Onsala 20 m data properties
Important information for observers

NB! This is very much a β version!

May 26, 2020

Contents

Introduction

This document is of the work-in-progress type and will likely remain so for the foreseeable future. The graphs shown here are mostly the results of individual test measurements which may vary from one occasion to the next due to weather and other factors. Also the hardware is continously upgraded and modified which may affect the performance. It would require regular extensive test campaigns to arrive at the best average (and current) values to use and that is not feasible since this facility is run with a minimum of staff support compared to many other similar telescopes.

Some formulae and methods described here have been checked once but not double-checked and there could be remaining sign errors or similar faults. We urge users to verify themselves that the results are reasonable if e.g. the CLASS method for resolution correction is employed (e.g. check that lines in overlapping tunings get narrower when averaged together and not broader although in many cases the effect will be too small to be noticable).

We also note that some FITS header keywords mentioned here may not be present in older data and that the headers of data files from the older Pegasus and current Bifrost control systems are different.

Henrik Olofsson
20-m telescope manager & staff scientist

Chapter 1

Frequency axes

1.1 How the system is tuned

The user supplies a reference frequency to the control system which is given in the source's rest frame. Note that this does not have to become the centre frequency of the spectrum since the user may request an additional offset. This is a handy mechanism in case we want to move the main line to the side in order fit other lines within the same bandpass, but we still want to have the frequency and velocity axes related to the main line.

The system then calculates the corresponding Doppler-shifted frequency in the topocentric frame based on the total radial velocity difference between the observatory and the source. If any offset is needed or requested, it is added as the last step. The LO is then set to a value that will make this sky frequency the centre of the bandpass.

Note that in the 4 GHz mode, even without any offset requested, the reference pixel will never be in the centre of each 2.5 GHz "half" that build up the 4 GHz bandpass (without any offset requested, it will however be at the centre of the receiver bandpass). The reason for this is that we want to have the same reference frequency in both spectra which by default is in the middle of the overlap of the two "halves", see Fig. 1.1. If the user requests a frequency offset of >0.5 GHz, the reference pixel will actually be outside the spectrum edge in one of the halves but that is fully allowed in FITS files and should not cause any problem with standard reduction software.

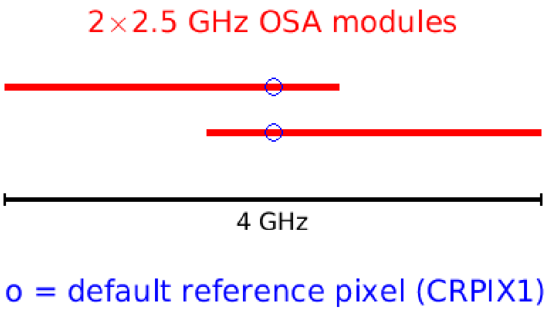


Figure 1.1: Illustration of how the OSA 4 GHz bandwidth mode works. Note that this setup is duplicated to cover both polarisations and the products that cover one and the same side are either named OSA1 and OSA2 in the FITS header keyword INSTRUME, or named OSA3 and OSA4 in case of the opposite side.

1.2 Relevant keywords in the FITS file

The reference frequency will appear in the RESTFREQ keyword, and the CRPIX1 keyword defines to which channel this frequency refers. The OBSFREQ gives the sky frequency at this channel. These are all unambigous and strictly correct quantities.

The CDELT1 keyword however – that defines the channel spacing – is only truly valid in the topocentric rest frame which has implications as seen in the next Sect.

VELO-SOU gives the source velocity w.r.t. the LSR system and VLSR is the LSR velocity of the reference channel. Normally these two are the same but the VELO-SOU keyword is there to help keep track of the orginal velocity in case we want to modify the reference channel at a later stage.

VELO-LSR gives the velocity of the observer w.r.t. the LSR system. Note that it has a different sign convention than VELO-SOU (i.e. the observer and the source have opposite signs when moving in the same direction; they do have the same sign if both are moving towards LSR though). We have followed the commonly used standard in this matter.

Note that FITS files saved by CLASS is using the keyword VELO-LSR for the function of VLSR which itself is absent in such headers. Thus VELO-LSR has a different meaning in FITS files saved by CLASS which is important to keep in mind if method 2) in Sect. 1.3.1 is attempted.

1.3 The "accordion effect" on the frequency axis

In the source's rest frame the channel separation is in fact slightly different from CDELT1 depending on the total red- or blueshift (they are only the same if the source is perfectly at rest w.r.t. the observer). One can see this as an expansion or compression of the bandpass. If we e.g. observe a red-shifted source with bandwith 4.0 GHz here at Earth we are in fact covering slightly more than 4 GHz in the source's rest frame due to compression.

We do not correct for this effect but are leaving the topocentric CDELT1 as is, which is the built-in instrumental values for a given spectrometer configuration.

This means that a frequency axis built up using the CRPIX1, RESTFREQ, and CDELT1 values will have a slight error in it which will increase with the distance from the CRPIX1 channel (at CRPIX1 itself it is exactly correct).

An example: consider the rather extreme case of observing a source with a net Doppler velocity of $+100\,\mathrm{km\,s^{-1}}$ in a 2.5 GHz spectrometer bandwidth configuration centred at 91 GHz but with a reference frequency far to the side, at 90.0 GHz. The error in the frequency axis near 92.0 GHz will then be about 667 kHz. Note, however, that a sky frequency axis, built up of CRPIX1, OBSFREQ, and CDELT1 will always be correct at every channel. The simplest way to correct for the effect is to simply recalculate the CDELT1 value according to

$$\mathrm{CDELT1}' = \mathrm{CDELT}\frac{\mathrm{RESTFREQ}}{\mathrm{OBSFREQ}}$$

or, using the net source↔topocentric relative velocity,

$$\mathrm{CDELT1}' = \frac{\mathrm{CDELT1}}{1 - V_{\mathrm{Dop}}/c}$$

The reason the system does not do this automatically is that every spectrum would then have a (very slightly) different frequency axis since the Doppler correction is continuously updated due to the Earth's rotation and orbital motion as can be seen in the variation in the OBSFREQ value. *Therefore, it is up to the observer to consider this effect in the data reduction and analysis.* In most cases, a satisfactory solution would be to apply the correction according to the expression above, using a V_{Dop} average over the whole observational campaign. However, if the campaign is drawn out over different parts of the year and the source is near the eccliptic plane, this may not be feasible since the difference in contribution from the Earth's orbital motion may be up to ±30 $\mathrm{km\,s^{-1}}$. In such cases, frequency corrected spectra from each part of the campaign may need to be resampled to have a common axis before averaging them together.

But, again, this is never a problem if the lines to be analysed are at or close to the reference frequency. And it is never a problem anywhere in the spectra if the total Doppler correction for the source is always small.

The perhaps most intuitive way to check how large the effect is, is to calculate the error per channel and then multiply this error with the number of channels between your line and the reference frequency/pixel in the spectrum. However, when we do this using the expression above, the frequency resolution cancels out and we end up with

$$|f_{\mathrm{err}}| \; = \; | \; (1 - \tfrac{\mathrm{RESTFREQ}}{\mathrm{OBSFREQ}}) \times (f_{\mathrm{line}} - \mathrm{RESTFREQ}) \; |$$

Example: We have tuned to ^{13}CO at 110201.354 MHz and for this particular source the total Doppler shift at the time of observation was such that the commanded sky frequency was 110215.259 MHz. We want to check how large the error is at the position of the C^{18}O line at 109782.176 MHz. Using the expression we find an error magnitude of 53 kHz which is quite small but the distance to the reference frequency was only 400 MHz in this example. Also note that the error will change over the year so when averaging data from different campaigns there is a risk of smearing on top of an offset/error.

1.3.1 Using CLASS to correct for the "accordion effect"

There is a built-in mechanism in CLASS that reads a parameter called the "Doppler factor" for certain types of data sources. Unfortunately, CLASS does not know how to look for the Doppler factor when importing Onsala data as single FITS files. In recent versions of CLASS it will in fact issue a warning about missing DOPPLER information and it will not even allow an AVE command on a stack of spectra. The fix is to do the correction yourself, and it can be done in two ways:

1) insert the correction directly by modifying the FRES CLASS keyword

   ```
   sic\let R%HEAD%SPE%FRES /status Write
   sic\let R%HEAD%SPE%FRES = 'dfcur*fcur/fsky'
   sic\let R%HEAD%SPE%DOPPLER /status Write
   sic\let R%HEAD%SPE%DOPPLER = '0.0'
   ```

   where dfcur, fcur, and fsky can be set from CDELT1, RESTFREQ, and OBSFREQ from the FITS header using the DEFINE FITS HDR command, respectively. dfcur and fcur and can also be set from the programme variables R%HEAD%SPE%FRES and R%HEAD%SPE%RESTF once a spectrum has been read with FITS READ but note that fcur will be in units of MHz in that case and the value of fsky should then also be converted from Hz to MHz to match.
   Using this method, we manually fix the frequency resolution and we can then continue as if the source and observer is in the same rest frame. This is the simplest solution since CLASS won't have to keep track of two different values for the frequency resolution. However, it appears the newest versions of CLASS[1] will no longer allow modification of 'programme variables' and the first and third command above will generate an error. If this is the case, the next option below should hopefully provide a working alternative.

2) calculate the Doppler factor and set it using the MOD DOP command. Below follows the full code needed to correct one

[1] Meaning at some point after GILDAS version jul17c which still works.

spectrum and add it to a stack of corrected spectra.[2]

```
file out myclassfile.20m single /append
def fits HDR myfitsfile.fits  ! Define struct with all header keywords
let VELOLSR = 'HDR%VELO_LSR'  ! Topocentric vel. wrt LSR
let VELOSOU = 'HDR%VELO_SOU'  ! Same as VLSR but *not* same as VELO_LSR
del /variable HDR             ! Structure not needed anymore
fits read myfitsfile.fits     ! Read the actual spectrum
mod dop '-(VELOSOU+VELOLSR)/2.99792458e8' ! -Vdop/c is the Doppler factor
! Note that the sign convention is different for VELOSOU and
! VELOLSR, that is why they are added and not subtracted above.
write
```

After this has been done, one can now see that CLASS shows two different values for the frequency resolution in the output of a DUMP command, e.g.:

```
r%head%spe%fres  =  1.9073486E-002 (obs.),  1.9071953E-002 (rest)
```

To average all spectra together while taking into account the slightly varying resolution, one can do:

```
file in myclassfile.20m
find
set align f i
ave /resam /nocheck
```

Note that the /nocheck argument is absolutely necessary to force CLASS to use the new (rest) values instead of the (obs.) values. It should also be mentioned that the set align command should have “c” instead of “i” as last argument if several overlapping tunings are averaged together.

Note that method 2) as described above cannot be used for FITS files already previously saved by CLASS since VELO-LSR has a different

[2] For science reductions where T_{mb}-scale is desired, this is also a good opportunity to apply the main beam efficiency which is elevation-specific for the 3- and 4-mm receivers. This can be done by inserting the line `let ETAMB = 'HDR%BEAMEFF'` before the header structure is deleted, and the line `let RY = RY/ETAMB` before the write command at the end.

meaning in such spectra, see Sect. 1.2 (and the sky frequency information is lost as well so correction is no longer possible at all).

And it should also be mentioned that one advantage with method 1) is that it is possible to save corrected spectra as individual FITS files if that is desired since we have changed the one and only frequency resolution value which will go into the FITS header when saved. If method 2) is used, it seems unfortunately like CLASS is plugging in the uncorrected resolution in the header when spectra are saved individually. Please contact the author of this document if a workaround is found for this.

1.3.2 Using XS to correct for the "accordion effect"

This feature is only present in more recent versions of the software. Under the menu Prefs, select Edit current preferences. In the bottom right corner, at Apply Doppler shift res. when reading FITS:, change from No to Yes and save preferences. Now the frequency axes will be automatically corrected when reading FITS files. Before you can apply the normal averaging procedures in XS, you must still find the best set of overlapping frequency channels in all spectra. This can be done by resampling to slightly wider resolution using Redres scans...in the Reduction meny. Or possibly by using Data Sets→Shift x-scale... and then select Align in frequency with clipping and then Apply. This latter will forcefully shift the frequency of nearest channels to each bin so some smearing will occur.

NB! DO NOT FORGET TO TURN THE Apply Doppler shift OPTION OFF AGAIN. As long as it is turned on, all spectra loaded will be corrected even if you are e.g. reading a spectrum that was previously corrected and saved. It will thus actually corrupt frequency axis when reading it again.

1.3.3 Correction of historical data

Before December 2016, the OBSFREQ keyword did not contain the correct value[3]. At that time, the VELO-LSR keyword was not present either so

[3] This bug, which before the fix was present as far as we can trace historically, made the system recalculate the sky frequency based on an old local Az/El direction, either the one at the time of the last receiver tuning, or the one at the very start of an observation sequenence. Since the Earth is turning and changing its orientation w.r.t. the LSR system,

neither of the two CLASS methods mentioned above would work for correction.

Although the OBSFREQ value in the very first spectrum taken in each observation command (Single or Map) *might* still be correct, we recommend using any of the web sites that are available that let's one calculate the quantity corresponding to VELO-LSR for a given time, celestial direction, and position on the Earth's surface and then use method 2 CLASS correction. It should be sufficient to use one and the same VELO-LSR for, say, a week of observations at a time since the effect of day-to-day variation should be tiny.

1.4 Image band frequency

The IMAGFREQ keyword should reflect the image band *sky frequency* of the reference channel. Strong lines from the image band can leak into the signal band even when the sideband suppression is quite good and it is always a good idea to check if any unidentified lines could come from the image band. Lines will be mapped in mirror image style between the sidebands. Example: if an LSB signalband is centred at 91 GHz and the frequency we are interested in is 90.5 GHz, it means that possible leakage to that channel could come from a USB frequency of 90.5+0.5+2*IFfreq+0.5 = 103.5 GHz for an IF frequency of 6 GHz. If we are interested in 91.5 GHz instead, the USB frequency would be 91.5-0.5+2*IFfreq-0.5 = 102.5 GHz. A more convenient way to write it would be using a band centre frequency, $f_{\rm c}^{\rm sig}$.

$$f^{\rm im} = 2f_{\rm c}^{\rm sig} - f^{\rm sig} \pm 2 \times {\rm IFfreq}$$

Now please note the following:

- The quantities in the expression above are in the sky frequency domain. The $f^{\rm im}$ should then be calculated back to the source rest frame by accounting for the Doppler shift at that frequency. Note

the OBSFREQ thus changed more than it should over time. This only pertains to the FITS header value, the receiver LO was still commanded correctly and the RESTFREQ value is still thus correct.

that the difference between the upper and lower sideband centres are only identically equal to 2×IFfreq in the sky frequency domain. In the source rest frame, the difference will vary since the Doppler corrections, in terms of frequency shift, are slightly different for the two sidebands as discussed in Sect. 1.5.

- Note that $f_c^{\rm sig}$ is not necessarily the middle of the spectrum range, it is rather the middle of the receiver bandpass of the signal band which is in turn determined by the LO setting (FITS keywords LO1-FREQ ± IF1-FREQ). When the 4 GHz spectrometer mode is used, for instance, the bandpass is covered by two offsetted but partly overlapping spectrometer halves and $f_c^{\rm sig}$ would correspond to a position half a gigahertz inside the upper and lower edges of each 2.5 GHz spectrum, respectively, see the location of the default reference pixel in Fig. 1.1.

- Up until March 2016, the IMAGFREQ keyword did not provide a correct value for offset tunings; it simply added or subtracted 2×IFfreq from the OBSFREQ value.

- The XS reduction software does not properly display the image frequency for data taken with offset tunings and/or using the 4 GHz mode (this is because XS always recalculates the reference pixel and moves it to the centre upon loading a spectrum).

1.5 2SB observations

When observing with two sidebands, it is important to be aware that the distance between the lower and upper sidebands is only identically 12 GHz in the topocentric rest frame. In the source's rest frame it varies slightly with the total Doppler velocity at the time of observation. This is already taken into account in the headers of the spectra from the "other-sideband", i.e. the one we are not tuned to, which is evidenced by a varying RESTFREQ value (in the signal band RESTFREQ will always be one and the same value for a given tuning). But the consequence of this is that lines in the other sideband can end up at rather different channel locations depending on at what time of year the observation was done, in contrast to the signal band where lines will always end up at the same channels (at least if they are close to the reference channel, see Sect. 1.3 for further discussion about the "accordion effect").

Thus, *data from the other sideband taken at different times of the year should never be combined by a simple channelwise averaging* unless the net velocity difference observer-source does not change much seasonally as can be the case for sources close to the eccliptic pole.

1.6 Frequency switched data

In XS, folding of frequency switched data can be done using the Reduction→Fold scans... menu option. The "Order of pos. and neg. peaks:" should always be set to Neg-Pos[4]. The frequency throw should best be entered manually rather than clicking the "or use header values" option which may not always work due to an unknown bug.

In CLASS, there is to our knowledge no built-in functions that can fold Onsala data, the existing functions appear to be designed around a different way of saving the FSW data. Below we present an example of a code snippet that will fold a spectrum in a brute-force way, we manually clone spectrum and manipulate the axes to achieve the wanted result. Note that in principle this should be done before any correction described in Sect. 1.3.1 is done since the frequency throw is given in the sky frequency domain.

```
! In this example, the spectrum was observed using a throw
! of 20 MHz so we now need to shift our spectrum f-axis 10 MHz
! to either side.
! There are several ways to do this but using "mod recenter"
! to bring the current f and v to a new reference channel
! should be the most logical choice.

set align f i
set weight e
```

[4] Very old data (more than at least ∼one decade old) may require the Pos-Neg option but that should be obvious from the result; signals will simply be inverted if the wrong option is used

```
file out tmptmp.class single /overwrite  ! temporary output

fits read myfswscan.fits
mod rec 'r%head%spe%rchan-10/r%head%spe%fres'
mod dop 0
mult -1             ! Make the negative component positive
write

fits read myfswscan.fits
mod rec 'r%head%spe%rchan+10/r%head%spe%fres'
mod dop 0
write

file in tmptmp.class
find
ave /resam

! Now we can WRITE this spectrum to a CLASS stack or save
! it using FITS WRITE.
```

Note that the `mod dop 0` commands are needed in recent versions of CLASS, otherwise the average command will be refused. But that also means that method 1) described in Sect. 1.3.1 for frequency resolution correction is (probably) no longer possible. If correction is desired, it should be done directly after this folding procedure, before the spectrum is pushed to a stack.

Chapter 2

Data product fidelity

2.1 Potential spectrometer edge aliasing

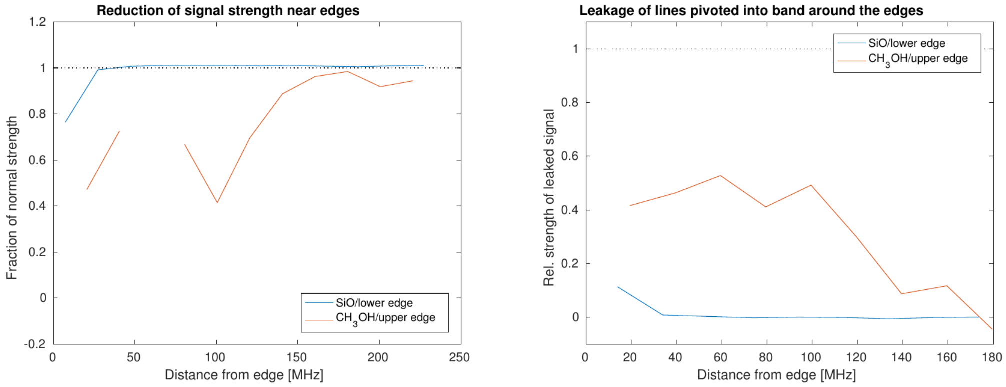
By the term aliasing effects, we here mean two ways data can be affected near the edge of the spectrometer bandpass: a line just outside the spectrometer edge can be symmetrically folded inside the edge and thus appear as a false spectral line which could lead to misidentications. In the range where such effects can occur, the normal spectral lines that *are* located inside the edge can appear weaker than normal ("when the gain for aliased lines goes up, the gain for real non-aliased lines goes down"). The affected frequency range is usually small except for the case of the broadest OSA configuration (modules of BW 2.5 GHz), and even in this case it's just one of the ends that is the main concern.

We will here present some plots that indicate the behaviour, but the main point that needs to be kept in mind can be stated in the following manner. In an unclipped 2.5 GHz OSA module, one edge can start to have noticable aliasing effects closer than 200 MHz to the edge, the location of which is determined by which subband is used for the tuning: in LSB it is the lower edge that is problematic, in USB it is the upper edge (in terms of frequency).[1] At 100 MHz from the edge a line aliased from the outside can

[1] In the 4 GHz mode, the relative location of the edge in question is the same for both modules. E.g. for an LSB tuning, it would correspond to the leftward edges of both red lines in Fig. 1.1, meaning that the upper module's less reliable lower edge will be within the centre overlap region and this should be taken into consideration if an attempt is made

appear at half its real strength, i.e. suppressed by only -3 dB. At 200 MHz from the edge, the aliasing suppression is roughly -20 dB which is an acceptable value. Further from the edge it is even better and not a concern.

If a user needs to use spectrometer bandpass closer than ∼40 MHz to the “good edge”, or closer than ∼200 MHz to the “bad edge”, we recommend that a test is conducted where a strong line is observed first well away from an edge, and then the tuning is shifted so that it moves to the position of interest in the near-edge bandpass to allow a strength comparison. Note that the change in aliasing suppression is not linear near the edge.


Reduction of signal strength near edges

Leakage of lines pivoted into band around the edges

Figure 2.1: Results from on-sky tests to determine aliasing effects in OSA 2.5 GHz spectra by simultaneously observing maser lines in SiO and CH3OH in the 43 GHz band, the latter of which is much weaker and hence the uncertainty in the correponding result is larger (comparison with an earlier spot-check suggests ∼10 percentage points error). The gap in the curve is due to a coincidence with an RFI. The worse performing edge is here the upper one since USB-tunings were used. The 4 GHz mode was employed so that the concurrent unaffected line strength could be retrieved from the other, overlapping, 2.5 GHz module where the relative line placement is different, see Fig. 1.1.

to stitch the two “halves” together to form a continuous 4 GHz spectrum.

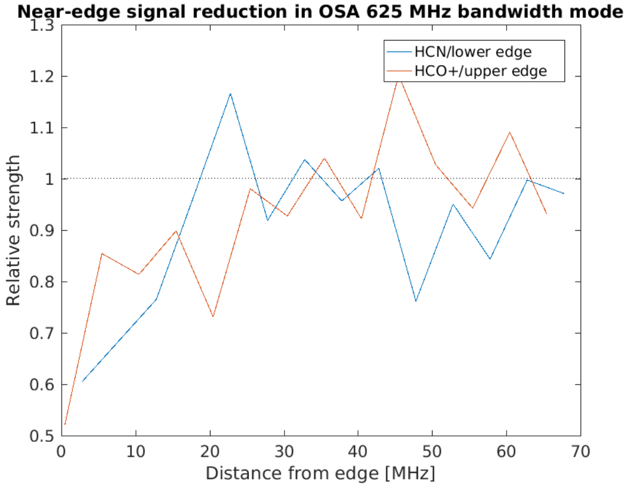


Figure 2.2: Test observation of Orion using the OSA 625 MHz bandwidth mode with the 3-mm receiver and LSB tunings near 89 GHz. Note that the unaffected line strength could not be measured simultanously here, and an average of observations just before and after the edge sweep was used for normalisation. This, in combination with unstable weather conditions, probably lead to an uncertainty of 15-20 percentage points. Additional test observations are planned, but at current time we recommend that science lines are not placed closer than 20 MHz to an edge in this mode without some initial spot-check test observation.

2.2 Centre channel spikes

All spectrometer products apart from the high resolution mode with the OSA (156/5 MHz/kHz BW/res) have a spike at the middle channel[2]. The spike can effectively be removed by selecting interpolation in the Bifrost spectrometer configuration tool. But the user should be aware that one channel (out of 32768 in the OSA) has in that case been set to the average of its immediate neighbours, it could in theory affect the line shape of a very narrow line. The spike in the high resolution mode of the SPE spectrometer (100/12 MHz/kHz BW/res) is unfortunately broader than one channel and we recommend that an offset tuning is used here to avoid placing a line in the middle of the spectrometer bandpass.

[2] If the automatic clipping function has been used it may no longer be located at the exact middle channel.

2.3 Known parasitic signals

There is an artifical line-like signal sometimes seen in one polarasation when the LO is the vicinity of 92 GHz. It moves in RF frequency when the LO is changed so interferences with science lines can be avoided by retuning slightly (an Offset can be plugged in in the tuning menu to still keep the same reference frequency).

We have recently (as of May 2020) discovered transient spurious signals in the range 76.4-76.7 GHz that *may* be due to RFI from anti-collision radar systems present in some modern vehicles but this remains to be clarified.

The 22/36/43 GHz receivers have a significant amount of RFI. One should always verify that the frequencies of interest are not affected by RFI at all possible source Doppler shifts.

2.4 Image sideband intrusions

The sideband suppression is usually quite good but there is always a risk of a false line appearing in the signal band that originates in the image band, or “opposite sideband”. A 2 K signal in the image band would for instance still appear with a strength of 100 mK in the signal band for a sideband suppression level of -13 dB. How to check what frequency in the image band a certain position in the signal band corresponds to, is discussed in Sect. 1.4.

2.5 Standing waves

The radome can cause fast standing waves, or ripples, in the spectra. It is most noticable in position switching mode, but starts to become noticable also when using beam-switching mode below ca 85 GHz. The ripple component with the longest period (ca 17 MHz) and the largest amplitude can be effectively removed by using the Path Length Modulator (PLM) in the observations (see the Bifrost manual on how it’s activited). When this option is used, the receiver is moved one wavelength along the telescope axis during each integration. The direction of motion is typically alternating and the speed of motion is automatically adjusted when the integration time is changed. The PLM is only available for the 3/4 mm receivers and will only be used for integration times equal to or larger than 30 seconds due to speed limitations in the mechanism. See the Figure below for an example of radome ripples as seen after Fourier transforming a spectrum, and the improvement when using the PLM. The ripple can be stronger at low elevations or in poor/unstable weather.

In deep observations towards the lower end of the 4 mm band, the residual ripples are often visually noticeble in the averages but their contribution to the RMS of the thermal noise is typically not as large as the human eye is led to believe. Nevertheless it is often feasible to try to identify spike features in Fourier transforms of the spectra, interpolate them out, and then inverse transforming back to normal spectra. Unless the real signals in the spectra have very particular properties, the spike interpolation in the Fourier domain should not affect the line shapes.

There are also two types of longer period standing waves that are sometimes seen in certain situations, with periods of 150 and 250 MHz, respectively. The former is associated with observing continuum sources and can be seen in deep beam-switched observations even if the antenna temperature of the source is as weak as a few tenths of a Kelvin. The latter is more elusive and is likely due to inescapable minor asymmetries in the beam switch mechanism. While the 250 MHz standing wave is relatively easy to see in single beam-switched observations, it is often fully cancelled out when using dual beam-switch. However, for unknown reasons, at certains times and at certain tuning frequencies, this standing wave does not fully cancel out and after an e.g. 12 h observation there could occasionally be a residual as high as 5 mK. When present, the amplitude can unfortunately vary over gigahertz scales which makes it more difficult to remove.

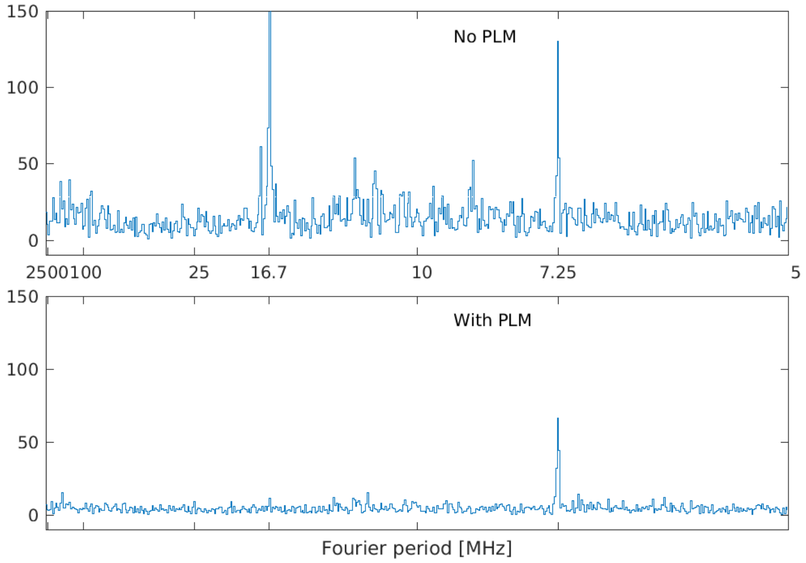


Figure 2.3: Fast Fourier Transforms of averaged spectra observed at 72.25 GHz. In the bottom example, the Path Length Modulator was used. Only the lowest 500 amplitude coefficients are shown. The channel count from left to right corresponds to the number of cycles that will fit within a bandwidth of 2.5 GHz but for simplicity the horizontal axis values noted have been converted to the corresponding cycle period in MHz. Note the observation with the PLM active was about 7 times longer than the 30 minute on-source integration time of the no-PLM data.

Chapter 3

Intensity calibration

3.1 Main beam efficiency

The model value for the main beam efficiency is listed in the FITS header keyword BEAMEFF. Note that it is elevation dependent for the 3- and 4-mm receivers so the conversion from T_{A} to T_{mb} scale should be done per individual spectrum for measurements that occur over a larger range of elevations (or use an average value appropriate for that range). The best performance at 86 GHz (η_{mb} ~57%) occurs at elevations around 65° and here the strongest near sidelobes are at a very low level, on the order of 2%. At lower elevations the sidelobe level increases as the main beam efficiency falls.

3.2 Other efficiencies

While the Moon efficiency has been measured to lie in the range 66-78% near 30° elevation at all frequencies in the 3- and 4-mm bands, these results are rather uncertain. In particular the values <70% were not consistent with the otherwise observed (and expected) trend with numbers improving with decreasing frequency. We currently recommend adopting a value of 73%.

The Moon efficiency *can* in theory be useful as an alternative to main

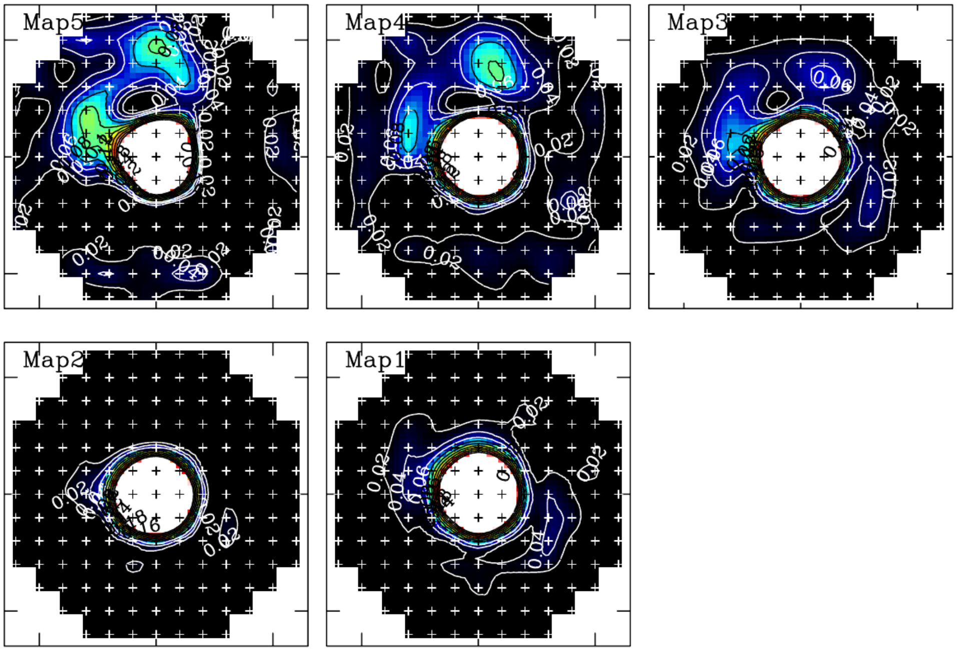


Figure 3.1: 86 GHz beam maps with the 3-mm receiver. The labels Map5 through Map1 corresponds to elevations 23, 34, 45, 60, and 65 degrees, respectively. The contour levels are relative to peak strength. Map2 is an experimental case which is not representative of standard observations (but may be so in the future).

beam efficiency for very extended emission where some part of the signal strength has been picked up by the the near sidelobe pattern and dividing by main beam efficiency will risk overestimating the average brightness temperature originating in the solid angle subtended only by the main beam. In practice, it is rarely used since interstellar radio emission is typically clumpy in nature and the distance to the edge of the emitting region for each position will play a major role on the appropriateness of applying the moon efficiency. Without advanced deconvolution algorithms, there is no simple solution to this problem even if a full extended 2D beam pattern were to be provided.

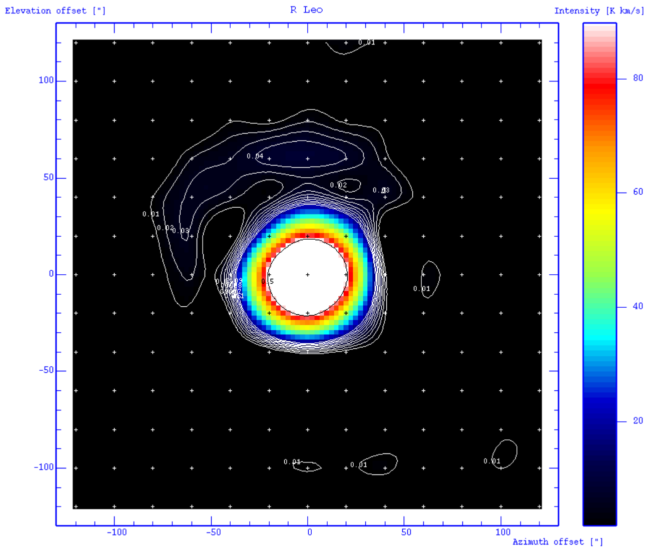


Figure 3.2: Example 86 GHz beam map at elevation 44° with the 4-mm receiver.

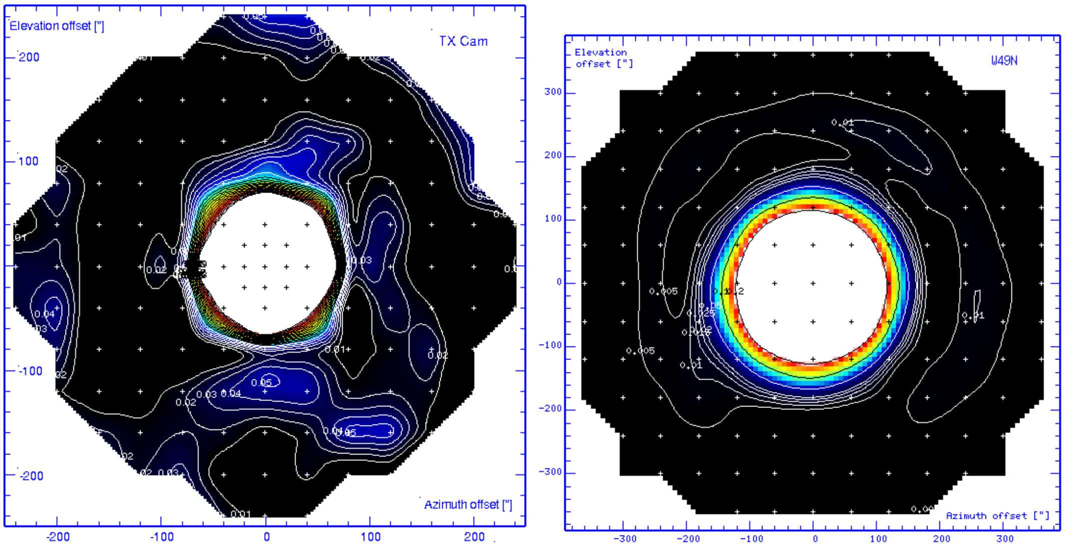


Figure 3.3: Example beam maps with the 43 and 22 GHz receivers.

Chapter 4

Data logistics

4.1 Format

All scientific and technical integrations are saved in single spectrum FITS files with no extension tables. Thus each spectrum has its own separate ASCII header at the beginning of the file where all meta data are given.

Calibrated integrations per polarisation are always saved, this can be considered the most basic data product[1].

Averages between polarisations for each integration can also be saved. This is recommended and in fact the default setup in an observation script, see the Bifrost manual. The weighting takes into account possible differences in T_{sys} between the polars.

The system can also produce continuously updated grand averages over many observations if requested to do so in the setup. The noise has here been accounted for through the radiometer formula. Note however that these are simple channelwise averages and there is no consideration for the possible frequency alignment effects (Sect. 1.3), nor for possible drifting of line positions in other-sideband data (Sect. 1.5). Also, these averages are made in the T_{A} domain and conversions to T_{mb} should best be done per

[1] Raw total power spectra from each phase in switched observations can also be saved if setup to do so in the observation script but will generate large data quantities. We do not recommend to use this unless in special cases.

spectrum *before* any averaging for the 3- and 4-mm receivers (Sect. 3.1).

4.2 How the data files are sorted in a project's home directory

There are plenty of different data file types sorted in different ways produced by the Bifrost control system. While this may seem confusing at first, there is a logic to it aimed at relieving the user from having to keep track of which scan number corresponds to which type of measurement.

Science data can have names in the forms
`<scan-number>d/a/D/A/dA/aA/DA/AA.fits`
whose meanings are all explained below.

If a project is called O2018b-99 and certain data were observed on date 2019-01-23, all the basic data products will be located under
`/projects/20m/2018b-99/Data/2019-01-23/`
Individual science integrations per polarisation will be stored in
`/projects/20m/2018b-99/Data/2019-01-23/Data/`
and will be called e.g. 12345678d.fits if they are Single observations, and 12345678D.fits if it was a spectrum observed as part of a Map.

Science integrations averaged over polarisations will be located in
`/projects/20m/2018b-99/Data/2019-01-23/Coadded/`
and will be called e.g. 12345678a.fits if they are Single observation, and 12345678A.fits if it was a map spectrum.

The system averages described in Sect. 4.1 are located in
`/projects/20m/2018b-99/Data/Averages/`
where Single type observation averages are called
`12345678dA.fits`
per polarisation, and
`12345678aA.fits`
for polarisations combined.

Map observation averages per position are called
`12345678DA.fits`
per polarisation, and

`12345678AA.fits`
for polarisations combined.

4.3 Technical data files

Files pertaining to such measurements as calibrations, pointings, and focussings are found in dedicated directories under
`/projects/20m/<projecthome>/Data/<obsdate>/`
Calibrations ($T_{\rm sys}$ profiles) have file names ending in c.fits. Pointings and focussing files have descriptive file name endings such as pl, pr, pc, pu, pd, where the p indicates that it is per individual polarisation, and l, r, c, u, d stand for left, right, centre, up, down. If the p is replaced by an a, it indicates that it is the average of two polarisations. Focussing files follow a similar pattern but have i, m, o as positional indicators (inner, middle, outer).

4.4 Other files

There are a number of subdirectories under the project home, e.g. `Catalogs/` for .cat-files which contain the source definitions in terms of coordinates and velocities. Other examples are `Configs/`, `Logs/`, and `Diaries/`, the meaning of which are explained in the Bifrost control system manual and help functions.

4.5 Misc. notes on reduction software

While there already is some embedded information and advice about reduction software related to particular subjects elsewhere in this document, we use this last Section to gather various other useful tips, listed in no particular order.

Information on how to install XS for Linux can be found here:

`https://nain.oso.chalmers.se/oso/ubuntu/README.txt`
This is now the recommended version to use since compatibility issues have arisen when running on Mac computers[2]. There is no Windows version at all.

Mac versions may still run but suffer from graphics window update issues to the point where it is almost not usable at all. Contact Onsala staff for instructions if you nevertheless wish obtain the source code and attempt your own compilation. The formerly recommended option to download source code and compile from
`ftp://yggdrasil.oso.chalmers.se/pub/xs/` by following the instructions in the README file *might* still be possible but this repository is currently not updated with the latest versions and may no longer be maintained at all in the future.

Mac users should however be able to run the Linux version in a virtual machine (VM) environment. For instance, the following setup has been shown to work ok: macOS Mojave (10.14.6) with VirtualBox (Oracle) Version 6.0.18 r136238 (Qt5.6.3) with Ubuntu 18.04.3 LTS. No special requirements were needed to install the VM on Mac, or Ubuntu on the VM, it was sufficient to follow the standard procedures given by their corresponding suppliers. To install a Linux based system on the VM you will need to download an image of the distro you want, and create a Live-CD/USB. Note that this has been tested on Ubuntu but should work on any Debian based distro, and most likely on Red Hat (the syntax will however be different since Debian uses APT and Red Hat uses RPM). For Debian-based systems, the installation instructions for XS are then the same as those given in the first URL above (the `nain` server). A separate one-page PDF document is available upon request that contains the above information about VM on Mac plus the Ubuntu instructions on `nain`.

If the installation of the Linux version fails or get stuck for some reason (whether it's for a proper Linux box or a Mac VM), you might need to do the following before trying again (by starting from the
`sudo apt-get update` command):

```
# Identify PID of the stuck process and kill it interactively:
```

[2] These problems likely originate in incompatibilities between Motif (a graphics package that XS is using for GUI functions) and newer versions (the issues started approximately in 2016) of X system libraries.

```
sudo fuser -vki /var/lib/dpkg/lock
# Remove the file:
sudo rm -f /var/lib/dpkg/lock
# Repair any broken packages in dpkg:
sudo dpkg --configure -a
```

Lastly, there is a separate package on `nain` called 'xs-data' that has some useful frequency files for XS allowing you to place markers at catalogued line position from several sources. If `nain` has already been added as repository according to the README.txt file, xs-data can be installed by `sudo apt-get install xs-data`.

Note that the CLASS software is adding the values of CRVALn (usually CRVAL1 for the f-axis) and the RESTFREQ keywords when reading FITS files while XS saves spectra where both keywords contain the same value[3]. Thus a spectrum saved by XS will get a twice too high frequency axis when loaded with CLASS. The unaltered data files from the 20m telescope however will not cause this effect since CRVAL1 is set to zero.
If you need to import a FITS file saved by XS in CLASS while still obtaining a correct frequency axis, you can do so after manually setting the corresponding CRVAL header value to 0.0.

There is also an older package that can be used for rudimentary data reduction using executables run in Shell scripts called DRP. It has not been maintained officially for a long time but there was a fairly recent effort to update the code to at least seemingly run correctly in simple test cases on new Linux systems and hardware. This – highly non-official version – is available upon request without any guarantee of functionality nor support.

Python is a popular programming language which has some support for handling FITS files when using the astropy and specutils

[3] RESTFREQ is only supposed to be the 'frequency of interest' whereas CRVAL is the actual reference pixel value that helps define the axis according to the official FITS standard recommendation so XS is not wrong in doing this.

(https://github.com/astropy/specutils) packages. Below follows some example code that may help as a foundation for anyone wanting to write reduction scripts in Python.

```
import numpy as np
from matplotlib import pyplot as plt
import astropy.wcs as fitswcs
from astropy.io import fits
from astropy import units as u
from specutils import Spectrum1D
import glob

# create a figure and one subplot
fig, ax = plt.subplots()
# use interactively, else one needs to close every plot before the
# next one is shown
plt.ion()

# get a list of fits files from subdirectory
fitsfiles = glob.glob("FITS/*.fits")
for fitsfile in fitsfiles:
    print(fitsfile)
    f = fits.open(fitsfile)
    # get the FITS header and data array
    header = f[0].header
    data = f[0].data
    f.close()

    # build a WCS structure (frequency axis) from FITS header info
    crval = header[’CRVAL1’]
    cdelt = header[’CDELT1’]
    crpix = header[’CRPIX1’]
    restf = header[’RESTFREQ’]
    wcs = fitswcs.WCS(header={’CDELT1’: cdelt, ’CRVAL1’: crval,
                          ’CUNIT1’: ’Hz’, ’CTYPE1’: ’FREQ’,
                          ’RESTFRQ’: restf, ’CRPIX1’: crpix})

    # reshape the data into one dimensional array,
    # assumes that frequency axis is NAXIS1
    # and assumes that all other axes are of length 1
    nchan = header['NAXIS1']
    specdata = data.reshape(nchan)
    # create vector of frequencies from WCS object
    freqdata = wcs.wcs_pix2world(np.arange(nchan), 0)[0]
    # create a Spectrum1D with frequencies in MHz, flux in K
    spec1d = Spectrum1D(spectral_axis=(freqdata*u.Hz).to(u.MHz),
flux=specdata*u.K)

    # clear previous plot and plot new spectrum
    plt.cla()
    ax.plot(spec1d.spectral_axis, spec1d.flux)
    ax.set_xlabel("Frequency [MHz]")
    ax.set_ylabel("Intensity [K]")
    plt.show(block=False)
    plt.pause(0.001)
```

A spectrum can be read and plotted with frequency axis in MatLab in the following manner:

```
fil='myfitsdir/1234567a.fits';
data=fitsread(fil);              %% Actual data vector
meta=fitsinfo(fil);              %% Meta data struct
cards=meta.PrimaryData.Keywords;  %% Cell array of FITS keywords
%% If you want to see all keywords, just type "cards"
%% or "fitsdisp(fil,'Mode','full')"
nchan=cards{find(strncmp('NAXIS1',cards(:,1),8)),2};
frest=cards{find(strncmp('RESTFREQ',cards(:,1),8)),2};
%% Note that "RESTFREQ" in the line above should be replaced
%% with "CRVAL1" if you read a spectrum saved by XS
fdel=cards{find(strncmp('CDELT1',cards(:,1),8)),2};
refpix=cards{find(strncmp('CRPIX1',cards(:,1),8)),2};
source=cards{find(strncmp('OBJECT',cards(:,1),8)),2};
fax=frest+fdel*([1:nchan]-refpix); %% f-axis in Hz
plot(fax,data,'k'); title(['Spectrum of ' source]);
```