H2020 Grant Agreement No. 730562 – RadioNet

PROJECT TITLE:	Advanced Radio Astronomy in Europe
STARTING DATE	01/01/2017
DURATION:	48 months
CALL IDENTIFIER:	H2020-INFRAIA-2016-1
TOPIC:	INFRAIA-01-2016-2017 Integrating Activities for Advanced Communities

Deliverable 6.4

Description of the Control of the digital frontend and backend, Recording and Correlation software.

Due date of deliverable:	2020-06-30
Actual submission date:	2020-12-07
Leading Partner:	STICHTING NEDERLANDSE WETENSCHAPPELIJK ONDERZOEK INSTITUTEN (NWO-I)

Document information

Document name:	Description of the Control of the digital frontend and backend, Recording and Correlation software.
Type	Report
WP	WP6 – BRAND EVN
Version date:	2020-12-04
Authors (Institutes)	Walter Alef (MPG) Hans van der Marel (NWO-I)

Dissemination Level

Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Index

RadioNet has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 730562

ELUCIDATION:

COVID-19 affected the Deliverable D6.4, to which the Art.51 applies as follows: Since the hardware (backend and frontend) were only delivered in autumn 2020, only a description and some testing of the control, recording and correlation software were feasible before the end of the RadioNet. The full evaluation will be done in 2021.

The control software of the digital frontend could be tested to the extent that a 0-baseline test between two independent samplers in the sampler chip could be performed and analysed (see deliverable D6.3). The recording could partially be tested with EHT data. The correlation tests with GMVA and EHT data signify roughly a 90% validity for BRAND data.

1 Introduction

The BRAND receiver needs to be integrated electronically into the Effelsberg antenna system for setting it up and controlling it via the standard Effelsberg station software. As the BRAND receiver prototype has been designed with the intention of making it most suitable for observations in Very Long Baseline Interferometry (VLBI) networks — in particular the European VLBI network (EVN) — it should be easily adaptable to other antennas. An important aspect is therefore the description of both the hardware and software interfaces.

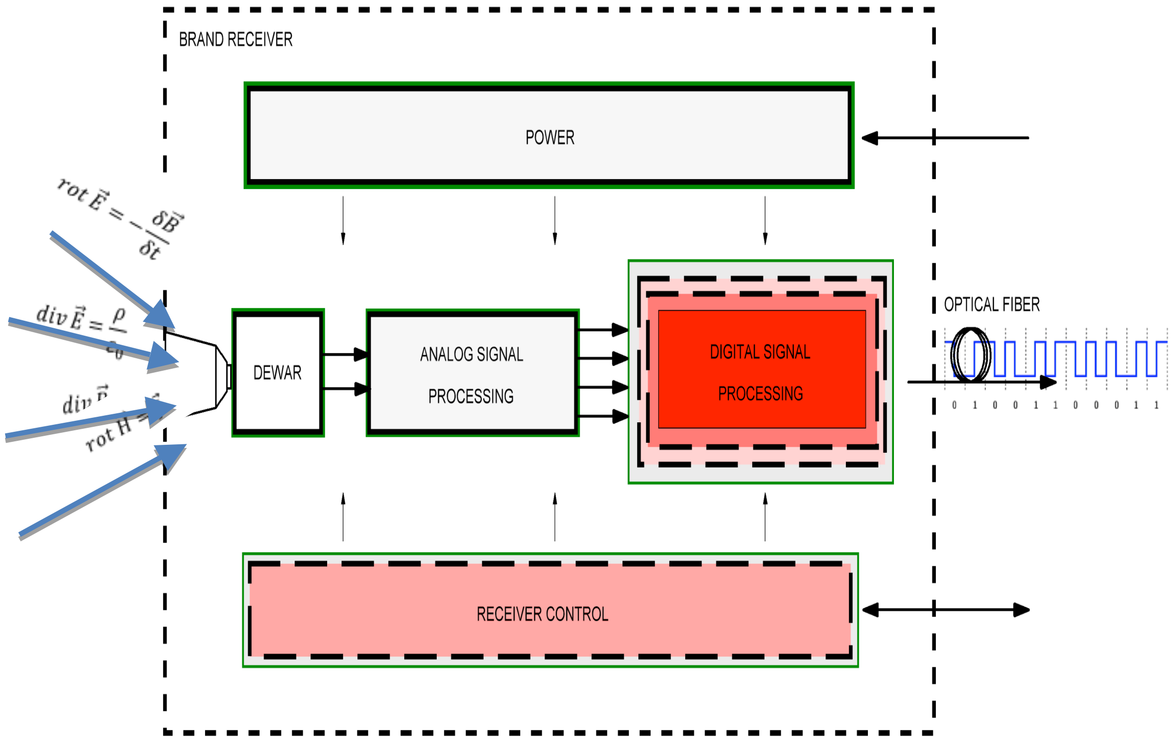

Figure 1: Block diagram of the BRAND receiver. At the left the electro-magnetic radiation is received by the feed horn. The dewar contains the directional coupler, filters, and the Low Noise Amplifiers, which are cooled to about 20K. This is followed by the analogue signal processing. The analogue signals are digitised followed by a digital band selection. The digitised data is output on the right via optical fibres and will be received at the telescope control room for further processing and recording. In addition, inside the receiver box are the power supplies for all units and the hardware interface for the receiver control. Power supplies and receiver control are connected to the outside for obtaining power and network connectivity.

RadioNet has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 730562

Here we have to distinguish between the analogue frontend and the digital frontend, which are both very different with respect to setup, control and monitoring. The analogue part is not very different from the frontends of other Effelsberg receivers. The digital frontend on the other hand is a novelty and only a basic software is being implemented initially to enable test observations as soon as possible. Later the control software will be enhanced to control more complex operations of the digital frontend. The output of the digital frontend is formatted as standard VDIF (VLBI Digital Interface Format)[1] packets and is transported over up to 64 10 GE fibres. This data can either be recorded directly for debugging and testing purposes, or it can be further processed by the digital backend.

The COVID-19 induced delays in the development of the digital backend board “BRAND_C” led to delays also in the development and testing of the corresponding control software. So this whole deliverable has been delayed by about six months. Other delays due to the pandemic have been introduced in the integration of the analogue frontend and the corresponding control software.

As digital backend serves a modified Digital Base-Band Converter 3 (DBBC3). The DBBC3, developed by a JRA of RadioNet3 (contract 283393), has been operational at several radio astronomy stations for a few years now under the control of the VLBI Field System [1]. It has its own set of control commands and software interface for the DBBC3. For the BRAND receiver this has to be extended. Due to COVID-19 access to the labs has been restricted. This lead to a delay of one to two months in this subtask.

The output of the modified DBBC3 is also formatted as standard VDIF packets and is transported over several 10 GE fibres. The enormous amount of data, which can reach up to about 100 Gb/s, will be recorded with Mark 6 disk recorders[2] (or similar fast RAID systems). This is a standard setup as is also being used by the Event Horizon Telescope (EHT) which records at the time of this report 64 Gb/s onto four such recorders with a total of 16 disk modules with 8 disks each. Due to delays in the digital frontend development this part could not be tested.

On the correlation side, in a “BRAND VLBI network”, a rather complex situation arises. The usable parts of the BRAND band from 1.5 – 15.5 GHz will be different at each telescope due to different Radio Frequency Interference (RFI). RFI above a certain limit will render the data in the affected part of the band useless, and this data will be removed/suppressed by the High Temperature Superconductor Filters, the digital frontend, and backend.

Rather than just one frequency setup, as is the standard in present day VLBI correlation — with exception perhaps of the EHT and VGOS —, a BRAND network will have a different frequency setup for each telescope! This complicates the correlation of the data significantly. Complicated control files would have to be created by hand with a lot of effort and a high probability of errors. The DiFX (Distributed FX) correlation software has been enhanced to make the correlation of BRAND data simple and transparent to the user.

2 Control software for the analogue frontend

No documentation of the control system for the BRAND analogue part of the frontend has been provided. This part of the BRAND receiver has the standard Effelsberg receiver interface. Its functions will be accessible to the receiver maintenance group at Effelsberg and to the operators during observations. It is expected that appropriate documentation will become available within the next months.

[1] *http://ivscc.gsfc.nasa.gov/publications/gm2010/whitney2.pdf*

[2] *https://www.haystack.mit.edu/mark-6-vlbi-data-system/*

RadioNet has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 730562

3 Control software for the digital frontend

3.1 Introduction

The BRAND_C digital backend consists of one Quad 8-bit 56GSps ADC and up to four Kintex Ultrascale FPGAs. The central hub for controlling these devices is the Control Software, which runs on a small computer connected to the board via USB. It is connected to a small Spartan3 FPGA on the board, which is used as a controller to configure and monitor the other devices.

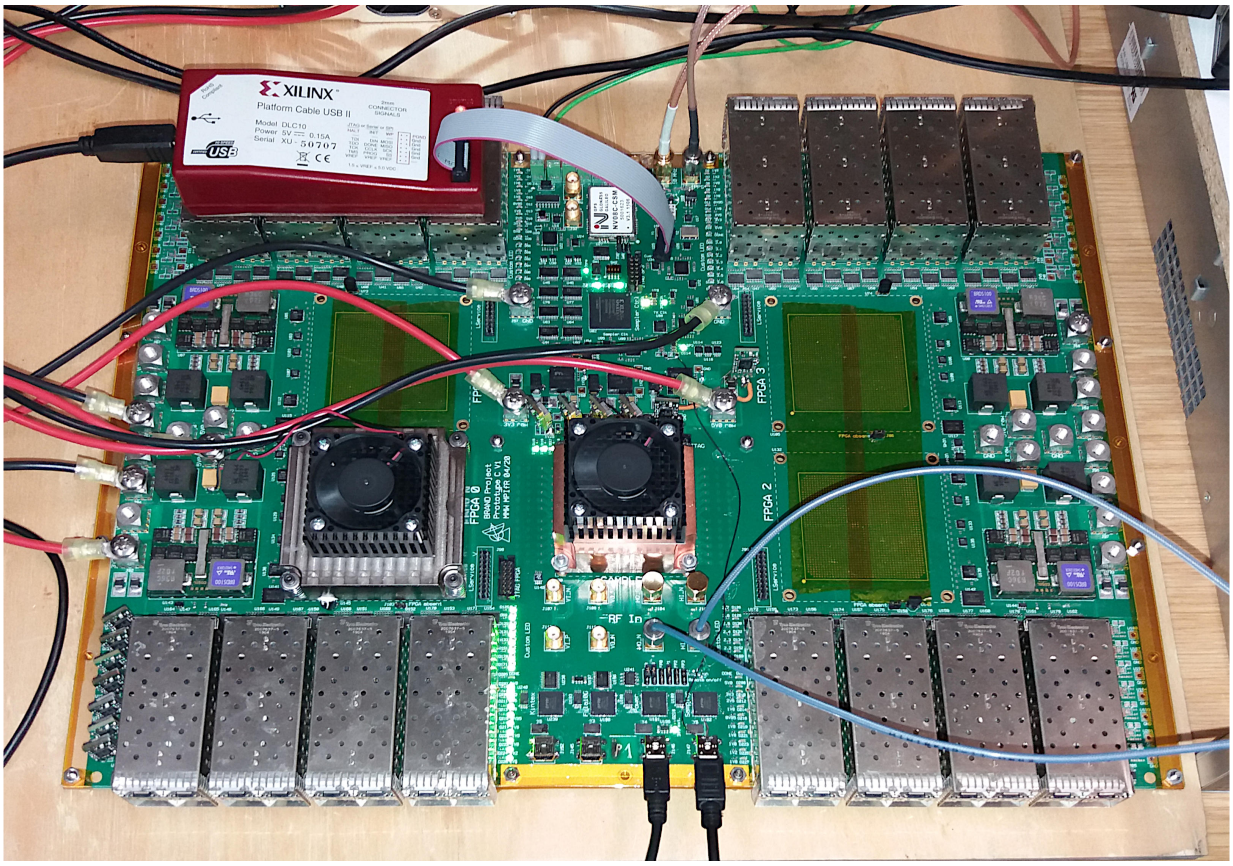

Figure 2: Digital Frontend board. The sampler chip with a fan for cooling is seen in the centre. The lower left of the four Kintex FPGA sockets is populated. The small Spartan3 FPGA is seen on the top centre of the board just above the black cable.

One part of the software developed for this project is the implementation of the central control software, and also the modification and development of the firmware for the Spartan3 FPGA on the digital backend board[3].

The Kintex FPGAs on the board will receive the sampled data from the ADC and perform the first filtering stage, which is necessary to deal with the large bandwidth and data rates. The filtered data is then packetized and sent via 10G Ethernet connections to a DBBC3 digital backend, which performs the final digital post-processing. This part of the project is described in deliverable D6.3.

[3] *In this deliverable only the central control software will be described. The firmware is described in the report of the digital hardware (D6.3).*

RadioNet has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 730562

3.2 Control software for the Spartan3 FPGA

The control software is the central hub for the operator to interact with the BRAND_C digital frontend. It runs on a small computer that is connected to an FTDI USB controller[4] on the digital frontend board. This FTDI controller is connected to the Spartan 3 Controller FPGA via a parallel bus. This interface allows the Control Software to interact with the ADC, which is also connected with the Spartan 3 via control lines.

In addition there are two synthesizers on the board, which are used to produce the necessary reference frequencies for the sampling process on the ADC and for the data transmission via HSSI (High Speed Serial Interconnect) between the ADC and the Kintex FPGAs. These synthesizers are also connected to the Spartan3 to be able to configure them via the control software.

The board also includes a couple of small ADCs, which are used to monitor the voltage levels. These ADCs are also directly connected to the Spartan3, which allows the Control Software to monitor the voltage levels and ensure that the voltages are in the correct range for the ADCs and FPGAs to operate properly.

The control computer is connected to each Kintex FPGA on the board via USB using the RS232 protocol, which allows a serial communication between each FPGA and the control computer.

The control software has been developed in Python 3. This allows the control software to run on Windows and on Linux operating systems without any changes to the source code. Python is very well suited for this kind of application, where different kinds of hardware APIs need to be integrated. There are a lot of open source libraries available in Python for the different functionalities needed, including APIs for communicating with the FTDI controller, the serial communication via RS232, Client/Server frameworks and graphical user interfaces.

The BRAND_C board is based on a development board purchased from the manufacturer of the ADC, which came with a set of Matlab scripts for interacting and testing the ADC device. These scripts include the following functions:

- communication with the ADC over SPI protocol (Serial Peripheral Interface)
- power on and start up the board and ADC
- configure the ADC
- perform the necessary calibration routines
- capture and plot data from the internal memory on the ADC device.

These Matlab scripts where used as a base for the control software development in Python, which now includes the same functionality as provided by the Matlab routines. It was further extended to include all additional functionality needed to perform the initial tests, including a zero-baseline test between two ADCs on one board (in the same chip).

The largest workload by far was the translation of the several thousands of lines of m-Code (used by Matlab) to Python3. This had to be done manually, because of subtle differences in these two programming languages. An automatic translation routine could not be used.

Using Matlab to implement the Control Software was not an option, due to the performance requirements of Matlab and general stability and reliability issues with the Matlab environment.

The first step in the development process was to implement the communication interface between the Control Software and the ADC. The control computer is connected to the board using a USB connection to an FTDI chip, which is connected to the Spartan3 Control FPGA. The FTDI uses a parallel communication protocol to talk to the Spartan3, which in turn uses an SPI (Serial Peripheral Interface) protocol to communicate with the ADC.

[4] *https://en.wikipedia.org/wiki/FTDI*

RadioNet has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 730562

We found a library for Python 3 that implemented the functionality to talk to the FTDI device, which was another reasons to use Python for the control software. The SPI protocol was implemented using the Matlab example scripts provided by the sampler manufacturer.

The next step was to implement the routines necessary to start, initialize, configure and calibrate the ADC device, based on the examples in the Matlab script provided. Since the scripts use internal MatLab libraries for plotting data or performing FFT calculations, we had to find alternative libraries for Python to implement this function.

To make the software usable, a simple terminal window function was implemented, using the cmd library included in Python. This provides simple shell functionality, allowing to type commands to issue the steps for setting up the system.

As soon as the communication with the sampler was verified, and the calibration routines worked as with the development board of the sampler manufacturer, the next step was to implement the functionality to perform a zero baseline test between two ADC channels on the same device and between two different boards, which is crucial to prove that the board is working as intended. For this it was necessary to modify the capturing procedure, which is triggered by an external signal, so that both ADCs start the data acquisition at exactly the same time. The Sampler device has 16 MB of on-chip memory for this purpose. Due to the huge bandwidth (28 GHz per channel) a precise triggering of the data acquisition was crucial to generate matching data streams.

Below follows a list of all implemented commands available in the control software:

- startup (starts up and initializes the synthesizers for the ADC and activates the ADC clock)
- poweron (checks the system voltages, performs a system reset, and load the initial configuration of the ADC)
- activatechannel (the ADC has four internal channels, this commands selects which channels should be activated)
- deactivatechannel (deactivates the selected channels of the ADC)
- calibrategain (performs the gain calibration, necessary for the correct mapping of the input channel voltages to the sample range)
- calibration (performs the power and offset calibration for the selected channels)
- capture (capture the specified amount of samples from the internal memory. If multiple channels are activated, they are captured in sequence, not in parallel)
- captureparallel (captures two data streams from the first two channels to the internal memory in parallel. Necessary for the zero-baseline test)
- plot (plots the frequency spectrum of the selected and captured data stream)
- plottimeseries (plots the captured data streams in the time domain)
- save (saves the selected data stream into a file, in ASCII format)

The following functions are still under development:

- Serial communication (via RS232) with the FPGAs. The library that will be used for this functionality has already been tested successfully with the DBBC3 backend, which has the same type of communication. It will be implemented as soon as the Firmware for the Kintex FPGAs will be available, which is required for further testing and verification.
- Server/Client support. In its final version it is planed to have a Client/Server Architecture similar to the Control Scheme of the DBBC3, but this has still to be determined, depending on the requirements of the stations using the BRAND receiver. The library for this functionality has already been tested successfully, and is in use for the DBBC3 as multicast-parser.

RadioNet has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 730562

- GUI (Graphical User Interface): At the moment the control software is used in a terminal window. In the final version a graphical user interface may be implemented, but this has not yet been decided, and depends on the requirements of the stations using the BRAND receiver.
- Control of the Power Supply: It was planned to have the ability to power on the board via the control software. But this requires a hardware modification of the actual design, so it is not implemented yet.

3.3 Firmware for the Control FPGA

The control architecture of the BRAND digital backend is very similar to the development board of the sampler manufacturer, so it was possible to use the firmware for the Spartan3 control FPGA provided by them with minor modifications. The following modifications were made to the firmware:

- the original design has three small ADCs connected to the Spartan3, used to check the system voltages. We connected additional five ADCs to read out the voltages for the Kintex FPGAs.

The firmware was modified to include these connections.

- The BRAND board has additional synthesizers, which are also connected and can be controlled by the Spartan3.
- The provided firmware did not include the functionality to trigger a data acquisition with an external signal. This functionality was necessary for the zero-baseline test and thus included.
- The clock line of the first design revision of the BRAND board was not shielded properly and thus we had stability issues due to the 60 MHz reference clock line. A modification to the board fixed this problem, but it was necessary to change the clock input pin.

4 Software for control of the digital backend

A DBBC3 serves as digital backend. It will be used as the final processing stage, and will receive and process the data from the BRAND_C digital frontend via up to 64 10G Ethernet connections. It will output the processed data via several 10G Ethernet lines to data recorders or the Internet.

In its original form the DBBC3 consists of pairs of up to eight ADB3L ADCs and Core3H processing boards, which perform the digital processing of the data sampled by the ADCs. For the BRAND project the ADB3L ADCs will not be used, instead the Core3H boards will receive the data over the available 10G Ethernet lines.

The currently available firmware does not have the capability to receive data via Ethernet, only the transmitting part is used. This was left out for performance reasons in the original project[5].

For the BRAND project the number of Ethernet transceivers has been increased from four to eight. The following firmware modifications are necessary for the BRAND project, and are currently implemented and tested:

- support for eight transceivers, instead of four.
- reactivation of the receiving part in the transceiver IP cores.
- synchronization of the incoming data to produce one parallel data stream

The synchronized data will then be processed by additional filters, either fixed bandpass filters (OCT mode) or variable Digital Down Conversion units (DDC mode). These modes are already implemented and in use for observations with the DBBC3 digital backend. The data is output in standard VDIF format, ready for recording and correlation.

[5] *JRA DIVA of RadioNet3 (contract 283393)*

RadioNet has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 730562

For the additional functions new commands will be defined and implemented in the internal control software of the DBBC3. They will be accessible via the Ethernet connection to the DBBC3 control computer. It is expected that this software will be implemented in Q1/2021.

5 Software for handling the data recording

The amount and rate of the data generated by the BRAND receiver is comparable to that of the EHT. We expect that when the full bandwidth is selected in the digital frontend and backend a maximum data rate of 112 Gbps can be reached with a BRAND receiver system.

The data rate of the EHT is presently 64 Gbps. The present upgrade plans foresee that when two frequencies will be observed, a data rate of 128 Gbps can be reached. The EHT records their data on four Mark 6 recorders, each with four disk modules containing eight disks each. For the APEX and Pico Veleta telescopes a DBBC3 has been used for the EHT.

It follows that for the recording of BRAND data the same setup and software will be used. Thus no special software development is needed for recording BRAND data. The DBBC3 will deliver the data to the data recorders, which will store it in the same way as done for the EHT.

6 Modifications to the DiFX software correlator[6]

The open source DiFX software correlator has been widely adopted by the astronomical and geodetic VLBI communities. It is in production use at several large VLBI networks including the LBA, VLBA, IVS, VGOS, GMVA, EHT, EAVN, and others. It supports multiple phase centres, near field, line/continuum VLBI, space VLBI, pulsar VLBI, and radio transient searches.

With the “zoom bands” supported since DiFX 2.5+ one can successfully correlate experiments where stations have mismatched frequency setups. Zoom bands are narrower frequency bands which fall within the recorded frequency bands and are extracted via FFT during correlation.

Highly mismatched frequency setups occur in VLBI with radio interferometer arrays such as the phased SMA, ALMA, and NOEMA. Being designed for interferometric observations independent of each other and of VLBI, the characteristics are uncommon for VLBI; recorded bands are narrow, may overlap, sampling rates may not be a power-of-2 in MHz. Lesser incompatibilities may also occur at VLBI stations that use direct RF sampling or full IF sampling systems such as BRAND, CSIRO Bluering, Effelsberg EDD. BRAND for example is anticipated to provide consistent 128 MHz wide channels, but tuning is expected to differ between BRAND-equipped and other telescopes.

Between BRAND antennas, on the other hand, the RFI situation will be quite different resulting in different usable parts of the band. Rather than just one or very few frequency setups, as is the standard in present day VLBI correlation, a BRAND network will always have a different frequency setup for each telescope! This complicates the correlation of the data significantly.

Correlation is usually possible with zoom bands. In the extreme, however, experiments with two or more stations having mismatched frequency setups may require an excessive number of DiFX zoom bands that additionally may have unequal bandwidths. The DiFX correlator supports such unusual setups natively, but the common export formats of the DiFX correlator FITS-IDI[7] and HOPS[8] do not.

The GMVA-ALMA 2017 campaign encountered precisely these issues and was the trigger for the DiFX developments. It serves also as the example for the implemented generic workaround, shown in Figure *3*. Recorded frequencies are first split into suitably fine-grained zoom bands that are used in a

[6] *A detailed report will be published on Zenodo at https://zenodo.org/communities/radionet-eu-brand/*

[7] *http://www.aips.nrao.edu/FITS-IDI.html*

[8] *https://www.haystack.mit.edu/haystack-observatory-postprocessing-system-hops/*

RadioNet has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 730562

normal DiFX zoom band correlation. Neighbouring zoom bands are concatenated along the frequency axis (or, “de-zoomed”) to form wideband visibilities. A spectral averaging step can be added to yield the PI-requested number of channels. The final frequencies (or, “output bands”) of the visibilities cover the original observed sky frequency range and have a FITS-compatible uniform bandwidth. The method described above for recouping FITS-IDI/HOPS compatible visibility data was implemented as a standalone DiFX utility at first (difx2difx.py). Later it was also implemented natively in the DiFX correlator (DiFX Output-bands branch; likely future DiFX 2.7 release).

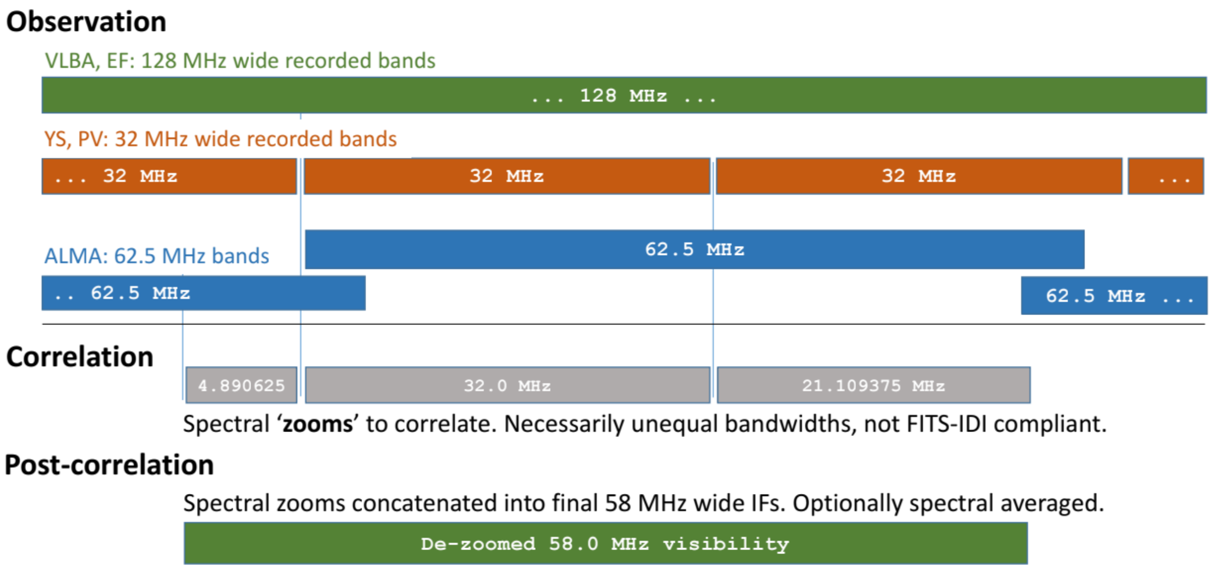
Observation
VLBA, EF: 128 MHz wide recorded bands
... 128 MHz ...
YS, PV: 32 MHz wide recorded bands
... 32 MHz
32 MHz
32 MHz
...
ALMA: 62.5 MHz bands
62.5 MHz
.. 62.5 MHz
62.5 MHz ...
Correlation
4.890625
32.0 MHz
21.109375 MHz
Spectral ‘zooms’ to correlate. Necessarily unequal bandwidths, not FITS-IDI compliant.
Post-correlation
Spectral zooms concatenated into final 58 MHz wide IFs. Optionally spectral averaged.
De-zoomed 58.0 MHz visibility

Figure 3: Example of a mixed-frequency setup, not to scale, showing the first GMVA-ALMA observation Sgr~A in 2017. Correlation with requested 58~MHz wide zoom bands was impossible due to the inadvertent narrow-band 32~MHz setup at two stations (orange boxes). Correlation under DiFX~2.5 with manually configured zoom bands (grey boxes) and post-processing the visibilities recovered the desired 58~MHz IF(s) (bottom) and allowed data to be converted to FITS-IDI.*

The correlator workflows of the two approaches are illustrated in Figure *4*.

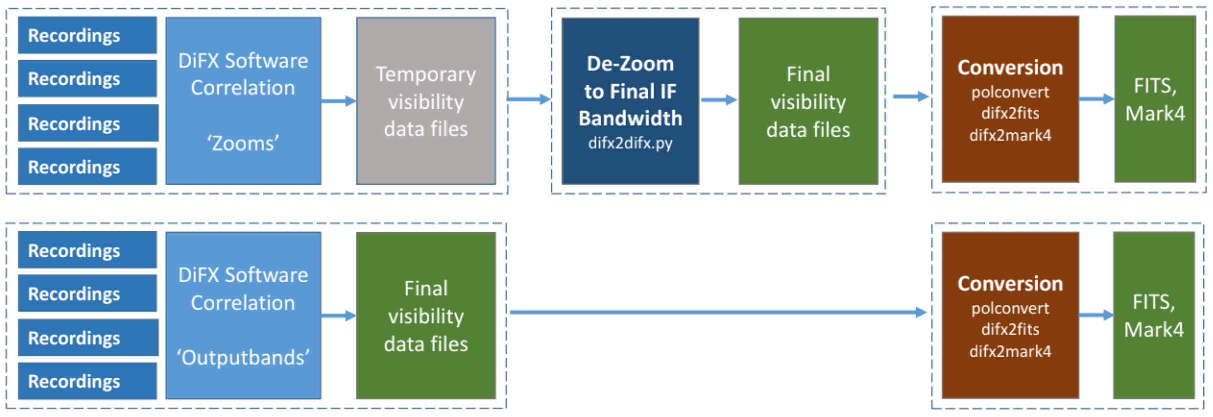
Recordings
Recordings
Recordings
Recordings
DiFX Software Correlation 'Zooms'
Temporary visibility data files
De-Zoom to Final IF Bandwidth difx2difx.py
Final visibility data files
Conversion polconvert difx2fits difx2mark4
FITS, Mark4
Recordings
Recordings
Recordings
Recordings
DiFX Software Correlation 'Outputbands'
Final visibility data files
Conversion polconvert difx2fits difx2mark4
FITS, Mark4

Figure 4: Processing steps for an offline implementation (top) using visibility data from plain DiFX~2.6. Native processing in DiFX Outputbands (bottom) avoids the overhead of temporary files, at the cost of changes to DiFX

RadioNet has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 730562

6.1 Script for manual conversion of output bands

The DiFX correlator generates correlated data in the cross-spectral domain. This simplifies the task of generating "legal" output bands for data export significantly. All zoom bands can quite easily be concatenated in cross-frequency space followed by cutting the resulting band(s) into sub-bands of the same width. In the extreme case the user can choose whatever width is considered useful for further data reduction.

A Python-based spectral reassembly script (`difx2difx.py`) was implemented in 2017 and is in production use for GMVA VLBI correlation. Given a set of DiFX visibility data files and a configuration file `difx2difx.py` carries out the concatenation of zoom bands present in the data, as sketched in Figure *3*, and frequency averages the final output visibilities. Output are a new set of DiFX visibility data files and their metadata.

The offline processing was tested for geodetic-like VLBI using visibility data from a simulated VLBI observation. The devised frequency setup had consistent BRAND-like 128 MHz recorded bands that were purposefully offset by multiples of 1 MHz at the different stations. The synthetic observation was correlated in DiFX 2.6.1 with 1 MHz zoom bands. Output bands could be synthesized successfully across the covered common sky frequency range. Processing had lower overheads than GMVA VLBI processing. Overall results indicated that BRAND-utilizing VLBI observations when correlated under DiFX 2.6 can indeed be processed using the offline method.

6.2 Native Implementation in DiFX Correlator

The zoom-band/difx2difx solution can become quite cumbersome and error prone for complex scenarios as we envisage for BRAND and other mixed networks. A native implementation was thus made in DiFX to ease the operation and improve upon the correlator workflow and its robustness. This will be particularly important once the EVN or other networks will observe regularly with mixed setups.

Changes to DiFX were kept to a minimum; the visibility data format remains unchanged, configuration files likewise except for the addition of an optional new parameter. Software changes included non-architectural changes to the correlator (*mpifxcorr*), extension of the correlation setup utility (vex2difx), and extension of a library (*difxio*). Converter utilities (*difx2fits*, *difx2mark4*) were updated negligibly by calls to a new frequency lookup function in *difxio*.

6.2.1 Vex2difx

New parameters were added to the DiFX configuration file (v2d file). Correlation jobs that want to use the outputband feature need to specify the following optional parameter:

- SETUP section: outputBandwidth = [auto | <bandwidth> in MHz]

When the *outputBandwidthparameter* is specified and has a setting of auto, an output IF bandwidth is derived based upon the observed sky frequencies and recorded bandwidths of all stations as defined in the VEX input file. Alternatively, the output bandwidth can be set explicitly with <*bandwidth*>, e.g, 58.5 for 58.5 MHz. Note that there are no guarantees that a particular output IF bandwidth can indeed be synthesized from the frequency setup in the VEX file. *Vex2difx* will show an error in this case, and a different bandwidth setting should be tried.

A second optional parameter, *gainOffsets*, was added to allow experimental rescaling of amplitudes in the recorded bands:

- ANTENNA section: *gainOffsets* = < *Δgain of freq 1*, *Δgain of freq 2*, ...> *with Δgain* ∈ [–1,1] that are relative to unity gain

The new *gainOffsetsparameter* permits manual alignment of the bandpasses of the recorded bands that contribute to an output band. Similarly, DiFX 2.6 parameterf *reqClockOffs* allows removal of delay offsets (phase offsets) between recorded bands. Both parameters can be used to compensate for discontinuities that may occur inside an assembled output band. This

RadioNet has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 730562

manual approach is tedious and *gainOffsets* in particular will affect amplitude calibration. The preferred approach is to instead use a station-based complex bandpass calibration during postprocessing.

6.2.2 Automatic Zoom Band Generation

In both modes of the *outputBandwidthparameter* (auto, or explicit bandwidth) vex2difxintroduces new zoom bands as necessary.

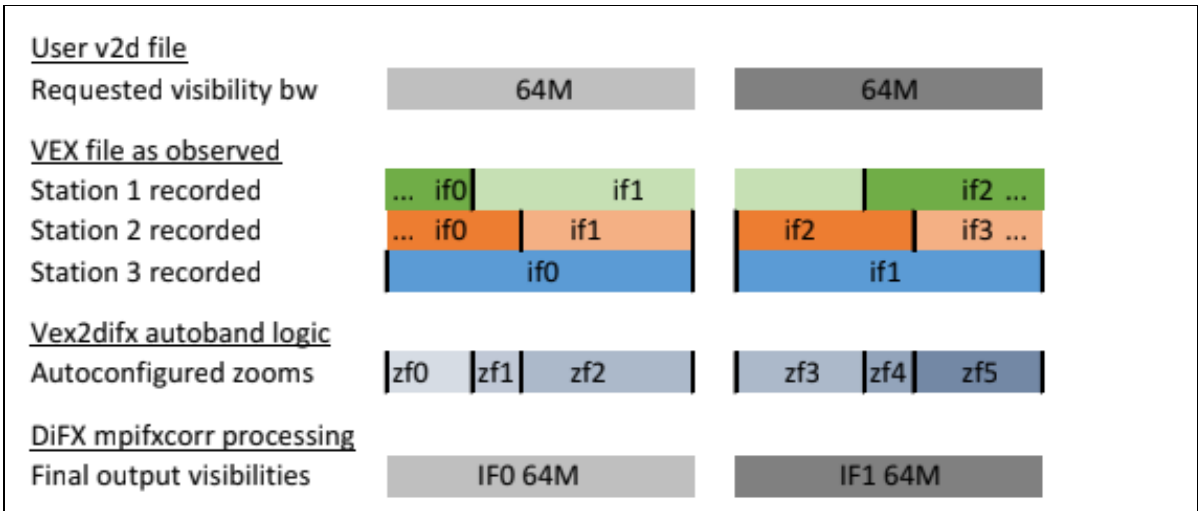

Figure 5: Illustration of automatic zoom band generation in vex2difx driven by band edges of recorded bands and the requested output visibility bandwidth. Each zoom band gets assigned a target outputband (N: 1 mapping) and the required assembly of visibility data is performed in mpifxcorr at correlation time.

The process of automatic zoom band generation is illustrated in Figure 5. Starting from a target bandwidth (v2d file) the sky frequency range covered by the recorded bands (VEX file) is segmented into spectral slices, dictated by band edges of the recorded bands. Any spectral slices with insufficient stations are discarded. Spectrum towards a new output band is accumulated from the remaining set of consecutive spectral slices. This is done by injecting matching-size or narrower-sized zoom band definitions (zf0 to zf2 in Fig. 3) into the spectral slice. The zoom bands are added into the DiFX .input file. This continues until one complete outputband IF is covered (IF0 in in Fig. 3). The bandwidth accumulation process is repeated for the next outputband IF using the remainder of spectral slices, using, e.g., zf3 to zf5 to produce IF1 in Fig. 5.

6.2.3 Internal Mapping of Visibilities into Outputbands

DiFX *mpifxcorr* was extended to allow concatenation of visibility data, with time and spectral averaging. The implementation redirects visibility data of zoom and recorded band(s) directly into the correct spectral channels within the final target band (N: 1 mapping). The mapping is defined in the .input file. The spectral concatenation process has a relatively compact implementation. For details see report at *https://zenodo.org/communities/radionet-eu-brand/.*

6.2.4 Other minor problems solved

1. Frequency Averaging: DiFX 2.6 computes fringe-stopped auto spectra at high spectral resolution. A frequency averaging step is applied optionally and at an early stage, while the cross-products are formed. All visibility data are stored in averaged form. With output bands, the high resolution auto spectra of zoom bands can have a number of spectral channels that is not divisible by the averaging factor. Bookkeeping of fractional channels was added.

RadioNet has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 730562

2. Spectral Channel Flagging: Channels of an outputband may need to be flagged when they carry no meaningful phase information, or their amplitude is hard to calibrate. It is also possible that one spectral output channel is the average of the channels of two contributing bands.

The easiest approach is to flag such channels. *Vex2difx* was extended to produce a flag file (*<jobname>.channelflags*) for each correlation job. Each line contains one flagging entry, consisting of an antenna name, MJD time range, numerical ID of a DiFX frequency, the range of channels (zero-based) to flag in that frequency, and lastly a flagging reason.

The converter program *difx2fits* was extended to import the channel flags directly into FITS-IDI table FG#1, making them available to CASA and AIPS postprocessing.

6.3 Special Calibration Steps in Post-Processing

In addition to basic flagging, there are two critical calibration steps for DiFX output-bands. Namely, a priori amplitude calibration with modified T^*_{sys} data that account for combined bands, and complex bandpass calibration to remove in-band discontinuities.

1. Flagging: In case of FITS-IDI data the FITS file comes pre-populated with channel flag records for all outputbands,
2. Calibration of Flux Density Scale: Calibration is based on a time series of station T^*_{sys} measurements, and a static DPFU and gain-elevation curve model. The T^*_{sys} data are usually determined over frequency ranges identical to the recorded VLBI bands. With outputbands data of several recorded bands might be combined.
3. Calibration of In-Band Amplitude and Phase: When the frequency setup of an observation is such that several narrow recorded bands must unavoidably be combined in order to provide an array-wide common output band, certain inter recorded band mismatches will produce discontinuities in the visibility amplitudes and phases within the output band.

   This problem will most likely not apply to BRAND data, but might show up in observations in mixed networks. Spectral mismatches occur due to per-band differences in digital signal processing in a VLBI backend, or in a phased array correlator and VLBI formatter. Such effects are known from GMVA and EHT observations.

   Future study is still needed and should inspect whether a single bandpass calibration, which could solve these problems, remains stable throughout an observation, whether such a calibration works in position-switched spectral emission line observations, as well as how geodetic VLBI observables may be affected by residual in-band phase discontinuities or residual mismatched delays between bands.

7 Literature

[1] Himwich, E., "Introduction to the Field System for Non-Users", in: International VLBI Service for Geodesy and Astrometry 2000 General Meeting Proceedings, 2000, pp. 86–90.

[2] Deller, A. T., "DiFX-2: A More Flexible, Efficient, Robust, and Powerful Software Correlator", Publications of the Astronomical Society of the Pacific, vol. 123, no. 901, p. 275, 2011. DOI:10.1086/658907.

RadioNet has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 730562

8 Acronyms

ADC	Analogue Digital Converter
AIPS	Astronomical Image Processing System
ALMA	Atacama Large Millimetre Array
APEX telescope	Atacama Pathfinder Experiment Telescope
API	Application Interface
BRAND	BRoad-bAND
CASA	Common Astronomy Software Applications
COVID-19	Corona Virus disease 2019
CSIRO	Commonwealth Scientific and Industrial Research Organisation
DBBC3	Digital Base-Band Converter 3rd generation
DDC	Digital Down Conversion
DiFX	Distributed FX
DPFU	Degrees Per Flux Unit
EAVN	East Asian VLBI Network
EDD	Effelsberg Digital Device
EHT	Event Horizon Telescope
EVN	European VLBI Network
FFT	Fast Fourier Transformation
FITS	Flexible Image Transport System
FITS-IDI	Flexible Image Transport System - Interferometry Data Interchange
FPGA	Field Programmable Gate Array
FTDI	Future Technology Devices International company
FX	Refers to an FX correlator: Fourier transformation followed by cross correlation
GMVA	Global Millimetre VLBI Array
GSps	Giga samples per second
GUI	Graphical User Interface
HOPS	Haystack Observatory Processing System
HSSI	High Speed Serial Interconnect
ID	Identifier
IF	Intermediate Frequency
IVS	International VLBI Service for Geodesy and Astrometry
LBA	Long Baseline Array
Matlab	Software
MHz	Mega Herz
MJD	Modified Julian Date
NOEMA	Northern Extended Millimetre Array
OCT	flexible in input band position and width filtering mode
RAID	Redundant Array of Independent Disks
RF	Radio Frequency
RFI	Radio Frequency Interference
RS232	Recommended Standard 232: Serial data protocol
SMA	Submillimetre Array telescope
SPI	Serial Peripheral Interface
T_{svs}	System Temperature
USB	Universal Serial Bus
VDIF	VLBI Digital Interface Format
VEX	VLBI EXperiment definition
VGOS	VLBI Global Observing System
VLBA	Very Longa Baseline Array
VLBI	Very Long Baseline Interferometry
10 GE	10 Gbit Ethernet

RadioNet has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 730562