JIVE UniBoard Correlator Memo 17: A comparison of on- and off-board averaging of UniBoard spectra

Des Small, Harro Verkouter, and Jonathan Hargreaves

JIVE

November 2, 2015

1 What this is and why we're doing it

The current UniBoard firmware design was based on the use of 1024 spectral points across each of its 16 MHz subband. This design decision has not aged well – it is not enough spectral resolution for spectral line experiments and too much for continuum work, where 32 or 64 points across the band are typically used.

Not only is the resulting output of the UniBoard very large and therefore inconvenient to store, every post-processing step has to process more data than is really necessary. It rapidly became clear that we would like to average data from continuum experiments down to, say, 32 points, which is the industry standard resolution for continuum data. Originally this averaging was implemented in the standard tool `j2ms2` which is used to convert the data into the standard 'Measurement Set' format used by CASA and other tools. But since the averaging simply consists of summing the values of successive frequency bins and the corresponding weights, we subsequently realised that it could also be implemented directly in the UniBoard firmware. Note that we also refer to this process as 'aggregation' below in contexts where we are thinking particularly in terms of integer summation. Both averaging strategies were implemented; the purpose of this document is to compare the results of the two averaging procedures.

We note in passing that SFXC also currently uses a high spectral resolution internally and averages down the results before exporting them. At one time this was done with the simple binning scheme we use here, but later this was replaced by a more sophisticated scheme that I don't understand and which cannot in any case be implemented with the UniBoard data without major changes to the firmware. Informal accounts from the SFXC team suggest that there was not a drastic change in the results when the improved scheme was introduced.

2 Method and Results

Data from scan 11 of the EVN experiment EG087A was used, simply because it was used for previous testing and the data was thus available in VDIF format. In both cases the correlated data was averaged down from 1024 to 32 spectral points (a factor of 32 reduction).

The amplitudes of the correlator for a sample of auto- and cross-correlation baselines for our chosen scan are shown in Figure 1; the amplitudes for the auto-correlation are of course much larger than for cross-correlations.

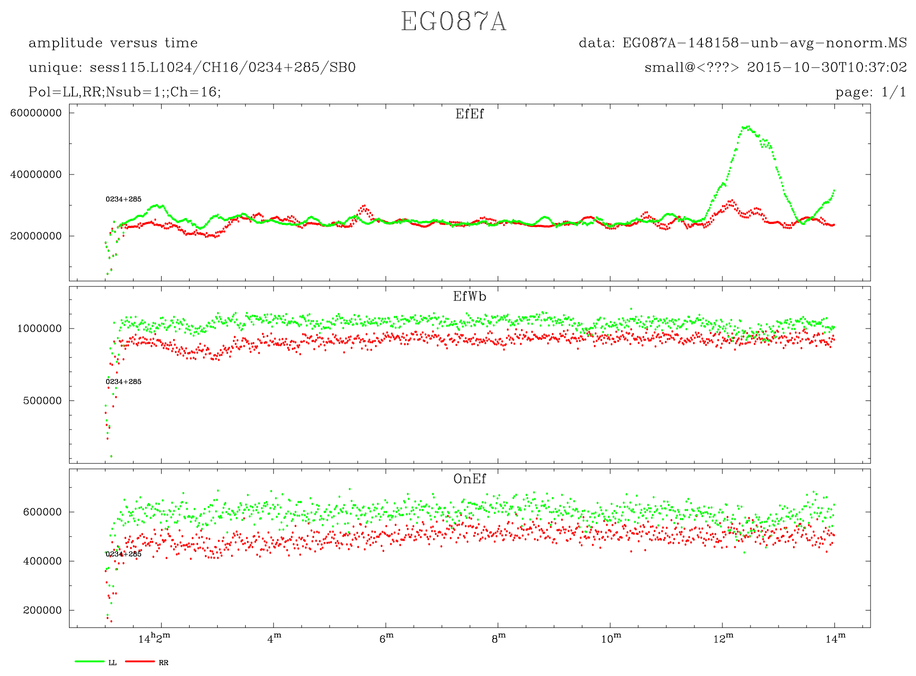

Figure 1: The unnormalized amplitudes for selected baselines

As part of our research into the subtle differences in amplitudes that can result from different aggregation methods, an existing Python tool for inspecting raw UniBoard output was extended by Harro Verkouter to allow aggregation of the integer-valued complex output using integer arithmetic.

Figure 2 shows the difference between UniBoard data without any on-board aggregation post-processed to aggregate frequencies in two different

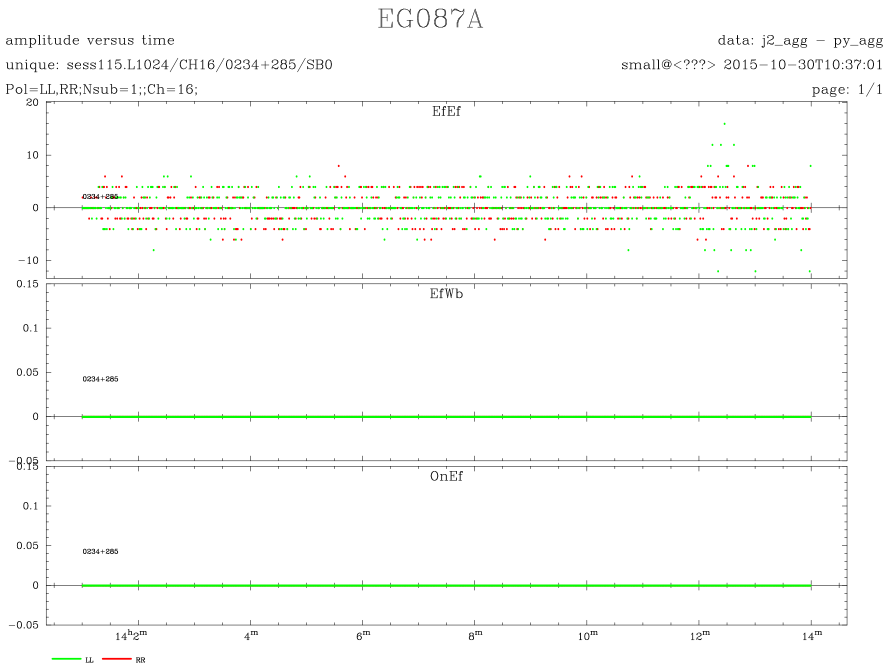

Figure 2: The difference between aggregation by Python and j2ms2

ways: firstly, using this Python analysis tool and secondly using the standard tool j2ms2, which converts each frequency bin's real and imaginary values output to single-precision floating point numbers before aggregation. For the cross correlations the differences are zero, but for the auto-correlations the differences are quantized as integers distributed around zero. Since both aggregators are working on identical UniBoard output, this must be a result of the floating point conversion in j2ms2. Single precision arithmetic can only exactly represent values up to 16777216; values between that and 33445532 are rounded to even numbers[1].

The Python tool was built precisely because we wanted to isolate the effects of floating point arithmetic; the result of the comparison between this tool shows that the effects are real but very small.

A second small effect results from internal truncation of the data products on the UniBoard before they are exported. Internally the UniBoard firmware represents each data product during an integration in a 36-bit register for each of the real and imaginary components, but at the end of an integration it exports only the high 32-bits of this register. This of course

[1]`https://en.wikipedia.org/wiki/Single-precision_floating-point_format#Precision_limitations_on_integer_values`

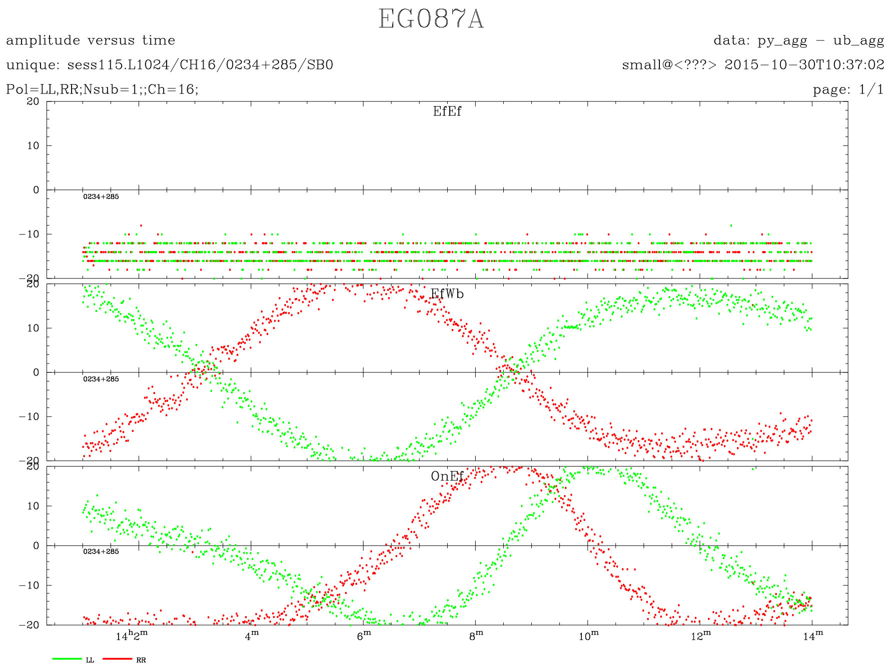

Figure 3: The difference between aggregation by Python and the uniboard

means that the least significant four bits are not exported, but when the UniBoard's onboard frequency aggregation is used the aggregation is done using the full 36-bit range.

Figure 3 shows a comparison between the UniBoard's full-precision frequency-bin aggregation and off-board aggregation using the Python integer aggregator working on results that have been truncated during export from the UniBoard. The results are what we expect: the cross correlation shows that (onboard) aggregation before truncation always gives a slightly higher value than truncation first and then aggregating, leaving a negative result when the former is subtracted from the latter as in the figure.

For the sake of completeness, Jonathan Hargreaves developed a custom variant of the UniBoard hardware in which data products are truncated to 32 bits by dropping instead the four most significant bits. This would not be appropriate for general use, because of the risk of overflowing the registers in auto-correlations, but based on the above results we established that it is safe in the case considered here. Figure 4 shows difference between on-board aggregation with this custom firmware and off-board aggregation using the Python tool using the untruncated results from the same firmware. As expected, the differences are exactly zero for both auto- and cross-correlation products.

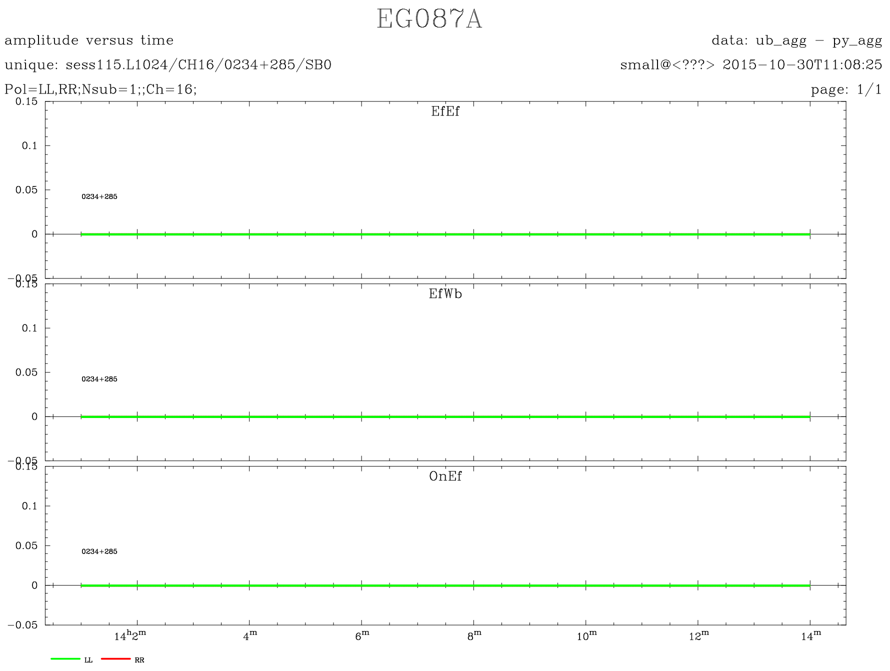

Figure 4: Python and uniboard comparison with non-truncating firmware

3 Conclusion

Frequency bin aggregation on the UniBoard drastically reduces the amount of storage required for correlator products and the amount of effort required to postprocess them. We anticipate that averaging down to 32 spectral points will be standard for production correlation with the UniBoard correlator.

We have shown that differences in frequency aggregation strategies can give rise to very small differences in correlator products – of order 10^{-7}, comparable to the accuracy of single-precision floating point representations (and in some cases arising directly from that precision) – depending on how and where aggregation is placed with respect to truncation of the results and conversion to floating point. Given that truncation of UniBoard results to 32-bit output is necessary then the best possible results come from doing the aggregation on-board with the full 36-bit registers, and then converting the totals to floating point in j2ms2 to store the data in the Casa Measurement Set format. This is of course exactly what we wanted to do in the first place.