

# THE EFFECT OF FRINGE RATE FILTER SHAPE ON THE POWER SPECTRUM

MATTHEW KOLOPANIS

This memo outlines the changes to the PAPER-64 power spectrum pipeline after Ali et al 2015. The application of the fringe rate filter is the final step in the PAPER analysis pipeline. The filter sets the total effective integration time and number of independent sky samples in the l<sub>1</sub>-binned. Fringe rate filtering also suppresses "non-sky" systematic modes. Small changes to the shape of the filter can have a fairly dramatic impact the power spectrum.

The fringe rate filter used in Ali 2015 (A15 hereafter) report on the PSA64 power spectrum is notably different from that described in Parsons et al Beam sculpting and fringe rate filter paper. This memo steps through the differences between the two to reconstruct the shape of the A15 filter and assess the impact on the data.

First, we compute the power spectrum of the data used to make the A15 power spectrum. Next we reconstruct the filter applied to the data and get a power spectrum which is close to the A15 result. Finally, we apply the "optimal" filter and get a power spectrum. All results ahve been corrected for signal loss using the method in A15.

## 1. POWER SPECTRUM OF A15

We use the data from the A15 as input to the power spectrum pipeline to ensure the local version of the pipeline is in agreement with the A15 results.

Figure 1 displays the A15 power spectrum, to a spectrum made with the fringe-rate filtered data used by A15. The spectra show good agreement in the range  $.15 < k < .5 h \text{ Mpc}^{-1}$ , as such it is reasonable to assume the power spectrum pipeline is working properly on enterprise at ASU.

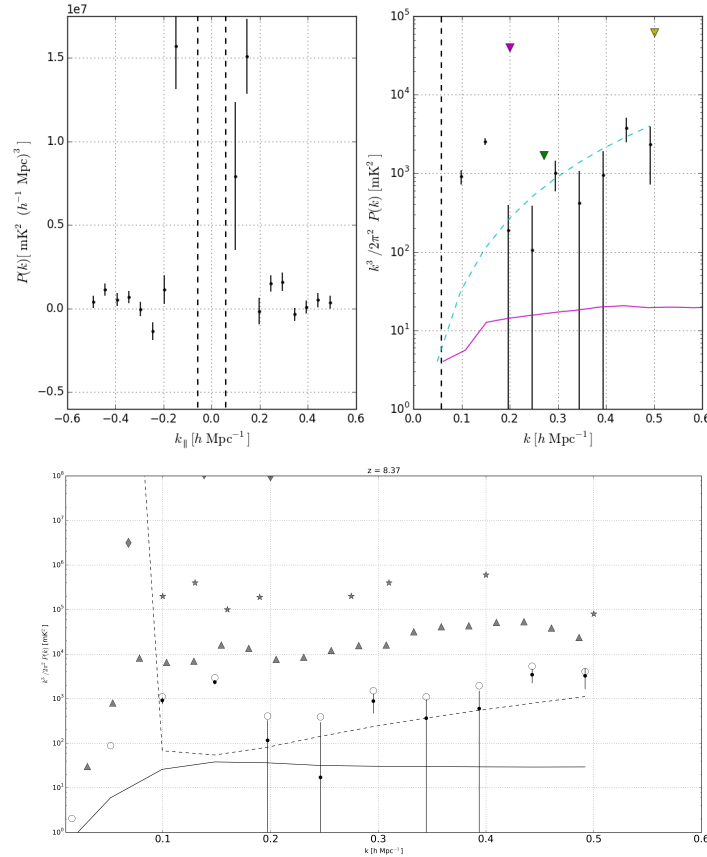


FIG. 1.— Top: the apj v002 power spectrum from Ali et al 2015. Bottom: Open circles are  $2\sigma$  A15 published data points (same as plotted in top figure). Black dots are our run of the power spectrum code on archived l<sub>1</sub>-binned frf filtered data used in the A15 paper. The only difference is a slightly higher signal loss which accounts for the offset. All error bars are 2 sigma

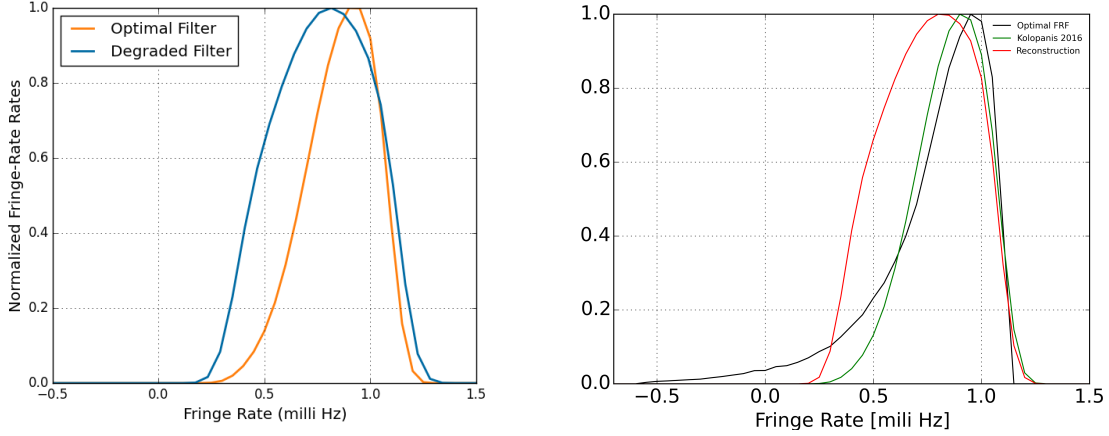


FIG. 2.— Left: the A15 fringe rate filter. Right: the optimal frf, K16 frf, and the A15 reconstruction

## 2. RECONSTRUCTING THE A15 FILTER

The following steps were taken to try to recreate the A15 frf:

1. `capo/src/frf_conv.py mk_fng(bl, eq)`
  - (a) this is the `mk_fng` for the optimal frf
2. `capo/src/frf_conv.py mk_fng_alietal(bl,eq)`
  - (a) This is the `mk_fng` used in A15

`frf_conv.py` should look like:

```

24 def mk_fng(bl, eq):
25     '''Return fringe rates given eq coords and a baseline vector (measured in wavelengths)
      in eq coords'''
26     return -2*n.pi/a.const.sidereal_day * n.dot(n.cross(n.array([0,0,1.]), bl), eq)
27
28 fringe used in ali et.al to degrade optimal fringe rate filter.
29 def mk_fng_alietal(bl, eq):
30     '''Return distorted fringe rates for given eq coordinates and a baseline vector (
      measured in wavelengths) in eq coords. This was the version used in ali et.al'''
31     ey, ex, ez = eq#yes, ex and ey are flipped.
32     return 2*n.pi/a.const.sidereal_day * (bl[0]*ex + bl[1]*ey * n.sqrt(1-ez**2))

```

3. `capo/mjk/scripts/frf_filter.py`<sup>1</sup>

- (a) this script applies the frf selected by the user on data passed as arguments
- (b) command line options allow for the application of the optimal or ali et al frf
- (c) can fine tune the frf by setting channel, maxfr, baseline length scaling factor, alietal flag
- (d) setting the following parameters for the ali et. al filter
  - i. `chan=160`
  - ii. `-alietal`

Note use of channel 160 to create the sky fringe as per Zaki's instructions. In A15, this fringe is used to make a fringe rate profile, this profile is then interpolated between frequency bins.

A comparison of the A15 filter, the optimal filter, the K16 filter and the A15 reconstruction filter are shown in Figure 2.

The reconstruction filter is then applied to the data, the resulting power spectrum is shown in Figure 3.

<sup>1</sup> While it is not necessary to use this python script to apply a fringe rate filter to data, I used this script in accompaniment with `capo/mjk/scripts/batch_frf_filter.sh` to filter all data contained in a directory whose sub-directories are named "even" and "odd" with the A15 reconstruction filter (e.g. `lstbin_psa64_no_frf/even,odd/sep0,1 ..etc`). This script would be executed in `lstbin_psa64_no_frf/`

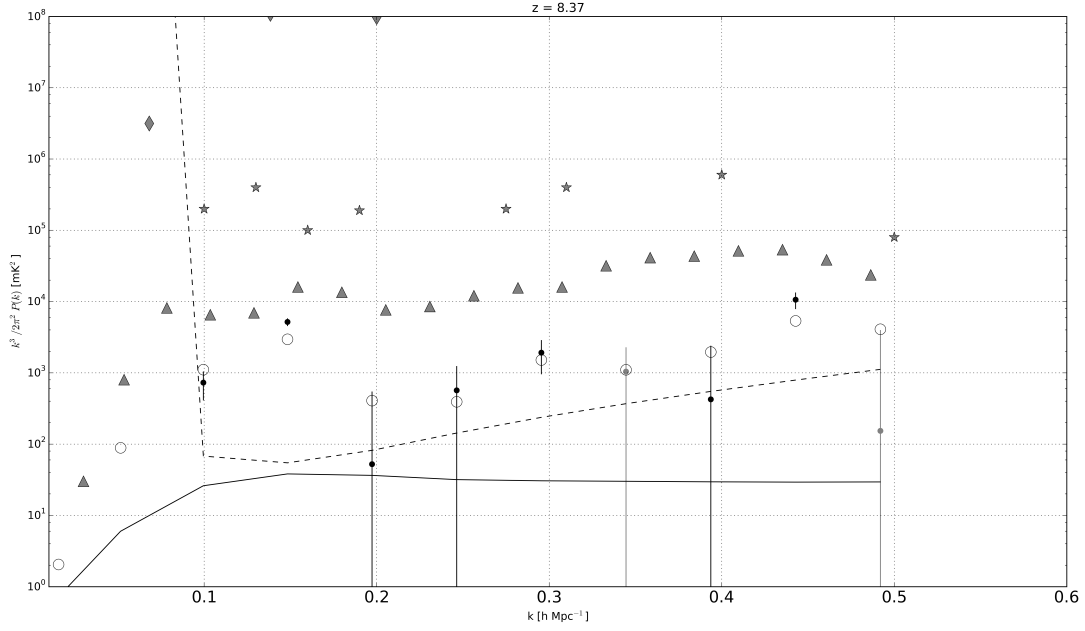


FIG. 3.— Black dots with error bars show the power spectrum after applying the reconstructed A15 frf. Open circles in the lower image indicate  $2\sigma$  upper bounds from the A15.

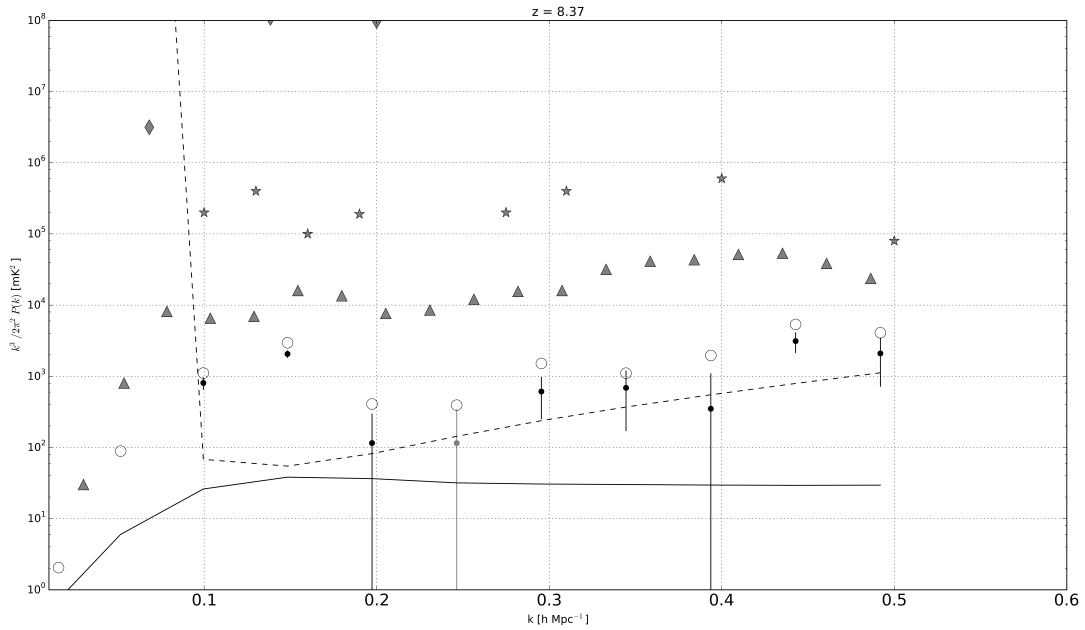


FIG. 4.— Black dots with error bars show the power spectrum after applying the optimal filter. Open circles in the lower image indicate  $2\sigma$  upper bounds from the A15.

### 3. OPTIMAL FRINGE RATE FILTER

To construct the optimal fringe rate filter used in the following alterations should be made:

1. `capo/mjk/scripts/frf_filter.py`
  - (a) do not set the `alietal` flag
  - (b) use chan 101 or 105

Any script can be used to filter unfiltered data. I use `capo/mjk/scripts/batch_frf_filter.sh` which again should be executed in a directory like `lstbin_psa64_nofrf/` whose sub-directories are "even" and "odd".

A power spectrum from data filtered with the optimal fringe rate filter is shown in Figure 4.