SUBJECT: **FTS Pipeline Scientific Validation, Phase 2: Report of Module Testing**

PREPARED BY: **Ed Polehampton, Peter Davis-Imhof, Jean-Paul Baluteau**

DOCUMENT No: **SPIRE-RAL-DOC-003216**

ISSUE: **1.0** **Date:** **23rd February 2009**

APPROVED BY: **Date:**

Distribution

Jean-Paul Baluteau	LAM
Peter Davis-Imhof	Blue Sky Spectroscopy
Ed Polehampton	RAL
Peter Ade	Cardiff University
Trevor Fulton	Blue Sky Spectroscopy
Nanyao Lu	IPAC
David Naylor	University of Lethbridge
Giorgio Savini	University College London
Christian Surace	LAM
Bruce Swinyard	RAL
Dominique Benielli	LAM
Scott Jones	University of Lethbridge
Sarah Leeks	RAL
Chris Pearson	RAL
Tanya Lim	RAL
Matt Griffin	Cardiff University
Michael Pohlen	Cardiff University
Pasquale Panuzzo	CEA

Change Record

ISSUE	DATE	Changes
Draft 1.0	19 February 2009	
Issue 1.0	23 February 2009	Minor corrections from Nanyo & Dominique (Non-linearity corr & clipping sections)

1 INTRODUCTION

1.1 The SPIRE FTS Validation Group

1.1.1 Group Membership

Coordinators:
Ed Polehampton (RAL)
Jean-Paul Baluteau (Marseille)
Peter Davis-Imhof (Blue Sky Spectroscopy)
Members:
Peter Ade (Cardiff)
Trevor Fulton (Blue Sky Spectroscopy)
Nanyao Lu (IPAC)
David Naylor (Lethbridge)
Giorgio Savini (Cardiff)
Bruce Swinyard (RAL)
Christian Surace (Marseille)
Dominique Benielli (Marseille)
Scott Jones (Lethbridge)
Cross-members (coordinating across all 4 groups):
Sarah Leeks (RAL)
Chris Pearson (RAL)

1.1.2 Objectives

The Objectives of the validation group are:

1. To ensure the pipeline conforms to the top-level documentation in terms of their overall architecture and detailed implementation.
2. To ensure that the developer documentation for individual modules conforms to the top-level documentation in terms of requirements and algorithms.
3. To verify that testing carried out at the developer module level is adequate and documented
4. To test the pipeline to validate the correct operation of individual modules and end-to-end systems.
5. To identify and initiate correction of errors or omissions in the pipeline documentation.
6. To identify and report errors in the module implementation and operation.
7. To document all results from the test phases.

The software test of pipeline modules aims to check:

- Consistency with the (already reviewed) top-level documents and module requirements
- Consistency with calibration file definitions (as described in the Pipeline Description Document)
- Correctness and clarity of implementation (i.e. algorithms used are correct and method clear)
- Commonality in use of symbols and terminology (i.e. inputs/outputs to each module use consistent terminology, algorithms use consistent symbols)
- Status of module-level testing (i.e. testing that has been carried out so far)

1.2 Structure of this Document

This document contains a review of all modules of the spectrometer pipeline that were not tested in Phase 1 (except Spatial Regridding). Also and update to the report on 1st level deglitching is included (a preliminary report was contained in the Phase 1 review document). The module testing was carried out by splitting the FTS Validation Group into several sub-teams, divided by institute. Each major module was assigned to two teams to provide independent testing. Other modules were tested by one team only.

1.3 Documents

1.3.1 Applicable Documents

	Deglitching SPIRE Interferograms, version 0.5, December 17, 2008

1.3.2 Reference Documents

User Guide	SPIRE Pipeline User Guide, version 0.02, 18 September 2008
Chris' Document	SPIRE Pipeline Description (SPIRE-RAL-DOC-002437) Issue 2.0, 31 January 2009
Trevor's Document	SPIRE Spectrometer Pipeline Description (SPIRE-BSS-DOC-002966) Issue 1.2, 5 December 2008
Module Requirements	SPIRE Data Processing Pipeline Module Requirements (SPIRE-ICS-DOC-002998) Draft 1.4, 27 August 2008

2 LIST OF MODULES TESTED

In Phase 2, the remaining modules in the pipeline were tested using the 0.6.6 release of the HCSS software. The calibration files and the changes carried out to the top level documents after the documentation review were also checked.

The objectives of the module testing in Phase 2 were:

- Check input data before module
- Check calibration files
- Look at available A/I parameters that can be changed/adjusted and make recommendations
- Ensure module does what it is supposed to do
- Check boundary/extreme conditions
- Validate the output before moving on to the next processing stage

The modules tested in Phase 2 were:

Module Name	**Owner**
Electrical Crosstalk	CEA
Non-linearity Correction	IPAC
Clipping Correction	LAM
Bath Temperature Fluctuation Correction	IPAC
SCAL, Telescope & Beamsplitter Correction	LAM
Level 2 Deglitching	BlueSky
Flux Conversion	BlueSky
Optical Crosstalk	BlueSky/CEA

By the time of the code freeze for version 0.6.6, all modules except Spatial Regridding were included in the pipeline script and ready to be tested.

Major Modules to be tested by 2 groups

Module Name	Group 1	Group 2
Non-linearity Correction	IPAC	Lethbridge
Clipping Correction	Marseille	Cardiff/RAL
Bath Temperature Fluctuation Correction	IPAC	Cardiff/RAL
SCAL, Telescope & Beamsplitter Correction	Marseille	Lethbridge

Minor Modules to be tested by 1 group

Electrical Crosstalk	Cardiff/RAL	
Optical Crosstalk	Cardiff/RAL	
Flux Conversion	IPAC	

To be tested at Blue Sky

Level 2 Deglitching	Blue Sky	

To be tested by Chris Pearson using tests from Scan Map pipeline

Compute BSM Angles	RAL	

3 Software Testing of Modules

For this Scientific Validation campaign, the selected SPIRE DP version 0.6.6.x build is used. This corresponds to DP-SPIRE #242 (New style).

3.1 Electrical Crosstalk

Tested by Giorgio Savini and Ed Polehampton.

The purpose of this module is to correct the timeline detector data for electrical crosstalk using an electrical crosstalk matrix calibration file. Currently this calibration file is filled with zeros except the diagonal which is filled with ones (i.e. no crosstalk).

3.1.1 Input Data

Real data from PFM4 was used as input. The observation used was:

OBSID	Date and Time	Mode	Number Scans	Source details
0x300117FE	8/12/06 18:42-18:50	H	4 High	SCAL4 @ 67.9 K, Laser on SSWD3

In order to process these for input to the module, we used the obsExporter tool to obtain Level-0.5 data from the database in a pool. We ran the level 0.5 detector timeline directly through the electrical crosstalk module.

3.1.2 Test Procedure

We aimed to test the following points:

1. Timeline output unchanged with identity matrix
2. Populate crosstalk matrix and check that timeline is affected in expected way

When carrying out initial tests, we found that the output was not changed at all for any values we put into the calibration file containing the matrix. This was due to a bug in the Java code for the module and has been raised in **SPIRE SPR-1115** (*SciVal-FTS: Elec Crosstalk module does not apply matrix for the Spectrometer*). It was fixed by René Gastaud in CVS and we compiled the updated module and inserted the compiled Java file into the build to be used for this phase of the validation.

In order to determine if the correction had been applied correctly, we summed the difference between the output timeline and the input timeline for each detector. We changed the crosstalk matrix to make one of the non-diagonal elements greater than zero. We did not make the assumption that the matrix should be symmetrical, even if this should be the case.

The crosstalk matrix (*N*x*N*, where *N* is the number of detectors) works in the following way: changing the value of one row in column "SSWA1", indicates crosstalk on SSWA1 coming from the detector in that row. The module multiplies the originating detector timeline (the row) by the value in the matrix and adds it to the final detector (the column).

If the module was applied correctly, this means that there should be a non-zero number for the column detector in the sum of the differences.

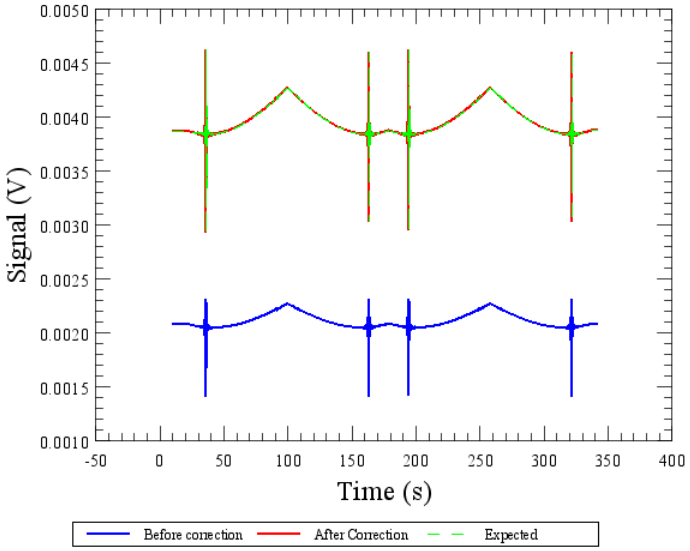

Figure 1: Comparison of the timeline before and after the correction when crosstalk (magnitude 1.0) is applied between detectors SLWA3 and SLWA1. The expected value, calculated by adding SLWA3 to SLWA1 is shown in green – this agrees with the results after applying the module.

3.1.3 Conclusions, Recommendations & Comments

- Bug in Java code found – raised in SPR-1115 which was closed
- After bug was fixed, we confirmed that when the input calibration file contained the identity matrix the output was unchanged
- We confirmed that when a non-diagonal element in the calibration file was added, the factor was applied to the timeline as expected

3.2 Non-Linearity Correction

The SPIRE spectrometer nonlinearity correction module (specNonLin) converts a voltage timeline, *V*, of an input SDT into a "linearized" voltage time line *V'*. The module corrects for the nonlinear dependency of the detector responsivity on voltage so that *V'* will be proportional to the actual detector optical load.

The module does the following simple calculation:

$$V' = K_1*(V - V_0) + K_2 \ln [(V-K_3)/(V_0-K_3)]$$

where V_0, K_1, K_2 and K_3 are provided in the calibration file associated with the module.

The non-linearity module was tested by two groups:

- Nanyao Lu
- David Naylor and Scott Jones

Module tests carried out by Nanyao Lu

In this report, we validate only the science requirements of the module, namely, *V'* is proportional to the input optical load on the detector.

The pipeline was run on a MacBook Pro laptop with Leopard OS and 4GB ram.

3.2.1 Test Data

We use test data generated by the same bolometer module from which the calibration table of the module was derived. Based on the PFM5 detector parameters, the bolometer model is coded in IDL.

We used:

(a) detector bath temperature of 300 mK
(b) bias voltage amplitudes of 30 mV for both SSW and SLW
(c) zero telescope background

It should be pointed out that the test results should remain valid even if (a), (b) or (c) changes.

For each detector channel, we calculated its voltage for each of the following optical loads: 2.5, 5.0, 10, 15, 20 pW. The ASCII file containing these voltages in Volts is "vbolo.data" and has been placed on the TWiki page:

http://www.herschel.be/twiki/bin/view/Spire/FTSPipelineGroup

3.2.2 Calibration Table

We use a calibration table that was based on the same detector model and configuration of (a) - (c) above. In this way, there is guaranteed consistency between the calibration table and simulated voltage readings. This is a crucial requirement for this validation report. Both the ASCII version (file: spec_response.txt) and FITS version (file: SpecNonLinCorr.fits) of this calibration table are available on the TWiki:

http://www.herschel.be/twiki/bin/view/Spire/FTSPipelineGroup

This calibration table is based on a fit of the function $K_1 + K_2/(V-K_3)$ to the model calculated responsivity over a range of optical load from 0 to 20pW. Figure 2 shows how well a curve based on the parameters K_1, K_2 and K_3 fit the model calculated responsivity.

It should be pointed out that a range of 0 to 20pW is much wider than the range expected in flight. So the function fit to the data could be improved if a smaller power range is used. Also, SSW_D5 is a dead channel. SSW_F4 has a failed function fit to its model response curve.

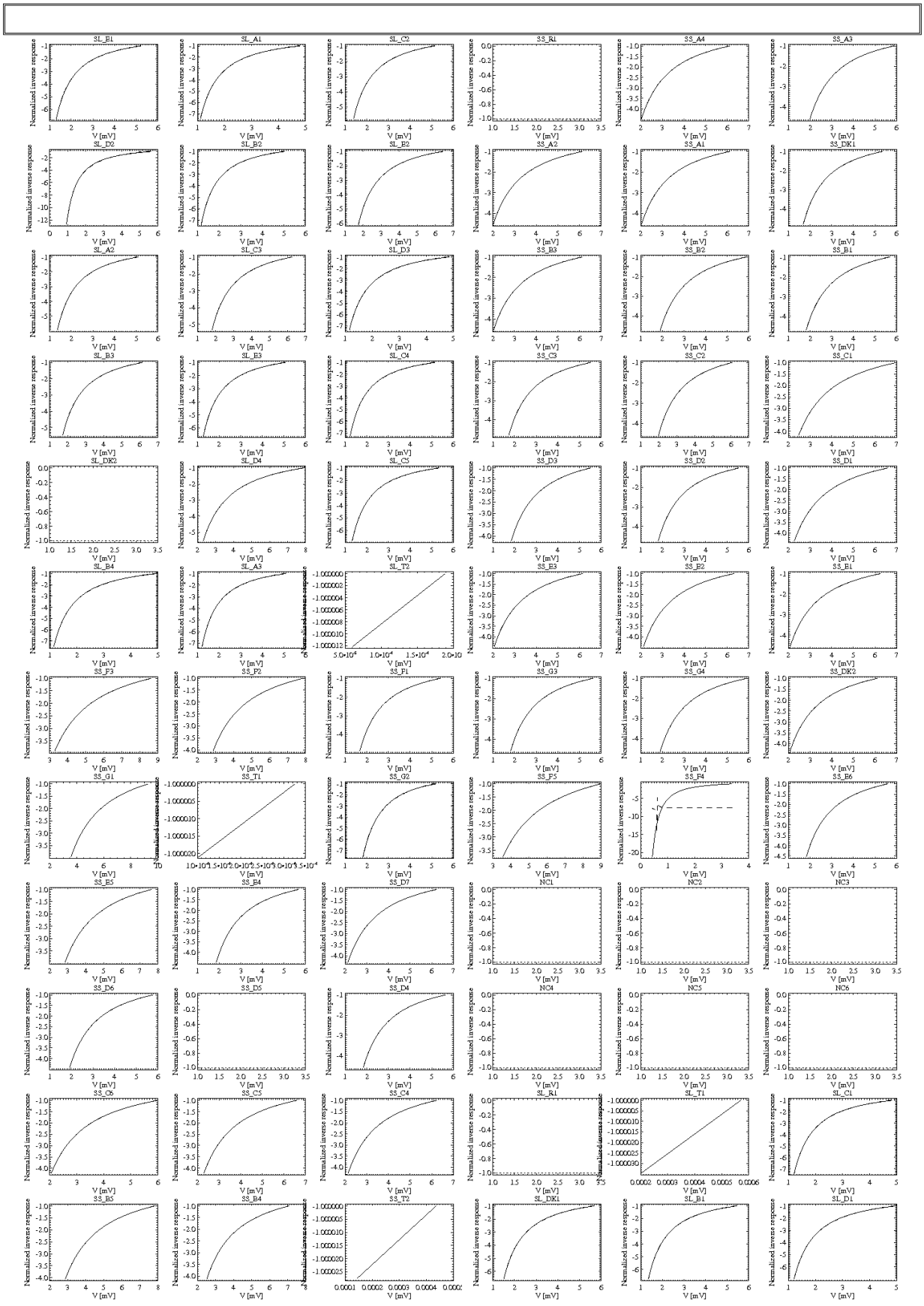

Figure 2: Plot of the functional fits to the model response curves for each detector.

3.2.3 Validation Criteria

We have only one science requirement to test:

Degree of the Proportionality of *V'* to Optical Load.

For a given detector channel, let

Q = Array of optical loads on the detector
V = Array of the corresponding bolometer voltage readings
V' = Array of the voltages output from the specNonLin module

The module passes the science requirement if *V'* is proportional to *Q* to within an uncertainty of *E*, set by the uncertainty of the function fit in the calibration table. As evident in the spectral response calibration table, *E* < 1% for all SSW channels except for SSWG2, SSWD5 and SSWF4. These 3 channels are either noisy or dead (SSWD5). For the SLW channels, *E* could be up to 3-4% except for the noisy channel SLWC2.

3.2.4 Validation Process.

We use the Jython script, bolo_nonlin_testing.py, which is available on the TWiki. It contains the following 3 key steps:

(a) Import the voltage table "vbolo.data" into the voltage time line of a SPIRE spectrometer SDT

(b) Feed this SDT through the module "specNonLin"

(c) For any given detector channel, compare the output voltage time line from "specNonLin" with the following optical load data array [0.25, 5.0, 10.0, 15.0, 20.0] in pW

3.2.5 Validation Results

Figure 3 shows how the module input and output voltages vary as a function of the optical load *Q*. Clearly, the relation between the raw input voltages and *Q* is highly non linear. In contrast, the output voltages appear to be linearly correlated with *Q*. The same conclusion can be drawn on other channels, such as on SLWC2 (see the Fig. 3 right).

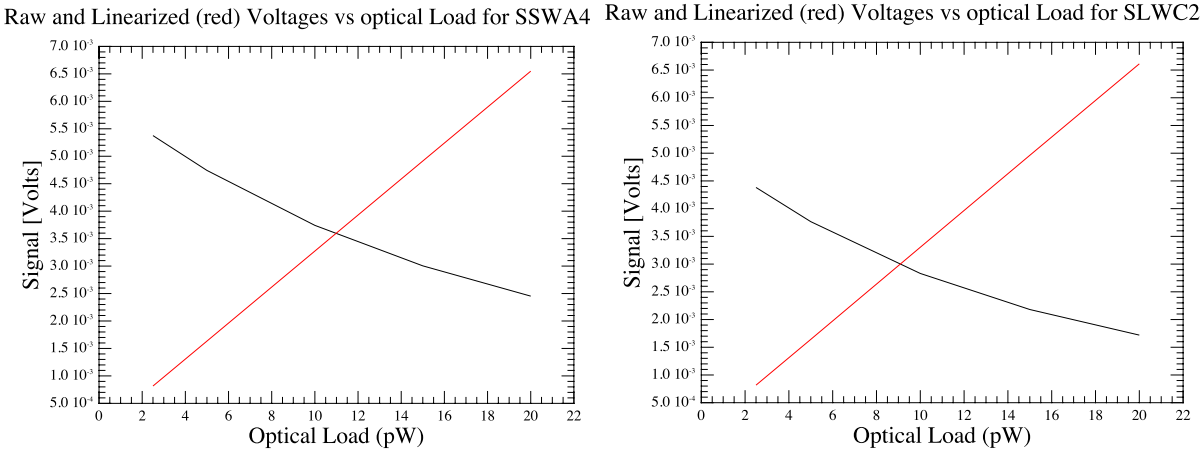

Figure 3: Plot of the raw bolometer voltages (in black) and linearized voltages (in red) as a function of the corresponding optical loads for the SSW channel A4 (left) and the SLW channel C2 (right).

Figure 4 (top left) shows the channel ratios of the voltages corresponding to the optical loads of 20 and 2.5pW. The black curve is using the raw voltages, and the red curves using the voltages output from the specNonLin module. If the module works as required, the red curve should be at a value of 20pW/2.5pW = 8. This is indeed what is observed here. Similarly, Fig. 4 (top right) shows the case with optical loads of 10 and 5pW. This range in optical load represents the real operational conditions of SPIRE in flight. Here the red curve remains at its expected value of 2.

We can examine more closely the observed deviations from their expected model values. The voltage ratios in the top two plots in Figure 4 were normalized by their respective expected values and the results are plotted (in red) as a function of detector channel index in the bottom two plots in Figure 4. Also shown in these plots are the upper and lower limits based on the uncertainties from our calibration fit to the bolometer model responsivity. It is evident that the normalized voltages are all within the calibration uncertainties.

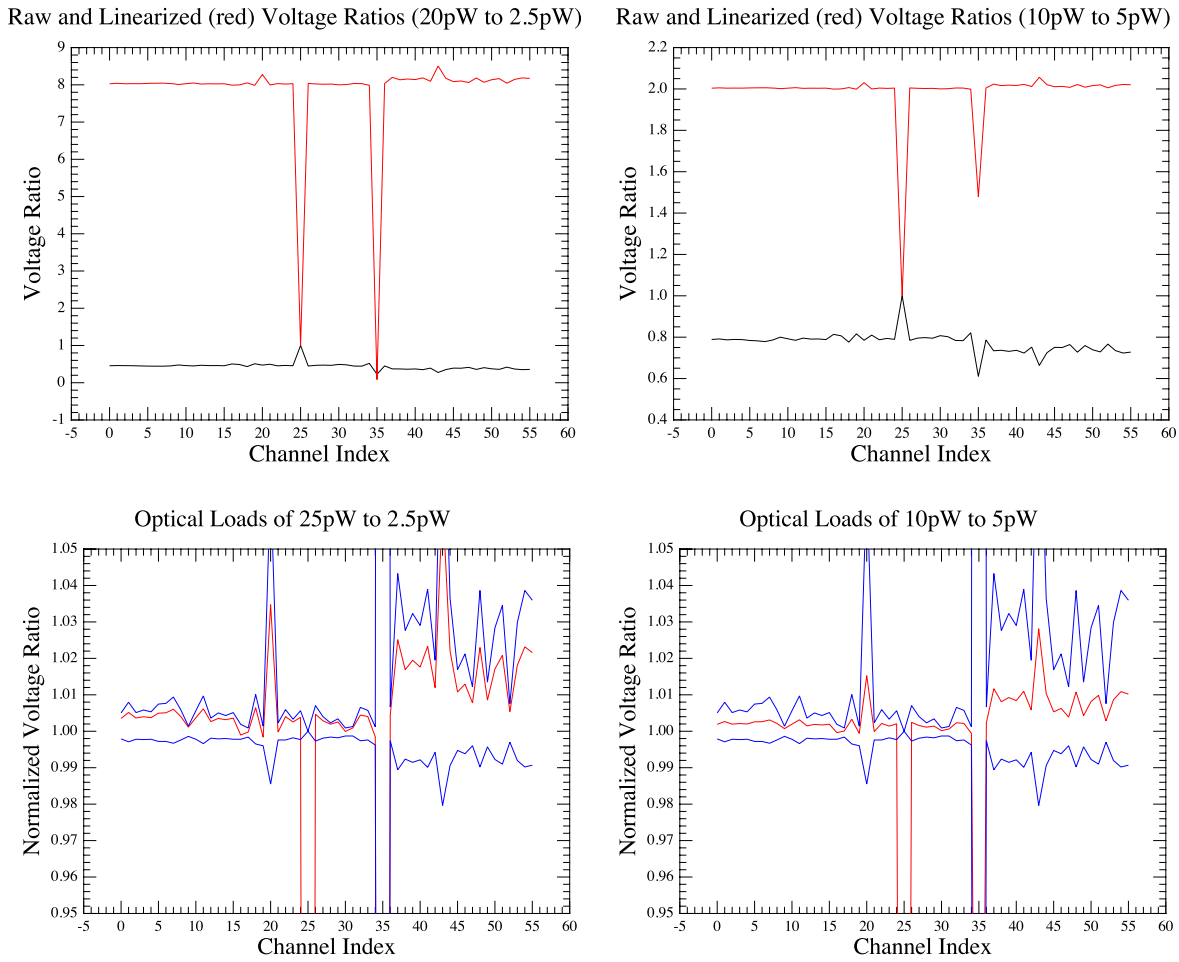

Figure 4: Top: Plot of channel voltage ratios using either bolometer raw voltages (in black) or the linearized voltages (in red). The left hand plot is for the two optical loads of 20 and 2.5pW, and the right hand plot is for the two optical loads of 10 and 5pW.
Bottom: Plots of the linearized voltage ratios normalized by their model expected values (in red) as a function of detector channel index (using the voltage ratios from top two plots). Also shown (in blue) are the expected upper and lower limits based on our calibration uncertainties.

Module tests carried out by David Naylor and Scott Jones

3.2.6 Input Data

The input data used was a SMEC scan from the PFM4 campaign (not an AOT observation).

OBSID	Date and Time	Mode	Number Scans	Source details
0x300114CE	27/11/06 18:53-19:17	H	16 High	SCAL off (~4.5 K) and CBB @ 9K

3.2.7 Calibration Data

As the detector settings in this observation were different from the ones used to derive the calibration table that is currently implemented in the build, Nanyao Lu derived a new calibration product to match observation 0x300114CE. This is available on the TWiki and used the following parameters:

- Bath temperature = 300mK
- V_{bias} = 0.031 V (peak amplitude)
- Background, Q_0 = 30 pW

This is a high flux case corresponding to the lab conditions. The actual telescope background is on the order of 4-5 pW. The background is important because the nonlinearity module outputs a linearized voltage relative to a reference voltage V_0, which is set to correspond to Q_0 in the current pipeline implementation. Mathematically, it is given by the K parameters proposed in Matt's pipeline document:

$$V_{linearized} = \int_{V_0}^{V} K_1 + \frac{K_2}{V - K_3} dV$$

where V is the voltage reading, and $V_{linearized}$ is the output voltage from the module.

The effect of V_0 is removed at the pipeline stage when a similarly reduced SCAL interferogram is subtracted out. However, note that this step was not performed in the testing carried out here.

The calibration of these K parameters would be done over a voltage range of V_{min} to V_{max}. K_3 is usually still significantly smaller than V_{min}. Normally, observational voltage readings are all > V_{min}, and we always have $V > K_3$. If something went wrong, and it caused $V < K_3$, the module would have to (i) label such samples as unusable in the output SDT, and (ii) generate a QC flag.

3.2.8 Test Procedure

To examine the non-linear correction module of the SPIRE pipeline, as it pertains to spectroscopy, we compared the results of its application using interferogram data that had been analyzed off line with our IDL pipeline code, and which clearly exhibited nonlinearity (see "*Non-linear Behaviour of Bolometric Detectors in Fourier Spectroscopy*", David A. Naylor, Brad G. Gom, Scott C. Jones, Locke D. Spencer for a summary of non-linear effects in spectrum).

Initial tests showed that it was very important to

(a) have the correct detector offsets added to the test data – the correction works on the absolute measured voltage values
(b) have a calibration table containing parameters derived with the correct detector settings and assumptions about the background power

When either of these criteria were not met, the results from the module showed spurious spikes and features.

3.2.9 Conclusions, Recommendations & Comments

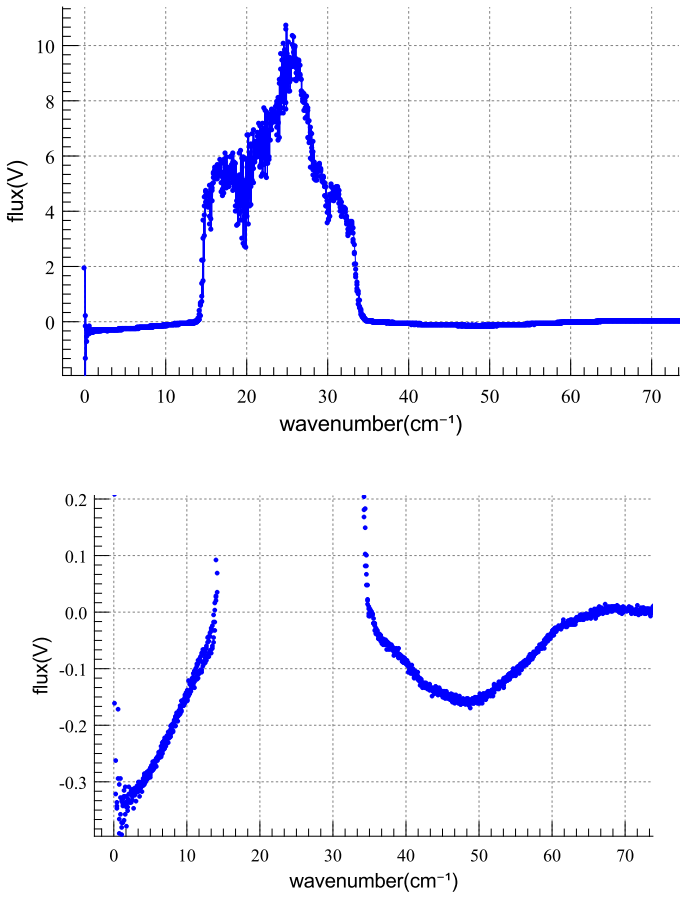

Figure 5: Spectrum (and zoom) resulting from pipeline without Non-Linearity correction.

The top plot of Figure 5 shows the full spectrum derived from the SPIRE pipeline, uncorrected for non-linearity, for central pixel, SLWC3, for observation 0x300114CE. The only pre-Fourier transform modules activated were "deglitch", "timeDomainPhaseCorrection", and "createIfgm" for these spectra. The resulting spectra are identical to those determined with our IDL pipeline code.

Results for uncorrected case:

Integrated area, fundamental band	109.09199
Integrated area, 0th harmonic	-2.9198
Ratio of 0th harmonic to fundamental band	-0.026765
Integrated area, 2nd harmonic	-2.76545
Ratio of 2nd harmonic to fundamental band	-0.025350

Key point: the integrated power in the 0^{th} and 2^{nd} harmonic bands is very similar as theory predicts (but 1/f noise would tend to effect)

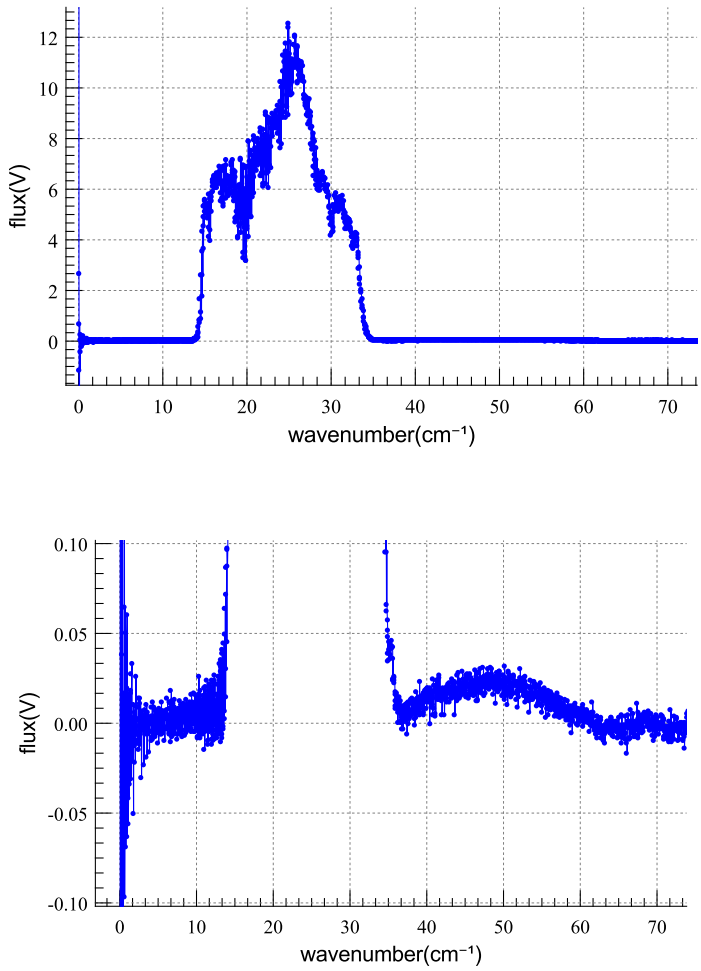

Figure 6: Spectrum (and zoom) resulting from pipeline including Non-Linearity correction.

The top plot of Figure 6 shows the full spectrum derived from the SPIRE pipeline, now using the non-linearity correction module, for central pixel, SLWC3, for observation 0x300114CE. In addition to the pre-Fourier transform modules discussed above, the module "specNonLin" was activated.

Results for corrected case:

Integrated area, fundamental band	127.45429
Integrated area, 0th harmonic	0.0347154
Ratio of 0th harmonic to fundamental band	0.00027238
Integrated area, 2nd harmonic, corrected	0.365095
Ratio of 2nd harmonic to fundamental band	0.00286452

Key points:

1. The non-linear correction is clearly working if the detector voltages are calculated correctly which means taking full account of the detector offset history. The engineering conversion module should raise a quality control flag if the offset history shows any signs of ambiguity since the effects of non-linearity correction on the resulting timelines can be catastrophic.
2. The integrated power in the 0^{th} and 2^{nd} harmonic bands is much smaller, by one order of magnitude. The correction appears to be doing a better job in the lower frequency band and appears to have over-corrected the 2^{nd} harmonic region. It should be explored if the reduction

of the power in the 0th and 2nd harmonic can be used as a quality control metric for the non-linearity correction module.

3. Although the integrated area under the uncorrected harmonic terms is around 3%, applying the correction increases the amplitude in the band by close to 20%. Non-linearity has a large effect on these data.

In addition, the science validation team recommends to

- Clarify in the description of this data processing module given in both Trevor and Chris' documents that this module not only linearizes the relationship between the signal and the optical load but also makes the slope positive.
- Verify that the tools are available to monitor the thermistor temperatures and the sub-K temperature sensors to verify that the fundamental assumptions behind the non-linearity correction module hold during flight.

3.3 Clipping Correction

The purpose of this module is to correct the detector timeline data for “clipped” samples. These occur where the readout reaches the limits of the available ADC dynamic range. These clipped samples are problematic as they represent corrupted samples in the timeline. If left uncorrected, these samples can lead to further complications in particular when the timelines are converted into interferograms.

The module makes use of the "truncated" data flag, and without using any of the clipped points performs a polynomial fit (8^{th} order) to restore the interferogram.

The dynamic range of the ADC runs between 0 and 65535 (65K). Samples are clipped when the signal takes values of 0, (cf Fig 7a) or 65K (see Fig 7b).

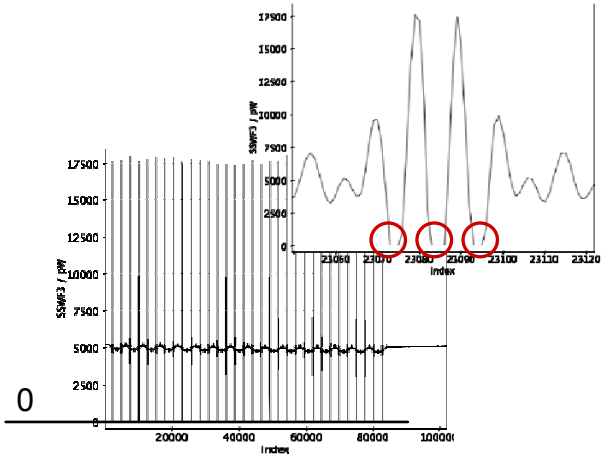

Figure 7 a: Example of clipping to 0 (SDT in PFM5), The zoom shows the interferogram part where clipping at 0 occurs: clipping is present in the central burst and in the two secondary fringes.

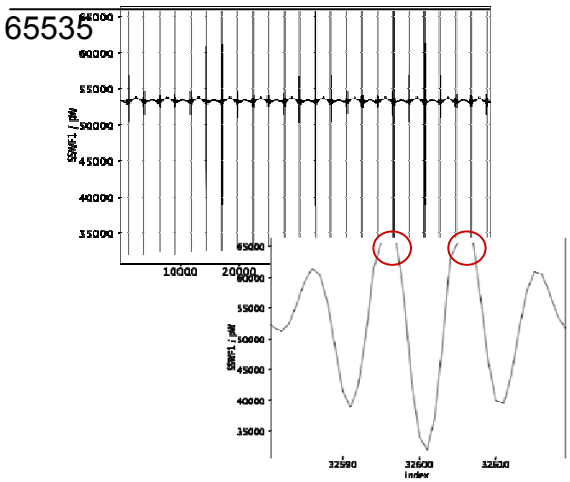

Figure 7 b: Example of clipping to 65k : The zoom shows the interferogram part where clipping at 65k occurs: clipping is present in the central burst and in the two secondary fringes.

The clipping module was tested by two groups:

- Dominique Benielli, Jean-Paul Baluteau and Christian Surace
- Giorgio Savini and Ed Polehampton

Module tests carried out by Dominique Benielli, Jean-Paul Baluteau and Christian Surace

3.3.1 Statistical Analysis of PFM5 tests data : number of clipped points

All of the PFM5 data has been analyzed. Table 1 gives the total number of measurements, the number of clipped samples, and the percentage of clipped measurements for each run. However, clipping occurs in the most important part of signal (i.e. around ZPD), indeed the maximum (or minimum) at ZPD corresponds to the integral of total flux of the corresponding spectrum integral. Therefore part of information concerning the interferogram disappears in the case of clipping.

	Total number of samples	Number of clipped samples	Clipped number %
ILT_PERF_SMC_300124ED_SPECF.fits	555120	2116	0.38%
ILT_PERF_SMC_300124EF_SPECF.fits	482616	180	3.73%
ILT_PERF_SMC_300124F4_SPECF.fits	482832	389	8.06%
ILT_PERF_SMC_300124F9_SPECF.fits	4442400	3639	8.19%
ILT_PERF_SMC_3001248B_SPECF.fits	7338600	7	0.01%
ILT_PERF_SMC_30012495_SPECF.fits	7339104	0	0.00%
ILT_PERF_SMC_30012497_SPECF.fits	7339104	1128	1.54%
ILT_PERF_SMC_30012568_SPECF.fits	2132640	0	0.00%
Total	**30112416**	**7459**	**0.02%**

Table 1: Analysis of PFM5 test data

3.3.2 Statistical Analysis of PFM5 tests data : number of clipped "areas"

The statistical results of the PFM5 30012497 file, where 0.02 % of samples were clipped, are presented in Table 2.

Clipped samples were found in data from several different SSW and SLW detectors, with clipping at 0 or 65 K, with 1 to 3 or more consecutive clipped samples (defined as a clipped "area"), and 1 to 5 clipped areas within the same SDT.

Detector Name	Total number of clipped samples	clipped value	number of clipped areas per scan	max number of consecutive clipped samples per scan
SSWE1	61	65K	1-3	2
SSWE2	58	0	1-2	2
SSWF3	326	0	3-4	3
SSWG2	301	0	2-4	3
SLWD2	34	0	1-2	1
SLWD4	149	0	4-5	2
SLWE1	96	0	2-4	2
SLWE2	103	65K	3-4	1

Table 2: Results from PFM5 30012497

3.3.3 Implementation of the Clipping package

In «herschel/spire/ia/pipeline/spec/» the clipping package has been implemented, with two classes, the ClippingTask itself and another intermediate class which compiles clipping areas and does the detection of each clipped area (with consecutive clipped measurements).

The ClippingTask extends the class, Task. The task takes two input parameters:

- a mandatory parameter «input» which is an SDT product, and
- an optional parameter «number» whose value has been defined and optimized (see Fig. 10; default value fixed to 5 after optimization).

The parameter «number» is equal to half the number of samples which are used in the fit to restore the clipped measurements. The reconstruction of the signal is performed for each clipped area: a **polynomial function of 8th order** is fitted using un-clipped samples using 5 (or the value entered for «number r») samples surrounding the clipped area (5 before, 5 after).

An output parameter named «result» which is an SDT product contains the restored corrected SDT.

3.3.4 Reconstruction tests simulations for Clipping

To study the quality of the reconstruction of the clipping task some simulations were carried out, using the following procedure:

- Take each good detector (i.e. no clipped samples) from the PFM5 30012495 run (i.e. 72-28=44 detectors), ignoring the bad (not used) detectors (with clipped measurement or only noise): bad_detectors are ={"SSWR1", "SSWDP1","SSWDP2", "SSWT1", "SSWE2", "SSWE1", "SSWF3", "SSWG2","SSWD5", "SSWT2", "SSWN1","SSWN2", "SSWN3", "SSWN4", "SSWN5", "SSWN6", "SLWR1", "SLWA3", "SLWT1", "SLWDP1", "SLWE1","SLWE2", "SLWB3","SLWC2", "SLWD2", "SLWDP2", "SLWD4", "SLWT2}
- A "clipped" interferogram is derived from the good detector data: where any measurement exceeds a clipping value, it is replaced by that clipping value
- A "restored" interferogram is computed from the clipped interferogram using clippingTask reconstruction.
- The SDT mask is flagged as TRUNCATED_UNCORR if the reconstruction does not perform (fail of polynomial fitting, or more than 9 consecutive clipped samples, or clipped samples at the beginning or the end of file)

3.3.5 Derived clipped interferogram

In the following we define the clipping value as the amplitude of the signal left over compared to the amplitude at ZPD, clipping of 20% means that the signal is cut out at a value of the amplitude at ZPD*0.2.
The derived clipped interferogram for clipping values from 20 % to 80% of the amplitude at ZPD is analyzed. As can be seen in Fig. 8a, the mean number of clipped samples per scan decreases with increasing clipping value.

The mean number of clipped samples per scan versus the percentage of clipping is presented in several cases:

- for SLW detectors clipping at the top of the signal (the central ZPD burst)
- for SSW detectors clipping at top of the signal secondary maxima
- for SLW detectors with clipping at the bottom
- for SSW detector with clipping in the bottom

The mean number of clipped samples per scan goes from a few samples to around 20 samples when the clipping value decreases from 80 % to 20% of the amplitude at ZPD.

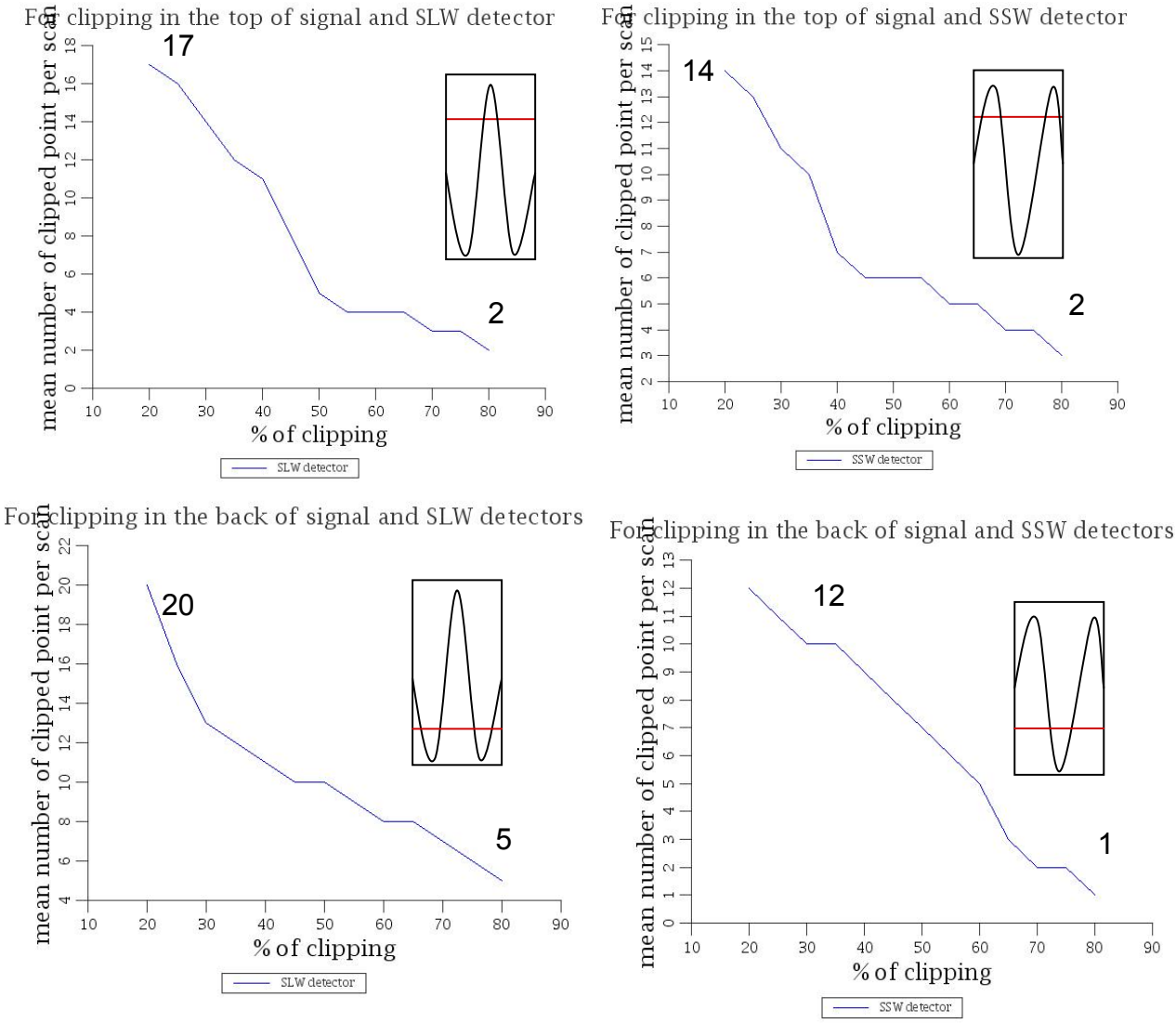

Figure 8a: Mean number of clipped point per scan vs clipping value from 20% to 80% of amplitude at ZPD.

3.3.6 Interferogram Reconstruction

Examples of clipping reconstruction with the clippingTask reconstruction are presented in Fig. 8b for several cases with the clipping value set to 20 %. Top: In the left panel the SLW detector presents five clipped areas, and the SSW detector (in the right panel) presents four clipped areas. Bottom: the left panel for the SLW detector shows four clipped areas, and the right panel for the SSW detector shows 3 clipped areas. We can see that the reconstruction is often **under estimated**.

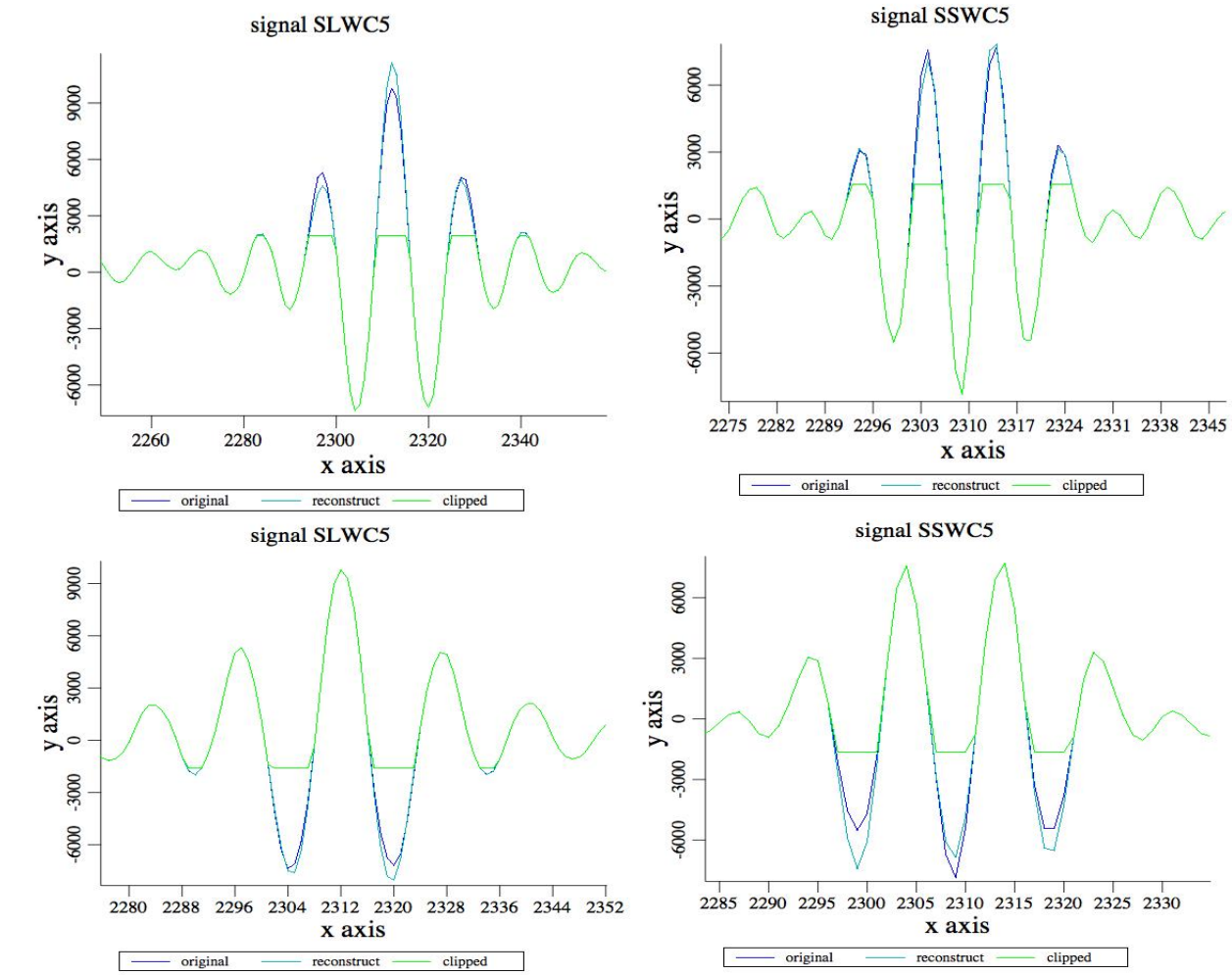

Figure 8b: Reconstruction of clipped measurements with a polynomial fit (8th order; number of fitting samples 2 * 5); clipping value is 20%, In green color the clipped signal, in dark blue the original signal and in light blue the restored signal.

3.3.7 Error on restored signal

3.3.7.1 Mean RMS error

The average (for all SLW or SSW detectors) of the RMS error of the restored measurements vs the percentage of clipping is shown in Fig. 9. The error obtained (i.e. the difference between restored and original measurements over the central ZPD burst amplitude) decreases when the percentage of clipping increases from 1-1.5 % to 4-6%.
Because the error is only calculated on the restored samples, the curve shown in the figure is not regular. So a few percent uncertainties in the restored measurements were found.

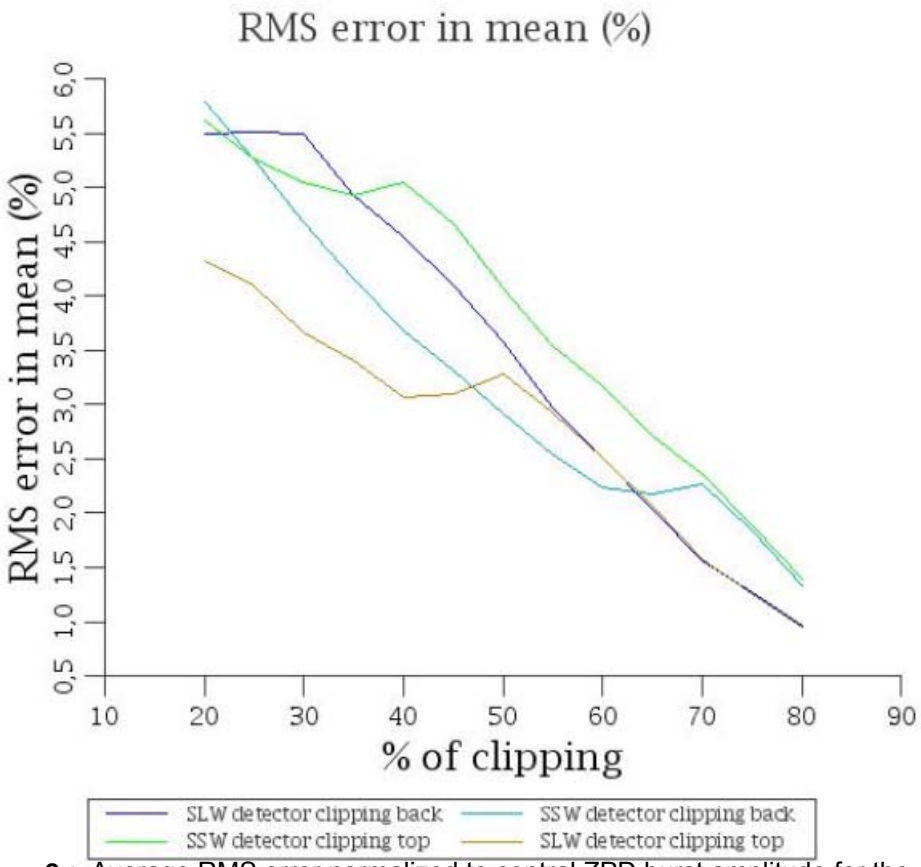

Figure 9 : Average RMS error normalized to central ZPD burst amplitude for the four situations described in Figure 8.

3.3.7.2 Result for RMS error with different 2nd parameter values

For different values of the second task parameter («number») of 4, 5 and 6, the average RMS error of restored measurements vs clipping values is shown in Fig 10. The error obtained is smaller for «number» equal to 5, so the value of 5 for the «number» parameter leads to the better optimization. The «number» value is now fixed to 5 for all following studies.

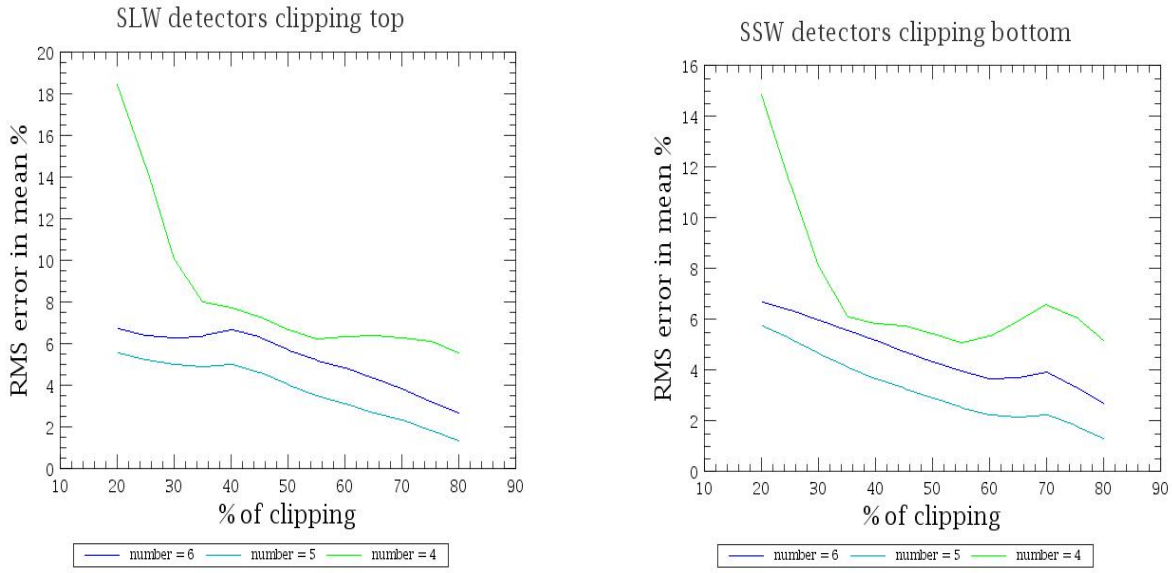

Figure 10 : Normalized RMS error in clipped area (SLW top clipping and SSW bottom clipping) for different number values.

3.3.7.3 RMS error on central ZPD clipped area only

The RMS error is computed only for the central ZPD clipped area (SLW top clipping and SSW bottom clipping). These clipped data are the most important ones to restore since the central ZPD area provides the total spectral power. In this central ZPD area the derived error (Fig. 11a) is found to be slightly larger than the error derived for the overall scan (Fig. 9).

The mean of the difference between original and reconstructed signals represents the systematic part of the error. This mean systematic difference (normalized to the central ZPD burst amplitude) is shown in Fig. 11b, and is found to be less than ~1% for SSW detectors and for clipping value higher than 40% for SLW detectors. Indeed, Figure 11b shows that the restored measurements are generally underestimated compared to original ones (negative difference).

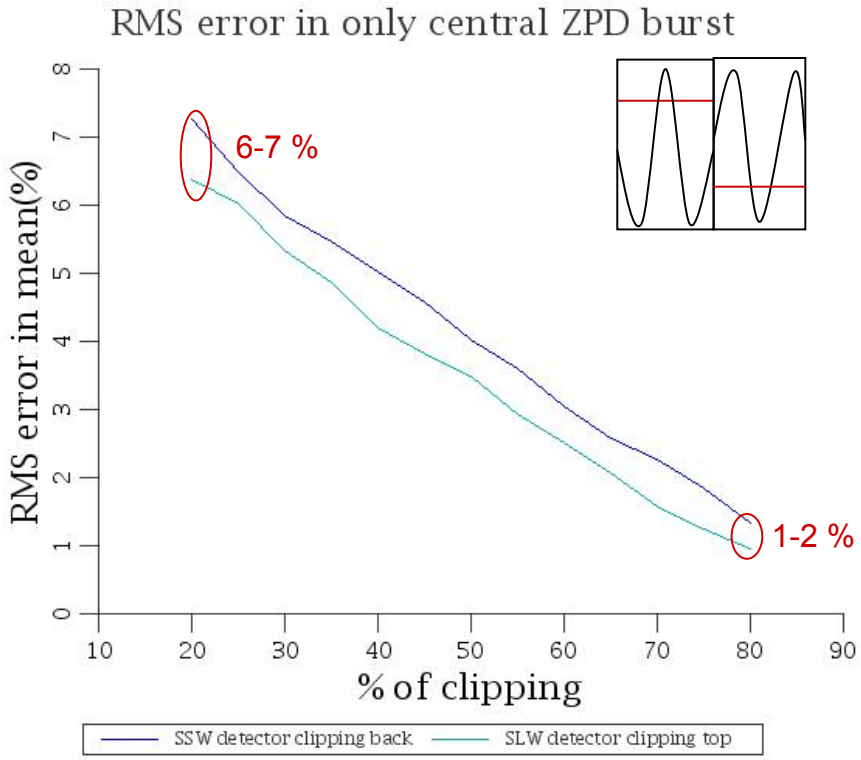

Figure 11 a: Normalized RMS error for the central ZPD clipped area only (SLW clipping high and SSW clipping low).

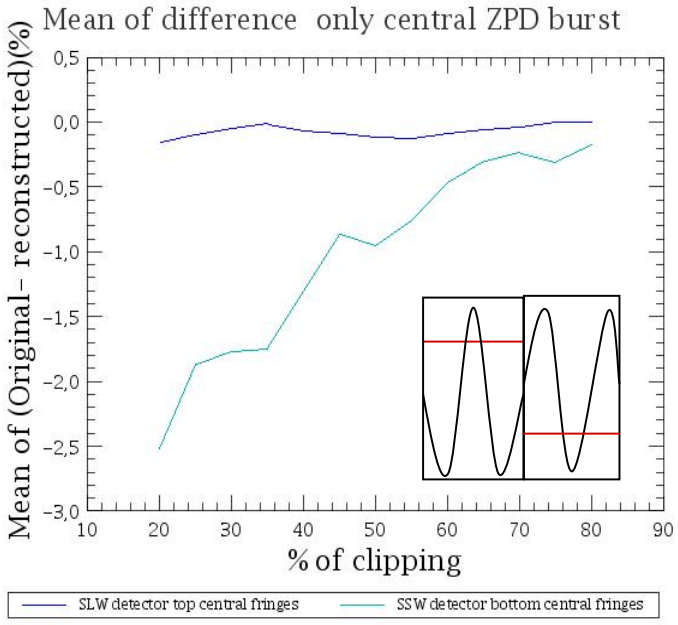

Figure 11 b: Systematic error: Mean of the difference for SSW detectors (clipped at bottom) and SLW detectors (clipped at top) only for central burst vs clipping value.

3.3.8 Difference between original spectrum and restored one

The error on the signal timelines seems to be quite important in certain cases (7% in the worst case) but it is also necessary to look at the impact of the clipping task on the final spectrum which is the end product of the whole pipeline.

In Fig. 12, in the spectrum significant part, there is no visible difference between the spectrum obtained from the original signal and from the spectrum obtained with the restored signal (for the two cases: SLW top clipped and SSW bottom clipped signals).

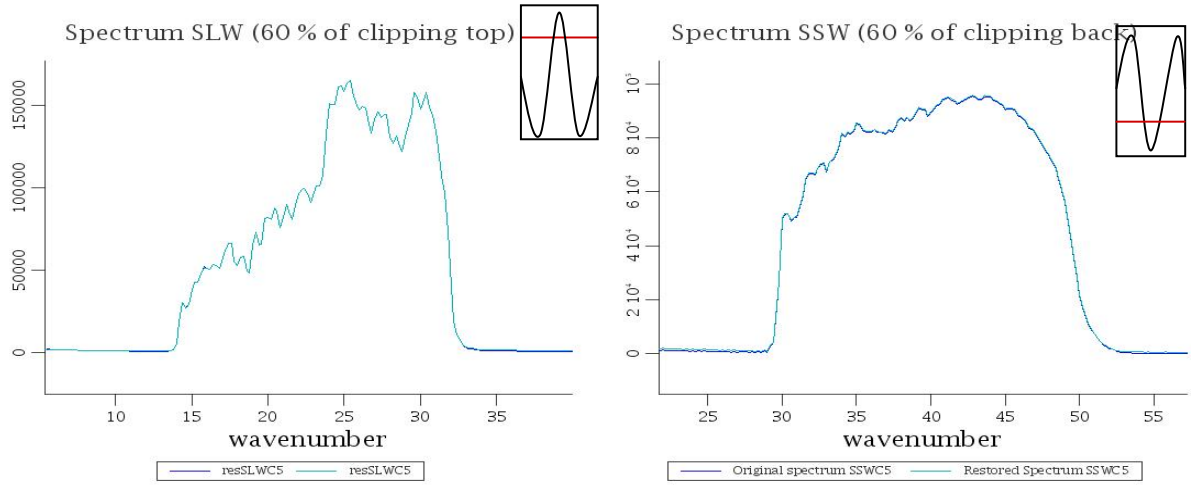

Figure 12 : Original spectrum (dark blue) and restored one (light blue) for SLWC5 (left) and SSWC5 (right), for a clipping value of 60%.

3.3.9 Spectral error and noise

The spectral error (difference between original spectrum and restored one) is plotted on Fig 13a for an SSW detector and on Fig. 13c for an SLW detector. This error on the spectrum for clipping values equal to 60% is approximately 0.4 % and 0.1% for SSW and SLW respectively (whereas the RMS error was about 2-3% for the interferogram signal). In all cases the error is lower than 1%.

To evaluate the eventual noise increase due to the reconstruction, the standard deviation is calculated. The spectral RMS noise error (individual spectrum minus mean spectrum) against scan number is computed (Fig. 13b and 13d) for both original and reconstructed spectra. The two curves for the SLW detector (Fig. 13d) are not significantly different, so there is no significant increase of noise between the original spectrum and the restored one. In the same way for the SSW detector (Fig. 13b) there is no modification of the noise (same curve of RMS noise for original signal and restored ones).

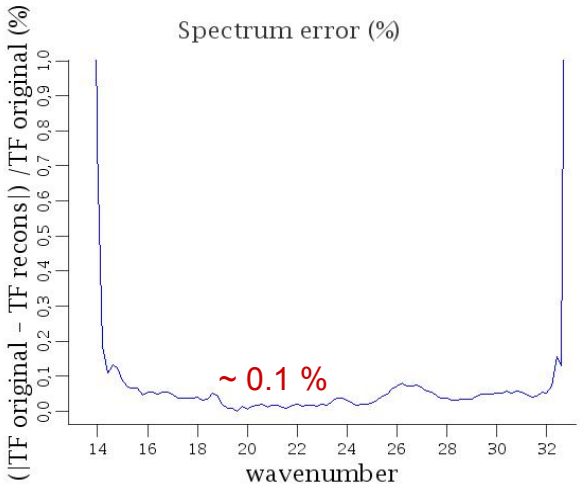

Figure 13 c : Spectral error for SLWC5 for a clipping value of 60%.

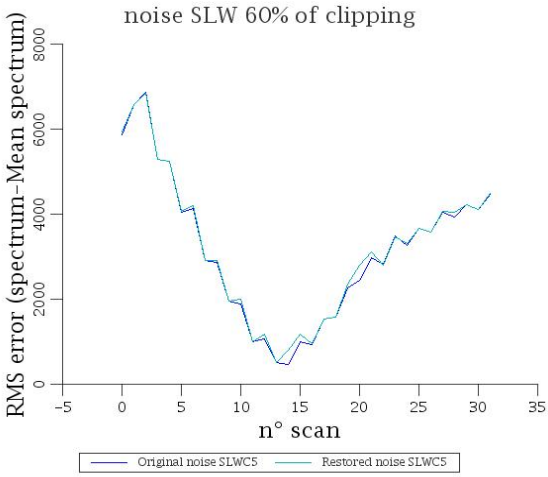

Figure 13 d: RMS noise for SLWC5 in dark blue for original signal and in light blue for restored signal (clipping value 60%).

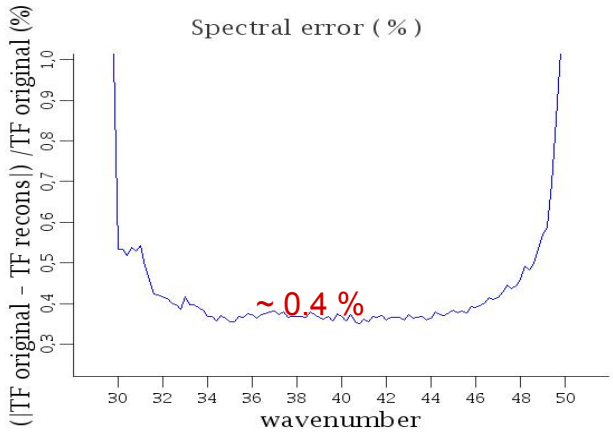

Figure 13 a : Spectral error for SSWC5 for a clipping value of 60%.

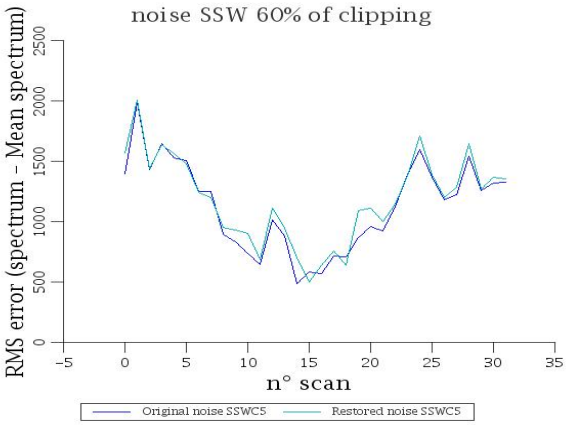

Figure 13 b : RMS noise for SSWC5 in dark blue for original signal and in light blue for restored signal (clipping value 60%).

The large variation of the noise on spectrum (Fig. 13d) with time (scan number) seems to be due to a different effect (detector bath temperature?) and this effect provides the main error in the measured spectra.

3.3.10 Conclusion

- The Clipping task detects clipped measurements and restores corrupted measurements
- Clipping Task sets the mask TRUNCATED_UNCORR to False if the reconstruction is successful
- RMS error on restored spectra is found to be less than 1%

- Tests on PFM5 have shown less than 0.1% clipped samples on all detectors
- The task can reconstructed the signal to a clipping value of 20% with a RMS error of 7%
- RMS error of restored interferometric signal is about 2% to 6-7% depending on % of clipping
- The 2nd parameter is optimized at the value of 5 for PFM5 testing conditions

This analysis has been performed using PFM5 data without correcting for detector Non-Linearity. This analysis should be repeated including the Non-Linearity Correction Task, to check that our conclusions are still valid.

Module tests carried out by Giorgio Savini and Ed Polehampton

3.3.11 Input Data

Real data from PFM4 was used as input. The observation used was:

OBSID	Date and Time	Mode	Number Scans	Source details
0x300117FE	8/12/06 18:42-18:50	H	4 High	SCAL4 @ 67.9 K, Laser on SSWD3

In order to process these for input to the module, we used the obsExporter tool to obtain Level-0.5 data from the database in a pool. We ran the level 0.5 detector timeline directly through the clipping module.

The data were saved before and after the clipping module and then loaded in an IDL for analysis.

3.3.12 Test Procedure

We aimed to test the following points:

1. Verify that unclipped timelines are left unchanged.
2. Check that clipped data in the ZPD range was seemingly restored (both at the top (ZPD) and bottom (sidelobes) and for multiple occurrences of clipping (ceiling close to the baseline).
    a. By comparing interferograms from neighbouring pixels with no clipping
    b. By comparing corrected interferograms in the same detector-scan.
    c. By comparing spectra obtained trasnforming different interferograms of the same pixel-scan.
3. Acknowledge the effect of clipped data in the High ZPD range.

3.3.13 Results

The first test was carried out and the result was as expected that no modification to the timeline was performed.

For the second test we observed the following:

- We found detector timelines with the baseline relatively close to the ceiling or floor of the readout. These presented multiple areas of clipping with different number of subsequent clipped points. In the corrected interferograms these seemed to be restored as in Figure 14 for detector SLWA1.
- The four uncorrected interferograms are visible in green overplotted on the corrected interferograms (black) after the clipping correction module. An imbalance is visible on alternate sidelobes in the reconstructed data. The imbalance is not always periodic (i.e. scan 1 similar to scan 3 and scan 2 to scan 4, as in the case of phase error associated to scan direction).
- Similarly in Figure 15, which shows the results for SLWC5, the same thing presents itself, this time at the floor and the difference in reconstruction is manifest in different heights of the central maximum.
- Subsequent Time Domain Phase Correction does not correct this imbalance. More specifically it applies an almost identical correction to all four interferograms (see Fig.14b and 15b).

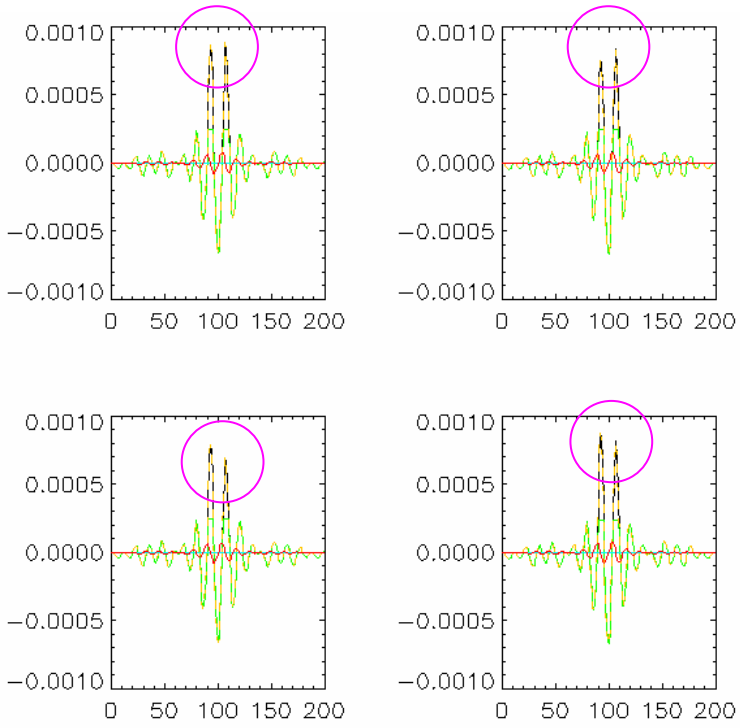

Figure 14a: The reconstruction of the interferogram for the 4 scans has visible asymmetry in the sidelobes which is different in different scans.

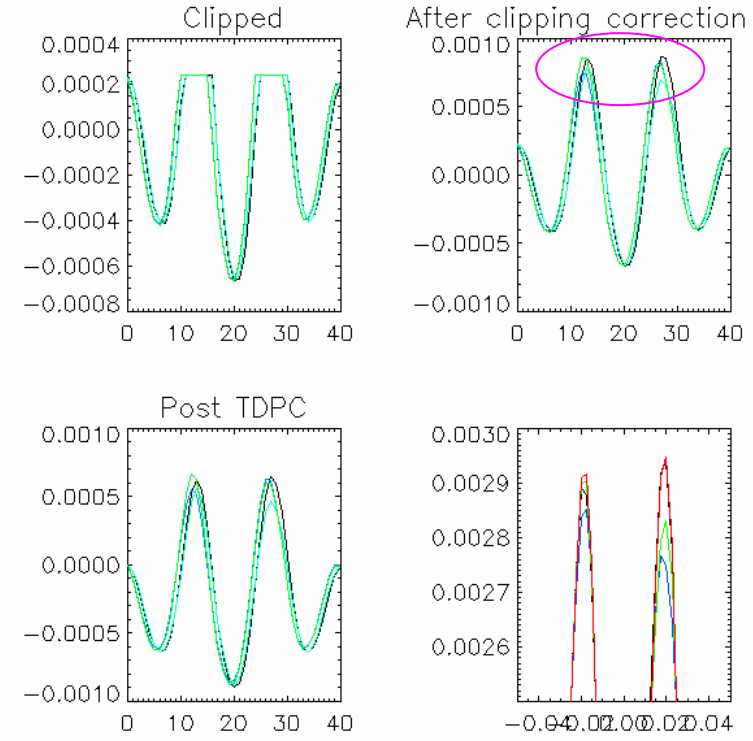

Figure 14b: The top left and right plots show the 4 superimposed interferograms before and after clipping correction. Lower right is the post-TDPC (time domain phase correction) interferogram to show that this does not correct for the (possibly) introduced imbalance.

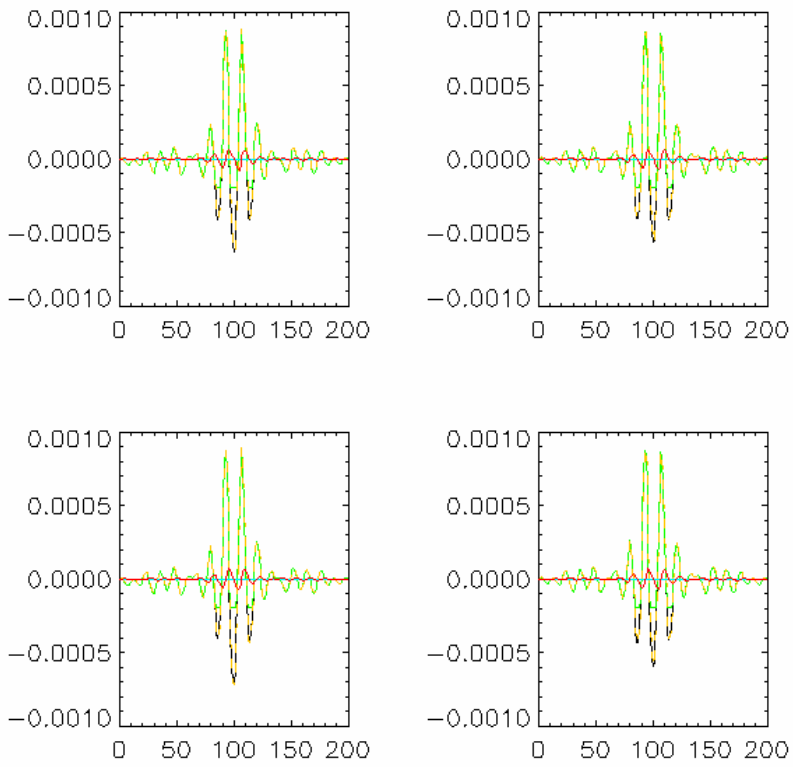

Figure 15a: The same as Fig. 14a, but for detector SLWC5 (bottom clipping at ZPD).

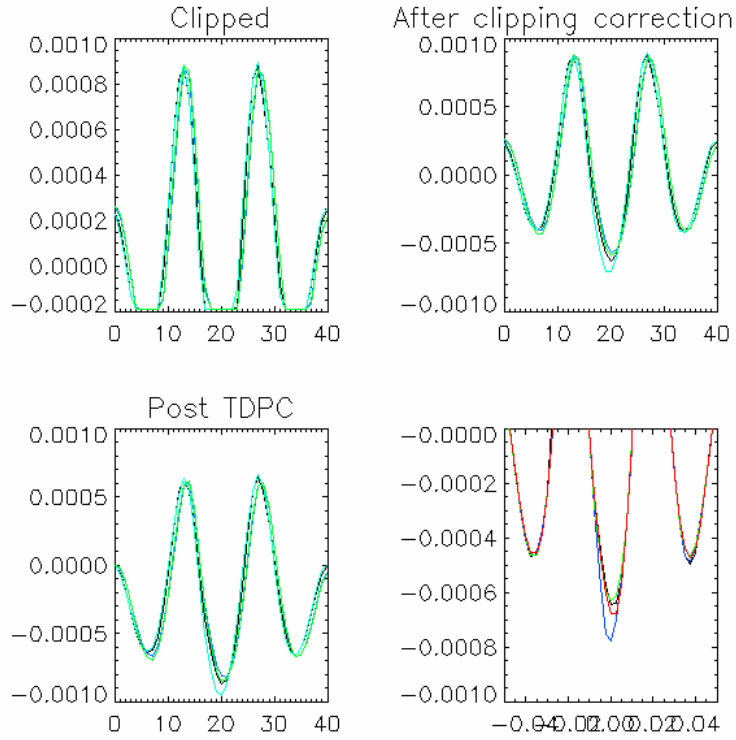

Figure 15b: The same as Fig 14b, but for detector SLWC5 (bottom clipping at ZPD). The difference in reconstructed ZPD value is apparent while the rest of the interferogram is nicely overlayed.

We also compared neighbouring detectors which did not show clipping, to look for evidence of an imbalance in the lobes of unclipped data. These detectors did not show as large a magnitude of lobe imbalance as the reconstructed clipped data.

We proceeded to run the rest of the pipeline to observe if there was correlation between obvious spectral artefacts. There is none.

The final spectrum from one detector which showed strong clipping on all 4 scans (SLWC5) is shown in Figure 16. In this case, the 4 spectra show strong phase errors which are different in each of the 4 scans. These may be related to the imbalance in lobes found in the clip corrected interferograms.

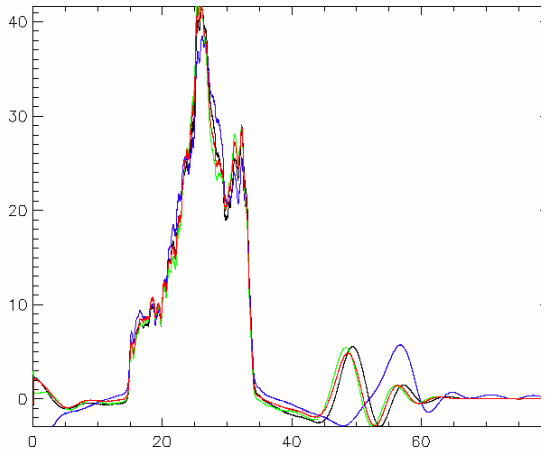

Figure 16: The spectra from the 4 repeated scans for SLWC5 – all 4 scans had heavy clipping.

3.3.13.1 Clipped particle hits

The prospect of a large glitch present in an interferogram which produces 1 or more clipped points deserves a separate mention. It is unclear under the present pipeline order (i.e. temporal sequence of modules) if the deglitching routine would remove such an event (especially if the number of clipped points is greater than 1). If it were so, would the clipped flag still try and correct for this or does the glitch flag make the first one void?

With the clipping correction module in its present position in the pipeline, it would seem reasonable to multiply the clipped flag with one-minus-the glitchflag (if possible) in order to void the clipping correction if a glitch is present.

Ideally (although this is not currently the case), the temporal sequence of events (as described in Sect. 3.1) should force us to correct for the clipped glitch (with a procedure different from that of a non-glitch clipping) prior to the glitch removal.

3.3.13.2 High OPD clipping

A test was also performed on a highly vignetted detector (SSWB1) with ~10^3 clipped points at the high OPD end of the scan. As expected, the clipping correction module attempts to perform the same polynomial fitting and hence introduces a large unwanted peak (see Figure 17). Subsequent analysis shows that this introduces unwanted features in the spectrum.

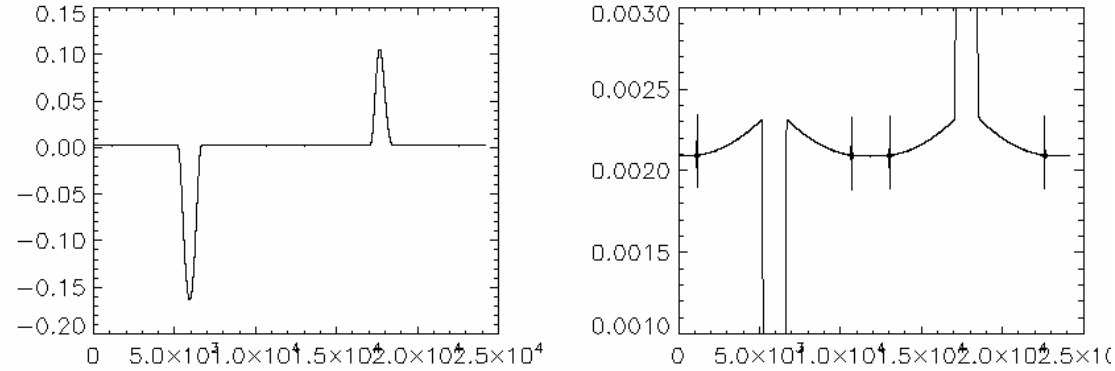

Figure 17: The effect of applying the clipping correction to an observation with clipping at high OPD. The left plot shows the reconstructed timeline and the right plot shows a zoom.

3.3.14 Conclusions, Recommendations & Comments

- We confirmed that no modification takes place in absence of clipping
- We identified the possibility that the clipping correction might introduce artifacts but the verification of this is non trivial due to pipeline module interactions. This could be investigated further in Phase 3.
- We confirmed that the module attempts to clip-correct High-OPD clipped interferograms, resulting in a huge polynomial being introduced and the scan resulting void. The following questions arise:
  - Is it possible to add a flag that identifies this type of clipping and void the scan/detector?
  - Is the scan salvageable by reducing the considered OPD to the non-clipped range, reducing the overall resolution of the final spectra for that detector?
- The task parameter "number" (and also the names of the other task parameters) should have a more meaningful name.
- The following are examples of messages that are printed to the console during the operation of the task:

```
    10-Feb-09 09:38:47.468 INFO ClippingTask: detector :SLWA1
    10-Feb-09 09:38:47.468 INFO ClippingTask: reconstruction not be
done for the1141 points of the sample
```

The meaning of these messages is not self-evident. Is each detector listed as the task processes the data? What does the number of points of the sample refer to? The text of the message should be tidied up in agreement with the logging policy defined during the last integration meeting (see http://www.herschel.be/twiki/bin/view/Spire/PipelineIntegration07PostMortem#Logging).

Further points and considerations for future testing

- A test could be carried out to compare the effectiveness of the reconstruction when different numbers of consecutive points are clipped (e.g. 1, 2, 3 points)
- What happens when the input parameter to the clipping task, "number", is zero, negative, or more generally too small to perform an 8th order fit?

3.4 Bath Temperature Fluctuation Correction

The purpose of this module is to correct the timeline detector data for the bath temperature fluctuations. This is done by removing a correction timeline calculated with an algorithm that uses a table of calibration coefficients and the two thermistor timelines. This correction accounts for the difference between the instantaneous bath temperature T and a reference bath temperature T_0. The output from the module is the signal one would get if the bath temperature remained at T_0.

This module was tested by two groups:

- Nanyao Lu
- Giorgio Savini and Ed Polehampton

Module tests carried out by Nanyao Lu

3.4.1 Test Data

PFM 5 data were used.

OBSID	Date and Time	Mode	Number Scans	Source details
0x300223C	16/02/07 22:21-00:10	N/A	N/A	Multi-phase spectrometer load curve

We processed only one Building Block from this dataset: BBID = 0x88010005, for which the detector bias voltages (amplitude) were 31.12 mV. The mean detector bath temperatures from the thermistors was approximately 298 mK, so we set T_0 = 298 mK.

We generated a specNonLin calibration table assuming T_0 = 298 mK, V_{bias} = 31.12 mV (amplitude) for both SSW and SLW, and zero telescope background. A corresponding calibration table for the temperature drift correction module was also generated. These tables are available on the TWiki.

The pipeline was run on a MacBook Pro laptop with Leopard OS and 4GB ram.

3.4.2 Validation Process

We have two science requirements to test:

1. Removal of much of the low-frequency correlated noise from each detector channel.
   The module should remove most of the "signal drift" that is known to be correlated with the similar signal drift of the thermistors.

   For each detector channel, if we fit the voltage timeline to a 2nd-order polynomial: $a_0 + a_1$*time + a_2*time^2, then we require:
   $|a_1|$ (prior tempDrift module) > $|a_1|$ (post tempDrift module)
   $|a_2|$ (prior tempDrift module) > $|a_2|$ (post tempDrift module)

2. The module should not increase the white noise significantly.
   Let RMS = the root mean square noise with respect to the polynomial fit defined above. Then we require:
   RMS (prior tempDrift module) ≈ RMS (post tempDrift module)

We used a Jython script adapted from the standard pipeline script, containing the following key steps:

(a) Load the test data (0x3001223C) from the local pool, and process only BBID 0x88010005
(b) Only the following 3 pipeline modules are executed: deglitching, specNonLin, and tempDrift. The calibration tables are imported at the appropriate places.
(c) Compare the voltage timelines before and after the tempDrift module, and make polynomial fits and derive RMS noise.
(d) Carry out independent calculation of the temperature drift correction.

3.4.3 Validation Results

Figure 18 (left) compares the voltage timelines before and after the temperature drift correction module for channel SSWA4.

The polynomial fits gave:
(a_0, a_1, a_2) = (-3.8786486799187914E-4, 2.670400165204788E-9, -2.7830982665915753E-12) before the temperature drift correction
(a_0, a_1, a_2) = (-3.3194110291859957E-4, 2.609456823952025E-10, -4.7644947946299151E-13) after the temperature drift correction

Clearly,
$|a_1|$ (before module correction) > $|a_1|$ (after module correction)
$|a_2|$ (before module correction) > $|a_2|$ (after module correction)

The calculated RMS with respect to the polynomial fit is:
RMS_{before} = 1.3048415434771173E-7 V (before module correction)
RMS_{after} = 1.3029340095122256E-7 V (after module correction)

Similarly, Figure 18 (right) compares the voltage timelines before and after the temperature drift correction module for channel SLWA1.

The polynomial fits gave:
(a_0, a_1, a_2) = (-2.6127242027120293E-4, 2.379045267746436E-9, -2.89489673897373E-12) before the temperature drift correction
(a_0, a_1, a_2) = (-2.5850577006557556E-4, 1.8676008005288797E-10, -4.399444081015872E-13) after the temperature drift correction

Clearly,
$|a_1|$ (before module correction) > $|a_1|$ (after module correction)
$|a_2|$ (before module correction) > $|a_2|$ (after module correction)

The calculated RMS with respect to the polynomial fit is:
RMS_{before} = 1.054939199E-7 V (before module correction)
RMS_{after} = 1.041207475E-7 V (after module correction)

Similar quantitative results are observed on all the other valid detector channels. In conclusion, the module passed both our science requirements.

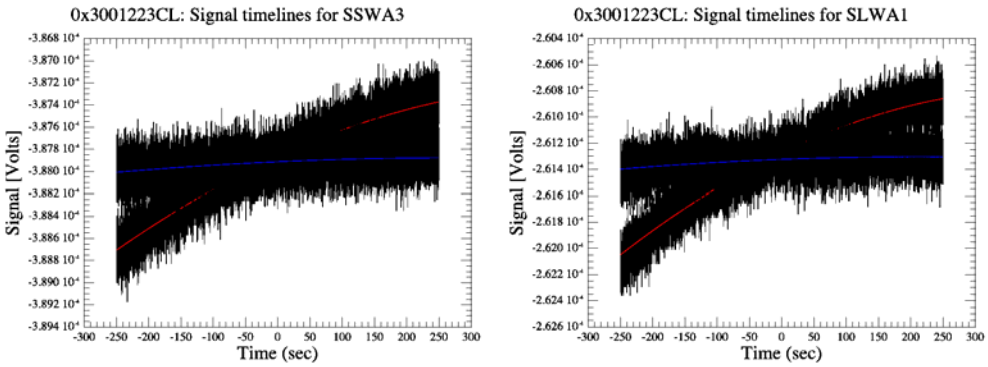

Figure 18: Comparison of the voltage timelines before and after the temperature drift correction module on channels SSWA3 and SLWA1. The 2nd-order polynomial fits are shown in red and blue, respectively. To plot the two data sets on the same scale, the corrected timeline has been shifted vertically by -5.597x10^{-5} V for SSWA3 and -2.818x10^{-6} V for SLWA1.

Module tests carried out by Giorgio Savini and Ed Polehampton

3.4.4 Input Data

Real data from PFM4 was used as input. The observation used was:

OBSID	Date and Time	Mode	Number Scans	Source details
0x300117FE	8/12/06 18:42-18:50	H	4 High	SCAL4 @ 67.9 K, Laser on SSWD3

In order to process these for input to the module, we used the obsExporter tool to obtain Level-0.5 data from the database in a pool. We ran the level 0.5 detector timeline directly through the clipping module.

The data was saved before and after the temperature drift module and then loaded into IDL for analysis. A similar analysis was also performed in Hipe.

3.4.5 Test Procedure

We aimed to test the following points:

1) Verify that the correction is applied to all detectors.
2) Check that the removed timeline corresponds to the calibration coefficients in the table (by reproducing the result).
3) Verify that small scale features in the thermistor timeline (e.g. peak at ZPD) are not propagated in the correction.

The first test was carried out and the result was that a correction is applied to all of the optical-detectors. The dark detectors and the resistor (and also the thermistors themselves) are not corrected by this module.

The second test was performed by comparing the difference between the timelines before and after the temperature drift correction module was applied. By taking this difference we reconstructed the correction that had been applied.

This was compared with the expected correction timeline obtained by combining the two thermistor detectors as indicated in the Pipeline Description Document,

$$S_i = D_i - TC_i$$

$$TCi = A_1 \times \mathrm{SMOOTH}(T_1 - V_{0T1}) + 0.5 \times B_1 \times [\mathrm{SMOOTH}(T_1 - V_{0T1})]^2 + A_2 \times \mathrm{SMOOTH}(T_2 - V_{0T2}) + 0.5 \times B_2 \times [\mathrm{SMOOTH}(T_2 - V_{0T2})]^2$$

The values for the parameters A_1, A_2, B_1, B_2, V_{0T1}, V_{0T2} were taken from the calibration product and saved to fits for loading in IDL.

We found that the results derived using IDL matched the output of the module. Figure 19 shows the differences for the two arrays (Output of module – expected output derived in IDL). The smoothing adopted in IDL was 5s x 80Hz = 400 points. Figure 20 shows the correction applied by the module, with the expected correction calculated in IDL. This shows that we can reproduce the correction applied by the module very well using the equations.

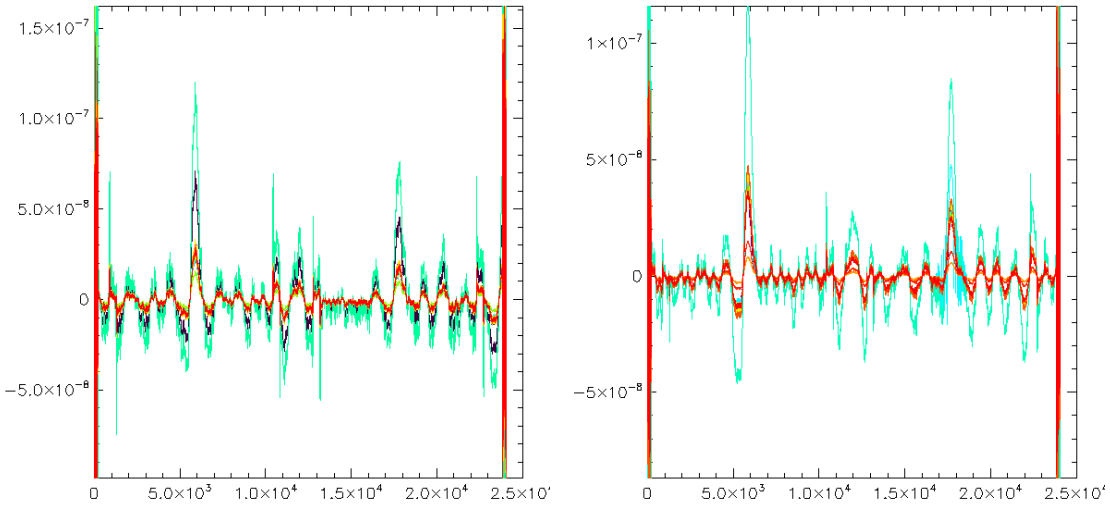

Figure 19: The left plot shows the SLW difference between output of temperature drift correction module and IDL-generated desired output.

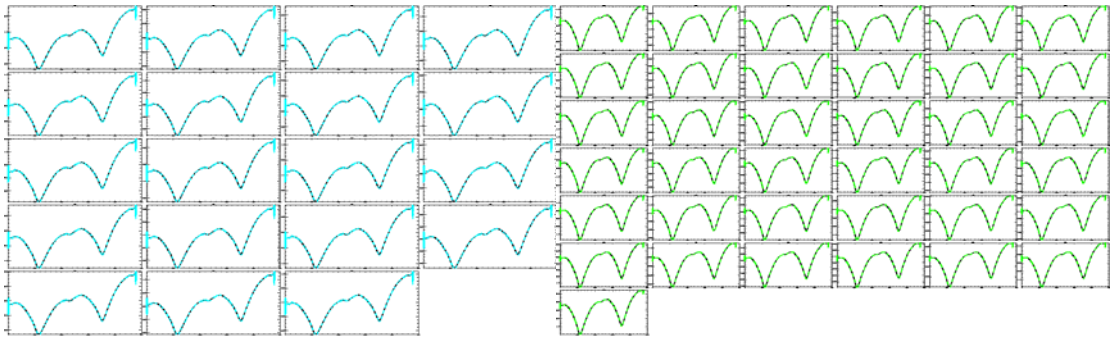

Figure 20: The left hand plots show the SLW temperature drift correction applied by the pipeline (black) and the correction calculated using IDL overplotted in blue. The right hand plots show the same thing for SSW.

We also applied the equations to the unsmoothed thermistor timelines to see which features are smoothed out (Figure 21). This shows that there is some peak around ZPD in the thermistor timeline (either due to temperature fluctuation due to changing optical load on detector plate, or due to electrical crosstalk), but it is indeed smoothed out and so is not applied to the data. The only potential problem of the smoothing is the join between SMEC scans. There seems to be a discontinuity in the data there that is smoothed out in the correction. If any of these large scale changes in the thermistor timelines, which are not smoothed out, are due to electrical crosstalk rather than temperature fluctuations, they will be propagated to all other detectors by this module.

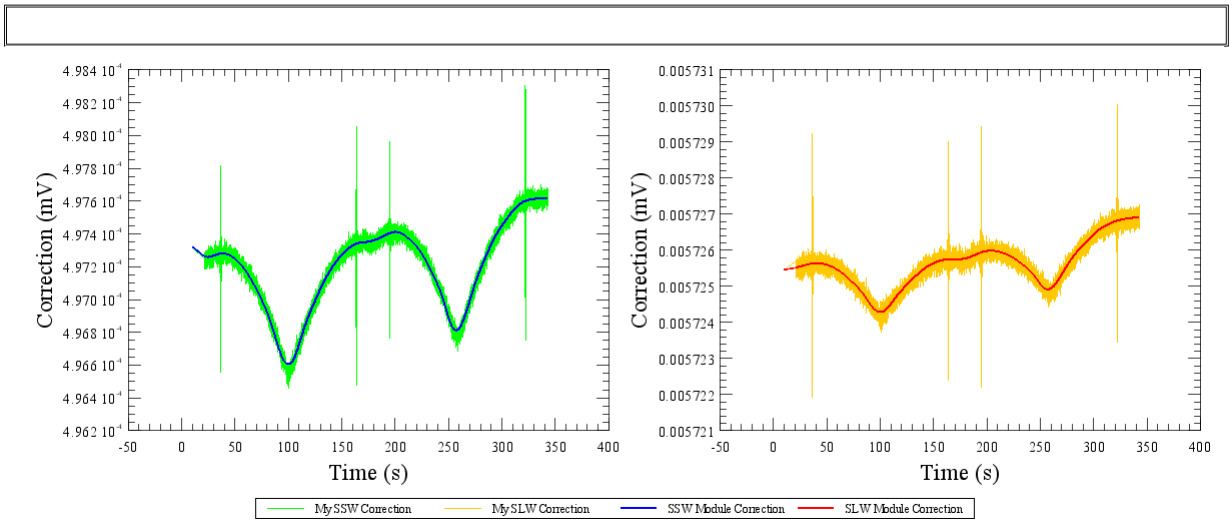

Figure 21: The correction applied by the temperature drift correction task for SSWD4 (left) and SLWA1 (right) plotted on top of the calculation using unsmoothed thermistor timelines.

3.4.6 Conclusions, Recommendations & Comments

The desired temperature drift is removed to a good degree by the module.

Other considerations:

- It is of paramount importance that some previous modules (especially the electrical crosstalk) have worked efficiently in order for this module not to introduce some significant and dangerous artefacts in the data. This consideration does not affect the validity of the module which performs as required. On the other hand it might affect the calibration parameters if this has not already been accounted for.

3.5 Compute BSM Angles

Tested by Chris Pearson.

The *Compute BSM Angles* Module converts the angles of the Beam Steering Mirror to physical units (decimal degree in the sky). The method is simple: interpolation from a calibration table containing the two converted angles (Y, Z) versus the two raw angles (chop angle, jiggle angle). Note that for FTS observations there is no need to create a chop-jiggle timeline as in the case of photometer jiggle observations; however it is still necessary to create a BSM angle timeline

- **Input Data Products: Nominal House Keeping Timeline (NHKT)** This contains the timeline of the positions of the BSM as sensor signal values (the chop and the jiggle positions), in raw format in the chopsenssig and jiggsenssig columns.

- **Output Data Products: BSM Angles Timeline (BAT)** It contains timelines of angular distance on the sky from its zero position (in spacecraft Y, Z coordinates)

- **Calibration Products: BSM Position Table (BSMPT)** This table contains BSM sensor signal in raw units (in chop and jiggle directions) and angular distance on the sky from its zero position (in spacecraft Y, Z coordinates). The BSMPT calibration file name in the build is listed as *SCalSpecBsmPos*.

- **Calling Procedure:** Assuming that *nhkt*, a NHKT product, and *bsmPos* is a BSMPT calibration product
  bat=calcBsmAngles(nhkt,bsmPos)

- **User Parameters:** None

3.5.1 Input Data

PFM 4 data were used:

OBSID	Date and Time	Mode	Number Scans	Source details
0x300117FE	8/12/06 18:42-18:50	H	4 High	SCAL4 @ 67.9 K, Laser on SSWD3

The data were input using the FitsArchive method within HIPE. No local store or pool was used.

3.5.2 Test Procedure

The module is tested using real test data from the PFM-4 AOT ground tests. The test proceeds under the following algorithm.

- Input a NHK timeline (NHKT)
- Input a BSM Positions Table (BSMPT)
- From the NHKT, extract the time, chop sensor signal, jiggle sensor signal columns from the table dataset (for testing purposes only)
- Run the module with the NHKT and BSMPT as inputs
- Output a BSM Angles Timeline (BAT)
- Plot the Chop sensor from the NHKT and the Y angle from the BAT and then overplot the Chop sensor and Y angle from the BSM positions Table.
- Plot the Jiggle sensor from the NHKT and the Z angle from the BAT and then overplot the Jiggle sensor and Z angle from the BSM positions Table.

The testing script is available on the Science Validation Group TWiki at:

http://www.herschel.be/twiki/bin/view/Spire/FTSPipelineGroup

When plotting the sensor position from the NHKT and the angle from the BAT then overplotting the sensor and angle from the BSM positions Table should result in overlapping data in the range covered by the NHKT if the conversion was done correctly

3.5.3 The "BSM Positions Table" Calibration File

The BSMPT is imported in the same way as the NHKT file. The BSM Chop and Jiggle sensor rest positions are stored in the (first) metadata. The values are

BSM chop rest position = 37632
BSM jiggle rest position = 39520

The chop, jiggle positions and Y,Z values in the BSMPT can be obtained in JIDE/HIPE by either
Java style: *bsmposY=bsmPos.getYangle()*
Jython Syle: *bsmposY=bsmPos.yangle*

In Figure 22 the relationship between the chop sensor and Y angle, and the Jiggle sensor and Z angle in the BSMPT calibration file is shown. It can be seen that the chop-Y relation is modelled by a polynomial while the jiggle-Z relationship is assumed to be linear. Note that the Y=0, Z=0 positions in the calibration files have the chop and jiggle values of 37413 and 39936. The Y angle value of the rest position of the chop sensor (37632) is 1.44 arcsec. Note that since the values in the calibration table are interpolated from a polynomial fit it is possible that the polynomial fit does not exactly pass through the rest position described by the metadata. The difference in the chop rest positions in the metadata and the table is 37632-37413 =219 ADU such that 1 arcsec shift corresponds to approximately 152 ADU. The pitch of the PSW array pixels is 28.15 arcsec so a 1.44 arcsec offset corresponds to approximately 5% of a pixel shift in the focal plane.

The BSM positions table stores the BSM chop and jiggle sensor signals in raw units and angular distance on the sky from its zero position (in spacecraft Y, Z coordinates). These values are calculated from a polynominal fit (or a liner fit in the case of the Jiggle sensor / Z-angle case) derived from Laboratory test data. The original BSM Positions Table is shown in Figure 22. The polynominal causes duplicate values of the Y-angle for a given value of the Chop Sensor (Y angles in the range 100-180 arcsecs) causing the BSM angles conversion module to fail with a "*Values must be in ascending order*" error. This was already raised in **SPIRE SPR-1137** (*Cannot create BSM Angles timeline*). Printing the values of the Chop sensor shows the duplicate values (in blue);

```
print bsmPos['table']['chopSensor']
{description="Chopper Sensor",
data=[16532,16537,16574,16642,16741,16869,17025,17209,17420,17658,17921,18208,18519,18853,19209
,19587,19985,20402,20839,21293,21765,22253,22757,23276,23808,24354,24912,25482,26062,26652,2725
2,27859,28474,29096,29724,30356,30993,31633,32276,32920,33566,34211,34856,35500,36141,36779,374
13,38042,38666,39283,39894,40496,41089,41673,42246,42808,43358,43895,44419,44928,45421,45899,46
359,46802,47226,47630,48015,48378,48720,49039,49334,49605,49851,50071,50265,50431,50568,50676,5
0755,50802,50818,50802,50752,50668,50550,50395,50204,49976,49710,49405,49060,48674,48247]
```

Therefore the BSM Positions Table is artificially truncated at a lower value of Y-angle to produce a new table shown in the right panel of Figure 22. The Table is truncated by removing the duplicate rows using the *bsmPos['table'].removeRow(row)* command.

The new truncated values are used for the purpose of this test. In future the module should use the "*softlimits*" parameter in its metadata to avoid this problem.

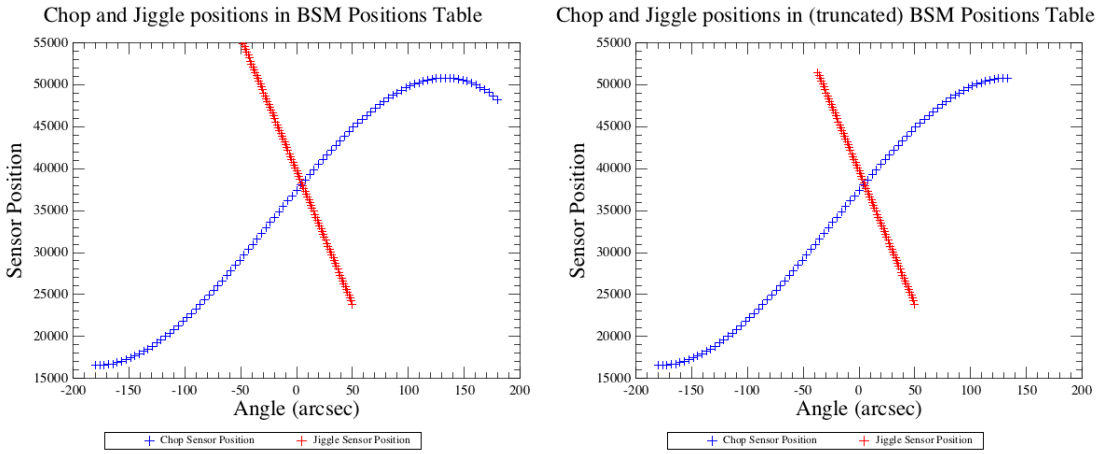

Figure 22: (left) The BSM Positions file contains the *relationship between chop sensor and Y angle, and the Jiggle sensor and Z angle. In the case of the chop-Y relation a polynominal is used to model the fit. The polynominal causes duplicate values of the Y-angle for a given value of the Chop Sensor (Y angles in the range 100-180 arcsecs) causing the BSM angles conversion module to fail. Therefore the BSM Positions Table is artificially truncated at a lower value of Y-angle to produce a new table shown in the **right** panel.*

3.5.4 The "NHKT" Data

The chop and jiggle positions are stored in the NHKT in the "CHOPSENSSIG" and "JIGGSENSSIG" columns. In Figure 23 these chop and jiggle sensor positions are plotted as a function of sample time. As expected the sensors positions are more or less constant around a chop sensor value of 37632 and a jiggle sensor value of 39519. Figure 24 shows the dispersion of chop and jiggle positions for all time samples. The dispersion is small of the order of 10 ADU in sensor signal (about 1/15th of an arcsec at the centre of the array).

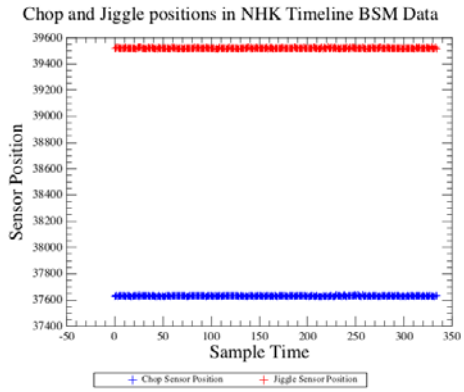

Figure 23: *NHK timeline Chop sensor value and Jiggle senor value as a function of sample time*

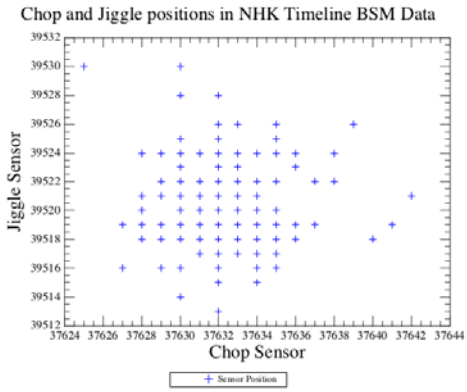

Figure 24: *NHKT chop and jiggle positions relative to each other. Note the small range on the axes.*

3.5.5 Running the module and creating the BSM Angles Timeline

The BSMPT calibration file and NHKT product are inputs to the Compute BSM Angles Module. The module is called as *bat=calcBsmAngles(nhkt,bsmPos)* where *bat* is the output BSM Angles Timeline which contains;

- the sample time (s)
- Y angle (arcsec)
- Y angle error (arcsec), all the errors are zeroes*.
- Z angle (arcsec)
- Z angle error (arcsec), all the errors are zeroes.

The Resulting BAT is shown in Figure 25 and should correspond to Figure 23.
* Since the offsets for the Y-Z angle spacing are derived from an optical model the errors are zero (because its a model). In reality the errors may be 1% of the absolute position so at the edge of the array 0.01x180 arcsec which is still small. The errors will be calculated in flight. The polynomial fit to the data produces errors and these should indeed be propagated.

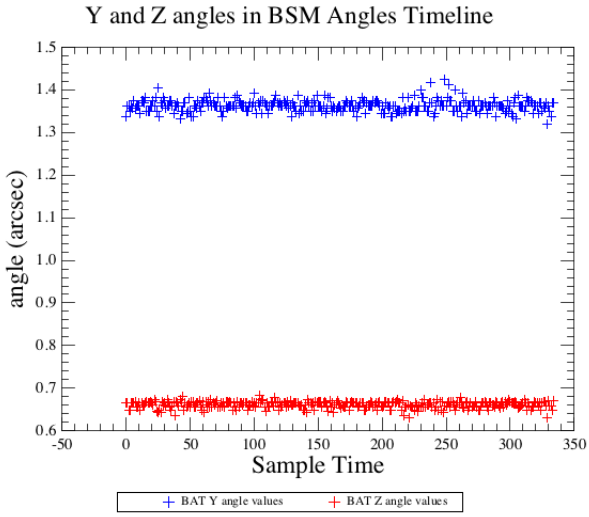

Figure 25: *BSM Angles Timeline (BAT), Y and Z angles with sample time*

In Figures 26 and 27, the BAT Y,Z angle dispersion is shown. Since the BSM is not moving the values cluster around a fixed point with locus of Z=0.66, Y=1.36.

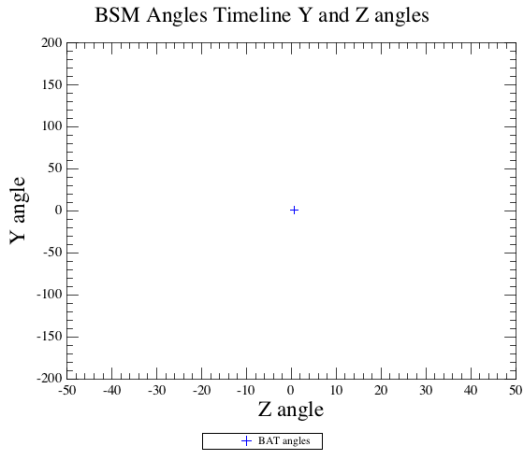

Figure 26: *BAT Y and Z angles*

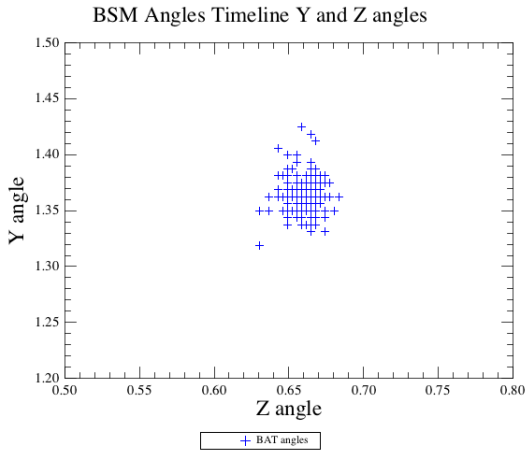

Figure 27: *As previous figure but zoomed in.*

Since the NHKT contains the chop position and Jiggle position sensor values as a function of sample time for the observation and the BAT contains the Y, Z angles for an observation as a function of sample time, a simple check can be made that the time is written correctly to the BAT from the NHKT. Figure 28 shows the NHKT and BAT sample times plotted against each other to show the time was copied across correctly

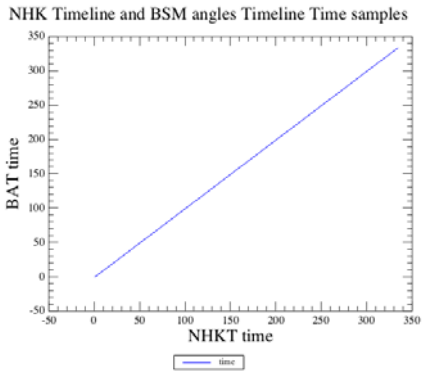

Figure 28: Comparison of NHKT and BAT sample times

In Figures 29 and 30, the relation between the NHK chop sensor and BAT Y angle is shown for the same sample times (i.e. the sensor values and angles corresponding to the NHKT time and BAT time respectively). By comparing these values with the calibration curves for the chop-Y relation in the BSMPT, we can check that the "Compute BSM Angles" module has successfully converted the chop sensor values into angles. In Figures 31 and 32, the corresponding calibration curves are overlaid to indeed confirm a successful conversion. In Figures 33, 34, 35 and 36, the results are shown for the jiggle-Z conversion.

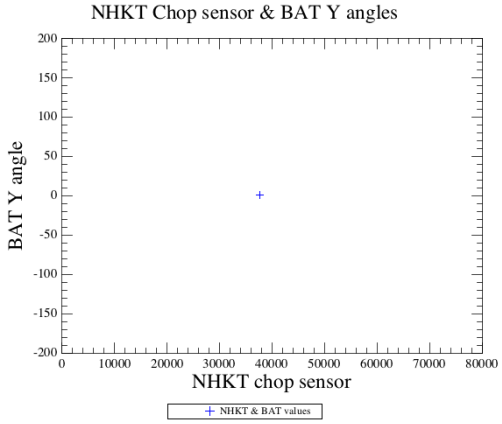

Figure 29: *Relation between NHKT chop sensor and BAT Y angle.*

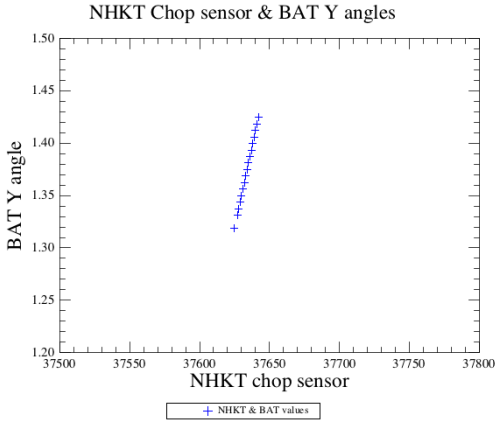

Figure 309: *Relation between NHKT chop sensor and BAT Y angle with axes zoomed in*

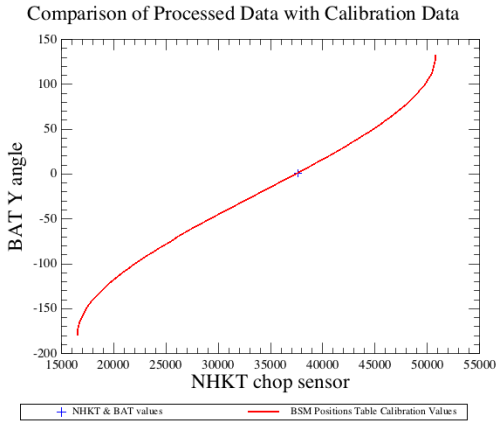

Figure 31: *Relation between NHKT chop sensor and BAT Y angle. The calibration curve from the BSMPT is overplotted.*

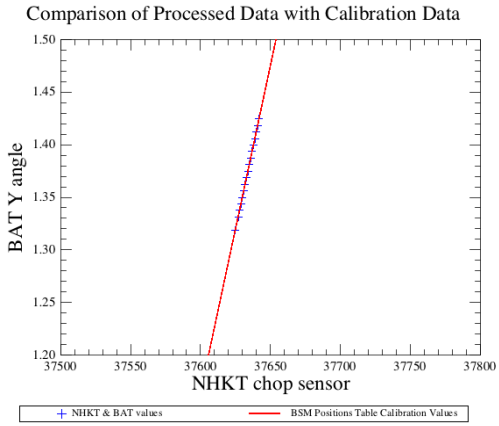

Figure 32: *Relation between NHKT chop sensor and BAT Y angle. The calibration curve from the BSMPT is overplotted.*

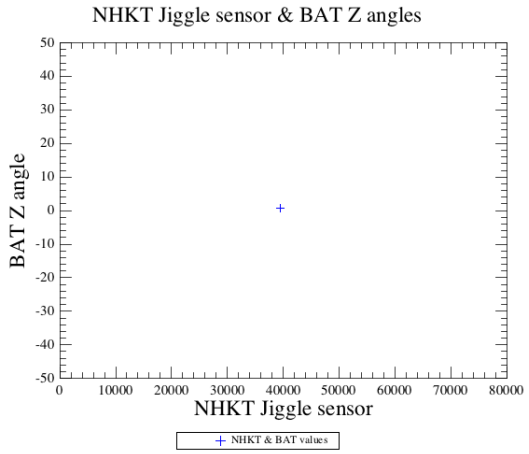

Figure 33: *Relation between NHKT jiggle sensor and BAT Z angle.*

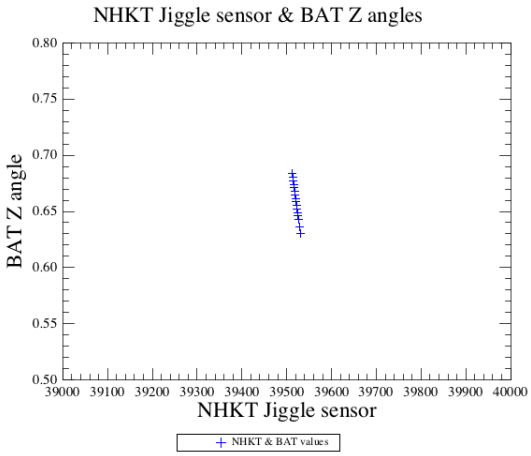

Figure 34: *Relation between NHKT jiggle sensor and BAT Z angle with axes zoomed in*

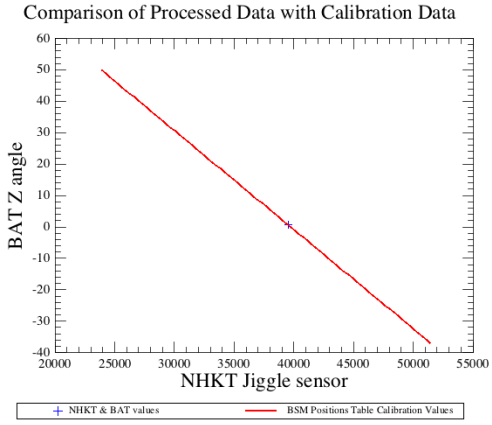

Figure 35: *Relation between NHKT jiggle sensor and BAT Z angle. In addition the calibration curve from the BSMPT is overplotted.*

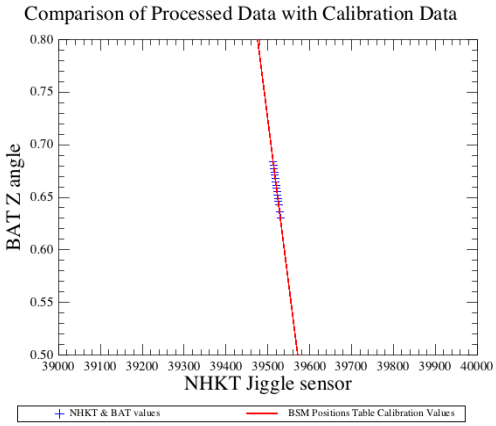

Figure 36: *Relation between NHKT jiggle sensor and BAT Z angle. In addition the calibration curve from the BSMPT is overplotted.*

3.5.6 Module Testing with Dummy Data

3.5.6.1 Test Strategy

In order to exercise the “Compute BSM Angles” module more thoroughly and to check that the BSM conversion works with a wider range of input values an additional test is carried out where a dummy NHKT product is created with the NHKT sample time, chop sensor and jiggle sensor columns populated by an incremental array from 1 to 65536 as shown in Figure 37.

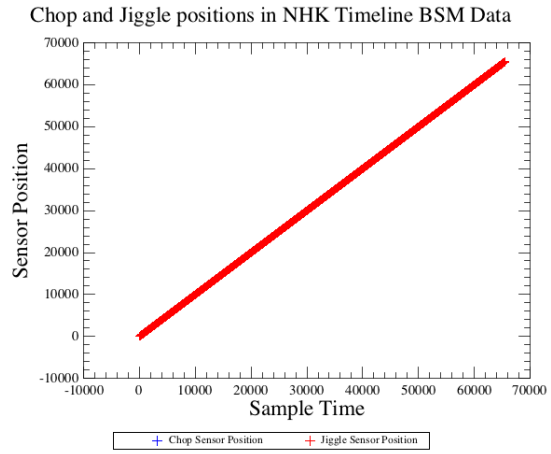

Figure 37: *Chop and Jiggle positions entered as incremental arrays from 1 –65536. Note the two lines are on top of one another.*

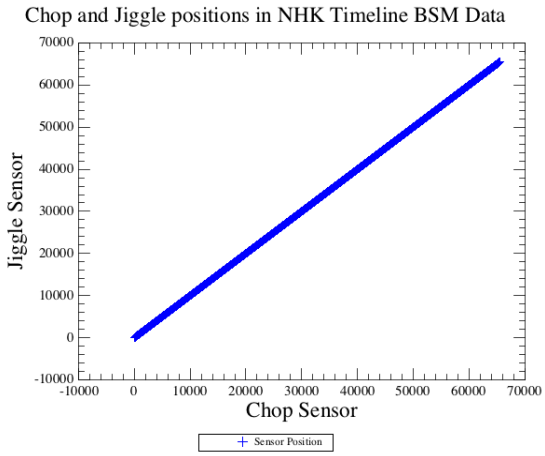

Figure 38: *Chop and Jiggle sensor position distribution in dummy NHKT product*

3.5.6.2 Running the module and creating the BSM Angles Timeline

The module is called as *bat=calcBsmAngles(nhkt,bsmPos)*, and the BSM Angles timeline is created. The same plots are made of the resulting values with the calibration table values being over plotted. The results are shown in Figures 39, 40, 41 and 42 for the chop sensor – Y angle relation. Note that again the module does indeed carry out a successful conversion. However, the chop calibration is defined over a range smaller than the 0-65535 interval (the chop sensor is 16 bits); the algorithm makes an extrapolation outside the range given in the calibration table. Although the calibration does contain the validity ranges for the calibration table they are not yet implemented in the module.

The corresponding values for the jiggle sensor – Z angle relation are shown in Figure 43, 44, 45 and 46 where we can again see a good conversion but a lack of range checking on the data.

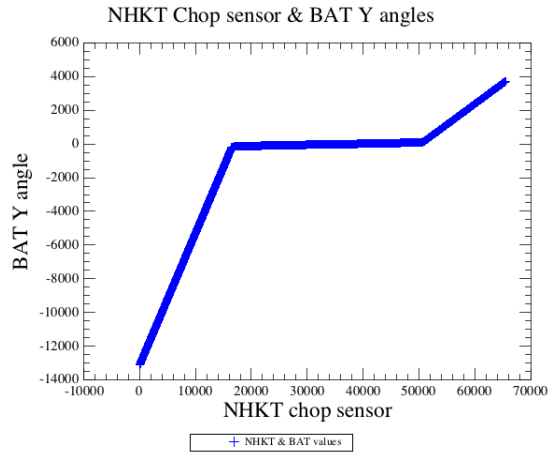

Figure 39: *Relation between NHKT dummy values chop sensor and BAT Y angle*

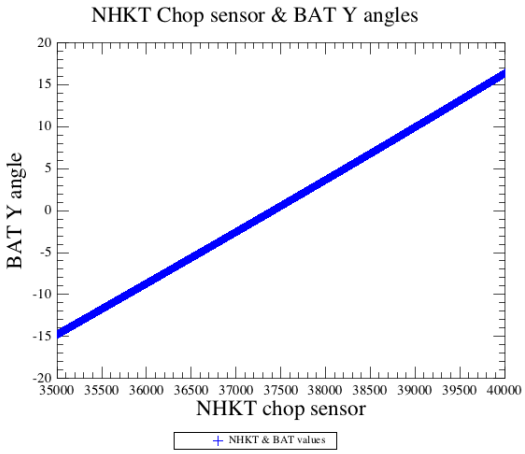

Figure 40: *Relation between NHKT dummy values chop sensor and BAT Y angle with axes zoomed in*

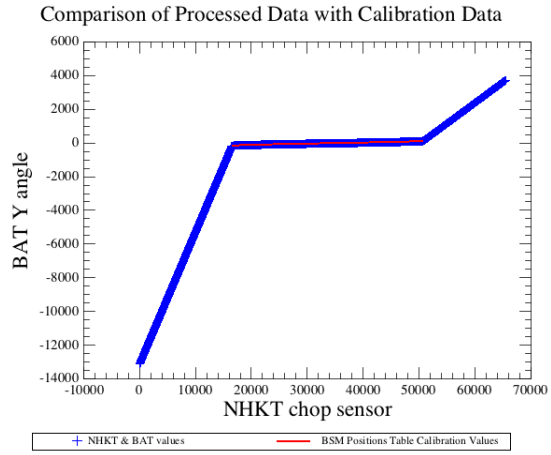

Figure 41: *Relation between NHKT dummy values chop sensor and BAT Y angle. In addition the calibration curve from the BSMPT is overplotted.*

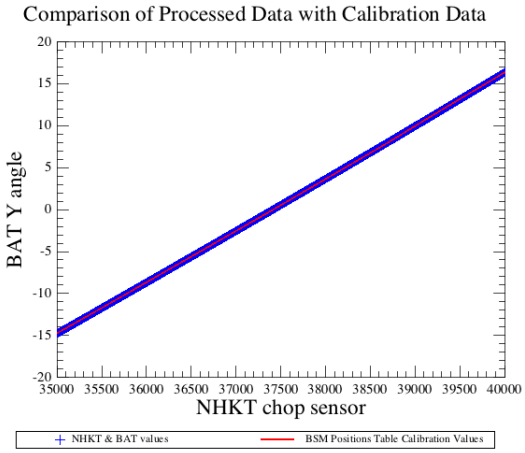

Figure 42: *Relation between NHKT dummy values chop sensor and BAT Y angle. In addition the calibration curve from the BSMPT is overplotted.*

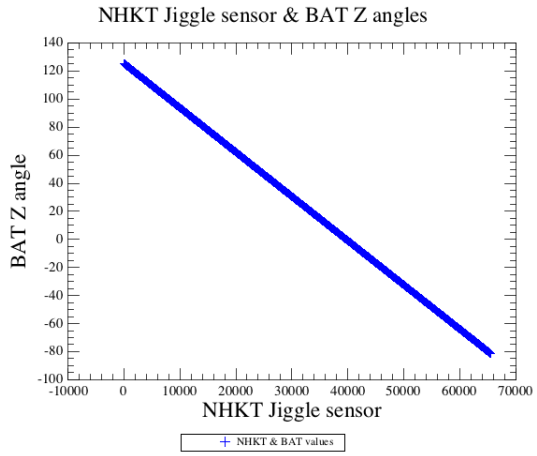

Figure 43: *Relation between NHKT dummy values jiggle sensor and BAT Z angle*

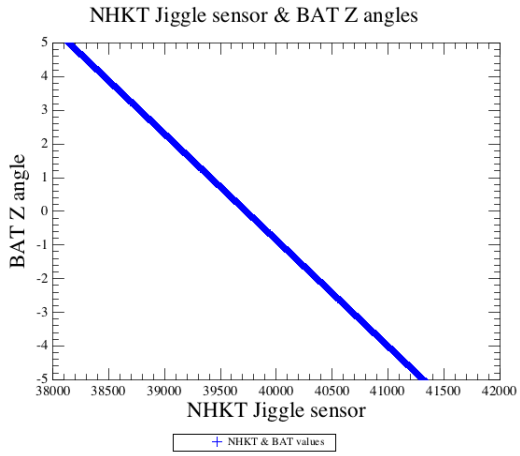

Figure 44: *Relation between NHKT dummy values jiggle sensor and BAT Z angle with axes zoomed in*

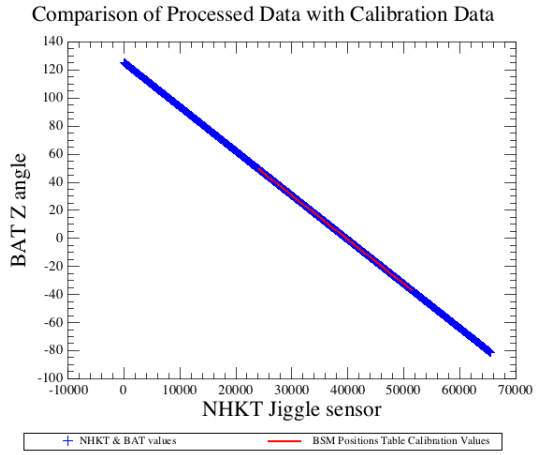

Figure 45: *Relation between NHKT dummy values jiggle sensor and BAT Z angle. In addition the calibration curve from the BSMPT is over plotted.*

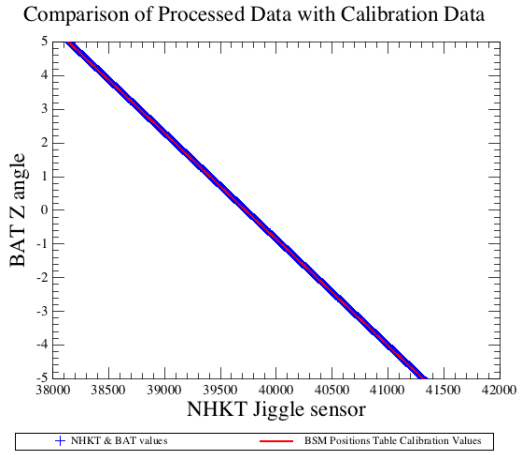

Figure 46: *Relation between NHKT dummy values jiggle sensor and BAT Z angle. In addition the calibration curve from the BSMPT is over plotted.*

3.5.7 Conclusions

3.5.7.1 Summary of Test Results

The “Compute BSM Angles” module has been tested on Spectrometer Data. The following conclusions are drawn:

1. The BSM Positions Table calibration file contains duplicate values for the chop sensor for different values of Y-angle. The module fails to compute the conversion due to this reason and the BSM positions table had to be manually edited for this test. Already reported in **SPR-1137**.
2. The chop and jiggle sensors (i.e. the BSM) does not move during the observation (except for a negligibly small fluctuation). However it is not situated at the zero (Y=0,Z=0) position.
3. The chop rest position in the metadata does not match the Y=0 position in the BSM Positions Table Calibration file (it instead has a Y=1.44 arcsec).
4. The jiggle rest position does match the Z=0 position in the BSM Positions Table Calibration file.
5. The errors in the calibration file are zeroes. The errors from the polynomial fit should be propagated to the calibration file.
6. The module successfully copies the sample time to the output BSM Angles Timeline product.
7. The module converts the sensor values successfully and accurately into angles.

The module was also tested on a greater range of data comprising of incremental arrays in sample time, chop sensor position and jiggle sensor position in the NHKT. The following conclusions are drawn:

1. The module converts the sensor values successfully and accurately into angles.
2. The module does not seem to have any checks for out of range values in the input NHKT and will extrapolate to any value input.

3.5.7.2 Recommendations

The following recommendations are made;

1. The BSM Positions Table should be updated to either include unique chop values or preferably the module should check for soft limits on the BSM conversion using metadata contained within the calibration file (i.e. this is the solution to **SPR-1137)**.

2. The module should check for out of range values.

3. Although the BSM is stationary during the observation it is not at the rest (Y=0, Z=0 or in this case Y=1.44, Z=0) position but is slightly offset at around of Z=0.66, Y=1.36. It is important to understand this in terms of the BSM errors etc.

4. The errors from the polynomial fit should be propagated to the calibration file.

3.6 SCAL, Telescope and Beamsplitter Correction

Tested by David Naylor and Scott Jones.

The purpose of this module is to subtract a reference interferogram from the data to remove the effect of the telescope, SCAL and the beamsplitters.

3.6.1 Input Data

PFM 4 data were used:

OBSID	Date and Time	Mode	Number Scans	Source details
0x300117FE	8/12/06 18:42-18:50	H	4 High	SCAL4 @ 67.9 K, Laser on SSWD3

3.6.2 Test Procedure & Results

In order to test the SCAL module subtraction, the first two scans of 0x300117FE were used as the reference. These were entered into the calibration product and then used as input to the module. If the module was working correctly, the data should have been subtracted from itself leaving an array of zeros.

Activating the *telScalCorrection* module did indeed return the expected result - an array of zeros.

The module did not execute within the pipeline using the calibration product that comes with the build. The console shows the following result:

```
        10-Feb-09 10:01:36.015 WARNING ScalTask: warning : Calibration product sdical is not
compatible with Enter product return = sdi enter
        10-Feb-09 10:01:36.015 INFO ScalTask: result{description="Spectrometer Detector
Interferogram", meta=[type, creator, creationDate, description, instrument, modelName,
startDate, endDate, aorLabel, aot, author, cusMode, dec, decNominal, equinox, instMode,
fileName, missionConfig, naifId, object, observer, obsid, obsMode, odNumber, pointingMode,
posAngle, proposal, ra, raDeSys, raNominal, telescope, subsystem, bbid, source, numScans,
commandedResolution, QcPhaseWrap, biasFreq,elecSide, bbTypeName, adcErr, offsetApp,
slwBiasAmpl, sswBiasAmpl, rcRollApp, jiggId, pointNum, maskMaster, maskInvalidTime,
maskAdcLatch, maskTruncated, maskUncorrectedTruncation, maskGlitchDetected,
maskGlitchNotRemoved, maskDead, maskNoisy, maskNotChoppedToSky, maskVoltageOol,
maskGlitchL1Detected, maskGlitchL1NotRemoved, maskGlitchL2Detected, maskGlitchL2NotRemoved,
ElectricalCrosstalkCorrectionDone], datasets=[0001, 0002, 0003, 0004, History], history=None}
```

The problem is that the reference interferogram only goes from -1.3 to 1.425 cm OPD. The reference interferogram should cover the OPD range for High and Medium res observations. The warning message should be more specific to support the identification of this problem.

3.7 Level 2 Deglitching

See the technical note *Deglitching SPIRE Interferograms*, version 0.5, December 17, 2008.

3.8 Flux Conversion

Tested by Nanyao Lu.

The Flux Conversion module (specFluxConv) converts the units of an input spectrum from volts to Jy using the conversion factors from the calibration table "SCalSpecFluxConv".

3.8.1 Test Data

PFM 4 data were used.

OBSID	Date and Time	Mode	Number Scans	Source details
0x3001172F	06/12/06 15:29-15:42	H	8 High	CBB @ 13 K

The flux conversion calibration table in the current build (242) contains dummy values of unity.

The pipeline is run on a MacBook Pro laptop with Leopard OS and 4GB ram.

3.8.2 Validation Process

Let *ssds1* = Spectrometer Detector Spectrum prior to the spectrum flux conversion module
ssds = Spectrometer Detector Spectrum after passing through the spectrum flux conversion module.

The module passes our tests if the numeric ratio of *ssds*.getScan(scan).getPixel(name).getData("flux") to *ssds1*.getScan(scan).getPixel(name).getData("flux") is always a data array of values of unity, for any valid scan number "scan" and any valid channel "name".

- Download the observation context with Level-0 data of OBSID 0x3001172F from the PFM4 RAL database to the local pool using obsExporter.
- Process the data per OBSID using a Jython script called "SpecFluxConv_0x3001172F.py," (available on the TWiki). This script, which is a modified version of the standard pipeline script SOF1.py, processes the data all the way to the execution of the module "specFluxConv". We keep two copies of SSDS (= herschel.spire.ia.dataset.SpectrometerDetectorSpectrum), one (SSDS1) represents the data prior to this module, and one (SSDS) after the data was fed through the module.
- For any scan number (SCAN, e.g., "0001") and any valid detector channel (NAME, e.g., "SSWA3"), we plot the ratio of SSDS.getScan(SCAN).getPixel(NAME).getData("flux") to SSDS1.getScan(SCAN).getPixel(NAME).getData("flux").

3.8.3 Validation Results

It is found that, for any valid scan (SCAN) and channel (NAME) tested, the ratio of input to output spectra is always an array of values of 1. Figure 47 shows examples for 2 channels.

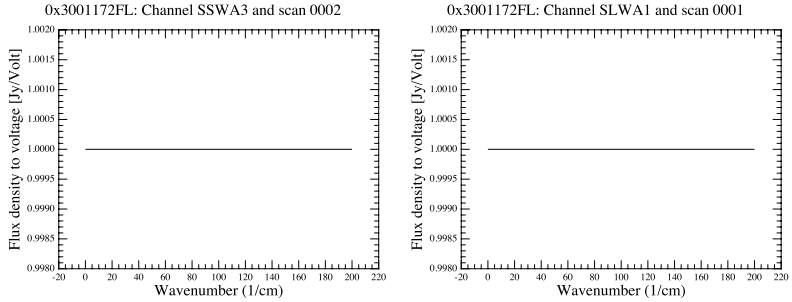

Figure 47: Flux-to-voltage ratio plot for SSWA3, scan 0002 (left) and SLWA1, scan 0001 (right).

3.9 Optical Crosstalk

Tested by Giorgio Savini and Ed Polehampton.

The purpose of this module is to correct the spectrum for optical crosstalk using a crosstalk matrix calibration file. Currently this calibration file is filled with zeros except the diagonal which is filled with ones (i.e. no crosstalk).

3.9.1 Input Data

Real data from PFM4 was used as input. The observation used was:

OBSID	Date and Time	Mode	Number Scans	Source details
0x300117FE	8/12/06 18:42-18:50	H	4 High	SCAL4 @ 67.9 K, Laser on SSWD3

In order to process these for input to the module, we used the obsExporter tool to obtain Level-0.5 data from the database in a pool. We ran the level 0.5 detector timeline directly through the electrical crosstalk module.

In order to prepare the data for the Optical Crosstalk module, we ran the following pipeline steps to create the spectrum:

- Time Domain Phase Correction
- Interferogram Creation
- Baseline Correction
- Phase Correction:
  - Apodize Interferogram (double sided)
  - FourierTransform (no zero padding)
  - Phase Correction
- Apodization
- Fourier Transform

3.9.2 Test Procedure

We aimed to test the following points:

1. Timeline output unchanged with identity matrix
2. Populate crosstalk matrix and check that timeline is affected in expected way

We ran the pipeline with the current calibration file, which should apply no correction. This was the case for optical detectors, but not for dark detectors, thermistors and resistors. A spectrum for these channels was included in the SDS product, but no information provided in the calibration file. Therefore, after correction, these channels were set to zero (see Figure 48).

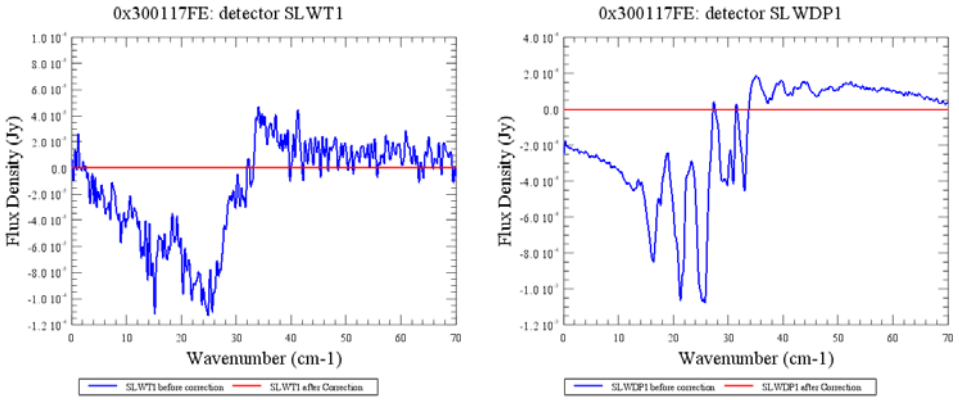

Figure 48: The spectra for thermistor and dark detector before and after correction by the optical crosstalk module.

We populated the crosstalk matrix by applying crosstalk between two detectors. According to Matt's pipeline document, the diagonal elements should not necessarily be equal to one. We filled the crosstalk matrix in as follows (but filling the appropriate rows and columns to apply crosstalk between SLWA3 and SLWC1):

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0.75 & 0.25 \\ 0 & 0 & 0.25 & 0.75 \end{bmatrix}$$

We calculated sum of the difference between the corrected and uncorrected spectra to see which detectors had been affected. The results for the four scans in this observation are as follows:

SSW

Detector	Scan 0001	Scan 0002	Scan 0003	Scan 0004	Detector	Scan 0001	Scan 0002	Scan 0003	Scan 0004
SSWA1	0.0000	0.0000	0.0000	0.0000	SSWE1	0.0000	0.0000	0.0000	0.0000
SSWA2	0.0000	0.0000	0.0000	0.0000	SSWE2	0.0000	0.0000	0.0000	0.0000
SSWA3	0.0000	0.0000	0.0000	0.0000	SSWE3	0.0000	0.0000	0.0000	0.0000
SSWA4	0.0000	0.0000	0.0000	0.0000	SSWE4	0.0000	0.0000	0.0000	0.0000
SSWB1	0.0000	0.0000	0.0000	0.0000	SSWE5	0.0000	0.0000	0.0000	0.0000
SSWB2	0.0000	0.0000	0.0000	0.0000	SSWE6	0.0000	0.0000	0.0000	0.0000
SSWB3	0.0000	0.0000	0.0000	0.0000	SSWF1	0.0000	0.0000	0.0000	0.0000
SSWB4	0.0000	0.0000	0.0000	0.0000	SSWF2	0.0000	0.0000	0.0000	0.0000
SSWB5	0.0000	0.0000	0.0000	0.0000	SSWF3	0.0000	0.0000	0.0000	0.0000
SSWC1	0.0000	0.0000	0.0000	0.0000	SSWF4	0.0000	0.0000	0.0000	0.0000
SSWC2	0.0000	0.0000	0.0000	0.0000	SSWF5	0.0000	0.0000	0.0000	0.0000
SSWC3	0.0000	0.0000	0.0000	0.0000	SSWG1	0.0000	0.0000	0.0000	0.0000
SSWC4	0.0000	0.0000	0.0000	0.0000	SSWG2	0.0000	0.0000	0.0000	0.0000
SSWC5	0.0000	0.0000	0.0000	0.0000	SSWG3	0.0000	0.0000	0.0000	0.0000
SSWC6	0.0000	0.0000	0.0000	0.0000	SSWG4	0.0000	0.0000	0.0000	0.0000
SSWD1	0.0000	0.0000	0.0000	0.0000	SSWDP1	-0.0852	0.2249	-0.0958	0.1803
SSWD2	0.0000	0.0000	0.0000	0.0000	SSWDP2	0.4635	-0.2446	0.4315	-0.2860
SSWD3	0.0000	0.0000	0.0000	0.0000	SSWT1	0.0752	-0.0074	0.0665	-0.0240
SSWD4	0.0000	0.0000	0.0000	0.0000	SSWT2	-0.0555	0.0688	-0.0470	0.0674
SSWD6	0.0000	0.0000	0.0000	0.0000	SSWR1	0.0088	0.0088	-0.0024	0.0064

SLW

Detector	Scan 0001	Scan 0002	Scan 0003	Scan 0004
SLWA1	0.0000	0.0000	0.0000	0.0000
SLWA2	0.0000	0.0000	0.0000	0.0000
SLWA3	1508.1946	-808.1354	1549.0040	-951.1779
SLWB1	0.0000	0.0000	0.0000	0.0000
SLWB2	0.0000	0.0000	0.0000	0.0000
SLWB4	0.0000	0.0000	0.0000	0.0000
SLWC1	-1131.1460	606.1015	-1161.7530	713.3834
SLWC3	0.0000	0.0000	0.0000	0.0000
SLWC4	0.0000	0.0000	0.0000	0.0000
SLWC5	0.0000	0.0000	0.0000	0.0000
SLWD1	0.0000	0.0000	0.0000	0.0000
SLWD2	0.0000	0.0000	0.0000	0.0000
SLWD3	0.0000	0.0000	0.0000	0.0000
SLWD4	0.0000	0.0000	0.0000	0.0000
SLWE1	0.0000	0.0000	0.0000	0.0000
SLWE2	0.0000	0.0000	0.0000	0.0000
SLWE3	0.0000	0.0000	0.0000	0.0000
SLWDP1	-0.7894	0.1391	-0.7162	0.2777
SLWR1	0.0046	-0.0080	0.0032	-0.0140
SLWT1	-0.0790	-0.0094	-0.0686	0.0012
SLWT2	0.0070	-0.0684	0.0023	-0.0566

This shows that the two detectors SLWA3 and SLWC1 were modified, as well as the dark detectors, thermistors and resistors.

In order to check the correction applied was as expected, we plotted the spectrum before and after and compared with that calculated as follows:

$$SLWA3_{corrected} = 0.75 \times SLWA3_{uncorr} + 0.25 \times SLWC1_{uncorr}$$
$$SLWC1_{corrected} = 0.25 \times SLWA3_{uncorr} + 0.75 \times SLWC1_{uncorr}$$

Figure 49 shows that the module applied the correction as expected for SLWA3, but not for SLWC1 (see zoom – the cyan line does not agree with the orange line). Further investigation showed that the actual equation applied to the second of the two detectors is:

$$SLWC1_{corrected} = 0.25 \times SLWA3_{corrected} + 0.75 \times SLWC1_{uncorr}$$

i.e. the factor for SLWA3 is applied to the spectrum after it has already been corrected.

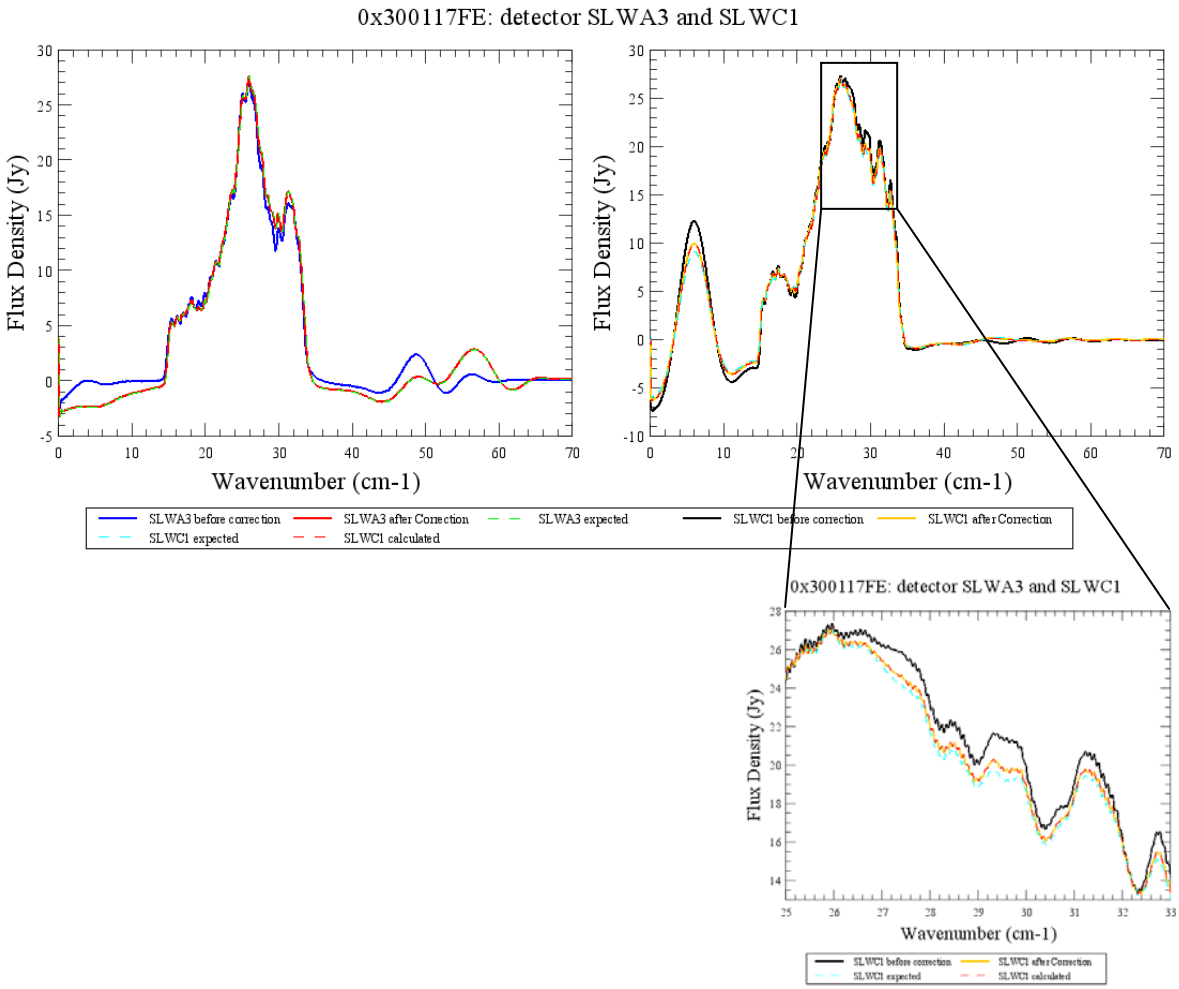

Figure 49: Results of applying the optical crosstalk matrix with crosstalk between detectors SLWA3 and SLWC1.

3.9.3 Conclusions, Recommendations & Comments

- The module propagates the original spectra when the identity matrix is used as calibration file, except for the non-optical detectors. These are set to zero. We recommend that these non-optical detectors are removed from the spectrum product earlier in the pipeline. This has already been proposed in **SPIRE SPR-1107**.

- The module applies the crosstalk incorrectly to the second detector. This should be corrected.

- There is some confusion in the documentation because the photometer optical crosstalk correction module still says that it also applies to the spectrometer (on SDT products). A brief examination of the java code shows that the photometer and spectrometer modules implement the correction in different ways. It should be made clearer which module applies to the spectrometer and photometer. Could the method used inside the code be harmonised between the two?

3.10 First Level Deglitching

This report is a recap and update of that given in Phase 1.

3.10.1 General considerations

As the module has been designed to detect Dirac like glitches we have used first such kind of glitches to validate the task and to better understand its performances.

At this stage *ONLY THE GLITCH **IDENTIFICATION** TASK HAS BEEN TESTED*.

3.10.2 First Analysis (achieved during Phase 1)

3.10.2.1 Method

We used a real observation, i.e. PFM4: 300114C8, which provides 16 HR interferograms in quite dark conditions. Very few strong glitches were found (visually) during this observation.

OBSID	Date and Time	Mode	Number Scans	Source details
0x300114C8	27/11/06 18:04-18:29	H	16 High	CBB @ 6.7 K

In order to validate the task we made use of the record from 2 detectors (SSWD4 and SLWC3) for which no glitches were detected (with the module run with default parameters). Each of these two detectors may be representative of the corresponding array detectors. Each record includes more than 100,000 samples.

300 glitches were added automatically to the recorded signal in the following way:

- all glitches have the same amplitude
- the first glitch is added at sample number 2000 and next ones are added every 350 samples (this provides quite a random distribution of glitches with respect to the ZPD positions for the 16 scans)

For this study we defined 3 regions:

- the LowRes part (SMEC position is less than 1250 µm away from ZPD)
- the MedRes part (SMEC posistion is less than 5000 µm away from ZPD, therefore MedRes includes also the LowRes part)
- the HighRes part (all OPDs, so HighRes includes MedRes)

Within this scheme we should be able to find:

- 18 glitches in the LowRes part
- 62 glitches in the MedRes part (18 LowRes + 44),
- 300 glitches in the HighRes part (62 MedRes + 238)

This should provide significant information regarding the percentage of glitches present that were actually detected for these three resolution modes.

3.10.2.2 Results

Four tests were made with a glitch amplitude of 5, 10, 20 and 40 times the rms signal noise (computed at the end of the record when the SMEC position is stable) with the default value for the three main parameters (**ScalMin = 1, ScalMax = 8, Holder in the range from -1.4 to -0.6, Correlation Threshold above 0.75**).

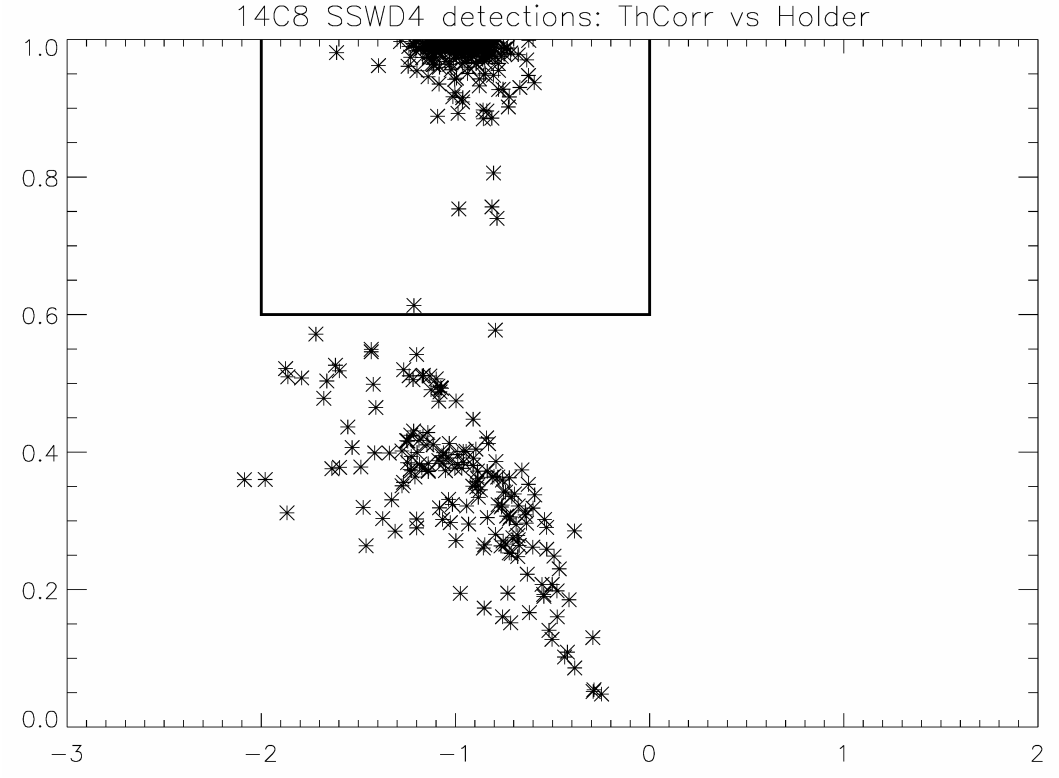

Figure 50a: Plot of SSWD4 detections in the Holder(X) – Correlation Threshold(Y) plane.

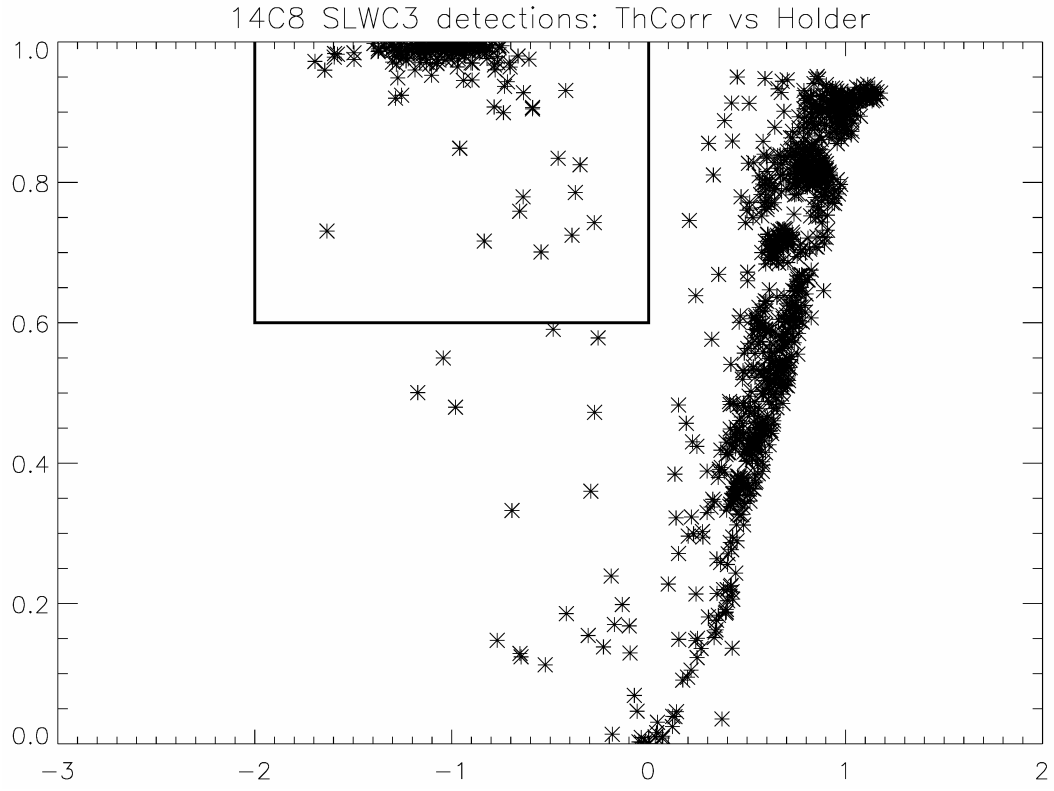

Figure 50b: Plot of SLWC3 detections in the Holder(X) – Correlation Threshold(Y) plane.

The glitch detection results (only the glitches found at the right position in the timeline are considered) are as follows for the two pixels:

SSWD4	Low	Medium	High	Percentage Low	Medium	High
5 x rms	0	2	17	0 %	3 %	6 %
10 x rms	0	28	201	0 %	45 %	67 %
20 x rms	4	45	278	22 %	73 %	93 %
40 x rms	9	53	291	50 %	85 %	97 %

SLWC3	Low	Medium	High	Percentage Low	Medium	High
5 x rms	0	0	2	0 %	0 %	1 %
10 x rms	0	3	102	0 %	5 %	34 %
20 x rms	0	9	186	0 %	15 %	62 %
40 x rms	0	12	204	0 %	19 %	68 %

3.10.2.3 Conclusions on these first results

These results are found to be close to our expectations. The large difference between the results for the two pixels can be understood taking into account the following two (conflicting) points:

- The true interferometric signal is much higher for SLWC3 than for SSWD4. The SNR (signal/amplitude of the central burst at ZPD compared to the rms noise) is 1600 for SLWC3 but only 120 for SSWD4. This means that it is harder to detect glitches for SLWC3 because the modulation due to real signal is higher (compared to the noise), and the glitches must be detected against a background that varies more than for SSWD4.
- As the signal frequency in SW detectors is twice the one in LW detectors, the ability to detect glitches is roughly a factor 2 worse for SW pixels than for LW ones. This is due to the higher slope in the interferogram modulation (i.e. frequency of modulations is higher for SW detectors).

Therefore we should expect a factor about 6 in glitch detection efficiency in favour of SSWD4 compared to SLWC3 which is roughly the kind of figure given in the results above.

Conclusion from these first tests:

- The ability to detect glitches is directly linked to the inverse of the maximum interferogram modulation (something which was expected).
- The conditions for in-flight observations (quite low interferometric signals after removal of telescope + SCAL emission) should be "fine" for a "good" glitch detection level. This is already quite encouraging.

3.10.3 Tuning module parameters

We used the same method that was used for the first analysis and ran the deglitching module with different sets of input parameters.

3.10.3.1 Tuning the ScalMin & ScalMax parameters

No significant change in the detection rate is achieved when tuning these two parameters.
So, in the following, we have kept them at their default values.

3.10.3.2 Tuning the Holder & Correlation Threshold parameters

We made module runs with an enlarged range of allowed values for these two parameters. The first runs were made on the same PFM4 300114C8 file, with no restriction on these two parameters.

The results are shown in Figures 50a (for SSWD4) and 50b (for SLWC3) for the run made with 300 manually added glitches with an amplitude of 20 times the noise level (as was done previously). In these figures, each « glitch detection » is marked as a star. With this large range of accepted values for the two parameters 460 and 1163 « detections » are found for SSWD4 and SLWC3 respectively (instead of 300 for each). This result implies, of course, a very large number of « false detections » and the pattern of scatter for such false detections differs very strongly for these two detectors.

A thorough analysis of the « detections » plotted in Figures 50a and 50b show that the major part of the « true » detections are achieved in the parameter area delimited by the thick line in these figures: i.e. Holder between -2 and 0, and Correlation Threshold between 0.6 and 1.

In the restricted area (of these two parameters) we find now 291 and 204 detections for SSWD4 and SLWC3 respectively, a result better than the one already noted in the first analysis (Phase 1) as given in Section 3.10.2 above. For easier inspection, these detections are plotted (as squares) in Figures 51a and 51b for SSWD4 and SLWC3 respectively.

An even deeper analysis of these selected « detections » indicates that some of the detected glitches are not found at the expected position in the timeline: they are detected a few samples (generally 5 at maximum) away from the expected one.

The list below gives the values for these detections with incorrect position:

Glitch #	Detector	Position	Holder	ThCorr	Constant1	Constant2	Noise	Deviation
29	SSWD4	12499.0	-0.937987	0.950700	6.90766	6.77396	5.07882	-1
31	SSWD4	13199.0	-0.594322	0.937100	6.24777	6.47707	5.07882	-1
60	SSWD4	23349.0	-0.811962	0.756601	6.91074	6.77586	5.07882	-1
92	SSWD4	35601.0	-0.982830	0.753692	7.39753	6.88005	5.07882	+1
174	SSWD4	65351.0	-0.884778	0.979244	6.47318	6.56533	5.07882	+1
183	SSWD4	68499.0	-0.804214	0.948036	6.83530	6.75437	5.07882	-1
184	SSWD4	68847.0	-1.21366	0.613236	5.88330	5.74676	5.07882	-3
299	SLWC3	6203.00	-0.834833	0.716342	5.93291	5.51480	5.05611	+3
328	SLWC3	25096.0	-1.10234	0.952670	5.43360	5.05747	5.05611	-4
348	SLWC3	33854.0	-0.589409	0.906651	5.76677	5.55104	5.05611	+4
417	SLWC3	74797.0	-0.459301	0.834252	5.41930	5.31268	5.05611	-3
427	SLWC3	80401.0	-0.347825	0.825067	5.97614	6.23533	5.05611	+1
437	SLWC3	85299.0	-0.637690	0.779412	6.54212	6.40293	5.05611	-1
438	SLWC3	85996.0	-1.63313	0.730625	6.31034	5.30928	5.05611	-4

Notes: Position should be 2000 + a number of time 350
Constant 1 & 2 are the values at origin of the linear regression fit and of the true ‘maxima line’ respectively (see text below)

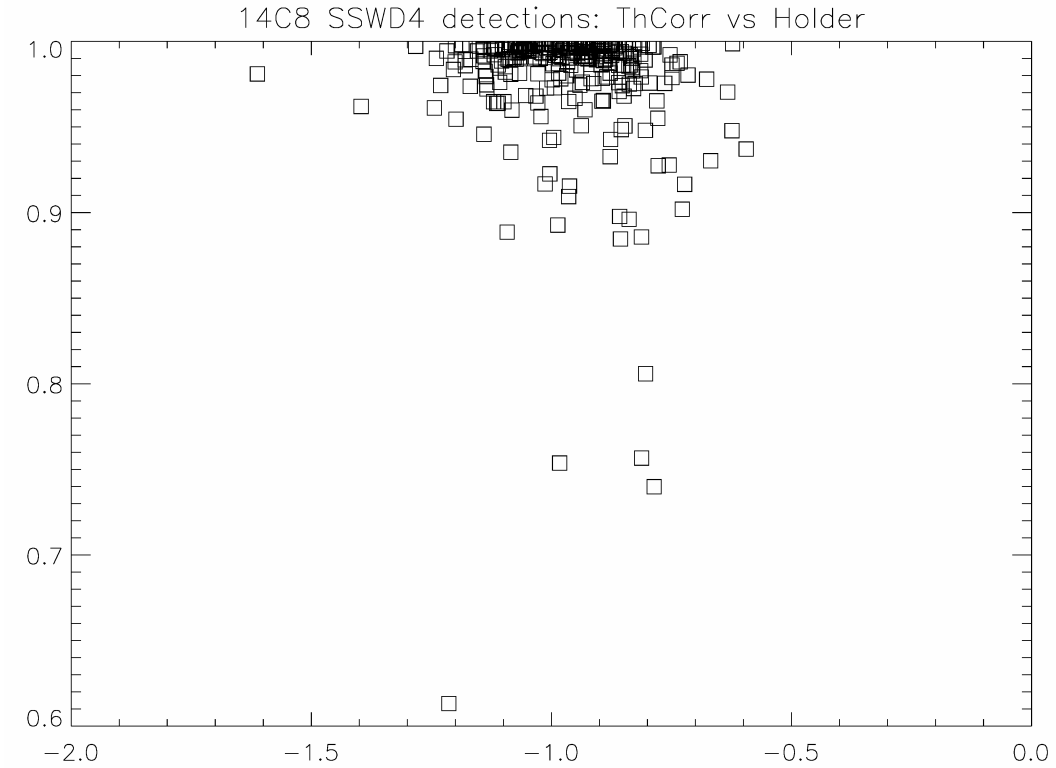

Figure 51a: Plot of SSWD4 detections in the restricted area of the Holder– ThCorr plane.

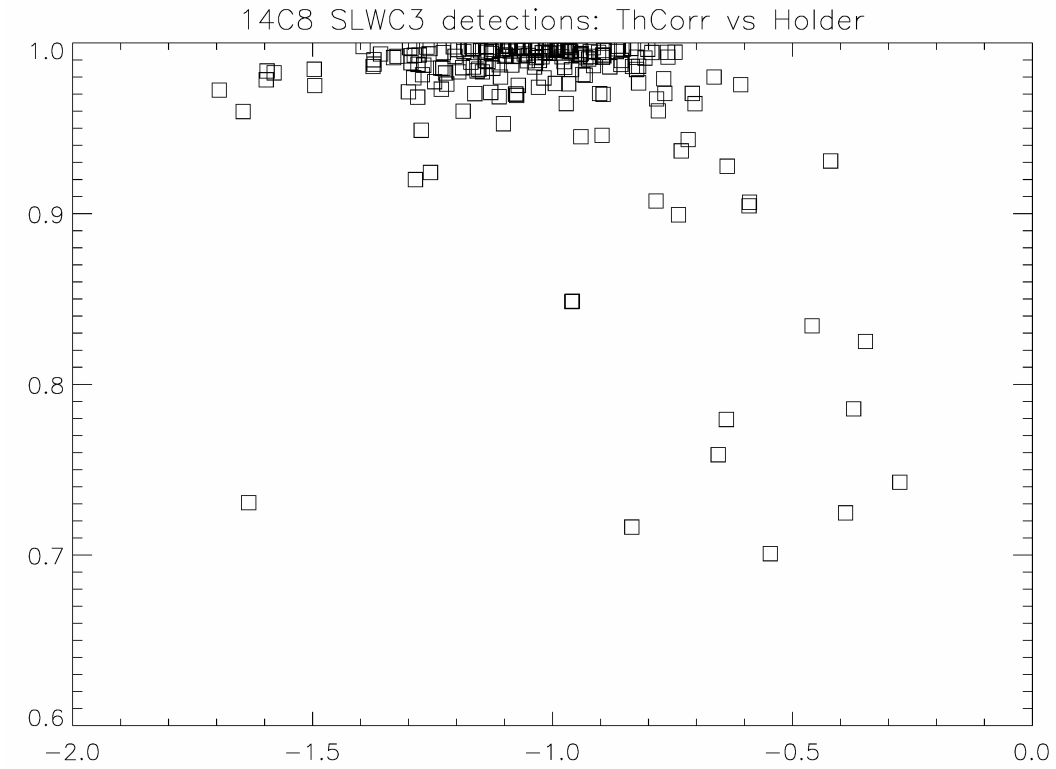

Figure 51b: Plot of SLWC3 detections in the restricted area of the Holder– ThCorr plane.

One can use the Holder and Correlation Threshold values from this list to locate these « shifted » detections in the Figures 51a and 51b. According to expectations, a very good detection should be located very close to the site with Holder = -1 and ThCorr = 1, and effectively, most of the incorrectly located detections are found away from this site.

We believe that the reason for detecting glitches at the incorrect position is due to the underlying modulated radiation signal where this signal has comparable (or even more) strength than the added glitches. The position separation increases as the underlying signal increases (see also Figure 55 below).

The detailed values of detection percentages for the three resolution modes (as already done in Phase 1, see above) are now, at this step, as follows:

					Percentage		
		Low	**Medium**	**High**	**Low**	**Medium**	**High**
SSWD4	20 x rms	10	53	291	56%	85%	97%
	40 x rms	13	57	295	72%	92%	98%
	200 x rms	17	61	299	94%	98%	99%

					Percentage		
		Low	**Medium**	**High**	**Low**	**Medium**	**High**
SLWC3	20 x rms	0	12	204	0%	19%	68%
	40 x rms	0	20	233	0%	32%	78%
	200 x rms	7	51	289	39%	82%	96%

Using a larger range for the values of these two parameters provides a significant increase in the overall detections. The LR mode (where the modulated signal is strong, compared to the noise level) yields the lowest detection rate: the results are not too bad for SSWD4 (however the few missing detections are all in this OPD part, even at low glitch strength) but, for SLWC3, we certainly need very strong glitches to be able to detect them (even with an amplitude of 200 times the rms noise level the detection rate in this part is less than 40%: this glitch amplitude is nevertheless 1/8 the maximum modulated signal at ZPD !).

With the restricted range (for Holder and Correlation Threshold parameters; see above) we made a run still using the PFM4 300114C8 file but, this time, with no manual addition of glitches and looking for all the detectors in the two arrays. The results are shown in Figure 52 where squares are detections of glitches from SSW detectors and stars for those from SLW detectors.

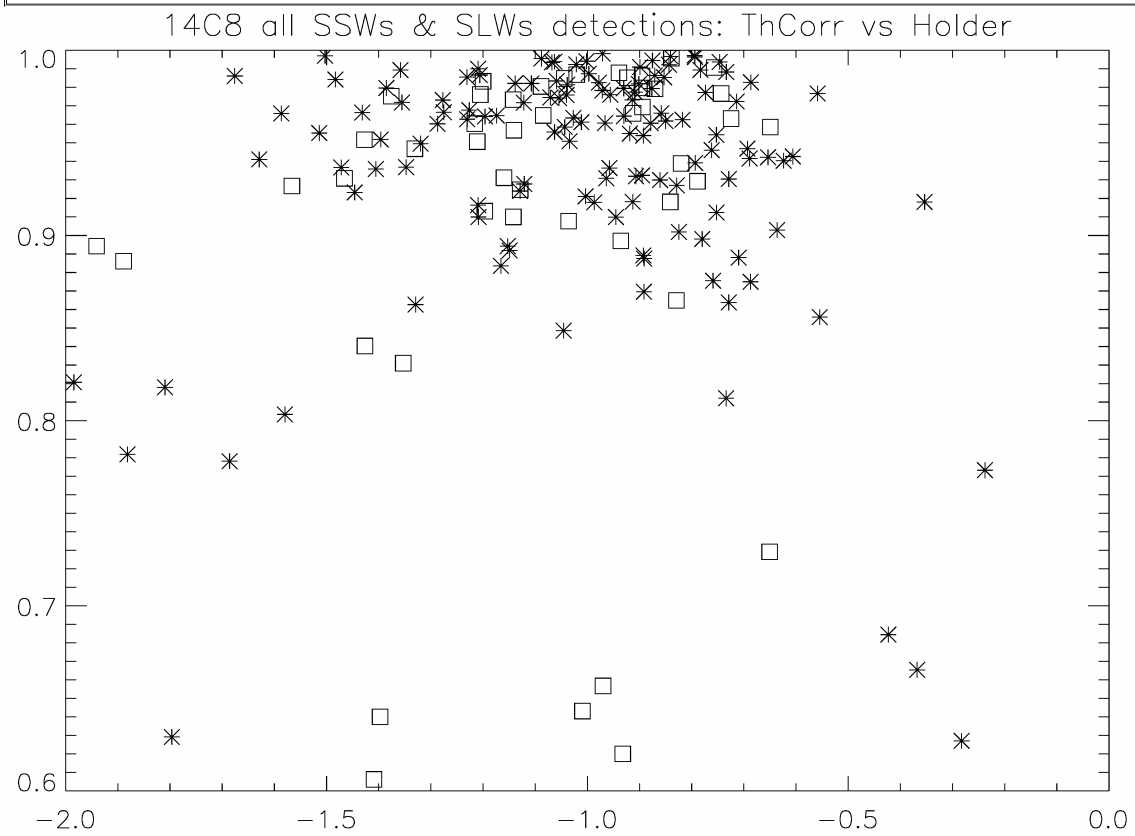

Figure 52: Plot of SSW detector (squares) and SLW detector (stars) detections for the whole PFM4 300114C8 run (with no addition of glitches) in the restricted area of the Holder– ThCorr plane.

We can see clearly from Figure 52 that the glitch detections are highly scattered (in the Holder-ThCorr plane), compared to Figure 51, but again with a definite concentration around the site for « good » detection. The total number of detections is 52 and 132 for SSW and SLW respectively: a large fraction of these are likely due to « false » detections however.

3.10.3.3 Tuning the Constant parameter

This « Constant » parameter, defined as the measured value of the « maxima line » at origin, is presently a « masked » one: in fact the present task considers only detections providing a « Constant » value greater than the « Noise » value. In the following we investigate the actual role and effect of the Constant over Noise ratio value.

This analysis is performed using histograms of the constant over noise ratio values. They are presented in Figures 53a to 53c for the runs with added glitches whose amplitudes are 20, 40 and 200 times the noise value respectively: the left panel is for SSWD4 and the right one for SLWC3. Figure 53d gives the result for the PFM4 300114C8 for the whole array (without addition of glitches): left panel for all SSW detectors and right one for SLW. In each panel the thicker line corresponds to the actual histogram divided by 10 as the plot has been clipped at 20 detections for better inspection of « side » detections.

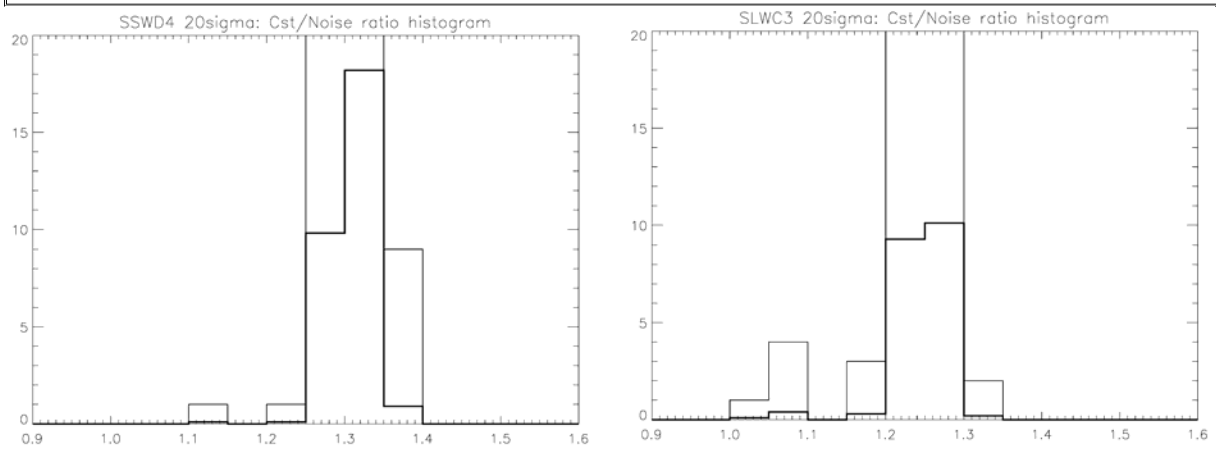

Figure 53a: Histogram of the constant/noise ratios for SSWD4 (left) and SLWC3 (right) detections [glitch amplitude = 20 times the rms noise value]

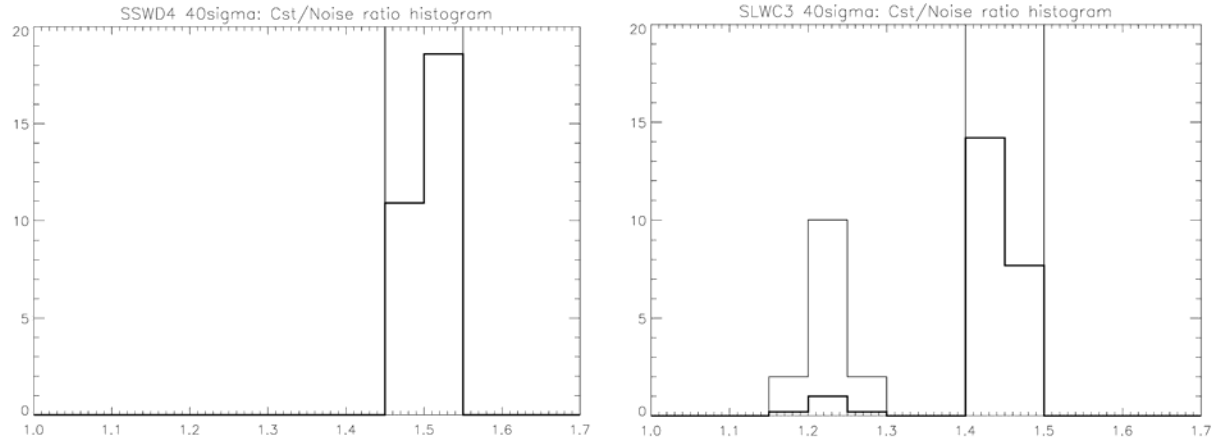

Figure 53b: Histogram of the constant/noise ratio for SSWD4 (left) and SLWC3 (right) detections [glitch amplitude = 40 times the rms noise value]

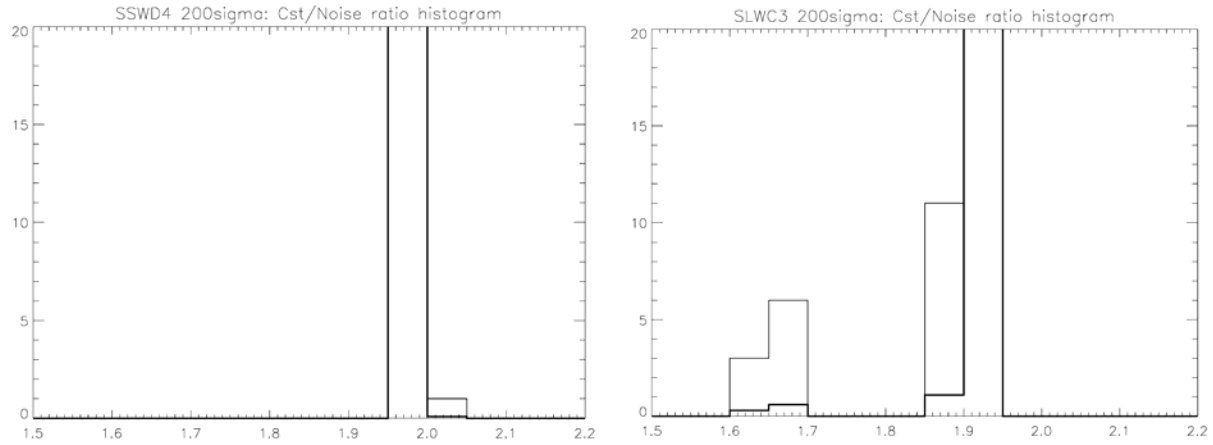

Figure 53c: Histogram of the constant/noise ratios for SSWD4 (left) and SLWC3 (right) detections [glitch amplitude = 200 times the rms noise value]

We can see from the figures 53a to 53c that, in general, the constant over noise ratio is well concentrated around a mean value (slightly higher for SSWD4 than for SLWC3) as one would have expected since all the glitches have the same amplitude. However, a few detections are found at

slightly lower ratio value and their number, as well as their departure, is bigger for SLWC3 than for SSWD4.

Glitches departing from mean value (left side of both histogram in Fig.53a) are given here:

	Glitch #	Position	Constant/Noise ratio
SSWD4	184	68847.0	1.13151
	207	76900.0	1.24869
SLWC3	299	6203.00	1.09072
	316	16350.0	1.18460
	328	25096.0	1.00027
	348	33854.0	1.09789
	375	49600.0	1.17992
	417	74797.0	1.05074
	430	81450.0	1.18874
	438	85996.0	1.05007

From this list of departing glitches we confirm that the separated secondary peak (on left side) in the histogram corresponds to glitches which are not found at the expected position. In fact, all already detected glitches which are away from more than 1 sample are within this left side secondary peak.

Figure 53d gives the results for the whole PFM4 300114C8 run concerning all the detectors with no manually added glitches. For both arrays the histogram peak corresponds to a mean constant over noise ratio value around 1.0 or less (detections with ratio values less than 1 have been rejected by the task). We note that the right side of the histogram peak have a significant smooth fall off (more pronounced for SLW than for SSW): however we should note that true glitches would yield values in this region. In fact four detections yield ratio values larger than 1.4 (SSWDP1 at position # 95751, SSWF4 at 97501, SLWB3 and SLWC2 at 24041) and visual inspection shows definite real glitches in these four cases. Therefore we may anticipate that most (or at least some) of the detections in the right side lobe may be due to real (but fainter !) glitches.

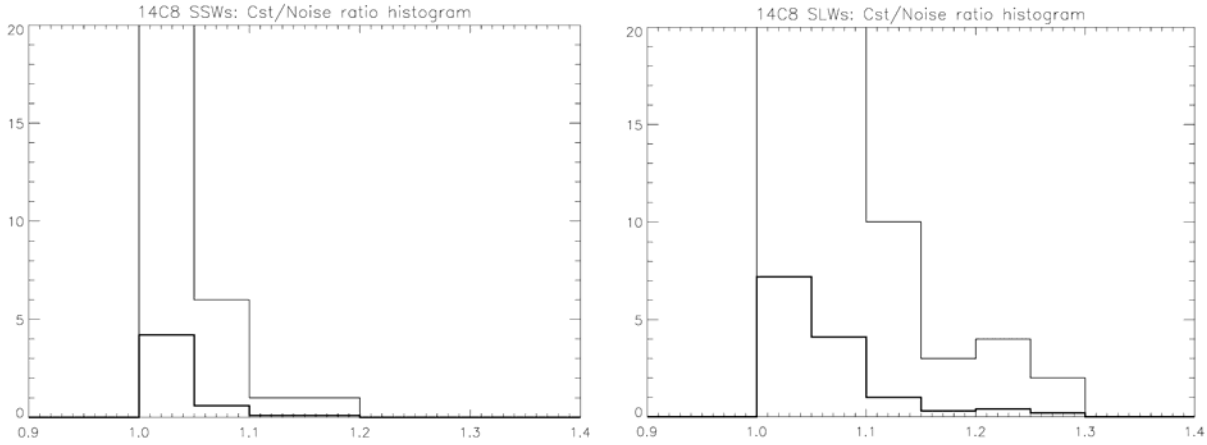

Figure 53d: Histogram of the constant/noise ratios for all SSW (left) and SLW (right) detections [without manually added glitches]

Figure 54 shows a composite version of the histograms already plotted in Figure 53 in two parts, one for the SSW detectors and the other one for the SLW detectors. In each panel the results for all the detectors from the PFM4 300114C8 run (with no added glitches) is plotted as the thick line. Results for SSWD4 and SLWC3 are also plotted as dashed and dotted lines, corresponding to addition of glitches with amplitude of 20 and 40 times the rms noise value respectively. Again, for clarity, the plots have been clipped at a value of 20.

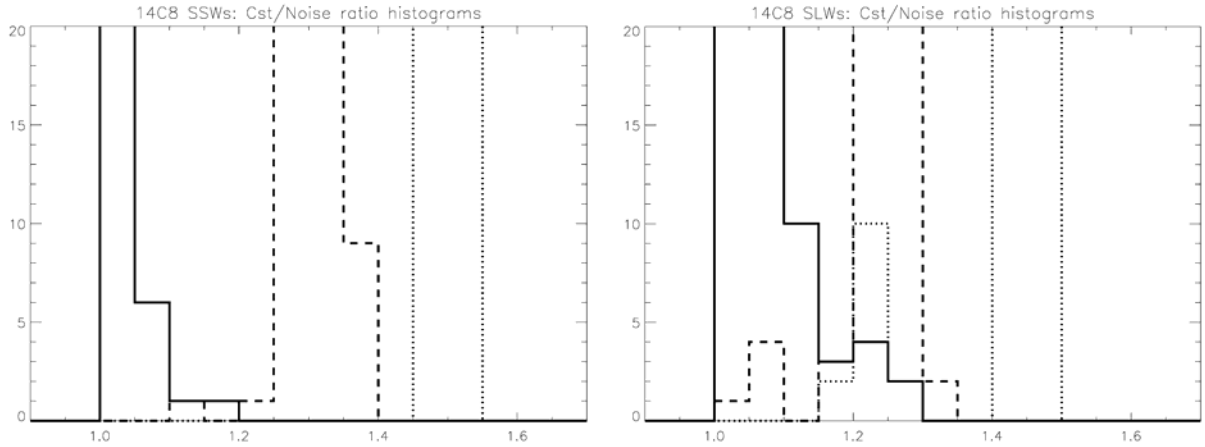

Figure 54: Composite histogram of constant/noise ratios **Left:** all SSW detectors with no added glitches and SSWD4 with added glitches **Right:** all SLW detectors with no added glitches and SLWC3 with added glitches (thick, dashed and dotted lines for no glitch, glitch amplitude of 20 and 40 times the rms noise respectively)

From Figure 54 it is worthwhile to note a first, but significant effect: as the strength of the glitch increases the histogram peak shifts toward higher values of the constant/noise ratio.
From the run made with large amplitude (200 times the rms noise) added glitches the histogram peak is found at a constant/noise ratio value close to 2 and 1.9 for SSWD4 and SLWC3 respectively. Therefore, as expected, this constant/noise ratio is a clear indicator of the glitch strength.

As concerns the SSW array, the overlap between the two histograms from « no added » and « 20 rms noise amplitude added » glitches yields a very small number of detected glitches (3 in total in the range of constant over noise ratio from 1.1 to 1.2). The overlap region for the SLW array (again from 1.1 to 1.2) is more populated: 13, 8 and 2 for « no added », « 20 rms noise amplitude added » and « 40 rms noise amplitude added » glitches respectively.

Figure 55 shows where the detected glitches with an incorrect position are located when the constant/noise ratio indicator is used (left panel) and when the corresponding normalized (i.e. current over maximum ratio) OPD is used (right panel).

From the left panel of Figure 55 we see that the constant/noise ratio is decreasing towards 1 as the position deviation gets larger. From the right panel we see that these « shifted » glitches appear mainly in the LowRes and MedRes part with the exception of 4 SLWC3 glitches falling at normalized OPD between 0.75 and 0.9 (i.e. in a region where the modulated signal is again high because of the « echo » feature) and of one SSWD4 glitch at normalized OPD around 0.45 (no clear explanation for this one !). From these indications we consider that these incorrect (in position) detections are likely due to the impact of the underlying modulated signal (much higher for SLWC3 than for SSWD4).

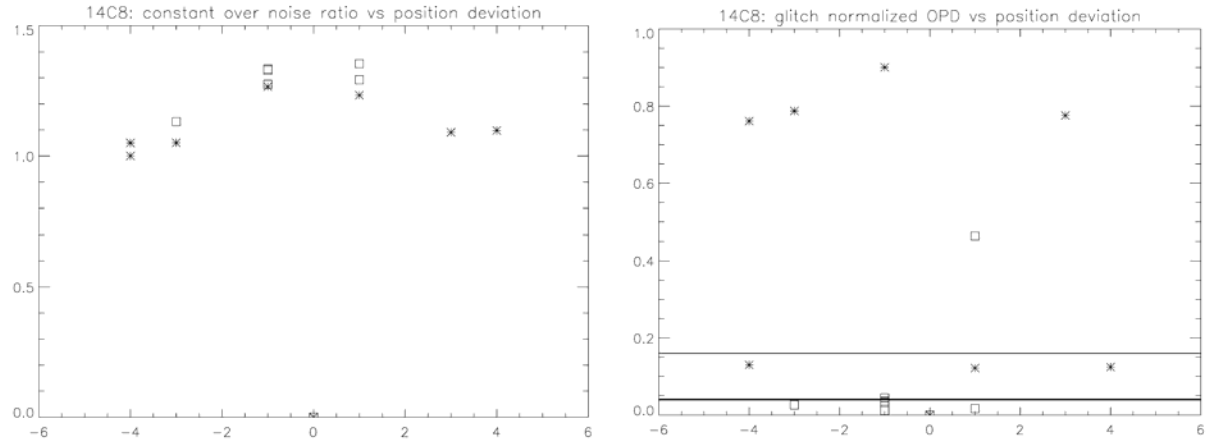

Figure 55: Plot of the constant/noise ratio (left panel) and of the normalized OPD (right panel) vs the position deviation of detected glitches (squares and stars for SSWD4 & SLWC3 detections)

3.10.3.4 Preliminary conclusions

As found in previous sections we consider that an extended range for both Holder values (now accepted from -2 to 0 instead of -1.4 to -0.6) and Correlation Threshold values (now from 0.6 to 1 instead of 0.75 to 1) gives satisfactory glitch detection results.

We have demonstrated that modifying the 'Constant over Noise ratio' Threshold value (in the build this threshold is set to 1) allows to eliminate a number of glitch detections which are either « false » or corresponding to faint glitch strength (i.e. roughly less than 20 times the rms noise value) without reducing significantly the number of « true » glitch detection with amplitudes about (or above) 20 times the rms noise value. The value for the Threshold to use (we suggest presently a value between 1.1 and 1.2) will be the result of a compromise : if we want to detect very faint glitches at the expense of detecting a larger number of « false » glitches then a value close to 1.1 should be selected ; on the other way if we want to minimize the number of « false » glitch detections at the expense of detecting a slighly lower number of glitch detections with amplitude about 20 times the rms noise value then a value close to 1.2 has to be selected. We see clearly different behaviour for SSWD4 and SLWC3: it is not clear presently if all detectors yield similar behaviour to the two studied above. We will certainly need to do this kind of arbitration with a much larger set of real measurements, and certainly to revise our selection when the in-flight data are available.

3.10.4 Results for Modelled Glitches

3.10.4.1 Method

The response of the detectors to a Dirac input has been modelled at the University of Lethbridge as the transfer function of their thermal responses and the electrical filter. The code, provided by Trevor, has been used to compute the impulse response function corresponding to the detector's time constant. According to a study made in Lethbridge, the time constants derived for each detector are found to be between 2 ms and 13 ms, so we used these two extreme cases to produce the impulse response function.

As the detector timelines are sampled at a frequency close to 80 Hz we have undersampled the output function from the code in order to match the SDT configuration. Five distinct origins have been used to describe the function: as the sampling interval is 12.5 ms, the 5 origin samples are separated by 2.5 msec. The results are shown in Figure 56: left and right panels correspond to time constant of 2 ms and 13 ms respectively.

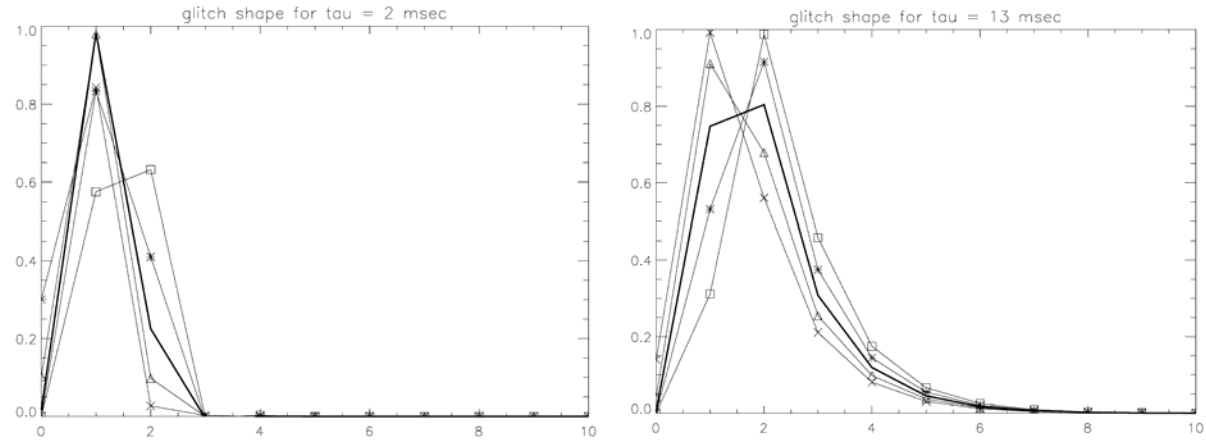

Figure 56: Plot of the impulse response function sampled at 80 Hz for a time constant of 2 ms (left) and 13 ms (right) the 5 origin samples are separated by 2.5 ms: symbols are squares, stars, thick line, triangles and crosses for the 1st one to the 5th one respectively.

These modelled glitches, with an amplitude of 20 times the rms noise level, were added to the SSWD4 and SLWC3 PFM4 300114C8 run SDT. The distribution of the glitches are performed in the same way as done in Phase 1 (see Section 3.10.2 above).

3.10.4.2 Results

The resulting glitch detection numbers by the module are given here below:

Modelled glitches	SSWD4	SLWC3	Total
GL02-1	293	199	492
GL02-2	294	203	497
GL02-3	294	205	499
GL02-4	294	206	500
GL02-5	292	203	495
Modelled glitches	**SSWD4**	**SLWC3**	**Total**
GL13-1	294	209	503
GL13-2	293	209	502
GL13-3	292	208	500
GL13-4	293	206	499
GL13-5	294	207	501

If the glitch detection is perfect we should end with a Total = 600

The corresponding results for Dirac's glitches with same amplitude (20 times the rms noise) were (see section 3.10.3.2):

SSWD4	SLWC3	Total
291	204	495

3.10.4.3 Preliminary conclusions

The detection rate with modelled glitches is very similar (or slightly better) than the one obtained with Dirac's glitches. This can be explained by the fact that the energy of the modelled glitch is larger than that of a Dirac type glitch. There is however some very marginal effect due to the way the sampling of the modelled glitch is achieved.

The list below gives for each of the module run the glitches which have been detected at an incorrect position:

Run	glitch#	position	deviation	Constant/Noise
GL02-1	59	23354.0	+3	**1.28**
GL02-1	90	34554.0	+3	1.15
GL02-1	294	3398.00	-3	1.01
GL02-2	60	23354.0	+3	**1.27**
GL02-2	295	3398.00	-3	1.05
GL02-2	321	19498.0	-3	1.04
GL02-2	350	33855.0	+4	1.05
GL02-3	60	23354.0	+3	**1.28**
GL02-3	295	3398.00	-3	1.08
GL02-3	322	19498.0	-3	1.06
GL02-3	351	33855.0	+4	1.07
GL02-4	60	23354.0	+3	**1.27**
GL02-4	295	3398.00	-3	1.07
GL02-4	303	6204.00	+3	1.04
GL02-4	323	19497.0	-4	1.06
GL02-4	352	33855.0	+4	1.07
GL02-5	293	3398.00	-3	1.05
GL02-5	301	6204.00	+3	1.02
GL02-5	320	19497.0	-4	1.03
GL02-5	348	33854.0	+3	1.05
GL13-1	196	72358.0	-3	1.02
GL13-1	217	79699.0	-2	**1.21**
GL13-1	324	18448.0	-3	1.07
GL13-1	385	52049.0	-2	1.11
GL13-2	216	79699.0	-2	**1.21**
GL13-2	323	18448.0	-3	1.08
GL13-2	384	52049.0	-2	1.11
GL13-3	322	18448.0	-3	1.03
GL13-4	60	23354.0	+3	**1.27**
GL13-4	294	3398.00	-3	1.09
GL13-4	322	18448.0	-3	1.05
GL13-5	60	23354.0	+3	**1.26**
GL13-5	353	33848.0	-3	1.02

The ‘deviation’ reported in the above list assume a correct position when the detected glitch is found shifted by one sample just after the actual position of the added glitch, since, as we can see in Fig. 56 the glitch is shifted with respect to the first sampled point in the glitch pattern. In fact, for a time constant of 2 ms the shift is mainly of 1 or 2 sampling intervals with a very few at 0, while for a time constant of 13 ms the shift is also mainly of 1 or 2 sampling intervals with a very few at 3. So, only glitch shift of less than 0 or greater than 3 sampling intervals should be considered as having an « incorrect » position and has been reported in the above list.

The net result is that, compared to the findings for the Dirac’s glitches where the shifted (by more than 1 sample point) detected glitches number was 1 and 5 for SSWD4 and SLWC3 respectively (see section 3.10.3.2), with a modelled glitch the rate of shifted detected glitches is about the same or slightly less : number of shifted glitches are 2/1/1/1/0/2/1/0/1/1 and 1/3/3/4/4/2/2/1/2/1 for the 10 runs and for SSWD4 and SLWC3 respectively. If we also include the selection on the « Constant/Noise ratio » parameter only a few glitches from these « incorrect » ones are still remaining (value in bold in the above list).

3.10.5 Conclusions for the « identify glitch » task

- From this analysis we can state that there is no significant change on the glitch detection rate when moving from Dirac type glitches to « real » modelled glitches whatever the detector time constant is (at least if the time constant do not exceed too much 13 ms).

- We found that the best detection rate is achieved with the following set of parameters:
  - ScalMin = 1 & ScalMax = 8
  - Holder values between -2. to 0
  - Correlation Threshold = 0.6

- The « Constant over Noise » ratio for each detected glitch has been shown to be a good indicator of the 'reality' of the glitch. Presently, in the build, only glitches which yield a ratio value greater than 1 are accepted as retained glitches. From this analysis we have given arguments to use as threshold for real detection a slightly higher value of this ratio: we believe that a threshold of 1.2 should provide better results, rejecting most of « false » or « faint » glitches without compromising the detection of (less fainter) « real » glitches. However the threshold value to be adopted may be discussed within the Validation Group and could be revisited with real data obtained with SPIRE when in transfer orbit to L2.

- A present estimate of the performance of the glitch identification task are as follows :
  - At OPDs where the detector noise exceeds (or is comparable with the strength of the modulated signal) the detection rate is close to 100% for glitches with amplitude about 20 (or above) times the rms noise.
  - At OPDs where the modulated signal exceeds the detector noise, then the detection rate is directly linked to the signal strength: from this analysis we derive a detection rate of about 40% for glitch amplitudes which are about 1/8 the signal strength, and a rate greater than 90% for glitch amplitudes comparable to the modulated signal strength.