

spi_psd_postprocess

User Manual

Version 1.3.1
21 January 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	3
5	Algorithm	4
6	Error codes	4

1 Introduction

The executable `spi_psd_postprocess` is part of the SPI scientific analysis preparation software component (SAP). It performs post-processing of the PSD discrimination based on the onboard pulse shape fitting results that are transmitted in the scientific event word of PSD events (**PE**) and PSD curves (**CRVE**). The pulse shape discrimination criteria for post-processing are defined by the PSD discrimination parameters that are stored in a table of type `SPI.-DISC-PSD`. This table describes the energy dependent discrimination threshold for all possible pulse shape fitting results. The optimum PSD discrimination parameters for a given observation may be obtained using the task `spi_psd_optimise`, hence `spi_psd_postprocess` has to be executed after this task. Post-processing information is requested for event spectra building, hence `spi_psd_postprocess` has to be executed before `spihist`.

`spi_psd_postprocess` is written in the ANSI C++ language and has been developed under ISDC support platform 5.1. It requires `spi_psdlib` version 2.0.0 or higher and `spi_toolslib` version 2.0.0 or higher.

2 Getting started

Before installing `spi_psd_postprocess`, make sure that the ISDC support platform 5.1 or higher is installed on your system, and that you have installed the libraries `spi_psdlib` version 2.0.0 and `spi_toolslib` version 2.0.0 or higher.

After downloading the `spi_psd_postprocess.tar.gz` file, step into a directory that should hold the distribution, move the `spi_psd_postprocess.tar.gz` file into this directory and type after the UNIX prompt `$` (don't type this prompt):

```
$ gunzip spi_psd_postprocess.tar.gz
$ tar xvf spi_psd_postprocess.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

```
$ ${ISDC_ENV}/bin/ac_stuff/configure
```

Finally, build the distribution by typing

```
$ make global_install
```

To perform a unit test, type

```
$ make test
```

Make sure that the test data `spi_test_data-1.0.tar.gz` are available at your site (they should reside in a directory whose name is defined by the `ISDC_TEST_DATA_DIR` environment variable).

3 Parameter file

```
#####
#
#           Centre d'Etude Spatiale des Rayonnements           #
#           (in collaboration with ISDC)                       #
#
#           SPI PSD discrimination post-processing             #
#
# -----#
#
# File:      spi_psd_postprocess.par                          #
# Version:   1.3.1                                           #
# Component: SAP                                             #
#
# Author:    Juergen Knoedlseder                             #
#            knodlseder@cesr.fr                              #
#            CESR                                             #
#
# Purpose:   Parameter file of the SPI PSD post-processing   #
#
# History:   1.3.1 21-Jan-2003 First ISDC delivery (Rev. 3)  #
#
#####
#
# The input DOLs/filenames
#=====
ingrpDOL,s,ql, "og_spi.fits[GROUPING]",,, "Input Observation Group DOL or filename"
indisDOL,s,ql, "psd_disc.fits[SPI.-DISC-PSD]",,, "PSD discrimination DOL or filename"
#
# The output DOLs/filenames
#=====
outgrpDOL,s,ql, "og_spi_psd.fits[GROUPING]",,, "Output Observation Group DOL or filename"
#
# Standard parameters
#=====
clobber,b,h, yes,, "Overwrite existing output data ?"
verbose,i,h, 2,0,3, "Information logging level"
```

Instead of specifying complete DOLs (Data Object Locations), which are composed of a filename plus the data structure extension (HDU), `spi_psd_postprocess` accepts also simple filenames and adds the appropriate data structure extensions. This means that **specified data structure extensions are ignored**.

The parameters have the following meaning:

- `ingrpDOL` (optional) specifies the DOL or filename of the input Observation Group (HDU `[GROUPING]`) for which the PSD post-processing should be performed. The input group has to be of level `COR` and gain correction has to have being performed before executing the task (using the `spi_gain_corr` executable).

If an output Observation Group has been specified (parameter `outgrpDOL`), the specification of this parameter is optional. If the parameter is left blank, the output Observation Group will then be used as input Observation Group. Otherwise, the input Observation Group will be copied into the output Observation Group.

- `indisDOL` (optional) specifies the DOL or filename of the PSD discrimination parameters (HDU [SPI.-DISC-PSD]).
If a [SPI.-DISC-PSD] element exists already in the input Observation Group, this element will be replaced by the specified DOL in the output Observation Group. Otherwise, the specified DOL will be attached to the output Observation Group. If left blank, it is assumed that a [SPI.-DISC-PSD] element exists already in the input Observation Group (this is the normal case if the executable `spi_psd_optimise` has been executed before). If no such element is found, however, the task execution is aborted with an error message.
- `outgrpDOL` (optional) specifies the DOL or filename of the output Observation Group (HDU [GROUPING]). The output Observation Group will be a copy of the input Observation Group. If an input Observation Group has been specified (parameter `ingrpDOL`), the specification of this parameter is optional. If the parameter is left blank, the input Observation Group will then be used as output Observation Group.
- `clobber` specifies if existing output data structures should be overwritten or not. If `yes` is specified, the executable will notify the user about the deletion of any file. If `no` is specified and the executable attempts to overwrite existing data (e.g. an existing output Observation Group), the task will exit with an error message. **An exception to this rule is the COR data structure for PSD events and PSD curves, for which the column PSD_CORFLAG is always overwritten since it existed already before `spi_psd_postprocess` has been called.**
- `verbose` specifies the verbose level of the executable. For `verbose=0`, no information will be logged in case of an error. For `verbose=1`, only errors will be logged, while for `verbose=2` also actions (such as DOL detachments and attachments and DOL deletion) will be logged. `verbose=3` provides a detailed report about the SPI PSD post-processing.

4 Interface definition

`spi_psd_postprocess` performs post-processing (i.e. onground pulse shape discrimination) for all PSD events (PE) and PSD curves (CRVE) that it finds in the input Observation Group. To perform the discrimination, `spi_psd_postprocess` extracts PSD discrimination parameters from the SPI.-DISC-PSD data structure that is either already attached to the input Observation Group, or specified by the task parameter `indisDOL`. Note that there may be several sets of PSD discrimination parameters in the SPI.-DISC-PSD data structure that correspond to different time intervals, and in this case `spi_psd_postprocess` correctly performs time dependent PSD discrimination.

On output, `spi_psd_postprocess` fills the PSD_CORFLAG column of the corrected PSD event and PSD curve data structures. Note that `spi_psd_postprocess` requires the existence of these data structures. In any case, `spi_psd_postprocess` requires calibrated energy information (in keV) for each event to apply the energy dependent pulse shape discrimination, and since the calibrated energy is stored in the same table as the corrected PSD flag, the PSD_CORFLAG column should exist.

`spi_psd_postprocess` may be applied repeatedly to the same Observation Group since it always replaces (or updates) the PSD_CORFLAG column of the group. This feature is useful for studying the impact of different PSD discrimination parameters on the scientific results.

The PSD_CORFLAG column may take the following values (see also the `spi_psdlib` User Manual):

- 0 (PSDLIB_POSTPROCESS_SINGLE): Single-site event (i.e. background)
- 1 (PSDLIB_POSTPROCESS_MULTIPLE): Multiple-site event (i.e. gamma-ray photon)
- 128 (PSDLIB_POSTPROCESS_ERROR_NO_PARA): No discrimination parameters exist to perform post-processing

- 129 (`PSDLIB_POSTPROCESS_ERROR_BAD_TTP1`): TTP1 value of the event/curve has no correspondent entry in the discrimination table (probably a wrong discrimination table has been specified)
- 130 (`PSDLIB_POSTPROCESS_ERROR_BAD_TTP2`): TTP2 value of the event/curve has no correspondent entry in the discrimination table (probably a wrong discrimination table has been specified)
- 131 (`PSDLIB_POSTPROCESS_ERROR_NO_PRP`): No prepared data has been found in the input group (the input group is of wrong ISDC level; `spi_psd_postprocess` needs corrected data)
- 132 (`PSDLIB_POSTPROCESS_ERROR_NO_COR`): No corrected data has been found in the input group (the input group is of wrong ISDC level; `spi_psd_postprocess` needs corrected data)
- 133 (`PSDLIB_POSTPROCESS_ERROR_NO_OSM`): No derived data has been found in the input group (the input group is of wrong ISDC level; `spi_psd_postprocess` needs corrected data)
- 134 (`PSDLIB_POSTPROCESS_ERROR_BAD_ENG`): The energy information of the PSD event/curve or the discrimination table was unusable (in principle this error should never occur)
- 135 (`PSDLIB_POSTPROCESS_ERROR_PSD_ERR`): A PSD analysis onboard error occurred for this event, hence no post-processing is possible (this should occur for each event which has a PSD error flag comprised between 0 and 15)

5 Algorithm

For each PSD event or PSD curve, the scientific event word contains the best fitting time-to-peak values of a pair of two pulse templates (`PSD_TTP1` and `PSD_TTP2`) and the normalised amplitude of the smaller pulse (`PSD_AMP`). Note that the time-to-peak of the smaller pulse is always stored in `PSD_TTP1` and by definition it is always smaller than 0.5 (since it is the normalised amplitude of the smaller pulse). These three parameters fully define the result of the PSD onboard pulse shape analysis, and also onboard discrimination is based on these parameters. The parameters are found in the prepared photon/photon data (the extraction of these parameters from telemetry is done by the task `dp_spi_psd`; see the corresponding User Manual).

An event is defined as single-site interaction if the normalised amplitude of the smaller pulse (i.e. of `PSD_TTP1`) is below a given threshold `MAXTHRES`. In general, this threshold depends on the time-to-peak values of the best fitting pulse pair and on the energy of the event.

For onboard PSD discrimination, the threshold is set to 0.5 if the time-to-peak difference is smaller than a given spacing (as default, a threshold of 0.5 is used for peak spacings of ≤ 2). This means that events which show only a small peak spacing are considered as single-site interactions. For larger peak spacings, the threshold is generally set to about 0.2, i.e. events for which the secondary pulse is small are also generally considered as single-site events. All remaining events are multiple-site interactions.

Onground post-processing allows for a more general discrimination scheme which should be more efficient. Preliminary investigations have shown that improved results are obtained for thresholds that decrease with increasing peak spacing. Also the order of the time-to-peak values may play a role. Onground post-processing allows for the most general pulse discrimination that is possible with the available information since the PSD discrimination table `SPI.-DISC-PSD` specifies the maximum threshold of single site events for all possible combinations of time-to-peak values.

6 Error codes

The executable `spi_psd_postprocess` may stop with the following error codes:

```
SPI_PSD_POSTPROCESS_ERROR_MEM_ALLOC          -230200
```

They have the following meaning:

- **SPI_PSD_POSTPROCESS_ERROR_MEM_ALLOC** : the allocation of dynamical memory has failed. Probable your system resources are too limited to run this task. The user may try to reduce the task's buffer size parameters **SPI_PSD_POSTPROCESS_PSD_BUFFER_PE** and **SPI_PSD_POSTPROCESS_PSD_BUFFER_CRVE** in the **spi_psd_postprocess.h** header file. These buffers can be much smaller than those specified by default, since **spi_psd_postprocess** always processes data in junks (hence smaller buffer sizes simply imply more disk access and hence slower execution). By default, the buffer sizes have been adjusted so that **spi_psd_postprocess** can handle entire science windows by a single DAL access.

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.