NOTE 254 – Multi-IF, FEED, BANK Implementation for the GBT

J. McMullin, R. Garwood, J. Braatz

17 January 2003

Contents

1 Compiled Targets

- GB M&C – provide information on numWindows, restFrequencies(numWindows), and offsets(numWindows)
- GB M&C – Permit a single integration time for the different banks of the Spectrometer
- AIPS++ – DOPPLER table filled by gbtmsfiller

  Enhancement: AOCso04231
- AIPS++ – SYSVEL in SOURCE table filled

  Enhancement: AOCso04231
- AIPS++ – RESTFRQS vector filled (once available from M&C)

  Enhancement: AOCso04232
- AIPS++ – Add sditerator function to switch between RESTFRQS

  Enhancement: AOCso04233
- AIPS++ – NUM_LINES in SOURCE table filled (once available from M&C)

  Enhancement: AOCso04232
- AIPS++ – Add REST frame option on plotter.

  Enhancement: AOCso04235
- AIPS++ – Revise center channel calculation (obtain from IF FITS)

  Enhancement: AOCso04234
- AIPS++ – Enable calibration of multi-beam procedures (NOD)

  Target: 153 beam switched calibration refinements

  Target: 119.1 msconcat: works for all subtables
- AIPS++ – Enable use of the FEED information in data access

  Target: 111.15 gbtmsfiller: Support BS data

  Target: 111.16 gbtmsfiller: Support multi-beam data

  Target: 153 beam switched calibration refinements

- AIPS++ – Enable use of the BEAM_OFFSETs for imaging

  Target: 172 Multi-beam Imaging

- AIPS++ – Enable dual beam-switched imaging

  No Targets set.

- AIPS++ – MSconcat procedure

  Target: 119.1 MSCONCAT works for all subtables

- AIPS++ – Enhance calibration routines to operate on multi-bank data.

  No targets set.

2 Multi-IF

2.1 Introduction

This chapter provides brief guidelines on how multi-frequency observing at the GBT will be enabled. Specifically, this note addresses the situations described in the Maddalena Memo ("Requirements for Multifrequency Observing with the GBT"). In the first section, I review the observing scenarios posed and the associated information which will be available in FITS files. The association with MeasurementSet v2.0 keywords and subtables is made and the implementation of tools for virtual and permanent alterations to the observed frequency scale are listed.

2.2 Observing Scenarios

- SFSW: Single frequency observing in a single spectral window: The observer will want to observe a single spectral line transition of a molecule or atom or a band centered at a single frequency for continuum observing. An example is observing galactic atomic hydrogen with a bandpass sufficiently wide to encompass all of the emission.
- MFSW: Multiple frequencies in a single spectral window: The observer will want a single spectral-line bandpass to encompass two or more transitions from different atoms or molecules or different transtions from the same molecule. This is a spectral-line observing mode only. Two examples are: 1) Observing the 1665 and 1667 MHz lines of OH in a 2.5 MHz or wider bandpass; 2) Observing the 4.8 GHz H_2CO and 5.3 GHz CH_2NH with an 800 MHz bandpass.
- MFMW: Multiple frequencies in multiple spectral windows:
  1. MFMW-1: Multiple frequency spectral-line observing with each spectral window covering a single transition or continuum observing at multiple frequencies. This is an extension of SFSW observing. An example is observing the 1612 and 1720 MHz lines of OH, each in a 5 MHz bandpass. Or, continuum observing at 8.4 and 10 GHz with a bandwidth of 200 MHz.
  2. MFMW-2: Multiple frequency observing with each spectral window covering multiple transitions. This is an extension of MFSW.
  3. MFMW-3: Multiple frequency observing with some spectral windows covering a single transition while others cover multiple transitions. This is a combination of SFSW and MFMW.

- SFMW: Single frequency in multiple spectral windows: Observers will pick this kind of observing for two common reasons:

  1. when they know the rest frequency of the atom or molecule but don't know the velocity of the object along the line of sight. An example of this kind of observing would be an attempt to discover galaxies in the zone of avoidance from their HI emission.
  2. when the expected line width is broader than the maximum bandwidth of the backend. Here, the observer is trying to synthesize a broad bandpass from narrow ones.

2.3 FITS to MeasurementSet v2.0 mapping

The Maddelena Memo describes the required inputs for configuring the hardware and observing. The relevant parameters available in the FITS files are:

- numWindows
- restFrequencies(numWindows); the first is considered the master for LO1 Doppler tracking.
- offsets(numWindows)
- velDefinition
- restFrame
- velocity
- xpos, ypos
- coordMode
- epoch

These parameters map to the following locations in the MeasurementSet.

- **numWindows**

  MS SUBTABLE:

  DATA_DESCRIPTION SPECTRAL_WINDOW:

  PARAMETER:

  DATA_DESC_ID-¿SPECTRAL_WINDOW_ID Int

Based on spectral tolerance, each window has a designated spectral window ID. This is mapped by the DATA_DESC_ID in the MAIN table to the DATA_DESCRIPTION table which will list the appropriate spectral window IDs for a given scan/observation. Information on each spectral window ID is found in detail in the SPECTRAL_WINDOW subtable.

- **restFrequencies(numWindows)**

  MS SUBTABLE:

  SOURCE

  PARAMETER: Format Units Measure

  REST_FREQUENCY Double(NUM_LINES) Hz Frequency

  Several lines/transitions can be associated with a source and spectral window ID. REST_FREQUENCY will be a vector of doubles. The lines or transitions can also be specified in a string vector, TRANSITION: TRANSITION STRING(NUM_LINES)

- **offsets(numWindows)**

  MS SUBTABLE:

  FREQ_OFFSET

  PARAMETER: Format Units

  OFFSET Double Hz

  This subtable contains frequency offset information, to be added directly to the defined frequency labeling in the SPECTRAL_WINDOW subtable as a measure offset. This allow bands with small, time variable, ad hoc frequency offsets to be labeled as the same SPECTRAL WINDOW if desired.

- **velDefinition**

  MS SUBTABLE:

  DOPPLER

  PARAMETER:

  VELDEF Double m/s DOPPLER

  VELDEF is a DOPPLER measure which contains the velocity definition (RADIO, OPTICAL, TRUE) and the velocity in m/s.

- **restFrame**

  MS SUBTABLE:

  SPECTRAL_WINDOW, SOURCE

  PARAMETER:

  CHAN_FREQ Double(NUM_CHAN) Hz FREQUENCY

  The restFrame (LSRK, LSRD, BARY, GEO, TOPO, GALACTO, etc) is encoded in the FREQUENCY measure of the spectrum's abcissa, CHAN_FREQ.

- **velocity**

  MS SUBTABLE:

  SOURCE

  PARAMETER:

  SYSVEL Double(NUM_LINES) m/s RADIAL VELOCITY

  This has the source systemic velocity for each line/transition and the appropriate reference frame (LSRK, LSRD, BARY, GEO, TOPO, GALACTO).

- **xpos,ypos**

  MS SUBTABLE:

  OBSERVATION

  PARAMETER:

  TELESCOPE_NAME String

  The telescope name can be used to create a POSITION measure which has the earth location from which the observation was made.

- **coordMode**

  MS_SUBTABLE:

  SOURCE others

  PARAMETER:

  DIRECTION Double(2) rad DIRECTION

  The source direction and coordinate frame are in the DIRECTION measure.

- **epoch**

  MS SUBTABLE:

  MAIN, almost any

  PARAMETER:

  TIME Double s EPOCH

  The TIME may be used as a EPOCH measure to determine frame information. In the MAIN and POINTING tables, this is the mid-point of an observing interval.

2.4 Implementation

The observing scenarios may be implemented through the support of the two canonical modes: SFSW and MFSW. The SFSW mode is already supported and has been the default. The MFSW mode needs to be supported for cases in which the observation was planned as an MFSW and also in cases where it was not anticipated such that key information isn't necessarily available natively in the data. The MFMW-1 mode is essentially the same as the SFSW mode in terms of the implementation; each window has the information and behavior of an SFSW. Similarly, the MFMW-2, MFMW-3 will behave as the the SFSW or MFSW case as appropriate. The SFMW is essentially the same as the SFSW case in terms of the spectral axis tracking requirements, however, its processing relies on the ability to combine the banks (in a properly weighted-way) into a single spectrum.

2.4.1 Spectra

The filler constructs the spectral axis using the following information:

- sky frequency

  For each pixel along the IF axis of the backend data (RECEIVER axis for SP, SAMPLER for ACS data), the filler determines what is the matching row in the IF FITS file; that row gives the polarization and feed information and it gives all of the components for working out the sky frequency except for the LO1. The LO1 information is taken from LO1A or LO1B (based on information from the IF FITS file) and also provides the doppler information. The center channel is calculated (currently ad hoc - can now use information from the IF FITS file) that corresponds to the sky frequency.

The filler then calculates the LO1 frequency for each switching state at the first time it finds in the LO1 FITS file. It now has for each state a sky frequency axis at a specific time, antenna position, and pointing direction (this is a predicted position - the LO1 file contains the predicted LO1 frequencies) as well as a rest frequency, reference frame, and velocity to track.

- spectral axis

  Using the above and the doppler information, the sky frequencies are converted to a frequency axis in the tracked reference frame. The filler assumes that the M&C doppler tracking applies throughout the scan (hence the frequency axis is constant throughout the scan). As subsequent scans are filled, if their frequency axis is within the maximum of the specified tracking tolerance or 5 Hz of a previously filled spectral window, that spectral window is also assigned to the new data.

The **dish** plotter function, plotrec, handles the appropriate display of the data. Each spectrum is treated as a 1-d image with a DIRECTION and SPECTRAL coordinate. The SPECTRAL coordinate is constructed based on:

- native x-axis (based on the CHAN_FREQ in SPECTRAL_WINDOW subtable
- restfrequency
- reference frame (LSRK, LSRD, BARY, GEO, TOPO, GALACTO)
- velocity definition (OPTICAL, RADIO, TRUE)

Frame information is also provided (for the frame conversions that require it) on:

- epoch (time of observation)
- telescope (provides the DIRECTION measure with position and frame info)
- referencevalue (provides a DIRECTION measure with direction info)

The following details the handling for the unplanned and planned cases.

- Unplanned MFSW

```
d.open('stjul13SP'); #using the standards_jul13 data
myscan:=d.getc(35);
myscan.data.desc.restfrequency
1.4204058e+09
myscan.data.desc.restfrequency:=1.4104058e+9
d.plotscan(myscan) #if you have velocity displayed you should see the
                        #shift far from 0
```

A tool for manipulating the REST_FREQUENCY will be required for enduring changes (saved to disk for an MS). The tool implementation would look as:

```
- d.restfrequency(spectral_window_id,new_restfreq_value);
```

The spectral window can be obtained from the d.summary(T) on the MS.

Alternatively, if multiple REST_FREQUENCY values were specified in the observation, an additional argument to the plotter functions can be made available to specify which should be used. This would then be available upon the specification of the coordinate system and used for subsequent display and access.

- Planned MFSW

  The data will have the RESTFRQS and NUM_LINES information filled. The sditerator will have a function which will enable toggling between members of this vector. The proposed syntax would be:

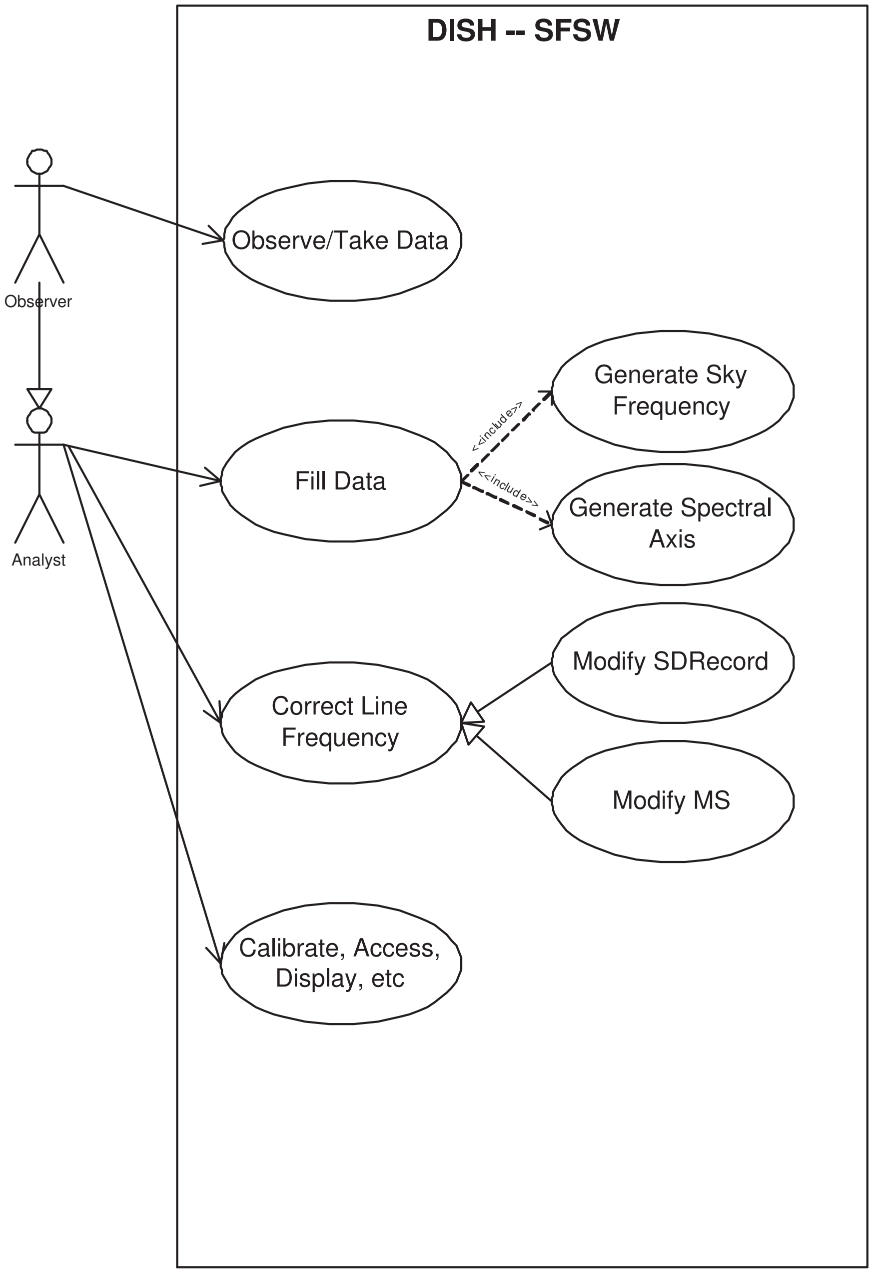
```
- d.open('myms_SP'); #creates the sditerator myms_SP
- myms_SP.rfindex(2); #change the spectral axis according to the rest
#frequency element 2
```

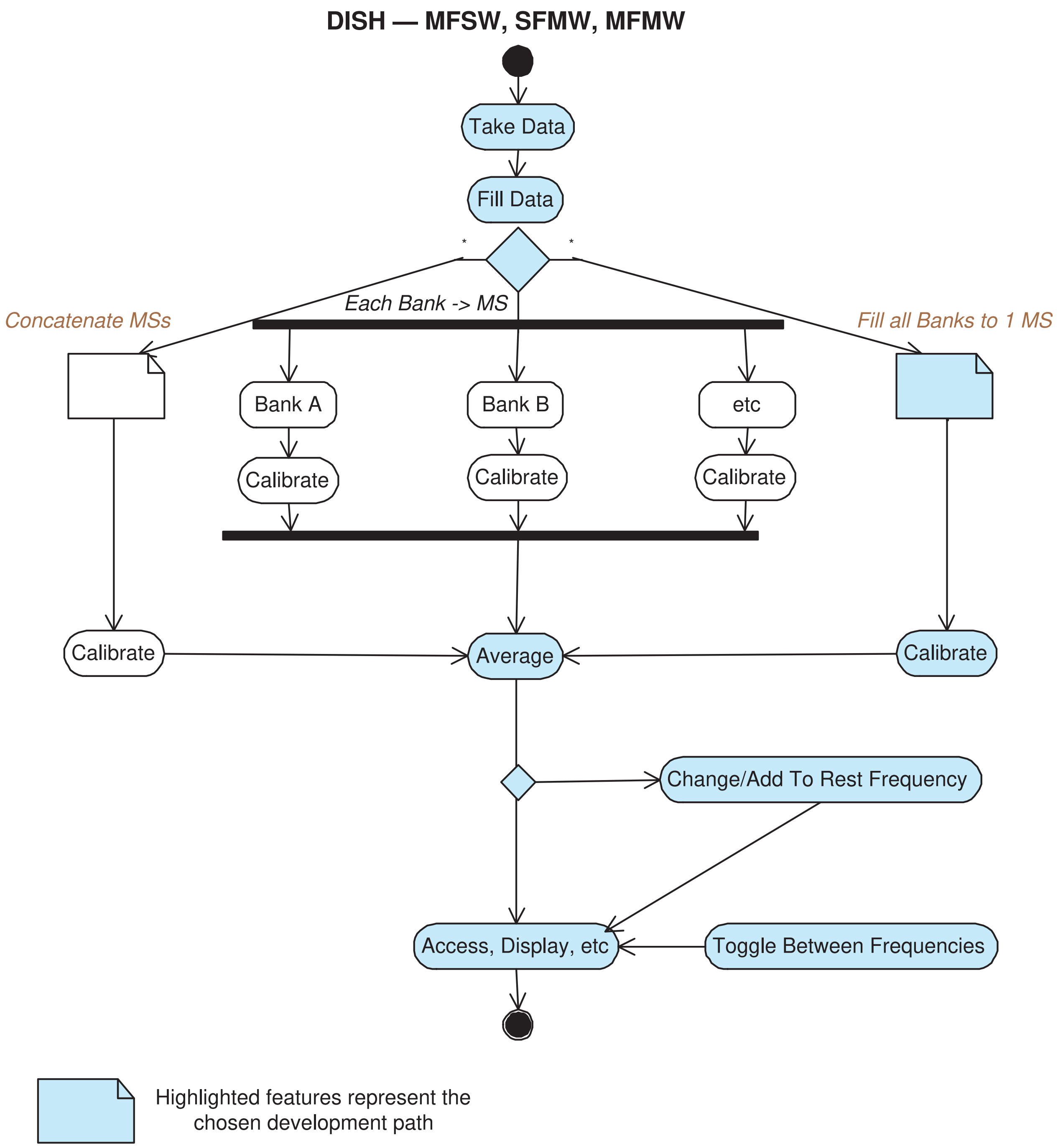
2.4.2 Basic Use Cases

Some graphical representations of the intended flow through the package.

2.5 Images

Each **image** tool also provides access to its underlying coordinate system. The **coordsys** tool has the function setrestfrequency (shorthand srf) which


DISH -- SFSW
Observer
Analyst
Observe/Take Data
Fill Data
Generate Sky Frequency
Generate Spectral Axis
<<include>>
<<include>>
Correct Line Frequency
Modify SDRecord
Modify MS
Calibrate, Access, Display, etc



DISH — MFSW, SFMW, MFMW
Take Data
Fill Data
*
*
Each Bank -> MS
Concatenate MSs
Fill all Banks to 1 MS
Bank A
Bank B
etc
Calibrate
Calibrate
Calibrate
Calibrate
Average
Calibrate
Change/Add To Rest Frequency
Access, Display, etc
Toggle Between Frequencies
Highlighted features represent the chosen development path

allows one to set the rest frequency by either over-writing the value or appending to a list. An argument of this function specifies which rest frequency is active.

Example:

```
- im:=image('my_image');
- csys:=im.coordsys();
- csys.summary();
Image name       : g29_pre.im
Image type       : PagedImage
Image quantity   : Intensity
Pixel mask(s)    : None
Region(s)        : None
Image units      : Jy/beam

Direction reference : GALACTIC
Spectral  reference : TOPO
Velocity  type      : RADIO
Rest frequency      : 1.42041e+09 Hz
Pointing center     :  29.300000  -6.500000
Telescope           : GBT
Observer            : UNKNOWN
Date observation    : UNKNOWN

Axis Coord Type      Name       Proj Shape Tile   Coord value at pixel    Coord
incr Units
--------------------------------------------------------------------------------
----------
1    1     Direction Longitude   SIN    20   10        29.300    11.00 -2.176708
e+02 arcsec
2    1     Direction Latitude    SIN   152   38        -6.500    77.00  2.176708
e+02 arcsec
3    2     Stokes    Stokes             1    1             I
4    3     Spectral  Frequency         300  100  1.420981e+09     1.00 -4.882812
e+03 Hz
                     Velocity                  -1.213028e+02     1.00  1.030572
e+00 km/s
T
- csys.setrestfrequency('1.41041e+09Hz',which=2,append=T);
T
```

```
- csys.restfrequency()
[value=[1.41041e+09 1.4204058e+09] , unit=Hz]
- im.setcoordsys(csys);
T
- im.summary()
Image name       : g29_pre.im
Image type       : PagedImage
Image quantity   : Intensity
Pixel mask(s)    : None
Region(s)        : None
Image units      : Jy/beam

Direction reference : GALACTIC
Spectral  reference : TOPO
Velocity  type      : RADIO
Rest frequency      : 1.41041e+09 [1.42041e+09] Hz
Pointing center     :  29.300000  -6.500000
Telescope           : GBT
Observer            : UNKNOWN
Date observation    : UNKNOWN

Axis Coord Type      Name       Proj Shape Tile   Coord value at pixel    Coord
incr Units
--------------------------------------------------------------------------------
----------
1    1     Direction Longitude   SIN    20   10        29.300    11.00 -2.176708
e+02 arcsec
2    1     Direction Latitude    SIN   152   38        -6.500    77.00  2.176708
e+02 arcsec
3    2     Stokes    Stokes             1    1             I
4    3     Spectral  Frequency        300  100  1.420981e+09     1.00 -4.882812
e+03 Hz
                     Velocity                  -2.246839e+03     1.00  1.037876
e+00 km/s
T
- im.view(); #use the ImageAnalysis Positions tool to view a spectrum in
             #velocity - it is far offset from 0 due to the shift in
             #restfrequency.
```

2.6 Targets

- GB M&C – provide information on numWindows, restFrequencies(numWindows), and offsets(numWindows)
- GB M&C – Permit a single integration time for the different banks of the Spectrometer
- AIPS++ – DOPPLER table filled by gbtmsfiller

  Enhancement: AOCso04231
- AIPS++ – SYSVEL in SOURCE table filled

  Enhancement: AOCso04231
- AIPS++ – RESTFRQS vector filled (once available from M&C)

  Enhancement: AOCso04232
- AIPS++ – Add sditerator function to switch between RESTFRQS

  Enhancement: AOCso04233
- AIPS++ – NUM_LINES in SOURCE table filled (once available from M&C)

  Enhancement: AOCso04232
- AIPS++ – Add REST frame option on plotter.

  Enhancement: AOCso04235
- AIPS++ – Revise center channel calculation (obtain from IF FITS)

  Enhancement: AOCso04234

3 Multi-Feed

3.1 Introduction

This note provides brief guidelines on how multi-feed observing analysis at the GBT will be enabled particularly the multi-feed imaging case. This note compliments the PTCS Project Note 1.1 (Proposed Implementation of Multi-Beam Capability – Richard M. Prestage).

3.2 GBT required implementation

The GBT will have implemented the following additional information required for multi-feed analysis:

- the addition of beam offset information to the Antenna FITS file to deduce the position of all of the beams on the sky.
- the addition of electronic beamswitching information to the IF FITS files to describe how backend data should be collated.
- extension of the existing GO functionality, and the addition of new GO procedures (Nod variants) to allow beam-switched point-source and mapping observations to be commanded through GO.

3.3 Beam information

The Antenna FITS file will contain a static table containing the beam offsets for all beams from the receiver fiducial point, as minutes on the sky in the directions of increasing azimuth and elevation. This information will be filled into the BEAM_OFFSET column of the FEED table in the MeasurementSet (v2.0).

The BEAM_OFFSET is a direction measure containing the beam position offset on the sky in the antenna reference frame (Az/El for the GBT). The receiver configuration file would have the form:

```
name    beamXelOffset   beamElOffset    SRFeed1 SRFeed2
1       -10.2           10.4            1       2

Where,

name            = integer number designating the feed (if physical)
beamXelOffset   = offset of the feed from the fiducial point (tracking beam)
                  in the cross-elevation direction, in arcminutes. The
                  corresponding azimuth offset applied will be:

                  AzOffset = Xel/cos(El)

beamElOffset    = offset of the feed from the fiducial point (tracking beam)
                  in the elevation direction.
SRFeed1         = if non-zero, indicates that this feed is part of a sig-ref
                  pair and specifies which feed has been denoted the first feed.
```

`SRFeed2          = for a sig-ref pair, indicates which feed has been designated the second feed.`

The beam offset information will be recorded in the Antenna FITS file while the sig-ref feed information will be recorded in the IF FITS file.

The following figure represents the data and interaction for the multi-beam observing currently available at the GBT. This is accurate only for the ACS (Spectrometer); the DCR and Spectral Processor write out a single FITS file (regardless of the number of IFs and FEEDs) and so will need to be treated differently within the calibration/data access routines.

The figure forks at the observing modes now available (the nomenclature is based on the PTCS Project Note 1.1 (underlined values) and their corresponding triptych of Observing Procedure:Switching State:Switching Signal). Below this is an example of the representation in the MeasurementSet of a single integration. Some key issues to be clarified are:

- Are there cases under which multi-feed data can be put into a single bank (e.g. at the expense of lower spectral resolution)?
- Are there cases under which beam switched data can be put into a single bank?

  Yes - 1/14/03
- How will the beams be labeled for >2 feeds for electronic beam switching?

  Yes - 1/14/03
- How can the different banks be properly associated with the data? The example data set has Bank A=IF 1, Bank B=IF 2, Bank C=IF 1, reverse sense of SIG/REF, Bank D=IF 2, reverse sense of SIG/REF. Is this always the case for Multi-IF or can this be setup differently?

  It can be setup in different ways - 1/14/03

3.4 Using the BEAM_OFFSET information

Two key uses of the FEED information and BEAM_OFFSETS are required.

- data access:

  Data from feeds within a scan should be independently accessible. This will be enabled through the use of an additional argument specifier in

getr,plotr,getc,plotc called nfeed. The nfeed will specify which feed to plot. The default will be the first feed in a list.

- imaging:

  Imaging will be enabled for multi-beam data through the following routine:

  - Read the POINTING table value for the RAJ2000,DECJ2000.
  - Read the BEAM_OFFSET information (az,el)
  - Convert the J2000 positions to Az,El positions for the time given.
  - Correct these values by beamXElOffset/cos(El) and beamElOffset

    In AIPS++ this will be:

    az = az + BEAM_OFFSET(0,0)/cos(el)

    el = el + BEAM_OFFSET(1,0)
  - Convert the modified Az,El positions back to J2000 positions
  - Grid the data

  This will be done for each feed.

3.5 Multi-beam GO Procedures

- non-switched pointed observations:

  Performed using Track, with data from the second feed being used to provide the reference or using OffOn, in which case the second feed will be ignored.

  Action: The non-switched track mode is not supported.

- beam-switched pointed observations:

  Performed using Track with electronic beam switching enabled. Supported with the current calib procedure.

  Action: requires refinement to use the new FITS information.

- nodded pointed observation:

  Action: Support new NOD procedure

- double-beam switched pointed observation:

  Action: Currently supported through calib on each scan and a getbs command. Full support utilizing the new FITS information is required.

- non-switched mapping observations:

  Performed using any of the current procedures (e.g., RALongMap) with switching disabled.

  Action: Support multi-beam imaging.

- beam-switched mapping observations:

  Performed using any of the current procedures (e.g., RALongMap) with switching enabled.

  Action: For small maps (<feed separation), the above multi-feed gridding will be adequate. For larger maps, dual-beam deconvolution algorithms need to be developed.

3.6 Targets

- AIPS++ – Enable calibration of multi-beam procedures (NOD)

  Target: 153 beam switched calibration refinements

  Target: 119.1 msconcat: works for all subtables

- AIPS++ – Enable use of the FEED information in data access

  Target: 111.15 gbtmsfiller: Support BS data

  Target: 111.16 gbtmsfiller: Support multi-beam data

  Target: 153 beam switched calibration refinements

```
The following 3 GBT-specific columns will be added to the FEED table
for GBT data:

GBT_SRFEED_ID - the FEED_ID corresponding to the associated feed when
        beam switching is in effect.  This uniquely identifies the other
        feed within a scan (and since the feed numbers should stay the
        same throughout a procedure set, this will also be the other
        feed during NODing - what changes is the BEAM_OFFSET).
GBT_FEED_NAME - from the NAME column of the ANTENNA file.  This preserves
        the original name since the filler will renumber things to start
        from zero (and it might reorder things, although I've tried to
```

```
        make that not happen).
GBT_TRCKBEAM - the name of the tracking beam.  Note that the non-physical
        beam information is not filled to the FEED table so if you were
        tracking, for example, "MR12", you won't find information about
        that beam in the MS.

In the main table, I propose adding this column for SPECTROMETER data
        only.

GBT_SRFEED_BANK - the BANK where the GBT_SRFEED_ID can be found.
        This will allow the calibration code to find the bank where
        the data from the other beam can be found.  If there is no
        other beam, this will be whatever bank that data is in.
        The reason this isn't in FEED is that I'm trying to avoid
        any dependence on SPECTRAL_WINDOW_ID in the FEED table and
        for multi-IF data, there will be different GBT_SRFEED_BANK
        value - one for each IF, for the same FEED_ID>
```

- AIPS++ – Enable use of the BEAM_OFFSETs for imaging

  Target: 172 Multi-beam Imaging

- AIPS++ – Enable dual beam-switched imaging

  No Targets set.

4 Multi-bank (ACS)

4.1 Introduction

Multi-bank observing may be considered a special case of multi-backend observing at the GBT; Each bank may be driven independently (different sampling rates, etc).

4.2 Data Access

Multi-bank data is filled to separate MeasurementSets. Data access on the separate MSs can be done serially by changing the pointer to each spectrometer bank, e.g.,

```
- d.open('ohsurvey_SPECTROMETER_A');
- d.gms();
```

```
- d.open('ohsurvey_SPECTROMETER_B');
- d.gms();
- d.calib(34);
- d.plotc(34,1);
- d.filein('ohsurvey_SPECTROMETER_A');
- d.plotr(34,1,1);
```

The problem is that for some multi-IF/multi-feed modes the data from these separate MSs must be dealt with simultaneously (e.g. SFMW multi-IF mode, NOD beam switched mode). This requires either an MS concatenation procedure that will properly index and sort the data and fill it to a single MeasurementSet or a nimble calibration routine which can deduce the appropriate relationships between the different banks.

4.3 Targets

- AIPS++ – MSconcat procedure

  Target: 119.1 MSCONCAT works for all subtables

- AIPS++ – Enhance calibration routines to operate on multi-bank data.

  No targets set.

5 Internal Decision Log

07 Jan 03

- Currently, the FEED table supports time dependent feed offsets and the GBT suggests its offsets will depend on elevation. Does this cause data bloat? Can we represent the elevation-dependence in another way consistent with the MS? Should we revisit the use of the MAIN table UVW for single dish pointing use (problem combining data).

  Decision: Existing MS definition of FEED table will work for GB case. The bloat issue should be mild as the BEAM_OFFSET changes should initially be per scan only. Long term, the addition of a single dish POINTING direction measure in the MAIN table should be assessed (analogous to synthesis UVW). Also explore possibility of storing function forms for offsets.

- FEEDID access within the sditerator/DISH. Do we need to provide accurate pointing information each feed or for just the tracking beam of a given observation. If the former, this requires the ability to parse the beam offsets and do appropriate transforms in several places. (Favor: only use pointing for tracking beam and offer a function (e.g., header) which will provide the true pointing position for a given feed).

  Pointing information in the header (and through sditerator) will be available only on the tracking beam. Detailed information on the pointing for each feed is only required for imaging applications which must handle this. Some convenience tool additions will be considered for enabling the pointing direction for a given feed to be provided.

- How do we handle the offsets for cross-polarization products?

  Not a high priority at this time. Nominally, the effective pointing direction for cross-correlated feeds would be derived from the pointing directions of the individual feeds and the individual beam patterns.

- For multi-feed imaging, we can handle non-beam switched data (have all positions, offset to grid) and beam switched data within a region smaller than the offsets. Beyond this, requires special deconvolution routines. Do we have these identified in general?

  We will be able to handle the first two cases: 1) non-beam switched multi-feed data and 2) beam switched with single and multi-feed for regions less than the switching or feed separation. Guidance should be provided (e.g. by the GBT SSR or NAUG).

- Will the msconcat procedures be delivered in time? Is this the best solution to the GBT problems or would a re-engineer of the filler's behavior be faster/cleaner (favor: msconcat solution since this is a pervasive issue in AIPS++ processing).

  The msconcat routines will be likely be delayed based on increased ALMA needs and the timing of the project reviews. Some discussion of the adaptation of the gbtmsfiller to provide this need - this would require some re-engineering of the filler as it was designed to support the on-line requirements and hence restricts filling to scans increasing in time. Given the impending take-over of the filler by GB M&C, analysis processing functions, which can handle the multiple MSs currently produced, will be adapted. Eventually, AIPS++ tools need to be able to handle multiple MSs seamlessly (there are several deferred targets to study this).

- Can the sampling times (integration times) for the different banks be assumed to be the same?

  Based on decisions from several years ago, the handling of multi-bank data by the filler is a special case of multi-backend observing; essentially, the banks can be completely independent in their sampling. However, in practice, it appears that this is a generalization that is unlikely to be utilized/needed. As a result, we currently have a series of targets that will need to deal with cumbersome association of data across banks which to the observer were always intended as a single spectrum/observation. We propose making a GBT policy that the observations using the spectrometer will simply have the same sampling. This assumption will enable a simplification in the filler, which matches the user intent, and allows the spectra to be stored within a single MS. In addition, this unites the treatment of multi-if/feed modes between the Spectral Processor, DCR, and Spectrometer. There appears to be no real restriction in scientific throughput as a result of this restriction since only the sampling needs to be the same (frequencies, BWs, resolutions, etc can all be different). – Approved Phil Jewell 1/14/03

5.1 Proposed SD MS changes

```
STATE table
-----------
o The information from SIG and REF is redundant; only one is
        needed.
o Add information on the procedure size (OBS\_MODE\_SIZE?).
        The sequence number information is obtained from SUB\_SCAN
o CAL should be a Bool to indicate whether the cal was fired
        (True) or not (False). The cal temperature (TCAL) is in
        the SYSCAL table.

MAIN/other tables
-----------------
o BEAM\_OFFSET is listed as a direction measure. Should be
        Float(2,num receptor) with units like arcminutes.
o Add information (boolean) to MAIN or SYSCAL indicating whether
        data had been calibrated or not. This is an issue for
        single dish data where the CORRECTED\_DATA exists upon
        filling and is altered only after calibration.
o Allow more cases with row-based measures. Currently, this is
```

avoided (Note 229), however many measures columns (particularly in the SOURCE table) should allow this. Even in cases where a constant measure reference is required for efficiency, there needs to be some way to record what the original reference was.

- o Split column keyword/unit information into a separate column. There are several cases where the column units or keywords should be able to change per row but not in the current definition. (e.g., units for CORRECTED_DATA, frame information for SYSVEL in SOURCE table).
- o The units for the CORRECTED_DATA column need to be optionally row-based.