TITLE: # Planck LFI – FS_SC: A Module to Handle a Tabulated Sorption Cooler Signal in the Flight Simulator

DOC. TYPE: OAT-TN

PROJECT REF.: PL-LFI-OAT-TN–022 **PAGE:** I of IV, 00

ISSUE/REV.: 0 **DATE:** July 20, 2001

Issued by	**Michele Maris** **LFI DPC Development Team**	**Date:** **Signature:**	**20 Jul 2001** _____
Agreed by	**F. PASIAN** **LFI Program Manager**	**Date:** **Signature:**	**December 12, 2001** _____
Approved by	**R.C. BUTLER** **LFI Program Manager**	**Date:** **Signature:**	 _____
Approved by	**N. MANDOLESI** **LFI Principal Investigator**	**Date:** **Signature:**	 _____

DISTRIBUTION LIST

Recipient	Company / Institute	E-mail address	Sent

CHANGE RECORD

Issue	Date	Sheet	Description of Change	Release

TABLE OF CONTENTS

1 SCOPE

Description of the package FS_SC to handle a tabulated sorption cooler signal in the framework of the PLANCK mission simulation.

1.1 LIMITS OF APPLICABILITY

The module uses a file containing simulated perturbations induced by the sorption cooler already tabulated for a given frequency channel. Then different horns of the same frequency channel will receive the same signal.

2 APPLICABLE/REFERENCE DOCUMENTS

2.1 APPLICABLE DOCUMENTS

[AD1] FORTRAN 90 Programming Guidelines for PLANCK/LFI
M. Maris
1999 March 8, Issue 0.1, LFI-OAT- 0002.01

[AD2] Effect of Sorption Cooler Temperature Variations on LFI Front-End
A. Mennella,
October 2000, PL-LFI-PST-TN-005, Issue 2.0

2.2 REFERENCE DOCUMENTS

2.3 ACRONYMS LIST

SC	Sorption Cooler
FS	Flight Simulator
Sys	Systematic
TOD(s)	Time Ordered Data

3 GENERALITIES

This module is designed to add the perturbation introduced by the Sorption Cooler to the LFI signal using a simulated sorption cooler signal provided by an external program and tabulated in an external file.

The Sorption Cooler (SC) [AD2] will introduces two main sources of fluctuation in the PLANCK/LFI signal, the main periods of these features being: 4000 sec and 600 sec. In addition a sub-minute perturbation whose leading harmonic has a of 17 sec is generated.

Some simulations show a peak-to-peak variation equivalent to 2 mK i.e. less than two quantization steps (at the nominal ?/q ~ 2).

The weight of such components and the amplitude in the not averaged signal, depends on the details of the mechanical assembly spacecraft, the sampling frequency, and the tolerance to SC failures. Consequently a detailed model will evolve in the near future. It is likely that the 4000 sec and 600 sec components will introduce nearly sinusoidal signals with about or less the same amplitude of the cosmological dipole. On the contrary, the 17 sec component amplitude is equivalent to about 1% of the main components amplitude, i.e. some hundredth of mK and its effect will be completely negligible for the sake of compression efficiency. If so, the SC will act as a time varying baseline which will not affect the signal compressibility more than the cosmological dipole, and likely will affect it less.

4 REQUIREMENTS

Basic requirements for the module are:

FS-SC-URD-1. The module shall handle a SC signal introduced in tabulated form.
FS-SC-URD-2. Tabulated signal shall be represented by a plain text file, with one column and one sample per line.
FS-SC-URD-3. Tabulated samples shall be not averaged in time.
FS-SC-URD-4. Tabulated signal shall be sampled at the same sampling rate of the frequency channel of choice.
FS-SC-URD-5. Tabulated signal shall be tabulated over a time long enough to assure covering of all the characteristic periods of the SC.
FS-SC-URD-6. The first sample of the Tabulated Signal shall simulated the first sample of the TOD of the full mission.
FS-SC-URD-7. Consecutive samples of the Tabulate Signal shall enter consecutive samples in the TOD.
FS-SC-URD-8. Tabulated signal shall be repeated cyclically to enter TODs longer than them.
FS-SC-URD-9. Damages in one or more units of the SC shall be simulated by different Tabulated Signals.

5 PACKAGE STRUCTURE

The package is written in FORTRAN 90 according to the conventions described in [AD1].

The package is structured in three main levels:
1. The library module: `fs_sc_tab.f90`.
2. The package environment: `fs_sc_tab_env.f90`.
3. The example driver used for testing: `fs_sc_tab_test.f90`.
4. The interface with the Flight Simulator

The library module is designed according to the principles of OOP in F90. The library module assures maximal reusability of the software. It carries the definition of the *Sorption Cooler* object plus the related methods to handle it. It has to carry only general purpose methods.

The package environment holds definitions and methods required to integrate the package inside an application. This module is less standardised, it is designed to simplify the integration but it may be modified whenever required during the integration phase introducing methods that are not rigorously standardised or of general use.

The example driver used for testing documents how to use the SC module and its environment and gives some simple tests to validate it.

The interface with the FS is based on the methods used in the `fs_sc_tab_test.f90` and the package environment.

6 LIST OF ROUTINES AND SERVICES

Routines and services are listed and described here.

6.1 UNIT: FS_SC_TAB.F90

6.1.1 REAL FUNCTION FS_SC_TAB_T_CALC(T,T0,SC)

Given the time t (from the mission start) computes the SC signal by linear interpolation
SC is a keyword for a structure of type T_SC_TAB, if not set, the GLOBAL SC_TAB is used instead
t0 is used to add a time shift to the SC signal, if omitted t0=0. is assumed

6.1.2 REAL FUNCTION FS_SC_TAB_CALC(NS,SC)

Given the sample number Ns returns the SC signal in the table
SC is a keyword for a structure of type T_SC_TAB, if not set, the GLOBAL SC_TAB is used instead
Ns is the sample number, if omitted SC%LAST is incremented by 1 (circularly) and the resulting element of the SC table is given in output

6.1.3 SUBROUTINE FS_SC_TAB_CREATE

Parameters: This, Name, TSampling, Half, TTotal0, NSAMP0, INUnit
Creates an T_SC_TAB object, if omitted the GLOBAL_T_SC_TAB variable is created
Beware: it uses the UNIT 10 if INUnit is omitted

6.1.4 SUBROUTINE FS_SC_TAB_COPY(THAT,THIS)

Copies the T_SC_TAB object *This* into the equivalent object *That* allocating the needed memory space.

6.1.5 SUBROUTINE FS_SC_TAB_DESTROY(THIS)

Destroys (deallocate) the data contained in the T_SC_TAB structure

6.1.6 SUBROUTINE FS_SC_TAB_SHOW(THIS)

This subroutine displays the scalar content of an T_SC_TAB object. Useful for debug purposes

6.2 UNIT: FS_SC_TAB_ENV.F90

6.2.1 SUBROUTINE FS_SC_TAB_ENV_GET(UNIT,NAMELIST,INIT,VERBOSE)

Gets from already open UNIT the data related to SC_TAB environment
If UNIT is omitted then data are stored from standard input.
If NameList = T or it is omitted reads data using a fixed sequence otherwise reads data using a namelist structure named: TABULATED_SORPTION_COOLER the global variable GLOBAL_SC_TAB is filled.
If INIT = .false. data are readed but not given in output
If VERBOSE .ne. 0 then a message is shown with the main parameters
Beware: Without INIT=.true. data are readed but not passed

6.2.2 SUBROUTINE FS_SC_TAB_ENV_CREATE(CIRCLESIZE,ECHO_UNIT)

Initialises auxiliaries data structures for the Sorption Cooler generation. It uses parameters already read by FS_SC_TAB_ENV_GET.
The parameter CircleSize is the number of samples in a circle.
Short messages are generated and whenever required are written in the unit ECHO_UNIT.

6.2.3 SUBROUTINE FS_SC_TAB_ENV_DESTROY()

Kills all the allocatable data structures and close all the units associated with the FS_DIPOLE_ENV

7 THE FLIGHT SIMULATOR INTERFACE

This section describes how the module is used inside the FS.

To link to the flight simulator introduce the following declarations in the declaration part:

```
!!!! SORPTION COOLER TABULATED
!
! SORPTION COOLER TABULATED libraries
! By M. Maris
!
    USE FS_SC_TAB
    USE FS_SC_TAB_ENV
```

Parameters required for the initialisation are read calling:

```
! Reads the sorption cooler tabulated related parameters from standard input
! using NameList method

        CALL FS_SC_TAB_ENV_GET(NAMELIST=.true.,Verbose=11)
```

The environment is then initialised using even other FS parameters:

```
!
!   ===============================================================
!   Initializes the SC_TAB environment with the required parameters
!   ===============================================================
    print*,'Initializes the SC_TAB environment with the required parameters'
    CALL FS_SC_TAB_ENV_CREATE(ECHO_UNIT=16,CircleSize=dimpsi+1)
```

After the spacecraft pointing is generated by the FS the dipole contribution is generated and added to the signal:

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Summs the sorption cooler contribution
!
! Note how this computation is performed for each noise point
!
! No arguments are passed, so that the SC contribution is automatically
! recovered from:
!  1) the GLOBAL_SC_TAB
!  2) the element of the tab next to the one previously readed
!
                 if (iWantSorptionCoolerTabulated) then
                    TSorption(ipsi) = fs_sc_tab_calc()
                    totsignal = totsignal + TSorption(ipsi)
                 endif
```

The simulated signal is then processed as usual.

This is to help debug or to have a separated sorption cooler:

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! If required writes out the SC signal (for all the circles per pointing)
            if (iWantWriteSorption) then
               print*,'Writing SC'
               write(IOUnit_Sorption) (real(TSorption(ipsi)),ipsi=0,dimpsi)
            endif
```

Before to leave destroy the environment to close files:

```
!
!   ==============================
!   Destroyes the SC environment
!   ==============================
    CALL FS_SC_TAB_ENV_DESTROY()
```

8 THE PARAMETERS FILE

A typical parameters file is as follow:

```
&TABULATED_SORPTION_COOLER
        iWantSorptionCoolerTabulated = T
        SCName                       = 'SC/TOD100_6_equal.dat'
        TSampling                    = 0.00925926
        Half                         = T
        TTotal0                      = 4000.
        NSAMP0                       = 432001
        iWantWriteSorption           = F
        FName_Sorption               = '100GHZ/Q12MK_ALL/sorption.bin'
/
```

9 APPENDIX: FILES USED BY THE MODULE

The following files of Tabulated SC signals have been generated by A.Mennella (private communication, 2001 May 15)

Files are TOD describing the signal oscillations for LFI at 30 and 100 GHz due to the Sorption Cooler (SC) oscillations.

Table of SC Tabulated Signal Files

Filename:	Freq. (GHz)	Sampling Time (sec)	Units	TOD Length (sec)	Notes:
TOD30_6_equal.dat	30	0.0305556	Kelvin	4000	SC with 6 equal compressor beds
TOD30_1_bad.dat	30	0.0305556	Kelvin	4000	SC with 5 equal compressor + 1 non homogeneous compressor bed
TOD30_3_bad.dat	30	0.0305556	Kelvin	4000	SC with 3 equal compressor + 3 non homogeneous compressor bed
TOD100_6_equal.dat	100	0.00925926	Kelvin	4000	SC with 6 equal compressor beds
TOD100_1_bad.dat	100	0.00925926	Kelvin	4000	SC with 5 equal compressor + 1 non homogeneous compressor bed
TOD100_3_bad.dat	100	0.00925926	Kelvin	4000	SC with 3 equal compressor + 3 non homogeneous compressor bed