Timing of the FOS in RAPID mode – "Too Rapid RAPID"

J.A. Christensen[1], W.F. Welsh[2], I.N. Evans[1,3], M. Reinhart[1], and J.J.E. Hayes[1]

January 8, 1997

[1] STScI, [2] Keele University, [3] Smithsonian Astrophysical Observatory

---

ABSTRACT

We present a description of the problems that can occur when using the FOS in RAPID mode, including the "too-rapid RAPID" phenomenon. At high time resolution (i.e. READ-TIMES $\lesssim$ 10s) the data sampling rate is in general, faster than expected. Standard pipeline handling will add an additional $\frac{1}{8}$s uncertainty to the timings due to truncation. However the exposure times (thus count rates and fluxes) are unaffected, and the relative timings can be accurately reconstructed.

This ISR is written to make the General Observer (GO) aware of these problems and to recognise the symptoms in their data.

A technical appendix is included describing in detail the data acquisition operations of the FOS. This low-level description is mainly for those interested in the fine details of how the FOS acquires and stores data, and is included for completeness and clarity of technical issues.

A second appendix containing information on a program to determine group start times, and a third appendix containing useful equations for calculating exposure times are also included.

---

1 Introduction

The FOS can be used in RAPID mode to obtain spectrophotometric data with very high time resolution (for example see CAL/FOS ISR 124). However, because of the instrument's design, there is an inherent and unavoidable problem when pushing the FOS to its high speed limits: the actual sampling rate of the data will not be exactly the same as the requested sampling rate. This is known as the "too rapid RAPID" problem. We loosely refer to this aspect of the FOS's RAPID mode data acquisition as a "problem", though in fact it is merely the normal operation of the FOS.

The "too–rapid RAPID" phenomenon is most noticeable when the FOS is pushed to its high speed limits (READ–TIMES $\lesssim$ 10 s), though the effect is present for all READ–TIMES (and even non–RAPID mode operation). This effect shortens the amount of time between exposures, and can result in the data set containing more groups than anticipated. The term "too–rapid RAPID" is a misnomer, but in this ISR we continue to use it in its most broad sense, meaning *any* timing problems associated with FOS RAPID mode observations.

We briefly explain the symptoms, the cause, and the cure for these timing "problems". The description is purposefully kept simple and focused on the needs of the GO. For a more detailed description of how the FOS works and why the too–rapid RAPID phenomenon occurs, a technical appendix is included.

2 Problems and Symptoms

When using the FOS in RAPID mode, the data sampling rate may not be what was requested. On rare occasions the sampling rate may not even be regular. This can be a major problem for analyses which require exact temporal sampling (*e.g.* fast Fourier transforms).

Another problem is that of absolute timing. The absolute start time of an observation is not known to better than 0.255s. This determination cannot be improved after the data have been taken. However, while the gaps between observations are not what was requested, they are precisely known (i.e. the relative times with respect to the first observation are known to $< 1\ \mu$s).

The most obvious symptom of the too–rapid RAPID effect is the unexpected sampling rate. Another aspect of the phenomenon is that the sampling rate is faster than anticipated. This is because the actual time needed to store the data is less than the allocated time. Because in many cases the FOS is commanded to take data for a certain length of time (not a certain number of spectra), the faster sampling rate may result in more data (groups) than expected. But this is not always the case– it is possible for the observer to get the exact number of groups expected, but since the sampling rate is faster, the elapsed time (duration of the observation) is less. Another symptom is that the keyword EXPTIME in the data header is the proposed time not the actual time spent observing, so it appears to be too short.

Note that the actual dwell or live time (amount of time spent actually accumulating photons) remains constant — it is the spacing between the start times of the integrations that can vary. The count rates (and hence fluxes) are correct.

3 Why the problems occur

There are two major factors involved, one dealing with the data processing and the other with the FOS firmware itself. First, the standard CALFOS pipeline reduction does not record the times of observations more accurately than 0.125s. Hence the times derived from the data header keyword, FPKTTIME, suffer this $\frac{1}{8}$s truncation. To find the actual time of observation requires going back to engineering telemetry data. This is now possible (see section 5) and very accurate *relative* timings are possible.

The second factor is inherent to the FOS itself. The FOS collects data by performing a series of operations inside several levels of nested loops. These include such things as accumulating data, transferring data to memory, substepping, overscanning, etc. The FOS microprocessor which controls these functions does not operate alone, but must coordinate activity with other HST operations.

In a very simplified picture, the FOS collects data in its accumulators, then temporarily stores this single observation in its internal memory. The contents of the FOS memory is then transfered to an external storage device (*e.g.* the science tape recorders).

The accumulators can acquire data while the internal memory is being read out and stored to tape. This parallel process effectively reduces the readout time of the instrument, resulting in a higher sampling rate than anticipated. This is the cause of the too–rapid RAPID. However, it is also possible that when the accumulators finish acquiring data and are ready to place the information into the FOS memory, the FOS memory has not been completely read out and stored onto the tape recorder. This "collision" results in the contents of the accumulators being discarded and the integration is re-executed. Thus because of the "handshaking" between the FOS microprocessor and the data storage systems, the start times of each observation (group) can vary.

While in almost all cases the *de facto* sampling rate is regular, (*i.e.* constant to $< 1\mu$s), there may be cases when this is not true. Investigations have shown, for example, that irregular sampling (greater than a 4% deviation from a constant sampling rate) occurred when SUBSTEP=4 with READTIMEs close to 6.4s. Thus regular sampling can not be ***guaranteed***. Keep in mind that the minimum possible timing "jitter" will be one LIVETIME + DEADTIME cycle, and in this case, due to the parameters chosen, that happens to be 4% of the READTIME.

One final additional factor: there is a delay between the time the FOS hands over the data and the time this action is acknowledged and a timestamp placed on the data. This delay is random and impossible to determine, and results in an uncertainty of 0.255s in the absolute timing. The relative timing between groups is not affected by this.

4 When does the too rapid RAPID occur?

Because the too–rapid RAPID phenomenon originates from low–level hardware/software design, it is impossible to prevent its occurrence. It is also impossible to predict its occurrence, other than giving broad regimes of when the effect may or may not occur.

The problem arises if the maximum Read–Out Time (ROT) is less than one live-time+deadtime cycle (known as an integration or "INT"). From the header keywords, an INT = (LIVETIME+DEADTIME) x 7.8125E–6s (the numerical factor converts spacecraft clock ticks to seconds).

If the minimum ROT is greater than an INT, the too–rapid RAPID will not occur. If the ROT is comparable to an INT, the too–rapid RAPID ***may*** occur.

The minimum and maximum read–out times (in seconds) can be computed via:

max ROT = **[(15/14)x(1.024/COMRATE) x SEGMENTS x LINES] + F**

min ROT = **[(15/14)x(1.024/COMRATE) x (SEGMENTS–1) x LINES] – F**

where

LINES = SUB–STEP x StepPatt x SLICES,
SEGMENTS = 1 + *ceiling*{**(WORDS–50)/61**},
WORDS = NCHANNELS + (OVERSCAN - 1),

and

COMRATE = telemetry rate in kbits/s; = 32 (for low telemetry rate), or 365 (for high telemetry rate);
SUB–STEP = header keyword NXSTEPS;
StepPatt = also called YSTEPS and is usually 1 (unless using a paired aperture or the obj–sky–background option);
SLICES = 1;
NCHANNELS = number of diodes read out, usually 512;
OVERSCAN = header keyword, usually 5;
ceiling { } = function, round up to the next highest integer;
F = a safety factor, set at 0.010s.

As an example, consider the following: say an observation required 20 INTS to obtain the proper amount of flux. INT number 1 is accumulated in one section of memory (called memory 1). When that INT is complete it is read into another section of memory (called memory 2), where it waits to be transfered to the tape recorder. If the INT accumulating in memory 1 finishes before memory 2 is read to the tape recorder there is no where for the INT in memory 1 to go. Therefore, it is ignored and the hardware pretends that it never took that particular INT. The counter, which believed it was accumulating INT number 2 never advances so the instrument then begins again and re-executes INT number 2. Meanwhile the amount of *time* that has passed is equal to the amount of time required to take 3 INTS. Hence the flux accumulated is unaffected, but the time between the start of one observation and the start of another can vary due to INTS being disregarded when there is a memory allocation traffic jam.

5 What to do about it

Under normal circumstances, the header keyword FPKTTIME can be used to determine the times of each observation (group), remembering that FPKTTIME is accurate only to –0.0/+0.125s. The –0.0/+0.125s accuracy on the FPKTTIME is due to the fact that FPKTTIME is truncated, not rounded. Since it is truncated, it will never be less than what was recorded, but it may be up to +0.125s later than recorded. There is an additional uncertainty of –0.255s/+0.125s on the absolute start time where the –0.255s comes from up to a 0.255s delay between the accumulator closing and the posting of the FPKTTIME. Hence the *actual* FPKTTIME could have occurred as much as -0.255 seconds ago, but will never be *later* than the recorded FPKTTIME. The positive uncertainty of +0.125s on the absolute start time is due to the truncation mentioned before. The ***start*** of the observation can be determined from the following equation where the variables are header keywords:

t_{start} = FPKTTIME – (((LIVETIME + DEADTIME) * INTS * NXSTEPS * OVERSCAN * YSTEPS * SLICES * NPAT) – DEADTIME) * 7.8125E–6

To get an estimate of the interval between the start times of group 1 and group 2 you take the FPKTTIME from group 2 and subtract the FPKTTIME from group 1. Remember the accuracy of this operation will only be good to -0.0/+0.125 seconds. (Note: FPKTTIME is in units of Modified Julian Date–days.)

If you need higher precision for your science, you will need to contact the FOS team. The telemetry data will be de-archived and processed though some very specific software which will not work outside of STScI. The output will give the start time of the observation (good to –0.255/+0.125s) and the relative times of all subsequent observations (good to $\sim 1\mu$s; the actual precision is better, but the times are stored in single precision). Information regarding this program are included in Appendix B.

6 Advice to Observers

Because of the too–rapid RAPID effect, the advice given in CAL/FOS ISR #124 on RAPID mode operations of the FOS should be used only as a rough guideline. The longer the READ–TIMEs, the less of an effect the too–rapid RAPID will have, so for observations which have READ–TIMES longer than $\sim$ 10s, ISR 124 should hold.

For observations which do not require equal temporal sampling better than 0.125s ($\frac{1}{8}$s), the too–rapid RAPID phenomenon should have little to no impact.

Because it is impossible to obtain data of the exact proposed sampling when using short READ–TIMES ($\lesssim$ 10s), any science that demands high temporal resolution and strictly regular sampling, and/or very accurate absolute timing should not be carried out with the FOS.

Observations that require precise timing information but can tolerate sampling that is different than the requested READ–TIME are in the regime where the FPKTTIMEs are too crude, and the FOS team must be contacted to compute the start times of each group.

In general, if the too–rapid RAPID effect is occurring, the data sampling rate will be shorter then specified (shorter than the READ–TIME parameter). As a result, more spectra (groups) will be obtained. This is not strictly true however, as things like filling the tape recorder or scheduling constraints can result in obtaining fewer data than expected.

Observers attempting to use the FOS as a very precise high–speed photometer should remain aware that the absolute start time of an observation cannot be determined to better than 0.255s. While the relative times can be known to better than a microsecond, the effects of light travel time across the HST orbit is not accounted for.

7 For Further Help

Consult the most recent FOS Instrument Handbook. We recommend reading through ISR 124, "High Speed Spectroscopy using the FOS in RAPID Mode". Both documents are available on line at:

`http://www.stsci.edu/ftp/instrument_news/FOS/fos_doc_access.html`

The 1990 FOS Instrument Handbook V1.1 contains additional useful engineering– level information, but we offer this as a suggested reference document, rather than required reading.

If you find that you are in the "too–rapid RAPID" regime, or want further clarification, you should contact the FOS team. Send your query to help@stsci.edu.

Acknowledgements: We wish to thank Tony Keyes, Don Chance, Michael Dahlem, and Ed Smith for useful discussions and assistance.

8 References

Welsh, W.F, Chance, D., Keyes,C., Reinhart, M.; 1994 "High Speed Spectroscopy using the FOS in Rapid Mode – CAL/FOS ISR 124", (Baltimore, STScI)

Ford, H. C. & Hartig, G. F. (eds.); 1990 "Hubble Space Telescope Faint Object Spectrograph Instrument Handbook V1.1", (Baltimore, STScI)

Keyes, C.D.,*et al.* ; 1995 "Hubble Space Telescope Faint Object Spectrograph Instrument Handbook V6.0", (Baltimore, STScI)

Also: "The Case of the Too Rapid Rapid", an internal engineering memo by John Fitch and Rick Hier, 1992 April 7, (Baltimore, STScI; not publicly circulated)

9 APPENDIX A: FOS Data Acquisition

(Adapted from a memo circulated by Ian Evans, 1995 September)

Below is included a simplified description of FOS data taking under normal operations with standard instrument modes, which includes a description of the "too–rapid RAPID" phenomenon and information about the accuracy of "First Packet Time", FPKTTIME.

The description assumes that the instrument is in reject mode, double precision adder mode, and non–time–tagged mode. These correspond to normal operations (e.g., ACCUM, RAPID, PERIOD). Timing and other information may vary for special, non–standard instrument modes. The description assumes also that the minimum LIVETIME and DEADTIME values required for operation are honored.

I. FOS Observation Loop Control

In the text below, the term "parameter" is used in a loose sense to describe both the commanded value and the telemetered value. For convenience, the parameters are referred to by the name of the telemetered value as stored in a generic edited uncalibrated data set header file (D0H).

FOS observations are performed under the control of several nested loops, executed by the FOS microprocessor.

The fundamental observation time element is the LIVETIME, which determines the length of time that the FOS accumulators are open (i.e., the interval during which the accumulators will count photon events detected by the diode array) during each execution of the innermost loop. The LIVETIME is specified as a number (0–65535, with a value of zero meaning 65536) of 7.8125 μs time intervals. The 7.8125 μs time interval is equal to 8 ticks of the spacecraft 1.024 MHz (+/– 50 Hz) clock. The accumulator opening and closing is controlled in hardware by the spacecraft clock, and thus the duration of a LIVETIME interval should have a similar accuracy to the accuracy of the clock.

After the LIVETIME expires, the accumulators close (i.e., they will no longer count photon events detected by the diode array) and remain closed for a period determined by the DEADTIME, which is also specified as a number of 7.8125 μs intervals of the spacecraft 1.024 MHz clock.

During the DEADTIME, the FOS microprocessor performs several computations, and controls several instrumental functions, relevant to the observation. One of the most important of these operations is to transfer the contents of the accumulators to the appropriate locations in the FOS memory (see section III, below).

Each LIVETIME + DEADTIME cycle is termed an INT (integration). As described in section III of this appendix, during each DEADTIME, noise burst rejection (controlled by the REJLIM parameter) is applied to the data gathered during the most recent LIVETIME.

The innermost loop repeats the LIVETIME + DEADTIME cycle serially the number of times specified by the INTS parameter, which may range from 0 to 255 repetitions, with a value of 0 meaning 256.

The second level loop controls substepping of the diode array in the X–direction. The (LIVETIME + DEADTIME) * INTS cycle is repeated serially the number of times specified by the XSTEPS parameter (0 to 255 repetitions, with 0 meaning 256). Between each repetition, the Digicon X–deflection is altered so that each successive XSTEP (also called SUB–STEP) will cause the spectrum to be deflected by an additional value of 1/XSTEPS times the diode X–spacing. The section of the photocathode that would be imaged onto the center of diode n if XSTEPS is set equal to 1, will be imaged onto the following locations if XSTEPS = 4: (center of diode n), (center of diode n + 1/4 diode X–spacing), (center of diode n + 2/4 diode X spacing), and (center of diode n + 3/4 diode X–spacing).The steps go in the +X direction. The accumulated photon counts for each XSTEP are recorded in separate locations in the memory, so they can be reconstructed in wavelength order during later processing.

The third level loop controls stepping of the diode array in the X–direction over an integral number of diodes (as opposed to fractional diode stepping controlled by the XSTEPS parameter). The (LIVETIME + DEADTIME) * INTS * XSTEPS cycle is repeated serially the number of times specified by the OVERSCAN parameter (0 to 255 repetitions, 0 meaning 256). Since the diode array consists of 512 diodes, values larger than 128 are not meaningful. Similar to the XSTEP loop, the Digicon X deflection is altered so that each successive OVERSCAN step will cause the spectrum to be deflected by an additional diode X spacing. Having a value of OVERSCAN > 1 allows for compensation for disabled diodes, which would otherwise leave an unrecoverable gap in the data. The accumulated photon counts for OVERSCANned data are summed into the memory locations for each XSTEP, as appropriate. For example, with XSTEPS = 4 and OVERSCAN = 5, a total of 20 successive X deflections would occur, with each deflection separated by 1/4 diode X spacing. The memory location that stores the counts from diode n for the first of these steps would be incremented by the counts from diode n + 1 for step 5, from diode n + 2 for step 9, and so on.

The fourth level loop allows stepping the diode array in the Y–direction. The (LIVETIME + DEADTIME) * INTS * XSTEPS * OVERSCAN cycle is repeated serially the number of times specified by the YSTEPS parameter (0 to 255 repetitions, 0 meaning 256). Between each repetition, the Digicon Y–deflection is altered so that each successive YSTEP will cause the spectrum to be deflected by an additional amount equal to the value YSPACE = (YRANGE * 32) / YSTEPS Y–bases in the +Y direction.

The fifth level loop controls gathering of time–resolved data (e.g., PERIOD mode). The (LIVETIME + DEADTIME) * INTS * XSTEPS * OVERSCAN * YSTEPS cycle is repeated serially the number of times specified by the SLICES parameter (0 to 255 repetitions, 0 meaning 256). Data from each SLICE are stored in separate memory locations, allowing subsequent processing by the TIME RESOLVED ground mode software to distinguish data collected in each bin for PERIOD mode observations.

The above sequence of nested loops, (LIVETIME + DEADTIME) * INTS * XSTEPS * OVERSCAN * YSTEPS * SLICES, comprises a unit termed a PATTERN.

The sixth level loop controls the number of PATTERNs executed between memory readouts. The PATTERN is repeated serially the number of times specified by the NPAT parameter (0 to 255 repetitions, 0 meaning 256). After each NPAT PATTERNs have executed, the data stored in memory are read out and dumped to the Science Data Formatter (SDF). This readout is non-destructive (i.e., the data stored in FOS memory are not changed), and is termed a Science Data Dump. Each Science Data Dump will produce a new group of data in the generic edited uncalibrated data set (D0H). Because the length of time between the accumulators closing and the completion of the Science Data Dump is dependent on hardware handshaking and other loads on both the FOS microprocessor and the SDF, time phasing across a readout cannot be guaranteed.

The seventh level loop controls the number of times the PATTERN * NPAT cycle executes between FOS data memory clears. The PATTERN * NPAT cycle is repeated the number of times specified by the NREAD parameter. The value of NREAD can range from 0 to 255 repetitions. A value of zero results in an indefinitely long observation, which will continue until stopped by a separate command to terminate the observation. If NREAD = n, then after the readout corresponding to the nth repeat of the PATTERN * NPAT cycle, the locations in FOS memory storing the count data are filled with zeros.

The outermost level loop controls the number of times the PATTERN * NPAT * NREAD cycle must execute to complete an "observation" (termed an "acquisition" in the instrument related documentation). The PATTERN * NPAT * NREAD cycle is repeated the number of times specified by the NMCLEARS parameter. Like NREAD, the value of NMCLEARS can range from 0 to 255 repetitions, with a value of zero resulting in an indefinitely long observation, which will continue until stopped by a separate command to terminate the observation.

II. ACCUM, RAPID, and PERIOD Mode

One can now see immediately that ACCUM mode, RAPID mode, and PERIOD mode result simply from changes to the NREAD, NMCLEARS, and SLICES parameters.

For an ACCUM mode observation, only one set of data is accumulated at a time, the data are read out regularly, and the memory is not cleared between readouts, so that each successive readout contains the sum of the counts observed during the current and all previous readouts. Thus, SLICES must be set to 1, NMCLEARS must be set to 1, and NREAD will be set to the number of repeats needed to accumulate the required observing time (or to 0, in which case the observation will be terminated by an End Data Acquisition command).

For a RAPID mode observation, only one set of data is accumulated at a time, the data are read out regularly, and the memory is cleared between successive readouts. SLICES must again be set to 1, but in this case, NREAD must be set equal to 1, and NMCLEARS will be set to the number of repeats needed to accumulate the required observing time (or to 0, in which case the observation will be terminated by an End Data Acquisition command).

For a PERIOD mode observation, SLICES sets of data are accumulated at a time, with each slice representing a different phase interval of the target period. The data are read out regularly, and the memory is not cleared between readouts, so that each successive readout contains (for each slice individually) the sum of the counts observed during the current and all previous readouts (for each slice individually). SLICES will be set to the value specified by the observer for the number of phase intervals (i.e., BINS), NMCLEARS must be set to 1, and NREAD will be set to the number of repeats needed to accumulate the required observing time (or to 0, in which case the observation will be terminated by an End Data Acquisition command).

III. What Happens when the Accumulators Close?

The accumulator close routine in the FOS firmware checks the status of the Science Dump In Progress (SDIP) flag prior to transferring the contents of the accumulators to FOS memory. If the SDIP flag is set, indicating that a Science Data Dump is occurring, then the accumulator contents are not transfered to FOS memory because the memory is being accessed for data transfer. Instead, the accumulator contents are discarded, and the PATTERN parameters (including the count of INTs, Digicon deflections, etc.) are not updated. The next INT will be a repeat of the discarded INT, and will use the same PATTERN parameters. If the SDIP flag is clear, then reject array processing occurs (as stated earlier, we assume REJECT mode in this description), and then the accumulator contents are transfered to the FOS memory (assuming that REJLIM is not exceeded). In normal operation, without a user specified value for REJLIM, REJLIM is set to FFFF hex, which makes the rejection limit unattainable. Finally, the PATTERN parameters are updated according to the next level loop control step. Note that the setting of the SDIP flag at the time it is tested determines whether the current INT is discarded. A Science Data Dump can occur in parallel with the accumulators being open; if the Science Data Dump terminates and clears the SDIP flag prior to the flag being tested by the accumulator close routine, then the current INT will not be discarded, and the data are valid.

If a Science Data Dump should be required after the end of a PATTERN, then the following timing applies. The accumulator close routine, which takes a maximum of 0.030s to complete, initiates the Science Data Dump. The Science Data Dump is initiated when the FOS issues a FRAME START pulse on the interface to the SDF. The amount of time that elapses between the accumulator close routine initiating the dump and the FOS issuing a FRAME START is not defined, but the Activate Science Data Dump command (which performs this action) is not included in the list of commands that require more than 0.050s to complete. Under normal circumstances, the FRAME START pulse will be acknowledged by the SDF issuing a LINE START pulse within 0.001–0.175s. In principle, if the science data are being written to a Science Tape Recorder (STR), and the STR must be brought up to speed prior to writing data, then the FRAME START pulse could be as long as 10 seconds. However, observation commanding currently writes the Science Header Packet (and Unique Data Log) prior to dumping science data, so the STR is currently always up to speed prior to the Science Data Dump, and so this caveat should not apply. The time code recorded in the data packet (the "First Packet Time," FPKTTIME) is the spacecraft clock time at the time the LINE START pulse is initiated. The FPKTTIME will be RANDOMLY delayed by up to 0.255s after the accumulator's closing at the end of the LIVETIME comprising the last INT in the pattern. So FPKTTIME cannot be used to determine the timing of the observations more accurately than the uncertainty associated with this random delay.

Although the delay between accumulator close and LINE START is not constant, as long as neither the FOS microprocessor nor the SDF are busy with other operations, variations from readout to readout should be fairly small. Inspection of a limited set of RAPID mode data showed that the spacing between successive FPKTTIMEs was approximately constant, with a one sigma variation of order 0.0002 s, and with maximum deviations of +/– 0.002s. Although this is not necessarily true for all observations, one can reasonably assume that an uncertainty of +/– 0.010–0.020s should be fairly robust.

IV. Exposure Start Time

The start time of the exposure associated with a group of data can be estimated from the FPKTTIME of that group of data using the equation

START TIME = FPKTTIME – ([(LIVETIME + DEADTIME) * INTS * XSTEPS * OVERSCAN * YSTEPS * SLICES * NPAT] – DEADTIME) * 7.8125E–6.

The extra deadtime term comes from the fact that the FPKTTIME is set by the science data dump initiation, which occurs shortly after the livetime expires. The actual beginning of the observation may occur up to approximately 1/4 second before the START TIME computed from this formula (because of the delay between accumulator close and posting of FPKTTIME), or up to approximately 1/8 second after the calculated START TIME. The latter possibility arises because the FPKTTIME stored in the generic edited information set is obtained from the actual spacecraft time by truncating the low order bits, rather than by rounding the value.

Within a single "observation" ("acquisition"), the relative timing between readouts (i.e., groups of data) must be an integral number of INTS, since the accumulator open timing is controlled by the hardware clock.

More precise absolute timing is only possible using the Synchronous Start instrument mode, which ties the first accumulator open to a predetermined spacecraft clock time.

V. Alignment Times and the "Too–Rapid Rapid" Phenomenon

The amount of time allocated in an alignment for each PATTERN * NPAT cycle (i.e., the alignment time contributing to each group/readout) is the exposure time plus the ***maximum*** readout time which is computed using

ROT = (15/14) * (1.024 / COMRATE) * SEGMENTS * XSTEPS * YSTEPS * SLICES.

In this equation, COMRATE is either 32 (low) or 365 (high) (in units of kbits/sec), and SEGMENTS is 1 + CEILING[(WORDS – 50) / 61] for WORDS > 51, or 1 otherwise. WORDS = NCHANNELS + (OVERSCAN – 1), where NCHANNELS is the number of diodes to be read out. The factor (15/14) accounts for the Reed-Solomon ECC coding.

The actual time taken for the readout will be less than the allocated ROT if (WORDS - 50) is not an integer multiple of 61.

Although the alignment time allocated for each PATTERN * NPAT cycle plus readout is [(LIVETIME + DEADTIME) * INTS * XSTEPS * OVERSCAN * YSTEPS * SLICES * NPAT] * 7.8125E-6 + ROT, since a Science Data Dump can occur in parallel with data taking (see section III), the actual time taken may be less than the allocated time. This can result in an effect that has been termed "Too–Rapid Rapid" which may alter relative start times of the groups from their "expected" values. The term "Too–Rapid Rapid" is a misnomer, since the effect can occur for any mode.

No allowance is made for the delay that occurs between accumulator close and LINE START — the assumption is that this delay is constant from readout to readout (and so the time between readouts is assumed to be the same). Although not strictly correct, this assumption is sufficiently accurate in terms of instrument operation. The only effect is to delay the end of the final readout by up to $\sim$ 1/4 second after the expected time, and there is sufficient pad time at the end of an observation so that this delay does not cause a problem.

After the accumulators close following the last LIVETIME prior to a readout, the Science Data Dump is initiated as described in section III. If the readout completes and the SDIP flag is cleared prior to being tested during the accumulator close routine execution during the DEADTIME of the following INT, then the Science Data Dump has effectively required zero additional alignment time beyond the exposure time, even though an amount ROT was allocated.

VI. Examples

Consider the following example, where LIVETIME = 12800 (0.100s), DEADTIME = 14080 (0.110s), and INTS, XSTEPS, OVERSCAN, YSTEPS, SLICES, and NPAT are all unity. (Note that this DEADTIME value is only an example and is actually a factor of ten larger than expected). If 512 diodes are read out, then at the low data rate, ROT is approximately 0.309s. In this case, 0.519s (= 0.100s + 0.110s + 0.309s) alignment time is allocated for each group exposure plus readout. The actual readout time will be somewhat smaller than 0.309 s, as the last SEGMENT will be short. However, the actual readout time (ROT) is considerably longer than the length of the INT (=0.210s), so the Science Data Dump for group n will still be in progress (and therefore the SDIP flag will be set) when the following LIVETIME completes. Thus the data from the successive LIVETIME will be discarded, as per section III. Although this data is lost, the observation will repeat, since the successive INT will repeat, and so there will be no missing groups. The exposure start (accumulator open) for group n + 1 will occur 0.420s [= 2 * (0.100s + 0.110s)] after the exposure start or group n. (We are assuming here for simplicity that the delay between accumulator close and testing the SDIP flag is constant from readout to readout.) Thus while the expected time between two groups is 0.519s, the actual time is 0.420s, and although it is hidden from the observer (who actually gets data faster than expected), every other observation has been discarded.

If in this example we change LIVETIME to 64000 (0.500s), then 0.919s (= 0.500s + 0.110s + 0.309s) alignment time is allocated for each group exposure plus readout. Since the ROT is considerably shorter than the length of the INT, the readout of group n will occur completely in parallel with the data taking for group n + 1. Thus, the exposure start (accumulator open) for group n + 1 will occur 0.610s (= 0.500s + 0.110s) after the exposure start for group n, much sooner than the expected 0.919s between groups. Again the sampling rate is faster than expected, but in this example no data are discarded.

Suppose further in this second example that 250 groups of data are expected. In RAPID mode, this would be commanded by setting NMCLEARS to 250. The total alignment time allocated would be 250 * 0.919s = 229.75 s. However, the observation would only require 152.5s (=250 * 0.610s), and would finish early. Since an exact number of repetitions was commanded, that number will be received. However, the start times of the groups would be separated by 0.610 s rather than 0.919s as may be naively expected. Thus while the expected number of groups are in fact obtained, the sampling rate is faster and the total elapsed time is less (the observations end early).

If however 1000 groups of data are expected, this number of repetitions cannot be commanded explicitly (section I), so instead NMCLEARS will be set to zero, and an "end data acquisition" command will be issued at the appropriate point near the end of the alignment time to terminate the observation. The computed alignment time for 1000 groups would be 1000 * 0.919s = 919 seconds. Since each group start time will be separated by 0.610s from the previous group start, 1506 groups of data will be received by the time the "end data acquisition" command is issued, instead of the 1000 expected. Thus the observer gets 50% more data than expected!

This latter example has been called the "Too–Rapid Rapid" case. Note however that even in the first example, which would not be called "Too–Rapid Rapid" as the term has been used, the same effect occurs. In fact, the "Too–Rapid Rapid" effect must always occur since the spacing between the exposure start times for successive groups is always less than the allocated alignment time.

VII. Relative Group Timing

The relative group start times are separated by an interval

Δt = ((LIVETIME + DEADTIME) * INTS * XSTEPS * OVERSCAN * YSTEPS * SLICES * NPAT + [N * (LIVETIME + DEADTIME)]) * 7.8125E–6,

where N is given by FLOOR[SDIP$_{time}$ / ((LIVETIME + DEADTIME))* 7.8125E–6]. In this equation, SDIP$_{time}$ is the actual duration when the SDIP flag is high. This will vary slightly from readout to readout, because of the random delay between accumulator close and LINE START. However, the time will be approximately equal to the actual readout time.

Note that the actual readout time will fall between the minimum value ROT = (15/14) * (1.024 / COMRATE) * (SEGMENTS – 1) * XSTEPS * YSTEPS * SLICES and the maximum value ROT which is a factor of (SEGMENTS)/(SEGMENTS–1) larger.

Note that the relative timing between each pair of groups will normally be the same (although it will always be less than the alignment time value one naively expects), since the value N above should be relatively stable if the scatter in the SDIP$_{time}$ is similar to the FPKTTIME scatter observed in section III.

However, if SDIP$_{time}$ $\sim$ [(LIVETIME + DEADTIME) * 7.8125E–6] then the relative timing between each pair of groups may vary. Assuming SDIP$_{time}$ is approximately equal to the actual readout time, we can divide phase space into three regions:

- If (15/14) * (1.024 / COMRATE) * (SEGMENTS – 1) * XSTEPS * YSTEPS * SLICES > [(LIVETIME + DEADTIME) * 7.8125E–6] + (0.010 to 0.020s),
then relative group timing should be stable. This corresponds to the non–"too–Rapid Rapid" mode as the term has been used.

- If (15/14) * (1.024 / COMRATE) * SEGMENTS * XSTEPS * YSTEPS * SLICES < [(LIVETIME + DEADTIME) * 7.8125E–6] – (0.010 to 0.020s),
then relative group timing should be stable. This corresponds to the "too-Rapid Rapid" mode as the term has been used.

- Otherwise, the relative group timing may vary from group to group.

Here we have used a pad of 0.010–0.020s for the same reason as discussed in section III in association with FPKTTIMEs, and the same cautions and assumptions apply.

10 APPENDIX B: INSTRUCTIONS FOR USING THE RAPID-TOOL

This program, created by Merle Reinhart, is not exportable and will only work on the STScI VMS computer system. If you are not at STScI and feel that your data requires timing information more accurate than $\frac{1}{8}$of a second, please contact help@stsci.edu. The FOS team will then be notified and your data will be run through this program. If you are at STScI you should use the following procedure.

Due to the nature of the calculations and file structures involved this must be a VMS specific program.

In your login.com add this line:

```
@disk$user4:[reinhart.fos-rapid]setup
```

This will define some logicals.

Next get the .d0h, .shh and the .pkx files out of the archive. Be sure to retrieve them to the VMS side. Transfering them from unix causes problems. Retrieve the .d0h and .shh files as normal (override standard file options and ask for CAL class, extensions .d0f,.shf). To retrieve the .pkx files you will have to jump through some hoops. Do everything the same, but again, override standard file options. For class enter EDT and for extensions enter nothing – just leave it blank. It will retrieve all EDT files but this is the only way to do it. There will be 6-8 per dataset name.

The tool requires that the .d0h,.shh and .pkx files all be present in the same place. It will not do anything if it doesn't find all three.

The command to run the tool (should work from any vax on the science cluster) is:

```
> rapid_times rootname
```

where rootname is the rootname of your dataset (ie y20pq301t). It understands wildcards (*).

It will print output to your screen. This is mostly to help with diagnostics, but to be safe, you may want to save this output to a file. The output will warn you of FILL data–a common occurance in rapid-mode. It will produce an ascii output file with the same rootname as your observation but with the extension .rpd. This file will list the absolute start time (with errors) and then the relative times of each successive group. To get the absolute start time of each group add the relative time to the absolute start time. The units are seconds. It will also flag the first set of groups that have FILL data in them.

11 APPENDIX C: USEFUL EQUATIONS FOR CALCULATING EXPOSURE TIMES

Upper case type indicates a keyword that can be found in the data headers.

Exposure time per group is the integration time for a given group, or rather, integration time per diode in units of seconds.

$Exposure\ Time_{group}$ = EXPOSURE * NXSTEPS

Elapsed time per group will be longer than the exposure time per group because of the added overhead time due to DEADTIME etc.

$Elapsed\ Time_{group}$ = (((LIVETIME + DEADTIME) * INTS * NXSTEPS * OVERSCAN * YSTEPS*SLICES*NPAT)–DEADTIME) * 7.8125E–6

Start time of a particular group will be:

$Start\ Time_{group}$ = FPKTTIME$_{group}$ – $Elapsed\ Time_{group}$

Read–Time, which is a proposal keyword, is not found in the headers. This will be the sampling rate the observer requested. Read–Time = $Elapsed\ Time_{group}$ + Read Out Time(ROT). The Read-Time is equivalent to the "alignment time".

The Exposure Time for the whole observation will be the Exposure Time per group times the number of reads and clears:

$Exposure\ Time_{observation}$ = EXPOSURE * NXSTEPS * NREADS * NMCLEARS

And finally, the Elapsed Time for the observation will be:

$Elapsed\ Time_{observation}$ = $Elapsed\ Time_{group}$ * NREADS * NMCLEARS