epexposure

August 16, 2005

Abstract

This meta task constructs livetime factors for EPIC pn scientific observations and stores them in a CCD specific exposure extension. Note: this description is currently outdated. The task now only removes a couple of columns in the EXPOSUnn extensions and reads in and analyses DLIMAP extensions (if present)

1 Instruments/Modes

Instrument	Mode
EPIC PN	IMAGING, TIMING, BURST

2 Use

pipeline processing	yes
interactive analysis	yes

3 Description

There is a need to calculate the fractional exposure `FRACEXP` column entries in the CCD specific exposure extensions (see [2] and SSC_LUX_TN_0038) per frame which will be addressed by higher level SAS tasks for the computation of exposure times or exposure maps. In addition one needs to create GTIs for timing analysis, e.g. light-curves or for determination of absolute quantities like fluxes.

The plan is to use the counter and discarded line info stored within the EPIC pn Auxiliary (AUX) File ([1], p. 50) to check if and how many events have been lost during the on-board processing and telemetry or have been flagged as bad in **epframes**.

A problem arises when comparing these counters with the number of events in the actual CCD specific events list: counters are given every m CYCLEs and they come *per quadrant, i.e. not per CCD*. Here m gives the number of cycles which have been integrated prior to the creation of the counter information. It is a programmable parameter defaulted to 20 to reduce the housekeeping telemetry and needs to be extracted from the periodic housekeeping data (PMH) (parameter name is `An_MAXFRC`, for quadrant $n = 0..3$, see Table 1). For a description of all HK-parameters see [3].

HK Parameter	Storage	Values	Meaning
An_MAXFRC	PMH	0 - 255, def. 20	number of cycles after which count info is sent
An_CCD0SWISTAT	PMH	ON/OFF	status of CCD 0 of this quadrant
An_CCD1SWISTAT	PMH	ON/OFF	status of CCD 1 of this quadrant
An_CCD2SWISTAT	PMH	ON/OFF	status of CCD 2 of this quadrant
An_CHOP	PMH	0 - 31	number of frames excluded from the onboard processing after one was included
NDISCLIN	AUX	0 - 16383	number of discarded columns
NEPDH	AUX	0 - 65535	number of events which have been transmitted from Event Analyser to Data Handling Unit

Table 1: Counters used for livetime computation

Basic Idea

There are several different deadtime contributions that have to be taken into account during the EPIC pn ODF processing: one factor depends solely on the readout mode, the second is related to deadtimes due to discarded CCD columns because of on-board MIP (Minimum Ionizing Particle) rejection, the third factor represents additional deadtimes from MIP rejection per frame per CCD by ground software. The fourth livetime factor handles the case that not all frames have been transmitted because of telemetry limits. Finally, quadrant and CCD specific deadtimes are caused by the bad frame flagging in **epframes**.

Mode dependent livetime factor FC0

The first livetime factor is mode dependent and constant for each exposure. It corresponds to the ratio of integration time to total time. The frame read-out dead-time is calculated in **epframes** already and appears (implicitly) in the `TIMEDEL` entries in the EXPOSURE extension which are smaller than the frame interval by a factor FC0. This factor is also written by **epframes** into the keyword `AX_FC0`.

MIP dependent quadrant livetime factor FC1

A counter in the AUX file is the Discarded Line counter `NDSCLIN` (see Table 1). This value comes per quadrant and per n CYCLEs: In case of full frame, extended full frame and large window modes `NDISCLIN` gives the number of discarded columns. The corresponding livetime factor is

$$FC1 = 1 - \mathrm{NDISCLIN}/(64 \cdot \#CCD \cdot \mathrm{An_MAXFRC}) \; . \tag{1}$$

The factor $\#CCD$ gives the number of active CCDs in the quadrant under consideration. Under normal circumstances this should be 3 for the above three imaging modes. To check for possibly switched off CCDs, however, one needs to extract the counter `An_CCDSEL` (see Table 1).

In case of small window, timing, and burst mode there is only one active CCD. For these modes the onboard MIP rejection is switched off (all events transmitted, NDSCLIN = 0) and the equation reduces to $FC1 = 1$.

MIP dependent frames livetime factor FC2

If in **epframes** MIP rejection is switched on (screening of residual MIPs) then the EXPOSURE extension contains also information about the discarded columns (FDISCLIN) per frame for each individual CCD.

This yields a similar livetime factor

$$FC2 = 1 - \mathrm{FDISCLIN}/64\,. \tag{2}$$

Chopper dependent livetime factor FC3

Because of telemetry limits an exposure might be programmed such that not all frames will be transmitted. This info is stored as quadrant specific periodic housekeeping parameters `An_CHOP` (see Table 1). These chopper parameters (normally 0) control the number of frames excluded from the onboard analysis before one is analysed and sent to telemetry. The livetime due to such discarded frames is represented by the factor

$$FC3 = 1/(1 + An_CHOP)\,. \tag{3}$$

As the chopper parameter and thus the associated livetime is constant for a specific exposure the values be stored in the keywords `An_CHOP` and `AX_FC3` by **epframes** in the CCD specific EXPOSURE extension to be addressed by higher SAS tasks. If one is just interested in a correct count rate then it is sufficient to divide the rate by FC3.

CCD specific frame quality dependent livetime

The SAS task **epframes** flags raw events as bad if events come in wrong readout order or if coordinates have invalid values (outside of the active window) because of data transfer problems. The **epframes** task handles this livetime via the creation of GTIs.

Quadrant specific frame quality dependent livetime

If there is a difference between the NEPDH counter (see Table 1), giving info on the number of events which have been transmitted from the Event Analyser [EPEA] to the Data Handling Unit [EPDH]) and the number of raw events in the corresponding time interval in all active CCDs for this quadrant then the interval associated with these raw events has to be considered as bad (handled via GTI creation) Both frame counters have to be computed over the time interval of m CYCLEs from the quadrant under consideration.

4 Parameters

This section documents the parameters recognized by this task (if any).

Parameter	Mand	Type	Default	Constraints
eventsets	yes	dataset list	events01.dat,events02.dat,events03.dat	

list of input event files (from a single quadrant)

randomizetime	no	boolean	N	Y/N

randomize TIME within one readout frame

screenexposure	no	boolean	Y	Y/N

remove extra columns in EXPOSU** extensions

spatialexposure	no	boolean	N	Y/N

determine spatial exposure inhomogeneities (CCD columns)

5 Errors

This section documents warnings and errors generated by this task (if any). Note that warnings and errors can also be generated in the SAS infrastructure libraries, in which case they would not be documented here. Refer to the index of all errors and warnings available in the HTML version of the SAS documentation.

inconsistent dataset list *(error)*
input data sets do not belong to the same quadrant and exposure - *does* ***evlistcomb*** *already check for this?*

badMode *(error)*
readout mode currently not supported

badSubMode *(error)*
IMAGING submode currently not supported

badMAXFRC *(error)*
number of cycles used to create counters changed during exposure

badCCDSEL *(error)*
number of active CCDs changed during exposure

badCHOP *(error)*
value of chopper counter changed during exposure

6 Input Files

1. Exposure and quadrant specific EPIC PN events files (from **epevents**)
2. Auxiliary data set (part of corresponding ODF)
3. Main housekeeping data set (part of corresponding ODF)

7 Output Files

1. Filtered (=CCD specific) auxiliary file(s)
2. CCD specific frames list(s) with columns `FRAME`, `TIME`, `NEPDH`, `NEVENTS`, `NBADEVTS`, `FLAG`, `NDISCLIN`, `FC1`, `FC2`

3. CCD specific EXPOSURE extensions (added to the unmerged event lists). As **epframes** has already created such an extension and has filled in columns TIME, TIMEDEL= FRMTIME $\times FC0$ and FRACEXP= 1 for FRAME numbers listed in the CCD specific event list, this exposure extension will be further modified by **epexposure**: The number of frame counters will be increased to cover all changes in livetime factors and the FRACEXP will be multiplied by $FC1$ and $FC2$.

8 Algorithm

```
   subroutine epexposure

    Read all parameters
    Open input event list(s) and auxiliary file from single exposure

    * call evlistcomb
    in addition to epn default also propagate the FRAME column
    (this is needed to link the event list with CYCLE counters in the AUX file)

    * call evselect
    to create CCD specific filtered versions of the AUX file

    * call make_epexposure
    to actually perform the livetime computation

   end subroutine  epexposure

------------------------------

   subroutine make_epexposure

    Read all parameters (passed on from epexposure)
    Open merged events file
    Open filtered(=CCD specific) auxiliary file
    Open housekeeping file

    Open frames output file and fill the primary header
    Create frames table extension (one per CCD)
    (this is temporarily used to collect counter info
     and to store livetime factors per frame)

    Create and fill columns:
    FRAME
    TIME: computed via call OAL_frameCounterToObt(aux1frameCol,obttags)
                        call OAL_obtToTimeTag(obttags,timetags)
          for all frames which have an entry in the auxiliary file
          compute time tags for all remaining frames from time steps

    Extract chopper value of quadrant under consideration: An_CHOP
      from housekeeping file, error if An_CHOP not constant,
      store An_CHOP as attribute

    Extract number of cycles after which count info is sent: An_MAXFRC
      from housekeeping file, error if An_MAXFRC not constant
```

```
  NDISCLIN: extracted from auxiliary file, valid for An_MAXFRC frames

  Extract number of active CCDs in quadrant under consideration: An_CCDSEL
    from housekeeping file, error if An_CCDSEL not constant

  FC1: MIP dependent quadrant livetime factor
       (based on NDISCLIN, An_MAXFRC and An_CCDSEL)

  NEPDH: extracted from auxiliary file, valid for An_MAXFRC frames

  NEVENTS: computed number of raw events per frame from merged event list

  FC2: MIP dependent CCD livetime factor
       (based on EXPOSURE extension prepared by 'epframes')

  flag all frames as bad where NEPDH /= NEVENTS
  (if ground MIP correction is active, NEPDH has to be reduced by rejected
  counts)

  Modify CCD specific EXPOSURE extension: add enough FRAME entries
    to store changing livetime factors

  Compute FRACEXP entries: epframes took care of FC0, now
    additional multiplication with FC1 and FC2

  Compute livetime corrected exposure time and store it as attribute

end subroutine  make_epexposure
```

9 Comments

- Handling of out-of-time events: the number of out-of-time events depends on the spatial selection, i.e. on the amount of events which are assumed to originate from the target source, as well as on the readout mode. So this effect can not be handled as a livetime factor in an early stage of the pipeline processing but needs to be addressed as a mode-dependent PSF effect by higher SAS tasks.
- It has to be verified in orbit that there is a flat MIP distribution before one just can use the NDISCLIN counter to calculate $FC1$. If not this will imply the MIP deadtime correction has to be done properly on a column-by-column basis by looking at the Discarded Lines Map which will be transmitted at least once at the end of each observation.
- Handling of counting mode: One will need to check the EPDH counter to see how much time of an exposure was observed in counting mode (and hence did not produce any scientific data and entries in an event list). This deadtime will be stored via a good time interval (GTI) by the task **epframes**.
- Document EPIC-EST-OP-002, Sect. 14.7 mentions the scientific data structures 'First and Second Lost Synch Data' which are related to a loosing of the nominal scientific data flow. It needs to be verified that such a gap in the scientific data flow is addressed in the ODF creation already.
- For (e.g.) timing analysis there is a need to not only compute the statistical livetime factors but also to create a list of good time intervals. **epframes** already creates such

GTIs corresponding to the flagged frames. But also the deadtimes due to the chopper, the counting mode and CCD readout (which is significant for burst and small window modes) should be accessible as GTIs to be created by higher SAS tasks.

10 Future developments

References

[1] ESA. XMM Interface Control Document: Observation and Slew Data Files (XSCS to SSC) (SciSIM to SOCSIM). Technical Report XMM-SOC-ICD-0004-SSD Issue 2.5, ESA/SSD, June 2000. Found at the URL: `ftp://astro.estec.esa.nl/pub/XMM/documents/odf_icd.ps.gz`.

[2] SSC. XMM Survey Science Centre to Science Operations ICD for SSC Products. Technical Report XMM-SOC-ICD-0006-SSC Issue 2.1, SSC, Mar 2000.

[3] EPIC Team. EPCS electrical I/F specification. Technical Report EPIC-EST-SP-002 Issue 4, LABEN, April 1998.