Data Compression study with E2 data

S. Klimenko, B. Mours, P. Shawhan, A. Sazonov

March LSC 2001 meeting

Baton Rouge, Louisiana

LSC Session : Detector Characterization

Purpose of the study

- Use the E2 data as bench mark for
  - existing compression methods
  - new compression methods

- Issues investigated:
  - effective compression factor
  - compression/uncompression speed
  - bias introduced in the data for lossy compression

- Detailed report available (LIGO-T010033-00-E)

Available Frame Compression (Format Spec. & I/O library)

- Only lossless compression methods
- Compression done at the vector level
  - » No need to uncompress unused channels.
- Standard gzip
  - » Integer are differentiated to improve compression factor.
- Zero suppress
  - » Differentiated data are stored with the minimal number of bits needed.
  - » Available only for integer.

Lossless Data Compression Performances

	Full Frames	RDS Frames
Number of bits to store a short (z. sup.)	6.1	7.6
Number of bits to store a float (gzip)	21.4	23.7
Compression/uncomp. speed (short; z. sup.)	13/12 Mb/s	12/20 MB/s
Compression/uncomp. speed (float; gzip)	1/3 MB/s	2/11MB/s
Fraction of float	31%	64%
Raw Size	3.2 MB/s	1.5 MB/s
Size after gzip + Zero supp.	1.7MB/s	1.0 MB/s

Speed measured on a Sun Ultra 10 @ 450 Mhz

The IFO was lock for this data set.

- Remarks:
  - » Poor speed and compression factor for float
  - » People use floating points

New Lossless Compression for float

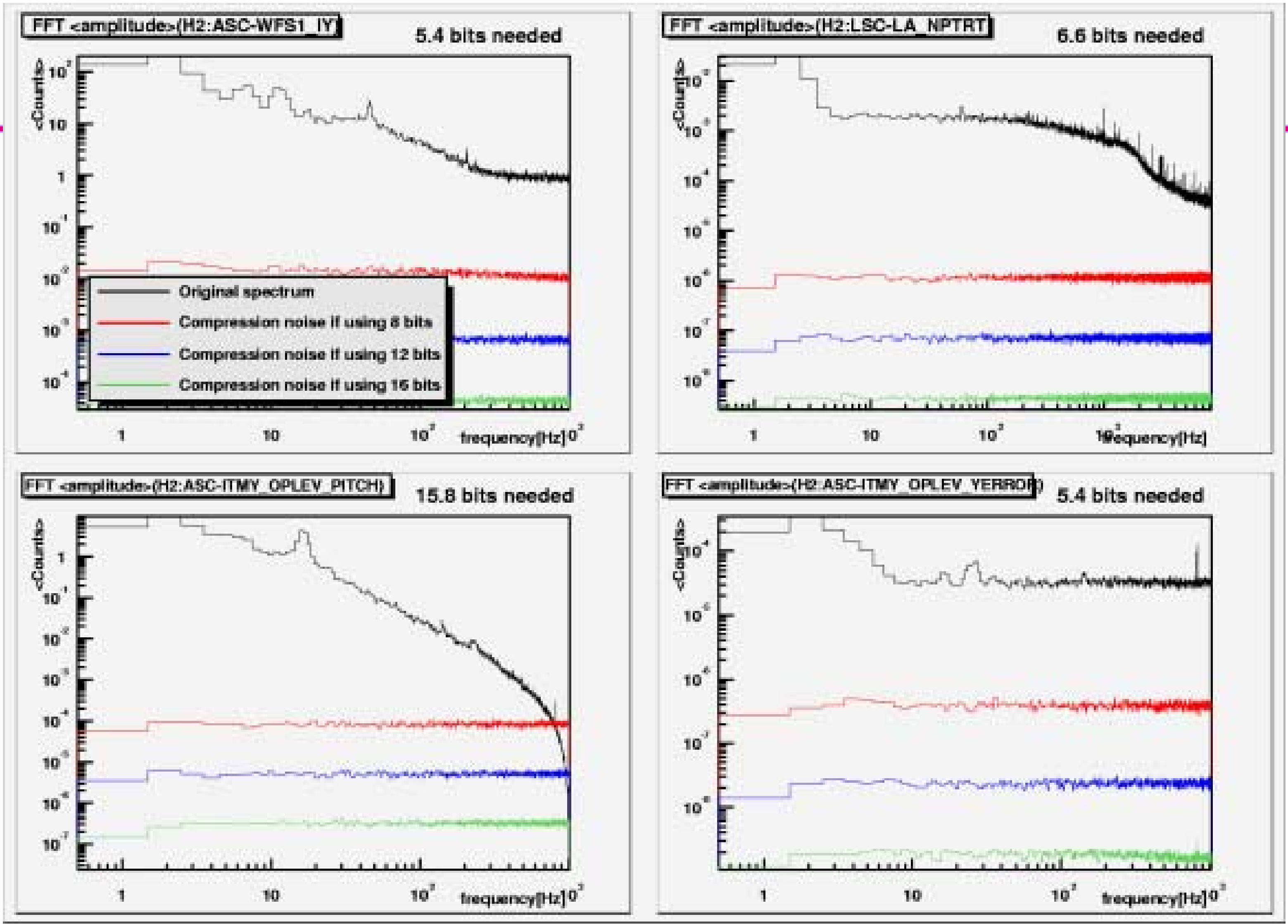
- Method:
  - Handle floating point numbers as if they are integer numbers
  - Apply differentiation and zero suppress algorithms.
  - It works because the sign and exponent stored in the most significant bits change slowly from one data word to the next one.
- Same compression factor as gzip
  - 22-24 bits per word
- Much faster method:
  - 30MB/s compression
  - 60MB/s uncompression

New Lossy Compression for float
Convert float to integer

- Method: Convert floating point to integer.
- Technique
  - » Differentiate the data and digitize the differences: $(s_{i+1} - s_i)/k$
  - » Round off is done by checking that the rebuilt data do not diverge from the original data.
- Data saved
  - » First value, the differences converted to integers, a scaling factor.
- One parameter: number of bits to store the integers
- Speed: Fast
  - » compression 11 MB/s*, uncompress 24 MB/s

*(assuming a predefined number of bits for storage)

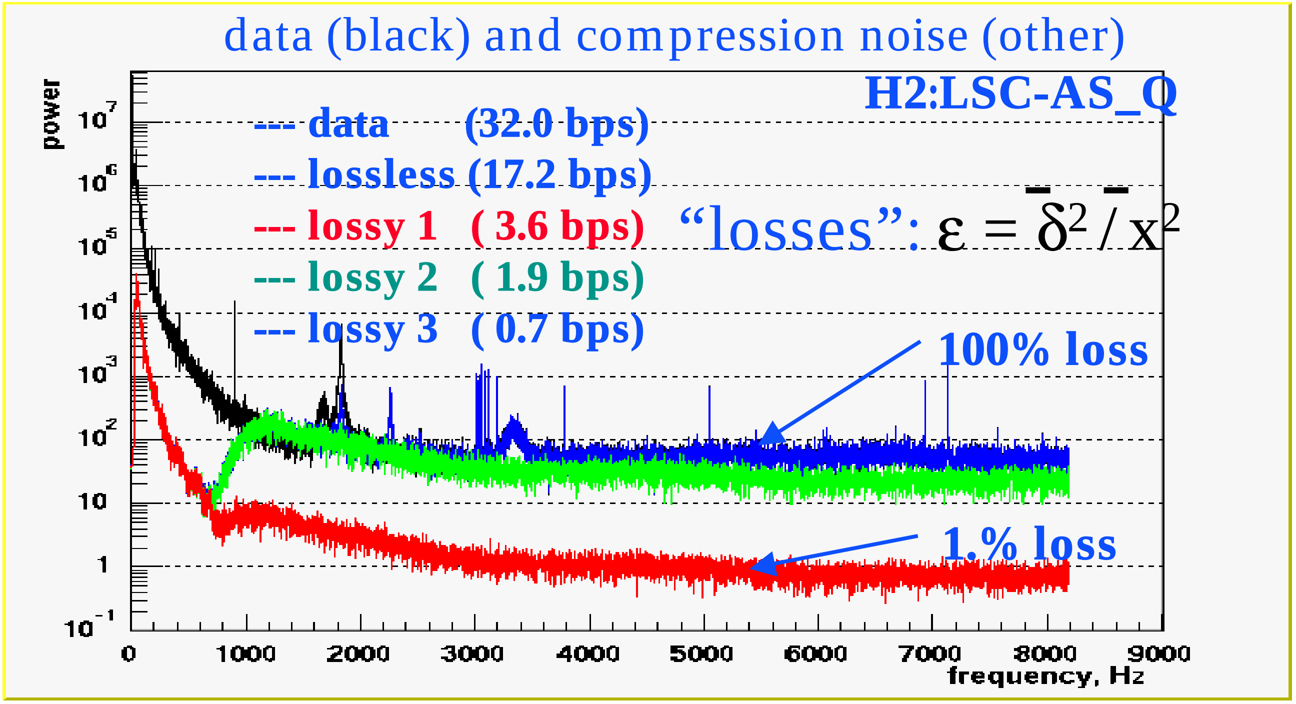
Noise for the float to int. method



Need on average 7.5 bits per word

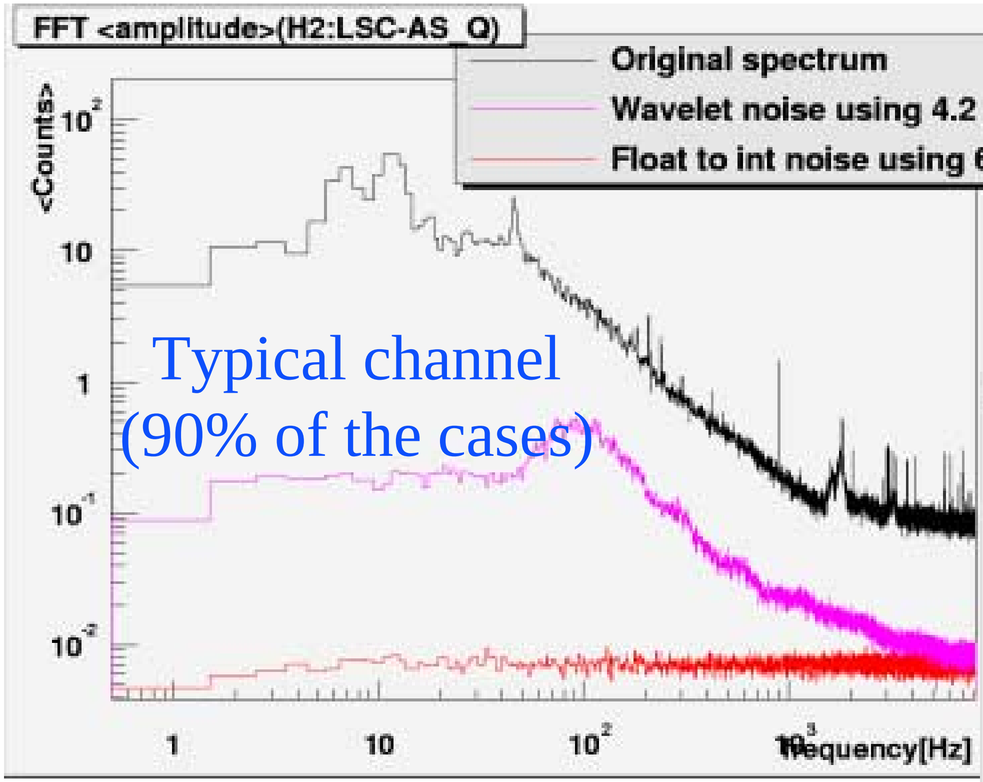
Wavelet compression

- Signal dynamic adjusted to follow the noise floor
- Large compression could be achieved



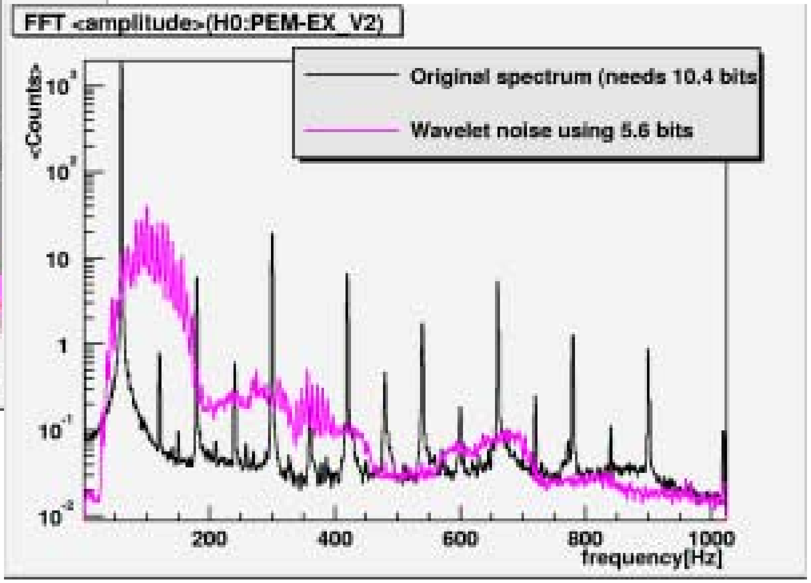
“losses”: $\varepsilon = \overline{\delta^2} / \overline{x^2}$

Wavelet: noise introduced



Typical channel
(90% of the cases)

The few channels
with strong lines



Summary

- The best 'compression' is to record only what you need:
  - » right channel, right frequency, right type (integers are better than floats)
- Existing tools
  - » Work OK short integer, Poor on float
- New lossless compression for floats
  - » Solves the gzip speed problem
- Lossy compression techniques
  - » Their use requires care
  - » Adapted to reduced data sets or specific studies
  - » Float to integer method: simple and fast technique (8 bits per sample)
  - » Wavelet: more powerful technique (4 bps or less) but less straightforward (More details in S. Klimenko talk)