Polar and other new modes of the GMRT Correlator

Jayaram N Chengalur

12/Dec/2000

1 Introduction

This writeup documents the polar and other new modes of the GMRT correlator. A block diagram of the GMRT correlator is given in Fig. 1.

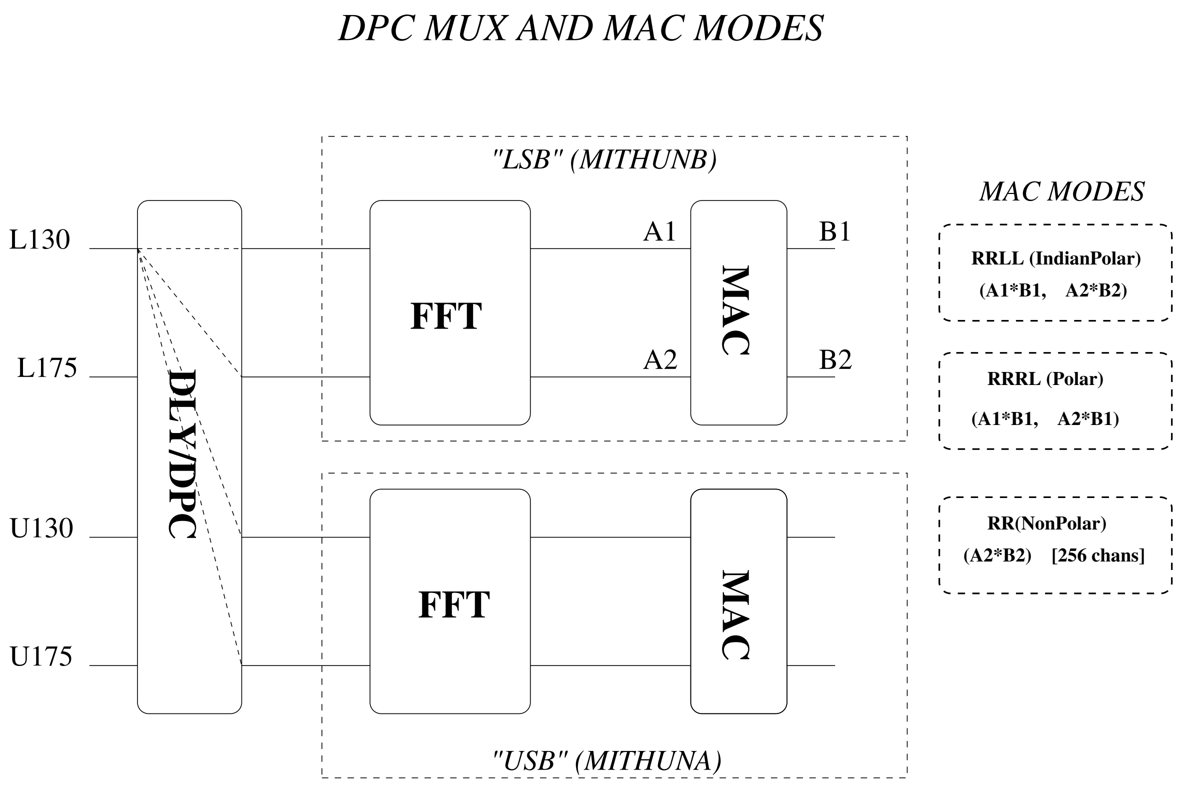

Figure 1: A block diagram of the GMRT correlator (excluding the samplers). The four signal outputs from a given GMRT antenna (see Sec 1 for details) are connected to the Delay/DPC block. After signal conditioning, these signals are sent to the FFT block. Any one of the input Delay/DPC signals can be sent to any one (or more) of the the FFT inputs. The FFT output is then sent to the Multiply and ACcumulate (MAC) block. Each MAC block has four inputs. As shown in the figure (and described in Sec 1) it can be configured into one of three different modes.

The GMRT antennas are dual polarized, so each antenna produces two output signals. These are conventionally called the "130" and the "175" polarizations[1] Each signal has a maximum bandwidth of 32 MHz. At the time of conversion to base-band, each of these signals is split into two sidebands, (the upper sideband "USB" and the lower sideband "LSB"), with each sideband having a maximum bandwidth of 16 MHz. Thus for each GMRT antenna, there are four input signals at the correlator. These signals are conventionally labelled as "USB-130" (U130), "USB-175" (U175), "LSB-130" (L130) and "USB-175" (L175). The GMRT correlator can process a maximum of 120 such 16 MHz wide inputs (i.e. corresponding to 30 antennas) in real time. As desgined the correlator consists of two esentially identical units, each capable of processing 60 (16 MHz wide) inputs. Conventionally, the USB signals from all antennas are connected to one unit and the LSB signals from all antennas are connected to the second unit. These units are hence often reffered to as the "USB" and the "LSB" halves of the correlator. Data from the "USB" half is acquired on a host called "corracqa", and then transferred to a server called "mithuna" for further processing and recording. Simialrly, data from the "LSB" half is acquired on a host called "corracqb", and then transferred to a server called "mithunb" for further processing and recording. Throughout this doccument, these two halves of the correlator will be called the "A" and "B" halves respectively. This is becuase (as will be clear shortly), the "USB" and "LSB" labels are not appropriate for most of the correlator configurations. For completeness, it should also be noted that there is a third host associated with the correlator, viz corrctl. This host is responsible for intial configuration and run time control of the correlator hardware. Finally, the user generally issues configuration commands from yet another host, which we will call the ONLINE host.

The block diagram in Fig. 1 starts after the sampler unit. After sampling, the signals go to a signal conditioning ("data preparation") block, called the DELAY/DPC. At the time of writing, some information on what happens in this block can be found in http://www.ncra.tifr.res.in/ sirothia/project/newdlydpc/NEWDLYDPC.htm. A slightly dated, but still relevant description can also be found in Roshi (1999). For the current purpose, the important aspect of the DELAY/DPC is that it allows for a flexible mapping between its input and output signals. Any given input signal can be sent to any one (or more) of the output ports. After signal conditioning, the signals are sent to the FFT block, where a 512 point (256 channel) complex FFT is done. After the FFT the signals are sent to the Multiply and ACcumulate (MAC) block. Each MAC block accepts four input signals. In Fig 1 they have been labelled as A1, A2, B1 and B2. Depending on how the MAC has been configured, it performs different operations on these inputs. There are three possible MAC configurations (or modes). In the RRLL mode (also called the "INDIAN POLAR" mode) the products formed in the MAC are A1*B1 and A2*B2. Adjacent frequency channels from the FFT are averaged together in this mode. One therefore has two output data streams, each of which has 128 complex channels. In the RRRL (or POLAR) mode, the products formed are A1*B1 and A2*B1[2]. Once again, adjacent frequency channels are

[1]So called because they are transported at IF frequencies of 130 and 175 MHz respectively.

[2]Given the current convention of sampler connections, a better label would have been RRLR.

averaged together in the MAC, so each output data stream has 128 complex spectral channels. The last mode is RR, in which only one product, viz. A2*B2 is produced. Adjacent frequency channels are not averaged together, and hence the output data stream has 256 complex spectral channels.

2 Correlator Configurations

2.1 The Indian Polar mode

In this mode the correlator gives four products for each pair of antennas viz. U130*U130, U175*U175, L130*L130 and L175*L175. There are 128 spectral channels for each of these products and the maximum bandwidth is 32 MHz. Since visibilities are measured for two orthogonal polarizations, this mode is appropriate when one is interested only in the total intensity (Stokes I) of the source. The DELAY/DPC and MAC have to be configured as shown in Fig. 2. As can be seen from the figure the "A" half of the correlator (see Sec 1) will produce USB visibilities while the "B" half will produce LSB visibilities.

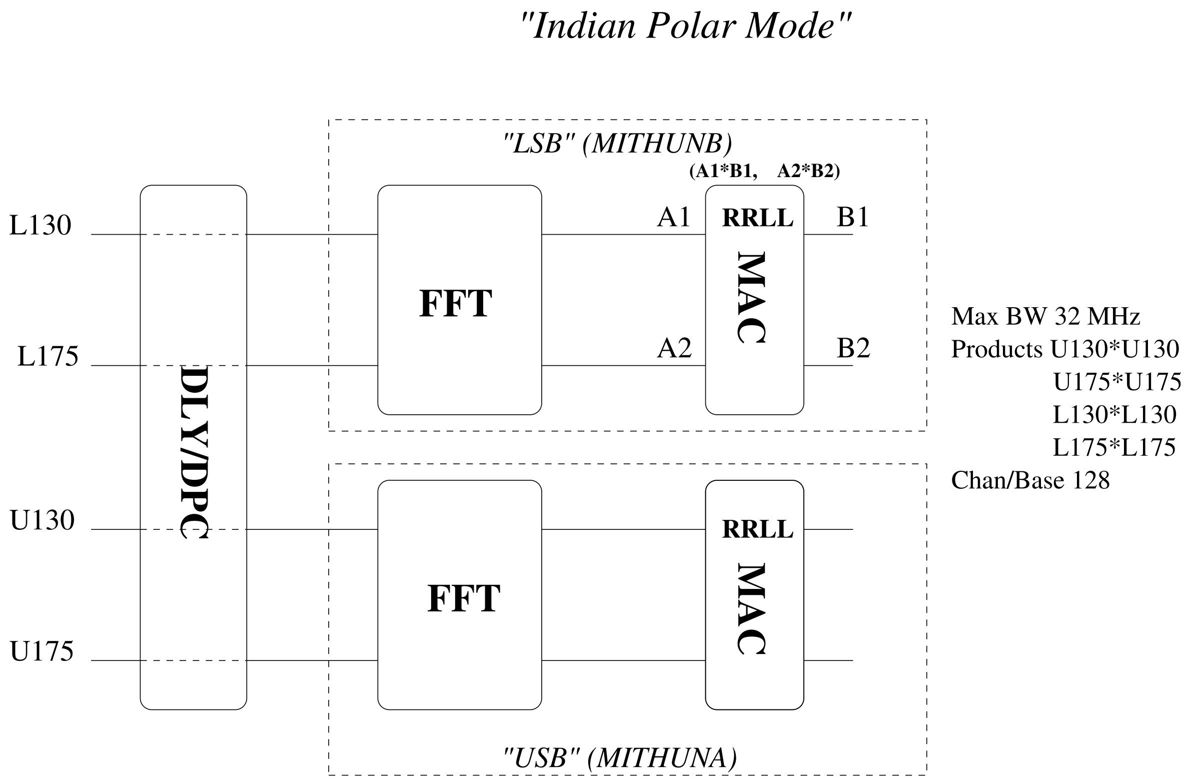

Figure 2: Schematic configuration for the Indian Polar Mode. See Sec. 2.1 for details.

2.2 The USB Polar mode

This modes forms all the products required for polar observations in the upper sideband. The correlator again gives four products for each pair of antennas, viz. U130*U130, U175*U130, U175*U175 and U130*U175. There are 128 spectral channels for each of these products and the maximum bandwidth is 16 MHz. This mode is suitable for observations where it is desired to measure all four stokes parameters.

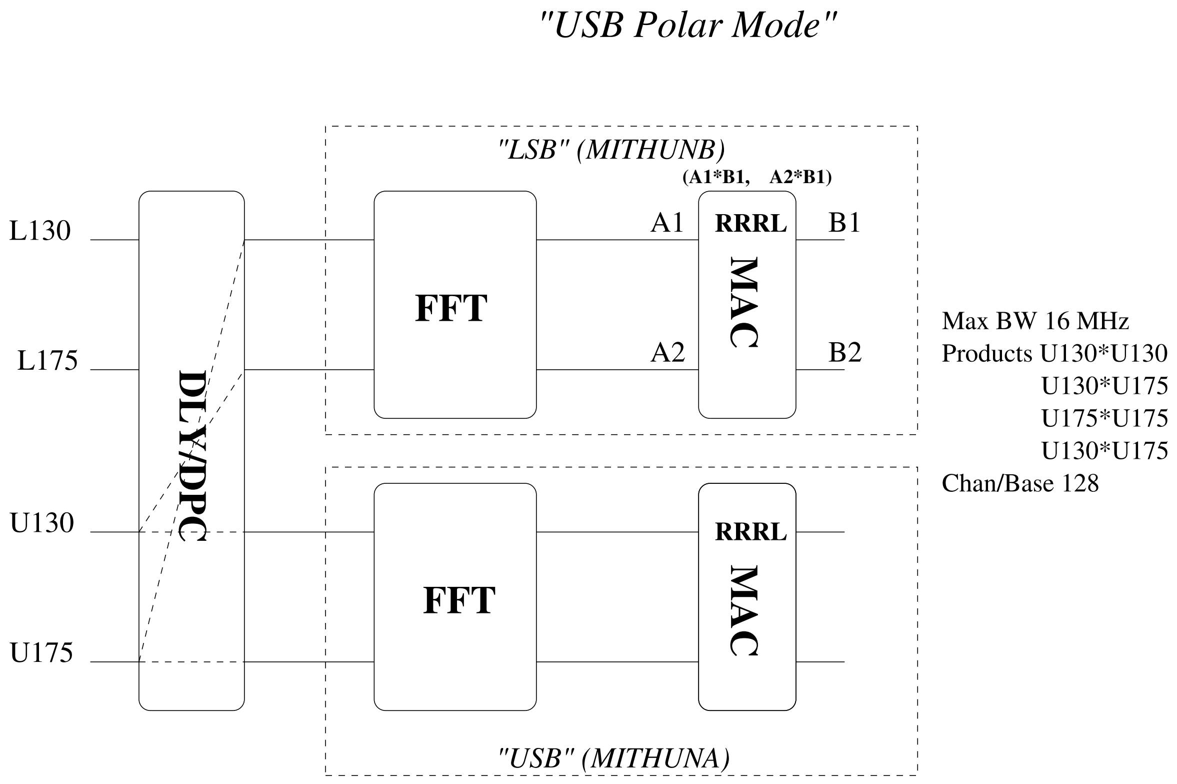

Figure 3: Schematic configuration for the USB Polar Mode. See Sec. 2.2 for details.

The DELAY/DPC and MAC configurations are shown schematically in Fig. 3. As can be seen, USB data from the baseband reciever system is sent to both halves of the correlator. The Delay/DPC is configured so that in one half of the correlator ("A") the data flow is as for the Indian Polar mode. For the other half ("B") the U130 data stream is redirected to the port where the L175 data used to go in the Indian Polar mode. Similarly the U175 data stream is redirected to where the L130 data used to go in the Indian Polar mode. The MAC is configured in the RRRL mode. The "A" half of the correlator produces U175*U175 and U130*U175 visibilities, while the "B" half produces U130*U130 and U175*U130 visibilities.

2.3 The USB High Resolution mode

In this mode the correlator gives only two products for each pair of antennas, viz. U130*U130 and U175*U175. There are 256 spectral channels for each of these products and the maximum bandwidth is 16 MHz. This mode is suitable for observations

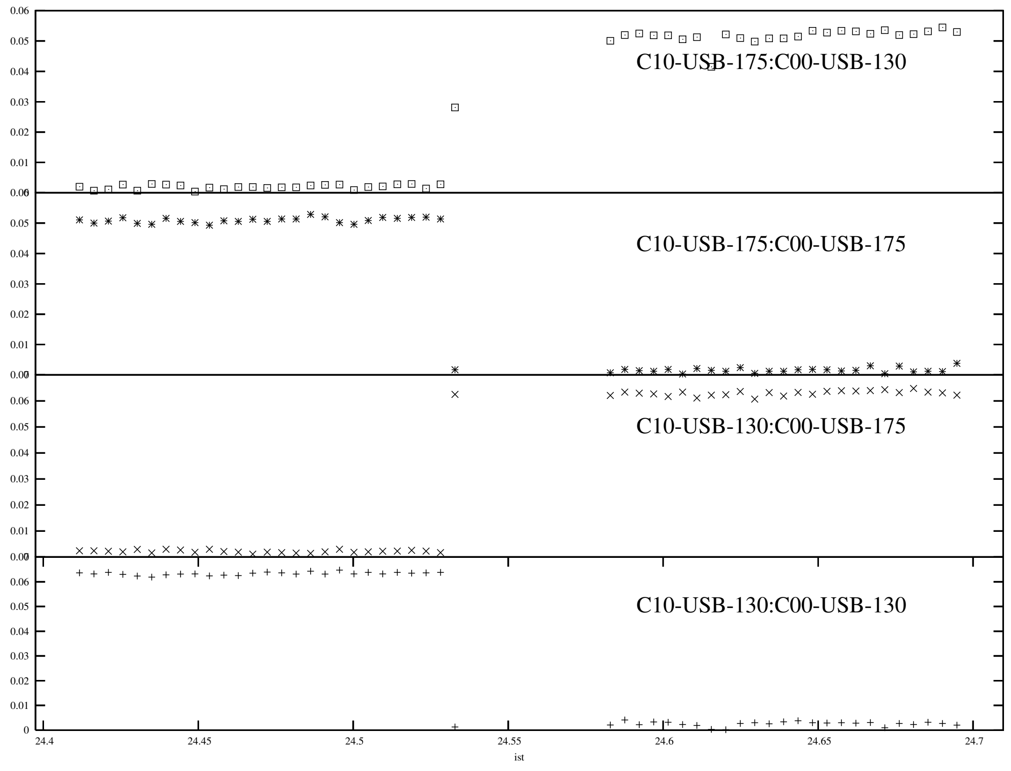

Figure 4: USB Polar mode calibrator data for visibilties between the antennas C00 and C10. The C10 antenna was given an RF swap midway through the observations. Only the amplitude of the visibilities is shown.

where the bandwidth of interest is 16 MHz or smaller (e.g. for spectral line observations, or continuum observations at frequencies where 32 MHz RF is not available) and only stokes I needs to be measured.

The DELAY/DPC and MAC configurations are shown schematically in Fig. 6. As can be seen, the Delay/DPC configuration in this mode is identical to that in the USB Polar mode. The MAC mode is however different, viz. RR. In this mode the "A" half of the correlator produces U175*U175 while the "B" half produces U130*U130 visibilities.

2.4 Other General Configurations

Three possible configurations were illustrated above, but there are in fact an immodest number of modes. For example, analogous to the USB Polar and USB HighRes modes one can image LSB Polar and LSB HighRes modes. One could also imagine modes where the same data is correlated one or more times, either for testing of the correlator, or for possible improvements in the signal to noise ratio. All combinations of DELAY/DPC and MAC configurations are supported by the current version of the DAS software. Details on how to actually configure and run these modes can be found in the following sections.

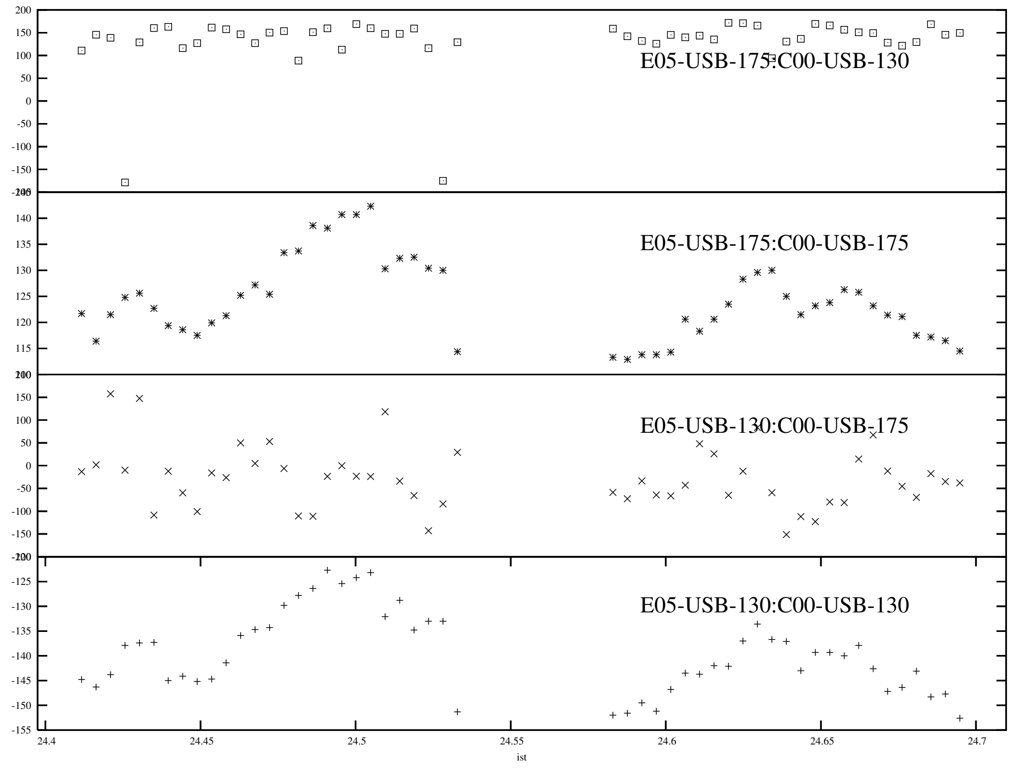

Figure 5: USB Polar mode data calibrator for visibilties between the antennas C00 and E05. Only the phase of the visibilities is shown.

3 Configuring and running different modes

In earlier DAS versions, the correlator was configured by editing (by hand) various configuration files. In this scheme one would have to consistently edit several configuration files on upto four different hosts to properly run these new modes. Given the very large number of possible configurations of the correlator, this scheme is no longer viable. The DAS programs[3] were hence modified so that all the user configuration information is contained in only one file, "corrsel.hdr". In addition to this, the DAS chain gets information on which antenna data stream is connected to which sampler via the "sampsys.hdr" file[4], while the information on the positions and instrumental delays of each antenna are contained in the "antsys.hdr" file[5]. These files are contained in the directory pointed to by the $SYS_DIR environmental variable. For

[3]It is assumed that readers are familiar with the DAS chain. Introductory material on the DAS chain can be found in Singh (1998); and detailed descriptions of an earlier version of the chain in Chengalur (2000).

[4]The contents of this file should be modified only by the members of the correlator group.

[5]This is the file that would need to be updated by, for e.g. **fdelay** when one wants to put in new instrumental delay values. Since correlator control happens only from the corrctl machine, the antsys.hdr file on this PC is the only one that affects the fringe stopping and delay tracking. However the antenna positions and delay values copied into the header of the LTA files are taken from the files in corracqa and corracqb. It is hence important to make sure that the files on all three hosts are identical

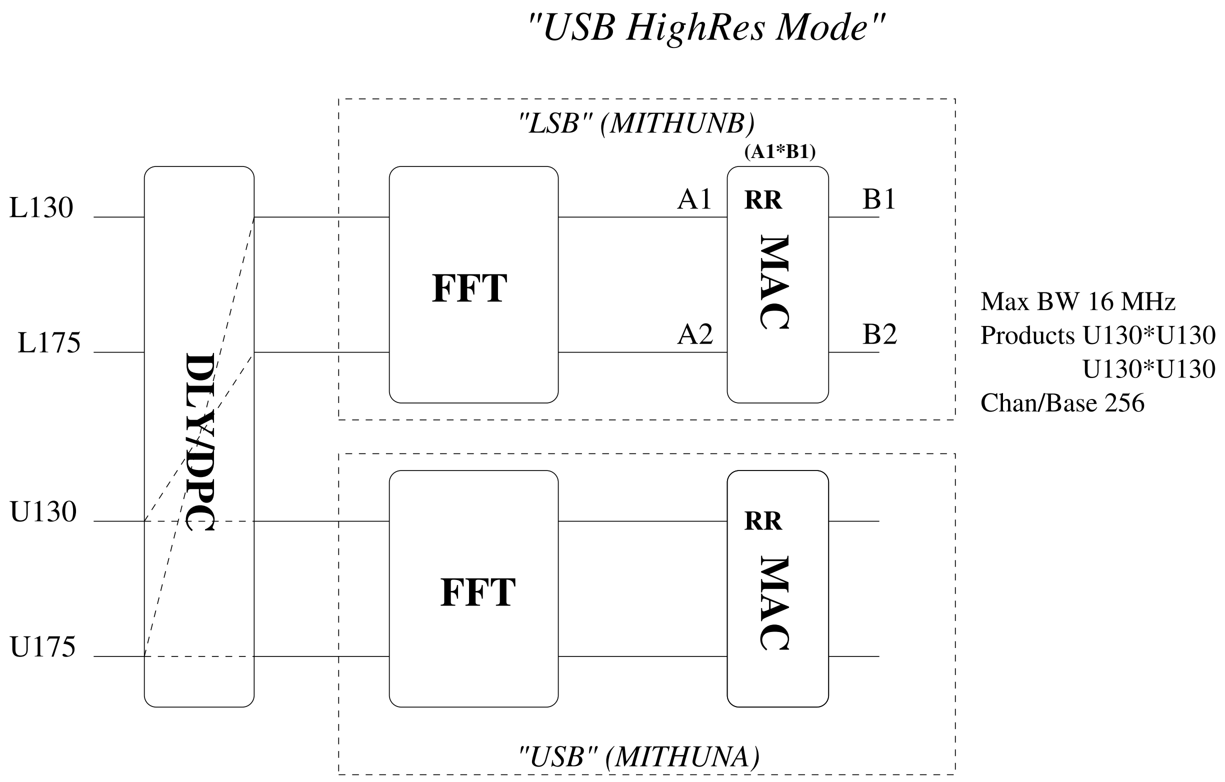

Figure 6: Schematic configuration for the USB High Resolution Mode. See Sec. 2.3 for details.

completeness, it is also worth noting that the "sampsys.hdr" file contains information on the connection of all 120 samplers. The contents of this file should be identical on all the three hosts. In the earlier scheme, the corrctl host had information of 120 samplers, while the corracqa and corracqb hosts had information on only 60 samplers each, and the number assigned to a given sampler differed on the different hosts.

The "corrsel.hdr" file resides on the ONLINE host (i.e. currently aditya or bhaskar). Configuration information from this file is consistently passed on to all relevant programs on all hosts. The file itself is also no longer meant to be edited by hand[6]. Instead the file is generated by a program which presents the user with a menu of possible configurations. This program is called **corrsel** .

The **corrsel** menu starts with the observing mode. At the moment ten observing modes are predefined. The predefined modes are 1. IndianPolar, 2. UsbPolar, 3. LsbPolar, 4. UsbHighRes, 5. LsbHighRes, 6. UsbCopy, 7. LsbCopy, 8. AllU130, 9. AllU175, 10. AllL130, and 11. AllL175

The names are hopefully self-explanatory. A given mode corresponds to some particular configuration of the DELAY/DPC and the MAC. For example, in the Indian Polar mode, the DELAY/DPC has a straight one to one mapping between its input and its output, and the MAC mode is RRLL. It should be noted that strictly

[6]Although since it is an ASCII file, one could if one wanted to. Details on what this file contains can be found further in this section.

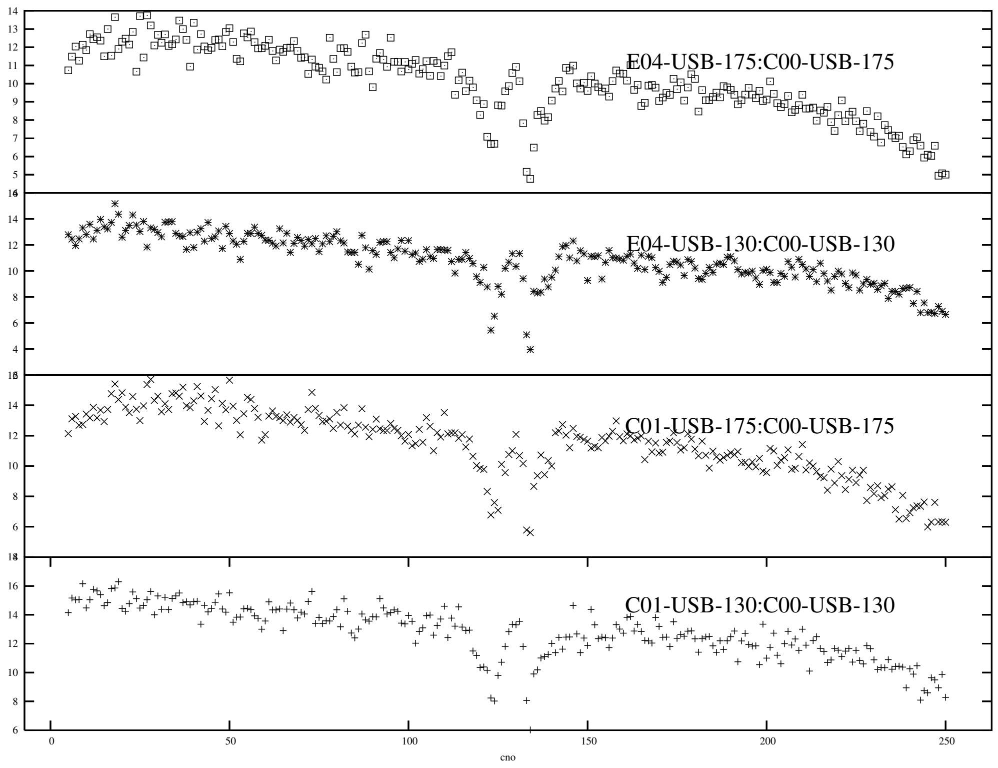

Figure 7: USBHighRes mode data spectra of galactic HI absorption towards 3C147.

speaking, the straight DPC map and the RRLL MAC mode *define* the IndianPolar mode. The actuall correlations that will be measured in this mode depend on the sampler connections in the "sampsys.hdr" file. Through out it has been assumed that the sampler connections are the default ones, viz. that SMP4n = L130, SMP4n+1 = L175, SMP4n+2 = U130 and SMP4n+3 = U175 (where n is an integer from 0 to 29). This means, that to preseve the meanings of these defenititions care must be taken while reconnecting the samplers. Any interchange of antennas must involve *all* samplers associated with those two antennas. Exchanging just the USB or LSB samplers would lead to inconsitencies.

The next menu item presented by **corrsel** is LTA1. This is the integration to be done by **acq** 30. The value is in units of STA cycles (i.e. 128 ms). The typical value used at the GMRT is 16, which corresponds to $\sim$ 2 sec. Running of the entire DAS chain at the full time resolution (i.e. 128ms; LTA1=1) is supported by the current version of the DAS software. Users should be aware however that only limited testing of this mode has been done so far.

The next item on the menu is the bandwidth, this can be any value from 16 MHz to 0.125 MHz in steps of 2. This is the bandwidth that each half of the correlator is set to. Hence, in the IndianPolar mode the total bandwidth is twice the value set here. In the Polar modes the total bandwidth is the same as the value set here.

The final item on the menu is the spectral channels for which data is to be acquired. **corrsel** produces an output file called "corrsel.hdr", a sample of which shown below.

```
{ Corrsel.def
LTA     = 16
CLK_SEL = 0           /* bandwidth = 16 MHz */
CHAN_NUM= 0:127:1
MAC_MODE= RRLL        /* legal values:  RRLL RRRL RR */
DPC_MUX = IndianPolar
MODE    = 0
FFT_MODE= 0           /* fft_size = 512/2^ fft_mode*/
} Corrsel
*
```

Most of the items in this file should be self explanatory. The MODE entry specifies which kind of fringe stopping is to be done (i.e. whether this is for e.g. a TransitObs or not). The DPC_MUX specifies the mapping between the input and output of the DPC. The following special names are recognized here: IndianPolar, UsbPolar, LsbPolar, UsbCopy, LsbCopy, AllU130, AllU175, AllL130, AllL175[7] In addition, strings of the form "dpc_abcd" are recognized. A DPC_MUX value of dpc_abcd means that input channel a is mapped to output channel 0, input channel b is mapped to output channel 1 and so on.

The "corrsel.hdr" file is generated on the ONLINE machine. Information in this file is read by the program **dassrv** and then passed on to **sockcmd** , **acq** 30 etc. on all relevant hosts. This ensures that information on the selected configuration is properly passed on to the entire DAS chain. In addition one has to ensure that the correlator hardware itself is also configured as specified in the corrsel.hdr file. The program **set_corr** (which is run on the ONLINE host) does this. **set_corr** reads the corrsel.hdr file, and then passes the configuration information on to the correlator configuration programs. **set_corr** takes the following options

```
set_corr --subsys SubSystem [--host CorrHostName][--file ConfigFile]
[--version Version]
Where SubSystem is one of:
delay_full (Complete Delay Config)
delay (Set DpcMode & ClkSel)
fft_mac (Complete Corr Config)
mac (Config Mac Mode)
pulsar (Corr Config for Pular Mode)
and Version is one of:  new old tst dvl
```

set_corr is a fairly simple program; all it does is parse the corrsel.hdr file and then run the appropriate program on the corrctl host using ssh. The actual configuration programs being run on corrctl are **corr_config** , **corr_configa newdly_config** and **dlyconfig** . Of these the first three are pretty much as provided by Sandeep Sirothia. **corr_config** configures the FFT and MAC blocks. **corr_configa** configures the FFT as appropriate for pulsar observations. **newdly_config** does a complete initialization

[7] In addition, for sentimental reasons, strings of the sort "arar_arar", "aral_brbl" are also recognized; but this is really very arcane and probably of no interest to anyone currently at NCRA.

of the DELAY/DPC block. **dlyconfig** on the other hand, only sets the DPC mode and the CLK_SEL (i.e. the desampling value or bandwidth). All of these programs can also be run from the shell prompt on corrctl. **dlyconfig** understands all of the special modes described above (e.g. IndianPolar, AllL130...) as well as strings of the form dpc_abcd. The mode is specified to **dlyconfig** using the -m option, and the CLK_SEL using the -c option. When configuring the correlator from the shell prompt, the onus lies on the user to ensure that the DAS chain configuration corresponds to the actual hardware configuration. While this is an extra burden, it may also be convinient for users who wish to spoof the system in some way.

For ease of use, the **set_corr** program is bundled into a shell script called **corr** . This shell script takes one argument, which has to be one of init,reconf, or pulsar. **corr** init causes a full initialization of the correlator. This intialization puts the correlator into the IndianPolar mode with a bandwidth of 16 MHz irrespective of the contents of the corrsel.hdr file. **corr** reconf reconfigures the corrrelator as specified in the corrsel.hdr file. And finally **corr** pulsar configures the correlator as appropriate for pulsar observations[8].

An SOP for running the DAS programs can be found in Appendix A. The procedure is very similar to the earlier one, except that all configuration information is passed to the programs automatically, hence there are no run time configuration arguments that have to be given. Since the correlator configuration is common for all correlator control cards, all subarrays will have the same observing mode. It is not possible to simultaneously have one subarray in the IndianPolar mode and another in the Polar mode.

As discussed in Sec 4 , the data format, and in particular the corr structre has changed. The changes in the corr structure mean that the shared memories used to communicate between programs are different in the current DAS version as compared to the older DAS versions. This means that many of the old data monitoring programs cannot be used. However, Manish Galande's DAS Console tool, which supplants all of these programs, is available. Further, as discussed in more detail below, in many of the modes the "normalization" option is not appropriate because the required normalization data is not available on the given host.

The size of the shared memory used for data transfer between the **acq** and **acq** 30 programs has also been changed. This, along with hardware upgradation of the hosts on which these programs run allows for running the entire DAS chain at the full (128 ms) time resolution. All programs which use this shared memory will hence have to be recompiled using the new version of the header file acqbuf.h

4 Disk Files in the new correlator modes

The correlator output is finally recorded into disk in a format called the "LTA" format (see Chengalur (2002)) for more details. As described in Sec. 1, correlator data is acquired in two different paths, viz. "A" (hosts: corracqa, mithuna) and "B" (hosts: corracqb, and mithunb). Each of these paths are independent (but time synchronized,

[8]In practise this means calling the program **corr_configa** instead of **corr_config** on corrctl.

see Sec 5). At the end of the observing session, the user hence has two disk files in which the data resides. Traditionally these are called the “lta” and “ltb” files. In the IndianPolar mode each of these files can be regarded as complete in itself. In most other modes one would require visibilities from both the lta and ltb files to properly interpret the data. Before describing the procedure for combining lta and ltb files the changes required in the LTA format in order to run these new correlator modes are described.

Two esential changes were to increase the maximum number of samplers from 60 to 120, and the maximum number of spectral channels from 128 to 256. This has cascading implications for the “corr” structure of the binary header. In addition, the maximum number of baselines that an LTA file can contain had to be increased from 930 to 1860. In the ASCII header therefore, the format of the baseline specification keyword changed from, for e.g. BAS003 to BAS0003. While this has no implications for the LUTE programs, the xtract suite of programs are unable to handle this change. The ASCII header also contains a keyword called SAMP_NUM. The value is a list of samplers whose data is contained in the file. Since LTA header records are limited to 80 characters in length, the SAMP_NUM keyword often overflows. This overflow also occured sporadically in the past when using sub-arrays consisting of a random subset of antennas. This keyword is actually redundant, and its use is depcracated. However since the xtract suite of programs uses it, a work around as been implemented. The SAMP_NUM keyword is now supplemented by the keywords SMP_NUM1, SMP_NUM2... SMP_NUM9 keywords. The xtract routines have been modified so that they can read both BASNNN and BASNNNN type keywords (i.e. they work on either new or old LTA files), and are also sensitive to the extended SAMP_NUM keywords. Finally, for completenes, it is noted here that (i) the timestamp in the LTA file now strictly corresponds to the first STA cycle that was folded into the given record, and (ii) the DAS version and LTA file format version are correctly recorded in the LTA file’s version keyword. The current LTA file version is 1.10.

In this version of DAS, the labeling of the samplers and data streams is consistently done. That is a sampler labelled as e.g. 30 corresponds to the physical sampler numer 30, and a data stream labelled as e.g. C00-USB-130 corresponds to the USB-130 data stream of antenna C00. This is true for all possible combinations of DELAY/DPC and MAC modes. The fact that the sampler numbers in an LTA file are now the same as the physical sampler numbers (as opposed to the earlier scheme, where the physical sampler numbers were remapped in peculiar ways for the two data paths) has implications for the phased array software. This software reads the output of **rantsol** and applies the appropriate phase correction to each data stream. For backwards compatibility, **rantsol** was modified to relabel the samplers identically to the old remapping scheme, i.e. where the LTA file sampler numbers do not correspond to the physical sampler numbers.

The data in the lta and ltb files can be merged together using the LUTE utility **ltamerge** . This utility can merge any two files that correspond to the same observation. All possibe observing modes are handled. **ltamerge** does some sanity checks on the header before proceeding. It also checks the data timestamps. In general (see Sec. 5 time stamps of corresponding records should either match to withun $\sim 3-4\times10^{-4}$ seconds, or should differ by an integral multiple of the integration time. **ltamerge** will use the timestamp information to make sure that data that is merged together correspond to the same timestamp. Data for timestamps that are present in only one file are ignored (in this case a warning is issued to the user). In addition, in case the time synchronization lies outside the acceptable range, a warning is issued to the user. Please also note that initial synchronization of the PC clocks to the master time server is a slow process, that could take upto a few hours.

Files that have been merged using **ltamerge** can be converted to FITS using **gvfits** [9]. **gvfits** will properly understand the observing mode and create the appropriate FITS file (i.e. with the data in separate STOKES, for Polar or HighRes modes or separate IFs for the IndianPolar mode. Modes in which the same data stream appears more than once are not handled.) At the GMRT, the L band feed is linearly polarized while all other feeds are circularly polarized. Since most of the data analysis is done in AIPS, which understands only circular polarization, by default **gvfits** will label even L band data as circularly polarized. However, there is an option to force linear polarization labels. For Polar mode data, it would be appropriate to not spoof the polar labels.

5 Time and Synchronization.

As mentioned in Sec 1 the correlator has three hosts associated with it, viz. corracqa, corracqb, and corrctl. The first two of these hosts acquire data from the MAC (via an addon PCI card). The function of the third host, corrctl, is to control the correlator hardware. There are two kinds of control that are of relevance, 1) configuring the correlator (which was discussed in Sec 3 above) and 2) setting the fringe stop and delay track parameters. Since these parameters are time variable, it is essential for proper functioning of the correlator that the clocks on these different hosts be synchronizd. This is currently acheived using the Linux ntp utility[10]. Further, as discussed above, GMRT data is generally converted into random group FITS using **gvfits** and then analyzed in AIPS. In this FITS format, it is assumed that all the different IF or STOKES data corresponding to a given integration have the same timestamp. Since the correlated data is split over the two halves of the correlator, and are acquired and recorded by two different hosts. It is hence essential to make sure that the data acqusition on the "A" and "B" paths are synchronized. Prior to the enabling of these new modes, the guiding philosphy in the DAS chain was to minimize data loss. This meant that even if data acquisition on the two halves of the correlator were started in sync, a glitch which occured only on one half of the correlator could cause synchronization to be lost. Since glitches are fairly common, the time difference between corresponding records in the two halves of the data acquisition chain could vary substantially through the observations. An example of this is shown in Fig. 8 In the Indian Polar mode, where data from the two halves were converted

[9] version 1.11 and higher

[10] ntp configuration on the correlator hosts was first done by Sandeep Sirothia

into two different FITS files and analyzed separately, this lack of synchronization was irrelevant.

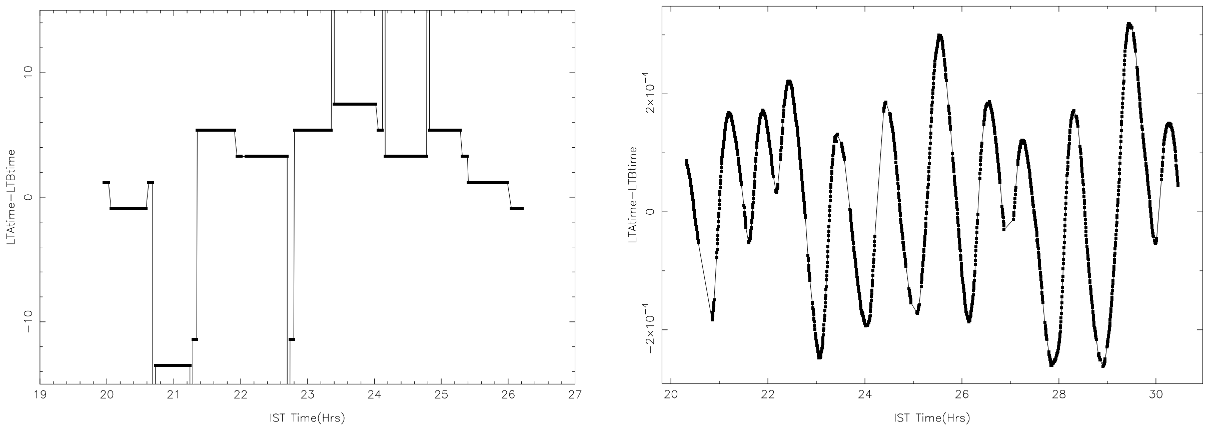

Figure 8: **[Left]** Difference between the timestamps (in seconds) of corresponding records acquired by the “A” and “B” chains of the previous version of the Data Acquisition System. The data is taken from the 01/Oct/05 observations of the project 08skc01. **[Right]** Difference between the timestamps (in seconds) of corresponding records acquired by the “A” and “B” chains of the current version of the Data Acquisition System. The data is taken from the 06/Dec/05 observations of the project 09gka01.

In the Polar modes of the correlator however, it would be extremely inconvinient to have different timestamps on data acquired by the two halves of the correlator. The entire DAS chain was hence reworked. In the new scheme keeping synchronization between the two chains has the highest priority and some amount of data loss in each chain is permitted. A rough overview of how synchronization is achieved is as follows. The **acq** program assigns a number to each STA block that it recieves from the correlator. STA blocks are recieved once every STA cycle (128 ms), and since the same STA interrupt is given to both halves of the correlator, in principle the blocks arriving in corracqa and corracqb should be in sync. However glitches in any one pipeline could cause a block to be missing. The **acq** program assigns block numbers to the STA data based on the current time, and is hence sensitive to these missing blocks. A missing block would result in a gap in the STA block sequence. In a similar way, all the down stream programs can keep track of the fidelity of the data chain, and take appropriate action when a data block is lost. At each stage in which integration of the data is done, it is ensured that the integration cycles have no phase jumps. For example, if one is doing an M point integration, and the first integration started at datablock M*n+P, (where n is an integer) it is ensured that each successive integration starts at datablock M*m+P, (with integer m). Whenever datablocks are renumbered, it is ensured that the renumbering uses the time information to proprely account for any missing blocks. Finally, the **acq** programs on the different hosts are started in sync[11] This means that at the end of the data chain, i.e. in the recorded LTA and LTB files, corresponding records should either have the same timestamp (within the accuracy with which time is being kept on the different hosts) or should have timestamps that differ by an integer multiple of the integration time. This synchronization scheme has been implemented and tested and appears to be fairly robust. An example of the difference in the timestamps between data acquired by the two pipelines by the new DAS software is shown in Fig. 8. As can be seen, the time is now synchronized to typically $3 - 4 \times 10^{-4}$ seconds or better.

It is still possible that some (hopefully rare!) glitches cause the DAS chains to go out of sync. This is most likely to happen when the MAC cards are reconfigured while data acquisition is going on. Reconfiguring the MAC card causes very large bursts of spurious data to flow into the DAS chain. Operators should hence ideally halt the DAS chain if they wish to reset the MAC, or should carefully monitor the time synchronization immediately after reconfiguring the MAC. Loss of time synchronization is generally readily apparaent by looking at the timestamps reported by **collect** or **record** running on the two pipelines. In addition the utility **ltatime** can be used to report the time difference between corresponding records in the LTA and LTB files.

6 Useful Routines

The LUTE programs **ltahdr** , **ltatime** , **ltamerge** and **ltaprint** all recognize these new correlator modes. **ltahdr** reports the observing mode and other useful information regarding a given LTA file. **ltatime** reports the time difference between corresponding records in lta and ltb files. **ltamerge** will merge together files corresponding to a given observation. All possible correlator modes can be merged using **ltamerge** . Finally **ltaprint** will print out the actual visibility data in a given LTA file. All LUTE programs will work on files produced by **ltamerge** , as well as the raw lta/b files produced by **record** . **ltaprint** has a host of options, some of which are particularly useful for polar mode data.

7 References

1. 'Subarray Operation at the GMRT', Jayaram N. Chengalur, NCRA Technical Report, Dec 2000.

2. 'The GMRT LTA format', Jayaram N. Chengalur, NCRA Techincal Report, Feb 2002.

3. 'The GMRT Correlator', Anish Roshi, in 'Low Frequency Radio Astronomy, Jayaram N. Chengalur, Y. Gupta, & K.S. Dwarakanath, eds. (1999).

[11] In sync means within one 128 ms STA cycle. This can demonstrably be done by hand. However, it can now be automatically done by Manish Galande's DASControl tool.

4. ‘The Data Acquisition System for the GMRT’, R. K. Singh, in ’Low Frequency Radio Astronomy, Jayaram N. Chengalur, Y. Gupta, & K.S. Dwarakanath, eds. (1999).

8 Appendix A: Running the Correlator

```
1.  log in to corracqa, corracqb, corrctl as observer, chose the "d"
   ("dvl") when prompted for which software system to use.
   (This sets the paths and environment variables appropriate for the
   new "development" software.  NB: if there are any shared memories
   floating around from the older das versions, you will need to
   delete them).
   2.  login in to mithuna and mithunb as observer and chose "d" when
   promoted for which software system to use.
   (Again, this sets the paths and environment variables appropriate for
   the new "development" software.  NB: if there are any shared memories
   floating around from the older das versions, you will need to
   delete them).
   3.  login into aditya as observer, and cd to /home/observer/dassrv-dvl
   (The old area, from which dassrv used to be run from 17/Nov/05 to
   08/Dec/05 has been renamed dassrv-tst4).
   4.  If you want to initialize the correlator, run
   corr init
   on *aditya*
   (This will run corr_config and newdly_config on the corrctl PC).
   5.  Choose the observing setup by running
   corrsel
   on *aditya*
   (This will setup the file /temp2/data/corrsel.hdr, which contains
   information on how you want the correlator configured.  corrsel
   allows a menu based selection of the observing mode, integration time,
   clksel etc.
   At the moment, the following modes are predefined in corrsel:
   0 IndianPolar
   1 UsbPolar
   2 LsbPolar
   3 UsbHighRes
   4 LsbHighRes
   5 UsbCopy
   6 LsbCopy
   7 AllU130
   8 AllU175
   9 AllL130
   10 AllL175
```

The names are hopefully self explanatory.
In addition, you can configure a vast number of modes by editing the
DPC_MUX and
MAC_MODE fields in the corrsel.hdr file.
The valid MAC_MODE values are RRLL RRRL and RR
The valid DPC_MUX values are any of the strings:

```
IndianPolar, UsbPolar, LsbPolar, UsbCopy, LsbCopy,    /* new modes */
AllU130, AllU175, AllL130, AllL175,
arar_arar, alal_alal, brbr_brbr, blbl_blbl,           /* historical */
aral_brbl, aral_alar, brbl_blbr, arbr_albl,
```

Again the names should be self explanatory. In addition, the mode can
be
a string of the form dpc_abcd, where 0<= a,b,c,d <=3. A mode of the
form
dpc_abcd means that the dpc mux sends channel a data to where channel
0
data normally goes, channel b data to where channel 1 data normally goes
and so on.)
NB: In the UsbHighRes and LsbHighRes modes the DPC_MUX is set to UsbPolar
and LsbPolar respectively. However the MAC_MODE will defer for the
HighRes and Polar modes.
6. reconfigure the correlator to the selected mode. To do this type
corr reconf
on *aditya*. PLEASE NOTE THAT IF YOU ARE CHANGING THE MODE, IT IS
NECCESSARY TO DO A hltndasc AND RESTART acq30 ETC. (IT MAY BE
SAFEST TO KILL AND RESTART ALL THE acq PROGRAMMES. NOT KILLING acq
MAY AFFECT SYNCHRONIZATION OF THE TWO PIPELINES. MOST OF THE TIME
IT SHOULD BE FINE TO KEEP acq RUNNING, AND JUST CHECK FOR
SYNCHRONIZATION PROBLEMS AFTER YOU RESTART. IF THINGS ARE SYNCHRONIZED
THEN FINE, OTHERWISE YOU WILL NEED TO KILL AND RESTART acq).
7. start acq, acq30, sockcmd fstop, dlytrk,dassrv,collect on the appropriate
machines. Note that none of the programs need command line options,
but
instead, as explained in the introduction, will automatically understand
the selected observing mode.
NB: The acqs on corracqa/corracqb/corrctl should be started within
1 STA cycle(i.e. 128 ms!) of each other. The easiest way to do this
is:
(a) export the corracqa/b/ctl window to all desktops.
(b) go to any one desktop and align the acq windows one
below the other
(c) type "acq" in all windows (but don't type ENTER!)
(d) hit the ENTER button in all windows in quick succession.
I agree that this is a pretty silly way to do things. A more automatic
method is in the prorverbial pipeline.

8. Ask the operator to give an init and add a project from the master terminal.

The bandmask while initing and adding a project should be correspond to the chosen mode.

NB: The bandmask is a bit mask with bits in the order USB130 USB 175 LSB130 LSB175

For a pure USB observation the bandmask is 3 (e.g. UsbPolar, UsbCopy, AllU130..)

For a pure LSB observation the bandmask is 12 (e.g. LsbPolar, LsbCopy, AllL130..)

For a dual sideband observation the bandmask is 15 (e.g. IndianPolar).

The actual bandmask for each mode is tabulated below:

Mode Name	Band Mask
IndianPolar	15
UsbPolar	3
LsbPolar	12
UsbHighRes	3
LsbHighRes	12
UsbCopy	3
LsbCopy	12
AllU130	1
AllU175	2
AllL130	4
AllL175	8

NB: All of these modes use both halves of the correlator, so will use CMODE 3 in ONLINE.

9. start and stop scans, record as usual. Median filter can be used. as always it is upto the user to decide if s/he wants median filter or not. The maximum filter length is 128. RECORD NOW TAKES AN ADDITONAL OPTION "SELF". IF THE LAST [OPTIONAL] ARGUMENT IS "SELF" THEN RECORD WILL PUT ONLY THE SELF CORRELATIONS IN THE LTA FILE. For example.

will create a file test.lta with only self correlations in it.

10. The only way to look at the data in real time is Manish's dasmon program. The version you get by default works for the "dvl" mode. If you do ssh -X dasmon mithuna/b (password: dasmon) you will again get the correct dasmon for the "dvl" mode.

NB: In the new modes the data is divided between the two halves of the
correlator.
For example in the UsbPolar mode U130xU130 and U175xU130 come in mithunb
(i.e. the "ltb" file) while U175xU175 and U130xU175 come in mithuna
(i.e.the
"lta" file). Hence
(a) dasmon in mithuna or mithunb will only show you one half of the data
(b) "normalization" cannot be applied in some modes. To normalize U175xU130
one needs the U130 and U175 selfs, but both will be available only when
one merges the data streams.
11. ltahdr and other lute programs are backwards compatable, as are
xtract, tax and
rantsol. For looking at the polar mode data, there is a special program
taxx.
For the IndianPolar and HighRes modes tax will work.
13. you can merge the lta and ltb files using ltamerge. This merged
file will have the
complete data, and is easier for playing around with taxx.
14. the merged file can be converted to FITS using listscan and gvfits.
These programs
are also backwards compatable. The individual lta/b files can also be
converted
using these programs, but I don't seem to be able to get AIPS to understand
the
polarizations in a file with only RR RL or only LL LR. I think that only
some
partitioning produced by the VLA correlator is understood. The combined
RR RL LR LL
FITS file is interpreted without problem.