JIVE/EVN CORRELATOR OUTPUT; FORMATS & DIAGNOSTIC TOOLS

Huib Jan van Langevelde
JIVE, Radiosterrenwacht Dwingeloo

Roger Noble
NRAL, Jodrell Bank

`version 2.00, December 1996`

This is a rewrite of EVN #68, formerly called "JIVE/EVN correlator output; thoughts on formats & tools". In its current form it formulates a high–level plan for the output format and diagnostic software for the JIVE/EVN correlator, as adopted by the EVN/JIVE correlator project team. Besides an introduction on desired software functionality, it also concerns the hardware implementation of the off–line interface.

1. Introduction

The JIVE/EVN MkIV correlator will flush out data at a very high rate. Raw correlation coefficients can be created with a typical rate between 50 kByte/s and 1 MByte/s. Peak output rates could be several MByte/s. At that stage the correlator hardware has done its job, but there is a considerable amount of work left to do in the daily operations of the processor. The data will need to be archived, and it will be necessary to review the quality and validity of the correlator output. In this stage one can also anticipate the collection of data necessary for feedback to the telescopes. The data has to be made available to the astronomer, together with the complete calibration data. And finally the correlator product needs to be archived.

This document defines the functions that need to be performed and discusses possible implementations of these tasks. It arrives at a plan in which is formulated what will be the boundary between on– and off–line software, it proposes to archive data in "internal" format, in a similar fashion as is anticipated for WSRT DZB development. It is thought that it is best to start the development of diagnostic software in AIPS^{++}. The same layer would also be used to convert data to a UVFITS distribution format.

2. The Constraints

Different boundary conditions arise from hardware and software decisions that have already been agreed on. Before it is possible to discuss possible implementations, or even the functionality that is desired, it is important to agree what these constraints are.

2.1. Data Capacity

First we estimate the data rate that the correlator can be expected to produce. It is not useful to ask what data rate the correlator could possibly deliver. Especially with recirculation, the data rate could become so high that any currently feasible off–line system would become too limited and no astronomer could process the data.

Therefore, it is a useful exercise to look at some typical projects. An 8 station EVN continuum experiment would possibly be processed with 16 channels per baseline, and typically 8 IF's. With 1 second integration time this would yield a typical data rate of 64 $\times$ 16 $\times$ 8 $\times$ 3 $\times$ 4 bytes/s, including some overhead this would be 100 kByte/s. For a reasonable integration time of 4 seconds this would be a data rate of 25 kByte/s. Including the effects of speed–up, this number could be increased by 2 or even 4, but there will be no cases where we are forced to use speed–up as is the case for the VLBA. For a global experiment the total number of products increases by a factor of 4 to 100 kByte/s. For a simple EVN spectral line experiment with a single transition and 256 channels, we can anticipate a similar rate as the number of IF's is reduced by 4 at the same time. When a higher resolution is required, or with polarization the data rate increases to 200 – 400 kByte/s. When doing high frequency (e.g. SiO masers) global VLBI, there is a need to decrease the integration time and 1.6 – 3.2 MByte/s data rates would be favorable. This, we estimate, is currently a reasonable upper limit to constrain the required data rate. It is always possible to process a certain data set more than once to obtain more output. However, for very high spectral resolution or other special purposes (possibly Space VLBI) there might come a need for higher rates.

Such data rates (25 kByte/s – 3.2 MByte/s) produce respectively 720 MByte or 92 GByte in a 8 hour experiment. This last number can be taken as an indication that few astronomers would currently be able to process the data from such an experiment. Below we will argue that 0.3 – 1.0 MByte/s and a total volume of approximately 15 GByte is a feasible production rate with the current hardware. This seems a reasonable match to current astronomical needs.

2.2. Data Structure

The correlator will run a sequence of jobs. A job can be made up from multiple tape–passes or more appropriately scans. Where a scan is defined as a period, during which the source, observing mode and number of telescopes remain constant and there are no tape stops. Multiple correlator configurations are allowed in one job, but data for different projects does not occur in a single job. There are some operational constraints on the size of a job. A single job should not produce too much data, because a problem may call for recorrelation of such a job, and the output volume should be small enough to fit comfortably on a single physical output device

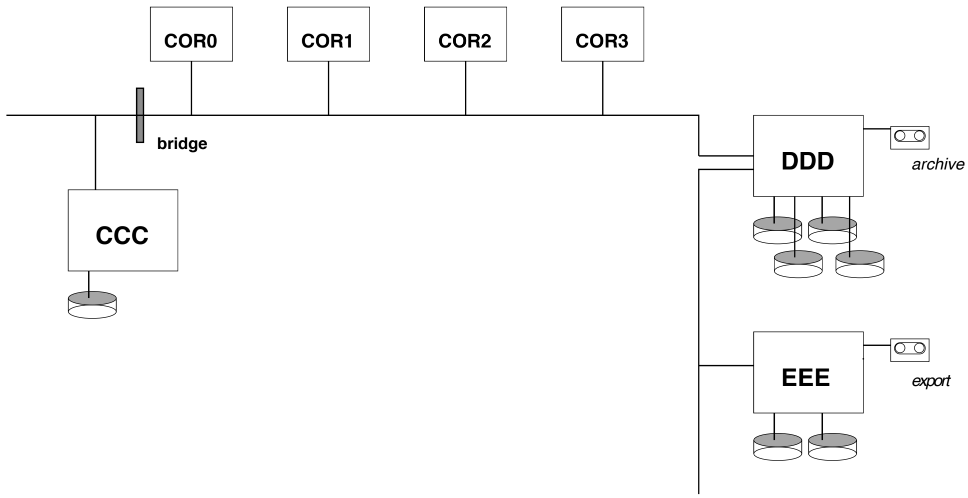

Fig. 1. Planned configuration for the JIVE/EVN correlator output infrastructure. D^3 *is the Data to Disk Distributor, a specially equipped workstation for fast data storage.* E^3 *is the Evaluation and Export Engine. The latter is a relatively standard workstation which accesses the data produced by the correlator. It can be thought of as a "logical unit", rather than a physical workstation. In some (stages of) implementations it could be any or many workstations, possibly* D^3 *or even* C^3 *(the Correlator Control Computer) in the development stage.*

Every scan in each job can be thought to consist of several "baselines", which contain the data on a single telescope pair (an autocorrelation should be considered such a pair too). On a single baseline data could exist in different polarizations and IF bands. Traditionally in VLBI it is logical to keep all the IF's on one baseline together, but in principle these are created in different places in the correlator hardware; these data entities are often referred to as "interferometers".

During operation the correlator chips in one of the correlator segments in one of the correlator units produces correlation functions at a number of delays for a particular baseline and IF/polarization (an "interferometer"). Which data are created where, is under control of the configuration software. After correlation the data is accumulated on one of the DSP's, which are shared between 32 chips on one correlator board. After accumulation in the DSP, the data are handed to the RT board of that correlator unit. The integration time is typically 1 second, but some astronomical problems require much shorter integration (from field of view constraints $\lesssim 0.1$ second) or longer integration (to allow large spectral datasets $\gtrsim 20$ second).

From the RT the data are flushed onto an Ethernet network. In the operational environment this piece of Ethernet is separated from the rest of the world by bridges in order to have optimal traffic between the correlator segments and the "data handler" (let's call it the Data to Disk Distributor: D^3). It is envisioned to be a workstation with disk capacity to store the data of several jobs. The correlator RT units currently have 10 Mbit/s Ethernet connections. In Fig. 1 a systematic description of the hardware is given.

The data from a certain "time stamp" is collected simultaneously on the off–line system. Thus, the data comes naturally time sorted. There are 4 RT systems, one for each correlator unit. Each collects data from a number of "interferometers". We assume that for each time stamp the data is labeled, but in principle randomly ordered. The on–line software for this has not been finalized.

2.3. Software Tasks

The software on D^3 that accepts the data is considered a part of the real time system. Apart from flushing the data from the correlator units to disk, it can also be expected to collect data from other correlator subsystems, such as header data, playback statistics and e.g. phase cal detection. In principle it is here that the data could be reformatted if we would decide to store it in a manner different from the correlator output, in particular if the data needs to be sorted by baseline rather than time first. Simple diagnostics on the data that comes in could possibly also be performed by D^3, but IO operations should probably be limited. It is currently thought that the Ethernet connection and disk access will be the bottleneck of the system.

Finally, there is an issue of spectral versus lag domain for the output data. The correlator can produce both under control of the configuration software. It is planned that data in the spectral domain will be calculated eventually in the RT or DSP hardware. It seems possible that production of spectral data will not be available initially, as it requires implementation of FFT calculation in the DSP or RT. However, in the long run there is a preference to flush out spectra, as it would limit the amount of data by a factor two.

2.4. Manpower

The system for data validation and export, should be developed under the constraint that manpower for correlator operations is limited. We try to design a system that automatically generates condensed quality control information. Time consuming data copying processes should be avoided; as data transfer is the limiting factor for the maximum out put rate, it can be understood that copying the data from one medium to another can easily take as long as the correlation process itself. Finally, in order to use resources efficiently one should be able to access the data quality as quickly as possible, preferably when correlation is in progress.

Therefore, the correlator operator should be considered the first step in the data validation process. First, because we need to use this source of manpower; when the correlator runs smoothly the operator has probably time to run a few simple diagnostic tasks to check whether the machine creates sensible data. We can also hope to detect problems early and not correlate blindly and find out later what went wrong. Furthermore, it allows real time investigation of both correlator and observing project specific problems by correlator staff. Another advantage is that this way the correlator operators get an interest in the data they are creating. Some of the tools for this are covered by the "on–line diagnostic software", but it would certainly be advantageous to make the outcome of the off–line software available quickly too.

A final consideration is that these functions should be implemented with limited manpower. We estimate that the amount of software covered in this memo requires almost a full person for *maintenance*; small changes to hardware, on–line software or data reduction packages will need to be tracked constantly. Hence, it seems favorable to base our efforts on existing platforms and ongoing projects. We considered in particular AIPS^{++}, HOPS and TMS (the WSRT Telescope Management System, which covers the output format of the DZB).

3. Functionality

In this section the functionality of the components considered is discussed. There are five sections. The first considers the on–line diagnostic software. This we argue to be a part of the real time system. It runs on C^3 or D^3 and displays messages and graphs that monitor the systems performance. The other four items are off–line diagnostics, the collection of calibration data, distributing the data to the astronomer and archiving. These systems we consider to be the off–line system.

3.1. On–line diagnostics

In this section we describe the tools necessary for real–time display. In short these are the items that the correlator operator needs access to instantaneously. In our philosophy it is important that the operator has a way to estimate whether the correlation he is doing is producing any valid data. This will enable him to decide to go ahead, or to try to move the tapes to different drives, or even to skip to the next project. Other reasons to have real–time display are debugging of the correlator and investigation of recording problems. Finally, such tools will be valuable during fringe searching. We list a simple list of tasks that could help meet these goals.

- *Observation summary.* After preparation of correlator jobs, configuration of the correlator etc. an observation summary is needed. It is important that the operator can know at any instant what was supposed to happen during the observation (the schedule) and what did happen (the logs). In this way he can understand why only a few tapes are spinning, or whether strong correlations are to be expected or not.
- *Playback statistics.* Next obvious thing are the playback statistics, i.e. a plot or list of the failure rate of the frames and time decoding of the different tracks of the data on each drive. Here it is important to emphasize the word track; a data stream in VLBA or MkIV format may come from several different physical tracks, because of 2–bit sampling flows into separate bit streams, fan–out and finally barrel–rolling. Statistics per track are necessary to point to recording or playback problems, statistics per data stream are necessary to weight the data. Another nice feature would be to display the head position history, so that the operator can get a feel for how well the data is being tracked.
- *Observables.* It will be necessary that the correlator operator has the opportunity, by means of the on–line software, to inspect some of the observables, for instance amplitude and phase, probably versus time or in the form of (lag) spectra. With some astronomical knowledge this allows a check of a proper functioning VLBI telescope. It may provide direct feedback in some cases of problem shooting or clock searching. For this it would be very favorable to have a delay and rate measurement returned by the system.

3.2. *Off–line diagnostics*

This item covers the tools to validate the correlation and investigate problems in an efficient way. In any case, this can be performed in the off–line data reduction package (AIPS or AIPS^{++}). This route will have to be available, because the users will have to be able to read the data into their favorite package. Once the data is read into such a package, a variety of tasks for plotting UV–data is available which can be used to check the functioning of the correlator in detail by inspecting the final data product.

But it is not feasible to read all data from all projects into AIPS and perform detailed checks, even though a certain level of automatic processing is available in AIPS (and surely in AIPS^{++}). Especially as it is anticipated that most projects will have to be released (i.e. having the data tapes recycled) without intervention by the PI. This calls for some automatic procedure to verify the data quality.

At the same time we will need to collect extensive data on the performance of individual stations in the network. Experience over the last couple of years has shown it to be essential to get feedback to the telescopes in the field as quickly as possible, ideally within a month. Otherwise problems cannot be identified clearly and fixed on a time scale where a fix can be confirmed. Such information consists for example of detailed information on playback, phase cal detection and bandpass response, and, in a later stage, sensitivity and clock evaluation.

Based on experience at the VLBA a list of *possible* checks contains the following items.

- *Check of several correlator parameters.* Before releasing data it is necessary to check whether the data was correlated according to the astronomer's wishes. These items include source position, time and frequency resolution, stations, and possible cross polarizations.

- *Playback statistics.* These should measure the read–back per track on the tape. For instance by showing the success rate for decoding frames per output visibility (the "weight" in AIPS jargon). It would be best to have these by track, rather than data stream. In the case of barrel–rolling this can identify head problems at the playback unit or recorder in the field. These statistics serve two purposes. First they are needed by the PI in order to understand his data. Second we want to pass this information back to the stations to allow them to correct recorder problems. So, we need to collect the playback statistics per antenna over many projects as well as all statistics for a certain project. In principle this holds for many of the item listed, but recorder problems are certainly the main station based problem we have to watch out for.

- *Cross power spectra.* These show for strong sources whether fringes are present. They immediately bring forward problems at the station acquisition rack and are the main place to spot gross problems in the correlator hardware. Cross–spectra should be plotted for example every hour or for every new source.

- *Autocorrelation spectra.* These help to distinguish whether the problems above are station based (e.g. interference) or a correlator problem.

- *Fringe solutions.* It should be possible to follow these in time for strong sources or calibrators. The ultimate check whether VLBI works is to see delay and rate solutions for strong sources. They are also the last step in our effort to monitor the clocks at the stations and this data should be kept for future reference. It would be important to have a reliable determination of the signal to noise ratio of the detected fringes.

- *Phase and amplitude in time.* Some correlator problems, like inaccurate model interpolations etc, will show up as modulations of correlated amplitude and especially phase.
- *Distribution tape.* One test implicitly done in the VLBA system is reading back the distribution medium. This turns out to be a non–trivial check as these overloaded DAT and Exabyte tapes often show problems.

3.3. Archiving

From the archives it should be possible to recreate the distribution data, as it was sent to the astronomer, at any time in the future. There are basically two stages at which we can do this.

- *In "raw" format.* The most direct way is to archive directly any output from the correlator in its internal format. This makes the archiving task simple. Any re–formatting that needs to be done can be left for later. An advantage is that it is in principle fast, we probably don't want to create additional overhead in the tasks that will be emptying the data disks for the correlator. Furthermore, correlators notoriously run at 60% efficiency only, so no time would be wasted on processing data which might not be useful after all.

  In this way the correlator output comes on archive tapes in the order they were ran. Retrieving something from the archive or distributing all data in a project will be a little more difficult. Some database system is needed to find back where which correlator job is to be found, the conversion to the astronomer FITS file needs to be done, and will have to be written to the distribution medium. We have to be aware that it will be quite common that certain correlator jobs will have to run more than once, and a mechanism is needed to decide which of the correlator jobs is the one that has to go onto the final product. In such a scheme the correlator output is stored independent of the off–line layer. This has the advantage that bug fixes and enhancements in the off–line layer can be applied to historical data. On the other hand will it be necessary to have the off–line layer detect in which version of the on–line software the data was created and assure backward compatibility of the off–line software. This could be considered a disadvantage; we are defining a format that basically only exists in our archive.
- *In distribution format.* Another scheme is to archive the data only in the final distribution form. This probably implies that the data remains on the output disk quite a bit longer and will be sorted in the archive step. By this is meant that a certain archive tape contains data for one project only. It would make the archive recovery a bit easier, as it reduces to a copy operation, although some database program to maintain the archive is still needed. This would be easiest implemented when the data for an entire project could be stored on disk until it is completed. However, this would

probably require 200 Gigabyte disk space. Working with a 20 Gig pool, in principle enough to keep a days work, requires additional effort.

Having the data archived in output format calls for more robustness in the off–line layer; "raw" correlation output is not easily recoverable in this scheme. This is a serious disadvantage. Another problem is that archiving will require processing power, especially because the data are only accessible directly by computers that are critically involved in the data collection.

3.4. Conversion to distribution format

An important first assumption is that the production of data for the astronomy community is the prevailing task of the correlator. However, we also wish to produce data that is usable for the geodetic community. To our best knowledge there is no strong format requirement for geodetic data, although HOPS output has been used directly by geodesists in the past. At the VLBA work is in progress to make data available to the geodetic packages though AIPS. Most importantly it should be possible to derive totals from the correlator output to input to CALC/SOLVE. Even if the processing path would exist in AIPS or AIPS^{++}, it will require high level JIVE manpower to test the geodetic and astrometric properties of the data.

We assume that the off–line data reduction will be mostly in AIPS, AIPS^{++}or, after calibration, in DIFMAP. To read the data into these packages a FITS distribution format is needed (although if an implementation of diagnostic software in AIPS^{++} is chosen we have in principle a tool that reads JIVE/EVN internal format data into AIPS^{++}). A complete description of the content of a valid UV–data FITS file, suitable for the AIPS VLBI data reduction package is needed. Work on such a description is in progress at NRAO (Diamond et al. in preparation). We estimate that the development of software to write UVFITS is a major job. The reason is that the UVFITS file will need to contain all the auxiliary data about geometry etc. in the right form to allow the use of VLBI specific calibration in AIPS.

An important requirement for modern VLBI is that full accountability of the correlator model should be provided. For geodetic, astrometric and phase referencing purposes it may be necessary to construct the total phase from the data, or to correct for certain model components. This requires us to store a polynomial description of the phase that was taken out by the correlation, as well as numerical values for all the components used in the model. Comparable data are available from the VLBA correlator.

3.5. Calibration data

There are several streams of calibration data that have to be made available to the user. An obvious example is the attachment of logged total power data and flags. These are obtained during the observations and kept with the recording logs. It is preferred that this is accommodated in our off–line layer in order to provide the calibration data directly appended to the (proper) FITS files. Therefore, it should be made available along the same lines as calibration data from the correlator (see below).

Another category are the phase–cal detection and possibly state count statistics. In principle these are a product of the correlator, but they are not produced within the correlator hardware, but in the station units. The phase–cal detections allow lining up multiple IF bands and possibly amplitude calibration, the state counts allow an amplitude correction for each station that takes into account threshold levels of the samplers, probably most important for two bit sampling. Both these calibration factors are ideally appended to the final distribution product. In principle they could also be applied to the output data, but we feel it should be left to the astronomer to decide whether and how he wants to apply these.

The case for correlator based corrections is a bit different. Maybe some corrections can be given in extension tables. But corrections that arrive from detailed knowledge on the correlator configuration are best applied to the data before they are distributed.

One important constraint coming from all this is that these measurements must be made available at the same place where the output data reside, in particular it will need to be archived in a similar way as the main data.

4. Solutions

Taking the constraints into consideration we have reached the following plan, which is described first on an infra–structural level (sect. 4.1). We outline an implementation in the sect. 4.2.

4.1. Infra–structural choices

Our first consideration is that it is anticipated that in the current system the Ethernet connection to D^3 and the access time to its disks will be the bottle necks. The current Ethernet connections on the correlator RT systems are realistically limited to 0.3 MByte/s, a modest data rate for an average spectral line project. The hardware configuration in Fig. 1 can be upgraded to a system with 4 times higher throughput without upgrading the RT systems in every correlator unit, by connecting the four 10 Mbit/s outputs to a router/bridge with a 100 Mbit/s connection to D^3. Such a configuration would cover the output rate requirements described in section 2.1. To obtain eventually a speed higher than $\gtrsim$ 1.2 MByte/s for the output rate of the correlator, a 100 Mbit/s Ethernet connection seems needed. This will probably allow data rates up to 3 (12) MByte/s without special measures, sufficient for most thinkable applications, but we are concerned that at that point the access time for the data disks, even with special hardware, will become a problem.

In any case, we feel that these considerations demand that only as limited reformatting or sorting of the data as possible should be done in D^3. Therefore, the data will be flushed to D^3 in its “raw” form. In this context “raw” means without processing after the DSP and RT systems in the correlator units. These processors may tag and label the data and possibly even Fourier transform it or perform digital corrections. This is possible because they represent distributed processing power, however we want to exclude any processing by central computers (C^3 or D^3) on the combined data stream. Therefore, the “raw” or internal data format cannot be sorted by data types, or by baseline, in cannot be relabeled or calibrated by central processing power.

Because the assignment of each "interferometer" is in principle flexible we could possibly still organize a certain sort order for the correlator output. However, we think it is not wise to rely on this and we propose that the data from each "interferometer" and time stamp should have a unique tag. With this identification and the correlator configuration one should be able to identify each of these data records eventually. However, it remains feasible, by means of the correlator configuration, to choose a mapping of "interferometers" that reflects as closely as possible the optimum structure for off–line access, based on the implementation strategy adopted. As an example it can be imagined that for the VLBI application it is desirable to group all "interferometers" that produce data on a certain baseline together (time – baseline – IF sorted).

Furthermore, the proposed scheme implies that complex mathematical operations in D^3 are to be avoided. This excludes Fourier transforming data into the desired domain at this stage. This imposes a constraint: either we ask the data to come out of the correlator in the same domain every time, or we need our processing down stream to be able to handle both cases.

With these large data volumes we aim for a solution in which it will not be necessary in the off–line analysis to duplicate the data. Loading VLBI projects like these from tape or copying across machines is subject to the same data bandwidth constraints and may take many hours. In particular, we think it is best to run the off–line diagnostic software as the data is created on disk. This is also desirable in order to have fast feedback from off–line diagnosis. Within the frame–work sketched here, one would think D^3 could perform its task with something like 20 GByte of disk capacity, based on daily recycling.

To allow access to data on D^3 disks by diagnostic software on another machine, it would be required for D^3 to have two Ethernet controllers, in order to avoid interference with the data collection process. This would remove the necessity for a bridge between $D^3$and the outside world (Fig. 1). We envision an Evaluation and Export Engine (E^3) that may access the data on the D^3 disks. This machine is also used to reformat the data into the distribution format, either directly by accessing the D^3 disks or after archive data are read back onto its local system. Total disk space on E^3 should be 10 GByte.

Although the decision on the archiving medium is not yet made, we think this does not affect the choice in what format this should be done. We favor an implementation in which the data are archived in their "raw" form. This will allow the fastest archiving. The most obvious archiving media have access rates that are of the same order of the median correlator output rate (0.1 – 0.3 MByte/s). This approach will make the data in the archive dependent on the version of the on–line software and independent of conversion software. It requires the conversion software to remain compatible with older versions of the on–line software.

Archiving could be done locally on D^3, but should be postponed during peak output rates. Another solution for this problem would be swappable disks, that could be mounted on D^3 or E^3, but this doesn't seem to be required at the moment. We think that the archiving medium could be either DAT or CD. DAT's offer the advantage of larger capacity and faster write times. Moreover, they are cheaper and we could possibly afford a couple of drives on the system simultaneously. Most other correlators have DAT's. At the VLBA the philosophy is that every 5 – 10 years when new technology becomes available the complete archive data has to be copied. Data on CD seem to have a better lifetime, however the write times are slightly longer and the data capacity per unit is lower. The distribution to users will be predominantly on DAT as far as it is possible to foresee.

In our scheme it is required that from the "raw" format distribution data can be produced. This implies that the "raw" data on D^3 has all auxiliary and calibration data attached to it. So we think it is the responsibility of the on–line software developed in Jodrell Bank to deliver all this data to D^3. Not only correlator configuration data and job dependent calibration, but also tape statistics and even telescope calibration data, which the correlator in practice does not need to operate. Note that this is a very limited volume of data that has to be transferred from C^3 to D^3, so it does not seriously limit the data rate. However, all these different products complicate the definition of the format of the "raw" data considerably.

Concluding, we describe a system in which data is gathered on the disks of D^3. Direct archiving is done of "raw" data on D^3, preferably when the correlator is not processing high data rate projects. Automatic processes on E^3 attempt to run the diagnostic software on–the–fly, as the data are created by the correlator. If the (Ethernet) resources are not available to keep up with the data rate of the correlator, this process may fall behind without any consequences to the correlator operations (simple diagnostic tools are available in the on–line software after all). Distribution in user format is done on E^3, either by accessing D^3's disks or after restoring archived data locally.

4.2. Software choices

It seems desirable to implement the required functionality on existing platforms. From the available options we favor a scheme in which most functionality is developed in AIPS^{++}. Below we describe briefly how we envision implementation in this case, covering the required functionality. Then we discuss in detail the cases for both HOPS and AIPS^{++}; which have both advantages for implementation of the diagnostic software,

- *On–line diagnostics.* We feel that these should be part of the real–time system. Development for this is to be done at Jodrell Bank.
- *Data archiving.* The data archiving is done in internal format. We wish to collaborate with the efforts in Dwingeloo to establish a structure for the archiving of WSRT data. This can only work if the formats are of comparable information content. The responsibility for these efforts lies with JIVE in Dwingeloo.
- *Off–line diagnostics.* The automatic data evaluation will be done in AIPS^{++}, using both built–in and correlator specific tasks, combined in a high level script.
- *Distribution.* The distribution to the user will be implemented in AIPS^{++}. Distribution in UVFITS format is an expected functionality of AIPS^{++}, but we anticipate considerable effort in this area is required to take place in Dwingeloo. Completion of this path has highest priority in the off–line/diagnostic software project, because it will enable data processing and quality control in standard packages.

- *Calibration Data.* In our view, the gathering of the calibration data is part of the effort of making the internal format data available in AIPS^{++} as a Measurement Set.

The responsibility to implement the three AIPS^{++} items lies with JIVE in Dwingeloo. A detailed description of the format of both the "raw" output data and auxiliary data needs to be established early between all parties. This is the necessary interface definition between the on–line software group, the correlator construction team and the people responsible for the implementation of diagnostic/distribution software described in this document.

Below we discuss the cases for AIPS^{++} and HOPS, as well as the possibility to install both. A best estimate of the consequences for implementing each is given. In a previous version of this document it was discussed to have FITS as the output format of the correlator. This option would require reformatting in D^3 and copying the data before inspection. Therefore we see it no longer as a reasonable alternative and it is left out.

4.2.1. HOPS

The MkIII correlators, notably the one in Haystack, run HOPS. It seems to offer most of the functionality desired. It is transparently written and could be easily extended. Another advantage is that Haystack will eventually use the package to access data quality from their own correlator, based on the same hardware, although a smaller version. However, there are still many places where implementation choices are based on the old HP 1000 architecture and with geodesy applications in mind. Some of these choices will affect the flexibility and efficiency of connecting HOPS to the JIVE/EVN correlator at some point.

A brief description on how things are organized in HOPS follows before the different aspects of an implementation in HOPS are discussed. Some documentation can be found in `http://dopey.haystack.edu/vlbi/hops.html`.

HOPS data are stored in a directory structure by "baseline/frequency scans". These are identified by the encoded time at which they were ran. Internally the data is stored in lag space. Fringe fits run normally on the data after it has been correlated and this results in a summary plot and a quality code assigned to the data. The fringe fitting is on a scan basis with a single multi band delay and rate for each scan, using the phase tone detection. A quality factor is attached to the data, and carried along.

As correlation for the project continues, the disk fills up with data for a project. There are powerful tools to investigate which parts of the data are correlated and one can view the quality codes of the entire project at a glance. In addition, there are several ways to deal with redundant data, for instance one can select the data with the highest quality or from the latest correlation for "baseline/frequency scans" that were produced in more than one job. When the inspection shows that the project is completed satisfactorily the data can be archived in a project based manner. Distribution is currently in a MkIII specific format, but could be in FITS in the future.

We continue to discuss how all the functional demands are met in this option and what the implications are for the implementation.

- *Distribution format.* Currently it is possible to write a "baseline/frequency scan" into a FITS file. A Fourier transform to frequency space is performed in this step. As is clear from the discussion in the previous section it is not trivial to go from there to the FITS data structure we desire for the end user. This area would require considerable development effort. The problem is not too severe because of the fact that HOPS has knowledge about the data structure of the entire experiment. So all the information is available to write FITS files of the desired data content.

- *Data validation and feedback.* As described above, HOPS has very impressive tools to do fringe fits and display data quality based on many different characteristics. It seems clear that it could in fact be set up in such a way that some of the results are made available to the correlator operator during correlation. One concept may need to be changed, namely the idea that a fringe fit is usually done over the period of a scan. There is work in progress at Haystack to allow automatic processing of spectral line data. There are some other issues that need attention in the automatic quality assignment. Examples are phase referencing observations, where there is not necessarily a detection in good quality data and observations without phase cal tones.

  The access of playback statistics of MkIV data requires some enhancements of HOPS. The current situation is of course that a data stream originates from a specific track on the tape. Tools to examine the playback statistics before the data streams are reconstructed are needed (as they are in any of the proposed solutions). A simple task to archive playback statistics by station is needed.

- *Calibration data.* We expect some work to be involved to make telescope based calibration data available (i.e. flagging and system temperatures) through HOPS. The same thing holds for the correlator based calibration, (e.g. phase–cal and state counts).

- *Archiving.* In the current HOPS implementation the data for a entire project remains on disk until a project is completed. Tools to do this exist with the HOPS framework, but in the JIVE/EVN case we would choose to archive the data in the format in which it resides on D^3. It is estimated that the HOPS data structure has all the information available to reconstruct the distribution format from archived data. The currently used scheme in HOPS to have all data reside on disk until a project is finished is deemed to be impossible, because it is estimated to require 200 GByte of disk space.

- *Implementation & Maintenance.* As is clear from the above, a number of enhancements of HOPS is necessary to accommodate HOPS for MkIV and specifically the needs of the JIVE/EVN correlator. For some of this initial work has been carried out at Haystack. It is anticipated that for a number of things desired in the JIVE/EVN correlator JIVE manpower will be needed. These are notably the things that interact with the on–line system (as these will differ between Haystack and JIVE) and some of the astronomy specific things. In addition one would think that a lot of work on the FITS distribution would have to be maintained in Dwingeloo. It looks like it will not be difficult to learn to code HOPS programs, although the data structure in which the data and parameters are maintained does not appear to be efficient for our needs.

To accommodate HOPS at the JIVE/EVN correlator, D^3 will need to distribute data coming from different baseline scans to different files. This we think would be a problem.

With up to 256 baselines, considerable sorting is implied. The task to open 256 files quickly or continuously either puts a burden on D^3 or would require special software effort on a low level. Furthermore, it would require the correlator to produce data in the lag domain. A solution in which the data is copied from "raw" format into HOPS format is not in accordance with the description in the previous section.

4.2.2. AIPS++

There seem to be a number of advantages for implementing our software layer in AIPS++. In this scheme it is envisioned that the "raw" data is dumped on disk without further sorting. While the data is written, table structures are being build up by the AIPS++ table manager, running on either E^3 or D^3(Fig. 2). This makes the data available to AIPS++ on E^3 (see AIPS++ memo 111). In principle, the storage manager allows the data to be in any form on disk. In practice it would be best if this would follow the logical data structure in AIPS++ for rapid access, this is time, baseline, IF, polarization and spectra. AIPS++ will allow data to be stored in lag domain, although this is not currently a part of the Measurement Set. For most applications it will transform the data to frequency domain, which is the "normal" way to store data in AIPS++, as it is for AIPS.

The data storage manager allows data from different sources and on different devices to be in a single logical structure. It is estimated that to create such a storage manager for JIVE/EVN correlator data, would be relatively simple. To make available all auxiliary data would require more work. This is necessary to turn the JIVE/EVN table into an AIPS++ Measurement Set, required to allow calibration software to access the data. Non–astronomic data from the correlator, like tape statistics, could be stored in other AIPS++ tables.

It is thought that AIPS++ is sufficiently mature for such an application. Simple plotting and manipulation tasks are already available, complicated (VLBI) processing tasks are anticipated in the next year. We are lucky in the sense that the table structure and storage mechanism is the single most established part of AIPS++, and it is this part that we are most critically dependent on. The WSRT DZB effort has (independently) chosen this route for their development. We note here that the WSRT implementation differs from our plan; there the AIPS++ Storage Manager to read the data is a standard AIPS++ product. To accommodate this, sorting of the data is performed as a part of the (AIPS++) output routines in that system. We propose to avoid a processing step (at least for the correlator data output) and write a specific Storage Manager for the JIVE/EVN correlator data.

- *Distribution format.* Writing out the data in FITS is a task which will have to be available in AIPS++ one day. We would probably have to implement an early or special version of this task to make the desired FITS data structure. It is probably possible to make a logical AIPS++ dataset encompass all the data on disk for a certain project, but as it will be difficult to store an entire project on disk or tape, there is a need to be able to break these structures into smaller pieces, with the information of the global data set remaining available.

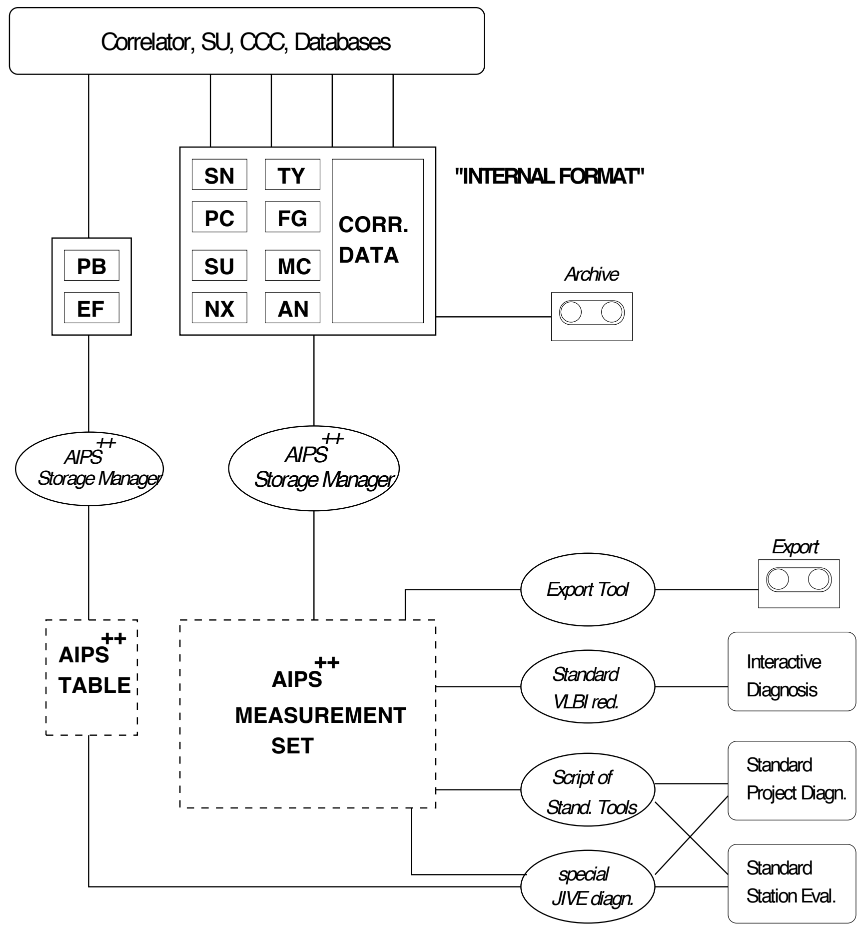

Fig. 2. Possible data structure for the JIVE/EVN correlator. The internal format contains all the data necessary to create distribution data. Besides correlator output there are other pieces of information required for processing and calibration. Examples of some of these are denoted in this figure by their "classic AIPS" table two letter codes: NX for indexing information, AN for antenna location and characteristics, PC for phase cal data, SN for amplitude calibration for example based on state counts, TY for system temperatures, FG for flagging, SU for source positions and MC for correlator model components. A separate route is possible for playback statistics and other measures of operational efficiencies. The data are made logically available as AIPS^{++} tables, allowing standard and specialized tools to deliver the desired functionality.

- *Data validation and feedback.* AIPS^{++} supports some form of nearly real–time data management, so it will be possible to display or process some of the data directly after it is put on disk. Already there are simple tools available to access and display almost anything from AIPS^{++} tables in Glish. Fringe fitting will have to be written for AIPS^{++}; work for this is planned to start soon. Standard scripts to display and plot quantities after correlation can easily be written in Glish and would run on E^3.

This could include tape statistics and other things which will not be a part of the distribution data eventually.

- *Calibration data.* The calibration data from the telescopes will have to be read in and attached to the data. One would imagine that filtering, sorting and attaching flagging and amplitude calibration data is a standard AIPS++ task. As before it depends on the AIPS++ and correlator time lines how much in house development is needed. It is not expected to be difficult to implement the interfaces for calibration data to the appropriate AIPS++ tables.
- *Archiving.* Archiving would be done in the internal format. To construct distribution data it would have to be read into AIPS++. The storage manager would in that case serve as an AIPS++ JIVE/EVN data reader. We would have to make sure that the internal data format includes all the necessary auxiliary data.
- *Implementation & Maintenance.* The AIPS++ route requires considerable JIVE resources in the implementation stage. Fortunately much of the know–how to start this work is available in Dwingeloo. Locally we would need to write the JIVE/EVN specific "storage manager" that makes the output data available to the AIPS++ table structure. Possibly this effort could be shared with the WSRT TMS as well. Special care is needed to make sure the necessary header and indexing data and auxiliary data is available in the internal correlator data, in order to make the structure qualify as a Measurement Set. We can then define special scripts to plot various diagnostic information in the process. In the long run we will benefit from the AIPS++ efforts directly, for instance when fringe fitting becomes available. In addition the package will provide programming infrastructure, user interfaces and maintenance as well as documentation tools. It would be desirable if the correlator would produce data in frequency space, rather than lag space and probably time–baseline–IF sorted would be most appropriate, although the storage manager could map to other structures. Separate storage managers for data streams with other data (e.g. read back statistics) will be implemented.

4.2.3. Could HOPS & AIPS++ co–exist?

With the different time scales on the availability of these two software packages, it is a legitimate question if the two could run at the same time. This would in our view only be interesting if this could be done without making an extra copy of the data.

The AIPS++ data structure allows virtually any data structure on disk to be mapped to an AIPS++ table. However, there will be a price in performance. First, the HOPS format has a specific sort order. Most often, this sorting will slow down the access of AIPS++ to the data (it will have to be un–sorted on the fly). A similar complication arises from the fact that HOPS requires the data to be stored in the lag–domain.

Hence, it is probably possible to make a HOPS data file visible to AIPS++. This requires a similar effort to that to make the internal format data available to AIPS++. But of course we still face the extra sorting and copying step to convert the data into the HOPS format.

4.3. *Time-line & Priorities*

Below we attempt to prioritize the software efforts that are supposedly to be carried out at JIVE. A very crude estimate, based on nothing but personal experience, is appended at the bottom. Such estimates have been wrong by large factors and require a detailed design plan and more detailed knowledge about AIPS^{++} to be accurate.

- Development in this area should start with a definition of the internal output format. Such a format definition includes the labeling and representation of correlator output, but also storage of the correlator job descriptor, auxiliary and calibration data, tape statistic etc.
- When such a description is available, work can start on developing both the AIPS^{++} storage manager and archive software. The AIPS^{++} part has highest priority, as it could be useful to have this part operational during the integration phase of the correlator.
- Of all the desired functionality, the ability to export data in (VLBI) UVFITS should have priority. This would allow us to export any data to existing software packages, where test data could be examined or first results could be processed.
- Together with archiving software, one can imagine an operational situation, albeit very inefficient. This should have the next highest priority.
- Making calibration data, specific to the JIVE/EVN correlator, available, we think this should be the next most important step. This requires not only software development but also engineering and scientific effort.
- Writing automated, almost real–time, diagnostic software, based on existing AIPS^{++} tools, would be the next important thing. This will allow the correlator to run with higher efficiency.
- Integration of all data from the correlator into proper AIPS^{++} data sets, to allow complete trouble shooting with the data on disk and the export of "calibrated" data.

We now estimate that work on the data format could start in early 1997. The first couple of months will be used to learn the necessary tools and to be introduced to AIPS^{++}. This learning stage could involve a pilot project, possibly reading preliminary (first fringes and further tests) correlator results into AIPS^{++}, by writing a data filler or by working on an administration system for archiving in AIPS^{++}. After that work could start on a preliminary version of the Storage Manager. Because a definition of the "internal format" is involved in this step we do not anticipate this work to be ready before the summer. One would hope that by October some real correlator data could be displayed with the AIPS^{++} β–release. By the time the correlator is supposed to be integrated, early 1998, we should have a way of exporting data. Work on an archiving mechanism is necessary before the correlator can be declared operational. Over 1998 we could continue to work on streamlining the operation and getting the calibration and diagnostic parts implemented.

These are very rough estimates, based on the availability of a single person. A sensible guess really requires knowledge of how much is in place with AIPS^{++} and experience implementing within the AIPS++ context. An understanding with NFRA/the AIPS++ management needs to be sought on the issues of shared effort and development support.

After the development phase we estimate the system will need continuous support by a considerable fraction of a single person, to maintain interfaces to the reduction packages, enhance the correlator efficiency, upgrade to larger data rates and support special observing modes.

5. Conclusions

We will adopt the following solutions:

- The data handler, D^3 collects output data and copies this to disk with very limited operations. It constructs index and header information for the data based on labels internal to the data and the job description it receives from C^3. This is part of the on–line system.
- A limited number of simple diagnostic tools run on C^3 and D^3 and are part of the on–line system.
- Calibration and other auxiliary data is collected on D^3 and needs to be archived together with the main data. This definition is a part of the on–line and off–line software interface.
- The data are archived in internal ("raw") format by D^3, mostly during night time or when correlator data rates are low.
- The development of an archive system should most logically be a joint effort with NFRA development for the WSRT.
- A third workstation (E^3) will run diagnostic software. This can be invoked both interactively or it will start automatically after completion of a part of the data.
- Distribution in user format will be provided by E^3, preferably on data still present on the D^3 disks, or data read back from archive.
- The data in internal format will be made available on E^3 (and D^3) through AIPS++.
- Development for diagnostic, calibration and distribution software will be done within the AIPS++ environment and needs to be synchronized with efforts within the AIPS++ consortium.
- A detailed description of the internal format is needed before any development can start.
- The highest priority items for software development are an AIPS++ Storage Manager for JIVE/EVN correlator output and UVFITS export.