PACS Spec - Pointing offset correction

The goal of this document is to describe the "pointing offset determination and flux correction" technique developped in Heidelberg for the PACS ICC. This method makes use of raster observations of Neptune to determine any eventual mispointings during science observations. The same raster observations are then used to estimate the flux lost because of the mispointing. In the following, we first describe how the calibrator observations were done, how they are processed in order to be used in combination with science observations. Finally, we explain how the flux correction is then applied.

1. Neptune raster observations

Several raster observations (8) of Neptune were conducted as part of the calibration plan of the ICC[1]. Each of the rasters was performed at one given grating position. A total of 8 different rasters therefore provide 16 grating positions in total (8 in blue band and 8 in red band). The main idea is to interpolate these raster observations onto a finer grating position grid, and use them to determine the pointing offset and subsequent flux correction as a function of time.

Each individual raster consists of a square grid of 25 × 25 Neptune observations (40 × 40 in one case). Each observation is separated by ∼2.5 arcsec in RA and Dec. In the point of view of the detector, we therefore "see" Neptune moving from e.g., the top-left corner of the 5×5 spaxels to the bottom-right corner, with a ∼2.5 arcsec sampling. Consequently, such observations mimic mispointed observations (as the source is moving along the detector) and enables us to derive the pointing offsets of science observations. It also means that for each positions of Neptune on the detector, we have access to its flux distribution (referred to as "beam profiles" in the following). For a given grating position, once the pointing offset has been determined ($\Delta\alpha \neq 0$ and $\Delta\delta \neq 0$), one can compare the beam profile of Neptune for a similar pointing, with the beam profile of Neptune when perfectly pointed ($\Delta\alpha = 0$ and $\Delta\delta = 0$). The flux difference (for the central spaxel, or the sum of any spaxels combination) is then used to estimate the correction factor to be applied to flux derived from the science observations.

One should note that all calibration and science observations are processed with the "background normalization" technique, that make use of the OFF positions during the chopping observations. The Neptune raster observations were performed slightly differently compared to usual science observations. No nodding was performed, and the chopping was done only on one side of the source (asymmetric chopping), while we have a symmetric chopping pattern with nodding (chop and nod throws being the same distance). This requires specific ajustements in the data reduction process as the telescope background is not the same in chopper position A and B (different regions of the mirror being used in two chopper positions).

[1] OBSIDs: 1342186678, 1342208866, 1342219846, 1342219847, 1342219848, 1342219849, 1342219850, 1342222162

1.1. Reduction method for the Neptune raster

The following was written by A. Poglitsch. Here we describe the way we normalize the raster observations to the background. This is the specific case of asymmetric chopping. The normalized signal *norm* is defined as follows,

$$norm = \frac{A - B}{A + B}, \tag{1}$$

where A and B are the signals in the two choppers positions. We assume that the source, when detected, will be detected in A position. If we define T_A and T_B the pure telescope fluxes in positions A and B, and s the source signal, we can then re-write the *norm* factor as follows:

$$norm = \frac{(T_\mathrm{A} + s) \times g - T_\mathrm{B} \times g}{(T_\mathrm{A} + s + T_\mathrm{B}) \times g} \tag{2}$$

$$= \frac{(T_\mathrm{A} + s)/T_\mathrm{B} - 1}{(T_\mathrm{A} + s)/T_\mathrm{B} + 1} \tag{3}$$

$$= \frac{f + f \times s/T_\mathrm{A} - 1}{f + f \times s/T_\mathrm{A} + 1}, \tag{4}$$

where f is the ratio:

$$f = \frac{T_\mathrm{A}}{T_\mathrm{B}} = \frac{A}{B} \tag{5}$$

In Eq. 5, A and B are the signals when no source is seen. This ratio is stable over time for all the spaxels and all the pixels (provided the source is not visible). The quantity x we are interested in is the source signal s as a function of telescope background (in that case T_A according to the above assumption that the source always falls in position A):

$$x = \frac{s}{T_\mathrm{A}} \tag{6}$$

Following this, we can re-write *norm*:

$$norm = \frac{f + f \times x - 1}{f + f \times x + 1} \tag{7}$$

And we can in the end go back to x, the background-normalized signal of the source:

$$x = \frac{f - 1 - norm \times (f + 1)}{f \times (norm - 1)} \tag{8}$$

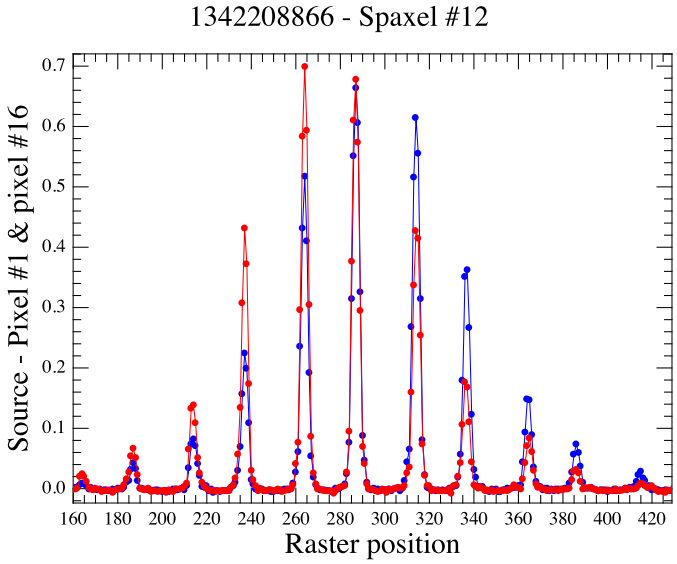

Fig. 1. Source signal normalized to the telescope background as a function of raster positions for pixel number 1 (blue) and pixel 16 (red) for the central spaxel

1.2. Output of the rasters

Figure 1 shows the evolution (flattened to 1D) of the background-normalized signal x as a function of the raster positions ($25 \times 25 = 625$) for pixel number 1 and 16 of the central spaxel # 12 (blue and red, respectively). As Neptune enters and leaves the field of view of the central spaxel, the signal increases and decreases. One can immediately notice there is a non negligible shift between the two x variables. This demonstrates that for a **fixed** grating position angle, each pixels of a single spaxel sees a different field of view.

These raster observations provide very detailed, and crucial informations that we can use to process spectroscopic observations. We are in principle able to account for most of the effects that are responsible for poorly reduced data. To summarize, from the raster observations, we extract the following observables:

1. 8 sets of Neptune rasters, with different, but fixed grating positions
2. For one given Neptune raster, we have a grid of 25×25 positions on the sky, with step-sizes of ~ 2.5 arcsec
3. For each position on the sky, we have the background normalized intensity distribution x for the 5×5 spaxels
4. For each spaxel, we have background normalized signal x for each of the 16 spectral pixels

1.3. Reduction method for the science data

Since the raster observations were done with one-side chopping and no nodding, the normalization of the science data must be done in a comparable way. The observations we are interested in are done with two-sides chopping (symmetrically with respect to the source) and chopping.

Jeroen: Could you remind me if there are any differences on the way the science data are reduced ?

1.4. A better constraint on the f factor

The f factor, described above as the ratio $T_{\rm A}/T_{\rm B}$ can be derived directly from the observations by using the off positions when chopping. However, $T_{\rm A}$ and $T_{\rm B}$ are not measured simultaneously. If A is in the off position, B is on source and vice-versa. This time difference between the measurements of $T_{\rm A}$ and $T_{\rm B}$ may introduce uncertainties in the f ratio. Therefore, to obtain a more reliable estimate of this ratio, we made use of so-called "dark sky measurements" for which no sources are in the field-of-view of the detector (**which OBSID ?**). These observations were performed with different values for the chopper throws, similar to ones used in scientific observations. Because no source is observed, $T_{\rm A}$ and $T_{\rm B}$ are measured at the same time, therefore providing a reliable estimate for f. These values, that can be found in the calibration files, appear to be slightly time-dependent. This dependency correlates with the mirror temperatures (**is it correct ?**), and can be corrected for using PACS housekeeping measurements.

2. Description of the workflow

In the following, we provide a first, rough description of the method, simply to detail the two main steps of the procedure: the determination of the offset and the subsequent flux correction. Once these two fondamental steps are explained, we will then present a more refined approach to the data processing.

2.1. Pointing offset

The biggest challenge is to obtain an accurate determination of the pointing. For the sake of simplicity, we will only consider one given grating position from now on. At first, we will use only one spectral pixel (e.g., pixel # 8). First thing is to build the so-called beam profile (i.e., the intensity distribution from the signal x) of the science target. We can refer to this quantity as $P_{\rm SCI,\,i,\,j}$, where i denotes the pixel number, j the spaxel number. In the current version of the script, one can define which spaxels are to be used. One can choose to use the central spaxel only, the 3×3, or the 5×5 spaxels. The beam profiles P will have the dimensions of the number of spaxels selected (i.e., 1, 3×3 or 5×5, respectively). Therefore, we can simplify the aforementioned notation to $P_{\rm SCI,\,i}$. Second step is to do the same for the Neptune raster observations, choosing the same spaxels as for the science data ($P_{\rm CAL,\,i}$). Because we have observations with different pointing, we therefore have $P_{\rm CAL,\,i}(\Delta\alpha, \Delta\delta)$. In principle, this should result in 25×25 different beam profiles. However, in order to have a finer sampling of the pointing offset, these 25×25 beams profiles are oversampled by a factor $n_{\rm over}$, which can be modified in the script. The default value that has proven to work efficiently is of about 10. This means that the step-size of 2.5 arcsec becomes $n_{\rm over}$ times smaller.

For a given spectral pixel i, and the considered spaxels, we then build a χ^2 map which is computed as follows

$$\chi^2(\Delta\alpha, \Delta\delta, i) = \left(\frac{P_{\rm CAL,\,i}(\Delta\alpha, \Delta\delta) - P_{\rm SCI,\,i}}{\sigma_{\rm i}} \right)^2 , \tag{9}$$

Fig. 2. Oversampled χ^2 map of the difference between beam profiles for the science and calibrator data (on the 3×3 central spaxels). Dark regions trace small χ^2 values.

where $\sigma_{\rm i}$ are the estimated uncertainties (**how are they computed ?**). Figure 2 shows an example of such a χ^2 map, the darker regions trace low χ^2. We therefore obtain a first estimate of where the true pointing is (at the place of the smallest χ^2). We can produce similar maps for every spectral pixels. If one notices that one or more pixels are affected by strong noise, a selection of the pixels to be used can be done in the procedure.

2.2. Flux correction

For a given spectral pixel, we can first assume that the true pointing offset is where the χ^2 reaches a minimum. We can refer to this on-sky position as $[\Delta\alpha, \Delta\delta]$ (which should be different from $[0, 0]$; a perfectly pointed observation). To correct for the flux loss due to mispointing, one can then simply multiply the flux of the science target, at the considered grating position, to the ratio $\sum_{spaxels} P_{\rm CAL,\,i}(0,0) / \sum_{spaxels} P_{\rm CAL,\,i}(\Delta\alpha, \Delta\delta)$. Indeed, by comparing the total amount of flux in the beam profiles at the on-sky position $[\Delta\alpha, \Delta\delta]$ to the total flux at the on-sky position $[0, 0]$, we have a good estimate of the flux lost due to the offset in pointing.

Such a correction can be applied for all spectral pixels, and we can proceed in a similar way for the grating positions at which the raster were observed.

3. Toward a more refined procedure

In the following, we present several improvements of the data processing technique we described above.

3.1. Exploring the χ^2 map: flat-fielding

In Section 2.1 we assume the pointing to be at the minimum of the χ^2 map. However, the signal-to-noise of the observations may not always ensure this assumption is always true. We proposed another route to find the best pointing offset. For a given grating position, from the χ^2 maps, which are very similar pixel-to-pixel, we define an on-sky region of low-χ^2. This region is defined as all $[\Delta\alpha, \Delta\delta]$ positions for which $\chi^2[\Delta\alpha, \Delta\delta] < \mathrm{MIN}(\chi^2) \times K$, where K is a threshold value (default value is **2?**).

Then, for all $[\Delta\alpha, \Delta\delta]$ positions within the low-χ^2 values, we correct the fluxes of the science target for all the 16 pixels (or all selected pixels). We then need to define a criterium to select which of all these positions traces best the true pointing. We estimate the flat-fielding to be a good proxy for that: if the pointing is perfect, then the fluxes of all spectral pixels should be the same. We therefore define a new χ^2_{pix} as the sum of the absolute values of the differences between all the pixels. This is implemented as follows in the jython script

```
for ipix in range(npix):
```
$$\chi^2_{\mathrm{pix}}[\Delta\alpha, \Delta\delta] \texttt{ = } \chi^2_{\mathrm{pix}}[\Delta\alpha, \Delta\delta] \texttt{ + SUM(ABS(}F_{\mathrm{corr}}\texttt{[ipix] - }F_{\mathrm{corr}}\texttt{[ipix+1]))},$$

where F_{corr}`[ipix]` is the corrected flux for the spectral pixel `ipix` of the science observations. From the χ^2_{pix} we can then find which $[\Delta\alpha, \Delta\delta]$ value provides the best flat-fielding, and we assume this is a reliable estimate for the true pointing.

3.2. The grating interpolation

As mentioned earlier, the correction factor can only be applied at a few grating positions (8 per band). One possible way to correct for flux losses over the entire specral range would be to interpolate the correction factors over a finer grid of grating position. However, proceeding this way means that we rely on our estimation of the pointing offset. One may therefore argue that a "bad estimation" would then propagate to other grating positions. We consequently opted for an alternative way; we interpolate the beam profiles over a fine grid of grating positions, and use the pointing determination plus flux correction procedure at other grating angles. In the end, we *have to* "create" informations for the missing grating positions, and we chose to rely on the observed data (and interpolate from them) rather than doing so on the fitting results, which may be subject to noise or other atrifacts.

3.3. Tweaking parameters

Here we simply summarize which parameters can be changed in the procedure.

- `spaxels`: a 1D array that tells the script which spaxels are to be used. Can be the central one, the 3×3 central ones, or any spaxels you want.
- `pixels`: a 1D array that tells the script which spectral pixels are to be used. Can be all 16, or a selection of pixels you want
- `smoothing`: the oversampling factor by which you wish to refine the raster observations (finer stepsize than the original 2.5 arcsec)
- `grating`: the number of grating positions you wish to interpolate on

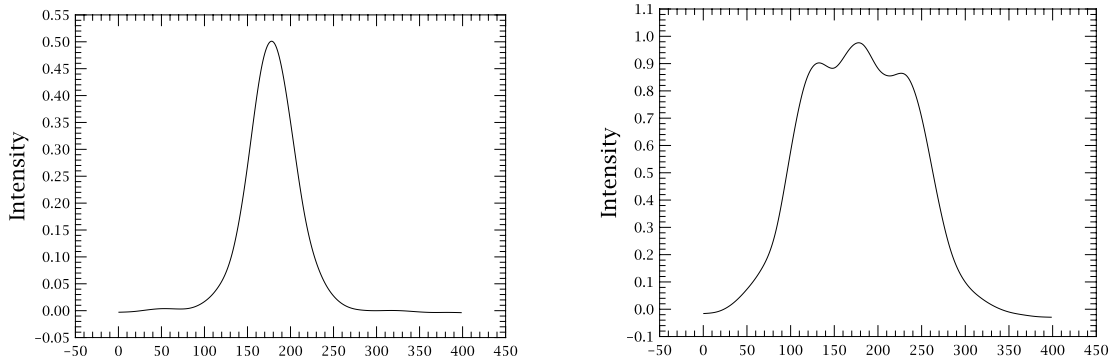

Fig. 3. Cuts in 1D of the oversampled (`smoothing`= 16) beam profiles for the central spaxel (left panel) and the sum of the 3×3 spaxels (right panel)

– `K`: the χ^2 threshold by which you define the number of $[\Delta\alpha, \Delta\delta]$ positions to be tested for the flat-fielding

4. Central spaxel vs. 3×3 central ones

Originally, only the central spaxel # 12 was used in the procedure. However, we encounter some problems for several test cases that we could solve by using the central 3×3 spaxels. The explanation lies in the shape of the beam profiles. The calibration beam profile for the central spaxel will look like a 2D Gaussian, and therefore the wings of the profile will decrease very fast (see left panel of Fig. 3). If the true pointing is quite off-centered compared to $[0, 0]$, then the correction factor, will be estimated where the beam profile is steep. We found that this translates into high uncertainties on the final correction factor.

Instead, if for the same mispointing, we use the 3×3 spaxels the correction factor will be estimated from the sum of the 3×3 beam profiles (see Sect. 2.2). In this case, the estimation of the correction factor will be much more reliable.

However, it also means that we cannot provide a correction "per spaxel" independantly, for the same reason as described above. If the true pointing lies too far away from the center of a given spaxel, the correction to be applied will become unreliable.