

# On the calculation of observing time with APEX in the spectral line OTF mode and the optimization of OTF parameters

L. Pagani, A. Belloche  
Observatoire de Paris, MPIfR Bonn

June 1, 2021

## Abstract

Running the latest version of the APEX calculator (V9.2)<sup>1</sup>, we found some discrepancies and some information which is missing that could help the user to evaluate correctly the observing time her/his maps will need. We also demonstrate that it is inadequate to use CLASS\xy\_map blindly in the observing configuration APEX has chosen because of beat problems. For a map sampling of one third of the beam, we recommend to use a xy\_map convolution kernel of FWHM equal to one half of the beam in order to produce maps with more uniform noise level for any scanning direction and a reasonably good noise decrease (factor  $\sim 2$ ) at the expense of a limited resolution degradation (+12%). Alternatively, if getting the highest angular resolution is a priority and increasing the data volume is not an issue, then we recommend to perform the OTF maps with a sampling of one fourth of the beam while keeping the default parameters of xy\_map.

## Presentation

Preparing together a proposal, we compared the merits of OTF and ON/OFF modes with APEX. We wanted to map a  $54'' \times 54''$  region at 691 GHz, i.e. with a  $9''$  beam down to 50 mK rms at  $60^\circ$  elevation, PWV = 0.5 mm and with a resolution of  $0.1 \text{ km s}^{-1}$  using a dump time of 1 sec.

The OTF calculator said :

```
We have used the OTF observing time calculator at APEX V9.2 to estimate the total time needed
to achieve our goal. We plan to do an OTF of 54 x 54 arcsec and for the calculation we assume
a dumptime of 1 seconds and a sampling corresponding to 1/3 of the beam. Using SEPIA660
tuned to 691 GHz in the LSB, selecting a spectral resolution of 0.1 km/s and assuming a typi-
cal source elevation of 60 deg and a typical PWV of 0.5 mm, we could get down to a noise of
50 mK[Ta*] in 22.03 hours (including telescope and calibration overheads).
```

Other informations provided by the OTC :

- Time per submap : 35.9 s
- Number of calibrations per coverage : 65
- Number of submaps : 9
- $T_{\text{sys}}$  : 762 K
- Rows per off (*n<sub>on2off</sub>*): 2
- Scanning speed :  $3''/\text{s}$
- Number of coverages : 93
- Sigma reached after 1 coverage : 481 mK, after 93 : 49.9 mK
- On source time : 500.5 min
- Off source time : 144.7 min
- Overhead time : 676.4 min.

---

<sup>1</sup><http://www.apex-telescope.org/heterodyne/calculator/ns/otf/index.php>

## Analysis

First, note that the number of calibrations is huge (almost 7 per submap) and therefore completely erroneous but this is not a problem we are concerned with presently. We just mention it for you to cure it.

### number of steps

The beam size being  $9''$ , the beam step is  $3''$ . We suppose horizontal axis is the sweeping axis, and vertical axis is the shift axis between rows (x and y, resp.). While we can understand that  $54''$  represents 18 pixels of  $3''$  (the other solution could be to have 19 pixels spaced by  $3''$ ), we see no reason why the  $54''$  along the shift axis should not be swept 19 times because only the 19th row is  $54''$  away from the initial row. The OTC proposes 9 submaps of 2 rows, that is 18 rows. One is missing.

Looking at the old help page about OTF<sup>2</sup>, the number of submaps given by eq. 10 on that webpage is

$$n_{submap} = \frac{L_y}{\Delta y \times n_{on2off}} + 1$$

Here :

$$n_{submap} = \frac{54''}{3'' \times 2} + 1 = 10$$

Therefore we agree with the webpage eq.10 but not with the OTC which returns 9 submaps instead.

### noise estimate

We agree with the equations 1 to 7 of the webpage which lead to the single dump noise of 1.176 K in our case :

$$t_{off} = t_{on} \times \sqrt{(3 \times n_{dumps} \times n_{on2off})} = 1 \times \sqrt{3 \times 18 \times 2} = 10.39s$$

$$\sigma^1 = \frac{T_{sys}}{\sqrt{2\Delta\nu}} \sqrt{\frac{1}{t_{on}} + \frac{1}{t_{off}}} = \frac{762}{\sqrt{2 \times 2.3 \times 10^5}} \sqrt{\frac{1}{1} + \frac{1}{10.39}} = 1.176K$$

neglecting backend efficiency but taking into account the presence of two polarizations averaged together (hence a  $\sqrt{2}$  in the denominator)<sup>3</sup>. The velocity resolution of  $0.1 \text{ km s}^{-1}$  corresponds to 230 kHz at 690 GHz. However, we totally disagree on the following equation (eq. 8) which states that

$$n_{cover} = \left[ \frac{\sigma^1}{\sigma} \right]^2$$

Past the confusion that the  $\sigma$  of eq. 2 is the same as  $\sigma^1$  of eq. 8 and is different from the  $\sigma$  of the same eq. 8 in the webpage, the computation here gives

$$n_{cover} = \left[ \frac{1.176}{0.05} \right]^2 = 554$$

Instead the OTC introduces a noise number for a single map of 0.48 K :

$$n_{cover} = \left[ \frac{0.48}{0.05} \right]^2 = 93$$

The question becomes "where does this 0.48 K figure come from and why is it not in concordance with the webpage explanations ?"

Another (very small) problem is the fact that the total ON+OFF time is slightly smaller than what it should. 36 dumps of 1 s (in 35.9 s !!), off time is 10.39, 9 submaps, therefore the total map time should be  $9 \times (36 + 10.4) = 417.6 \text{ s}$ . Repeated 93 times, one finds 647.3 min while the OTF calculator says  $500.5 + 144.7 = 645.2 \text{ min}$ . About 2 minutes are lost. If we use 35.9 s instead of 36 s we get almost the OTC time:  $9 \times (35.9 + 10.4) \times 93 = 645.9$ . The remaining difference is probably due to some round-off in our numbers but the main difference is due to the strange result that  $36 \times 1 = 35.9$ .

<sup>2</sup>[http://www.apex-telescope.org/heterodyne/calculator/otf/inf\\_new4.php](http://www.apex-telescope.org/heterodyne/calculator/otf/inf_new4.php)

<sup>3</sup>the presence of the two polarizations is not indicated in the calculations, which can lead to misunderstandings for the user

## is CLASS the explanation ?

We therefore thought that this 0.48 K was the noise of a single pixel after processing the data with CLASS following the recipe given in the webpage.

```
table mydata new
xy_map mydata
let name mydata
let type lmv
go noise
```

but here we enter a new set of problems. To test the reduction of noise from the raw OTF map to the xy\_map, we produced a  $19 \times 20$  pixel map with noise only, i.e. no fake signal, produced by the internal noise generator of CLASS, with  $3''$  spacings. Noise was set to match the calculated noise for the particular OTF mode requested here. We produced a OFF spectrum of noise

$$\sigma_{off} = \frac{T_{sys}}{\sqrt{2\Delta\nu t_{off}}} = \frac{762}{\sqrt{2 \times 2.3 \times 10^5 \times 10.67}} = 0.344K$$

which we subtracted to every 38 spectra spread on two consecutive rows for which we produced 38 realizations of the expected ON noise (38, instead of 36 with the OTC, makes the OFF time lasts 10.67s instead of 10.39s)<sup>4</sup>:

$$\sigma_{on} = \frac{T_{sys}}{\sqrt{2\Delta\nu t_{on}}} = \frac{762}{\sqrt{2 \times 2.3 \times 10^5 \times 1}} = 1.124K$$

The expected total noise for each spectrum is therefore

$$\sigma = \sqrt{\sigma_{on}^2 + \sigma_{off}^2} = \sqrt{1.124^2 + 0.344^2} = 1.175K$$

slightly less than the 1.176 K from the OTC because we used 19 rows instead of 18. The difference is too small to warrant repeating the whole process.

We get the noise distribution as displayed in Fig. 1a.

If we plot the noise of individual pixels from the OTF file directly, we get a noise map (Fig. 1b) which is quite random, as expected. Now we process these spectra following the recipe above (convolution kernel of beam\_size/3, regridding by half beam steps), and get this information from CLASS :

```
[...]
I-XY_MAP, Field of View:      -81.8" x      86.3"
Pixel size:      -4.5" x      4.5"
Spatial resolution:      9.6"
Telescope Beam:      9.1" (from table header)
I-XY_MAP, Creating file: simu_otf_C065_9asec.lmv
I-XY_MAP, Creating file: simu_otf_C065_9asec.wei
LAS> let name simu_otf_C065_9asec
LAS> let type lmv
LAS> go noise
I-GO, Changing to new (or updated) file  simu_otf_C065_9asec.lmv
W-CLEAR, CLEAR ALPHA is obsolete
MAX_NOISE      =      1.055768      ! Real      LOCAL  2
NOISE          =      0.8109840      ! Real      GLOBAL
```

---

<sup>4</sup>Originally, we wanted to produce a map that would have been what we expected from our requests, i.e.  $19 \times 19$  pixels. We did not think necessary to repeat the study for a  $18 \times 18$  map to exactly match the OTC. The results are close enough to be considered equivalent.

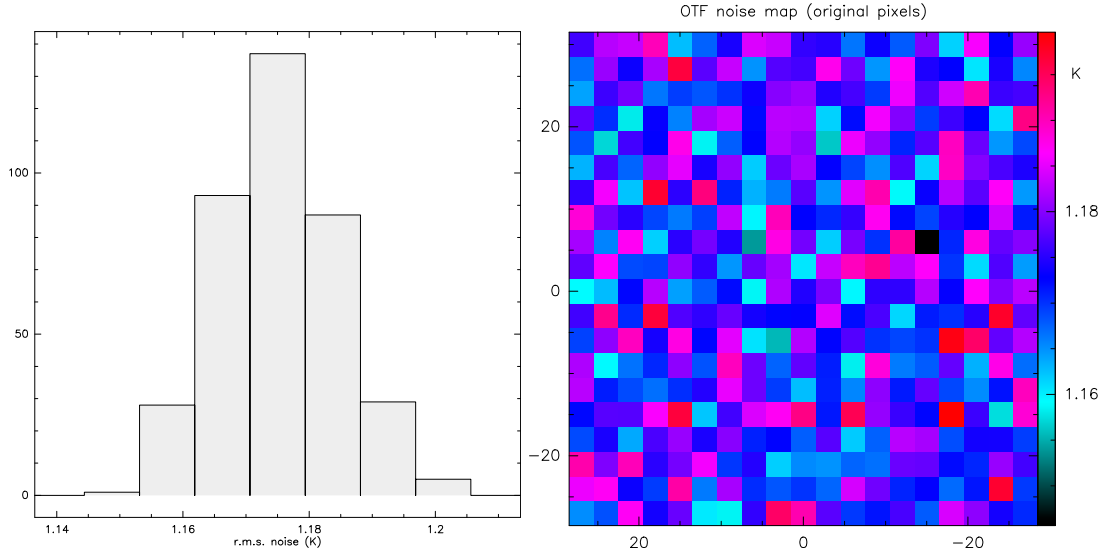


Figure 1: **left - a** Noise distribution of the 380 synthetic spectra generated by CLASS. **right - b** Original spectral noise per pixel from the simulated OTF observations.

The noise from CLASS estimate is 0.81 K and not the announced 0.481 K and we would need

$$n_{cover} = \left[ \frac{0.81}{0.05} \right]^2 = 263$$

repeats to get the noise down to 50 mK instead of 93 from the OTC. So CLASS does not seem to give the same answer as the OTC, however, it appears that this specific use of `xy_map` in CLASS is problematic. First, if we measure the noise spectrally pixel per pixel and build an histogram, we get Fig. 2a.

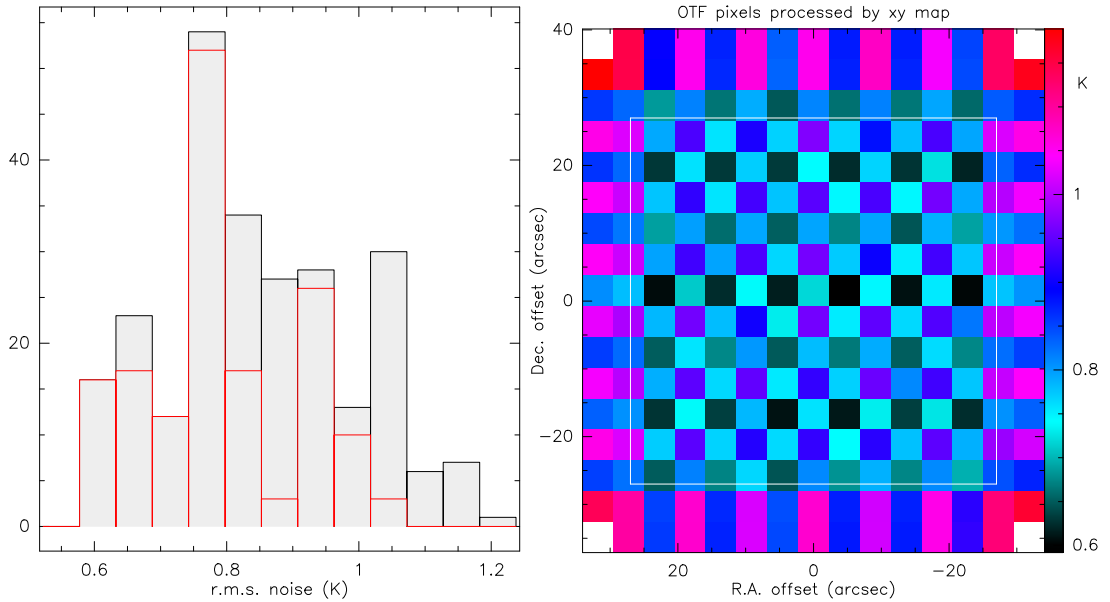


Figure 2: **left - a** Histogram of the noise in the 251 pixels issued from the `xy_map` default convolution and regridding of the data. The red histogram is obtained by excluding pixels outside the original region (white box in Fig. 2b). **right - b** Noise from the OTF map after convolution by 1/3 beam kernel and regridding by half beam steps. The white box traces the original  $54'' \times 54''$ .

One can see that the noise is strangely distributed with a minimum of 0.59 K (0.81 K is the median of the range of values which is not forcibly representative of what happens in the useful part of the map, GO NOISE should be used with caution)<sup>5</sup>. We should therefore check the noise map again (Fig. 2b): a regular pattern

<sup>5</sup>GO NOISE measures the noise over the map per channel while we measure the noise per pixel over the frequency. If the

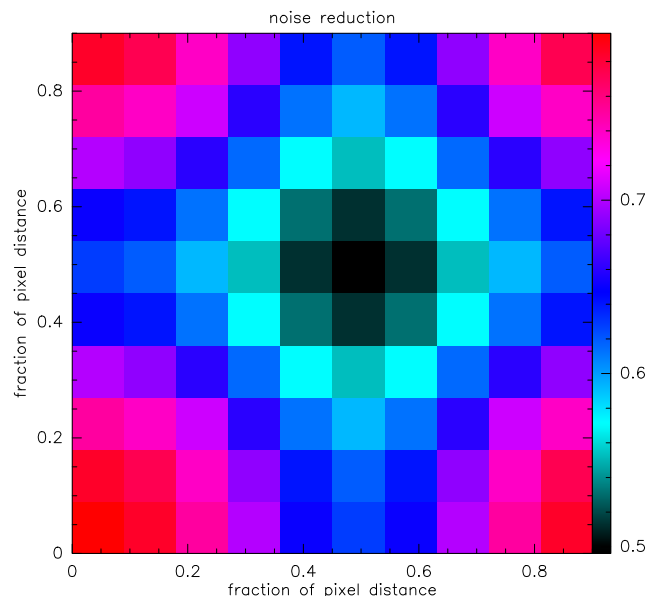


Figure 3: Noise reduction for a kernel of size  $1/3^{rd}$  of a beam applied to pixels regularly spaced by  $1/3^{rd}$  of a beam as a function of the distance of the center of the kernel to the reference pixel center.

of noise appears! This beating effect is due to the use of OTF dumps of  $1/3^{rd}$  beam steps with a regridding of  $1/2$  beam steps computed by convolving the original data with a Gaussian kernel of FWHM  $1/3^{rd}$  beam.

What happens is that depending on the position of the pixels with respect to the kernel, the averaging can diminish the noise from 0.8 when the kernel is coincidental with one pixel to 0.5 when the kernel is in the middle of 4 pixels (Fig. 3). In our test case with an input noise of 1.175 K, this means a convolved pixel noise between 0.58 and 0.94 K, if the noise of all pixels is independent in between them. The predictions remain quite similar to the range of values in the inner part of Fig. 2, despite their common OFF position, because only a few spectra are included, therefore the OFF noise contribution remains small. This also implies that the resolution varies depending on the position of the pixels during convolution. However, the spatial resolution change is not very large (in the present case, it is probably between  $9.6''$  and  $10.2''$ ).

It is clear to us however that the default parameters of `xy_map` should never be used with the standard OTF observing mode ( $1/3^{rd}$  beam step) at APEX mostly because we end up with a map of slightly fluctuating resolution and strongly fluctuating noise.

One could contest the fact that the sampling is never that regular in real life and that could solve the beating problem. We got two small maps from APEX OTF that do show regularity but our experience with the 30-m OTF observations is that each direction is very regular but both directions have a small jitter. We have therefore "moved" every other line by  $1''$  and recomputed the histogram. GO NOISE indicates a similar noise (0.83 K) while the histogram looks marginally better but not the noise map (Fig. 4). In both cases, the weight map produced by `xy_map` shows clearly the beating pattern too.

## Solutions

### First solution : avoid the regridding ?

To avoid the beating between  $1/3$  beam dumps and kernel, and  $1/2$  beam gridding, one can ask `xy_map` to keep the original gridding <sup>6</sup>:

```
let map%cell -3 3 !note the - for the first axis to keep it increasing to the left
```

number of pixels is large enough and in absence of signal, the results should be comparable if the noise is uniform over the map (GO NOISE clips the results between  $\pm 2.5\sigma$  to eliminate the signal but GO NOISE was developed to measure noise in interferometer maps, not convolved maps from single-dish telescopes)

<sup>6</sup>however the pixels are not forcibly placed at their original position, only the spacing is the same, but the result is not very different because `xy_map` tries to keep the pixels in a sensible placing with respect to the original ones. It is possible to force `xy_map` to compute the kernel exactly on the original grid with the `map%like` option

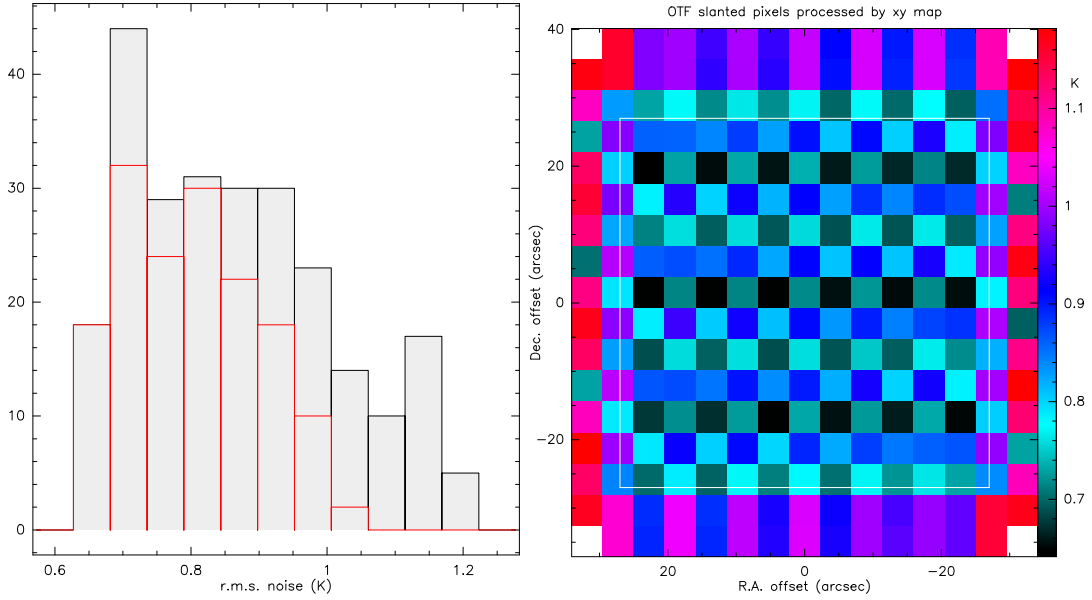


Figure 4: **left - a** Histogram of the noise from the whole 251 half-beam step pixels from the jittered OTF map, red histogram for the pixels inside the white box (original region). **right - b** same as Fig. 2b for jittered pixel distribution

xy\_map my\_data my\_data\_onethird

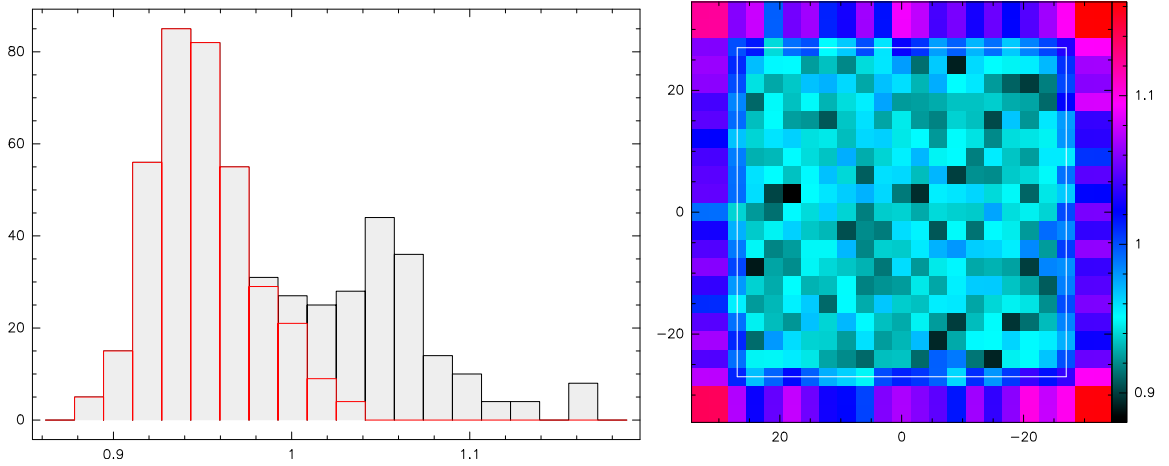


Figure 5: **left - a** Histogram of the noise for the xy\_map processed data with identical regridding (1 pixel every  $1/3^d$  of a beam), red histogram for the pixels inside the white box (original region). **right - b** Corresponding map of the noise (same as Fig. 2b).

Now the noise looks much better in the histogram and in the noise map (Fig. 5). The weight map is now uniform. GO NOISE indicates a noise of 0.971 K while the noise in the central region traced by the red histogram in Fig. 5a is centered on 0.95 K which is identical to what one can get by hand. We therefore definitely cannot match the 0.48 K found by the OTC and in reality we would need

$$n_{cover} = \left[ \frac{0.95}{0.05} \right]^2 = 361 \text{ repeats}$$

to reach the desired sigma (50 mK). This represents 42 hours (2527 min) without the overheads instead of the 645 min indicated by the OTC.

## Second solution : increase the kernel size

If instead of using a convolution kernel of beam/3 size, one uses a kernel of beam/2 size, the final beam size is slightly more degraded (from 9.1'' to 9.6'' originally, to 10.2'' presently)<sup>7</sup> but the beating pattern has disappeared (except for the central  $-\Delta\delta = 0$  - scan which seems slightly better in terms of noise, by chance in principle, since the weight map is uniform presently in the inner white box), Fig. 6b). GO NOISE indicates 0.64 K while the red histogram (Fig. 6a) has a median value of 0.565 K (remember that GO NOISE includes the fake pixels outside the white box).

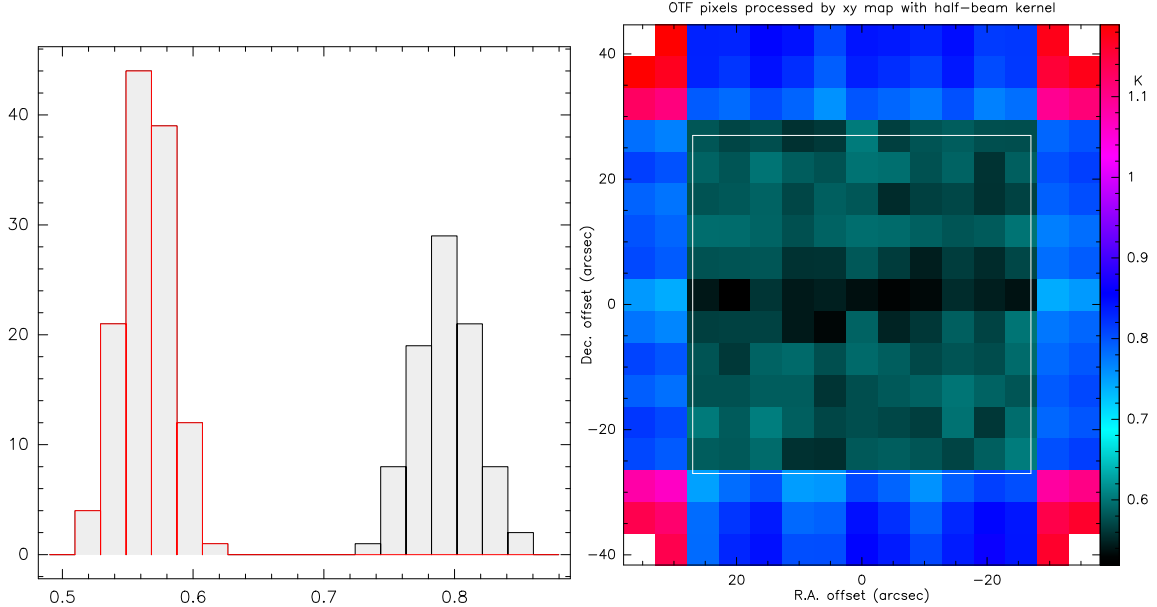


Figure 6: **left - a** Histogram of the noise after a convolution with a half-beam kernel and regridding on a half-beam spacing. The red part only contains those pixels in the white square on the noise map on the right. **right - b** Corresponding map of the noise (same as Fig. 2b).

Now the number of repeats is

$$n_{cover} = \left[ \frac{0.565}{0.05} \right]^2 = 128 \text{ repeats}$$

to reach the desired sigma (50 mK). This is closer to the OTC figure of 93 and the time is now 14.9 hours (896 min) without the overheads and with a minimal degradation of the spatial resolution. Checking the noise reduction as a function of the position of the kernel with respect to the pixels (as in Fig. 3), one finds a variation between 0.43 and 0.46 times the original noise (we measure 0.48 though in our test case because of the correlated OFF noise). To ask `xy_map` to use a kernel of 1/2 beam, one simply needs to set `map%reso` to the desired final beam size. For a kernel of 1/2 beam this is simply:

```
let map%reso = beam*sqrt(1.25)
```

where beam is to be replaced by the size returned by the command `TABLE` run just before (`sd_beam`):

```
[...]
I-TABLE, freq: 691.66048 GHz   sd_beam: 9.089 arcsec
I-OBSERVATORY, Tel: APEX      -67:45:32.90 -23:00:20.8 Alt.   5104.0 Diam   12.0
```

## Third solution : increase the dumps per beam to 4

If now, we dump the data every 1/4<sup>th</sup> beam instead of every 1/3<sup>rd</sup> but without changing the total time ON source to cover the map, we have the following numbers (we keep the wrong estimate of  $n_{rows}$  from the OTC to be comparable):

<sup>7</sup>these are quite theoretical values since the beam has been elongated by 1/3<sup>rd</sup> of its size along the scanning direction

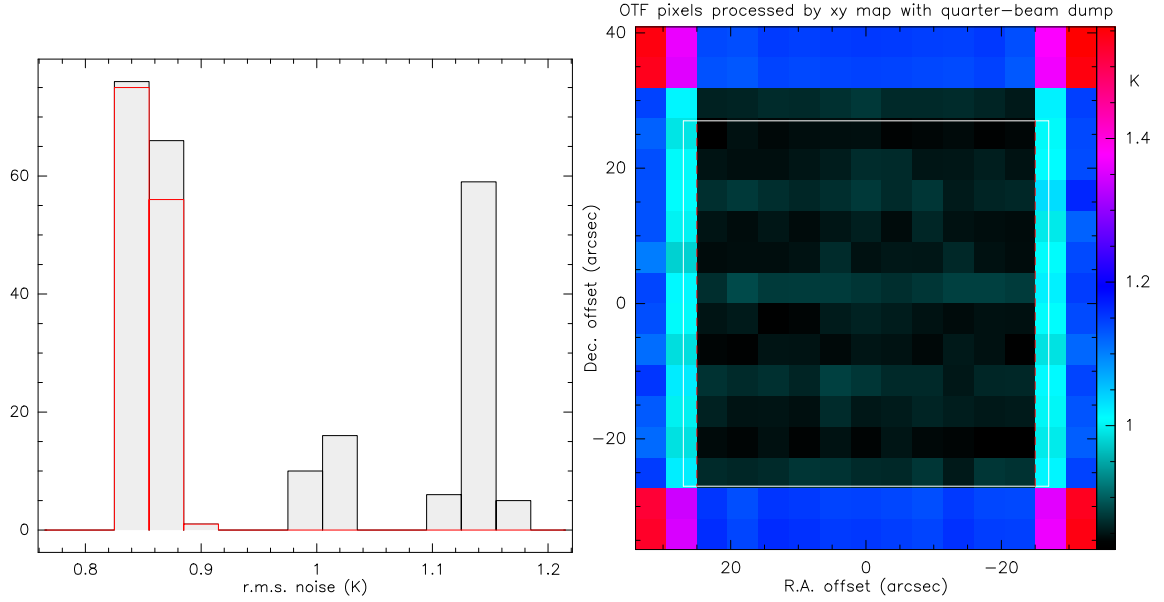


Figure 7: **left - a** Histogram of the noise after a convolution with a third-beam kernel and regridding on a half-beam spacing of pixels sampled every quarter-beam. The red part only contains those pixels in the white square on the noise map on the right. **right - b** Corresponding map of the noise (same as Fig. 2b).

$$\Delta x = \frac{beam}{4} = \frac{9''}{4} = 2.25''$$

$$n_{dump} = \frac{54''}{2.25''} = 24$$

$$n_{rows} = n_{dump} = 24$$

$$n_{pixels} = 24 \times 24 = 576$$

$$t_{on} = \frac{36 \times 9}{576} = 0.5625 \text{ s}$$

$$n_{on2off} = 3$$

$$t_{submap} = 72 \times 0.5625 = 40.5 \text{ s}$$

$$n_{submaps} = 8$$

$$t_{off} = 0.5625 \sqrt{4 \times 24 \times 3} = 9.55 \text{ s}$$

$$\sigma_{on} = \frac{762}{\sqrt{2 \times 2.3 \times 10^5}} \frac{1}{\sqrt{0.5625}} = 1.498 \text{ K}$$

$$\sigma_{off} = \frac{762}{\sqrt{2 \times 2.3 \times 10^5}} \frac{1}{\sqrt{9.55}} = 0.364 \text{ K}$$

$$\sigma_{dump} = \sqrt{\sigma_{on}^2 + \sigma_{off}^2} = \sqrt{1.498^2 + 0.364^2} = 1.542 \text{ K}$$

The total time for one map is  $8 \times (40.5 + 9.55) = 400.4 \text{ s}$  (without overheads, to be compared with the original 417.6 s). After processing with `xy_map` with 1/3 beam regridding (as in the first solution), GO NOISE gives 0.936 K, while central pixels are around 0.85 K (Fig. 7), similar to the result computed by hand (0.855 K). This faster dumping 1) saves time (because the OFFs are shorter and less numerous) and 2) gets lower noise than the default dumping spacing. We get

$$\left(\frac{0.95}{0.85}\right)^2 \times \left(\frac{418}{400}\right) = 1.30$$

times faster. It also provides beams slightly less elongated. These advantages are however obtained only at maximum resolution. Note also that we have not tried to optimize the number of ONs per OFF but tried



to stay within the 30–60 s stability region.  $n_{on2off} = 4$  is also a possibility, giving 6 submaps of 54 (on) + 11 (off) = 65 s each, and 390 s total for one map. The noise would be higher though since slightly less total time is spent on the map and the OFF spectrum being common to more pixels, the convolution would be less efficient at decreasing the noise.

Since the default xy\_map convolution is not improving very much the noise when used consistently, we can again degrade slightly the spatial resolution by convolving with a 4.5'' kernel. However, the advantage of the higher number of spectra disappears and we end up with the same numbers for the noise (0.58 K) modulo the observing time (0.58 K for 400 s, 0.56 K for 418 s).

## where does the 0.48 K come from ?

To obtain such a low number, for a single map, one needs to convolve the data with a 5.6'' kernel in the present case ( $\sim 0.6$  beam), leading to a final beam size of  $\sqrt{5.6^2 + 9.1^2} = 10.7''$ . This is possibly an acceptable beam size degradation but this is inconsistent with the OTC. Having obtained the code behind the OTC webpage, one sees that the OTC uses a kind of ad-hoc estimate of the number of spectra in one half-beam (which is the regridding step but has nothing to do with the default kernel used by xy\_map) that ends up with 6 spectra flatly averaged. In our case,  $1.175 \text{ K} / \sqrt{6} = 0.480$  indeed. The flat weight of 6 has no justification and no relation with the gaussian weight of 1.26 of the xy\_map kernel. Even including the correction factor of 1.3 in time (1.14 in noise) invoked in the web help page from the comparison with real observations, one cannot reconcile the two.

## comparison with APEX observations

Since the webpage claims good coincidence with the observations, we have requested a set of APEX OTF observations. Two such maps (2020, scans 16383 and 16415) have been given to us, with the commands used to run the observations. We have injected the parameters in the OTF OTC calculator and we still find the same discrepancy factor of 2 as explained above when one keeps the regridding so that the resolution does not change (first solution). Therefore we still disagree with the OTC by a factor of 4 in time with respect to its predictions (we ignore the overheads, considering that they should have no impact on the resulting noise).

## ON/OFF full beam raster map

Since the beam is not a Dirac function but a gaussian, the contrast of the small details one can obtain by doing Nyquist sampling (half-beam sampling) instead of a simple full-beam sampling is very low and these details are lost if the map is not of very high signal/noise ratio. It is not our purpose to demonstrate this fact here but simply to ask whether it is better or not to do an OTF map or an ON/OFF map when sensitivity is an issue. In our present case, to map a  $54'' \times 54''$  area with a  $9''$  beam needs 36 pointings. To reach 50 mK rms with the same conditions as given above, one needs (supposing  $t_{on} = t_{off}$ ):

$$\sigma_{pixel} = \frac{2 \times T_{sys}}{\sqrt{2\Delta\nu(t_{on} + t_{off})}} = \frac{2 \times 762}{\sqrt{2 \times 2.3 \times 10^5 \times 2021}} = 0.05 \text{ K}$$

2021 s = 33.7 min per position, that is 20.2 hours of ON+OFF integration time, without overheads. With the standard treatment of the data in OTF mode, one needs 42 hours, twice as much. Nyquist sampling in ON/OFF raster mode would take 4 times more and therefore would be twice longer than OTF mode but if Nyquist sampling is not requested, **OTF mode is slower than raster mode**. By slightly decreasing the resolution of the OTF map to  $10.2''$ , i.e. using a kernel of  $\frac{1}{2}$  beam, the integration time drops to  $\sim 15$  hours and is now faster than the ON/OFF mode.

## frequency-switching

This mode is absent from APEX, though it has been tested (see report of Belloche et al. 2010). The cons is a wiggled baseline but for narrow lines, this is not a problem. The pros is a huge time saving compared to ON/OFF mode and in the case of OTF maps, the possibility to smooth more efficiently the pixels. If OTF map is used, no OFF pixel integration, so the same 418 sec are spent ON time, and the 324 pixels can be

observed  $418/324 \approx 1.3$  s/pixel. The pixel noise is now

$$\sigma_{pixel} = \frac{\sqrt{2}T_{sys}}{\sqrt{2\Delta\nu(t_{on1} + t_{on2})}} = \frac{\sqrt{2} \times 762}{\sqrt{2 \times 2.3 \times 10^5 \times 1.3}} = 1.394 \text{ K}$$

For standard processing with a  $1/3^{rd}$  beam kernel, the noise only decreases to 1.1 K, instead of 0.94 K for the standard position-switching OTF mode. But when a larger spatial smoothing is needed, the frequency-switching gets a better noise decrease since there is no common OFF spectrum being added, all OFFs being independent, there is no partial correlation in noise between adjacent pixels. For example, degrading the resolution from  $9''$  to  $15''$ , the standard position-switching OTF mode obtains an rms of 0.283 K in our present case while the FSw mode gets down to 0.235 K. The ratio of the two represents 45 % increase (+ overhead increase!) in observing time for the PSW OTF mode to catch the FSw mode efficiency.

## Conclusion

- There is a strong discrepancy with the OTC calculator. We have a factor 2 difference on the noise of the spectra after convolution with a small kernel. This means a factor 4 on the integration time.
- The xy\_map program should not be used in the basic mode because the beating between  $1/3^{rd}$  beam steps,  $1/3^{rd}$  beam kernel size and  $1/2$  beam step regridding will always be there producing pixels with strongly varying noise and slightly varying spatial resolution.
- GO NOISE should be used cautiously after running xy\_map because the edges introduced by the convolution are noisier than the useful part of the map. A solution is to force the regridding in xy\_map to remove the artificial edges with map%like.
- To make the OTF faster than the simple ON/OFF full beam sampling, as far as our tests go, one has to slightly degrade the resolution by applying a half-beam kernel to the data (12% linearly, that is from  $9.1''$  to  $10.2''$  at 691 GHz, e.g.). This also has the advantage of producing a relatively constant noise output for the standard  $1/3^{rd}$  beam sampling and we strongly recommend to perform the data reduction along this setup.
- If the signal/noise ratio is not an issue and maximum resolution is requested, another winning strategy is to perform the OTF mode with a  $1/4$  beam dump. This also produces constant noise output even with the default xy\_map convolution parameters, and slightly better S/N ratio (only at maximum spatial resolution). One drawback is the higher data volume that will be produced.
- The lack of a frequency-switching mode is a loss of time since integrations would be twice as fast and overheads much less in any case. Given the hourly cost to run this telescope (or any radio-telescope in general), it is also a considerable loss of money and energy which has no justification for one of us (LP). LP would urge to offer this mode at APEX.