RHESSI Calibration User Manual

Andrew R. Inglis, Albert Y. Shih

August 14, 2024

This document provides a brief description of how to run the RHESSI calibration codes.

1 Introduction

RHESSI was a solar X-ray mission that operated betwen 2002 - 2018, sensitive in the 3keV to 17 MeV range. Following the end of mission operations, RHESSI energy gain and resolution were recalibrated as described in a companion document. This document contains technical information for expert users on how to run the RHESSI Pass 2 calibration codes.

This code is written in Interactive Data Language (IDL). To run these codes, and up-to-date installation of SolarSoftWare (SSW) is also needed, with the `hessi` branch installed.

2 Calibration workflow

The workflow for the Pass 2 calibration involved several steps. The first task was to define the times at which calibration was to be performed. This was nominally done at a 3 day cadence for the entire mission, from 2002 – 2018, though there were some additional times where extra calibration points were taken. Once the list of times is defined, level 0 calibration files can be created using the `super_log` procedure. In principle, a file can be created for any set of calibration times (i.e. any time period). For convenience, in Pass 2, calibration time periods were organised around science campaigns, where a science campaign is defined as all observations between detector anneals. The campaigns are defined in Table 1.

Campaign 1 is sufficiently long that it was separated into two components, denoted 'Campaign 1a' and 'Campaign 1b'. Thus, in total seven level 0 files were created for Pass2.

Once the level 0 files are created, each level 0 calibration file is loaded into a GUI that was created specifically for the purpose of viewing and making corrections to the raw calibration data. Once corrections are complete, they are saved as a separate 'changes file' in CSV format. Once this is done for all campaigns the level 1 Calibration data can be created, by applying the desired changes to the level 0 data. The purpose of this workflow is to preserve all the steps of the recalibration process, so that the changes from level 0 to level 1 can be easily inspected.

Campaign	Start Date	End Date
Campaign1	2002-02-13	2007-08-31
Campaign2	2007-12-01	2010-03-15
Campaign3	2010-05-10	2012-01-16
Campaign4	2012-02-24	2014-06-25
Campaign5	2014-08-14	2016-02-22
Campaign6	2016-04-29	2018-04-11

Table 1: RHESSI science campaign definitions

Filename convention	Description
calibration_pass2_campaignX_lvl0.csv	Raw output fit positions and widths from automatic codes
changes_pass2_campaignX.csv	Record of changes to apply to the lvl0 fit data
calibration_pass2_campaignX_lvl1.csv	Fit positions and widths after corrections
gain_pass2_campaignX.csv	Gain and offset values derived from lvl1 fit data
resolution_pass2_campaignX.csv	resolution parameters derived from lvl1 fit data

Table 2: RHESSI calibration filename conventions for Pass 2.

Once all the changes to the level 0 data have been determined, the calibration results can be promoted to level 1 by applying all the changes to the level 0 data. This is done with the `lvl0_to_lvl1` routine. Once the level 1 files are created, the actual gain and resolution parameters can be calculated using the `log2gain` and `log2resol` procedures. These return the actual parameters that are used in the RHESSI response function. One gain and resolution file is created for each campaign. The final step is to combine these files into a single gain file and a single resolution file to be used by the RHESSI software in SSW. This workflow is illustrated graphically in Figure 2.

For reference, in Pass 2 the filename conventions used are described in Table 2.

3 The Code and Examples

The RHESSI calibration code is written in IDL and may be found here: `https://gitlab.com/ayshih/rhessi_calibration.`

This code requires an expert user who is familiar with the RHESSI instrument. Here we show examples of how to use some of the core routines in the recalibration software. This is not an exhaustive manual, but should help the user understand the basics of the process.

The code repository also contains many functions for making various visualizations. For brevity, these are not described in this manual since many of these plots are no longer relevant.

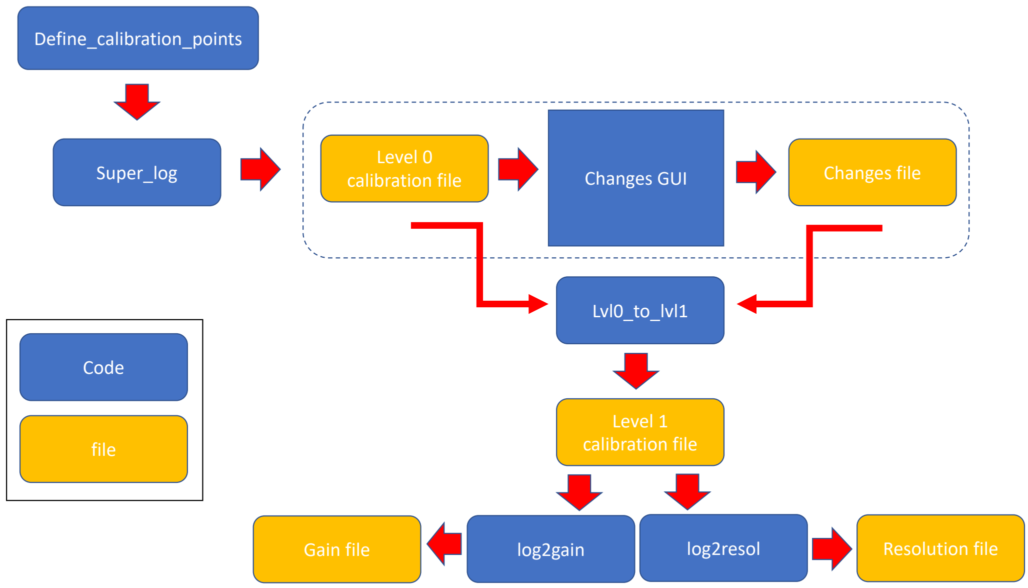

Figure 1: The RHESSI calibration workflow, where code is represented in blue and files represented in orange. First, the points for which calibration is desired are defined. Then, for each of those points the calibration lines are fit for each detector using the `super_log` wrapper procedure. This produces level 0 calibration files. The 'Changes GUI' is used to make corrections to this level 0 data. These changes are saved in a changes file. Then, the `lvl0_to_lvl1` procedure uses the changes file to promote the level 0 data to level 1 data, which is ready to be used to calculate the gain and resolution. The final step is to calculate gain and resolution using the `log2gain` and `log2resol` procedures.

3.1 Retrieving the list of calibration times

The times at which the Pass 2 calibration was performed can be easily retrieved using the define_calibration_points procedure.

Listing 1: How to retrieve the list of calibration times using in the Pass 2 effort

```
IDL> define_calibration_points ,calibration_times
MRDFITS: Binary table.  3 columns by  1650 rows.
IDL> help ,calibration_times
CALIBRATION_TIMES
                    DOUBLE     = Array[1865]
IDL> print ,anytim(calibration_times[0] ,/vms)
13-Feb-2002 12:00:00.000
```

3.2 Performing the level 0 automatic line fitting

To generate the level 0 calibration results using the automated fitting procedures, we can use the super_log_exact_times procedure, which recursively calls the log_lines routine, which controls the fitting. A basic example is shown in the following code block.

Listing 2: Example of performing RHESSI calibration for many time intervals

```
IDL> .r super_log_exact_times.pro
IDL> super_log ,year='2002',propagating_guesses=1,extra_failsafes=1
IDL>␣super_log ,year='2003',propagating_guesses=1,extra_failsafes=1,$
 previous_year_file = 'calibration_2002.csv'
IDL>␣super_log ,year='2004',propagating_guesses=1,extra_failsafes=1,$
 previous_year_file = 'calibration_2003.csv'
```

The above code example first fits the calibration lines every three days for the entirety of 2002, the first year of science operations. By using the propagating_guesses keyword (strongly recommended), the estimated position of each line for each detector is based on the fit to that line at the previous calibration point (if successful). Use of this keyword helps ensure continuity over time. The keyword extra_failsafes performs some additional checks on the results of each fit to determine whether the fit was successful and reasonable. Fits that are determined to have failed are set to zero in the level 0 calibration results file and have to be corrected.

The next line performs the same procedure for the year 2003. The previous_year_file keyword is used so that the last calibration point of 2002 can be used as the input guess for the line positions at the beginning of 2003. The final line accomplishes the same thing for 2004.

Note that the year keyword is not necessary. If desired, an arbitrary list of times spanning multiple years can be run and saved in a single file.

3.3 Running the changes GUI

As part of the recalibration effort, a graphical user interface that can be used to view and edit calibration data as a function of time. Internally it is referred to as the changes_gui. This GUI is needed because, although the level 0 fitting is reasonably robust, inevitably corrections need to be made when the automatic fitting makes errors.

In the GUI, edits can be performed via various click, or click-and-drag actions. The edited calibration can be previewed. Finally, the edits can be saved to an output file. The GUI can be initialized like this:

```
IDL> changes_gui, 'calibration_pass2_campaign2_V02_lvl0.csv',$
'changes_pass2_campaign2_V02_v17.csv',$
old_changes = 'changes_pass2_campaign2_V02_v16.csv',$
width1369_csv = '1369_measurements/1369_widths_campaign2_cleaned.csv'
```

where we have the following input parameters:

- **calibration_pass2_campaign2_V02_lvl0.csv:** A level 0 input calibration file. A file of this type is created by the `super_log` routine. In this example, the GUI will load the calibration data for Campaign 2 of the RHESSI mission.
- **changes_pass2_campaign2_V02_v17.csv:** This is the output file where any changes to the level 0 calibration file will be documented.
- **old_changes:** This keyword is used to load any previously saved changes file. So, the user can load in an existing changes file, make more changes, and the output will be saved into a new changes file.
- **width1369.csv:** Optional. If set, measurements of thee 1369 keV line are loaded and plotted in the GUI for reference when certain options are set. This data is not used in any fitting and is not editable in thee GUI, it is for reference only.

For the full list of GUI parameters, see the code itself. An example of the GUI being used is shown in Figure 2.

Note that several interactive buttons are available in the GUI along the top edge of the viewer.

3.4 Promoting calibration results from level 0 to level 1

Once all of the changes to apply to the level 0 results have been recorded in a changes file, the calibration results can be promoted to level 1 using the `lvl0_to_lvl1` routine, as follows:

```
IDL> lvl0_to_lvl1 , lvl0_csv , changes_csv , lvl1_csv
```

Parameter Explanation:

- **lvl0_csv:** An input level 0 calibration file.
- **changes_csv:** An input changes file.
- **lvl0_csv:** An output level 1 calibration file.

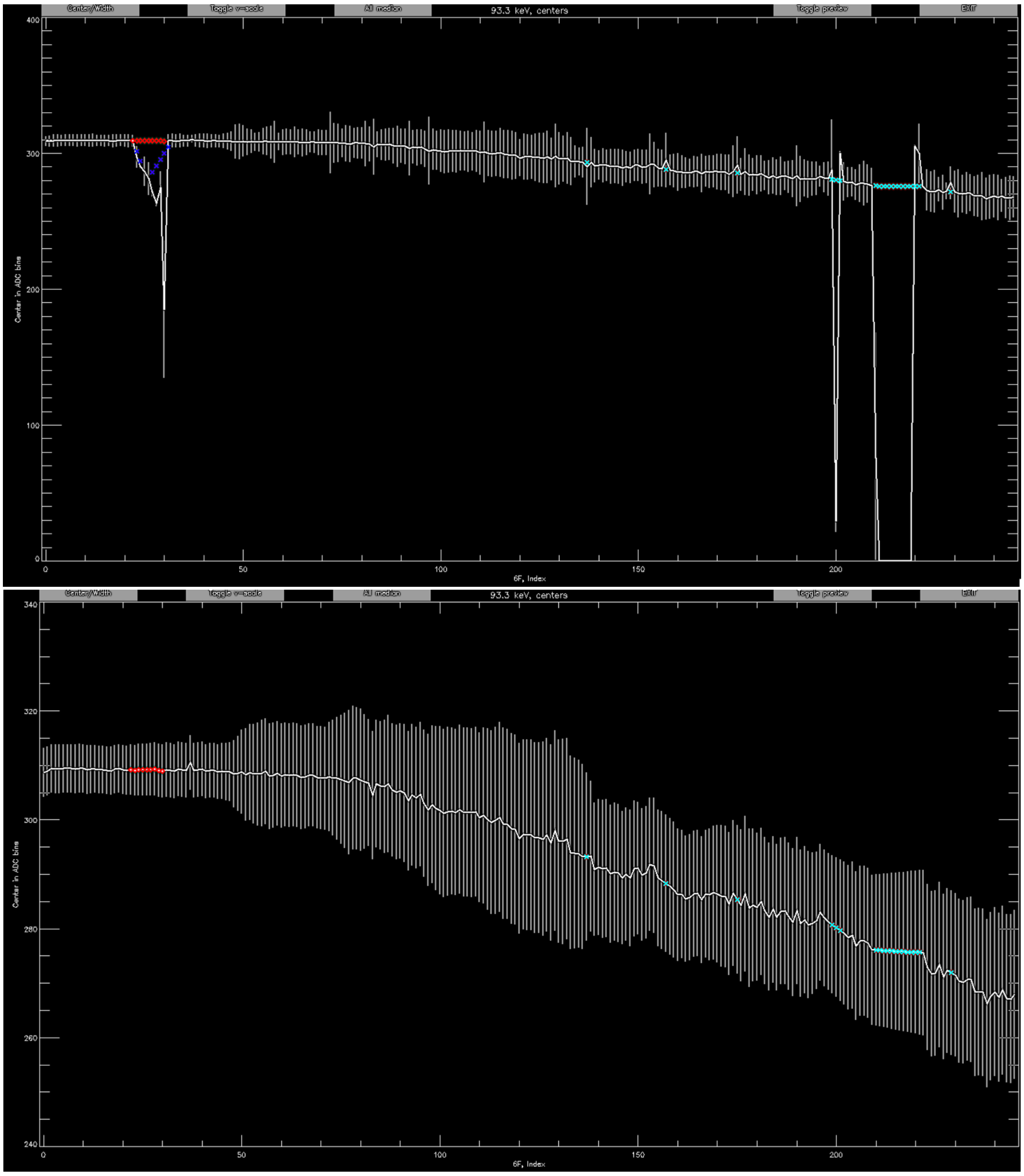

Figure 2: Example of the calibration GUI being used. Shown in white are the raw fitted positions of the 93.3 keV line in ADC space for the front segment of detector 6 during the Campaign 2 time period. The x-axis shows time in data point units. Cyan points indicate where the running median value has been substituted in place of the raw fit. Red points indicate where a manual value has been used to replace the raw fit value. Dark blue points are suggested median values that were **not** adopted by the user. **Bottom:** A preview of the new, cleaned line locations for the D6F 93.3 keV calibration line, obtained by applying the changes shown in the top panel.

3.5 Calculating the Energy Gain parameters

Given a level 1 calibration result file containing the fit results to the three spectral calibration lines, `log2gain` will use those results to obtain the best fit gain function for each RHESSI detector segment. This is done by performing a linear fit to the line positions. `log2gain` returns the slope and offset of this best fit for each detector segment and each desired time, saved to a CSV file if desired.

Code example:

```
IDL> log2gain,'calibration_2002.csv', write_filename = 'gain_2002.csv'
```

Parameter explanation:

- **'calibration_2002.csv'**: An input calibration file. Typically this would be a level 1 file in the format defined by the `super_log` procedure or a derivative thereof.
- **write_filename:** sets the output filename where the gain parameters will be stored.

3.6 Calculating the Energy Resolution Parameters

Given a level 1 calibration result file containing the fit results to the three spectral calibration lines, `log2resol` will use those results to obtain the best fit resolution function for each RHESSI detector segment. This is done by fitting a quadratic function to the line width measurements. `log2resol` returns the best-fit parameters for the resolution equation for each detector segment.

Code example:

```
IDL> log2resol,'calibration_2002.csv', write_filename = $
'resolution_2002.csv'
```

Parameter explanation:

- **'calibration_2002.csv'**: An input calibration file. Typically this would be a level 1 file in the format defined by the `super_log` procedure or a derivative thereof.
- **write_filename:** sets the output filename where the resolution parameters will be stored.