Apply the ACIS CTI Correction

CIAO 3.4 Science Threads

Table of Contents

Apply the ACIS CTI Correction

CIAO 3.4 Science Threads

Overview

Last Update: 21 May 2007 − need to set `stop=none` if aspect solution is not provided

Synopsis:

The loss of charge in a CCD as it is shifted from one pixel to the next during readout is known as charge transfer inefficiency (CTI). As of CALDB 3.1.0 (23 June 2005), the parallel CTI of all of the front−illuminated CCDs has been calibrated.

NEW The CIAO 3.4 and CALDB 3.3.0 releases (18 December 2006) make it possible to apply parallel and serial CTI correction to the back−illuminated CCDs (ACIS−S1 and S3).

Purpose:

To create a new level=2 event file with the ACIS CTI correction applied. This thread also applies the newest available gain map to the data, so it ***is not necessary*** to run the Apply an ACIS Gain Map thread afterwards.

Read this thread if:

you are working with −120 C ACIS data and you are doing spectral analysis. The CTI adjustment may marginally improve grating spectral analyses in the sense that the adjustment may improve order sorting.

It is not necessary to run this thread if you are working with data taken at a focal−plane temperature of −90, −100 or −110 C, as there is no calibration available for these temperatures. It is also unnecessary to apply the correction if the analyses involve only timing or spatial studies.

Calibration Updates:

- ***CALDB v3.3.0*** ***(18 Dec 2006):*** New time−dependent CTI corrections for the back−illuminated ACIS chips were released (ACIS−S1 and S3). This is the first release where the serial and parallel CTI correction is available for the BI chips. Only calibration for the BI chips (S1, S3) has changed in these files; calibration for the FI chips is identical to the v5 files. The How CIAO 3.4 and CALDB 3.3.0 Affect Your Analysis section of the CIAO release notes explains how the files will affect your analysis.
- ***CALDB v3.2.0*** ***(21 Nov 2005):*** New time−dependent CTI corrections for all front−illuminated (FI) ACIS chips were released. The CIAO 3.3 release notes explain how they will affect your analysis.
- ***CALDB v3.1.0*** ***(23 Jun 2005):*** A new calibration file to correct for the CTI was added to the CALDB. The file includes calibration for the S0, S4, and S5 ACIS chips; prior to this release, CTI correction for those chips was unavailable.

Related Links:

- Analysis Guide: ACIS Data Preparation
- Why topic: ACIS CTI Correction
- Caveats at the end of this thread

Proceed to the HTML or hardcopy (PDF: A4 | letter) version of the thread.

How the Data is Affected

There are a number of values in the event file that are affected by applying the CTI correction. The columns that can be changed include FLTGRADE, GRADE, PHA, ENERGY, PI, and STATUS. Here is an outline of what takes place:

- The adjustment computes a PHAS_ADJ column.
- The FLTGRADE is computed using the PHAS_ADJ column instead of the PHAS column.
- GRADE is computed from FLTGRADE.
- PHA is computed from PHAS_ADJ instead of PHAS.
- ENERGY is computed from PHA.
- PI is computed from ENERGY.

Additionally, the STATUS column can be modified if the adjustment does not converge in the specified maximum number of iterations. An explanation is provided in the apply_cti parameter description for acis_process_events; also see the max_cti_iter and cti_converge parameters.

New CTI header keyword: CTI_APP

Prior to CIAO 3.4 and DS 7.6.10, the CTI_CORR header keyword was used to record whether the CTI correction had been applied to a dataset. The value of the keyword was a boolean: "TRUE" (or "1") if the CTI had been applied and "FALSE" (or "0") if it hadn't:

```
unix% dmkeypar acis_1838_evt2_yes.fits CTI_CORR echo+
1

unix% dmkeypar acis_1838_evt2_no.fits CTI_CORR echo+
0
```

The updates to CTI support in the CIAO 3.4 and DS 7.6.10 releases include a new, more descriptive header keyword, CTI_APP. The value of this keyword is a ten−character string which records the CTI applied to each of the ACIS chips (0−9, left to right). Allowed character values are:

- ***N***: no CTI correction applied
- ***P***: parallel CTI correction applied only (expected value for FI chips)
- ***S***: serial CTI correction applied only (currently not a possible case)
- ***B***: both serial and parallel CTI correction applied (expected value for BI chips)

Here are some examples:

- Standard CTI correction to FI and BI chips:

```
unix% dmkeypar acis_evt2.fits CTI_APP echo+
PPPPPBPBPP
```

- Only FI chips corrected for parallel CTI:

```
unix% dmkeypar acis_evt2.fits CTI_APP echo+
PPPPPNPNPP
```

- No CTI correction applied:

```
unix% dmkeypar acis_evt2.fits CTI_APP echo+
NNNNNNNNNN
```

Note that all ten spaces in the value are always populated, even though there are only a maximum of six chips on at a time.

The `CTI_CORR` keyword is retained in the output for backward−compatibility, but all CIAO tools have been updated to use the value of `CTI_APP` when checking for the CTI correction.

Get Started

Sample ObsID used: 1838 (ACIS−S, G21.5−09)

File types needed: evt1; flt1; bpix1

Since the CTI correction affects the assignment of event grades, among other things, ***it must be applied to the level=1 event file***.

If you created a new bad pixel file by running the Create a New ACIS Bad Pixel File: Identify ACIS Hot Pixels and Cosmic Ray Afterglows thread, use that file in this analysis. Otherwise, use the `bpix1.fits` file from the Archive.

There are several pieces of information needed from the file header before proceeding:

- ***If the CTI correction has been applied (`CTI_CORR` and/or `CTI_APP` keyword)***: the CTI correction may have been applied in SDP or during previous CIAO analysis. The `CTI_APP` header keyword was added at CIAO 3.4 and DS 7.6.10 (see the New CTI header keyword section for more information); the older `CTI_CORR` keyword is retained for backward−compatibility.
- ***What version of the software was used to process the data in SDP (`ASCDSVER` keyword)***: to see if the data were processed before or after the correction was included in SDP.

  Components of the CTI correction were added to standard data processing (SDP) at different times:

  - ***DS 6.11***: parallel CTI correction for timed−exposure (TE) mode data taken on front−illuminated ACIS chips
  - ***DS 7.6.3***: correct CTI calibration for continuous−clocking (CC) mode data
  - ***DS 7.6.10***: parallel and serial CTI for observations done with the back−illuminated chips
- ***Which chips were on for the observation (`DETNAM` keyword) and what mode was used (`READMODE` keyword)***: this is needed to determine which version of SDP would be applicable.

- ***Focal plane temperature (`FP_TEMP` keyword)***: the CTI correction only applies to data taken at −120 C (153 K). See this FAQ for more information on checking the temperature of your observation.

The tool dmkeypar is used to get the header information:

```
unix% dmkeypar acisf01838_000N001_evt1.fits CTI_CORR echo+
# dmkeypar (CIAO 3.4): ERROR: Keyword 'CTI_CORR' was not found in file 'acisf01838_000N001_evt1.fits'.

unix% dmkeypar acisf01838_000N001_evt1.fits CTI_APP echo+
# dmkeypar (CIAO 3.4): ERROR: Keyword 'CTI_APP' was not found in file 'acisf01838_000N001_evt1.fits'.

unix% dmkeypar acisf01838_000N001_evt1.fits ASCDSVER echo+
R4CU5UPD8.2

unix% dmkeypar acisf01838_000N001_evt1.fits READMODE echo+
TIMED

unix% dmkeypar acisf01838_000N001_evt1.fits DETNAM echo+
ACIS−012367

unix% dmkeypar acisf01838_000N001_evt1.fits FP_TEMP echo+
153.446014
```

This is a 120 C (153 K) timed−exposure mode observation which uses both front− (ACIS−01236) and back−illuminated (ACIS−7) chips. It was processed before any of the CTI correction became part of SDP, i.e. the `ASCDSVER` is lower than DS 6.11. Note that the version naming convention changed after version R4CU5UPD14 to the "DS" system, starting with DS 6.0.0.

Since the CTI correction has not previously been applied, complete this thread in its entirety. Note that in some cases, the CTI calibration has been applied, but the data should still be reprocessed to apply a newer version of the calibration. Refer to the Calibration Updates for a history of CTI calibration changes and the relevant observation dates.

Related acis_process_events threads

There are other threads that should be considered, since they may affect how `acis_process_events` is run. ***The Create a New Level=2 Event File thread shows how to combine all of these options into a single run of `acis_process_events`.***

- Apply the Time−Dependent ACIS Gain Correction
- Remove Pixel Randomization
- Apply/Remove PHA Randomization
- Clean ACIS Background in VFAINT Mode

Generate A New Level=1 Event File

Determine the eventdef parameter

The `eventdef` parameter specifies the names and data types of the columns in the output event data file. Two *relevant* predefined strings are included in the parameter file for acis_process_events:

READMODE	DATAMODE	event mode	eventdef string
TIMED	(V)FAINT	timed exposure (very) faint	stdlev1
CONTINUOUS	CC(33)_FAINT	continuous clocking (3x3) faint	cclev1

As explained in How the Data is Affected, it is necessary for the PHAS column to exist in the input file. This column is not created for observations taken with the data mode GRADED, so this thread can only be run on FAINT or VFAINT mode data. If you are unsure of the event mode of your observation, the information can be found in the READMODE and DATAMODE values stored in the file header:

```
unix% dmkeypar acisf01838_000N001_evt1.fits READMODE echo+
TIMED

unix% dmkeypar acisf01838_000N001_evt1.fits DATAMODE echo+
FAINT
```

This is a timed exposure faint observation, so the proper eventdef parameter is "stdlev1." The full parameter syntax of each eventdef string may be found in plist acis_process_events.

If you are working with VFAINT mode data and wish to perform the background cleaning as well, make sure to read this caveat before continuing.

Optional Step: when the parameters to apply the CTI correction are set (see the next section), the values of the column PHAS are adjusted and saved in a column named PHAS_ADJ. If you are interested in the PHAS_ADJ column to be present in the output file, you must define a custom eventdef string which includes "f:phas_adj". For example, modifying the "stdlev1" string:

```
unix% pset acis_process_events eventdef="{d:time,s:ccd_id,s:node_id,i:expno,s:chip,s:tdet,f:det,f:sk
```

The *only* effect of doing this is that the extra column will be written to the output. Otherwise all the calculations are done, then the column is omitted from the final event file. ***In the interest of reducing the output file size you can choose to not to keep the PHAS_ADJ column. In this thread, the unaltered "stdlev1" event definition is used.***

Run acis_process_events

Running this tool with the SDP level=1 event file as the input will produce a *new* level=1 event file that has the latest CALDB applied. This means that the newest gain map will be picked up as well.

Since the event grades, energy, and PI values are all affected by the values in the PHAS column, they need to be recalculated using the PHAS_ADJ values; by default the doevtgrade and calculate_pi parameter values are set to "yes", so they do not need to be changed.

The apply_cti parameter indicates that the CTI correction will be applied. Note that the CTI correction is on by default (apply_cti=yes); the pset command is used for emphasis in this example:

```
unix% punlearn acis_process_events
unix% pset acis_process_events infile=acisf01838_000N001_evt1.fits
unix% pset acis_process_events outfile=acis_1838_new_evt1.fits
unix% pset acis_process_events badpixfile=acis_1838_new_bpix1.fits
unix% pset acis_process_events eventdef=")stdlev1"
unix% pset acis_process_events stop=none
```

```
unix% pset acis_process_events apply_cti=yes
unix% acis_process_events
Input event file or stack (acisf01838_000N001_evt1.fits):
Output event file name (acis_1838_new_evt1.fits):
aspect offset file ( NONE | none | <filename>) (NONE):
```

It is important to note the unusual syntax of the `eventdef` parameter; the tool will not access the predefined string if the leading ")" is missing (see example 6 of `ahelp parameter`). If you are using a custom string to retain the `PHAS_ADJ` column, use the `pset` command given in the Determine the eventdef parameter section.

You may see a warning about the number of event islands that contain one or more bad pixels:

```
# acis_process_events (CIAO 3.4): The following error occurred 26941
  times: dsAFEBADPCNTERR −− WARNING: Event island contains 1 or more bad pixels.
```

It is explained in this FAQ and may be ignored.

The content of the parameter file may be checked using plist acis_process_events.

Generate A New Level=2 Event File

Apply grade/status filters

Filter for bad grades (using ASCA grades) and for a "clean" status column (ie all bits set to 0):

```
unix% punlearn dmcopy
unix% dmcopy "acis_1838_new_evt1.fits[EVENTS][grade=0,2,3,4,6,status=0]" \
      acis_1838_flt_evt1.fits
```

Apply GTI filters

The Good Time Intervals (GTIs) supplied by the pipeline now need to be applied. Simultaneously, an unnecessary column is eliminated from the output:

```
unix% punlearn dmcopy
unix% dmcopy "acis_1838_flt_evt1.fits[EVENTS][@acisf01838_000N001_flt1.fits][cols −phas]" \
      acis_1838_evt2.fits
```

Be sure to include the ***@ symbol*** in the filter expression; the command will not be executed properly if it is omitted.

Examining the Output

The new CTI−corrected level=2 event file is named `acis_1838_evt2.fits`. Header keywords have been created to record information about the correction:

```
unix% dmlist acis_1838_evt2.fits header | grep CTI
0077 CTI_CORR             TRUE                           Logical
```

```
0078 CTI_APP              PPPPPBPBPP                        String
0079 CTIFILE              acisD2000−01−29ctiN0006.fits   String
```

The CTI_APP value shows that the parallel CTI correction was applied to the FI chips, while both serial and parallel corrections were applied to the BI chips. For a complete explanation of the CTI_CORR and CTI_APP keywords, refer to the New CTI header keyword section. The CTIFILE is the name of the CTI calibration file applied to the data. A CTI−corrected gainfile has also been applied to the data:

```
unix% dmkeypar acis_1838_evt2.fits GAINFILE echo+
acisD2000−01−29gain_ctiN0006.fits
```

If you kept the PHAS_ADJ column in the output file and decide later you would like to remove it, use dmcopy:

```
unix% punlearn dmcopy
unix% dmcopy "acis_1838_evt2.fits[EVENTS][cols −phas_adj]" acis_1838_evt2_nophas.fits
```

Notes

A common warning message

While running this thread, users may see the following warning:

```
# acis_process_events (CIAO 3.4): WARNING: No files found matching CALDB search:
        tel=CHANDRA
        inst=ACIS
        det=−
        filt=−
        product=CTI
        start_date=1999−08−20T00:07:44
        start_time=00:07:44
        stop_date=1999−08−20T02:26:38
        stop_time=02:26:38
        query=−
# acis_process_events (CIAO 3.4): WARNING: PROBLEM READING CTIFILE, CTI ADJUSTMENT WILL NOT BE APPLI
```

The most common cause for seeing this warning is that the observation does not contain −120C data. Since the CTI correction is only calibrated for this focal place temperature, it is not possible to use it on other observations. Note that this is *not an error*; if no CTI file is found, acis_process_events continues running as if apply_cti=no and produces a valid output file that has not been CTI−corrected.

Reapplying or removing the CTI correction

If the input file has already been CTI−corrected and "***apply_cti=yes***", the CTI correction is recalculated. It is not, however, applied on top of the first correction; since the PHAS column remains unadjusted, the new correction will start from there to redetermine the PHA, ENERGY and PI values.

If the input file has already been CTI−corrected and "***apply_cti=no***", the CTI correction is ***removed*** (PHA, ENERGY and PI values are recomputed using the original, pre−CTI−corrected data).

Caveats

Frame times, subarrays, and CC mode

The CTI adjustment has been fully tested and calibrated on −120 C observations in TIMED mode with frame time of 3.2 s. The calibration group knows that the CTI process is slightly different in modes which have longer or shorter frame times, such as subarrays. This is because different amounts of cosmic ray and other background charge can partially fill traps. They expect that CTI−corrected data will still be an improvement, and will be as accurate as our calibration base allows at the present time. Please let us know (via Helpdesk) if you have evidence of specific discrepancies due to different frame times.

In CC mode there is also a change in parallel transfer rate; the rate for TE mode is 40 ***microsec***/row, and for CC mode it's 2.8 ***millisec***/row. Different traps become more or less important for the different transfer rates. There is very little calibration data in this mode, so it is suggested that CTI correction be compared with uncorrected data, and the results carefully considered for reasonableness.

```
Parameters for /home/username/cxcds_param/acis_process_events.par


#−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
#
#  acis_process_events.par− Parameter file for acis_process_events program
#
#−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
        infile = acisf01838_000N001_evt1.fits   Input event file or stack
       outfile = acis_1838_new_evt1.fits Output event file name
    acaofffile = NONE             aspect offset file ( NONE | none | <filename>)
    (apply_cti = yes)             Apply CTI adjustment?
  (apply_tgain = yes)             Apply time−dependent gain adjustment?
(alignmentfile = )acaofffile −> NONE) sim/fam alignment file ( NONE | none | <filename>)
      (obsfile = NONE)            obs.par file for output file keywords ( NONE | none | <filename>)
      (geompar = geom)            Parameter file for Pixlib Geometry files
      (logfile = stdout)          debug log file ( STDOUT | stdout | <filename>)
    (gradefile = CALDB)           grade mapping file ( NONE | none | CALDB | <filename>)
     (gainfile = CALDB)           acis gain file ( NONE | none | CALDB | <filename>)
   (badpixfile = acis_1838_new_bpix1.fits) acis bad pixel file ( NONE | none | <filename>)
   (threshfile = CALDB)           split threshold file ( NONE | none | CALDB | <filename>)
      (ctifile = CALDB)           acis CTI file ( NONE | none | CALDB | <filename>)
    (tgainfile = CALDB)           gain adjustment file ( NONE | none | CALDB | <filename>)
     (eventdef = )stdlev1 −> {d:time,s:ccd_id,s:node_id,i:expno,s:chip,s:tdet,f:det,f:sky,s:phas,l:pha,l:ph
f:energy,l:pi,s:fltgrade,s:grade,x:status}) output format definition
   (doevtgrade = yes)             Determine event flight grade?
 (check_vf_pha = no)              Check very faint pixels?
(calc_cc_times = no)              Estimate the times of arrival for CC−mode observation?
        (trail = 0.027)           Trail fraction
     (spthresh = 13)              Default split threshold level (overridden by values in threshfile)
  (time_offset = 0)               Offset to add to event time field to synch w/ fam data
   (docentroid = no)              Determine pixel centroid for coord. conversion?
 (calculate_pi = yes)             perform pha−>pi conversion? (requires gain file)
 (pi_bin_width = 14.6)            Width of Pi bin in eV
  (pi_num_bins = 1024)            Number of values to bin energy into
 (max_cti_iter = 15)              Maximum iterations for the CTI adjustment of each event
 (cti_converge = 0.1)             The convergence criterion for each CTI−adjusted pixel in adu
       (tstart = TSTART)          header key containing start/default time value
```

```
        (tstop = TSTOP)          header key containing time of last event
      (clobber = no)             Overwrite output event file if it already exists?
      (verbose = 0)              level of debug detail (0=none, 5=most)
         (stop = none)           end transformations at [chip,tdet,det,tan,sky,none]
     (instrume = acis)           axaf instrument− used for instrument parameter file
    (rand_seed = 1)              random seed (for pixlib), 0 = use time dependent seed
     (rand_pha = yes)            Randomize the pha value used in gain calculations
(rand_pix_size = 0.5)            pixel randomization width (−size..+size) 0=no randomization
      (stdlev1 = {d:time,s:ccd_id,s:node_id,i:expno,s:chip,s:tdet,f:det,f:sky,s:phas,l:pha,l:pha_ro,
s:fltgrade,s:grade,x:status}) TE faint modes event definition string
      (grdlev1 = {d:time,s:ccd_id,s:node_id,i:expno,s:chip,s:tdet,f:det,f:sky,l:pha,l:pha_ro,s:corn_
s:fltgrade,s:grade,x:status}) TE graded event format definition string
       (cclev1 = {d:time,s:ccd_id,s:node_id,i:expno,s:chip,s:tdet,f:det,f:sky,f:sky_1d,s:phas,l:pha,
f:energy,l:pi,s:fltgrade,s:grade,x:status}) CC faint event format definition string
    (ccgrdlev1 = {d:time,s:ccd_id,s:node_id,i:expno,s:chip,s:tdet,f:det,f:sky,f:sky_1d,l:pha,l:pha_r
f:energy,l:pi,s:fltgrade,s:grade,x:status}) cc graded event format definition string
         (mode = ql)
```

History

16 Dec 2004	updated for CIAO 3.2: use ACIS bad pixel file (badpixfile parameter)
01 Feb 2005	added note about "Event island contains 1 or more bad pixels" warning
23 Jun 2005	CIAO 3.2.2 patch: new calibration file in CALDB 3.1.0; minor acis_process_events parameter change (default value of threshfile is CALDB instead of NONE)
12 Dec 2005	updated for CIAO 3.3: there was a fix to the CTI correction for CC−mode data in SDP version DS 7.6.3 (12 August 2005); new CTI files in CALDB 3.2.0; output filenames include ObsID
18 Dec 2006	updated for CIAO 3.4: new CTI files in CALDB 3.3.0, including calibration for serial CTI; New CTI header keyword section; CIAO version in errors and warnings
21 May 2007	need to set stop=none if aspect solution is not provided

URL: http://cxc.harvard.edu/ciao/threads/acisapplycti/ Last modified: 21 May 2007