Outline

• Introduction to single dish calibration.
• Tracking observations:
  ▪ Data reduction in GBTIDL.
• Mapping observations:
  ▪ Data reduction with the gbtpipeline.
• High-frequency calibration.
A single dish telescope measures power:

\[ P^{[\text{cal}]}(\nu) = G(\nu) \left[ T_{\text{sou}}(\nu) + T^{[\text{cal}]}_{\text{sys}}(\nu) \right] \]

with

\[ T^{[\text{cal}]}_{\text{sys}}(\nu) = T_{\text{atm}} + T_{\text{spill}} + T_{\text{sw}} + T_{\text{rx}} \left[ + T_{\text{cal}} \right] \]

we are after

\[ T_{\text{sou}}(\nu) \]

See e.g., Winkel+2012 & O'Neil 2002
How do we get to $T_{sou}(\nu)$ from $P^{[\text{cal}]}(\nu)$?
From $P(\nu)$ to $T_{\text{sou}}(\nu)$

$$T_{\text{sou}} + \Delta T_{\text{sys}} = T_{\text{sys,off}}^{[\text{cal}]} \frac{P_{\text{on}}^{[\text{cal}]} - P_{\text{off}}^{[\text{cal}]}}{P_{\text{off}}^{[\text{cal}]}}$$

we also need to know $T_{\text{sys,off}}^{[\text{cal}]}$

$$T_{\text{sys,off}} = T_{\text{cal}} \left[ \frac{P_{\text{cal}}^{\text{off}}}{P_{\text{off}}^{\text{cal}}} - 1 \right]^{-1}$$

All quantities are frequency dependent

See e.g., Winkel+2012 & O'Neil 2002
From $P(\nu)$ to $T_{\text{sou}}(\nu)$

$$T_{\text{sou}} + \Delta T_{\text{sys}} = T_{\text{cal}} \left[ \frac{P_{\text{cal}}}{P_{\text{off}}} - 1 \right]^{-1} \frac{P_{\text{on}} - P_{\text{off}}}{P_{\text{off}}}$$

Changes to $T_{\text{cal}}$ are linear in $T_{\text{sou}}$ 😊

See e.g., Winkel+2012 & O'Neil 2002
Observing strategies

Position switching:
1. On-Off
2. Beam nodding
3. Sub-beam nodding

Frequency switching:
1. In band
2. Out of band

Strategy will depend on source properties and science goals. There are other options, and you can mix these alternatives.

See e.g., Winkel+2012 & O'Neil 2002
Position switching

- The science target is the "On" position.

\[ P_{\text{On}}^{[\text{cal}]}(\nu) = G(\nu) \left[ T_{\text{sou}}(\nu) + T_{\text{sys}}^{[\text{cal}]}(\nu) \right] \]

- An "empty" region is the "Off" position.

\[ P_{\text{Off}}^{[\text{cal}]}(\nu) = G(\nu) \left[ T_{\text{sys}}^{[\text{cal}]}(\nu) \right] \]

*It is better to keep the elevation constant*

---

1 # In AstrID:
2 OnOff( location, referenceOffset, scanDuration, beamName )
Position switching

The Off region should not have emission/absorption (the Off region should be more than a beam away from your source). The On-Off cycle should be faster than fluctuations in the telescope's gain (for narrow features ~few minutes <10 GHz, faster above).

Useful for:

- Observations of broad (>100 km s$^{-1}$) spectral lines.
- Observations of sources with crowded spectrum.

Drawbacks:

- Lost time slewing.
- Differences in $P_{on}$ and $P_{off}$ produce residual baselines.
Frequency switching

$P_{\text{off}}$

Noise Diode

Local Oscillator

Backend e.g., VEGAS

Frequency

Power P
Frequency switching

The Off region should not have emission/absorption. Be aware of the RFI environment (you do not want to switch to a region with RFI).

Useful for:

- Observations of narrow ($\leq 10$ km s$^{-1}$) spectral lines.

Drawbacks:

- Need to know source velocity \textit{a priori}.
- For larger $\Delta \nu$, larger residual baseline.
- No continuum measurements.
What is $T_{\text{cal}}$?

Is the equivalent temperature of a noise source injected to the signal.
By default the metadata includes a scalar value for $T_{\text{cal}}$. However,

- It is a scalar (good approximation for some receivers).
- You don't know when it was measured (the temperature of the noise diodes drifts).

→ Perform observations of a calibrator source!
A note on $T_{\text{cal}}$

→ Perform observations of a calibrator source!
A note on $T_{\text{cal}}$

Calibrators should:

- Have a known flux density at your observing frequency.
- Be stable in time (or you should know its flux density at the time of your observation).
- Be point-like.

Standard calibration sources and their properties:

- Perley & Butler 2017
- Ott et al. 1994
GBTIDL

- GBO supported data reduction software.
- Written in IDL.
- Locally available: `user@planck$ gbtidl`
- Supports spectral line data reduction.
- Knows about most spectral line observing modes, e.g., On-Off, Track, frequency switching, beam nodding.

Code & documentation: http://gbtidl.nrao.edu/
GBTIDL: getting help

1. GBTIDL -> usage,'show' ; Lists optional arguments.
2. GBTIDL -> usage,'show', /verbose ; Describes the command.
3. GBTIDL -> usage,'show', /source ; Show source code.
GBTIDL: data access

1. GBTIDL -> online; To access the active project.
2. GBTIDL -> offline, 'AGBT16B_037_04'; To access a project in /home/sdfits.
3. GBTIDL -> filein, 'mySDFITS.fits'; To access an SDFITS file in another location.
GBTIDL stores the data in containers (array-like structures). There are 16 of them, and the first (0) is called the primary data container (PDC).

```plaintext
; Save the contents of the third row of an SDFITS to variable `x`.
GBTIDL -> getrec,3 ; Read the third row into the PDC.
GBTIDL -> x = getdata() ; Copies the PDC into `x`.
; Zero the first 500 elements of x.
GBTIDL -> x[0:500] = 0
```

### Container arithmetic

```plaintext
; Add data containers 10 and 11 and save into 12
GBTIDL -> add,10,11,12
; Subtract data containers 10 and 11 and save into 12
GBTIDL -> subtract,10,11,12
; Divide data containers 10 and 11 and save into 12
GBTIDL -> divide,10,11,12
```
GBTIDL: observation information

1  GBTIDL -> summary ; Summary of loaded session.
2  GBTIDL -> header ; Metadata of container 0
3  GBTIDL -> list ; List the contents of each row in the SDFITS.
GBTIDL: baseline fitting

1. GBTIDL -> setregion ; Define region for baseline fitting.
2. GBTIDL -> nfit,3 ; Set polynomial order.
3. GBTIDL -> bshape ; Fit polynomials to selected ranges.
4. GBTIDL -> baseline ; Subtract baseline.
GBTIDL: smoothing

1 ; Smooth spectrum in PDC by 5
2 ; channels, keeping every 5th
3 ; channel.
4
5 ; Using a Gaussian kernel.
6 GBTIDL -> gsmooth,5,/decimate
7 ; Using a boxcar kernel.
8 GBTIDL -> boxcar,5,/decimate
GBTIDL: Gaussian fitting

; Fit a Gaussian to the spectrum on display. You will specify the region to be fitted and starting guesses using the GUI.

GBTIDL -> fitgauss
offline, 'TGBT20A_506_01' ; Connect to project TGBT20A_506 session 01.
summary ; List the contents of the sdfits file.
getps,6 ; Get position switched data for scan 6.
gsmooth,10 ; Smooth the data using a Gaussian kernel 10 channels wide.

Data courtesy of A. Bonsal & N. Butterfield
Try it yourself:

- Open GBTIDL, load the data for this example, TGBT20A_506_01, and plot the spectrum.
- Convert the frequency axis into velocity.
- What is the velocity of the object?
- Does it agree with the optically derived velocity? (see Barton+1996)
average_RSCG.pro

```pro
pro average_RSCG, scan_start, scan_end
 ; Average the position switched scans
 ; starting at scan_start and ending at
 ; scan_end, inclusive.

sclear ; Clears the default global accumulator.
freeze ; Turn off the plotter’s auto-update feature.

; Loop over scans, getting the position switched data,
; storing it in an accumulator for averaging.
for i=scan_start, scan_end, 2 do begin
    getps, i
    accum
endfor

ave
show
return
end
```

Save as a text file in `${HOME}/gbtidlpro`
In GBTIDL:

```plaintext
offline, 'TGBT20A_506_01'
.r average_RSCG
average_RSCG, 6, 9
show
gsMOOTH, 12
show
fileout, 'rscg31_spec.fits'
keep
```

- Display the averaged spectrum.
- Smooth using a Gaussian kernel.
- Show the smoothed spectrum.
- Save to this fits file.
- Save the primary data container (PDC) to the fits file.
In GBTIDL:

```plaintext
offline, 'TGBT20A_506_01'
.r average_RSCG
average_RSCG, 6, 9
show
smooth, 12
show
fileout, 'rscg31_spec.fits'
keep
```

Display the averaged spectrum.

Smooth using a Gaussian kernel.

Show the smoothed spectrum.

Save to this fits file.

Save the primary data container (PDC) to the fits file.
In GBTIDL:

display offline, 'TGBT20A_506_01'
display .r average_RSCG
display average_RSCG 6 9

```gfortran
fileout, 'rscg31_spec.fits'
```

**GBTIDL Plotter**

- **Scan**: 7
- **V**: 1600.0 RADI-LSR
- **Int**: 00 04 53.1
- **Natalie Butterfield**: LST: +15 56 24.3
- **LSR**: +15 56 24.3
- **RSCG31**

**Antenna Temperature (T_a)**

- **Antenna Temperature (T_a)**
- **1.410**
- **1.411**
- **1.412**
- **1.413**
- **1.414**

**LSR Frequency (GHz)**

- **Thu Feb 4 21:42:25 2021**
- **Tsys**: 20.77
- **Pol**: YY
- **Tcal**: 1.46
- **BW**: 23.4375 MHz
- **Fsky**: 1.41279 GHz
- **IF**: 0
Scalar vs vector calibration

Using averages while calibrating introduces biases, particularly if there are standing waves and/or a non-flat frequency response (e.g., power law in $T_{\text{cal}}$).
Scalar vs vector calibration

GBTIDL uses averages, but the GBT has an unblocked aperture :)

![Graph showing comparison between vector and scalar calibration, with temperature plotted against frequency.](image-url)
Mapping

The telescope "scans" an area while tracking the map center.
Mapping

There are different mapping patterns.

On-the-fly (see e.g., Mangum+2000):
- RaLongMap, DecLatMap
- Daisy

Grid:
- PointMap

Others:
- Spider
- Z17
Mapping

If there are regions free of emission in the mapped area, you can use these as Off positions.
GBTIDL example 3

to_ta.pro

pro to_ta, file_out, scan_ref, scan_start, scan_end, int_start, int_end, if_num
; Process mapping scans and save them to file_out.
; It uses scan_ref as reference scan.
; Starts with scan_start scan and ends at scan_end scan, inclusive.
; Process each integration individually, starting at int_start and
; ending at int_end, inclusive.
; Only process data for spectral window if_num.

freeze ; Turn off the plotter’s auto-update feature.
fileout, file_out

; Loop over scans, converting to antenna temperature
; using the reference position. It saves each integration
; into a new fits file.
for i=scan_start, scan_end, 1 do begin
  for j=int_start, int_end, 1 do begin
    getsigref, i, scan_ref, intnum=j, ifnum=if_num
    keep
  endfor
endfor

return
end

Save as a text file in ${HOME}/gbtidlpro
Once the data is calibrated use the `gbtgridder` to produce a data cube.

```

gbtgridder -c 11000:11251 -a 7 --noline --nocont -o output input.fits
```

This will produce a data cube `output_cube.fits` with channels 11000 to 11251 averaged by a factor of 7. Also, a map of weights `output_weight.fits`.

Source code at:

https://github.com/GreenBankObservatory/gbtgridder

(No documentation available)

```

gbtgridder -h ; get help on input options.
```
GBT pipeline

Calibrates and grids the data.
It uses the gbtgriddler for gridding.

Works for receivers with noise diodes.
Designed for processing KFPA observations (it has been tested with L band data as well).
Default values suitable for KFPA observations.

Documentation:
https://safe.nrao.edu/wiki/bin/view/GB/Gbtpipeline/PipelineRelease

Source code:
GBT pipeline

Some options:

- **-i** # Input SDFITS.
- **-m** # Mapping scans.
- **--refscan** # Scans to use as reference.
- **-w** # Spectral window to process.
- **-c** # Channels to grid.
- **--beam-scaling** # Multiply Tcal by this value.
- **--imaging-off** # If you do not want to grid.

Example:

```bash
gbtpipeline -i my.sdfits.raw.vegas -m 14:24 --refscan 13,26
```

Map scans: 14 to 24, reference scans 13 and 26
Continuum

Pick receiver, integration time and slew rate to minimize $\frac{1}{f}$ noise (timescale $\sim$1.4 s at C and K band, see e.g., Harper+2015).

There is no observatory supported continuum data reduction package.

Contact your project friend if you are interested in doing continuum science.
Continuum: example

How to fill continuum data?
See: https://github.com/GreenBankObservatory/gbtcal-nb/blob/master/gbtcal.ipynb

Contact your project friend if you are interested in doing continuum science.
Questions?

Send feedback to: warmentr@nrao.edu
psalas@nrao.edu

The Green Bank Observatory is a facility of the National Science Foundation
operated under cooperative agreement by Associated Universities, Inc.
Hot & cold loads

ARGUS and the W-band receiver do not use noise diodes, they use hot and cold loads.
The W band receiver can see two loads: $T_{\text{amb}} \& T_{\text{cold}}$.
ARGUS can see one load : $T_{\text{amb}}$
Hot & cold loads

W-band

\[ G = \frac{(T_{\text{amb}} - T_{\text{cold}})}{(P_{\text{amb}} - P_{\text{cold}})} \]

\[ T_{\text{sys}} = GP_{\text{off}} \]

\[ T_{a} = T_{\text{sys}} \frac{P_{\text{on}} - P_{\text{off}}}{P_{\text{off}}} \]
Hot & cold loads

ARGUS

\[ T^*_a = T^*_s \cdot \frac{P_{on} - P_{off}}{P_{off}} \]

\[ T^*_s(t) = \frac{T_{cal}}{\left( \frac{P_{amb}}{P_{off}} - 1 \right)} \]

\[ T_{cal} \approx (T_{atm} - T_{bg}) + (T_{amb} - T_{atm})e^{\tau_0 A} \]

\[ T_{cal} \approx T_{amb} \]
Temperature scales

- $T_a$ : Antenna temperature.
- $T'_a = T_a e^{\tau_0 A}$ : Antenna temperature corrected for atmosphere.
- $T^*_a = \frac{T'_a}{\eta_l}$ : Forward beam brightness temperature.
- $T_{mb} = \frac{T'_a}{\eta_{mb}}$ : Main beam antenna temperature.

For the GBT:

- $\eta_l \approx 0.99$, GBT memo #16 & #19
- $\eta_{mb} = 0.44 \pm 0.04$ @ 86 GHz, GBT memo #302
  $\eta_{mb} = 0.94$ @ 5 GHz
Single Dish FITS files.
Default data i/o for GBTIDL and gbtgriddler.
VEGAS and DCR data stored as SDFITS.
Definition: https://fits.gsfc.nasa.gov/registry/sdfits.html
Details: https://safe.nrao.edu/wiki/bin/view/Main/SdfitsDetails
Community developed data reduction tools

- DEGAS: ARGUS survey (https://github.com/GBTspectroscopy/degas)
- TMBIDL: general use (https://github.com/tvwenger/tmbidl)
- groundhog: general use (https://github.com/astrofle/groundhog)
- SDgridder: gridded (https://github.com/tvwenger/sdgridder)
- HCGrid: gridded (https://github.com/HWang-Summit/HCGrid)
- sdpy: ??? (https://github.com/keflavich/sdpy)
Questions?

Send feedback to: warmentr@nrao.edu
psalas@nrao.edu

The Green Bank Observatory is a facility of the National Science Foundation operated under cooperative agreement by Associated Universities, Inc.