

lbb-用于射电天文VLBI基带数据 读写的库*

郭绍光^{1†} 李纪云¹ 朱人杰^{1,2} 劳保强¹ 陆扬¹ 杨颖³

(1 中国科学院上海天文台 上海 200030)

(2 中国科学院射电天文重点实验室 南京 210023)

(3 中国科学院国家授时中心 西安 710600)

摘要 Library for Baseband (*lbb*)是一个自研的用于读取解析甚长基线干涉测量(Very Long Baseline Interferometry, VLBI)基带数据的工具库,主要用于VLBI观测中对终端基带数据的读取、解析及输出结果分析.该软件库通过对基带数据的自动判断,自动实现读取不同数据格式的功能,并提供了各种各样的API (Application Programming Interface)功能供用户调用.目前*lbb*软件库已经成功应用在了探月工程VLBI测轨任务中的硬件相关处理机配置项和测地VLBI观测数据的预处理.文章详细介绍了*lbb*软件库的设计、功能及用法.

关键词 射电天文学: 基带数据, 数据采集: 终端, 数据处理: 格式转换

中图分类号: P161; 文献标识码: A

1 引言

甚长基线干涉测量(Very Long Baseline Interferometry, VLBI)是目前角分辨率最高的天文观测技术,通过这种技术,世界各地许多射电望远镜的信号可以组合在一起,产生极高的角分辨率,使其在天文观测、大地测量和深空探测中得到了广泛的应用^[1-3].作为在VLBI接收系统中重要的数字设备之一的数字基带转换器(Digital Base-Band Converter, DBBC)承担着频段选择、数据采集等任务^[4],其把接收机接收的宽带中频信号数字化后分成多个通道并转换为基带信号,以供相关处理机进行后续处理.基带信号格式比如MarkIV、Mark5B和VDIF (VLBI Data Interface Format)¹,是目前国际上VLBI领域使用最普遍的格式.射电天文VLBI基带数据基本都遵循一定的规则,每个数据帧包含帧头和数据,或者同时包含原始二进制数据和ASCII描述文件.虽然规则相同,但细节

2019-11-01收到原稿, 2020-01-09收到修改稿

*国家自然科学基金项目(11873079、11703069、11911540462), 国家自然科学基金委员会-中国科学院天文联合基金项目(U1831204), 中国科学院“西部之光”西部青年学者基金项目(XAB2016A06)资助

[†]sgguo@shao.ac.cn

¹<https://www.aoc.nrao.edu/software/sched/labelSECRECSYSRecordingSystems.html>

不同. 因此, 在使用这些数据之前, 我们应该先对这些原始数据进行解码和分析, 然后将数据发送到硬件相关处理机^[5]或软件相关处理机^[6-7]进行相关生成可见度数据.

为了满足我国后续深空探测及测地观测对不同VLBI数据格式的支持, 本文描述了一个读取解析射电基带数据的框架库, 以简化数据处理使其标准化. 该框架库名为Library for Baseband data, 简称*lbb*软件库. *lbb*软件库对不同的基带数据将提供统一的读写接口, 方便用户调用, 并支持通过后缀名与解析帧头信息来自动判定数据格式并调用解析函数. 框架支持添加新的数据格式, 只需按照标准格式添加一些新的代码格式说明. *lbb*软件库使用C语言开发, 可以运行在多数类Unix系统中.

本文第2节将概述现有的射电天文基带数据格式, 第3节将对*lbb*软件库的设计和实现进行详细阐述, 第4节将使用真实数据进行实验测试分析, 最后对本文进行总结和展望.

2 基带数据概述

首先对基带数据记录系统进行介绍, 然后对基带数据进行阐述.

2.1 记录系统

数据终端采集和记录系统是VLBI的重要部分, 采集系统负责数据的采集和编码, 记录系统负责对终端的输出数据进行高速的存储. 为了应对未来对高输出速率、多种记录模式、海量数据的要求, 终端记录系统一直在更新迭代. 目前主流的为美国的Mark系列.

美国从1967年开始研发记录系统, 第1代为Mark1, 记录速率仅为1 Mbps, 随着技术的发展, 从2007年开始, 磁带系统被硬盘媒介代替, 记录速率也得到了质的提高. 因为Mark5B系统使用VLBI标准接口(VLBI Standard Interface, VSI)来记录数据, 改进了磁带系统遗留下来的格式化问题. 在每个模组中包含8个硬盘, 此时的记录速率达到了1 Gbps. 从2013年初开始, VLBA (Very Long Baseline Array)逐渐停止使用Mark5A系统, 并将大部分工程转移到Mark5C系统. 其他一些天文台也在逐步迁移到Mark5C系统^[8-9]. 同时随着国际和国内新开发的数字基带转换器, 比如美国的DBE (Digital Backend)、欧洲的EVN (European VLBI Network) DBBC^[10]以及我国的CDAS2 (Chinese Data Acquisition System 2), 均不再使用输出速率受限的VSI接口, 而是使用10GbE网络接口, 数据记录设备可以采用具有相关接口的设备或通用存储设备^[11-12]. 经过50 yr的不断更新和发展, 目前已经研发升级到了Mark6系统, 支持16 Gbps (4 Gbps/端口)的稳定记录速率和32 Gbps的爆发记录速率^[8].

除了美国的Mark系列系统, 日本使用自研的K系列系统^[13], 澳大利亚使用类PCEVN (Personal Computer European VLBI Network)系统. 中国也使用自研的VLBI网络(China VLBI Network, CVN)记录系统系列CVRs (China VLBI Recorder System), CVRS目前已经最大支持到4 Gbps/台站的记录速率, 在探月工程及其他项目中已经得到应用.

下面将主要针对最常用的Mark5B和VDIF的基带数据进行阐述.

2.2 Mark5B

Mark5B数据格式为基于硬盘的VLBI数据系统Mark5B的输出格式. 采样数据类型为实数, 且只有一种固定格式, 该数据格式的每一帧包含一个16字节的帧头和10000字

节的数据, 帧头信息中第1个字为同步字, 固定为0xEDDEADAB (大端模式, 小端反之), 第2个字的比特0到比特14为从0开始的秒内帧号, 第3个字以及第4个字为VLBI的BCD (Binary-Coded Decimal)码(精确到1 ms)和循环冗余校验码(Cyclical Redundancy Check Code, CRCC).

Mark5B数据支持 2^N 比特采样, 其中 $0 \leq N \leq 5$, 并且每一个秒内帧号为0的数据均为该时间的精确起点(精确到秒). 通道数为有效比特流除以每次采样的比特数(目前通常设置为1或者2), 当前版本假定所有有效比特流均有效. 另外Mark5B系统也会输出有效比特流掩码、秒内帧号及其他观测元数据, 当前*lbb*软件库版本暂时不读取这些信息, 需要指定这些参数.

2.3 VDIF

2.3.1 VDIF简介

随着新的VLBI数据采集与存储系统的发展, 与日俱增的数据交互需求以及实时与近实时的高速网络的需要, 都需要一个新的数据格式来解决这些问题, 此时出现了继VSI硬件接口协议(VSI Hardware, VSI-H)和VSI软件接口协议(VSI Software, VSI-S)之后的VSI-E接口协议(其中E是electronic-VLBI的含义), VSI-E基于标准的RTP/RTCP网络协议, 指定了数据格式和数据传输协议. 但由于VSI-E的复杂性, 并未被VLBI社区正式采纳. 此时就出现了VDIF, 该格式最早在2009年定义, 主要为了标准化VLBI数据的传输和存储并提升VLBI基带数据的可扩展性, VDIF的目标与VSI-E不太一样, VDIF不定义数据传输协议, 仅仅定义了一种不依赖于传输协议的数据交换格式, 这个格式匹配所有的VLBI数据传输, 包含实时与近实时的e-VLBI. VDIF的可扩展性主要体现在自定义扩展字, 可用于存储下列数据组合:

- Fully Corner-Turned Data (FCTD): 即一个thread包含多个通道;
- Not Corner-Turned Data (NCTD): 即多个threads, 每个thread包含一个通道;
- Partially Corner-Turned Data (PCTD): 即多个threads, 每个thread含多个通道.

上面的3种数据理论上可以通过软件进行重新组合来满足相关处理需要的, 可以很方便地将FCTD转换为NCTD和PCTD. 但是在将NCTD和PCTD的数据转换为FCTD数据时, 可能存在数据有效标记位无法表明VDIF数据是否有效的问题, 因为多个threads的数据标记位可能是不同的, 这就会引起部分数据权重不对造成的相关问题.

考虑到易识别性, VDIF文件的命名一般以vdif作为文件的后缀名, 格式为<exp code>_<station code>_<scan name>[_fd<datainfo1>_<datainfo2>].vdif. 其中exp code为experiment code即实验代码、station code为台站编码²、scan name为scan名、datainfo1及datainfo2为2.3.2节描述的格式信息, 比如2018年11月28日18点开始的rd1809_T6_332-1800_fd1024_16_2.vdif数据, 实验代码为rd1809, 参加台站为上海天马65 m望远镜, scan为332-1800, 记录速率为1024 Mbps, 16个通道, 2比特采样.

2.3.2 VDIF数据格式描述

与2.2节描述的Mark5B格式类似, VDIF数据文件也有一系列的数据帧组成. 每个数

²ftp://cddis.gsfc.nasa.gov/pub/vlbi/ivscontrol/ns-codes.txt

据帧由帧头(Data Frame Header)和数据(Data Array)组成, 统称为数据帧(VDIF Data Frame).

VDIF的数据来源如果只有一台设备, 比如DBBC或者CDAS2^[8], 这种情况下数据是严格意义上按照时间来排序的, 但是如果有多路输入数据流, 并且数据传输经过了交换机或者网络就不能保证数据到达的顺序了. VDIF数据不强制要求数据帧的次序, 但是对于某些比较旧的相关处理设备可能有这个需要, 目前仍旧在更新的软件基本可以处理并对部分数据进行重新排序处理^[7].

如2.3.1节所述, 数据包含单通道与多通道模式, 下面将分别阐述. 对于单通道而言, VDIF的采样比特数目可以从1到32, 实数按照比特采样递增, 而复数是按照成对比特采样出现的, 所以对于复数形式的数据而言, 最大支持的比特数其实为16, 如果大于该值, 将会占用2个相邻的字来描述. 虽然VDIF不强制要求采样比特数, 但对于标准的VLBI而言, 一般取1-4, 6-8和11-32.

对于多通道而言, 为了与VLBI数据格式兼容, 数据只支持 2^n 个通道, 2^k 个采样比特(其中 n 与 k 均为整数), 最大的支持通道为 2^{31} , 最大的采样比特为32. 对于多通道而言, 是按照一个完整的采样来描述的, 比如对于所有通道的一次实数采样, 包含 $2^n \times 2^k$ 个比特. 对于所有通道的一次复数采样, 包含 $2 \times 2^n \times 2^k$ 个比特.

考虑到VDIF的数据流最大可以支持1024个数据线程, 通常的使用方法为一个数据线程包含一个通道或者多个通道. 为了区分多线程的数据流, 定义了simple和compound数据流. 其中simple数据流包含2种类型: (1)含有相同采样率和采样比特数的一个或多个单通道线程; (2)含有相同通道数、采样率和采样比特数的一个或多个多通道线程. 对于simple类型的数据而言, 每个数据线程必须包含相同的通道数、采样比特数和数据类型(实数或者复数)、秒内帧号、数据帧头和数据负载长度. 对于描述VDIF数据一般格式为<data rate (Mbps)>-<total channels>-<bits/sample>-[<thread id>], 比如1024-16-2-16的含义为总的速率1024 Mbps, 分为16个通道, 2比特采样, 格式化为16个单通道线程. 对于compound数据流而言, 是由多个混杂的simple数据流组成的, 唯一需要注意的是, 不同的simple数据流必须保证有且仅有唯一线程号, 正常情况下不推荐使用这种模式.

2.3.3 VDIF文件帧头

与Mark5B格式的帧头不同, VDIF的帧头由8个32位的字共计32个字节(legacy VDIF除外, 依旧为16字节长)组成, 不同于Mark5B, VDIF数据的长度从32字节到134 M字节不等, 由帧头信息确定. 所有的信息均为小端设置. 在同一个观测中, 所有的VDIF文件帧头的前4个字遵循同样的定义规则, 但是后面的4个字由用户自定义. 这4个字的布局由文件扩展数据版本(Extended Data Version, EDV)来决定.

帧头的第1个字包含从参考纪元开始的秒数, 第2个字包含参考历元及从0开始的秒内帧号, 第3个字包含VDIF的版本号、数据通道以及数据帧长, 第4个字包含采样比特数、线程号、台站号等. 需要注意的是对于台站数目巨大的项目, 比如平方公里阵列望远镜(Square Kilometre Array, SKA)^[13], 该值将作为数字对待, 这种情况下, 8比特的值应该大于 48^3 . 后面的4个字为VDIF的扩展字(EDV), 这里对于用户自己生成的数据很有

³<https://www.skatelescope.org/>

帮助, 通过指定EDV的版本也可以使得解码软件自动识别数据格式, 目前提交的扩展版本参考图1, 如果不使用这4个字, 那么EDV将被设置为0, 其他的扩展也被设置为全0. 目前EDV已经有多个版本, 如图1所示, 详细的解析参考2.3.2节.

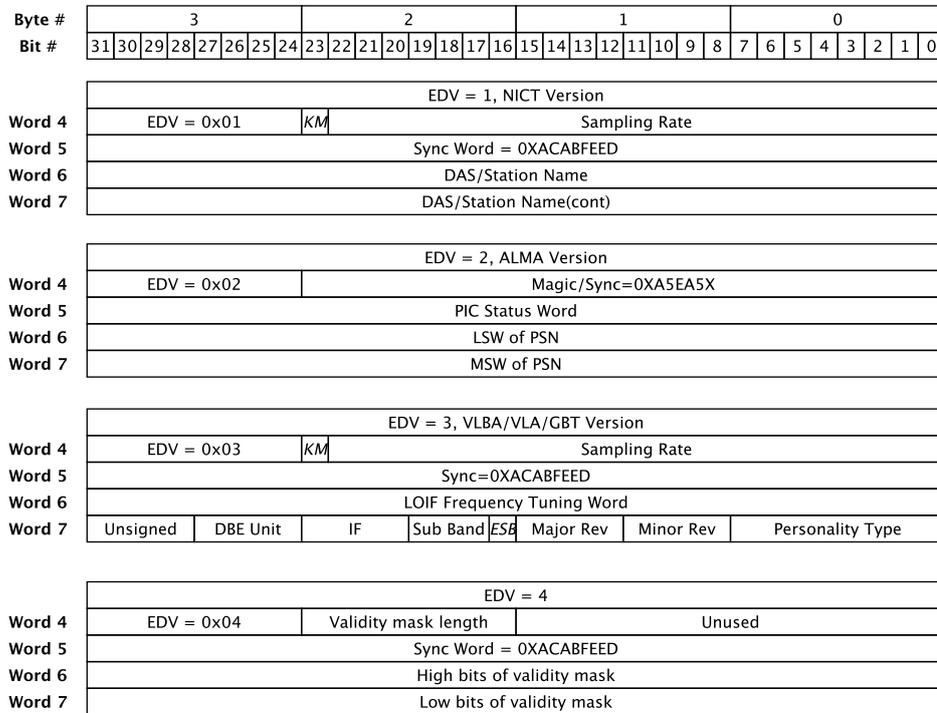


图 1 VDIF扩展数据版本示意图(图中缩写见3.2.2节定义)

Fig. 1 Scheme of VDIF extended data version (abbreviations are defined in Sec. 3.2.2)

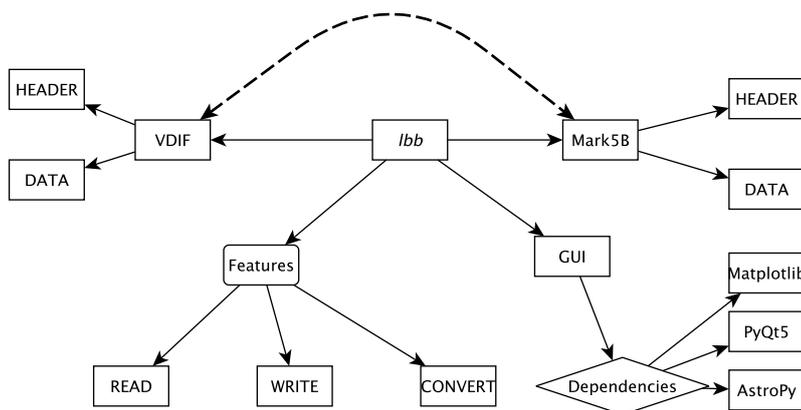
3 *lbb*软件库

3.1 设计理念

软件的开发受到其他一些软件的启发, 比如mark5access⁴, baseband⁵等. 考虑到可移植性, 软件全部使用C语言开发完成, 不依赖于任何程序库, 经测试可以在主流发行版上编译运行并使用. 考虑到易用性, 软件配套发布了基于Python的图形用户界面(Graphical User Interface, GUI)供用户方便调用. 软件数据解析核心代码使用C语言开发, 保证了软件的运行效率, GUI开发使用Python, 提高了开发效率. *lbb*软件库的整体架构如图2所示. 当前*lbb*软件库支持的基带数据格式为Mark5B和VDIF数据格式, 目前这两种数据格式最为通用. 探月工程中CVN的4个台站使用Mark5B数据格式, VLBI测地观测中使用VDIF数据格式. 最近已经开展了基于Mark6记录系统的观测实验^[8].

⁴<https://github.com/demorest/mark5access>

⁵<https://github.com/mhvk/baseband>

图 2 *lbb* 软件库架构示意图Fig. 2 General view of *lbb* architecture

由于终端基带数据的格式基本为帧头及数据组成的数据帧, 所以*lbb*软件库主要分为解析帧头和解析数据两种情况。

3.2 软件库介绍

*lbb*软件库主要提供了9个API (Application Programming Interface)供用户调用, 分别为:

- `read_frame`: 读取数据帧(自动判定数据格式, 包含帧头与数据);
- `read_frame_header`: 读取数据帧头(自动判定数据格式);
- `read_frame_data`: 读取数据(自动判定数据格式);
- `read_mark5b_frame`: 读取Mark5B数据帧(包含帧头与数据);
- `read_mark5b_frame_header`: 读取Mark5B帧头;
- `read_mark5b_frame_data`: 读取Mark5B数据;
- `read_vdif_frame`: 读取VDIF数据帧(包含帧头与数据);
- `read_vdif_frame_header`: 读取VDIF帧头;
- `read_vdif_frame_data`: 读取VDIF数据;

除了这9个主要的API以外, 还根据帧头信息增加了诸如`read_mark5b_frame_number` (读取Mark5B帧号)、`read_mark5b_mjd` (读取Mark5B简化儒略日)等函数。

3.2.1 Mark5B数据格式解析

Mark5B一个数据帧为10016字节, 超过了最大传输单元(Maximum Transmission Unit, MTU)的限制, 一般需要分为两组数据来传输, 每组数据包含5008字节, 这就导致数据可能不从每一个起始帧开始, 也可能不是一个整帧结束, 所以需要首先定位到第1个帧以及最后一帧的偏移量. 为了避免数据的跳帧情况, 此处的方法为确认连续的3帧均为正常帧, 即认为数据是连续的, 且第1帧为数据的起始帧。

图3和4分别展示了定位第1帧以及最后一帧的示意图, 其中红色为可能的偏移量(该值可能为0), 分别在第1帧和最后一帧出现, 绿色为帧头信息(Frame Header, FH), 黄色

表示10000字节的数据信息,中间为多组数据帧.

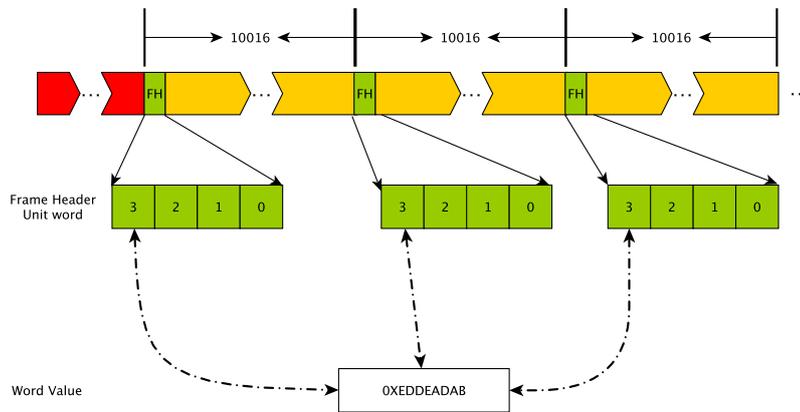


图 3 Mark5B数据的第1帧偏移值

Fig. 3 First frame offset of Mark5B data

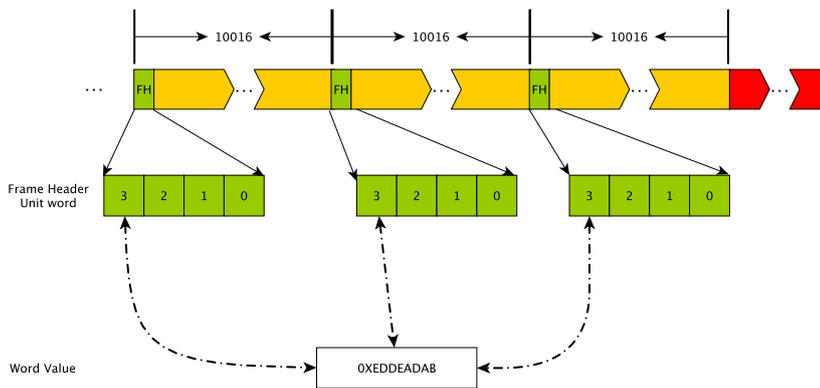


图 4 Mark5B数据的最后一帧偏移量

Fig. 4 Last frame offset of Mark5B data

在解析定位到数据帧头的同时,实际的数据也会存在系统缓存中,在每次重新定位到下一帧的同时将同一秒的数据写入到一个文件中直到到达数据帧尾.这个数据将用于后续的相关处理.

3.2.2 VDIF数据格式解析

如2.3.3节所述目前VDIF的解析支持5种扩展数据格式,对于VDIF格式的数据,一般认定第1帧即为起始帧或者作为一个偏移参数来进行处理,对于VDIF的扩展版本也可以通过同步字来进行识别和判断.

VDIF数据的解析流程如图5所示,其中FS为Frame Size的简称,为帧长度;FH为Frame Header的简称,为帧头,正常情况下如果不指定offset,默认前32字节为第1帧的帧头信息;MFS为Max Frame Size的简称,一般设定为20组最大FS的数据大小;详细的流程为,首先程序将对输入的参数进行解析,如果对数据的偏移有指定,需要跳过该部分数

据到达VDIF数据的帧头部分, 解析的参数也需要根据2.3.3节所述满足一定的条件, 比如数据的帧长必须是8的倍数, 且其表示的数据大小需要在32 kB到134 MB之间, 否则将作为异常处理, 另外程序还指定了详细的解析级别用于后续的数据完备性分析和检查.

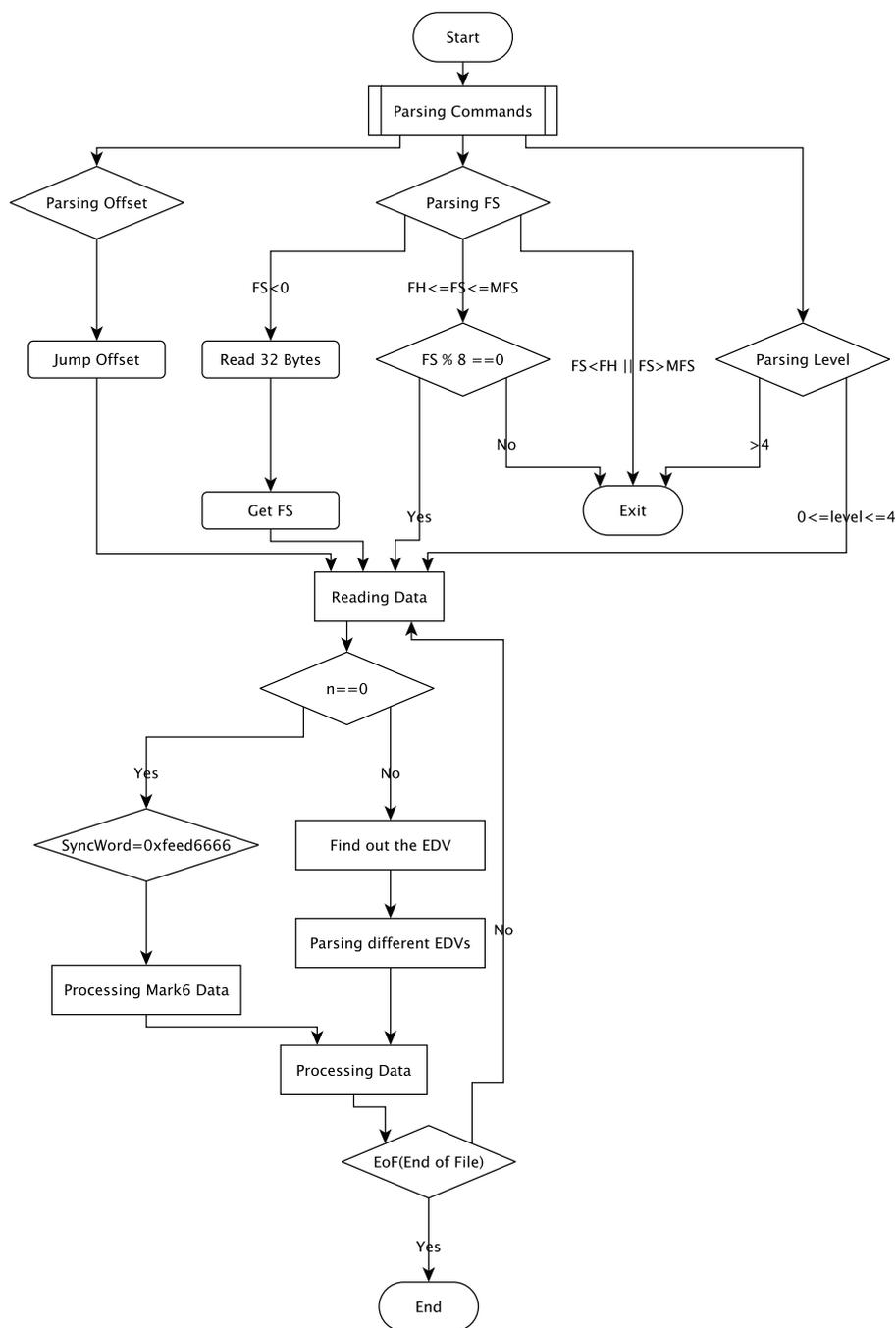


图 5 VDIF文件处理流程

Fig. 5 Processing flow of VDIF file

目前VDIF的EDV版本有5种,在解析出EDV版本后,不同的版本有不同的处理流程,其中对于使用Mark6设备将Mark5B在线转换为VDIF数据的流程单独处理,各个版本EDV的具体描述信息如下.当EDV的版本为1时,表示日本国家信息与通讯技术研究所(National Institute of Information and Communications Technology, NICT)定义的扩展版本,其中 KM (kHz MHz)标记采样率占1比特,为0时表示kHz,为1时表示MHz,这主要是因为K5/VSSP系统的采样率从40 kHz到64 MHz^[14]; Sampling Rate根据 KM 的单位,最大可以表示超过100 GHz的采样率; Word5为同步字0XACABFEED,用于查找VDIF文件的帧头位置; Word6和Word7共计8个字节用于描述数据采集系统或者台站的信息.

当EDV的版本为2时,表示阿塔卡玛毫米/亚毫米波阵列望远镜(The Atacama Large Millimeter/submillimeter Array, ALMA)定义的扩展版本,ALMA使用所谓的相位接口卡(Phasing Interface Cards, PIC)连接到基线观测相关处理中心输出数据.其中Magic/Sync为ALMA测试的子版本,可以通过这个参数来定义PICs观测、Mark6测试等其他应用; Word5用于指定PIC的状态或者定位Mark6的状态, Word6和Word7分别为VDIF传输协议(VDIF Transport Protocol, VTP)封包序列数(Packet Sequence Number, PSN)的最低有效位(Least significant word, LSW)和最高有效位(Most significant word, MSW).

当EDV的版本为3时,表示VLBA/VLA (The Very Large Array)与GBT (Green Bank Telescope)定义扩展版本,对于这个版本,有一些默认设置,比如每个VDIF数据帧长为5000,所以每帧的长度为5032字节,每个VDIF线程包含一个单独的基带通道;采样率与同步字采用NICT的版本,这个版本新增了数字后端优化频率(注意不是天空频率),对于应用在美国国立射电天文台(National Radio Astronomy Observatory, NRAO)的ROACH数字后端系统(ROACH Digital Backend System, RDBE)而言,这个值的范围为512 MHz到1024 MHz.第6个字表示本振频率(Local Oscillator Intermediate Frequency, LOIF), IF表示中频输入的数字后端设备, Sub Band表示DBE的子通道, ESB (Electronic SideBand)表示IF的上边带(为1)与下边带(为0).进阶的Major Rev (Revision)和Minor Rev为EDV3的主版本修订号和次版本修订号,后面的字节表示固件的类型与版本,其中对于Personality Type而言, 0X00表示多相滤波器组(polyphase filter banks, PFB), 0X80表示数字下变频器(Digital Down Converter, DDC) Mark5B, 0X81表示DDC complex, 0X82表示DDC VDIF.

当EDV的版本为4时, Validity mask length为1到64的整数,表示数据有效比特的掩码,另外还有16比特用于预留将来的功能扩展,其中同步字为32比特的0XACABFEED,与EDV1和EDV3一致,位于第5个字,用于确保VDIF数据流的同步,第6和7个字表示各个通道的数据是否有效,64比特分别对应64个通道,设置为1表示有效,反之无效.

其实除了上面描述的4个扩展版本,还有一个版本号为0XAB的扩展版本.在Mark6的处理中,会将输入的Mark5B格式的数据直接转换为VDIF格式^[8],因为Mark5B有16字节的帧头和10000字节的数据,网络传输分为2组5008的数据.在转换为VDIF格式时,16字节的帧头会复制一份作为32字节的VDIF格式的帧头,10000字节的数据紧随其后,此时数据帧为10032字节,因为Mark5B的同步字为0XABADDEED,所以版本号为0XAB.

在VDIF进行解析的时候还有一点需要特别注意,VDIF的数据是可以不连续的,比

如对于脉冲星观测而言, 只有在有pulsar pulse时才有数据, 所以对于此处需要特别注意, 此时就需要与当时的观测纲要文件进行比对以确定数据的详细情况.

3.3 图形用户界面

考虑到软件的易用性, 也开发了相应的图形用户界面GUI, 该GUI运行环境为Linux发行版本. 用户调用操作流程图如图6所示.

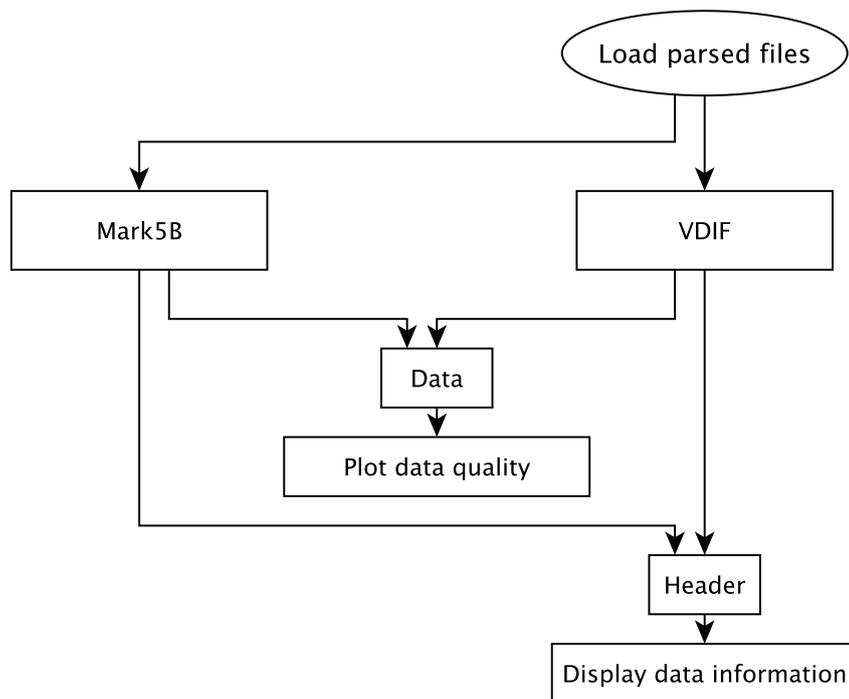


图 6 基带数据读取识别流程图

Fig. 6 Flow chart of reading and identifying baseband data

该GUI界面使用Python语言开发, Python是一个互动性及面向对象的脚本语言, 开发速度快. 该用户界面的开发使用PyQt⁶、matplotlib⁷等Python模块. 如图7所示, GUI软件目前具备的功能分别为: 输入文件区域、输出信息区域、控制区域及数据完整性可视化结果分析区域. 其中输入文件区域为输入经数据处理后的结果文件; 输出信息部分主要包括数据的时间信息和帧头信息; 控制显示区域主要为切换不同时间段及不同帧号并显示当前进度; 可视化区域主要为可视化输出当前数据的分析结果, 方便用户根据可视化区域快速定位数据问题.

⁶<https://www.riverbankcomputing.com/software/pyqt/intro>

⁷<https://matplotlib.org/>

4 应用案例

*lbb*软件库及相关测试软件已经在Linux系统的发行版CentOS7、Ubuntu18, 同时也在MacOSX上进行了测试. 鉴于*lbb*软件库的设计, 该软件库可以运行在大部分的主流Linux发行版上.

为了验证*lbb*软件库的通用性, 本章节将展示*lbb*软件库在探月工程VLBI测轨系统硬件相关处理机配置项的使用以及与mark5access、baseband相比所新增的功能和效率分析. baseband为加拿大多伦多大学Marten H. van Kerkwijk教授开发的一套基于Python的基带数据库; mark5access为美国国家射电天文台Walter Bristen教授及一批天文学家共同维护开发的基带数据库, 目前已经用于软件相关处理机DiFX (Distributed FX software correlator)^[7].

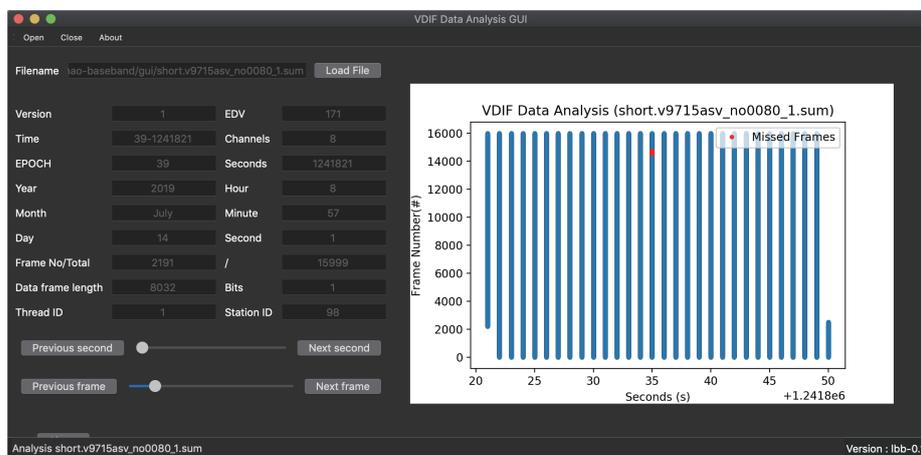


图7 VDIF数据分析用户界面

Fig. 7 Graphical user interface of VDIF data analysis

*lbb*软件库在相关处理机的应用主要为将CDAS2采集到的数据去帧头, 将时间信息发送给主控程序, 在几个测站的相关处理时间同步后, 开始进行相关处理操作; *lbb*软件库在VLBI全球观测系统(VLBI Global Observation System, VGOS)数据处理中的应用主要为将CDAS2终端输出的数据格式转换为满足VGOS国际联测任务的数据格式, 用来进行条纹检测和后续的相关处理.

测试使用相对复杂的VDIF数据格式进行解析对比, 使用的数据为VGOS的测地数据, 任务代号v9715a, 终端使用CDAS2, 数据格式为VDIF, 数据速率16 Gbps, 观测时间为2 h. 本案例将对几种不同大小的数据进行解析并输出秒级文件同时分析出数据完备性.

测试环境为Linux的CentOS发行版本, 版本号CentOS7.4, 基于i7 CPU、32 GB RAM、16核的运行环境.

使用基于*lbb*软件库自研的软件, 可以对数据进行解秒操作, 同时对数据的完整性进行解析并输出结果图像. 如图7所示对输出数据的秒内帧号进行解析, 并用红色标记出记录中缺失的数据帧. 同时基于*lbb*软件库还研发了支持一次性读取多个文件作为输入, 自

动对这一批数据进行处理和分析,并自动生成对应的文件结果信息.每个结果信息均可以使用*lbb*-GUI进行查看,极大地方便了用户使用.

在实例测试的同时,将*lbb*软件库与目前通用的mark5access以及baseband进行效率测试,对比结果如图8所示.

在和mark5access的比较中,因为均使用C语言开发,效率基本一致,而使用Python开发的baseband,可以看出在对小数据分析时,花费时间区别不大,但是随着数据量的增长,baseband对于数据的解析效率相较于*lbb*/mark5access就开始降低,随着数据量的增大这个差距会继续加大.

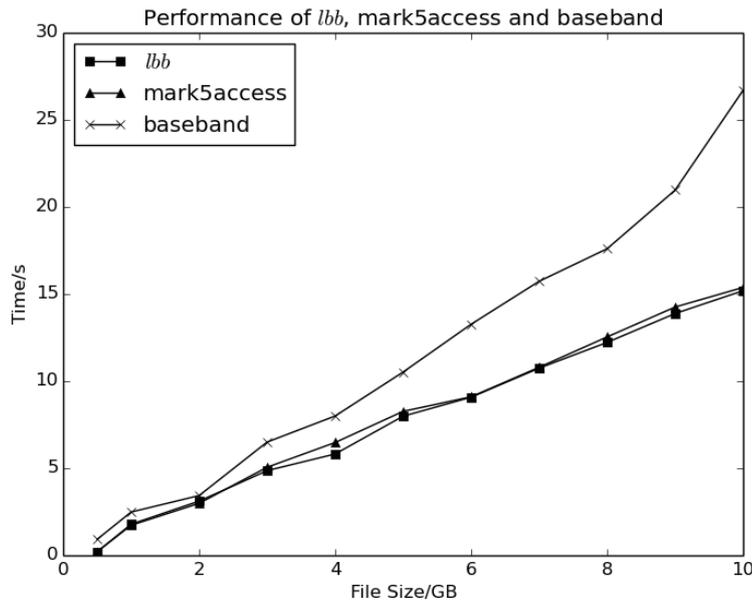


图 8 测试效率对比

Fig. 8 Comparison of the analysis efficiency

5 总结与展望

本文探讨了射电天文基带数据的历史及发展,并对通用的两种数据格式进行了相关的解析,为后续的数据处理提供了通用的软件库,方便直接调用.测试表明,该软件库可以在通用的Linux发行版上正常使用,可以对将来软件的开发与发展提供参考.

考虑到后续相关工程任务对实时数据处理的要求,VGOS常规数据观测8 Gbps或更高记录速率的观测以及诸如SKA等所产生的海量数据都对当前的软件和程序提出了更高的要求,直接输出数据流的方式可能会更有利于数据的处理.

参 考 文 献

- [1] 舒逢春, 郑为民, 蒋栋荣, 等. 天文学报, 2007, 48: 239
- [2] 黄勇, 胡小工, 黄城, 等. 天文学报, 2006, 47: 82
- [3] Huang Y, Hu X G, Huang C, et al. ChA&A, 2006, 30: 318
- [4] 罗近涛, 陈岚, 吴亚军, 等. 天文研究与技术, 2010, 7: 214
- [5] 朱人杰, 张秀忠, 项英, 等. 天文学进展, 2008, 26: 62
- [6] 郑为民, 杨艳. 世界科技研究与发展, 2005, 27: 7
- [7] Deller A T, Brisken W F, Phillips C J, et al. PASP, 2011, 123: 275
- [8] 郭绍光, 朱人杰, 郑为民, 等. 天文研究与技术, 2017, 14: 281
- [9] 杨文军. 天文研究与技术, 2012, 9: 26
- [10] Tuccari G. Proceedings of the 7th IVS General Meeting Proceedings. Madrid: Dirk Behrend and Karen Baver, 2012: 76
- [11] García-Miró C, Rogstad S P, Navarro R, et al. Proceedings of the 7th IVS General Meeting Proceedings. Madrid: Dirk Behrend and Karen Baver, 2012: 171
- [12] 朱人杰, 张秀忠, 韦文仁, 等. 天文学进展, 2011, 29: 207
- [13] 郭绍光, 郑小盈, 毛羽丰, 等. 科研信息化技术与应用, 2018, 9: 3
- [14] Kondo T, Amagai J, Koyama Y, et al. JGSJ, 2008, 54: 293

lbb-a Library for Reading and Writing VLBI Baseband Data of Radio Astronomy

GUO Shao-guang¹ LI Ji-yun¹ ZHU Ren-jie^{1,2} LAO Bao-qiang¹ LU Yang¹
YANG Ying³

(1 Shanghai Astronomical Observatory, Chinese Academy of Sciences, Shanghai 200030)

(2 Key Laboratory of Radio Astronomy, Chinese Academy of Sciences, Nanjing 210023)

(3 National Time Service Center, Chinese Academy of Sciences, Xi'an 710600)

ABSTRACT Library for Baseband (*lbb*) is a self-developed open source library for reading and analyzing VLBI (Very Long Baseline Interferometry) baseband data, which is mainly used for reading and analyzing terminal baseband data in VLBI observations. This software library can analyze different data formats based on the algorithm, and also provide a variety of Application Programming Interface (API) functions for the user. At present, *lbb* library had been successfully applied to the hardware correlator for Chang'e lunar project, and the data observation of geodetic VLBI. This paper introduces the design, function and usage of *lbb* in detail.

Key words radio astronomy: baseband data, data acquisition: terminal, data processing: format exchange