spi_psd_optimise

User Manual

Version 2.0.1
21 January 2003

Jürgen Knödlseder
Centre d'Etude Spatiale des Rayonnements
knodlseder@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

The executable `spi_psd_optimise` is part of the SPI scientific analysis preparation software component (SAP). It optimises the PSD discrimination parameters in order to maximise the sensitivity gain by the PSD sub-assembly.

This second version of `spi_psd_optimise` presents a major revision with respect to the first version, which only was a dummy executable. `spi_psd_optimise` now really optimises the PSD discrimination by maximising the sensitivity improvement for the *detection* of instrumental background lines that are known to originate from photon interactions. It also estimates the PSD response factors f_p and f_b at the line energies.

Several discrimination schemes have been implemented that gradually increase in complexity. However, more complex discrimination schemes are also more sensitive to the statistical limits of the data (i.e. the finit number of events in the instrumental background lines), which tends to produce an overestimation of the PSD discrimination performance. Thus, a trade-off has to be found between the complexity of the discrimination scheme and the reliability of the PSD optimisation. This is particularly important for the determination of the PSD response factors.

`spi_psd_optimise` is written in the ANSI C++ language and has been developpped 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_optimise`, 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_optimise.tar.gz` file, step into a directory that should hold the distribution, move the `spi_psd_optimise.tar.gz` file into this directory and type after the UNIX prompt `$` (don't type this prompt):

```
$ gunzip spi_psd_optimise.tar.gz
$ tar xvf spi_psd_optimise.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 optimisation              #
#                                                                    #
# ------------------------------------------------------------------ #
#                                                                    #
#  File:      spi_psd_optimise.par                                   #
#  Version:   2.0.1                                                  #
#  Component: SAP                                                    #
#                                                                    #
#  Author:    Juergen Knoedlseder                                    #
#             knodlseder@cesr.fr                                     #
#             CESR                                                   #
#                                                                    #
#  Purpose:   Parameter file of the SPI PSD discrimination           #
#             optimisation executable                                #
#                                                                    #
#  History:   2.0.1  21-Jan-2003  Second ISDC delivery               #
#                                                                    #
######################################################################
#
# The input DOLs/filenames
#========================
ingrpDOL,  s,q,"og_spi.fits",,,             "Input Observation Group DOL or filename"
ingtiDOL,  s,q,"",,,                        "Good Time Interval DOL or filename"
insetDOL,  s,q,"psd_cube_set.fits",,,       "PSD response cube definition DOL or filename"
incoeffDOL,s,q,"spi_gain_coeff_index.fits",,,"Gain correction DOL (File/IDX)"
#
# The output DOLs/filenames
#=========================
outgrpDOL,s,q,"og_spi.fits",,,  "Output Observation Group DOL or filename"
outdisDOL,s,q,"psd_disc.fits",,,"PSD discrimination DOL or filename"
#
# Task parameters
#================
alpha,  r,q, 0.0,,,"Signal-to-background fraction"
method, s,h,"2D",,,"Optimisation method (NO/SAID/ONBOARD/2D/2DSLICE/3D)"
nslices,i,h,  10,,,"Number of slices for 2DSLICE"
slice,  b,h,  no,,,"Optimise PSD for time sub-intervals ?"
ontime, r,h, 0.0,,,"Constant ONTIME slice (seconds)"
#
# Standard parameters
#====================
clobber,b,h,yes,,,"Overwrite existing output data ?"
verbose,i,h,3,0,3,"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_optimise` 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 discrimination parameters should be optimised. The input group may be of level `PRP` or `COR`. If no corrected data are available (i.e. level `PRP`), `spi_psd_optimise` provides the possibility to perform on-the-fly energy calibration.

  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.

- `ingtiDOL` not used so far (should remain blank).

- `insetDOL` (optional) specifies the DOL or filename of the PSD response cube definition (HDU `[SPI.-CUBE-SET]`). This data structure specifies which instrumental gamma-ray lines should be used for PSD optimisation and which energy intervals should be used for the analysis.

  If a `[SPI.-CUBE-SET]` 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.-CUBE-SET]` element exists already in the input Observation Group. If no such element is found, however, the task execution is aborted with an error message.

- `incoeffDOL` (optional) specifies the DOL or filename of the gain coefficients file or index (HDU `[SPI.-COEF-CAL]` or `[SPI.-COEF-CAL-IDX]`). This data structure is only required if no corrected data are present in the input Observation Group (i.e. if no `COR` data are found for the PSD events). In this case (and only in this case) an on-the-fly energy calibration will be performed based of the gain coefficients that are found in the data structure.

  If a `[SPI.-COEF-CAL]` or `[SPI.-COEF-CAL-IDX]` 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.-COEF-CAL]` or `[SPI.-COEF-CAL-IDX]` element exists already in the input Observation Group. 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 plus the resulting SPI PSD discrimination data structure `[SPI.-DISC-PSD]` attached.

  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.

- `outdisDOL` specifies the DOL or filename of the PSD discrimination parameter output data structure (HDU `[SPI.-DISC-PSD]`). After execution of the task, this data structure will contain the optimised PSD discrimination parameters and the PSD response factors at the node energies specified by the PSD response cube definition.

  This data structure will be attached to the output Observation Group. Any existing data structure of the same type that may already exist in the output Observation Group will be detached before. If the specified DOL is identical to an already existing DOL, this DOL will be deleted if the `clobber` parameter is `yes` (otherwise the task will abort with an error).

- `alpha` specifies the signal-to-background fraction that should be assumed for PSD discrimination optimisation. Since in any case where the PSD is expected to improve the sensitivity the signal is much weaker than the background, a default value of `0.0` is suggested (this corresponds to the background limited case). However, one may fine-tune the optimisation by selecting a fraction that is close to the expected value.

- `method` specifies the PSD optimisation method. The following methods are available (see section 5 for a detailed description): `NO`, `SAID`, `ONBOARD`, `2D`, `2DSLICE`, `3D`.
- `nslices` specifies for the PSD optimisation method `2DSLICE` the number of slices (see section 5).
- `slice` specifies if PSD optimisation should be performed on time intervals of constant `ONTIME`, leading to a time dependent PSD discrimination optimisation. Notice, however, that time-slicing will reduce the number of events available for optimisation, and the discrimination and PSD response factors may become biased due to the limited statistics of the data. Hence, if there is no evidence for a variation of the PSD performance with time it is suggested to avoid time slicing (hence by default this parameters should be set to `no`).
- `ontime` (optional) if `slice=yes`, specifies the constant `ONTIME` of the time intervals (in units of seconds) into which the PSD event data found in the input Observation Group will be devided for PSD discrimination optimisation.
- `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 or PSD discrimination parameter structure), the task will exit with an error message.
- `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 discrimination optimisation.

4 Interface definition

On input, `spi_psd_optimise` requires the following data structures (either in the input Observation Group or as DOLs or filenames specified in the parameter file):

- `GNRL-IDX.-GRP` science window group index that groups all science window groups of level `PRP` or `COR`. If the input data is of level `PRP` only (i.e. no `COR` data are found for the PSD events), on-the-fly energy calibration will be performed by using the gain calibration coefficients that are found in the `[SPI.-COEF-CAL]` or `[SPI.-COEF-CAL-IDX]` data structure. The PSD optimisation will be based on all PSD events that are found in this index group.
- `SPI.-CUBE-SET` PSD response cube definition, specifying the node energies and energy intervals for which PSD discrimination optimisation should be performed.

In addition, a `[SPI.-COEF-CAL]` or `[SPI.-COEF-CAL-IDX]` data structure is required if the level of the input Observation Group is `PRP`.

All members of the input Observation Group become members of the output Observation Group. In addition, the following additional HDU is attached to the output Observation Group (or replaced if it exists already):

- `SPI.-DISC-PSD` optimised PSD discrimination parameters.

If no output Observation Group is specified, `spi_psd_optimise` attempts to make the input Observation Group to the output Observation Group, which is only possible if the `clobber` parameter is set to `yes`. Vice versa, if no input Observation Group is specified, `spi_psd_optimise` uses the output Observation Group as input Observation Group. For more details on the group logic, the user is referred to the discription of the group API in the `spi_toolslib` User Manual.

From the photon data that are present in the Observation Group, `spi_psd_optimise` builds so-called PSD response cubes for a specified number of gamma-ray lines and adjacent continuum bands. A response cube is a 3-dimensional data cube that is spanned by the PSD parameters `PSD_TTP1`, `PSD_TTP2`, and `PSD_AMP` that are found in the `PRP` data for each PSD event.

The energy bands that are used for the response cube definition are defined by a PSD response cube definition of HDU type `[SPI.-CUBE-SET]`. Each row of this data structure defines an energy node at which the PSD discrimination is optimised. Each node is comprised of 2 sets of 10 energy intervals at maximum. One set is used to populate a response cube comprised of multiple-site PSD events (the so-called *m-cube*), and one set is used to build a single-site response cube (the so-called *s-cube*).

Typically, the multiple-site response cube is built from energy intervals that enclose instrumental gamma-ray lines that are known to arise from photon interactions, while the single-site cube is built from adjacent energy intervals that are free of gamma-ray lines. In addition, two narrow energy intervals are defined for each of the 10 m-cube intervals in order to estimate the contribution of single-site events to the m-cube. A linear interpolation of the number of events in these two intervals provides an estimate of the continuum under each of the gamma-ray lines (where typically one interval is to the left and the other interval is to the right of the gamma-ray line; yet one may also choose both intervals on one side of the line, the continuum is then estimated by extrapolation). For optimisation, the m-cube is then *cleaned* from single-site events by subtracting the s-cube, scaled to the estimated number of single-site events in the m-cube energy intervals. This cleaning is essential, in particular in the case of weak gamma-ray lines.

Notice also that separate m- and s-cubes will be built for each of the 19 detectors, hence PSD discrimination optimisation will be performed for all detection channels individually.

The columns of the `[SPI.-CUBE-SET]` data structure have the following meaning:

- `LINE_ID` 20 character identifier of the instrumental gamma-ray that is used for optimisation.
- `LINE_ENERGY` energy in keV of the instrumental gamma-ray line that is used for optimisation. This energy will be copied to the `ENERGY` field of the `SPI.-DISC-PSD` result data structure. The PSD discrimination and response factors is supposed to apply to this *node energy.*
- `NUM_MULT` number of energy intervals used to build the m-cubes. In general, only a single energy interval that encloses one gamma-ray line is specified, yet one may add several gamma-ray lines in a single response cube to improve the statistics. A maximum of 10 energy intervals may be specified.
- `NUM_SNGL` number of energy intervals used to build the s-cubes. Typically, two single-site energy intervals are specified, one to the left and one to the right of the specified gamma-ray line. Yet in order to improve statistics, several adjacent single-site intervals may be specified. A maximum of 10 energy intervals is possible.
- `MULT_EMIN` minimum energy in keV of the multiple-site energy intervals.
- `MULT_EMAX` maximum energy in keV of the multiple-site energy intervals.
- `MULT_BL_EMIN` minimum energy in keV of a narrow energy interval to the left of the gamma-ray line that is used to estimate the continuum background underneath the gamma-ray line.
- `MULT_BL_EMAX` maximum energy in keV of this energy interval.
- `MULT_BR_EMIN` minimum energy in keV of a narrow energy interval to the right of the gamma-ray line that is used to estimate the continuum background underneath the gamma-ray line.
- `MULT_BR_EMAX` maximum energy in keV of this energy interval.

- SNGL_EMIN minimum energy in keV of the single-site energy intervals.
- SNGL_EMAX maximum energy in keV of the single-site energy intervals.

Using the algorithms described in section 5, **spi_psd_optimise** optimises the PSD discrimination parameters and derives the PSD response factors f_p and f_b. The results are stored in a SPI.-DISC-PSD data structure. For each time-interval, detector, and gamma-ray energy one row is added to this data structure. The columns of this data structure have the following meaning:

- OBT_START OBT start time of the validity interval of the discrimination parameters. A value of DAL3_NO_OBTIME specifies that the discrimination parameters are valid from the first photon on that is found in the input Observation Group.
- OBT_END OBT end time of the validity interval of the discrimination parameters. A value of DAL3_NO_OBTIME specifies that the discrimination parameters are valid up to the last photon found in the input Observation Group.
- SET template library set (0-1) that has been used for this validity interval, detector, and energy.
- DETE detector identifier (0-18) for which the PSD discrimination parameters and response factors apply.
- PSD_PA PSD net pulse area for which the PSD discrimination parameters and response factors apply. The PSD net pulse area is the energy measure of the PSD sub-assembly. It is derived from the energy of the specified gamma-ray line, using the hard-coded calibration relation PA = 0.2798 × ENERGY.
- ENERGY energy (in keV) of the gamma-ray line that has been used for optimisation. This field is copied from the LINE_ENERGY of the SPI.-CUBE-SET data structure, and presents the *node energy* of the PSD discrimination parameters and response factors.
- PSD_FP PSD response factor that gives the fraction of photons that have been correctly identified as photons. Typically, a value around 0.5 is expected.
- PSD_FB PSD response factor that gives the fraction of background events that have been correctly identified as background events. Typically, a value around 0.8-0.9 is expected.
- N_TTP1 Number of templates used for PSD_TTP1. Typically, a value of 38 should be found.
- N_TTP2 Number of templates used for PSD_TTP2. This value should be identical to N_TTP1.
- PSD_TTP1 Vector of PSD_TTP1 values of length N_TTP1. The remaining entries of the 38 element array are padded with zeros.
- PSD_TTP2 Vector of PSD_TTP2 values of length N_TTP2. The remaining entries of the 38 element array are padded with zeros.
- MAXTHRES Vector of maximum smaller peak amplitudes of length N_TTP1 × N_TTP2. The remaining entries of the 1444 element vector are padded with zeros. The vector represents in fact a 2-dimensional array, where the vector index is given by PSD_TTP1 + PSD_TTP2 × N_TTP1.

5 Algorithm

spi_psd_optimise implements the following PSD discrimination optimisation methods:

- NO no PSD discrimination, all PSD events are treated as if they were multiple-site events, hence no PSD events is discarded as background.

- SAID PSD discrimination scheme used during the design and developement of the PSD sub-assembly by Said Slassi-Sennou. It is based on the absolute difference of the PSD_TTP1 and PSD_TTP2 values and a single threshold for the smaller peak amplitude PSD_AMP.
- ONBOARD PSD onboard discrimination scheme which extends the SAID scheme by considering the sign of the PSD_TTP1 − PSD_TTP2 difference.
- 2D 2-dimensional discrimination scheme which provides a variable smaller peak amplitude threshold as function of the PSD_TTP1 − PSD_TTP2 difference.
- 2DSLICE same as 2D, yet the PSD_TTP1 and PSD_TTP2 parameter space is split into slices of constant PSD_TTP1 + PSD_TTP2, and different smaller peak amplitude thresholds apply for each slice.
- 3D most general 3-dimensional discrimination scheme which provides a separate smaller peak amplitude threshold for each pair of PSD_TTP1 and PSD_TTP2.

If ONTIME slicing is requested (slice=yes), the data in the input observation group are sliced into intervals of constant ONTIME. The duration of the time intervals is specified by the parameter ontime. Note that too short time slices may lead to unreliable PSD discrimination parameters and response factors (in fact, it is not recommended to slice the data unless there is a reason to believe that the PSD performance changed with time). To avoid this situation for the last time slice (for which it is highly likely that the remaining time is considerably shorther than the requested time), the last time slice is merged with the previous time slice if the ONTIME of the last time slice is shorter than 90% of the requested ONTIME (the fraction is set by the parameter SPI_PSD_OPTIMISE_SLICE_FRACT in the spi_psd_optimise.h header file).

spi_psd_optimise also detects library set changes, which imply changes of the PSD discrimination parameters (since for example the number of templates may no be identical for different library sets). If a library set change is detected (by examining the PSD configuration parameters), a new time interval is added. Since the PSD configuration is set by 3 different telecommands (which are generally sent at different times), library set changes that occur within 100 seconds (defined by the parameter SPI_PSD_OPTIMISE_CONF_DOBT in the spi_psd_optimise.h header file) are considered as a single library set change (otherwise, three library set changes whould be detected each time that the PSD configuration is changed).

5.1 NO

No PSD discrimination, all events will be flagged as multiple-site (i.e. MAXTHRES = 0 for all PSD_TTP1 and PSD_TTP2 values).

5.2 SAID

The original discrimination scheme used by Said Slassi-Sennou during the developement phase of the PSD sub-assembly. Here, all events with an absolute time-to-peak difference ≤ SPC are considered as single site events, where SPC is typically 2. For larger time-to-peak differences, all events with a smaller peak amplitude PSD_AMP < TRS are considered as single-site events, with TRS typically 0.2.

5.3 ONBOARD

The onboard discrimination scheme which is an extension of Said's scheme.

5.4 2D

A 2-dimensional discrimination scheme where each time-to-peak difference PSD_TTP1 − PSD_TTP2 has a proper threshold value.

5.5 2DSLICE

An extension of the 2D scheme where the PSD_TTP1 / PSD_TTP2 plane is devided in slices of constant PSD_TTP1 + PSD_TTP2, and each time-to-peak difference PSD_TTP1 − PSD_TTP2 has a proper threshold value for each slice.

5.6 3D

The most general discrimination scheme where each PSD_TTP1 / PSD_TTP2 has a proper threshold value.

6 Error codes

The executable spi_psd_optimise may stop with the following error codes:

```
SPI_PSD_OPTIMISE_ERROR_MEM_ALLOC                 -230900
SPI_PSD_OPTIMISE_ERROR_BAD_METHOD                -230901
```

They have the following meaning:

- SPI_PSD_OPTIMISE_ERROR_MEM_ALLOC : the allocation of dynamical memory has failed. Probable your system resources are too limited to run this task.
- SPI_PSD_OPTIMISE_ERROR_BAD_METHOD : an invalid optimisation method has been specified. The only allowed optimisation method is ONBOARD.

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.