Simulating dipole calibration for PICO

Maurizio Tomasi

December 2018

Hypothesis

I assume to calibrate the instrument using the solar dipole of the CMB, whose peak-to-peak amplitude is ~7 mK. I use TOAST to produce the timelines, and DaCapo to compute the calibration factors. The scripts used to run the simulation have been prepared by Andrea Zonca and are available at pico-simulations.

As a reference, the dipole in Ecliptic coordinates has the following shape (the color scale represents the temperature in K):

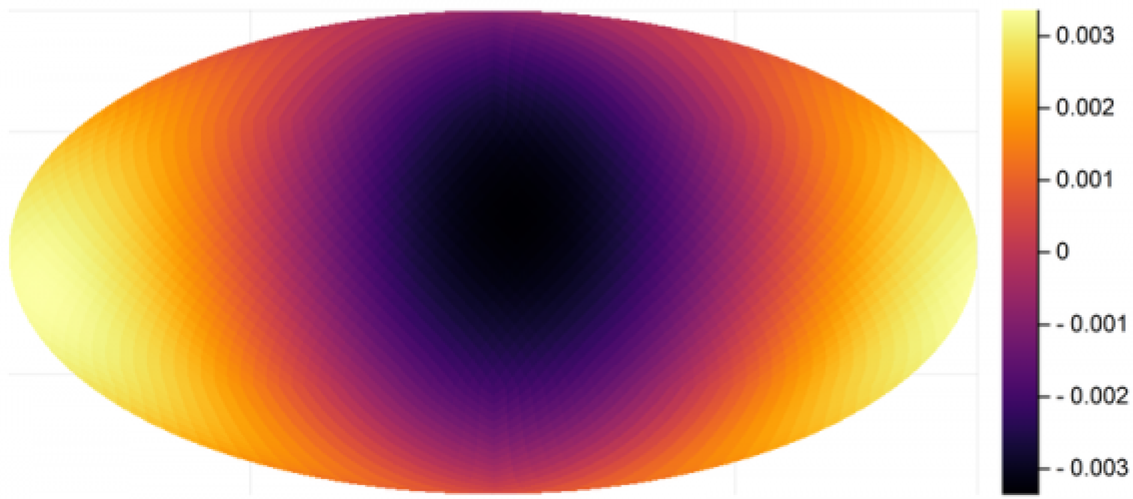


Figure 1. The CMB dipole, shown in Ecliptic coordinates

Observation of the dipole

I used the scanning strategy parameters listed at the page Optimizing Scan Strategy, with no kinematic dipole and the Galactic signal simulated by PySM for a W-band detector, which should be the best case in terms of Galactic contamination.

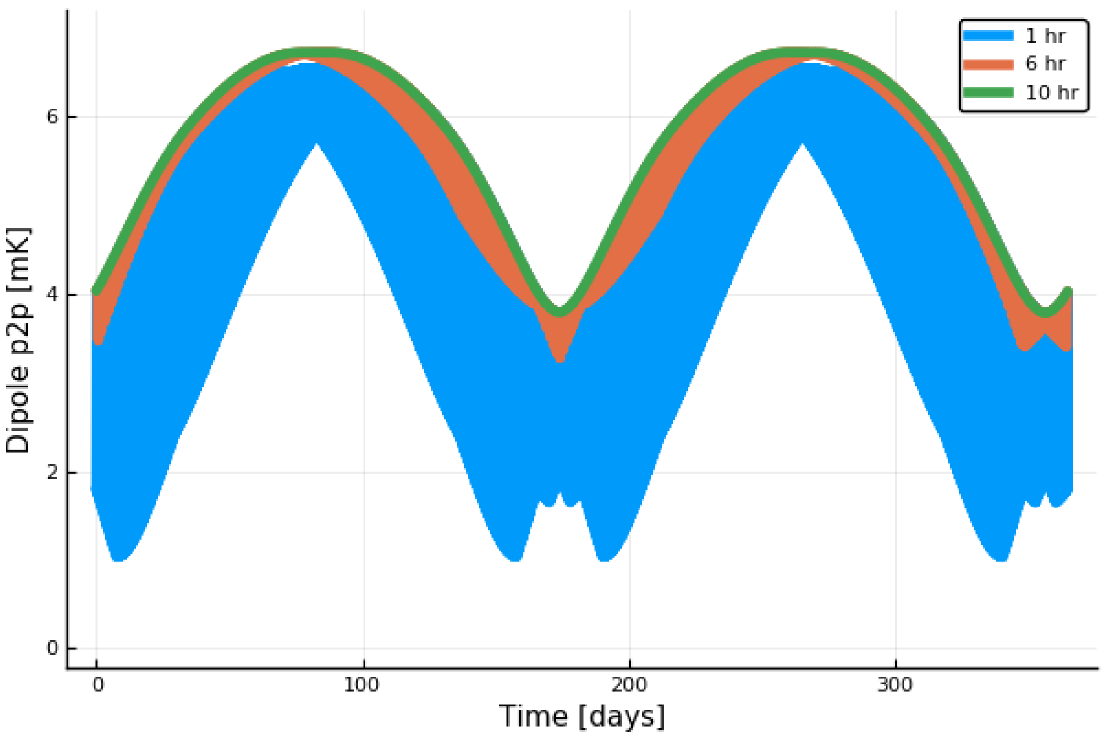
The sky coverage as a function of time shows a sharp rise in the first few hours, reaching ~50% of the sky in one precession period. Then, the value increases slowly till it reaches 100% after nearly 6 months. (For comparison, fsky varies more linearly in the case of Planck.) The following figure shows fsky as a function of time:

The precession allows PICO to observe wider portions of the sky in the same time. This is good for dipole calibration as well, as it allows the detectors to potentially sample larger variations of the dipole, thus increasing the S/N of the dipole measurement.

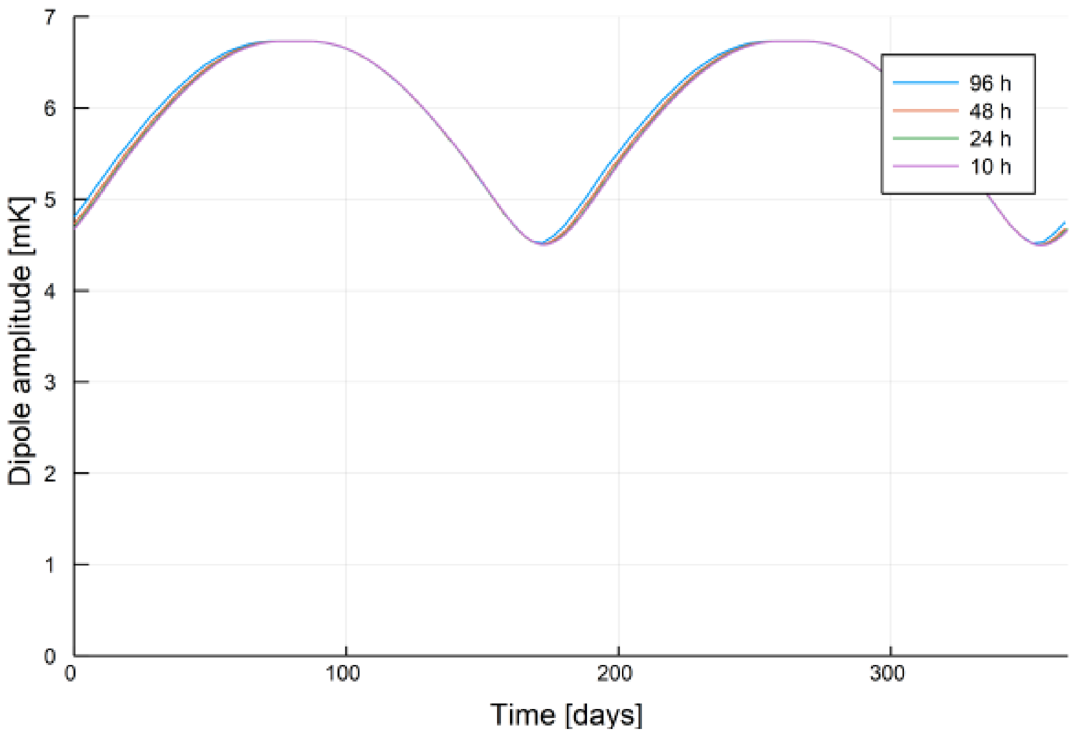
A simple parameter that encapsulates much of the details about photometric calibration is the peak-to-peak variation in the dipole signal during some fixed time span. As PICO detectors are not meant to perform absolute measurements, the modulation of the dipole signal consititutes our calibration source. The peak-to-peak amplitude of the dipole is therefore a good tracer of the S/N for the calibration signal.

Consider this image, which shows the simulated timeline produced by TOAST for one of the W-band detectors (temperature in K as a function of time in seconds, no Galaxy is present):

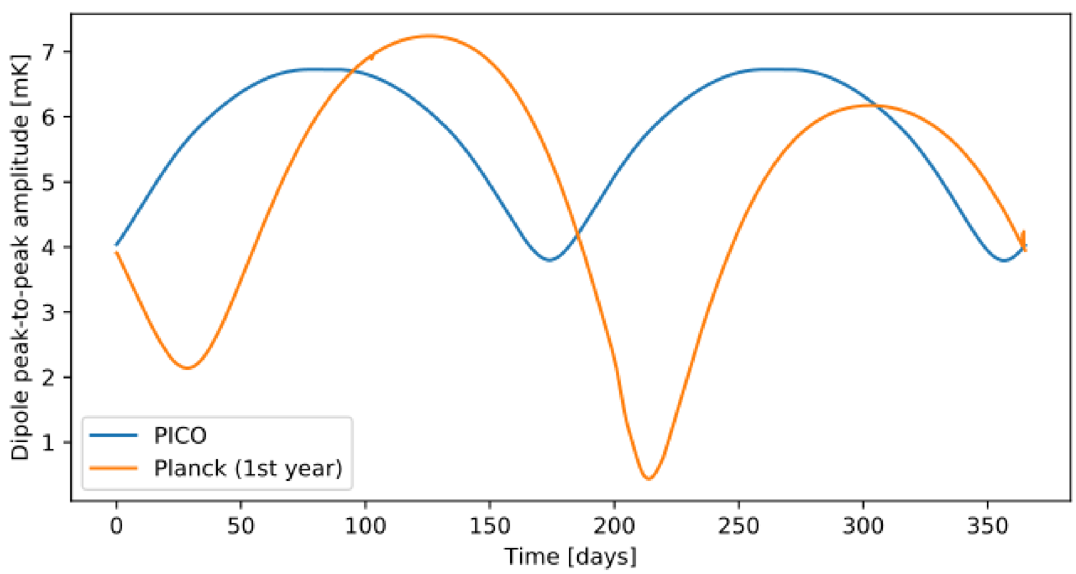
The low-frequency modulation is due to the dipole signal, while the high-frequency noise is detector noise. As the plot shows ~100 seconds of data, the peak-to-peak variation of the dipole signal within this time frame is less than 3 mK. The following image shows how the peak-to-peak dipole amplitude changes as a function of the time window used to estimate the peak-to-peak value, for three values of the window:



There are several interesting facts that become apparent from the image above. First of all, there are high-frequency fluctuations which disappear for longer time windows. Specifically, as the length of the time window approaches the precession period, high-frequency fluctuations are present no longer. A second interesting fact is that the plot shows low-frequency fluctuations as well. These are unavoidable and are due to the yearly variation in the angle between the spacecraft spin axis and the CMB dipole axis. As a matter of fact, these fluctuations are present even for much longer time windows:



It is interesting to note that the peak-to-peak amplitude of these fluctuations is significantly better than Planck’s scanning strategy:



Running DaCapo

I assume that the calibration code to be used in a real mission would include a component-separation step able to reduce the bias due to the contamination of the dipole signal by Galactic large-scale structures. Therefore, in our simulations I assume that the only signal in the sky apart from the dipole is the CMB itself. From the point of view of calibration, CMB acts as a noise signal which is correlated among detectors.

The DaCapo algorithm is explained in Planck 2015 results. V. LFI calibration and in the source code of the Python version of DaCapo (https://github.com/ziotom78/dacapo_calibration), which has been written with the aim to be readable as an article. The algorithm works as follows:

1. Apply an extended version of the destriping equation, which reconstructs the baselines of 1/f noise as well as the gain factors. This uses the following assumptions:
    - 1/f baselines have zero mean
    - Gains are constrained by a model of the dipole, which must be provided as input
2. The equation is solved iteratively using the Conjugate Gradient algorithm. At the end, the following estimates are available:
    - 1/f baselines
    - Timeline of gain factors
    - Sky map (temperature only) without the dipole
3. Iterate the algorithm again, subtracting the sky map obtained in the previous step from the timelines, in order to remove any systematic due to spurious signals (CMB, foregrounds).
4. When the 1/f baselines, the gain factors, and the sky map no longer change, save all the results and quit.
5. DaCapo must be run once per each detector, so that the sky map produced as output is typically noisier than the maps expected from a typical configuration of detectors.

I employed the DaCapo algorithm in the following pipeline:

1. Use TOAST to produce timelines using two detectors (0A and 0B) along the boresight direction, the nominal scanning strategy and realistic noise, and save them to FITS files. The gain used in this step is 1.0 (constant) for both detectors. TOAST produces a sky map (including the dipole) using the Madam map-maker (a destriper) and data from both detectors 0A and 0B.
2. Run DaCapo on the FITS files produced during the previous stage. As the algorithm works on single detectors, the code must be run twice (once for detector 0A and once for 0B).
3. If DaCapo converges, re-run TOAST as in the previous step, but this time use the gains calculated by DaCapo instead of assuming a constant (1.0) gain.
4. At the end of the simulation, take the difference between the maps produced by TOAST in steps 1. and 3. and compute the power spectrum. These spectra represent the error caused by an imperfect estimate of gain drifts during the mission.

Our simulations assume that each detector has a calibration factor equal to 1.0. The calibration code is therefore expected to retrieve the same value from the simulated data, plus some statistical noise. Of course, the code does not know that the gain is constant, so it always tries to estimate the dependency of gain on time.

I have run a few simulations using DaCapo on the simulated timelines produced by Andrea Zonca. Here are the details:

- 1 year of observation
- 2 boresight W-band detectors
- CMB with r=0, no Galaxy
- Planck2015 solar dipole, including the quadrupolar correction
- No orbital dipole
- 1/f + WN, with fknee = 10 mHz (corresponding to a period of 100 s).
- Constant gain (1.0)

The CMB realization I used is the following:

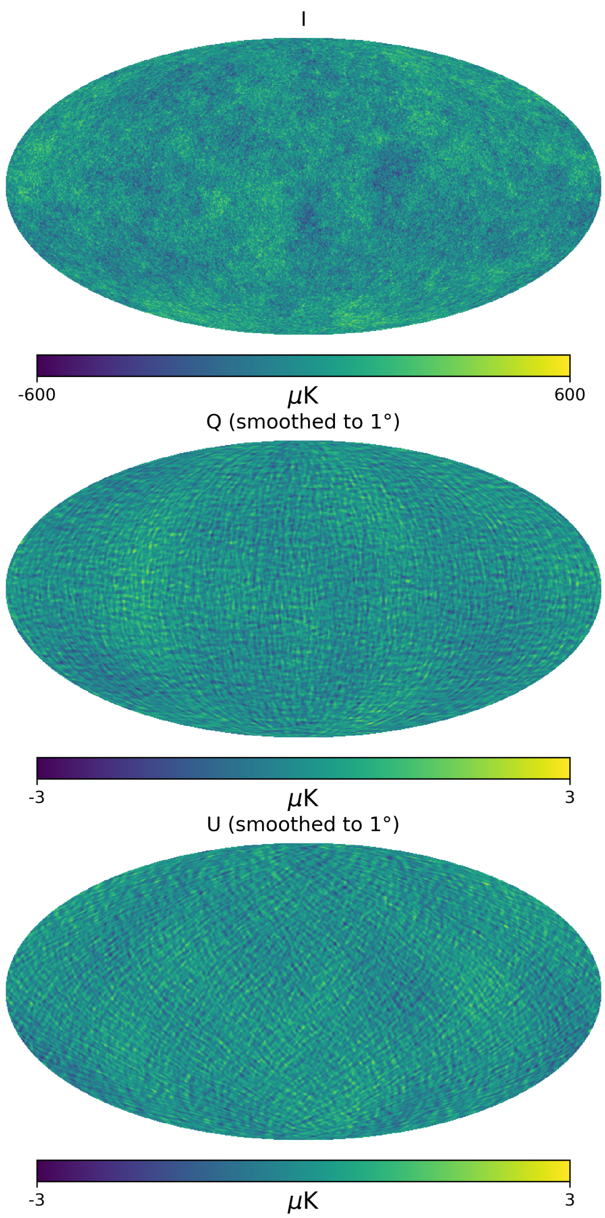
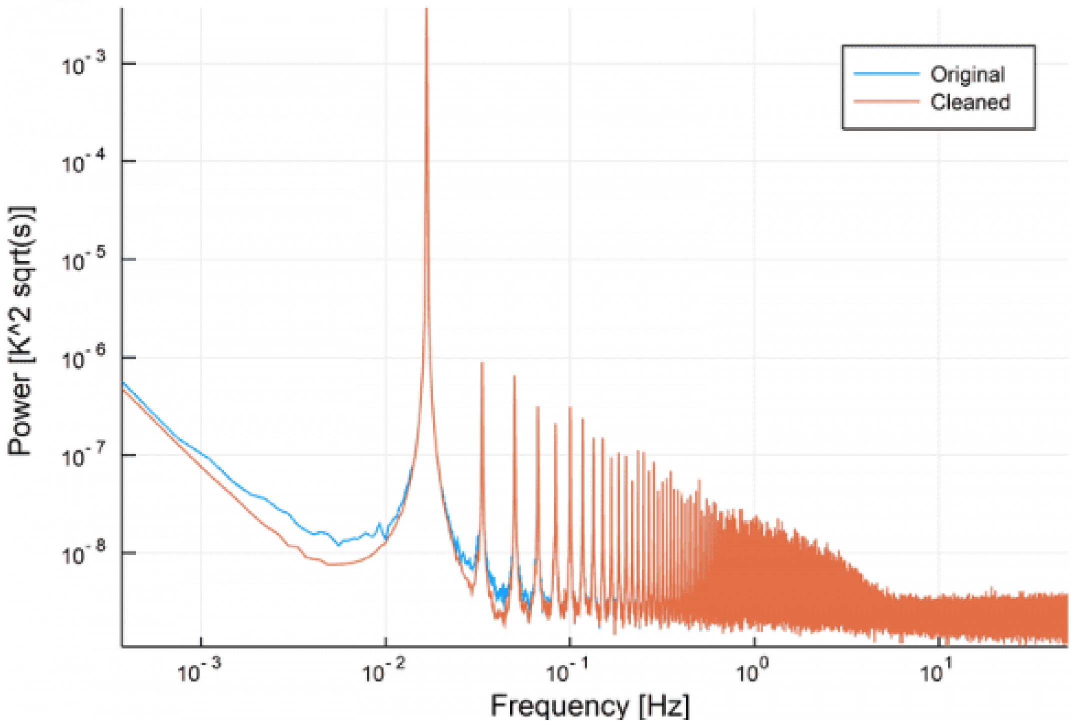
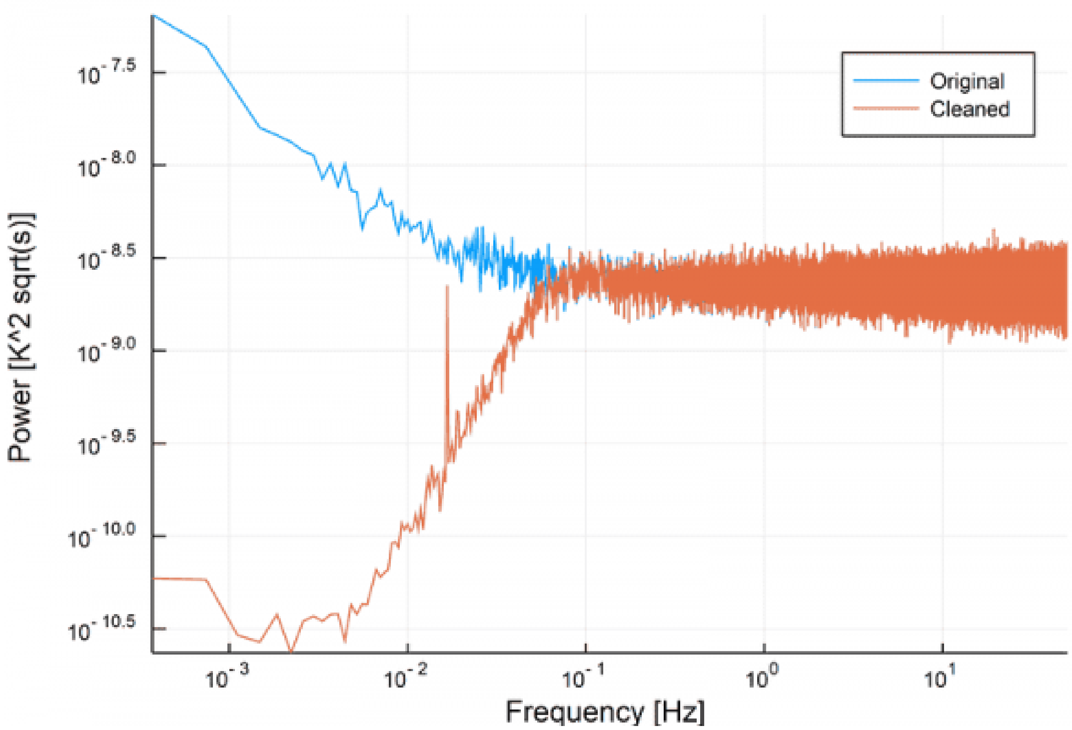


Figure 2. CMB map used in the simulations

As DaCapo implements a destriper as well, let’s check that the 1/f is correctly estimated. The baseline I used was 10 s:

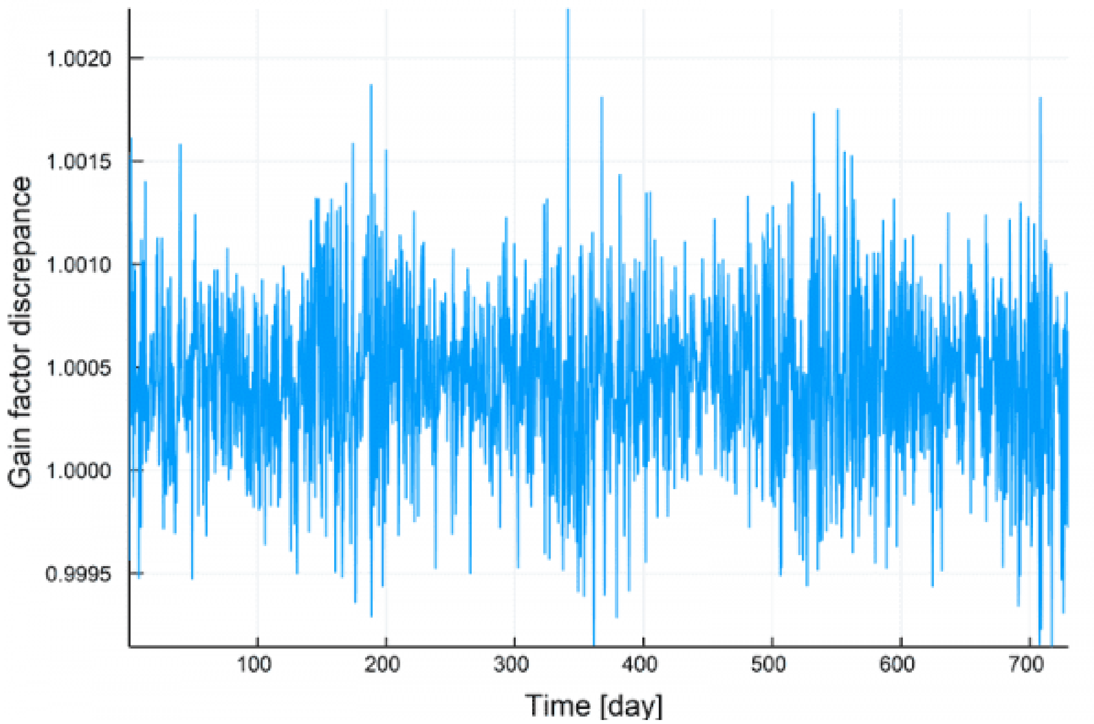


(“Original” is the TOD produced by TOAST and fed to DaCapo, while “Cleaned” is the same TOD minus the 1/f baselines calculated by DaCapo.) This plot is complicate to understand due to the presence of the sky signal (the dipole peak and its harmonics). Let’s compute the PSD of the noise component alone: this is easy, since TOAST saves the TOD of the foreground signal in a separate column in the FITS files.



The peak at 167 mHz corresponds to the spin period (60 s): note that the height of this residual peak matches the level of the high-frequency WN tail. Apart from that, 1/f is correctly suppressed by the code.

The estimated gains for a time window of 10 hours (the same as the precession period) are shown here:



You can see that the RMS of the gain factors varies with time. This is anticorrelated with the peak-to-peak amplitude of the dipole (see the plot above). The average value of the gains shows some residual bias, as it is not exactly 1.0: this bias is due to the fact that TOAST convolves the sky and the dipole signals with a beam of some finite size, but DaCapo considers a pencil beam (i.e., a Dirac delta). The usual way to deal with this is to use a 4π convolver, like the one used in Appendix A of Planck 2015 results. V. LFI calibration, but for our purpose it's easier to subtract an offset so that the average value of the gain is 1.0. The RMS of the timeline is 0.04%, which can be considered as an estimate of the relative calibration error.

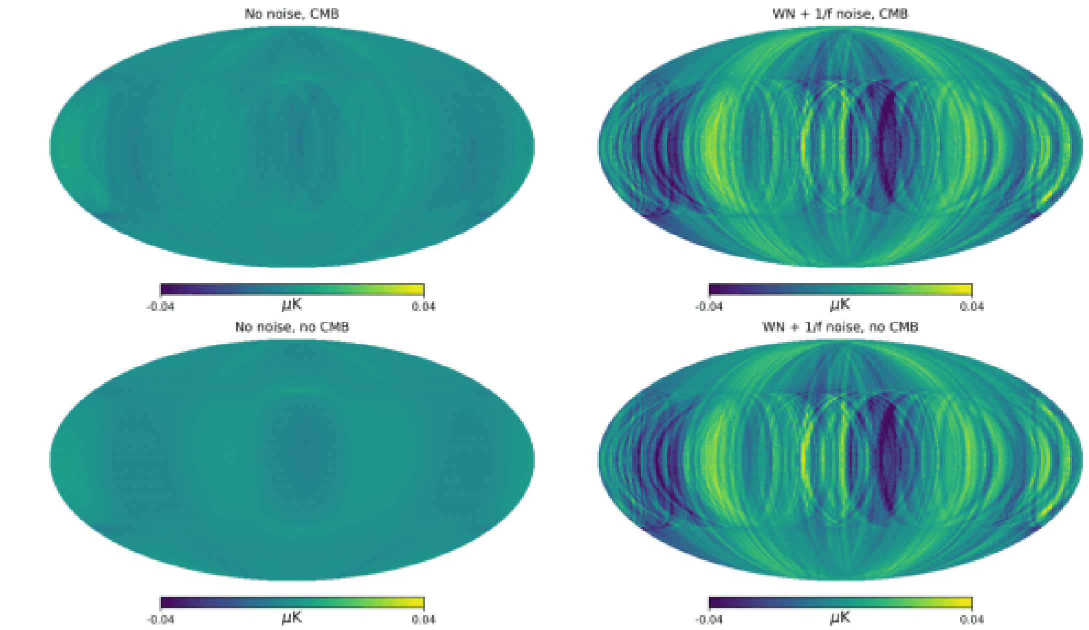
Output maps

As sketched in the previous section, I produced full-sky maps with TOAST, calibrate them with DaCapo and re-run TOAST with the gains estimated by DaCapo to get a new full-sky map which contains the effect of imperfect gain drift reconstruction. The difference between the two maps show the projected effect of gain drifts on the sky.

I have run two sets of simulations, using the following assumptions:

- Two noiseless receivers (0A and 0B);
- Two W-band receivers with nominal white noise and 1/f noise with knee frequency 10 mHz.

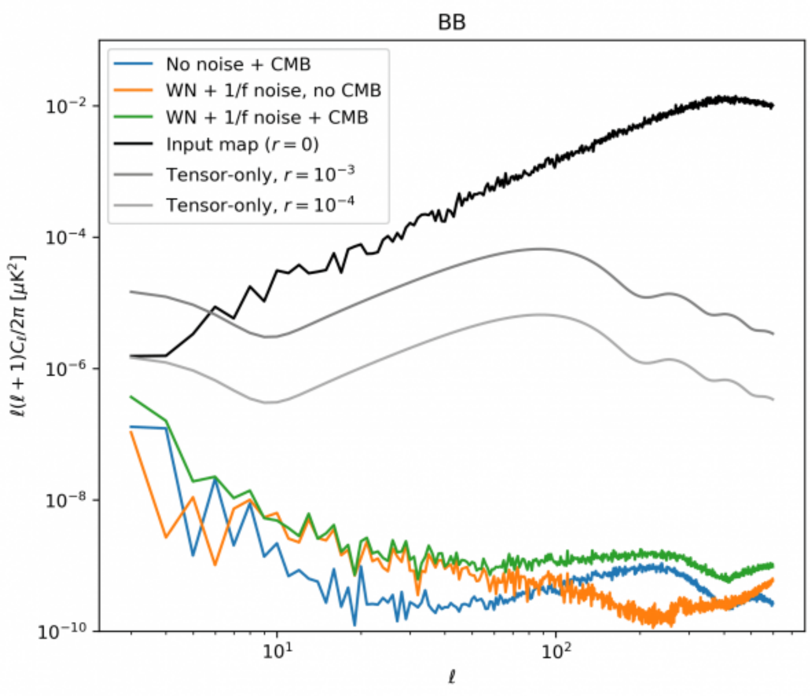
The differenced maps produced in the two cases are quite different:



Of course, the scale of the effect considers the presence of two detectors on the whole focal plane, which is not the case for PICO.

Computing the BB spectrum of the map in the two cases and confronting it with the input map shows the scale of the effect (as well as two power spectra assuming only tensor modes, provided as a reference):

In order to produce this plot, I rescaled the spectrum by 2/N, where N is the number of W-band bolometers in the focal plane of PICO, and then I performed a NET-weighted average over all the cosmology bands from 60 GHz (band 7) to 220 GHz (band 22): this assumes that the gain fluctuations are negligibly correlated among detectors. This is the case for the case with realistic noise, but not for the noiseless simulation; this should be expected, as the only noise here is caused by the CMB, which is the same for detectors 0A and 0B:



BB
No noise + CMB
WN + 1/f noise, no CMB
WN + 1/f noise + CMB
Input map ($r = 0$)
Tensor-only, $r = 10^{-3}$
Tensor-only, $r = 10^{-4}$
$\ell(\ell+1)C_\ell/2\pi$ [μK^2]
ℓ