

spi_gain_hist

User Manual

Version 1.5.0
25 September 2003

Jürgen Knödseder
Centre d'Etude Spatiale des Rayonnements
knödseder@cesr.fr
<http://www.cesr.fr/~jurgen/index.html>

Note to the user

This software has been written to analyse data of the SPI telescope onboard INTEGRAL. Particular care has been taken in making the software user friendly and well documented. If you appreciated this effort, and if this software and User Manual were useful for your scientific work, the author would appreciate a corresponding acknowledgment in your published work.

Contents

1	Introduction	1
2	Getting started	1
3	Parameter file	2
4	Interface definition	4
5	Error codes	4

1 Introduction

The executable `spi_gain_hist` builds calibration spectra in the format `SPI.-CALI-DSP` from SPI photon data. Calibration spectra may be built from single events (**SE**), PSD events (**PE**), multiple events (**ME**), onboard spectra, or combinations of those three types.

`spi_gain_hist` is an ANSI C++ executable that has been developed under ISDC support platform 6.3 and requires the libraries `spi_psdlib` and `spi_toolslib`.

2 Getting started

Before installing `spi_gain_hist`, make sure that the ISDC support platform 6.3 or higher is installed on your system, and that the libraries `spi_psdlib` and `spi_toolslib` are installed.

After downloading the `spi_gain_hist.tar.gz` file, step into a directory that should hold the distribution, move the `spi_gain_hist.tar.gz` file into this directory and type:

```
$ gunzip spi_gain_hist.tar.gz
$ tar xvf spi_gain_hist.tar
```

The first command uncompresses the distribution file, the second unpacks the files.

Before configuration, the distribution needs to be reset to a clean state. To do this, type

```
$ make distclean
```

Then, configure the distribution. It is assumed here that you have previously installed the ISDC support platform, thus you should type

```
$ ~/bin/ac_stuff/configure
```

Finally, build the distribution by typing

```
$ make global_install
```

3 Parameter file

```
#####
#
#           Centre d'Etude Spatiale des Rayonnements           #
#           (in collaboration with ISDC)                       #
#
#   SPI Gain Correction Event Binning (for energy calibration) #
#
# -----#
#
# File:      spi_gain_hist.par #
# Version:   1.5.0             #
# Component: CA                #
#
# Author:    Juergen Knoedlseder #
#            knodlseder@cesr.fr  #
#            CESR                #
#
# Purpose:   Parameter file of the SPI gain correction event binning #
#            executable #
#
# History:   1.5.0 25-Sep-2003 First release (Rev. 5) #
#
#####
#
# Input/Output DOLs
#=====
inGrpDOL, s, q, "swg_prp.fits[1]",,, "Input Group (SWG/OG/IDX)"
inGtiDOL, s, q, ,,, "DOL of a Good Time Interval"
outDOL,   s, q, "spec.fits",,, "Calibration spectra filename"
#
# OBT limits
#=====
minOBT, s, q, "",,, "Event usage minimum OBT"
maxOBT, s, q, "",,, "Event usage maximum OBT"
append, b, h, no,, "Append minimum OBT to last results ?"
slice,  b, q, no,, "Split time interval in constant ONTIME intervals ?"
nopart, b, q, no,, "Skip partial time intervals ?"
ontime, r, q, 0.0,, "Constant ONTIME slice (seconds)"
#
# Task parameters
#=====
useSE,   b,h, yes,, "Use SE to build spectrum ?"
usePE,   b,h, yes,, "Use PE to build spectrum ?"
useCRVE, b,h, no,, "Use CRVE to build spectrum ?"
useME,   b,h, no,, "Use ME to build spectrum ?"
useAON,  b,h, no,, "Use ACS-ON spectra to build spectrum ?"
useAOFF, b,h, yes,, "Use ACS-OFF spectra to build spectrum ?"
#
# ISDC Standard Parameters
#=====
clobber, b, h, yes,, "Overwrite existing output data ?"
```

`verbose, i, h, 2,0,4,"Information logging level"`

The parameters have to following meaning:

- **inGrpDOL** specifies the input group from which the calibration spectra should be built. The input group may be either a science window group, an observation group, or an index file.
- **inGtiDOL** specifies a Good Time Interval (GTI) DOL that may optionally be used for time interval selection. If an empty string is specified, no GTI DOL will be used for time interval definition. Instead the parameters **minOBT**, **maxOBT**, **append**, **slice**, **nopart** and **ontime** will be used for GTI interval definition. If a GTI DOL has been specified, these parameters will be ignored!
- **outDOL** specifies the filename of the result spectra. This result file will contain both the **SPI.-CALI-GTI** and the **SPI.-CALI-DSP** extensions that define a calibration spectra. Multiple calibration spectra can be added to the same file, where the calibration pointing identifier **PTID_CAL** is automatically incremented by one for each new spectra set. If a new file is created, **PTID_CAL** is reset to 0.
- **minOBT** specifies the minimum OBT limit of the events that should be used for performance determination. The OBT format is a character string. Leading 0 may be omitted. If the character string is empty, or if any non-number character is specified (such as "no" for example), no minimum OBT limit is applied (and data accumulation starts with the first event in the input group).
- **maxOBT** specifies the maximum OBT limit of the events that should be used for performance determination. The OBT format is a character string. Leading 0 may be omitted. If the character string is empty, or if any non-number character is specified (such as "no" for example), no maximum OBT limit is applied (and data accumulation stops with the last event in the input group).
- **append** this parameter is not used so far.
- **slice** specifies if the input group should be "sliced" into time frames of constant **ONTIME** (the **ONTIME** is the time, specified in seconds, during which SPI science data were accumulated and made available to the observer).
- **nopart** (optional) if **slice = yes**, specifies if partial time slices, i.e. time slices with durations that are shorter than the requested **ONTIME**, should be skipped. Partial time slices may occur at the end of a data stream, and to assure a uniform quality of the calibration spectra it is recommended to set this parameter to **yes**. Together with **append = yes**, re-execution of **spi_gain_hist** at a later time will append new time slices that start with the OBT of the last appended time slice.
- **ontime** (optional) if **slice = yes**, specifies the **ONTIME** duration of each time slice. Note that the last time slice has generally an effective **ONTIME** that is shorter than the specified value, since in general, the available **ONTIME** is not an integer multiple of the value specified by **ontime**.
- **useSE** specifies if single events (SE) should be used for spectra building.
- **usePE** specifies if PSD events (PE) should be used for spectra building.
- **useCRVE** specifies if PSD curves (CRVE) should be used for spectra building. If **usePE** and **useCRVE** are both set to **yes**, PSD curves are only collected for a time slice if no PSD events are found. Otherwise, one would double-count PSD events since normally each PSD curve has a corresponding PSD event in the telemetry. The mode **useCRVE=yes** is useful for the analysis of SPI PSD calibration data where only PSD curves but no PSD events are found in the telemetry.
- **useME** specifies if multiple events (ME) should be used for spectra building. Multiple events are added as so-called **broken ME**, which means that the partial energy that was registered in a given detector is stored in the spectra.

- **useAON** specifies if ACS ON onboard spectra should be used for spectra building. If combined with photon data, spectra will be grey-filtered in order to adjust the ONTIME.
- **useAOFF** specifies if ACS OFF onboard spectra should be used for spectra building. If combined with photon data, spectra will be grey-filtered in order to adjust the ONTIME.
- **clobber** specifies if output datasets should be overwritten.
- **verbose** verbose level.

4 Interface definition

spi_gain_hist extracts photon data of ISDC level PRP from an input group and creates calibration spectra in the SPI PHA energy format (AFEE energy channels). The input group may be either a single science window group, an observation group, or an index file.

Calibration spectra are built for each detector separate. Each detector produces two spectra, one for the high gain range (or PHA0) and one for the low gain range (or PHA1). Thus, multiples of 38 spectra are produced by **spi_gain_hist**.

spi_gain_hist writes the spectra information into two data structures:

- **SPI.-CALI.-GTI** which contains the OBT start and stop values for each set of 38 spectra, as well as the calibration pointing ID **PTID_CAL** and the **ONTIME** (i.e. the net integration time for the spectra).
- **SPI.-CALI.-DSP** which holds the spectra themselves, linked to the **SPI.-CALI.-GTI** by the calibration pointing ID.

If no calibration spectra exist prior to calling **spi_gain_hist**, the task will create the corresponding FITS file and assign a first calibration pointing ID (**PTID_CAL**) of 0 to the first set of spectra. If calibration spectra are added to an existing set of spectra, the calibration identifier will be incremented by one.

The input group may be sliced in time intervals of constant **ONTIME** to guarantee a homogenous exposure for all calibration spectra. In this case, **spi_gain_hist** searches for time intervals of constant **ONTIME** in the input group. Note that **ONTIME** means really the net exposure time, hence data gaps do not contribute to the **ONTIME** determination (this means that the OBT intervals are not necessarily of equal length).

If science mode changes occur during the input group, **spi_gain_hist** will integrate properly over the mode change boundaries, assuring a constant **ONTIME** (this is not completely true for the PSD calibration mode, since in this case no event data are produced but only PSD curves, yet the time is accounted for as valid **ONTIME**; a future version of **spi_gain_hist** may correct for this special case).

Note that the last time interval has not necessarily the requested **ONTIME**, but only the remaining **ONTIME** that was left.

5 Error codes

The executable **spi_gain_hist** may stop with the following error codes:

```
SPI_GAIN_HIST_ERROR_MEM_ALLOC -212000 // Memory allocation failure
```

They have the following meaning:

- **SPI_GAIN_HIST_ERROR_MEM_ALLOC** : the allocation of dynamical memory has failed. Probable your system resources are too limited to run this task.

In addition, all errors that may occur in calls to ISDC support functions (such as for example DAL, RIL or PIL) are forwarded. Please consult the ISDC web pages for getting information about these error codes.