Astron. Astrophys. Suppl. Ser. **147**, 51–74 (2000)

Data streams from the low frequency instrument on-board the PLANCK satellite: Statistical analysis and compression efficiency*

M. Maris[1], **D. Maino**[1], **C. Burigana**[2], **and F. Pasian**[1]

[1] Osservatorio Astronomico di Trieste, Via G.B. Tiepolo 11, I-34131 Trieste, Italia
e-mail: <name>@ts.astro.it
[2] Istituto TeSRE, Consiglio Nazionale delle Ricerche, Via Gobetti 101, I-40129 Bologna, Italia
e-mail: burigana@tesre.bo.cnr.it

Received May 22; accepted August 7, 2000

Abstract. The expected data rate produced by the Low Frequency Instrument (LFI) planned to fly on the ESA PLANCK mission in 2007, is over a factor 8 larger than the bandwidth allowed by the spacecraft transmission system to download the LFI data. We discuss the application of lossless compression to PLANCK-LFI data streams in order to reduce the overall data flow. We perform both theoretical analysis and experimental tests using realistically simulated data streams in order to fix the statistical properties of the signal and the maximal compression rate allowed by several lossless compression algorithms. We studied the influence of signal composition and of acquisition parameters on the compression rate $C_{\rm r}$ and develop a semiempirical formalism to account for it. The best performing compressor tested up to now is the arithmetic compression of order 1, designed for optimizing the compression of white noise like signals, which allows an overall compression rate $\overline{C_{\rm r}} = 2.65 \pm 0.02$. We find that such result is not improved by other lossless compressors, being the signal almost white noise dominated. Lossless compression algorithms alone will not solve the bandwidth problem but needs to be combined with other techniques.

Key words: methods: data analysis — methods: statistical — instrumentation: miscellaneous — space vehicles — cosmology: cosmic microwave background

1. Introduction and Scanning Strategy

The PLANCK satellite (formerly COBRAS/SAMBA, Bersanelli et al. 1996), which is planned to be launched in 2007, will produce full sky CMB maps with high accuracy and resolution over a wide range of frequencies (Mandolesi et al. 1998; Puget et al. 1998). Table 1 summarizes the basic properties of LFI aboard PLANCK. The reported sensitivities per resolution element – i.e. a squared pixel with side equal to the Full Width at Half Maximum (FWHM) extent of the beam –, in terms of antenna temperature, represents the goals of LFI for 14 months of routine scientific operations as recently revised by the LFI Consortium (Mandolesi et al. 1999).

The limited bandwidth reserved to the downlink of scientific data calls for huge lossless compression, theoretical upper limit being about four (Maris et al. 1999). Careful simulations are demanded to quantify the capability of true compressors for "realistic" synthetic data and improve the theoretical analysis, including CMB signal (monopole, dipole and anisotropies), foregrounds and instrumental noise.

During the data acquisition phase the PLANCK satellite will rotate at a rate of one circle per minute around a given spin axis that changes its direction every hour (of $2.5'$ on the ecliptic plane in the case of simple scanning strategy), thus observing the same circle on the sky for 60 consecutive times (Mandolesi et al. 1998; Mandolesi et al. 2000). LFI will produce continuous data streams of temperature differences between the microwave sky and a set of on-board reference sources; both differential measurements and reference source temperatures must be recorded.

The LFI Proposal assumes a sampling time $\tau_{\rm s} \sim$ 7 msec for each detector (Mandolesi et al. 1998, Bersanelli et al. 2000), thus calling for a typical data rate of $\sim$ 260 Kb/sec, while the allocated bandwidth to download PLANCK data to ground is in total $\sim$ 60 Kb/sec. Assuming the total bandwidth to be equally split between instruments, $\approx$ 30 Kb/sec on the average would be assigned to LFI asking for a compression of about a

Send offprint requests to: M. Maris, e-mail: maris@ts.astro.it
* This paper is dedicated to the memory of Richard Gispert, HFI Data Processing Center Manager, excellent colleague, brilliant scientist, and unforgettable friend.

factor 8.4. Data have to be downloaded without information losses and by minimizing scientific processing on board.

A possible solution would be to adapt the sampling rate to the angular resolution specific for each frequency. This should allow to save about up to a factor ≈ 9 for the 30 GHz channel, but since only $\approx 7\%$ of the samples come from such channel (see Table 1) the overall reduction in the final data rate would be $\approx 17\%$.

On the other hand, it is unlikely that the bandwidth for the downlink channel may be enhanced to solve the bandwidth problem, since the ground facilities are shared between different missions and there is the need to minimize possible cross-talks between the instrument and the communication system.

With the aim of optimizing of the transmission bandwidth dedicated to the downlink of LFI data from the Planck spacecraft to the FIRST/Planck Ground Segment, we analyze in detail the role that can be played by lossless compression of LFI data before they are sent to Earth.

We apply different compression algorithms to suitable subsets of Planck-LFI simulated data streams generated by considering different combinations of astrophysical and instrumental signals and for different instrumental characteristics and detection electronics.

The first considered contribution is that introduced by receiver noise: we consider here the case of pure white noise and of white noise coupled to $1/f$ noise with different knee frequencies. The reference load temperature is assumed to be 20 K for present tests; because of the strong dependence of the $1/f$ noise on the load temperature, this can be considered a worst case, since the actual baseline reference load is of 4 K.

Different sky signal sources are subsequently added to the receiver noise: CMB fluctuations, CMB dipole, Galaxy emission and extragalactic point sources. The signal from the different sky components is convolved with the corresponding antenna pattern shapes, assumed to be symmetric and Gaussian with the FWHM reported in Table 1.

We generate simulated data streams at the two extreme frequency channels, 30 GHz and 100 GHz and consider data streams with different time lengths. We explore also different signal offset and scaling. The large number of above combinations was systematically explored using an automated program generator as described by Maris et al. (1998).

In Sect. 2 we characterize quantitatively the LFI signal component by component. In Sect. 3 we discuss how the acquisition chain is modeled to perform compression simulations. A theoretical analysis of the compression efficiency is presented in Sect. 4. While Sect. 5 is devoted to the analysis of the signal statistics. The subject of quantization error is illustrated in Sect. 6. The experimental protocol and results about compression are reported in Sect. 7. Further constraints on the on-board data compression are reported in Sect. 8. A proposal for an alternative coding method is made in Sect. 9. The overall compression rate is estimated in Sect. 10. Conclusions are in Sect. 11. Appendix A is included to further illustrate the estimation of the overall compression rate.

2. Characterization of Planck-LFI signal components

The simulated cosmological and astrophysical components are generated according to the methods described in Burigana et al. (2000) and the data stream and noise generation as in Burigana et al. (1997b), Seiffert et al. (1997) and Maino et al. (1999). We summarize here below the basic points.

• *Modeling the CMB pattern* – The CMB monopole and dipole have been generated by using the Lorentz invariance of photon distribution functions, η, in the phase space (Compton–Getting effect): $\eta_{\rm obs}(\nu_{\rm obs}, \boldsymbol{n}) = \eta_{\rm CMB}(\nu_{\rm CMB})$, where $\nu_{\rm obs}$ is the observation frequency, $\nu_{\rm CMB} = \nu_{\rm obs}(1 + \boldsymbol{\beta} \times \boldsymbol{n})/\sqrt{1-\beta^2}$ is the corresponding frequency in the CMB rest frame, $\boldsymbol{n}$ is the unit vector of the photon propagation direction and $\boldsymbol{\beta} = \boldsymbol{v}/c$ the observer velocity. A blackbody spectrum at $T_0 = 2.725$ K (Mather et al. 1999) is assumed for η. For Gaussian models, the CMB anisotropies at $l \geq 2$ can be simulated by following the standard spherical harmonic expansion (see, e.g., Burigana et al. 1998) or by using FFT (Fast Fourier Transform) techniques which take advantage of equatorial pixelization (Muciaccia et al. 1997).

• *Modeling the Galaxy emission* – The Haslam map at 408 MHz (Haslam et al. 1982) is the only full-sky map currently available albeit large sky areas are sampled at 1420 MHz (Reich 1986) and at 2300 MHz (Jonas et al. 1998). To clean these maps from free-free emission we use a 2.7 GHz compilation of $\sim$ 7000 HII sources from C. Witebsky (1978, unpublished) at resolution of $\sim 1°$. They are subtracted for modeling the diffuse components and then re-added to the final maps. We use a spectral index $\beta_{\rm ff} = 2.1$ from 2.7 to 1 GHz and $\beta_{\rm ff} = 0$ below 1 GHz. We then combine the synchrotron maps producing a spectral index map between $408 - 2300$ MHz with a resolution of $\lesssim 2° \div 3°$ ($< \beta_{\rm sync} > \sim 2.8$). This spectral index map is used to scale the synchrotron component down to $\sim$ 10 GHz. In fact, for typical (local) values of the galactic magnetic field ($\sim 2.5\ \mu$G), the knee in the electron energy spectrum in cosmic rays ($\sim$ 15 GeV) corresponds to $\sim$ 10 GHz (Platania et al. 1998). From the synchrotron map obtained at 10 GHz and the DMR 31.5 GHz map we derive a high frequency spectral index map for scaling the synchrotron component up to Planck frequencies. These maps have a poor resolution and the synchrotron structure needs to be extrapolated to Planck angular scales. An estimate of the synchrotron angular power spectrum and of its spectral index, γ ($C_l \propto l^{-\gamma}$), has been provided by Lasenby et al. (1998); we used $\gamma = 3$ for the angular structure extrapolation (Burigana et al. 1998).

Table 1. Summary of LFI characteristics as recently revised by the LFI Consortium (Mandolesi et al. 1999). Data rates are tabulated for the case of a sampling rate equal to 8640 samples per circle and constant time and frequency

Center frequency ν [GHz]	30	44	70	100
Number of detectors $n_{\mathrm{dtc},\nu}$	8	12	24	68
Angular resolutions, FWHM [$'$]	33.6	22.9	14.4	10.0
Bandwidth [$\Delta\nu/\nu$]	0.2	0.2	0.2	0.2
$10^6 \Delta T/T$	1.6	2.4	3.6	4.3
ΔT_{ant} [μK]	5.1	7.8	10.6	12.4
ΔT_{ant} [mK] per sampling and receiver	2.06	2.61	3.16	4.36
Number of samples for beam	13.4	9.2	5.8	4.0
Data rate per Detector [Kb/sec]	2.3	2.3	2.3	2.3
Data rate per Frequency Channel [Kb/sec]	18.4	27.6	55.3	156.7
Uncompressed data rate partition function f_ν [%]	7.14	10.71	21.43	60.71

Schlegel et al. (1998) provided a map of dust emission at 100 μm merging the DIRBE and IRAS results to produce a map with IRAS resolution ($\simeq 7'$) but with DIRBE calibration quality. They also provided a map of dust temperature, T_{d}, by adopting a modified blackbody emissivity law, $I_\nu \propto B_\nu(T_{\mathrm{d}})\nu^\alpha$, with $\alpha = 2$. This can be used to scale the dust emission map to Planck frequencies using the dust temperature map as input for the $B_\nu(T_{\mathrm{d}})$ function. Unfortunately the dust temperature map has a resolution of $\simeq 1°$; again, we use an angular power spectrum $C_l \propto l^{-3}$ to scale the dust skies to the Planck proper resolution. Merging maps at different frequencies with different instrumental features and potential systematics may introduce some internal inconsistencies. More data on diffuse galactic emission, particularly at low frequency, would be extremely important.

• *Modeling the extragalactic source fluctuations* – The simulated maps of point sources have been created by an all–sky Poisson distribution of the known populations of extragalactic sources in the $10^{-5} < S(\nu) < 10$ Jy flux range exploiting the number counts of Toffolatti et al. (1998) and neglecting the effect of clustering of sources. The number counts have been calculated by adopting the Danese et al. (1987) evolution model of radio selected sources and an average spectral index $\alpha = 0$ for compact sources up to $\simeq 200$ GHz and a break to $\alpha = 0.7$ at higher frequencies (see Impey & Neugebauer 1988; De Zotti & Toffolatti 1998), and by the model C of Franceschini et al. (1994) updated as in Burigana et al. (1997a), to account for the isotropic sub-mm component estimated by Puget et al. (1996) and Fixsen et al. (1996). At bright fluxes, far–IR selected sources should dominate the number counts at High Frequency Instrument (HFI) channels for $\nu \gtrsim 300$ GHz, whereas radio selected sources should dominate at lower frequencies (Toffolatti et al. 1998).

• *Instrumental noise* – The white noise depends on instrumental performances (bandwidth $\Delta\nu$, system temperature T_{sys}), on the observed sky signal, T_{sky}, dominated by CBM monopole, and on the considered integration time, τ, according to:

$$\Delta T_{\mathrm{sky}} = \frac{\sqrt{2}(T_{\mathrm{sys}} + T_{\mathrm{sky}})}{\sqrt{\Delta\nu\ \tau}}. \qquad (1)$$

Under certain idealistic assumptions, Burigana et al. (1997b) and Seiffert et al. (1997) provide analytical estimates for the knee frequency, f_{k}, of LFI radiometers; it is predicted to critically depend also on the load temperature, T_{load}, according to:

$$f_{\mathrm{k}} = \frac{A^2\Delta\nu}{8}(1-r)^2\left(\frac{T_{\mathrm{sys}}}{T_{\mathrm{sys}} + T_{\mathrm{sky}}}\right)^2, \qquad (2)$$

where $r = (T_{\mathrm{sky}} + T_{\mathrm{sys}})/(T_{\mathrm{load}} + T_{\mathrm{sys}})$ and A is a constant, depending on the state of art of radiometer technology, which has to be minimized for reducing via hardware the knee frequency (current estimates are $A \sim 1.8\ 10^{-5}$ for 30 and 44 GHz radiometers and $A \sim 2.5\ 10^{-5}$ for 70 and 100 GHz).

Recent experimental results from M. Seiffert (1999, private communication), show knee frequency values of this order of magnitude, confirming that the present state of art of the radiometer technology is close to reach the ideal case.

A pure white noise stream can be easily generated by employed well tested random generator codes and normalizing their output to the white noise level ΔT_{sky}. A noise stream which takes into account both white noise and $1/f$ noise can be generated by using FFT methods. After generating a realization of the real and imaginary part of the Fourier coefficients with spectrum defined $S_{\mathrm{noise}}(f) \propto (1 + f_{\mathrm{k}}/f)$, we transform them and obtain a real noise stream which has to be normalized to the white noise level ΔT_{sky} (Maino et al. 1999).

• *Modeling the observed signal* – We produce full sky maps, T_{sky}, by adding the antenna temperatures from CMB, Galaxy emission and extragalactic source fluctuations. Planck will perform differential measurements and not absolute temperature observations; we then represent the final observation in a given i-th data sample in the form $T_i = R_i(T_{\mathrm{sky},i} + N_i - T_{\mathrm{ref},i})$, where N_i is the instrumental noise generated as described above.

$T_{\mathrm{ref},i}$ is a reference temperature subtracted in the differential data and R_i is a constant which accounts for the calibration. Of course, the uncertainty on R_i and the non reduced time variation of $T_{\mathrm{ref},i}$ have to be much smaller than the PLANCK nominal sensitivity. Thus, we generate the "observed" map assuming a constant value, $T_{\mathrm{ref},i}$, of $T_{\mathrm{ref},i}$ for all the data samples. We note that possible constant small off-sets in $T_{\mathrm{ref},i}$ could be in principle accepted, not compromising an accurate knowledge of the anisotropy pattern. We arbitrarily generate the "observed" map with $R_i = R = 1$ for all the data samples.

3. A model of acquisition chain

To test rigorously the efficiency of different compressors the best solution is to generate a realistically simulated signal for different mission hypotheses and apply to them the given compressors. To be realistical the simulation of the signal generation should contain both astrophysical and instrumental effects. It would be helpful that the final simulation would be able to given a hint about the influence of the various signal components and their variance. Of course it is useless to reproduce in full detail the LFI to obtain a signal simulation accurate enough to test compressors. A simplified model of the LFI, its front-end electronics and its operations will be enough.

At the base of the simplified model is the concept of *acquisition pipeline*. This pipeline is composed by all the modules which process the astrophysical signal: from its collection to the production of the final data streams which are compressed and then sent to Earth. In the real LFI, the equivalent of the acquisition pipeline may be obtained following the flow of the astrophysical information, from the telescope through the front-end electronics and the main Signal Processing Unit (SPU) to the memory of the Data Processing Unit (DPU) which is in charge to downlink it to the computer of the spacecraft and then to Earth. The acquisition pipeline is represented in Fig. 1. Since its purpose is to describe the signal processing and its parameters it must not be regarded as a representation of the true on-board electronics since some functionalities may be shared between different real modules. In this scheme *Front End* operations of the true LFI are assigned to the first simulation level, while on-board processing and compression to the second one.

The simulated microwave signal from the sky is collected and compared with the temperature of a reference load which, in our simulations, is supposed to have exactly the CMB temperature $T_0 = 2.725$ K (Mather et al. 1999)[1]. The difference ΔT expressed in μK is sampled along a scan circle producing a data stream of 60 scan circles with 8640 samples (pointings).

[1] Alternatively, sky and reference-load signals may be sampled separately and then ΔT may be compute numerically by the DPU.

Signal detection is simulated by (Bersanelli et al. 1996; Maris et al. 1998; Maris et al. 1999)

$$V_{\mathrm{out}} = \mathrm{AFO} + \mathrm{VOT} \cdot \Delta T, \tag{3}$$

where V_{out} is the detection chain output in Volts, VOT is the antenna temperature to the detector voltage conversion factor (-0.5 V/K $\le$ VOT $\le +1.5$ V/K) while AFO is a detection chain offset (-5 V $\le$ AFO $\le +5$ V). Of course in our simulation this offset takes into account all offset sources, including variations of the reference temperature, and not only of the electrical offset. Similarly the VOT factor takes into account also differences among the different detectors which affect the calibration of the temperature/voltage relation. The range for VOT and AFO is large enough to include the whole set of nominal instrumental configurations, allowing also for somewhat larger and smaller values.

The analog to digital conversion (ADC) is described by the formula:

$$V_{\mathrm{out}}^{\mathrm{adu}}(\mathrm{adu}) = \mathrm{trunc}\left(2^{N_{\mathrm{bits}}} \cdot \frac{V_{\mathrm{out}} - V_{\mathrm{min}}}{V_{\mathrm{max}} - V_{\mathrm{min}}}\right), \tag{4}$$

where trunc(.) is the decimal truncation operator, N_{bits} is the number of quantization bits produced by the ADC, while V_{min} and V_{max} are the lower and upper limits of the voltage scale accepted in input by the ADC. In our case: $N_{\mathrm{bits}} = 16$ bits, $V_{\mathrm{min}} = -10$ V, $V_{\mathrm{max}} = +10$ V. So the quantization unit "adu" (analog/digital unit) is

$$1\,\mathrm{adu} = \frac{V_{\mathrm{max}} - V_{\mathrm{min}}}{2^{N_{\mathrm{bits}}}} \tag{5}$$

or in terms of antenna temperature the quantization step is

$$\Delta = \frac{V_{\mathrm{max}} - V_{\mathrm{min}}}{2^{N_{\mathrm{bits}}}\mathrm{VOT}} \tag{6}$$

for a typical VOT = 1 V/K, $N_{\mathrm{bits}} = 16$ bits, $1\ \Delta \approx 3\ 10^{-4}$ K/adu. After digitization the simulated signal is written into a binary file of 16 bits integers and sent to the compression pipeline.

The simplified model of the Low Frequency Instrument is composed of four acquisition pipelines, one for each frequency, each one being representative of the set of devices which form the full detection channel for the given frequency. The overall data-rate after loss-less compression for LFI should be obtained summing the contribution expected from each detector. Since in the real device each radiometer for a given frequency channel, will be characterized by different values of VOT and AFO, the distribution of these parameters has to be taken in account computing the overall compression efficiency. In particular a greater attention should be devoted to the distribution of the VOT parameter since the compression efficiency is particularly sensitive to it. However, since the distribution of operating conditions and instrumental parameters are

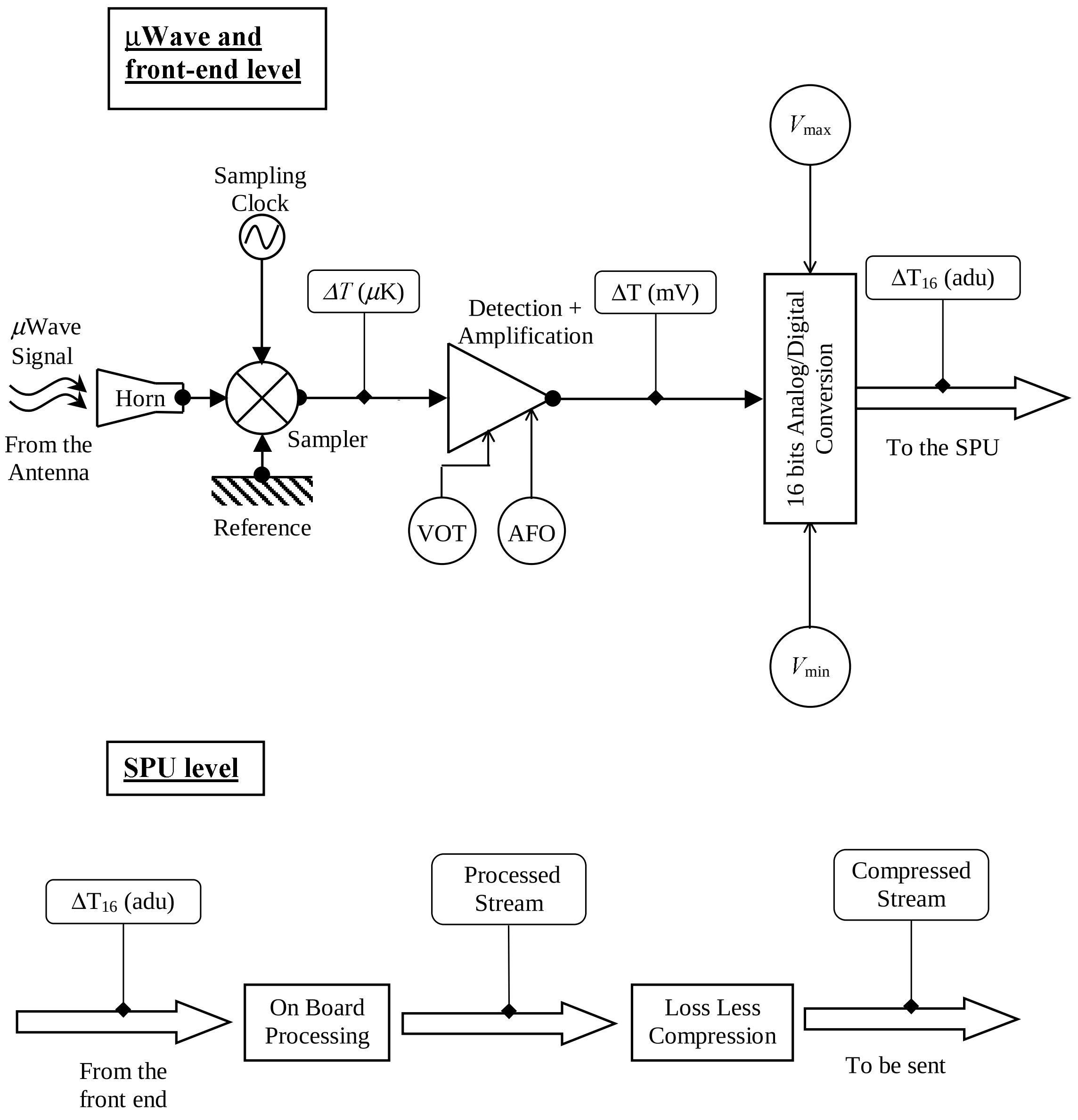

Fig. 1. Scheme for the functional model of the acquisition pipeline

not yet fully defined, we assumed that all the detectors belonging to a given frequency channel are identical[2] and located at the telescope focus.

4. An informal theoretical analysis about the compression efficiency

An informal theoretical analysis may be helpful to evaluate the maximum lossless compression efficiency expected from LFI and to discuss the behaviour of the different compressors. For further details we forward the reader to Nelson & Gailly (1996).

Data compression is based on the partition of a stream of bits into short chunks, represented by strings of bits of fixed length $N_{\rm bits}$, and to code each string of bits $S_{\rm In}$ into another string $S_{\rm Out}$ whose length $N_{\rm bits}^{\rm out}$ is variable and, in principle, shorter than $S_{\rm In}$. In this scheme, when the string of bits represents a message, the possible combinations of bits in $S_{\rm In}$ represents the *symbols* by which the *message* is encoded. From this description the compression operation is equivalent to map the input string set $\{S_{\rm In}\}$ into an output string set $\{S_{\rm Out}\}$ through a *compressing function* $\mathcal{F}_{\rm Comp}$. A compression algorithm is called *lossless* when it is possible to reverse the compression process reconstructing the $S_{\rm In}$ string from $S_{\rm Out}$ through a *decompression* algorithm. So the condition for a compression programs to be lossless is that the related $\mathcal{F}_{\rm Comp}$ is a one-to-one application of $\{S_{\rm In}\}$ into $\{S_{\rm Out}\}$. In this case the *decompressing algorithm* is the inverse function of $\mathcal{F}_{\rm Comp}$. Of course in the general case it is not possible to have at the same time lossless compression and $N_{\rm bits} > N_{\rm bits}^{\rm out}$ for any

[2] But see Sect. 10 and the Appendix for a more detailed discussion.

string in the input set. The problem is solved assuming that the discrete distribution $P(S_{\rm In})$ of strings belonging to the input stream of bits is not flat but that a most probable string exists. So a good $\mathcal{F}_{\rm Comp}$ will assign the shortest $S_{\rm Out}$ to the most probable $S_{\rm In}$ and, the least probable the input string, the longest the output string. In the worst case output strings *longer* than the input string will be assigned to those strings of $\{S_{\rm In}\}$ which are least probable. With this statistical tuning of the compression function the final length of the compressed stream will be shorter than the original length, the averaged length of $S_{\rm Out}$ being:

$$\overline{N^{\rm out}_{\rm bits}} = \sum_{S_{\rm In}\in\{S_{\rm In}\}} P(S_{\rm In}) N^{\rm out}_{\rm bits}(\mathcal{F}_{\rm Comp}(S_{\rm In})). \qquad (7)$$

Several factors affect the efficiency of a given compressor, in particular best performances are obtained when the compression algorithm is tuned on the specific distribution of symbols. Since the symbol distribution depends on $N_{\rm bits}$ and on the specific input stream, an ideal general-purpose self-adapting compressor should be able to perform the following operations: *i)* acquire the full bit stream (in the hypothesis it has a finite length) and divide it in chunks of length $N_{\rm bits}$, *ii)* perform a frequency analysis of the various symbols, *iii)* create an optimized *coding table* which associates to each $S_{\rm In}$ a specific $S_{\rm Out}$, *iv)* perform the compression according to the optimized coding table, *v)* send the coding table to the uncompressing program together with the compressed bit stream. The uncompressing program will restore the original bit stream using the associated optimized coding table.

In practice in most cases the chunks size $N_{\rm bits}$ is hardwired into the compressing code (typically $N_{\rm bits} = 8$ or 16 bits), also the fine tuning of the coding table for each specific bit stream is too expensive in terms of computer resources to be performed in this way, and the same holds for coding table transmission. So there are compressors which work as if the coding table or, equivalently, the compression function is fixed. In this way the bit stream may be compressed chunk by chunk by the compressing algorithm which will act as a filter. Other compressors perform the statistical tuning on a small set of chunks taken at the beginning of the stream, and then apply the same coding table to the full input stream. In this case the compression efficiency will be sensitive to the presence of correlations between difference parts of the input stream. In this respect self-adaptive codes may be more effective than non-adaptive ones, if their adapting strategy is sensitive to the kind of correlations in the input stream.

On the other hand other solutions may be adopted to obtain a good compromise between computer resources and compression optimization. For example all of the previous compressors are called *static* since the coding table is fixed in one way or the other at the beginning of the compression process and then used all over the input stream. Another big class of self-adaptive codes is represented by *dynamical* self-adaptive compressors, which gain the statistical knowledge about the signal as the compression proceeds changing time by time the coding table. Of course these codes compress worse at the beginning and better at the end of the data stream, provided its statistical properties are stationary. They are also able to self-adapt to remarkable changes in the characteristics of the input stream, but only if these changes may be sensed by the adapting code. Otherwise the compressor will behave worse than a well-tuned static compressor. Moreover, if the signal changes frequently, it may occur that the advantage of the dynamical self adaptability is compensated by the number of messages added to the output stream to inform the decompressing algorithm of the changes occurred to the coding table. Last but not least, if some error occurs during the transmission of the compressed stream and the messages about changes in the coding table are lost, it will be impossible to correctly restore it at the receiving station. This problem may be less severe for a static compressor since, as an example, it is possible to split the output stream in packets putting *stop codes* and storing the coding table on-board until a *confirmation message* from the receiving station is sent back to confirm the correct transmission.

It is then clear that each specific compression algorithm is *statistically optimized* for a given kind of input stream with its own statistical properties. So to obtain an optimized compressor for LFI it is important to properly characterize the statistics of the signal to be compressed and to test different existing compressors in order to map the behaviour of different compression schemes using realistically simulated signals and, as soon as possible, the true signals produced by the LFI electrical model.

In order to evaluate the performances of different compression scheme we considered the *Compression Rate* $C_{\rm r}$ defined as:

$$C_{\rm r} = \frac{L_{\rm u}}{L_{\rm c}} \qquad (8)$$

where $L_{\rm u}$ is the length of the input string in bytes and $L_{\rm c}$ is the length of the output string in bytes[3]. Other important estimators to evaluate the performances of a given compression code are the memory allocation and the compression time. Both of them must be evaluated working on the final model of the on board computer. Since this component is not fully defined for the Planck-LFI mission, in this work we neglect these aspects of the problem.

The measure represented by one of the 8640 samples which form one scan circle is white noise dominated, the rms $\sigma_{\rm T}$ being about a factor of ten higher than the CMB fluctuations signal. If so, at the first approximation it is possible to assume the digitized data stream from the

[3] Often compressors are evaluated looking at the compression efficiency $\eta_{\rm c} = 1/C_{\rm r}$ but we considered $C_{\rm r}$ more effective for our purposes.

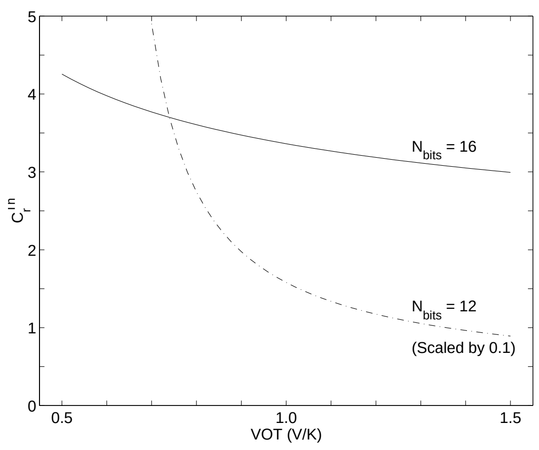

Fig. 2. $C_{\rm r}^{\rm Th}$ as a function of VOT and $N_{\rm bits}$. It is assumed $V_{\rm min} = -10$ V, $V_{\rm max} = +10$ V and $\sigma_{\rm T} = 2\,10^{-3}$ K. The curve for 12 bits is scaled by a factor 0.1 to allow a better comparison with the 16 bits curve

front-end electronics as a stationary time series of independent samples produced by a normal distributed white noise generator. In such situation symbols are represented by the quantized signal levels, and it is easy to infer the best coding table and by the information theory the expected compression rate for an optimized compressor is promptly estimated (Gaztñaga et al. 1998). In our notation, for a zero average signal:

$$C_{\rm r}^{\rm Th} = \frac{N_{\rm bits} \ln 2}{\ln(\sqrt{2\pi e}\sigma_{\rm l}/{\rm adu}) + \ln {\rm VOT}} \qquad (9)$$

where $\sigma_{\rm l}$ is the rms of the sampled signal[4].

From Eq. (9) it is possible to infer that the higher is the VOT, (i.e. higher is the ΔT resolution) the worse is the compression rate, as already observed in Maris et al. (1998) and Maris et al. (1999). The reason being the fact that as VOT is increased the number of quantization levels (i.e. of symbols) to be coded is increased and their distribution becomes more flat increasing $\overline{N_{\rm bits}^{\rm out}}$. Assuming that all the white noise is thermal in origin then $\sigma_{\rm l} \approx \sigma_{\rm T} \approx 2\ 10^{-3}$ K. With the adu defined in Eq. (5) together with the typical values of $V_{\rm min}$ and $V_{\rm max}$ assumed therein and $N_{\rm bits} = 16$ bits we have $C_{\rm r}^{\rm Th} \sim 11.09/(3.30 + \ln {\rm VOT})$. In conclusion, for VOT = 0.5, 1.0, 1.5 V/K the $C_{\rm r}^{\rm Th}$ is respectively 4.26, 3.36, 3.00. In addition Fig. 2 represents the effect of a reduction of $N_{\rm bits}$ on $C_{\rm r}^{\rm Th}$ compared to $C_{\rm r}^{\rm Th}$ for $N_{\rm bits} = 16$.

5. Statistical signal analysis

A realistic estimation of the compression efficiency must be based on a quantitative analysis of the signal statistics, which includes: statistics of the binary representation (Sect. 5.1), entropy (Sect. 5.2) and normality tests (Sect. 5.3).

[4] It has to be noted that Eq. (9) is an approximated formula which is rigorously valid when $\sigma_{\rm l}/{\rm adu} \gg 1$.

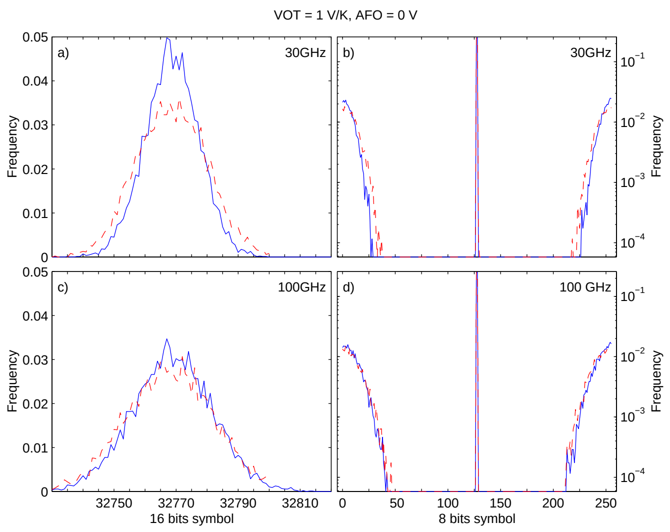

Fig. 3. Statistical distribution of 8 and 16 bits words for LFI simulated signals. Upper row is for 30 GHz, lower row for 100 GHz. Left column are the distributions of 16 bits words from the quantized signals, right column is for 8 bits words from the quantized signal, *full line* is the distribution for pure white noise, *dashed line* is the distribution for the full signal

5.1. Binary statistics

Most of the off-the-shelf compressors considered here do not handle 16 bits words, but 8 bits words. The 16 bits samples produced by the adc unit are splitted into two consecutive 8 bits (1 byte) words labeled: *most significant bits* (MSB) word and *least significant bits* (LSB) word. To properly understand the compression efficiency limits it is important to understand the statistical distribution of 8 bits words composing the quantized signal from LFI.

Figure 3 represents the frequency distribution of symbols when the full data stream of 60 scan circles is divided into 8 bits words. Since for most of the samples the range spans over ≈ 64 levels (5 bits) only the bytes corresponding to the MSB words assume a limited range of values producing the narrow spike in the figure. The belt shaped distribution at the edges is due to the set of LSB words. The distributions are quite sensitive to the quantization step, but do not change too much with the signal composition, the largest differences coming from the cosmological dipole contribution.

From the distribution in Fig. 3 one may wonder if it would not be possible to obtain a more effective compression splitting the data stream into two substreams: the MSB substream (with compression efficiency $C_{\rm r}^{\rm MSB}$) and the LSB substream (with compression efficiency $C_{\rm r}^{\rm LSB}$). Since the two components are so different in their statistics, with the MSB substream having an higher level of redundancy than the original data stream, it would be reasonable to expect that the final compression rate $2/(1/C_{\mathrm{r}}^{\mathrm{MSB}} + 1/C_{\mathrm{r}}^{\mathrm{LSB}})$ be greater than the compression rate obtained compressing directly the original data stream. We tested this procedure taking some of the compressors considered for the final test. From these tests it is clear that $C_{\mathrm{r}}^{\mathrm{MSB}} >> C_{\mathrm{r}}$ but since most of the redundancy of the original data stream is contained in the MSB substream the LSB substream can not be compressed in an effective way, as a result $C_{\mathrm{r}}^{\mathrm{LSB}} < C_{\mathrm{r}}$ and $2/(1/C_{\mathrm{r}}^{\mathrm{MSB}} + 1/C_{\mathrm{r}}^{\mathrm{LSB}}) \lesssim C_{\mathrm{r}}$. So the best way to perform an efficient compression is to apply the compressor to the full stream without performing the MSB/LSB separation. Apart from these theoretical considerations, we performed some tests with our simulated data stream confirming these result.

5.2. Entropy analysis

Equation (9) is valid in the limit of a continuous distribution of quantization levels. Since in our case the quantization step is about one tenth of the signal rms this is no longer true. To properly estimate the maximum compression rate attainable from these data we evaluate the entropy of the discretized signal using different values of the VOT.

Our entropy evaluation code takes the input data stream and determines the frequency f_s of each symbol s in the quantized data stream and computing the entropy as: $-\sum_s f_s \log_2 f_s$ where s is the symbol index. In our simulation we take both 8 and 16 bits symbols (s spanning over 0, …, 255 and 0, …, 65 535). Since in our scheme the ADC output is 16 bits, we considered 8 bits symbols entropy both for the LSB and MSB 8 bits word and 8 bits entropy after merging the LSB and MSB significant bits set.

As expected, since AFO merely shifts the quantized signal distribution, entropy does not depend on AFO. For this reason we take AFO = 0 V, i.e., no shift.

Table 2 reports the 16 bits entropy as a function of VOT, composition and frequency. As obvious entropy, i.e. information content, increases increasing VOT i.e. quantization resolution. The entropy H distribution allows to evaluate the C_{r} rms espected from different data streams realizations:

$$\mathrm{RMS}(C_{\mathrm{r}}) \approx C_{\mathrm{r}} \frac{\mathrm{RMS}(H)}{H}. \qquad (10)$$

Since data will be packed in chuncks of finite length it is important not only to study the entropy distribution for the entire data-stream, which will give an indication of the overall compressibility of the data stream as a whole, but also the entropy distribution for short packets of fixed length. So each data stream was splitted into an integer number of chunks of fixed length l_{chunck}. For each chunck

Table 2. Entropy for 16 bits samples at 30 and 100 GHz, for only White Noise and Full Signal as a function of L_{chunck}. Total Entropy refers to the entropy computed over the full set of samples (8640 × 60), Mean and RMS Entropy are the mean and RMS of different realizations of chunks of samples of length L_{chunck}. The same for C_{r} columns. Here C_{r} are derived from the corresponding values of the entropy. The quantization step is $\Delta = 0.305$ mK/adu

30 GHz, White Noise

	Entropy (bits)			C_{r}		
L_{chunck}	Total	Mean	RMS	Total	Mean	RMS
16	5.1618	3.5596	0.1989	3.10	4.49	0.251
32	5.1618	4.1815	0.1658	3.10	3.83	0.152
64	5.1618	4.6108	0.1262	3.10	3.47	0.095
135	5.1618	4.8791	0.0890	3.10	3.28	0.060
8640	5.1618	5.1561	0.0114	3.10	3.10	0.007
17280	5.1618	5.1589	0.0061	3.10	3.10	0.004

30 GHz, Full Signal

	Entropy (bits)			C_{r}		
L_{chunck}	Total	Mean	RMS	Total	Mean	RMS
16	5.5213	3.5602	0.1982	2.90	4.49	0.250
32	5.5213	4.1849	0.1664	2.90	3.82	0.152
64	5.5213	4.6162	0.1278	2.90	3.47	0.096
135	5.5213	4.8885	0.0893	2.90	3.27	0.060
8640	5.5213	5.5119	0.0176	2.90	2.90	0.009
17280	5.5213	5.5157	0.0118	2.90	2.90	0.006

100 GHz, White Noise

	Entropy (bits)			C_{r}		
L_{chunck}	Total	Mean	RMS	Total	Mean	RMS
16	5.7436	3.6962	0.1740	2.79	4.33	0.204
32	5.7436	4.4174	0.1521	2.79	3.62	0.125
64	5.7436	4.9627	0.1230	2.79	3.22	0.080
135	5.7436	5.3354	0.0875	2.79	3.00	0.049
8640	5.7436	5.7352	0.0115	2.79	2.79	0.006
17280	5.7436	5.7394	0.0063	2.79	2.79	0.003

100 GHz, Full Signal

	Entropy (bits)			C_{r}		
L_{chunck}	Total	Mean	RMS	Total	Mean	RMS
16	5.8737	3.6970	0.1734	2.72	4.33	0.203
32	5.8737	4.4186	0.1526	2.72	3.62	0.125
64	5.8737	4.9655	0.1224	2.72	3.22	0.079
135	5.8737	5.3419	0.0887	2.72	3.00	0.050
8640	5.8737	5.8604	0.0180	2.72	2.73	0.008
17280	5.8737	5.8655	0.0127	2.72	2.73	0.006

the entropy was measured, and the corresponding distribution of entropies for the given L_{chunck} as its mean and rms was obtained. We take $l_{\mathrm{chunk}} = 16$, 32, 64, 135, 8640, 17280 16-bits samples, so each simulated 8640 × 60 data stream will be splitted into 32 400, 16 200, 8100, 3840, 60, 30 chuncks. Small chunck sizes are introduced to study the entropy distribution as seen by most of the true compressors which do not compress one circle (8640 samples) at a time. Long chuncks distributions are usefull to understand the entropy distribution for the overall data-stream.

The entropy distribution per chunck is approximately described by a normal distribution (see Fig. 4), so the mean entropy and its rms are enough to characterize the results. Note however that the corresponding distribution of compression rates is not exactly normally distributed, however for the sake of this analysis we will assume that even the C_{r} distribution is normally distributed.

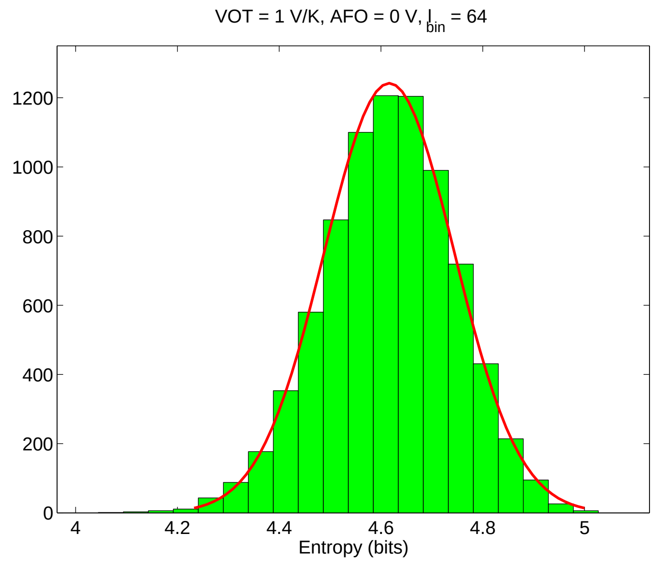

Fig. 4. The entropy distribution per bunch for $l_{\mathrm{chunk}} = 64$ samples, for the full signal at 30 GHz

The mean entropy measured over one scan circle ($l_{\mathrm{chunk}} = 8640$ samples) coincides with the entropy measured for the full set of 60 scan circles, the entropy rms being of the order of 10^{-2} bits. Consequently the expected rms for C_{r} compressing one or more circles at a time will be less than 1%.

The mean entropy and its rms are not independent quantities. Averaged entropy decreases as L_{chunck} decreases, but correspondingly the entropy rms increases. As a consequence the averaged C_{r} decreases decreasing L_{chunck}, but the fraction of chunks in which the compressor performs significantly worst than in average increases. The overall compression rate, i.e. the C_{r} referred to the full mission, being affected by them.

5.3. Normality tests

Since normal distribution of signals is assumed in Sect. 4 it would be interesting to fix how much the digitized signal distribution deviates from the normality. Also it would be important to characterize the influence of the $1/f$ noise and of the other signal components, especially the cosmic dipole, in the genesis of such deviations. To obtain an efficient compression it would be important that the samples are as more as possible statistically uncorrelated and normally distributed. In addition one should make sure that the detection chain does not cause any systematic effect which will introduce spurious not normal distributed components. This is relevant not only for the compression problem itself, which is among the data processing operations the least sensitive to small deviations from the normal distribution, but also in view of the future data reduction, calibration and analysis. For them the hypothesis of normality in the signal distribution is very important in order to allow a good separation of the foreground components. Last but not least, the hypothesis of conservation of normality along the detection chain, is important for the scientific interpretation of the results, since the accuracy expected from the PLANCK-LFI experiment should allow to verify if really the distribution of the CMB fluctuations at $l \gtrsim 14$ is normal, as predicted by the standard inflationary models, or as seems suggested by recent 4 years COBE/DMR results (Bromley & Tegmark 1999; Ferreira et al. 1999).

For this reason a set of normality tests was applied to the different components of the simulated signal before and after digitization in order to characterize the signal statistics and its variation along the detection process. Of course this work may be regarded as a first step in this direction, a true calibration of the signal statistics will be possible only when the front end electronics simulator will be available. Those tests have furthermore the value of a preparation to the study of the true signal.

Normality tests were applied on the same data streams used for data compression. Given on board memory limits, it is unlikely that more than a few circles at a time can be stored before compression, so statistical tests where performed regarding each data stream for a given pointing, as a collection of 60 independent realizations of the same process. Of course this is only approximately true. The $1/f$ noise correlates subsequent scan circles, but since its rms amplitude per sample is typically about one-tenth of the white noise rms or less, these correlations can be neglected in this analysis.

Starting from the folded data streams a given normality test was applied to each set of 60 realizations for each one of the 8640 samples, transforming the *stream of samples* in a *stream of test results* for the given test. The cumulative distribution of frequency was then computed over the 8640 test results. Since 60 samples does not represent a large statistics, significant deviations from theorethically evaluated confidence levels are expected resulting in an excessive rejection or acceptation rates. For this reason each test was calibrated applying it to the undigitized white noise data stream. Moreover, in order to analyze how the normality evolves increasing the signal complexity, tests was repeated increasing the information content of the generated data stream.

To simplify the discussion we considered as a reference test the usual Kolmogorow - Smirnov D test from Press et al. (1986) and we fix a 95% acceptance level. The test was "calibrated" using the Monte-Carlo white noise generator of our mission simulator in order to fix the threshold level D_{th} as the D value for which more than 95% of our samples show $D \leq D_{\mathrm{th}}$. From Table 3 the quantization effect is evident, at twice the nominal quantization step (VOT = 2 V/K) in 30% of the samples (i.e. 2592 samples) the distribution of realizations deviates from a normal distribution ($D > D_{\mathrm{th}}$). Since the theoretical compression rate from Eq. (9) is for a continuous

Table 3. Quantization effect on the Kolmogorow - Smirnov D test applied to simulated data, Δ is the quantization step

	Δ (mK/adu)		
	1.220	0.610	0.406
$\mathcal{F}(D < 0.1475,$ White Noise)	0.28	0.70	0.84
$\mathcal{F}(D < 0.1475,$ Signal)	0.27	0.71	0.86
$D^{\rm Q}_{95}$	0.2449	0.1851	0.1678
$\mathcal{F}(D < D^{\rm Q}_{95},$ Signal)	0.95	0.95	0.95

distribution of levels ($\sigma \gg \Delta$) a smaller $C_{\rm r}$ should be expected. Since the deviation from the normal distribution is a systematic effect, for the sake of cosmological data analysis one may tune the D test to take account of the quantization. As an example, the third line in Table 3 reports the threshold for the quantized signal $D^{\rm Q}_{\rm th}$ for which 95% of the quantized white noise samples are accepted as normal distributed. The line below represents the success rate for the full quantized signal. After the recalibration the test is able to recognize that in 95% of the cases the signal is drawn from a normal distribution, but at the cost of a growth in the threshold D which now is a function of the quantization step Δ.

As for the entropy distribution and the binary statistics, even in this case most of the differences between the results obtained for a pure white noise signal and the full signals are explained by the presence of the cosmological dipole. However these simulations are not accurate enough to draw any quantitative conclusions about the distortion in the sampling statistics induced by digitization, but they suggest that to approximate the instrumental signal as a quantized white noise plus a cosinusoidal term associated to the cosmic dipole is more than adequate in order to understand the optimal loss-less compression rate achievable in the case of the Planck-LFI mission.

6. Quantization and quantization error

A possible solution to solve the bandwidth problem is to reduce the amount of information of the sampled signal i.e. its entropy. Independently from the way in which this is performed, the final compression strategy will be lossy, and the final reconstructed (uncompressed) signal will be corrupted with respect to the original one, degrading in some regard the experimental performances. In this regard, any sort of lossy compression may be seen as a kind of signal rebinning with a coarser resolution (quantization step) in $\Delta T/T$.

There are at least six aspects in Planck-LFI operations which may be affected by a coarser quantization:

1. C_l and periodical signals reconstruction;
2. destriping;
3. foreground separation;
4. point like sources detection;
5. variable sources characterization;
6. tests for normality of CMB fluctuations.

Since the non linear nature of the quantization process, all of them are hard to be analytically evaluated and for this reason a specific simulation task is in progress for the Planck-LFI collaboration (White & Seyfert 1999; Maris et al. 2000). However a heuristic evaluation for the point (1) by analytical means is feasible.

Quantization operates a convolution of the normal distribution of the input signal with the quantization operator $(x : \Delta) = \texttt{sign}(x)\Delta * \mathrm{floor}(|x/\Delta|)$. If the quantization error: $(x - (x : \Delta))$ is uniformly distributed its expectation is $\Delta/2$ and its rms is $\Delta/\sqrt{12}$ (Kollár 1994). Quantization over a large amount of samples may be regarded as an extra source of noise which will enhance the variance per sample. If the quantization error is statistically independent from the input quantized signal and if it may be added in quadrature to the white noise variance $\sigma_{\rm WN}$, the total variance per sample will be $\approx {\sigma_{\rm WN}}^2 \left(1 + \frac{\Delta^2/{\sigma_{\rm WN}}^2}{12}\right)$. So for $\Delta \lesssim \sigma_{\rm WN}$ the expected quantization rms is $\lesssim 4\%$. From error propagation the relative error on the C_l is (Maino 1999):

$$\frac{\delta C_l}{C_l} = \sqrt{\frac{4\pi}{A}}\sqrt{\frac{2}{(2l+1)f_{\rm sky}}}\left[1 + \frac{\sigma^2\theta^2}{B_l^2 C_l}\right] \quad (11)$$

so that the quantization contribution to the overall error will be small and dominated by the cosmic variance for a large set of l. However the application of such encouraging result must be considered carefully in a true experimental framework. Apart from the assumptions, it has to be demonstrated indeed that a large quantization error like this will not harm significantly the aforementioned aspects, moreover the impact of signal quantization will depend on how and in which point of the detection chain it will be performed.

7. Experimental evaluation of off-the-shelf compressors

This section describes the evaluation protocol and the experimental results of the compression of simulated data streams for Planck-LFI.

7.1. Evaluation protocol

First tests were performed on a HP-UX workstation on four compressors (Maris et al. 1998) but given the limited number of off-the-shelf compression codes for such platform, we migrated the compression pipeline on a Pentium III based Windows/NT workstation.

As described in Sect. 2 the signal composition is defined by many components, both astrophysical and instrumental in origin. In particular, it is important to understand how each component or instrumental parameter, introducing deviations to the pure white noise statistics, affects the final compression rate.

To scan systematically all the relevant combinations of signal compositions and off-the-shelf compressors, a Compression Pipeline was created. The pipeline is based on five main components: the signal quantization pipeline, the signal database, the compression pipeline, the compression data base, the post-processing pipeline. The signal quantization pipeline performs the operations described in the upper part of Fig. 1. The simulated astrophysical signals are hold in a dedicated section of the signal archive, they are processed by the quantization pipeline and stored back in a reserved section of the signal archive. So quantized data streams are generated for each relevant combination of the quantization parameters, signal composition and sky pointing.

Each compressor is then applied by the compression pipeline to the full set of quantized signals in the signal archive. Results, in terms of compression efficiency as a function of quantization parameters are stored in the compression database. The statistical analysis of Sect. 5 are performed with a similar pipeline.

Finally the post-processing pipeline scans the compression data base in order to produce plots, statistics, tables and synthetic fits. Its results are usually stored into one of the two databases.

The pipeline is managed by `PERL` 5.004 script files which drive FORTRAN, C, IDL programs or on-the-shelf utilities gluing and coordinating their activities. Up to $\approx 75\,000$ lines of repetitive code are required per simulation run. They are generated by a specifically designed Automated Program Generator (APG) written in IDL (Maris et al. 1998). The APG takes as an input a table which specifies: the set of compressors to be tested, the set of quantization parameters to be used, the order in which to perform the scan of each parameter/compressor, the list of supporting programs to be used, other servicing parameters. The program linearizes the resulting parameter space and generates the `PERL` simulation code or, alternatively, performs other operations such as: to scan the results data base to produce statistics, plots, tables, and so on. The advantage of this method is that a large amount of repetitive code, may be quickly produced, maintained or replaced with a minor effort each time a new object (compressor, parameter or analysis method) is added to the system.

7.2. Experimental results

Purpose of these compression tests is to give an upper limit to the lossless compression efficiency for LFI data and to look for an optimal compressor to be proposed to the LFI consortium.

A decision about the final compression scheme for Planck-LFI has not been taken yet and only future studies will be able to decide if the best performing one will be compatible with on-board operations (constrained by: packet independence and DPU capabilities) and will be accepted by the Planck-LFI collaboration.

For this reason up to now only off-the-shelf compressors and hardware where considered. To test any reasonable compression scheme a wide selection of lossless compression algorithms, covering all the known methods, was applied to our simulated data. Lacking a comprehensive criteria to fix a final compressor, as memory and CPU constrains, we report in a compact form the results related to all the tested compressors. We are confident that the experience which will be gained inside the CMB community developing ground based experiments (i.e. Lasenby et al. 1998), long duration flight balloon experiments (i.e. De Bernardis & Masi 1998) such as BOOMERANG (De Bernardis et al. 2000; Lange et al. 2000) and MAXIMA (Hanany et al. 2000; Balbi et al. 2000), and specialized space missions such as the Microwave Anisotropy Probe (MAP) (Bennet et al. 1996), joined to the experience which will be gained inside the Planck collaboration developing on-board electronics prototypes, will provide us with a more solid base to test and improve the final compression algorithms applied to real data.

Tables 4 and 5 list the selected compression programs. Since the behaviour (and efficiency) of each compressor is determined by a set of parameters, one or more *macro file* operating a given combination of compressor code plus parameters is defined. It has to be noted that `uses` is a space qualified algorithm, based on Rice compression method, for which space qualified dedicated hardware already exists.

To evaluate the performances of each compressor, *figures of merit* are drawn like the one in Fig. 5 which shows the results for the best performing compressor: `arith-n1`. Looking at such figures it is possible to note as the compression efficiency does not depend much on the signal composition. This is true even when large, impulsive signals, as planets, affecting few samples over thousands are introduced. Again, this is a consequence of the fact that white noise dominates the signal, being the most important component to affect the compression efficiency. In this regard it has been speculated that the $1/f$ component should improve the correlation between neighborhood samples affecting the compression efficiency (Maris et al. 1998) no relevant effect may be detected into our simulations. As an example from Fig. 5 for the 30 GHz signal the addition of the $1/f$ noise to the white noise data stream affects the final $C_{\rm r}$ for less than 0.5%.

The only noticeable (i.e. about 6%) effect due to an increase in the signal complexity, occurs when the cosmic dipole is added. In the present signal the dipole amplitude is comparable with the white noise amplitude (≈ 3 mK) so its effect is to distort the sample distribution, making it leptocurtic. As a consequence compressors, which usually work best for a normal distributed signal, becomes

Table 4. Tested compressors and related parameters. The *Macro* column contains the names of the macros running a given compression *Code* with a given combination of *Parameters*

Macro	Code	Parameters	Note
ahuff-c	ahuff-c		Adaptive Huffman (Nelson & Gailly 1996)
AR	ar		
arc	arc		
arha	arhangel		http://www.geocities.com/SiliconValley/Lab/6606
arhaASC	”	−1	ASC method
arhaHSC	”	−2	HSC method
arith-c	arith-c		Arithmetic coding (Nelson & Gailly 1996)
arith-n	arith-n		Adaptive Arithmetic Coding (AC) (Nelson & Gailly 1996)
arith-n0	”	−o0	Zeroth order Arithmetic coding
arith-n1	”	−o1	First order AC
arith-n2	”	−o2	Second order AC
arith-n3	”	−o3	Third order AC
arith-n4	”	−o4	Fourth order AC
arith-n5	”	−o5	Fifth order AC
arith-n6	”	−o6	Sixth order AC
arith-n7	”	−o7	Seventh order AC
arj	arj		
arj0	”	−m0	method 0 (no compression)
arj1	”	−m1	method 1
arj2	”	−m2	method 2
arj3	”	−m3	method 3
arj4	”	−m4	method 4
boa	boa		
bzip	bzip2090		
bziprb	”	–repetitive-best	best compression of repetitive blocks
bziprf	”	–repetitive-fast	fast compression of repetitive blocks
gzip1	gzip	−1	fast compression
gzip9	”	−9	best compression
huff-c	huff-c		Hauffman (Nelson & Gailly 1996)
jar	jar32		
jar1	”	−m1	method 1
jar2	”	−m2	method 2
jar3	”	−m3	method 3
jar4	”	−m4	method 4
lha	lha		
lzss	lzss		
lzw12	lzw12		
lzw15v	lzw15v		

less effective. Since the dipole introduces correlations over one full scan circle, i.e. few 10^3 samples, while compressors establish the proper coding table observing the data stream for a small set of consecutive samples (from some tens to some hundred samples), even a self adaptive compressor will likely loose the correlation introduced by the dipole. A proper solution to this problem is suggested in Sect. 9. The other signal components do not introduce any noticeable systematic effect. The small differences shown by the figures of merit may be due to the compression variance and depend strongly on the compressor of choice. As an example a given compressor may be more effective to compress the simulated data stream with the full signal than the associated simpler data stream containing only white noise, $1/f$ noise, CMB and dipole. At the same time another compressor may show an opposite behaviour.

As shown by Fig. 6, and as expected from Eq. (9) increasing VOT, i.e. increasing the quantization step, increases the compression rate. In addition C_r increases increasing N_c up to a $\approx 20\%$. The increase is noticeable for $N_\mathrm{c} < 15$ and saturates after $N_\mathrm{c} = 30$. On the contrary its dependence on the offset (AFO) is negligible (less than 1%). For these reasons in the subsequent analysis the AFO dependency is neglected and the corresponding simulations are averaged.

7.3. Synthetical description

The full data base of simulated compression results takes about 14 MBytes, for practical purposes it is possible to synthesize all this information using a phenomenological relation which connects C_r with N_c and VOT whose free parameters may be fitted using the data obtained from the simulations. In short:

$$C_\mathrm{r}^\mathrm{Fit}(\mathrm{VOT}, N_\mathrm{c}) = \frac{C_{\mathrm{r},1}}{\mathcal{I}(N_\mathrm{c}) + \mathcal{S}(N_\mathrm{c}) \ln\left[\frac{\mathrm{VOT}}{1\ \mathrm{V/K}}\right]} \tag{12}$$

where $C_{\mathrm{r},1}$ is the C_r for $N_\mathrm{c} = 1$, VOT = 1.0 V/K, while $\mathcal{I}(N_\mathrm{c})$ and $\mathcal{S}(N_\mathrm{c})$ describe the C_r dependence on N_c. In particular the relation is calibrated for any compressor imposing that $C_\mathrm{r}(\mathrm{VOT} = 1\ \mathrm{V/K}, N_\mathrm{c} = 1) = C_{\mathrm{r},1}$.

The linear dependency of $1/C_\mathrm{r}^\mathrm{Fit}$ over $\ln \mathrm{VOT}$ is a direct consequence of Eq. (9), and is confirmed by a set of tests performed over the full set of our numerical results

Table 5. See Table 4

Macro	Code	Parameters	Note
pkzip	pkzip		from PKWARE
pkzip-ef	”	-ef	fast compression
pkzip-en	”	-en	normal compression
pkzip-es	”	-es	super fast compression
pkzip-ex	”	-ex	extra compression
rar-m0	rar	-m0	level 0 compression
rar-m1	”	-m1	level 1 compression
rar-m2	”	-m2	level 2 compression
rar-m3	”	-m3	level 3 compression
rar-m4	”	-m4	level 4 compression
rar-m5	”	-m5	level 5 compression
splint	splint		
SZIP00	szip		Rice Algorithm and Rice compression chip simulator
szip0ec	”	-ec	entropy coding compression mode
szip0nu	”	-nn	nearest neighbor compression mode
szipc0	”	-chip	compress exactly as chip
SZIPCEC	”	-chip -ec	as szip0ec + chip compression
SZIPCNU	”	-chip -nn	as szip0nu + chip compression
uses	uses	-n 16 -s 64 -rr	Universal Source Encoder for Space 16 bits per sample, 64 samples for scanline, correlates near samples (CNS)
uses008	”	-n 16 -s 8 -j 8	8 samples, 8 samples per block
uses008rr	”	-n 16 -s 8 -rr -j 8	as uses008 + CNS
uses016	”	-n 16 -s 16	16 samples per block
uses016rr	”	-n 16 -s 16 -rr	16 samples per block + CNS
uses032	”	-n 16 -s 32	32 samples per block
uses032rr	”	-n 16 -s 32 -rr	32 samples per block + CNS
uses064	”	-n 16 -s 64	64 samples per block
uses064rr	”	-n 16 -s 64 -rr	64 samples per block + CNS
uses320	”	-n 16 -s 320	320 samples per block
uses320rr	”	-n 16 -s 320 -rr	320 samples per block + CNS
uses960	”	-n 16 -s 960	960 samples per block
uses960rr	”	-n 16 -s 960 -rr	960 samples per block + CNS
zoo	zoo		

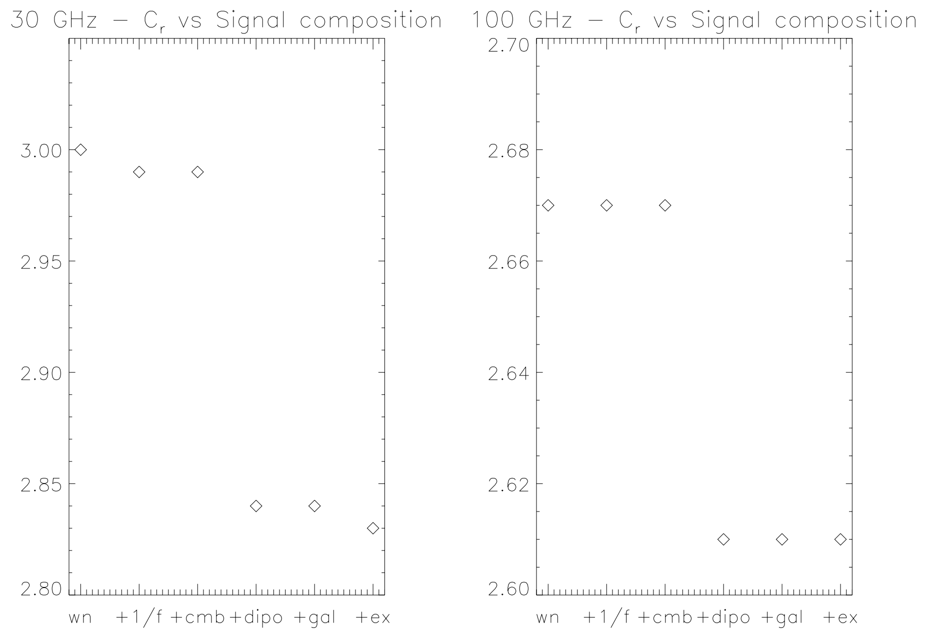

Fig. 5. Figures of merit for the arithmetic compression of order 1 (`arith-n1`) for 30 GHz and 100 GHz channels. Here AFO = 0 V, VOT = 1.0 V/K, $N_c = 1$. The compression efficiency is plotted as a function of the incremental complexity of the signal composition: `wn` means white noise only, `+1/f` plus $1/f$, `+cmb` plus CMB, `+dipo` plus dipole, `+gal` plus galaxy, `+ex` plus extragalactic sources

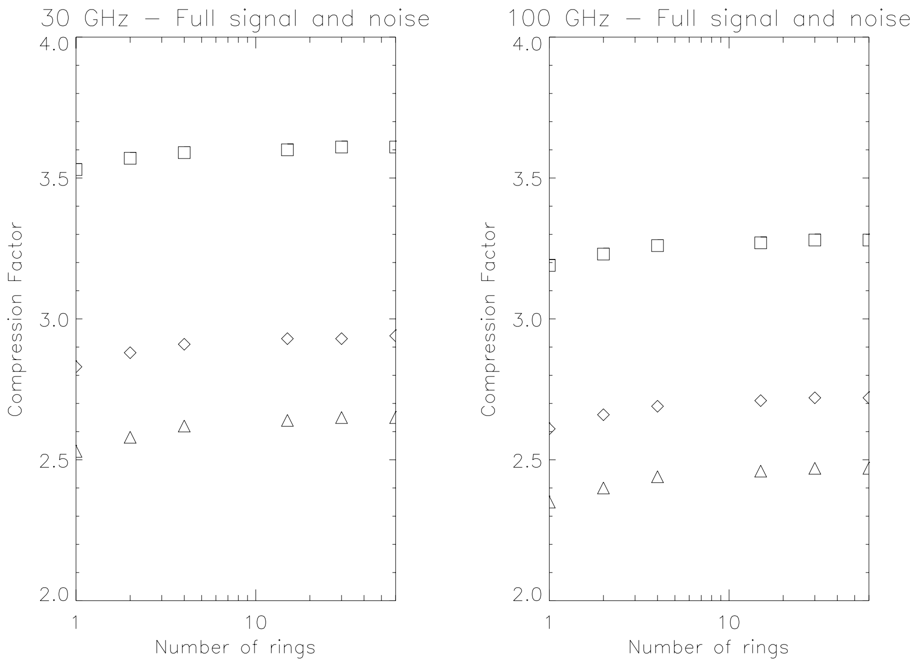

Fig. 6. Compression rates for `arith-n1` as a function of the VOT and N_c for a full simulated signal (`wn` + `1/f` + `dp` + `cmb` + `dipo` + `gal` + `ex`) (see also Fig. 5 for details). From top to bottom: *Squares*: VOT = 1.5 V/K, *Diamonds*: VOT = 1.0 V/K, *Triangles*: VOT = 0.5 V/K

for the compression efficiency, the rms residual between the best fit (12) and simulated data being less than 1.5%, in almost the 92% of the cases and less than 1% in 72% of the cases. The dependencies of its parameters $\mathcal{I}$ and $\mathcal{S}$ over N_c are obtained by a test-and-error method performed on our data set and we did not investigate further on their nature. For all practical purposes our analysis shows that these functions are well approximated by a series expansion:

$$\mathcal{I}(N_c) \approx \exp\left(\sum_{k=1}^{2} A_k (\ln N_c)^k\right), \qquad (13)$$

$$\mathcal{S}(N_c) \approx \mathcal{S}_1 \exp\left(\sum_{k=1}^{5} B_k (\ln N_c)^k\right). \qquad (14)$$

here $\mathcal{S}_1$, A_k and B_k are free parameters obtained by fitting the simulated data, in particular $\mathcal{S}_1$ is the slope for $N_c = 1$.

Since an accuracy of some percent in determining the free parameters of $C_r^{\mathrm{Fit}}(\mathrm{VOT}, N_c)$ is enough, the fitting procedure was simplified as follow. For a given compressor, signal component, swap status, and N_c value $\mathcal{I}$ and $\mathcal{S}$ where determined by a χ^2 fitting procedure. The list of $\mathcal{I}$ and $\mathcal{S}$ as a function of N_c have been fitted by using relations (13) and (14) respectively. The fitting algorithm tests different degrees of the polynomial in the aforementioned relations (up to 2 for $\mathcal{I}(N_c)$, up to 5 for $\mathcal{S}(N_c)$) stopping when the maximum deviation of the fitted relation respect to the data is smaller than 0.5% for $\mathcal{I}$ or 0.0001 for $\mathcal{S}$, or when the maximum degree is reached.

Tables 6-9 report the results of the compression exercise ordered for decreasing $C_{r,1}$. The first column is the name of compression macro (i.e. a given compression program with a defined selection of modificators and switches) as listed in Tables 4, 5. The third and fourth column are the fitted $C_{r,1}$ and $\mathcal{S}_1$ as defined in: (12), (13), (14). From the 5th to the 7th columns and from the 8th to the 13th columns the polynomial degree and the expansion parameters for Eqs. (13) and (14) are reported.

Many compressors are sensitive to the ordering of the Least and Most Significant Bytes of a 16 bits word in the computer memory and files. Two ordering conventions are assumed: *little Endian* memory layout i.e. Least Significant Byte is stored First or *big Endian* memory layout i.e. Most Significant Byte is stored First; Intel processors, as `PENTIUM III`, work with little Endian memory layout. For this reason each test was repeated twice, one time with the original data stream file, i.e. with the little Endian configuration, and the other after swapping bytes, so to simulate big Endian configuration. If the gain in $C_{r,1}$ after byte swapping is bigger than some percent, big Endian compression is reported, otherwise the little Endian is reported. These two cases are distinct by the second column of Tables 6-9 which is marked with a *y* if swapping is applied before compressing. It is interesting to note that not only 16 bits compressors, such as `uses`, are sensitive to swapping. Also many 8 bits compressors are sensitive to it, probably this is due to the fact that if the most probable 8 bits symbol is presented first at the compressor a slightly better balanced coding table is built.

Table 6. Compression rates at 30 GHz, white noise only. A *y* in the *Swap* column means that the compression efficiency is some percent better using a big Endian bytes ordering than with a little Endian bytes ordering. The polynomial degree D, as coefficients A_1, A_2, B_1, …, B_5 refer to Eqs. (12-14)

Macro	Swap	$C_{\rm r,1}$	$\mathcal{S}_1$	D	A_1	A_2	D	B_1	B_2	B_3	B_4	B_5
arith-n1		3.00	0.293	2	-0.0197	0.00314	5	-0.07749	0.03629	-0.00978	0.00153	-0.00010
BZIP	y	2.85	0.268	2	-0.0169	0.00273	5	-0.06467	0.06598	-0.03615	0.00918	-0.00084
bziprb	y	2.85	0.268	2	-0.0169	0.00273	5	-0.06467	0.06598	-0.03615	0.00918	-0.00084
bziprf	y	2.85	0.268	2	-0.0169	0.00273	5	-0.06467	0.06598	-0.03615	0.00918	-0.00084
arith-n2		2.82	0.324	2	-0.0453	0.00601	3	-0.10166	0.01394	-0.00060	0.00000	0.00000
boa	y	2.81	0.247	1	-0.0129	0.00000	5	0.06445	-0.08272	0.03984	-0.00629	0.00023
arhaHSC	y	2.68	0.281	2	-0.0367	0.00525	5	0.13641	-0.22929	0.14843	-0.03812	0.00343
arha	y	2.68	0.281	2	-0.0367	0.00525	5	0.13641	-0.22929	0.14843	-0.03812	0.00343
uses320rr	y	2.68	0.241	0	0.0000	0.00000	1	-0.00036	0.00000	0.00000	0.00000	0.00000
uses032rr	y	2.68	0.241	0	0.0000	0.00000	1	-0.00048	0.00000	0.00000	0.00000	0.00000
uses960rr	y	2.68	0.241	0	0.0000	0.00000	1	-0.00036	0.00000	0.00000	0.00000	0.00000
uses064rr	y	2.68	0.241	0	0.0000	0.00000	1	-0.00024	0.00000	0.00000	0.00000	0.00000
uses	y	2.68	0.241	0	0.0000	0.00000	1	-0.00024	0.00000	0.00000	0.00000	0.00000
uses016rr	y	2.68	0.241	0	0.0000	0.00000	1	-0.00043	0.00000	0.00000	0.00000	0.00000
uses960	y	2.67	0.241	0	0.0000	0.00000	1	-0.00035	0.00000	0.00000	0.00000	0.00000
arith-n3	y	2.67	0.329	2	-0.0586	0.00639	5	-0.03826	-0.02445	0.00438	0.00045	-0.00009
arith-n	y	2.67	0.329	2	-0.0586	0.00639	5	-0.03826	-0.02445	0.00438	0.00045	-0.00009
uses320	y	2.66	0.239	0	0.0000	0.00000	1	-0.00044	0.00000	0.00000	0.00000	0.00000
uses064	y	2.61	0.231	0	0.0000	0.00000	1	-0.00031	0.00000	0.00000	0.00000	0.00000
uses008rr	y	2.59	0.233	0	0.0000	0.00000	5	-0.01446	0.01486	-0.00603	0.00118	-0.00008
uses032	y	2.54	0.222	0	0.0000	0.00000	1	-0.00028	0.00000	0.00000	0.00000	0.00000
arith-n4	y	2.43	0.323	2	-0.0555	0.00272	3	0.06685	-0.04451	0.00450	0.00000	0.00000
uses016	y	2.42	0.204	0	0.0000	0.00000	1	-0.00043	0.00000	0.00000	0.00000	0.00000
arith-n5	y	2.27	0.298	1	-0.0484	0.00000	5	0.13951	-0.09635	0.02539	-0.00255	0.00001
arhaASC	y	2.26	0.260	2	-0.0356	0.00552	5	-0.07916	0.08148	-0.03842	0.00828	-0.00066
splint	y	2.24	0.202	2	-0.0143	0.00221	5	-0.04739	0.04058	-0.01775	0.00341	-0.00022
arith-n6	y	2.23	0.245	1	-0.0311	0.00000	5	0.13684	-0.11311	0.05274	-0.00867	0.00039
arith-n7	y	2.20	0.204	1	-0.0178	0.00000	5	0.07409	-0.09937	0.07372	-0.01523	0.00093
uses008	y	2.14	0.169	0	0.0000	0.00000	1	-0.00046	0.00000	0.00000	0.00000	0.00000
rar-m4		2.14	0.264	2	-0.0514	0.00693	5	0.07531	-0.21653	0.13994	-0.03549	0.00317
rar-m5		2.14	0.263	2	-0.0514	0.00688	5	0.06084	-0.19110	0.12472	-0.03163	0.00282
lha		2.13	0.259	2	-0.0288	0.00450	5	-0.03262	0.01456	-0.00261	-0.00001	0.00004
ar		2.13	0.259	2	-0.0288	0.00450	5	-0.03262	0.01456	-0.00261	-0.00001	0.00004
arj2		2.13	0.262	2	-0.0416	0.00632	3	-0.04587	0.01553	-0.00175	0.00000	0.00000
arj1		2.12	0.260	2	-0.0443	0.00666	5	-0.04412	0.01608	-0.00007	-0.00096	0.00014
arj		2.12	0.260	2	-0.0443	0.00666	5	-0.04412	0.01608	-0.00007	-0.00096	0.00014
gzip9		2.12	0.258	2	-0.0428	0.00614	5	-0.12294	0.15123	-0.07645	0.01726	-0.00143
rar-m3	y	2.11	0.262	2	-0.0393	0.00568	5	-0.02045	-0.08203	0.05953	-0.01523	0.00136
pkzip-en		2.10	0.255	2	-0.0464	0.00674	5	-0.03817	0.01428	0.00182	-0.00163	0.00021
pkzip-ex		2.10	0.255	2	-0.0486	0.00694	5	-0.04753	0.03824	-0.01546	0.00317	-0.00025
lzw15v		2.09	0.300	2	-0.0590	0.00907	5	-0.01981	-0.11244	0.09067	-0.02468	0.00227
arj3	y	2.06	0.245	2	-0.0344	0.00521	5	-0.02355	-0.02681	0.02433	-0.00677	0.00064
pkzip-ef	y	2.06	0.244	2	-0.0211	0.00323	5	-0.06763	0.07218	-0.03857	0.00950	-0.00085
RAR-M2	y	2.05	0.248	2	-0.0408	0.00573	5	0.12518	-0.24633	0.13212	-0.02959	0.00242
zoo		2.05	0.294	2	-0.0322	0.00527	5	0.24589	-0.43331	0.26504	-0.06708	0.00604
lzw12		2.04	0.286	2	-0.0654	0.01029	5	0.02352	-0.05311	0.03471	-0.00888	0.00080
rar-m1	y	2.03	0.244	2	-0.0310	0.00451	5	0.04862	-0.16067	0.09828	-0.02402	0.00210
arc		2.02	0.314	2	-0.0157	0.00263	5	-0.04756	0.07516	-0.04521	0.01165	-0.00107
jar4	y	2.00	0.245	2	-0.0774	0.01092	5	-0.05999	0.03876	-0.04084	0.01335	-0.00134
jar3	y	2.00	0.245	2	-0.0776	0.01095	5	-0.06825	0.05087	-0.04809	0.01525	-0.00152
gzip1	y	2.00	0.215	1	-0.0019	0.00000	5	0.01435	-0.02354	0.02012	-0.00621	0.00064
jar1	y	1.99	0.238	2	-0.0741	0.01074	5	-0.11661	0.11613	-0.08410	0.02431	-0.00236
jar2	y	1.99	0.238	2	-0.0748	0.01071	5	-0.11091	0.11461	-0.08690	0.02524	-0.00243
jar	y	1.99	0.239	2	-0.0747	0.01069	5	-0.13360	0.14765	-0.10536	0.02968	-0.00282
pkzip-es		1.95	0.178	1	-0.0031	0.00000	5	0.00212	-0.00385	0.00547	-0.00211	0.00025
arith-c		1.94	0.183	0	0.0000	0.00000	5	-0.02384	0.00689	0.00415	-0.00215	0.00028
ahuff-c		1.94	0.178	1	-0.0016	0.00000	5	-0.04099	0.02531	-0.00863	0.00171	-0.00014
huff-c		1.93	0.181	0	0.0000	0.00000	3	-0.02798	0.01088	-0.00093	0.00000	0.00000
arith-n0		1.91	0.193	0	0.0000	0.00000	5	-0.02497	0.03612	-0.02016	0.00504	-0.00045
arj4	y	1.87	0.228	1	-0.0050	0.00000	5	0.03257	-0.04890	0.02528	-0.00574	0.00049
lzss		1.60	0.337	2	-0.0281	0.00452	5	-0.00380	-0.01844	0.01154	-0.00261	0.00021

Table 7. Compression rates at 30 GHz, full signal

Macro	Swap	$C_{r,1}$	S_1	D	A_1	A_2	D	B_1	B_2	B_3	B_4	B_5
arith-n1		2.83	0.287	2	-0.0215	0.00322	5	-0.05481	-0.02562	0.02755	-0.00749	0.00068
boa	y	2.73	0.244	1	-0.0131	0.00000	5	0.03906	-0.00609	-0.01570	0.00802	-0.00099
BZIP	y	2.72	0.265	2	-0.0224	0.00316	5	-0.09587	0.09339	-0.04818	0.01169	-0.00104
bziprb	y	2.72	0.265	2	-0.0224	0.00316	5	-0.09587	0.09339	-0.04818	0.01169	-0.00104
bziprf	y	2.72	0.265	2	-0.0224	0.00316	5	-0.09587	0.09339	-0.04818	0.01169	-0.00104
arith-n2	y	2.68	0.313	2	-0.0444	0.00540	5	-0.05389	-0.04802	0.03071	-0.00701	0.00059
uses016rr	y	2.67	0.239	0	0.0000	0.00000	3	0.00500	-0.00238	0.00032	0.00000	0.00000
uses	y	2.67	0.239	0	0.0000	0.00000	1	0.00075	0.00000	0.00000	0.00000	0.00000
uses320rr	y	2.67	0.240	0	0.0000	0.00000	1	0.00063	0.00000	0.00000	0.00000	0.00000
uses064rr	y	2.67	0.239	0	0.0000	0.00000	1	0.00075	0.00000	0.00000	0.00000	0.00000
uses960rr	y	2.67	0.240	0	0.0000	0.00000	1	0.00063	0.00000	0.00000	0.00000	0.00000
uses032rr	y	2.67	0.240	0	0.0000	0.00000	2	0.00130	-0.00019	0.00000	0.00000	0.00000
uses960	y	2.66	0.239	0	0.0000	0.00000	1	0.00067	0.00000	0.00000	0.00000	0.00000
uses320	y	2.66	0.237	0	0.0000	0.00000	2	0.00192	-0.00029	0.00000	0.00000	0.00000
uses064	y	2.60	0.230	0	0.0000	0.00000	1	0.00072	0.00000	0.00000	0.00000	0.00000
uses008rr	y	2.58	0.231	0	0.0000	0.00000	3	0.00461	-0.00175	0.00022	0.00000	0.00000
arha	y	2.58	0.266	2	-0.0355	0.00330	5	0.10870	-0.10860	0.04159	-0.00656	0.00036
arhaHSC	y	2.58	0.266	2	-0.0355	0.00330	5	0.10870	-0.10860	0.04159	-0.00656	0.00036
uses032	y	2.53	0.220	0	0.0000	0.00000	1	0.00069	0.00000	0.00000	0.00000	0.00000
arith-n	y	2.53	0.318	2	-0.0590	0.00567	5	0.00372	-0.06319	0.02159	-0.00347	0.00025
arith-n3	y	2.53	0.318	2	-0.0590	0.00567	5	0.00372	-0.06319	0.02159	-0.00347	0.00025
uses016	y	2.41	0.203	0	0.0000	0.00000	3	0.00434	-0.00169	0.00021	0.00000	0.00000
arith-n4	y	2.32	0.300	1	-0.0447	0.00000	3	0.12234	-0.05802	0.00552	0.00000	0.00000
splint	y	2.23	0.211	1	-0.0052	0.00000	5	-0.00174	-0.04665	0.03588	-0.00974	0.00091
arhaASC	y	2.21	0.253	2	-0.0286	0.00444	5	-0.04378	0.04738	-0.02920	0.00788	-0.00074
arith-n5	y	2.18	0.270	2	-0.0431	-0.00114	5	0.21484	-0.10725	0.01519	0.00138	-0.00037
arith-n6	y	2.14	0.229	1	-0.0301	0.00000	5	0.15861	-0.08982	0.01690	0.00317	-0.00076
uses008	y	2.14	0.167	0	0.0000	0.00000	3	0.00609	-0.00214	0.00025	0.00000	0.00000
arith-n7	y	2.13	0.198	1	-0.0170	0.00000	5	0.09893	-0.12948	0.06164	-0.00701	-0.00005
rar-m4		2.08	0.260	2	-0.0410	0.00524	5	-0.01550	-0.04574	0.03289	-0.00784	0.00064
rar-m5		2.08	0.260	2	-0.0410	0.00521	5	-0.02235	-0.03104	0.02362	-0.00544	0.00042
rar-m3		2.08	0.262	2	-0.0377	0.00483	5	-0.05812	0.02990	-0.01191	0.00308	-0.00031
lha		2.07	0.253	2	-0.0179	0.00277	5	-0.00484	-0.01930	0.01629	-0.00464	0.00045
ar		2.07	0.253	2	-0.0179	0.00277	5	-0.00484	-0.01930	0.01629	-0.00464	0.00045
arj2		2.07	0.257	2	-0.0369	0.00545	5	-0.08369	0.06092	-0.02509	0.00530	-0.00043
gzip9		2.07	0.251	2	-0.0328	0.00451	5	-0.17111	0.24612	-0.13564	0.03249	-0.00282
arj		2.07	0.258	2	-0.0379	0.00553	5	-0.11502	0.10646	-0.05043	0.01137	-0.00095
arj1		2.07	0.258	2	-0.0379	0.00553	5	-0.11502	0.10646	-0.05043	0.01137	-0.00095
pkzip-ex		2.05	0.248	2	-0.0387	0.00532	5	-0.12672	0.17385	-0.09392	0.02222	-0.00191
pkzip-en		2.05	0.248	2	-0.0379	0.00528	5	-0.10223	0.13492	-0.07144	0.01661	-0.00141
arj3	y	2.04	0.247	2	-0.0229	0.00334	5	-0.12945	0.15774	-0.08167	0.01903	-0.00163
pkzip-ef	y	2.03	0.250	2	-0.0176	0.00257	5	-0.04261	0.07635	-0.04692	0.01187	-0.00105
RAR-M2	y	2.02	0.253	2	-0.0294	0.00374	5	-0.02304	-0.02755	0.02025	-0.00464	0.00037
rar-m1	y	2.00	0.251	2	-0.0204	0.00268	5	-0.02610	-0.03585	0.02533	-0.00575	0.00045
gzip1	y	1.99	0.231	0	0.0000	0.00000	5	0.01622	0.00472	-0.00556	0.00134	-0.00009
lzw15v		1.96	0.299	2	-0.0659	0.00998	5	0.03212	-0.25106	0.18178	-0.04888	0.00455
jar3	y	1.93	0.226	2	-0.0657	0.00879	5	-0.04399	0.07916	-0.07039	0.02054	-0.00194
jar4	y	1.93	0.226	2	-0.0658	0.00883	5	-0.04341	0.07650	-0.06826	0.01990	-0.00187
arith-n0		1.93	0.208	0	0.0000	0.00000	5	0.00419	-0.02682	0.01969	-0.00521	0.00048
zoo		1.92	0.294	2	-0.0309	0.00483	5	0.05366	-0.03406	0.00370	0.00166	-0.00030
jar1	y	1.92	0.222	2	-0.0642	0.00882	5	-0.02681	0.04817	-0.05718	0.01895	-0.00194
jar	y	1.92	0.223	2	-0.0647	0.00879	5	-0.03784	0.06398	-0.06601	0.02051	-0.00201
jar2	y	1.92	0.223	2	-0.0652	0.00888	5	-0.04045	0.07194	-0.07191	0.02219	-0.00217
arc		1.91	0.289	2	-0.0196	0.00308	5	-0.01469	0.11187	-0.08335	0.02535	-0.00271
pkzip-es		1.88	0.183	2	-0.0110	0.00183	5	0.21639	-0.37418	0.22967	-0.05846	0.00529
lzw12		1.87	0.313	2	-0.0618	0.00928	5	0.07349	-0.13354	0.08229	-0.02077	0.00185
arj4	y	1.86	0.232	1	-0.0047	0.00000	5	-0.00205	-0.02264	0.01446	-0.00319	0.00025
arith-c		1.85	0.177	1	0.0021	0.00000	5	-0.00521	-0.02581	0.02479	-0.00679	0.00062
ahuff-c		1.85	0.176	1	-0.0025	0.00000	3	-0.03264	0.01067	-0.00114	0.00000	0.00000
huff-c		1.85	0.178	1	0.0024	0.00000	5	-0.01291	-0.01606	0.02013	-0.00574	0.00053
lzss		1.54	0.331	2	-0.0205	0.00305	5	-0.02726	0.02733	-0.01291	0.00271	-0.00020

Table 8. Compression rates at 100 GHz, white noise only

Macro	Swap	$C_{\mathrm{r},1}$	S_1	D	A_1	A_2	D	B_1	B_2	B_3	B_4	B_5
arith-n1		2.67	0.269	2	-0.0239	0.00375	3	-0.08175	0.02282	-0.00223	0.00000	0.00000
boa	y	2.56	0.215	1	-0.0102	0.00000	5	0.08638	0.03130	-0.06070	0.02020	-0.00200
bziprf	y	2.56	0.248	2	-0.0189	0.00302	5	-0.11613	0.10497	-0.05095	0.01184	-0.00102
bziprb	y	2.56	0.248	2	-0.0189	0.00302	5	-0.11613	0.10497	-0.05095	0.01184	-0.00102
BZIP	y	2.56	0.248	2	-0.0189	0.00302	5	-0.11613	0.10497	-0.05095	0.01184	-0.00102
arith-n2	y	2.50	0.296	2	-0.0458	0.00555	3	-0.07707	0.00112	0.00090	0.00000	0.00000
uses320rr	y	2.44	0.219	0	0.0000	0.00000	2	0.00212	-0.00042	0.00000	0.00000	0.00000
uses032rr	y	2.44	0.218	0	0.0000	0.00000	2	0.00290	-0.00058	0.00000	0.00000	0.00000
uses960rr	y	2.44	0.218	0	0.0000	0.00000	2	0.00267	-0.00054	0.00000	0.00000	0.00000
uses	y	2.44	0.218	0	0.0000	0.00000	2	0.00261	-0.00052	0.00000	0.00000	0.00000
uses064rr	y	2.44	0.218	0	0.0000	0.00000	2	0.00261	-0.00052	0.00000	0.00000	0.00000
uses016rr	y	2.44	0.218	0	0.0000	0.00000	2	0.00254	-0.00051	0.00000	0.00000	0.00000
arhaHSC	y	2.44	0.224	2	-0.0281	0.00286	5	0.08921	0.08500	-0.12197	0.04051	-0.00416
arha	y	2.44	0.224	2	-0.0281	0.00286	5	0.08921	0.08500	-0.12197	0.04051	-0.00416
uses960	y	2.44	0.218	0	0.0000	0.00000	2	0.00230	-0.00047	0.00000	0.00000	0.00000
uses320	y	2.43	0.217	0	0.0000	0.00000	2	0.00243	-0.00049	0.00000	0.00000	0.00000
uses064	y	2.39	0.210	0	0.0000	0.00000	2	0.00227	-0.00044	0.00000	0.00000	0.00000
uses008rr	y	2.36	0.212	0	0.0000	0.00000	2	0.00297	-0.00057	0.00000	0.00000	0.00000
arith-n	y	2.36	0.288	2	-0.0529	0.00449	5	0.00176	0.00520	-0.02287	0.00734	-0.00068
arith-n3	y	2.36	0.288	2	-0.0529	0.00449	5	0.00176	0.00520	-0.02287	0.00734	-0.00068
uses032	y	2.33	0.202	0	0.0000	0.00000	2	0.00236	-0.00048	0.00000	0.00000	0.00000
uses016	y	2.23	0.188	0	0.0000	0.00000	1	0.00044	0.00000	0.00000	0.00000	0.00000
arith-n4	y	2.18	0.254	1	-0.0380	0.00000	5	0.18606	-0.05246	-0.00697	0.00413	-0.00043
arith-n5	y	2.07	0.221	2	-0.0243	-0.00354	5	0.34762	-0.13733	0.00720	0.00512	-0.00072
splint	y	2.07	0.196	2	-0.0133	0.00206	5	-0.00533	-0.02765	0.02599	-0.00789	0.00080
arith-n6	y	2.05	0.194	2	-0.0159	-0.00173	5	0.29121	-0.12851	-0.00241	0.01139	-0.00152
arhaASC	y	2.04	0.240	2	-0.0364	0.00562	5	-0.03234	-0.02616	0.02181	-0.00533	0.00045
arith-n7	y	2.04	0.177	2	-0.0098	-0.00050	5	0.25616	-0.18081	0.01542	0.01302	-0.00209
uses008	y	2.01	0.158	0	0.0000	0.00000	1	0.00072	0.00000	0.00000	0.00000	0.00000
lha		1.94	0.233	2	0.0018	-0.00548	5	0.19837	-0.54030	0.48419	-0.17127	0.02078
ar		1.93	0.231	2	-0.0273	0.00432	5	0.02245	-0.03921	0.02301	-0.00535	0.00044
rar-m4		1.92	0.237	2	-0.0521	0.00714	5	0.00661	-0.10494	0.07878	-0.02122	0.00196
rar-m5		1.92	0.237	2	-0.0522	0.00713	5	0.00965	-0.11367	0.08613	-0.02339	0.00217
rar-m3		1.92	0.240	2	-0.0489	0.00674	5	-0.03071	-0.04382	0.04149	-0.01171	0.00110
gzip9		1.92	0.231	2	-0.0451	0.00655	5	-0.12061	0.10263	-0.03799	0.00642	-0.00039
arj2		1.92	0.234	2	-0.0451	0.00695	5	-0.13463	0.10290	-0.03891	0.00716	-0.00050
arj		1.92	0.233	2	-0.0478	0.00727	5	-0.13045	0.11291	-0.04703	0.00936	-0.00070
arj1		1.92	0.233	2	-0.0478	0.00727	5	-0.13045	0.11291	-0.04703	0.00936	-0.00070
pkzip-en		1.90	0.228	2	-0.0481	0.00705	5	-0.15169	0.15484	-0.07118	0.01528	-0.00123
pkzip-ex		1.90	0.228	2	-0.0503	0.00725	5	-0.14571	0.15164	-0.06928	0.01452	-0.00113
arj3	y	1.88	0.211	2	-0.0311	0.00481	5	-0.00706	0.02981	-0.01898	0.00497	-0.00046
pkzip-ef	y	1.87	0.220	2	-0.0225	0.00351	5	0.03054	-0.05569	0.03241	-0.00778	0.00068
RAR-M2	y	1.87	0.219	2	-0.0395	0.00538	5	0.11148	-0.17429	0.08586	-0.01769	0.00132
lzw15v		1.85	0.285	2	-0.0602	0.00904	5	0.12796	-0.38889	0.26379	-0.07067	0.00664
rar-m1	y	1.84	0.219	2	-0.0307	0.00444	5	0.11383	-0.21122	0.11975	-0.02828	0.00241
gzip1	y	1.84	0.191	1	-0.0016	0.00000	3	0.03111	-0.01088	0.00119	0.00000	0.00000
jar4	y	1.82	0.218	2	-0.0744	0.00988	5	-0.09176	0.14991	-0.11208	0.03095	-0.00287
jar3	y	1.82	0.218	2	-0.0744	0.00985	5	-0.08278	0.13864	-0.10637	0.02971	-0.00277
jar1	y	1.82	0.214	2	-0.0723	0.01007	5	-0.06327	0.08005	-0.07213	0.02269	-0.00230
jar2	y	1.82	0.214	2	-0.0727	0.00970	5	-0.07442	0.10520	-0.08803	0.02572	-0.00246
zoo		1.82	0.279	2	-0.0245	0.00391	5	0.11582	-0.14072	0.07516	-0.01776	0.00154
jar	y	1.82	0.214	2	-0.0731	0.00978	5	-0.07412	0.10486	-0.08801	0.02574	-0.00247
pkzip-es		1.81	0.167	1	-0.0030	0.00000	5	-0.01860	0.02217	-0.00952	0.00190	-0.00014
ahuff-c		1.81	0.170	1	-0.0021	0.00000	3	-0.02288	0.00543	-0.00049	0.00000	0.00000
lzw12		1.81	0.282	2	-0.0556	0.00883	5	0.05272	-0.04581	0.02072	-0.00425	0.00033
arith-c		1.81	0.175	0	0.0000	0.00000	5	-0.00536	-0.03274	0.02661	-0.00703	0.00064
huff-c		1.80	0.174	0	0.0000	0.00000	5	-0.00677	-0.02247	0.01499	-0.00279	0.00016
arc		1.79	0.285	2	-0.0126	0.00207	5	0.00603	0.05162	-0.04296	0.01245	-0.00121
arith-n0		1.76	0.190	0	0.0000	0.00000	5	-0.01967	0.02962	-0.01854	0.00510	-0.00049
arj4	y	1.72	0.195	1	-0.0049	0.00000	5	0.06028	-0.08652	0.04954	-0.01210	0.00107
lzss		1.39	0.321	2	-0.0220	0.00355	5	0.07109	-0.07770	0.04328	-0.01090	0.00100

Table 9. Compression rates at 100 GHz, full signal

Macro	Swap	$C_{r,1}$	S_1	D	A_1	A_2	D	B_1	B_2	B_3	B_4	B_5
arith-n1		2.61	0.266	2	-0.0247	0.00367	3	-0.08195	0.02105	-0.00191	0.00000	0.00000
boa	y	2.52	0.215	1	-0.0104	0.00000	5	0.01603	0.14591	-0.12275	0.03379	-0.00305
BZIP	y	2.50	0.244	2	-0.0226	0.00350	5	-0.07152	0.02975	-0.00662	0.00104	-0.00008
bziprf	y	2.50	0.244	2	-0.0226	0.00350	5	-0.07152	0.02975	-0.00662	0.00104	-0.00008
bziprb	y	2.50	0.244	2	-0.0226	0.00350	5	-0.07152	0.02975	-0.00662	0.00104	-0.00008
uses016rr	y	2.44	0.218	0	0.0000	0.00000	1	0.00024	0.00000	0.00000	0.00000	0.00000
uses320rr	y	2.44	0.218	0	0.0000	0.00000	3	0.00368	-0.00209	0.00031	0.00000	0.00000
uses064rr	y	2.44	0.219	0	0.0000	0.00000	3	0.00231	-0.00159	0.00026	0.00000	0.00000
uses	y	2.44	0.219	0	0.0000	0.00000	3	0.00231	-0.00159	0.00026	0.00000	0.00000
uses960rr	y	2.44	0.218	0	0.0000	0.00000	3	0.00410	-0.00220	0.00033	0.00000	0.00000
uses032rr	y	2.44	0.218	0	0.0000	0.00000	3	0.00265	-0.00155	0.00024	0.00000	0.00000
arith-n2	y	2.43	0.294	2	-0.0481	0.00558	3	-0.08108	0.00117	0.00091	0.00000	0.00000
uses960	y	2.43	0.218	0	0.0000	0.00000	3	0.00385	-0.00218	0.00033	0.00000	0.00000
uses320	y	2.42	0.217	0	0.0000	0.00000	3	0.00338	-0.00197	0.00030	0.00000	0.00000
arhaHSC	y	2.41	0.217	1	-0.0191	0.00000	5	0.06705	0.13417	-0.14820	0.04462	-0.00431
arha	y	2.41	0.217	1	-0.0191	0.00000	5	0.06705	0.13417	-0.14820	0.04462	-0.00431
uses064	y	2.38	0.210	0	0.0000	0.00000	3	0.00318	-0.00186	0.00029	0.00000	0.00000
uses008rr	y	2.36	0.212	0	0.0000	0.00000	2	0.00052	-0.00016	0.00000	0.00000	0.00000
uses032	y	2.33	0.202	0	0.0000	0.00000	1	0.00010	0.00000	0.00000	0.00000	0.00000
arith-n3	y	2.30	0.284	2	-0.0533	0.00421	5	-0.01253	0.02478	-0.03086	0.00861	-0.00075
arith-n	y	2.30	0.284	2	-0.0533	0.00421	5	-0.01253	0.02478	-0.03086	0.00861	-0.00075
uses016	y	2.23	0.188	0	0.0000	0.00000	1	-0.00002	0.00000	0.00000	0.00000	0.00000
arith-n4	y	2.14	0.244	2	-0.0292	-0.00196	5	0.14329	0.03495	-0.05418	0.01476	-0.00130
splint	y	2.07	0.197	1	-0.0047	0.00000	5	-0.01741	0.01224	-0.00217	-0.00032	0.00010
arith-n5	y	2.03	0.208	2	-0.0183	-0.00466	5	0.29045	0.01512	-0.08376	0.02666	-0.00252
arith-n6	y	2.02	0.186	2	-0.0115	-0.00265	5	0.22665	0.03674	-0.10799	0.03718	-0.00370
arith-n7	y	2.01	0.173	2	-0.0066	-0.00130	5	0.17214	0.04752	-0.14136	0.05310	-0.00559
uses008	y	2.00	0.158	0	0.0000	0.00000	1	-0.00021	0.00000	0.00000	0.00000	0.00000
arhaASC		2.00	0.230	2	-0.0311	0.00469	5	-0.02170	-0.05814	0.04424	-0.01140	0.00102
lha		1.92	0.232	2	0.0098	-0.00698	5	0.20283	-0.53081	0.47763	-0.16992	0.02069
ar		1.91	0.230	2	-0.0193	0.00283	5	0.02404	-0.02337	0.01172	-0.00256	0.00020
rar-m4		1.90	0.246	2	-0.0463	0.00614	5	-0.07985	0.00548	0.01515	-0.00532	0.00053
rar-m5		1.90	0.246	2	-0.0464	0.00613	5	-0.08017	0.00354	0.01795	-0.00629	0.00064
rar-m3		1.90	0.248	2	-0.0435	0.00577	5	-0.08580	0.01333	0.01332	-0.00547	0.00059
gzip9		1.90	0.231	2	-0.0382	0.00528	5	-0.06539	0.00848	0.01219	-0.00488	0.00053
arj2		1.90	0.238	2	-0.0395	0.00581	5	-0.14722	0.11316	-0.04561	0.00913	-0.00070
arj1		1.90	0.236	2	-0.0420	0.00618	5	-0.12048	0.08350	-0.03106	0.00586	-0.00043
arj		1.90	0.236	2	-0.0420	0.00618	5	-0.12048	0.08350	-0.03106	0.00586	-0.00043
pkzip-en		1.88	0.229	2	-0.0418	0.00582	5	-0.11754	0.09781	-0.03959	0.00759	-0.00054
pkzip-ex		1.88	0.229	2	-0.0434	0.00602	5	-0.10145	0.07173	-0.02560	0.00452	-0.00030
arj3	y	1.87	0.224	2	-0.0249	0.00372	5	-0.04766	0.04250	-0.01611	0.00301	-0.00022
pkzip-ef	y	1.86	0.225	2	-0.0191	0.00280	5	-0.05886	0.07895	-0.04524	0.01169	-0.00109
RAR-M2	y	1.85	0.229	2	-0.0349	0.00451	5	0.02404	-0.11088	0.06660	-0.01558	0.00130
gzip1	y	1.84	0.196	0	0.0000	0.00000	3	0.01133	-0.00301	0.00026	0.00000	0.00000
rar-m1	y	1.83	0.225	2	-0.0250	0.00335	5	-0.03851	0.00944	-0.00595	0.00278	-0.00037
lzw15v		1.80	0.287	2	-0.0600	0.00879	5	0.07732	-0.33327	0.23644	-0.06447	0.00611
jar3	y	1.80	0.213	2	-0.0706	0.00908	5	-0.02884	0.07578	-0.07773	0.02412	-0.00238
jar4	y	1.80	0.213	2	-0.0710	0.00917	5	-0.02464	0.06882	-0.07334	0.02294	-0.00227
pkzip-es		1.79	0.170	1	-0.0031	0.00000	3	-0.00794	0.00285	-0.00030	0.00000	0.00000
jar1	y	1.79	0.212	2	-0.0693	0.00942	5	0.01875	-0.04234	-0.00304	0.00590	-0.00082
jar2	y	1.79	0.213	2	-0.0700	0.00910	5	-0.03402	0.06453	-0.07371	0.02387	-0.00241
jar	y	1.79	0.211	2	-0.0700	0.00910	5	-0.01391	0.04316	-0.06396	0.02185	-0.00226
arc	y	1.79	0.306	0	0.0000	0.00000	5	0.18310	-0.29880	0.15378	-0.03322	0.00261
ahuff-c		1.78	0.167	1	-0.0029	0.00000	3	-0.03978	0.01224	-0.00127	0.00000	0.00000
arith-c		1.78	0.173	0	0.0000	0.00000	5	-0.04911	0.03957	-0.01069	0.00088	0.00004
huff-c		1.77	0.171	0	0.0000	0.00000	5	-0.08077	0.10270	-0.05115	0.01166	-0.00099
zoo		1.77	0.282	2	-0.0219	0.00315	5	0.06067	-0.05962	0.02875	-0.00637	0.00053
arith-n0		1.75	0.198	0	0.0000	0.00000	5	0.00087	-0.01669	0.01603	-0.00493	0.00050
lzw12		1.75	0.293	2	-0.0519	0.00772	5	-0.01836	0.01817	-0.00615	0.00097	-0.00007
arj4	y	1.71	0.207	1	-0.0053	0.00000	5	-0.03454	0.04140	-0.02937	0.00871	-0.00088
lzss		1.37	0.329	2	-0.0187	0.00267	5	0.02328	-0.00459	-0.00898	0.00430	-0.00052

It should be noted that the coefficients reported here are obtained compressing one or more full scan circles at a time, so their use to extrapolate C_r when each scan circle is divided in small chunks which are separately compressed has to be performed carefully, especially for VOT ≈ 0.5 V/K where some extrapolated C_r grows instead of to decrease for a decreasing N_c as in most of the cases. However we did not investigate further the problem because the time required to perform all the tests over all the compressors increases decreasing N_c, and because up to now a final decision about the packet length has not been made yet. Moreover, short data chunks introduce other constrains which are not accounted for by Eq. (9) but which are discussed in Sect. 8.

Apart from the choice of the best compressor, Tables 6 to 9 allow interesting comparisons.

The performances of the arithmetic compression `arith` are very sensitive to changes in the coding order $n = 0, \ldots, 7$. The computational weight grows with n, while C_r is minimal at $n = 0$, maximal for $n = 1$ and decreases increasing n further.

Both non-Adaptive Huffman (`huff-c`) and Adaptive Huffman (`ahuff-c`) are in the list of the worst compressors, considering both the pure white noise signal and the full signal.

We implemented the space-qualified `uses` compressor with a wide selection of combinations of its control parameters: the number of coding bits, the number of samples per block, the possibility to search for correlations between neighborhood samples. We report the tests for 16 bits coding only, changing the other parameters. `Uses` is very sensitive to byte unswapping, when not performed `uses` does not compress at all. On the other hand, opposite to `arith` the sensitivity of the final C_r to the various control parameters is small or negligible. In most cases $C_{\mathrm{r},1}$ differs of less than 0.01 for changing the combination of control parameters, such changes are not displayed by the two digits approximation in the tables, but they are accounted for by the sorting procedure which fixes the table ordering. At 30 GHz most of the tested compressors cluster around $C_{\mathrm{r},1} = 2.67$ and at this level `arith-n3` is as good as `uses`. At 100 GHz the best `uses` macros clusters around $C_{\mathrm{r},1} = 2.43 - 2.44$, equivalent to `arith-n2` performance. In our tests `uses` performs worst at 8 samples per block without correlation search, but apart from it, in our case the correlation search does not improve significantly the compression performances. Some commercial programs such as `boa`, `bzip` compress better than `uses`.

8. Further constrains: Packet independence and packet length

As an example of global constrains to the on-board compression we discuss the problems related to Packets Independence and Packets Length.

Data from the LFI must be put into equal size packets before being sent to Earth. Packets independence is considered to be a requirement, then each packet must be self-consistent, its loss or its erroneous transmission must not interfere with the data retrieval from subsequent packets. More over each packet must carry in "clear" format (i.e. uncompressed) all the information needed to decode its content. That is: each packet must contain its own decoding table or decoding information. A typical packet length is about some hundred of bytes, but a smaller length may be planned if required; at the same time a typical decoding table holds something less than a hundred bytes leaving limited room for data.

In addition, for a fixed length L_u of a random input stream (expressed in bits) the output L_c will not be a constant but will change in time with respect to the averaged length $L_\mathrm{u}/C_\mathrm{r}$. Of course, it is not possible to predict in advance what will be the final length of a given bit stream. So either L_u is held fixed, loosing in compression efficiency, or L_u is adapted with some interactive method, maximizing the compression efficiency but at the cost of a significant slowing of the compression process.

In conclusion, the packets independence plus limited packet length prevents from sending the decoding table, leaving only two possibilities open: i) send the relevant bytes only (Maris 1999a), ii) to use a predefined coding table (Maris 1999b), both methods are described in the next section.

9. Proposed coding and compression scheme

The basic principle of the first method named *Least Significant Bits Packing* (LSBP) is to send only those bits of the 16 bits output from the ADC which are affected by the signal and the noise. This is effective for the nominal mission since with the planned quantization step of 0.3 mK/adu, at one sigma the noise will fill about 21 levels, this will require at least 5 bits over 16 and it is reasonable to expect a final data flow equivalent to $C_{\mathrm{r},1} < 3$. It is not possible to improve much the compression rate by compressing the resulting 5 bits data stream, since its entropy would be $H < 5.4$ bits and $C_\mathrm{r} \lesssim 1.08$.

In order to ensure the compression to be lossless all the samples exceeding the $[-\sigma, +\sigma]$ (5 bits) range have to be sent separately coding at the same time: their position (address) in the stream vector and their value. So, for $N_\mathrm{bits} < 16$ bits corresponding to a threshold $x_\mathrm{th} = 2^{N_\mathrm{bits}}$, each group of samples stored into a packet is partitioned into two classes accordingly with their value x:

Regular Samples (RS) ⟨def⟩ all those samples for which: $|x| \le x_\mathrm{th}$,

Spike Samples (SS) ⟨def⟩ all those samples for which: $|x| \ge x_\mathrm{th}$.

The coding process then consists of two main steps: i) to split the data stream in Regular and Spike Samples preserving the original ordering in the stream of Regular Samples, ii) to store (send) the first $N_{\rm bits}$ bits of the regular samples and, in a separated area, the 16 bits values and the location in the original data stream of each Spike Sample, i.e. Spike Samples will require more space to be stored than regular ones. The decoding process will be the reverse of this packing process.

In this scheme each packet will be divided into two main areas: the Regular Samples Area (RSA) which hold the stream of Regular Samples, the Spike Sample Area (SSA) which hold the stream of Spike Samples, plus a number of fields which will contain packing parameters such as: the number of samples, the number of regular samples, the offset, etc. Since the number of samples in each area will change randomly it will be not possible to completely fill a packet. The filling process will leave an empty area in the packet in average smaller than $N_{\rm bits}$.

In Maris (1999a) a first evaluation for the 30 GHz channel is given assuming that the signal is composed only of white noise plus the CMB dipole. As noticed in Sect. 7.2 the cosmological dipole affects the compression efficiency reducing it of a small amount. To deal with it a possible solution would be to subdivide each data stream in packets, subtract to each measure of a given packet the integer average of samples (computed as a 16 bits integer number) and then compress the residuals. Each integer average will be sent to Earth together with the related packet where the operation will be reversed. Since all the numbers are coded as 16 bits integers all the operations are fully reversible and no round off error occurs. However it cannot be excluded that the computational cost of such operation will compensate the gain in $C_{\rm r}$.

Two schemes are proposed to perform the cosmological dipole self-adaptement. In *Scheme A* the average of samples in the packet are subtracted before coding and then sent separately. In *Scheme B* $x_{\rm th}$ is varied proportionally to the dipole contribution. Both of them assumes that the dipole contribution is about a constant over a packet length. From this assumption: $Lpacket \lesssim 200$ samples i.e. $Lpacket < 512$ bytes, since for $Lpacket > 512$ bytes the cosmic dipole contribution can not be considered as a time constant. For larger packets a better modeling (i.e. more parameters) will be required in order not to degrade the compression efficiency.

A critical point is to fix the best $x_{\rm th}$, i.e. $N_{\rm bits}$, for a given signal statistics, coding scheme and packet length $L_{\rm p}$. Even here $C_{\rm r}$ grows with the packet length but it does not change monotously with $x_{\rm th}$. An increase in $x_{\rm th}$ ($N_{\rm bits}$) decreases the number of spike samples, but increases the size of each regular sample. While the opposite occurs when $x_{\rm th}$ is decreased, and when $N_{\rm bits} < 4$ bits $C_{\rm r} < 1$. For both the schemes the optimality is reached for $N_{\rm bits} = 6$ bits, but *Scheme A* is better than *B*, with: $C_{\rm r}$(*Scheme A*, $Lpacket = 512$ bytes) $= 2.61$, $C_{\rm r}$(*Scheme B*, $Lpacket = 512$ bytes) $= 2.29$.

Compared with `arith-n1`, this compression rate is worse by about a $14 - 30\%$. This is due to two reasons: i) coding by a threshold cut is less effective than to apply an optimized compressor; ii) the results reported in Tables 6-9 refer to the compression of a full circle of data instead of a small packet, resulting in a higher efficiency. However, the efficiency of this coding method is similar to the efficiency of the bulk of the other true loss-less compressors tested up to now, and when the need to send a decoding table is considered, is even higher. A compression scheme based on the same principle, but with a different organization of fields, has been proposed also by Guzzi & Silvestri (1999) which report a similar compression efficiency.

The second possible solution to the packeting problem is to use one or more standardized coding tables for the compression scheme of choice (Maris 1999b). In this case the coding table would be loaded into the on-board computer before launch or time by time in flight and the table should be known in advance at Earth. Major advantages would be: 1. the coding table has not to be sent to Earth; 2. the compression operator will be reduced to a mapping operator which may be implement as a tabular search, driven by the input 8 or 16 bits word to be compressed; 3. any compression scheme (Huffman, arithmetic, etc.) may be implemented replacing the coding table without changes to the compression program; 4. the compression procedure may be easily written in C or the native assembler language for the on-board computer or, alternatively, a simple, dedicated hardware may be implemented and interfaced to the on-board computer. The disadvantages of this scheme are: 1. each table must reside permanently in the central computer memory unless a dedicated hardware is interfaced to it; 2. it is difficult to use adaptive schemes in order to tune the compressor to the input signal, as a consequence the $C_{\rm r}$ may be somewhat smaller than in the case of a true self-adapting compressor code.

The first problem may be circumvented limiting the length of the words to be compressed. In our case the data streams may be divided in chunks of 8 bits and the typical table size would be $\lesssim 1$ Kbyte. Precomputed coding tables may be accurately optimized by Monte-Carlo simulations on ground or using signals from ground tests of true hardware.

The second problem may be overcome by using a preconditioning stage, reducing the statistics of the input signal to the statistics for which the pre-calculated table is optimized. In addition more tables may reside in the computer memory and selected looking to the signal statistics. With a simple reversible statistical preconditioner, about ten tables per frequency channel would be stored in the computer memory, so that the total memory occupation would be less than about 40 Kbytes. It cannot be excluded that the two methods just outlined can be merged.

10. Estimation of the overall compression rate

The overall compression rate (efficiency) is the average of $C_{\rm r}$ ($\eta_{\rm c}$) over the full set of detectors. Appendix A illustrates the mathematical aspects of such average. From Eq. (A4):

$$\overline{C_{\rm r}}(N_{\rm c}) = \left[\sum_{\nu} \frac{f_\nu}{C_{{\rm r},\nu}(N_{\rm c})}\right]^{-1}. \qquad (15)$$

We will limit ourselves to the most probable case $N_{\rm c} = 1$ and to the most effective compressor `arith-n1`. The compression parameters $C_{\rm r,1}$ and $\mathcal{S}_1$ at 30 GHz and 100 GHz are derived from our simulations, while $C_{\rm r,1}$ and $\mathcal{S}_1$ at 44 GHz and 70 GHz are obtained by linear interpolation of the simulated values as a function of $\ln \sigma_\nu$. After that we obtain:

$$\overline{C_{\rm r}} \approx \frac{2.66}{1 + 0.271 \times \ln \mathrm{VOT}}. \qquad (16)$$

As expected the overall compression rate is dominated by the 100 GHz channel. Taking in account the conservative VOT distribution considered in Eq. (A8) the overall compression rate becomes: $\overline{C_{\rm r}} \approx 2.63$ which represents a $\approx 2\%$ correction only. It is likely that this correction will be even smaller, since the amplifiers gain will be adjusted in order to cover a smaller VOT interval. So this 2% correction represents our greatest uncertainty in our estimation of the expected compression rate in the present conditions, and we may conservatively conclude that:

$$\overline{C_{\rm r,arith-n1}} \approx 2.65 \pm 0.02 \cdot \qquad (17)$$

Recently a new evaluation of the expected instrumental sensitivity leads to some change in the expected white noise rms. These changes affect in particular the 30 GHz channel, but does not change significantly the 100 GHz channel so that the overall compression rate will be practically unaffected.

11. Conclusions

The expected data rate from the PLANCK Low Frequency Instrument is $\approx$ 260 kbits/sec. The bandwidth for the scientific data download currently allocated is just $\approx$ 60 kbit/sec. Assuming an equal subdivision of the bandwidth between the two instruments on-board PLANCK, an overall compression rate of a factor 8.7 is required to download all the data.

In this work we perform a full analysis on realistically simulated data streams for the 30 GHz and 100 GHz channels in order to fix the maximum compression rate achievable by loss-less compression methods, without considering explicitly other constrains such as: the power of the on-board Data Processing Unit, or the requirements about packet length limits and independence, but taking in account all the instrumental features relevant to data acquisition, i.e.: the quantization process, the temperature/voltage conversion, number of quantization bits and signal composition.

As a complement to the experimental analysis we perform in parallel a theoretical analysis of the maximum compression rate. Such analysis is based on the statistical properties of the simulated signal and is able to explain quantitatively most of the experimental results.

Our conclusions about the statistical analysis of the quantized signal are: I) the nominally quantized signal has an entropy $h \approx 5.5$ bits at 30 GHz and $h \approx 5.9$ bits at 100 GHz, which allows a theoretical upper limit for the compression rate ≈ 2.9 at 30 GHz and ≈ 2.7 at 100 GHz. II) Quantization may introduce some distortion in the signal statistics but the subject requires a deepest analysis.

Our conclusions about the compression rate are summarized as follows: I) the compression rate $C_{\rm r}$ is affected by the quantization step, since greater is the quantization step higher is $C_{\rm r}$ (but worse is the measure accuracy). II) $C_{\rm r}$ is affected also by the stream length $L_{\rm u}$, i.e. more circles are compressed better then few circles. III) the dependencies on the quantization step and $L_{\rm u}$ for each compressor may be summarized by the empirical formula (12). A reduced compression rate $C_{\rm r,1}$ is correspondingly defined. IV) the $C_{\rm r}$ is affected by the signal composition, in particular, by the white noise rms and by the dipole contribution, the former being the dominant parameter and the latter influencing $C_{\rm r}$ for less than $\approx 6\%$. The inclusion of the dipole contribution reduces the overall compression rate. The other components ($1/f$ noise, CMB fluctuations, the galaxy, extragalactic sources) have little or no effect on $C_{\rm r}$. In conclusion, for the sake of compression rate estimation, the signal may be safely represented by a sinusoidal signal plus white noise. V) since the noise rms increases with the frequency, the compression rate $C_{\rm r}$ decreases with the frequency, for the LFI $\Delta C_{\rm r}/C_{\rm r} \lesssim 10\%$. VI) the expected random rms in the overall compression rate is less than 1%. VII) we tested a large number of off-the-shelf compressors, with many combinations of control parameters so to cover every conceivable compression method. The best performing compressor is the arithmetic compression scheme of order 1: `arith-n1`, the final $C_{\rm r,1}$ being 2.83 at 30 GHz and 2.61 at 100 GHz. This is significantly less than the bare theoretical compression rate (9) but when the quantization process is taken properly into account in the theoretical analysis, this discrepancy is largely reduced. VIII) taking into account the data flow distribution among different compressors the overall compression rate for `arith-n1` is:

$\overline{C_{\rm r,arith-n1}} \approx 2.65 \pm 0.02.$

This result is due to the nature of the signal which is noise dominated and clearly excludes the possibility to reach the required data flow reduction through loss-less compression only.

Possible solutions deal with the application of lossy compression methods such as: on-board averaging, data rebinning, or averaging of signals from duplicated detectors, in order to reach an overall lossy compression of about a factor 3.4, which coupled with the overall lossless compression rate of about 2.65 should allow to reach the required final compression rate ≈ 8.7. However each of these solutions will introduce heavy constrains and important reduction of performances in the final mission design, so that careful and deep studies will be required in order to choose the best one.

Another solution to the bandwidth problem would be to apply a coarser quantization step. This has however the drawback of reducing the signal resolution in terms of $\Delta T/T$.

Lastly the choice of a given compressor cannot be based only on its efficiency obtained from simulated data, but also on the on-board available CPU and on the official ESA space qualification: tests with this hardware platform and other compressors will be made during the project development. Moreover, we are confident that the experience which will be gained inside the CMB community developing ground, balloon and space based experiments, as the development of full prototypes of the on-board electronics, will provide us with a solid base to test and improve compression algorithms. In addition the final compression scheme will have to cope with requirements about packet length and packet independence. We discuss briefly this problems recalling two proposals (Maris 1999a, 1999b) which suggest solutions to cope with these constrains.

Acknowledgements. We warmly acknowledge a number of people which actively support this work with fruitful discussions, in particular F. Argüeso, M. Bersanelli, L. Danese, G. De Zotti, E. Gaztñaga, J. Herrera, N. Mandolesi, P. Platania, A. Romeo, M. Seiffert and L. Toffolatti and K. Gorski and all people involved in the construction of the Healpix pixelisation tools, largely employed in this work, G. Lombardi from *Siemens - Bocholt* and G. Maris from *ETNOTEAM - Milano*, for fruitful discussions about compression principles and their practical application, P. Guzzi and R. Silvestri from *LABEN - Milano* for explanations and suggestions about the PLANCK-LFI data acquisition electronics. At last we acknowledge the referee Miguel A. Albrecht, for the useful suggestions and corrections, which improve significantly the text readability and accuracy.

Appendix A: Formulation of the final data flow

In this Appendix we will discuss how to account for the distribution of the acquisition parameters between the different detectors in the computation of the overall compression rate. Since the formalism is simpler we will develop expressions for $\eta_{\rm c} = 1/C_{\rm r}$ instead of $C_{\rm r}$.

We have pointed out in Sect. 5.2 that the compression efficiency is a random variable, whose distribution is a function of all those parameters which are relevant to fix the statistical distribution of the input signal. In our case: ν, VOT, AFO, $N_{\rm circ}$ are the relevant parameters, so that the conditioned probability to have a compression efficiency in the range $\eta_{\rm c}$, $\eta_{\rm c} + {\rm d}\eta_{\rm c}$ is:

$$\mathcal{P}_{\nu,N_{\rm Circ}}\left(\eta_{\rm c}|{\rm AFO},{\rm VOT}\right){\rm d}\eta_{\rm c}. \tag{A1}$$

This probability may be obtained by our Monte-Carlo simulations for different combinations of AFO, VOT, $N_{\rm circ}$ and ν. Then the averaged compression efficiency is:

$$\overline{\eta}_{{\rm c},\nu,N_{\rm Circ}}({\rm AFO},{\rm VOT}) = \int_0^{+\infty} {\rm d}\eta_{\rm c}\,\eta_{\rm c}\,\mathcal{P}_{\nu,N_{\rm Circ}} \times \left(\eta_{\rm c}|{\rm AFO},{\rm VOT}\right). \tag{A2}$$

Of course we assumed that for any ν, VOT, AFO, $N_{\rm circ}$ the probability distribution is integrable and normalized to 1, while the integration limits 0, $+\infty$ are to be intended as formal. There are several detectors for any frequency channel, each one having its own AFO and VOT, so distributions of AFO and VOT values may be guessed among the different detectors. Assuming they are integrable and normalized to 1 as well it is possible to compute the most probable $\overline{\eta}_{{\rm c},\nu,N_{\rm Circ}}$ as[5]

$$\overline{\eta}_{{\rm c},\nu,N_{\rm Circ}} = \int_{\rm AFO_{min}}^{\rm AFO_{max}} {\rm dAFO}\,\mathcal{P}_\nu({\rm AFO}) \times \int_{\rm VOT_{min}}^{\rm VOT_{max}} {\rm dVOT}\,\mathcal{P}_\nu({\rm VOT})\,\overline{\eta}_{{\rm c},\nu,N_{\rm Circ}}({\rm AFO},{\rm VOT}). \tag{A3}$$

With this definition the final overall compression efficiency is:

$$\overline{\eta}_{{\rm c},N_{\rm Circ}} = \sum_{\nu=30,44,70,100\ {\rm GHz}} f_\nu \overline{\eta}_{{\rm c},\nu,N_{\rm Circ}} \tag{A4}$$

where f_ν is the partition function for the data flow through the different detectors, if $n_{{\rm dtc},\nu}$ is the number of detectors for the frequency channel ν (see Table 1), $n_{\rm dtc} = \sum_{\nu=30,44,70,100\ {\rm GHz}} n_{{\rm dtc},\nu} = 112$, is the total number of detectors and if the number of samples for frequency is a constant, then:

$$f_\nu = \frac{n_{{\rm dtc},\nu}}{112}, \tag{A5}$$

so that for $\nu = 30$, 44, 70 and 100 GHz respectively: $f_\nu = 0.0714$, 0.1071, 0.2143 and 0.6071, finally the expect data rate for each set of 60 circles is:

$$\overline{R}_{N_{\rm Circ}} = 16\,{\rm bits} \times 60\,{\rm circles} \times 8640\,{\rm samples} \times 112\,{\rm detectors} \times \overline{\eta}_{\rm c}^{\,N_{\rm Circ}}. \tag{A6}$$

Presently there are no data to know in advance the distribution of VOT and AFO values between the different detectors. For this reason in this work we assumed simply flat distributions, identical for each frequency for such

[5] Here

$$\int_{\rm AFO_{min}}^{\rm AFO_{max}} {\rm dAFO}\,\mathcal{P}_\nu({\rm AFO}) = 1, \quad \int_{\rm VOT_{min}}^{\rm VOT_{max}} {\rm dVOT}\,\mathcal{P}_\nu({\rm VOT}) = 1.$$

parameters. More over, the AFO contribution is negligible, so that the variance introduced by this parameter is neglected. From (9) we assumed that the compression efficiency is approximately a linear function of ln VOT or:

$$\overline{\eta}_{\mathrm{c},\nu,N_{\mathrm{Circ}}}(\mathrm{VOT}) \approx \overline{\eta}_{\mathrm{c},\nu,N_{\mathrm{Circ}},1} + \dot{\overline{\eta}}_{\mathrm{c},\nu,N_{\mathrm{Circ}}} \ln \mathrm{VOT} \qquad (\mathrm{A7})$$

where $\dot{\overline{\eta}}_{\mathrm{c},\nu,N_{\mathrm{Circ}}}$ is the first derivative of $\overline{\eta}_{\mathrm{c},\nu,N_{\mathrm{Circ}}}(\mathrm{VOT})$ with respect to ln VOT computed for VOT = 1 V/K, $\overline{\eta}_{\mathrm{c},\nu,N_{\mathrm{Circ}},1} \equiv \overline{\eta}_{\mathrm{c},\nu,N_{\mathrm{Circ}}}(\mathrm{VOT} = 1\ \mathrm{V/K})$. As an example, at 30 GHz for `arith-n1` the full signal compression rate is $\overline{\eta}_{\mathrm{c},\nu,N_{\mathrm{Circ}}}(\mathrm{VOT}) \approx 0.3534 + 0.287 \times \ln \mathrm{VOT(K/V)}$ with one interpolation error less than 0.2%. With these approximations Eq. (A3) becomes

$$\overline{\eta}_{\mathrm{c},\nu,N_{\mathrm{Circ}}} \approx \overline{\eta}_{\mathrm{c},\nu,N_{\mathrm{Circ}},1} + \dot{\overline{\eta}}_{\mathrm{c},\nu,N_{\mathrm{Circ}}} \times \int_{0.5\ \mathrm{V/K}}^{1.5\ \mathrm{V/K}} \mathrm{dVOT} \frac{\ln \mathrm{VOT}}{1.0\ \mathrm{V/K}} \qquad (\mathrm{A8})$$

and after integration we obtain the final formula

$$\overline{\eta}_{\mathrm{c},\nu,N_{\mathrm{Circ}}} \approx \overline{\eta}_{\mathrm{c},\nu,N_{\mathrm{Circ}},1} - 0.045229 \cdot \dot{\overline{\eta}}_{\mathrm{c},\nu,N_{\mathrm{Circ}}} \qquad (\mathrm{A9})$$

for the case in the previous example: $\overline{\eta}_{\mathrm{c},\nu,N_{\mathrm{Circ}}=2} \approx 0.34$ which is equivalent to $C_{\mathrm{r}} \approx 2.94$.

To understand the influence of the error in the VOT determination over the distribution on the final predictions the computation is made for a truncated (i.e. zero outside the VOT range of interest) normal distribution of VOT. The rms for the VOT distribution is chosen in the VOT range [0.5, 1.5] V/K we obtain respectively $\overline{\eta}_{\mathrm{c},\nu,N_{\mathrm{Circ}}} \approx 0.35,\ 0.34,\ 0.34$; which corresponds to $C_{\mathrm{r}} \approx 2.86,\ 2.91,\ 2.92$ respectively. Similar results are obtained with a quadratic VOT distribution. In conclusion these predictions are robust against the shape of the VOT distribution, at least for distributions which are symmetric around the nominal VOT = 1 V/K value.

References

Bennett C.L., et al., 1996, Amer. Astro. Soc. Meet. 88.05

Bersanelli M., Efstatihou G., Griffin M., et al., 1996, COBRAS/SAMBA, Rep. Phase A Study, ESA D/SCI(96)3

Bersanelli M., Burigana C., Butler C., et al., 2000, LFI Int. Rep. PLANCK-LFI: On-Board Data Processing, Compression and Telemetry Rate April 21, 2000, ISSUE 3.0

Balbi A., Ade P., Bock J., et al., 2000, astro-ph/0005124 (submitted to ApJ Lett.)

Bromley B.C., Tegmark M., 1999, ApJ 524, L79

Burigana C., Danese L, de Zotti G., et al., 1997a, MNRAS 287, L17

Burigana C., et al., 1997b, Int. Rep. TeSRE/CNR 198/1997

Burigana C., Maino D., Mandolesi N., et al., 1998, A&AS 130, 551

Burigana C., Maino D., Mandolesi N., et al., 2000, Ap. Lett. Comm., 37, Issue 3-6, (in press) astro-ph/9903137

Danese L., Franceschini A., Toffolatti L., et al., 1987, ApJ 318, L15

De Bernardis P., Ade P.A.R., Bock J.J., et al., 2000, Nat 404, 955

De Bernardis P., Masi S., 1998 Proc. of the XXXIII$^{\mathrm{rd}}$ Rescontres de Moriond, Les Arcs, France, January 17-24, 1998, p. 209

De Zotti G., Toffolatti L., 1998, ApJ Lett. (in press)

Ferreira P.G., Górski K.M., Magueijo J., 1999, 3K cosmology, Proc. of the EC-TMR Conf., Rome, Italy, October, 1998, Maiani L., Melchiorri F., Vittorio N. (eds.). Pub. Woodbury, N.Y.: American Institute of Physics, 476, 293

Fixsen D.J., Cheng E.S., Gales J.M., et al., 1996, ApJ 473, 576

Franceschini A., Mazzei P., De Zotti G., Danese L., 1994, ApJ 427, 140

Gaztñaga E., et al., 2000, Ap. Lett. Comm. 37, Issue 3-6 (in press) (also LFI-IEEC-TN-002.0)

Guzzi P., Silvestri R., 1999, A coding scheme for the LFI data, LABEN Technical Note, November 5, 1999

Hanany S., Ade P., Balbi A., et al., 2000, astro-ph/0005123 (submitted to ApJ Letters)

Haslam C.G.T., Stoffel H., Salter C.J., Wilson W.E., 1982, A&AS 47, 1

Impey C.D., Neugebauer G., 1988, AJ 95, 307

Jonas J.L., Baant E.E., Nicolson G.D., 1998, MNRAS 297, 997

Kollár I., 1994, IEEE Trans. Intrum. Meas. 43, 5, 733

Lange A.E., Ade P.A.R., Bock J.J., et al., 2000, astro-ph/0005004 (submitted to PRD)

Lasenby A., Jones, A.W., Dabrowski, Y., 1998, Proc. of the the XXXIII$^{\mathrm{rd}}$ Rescontres de Moriond, Les Arcs, France, January 17-24, 1998, p. 221

Maino D., Ph.D. Thesis, International School for Advanced Studies, SISSA/ISAS, Trieste, Italy, October 1999

Maino D., Burigana C., Maltoni M., et al., 1999, A&AS 140, 383

Mandolesi N., Laurence C.R., Bersanelli M., et al., 1998, Low Frequency Instrument for PLANCK, Proposal submitted to European Space Agency

Mandolesi N., Bersanelli M., Burigana C., Villa F., 2000, Ap. Lett. Comm. 37, Issue 3-6 (in press), astro-ph/9904135

Mandolesi N., Bersanelli M., Butler R.C., et al., 1999, PLANCK Low Frequency Instrument, Instrument Science Verification Review, LFI Design Report, October 1999

Maris M., Staniszkis M., 1998, Proc. of the Astronomical Data Analysis Software and Systems ADASS VIII, 470, Mehringer D.M., Plante R., Roberts D.A. (eds.)

Maris M., Pasian F., Smareglia R., et al., 1998, Proc. of the Astronomical Data Analysis Software and Systems ADASS VIII, 145, Mehringer D.M., Plante R., Roberts D.A. (eds.)

Maris M., Pasian F., Burigana C., et al., 1999, Proc. of the XLIII National Congress of the Italian Astronomical Society held in Naples, Italy from 4 to 8 May 1999, Agatino R. (ed.) (in press)

Maris M., 1999a, A Possible Coding Scheme for PLANCK-LFI, OAT-Tech. Rep. 55/99, PL-LFI-OAT-TN-005

Maris M., 1999b, A Proposal for a Fast Compression Method for PLANCK-LFI, OAT-Tech. Rep. 56/99, PL-LFI-OAT-TN-006

Maris M., Maino D., Bersanelli M., et al., 2000, Quantization Errors on Simulated LFI Signals, PLANCK-LFI Int. Rep.: PL-LFI-OAT-TN-011, OAT Tec. Rep. 69/2000, Pub. N. 2138

Mather J.C., Fixsen D.J., Shafer R.A. Mosier C., Wilkinson D.T., 1999, ApJ 512, 511

Muciaccia P.F., Natoli P., Vittorio N., 1997, ApJ 488, L63

Nelson M., Gailly J.L., 1996, The Data Compression Book II$^{\mathrm{nd}}$ edt. M & T Books, New York, U.S.A.

Platania P., Bensadoun M., Bersanelli M., et al., 1998, ApJ 505, 473

Press W.H., Flannery B.P., Teukolsky S.A., Vetterling W.T., 1986, Numerical Recipes. Cambridge University Press, Cambridge, U.S.A.

Puget J.-L., Abergel A., Bernard J.-P., et al., 1996, A&A 308, 5

Puget J.-L., Efstatihou G., Lamarre J.M., et al., 1998, High Frequency Instrument for Planck, Proposal submitted to European Space Agency

Reich P., Reich W., 1986, A&AS 63, 205

Schlegel D.J., Finkbeiner D.P., Davis M., 1998, ApJ 500, 525

Seiffert M., et al., 1997, Rev. Sci. Instr. (submitted)

Toffolatti L., Argueso G. F., De Zotti G., et al., 1998, MNRAS 297, 117

White M., Seiffert W., 1999, Noise Quantization LFI Tech. Rep., October 20, 1999