

상관결과 분석을 위한 파일 시스템 설계 및 소프트웨어 개발

File System Design and Software Development for Correlation Result Analysis

오세진*, 칸야 유키토시**, 노덕규*, 염재환*, 오충식*, 윤영주*, 정진승*, 정동규*

Se-Jin Oh*, Yukitoshi Kan-ya**, Duk-Gyoo Roh*, Jae-Hwan Yeom*, Chung-Sik Oh*, Young-Joo Yun*,
Jin-Seung Jung*, Dong-Kyu Jung*

요약

본 논문은 대전상관기로부터 출력되는 상관결과 파일을 데이터 분석에 활용할 수 있는 파일 시스템의 설계와 이와 관련된 소프트웨어 개발에 대해 기술한다. 상관결과는 천체의 가시도(visibility) 성분(진폭과 위상)이 나열된 구조이지만, 상관결과의 과학적 분석을 위해서는 천체를 관측할 때의 기상정보, 전파망원경 위치, 관측시간, 천체의 위치, 천체의 종류, 수신기 잡음온도 등의 여러 가지 정보가 필요하다. 데이터 분석을 위한 리눅스 시스템에서 이러한 정보들에 대해 디렉터리 구조의 파일 시스템을 설계하고 이 파일 시스템을 생성하는 소프트웨어를 개발하였다. 본 논문에서는 설계한 파일 시스템의 유효성을 확인하기 위해 상관처리 결과에 대해 파일 시스템 생성 시험을 수행하였으며, 천문학자들이 데이터 분석을 통하여 과학적 분석을 수행하는데 문제가 없음을 확인하였다.

ABSTRACT

In this paper, we designed file system in order to utilize data analysis by using correlation result from Daejeon correlator including related software development. Correlation results are consisted of visibility component (amplitude and phase) of radio source, but for data analysis of correlation result, various information such as weather, radio telescope position, observation time, radio source position, source type, and receiver noise temperature are needed. In this paper, we designed file system as a directory-structure for making use of these informations at Linux system for analyzing data and developed software to make file system. To verify the effectiveness of designed file system and developed software, file system generation experiment is conducted, and then astronomers accepted that there is no severe problem for scientific analysis using designed file system.

Keywords : Daejeon Correlator, CODA File System, CODA generator

I. 서론

대전상관기는 한국천문연구원과 일본국립천문대가 2006년부터 공동으로 개발을 시작하여 2009년에 완료하였으며, 2012년부터 전파간섭계인 한국우주전파관측망(Korean VLBI Network, KVN)과 VERA(VLBI Exploration of Radio Astrometry) 관측망으로 관측한 VLBI(Very Long Baseline Interferometry) 데이터의 상관처리를 수행하고

있다[1].

여러 종류의 시스템들이 융합되어 최종적인 자료를 만들어낼 때 중간 단계에서 발생하는 자료를 효율적으로 처리하기 위해서는 자료구조를 잘 정의하고 다음 단계에서 그 자료를 읽고 해석할 수 있는 하드웨어나 소프트웨어가 필요하다. 대전상관기는 여러 종류의 시스템들이 융합되어 하나의 시스템을 구성하고 있으며(그림 1), 상관결과는 32개 블록의 대용량 파일 시스템으로 구성되어 있다[2]. 본 연구에서는 이와 같이 대용량 파일 시스템을 다음의 자료 해석 단계에서 효율적으로 활용할 수 있도록 파일 시스템을 구성하는 소프트웨어 개발을 수행하였다. 전파천문의 VLBI 데이터에는 천체의 정보, 망원경 위치, 날씨, 관측시간 등의 정보가 필요하다[3]. 상관기의 출력은 전파망원경을 이용하여 천체의 전파세기(즉, Power spectrum)를 측정할 값에 대한 자기상관(Auto correlation)과 교차상관

* 한국천문연구원

** 일본국립천문대/Leonid Corporation

접수일자 : 2013. 5. 28 수정완료 : 2013. 7. 29

게재확정일자 : 2013. 8. 1

(Cross correlation) 결과를 출력한 값이다[4]. 따라서 자료의 해석을 위해서는 앞에서 기술한 다양한 관측 정보와 상관결과를 대상으로 파일 시스템을 구성하고 천문학의 자료 해석 프로그램인 AIPS(Astronomical Image Processing System)를 이용하여 데이터 해석을 수행하게 된다[5].

본 연구에서는 천체의 관측정보와 상관처리 결과를 분석하기 위한 CODA(Correlator Output Data Analysis) File System(CFS)을 설계하고 관련 소프트웨어 개발에 대해 기술하고자 한다.

본 논문의 구성은 다음과 같다. II장에서는 대전상관기의 상관결과 형식에 대해 기술하고, III장에서는 본 연구에서 제안한 상관결과 분석을 위한 파일 시스템의 설계에 대해 자세히 기술하며, IV장에서는 설계한 파일 시스템을 위한 소프트웨어 개발과 이를 이용한 파일 시스템 생성 및 시험결과에 대해 고찰한 후 마지막으로 V장에서 본 논문의 결론을 맺는다.

II. 한일상관기의 상관결과 파일구조

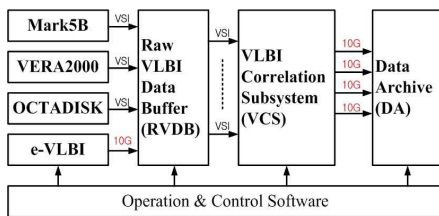


그림 1. 대전상관기의 구성

Fig. 1. Configuration of Daejeon correlator

대전상관기의 상관결과는 그림 1에 나타난 것과 같이 4개의 10GbE 출력포트와 기록장치(Archive system) 사이를 TCP/IP 통신으로 TCP 데이터 형식으로 기록장치에 저장된다. TCP 데이터는 IP 헤더(Header), TCP 헤더(Header), TCP Payload Data로서 출력되며, TCP 패킷의 구성을 표 1에 나타내었다.

표 1. TCP 패킷의 구성

Table 1. Configuration of TCP packet

TCP Packet 구성	Size
IP Header	20 byte
TCP Header	20 byte
TCP Payload Data	HEADER Packet의 경우 48 byte 상관 Packet의 경우 1036 byte

TCP Payload Data는 헤더 패킷(Header Packet)과 상관결과 데이터의 패킷으로 나누어져 있다. 출력순서는 헤더 패킷의 다음에 상관결과 데이터의 패킷이 출력된다. 하나의 상관결과는 표 2에 나타난 것과 같이 구성된다. 표 2는 대전상관기의 3가지 구성모드 중에서 16 관측국 모드를 나타낸 것이다.

표 2. 16관측국 모드의 TCP Payload 구성
Table 2. TCP Payload of 16 station-mode

HEADER 정보	HEADER PKT	HEADER Packet
자기상관 (Auto-Corr.)	상관 AUTO PKT#1	자기상관 Packet
	상관 AUTO PKT#2	
	⋮	
교차상관 (Cross-Corr.)	상관 AUTO PKT#16	교차상관 Packet
	상관 CROSS PKT#1	
	상관 CROSS PKT#2	
	상관 CROSS PKT#240	

표 2에 나타난 TCP Payload Data의 자세한 구성을 아래에 기술한다. 먼저 헤더 패킷은 표 3에 나타내었다.

표 3. 헤더 패킷의 상세 구성

Table 3. Detail configuration of HEADER packet

이름	Byte	내용	전체 크기
HEADMARK	4	"HEAD"(String)	48byte
Total Seqnum	4	0x00000000 ~ 0xFFFFFFFF	
Seqnum	4	항상 0	
Correlation Block#	1	(1~32)	
Array Mode	1	0:16 / 1:8+8 / 2:12+4	
FFT length	2	8~256(x1024-points)	
binning start channel	4	-0x3FFFF(256k-1)	
binning factor[16]	16	0~255(Setting value-1), (8-bits) x 16	
Stream No.	4	Serial number of streams. (0~63)	
IP length	4	256~02400(x0.1ms)	
IP count	4	0~0xFFFFFFFF	

표 4는 표 2에 나타난 자기상관 패킷의 상세한 구성을 나타낸 것이다.

표 4. 자기상관 패킷의 상세 구성

Table 4. Detail configuration of Auto correlation packet

Name	Byte	내용	전체크기
Seqnum	4	1~16	1036byte
X in	1	Station# in Correlation Block (0~15)	
Y in	1	Station# in Correlation Block (0~15)	
Stream No.	1	Serial No. of streams. (0~63)	
Data Kind	1	0: Auto / 2: Cross-Real / 3: Cross-Imag	
Valid segments	4	The number of valid segments	
Result Data[256]	1024	32-bits x 256-ch, offset binary	

대전상관기는 16관측국의 VLBI 데이터를 상관처리할 수 있기 때문에 자기상관은 16개, 교차상관은 120개가 출력된다. 교차상관의 경우 실수와 허수로 구분되어 총 240개가 출력된다. 따라서 표 5에 나타난 것과 같이 순서 번호는 표 4의 1-16, 그리고 17-256개가 출력된다. 표 5는 교차상

관 패킷의 구성을 자세히 나타낸 것이다.

표 5. 교차상관 패킷의 상세 구성

Table. 5. Detail configuration of cross correlation packet

Name	Byte	내용	전체크기
Seqnum	4	17~256	1036byte
X in	1	Station# in Correlation Block (0~15)	
Y in	1	Station# in Correlation Block (0~15)	
Stream No.	1	Serial No. of streams. (0~63)	
Data Kind	1	0: Auto / 2: Cross-Real / 3: Cross-Imag	
Valid segments	4	The number of valid segments	
Result Data[256]	1024	32-bits x 256-ch, offset binary	

표 3, 4, 5에서 "Stream No." 는 stream의 시리얼 번호이고, 하위 4-bits는 (Stream# -1)을 의미하고 상위 2-bits는 (Port#-1)을 나타낸다. 그래서 Stream No.는 (Port#-1)*16+(Stream#-1)이다. Y in은 상관 곱셈에서 공액(Conjugate)으로 변환된 입력을 나타내며, 자기상관의 경우 더미(Dummy) 데이터이다.

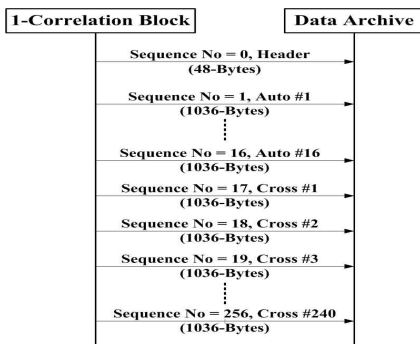


그림 2. 상관결과의 송신 순서(32개 상관출력 블록 중 1개 상관 블록을 나타냄)

Fig. 2. Transmission sequence of correlation result(This figure shows 1correlation block in 32 correlation block)

그림 2는 표 3, 4, 5의 내용을 바탕으로 상관기에서 데이터 아카이브로 전송되는 상관결과 출력 순서를 나타낸 것이다. TCP의 경우 상관기와 데이터 아카이브 사이의 연결 구성에 의해 데이터 아카이브에서 ACK 패킷의 요청이 있을 때 상관기는 데이터를 출력한다. 그리고 그림 2에서 괄호안의 byte 수는 TCP의 Payload Data의 크기를 나타내는데, TCP 헤더와 IP 헤더 등은 포함되어 있지 않다.

III. CODA 파일 시스템 설계

3.1 CODA 라이브러리

CODA File System(CFS)은 원래 일본국립천문대의 미

타카(Mitaka) 상관기에서 VLBI Space Observatory Programme(VSOP-1)의 데이터를 처리하기 위해 설계되었다. 이 CFS는 1994년에 포트란(Fortran77)으로 SURI GIKEN사에서 구축하였으며, CODA-1이라고 한다[6]. 그리고 2001년에 일본국립천문대의 VERA 관측망을 위해 후지츠(FUJITSU)사에서 2진(Binary) 데이터 형식을 지원하도록 설계변경을 수행하였다. 2001년 수정된 버전은 CODA-2.0이다. 본 연구의 대전상관기를 위한 CFS는 CODA-2.1로서 CODA-2.0이 미타카 상관기로부터 상관결과가 출력되면 그 결과를 직접 CFS로 변환시키면서 이진 파일 형식을 지원하는 것을 디렉터리 구조(Directory structure)를 지원하도록 수정하였다. 미타카 상관기는 최대 1Gbps의 데이터처리와 상관결과가 1024 주파수 채널을 가지는 반면 대전상관기는 최대 8Gbps 데이터처리와 상관결과가 8192 주파수 채널을 가지고 있기 때문에 상관출력을 그대로 CFS를 작성하는 것은 현재 네트워크 속도와 계산기의 성능에 비추어 보면 어려움이 따르기 때문에 우선 상관결과를 서버에 저장한 후 디렉터리 구조의 파일 시스템을 작성하도록 하였다. 이 부분이 CODA-2.1의 특징이라고 할 수 있다. 그리고 CFS와 연결을 담당하는 인터페이스인 IO 라이브러리를 CCcoda라고 한다. 이 CCcoda는 일본국립천문대의 VERA 관측망의 자료처리 소프트웨어인 VEDA(VERA Data Analyser)를 위한 라이브러리이다[7]. 현재 일본국립천문대에서 사용되고 있는 CCcoda 버전은 2.0이며 VERA의 DIR1000, DIR2000 관측장치들의 상관출력에 대해 대응하고 있다. 대전상관기의 CODA-2.1도 기존의 CCcoda 라이브러리를 그대로 활용한다. CCcoda 라이브러리는 CFS 데이터에 대해 기존의 포트란으로 작성된 CFS 라이브러리를 추상화해서 보다 알기 쉬운 인터페이스를 제공하고 있으며, 주요 목적을 정리하면 다음과 같다.

첫 번째는 CFS 라이브러리를 사용할 때 설정해야 하는 기존의 중요하지 않은 정보(가상 기계 번호, Dummy 변수, 설정이 번거로운 변수 등)를 쉽게 설정할 수 있다. 그래서 I/O 부분을 완전하게 분리하고, 각각의 라이브러리로 제공하는 것으로 소프트웨어 개발자가 CODA 특유의 부분을 편리하게 이용할 수 있도록 한다.

두 번째는 I/O 인터페이스를 추상화해서 DIR-1000 계 데이터나 FITS에 대응했을 때도 데이터 계산 부분의 변경이 필요 없게 한다.

위 두 번째 목적에서 CCcoda는 객체지향언어를 이용한 다형성(Polymorphism)의 기능을 사용하는 것이 장점이다. CCcoda는 C++ 언어로 구현되었는데, C++ 언어는 이전의 C 언어의 노하우나 CFS 라이브러리 등과 호환성을 갖고 있으므로 신뢰할 수 있는 프로그래밍 언어이다.

3.2 FITS-IDI 형식

FITS-IDI의 형식은 참고문헌[8]에 자세히 나와 있으며, FITS-IDI의 자료 구조는 표 6과 같다. FITS(Flexible Image Transport System)는 천문학적인 데이터를 디지털로 변환하거나 저장하는데 널리 사용되고 있다[5][8]. 그리

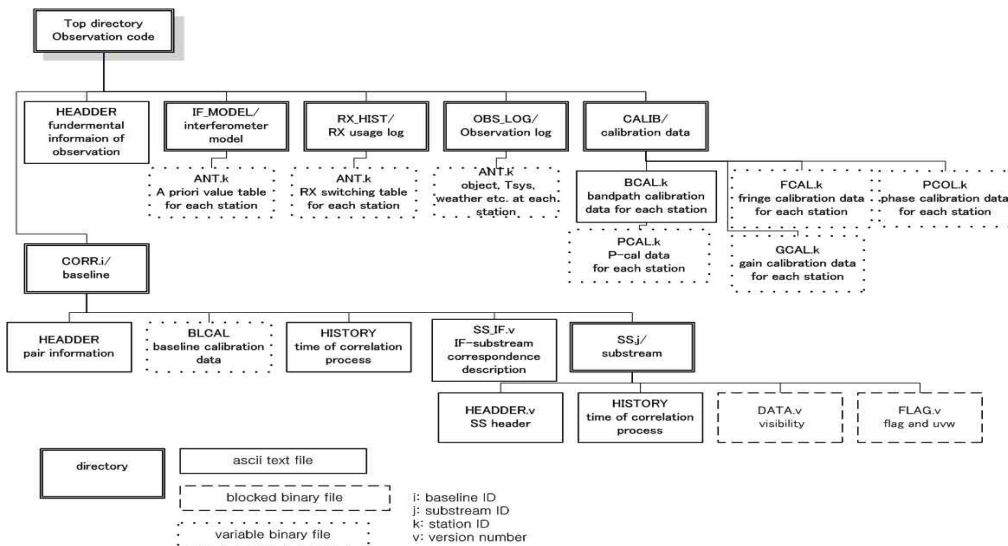


그림 3. 설계한 CODA 파일 시스템의 구조

Fig. 3. The designed CODA File System structure

고 FITS-IDI(Interferometer Data Interchange)는 전파천문학에서 전파간섭계로 관측한 데이터를 분석하기 위해 만든 자료형식이다. 표 6에 나타낸 것과 같이 FITS-IDI 자료구조는 Primary Header 이후에 각 간섭계 데이터에 대한 테이블의 헤더와 테이블 정보로 구성된다. 이 구조를 만들기 위해 상관결과와 관측로그 등으로부터 정보를 수집하여 CFS를 만들 때 활용하게 된다. 다음 절에서 이 FITS-IDI 구조를 만들기 위해 본 연구에서 제안하는 CFS의 구조에 대해 자세히 기술한다.

3.3 파일 시스템 설계

상관기 출력 데이터 및 상관운용 제어 계산기에 의한 출력은 인터페이스를 통해서 CFS 내에 저장된다. CFS는 파일 서버 내에 구축되며, 상관기와 계산서버는 네트워크를 통해서 연결된다.

CFS의 가장 중요한 점은 대용량의 상관결과 데이터를 보존하고 다음 단계에서 활용할 수 있도록 조작하는 것이다. 이를 위해 데이터는 취급할 수 있는 크기로 분할하고 데이터의 각 내용에 대해 계층화된 디렉터리 내에 둔다. 그리고 여러 개의 디렉터리와 파일로부터 구성되는 시스템을 용이하게 다룰 수 있도록 준비하는 것이다. 본 연구에서는 FITS-IDI 형식을 만족할 수 있도록 대용량 상관출력 파일로부터 디렉터리 구조를 갖는 CFS를 그림 3과 같이 설계하였다.

그림 3은 표 6에 나타낸 FITS-IDI 형식의 자료구조를 만들기 위해 본 연구에서 일본국립천문대의 미타카 상관기의 CFS를 바탕으로 수정하여 대전상관기의 상관결과에 적합하도록 설계한 디렉터리 구조를 갖는 CFS를 나타낸 것이다. 그림 3에서 다른 상자들은 파일이며, 이중선 상자(dual-solid line)는 디렉터리를 나타낸다. 그리고 단일선 상자(single-solid line)는 텍스트(ascii text, AT), 대시선(dashed-line) 상자는 블록된 2진(blocked binary, BB), 점선 상자(dotted-line)는 가변 2진(variable binary, VB)을 각각 나타낸다. AT 파일은 간단한 텍스트 파일이며, BB 파일과 VB 파일은 텍스트 헤더와 2진 데이터 영역으로 구성된다. 특히 VB 파일은 레코드 ID 번호로 규정된 다른 종류의 여러

가지 데이터 엔트리를 가진다. 각 파일들의 자세한 형식은 다음에 기술한다.

그림 3의 내용을 요약해서 각 VLBI 관측에 대해 CFS에 포함된 내용을 표 7에 나타내었다. 표 7에서 특정 파일 또는 디렉터리 이름에 첨자 i, j, k, v 등이 있는데, i는 기선 ID 번호, j는 서브스트림(Sub-stream) ID 번호, k는 관측국 ID 번호, v는 버전 번호로서 상태번호를 나타낸다. 각 파일의 버전 번호는 분석 프로그램에 의해 이 파일이 변경되었을 때 하나씩 증가한다.

표 6. FITS-IDI의 자료 구조

Table 6. Data structure of FITS-IDI

Primary Header
UV_DATA Table Header
UV_DATA Table
ARRAY_GEOMETRY Table Header
ARRAY_GEOMETRY
ANTENNA Table Header
ANTENNA Table
FREQUENCY Table Header
FREQUENCY Table
SOURCE Table Header
SOURCE Table
INTERFEROMETER Table Header
INTERFEROMETER Table
SYSTEM_TEMPERATURE Table Header
SYSTEM_TEMPERATURE Table
GAIN_CURVE Table Header
GAIN_CURVE Table
PHASE_CAL Table Header
PHASE_CAL Table
FLAG Table Header
FLAG Table
WEATHER Table Header
WEATHER Table
BASELINE Table Header
BASELINE Table
BANDPASS Table Header
BANDPASS Table
CALIBRATION Table Header
CALIBRATION Table
MODEL_COMPS Table Header
MODEL_COMPS Table

3.4 설계한 CODA F/S의 특징

본 연구에서 설계한 CODA-2.1의 CFS는 CODA-2.0을 사용하고 있는 일본국립천문대의 CFS와 비교하여 다음의 특징을 갖고 있다. 우선 개발환경이 유닉스(Unix)에서 리눅스(Linux)로 변경되었으며, C++ 언어를 사용하여 객체 지향적이고, 다형성, 계승 등의 기능과 표준 라이브러리를 최대한 활용하여 이식성을 높여 다른 시스템과의 호환성을 갖추었다. 두 번째는 상관처리 리포트를 참조하여 상관결과에 대한 정보를 미리 읽고 CFS로 어떻게 변환할 것인지 사전에 계획하는 기능을 추가하였다. 이는 상관처리된 결과 파일을 사전에 점검하고 CFS를 어떻게 생성하게 될 것인지 설정하는 기능을 갖는다. 세 번째는 VLBI 진파관측은 목적 천체가 하나인 경우와 여러 개인 경우가 있는데, 이때 수분에서 수 십분 정도 관측을 하며 관측천체를 변경하는 경우 관측과 관측 사이 즉, 진파망원경이 다음 관측을 위해 이동하는 사이에는 관측을 하지 않지만 데이터는 계속 기록되는 경우가 있다. 이 부분은 상관처리에서도 유효하지 않기 때문에 데이터 분석과정에서는 제외하고(Flagging out) 있다. 미타카 상관기의 CFS와 비교하여 간단하게 관측과 관측 사이를 제외하는 기능을 추가하였다. 네 번째는 상관처리를 한 번에 수행할 경우에는 문제없이 한 번에 CFS를 만들 수 있으나, 같은 관측에 대해 여러 번의 상관처리를 수행해야 하는 경우가 있으며, 이때마다 각각의 상관결과에 대해 CFS를 만들며, 이것을 하나의 CFS로 결합하는 기능도 함께 추가하여, 전체적인 성능 개선을 진행하였다.

표 7. 설계한 CODA F/S의 설명

Table 7. Explanation of the designed CODA F/S

file type	file name	type
the directory with the name of the observation code	the observation code	
observation header	HEADER	AT
the directory for the observation log of stations	OBS_LOG	
observation log	ANT.k	VB
the directories for baseline	CORR.i	
baseline header	HEADER	AT
baseline history	HISTORY	AT
baseline calibration data	BLCAL.v	VB
the directories for sub-streams	SSj	
substream header	HEADER.v	AT
substream history	HISTORY	AT
substream-IF correspondence info	SS_IF.v	AT
correlation data (visibility)	DATA.v	BB
flag and uvw data	FLAG.v	BB
the directory for calibration data	CALIB	
gain calibration data	GCAL.k.v	VB
fringe calibration data	FCAL.k.v	VB
band path calibration data	BCAL.k.v	VB
P-cal calibration data	PCAL.k.v	VB
phase calibration data (including 2B calibration data for VERA)	PCOL.k.v	VB
the directory for interferometer model	IF_MODEL	
interferometer model	ANT.k	VB
the directory for receiver history	RX_HIST	
receiver history	ANT.k	AT

IV. 소프트웨어 개발 및 시험

4.1 상관후처리 소프트웨어 개발

상관후처리 소프트웨어는 대전상관기의 상관결과를 CFS로 변환하고, 이를 과학적 목적에 활용할 수 있는 데이터 변환과정을 통해 AIPS 등으로 분석, 이미지 작업을 수행하는 일련의 소프트웨어를 말한다. 그림 4는 대전상관기를 위한 상관후처리 소프트웨어 구성을 나타낸 것이며, 간략한 기능을 설명하면 다음과 같다. CODA generator(CODAgen) 프로그램은 상관처리와 관측에 사용된 각종 파라미터를 이용하여 상관결과를 CFS로 변환한다. 이 파일 시스템을 이용하여 광역 프린지 탐색(Global Fringe Search) 프로그램은 프린지 탐색을 수행하여 관측 지연 파라미터의 시각 Offset을 보정한다. VEDA는 일본국립천문대에서 개발한 측지(Geodesy), 측성학(Astrometry)을 위해 개발한 소프트웨어로서 이 CFS를 사용하여 분석을 진행한다. 그리고 FITS generator(FITSgen) 프로그램은 CFS에 대해 천문학자가 과학적 분석을 위해 FITS-IDI 파일형식으로 변환하고 천문학자는 AIPS로 관측데이터를 분석하는 작업을 진행한다. 본 연구에서는 그림 4에서 점선으로 표기한 CFS 관련 소프트웨어의 개발에 대해 자세히 기술한다.

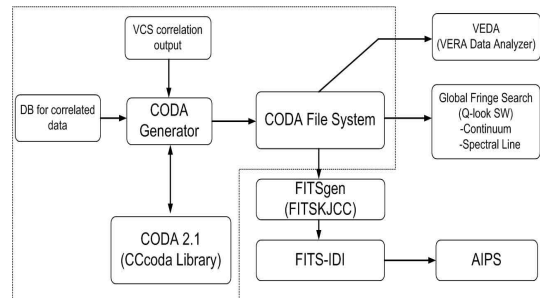


그림 4. 상관후처리 소프트웨어의 구성 요소들

Fig. 4. Constitution element of correlation post-processing software

그림 5는 상관결과를 CFS로 변환하는 프로그램과 제어 소프트웨어 사이의 구성을 나타낸 것이다. 제어 소프트웨어에서는 관측에서 상관처리까지의 일련의 로그(Log) 파일들, 상관처리 보고서, 파라미터 정보들을 포함한 리스트 파일을 CODAgen 프로그램으로 전달한다. CFS를 작성하는 CODAgen 프로그램은 KVN-CODA 서버에서 동작하는데, 이 프로그램은 상관결과가 저장되어 있는 데이터 아카이브 시스템을 접근해서 결과파일을 읽어 들인다. 그리고 데이터 아카이브 시스템과 프로그램이 설치된 KVN-CODA 서버는 네트워크 파일 시스템으로 연결되어 있으며, 상관결과도 동적으로 연결되어 있다.

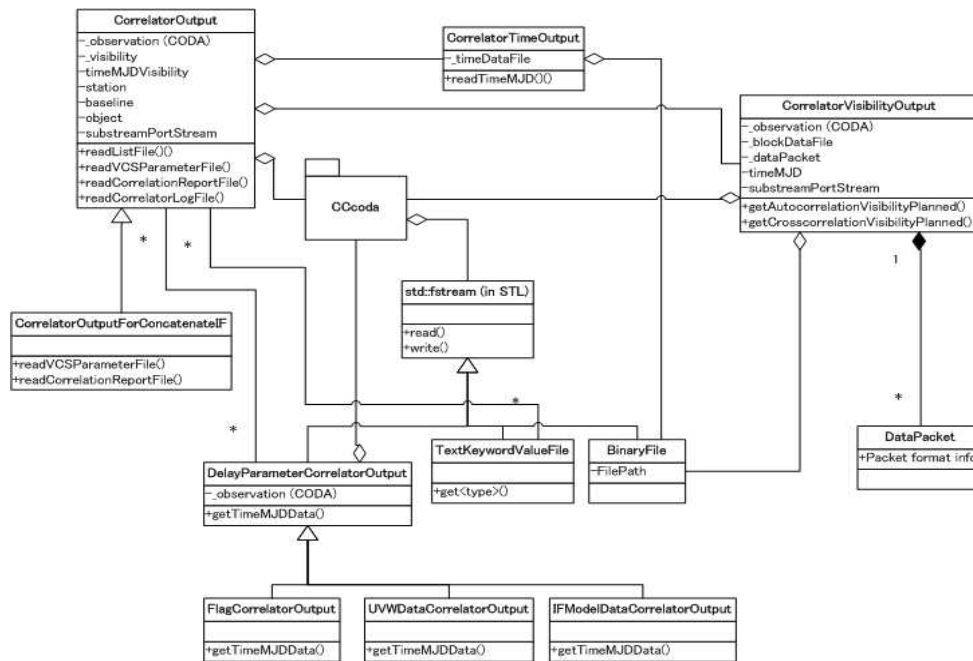


그림 6. CODAGenerate 프로그램의 Class UML(Unified Modeling Language)
Fig. 6. UML of CODAGenerate program Class

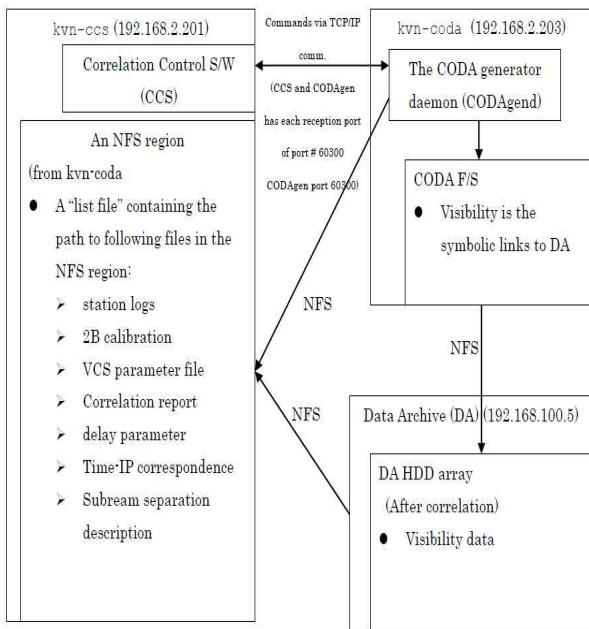


그림 5. 제어 소프트웨어, CODAgen, Data Archive 간의 통신 및 연결 구조
Fig. 5. Communication and connection structure between control software, CODAgen, Data Archive

표 8. 개발한 소프트웨어 리스트
Table 8. Developed software list

소프트웨어	사용 언어	기능
CODAgen.pl	Perl (5.8.0 이상)	CCS와의 TCP/IP에 의한 통신을 수행하고, CCS의 명령에 따라 다른 CODAgen 소프트웨어를 동작하고 CODA를 만든다.
CODAGenerateLocal.pl	Perl (5.8.0 이상)	로컬 머신 위에서 다른 CODAgen 소프트웨어를 동작하고 CODA를 만든다.
CODASkeleton	Perl (5.8.0 이상)	상관기 출력의 설정을 읽어와서 CODA 디렉터리 구조를 생성한다.
CODAGenerate.bin	C++11	CODASkeleton으로 생성한 디렉터리 구조 내의 파일에 상관기 출력 데이터를 기록한다.
CODAConcatenateIF	Perl (5.8.0 이상)	2개 이상의 CODA를 하나의 CODA로 결합한다.

CODASkeleton 프로그램은 경로에 지정된 리스트 파일로부터 상관출력의 설정을 읽고, 환경변수로 지정한 위치에 CODA의 디렉터리 구조를 만든다. 그리고 이 프로그램은 CODAGenerate가 파일을 쓰기 위해 필요한 헤더 정보만 기록된 비어있는 파일을 환경변수 CFS_SKELETON으로부터 CFS_DAT_HOME에 지정한 각 디렉터리에 적절한 이름으로 복사해 두는 역할을 하며, C++11로 작성하였다.

CODAGenerateLocal.pl 프로그램은 경로에 지정된 리스트(list) 파일로부터 상관 출력 결과들의 설정을 읽고 환경변수인 CFS_DAT_HOME(관측명마다 CFS의 루트(root)

4.1.1 CFS 작성 소프트웨어 개발

본 연구에서는 다음 표에 나타낸 CODAgen 소프트웨어들을 개발하였으며, 이하에 간략히 소개한다.

디렉토리를 정의한다)으로 지정한 위치에 CODA의 디렉터리 구조를 만들고 CFS 내에 데이터를 기록하는 기능을 담당한다. 환경변수에 지정한 디렉터리가 존재하지 않은 경우는 디렉토리를 만든다. 이 프로그램은 CODASkeleton과 CODAGenerate 프로그램을 실행하기 위해 스크립트로 작성한 것이다.

CODAGenerate.bin 프로그램은 경로에 지정한 리스트 파일로부터 상관출력의 설정을 읽으며 환경변수 CFS_DAT_HOME에 지정한 CODA의 디렉터리 구조 아래의 각 파일에 실제 상관결과를 직접 쓰는 역할을 하며, 앞에서 기술한 프로그램 중에서 가장 핵심적인 것이다. 이 소프트웨어는 C++11로 작성되었으며, 내부 클래스와 함수 등의 기능은 이후에 자세히 기술한다.

CODAConcatenateIF 프로그램은 미리 작성한 여러 CFS에 대해 명령어 라인에 지정한 리스트 파일들로부터 상관출력의 설정을 읽고, 여러 개로 지정한 CFS 디렉토리를 절대 경로로 지정한 2개 이상의 CFS를 환경변수 CFS_DAT_HOME에 지정한 1개의 CFS로 결합하는 역할을 한다.

이하에서는 본 연구에서 개발한 CODAGenerate.bin 프로그램의 클래스(Class)에 대해 자세히 기술한다. CODAgen 프로그램에서 가장 중요한 역할을 담당하는 것은 데이터 파일에 상관결과를 읽고 쓰는 CODAGenerate.bin 프로그램이다. 이하에 이 프로그램의 동작과 구성에 대해 자세히 기술한다.

CODAGenerate.bin 프로그램은 그림 6에 나타난 것과 같이 C++11 및 C++의 STL(Standard Template Library)를 이용해서 작성되었으며, 컴파일은 gcc 4.4.3 이상의 버전에서 수행하였다. 또한 CODA로의 연결은 앞에서 기술한 IO 라이브러리인 CCoda와 연결되어 있다. CODAGenerate.bin 프로그램은 메인(Main) 함수의 코드와 전처리(Preprocess) 정의를 기술한 코드, 그리고 표 9에 나타난 이 프로그램의 내부 클래스로 구성된다.

표 9의 클래스에 대한 UML(Unified Modeling Language)[9]을 그림 6에 나타내었다. 그림 6은 CCoda 라이브러리를 중심으로 CODAGenerate.bin 프로그램을 구성하는 각 클래스와의 관계이다. 그리고 전체 CODAGenerate.bin 프로그램의 동작 순서를 그림 7, 가시도를 CFS로 쓰는 것과 관련된 클래스 동작의 UML 순서를 그림 8에 각각 나타내었다.

그림 7에서 CODAGenerate.bin 프로그램의 실행순서는 다음과 같다. 우선 상관처리 로그(Log), 상관처리 파라미터 파일, 그리고 이들 파일을 담고 있는 리스트 파일을 실행 프로그램이 읽는다. 그리고 상관결과인 가시도 파일을 리스트 파일로부터 읽으며, 가시도 파일과 관련된 상관 uvw 파일을 읽는다.

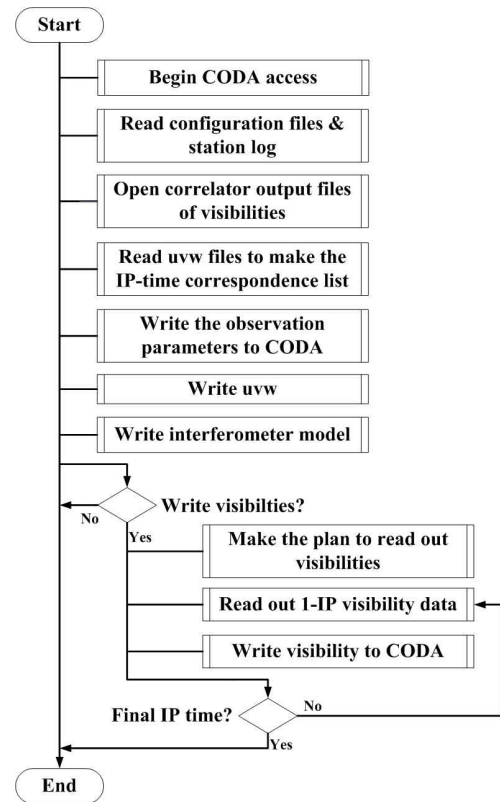


그림 7. CODAGenerate.bin 프로그램의 동작 순서
Fig. 7. Operating order of CODAGenerate.bin program

먼저 CFS에 기본적인 관측 파라미터들을 기록한다. 이와 함께 uvw, 상관모델 등을 CODA에서 지정한 파일 시스템에 기록한다. 이 단계가 끝난 후 실제 메모리상에 읽어 둔 가시도 파일에 대해 상관처리 로그에서 지정한 관측 모드 등을 참고하여 어떻게 상관결과인 가시도 파일을 읽을 것인지 계획단계를 수행한다. 이 계획단계를 수행할 때 상관결과의 가시도 파일이 규격대로 정상적인지를 확인하며, 계획단계가 완료되면 CFS에서 지정한 디렉터리에 가시도 파일을 기록한 후 CFS를 만드는 단계를 종료하게 된다.

그림 8의 상관출력인 가시도를 쓰는 메인 프로세스의 클래스에 대한 UML 순서를 살펴보면, TextKeywordValueFile 클래스로부터 CODA에 대한 설정파일들을 읽고, BinaryFile 클래스로부터 상관결과 파일을 읽고 IIP에 대한 파일 크기와 데이터 크기에 대한 정보를 얻는다.

그리고 UVWDataCorrelatorOutput 클래스로부터 전체 시간 정보를 얻은 후 CFS에 여러 가지 파라미터 정보를 기록한다. 이후 관련 파라미터 정보를 바탕으로 상관결과를 읽기 위한 계획단계에 대한 메모리 설정을 수행한 후 계획한 1-Time 가시도 데이터를 상관출력 파일로부터 읽은 후 CFS에 가시도 파일을 저장한다.

R11027B 관측명을 대상으로 CFS 생성시험을 수행하였다. 그림 9는 프로그램의 실행화면을 나타낸 것이며, 그림 10

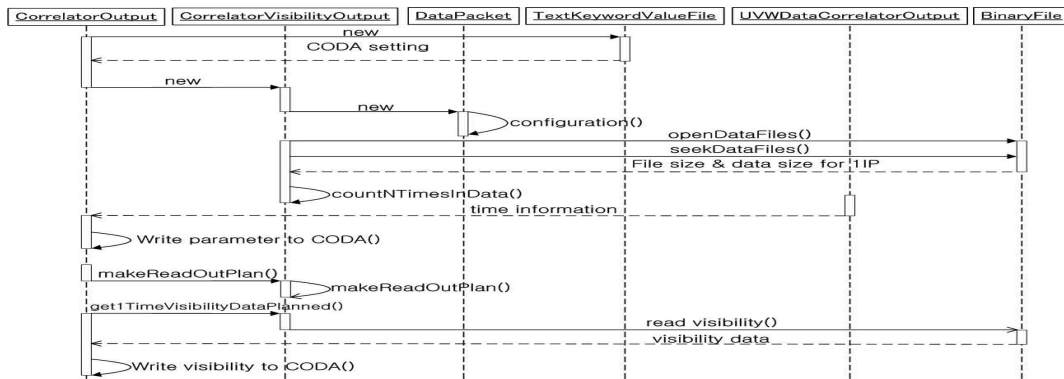


그림 8. 상관출력인 가시도를 쓰는 프로세스의 클래스에 대한 UML 순서

Fig. 8. UML sequence according to class of main process for writing the Visibility of correlation result

표. 9. CODAGenerate.bin 프로그램을 구성하는 클래스

Table 9. Class composed of CODAGenerate.bin program

클래스 명	기능
BinaryFile	STL의 fstream 클래스를 계승한다. 2진 파일의 IO를 담당한다.
CorrelatorLogFile	STL의 fstream 클래스를 계승한다. 관측국 로그를 읽는 역할을 담당한다.
CorrelatorOutput	각 상관출력 데이터에 포함된 정보를 VLBI 데이터로 통합하는 역할을 담당한다.
CorrelatorOutputForConcatenateIF	CorrelatorOutput을 계승한다. CODAConcatenateIF를 위해 CorrelatorOutput으로의 조금의 기능변경 부분을 담당한다.
CorrelatorTimeOutput	시각정보 파일을 읽는다. 현재는 각정보는 상관기의 uvw 출력 파일을 기본으로 하기 위해 사용하지 않는다.
CorrelatorVisibilityOutput	상관기 visibility 출력을 읽고 수정하는 역할을 담당한다.
DataPacket	상관기 visibility 출력의 내부단위가 되는 [데이터패킷]의 형식을 담당한다.
DelayParameterCorrelatorOutput	지연추적 파라미터와 flag 정보 등 IO로의 인터페이스를 정하는 추상 클래스
FlagCorrelatorOutput	DelayParameterCorrelatorOutput 클래스를 계승한다. Flag 정보를 읽는 역할을 담당한다.
IFModelDataCorrelatorOutput	DelayParameterCorrelatorOutput 클래스를 계승한다. 상관기 모델을 읽는 역할을 담당한다.
TextKeywordValueFile	STL의 fstream 클래스를 계승한다. 상관기의 list file과 VCS의 파라미터 파일 등의 키워드와 값의 설정을 기술하는 파일을 읽는 역할을 한다.
UVWDataCorrelatorOutput	DelayParameterCorrelatorOutput 클래스를 계승한다. uvw를 읽는 역할을 한다.

그림 7과 8에 나타낸 것과 같이 본 연구에서는 개발한 CODAGenerate.bin 프로그램에 상관결과로부터 CODA 구조를 만들기 위해 계획단계(Planning stage)를 도입하였다. 이 계획단계는 다양한 구조(상관 스트림(Stream)의 개수에 따른 상관결과 파일의 다양성)를 가지는 상관결과 파일을 CFS 파일 구조를 만들기 전에 프로그램이 상관처리에 사용된 로그 파일의 정보를 읽은 후 상관결과 파일구조를 미리 파악한 후 CFS 파일 구조를 만들 수 있도록 하는 것이다.

본 연구에서는 개발한 CODAgen 프로그램을 이용하여

은 생성된 CFS를 각각 나타낸 것이다. 그림 9의 '관측명.lst' 파일에는 그림 5에 나타낸 것과 같이 상관처리결과 리포트, 상관처리에 사용된 파라미터들, 관측로그, 지연 파라미터, 서브스트림의 구분 등에 대한 정보를 포함한다. 그림 10은 생성된 CFS 중에서 첫 번째 서브스트림에 대해서만 살펴본 것으로서 7개 관측국의 조합으로 가능한 교차상관에 대한 기선의 수는 $21(n(n-1)/2, n=7$ 일 때)기선이며, 자기상관을 포함하여 총 28개의 CORR.i를 생성하였으며, 각 CORR.i에 대해 상관모드 C5(16MHz BW, 16stream)인 경우 서브스트림 SS.j(여기서 j=16)를 만들고 그 서브 디렉터리에 가시도(DATA.1)와 관련 정보가 저장되어 있는 것을 볼 수 있다.



그림 9. CODA 생성 소프트웨어 실행화면

Fig. 9. Executive screen of CODA generation software

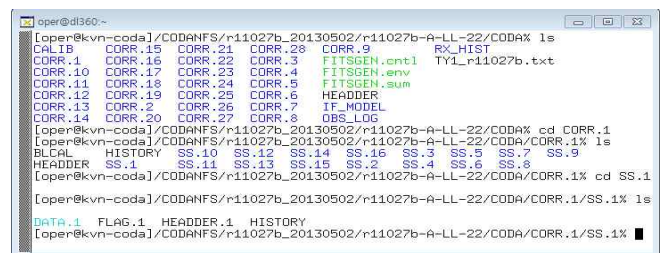


그림 10. 생성된 CODA 파일시스템

Fig. 10. Generated CODA file system

그림 11은 그림 10에서 생성된 CFS로부터 FITSgen 프로그램을 통해서 변환된 FITS 파일을 생성하고 데이터 분석을 위해 AIPS 프로그램에서 읽은 후 FITS 파일에 기록된 관측주파수, 관측정보, 가시도 정보 등을 자세히 나타낸 것이다.


```

CODA1 > FITLD1: Task FITLD (release of 31DEC11) begins
CODA1 > FITLD1: Create 027b267tst .UVDATA 1 (UV) on disk 1 cno 140
CODA1 > FITLD1: Sky frequencies (MHz) before re-ordering occurs:
CODA1 > FITLD1: Incoming FREQID # 1
CODA1 > FITLD1: 22226.06 22002.06 22034.06 22066.06
CODA1 > FITLD1: 22098.06 22130.06 22162.06 22194.06
CODA1 > FITLD1: 22226.06 22290.06 22290.06 22322.06
CODA1 > FITLD1: 22354.06 22386.06 22418.06 22450.06
CODA1 > FITLD1: CL table interval set at 0.05 minutes
CODA1 > FITLD1: Current file has 1960 visibilities
CODA1 > FITLD1: File ref. freq: 22002.06 MHz
CODA1 > FITLD1: Found 3C454.3 at 0/04:20:01 src # 2
CODA1 > FITLD1: fqid # 1
CODA1 > FITLD1: UV table spanned time: 0/04:20:01 - 0/04:21:54
CODA1 > FITLD1: 1960 vis. written
CODA1 > FITLD1: *****
CODA1 > FITLD1: Subarray or data out of order condition found.
CODA1 > FITLD1: NX/CL tables deleted.
CODA1 > FITLD1: Use USUBA to set up subarrays.
CODA1 > FITLD1: Rerun INDXR using CPARAM(3) and (4)
CODA1 > FITLD1: *****
CODA1 > FITLD1: Image=MULTI (UV) Filename=027b267tst .UVDATA 1
CODA1 > FITLD1: Telescope=KVN Receiver=VLBA
CODA1 > FITLD1: Observer=r11027b User #= 51
CODA1 > FITLD1: Observ. date=28-JAN-2011 Map date=28-MAY-2013
CODA1 > FITLD1: # visibilities 1960 Sort order TB
CODA1 > FITLD1: Rand axes: UU-L-SIN VW-L-SIN WV-L-SIN TIME1 BASELINE
CODA1 > FITLD1: SOURCE FREQSEL INTTIM CORR-ID
CODA1 > FITLD1: -----
CODA1 > FITLD1: Type Pixels Coord value at Pixel Coord incr Rotat
CODA1 > FITLD1: COMPLEX 3 1.0000000E+00 1.00 1.0000000E+00 0.00
CODA1 > FITLD1: STOKES 1 -2.0000000E+00 1.00 -1.0000000E+00 0.00
CODA1 > FITLD1: FREQ 128 2.2002062E+10 1.00 2.5000000E-01 0.00
CODA1 > FITLD1: IF 16 1.0000000E+00 1.00 1.0000000E+00 0.00
CODA1 > FITLD1: RA 1 00 00 00.000 1.00 0.0000000 0.00
CODA1 > FITLD1: DEC 1 00 00 00.000 1.00 0.0000000 0.00
CODA1 > FITLD1: -----
CODA1 > FITLD1: Coordinate equinox 2000.00
CODA1 > FITLD1: Maximum version number of extension files of type HI is 1
CODA1 > FITLD1: Maximum version number of extension files of type AT is 1
CODA1 > FITLD1: Maximum version number of extension files of type WX is 1
CODA1 > FITLD1: Maximum version number of extension files of type CT is 1
CODA1 > FITLD1: Maximum version number of extension files of type FQ is 1
CODA1 > FITLD1: Maximum version number of extension files of type AN is 1
CODA1 > FITLD1: Maximum version number of extension files of type SU is 1
CODA1 > FITLD1: Maximum version number of extension files of type TV is 1
CODA1 > FITLD1: Appears to have ended successfully
CODA1 > FITLD1: coda1 31DEC11 TST: Cpu= 0.8 Real= 1 IO= 51
coda1 LISTR(31DEC11) 51 28-MAY-2013 17:48:23 Page 1
    
```

그림 11. R11027B scan267번의 CFS로부터 생성한 FITS 파일을 AIPS에서 로딩한 상태

Fig. 11. Loading status at AIPS for FITS file generated from CFS of R11027B scan267

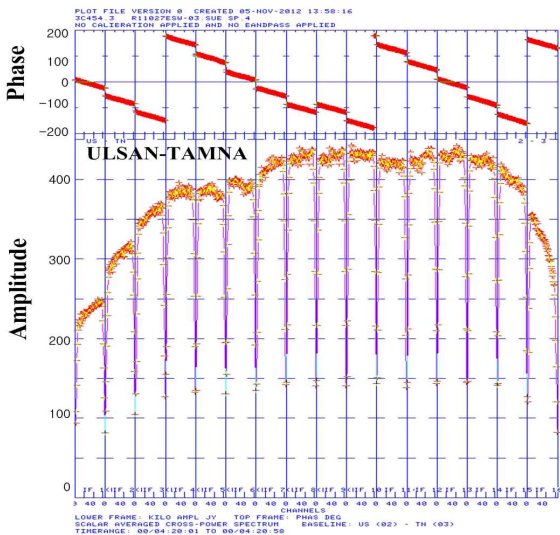


그림 12. R11027B scan267번에 대한 CFS로부터 생성된 FITS 파일을 AIPS에서 읽은 3C454.3 천체의 스펙트럼과 위상
Fig. 12. Spectrum and phase of 3C454.3 read FITS file at AIPS generated from CFS according to R11027B scan267

그림 12는 CFS로부터 생성한 FITS 파일을 AIPS에 로딩한 후 관측천체 3C454.3에 대한 스펙트럼과 위상정보를

나타낸 것이다. 그림 11과 12에 나타난 것과 같이 상관처리가 완료된 후 본 연구에서 설계·작성한 CFS로부터 FITS 파일을 변환하여 AIPS에서 읽혀진 경우 천문학자들이 분석작업을 진행할 수 있는 단계가 완성된 것이다.

이상의 시험결과를 바탕으로 리눅스 환경에서 호환성을 갖도록 C++ 언어를 이용하여 대전상관기의 상관후처리 소프트웨어의 핵심 역할을 담당하는 CFS 설계 및 작성 소프트웨어가 정상적으로 동작하고 있어서 본 연구에서 설계한 디렉터리 구조를 갖는 CFS가 유효함을 확인하였다.

V. 결론

본 논문에서는 대전상관기로부터 출력되는 상관결과 파일을 데이터 분석에 활용할 수 있는 CFS의 설계와 이와 관련된 소프트웨어를 개발하였다. VLBI와 같은 전파천문학에서의 상관결과는 천체의 가시도 성분(진폭과 위상성분)이 나열된 구조이지만, 상관결과의 과학적 분석을 위해서는 천체를 관측할 때의 기상정보, 전파망원경 위치, 관측시간, 천체의 위치, 천체의 종류, 관측장치 수신기 잡음온도 등의 여러 가지 정보가 필요하다. 리눅스 시스템에서 데이터 분석을 수행하기 위해 이러한 정보들에 대해 디렉터리 구조를 갖는 CFS를 설계하고 이를 생성하는 소프트웨어를 개발하였다. 본 논문에서 제안한 CFS는 일본국립천문대의 미타카 상관기의 CFS와 호환성을 가지며 대전상관기의 출력 파일을 자료분석에 활용할 수 있는 기본적인 파일 시스템으로 활용되고 있다. 본 논문에서는 설계한 파일 시스템의 유효성을 확인하기 위해 상관처리 결과에 대해 파일생성 시험을 수행하였으며, 이 파일 시스템을 FITS 파일로 변환하여 AIPS에서 천문학자들이 데이터 분석을 통하여 과학적 분석을 수행하는데 문제없이 정상적으로 동작되고 있어 설계한 상관결과의 CFS와 생성 소프트웨어가 유효함을 확인하였다.

참고 문헌

- [1] 노덕규, 오세진, 염재환 외 15명, “2008년도 한일공동VLBI상관기 및 수신기 개발 결과보고서,” 한국천문연구원, pp. 3-100, 2008.
- [2] 오세진, 노덕규, 염재환 외 6명, “VLBI상관서비스시스템 본제품의 제작현장 성능시험,” 신호처리시스템학회 논문지 제12권, 제4호, pp. 322-331, 2011.
- [3] Takahashi F., Kondo T., and Koyama Y., *Very Long Baseline Interferometer*, Ohmsha, pp. 35-55. 2000.
- [4] Thompson, A. R., Moran, J. M., and Swenson, G. W.Jr., *Interferometry and Synthesis in Radio Astronomy*, 2nd Ed., (New York:John Wiley & Sons), pp. 289, 2001.
- [5] AIPS webpage, <http://www.aips.nrao.edu/index.shtml>
- [6] 일본국립천문대, “CODA File System 사양서 ver 1.0,” 1994.

[7] Y. Kan-ya, The Specification of CODA File System, Leonid corporation, 2012.
 [8] Eric W. Greisen, "The FITS Interferometry Data Interchange Convention-Revised," AIPS Memo 114r, 2011.8.
 [9] Jacobson Ivar, Grady Booch, and James Rumbaugh. *The Unified Software Development Process*. Addison Wesley Longman, 1998.



오 세 진 (Se-Jin Oh)

正會員

1996년 2월 영남대 전자공학과(학사)
 1998년 2월 영남대 전자공학과(석사)
 2002년 2월 영남대 전자공학과(박사)

2001년 9월 ~ 2002년 12월 대구과학대학 교수
 2010년 6월 ~ 2011년 5월 한국천문연구원 상관계그룹장
 2002년 12월 ~ 현재 한국천문연구원 선임연구원
 ※주관심분야 : 디지털신호처리, VLBI상관계 개발, 천문 관측기기개발



칸야 유키토시 (Yukitoshi Kan-ya)

2000년 4월~2001년 3월 츠크바대 연구원
 2001년 4월~2007년 3월 일본국립천문대
 2007년 4월 ~ 2009년 2월 연세대학교 연구교수
 2010년 3월 ~ 현재 일본국립천문대 연구원

2010년 3월 ~ 현재 Leonid Corporation CEO
 ※주관심분야 : 전파천문, VLBI상관계 소프트웨어 개발



노 덕 규 (Duk-Gyoo Roh)

正會員

1985년 2월 서울대 천문학과(이학사)
 1994년 8월 동경대 천문학과(이학석사)
 1997년 8월 동경대 천문학과(박사수료)
 1985년 4월 ~ 현재 한국천문연구원 책임연구원

2005년 11월 ~ 2009년 3월 한국천문연구원 그룹장
 ※주관심분야 : 전파천문, VLBI상관계 개발



염 재 환 (Jae-Hwan Yeom)

2005년 8월 한양대 정밀기계공(석사)
 2005년~현재 한국천문연구원 선임연구원
 ※주관심분야 : 디지털신호처리, VLBI상 관계기 개발



오 충 식 (ChungSik Oh)

2002년 2월 서울대 천문학과(이학사)
 2006년 3월 동경대 천문학과(이학석사)
 2009년 3월 동경대 천문학과(이학박사)
 2009년 4월-2010년 11월 한국천문연구원 박사후연수원

2010년 12월 - 현재 한국천문연구원 선임연구원
 ※주관심분야 : 전파천문, Astrometry, VLBI상관계기



윤 영 주 (Young-Joo Yun)

1999년 2월 서울대 천문학과(이학사)
 2001년 2월 서울대 천문학과(이학석사)
 2011년 2월 서울대 천문학과(이학박사)
 2011년 3월 ~ 현재 한국천문연구원 선임연구원

※주관심분야 : 전파천문, VLBI상관계 소프트웨어 개발



정 진 승 (JinSeung Jung)

正會員

2008년 2월 경남대 전자공학과(공학사)
 2010년 2월 경남대 전자공학과(공학석사)
 2010년 8월 ~ 현재 한국천문연구원 연구원
 ※주관심분야 : 디지털신호처리, FPGA 설계, 천문관측기기 개발



정 동 규 (Dong-Kyu Jung)

2004년 8월 충남대 천문학과(이학사)
 2006년 8월 충남대 천문학과(석사수료)
 2012년 1월 ~ 현재 한국천문연구원 연구원
 ※주관심분야 : VLBI상관계기, 천문관측 기기 개발