Notes on GBT Ka band Data Analysis

v.10jan05 Brian S. Mason

This memo describes the calibration procedures for Ka band beamswitched photometric data. Some brief comments at the end address total power and mapping. A brief summary:

- Analysis algorithms appropriate for beamswitched data are presented and a prototype implementation presented. This is appropriate for the majority of accepted Ka band proposals, *e.g.*, high-z CO.
- Analysis algorithms for total power data are identical to existing approaches, with the proviso that up to $\sim 10\%$ cross-talk between the beams (or worse at the edges of the receiver band) may occur.
- SDFITS appears to work on Ka band data.
- AIPS++ MSFILLER appears to work on Ka band data but hasn't been extensively tested.
- Total power OTF maps are possible — with the caveats that MSFILLER hasn't been thoroughly tested (and is probably part of the mapping pipeline), and the feed crosstalk issue mentioned above.

1 Assumptions

We make the following assumptions:

- Data were collected in a Nod (2 or 4 scans) or OnOff procedure
- A **single** cal diode in the receiver was active (the LEFT diode)

Things we don't specify are:

- organization and averaging of multiple observations.
- implementational details like how you fill your data into SDFITS (a single project with all backends or one backend; individual scans; etc).

2 Analysis Algorithms

The following describes how to form a calibrated double-beamswitched spectrum from a single sampler. The spectrum from the other sampler should be independently calibrated; observers will want to compare plots of the separate sampler spectra, and average them for a final spectrum if all looks well. In the following, a capital letter denotes the raw spectrum for a given scan (ie lags have been converted into an uncalibrated spectrum, with VanVleck corrections and Hanning smoothing if desired); superscripts indicate the phase. A denotes the first scan in a NOD and B the second scan. For symmetric (4-scan) NODs the third and fourth scans are C and D. The following procedure will also work for beamswitched OnOffs though this is inefficient.

Note that this describes how to calibrate a single observation, defined as: two or four scans on a science target in a single system configuration, possibly with corresponding observations of a calibrator source. Multiple observations should be combined as desired by standard operations of averaging and interpolation.

- Average a selection of integrations for each phase of the scan. For instance, average integrations 2...6, omitting the first when the antenna was still not on source. The resulting averages are called $A^{sig,off}, A^{sig,on}, A^{ref,off}, A^{ref,on}$ for the first scan in the NOD and $B^{sig,off}, B^{sig,on}, B^{ref,off}, B^{ref,on}$ for the second scan in the NOD. Form an uncalibrated estimate of the sky spectrum as

$$sky = \left[(A^{sig,off} - A^{ref,off}) - (B^{sig,off} - B^{ref,off})\right]/2. \tag{1}$$

  Optionally the CAL ON phases can be included in the average to eliminate the $\sqrt{2}$ time penalty of calibration, like this:

$$\begin{aligned} sky = \quad & \left[(A^{sig,off} - A^{ref,off}) + (A^{sig,on} - A^{ref,on})\right]/4... \\ & - \left[(B^{sig,off} - B^{ref,off}) - (B^{sig,on} - B^{ref,on})\right]/4. \end{aligned} \tag{2}$$

- If symmetric 4-scan NODding was done, form sky from scans A and B as above, and the analagous quantity from scans C and D, and average with AB, ie, $(sky = sky^{AB} + sky^{CD})/2$.

- Form an uncalibrated estimate of the cal spectrum. For the first scan A the quantity to form is

$$cal = \left[(A^{sig,on} - A^{ref,on}) - (A^{sig,off} - A^{ref,off})\right]. \tag{3}$$

  Average this quantity over however many scans are in the procedure (2 for normal NODs; 4 for symmetric NODs)

- Normalize the sky spectrum by the cal spectrum , and apply elevation-dependent corrections which reference the spectrum to what would be observed above the atmosphere and at the nominal elevation of maximum gain for the GBT:

$$sky_{norm} = \frac{sky}{cal} \times \frac{G_{max}}{G(el)} \times e^{\tau/sin(el)}. \tag{4}$$

  Factors in this equation are

  - el: the telescope elevation during this scan. A mean value over the scan, or the elevation at the midpoint will suffice.
  - τ: the sky opacity at zenith with a default value of 0.05 for all of Ka band. This should be a user specifiable parameter.

– G: The GBT gain curve. In analyzing commissioning data I've assumed a rescaled form of Jim Condon's fit to 43 GHz data,

$$G(el) = Exp\left[-\left(\frac{el - el_0}{w}\right)^2\right] \tag{5}$$

with $w = 54^\circ \frac{43\,GHz}{\nu}$. Here ν is the center sky frequency of the given scan, in GHz. With this parameterization that Gmax = 1.

- Scale the data to flux density units using S_{cal}, defined as the equivalent flux density that would be produced by the calibration diode at the GBT peak gain, in absence of atmospheric attenuation. Rescaling the data is done as: $sky_{calib,Jy} = sky_{norm} \times S_{cal}$. The S_{cal} has been determined over the entire receiver band for the LEFT cal diode in the Ka band receiver but not the right diode. The signal to noise at the highest sky frequencies is not very good. S_{cal} can be determined from program data as $S_{cal} = \frac{cal}{sky} \times S_{calib} e^{-\tau} \frac{G(el)}{G_{max}}$

- If antenna temperature units are desired, rescale by the GBT gain $sky_{calib,K} = sky_{calib,Jy} \times \Gamma$. For the Kelvin to Jansky Gain I use Jim Condon's 43 GHz determination,

$$\Gamma = 2.0\,\mathrm{K/Jy} \times Exp\left[-\left(\frac{4\pi\sigma}{\lambda}\right)^2\right] \tag{6}$$

where lambda is the wavelength at the center sky frequency for this sampler, and $\sigma = 390\,\mu$m.

- Since we don't currently understand the sometimes significant differences between celestial S_{cal} and laboratory T_{cal} values, and the lab measurements may not be finely sampled enough for some applications, S_{cal} is recommended since it is most closely related to the data observers take.

3 Test Data sets

All of the following data are 4-bank 800 MHz spectroscopy nods.

- `CKA_14NOV04` scans 313 - 355 – blank sky (baseline tests). All have the same tuning.

- `CKA_14NOV04` scans $(111, 119, 121, 135, 137, 151, 153, 163, 165, 175, 179, 181)$ – various tunings on NGC7027, $S_{32} \sim 5.2\,Jy$.

- `CKA_14NOV04` scans $(217, 219, 225, 227, 233, 235, 241, 243, 249, 251, 257, 259, 265, 267)$ – various tunings on 3C48, $S_{32} \sim 0.9\,Jy$.

- `TKA_07NOV04` scans $(156, 158, 160, 162, 164, 166, 168, 170, 172, 174, 176)$ – various tunings on 3C48

- TKA_03NOV04 scans $(77, 83, 89, 95, 101, 115, 119, 123)$ – various tunings on NGC7027

Brian's Dec 04 summary of Ka band commissioning results is at http://wiki.gb.nrao.edu/bin/view/Projects/KaCommSumm

4 Further notes

4.1 Tcal table mappings

The calibration FITS file for the Ka band receiver has four binary table extensions: one for each receptor (currently there are two receptors), and for each receptor, a continuum and a spectral measurement table. The continuum and spectral tables are distinguished by the last latter ('C' or 'S' respectively) of the RECEPTOR keyword in the header of each binary table. The following table enumerates the mappings for the

```
2004_09_01_00:00:00.fits
```

version of the calibration FITS file, as well as showing the nominal order of the tables in the FITS file (which should never be used by software!).

Nominal Order	FEED	POLARIZE	RECEPTOR	Explanation
1	2	R	D2C	continuum Tcal for each frequency channel of Feed 2 Right pol'n
2	1	L	S2C	continuum Tcal for each frequency of Feed 1 Left pol'n
3	2	R	D2S	spectral Tcal for Feed 2 Right pol'n
4	1	L	S2S	spectral Tcal for Feed 1 Left pol'n

4.2 AIPS++/Filler issues

AIPS++ msfiller used to not work on beamswitched Ka band data (as of early fall 2004), however this seems to have nominally been fixed (as of 12jan05). Possibly the issue was the IF manager previously not reporting a Receiver in the Receiver column of the IF manager FITS file (as I recall it's tracing of the IF path terminated at the MMCONVERTER though it got frequencies etc correct).

Although the filler doesn't actually crash when run on data, it should be tested for correctness before observers are pointed at it. Some places where there could be problems:

- The beamswitch toggles between Feed 1 Left pol'n and Feed 2 Right pol'n. This shouldn't affect the data processing but may be an accounting issue. It is ok to pretend that you are toggling between like polarizations from the algorithm point of view.
- The Tcal mappings are somewhat different than ordinary, as enumerated previously. mmconverter/finding rx

4.3 Total Power and Mapping Data

Cross talk is also an issue for total power maps. There are at least three approaches to dealing with cross-talk in in a map:

- Ignore it, if $\sim 10\%$ confusion on arcminute scales is acceptable for the science.
- Make the map in a beamswitched mode and restore the total power map using existing techniques such as that of Emerson, Klein, & Haslam. Due to fundamentally missing information at harmonics of the beamswitch scale $1'.35$ (and amplified noise near that scale) this technique may not be useful for maps larger than a few arcminutes. Note that multiple chops (needed by the improved "EKH2" algorithm) are not possible although as the source rises and sets the chop will slowly rotate relative to the sky. No pipeline which does this has been tested with Ka band data (or GBT data to my knowledge), although perhaps AIPS could do it.
- Solve for the cross-talk. A method for doing so has been documented in "Pseudo-Correlation Radiometer Observing Modes & Calibration" (NRAO internal memo), available at `http://www.gb.nrao.edu/ bmason/gbt-dev/radiometer-modes.ps`. The relevant equation is Eq.20 in §2. In principle this algorithm could be implemented before passing the data to standard imaging tools. However the reconstructed total power may be too noisy to be useful, and further analysis is needed of this case.

Other than these issues imaging with Ka band is the same as for other GBT receivers.

5 IDL code

Following is a snippet of IDL code that illustrates how to do some basic operations getting your data into IDL. There are only about a half a dozen lines of actual active IDL code, the rest are comments.

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; reduceKaSample.pro - sample IDL program to access sdfits
;                      data and plot the average of integrations
;                      for a selected (scan + phase + sampler/bank)
;
; 12jan05 bsm
;
; this script uses functions in the astronomy library at
;   /users/astro-util/idl/astrolib/pro/
; which must be added to your IDL_PATH environment variable
; prior to startup.
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; CONFIGURATION
;  -select your scan, phase, and sampler

scan= 119
isSig='T'
isCal='F'
bankSamp='A1     '

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; The rest should just work.


;
; I filled my dataset using:
;
;   sdfits -backends=acs -scans=119:272 /home/gbtdata/CKA_14NOV04 \
;     CKA_14NOV04s119s272 |& tee sdFitsLog.txt
;
; read the whole huge project:
myTable = MRDFITS('CKA_14NOV04s119s272.raw.acs.fits', 1, myHeader)

; Obtain all integrations for a given scan, phase, and bank/sampler.
q=where(myTable.sampler eq bankSamp and myTable.scan eq scan and $
```

```
        myTable.sig eq isSig and myTable.cal eq isCal)
; get size info on the array of indices that meets our criteria
;  the second element of the return array is the length of the input array
qs=size(q)
; select all but the first integration, which we presume is still off source
;  NB: qs[1] is the length of q; subtract 1 b/c of zero indexing
qList=q[1:qs[1]-1]

;
; Make a 2-d array with all the data we just selected, and only that data.
;  I did this just for explicitness since it's not clear a priori
;  how IDL would store an array of structures, and thus, which
;  dimension to average over in the call to AVG (since you can pass AVG an array
;  of structures)
dataArr=myTable[qList].data[*]
; Note that if you type "print, dataArr" the separate integrations appear as consecutive
;  *ROWS* in the output, and the columns are different frequency channels.
;
; Better yet just type "size, dataArr"
;
; CONFUSINGLY, IDL's indexing convention is A[Col, Row] thus,
;  dataArr[freqCh, Integ]
;

; Now average the scan/bank/sampler/phase we've selected over integrations:
avgSpec=avg(dataArr,1)
; For the reasons above, averaging over the 1th (ie 2nd as opposed to 0th=1st)
;  dimension of the array, corresponds to averaging the integrations together for
;  each frequency channel.

plot,avgSpec
```