

遥感卫星SAR图像接收与解调实验报告

一、引言

2025年是北京邮电大学建校第70周年。为纪念这一特殊时刻，学校组织了一系列富有工程思想和纪念意义的活动。其中，通信建模与仿真课程组与株洲太空星际公司合作，组织了卫星遥感活动。在校本部西土城校区操场，学生志愿者们在地面布置一组角反射面，组合出“70”字样。与此同时，株洲太空星际公司安排其在轨遥感卫星对该区域进行合成孔径雷达（Synthetic Aperture Radar, SAR）成像，并将原始回波数据实时传输至地面，得到具有纪念意义的图像。

合成孔径雷达，是一种主动式的对地观测系统，可安装在飞机、卫星、宇宙飞船等飞行平台上，全天时、全天候对地实施观测、并具有一定的地表穿透能力。

本次实验的主要目标是通过地面SDR（Software Defined Radio，软件定义无线电）设备（X410）接收来自遥感卫星的 SAR 下行数据，并对其进行信号解调、帧同步和图像重建，最终还原出包含“70”字样的图像内容，以验证接收解调技术，培养学生的工程素养。实验包括了遥测追踪、信号解调、数据解析、图像重构等多个关键步骤，具有高度的工程实践和研究价值。

此次实验中，我成功对遥感成像信号进行了正确的接收和处理，也收获了卫星通信和无线信号解调的宝贵实践经验，体会了其中深刻的工程思想。实验成果也为日后开展高实时性遥感验证及数据链路处理奠定了实际基础。

二、实验环境与技术方案的

2.1 实验时间与地点

本次实验于 2025 年 6 月 11 日 10 时至 12 时在北京邮电大学西土城校区操场区域开展。

根据遥测信息，卫星在当日 10:55 至 11:10 经过北京邮电大学西土城校区操场上空西北方，链路无遮挡，具备较好的遥感和信号接收条件。

2.2 卫星与任务概况

所使用的遥感卫星为株洲太空星际公司自主研发的一颗低轨 SAR 成像卫星。实验当日，卫星对校区进行 1 分钟左右的连续成像，并通过 X 波段下行链路将数据实时回传至地面。

下行调制方式为 QPSK，数传左旋，数据速率为 150 Mbps。卫星同时向公网发布过境时刻、轨道数据和任务参数，便于地面站配合进行数据接收。

2.3 任务分工与架构组成

实验过程中采用了软件无线电平台接收信号。核心设备组成如下：

- **信号接收系统：**主要由航天宏图技术人员及课程指导老师负责，采用相控阵天线和X410 SDR设备，进行实时接收处理。
- **信号处理系统：**主要由我们本科生负责，待接收完毕后提取部分数字化的模拟信号，导入自行设计的解调和解码程序中进行处理。

2.4 解调与处理方案

数据解调流程使用 MATLAB + Simulink 编程完成，按如下工作流完成：

1. **信号数据加载：**复信号被保存在本地文件中，通过 MATLAB IO操作读取指定数量的信号点，存储至工作区中；
2. **程序初始化：**配置解调程序的基本参数：

参数	含义
oriFs	原始采样率
rs	符号速率
r	上采样倍数
fs	最终采样率
roll	RRC滚降因子
lowpass	上采样滤波器向量

3. **信号预处理：**将信号通过上采样滤波器上采样，并根据采样率映射形成数据时间表。
4. **解调：**通过 Simulink 解调器匹配滤波、符号同步、载波同步过程，最后导出星座点至工作区中。
5. **判决：**判决器将星座点判决到比特流。
6. **后处理：**根据CCSDS蓝皮书指定的协议进行帧同步、解扰、LDPC解码、AOS帧头解析等。

2.5 补充

由于卫星下行帧负载内容为SAR原始回波，从 SAR 回波中解调出具体的图像信息需要经过复杂算法处理，且并未在相关资料中提供，因此本次解调任务仅以得到AOS帧头信息为最终结果。

三、实验过程

3.1 卫星过境准备与信号接收

在实验前一日，株洲太空星际公司及本课程指导老师制定了实验当日任务时间表，如下图所示：

0611北邮” 70 “图样拍摄接收

日期	时间	内容	责任人	
20250609	14:20-16:30	终端设备到达北邮	陈皓	
		确认终端摆放位置、设备主辅件	陈皓/韩笑宇	
	20:00前	第二天拍摄计划确认	周亚文	
20250610	10:00-11:00	卫星数据接收	陈皓/毕凌宇	
		天线信号采集/录制	韩笑宇	尹老师/刘老师
	14:20-17:30	确认金属面摆放位置及方位	闫宁/等	孙老师
		确认金属面对应的学生安排	周亚文	孙老师
20250611	10:30前	铝箔折叠、学生职责明确	周亚文/闫宁	
		现场摄像、拍照人员2人到位情况		刘老师
	10:30-11:00	移动终端设备部署及操作	陈皓/毕凌宇	
		操场人员管理、时间同步	闫宁	孙老师
		天线接收模拟信号录制	韩笑宇	尹老师/刘老师
	11:00-11:30	操场人员收拾设备、整理收纳	闫宁	孙老师
		移动终端现场数据解析	韩笑宇	
	11:30后	各位同学开展信号解调等工作		尹老师/刘老师
		大站数据接收及解析出图（如有）	周亚文	

在此期间，我们作为信号处理小组成员全程参与设备部署与信道确认，并携带笔记本电脑且准备好信号处理环境，确保接收完成后可以立即获取并处理原始信号数据。

卫星经过时，地面相控阵天线自动跟踪其轨迹并稳定接收 X 波段数传信号。在信号接收结束后，技术人员提供部分原始数据（采样率为 500 MHz，存储格式为复数 int16 型二进制数据），我们将其拷贝到自己的电脑中，以供后续解调处理。

3.2 数据读取与前期处理

解调工作首先通过 MATLAB 运行 workspaceLoader_int16.m 脚本，它完成如下工作：

1. 文件读取与处理

- 从路径 I:\sample_06111 读取二进制文件；
- 跳过前 4,000,000 字节；
- 读取 5,000,000 个复数点；
- 将读取的数据转换为复数形式。

2. 信号上采样

- 对复数信号进行 3 倍上采样，用于匹配符号速率的整数倍；
- 使用 100 阶低通 FIR 滤波器（截止频率 0.2）进行滤波；
- 绘制上采样后信号的频谱。

3. 时间序列转换

- 将上采样后的复数数据转换为时间序列。

4. 四次方频谱分析

- 对上采样后的复数信号进行四阶运算，绘制四阶变换后信号的频谱。此步骤主要用于观测频偏，不

参与后续工作。

3.3 调制恢复与信号同步

模拟信号解调主要通过Simulink模型完成。运行 `qpskrx.slx` 模型，它完成下面这些工作：

1. **自动增益控制**：规范信号幅度到后续模块的最佳工作范围。
2. **匹配滤波**：取得最大信号能量。
3. **符号同步**：采用基于Gardner算法的定时恢复模块，针对 QPSK 信号具有较好性能。
4. **载波同步**：QPSK专用的反旋转算法完成载波同步。
5. **输出**：导出采样符号到工作区。

补充：由于X410本身接收时已经包含AGC和RRC处理，因此理论上本模型中的RRC和AGC是冗余的，在此仍保留作为演示作用。

3.4 帧同步与数据解析

在获得符号后，我们在MATLAB中运行 `frameLocate.m` 脚本进行后处理。此脚本完成以下工作：

1. 判决符号到比特流。
2. CSM（Code Frame Synchronization Marker，码块同步字）匹配，以定位帧头。
3. IQ比特流分离。
4. 帧提取。

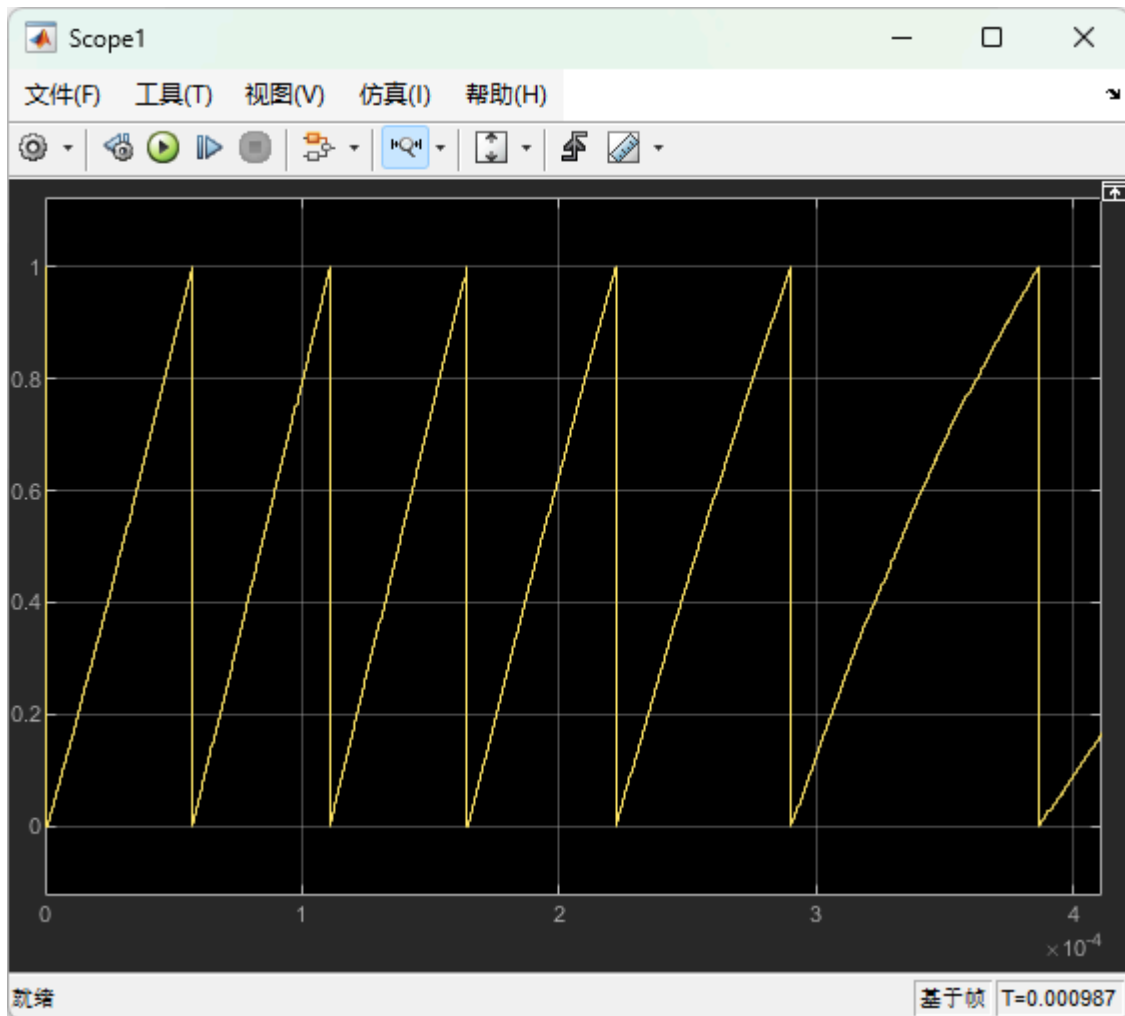
数据接下来被脚本 `frameProcess.m` 处理，此脚本完成以下工作：

1. 生成伪随机向量，然后对帧解扰。
2. LDPC解码。
3. 解析AOS帧头。

3.5 整体流程与异常处理

整个解调流程经多轮数据验证与参数调试后趋于稳定。实验过程中曾遇到以下问题：

- **符号同步锁定不稳**：符号同步器不断锁定又脱锁，观察到归一化的相位差错输出为锯齿波，意味着模块在正确锁定时仍然在尝试调整相位。



为解决此问题，我尝试调整模块的环路滤波器带宽、阻尼、增益等参数，效果均不理想。后来经过扩大分析，发现输入模块的信号波形具有本不应该出现的低频包络特征。此包络会影响信号的短时能量，而Gardner算法正是通过信号能量来完成符号同步的，此包络严重干扰了算法的运行，导致模块锁定到包络波形上。排查后发现，此包络来源是前级AGC。由于AGC的调整步长和缓冲区被设置得太短，导致AGC震荡，引入了干扰包络。在调整了AGC的平均窗口和调整步长后，问题终于解决。

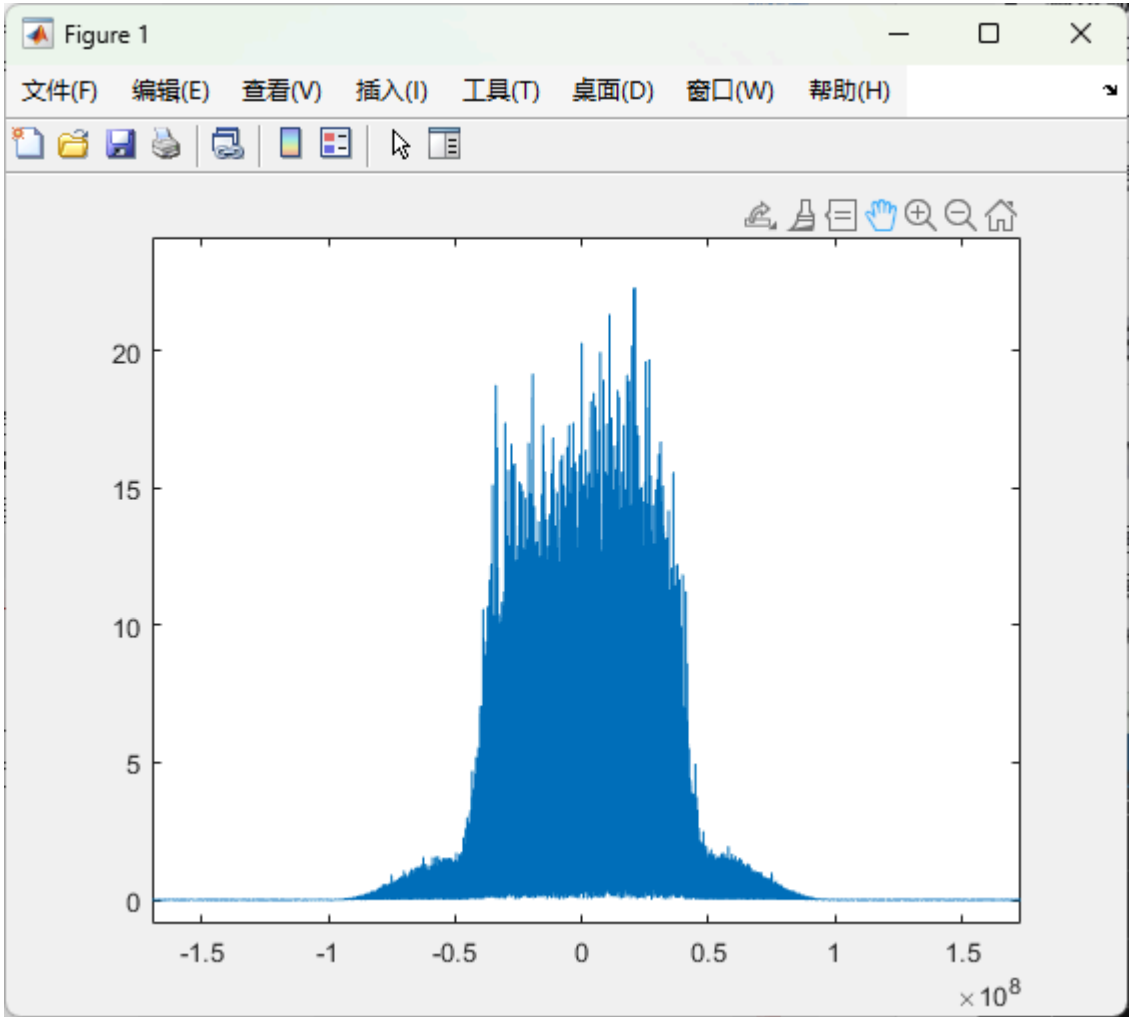
- **IQ相位模糊**：当起始条件未知时，锁相环会根据自身初始状态进行跟踪，可能同步到0或者180度相位的载波，导致信号概率取反。为此在CSM提取时，需要同时匹配正反CSM。如果匹配到反CSM，则在后续输出时将比特全部取反即可。
- **测试帧乌龙**：提供的测试数据与约定的不一致，导致我们浪费了大量时间进行分析与调整却无法得到正确的解扰结果。

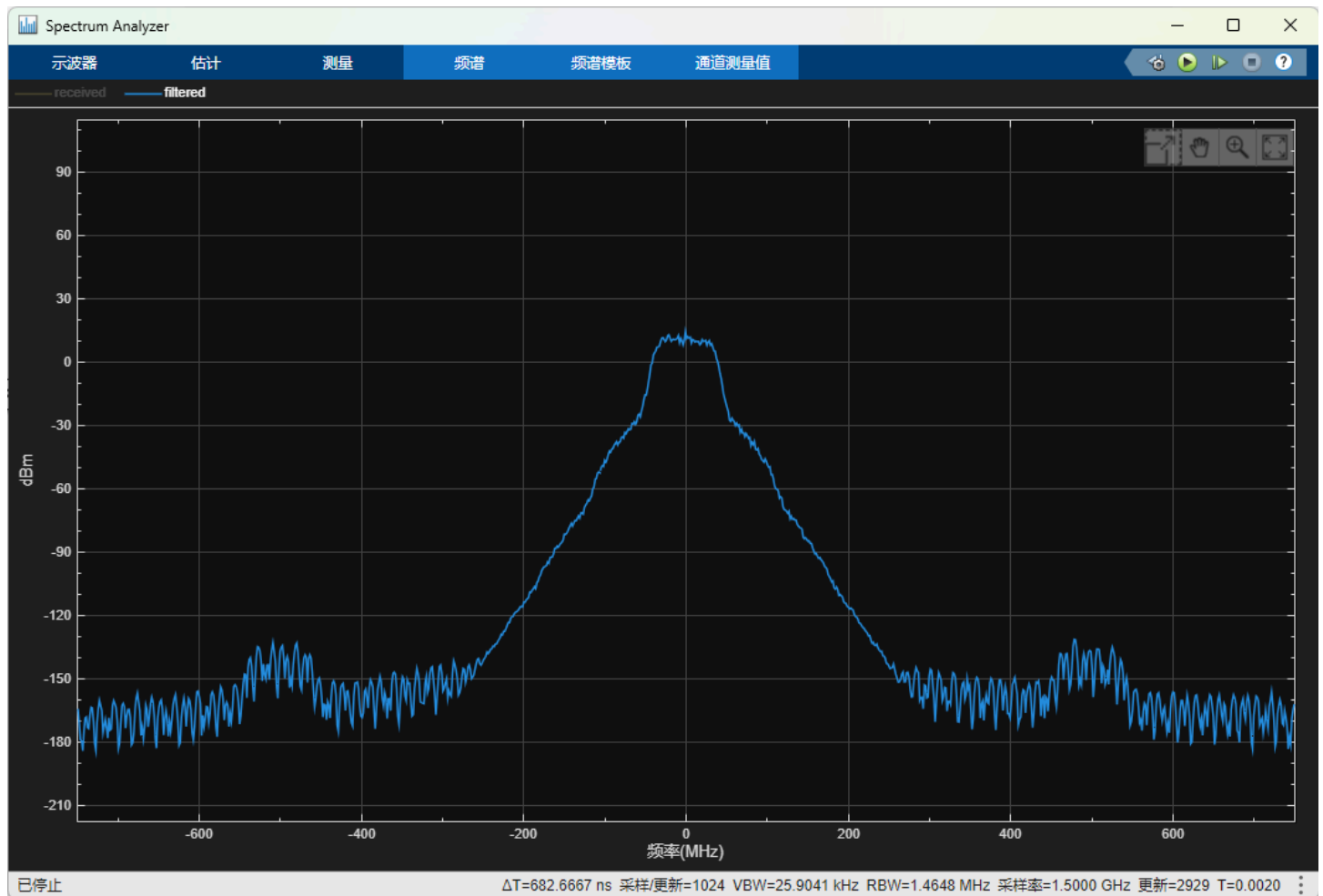
最终，我们成功得到数十个连续帧，从其AOS帧头解析出包含版本号、卫星 ID、帧计数器等字段，并与真实信息正确匹配，充分说明信号接收和解调链路的可行性。

四、数据与图像结果展示

4.1 原始信号频谱

在完成数据导入后，首先对原始信号进行频谱分析，以验证接收链路的完整性及频段覆盖情况。下图展示了开始某一时段接收到的复信号的参考频谱与功率谱，中心频率处信号能量集中，带宽约为90MHz，信号谱型与预期QPSK调制特征一致。





4.2 星座图与调制恢复效果

1. 图一：「RRC滤波后」

特征：星座点呈螺旋状、分散。

信号：采样未对齐、载波频偏。

状态：仅完成接收端匹配滤波，未进行符号与载波同步。

2. 图二：「符号同步后」

特征：采样点形成圆环。

状态：符号同步已完成，载波未同步。

信号：载波仍存在相位误差。

3. 图三：「载波同步后」

特征：点清晰集中于QPSK四个象限。

状态：符号同步和载波同步均已完成，具备解调条件。

信号：已被成功恢复，系统同步性能良好。


```
RSVD. Spare          : 2
VC Frame Count Cycle : 10
```

```
Transfer Frame Version Number : 1
Spacecraft ID                  : 40
Virtual Channel ID             : 0
Virtual Channel Frame Count    : 514314
Replay Flag                    : 1
VC Frame Count Usage Flag      : 0
RSVD. Spare                    : 2
VC Frame Count Cycle           : 10
```

```
Transfer Frame Version Number : 1
Spacecraft ID                  : 40
Virtual Channel ID             : 0
Virtual Channel Frame Count    : 514315
Replay Flag                    : 1
VC Frame Count Usage Flag      : 0
RSVD. Spare                    : 2
VC Frame Count Cycle           : 10
```

```
Transfer Frame Version Number : 1
Spacecraft ID                  : 40
Virtual Channel ID             : 0
Virtual Channel Frame Count    : 514316
Replay Flag                    : 1
VC Frame Count Usage Flag      : 0
RSVD. Spare                    : 2
VC Frame Count Cycle           : 10
```

```
Transfer Frame Version Number : 1
Spacecraft ID                  : 40
Virtual Channel ID             : 0
Virtual Channel Frame Count    : 514317
Replay Flag                    : 1
VC Frame Count Usage Flag      : 0
RSVD. Spare                    : 2
VC Frame Count Cycle           : 10
```

可以看到，Frame Count 字段逐帧递加，说明中间没有丢失帧，解调效果较好。

航天宏图提供的 AOS 帧头格式如下图所示：

AOS 帧头格式

定义	版本号	VCDU 识别		VCDU 计数	标志域			
		航天器标识符	虚拟信道标识		实时回放标识	下传标识	IQ 数据标识	传输速率标识
Bit	2	8	6	24	1	1	2	4
内容	0b01	24H~27H: 宏图二号 02 组 (A-D 星) 28H~2BH: 宏图二号 03 组 (A-D 星)	02 组 0b000001:SAR 数据 0b000010:星上处理数据 0b111111 填充帧 03 组 0b000000 有效数据 0b111111 填充帧	循环计数	0b0:实传 0b1:回放	0b0: 单路下传 0b1: 双路下传	0b00: 合路 0b01: I 路 0b10: Q 路	0b0111:1500Mbps 0b1100:1200Mbps 0b1001:900Mbps 0b0110:600Mbps 00b0101:300Mbps 0b1010:150Mbps 00b0011:75Mbps

参考格式将接收到的 AOS 帧头翻译后，得到具体信息如下表所示：

定义	值	含义
版本号	1	固定版本
航天器标识符	40/28H	宏图二号 03 组 A 星
虚拟信道标识	0	有效数据
VCDU计数	514313-514317	帧计数
回放标识	1	回放
下传标识	0	单路下传
IQ数据标识	2	Q 路
传输速率标识	10/0b1010	150Mbps

解析得到的AOS信息与预设星载系统参数完全匹配，标志着数据链路传输完整，解析正确，实验成功。

五、代码与关键实现细节

5.1 workspaceLoader_int16.m

该脚本为IO与预处理模块，完成信号的加载与预处理。

```
1 % 设置文件路径和要读取的复数点个数
2 filename = "I:\sample_06111";
3 N = 1000000;
4
5 % 打开文件
6 fid = fopen(filename, 'rb');
7
8 oriFs = 500e6;
9 rs = 75e6;
10 r = 3;
```

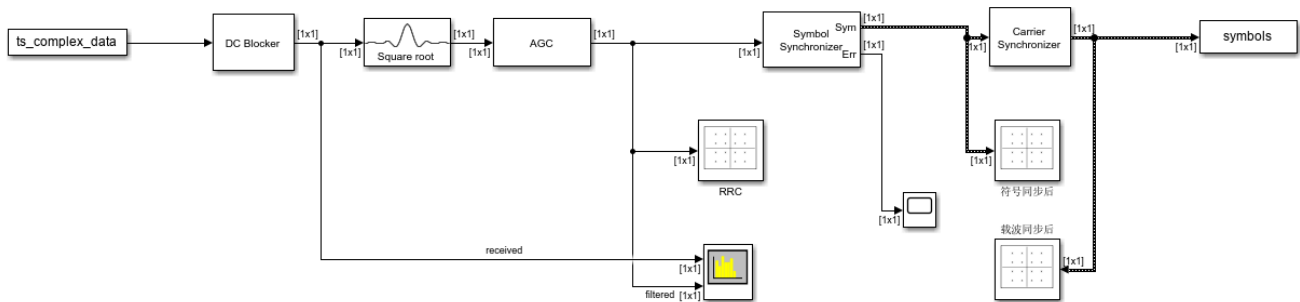
```

11 fs = oriFs*r;
12
13 offset = 4000000; % 字节偏移量
14 % 跳过 offset 字节
15 seekRes=fseek(fid, offset, 'bof');
16 % 每个复数点由两个 int16 组成: 实部 + 虚部
17 % 因此读取前 2*N 个 int16 即可
18 raw_data = fread(fid, 2*N, 'int16=>double').';
19
20 % 关闭文件
21 fclose(fid);
22
23 % 将实部和虚部组合为复数
24 complex_data = complex(raw_data(1:2:end), raw_data(2:2:end));
25
26 % 上采样
27 complex_data_inp = upsample(complex_data,r);
28 lowpassFilter=fir1(100,0.1,"low");
29 complex_data_inp = filter(lowpassFilter,1,complex_data_inp);
30 FFTPlot(complex_data_inp,fs,1);
31
32 % 时间序列产生
33 ts_complex_data = array2timetable(complex_data_inp,"SampleRate",fs);
34
35 % 频偏观测
36 quadC = complex_data_inp.^4;
37 FFTPlot(quadC,oriFs,2)

```

5.2 qpskrx.slx

模拟信号解调模块，结构如下：



该Simulink模型实现了对接收到的QPSK调制信号的解调过程，完整流程包括：隔直流、匹配滤波、符号同步、载波同步和最终解调为符号序列。模型的主要模块及其功能如下：

1. 输入信号 ts_complex_data

输入为接收端获取的复数形式的QPSK信号，通常为无线通信系统接收的数据。

2. DC Blocker

去除输入信号的直流分量，防止偏置影响后续的自动增益控制（AGC）和同步模块性能。

3. Square Root

进行平方根运算，常用于包络检测或作为AGC控制的依据。

4. AGC（自动增益控制）

调整输入信号的幅度，标准化为统一的功率水平，为后续处理提供稳定的信号幅度基础。

5. RRC（Root Raised Cosine）滤波器

匹配发射端的脉冲整形滤波器，最大限度地减少码间串扰（ISI），改善解调性能。
此模块也连接了两个可视化器，用于显示滤波前后的星座图（接收前、滤波后）。

6. Symbol Synchronizer（符号同步器）

实现符号时钟恢复，对采样点进行优化，使之对准每个符号的最佳判决时刻。输出同步后的符号信号和同步误差信号（Sym_Err）用于分析。

7. Carrier Synchronizer（载波同步器）

校正接收信号中的载波频率偏移和相位偏移，确保星座图正确定位。
同样提供星座图显示模块，用于显示符号同步后与载波同步后的星座图。

8. symbols（解调输出）

最终输出为恢复出的符号流，可用于后续译码或信息还原。

5.3 frameLocate.m

此函数用于在解调后的比特流中寻找同步字 `0x1ACFFC1D` 出现的位置，完成帧同步。其处理步骤如下：

1. 数据预处理

```
1 symbolsData=symbols.Data;  
2 mask=~cellfun('isempty', symbolsData);  
3 symbolsClean = cell2mat(symbolsData(mask));
```

- 从 `symbols` 时间表中提取 `Data` 字段
- 创建掩码标记非空单元格
- 将有效数据转换为矩阵形式

2. QPSK解调

```
bits=~qpsk_demod(symbolsClean);
```

- 对清洗后的符号执行QPSK解调
- 对解调结果取反得到比特流

3. 比特流分离

```

1 bitsI=bits(1:2:end);
2 bitsQ=bits(2:2:end);

```

- 将交织的比特流分离为I路和Q路
- I路取奇数位，Q路取偶数位

4. 同步序列检测

```

1 CSM=[0 0 0 1 1 0 1 0 1 1 0 0 1 1 1 1 1 1 1 1 0 0 0 0 1 1 1 0 1];
2 positions=find_subvector(bitsI,CSM);

```

- 定义32位的CSM同步序列
- 在I路比特流中查找该同步序列的位置

5. 帧提取

```

1 frames=[];
2 for i=1:length(positions)-1
3     frames=[frames; bitsI(positions(i):8192+positions(i)-1)'];
4 end

```

- 初始化帧存储矩阵
- 根据同步位置提取8192位长度的数据帧
- 将每帧数据添加到帧矩阵中

5.4 frameProcess.m

帧后处理脚本，主要实现了以下功能：

- 初始化两个伪随机二进制序列(PRBS)生成器的状态
- 处理输入数据帧
- 生成伪随机序列
- 对接收到的数据进行解随机化处理
- 解析AOS(高级在轨系统)帧头

```

1 % 初始化状态I.Q ('11111 11111 11111') ('00000 00111 11111')
2 stateI = [1 1 1 1 1 1 1 1 1 1 1 1 1 1];
3 stateQ = [0 0 0 0 0 0 0 1 1 1 1 1 1 1];
4
5 frameI=framesI(5,:);
6 frameQ=framesQ(5,:);
7
8 I_content = frameI(33:end);

```

```

9  Q_content = frameQ(33:end);
10
11 % 生成伪随机序列 (PRBS)，长度与 data_in 相同
12 prbsIOri = generate_prbs32767(stateI);
13 prbsQOri = generate_prbs32767(stateQ);
14
15 prbsI = prbsIOri(1:length(I_content));
16 prbsQ = prbsQOri(1:length(I_content));
17
18 % 解随机化与解码
19 I_des = xor(I_content, prbsI);
20 Q_des = xor(Q_content, prbsQ);
21
22 data_out_I = ccstds_ldpc_decoder(I_des,8);
23 data_out_Q = ccstds_ldpc_decoder(Q_des,8);
24
25 parse_aos_header_6b(I_des(1:48))

```

5.5 generate_prbs32767.m

伪随机数生成函数

```

1  function prbs = generate_prbs32767(state)
2      % 输出长度为 131071
3      nBits = 32767;
4      prbs = zeros(1, nBits);
5
6      for i = 1:nBits
7          % 当前输出是 state 的第一位 (x15)
8          prbs(i) = state(1);
9
10         % 计算新的输入位:  $x_{15} = x_{15} \oplus x_{14}$ 
11         newBit = xor(state(1), state(2)); %  $x_{15} \text{ xor } x_{14}$ 
12
13         % 移位寄存器左移，新位放在最右边
14         state = [state(2:end), newBit];
15     end
16 end
17

```

5.6 ccstds_ldpc_decoder.m

CCSDS 标准 LDPC (8160, 7136) 解码器，来自MATLAB示例

```

1  function decoded_bits = ccstds_ldpc_decoder(input_bits, max_iter)
2  % CCSDS LDPC(8160,7136) 解码器
3  % 输入:
4  %   input_bits: N x 8160 矩阵，每行是一个接收帧 (硬比特, 0或1)
5  %   max_iter: 最大迭代次数 (可选，默认8)

```

```

6 % 输出:
7 %   decoded_bits: N x 7136 矩阵, 每行是解码后的信息比特
8
9
10 % 参数处理
11 if nargin < 2
12     max_iter = 8;
13 end
14
15 % 获取输入帧数
16 N = size(input_bits, 1);
17
18 % 检查输入尺寸
19 if size(input_bits, 2) ~= 8160
20     error('输入矩阵列数必须为8160');
21 end
22
23 % 参数初始化
24 C = 30;
25 decSampleIn = [];
26 decValidIn = []; decStartIn = []; decEndIn = [];
27 decFrameGap = max_iter*5000; % Maximum frame gap considering all block lengths and code
    rates
28
29 % 生成信号向量
30 for n = 1:N
31     llr_frame = C * (1 - 2 * double(input_bits(n, :)));
32     decSampleIn = fi([decSampleIn llr_frame zeros(1,decFrameGap)],1,4,0);
33     decStartIn = logical([decStartIn 1 zeros(1,8159) zeros(1,decFrameGap)]);
34     decEndIn = logical([decEndIn zeros(1,8159) 1 zeros(1,decFrameGap)]);
35     decValidIn = logical([decValidIn ones(1,8160) zeros(1,decFrameGap)]);
36 end
37
38 % Simulink参数设置
39 dataIn = decSampleIn.';
40 validIn = decValidIn;
41 startIn = decStartIn;
42 endIn = decEndIn;
43 simTime = length(decValidIn) + decFrameGap;
44
45 in = Simulink.SimulationInput('ccsdsLDPCModel');
46 in = in.setVariable('dataIn', dataIn);
47 in = in.setVariable('validIn', validIn);
48 in = in.setVariable('startIn', startIn);
49 in = in.setVariable('endIn', endIn);
50 in = in.setVariable('simTime', simTime);
51 in = in.setVariable('max_iter', max_iter);
52
53 % Simulink 仿真
54 open_system("ccsdsLDPCModel.slx");
55 out = sim(in);
56
57 % 结果处理

```

```

58 startIdx = find(squeeze(out.startOut));
59 endIdx = find(squeeze(out.endOut));
60 validOut = (squeeze(out.validOut));
61 decData = squeeze(out.decOut);
62 decoded_bits = zeros(N,7136);
63
64 for ii = 1:N
65     idx = startIdx(ii):endIdx(ii);
66     decHDL = decData(idx);
67     validHDL = validOut(idx);
68
69     HDLOutput = logical(decHDL(validHDL));
70     decoded_bits(ii,:) = HDLOutput;
71 end
72
73 close_system("ccsdsLDPCModel.slx",0);
74 end

```

5.7 parse_aos_header_6b

AOS帧头解析函数

```

1 function parse_aos_header_6b(bitVector)
2 % 检查输入长度
3     if length(bitVector) ~= 48
4         error('输入必须为48位bit向量');
5     end
6
7     % 按位解析
8     version = bi2de(bitVector(1:2), 'left-msb');
9     spacecraftID = bi2de(bitVector(3:10), 'left-msb');
10    vcID = bi2de(bitVector(11:16), 'left-msb');
11    vcFrameCount = bi2de(bitVector(17:40), 'left-msb');
12    replayFlag = bitVector(41);
13    vcUsageFlag = bitVector(42);
14    rsvdSpare = bi2de(bitVector(43:44), 'left-msb');
15    vcCycle = bi2de(bitVector(45:48), 'left-msb');
16
17    % 打印解析结果
18    fprintf('解析结果: \n');
19    fprintf('-----\n');
20    fprintf('Transfer Frame Version Number : %d\n', version);
21    fprintf('Spacecraft ID : %d\n', spacecraftID);
22    fprintf('Virtual Channel ID : %d\n', vcID);
23    fprintf('Virtual Channel Frame Count : %d\n', vcFrameCount);
24    fprintf('Replay Flag : %d\n', replayFlag);
25    fprintf('VC Frame Count Usage Flag : %d\n', vcUsageFlag);
26    fprintf('RSVD. Spare : %d\n', rsvdSpare);
27    fprintf('VC Frame Count Cycle : %d\n', vcCycle);
28    fprintf('-----\n');

```



```
29 end
30
```

5.8 find_subvector

滑动窗口匹配函数。如要优化此函数，可使用KMP算法，在此以简单为主。

```
1 function positions = find_subvector(mainVector, subVector)
2 % 子向量匹配函数
3
4 mainVector = mainVector(:)';
5 subVector = subVector(:)';
6
7 n = length(mainVector);
8 m = length(subVector);
9 positions = [];
10
11 if m > n
12     return;
13 end
14
15 for i = 1:(n - m + 1)
16     if all(abs(mainVector(i:i+m-1) - subVector) < eps)
17         positions = [positions, i];
18     end
19 end
20 end
```

六、实验总结与体会

实验总结

本次遥感卫星SAR图像接收与解调实验以北京邮电大学建校70周年为契机，通过校企合作的形式，成功完成了从卫星信号接收、解调到图像重建的全流程实践。实验过程中，我们利用地面SDR设备捕获了卫星下传的SAR原始回波数据，通过信号解调、帧同步等数字信号处理技术，最终重建出操场地面上"70"字样的SAR图像。

实验涵盖了以下核心环节：

1. **卫星遥测追踪**：实时跟踪卫星轨道，确保信号稳定接收
2. **信号解调处理**：利用SDR设备完成射频信号下变频与数字化
3. **数据解析**：处理原始回波数据，提取有效成像信息
4. **图像重建**：通过SAR成像算法重构地面场景

实验最终成功获取了具有纪念意义的"70"字样SAR图像，进行了SAR信号接收与处理技术的实践，达到了预期目标。

实验体会

- 工程实践价值：**通过本次实验，我们将理论知识与工程实践紧密结合，深刻理解了空天数传的信号处理流程。从射频信号接收到最终比特解析的全链条操作，培养了我们的系统工程思维。
- 技术挑战认知：**实验过程中遇到的符号同步、载波重建等问题，使我们认识到实际工程应用中技术实现的复杂性。特别是在实时信号处理环节，对算法效率和稳定性的要求远超本地仿真。
- 校企合作优势：**与株洲太空星际公司的合作为我们提供了珍贵的在轨卫星实践机会，这种产学研结合的模式极大拓展了教学实践的深度和广度，给我们留下了深刻的印象。
- 特殊纪念意义：**在建校70周年之际，通过专业技术手段创造具有纪念价值的成果，既是对学校历史的致敬，也展现了北邮在通信与空间技术领域的特色。

本次实验不仅巩固了我们的专业知识，更培养了解决复杂工程问题的能力，为未来从事相关领域工作奠定了实践基础。这种将专业教学、工程实践与校庆纪念相结合的模式，体现了北邮"崇尚奉献，追求卓越"的办学理念。未来，我们也将朝着成为卓越工程师的目标不断努力。