：

$$\begin{aligned}\hat{t}_s(1) &= \arg \min_{t_s(1)} \left\{ \sum_{j=1}^J [\tilde{t}_s(j) - t_s(j)]^2 \right\} \\ &= \arg \min_{t_s(1)} \left\{ \sum_{j=1}^J [\tilde{t}_s(j) - t_s(1) - (j-1)90ms]^2 \right\} \\ &= \arg \min_{t_s(1)} \left\{ \sum_{j=1}^J [\text{mod}(\tilde{t}_s(j) - t_s(1), 90ms)]^2 \right\}\end{aligned}$$

□ 则每一帧的TOA观测可以简化为：

$$\begin{aligned}\hat{\tau}(j) &= \hat{t}_r(j) - \hat{t}_s(j) \\ &= \hat{t}_r(j) - \hat{t}_s(1) - (j-1)90ms \\ &= \text{mod}(\hat{t}_r(j) - \hat{t}_s(1), 90ms)\end{aligned}$$

□  $\hat{t}_s(1)$  已经估计，为21.025ms



# TOA定位

- 得到TOA观测量后，对于TOA定位方法，观测量残差为：

$$\Delta \mathbf{Z}_\tau = \begin{bmatrix} c\hat{\tau}_r^1(1) - \tilde{\rho}_r^1(1) \\ c\hat{\tau}_r^2(1) - \tilde{\rho}_r^2(1) \\ \dots \\ c\hat{\tau}_r^I(1) - \tilde{\rho}_r^I(1) \\ \dots \\ c\hat{\tau}_r^1(J) - \tilde{\rho}_r^1(J) \\ c\hat{\tau}_r^2(J) - \tilde{\rho}_r^2(J) \\ \dots \\ c\hat{\tau}_r^I(J) - \tilde{\rho}_r^I(J) \end{bmatrix}$$

- $\tilde{\tau}_r^i(j)$ 指rover接收到的第j帧信号的TOA观测， $\tilde{\rho}_r^i(j)$ 指根据TLE星历估计的距离

- 雅可比矩阵为：

$$\mathbf{G}_\tau = \begin{bmatrix} \mathbf{e}_r^1(1) & 1 \\ \mathbf{e}_r^2(1) & 1 \\ \dots & \dots \\ \mathbf{e}_r^I(1) & 1 \\ \dots & \dots \\ \mathbf{e}_r^1(J) & 1 \\ \mathbf{e}_r^2(J) & 1 \\ \dots & \dots \\ \mathbf{e}_r^I(J) & 1 \end{bmatrix}$$

- 雅可比矩阵中的元素指卫星指向估计的接收机位置的单位方向向量，为：

$$\mathbf{e}_r^i(j) = \frac{\partial \tilde{\rho}_r^i(j)}{\partial \hat{\mathbf{P}}_r(j)} = \frac{\hat{\mathbf{P}}_r - \mathbf{P}^i(j)}{|\hat{\mathbf{P}}_r - \mathbf{P}^i(j)|}$$

# TDOA定位

□ TDOA方法雅可比矩阵和TOA相同，观测量残差矩阵为：

$$\Delta \mathbf{Z}_{d\tau} = \begin{bmatrix} (c\hat{\tau}_r^1(1) - c\hat{\tau}_b^1(1)) - (\tilde{\rho}_r^1(1) - \tilde{\rho}_b^1(1)) \\ (c\hat{\tau}_r^2(1) - c\hat{\tau}_b^2(1)) - (\tilde{\rho}_r^2(1) - \tilde{\rho}_b^2(1)) \\ \dots \\ (c\hat{\tau}_r^I(1) - c\hat{\tau}_b^I(1)) - (\tilde{\rho}_r^I(1) - \tilde{\rho}_b^I(1)) \\ \dots \\ (c\hat{\tau}_r^1(J) - c\hat{\tau}_b^1(J)) - (\tilde{\rho}_r^1(J) - \tilde{\rho}_b^1(J)) \\ (c\hat{\tau}_r^2(J) - c\hat{\tau}_b^2(J)) - (\tilde{\rho}_r^2(J) - \tilde{\rho}_b^2(J)) \\ \dots \\ (c\hat{\tau}_r^I(J) - c\hat{\tau}_b^I(J)) - (\tilde{\rho}_r^I(J) - \tilde{\rho}_b^I(J)) \end{bmatrix}$$

□ 变量中的下标r表示rover，b表示base

# FOA定位

- FOA观测量为锁相环输出多普勒估计，对应的观测量残差矩阵为：

$$\Delta \mathbf{Z}_\omega = \begin{bmatrix} \hat{\omega}_{dr}^1(1) - \tilde{\omega}_{dr}^1(1) \\ \hat{\omega}_{dr}^2(1) - \tilde{\omega}_{dr}^2(1) \\ \dots \\ \hat{\omega}_{dr}^I(1) - \tilde{\omega}_{dr}^I(1) \\ \dots \\ \hat{\omega}_{dr}^1(J) - \tilde{\omega}_{dr}^1(J) \\ \hat{\omega}_{dr}^2(J) - \tilde{\omega}_{dr}^2(J) \\ \dots \\ \hat{\omega}_{dr}^I(J) - \tilde{\omega}_{dr}^I(J) \end{bmatrix}$$

- 雅可比矩阵为：

$$\mathbf{G}_\omega = \frac{2\pi f_c}{c} \begin{bmatrix} \dot{\mathbf{e}}_r^1(1) & 1 \\ \dot{\mathbf{e}}_r^2(1) & 1 \\ \dots & \dots \\ \dot{\mathbf{e}}_r^I(1) & 1 \\ \dots & \dots \\ \dot{\mathbf{e}}_r^1(J) & 1 \\ \dot{\mathbf{e}}_r^2(J) & 1 \\ \dots & \dots \\ \dot{\mathbf{e}}_r^I(J) & 1 \end{bmatrix}$$

- 雅可比矩阵中的元素指卫星指向估计的接收机位置的单位方向向量对时间导数，为：

$$\dot{\mathbf{e}}_r^i(j) = \frac{\partial \tilde{\omega}_{dr}^i(j)}{\partial \hat{\mathbf{P}}_r(j)} = \frac{1}{|\hat{\mathbf{P}}_r - \mathbf{P}^i(j)|} [\mathbf{e}_r^i(j) \times (\mathbf{e}_r^i(j) \times \mathbf{V}^i(j))]$$

- $\mathbf{V}^i(j)$  为：第j帧信号发射时卫星的速度

- $\hat{\omega}_{dr}^i(j)$  指rover接收到的第j帧信号的FOA观测， $\tilde{\omega}_{dr}^i(j)$  指根据TLE星历估计的多普勒

# FDOA定位

□ FDOA方法雅可比矩阵和FOA相同，观测量残差矩阵为：

$$\Delta \mathbf{Z}_{d\omega} = \begin{bmatrix} (\hat{\omega}_{dr}^1(1) - \hat{\omega}_{db}^1(1)) - (\tilde{\omega}_{dr}^1(1) - \tilde{\omega}_{db}^1(1)) \\ (\hat{\omega}_{dr}^2(1) - \hat{\omega}_{db}^2(1)) - (\tilde{\omega}_{dr}^2(1) - \tilde{\omega}_{db}^2(1)) \\ \dots \\ (\hat{\omega}_{dr}^I(1) - \hat{\omega}_{db}^I(1)) - (\tilde{\omega}_{dr}^I(1) - \tilde{\omega}_{db}^I(1)) \\ \dots \\ (\hat{\omega}_{dr}^1(J) - \hat{\omega}_{db}^1(J)) - (\tilde{\omega}_{dr}^1(J) - \tilde{\omega}_{db}^1(J)) \\ (\hat{\omega}_{dr}^2(J) - \hat{\omega}_{db}^2(J)) - (\tilde{\omega}_{dr}^2(J) - \tilde{\omega}_{db}^2(J)) \\ \dots \\ (\hat{\omega}_{dr}^I(J) - \hat{\omega}_{db}^I(J)) - (\tilde{\omega}_{dr}^I(J) - \tilde{\omega}_{db}^I(J)) \end{bmatrix}$$

□ 变量中的下标r表示rover，b表示base

# 04

## 代码简介



# 文件介绍

□ 打包程序文件夹内共有以下文件：

include	2024/10/8 5:23	文件夹	→	□ 工具函数
PLL相关函数	2024/10/8 5:23	文件夹	→	□ 锁相环函数
SGP4	2024/10/8 5:23	文件夹	→	□ 解星历相关函数
tleFile	2024/10/8 5:23	文件夹	→	□ TLE星历文件和EOP参数文件
65satellite mapping table.mat	2024/5/27 9:43	MATLAB Data 1 KB	→	□ 卫星信号ID和TLE映射表
getMeasurementPLL.m	2024/10/8 6:50	MATLAB Code 8 KB	→	□ 观测量估计函数
GPS原理与接收机设计 谢钢 着.pdf	2022/3/3 0:13	Microsoft Edge ... 48,578 KB	→	□ 附带参考的文献
init.m	2024/10/8 6:53	MATLAB Code 5 KB	→	□ 入口函数
initSettingsV101.m	2024/10/8 6:54	MATLAB Code 3 KB	→	□ 设置函数
lsqDcPositionEstimat.m	2024/10/8 6:12	MATLAB Code 4 KB	→	□ FOA、FDOA定位函数
measAssociate.m	2024/10/8 6:37	MATLAB Code 5 KB	→	□ 匹配base和rover的函数
TOAPositionEstimat.m	2024/10/8 6:11	MATLAB Code 3 KB	→	□ TOA、TDOA定位函数



# 设置函数介绍

□ 打开init.m文件，第一个重要的模块是初始设置，即第22行：

```
21  %% Initialize constants, settings and TLE =====
22  settings = initSettingsV101();
23
24  % Read the TLE data for all satellites
25  ephemeris = TLE_Reader(settings);
26
27
28  % Remove the redundancy of TLE data for the same satellite ID
29  ephemeris = sortSatData(ephemeris,settings);
30
```

□ 打开initSettingsV101函数，主要设置参数为：

10  %% Processing settings =====	
11  % Number of seconds to be processed	
12  settings.msToProcess = (30-1)*60;    %[s]	□ 处理数据的时间（秒）
13  % settings.msToProcess = (5-1)*60;    %[s]	
14  % Number of ship samples at the beginning of the signal file	
15  settings.skipNumberOfMs = 71067;	□ 开始时跳过多少毫秒的数据
16  % Name of the TLE file to calculate SV position and velocity	
17  settings.tleFile = 'IRI0723.txt';	□ TLE文件
18  %% Base receiver related setting	
19  % Base file	
20  settings.IfFileBs = 'F:\依星长基线等数据\data_V2\USRP_RX_326F3B5_V2.bin';	□ 数据位置 <small>% 长基线的数据，6.77km基线</small>
21  % Base start time of receiving	
22  settings.LinuxStartTimeBs = 1690115889.250850799999999999849;    % [second]	□ 数据接收的unix时间戳
23  [settings.startTimeBs, settings.startDay] = LinuxTime2UTC(settings.LinuxStartTimeBs);	
24  % Base receiver geo postion in LLA	
25  settings.referRxPos = [40.036446666666666667, 116.34000666666666667, 53];    % [Degrees]	□ 数据接收位置
26	

# 设置函数介绍

□ 打开init.m文件，第一个重要的模块是初始设置，即第22行：

```
21 %% Initialize constants, settings and TLE =====
22 settings = initSettingsV101();
23
24 % Read the TLE data for all satellites
25 ephemeris = TLE_Reader(settings);
26
27
28 % Remove the redundancy of TLE data for the same satellite ID
29 ephemeris = sortSatData(ephemeris,settings);
30
```

□ 打开initSettingsV101函数，主要设置参数为：

```
37 %% Raw signal file name and other parameter =====
38 % Name of the EOP file
39 settings.eopFile = 'EOP2_latest.txt';
40 % Data type used to store one sample
41 settings.dataType = 'int16';
42 % File Types
43 %1 - 8 bit real samples S0,S1,S2,...
44 %2 - 8 bit I/Q samples I0,Q0,I1,Q1,I2,Q2,...
45 settings.fileType = 2;
46 % Intermediate, sampling and code frequencies
47 settings.IF = 0e6; % [Hz] 1.364e6
48 settings.samplingFreq = 2.5e6; % [Hz]
49 % Iridium satellite visibility predict period
50 settings.visibT = 90*48/1000; % [s] 4.32s
51 % Elevation mask to exclude signals from satellites at low elevation
52 settings.elevationMask = 0; %[degrees 0 - 90]
53
```

□ EOP参数文件

□ 数据量化位数

□ 数据采集中频

□ 数据采样率

# 星历函数介绍

- 打开init.m文件，第二个重要的模块是读取星历，即第25-29行：

```
21  %% Initialize constants, settings and TLE =====
22  settings = initSettingsV101();
23
24  % Read the TLE data for all satellites
25  ephemeris = TLE_Reader(settings);
26
27
28  % Remove the redundancy of TLE data for the same satellite ID
29  ephemeris = sortSatData(ephemeris,settings);
30
```

- 读取TLE文件中的所有卫星信息

- 每一个卫星选取和接收时间最接近的时间点的信息

- TLE文件表达了轨道六根数，具有固定的格式

格式参考：[https://blog.csdn.net/klllo\\_/article/details/121519590](https://blog.csdn.net/klllo_/article/details/121519590)

```
IRIDIUM 106
1 41917U 17003A 23201.45118238 .00000035 00000+0 53139-5 0 9996
2 41917 86.4008 172.6427 0002384 97.5865 262.5602 14.34216059340887
IRIDIUM 103
```

- 星历使用方法参考measAssociate函数的第97行，第一个输入是目标卫星的时间点（转化为UTC的分钟，如3：20即180min），第二个输入时是对应的星历，可得卫星在ECEF坐标的位置（m）和速度（m/s）

```
97  [satPos, satVel] = getSatPosVel(combinMeas.measureTime(ind), ...
98  ephemeris(combinMeas.satIDRow(ind)),settings);
```

# 观测量提取函数介绍

- 打开init.m文件，第三个重要的模块是观测量提取，即第34-35行：

```
31 % Generate plot of the IF data =====
32 fprintf('Probing data (%s)...\n', settings.IfFileBs)
33 % Signal processing =====
34 baseMeas = getMeasurementPLL(settings, ephemeris, 'base');
35 roverMeas = getMeasurementPLL(settings, ephemeris, 'rover');
36 % load 'F:\0622论文数据信号处理\返修修改\长基线结果\V2_dataLongBaseline_PLL.mat'
37
```

- 根据阈值判断法确定信号大致的起始位置

```
114 for ind = 1:peakPerFrame
115     % 粗略位置检测
116     [mValue, mPos] = max(fftPeak);
117     if mValue < cn0Th
118         break
119     else
120         % fftResult(ind) = mPos;
121         mp1 = mPos - 8; if mp1 <= 0, mp1 = 1; end
122         mp2 = mPos + 8; if mp2 >= fftCnt, mp2 = fftCnt; end
123         fftPeak(mp1:mp2) = 0;
```

- 完成信号的捕获，同时为了提高观测精度，二次估计了信号频率，并初始化跟踪环

```
125 % ----- 粗略多普勒估计 -----

% ----- CW头起始位置估计 -----
129 subSig2 = data1(fftPointsCnt*(mPos-2)+1+len5ms : fftPointsCnt*mPos + len5ms + cwPlusUwordLen);
130 % ???
131 phasePoints1 = (0 : (length(subSig2) - 1)) * 2 * pi * ts;
132 localCarr = exp(-1i * coarseFreq * phasePoints1);
133 carrWiptOff = subSig2 .* localCarr;
134 corrResult1 = xcorr(carrWiptOff, syncWordSample);
135 corrResult2 = abs(corrResult1(end-length(carrWiptOff)+1:end));
136 [~, mPos1] = max(corrResult2);
137 cwPosEstimated = fftPointsCnt*(mPos-2)+1+len5ms + mPos1;
138 % ----- 估计精确的多普勒 -----
139 subSig3 = data1(cwPosEstimated: cwPosEstimated + cwLen1-1);
140 eastimatedFreq = freqEstimate(subSig3, settings.samplingFreq);
141
```

# 观测量提取函数介绍

- 跟踪模块：146行使用相位差分法进行频率估计和解码（简单明确），148行使用PLL进行处理（稍复杂）

```
146 % [measOutput,successFlag] = getDataAndFreq(sigWithPreamble,estimatedFreq,settings);
147 Bn = 50;use_corrTdex = 25;
148 [measOutput,successFlag] = use_tracking_loop_getdatainformation(sigWithPreamble,estimatedFreq,Bn,use_corrTdex,settings,index,cwPosEstimated,settings
```

- 打开use\_tracking\_loop\_getdatainformation函数

```
22
23 %解算symbol
24 [tmp_symbol_data,~,~,CN0] = TrackingLoop_from_data_get_symbol(origin_data_with_preamble,Frequent_estimate,settings,lock_time,Bn);
25
26 %symbol变成bit
27 [bit_output] = from_symbol_get_bit(tmp_symbol_data(65:end));
28
29 %检验数据类型
30 data_type = from_bit_get_datatype(bit_output);
31
32 %对相应的数据类型进行处理
33 if(data_type == -1)
34     time_after_pream = 0;
35     frequency_after_loop = 0;
36     beam = 0;
37     success_flag = 0;
38     sat_id = -1;
39 elseif(data_type == 1)
40     success_flag = 1;
41     [sat_id,beam,~] = find_SV_ID_BC(bit_output);
42     [time_after_pream,frequency_after_loop,~,fflag,~] = FLLLoop_get_frequency_with_lock(use_corrTdex,origin_data_with_preamble,Frequent_estimate,settings,Bn,lock
43     if(abs(frequency_after_loop - Frequent_estimate)>50 || fflag ==0)
44         %说明误差比较大了
45         success_flag = 0;
46     end
47 elseif(data_type == 2)
48     success_flag = 1;
49     [sat_id,~,~,beam,~] = find_SV_ID_RA(bit_output);
50     [time_after_pream,frequency_after_loop,~,fflag,~] = FLLLoop_get_frequency_with_lock(use_corrTdex,origin_data_with_preamble,Frequent_estimate,settings,Bn,lock
51     %相位差分法计算误差率：只计算能验证的部分
52     if(abs(frequency_after_loop - Frequent_estimate)>50 || fflag ==0)
53         %说明误差比较大了
54         success_flag = 0;
```

解出Symbol

BCH校验，判断信号类型

环路估计频率，解码

解码请参考解码.pdf文档

# 定位函数介绍

□ 打开init.m文件，第四个重要的模块是定位，以FOA定位为例：

```
38 %% FOA positioning for rover receiver =====
39 combinMeas = measAssociate(baseMeas,roverMeas,ephemeris,settings);
40 combinMeas.freqRefined=combinMeas.roverDoppler;
41 % True rover receiver position in ECEF
42 roverPos.Geo = settings.roverRxPos;
43 roverPos.Ecef = geo2cartd(roverPos.Geo(1), roverPos.Geo(2), roverPos.Geo(3), 5);
44 % Biased position
45 biasedPosEcef = roverPos.Ecef + [60000, 60000, 60000];
46 % LSQ doppler positioning
47 estimatedPos1 = lsqDcPositionEstimat(biasedPosEcef, combinMeas, ephemeris, settings,'rover');
48 % Positioning error output
49 disp('【 ----- Rover FOA定位结果 ----- 】')
50 posErrorCalc(roverPos,estimatedPos1)
```

□ 匹配Base和Rover

□ 由于使用freqRefined作为观测量，选择Rover的观测量进行覆盖，否则是差分后的观测量

□ 输入偏移的位置、匹配后的观测、星历、设置进行定位

□ 定位方法参考定位小节



# 谢谢

THANK YOU

