XMM-Newton CCF Release Note

XMM-CCF-REL-298

EPIC-pn time jump parameters

R. Saxton & M. J. Freyberg

April 17, 2013

1 CCF components

Name of CCF	VALDATE	List of Blocks changed	Change in CAL HB
EPN_TIMEJUMP_0001.CCF	2000-01-01	TIMEJUMPTOL	YES

2 Change

The apparent arrival times of EPIC-pn events are subject to onboard jumps for a variety of reasons. Within the OAL, an algorithm is used to identify blocks of events which have experienced a jump and reset the times of these events to their correct value. This algorithm relies on extrapolating event times, by integral multiples of the frame time, to distinguish between the different types of time jump. The technique is limited by the stability of the EPIC-pn oscillator which has slowly degraded, in a non-linear manner, such that frame times are now a little longer than they were at the beginning of the mission.

The change in the duration of a frame has been calibrated for each observing mode and at a number of discrete epochs of the mission. This calibration, is stored within the EPN_TIMEJUMP CCF as a *tolerance* which may be used with the XMM-SAS to identify when a genuine time jump has occurred.

2.1 Oscillator

The intrinsic EPIC-pn time resolution is 40 ns; derived from a nominal oscillator frequency of 25 MHz. In Figure 1 we show the evolution of the oscillator frequency as a function of revolution number from measurements taken with the Cal-closed filter. The time resolution used in the EPIC-pn auxiliary information is actually 512 times the intrinsic resolution, equating to 20.48 μ s.

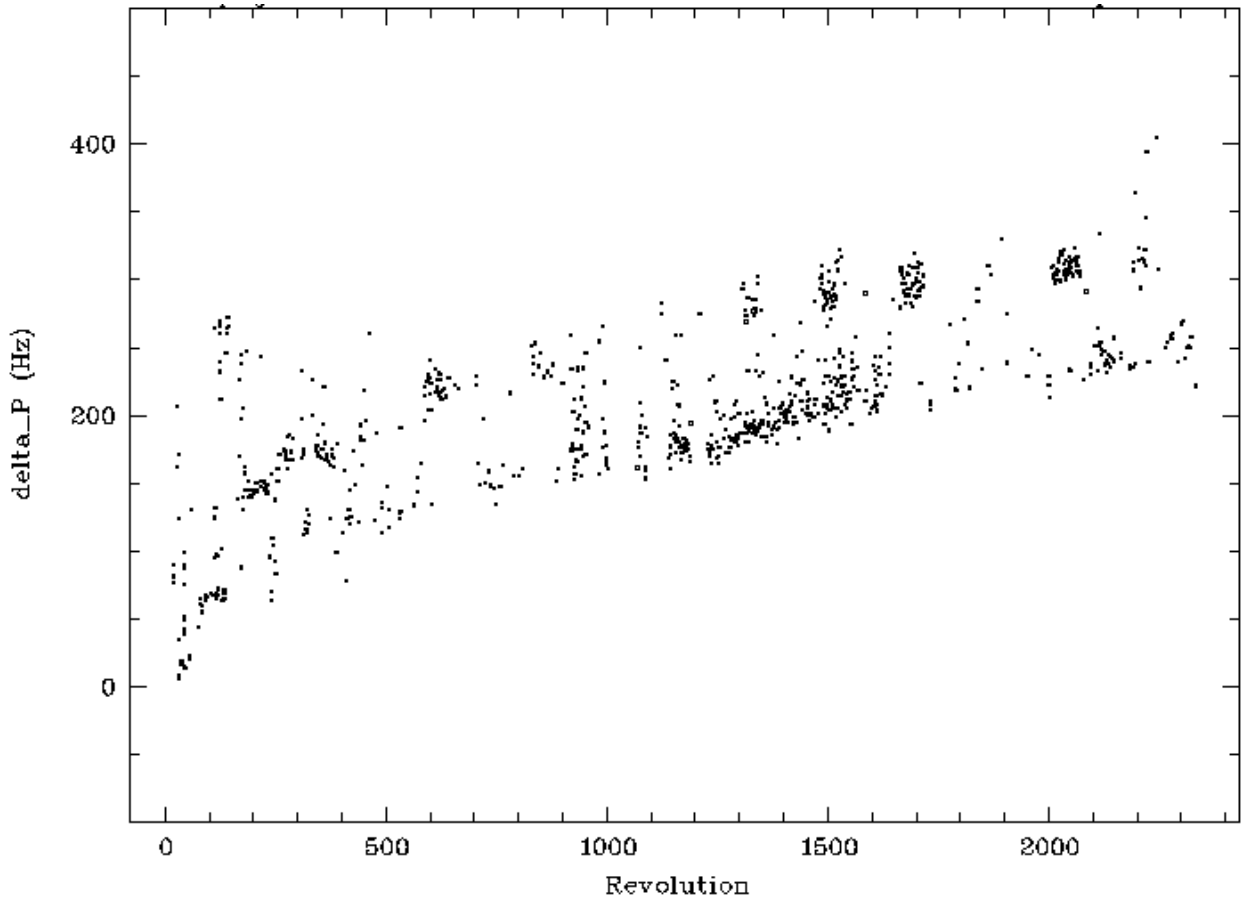

Figure 1: The frequency of the EPIC-pn oscillator as measured from CAL-closed observations at different epochs of the mission.

Table 1: Format of the CCF

Rev	TOL_FF	TOL_eFF	TOL_LW	TOL_SW	TOL_TI	TOL_BU
1	15	7	16	2	10	14
20	15	7	16	2	10	14
59	22	10	15	8	10	5
800	35	35	34	33	36	35

Event times are calculated by:

$$T = T0 + FTCOARSE + FTFINE * 20.48\mu s + Tcal$$

where FTCOARSE and FTFINE are the coarse and fine on-board counters.

The difference of 2 event times must be an integer multiple of the frame time; deviations indicate possible time jumps, for example when the FTCOARSE counter has been incorrectly incremented. Telemetry drops can lead to large gaps between events and hence, a high precision knowledge of the frame time is needed for an accurate identification of a time jump.

To avoid false detections of time jumps, caused by errors in the assumed frame time, the SAS uses a concept of a time jump tolerance. This is defined in FTFINE units and has been set in previous SAS versions to a default of $TOLERANCE = 22$. With this release, the CCF will be interrogated to find the appropriate tolerance to use for a particular revolution and observing mode. The algorithm is:

```
if ( MOD( (time2 - time1) / frame_time ) > (FTFINE * TOLERANCE) ) {
                    TIMEJUMP detected
}
```

The calibration data is stored in the CCF as revolution and time jump tolerance for each observing mode (see Tab. 1).

3 Scientific impact of this update

The false detection of time jumps has become an increasingly important problem during the processing of EPIC-pn datasets by `epproc` and `epchain`. This leads to blocks of events being assigned the wrong arrival times, which in turn leads to errors in downstream time-series analysis tasks.

With the new functionality, supported by this release, the fidelity of the time jump detection algorithm will be greatly improved, especially for more recent observations.

4 Estimated scientific quality

5 Test procedure and results

6 Future changes

This version of the CCF calibrates the time jump tolerance until revolution 2100. The tolerances for later revolutions are set equal to that of revolution 2100. This file should be updated periodically to include calibrations of the latest revolutions.

7 References