TITLE: # Planck LFI – GLISSANDO: a package to simulate signals with time dependent power spectra

DOC. TYPE: TECHNICAL NOTE

PROJECT REF.: PL-LFI-OAT-TN–005 **PAGE:** I of IV, 00

ISSUE/REV.: 0.1 **DATE:** 22 Apr 2003

Issued by	**Michele Maris** **Luca Terenzi**	**Date:** *22 Apr 2003* **Signature:**
Agreed by	**F. PASIAN** **LFI Program Manager**	**Date:** *Xxx* **Signature:**
Approved by	**R.C. BUTLER** **LFI Program Manager**	**Date:** *xxx* **Signature:**
Approved by	**N. MANDOLESI** **LFI Principal Investigator**	**Date:** *xxx* **Signature:**

DISTRIBUTION LIST

Recipient	Company / Institute	E-mail address	Sent

CHANGE RECORD

Issue	Date	Sheet	Description of Change	Release
0	2003 Apr 11		Draft Issue of the document	0
0	2003 Dec 28		Draft issued after improvements to figures and text.	1
1	2003 Jan 27		First Issue	0

TABLE OF CONTENTS

1 SCOPE

This document describes the package GLISSANDO for the LS of the DM pipeline.

GLISSANDO is designed to simulate slow time variability in the sorption cooler statistical properties.

The scope of the simulation is the generation of TODs of thermal instabilities to *test* and *train* the existing and the future procedures for the removal of the sorption cooler instabilities from Planck data in a not-nominal situation.

GLISSANDO is a F90 code, used to generate simulated data starting from already existing measured or simulated data streams and transfer functions forming a *simulation data-set*.

GLISSANDO accepts input *simulation data-sets* but since heterogeneous sources for the data and the transfer functions may be used in order to prepare a given *simulation data-set*, the data has to be introduced in GLISSANDO in a unique well defined format (basic a text format) avoiding GLISSANDO to deal with heterogeneous formats.

In order to simplify the effort to prepare a data set, an *Ingestion Package* written in Octave/MatLab has been developed as described in the appendix. However in order to use GLISSANDO the Ingestion Package is not needed, as it is not strictly required for the preparation of the simulation data-set.

1.1 LIMITS OF APPLICABILITY

GLISSANDO is designed to simulate slow variations of the sorption cooler statistical properties, where *SLOW* means that the time scales of the variation is large respect to the typical time scale over which the sorption cooler varies.

Since the purpose of GLISSANDO is *training* and *evaluation* of removal procedures, at the present level of development GLISSANDO is not designed to include a detailed modelling of the degradation mechanism of the cooler. Consequently at the present level of development GLISSANDO can not be used as a predictive tool.

Figures in the report refers to the Transfer Functions connecting the radiometers to the cold-end of the sorption cooler, the experimental TODs of the BBM version of the Sorption Cooler and the simulations of the Sorption Cooler currently distributed in November 2002. Since new data have been issued just before this report has been closed, these figures are no more representative of a realistic simulation of the nominal sorption cooler. However, the new TODs have a power spectrum similar to the old ones, but the amplitude which is lower. Consequently the figures in this issue of the report may still represent a valid exemplification for the use and the method of GLISSANDO.

2 APPLICABLE/REFERENCE DOCUMENTS

2.1 APPLICABLE DOCUMENTS

TBD

2.2 REFERENCE DOCUMENTS

TBD

2.3 ACRONYMS LIST

DFT	Discrete Fourier Transform
DM	Demonstration Model
F90/95	FORTRAN 90/95
FFT	Fast Fourier Transform
IDFT	Inverse Discrete Fourier Transform
LS	Level S
SC	Sorption Cooler
Sys	Systematic
TBD	To Be Defined
TF	Transfer Function
TFDT	Time – Frequency Domain Transform
TOD(s)	Time Ordered Data
par - file	parameters file
dat – file	data file

3 INTRODUCTION

This document describes GLISSANDO a software package designed to simulate long-term non-stationarities as long-term drifts in the sorption cooler (SC) of the Planck Mission within the Level S in the Demonstration Model (DM) of the Planck/LFI pipeline.

The scope of the simulation is the generation of TODs of thermal instabilities to *test* and *train* the existing and the future procedures for the removal of the sorption cooler instabilities from Planck data in a not-nominal situation.

The code simulates long term non-stationarities in the SC starting from the Discrete Fourier Transform (DFT) of the SC output at its interface with the instrument, the cold-end, allowing the coefficients of the DFT to vary in time with a time scale much longer than the time scale of the over which the cooler oscillates.

GLISSANDO is a F90 code, used to generate simulated data starting from already existing measured or simulated data streams and transfer functions forming a *simulation data-set*, which are given to GLISSANDO in a unique well defined format (basically a text format) avoiding GLISSANDO to deal with heterogeneous formats.

In order to simplify the preparation of a data set, an *Ingestion Package* written in Octave/MatLab has been developed as described in the appendix. However in order to use GLISSANDO the Ingestion Package is not needed, as it is not strictly required for the preparation of the simulation data-set.

4 ASSUMPTIONS, REQUIREMENTS, INTERFACES WITH THE BBM

This part of the document describes the list of assumptions, requirements for the simulation and the interface with the BBM.

4.1 ASSUMPTIONS

ASMP-1 The unperturbed cooler generates a periodical signal over a given duty-cycle.

ASMP-2 Drifts are slow respect to the SC duty cycle.
A typical drift time is the time spanned by the mission to be completed.

ASMP-3 Transfer Functions (TF) for any horn will not depend on time.
A time varying TF is not expected since it would indicate the occurrence of some bad hardware faillure.

4.2 SIMULATION REQUIREMENTS

SIM-1 Code shall simulate slow SC drifts over the nominal data acquisition phase of the full mission.
Launch, Transfer Phase and Commissioning are outside the scope of this simulation.

SIM-2 Code shall allow the simulation of slow SC drifts over a used defined time interval.

SIM-3 Code shall base its simulations on existing real and simulated data.

SIM-4 Code shall be able to generate TODs for any horn.

SIM-5 Code shall be able to generate data resampled to a given sampling frequency.

SIM-6 Code shall be able to generate coadded data.

SIM-7 Code shall be allowed to exclude resampling

SIM-8 Code shall be allowed to exclude coadding

SIM-9 Code shall be as fast as possible.

4.3 INTERFACE WITH THE BBM

BBM-IC-1 **Within the BBM the code shall work as an independent program.**
This of course does not exclude modularization to allow the use of basic functionalities in other parts of the DM.

BBM-IC-2 **Code shall receive parameters from the BBM through a parameter file, formatted as a F90 NAMELIST.**

BBM-IC-3 **Code shall be able to write data as BINARY tables in FITS**

BBM-IC-4 **Input from simulations for the code shall be stored in separated files.**
The reason for this is to allow improvements or changes in the basic information, as an example when a new simulated data will be at disposition, without to recompile the code.

5 SIMULATION METHOD

5.1 BASIC PRINCIPLE

The active cooling system (Sorption Cooler) in Planck/LFI is known to produce thermal fluctuations in the signal of sky and reference load, resulting in nearly periodical time variations of the baseline of each radiometer. The sorption cooler is composed of 6 cooling units, named *sorption beds*, each one forced to follow a cooling cycle with constant period $t_{duty-cycle}$ and constant phases whose differences are multiples of $2\pi/6$. The cooler is interfaced to the focal plane (and through it to the radiometers) through two cold-ends whose temperature T_{scce} depends on the total cooling efficiency of the cooler sum of the cooling efficiency of each unit. The cooling efficiency for each unit varies in time along with the duty-cycle, but in the ideal case the six units would be balanced to allow complete cancellation of fluctuations in the sum and then in the T_{scce}. However in the true case unavoidable slight unbalances lead to the observed residual thermal fluctuations whose main Fourier components are locked to period and phases of the duty-cycle. The transfer of T_{scce} to each radiometer is then calculated once the thermal transfer functions between the cold-ends and the cooler for each radiometer are known. In the nominal case the unbalances and the transfer functions are constant leading to a stationary signal.

The cooling system degrades along the mission, so that after some time its status will be no more nominal. In particular the most important degradation likely to occur is a gradual unequal loss of efficiency of each cooling unit, while phase and periods of the duty cycles as the transfer functions are unlikely to vary in time in a comparable manner. This is equivalent to assume a time dependence in the amplitudes of the different Fourier components of T_{scce}.

The code then simulates the effect of a slow variation of the amplitudes in the Fourier components of T_{scce} assumed to vary over a characteristic degradation time scale $t_{\deg}$ with the assumption

$$t_{\deg} >> t_{corr,sc},$$

where $t_{corr,sc}$ is the correlation time for the signal seen at a given radiometer which is a function of t_{sc} and the time lags $t_{transfer-lag}$ induced by the transfer functions. Since $t_{sc} \approx 4\times10^4$ sec, $t_{transfer-lag} \approx t_{sc}$ then $t_{sc} < t_{corr,sc} < 1/2$ day so that assuming a worst case $t_{\deg} \approx 1/2$ year then the condition is surely respected.

The code then will need in input the Discrete Fourier Transform (DFT) of the nominal T_{scce} and the law describing the time dependence of the power spectrum.

5.2 FORMALISM

Let be T_{scce} to denote the sorption cooler signal at the cold end, with $T_{sc,h}$ the sorption cooler signal as seen by a radiometer at the horn *h*, T_h the transfer function for the horn *h*, *t* the time.

In Fourier space $\tilde{T}_{scce,k}$, $\tilde{T}_{sc,k}$ and $\tilde{\mathrm{T}}_{h,k}$ will represents the Fourier Transforms of T_{scce} , $T_{sc,h}$ and T_h respectively.

Since T_{scce} is a stationary and periodical time serie with period t_{sc} the Fourier Transforms reduces to a Discrete Fourier Transform (DFT), so than:

$$T_{sc,h}(t) = \sum_k \tilde{\mathrm{T}}_{h,k} \tilde{T}_{scce,k} \exp(i\omega_k t), \qquad (1)$$

where *k* and ω_k denotes the index and the frequency of the Fourier modes.

A drift in the periodic behaviour of T_{scce} may be represented as a time dependence in the Fourier coefficients, so replacing $\tilde{T}_{scce,k} \rightarrow \tilde{T}_{scce,k}(t)$ in eq. (1):

$$T_{scce}(t) = \sum_k \tilde{T}_{scce,k}(t) \exp(i\omega_k t). \qquad (2)$$

But if T_{scce} is no longer stationary the convolution theorem is no longer valid, and eq. (1) and (2) are just approximations. However, if the time scale over which $\tilde{T}_{scce,k}$ varies t_{deg} is large compared to the correlation time scale for the T_{scce} induced by the transfer function then

$$T_{sc,h}(t) \approx \sum_k \tilde{\mathrm{T}}_{h,k} \tilde{T}_{scce,k}(t) \exp(i\omega_k t). \qquad (3)$$

is still a valid approximation.

The kind of time dependence for $\tilde{T}_{scce,k}(t)$ has to be inferred from data. Up to now no long term information for the behaviour of the cooler is know. As a consequence it is assumed that the model for the time dependence of the amplitudes of the various modes is linear (or polynomial) so that:

$$\tilde{T}_{scce,k}(t) = \sum_{j=0}^{n} a_{k,j} t^j \exp(i\varphi_k) \qquad (4)$$

where $a_{k,j} \in \Re$ are the coefficients of the polynomial dependence, φ_k are the phases of the various modes, which is left unchanged..

Equation (4) represents an absolute time dependence for the power spectra. Relative variations may be defined as well, in this case, assuming $t = 0$ as the origin of times:

$$\tilde{T}_{scce,k}(t) = \sum_{j=1}^{n}(1 + b_{k,j}t^{j})a_k \exp(i\varphi_k), \tag{5}$$

where $a_k \in \Re$ are normalizations, $b_{k,j} \in \Re$ are the coefficients of the polynomial dependences, φ_k are the phases of the various modes.

In order to speed-up the code the Inverse Discrete Fourier Transform (IDFT) of eq. (3) and since over a duty cycle the unperturbed SC is periodical, the exponential terms $\exp(i\omega_k t)$ are pre-computed and stored at start-up. This limits the number of samples per duty cycle which is possible to generate, but, since the measured SC signal is noise dominated at the highest frequencies, the cost is largely compensated by the saving in computational time time[1].

Presently data about the cooler allows to definedefining just the behaviour of nominal SC and of the degraded SC. The time evolution then may be defined by the power spectra of these two extreme cases. Assuming $t = 0$ as the origin of time, $t = t_{\deg}$ as the degradation time, and denoting with $\tilde{T}^{nominal}_{scce,k}$ the Fourier Transform of the nominal cooler, and with $\tilde{T}^{degraded}_{scce,k}$ the Fourier Transform of the degraded cooler, equations (4) and (5) have to assure that

$$\left|\tilde{T}_{scce,k}(t = 0)\right|^2 = \left|\tilde{T}^{nominal}_{scce,k}\right|^2, \tag{6a}$$

$$\left|\tilde{T}_{scce,k}(t = t_{\deg})\right|^2 = \left|\tilde{T}^{degraded}_{scce,k}\right|^2. \tag{6b}$$

For the absolute time dependence of equation (4), this translates into a linearlinear time dependence with:

$$a_{k,0} = \left|\tilde{T}^{nominal}_{scce,k}\right|, \tag{7a}$$

$$a_{k,1} = \frac{\left|\tilde{T}^{degraded}_{scce,k}\right| - \left|\tilde{T}^{nominal}_{scce,k}\right|}{t_{\deg}}. \tag{7b}$$

For the relative time dependence of equation (5), this translates into a linearlinear time dependence with:

$$a_k = \left|\tilde{T}^{nominal}_{scce,k}\right|, \tag{8a}$$

[1] We reserve ourselves to allow the use of an ordinary DFT, without precomputed terms as a future improvement.

$$b_{k,1} = \frac{\left|\tilde{T}^{\deg raded}_{scce,k}\right| - \left|\tilde{T}^{no\,min\,al}_{scce,k}\right|}{\left|\tilde{T}^{no\,min\,al}_{scce,k}\right|} \frac{1}{t_{\deg}} . \qquad (8b)$$

However while for $\left|\tilde{T}^{\deg raded}_{scce,k}\right|$ only simulated data exists, for $\left|\tilde{T}^{no\,min\,al}_{scce,k}\right|$ both simulated and measured data are present. As shown the next section, simulated data are not directly comparable to measured data, since measured data have fluctuations which are wider than simulated data, even in the case of the nominal SC. In addition, the real data have a richest spectrum of fluctuations. For this reason we only simulate relative variations with a_k defined by nominal, measured data and $b_{k,1}$ by nominal and degraded simulated data then

$$a_k = \left|\tilde{T}^{no\,min\,al,simulated}_{scce,k}\right|,$$

(9a)

$$b_{k,1} = \frac{\left|\tilde{T}^{\deg raded,simulated}_{scce,k}\right| - \left|\tilde{T}^{no\,min\,al,simulated}_{scce,k}\right|}{\left|\tilde{T}^{no\,min\,al,simulated}_{scce,k}\right|} \frac{1}{t_{\deg}} .$$

(9b)

While phases are assumed to be constant in time and defined as:

$$\varphi_k = \arg\left(\tilde{T}^{no\,min\,al}_{scce,k}\right), \qquad (10)$$

note that the fact that phases are left unchanged is equivalent to assume that the final signal will have the same *power spectra* expected by the simulations, but with the phases of the nominal signal. This means that the statistical properties (which depend on the *power*) will be those of the simulated data, but the time dependence may appear different. Having not data allowing a meaningful hypothesis of how to model the time dependence in phase, we did not attempt any simulation. However the code is written to allow the simulation of time dependence in phase which will be added as soon as data will allow it.

Data Streams In Input to the Simulator

This section describes the main features of the data in input to the simulator and how the time dependence of the Fourier coefficients is calculated.

Figures in the report refers to the Transfer Functions connecting the radiometers to the cold-end of the sorption cooler, the experimental TODs of the BBM version of the Sorption Cooler and the simulations of the Sorption Cooler currently distributed in November 2002. Since new data have been issued just before this report has been closed, these figures are no more representative of a realistic simulation of the nominal sorption cooler. However, the new TODs have a power spectrum similar to the old ones, but the amplitude which is lower. Consequently the figures in this issue of the report may still represent a valid exemplification for the use and the method of GLISSANDO.

5.3 THE MEASURED TSCCE

5.3.1

Figure 1 represents the measured data for $T_{scce}(t)$ as a function of time [1]. Data are sampled every second. The full data represents five consecutive cycles of the SC, represented by five different colours. It is clear that the data are only approximately periodical since there is an important stochastic component. The lower frame of the figure represents the same data after transferring to a 30GHz horn with the appropriated transfer function. Note the nearly complete removal of the stochastic noise. The amplitudes (square roots of power spectrum) for the Fourier decomposition of $T_{scce}(t)$ and of $T_{sc,h}(t)$ for a 30 GHz horn ($T_{sc,30\,\mathrm{GHz}}(t)$) are displayed in figure 2.

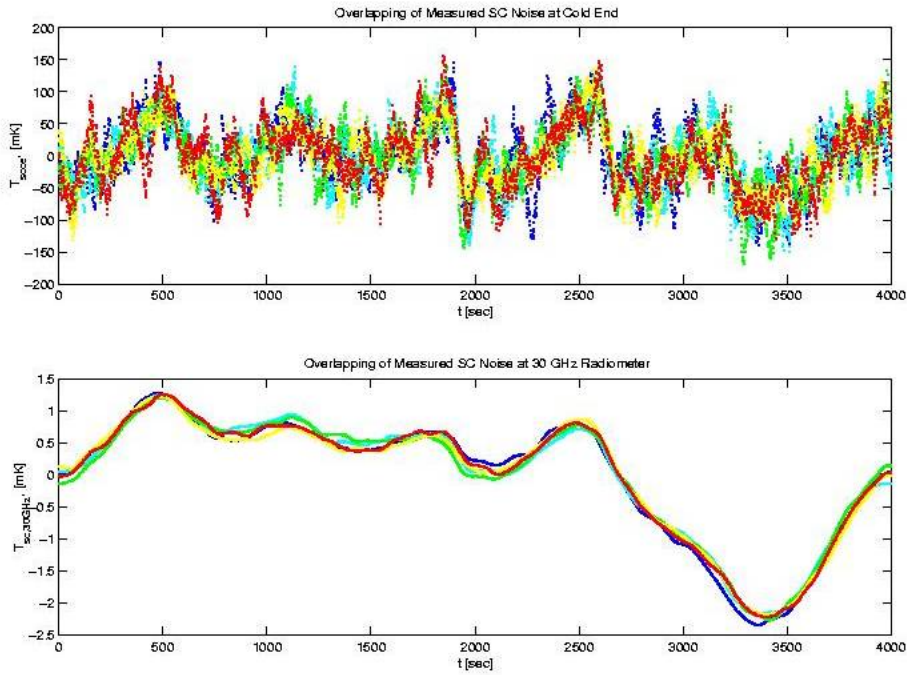

Figure 1 Measured data at the cold end. The sampling time is 1 Hz. The five colours represents the five consecutive measured cycles obtained from the SC BBM. The entire TOD lasts for about 20000 sec, the sorption cooler having a period of 4000 sec. Upper frame the original measures, lower frame expected signal in a 30 GHz horn. The signal in the lower frame has been shifted to take in account of the delay between the cold-end variation and the variation observed in the radiometric channel.

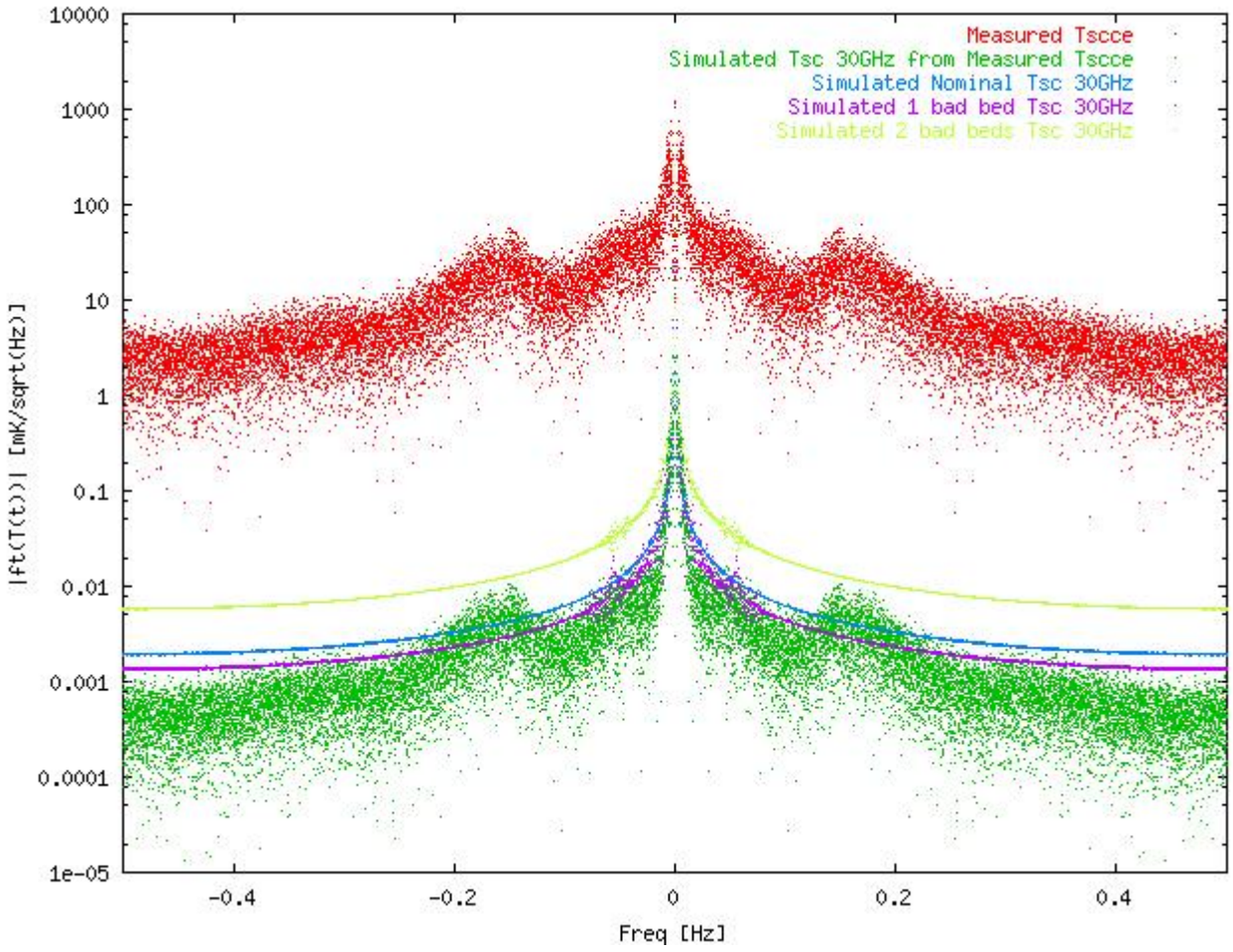

Figure 2 Amplitudes of the Fourier modes of T_{scce}, $T_{sc,30\,\mathrm{GHz}}$ for the 30 GHz channel from measured data of the BBM SC and simulations in the nominal case and in degraded cases. Red are points for T_{scce} measured at the BBM SC representing the nominal case. Green is the nominal expected $T_{sc,30\,\mathrm{GHz}}$ obtained after application of a transfer function to the measured T_{scce}. Blue, violet and yellow are $T_{sc,30\,\mathrm{GHz}}$ simulated assuming to be in the nominal case and with 1 or 3 bad cooling units (sorption beds) respectively.

5.4 THE SIMULATED TSCCE, NOMINAL, 1 BAD BED, 3 BAD BEDS

5.4.1

Figure 3 Simulated TODs for the 30 GHz and the 100 GHz frequency channel in the case of nominal SC (blue), 1 bad bed (magenta), 3 bad beds (green).Figure 3 represents simulated TODs for the 30 GHz and 100 GHz channels. At the moment no simulation for pure T_{scce} exists, however since we are interested to simulate the *relative* variation of the amplitudes of the various modes, this is unimportant since in Fourier space for a time independent transfer function, the relative variation of the FT of $T_{sc,h}$ concides with the corresponding relative variation of the FT of T_{scce}. So simulations for variations of $T_{sc,h}$ for a single horn of the 30 GHz channel are sufficient to obtain parameters valid for the full set of horns.

Figure 2 displays the FT of a simulated nominal cooler for the 30 GHz channel, obtained from a crude simulation. Note that despite the simplifications in the model, the FT of the simulated signal is a crude but, for our purposes, acceptable representation of the envelope of the Fourier Transform of $T_{sc,30\,\mathrm{GHz}}(t)$ at frequencies between 0.05 Hz and 0.015 Hz, while it underestimates the power at lower frequencies and overestimates it at the higher. An improved thermal and SC model will surely allow solving these problems. The simulation also misses the bump at 0.018 Hz.

Figure 4 represents the relative variations of amplitudes of Fourier modes, i.e. $b_{k,1}t_{\mathrm{deg}}+1$ as defined in eq. (9a), degrading from the nominal case to either 1 bad cooling unit or 3 bad cooling units. It is interesting to note that from the model it is possible to conclude that the failure of a single SC unit removes the power at high frequencies increasing the power at low frequencies even of more of a factor 20. In addition apart from a narrow band about 0 Hz, the $b_{k,1}t_{\mathrm{deg}}+1$ is constant. This is also true for the failure of 3 beds, but here the constant part is nearly equivalent to an increase of about a factor 3 of amplitudes.

Figure 5 represents the histogram of $b_{k,1}t_{\mathrm{deg}}+1$ plotted in figure 4.

5.5 SIMULATED TSCCE AND TSC AT 30 GHZ FOR ONE YEAR VARIATION

Putting all together it is possible to simulate the effect of linear degradations for the cases under study.

Figures 6, 7 compare the effect of 1 bed degradation or 3 beds degradation after 14 months against the nominal cooler. In figure 7 it is evident how the effect of the degradation can not be simply reduced to a rescaling of the nominal signal in the time space.

Figure 8, 9 shows the effect of degradation in time along the 14 months taken as a reference time it is evident how new features “born” as the cooler degrade and how the way in which degradation proceeds strongly influences the final shape.

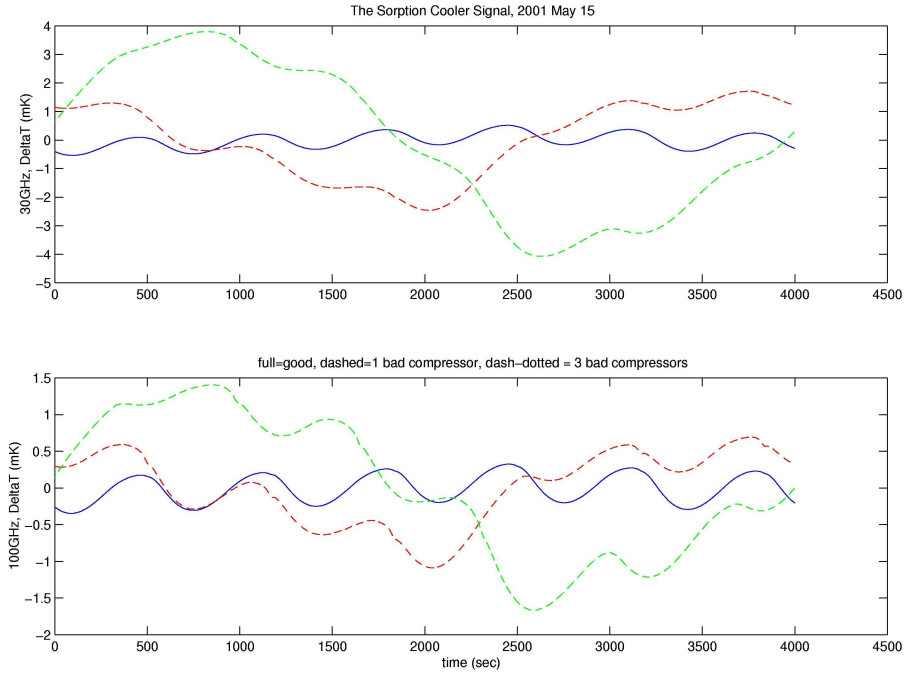

Figure 3 Simulated TODs for the 30 GHz and the 100 GHz frequency channel in the case of nominal SC (blue), 1 bad bed (magenta), 3 bad beds (green).

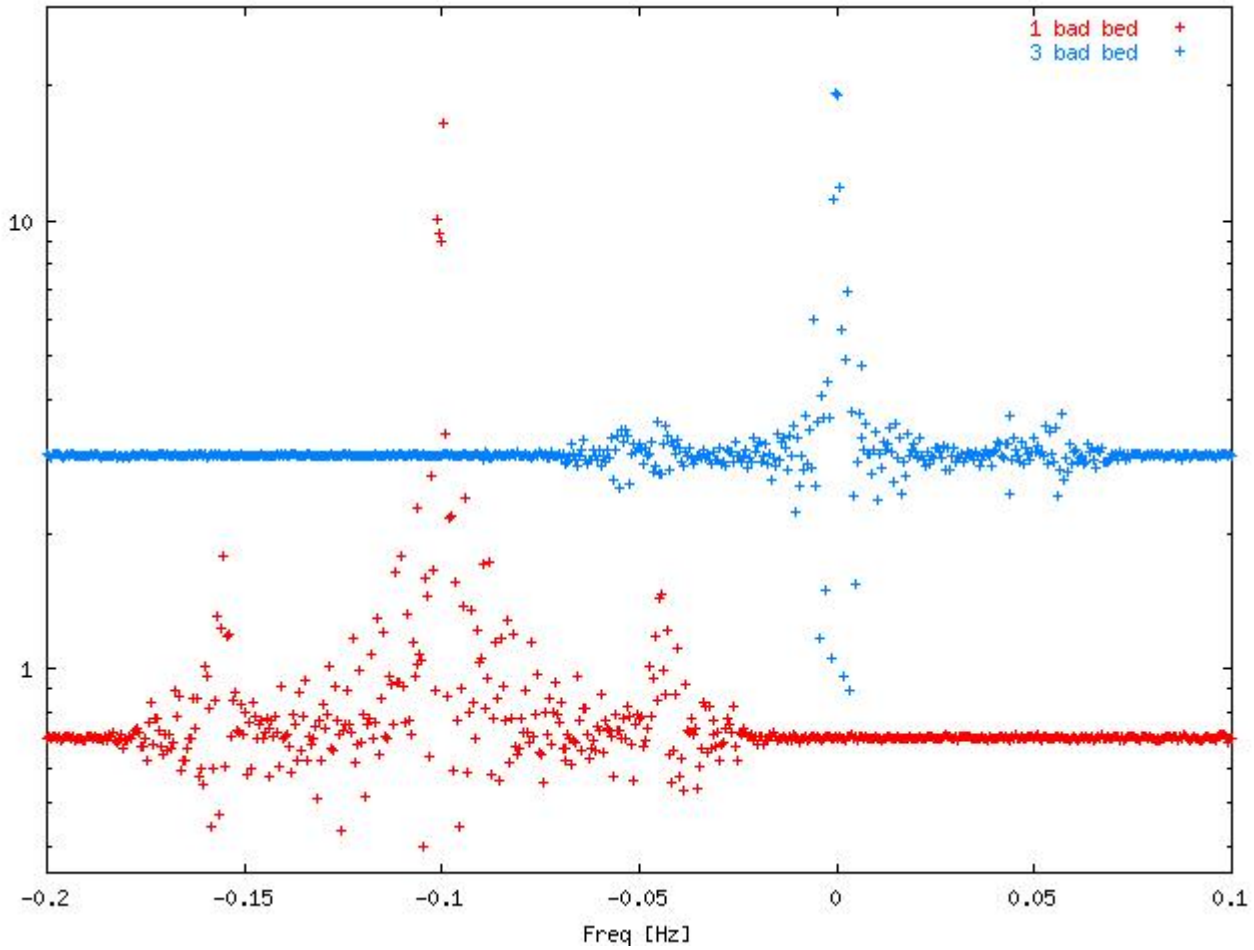

Figure 4 Ratio of amplitudes of Fourier modes for a simulated $T_{sc,h}$ at 30 GHz for 1 bad sorption bed (red) or for 3 bad sorption beds (blue) versus the nominal case. The plot for the 1 bad case is shifted of –0.1 Hz to avoid overlap with the other plot. To reduce noise, plots are binned with a bin width of 5×10^{-4} Hz.

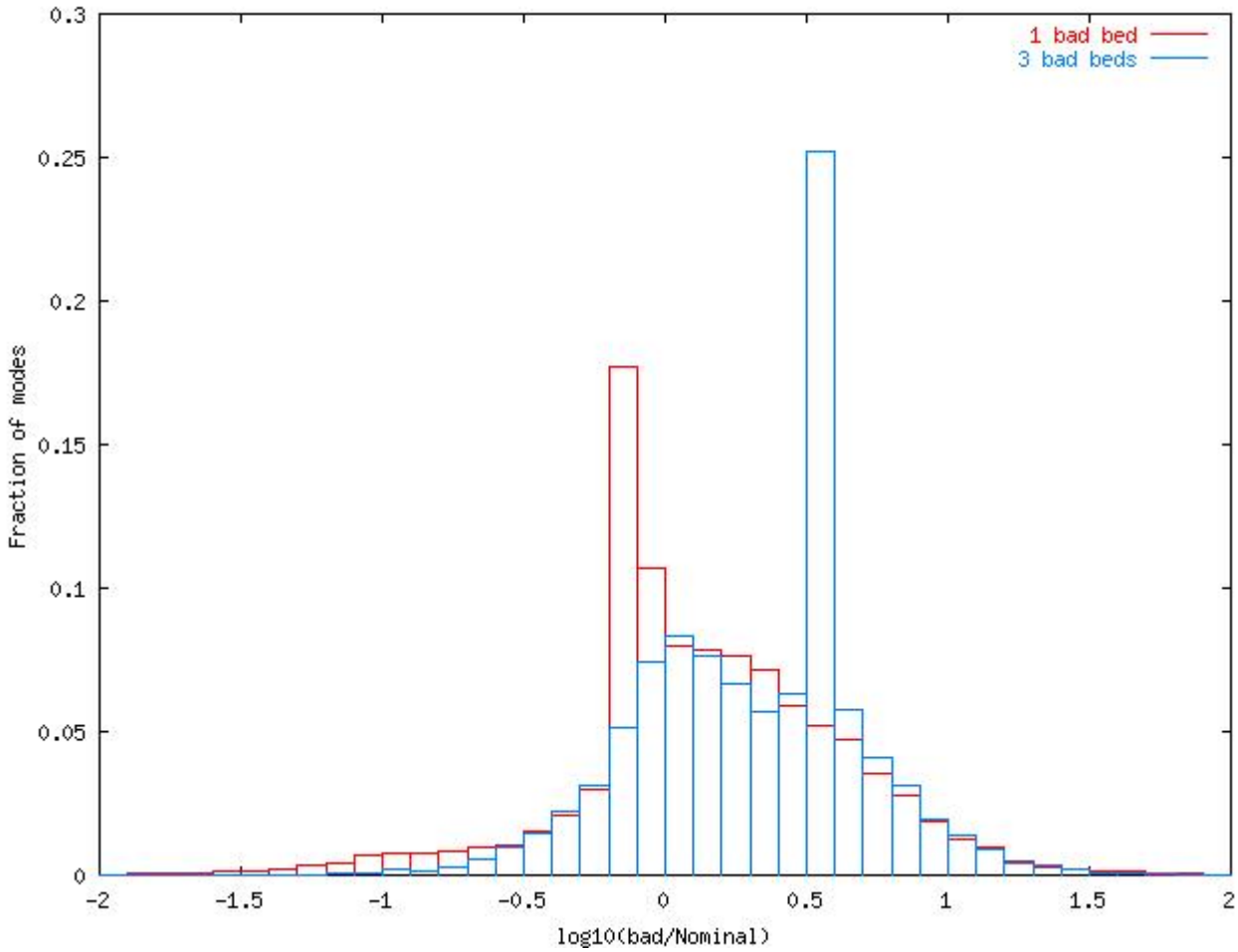

Figure 5 The histogram for the $b_{k,1}t_{\deg}+1$ ***values plotted in figure 4.***

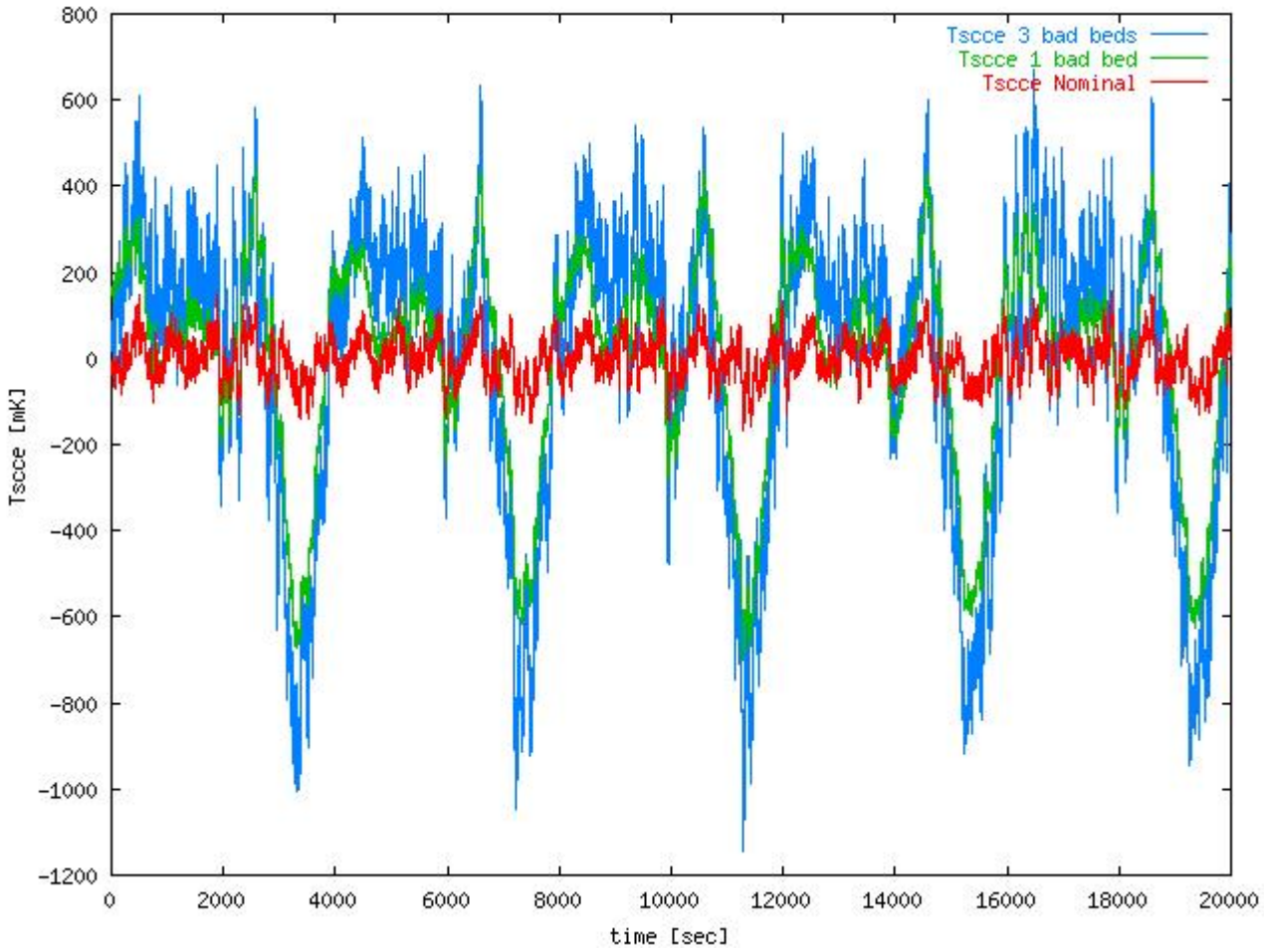

Figure 6 TODs of T_{scce} for the nominal case (red) and simulated for complete degrading of 1 sorption bed (green) and 3 sorption beds (blue).

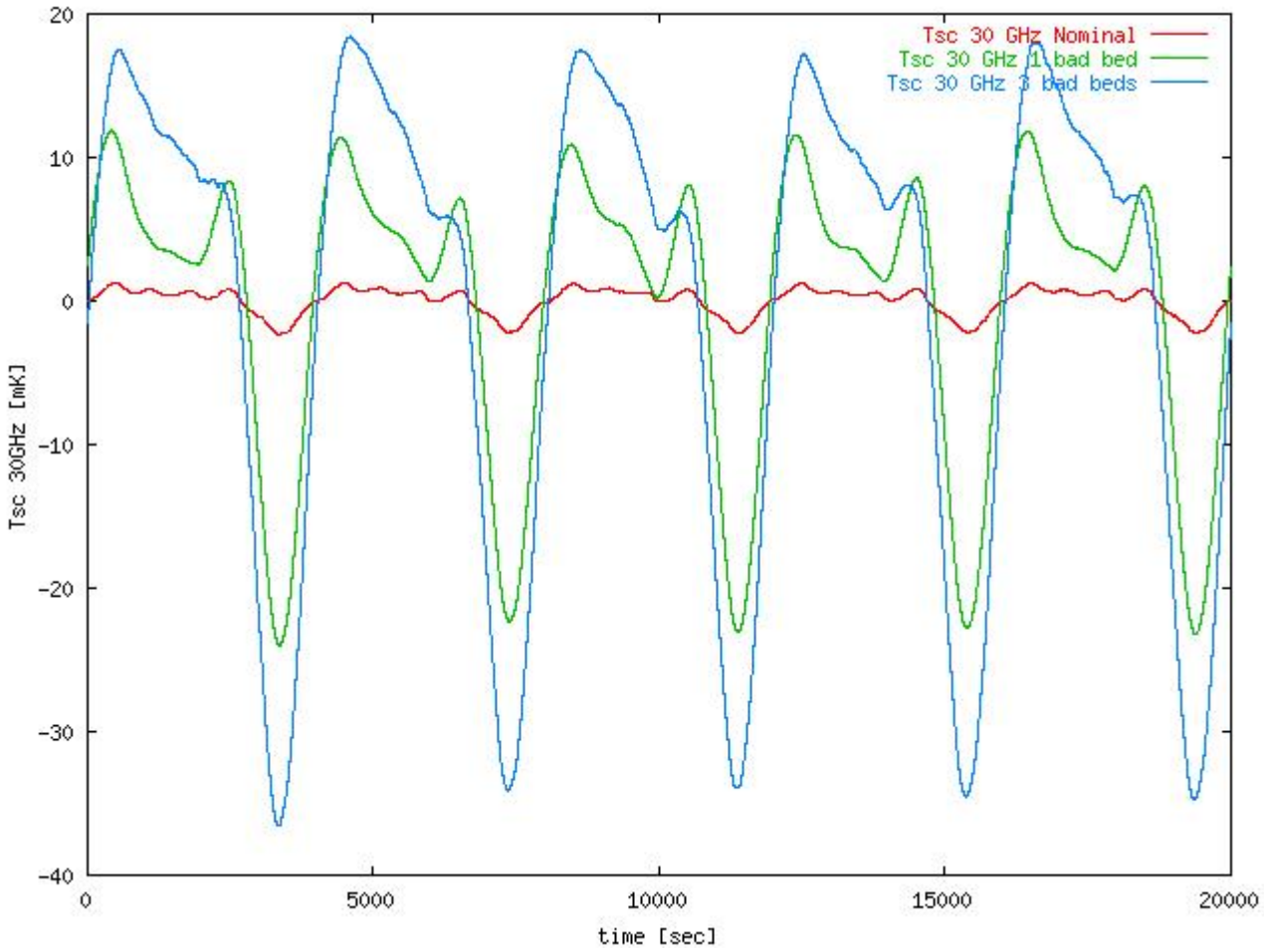

Figure 7 TODs of $T_{sc,30\,\mathrm{GHz}}$ for the nominal case (red) and simulated for complete degrading of 1 sorption bed (green) and 3 sorption beds (blue).

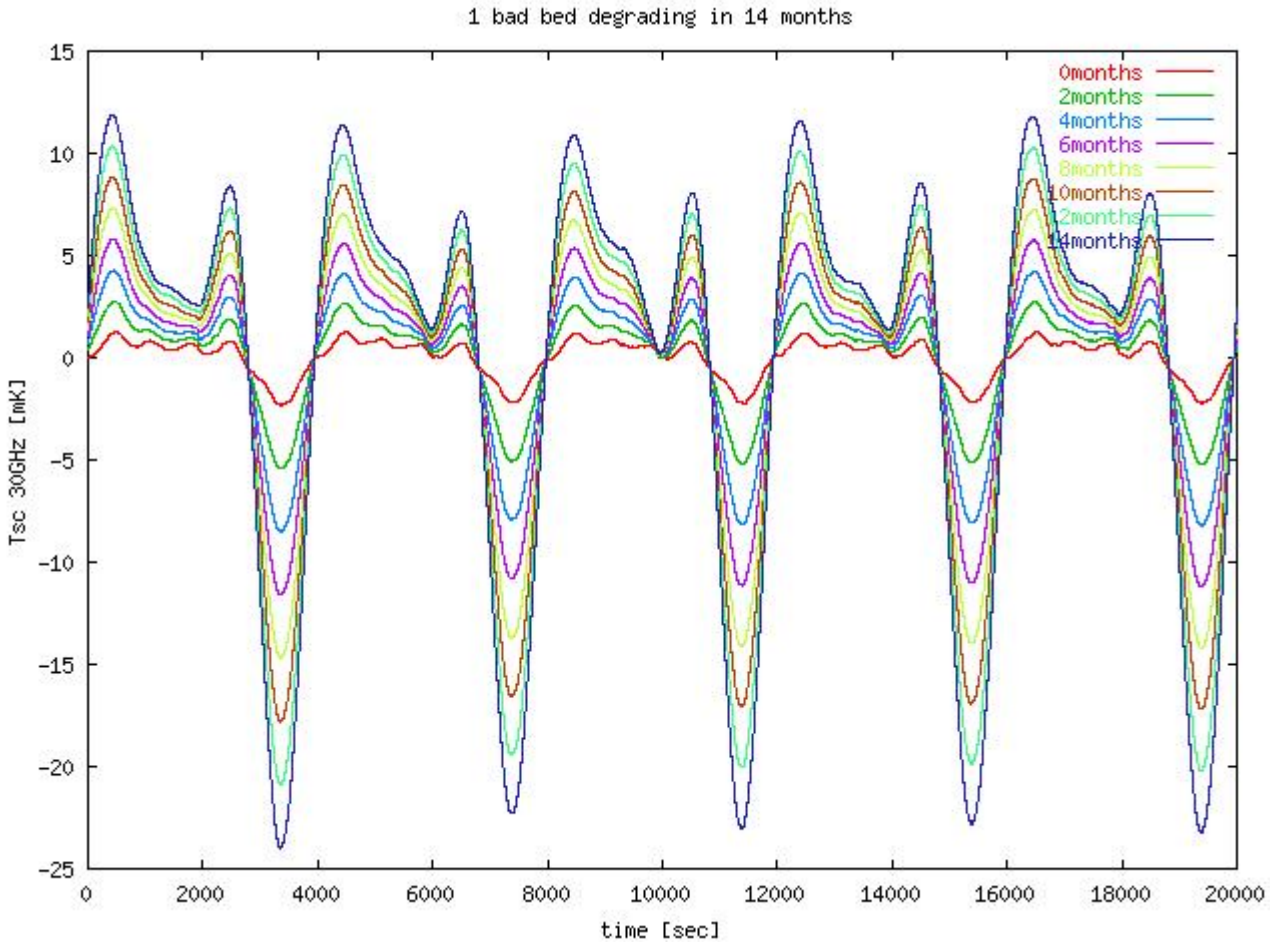

Figure 8 Simulated degradation of $T_{sc,30\,\mathrm{GHz}}$ for the case of a 1 bad bed in 14 months of mission. Curvers are generated in steps of 2 months from zero (red) to 14 months (black).

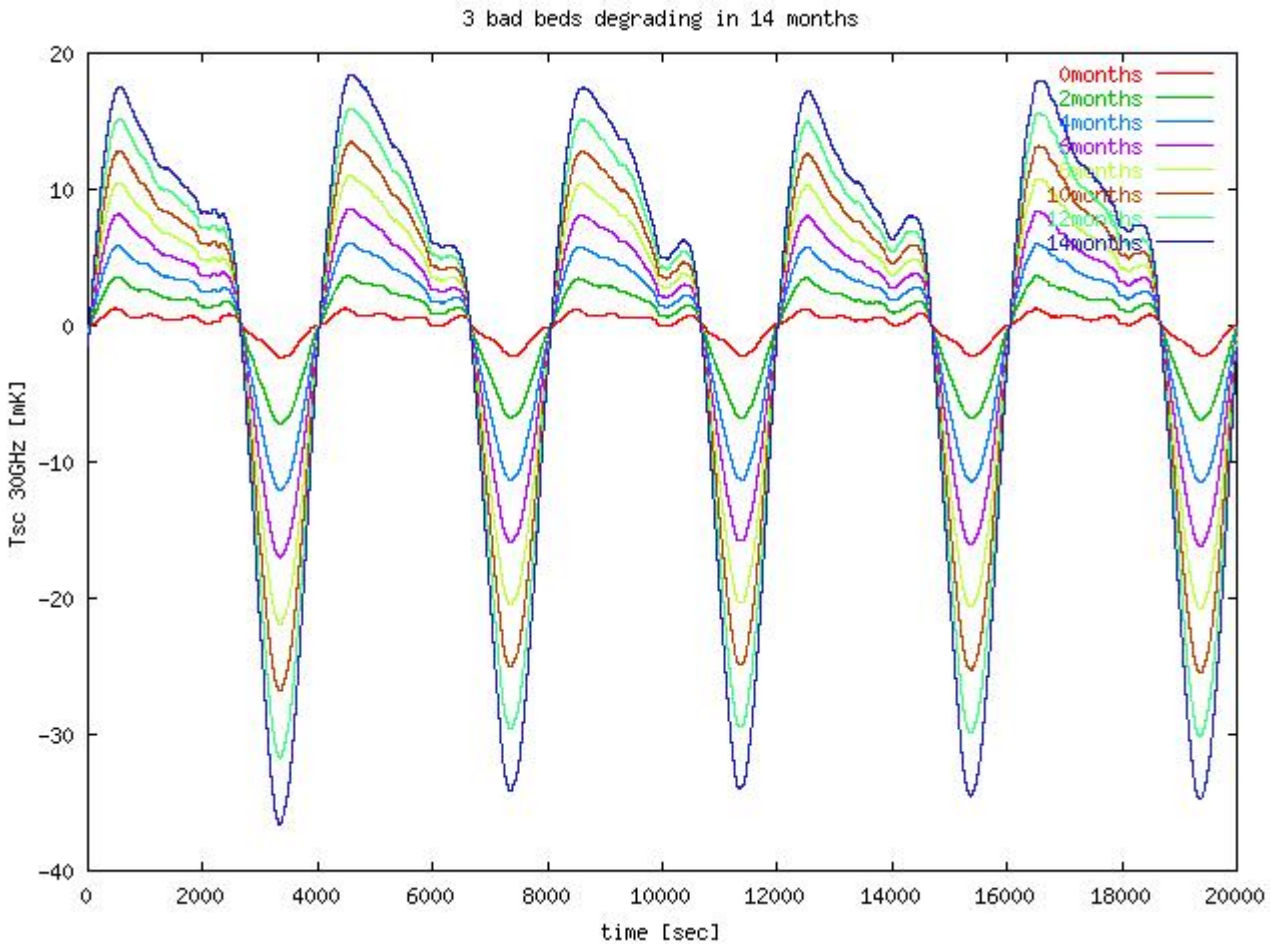

Figura 9 Simulated degradation of $T_{sc,30\,GHz}$ for the case of a 3 bad beds in 14 months of mission. Curvers are generated in steps of 2 months from zero (red) to 14 months (black).

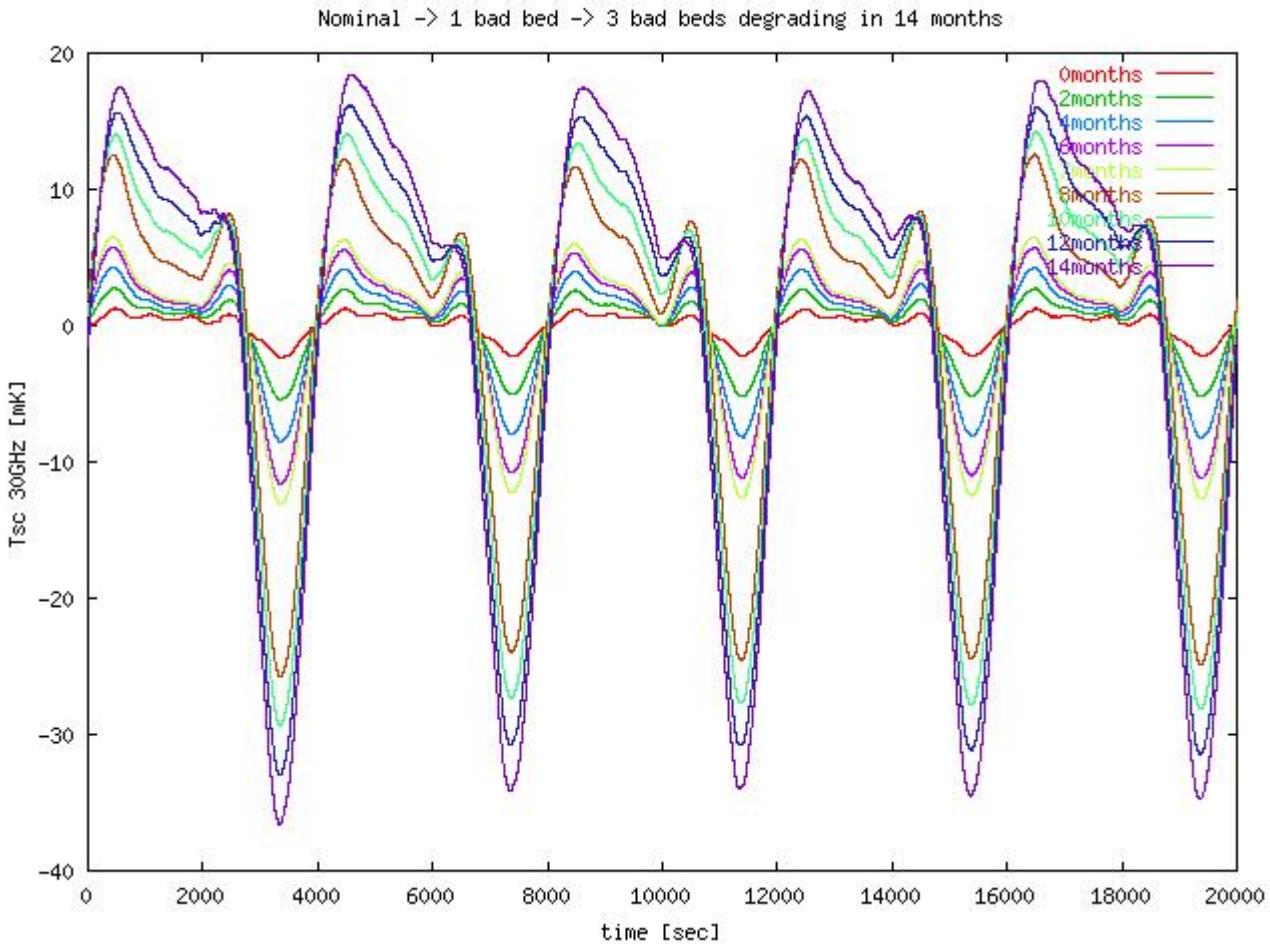

Figure 10 Simulated degradation of $T_{sc,30\,GHz}$ for the case of a system degrading from the nominal case to 1 bad bed in the first 7 months (curves from red to yellow) and from 1 bad bed to 3 bad beds in the remaining 7 months (from yellow to black).

5.5.1

5.6

6 FILES FORMAT

6.1

6.2 THE DATA FORMAT

Files containing the Fourier Transform Coefficients and the related Transfer Functions are written in ASCII (text) format so to allow simple inspection and editing. Since these data does not occupy too much space (a typical table takes at most 1 Mega Byte). There is no reason to use FITS and Binary Data Formats.

The ASCII Files carries both the data and the metadata required by GLISSANDO to operate, plus some simple comment.

GLISSANDO contains all the libraries needed to handle such data.

6.2.1 DATA FORMAT RULES

DFR-1 Files are organized in blocks, each block being either a header (containing metadata) or a vector of data.

DFR-2 Headers or metadata are stored using the conventions of F90, `NAMELIST`.

DFR-3 Each header begins with a line with “&” in the first column followed by the header name:

```
&header_name
```

DFR-4 Each header ends with a line with the “/” character in the first column.

DFR-5 Each line between the begin and the end may be either:

i. an empty line
ii. a comment line
iii. an entry line

DFR-6 an empty line is a line without characters or only with spaces or tabs

DFR-7 a comment line is a line starting with “!”

DFR-8 an entry line is a line definining a keyword name and a value according to the sintax:

```
keyword_name = value,
```

note the mandatory "," ending the line.

DFR-9 Strings are delimited by the single-quote character "'".

DFR-10 comment lines and empty lines are skipped.

6.2.2 STRUCTURE OF THE COEFFICIENT TABLES

Each file is structured in 5 blocks:

First block named: GLISSANDO_TABLE_VERSION
contains the description of the version of the structure

Second block named: GLISSANDO_TABLE
contains the description of the table

Third block named: PHASES
contains the description of the phase table
followed by a list of values of phases and their time dependences coefficients

Fourth block named: AMPLITUDES
contains the description of the amplitudes table
followed by a list of values of amplitudes and their time dependences coefficients

Fifth block named: FREQUENCIES
contains the description of the frequencies table
followed by a list of values of frequencies

7 THE SIMULATION CODE

This section describes the simulation code and its modules.

The code gets in input tables generated from the ingestion code plus the par-file for the simulation.

Simulation is organized in *cycles* the simulator generates *cycles* of SC data, eventually resampled and coadded.

A cycle is characterized by: the cycle number $I_{\rm cycle}$, the time span $t_{\rm cycle}$, the sampling time $\delta t_{\rm samp}$ and the number of samples in the cycle $N_{\rm samp}$, Of course $t_{cycle} = \delta t_{samp} N_{samp}$. Assuming all the parameters as constant, then the starting and ending times of the cycle number $I_{\rm cycle}$ are

$$t_{start}(I_{cycle}) = \delta t_{samp} N_{samp} I_{cycle},$$

$$t_{end}(I_{cycle}) = \delta t_{samp} N_{samp} (I_{cycle} + 1).$$

The cycle index ranges between 0 and `NumCycles-1`.

For data chunks not resampled and coadded, N_{samp} is fixed by the length of data stream representing a TOD of SC. The present input is $N_{samp} = 2000$ being the TOD of the SC sampled every 10 sec for 20 000 sec. The number of samples is stored in the field with keyword `Nmodes` of the file of DFT coefficients choose for input.

Data in Input are

Coefficient Files contained in GLISSANDO/DATA/DFT
Transfer Function Files contained in GLISSANDO/DATA/TF

Transfer function files are used to generate TOD for a given radiometer belonging to a given OMT.

Note that TF files has to be selected sampled in the same manned as coefficients files, i.e. with the same `Nmodes`.
It would be good practice to put `Nmodes` in file names.

So for `Nmodes` = 2000 one has to use (as an example):

DFT/1bed_1year_tvariation_nk=2000.dat
TF/tf_horn10_nk=2000.dat

Output may be in FITS or ASCII.

FITS has again to be written and implemented

The ASCII output is a simple list of numbers, one sample for line.

If ASCII output is selected in Issue 1.0 RecordLength is automatically calculated.

In Issue 1.1 the width of each row will be fixed by the `RecordLength` parameter.

For not resampled data `RecordLength` has to coincide with `Nmodes` parameter defined in the files of DFT coefficients given in input.

APPENDIX A

8 USE OF THE SIMULATION CODE

This section describes how to run the code and how to tune the parameters file.

8.1 THE PARAMETERS FILE

Parameters needed by GLISSANDO are passed through a specific par-file.

8.1.1 FORMATTING RULES FOR PAR-FILES

The par-files are structured according to the following rules:

PARF-1 Par-Files are coded using the conventions of F90/95, `NAMELIST`.

PARF-2 Each par-file carries information for a single simulation. If more simulations are queued in the same par-file, only the first one will be executed.

PARF-3 Each par-file begins with a line with:

```
&header_name
```

where the & is in the first column.

PARF-4 Each header ends with a line with the "/" character in the first column.

PARF-5 Each line between the begin and the end may be either:

i. an empty line
ii. a comment line
iii. an entry line

PARF-6 An empty line is a line without characters or only with spaces or tabs

PARF-7 A comment line is a line starting with "!"

PARF-8 An entry line is a line defining a keyword name and a value according to the syntax:

```
keyword_name = value,
```

note the mandatory "," ending the line.

PARF-9 Strings are delimited by the single-quote character "'".

PARF-10 Comment lines and empty lines are skipped.

8.1.2 AN EXAMPLE OF PAR-FILE

This is an example of par-file:

```
!
! List of parameters to operate a test1 for GLISSANDO
!
&GLISSANDO_PAR
! Name of the DFT  coefficients file
  FN_DFTCoeff = '1bed_1year_tvariation.dat',
!
! Name of the output file
  OFName      = 'glissando_1bed_1year.out',
!
! T for ascii, F for FITS
  iWantASCII  = T,
!
! Record Length in the Output File
  RecordLength = 1980,
!
! Number of cycles to be generated
  NumCycles   = 1,
!
! if true use Transfer Function
  iWantTransferFunction = T,
! the file name for the TF
  File_TransferFunction   = 'tf_horn10_nk=2000_phi=0.dat',
!
! T for resampling of data
    iWantResampling  = F,
! 0 for linear interpolation
    ResamplingMethod = 0,
!sec, Step Time for resampling
    ReSamplingTime   = 0.030303030303030303,
!
! T for coadding data
    iWantCoadding    = F,
! Number of (resampled) independent samples in a coadded chunck
    CoaddingL        = 1980,
! Number of chuncks to be coadded
    CoaddingN        = 60,
!
! Parameters for RTF calculation
!
/
```

8.1.3 KEYWORDS FOR THE PAR-FILES

Keyword Name	Type	Content	Ranges of values
`FN_DFTCoeff`	String	Name of the coefficients for the DFT	
`OFName`	String	Name of the Output File	
`iWantASCII`	Logical	T for output in ASCII, F for FITS	T, F
`RecordLength`	Integer	The length of the chunk of data generated at each cycle and written in output	positive, not null integer
`NumCycles`	Integer	Number of cycles to	positive, not null integer
`iWantTransferFunction`	Logical	Whether to use or not a transfer function. If a transfer function is not used $\widetilde{T}_{scce,k}$ will be generated.	F, T
`FN_TransferFunction`	String	Name of the file containing the transfer function. Beware, use a TF sampled as the DFT given in input (same `NMODES`).	
`iWantResampling`	Logical	T to resample data after DFT, F to not resample data	F, T
`ResamplingMethod`	Integer	The resampling method.	0 = linear method
`ReSamplingTime`	Float	The time step after resampling in sec.	positive, not null float
`iWantCoadding`	Logical	T to coadd data after resampling, F to not coadd data	F, T
`CoaddingL`	Integer	Number of samples in a coadded chunk (present version, must be equal to `RecordLength`)	positive, not null integer
`CoaddingN`	Integer	Number of chunks to be coadded and averaged	positive, not null integer

8.2 RUNNING THE CODE

See the installation section

9 TESTING OF THE SIMULATION CODE

Some standard testing for the code is executed by a validation data set and some validation code, all included in the `GLISSANDO/TEST` subdirectory.

9.1 THE VALIDATION CODE

Validation codes are written to test some specific functionality (modules), they are written to be self consistent and usually does not require any input.

9.2 THE VALIDATION DATA SET

The validation data set includes three simple representative cases:

Test 1 Simple sinusoidal signal
- script file `glissando_test1.pl`
- par-file `glissando_test1.par`
- input dat `glissando_test1.dat`
- output dat `glissando_test1.out`

Test 2 Amplified sinusoidal signal
- script file `glissando_test2.pl`
- par-file `glissando_test2.par`
- input dat `glissando_test2.dat`
- output dat `glissando_test2.out`

Test 3 Damped sinusoidal signal
- script file `glissando_test2.pl`
- par-file `glissando_test2.par`
- input dat `glissando_test2.dat`
- output dat `glissando_test2.out`

The output of the three tests represented in Figure 11 is stored in `GLISSANDO/TESTS/RELEASE`.

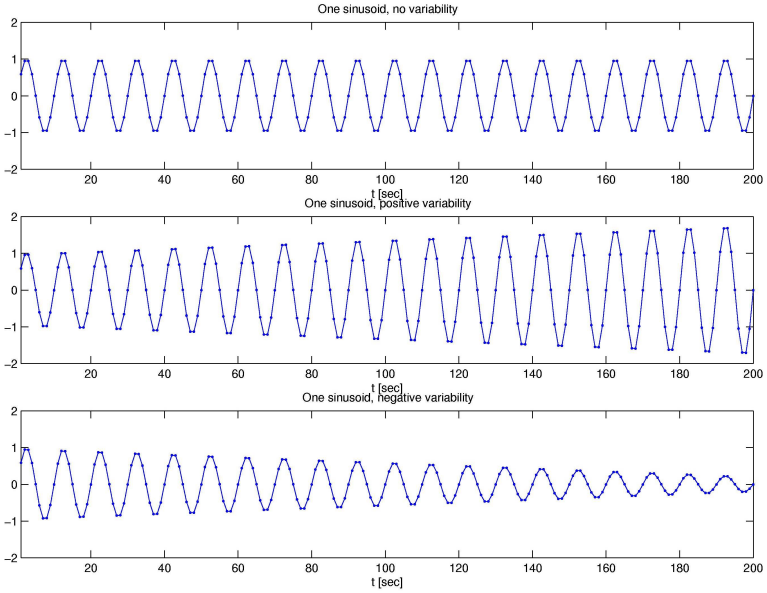

Figure 11 Results for test. From top to bottom: test 1, test 2 and test3.

10 PERFORMANCES

10.1

To be written.

11 INSTALLATION

11.1 INSTALLATION OF THE F90 CODE

11.1.1 REQUIREMENTS

11.1.1.1 ISSUE 1.0

Issue 1.0 is completely self-contained, so no extra libraries needs to be installed, the only extra-language, apart from `FORTRAN 90` compiler, is `PERL 5.0` (or higher release) to run the installation script.

In the case `PERL` is not present, the code may be compiled and linked by hand, module by module, respecting the order of the files listed in the `@LST` variable of the `make.pl` file.

Issue 1.0 have been compiled and linked with the following compilers/OS:

OS	Compiler
Windows 2000	Visual Fortran Professional Edition 5.0
Linux/Red Hat 7.3	pgf90
Linux/Red Hat 7.3	lf95
Linux/SUSE 8.0	lf95

11.1.1.2 ISSUE 1.1

Issue 1.1 needs

+ `PERL 5.0` or higher to run the installation script

The following libraries:

11.1.2 INSTALLATION

11.1.2.1

Unpack the tar file in the directory you want to install GLISSANDO

```
tar –xvf glissando_1_0.tar
```

A directory tree will be created including:

```
GLISSANDO      root directory
     |--- DOC  documentation
     |--- TEST tests
     |--- DAT  data files
     |--- PAR  example parameter files
     |--- SRC  source files
```

enter the `GLISSANDO/SRC` directory and run the `PERL 5.0` script file

```
perl make.pl
```

in the case the current `F90/95` compiler is not `pgf90`, edit the compiling options in the `make.pl` file accordingly.

The executable file

```
glissando_1_0.x
```

is created in the `GLISSANDO/SRC` directory and may be moved in the directory where the binary has to reside.

To run the code with a specific par-file call

```
glissando_1_0.x < par-file_name
```

.

11.1.2.2

12 DOCUMENTED BUGS AND UPGRADES

none!

APPENDIX A THE INGESTION CODE

A.1 THE INGESTION CODE

The ingestion code is a scripting package whose scope is to ingest simulated and real data about the Sorption Cooler, in order to generate a self-consistent set of tables of Fourier coefficients, the related time dependences and the tables of transfer functions used by the simulator.

The choice of a scripting mathematical language for this part of the package has been motivated by the fact that the ingesting code may need heavy changes each time a new data set (with a new data format) is available, and new ideas on how to calculate the set of Fourier coefficients and their time dependence are accepted. At the same time the ingestion code shall be used only if a new data set has to be generated. So a good strategy is to prepare a specialized ingestion code for each new data set. In addition such scripting languages allows for on-line graphical and numerical analysis of ingested data and the related produced data sets.

The choice of `MatLab/Octave` [2] has been motivated by the fact that `Octave` is a good clone of `MatLab`, full compatible for the numerical part, which may be download from network and which is usually already installed in standard `Linux` distributions.

A.2 INSTALLATION OF THE INGESTION CODE

To be written

A.2 USE OF THE INGESTION CODE

To be written

[2] `MatLab` is a proprietary tool distributed by *MathSoft Corporation*. `Octave` is a free software distributed under GPL license (www.octave.org).