NASA Planetary Data System GRaND Experimental Data Records: Overview of IDL functions in Extras

Tom Prettyman
Planetary Science Institute

V1.0

11-Jul-2013

Introduction

- IDL functions accompany the Experimental Data Records (EDR) for the NASA Dawn mission's Gamma Ray and Neutron Detector (GRaND)
  - See the Extras directory for these routines
- The purpose of these functions is to provide examples of how to read data from the EDR archive
- This presentation shows how the EDR can be explored using the IDL functions provided
- Supplementary quicklook reports, which summarize the data in each EDR directory, are also described

Nomenclature

- A list of acronyms and abbreviations relevant to the GRaND EDR is provided in Extras
  - GRD_nomenclature.doc

What are Extras?

- Extras are supplementary documentation, programs, and data that can be provided along with archived data
- Extras are not a required element of the Planetary Data System (PDS) archive and need not conform to specific standards; however,
- The information provided may be helpful to users in understanding or interpreting the archived data

What is IDL?

- The “Interactive Data Language” (IDL) is scientific programming language that is widely used in the planetary science and astronomy community for data reduction, analysis, and visualization
  - Presently distributed by Exelis Visual Information Solutions
- Why IDL? Why not Fortran, C, Python, or even Excel?
  - The EDR can be read using just about any programming language and platform
  - Since the GRaND team and many of the gamma/neutron community uses IDL for data processing, the examples are provided as IDL functions
  - IDL is sufficiently similar to other programming languages that it should be straight-forward to translate the provided functions, for example, into Fortran or C
  - All of the ASCII-formatted EDR data tables can be imported into Excel, provided your machine has ample resources (CPU & memory) to do so. Spreadsheet programs can be used to explore the data set.

Prerequisites

- IDL 8.1 or a subsequent version is installed on your computer
- You are familiar with IDL
- For IDL neophytes, there are various tutorials available on the internet. For example, David Fanning's Coyote's Guide to IDL Programming (http://www.idlcoyote.com) is an excellent resource for beginners.

The IDL functions in Extras

- There are two primary functions:
  - grd_read_l1a_soh: reads GRaND state-of-health (housekeeping) data from an EDR directory
  - grd_read_l1a_science: reads GRaND science data records from an EDR directory
- There are two supporting functions:
  - get_pds_values: extracts a parameter from a PDS label given the keyword
  - make_fast_histograms: constructs histograms from fast neutron event data

Let’s get started

- I’m interested in viewing data from GRaND Low Altitude Mapping Orbit (LAMO)
- I’m especially interested in periods of time when the instrument was fully configured for science data acquisition
- I’d like to plot time-series counting data and histograms during these times

How to find data

- The GRaND PDS data are organized by mission phase within the /DATA directory
- Where can I find the dates for different mission phases?
  - Under "CATALOG," search for "Mission Phases" in the *dawnmission.cat* file.
  - You'll find a table of start and end times for each of the mission phases. For example, "Vesta Science LAMO (VSL)" spans 12-Dec-2011 through 1-May 2012

GRaND data directories

- For each mission phase, you'll find directories containing GRaND EDR
- The name of the directory gives the range of spacecraft event times (SCET). For example, the directory
  - GRD-L1A-111215-111222_YYMMDD
- contains data acquired between 15-Dec-2011 and 22-Dec-2011. The first two date codes YYMMDD are the begin and end dates.
- The third is the creation date, which I've indicated generically by YYMMDD.

What was the instrument doing?

- Open the directory GRD-L1A-111215-111222_YYMMDD
- Look inside the subdirectory LEVEL1A_AUX
- You'll find a file called GRD-L1A-111215-111222_YYMMDD-STA.tab. This is the state file.
- The file lists the instrument configuration (aka "state"), which includes commandable parameters such as high voltage settings and accumulation times.
- Every time the instrument state changes, a new line with the new parameters is added to the state file
- Let's look at the contents of this file

What's in the state file?

2011-12-15T00:00:36 0 -999 377179303 70 35 1 1058.82 1 1000.00 1 1029.41 1 1058.82 1 735.29 1 1000.00 1 1 1 1 …

MODE

TELSOH

High voltage setting (V) followed by 1 if "ON"

TELREADOUT

- The file has just one line, which I've truncated. For a detailed description, see the accompanying format file and documentation (e.g. the SIS).
- That there is only one line indicates that the instrument settings did not change for the records included in this directory
- Furthermore, I can see key parameters at a glance:
  - The accumulation time for science data records (TELREADOUT) was 70s
  - The interval for recording housekeeping data (TELSOH) was 35s
  - The instrument mode was normal (MODE=1)
  - All the power supplies were enabled and the high voltages were at nominal settings

What else can I find out about GRaND?

- The housekeeping (SOH) data are broken up into two files, which can be found within the LEVEL1A_AUX directory, for example
  - GRD-L1A-111215-111222_YYMMDD-RDG.tab
  - GRD-L1A-111215-111222_YYMMDD-SOH-SCL.tab
- The RDG file provides instrument "readings" (temperatures, voltages) every 35s. Each sample (line of the file) is tagged with the SCET/UTC time and the spacecraft clock (SCLK) string.
- The SOH-SCL contains counting (scaler) data recorded in the housekeeping telemetry.
- Both of these files can be examined using Excel.

A word about SCLK

- Before going much further, you should know that SCLK is recorded as a four byte word in GRaND telemetry, and
- the example IDL functions (and the EDR data tables) represent SCLK as a long integer
- However, SCLK is properly represented as a string. For example, in the first line of the RDR file, SCLK = 377179303.
- For use in NAIF/SPICE (navigation) routines, this integer must be converted to a string. The full SCLK string looks like
  - '001/0377179303.000'
- The first four characters '001/' give the master frame and the last four characters '.000' give the seconds fraction (000 – 255); however, since GRaND does not record the seconds fraction, the selection of '.000' is somewhat arbitrary.
- It turns out that passing a SCLK string of '377179303' to NAIF/SPICE routines gives the same result as '001/0377179303.000'
- The conversion of the unsigned integer to string is easy in IDL: sclkstr=strtrim(string(SCLK),2)

Another fact about SCLK

- You should also know that the SCLK values reported in the science telemetry and the housekeeping telemetry do not have the same accuracy
- SCLK values recorded in the SOH telemetry (e.g. found in RDR, SOH-SCL, and the state file) give the most recent value provided by the spacecraft via a timestamp command. These generally occur every 60s. So, the time associated with an SOH packet could be "off" by as much as 60s.
- In contrast, the SCLK value provided in the science telemetry accurately records the end of each science accumulation interval within about 1s – which GRaND accomplishes by internally counting seconds since the last SCLK delivered via the timestamp command.
- This information may seem a bit obscure; however, these facts are important when comparing housekeeping and science data.

Quicklook reports

- Another way of getting a graphical overview of a data directory is to review the associated quicklook report
- You can find these in the /BROWSE directory of the archive
- For example, the quicklook report for GRD-L1A-111215-111222_YYMMDD is a Portable Document File (PDF) named
  - GRD-L1A-111215-111222_REPORT.PDF

What's in the Quicklook report?

- A 1-page summary, including record counts and instrument state data
- Followed by strip charts of scaler data and selected histograms

NASA Dawn Mission

GRaND@Vesta Activity Report

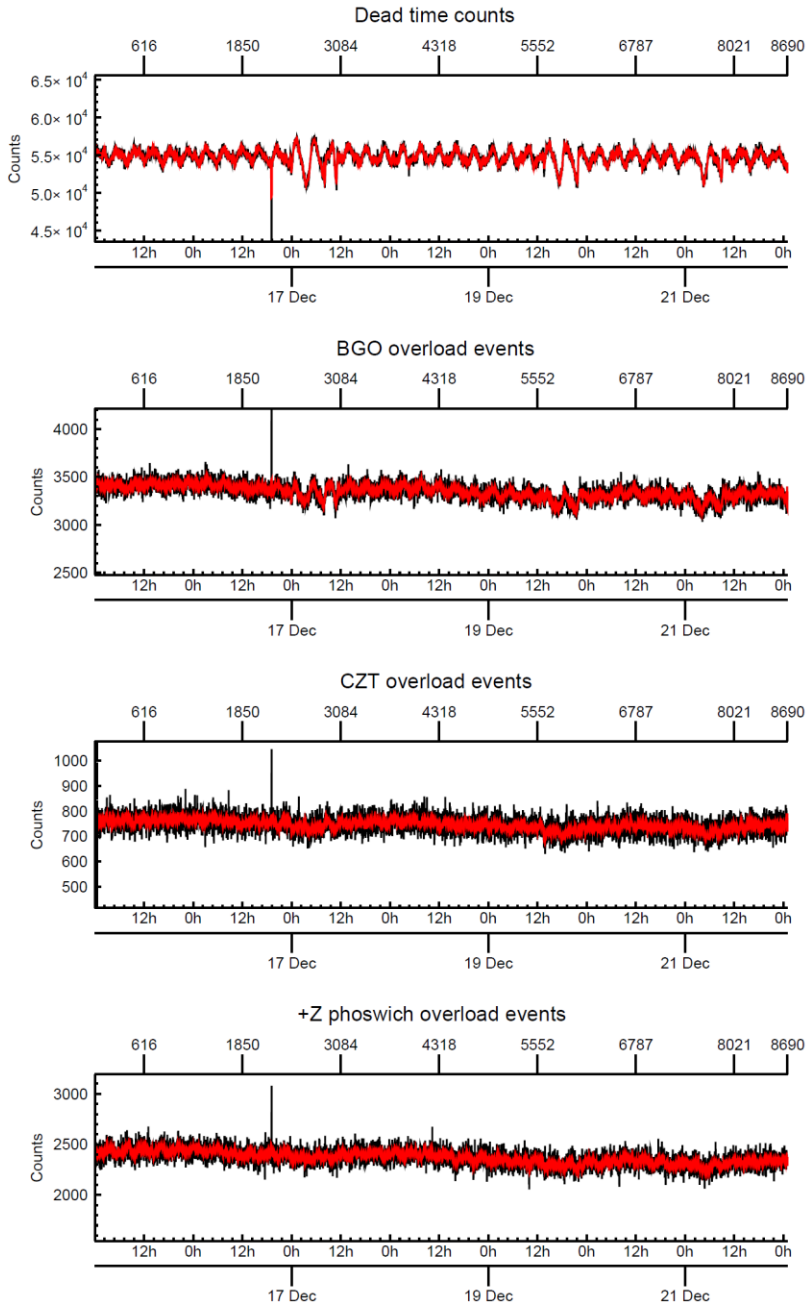
2012-02-29

- Activity: DA015 Data directory: cdr\VSL\11349\
- Number of science records: 8691
  23 missing science packets with 1 gaps.
  Start/End: 2011-12-15T00:00:07 to 2011-12-22T00:59:37
- Number of SOH records: 17383
  0 missing SOH packets with 0 gaps.
  Start/End: 2011-12-15T00:00:36 to 2011-12-22T00:59:36
- Consumables:
  136.5 power cycles.
  340.7 days of operation.
- Comment: Quick report.
- Instrument settings for last state (STATE_INDEX 0):

TELREADOUT	70	NEMG_TOT_EVTS	3876
TELSOH	35	NEMG_CZT_EVTS	3376
MODE	1	NEMN_TOT_EVTS	2800
HVPS1	1	L_BGO_CW	1
HVPS1_SET	180	H_BGO_CW	64
HVPS2	1	L_BGO_ROI	1
HVPS2_SET	170	H_BGO_ROI	64
HVPS3	1	L_BLP_MY_CW	1
HVPS3_SET	175	H_BLP_MY_CW	64
HVPS4	1	L_BLP_MY_ROI	1
HVPS4_SET	180	H_BLP_MY_ROI	64
HVPS5	1	L_BLP_PY_CW	1
HVPS5_SET	125	H_BLP_PY_CW	64
HVPS6	1	L_BLP_PY_ROI	1
HVPS6_SET	170	H_BLP_PY_ROI	64
PM5_LVPS	1	L_BLP_MZ_CW	1
P12_LVPS	1	H_BLP_MZ_CW	64
HEATERS_ENABLE	0	L_BLP_MZ_ROI	1
CZT_PM5_LVPS	1	H_BLP_MZ_ROI	64
CZT_ENABLES		L_BLP_PZ_CW	1
	0010001000000010	H_BLP_PZ_CW	64
		L_BLP_PZ_ROI	1
		H_BLP_PZ_ROI	64

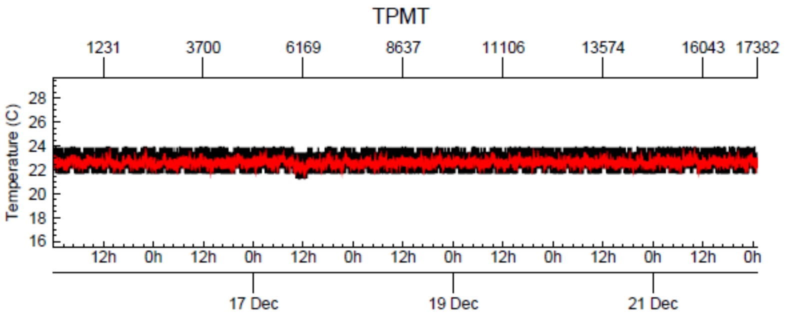
Scaler data

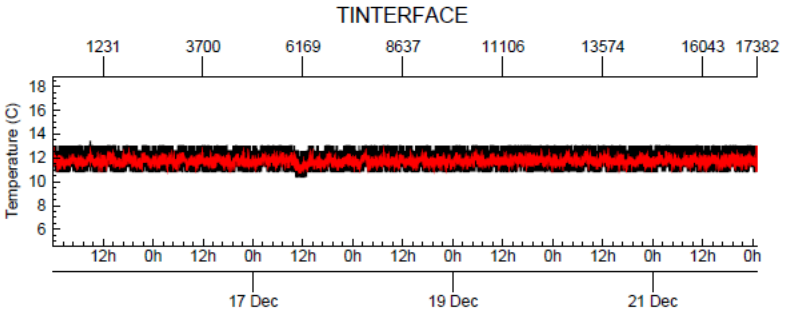
- Some scaler data from the Quicklook report
- We'll discuss the scaler data in more detail later in the presentation

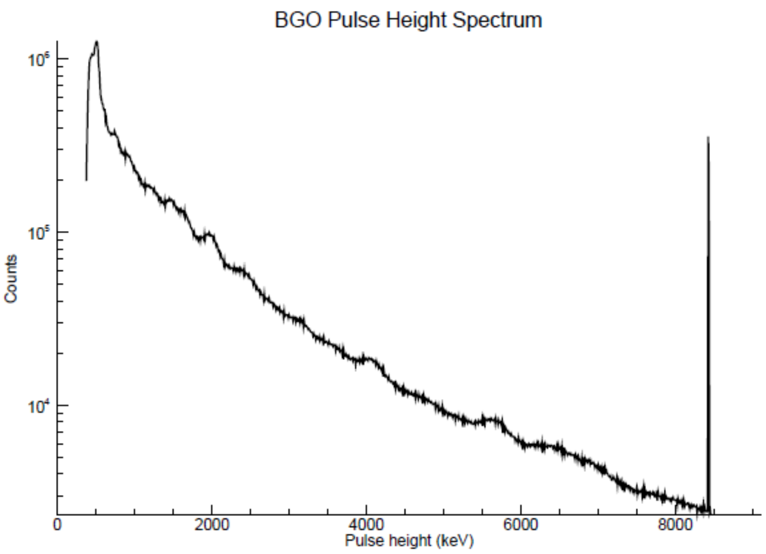


Histogram

- More scaler data and the BGO histogram are shown (right)
- The BGO histogram must undergo some correction before it can be analyzed (e.g. differential nonlinearity and gain corrections)
- These corrections are applied to produce the Reduced Data Records (RDR)







Using the IDL functions

- Now that we've covered the prerequisites, let's use the IDL functions.
- In the examples that follow, 'IDL>' refers to the IDL prompt, after which follows a relevant command.
- To explore the contents of the LAMO directory we selected, download the GRaND EDR directory (GRD-L1A-111215-111222_YYMMDD) from PDS to your computer
- Following the download, the directory might appear on your machine under
  - c:\users\myname\projects\GRaND_PDS\VSL\ (PC)
  - /home/myname/GRaND_PDS/VSL/ (Linux)
- Download and compile the IDL routines provided in Extra

Using the IDL functions

- Next, construct a variable called ‘directory’ as follows. The IDL routines will look for the data files there, e.g.
  - IDL> path='C:\users\myname\projects\GRaND_PDS\'
  - IDL> directory= path+'GRD-L1A-111215-111222_YYMMDD\'
  - IDL> print, directory
    - c:\users\myname\projects\GRaND_PDS\GRD-L1A-111215-111222_YYMMDD\
- Now, you can read in the science and housekeeping data using the IDL functions:
  - IDL> sci=grd_read_l1A_science(directory)
  - IDL> soh=grd_read_l1a_soh(directory)

Viewing the results

- Each of the function calls returns an array of structures, for example,
  - IDL> help, sci
  - SCI STRUCT = -> <Anonymous> Array[8691]
  - (SCI is an array of anonymous structures 8691 in length)
  - Each array elements corresponds to a science data record. So, there are 8691 science data records.
- The contents of the structure can be determined as follows:
  - IDL> help, sci, /str
  - ** Structure <ad10e70>, 13 tags, length=47224, data length=47220, refs=1:
  - SCET_UTC STRING '2011-12-15T00:00:07
  - SCLK LONG 377179274
  - PSC LONG 12485
  - SCALER ULONG Array[23]
  - BGO_HIST ULONG Array[1024]
  - CZT10 FLOAT Array[512, 16]
  - PHOS_PZ ULONG Array[256]
  - PHOS_MZ ULONG Array[256]
  - BLP2 LONG Array[64, 4]
  - BGO2 LONG Array[64, 4]
  - FAST LONG Array[64, 4]
  - SECOND LONG Array[64, 4]
  - TTSP LONG Array[256, 4]

Structure contents

- The structure for each array element contains the following info:
  - SCET_UTC STRING '2011-12-15T00:00:07 — UTC spacecraft event time (SCET) for the end of the accumulation interval
  - SCLK LONG 377179274 — SCLK (s)
  - PSC LONG 12485 — Packet sequence counter
  - SCALER ULONG Array[23] — Data for 23 scalers (counts)
  - BGO_HIST ULONG Array[1024] — BGO histogram
  - CZT10 FLOAT Array[512, 16] — CZT CAT10 single interaction histograms
  - PHOS_PZ ULONG Array[256] — +Z phoswich single interaction histogram
  - PHOS_MZ ULONG Array[256] — -Z phoswich single interaction histogram
  - BLP2 LONG Array[64, 4] — BLP histograms for the BLP-BGO coincidence
  - BGO2 LONG Array[64, 4] — BGO histograms for the BLP-BGO coincidence
  - FAST LONG Array[64, 4] — Fast neutron first interaction histograms
  - SECOND LONG Array[64, 4] — Fast neutron second interaction histograms
  - TTSP LONG Array[256, 4] — Fast neutron time-to-second-pulse histograms

Plotting the BGO histogram

- Plot the BGO histogram summed over all of the records

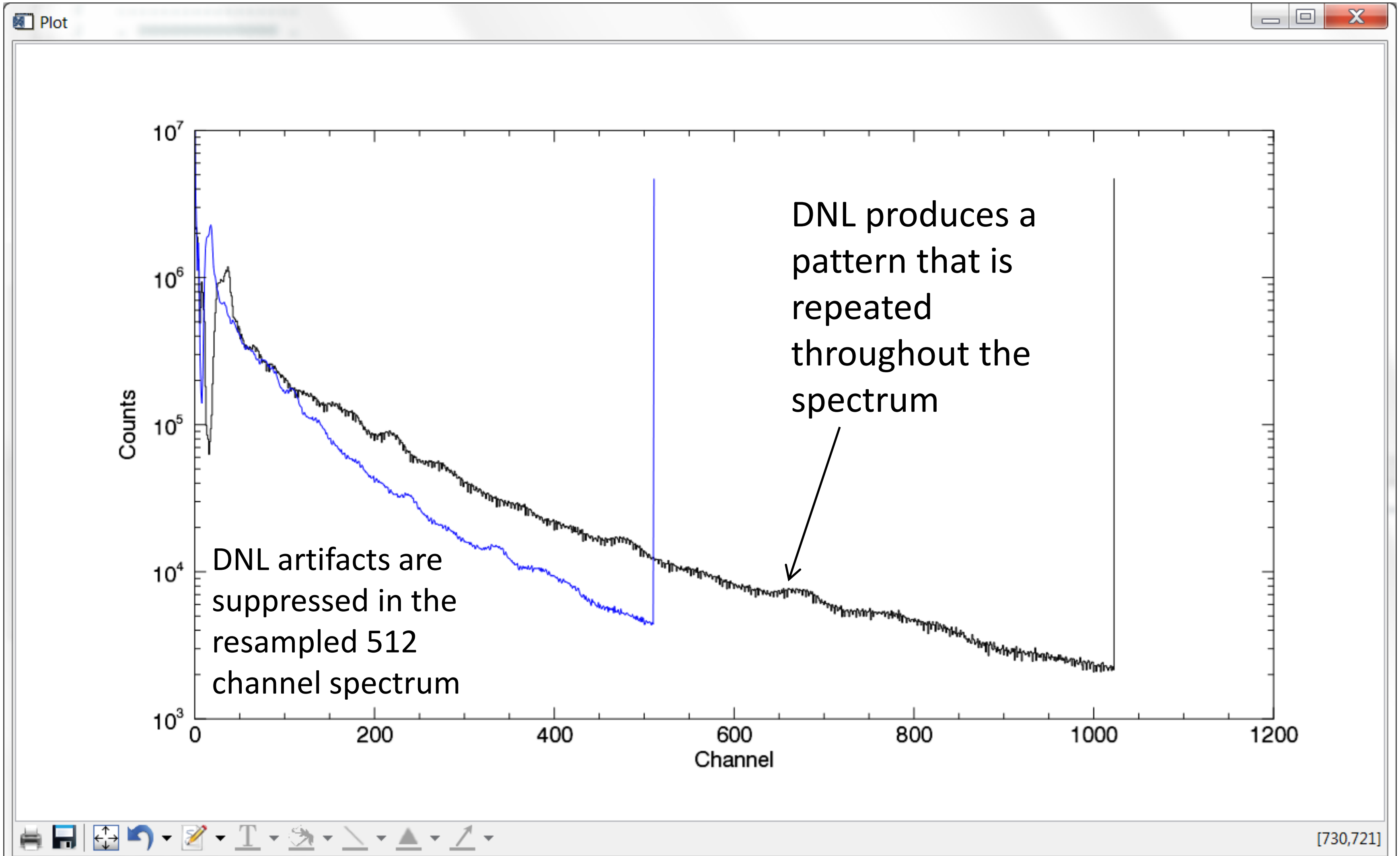
- IDL> p=plot(total(sci.bgo_hist,2),/ylog,/histogram, yrange=[1000.,1e7])

- Reduce differential nonlinearity artifacts by rebinning the data from 1024 channels to 512

- IDL> p=plot(rebin(total(sci.bgo_hist,2),512)*2L,/over, color='blue')
- IDL> p.xtitle='Channel' & p.ytitle='Counts‘

- Note that the rebin function uses nearest-neighbor averaging. So, the rebinned spectrum is multiplied by 2 to preserve total counts

Result

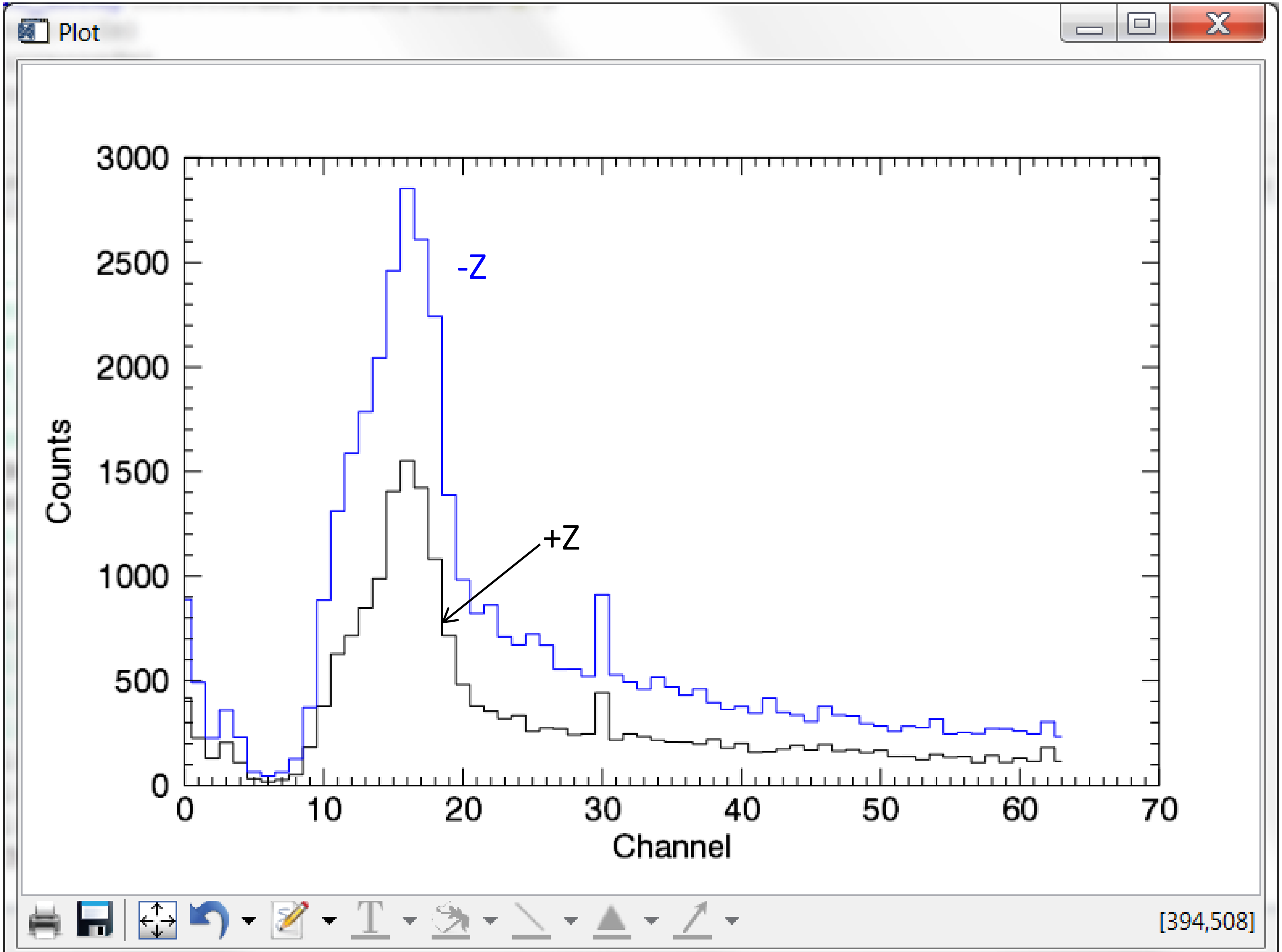


Plot
10^7
10^6
10^5
10^4
10^3
Counts
0
200
400
600
800
1000
1200
Channel
DNL produces a pattern that is repeated throughout the spectrum
DNL artifacts are suppressed in the resampled 512 channel spectrum
[730,721]

Plotting the CAT2 histograms

- Compare the +Z (PZ) and –Z (MZ) BGO CAT2 histograms summed over the first 100 science data records; we'll use IDL's plot function
- CAT2 events occur when there is a coincidence between the BGO scintillator and any of the four boron-loaded plastic (BLP) scintillators
- The code documentation indicates that the ID for the PZ scintillator is 0 and the MZ scintillator is 3
- Let's start with the PZ spectrum:
  - IDL> help, sci[0:99].bgo2[*,0]   ; these are the first 100
                                  ; spectra for +Z
  - <Expression>   LONG   = Array[64, 100]
  - IDL> p=plot(total(sci[0:99].bgo2[*,0],2),/histogram)
  - IDL> help, sci[0:99].bgo2[*,3]   ; these are the first 100
                                  ; spectra for -Z
  - <Expression>   LONG   = Array[64, 100]
  - IDL> p=plot(total(sci[0:99].bgo2[*,3],2),/histogram, color='blue',/over)
  - IDL> p.xtitle= 'Channel' & p.ytitle= 'Counts'

Result

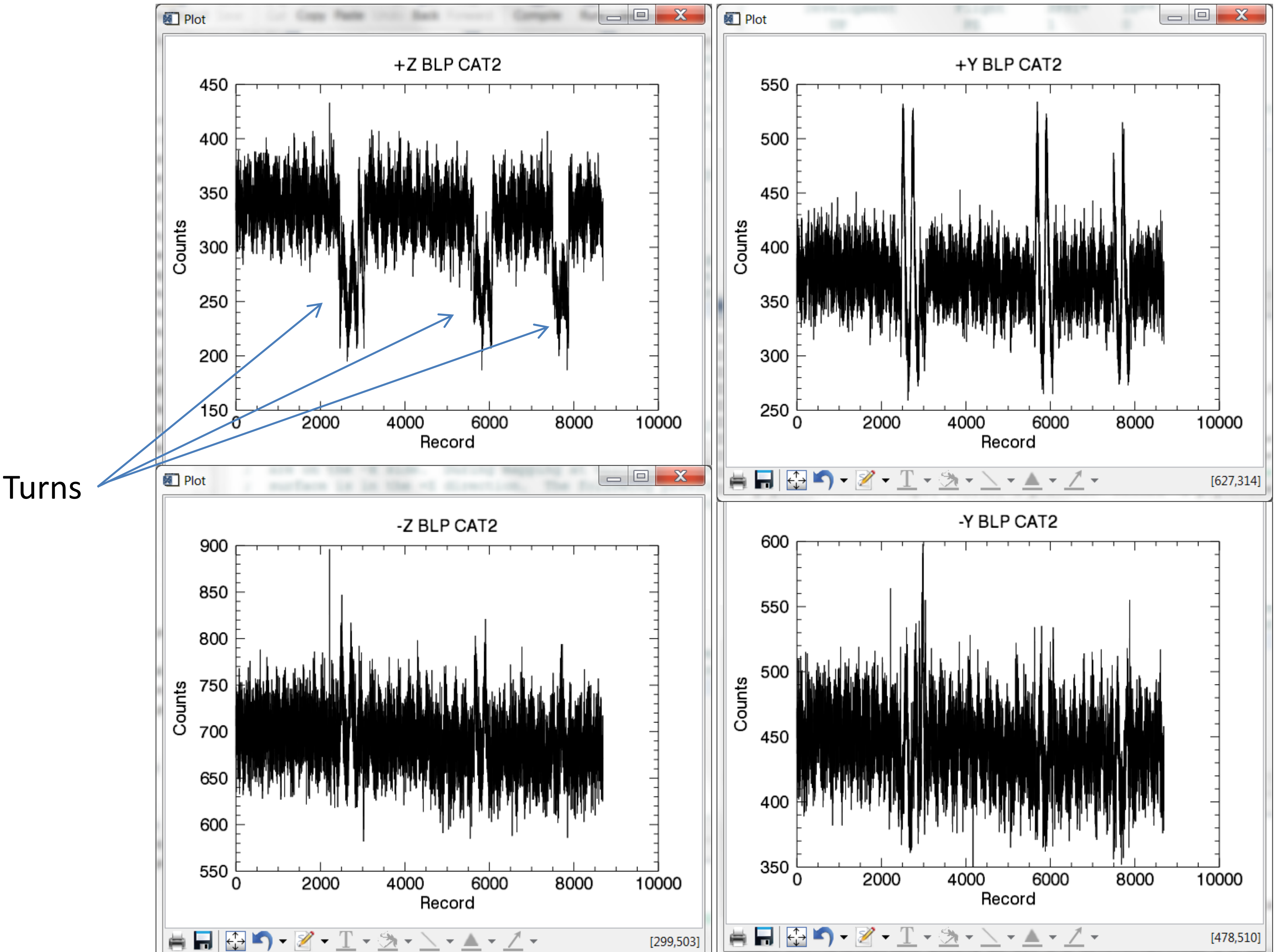


Plot
3000
2500
2000
1500
1000
500
0
Counts
-Z
+Z
0
10
20
30
40
50
60
70
Channel
[394,508]

Plotting time series data

- During LAMO, most of the time was spent nadir pointing, with occasional turns for telemetry/orbit-maintenance
- Look for evidence of turns in the BLP (CAT2) time-series counting data (increase or decrease in counts)
- IDL> p=plot(total(sci.blp2[*,0],1)) & p.xtitle='Record' & p.ytitle='Counts' & p.title='+Z BLP CAT2‘
- IDL> p=plot(total(sci.blp2[*,3],1)) & p.xtitle='Record' & p.ytitle='Counts' & p.title=‘-Z BLP CAT2‘
- IDL> p=plot(total(sci.blp2[*,1],1)) & p.xtitle='Record' & p.ytitle='Counts' & p.title=‘-Y BLP CAT2‘
- IDL> p=plot(total(sci.blp2[*,2],1)) & p.xtitle='Record' & p.ytitle='Counts' & p.title='+Y BLP CAT2‘

Results



Turns
Plot
+Z BLP CAT2
Counts
Record
+Y BLP CAT2
Counts
Record
-Z BLP CAT2
Counts
Record
-Y BLP CAT2
Counts
Record
[627,314]
[299,503]
[478,510]

Qualitative Analysis

- +Z sensor counts/science-data-record decrease when Dawn turns from nadir (the +Z sensor is usually tipped towards Vesta and is relatively well-shielded from the spacecraft by the other sensors and intervening materials)
- "Inboard" sensors (-Y, -Z) have large spacecraft background contributions that lessen their response to turns
- The signature for turns is more pronounced for the "outboard" (+Z, +Y) sensors

Scaler data

- In addition to histograms and events, GRaND records data for 23 scalers
- What is a scaler? It's just a pulse or event counter.
- The scalers are defined in the dataset catalog, the SIS, and the Data Processing Document (Table 2; the table is repeated on the following slide.
- The scaler data are recorded in the housekeeping and science telemetry
- In this presentation, we'll use scaler 0 (the deadtime counter) to determine live time for each science data record

Scaler definition table

Index	Description
0	Dead time counts
1	BGO overload events
2	CZT overload events
3	+Z phoswich overload events
4	-Y BLP overload events
5	+Y BLP overload events
6	-Z phoswich overload events
7	+Z phoswich CAT4 events
8	-Y BLP CAT4 events
9	+Y BLP CAT4 events
10	-Z phoswich CAT4 events
11	Early second interaction events
12	Multiple-crystal CZT events
12	Multiple-crystal CZT events
13	Valid CZT events (CAT10)
14	Coincidence BGO and CZT events (CAT7)
15	Coincidence of three or more sensor elements
16	Total events processed by GRaND
17	Number of single CZT events (CAT10) in the gamma ray event buffer
18	Number of BGO-CZT coincidence events (CAT7) in the gamma ray event buffer
19	Number of events (CAT4) in the neutron event buffer
20	Total number of events allowed in the gamma ray event buffer
21	Number of single CZT events (CAT10) allowed in the gamma ray event buffer
22	Number of events allowed in the neutron event buffer

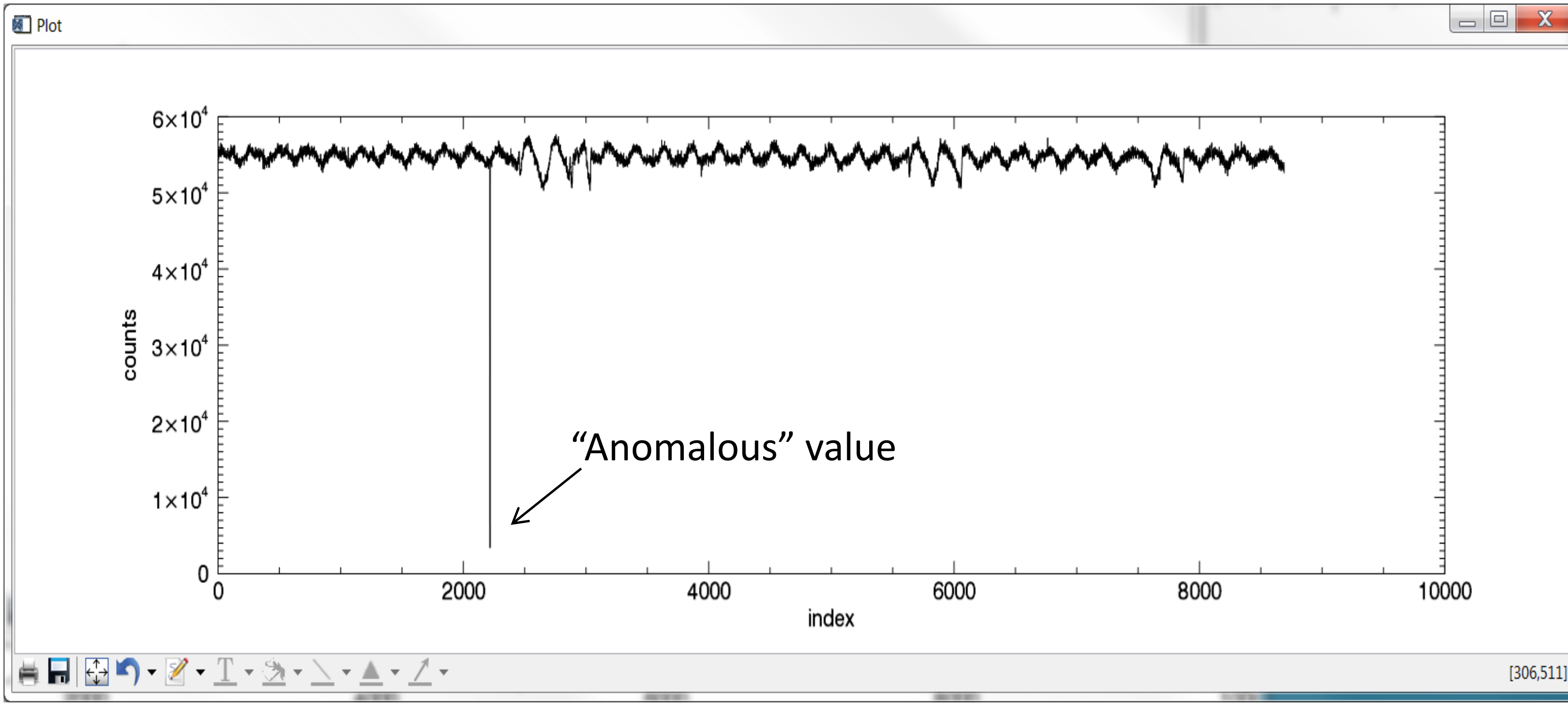
Dead Time Scaler (Index = 0)

- The "dead time" is the amount of time GRaND's counting circuits are busy processing events
- Corrected counting rates are determined by normalizing counts to "live time," which is TELREADOUT minus dead time
- The dead time for each event is recorded by a pulse counter (the dead time scaler). Each count indicates an elapsed time of 204.8 microseconds.
- The dead time scaler is incremented in proportion to the duration of each event
- The scaler is reset at the end of each science accumulation interval
- Thus, the total dead time is the value of the dead time scaler multiplied by 204.8×10^{-6} s.
- See Prettyman et al. (2011) *Space Science Reviews* 163:371–459 for a detailed explanation of this scaler

Plotting the dead time scaler

- Plot the dead time scaler
- IDL> p=plot(sci.scaler[0], xtitle='index', ytitle='counts')
- Notice that most of the records have values of about $5.5x10^4$; however, there is an anomalous low value to consider
- Get the index of the anomalous value
- IDL> index=where(sci.scaler[0] lt 1e4, count)
- IDL> print, index, sci[index].scaler[0]
- The result of the print indicates that there is one anomalous value (3401) with an index of 2217

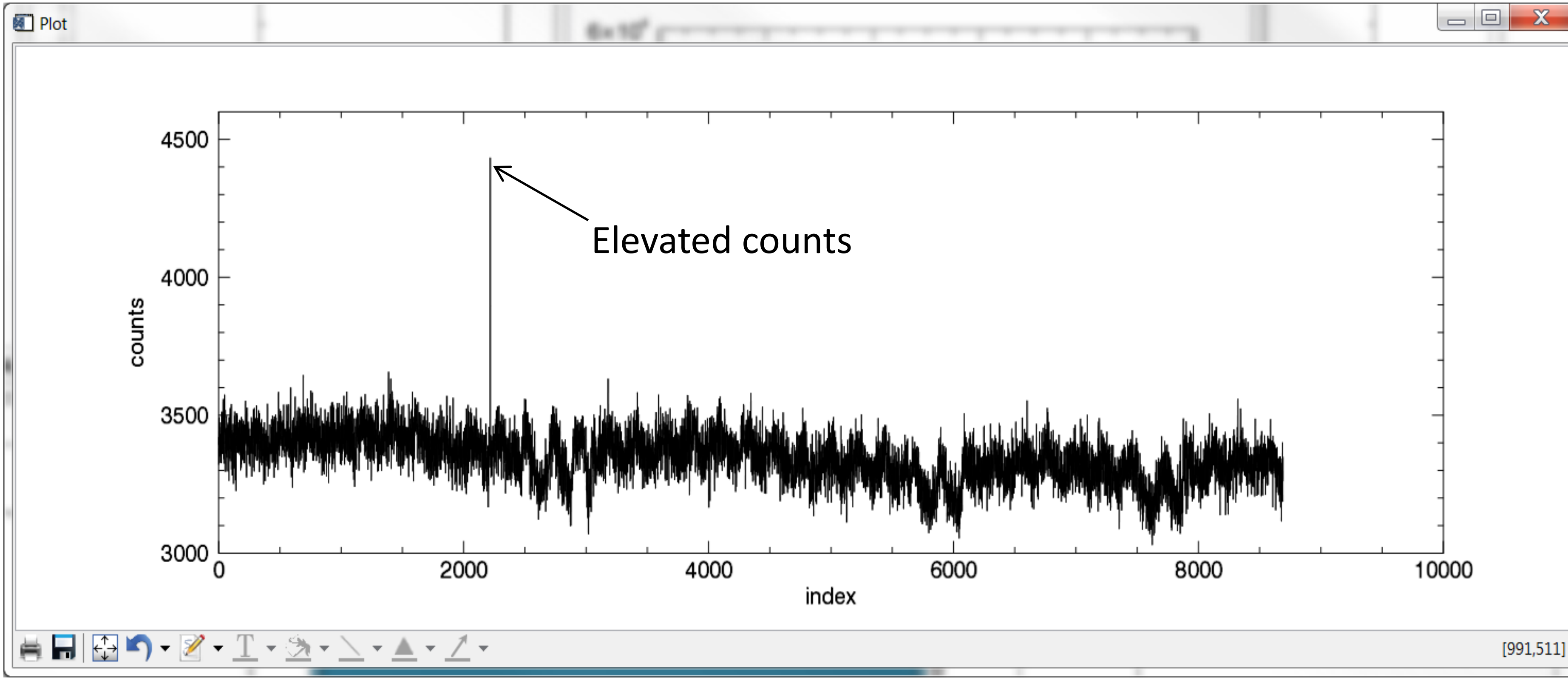
Dead time counter plot



Analysis

- The 16-bit dead time scaler may "roll over" one or more times during a science accumulation interval, depending on TELREADOUT and counting rate
- The low dead time counts for record 2217 indicates that the dead time counts may have exceeded 2^{16}-1 during the accumulation interval for the record in question
- This would be supported if counting rates were elevated for this record. To find out, plot the BGO overload scaler (scaler index of 1)
- IDL> p=plot(sci.scaler[1], xtitle='index', ytitle='counts')

BGO overload scaler plot

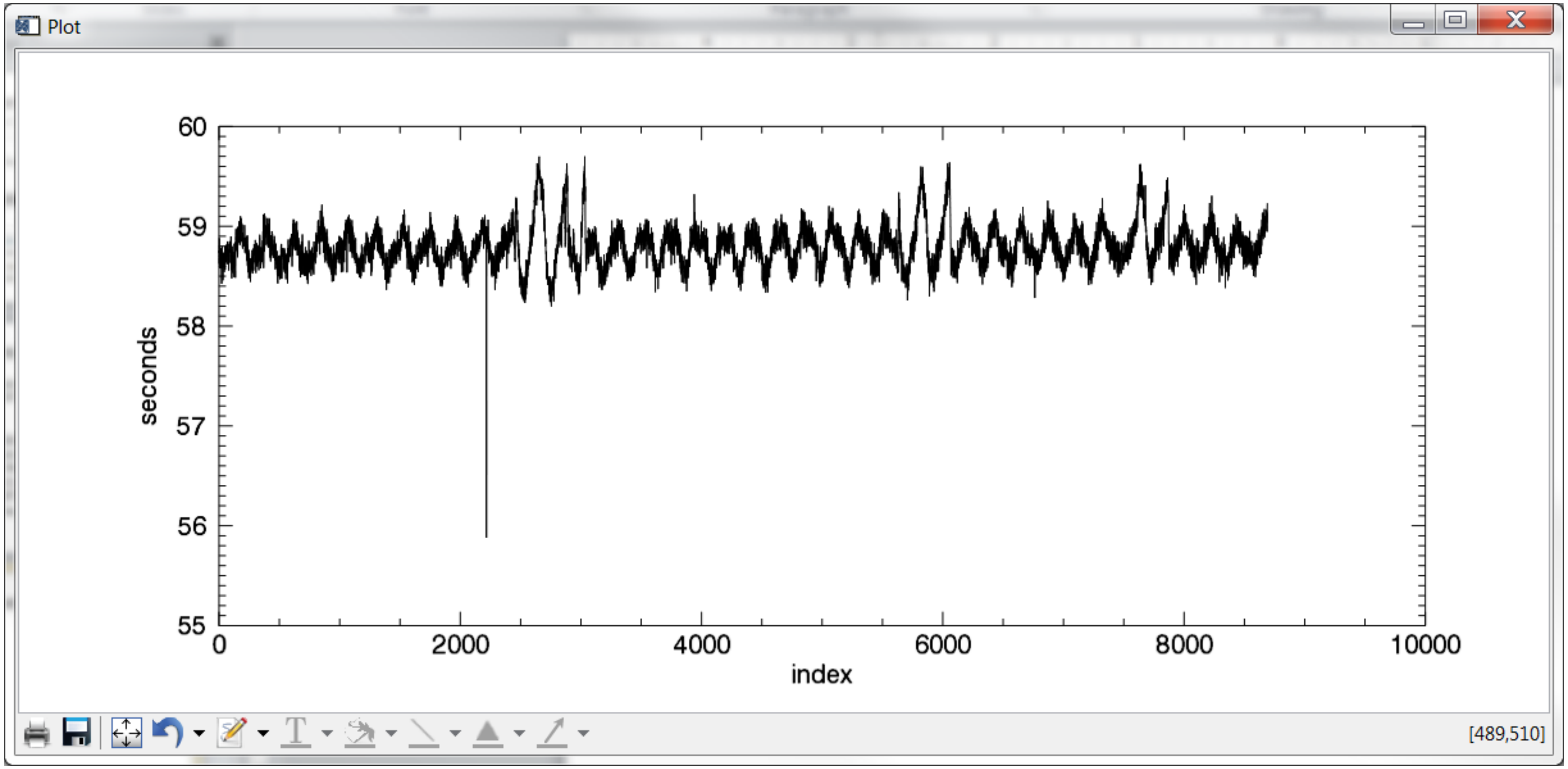


- IDL> index=where(sci.scaler[1] gt 4000, count)
- IDL> print, index
- The result of the print was 2217, which indicates that elevated BGO overload counts are associated with the anomalously low dead time count
- Increased counts for record 2217 can be observed for other scalers and time series of histogram sums

Determining Live Time

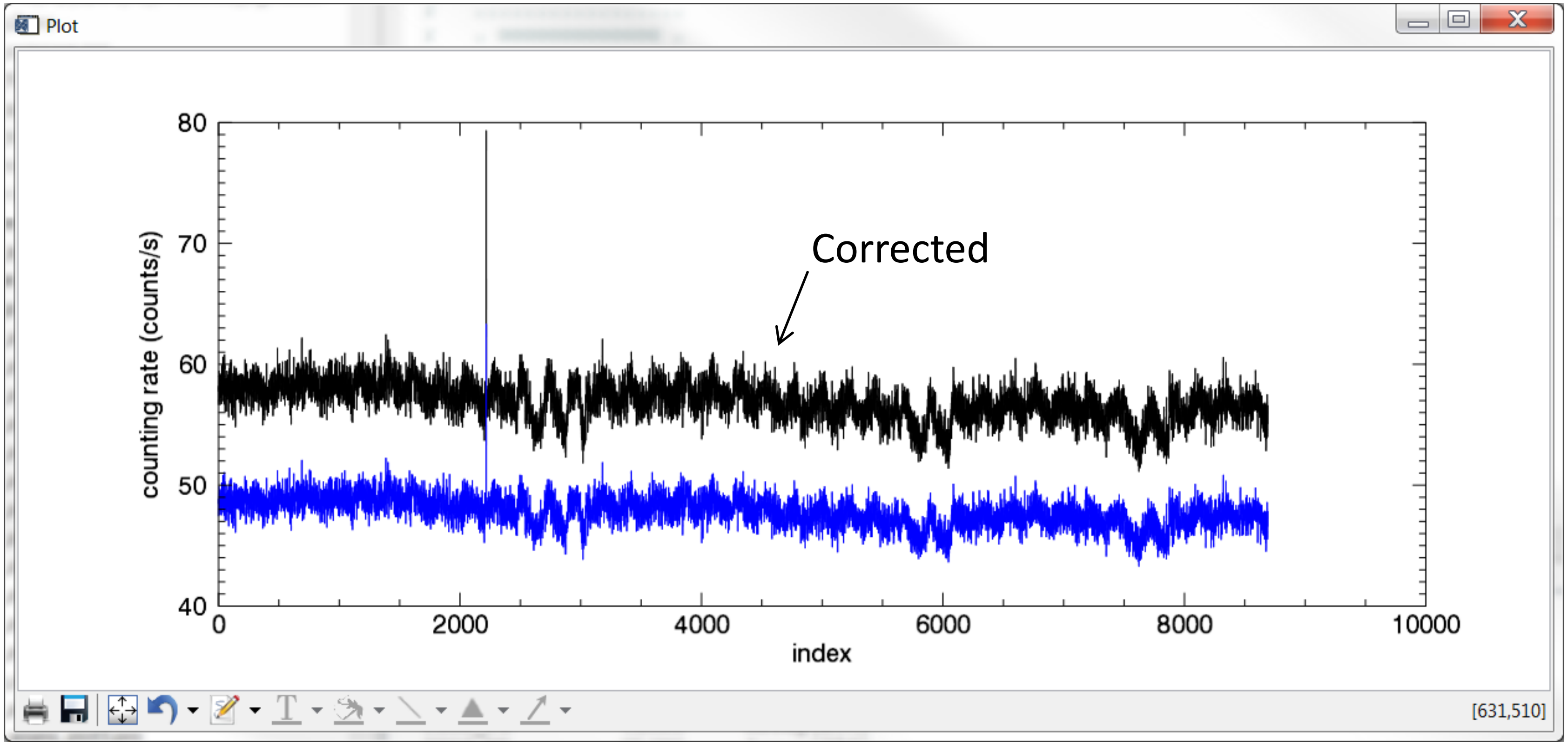
- Assuming the dead time counter usually does not roll over for TELREADOUT=70s in LAMO, live time can be calculated as follows:
- TELREADOUT=70.
- IDL> live_time = TELREADOUT - sci.scaler[0]*204.8e-6
- Assuming the dead time counter rolled over once for record 2217, then the following adjustment is needed:
- IDL> live_time[2217] = TELREADOUT - (2L^16 + sci[2217].scaler[0])*204.8E-6
- IDL> p=plot(live_time, xtitle='index', ytitle='counts')

Live Time Plot



- Note that when GRaND is nadir-pointed, the live time peak-to-peak variation is relatively small (about 1%)
- After correction for roll-over, the anomalous record is associated with low live time (high dead time), consistent with elevated event rates during this time period (unknown cause)

Live time correction

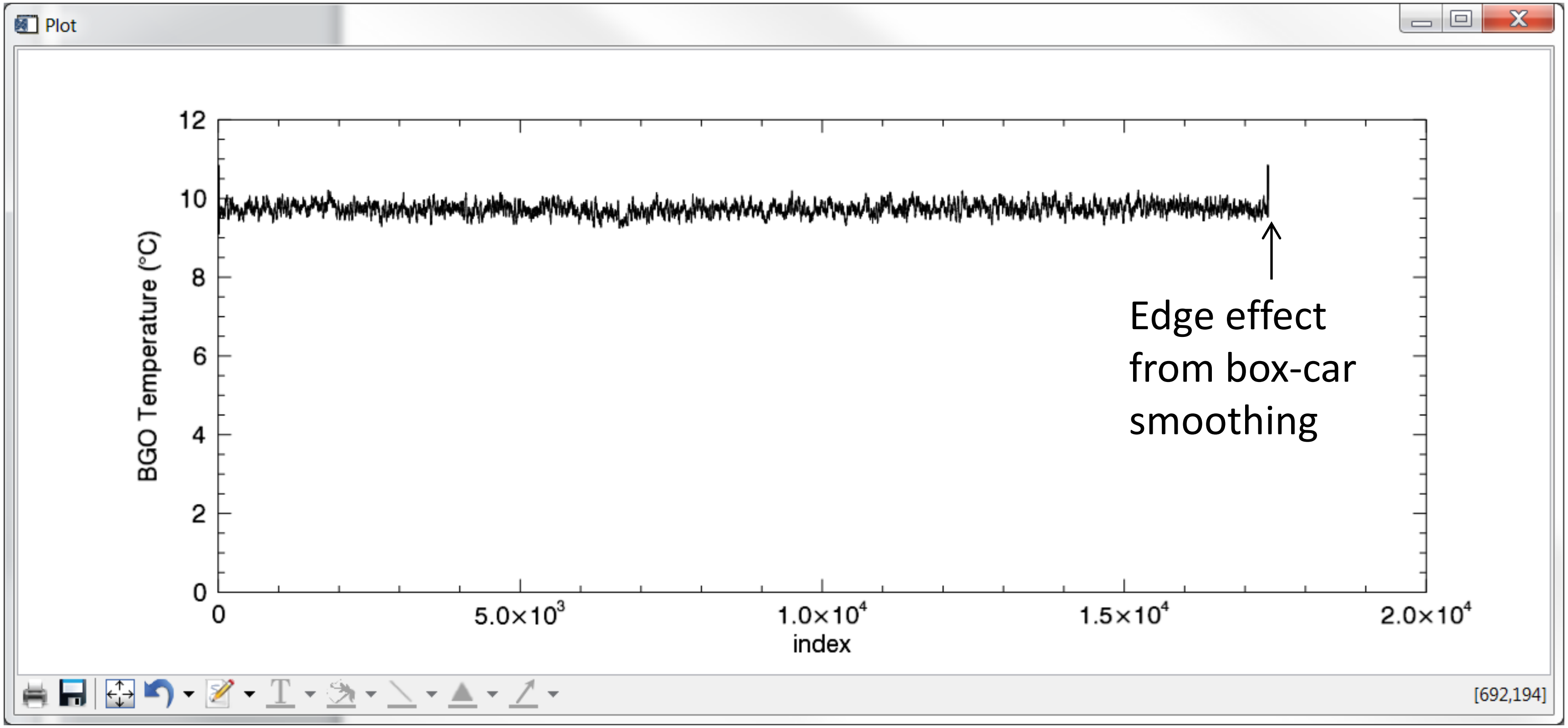


- Compare live-time corrected counting rates with gross counting rates for the BGO overload scaler
- IDL> p=plot(sci.scaler[1]/live_time, xtitle='index', ytitle='counting rate (counts/s)')
- IDL> p=plot(sci.scaler[1]/TELREADOUT, /over, color='blue')

What’s in the SOH structure?

- IDL> help, soh
  - SOH STRUCT = -> <Anonymous> Array[17383]
- IDL> help, soh, /str
  - ** Structure <2cb0e4f0>, 11 tags, length=144, data length=144, refs=1:
  - SCET_UTC STRING '2011-12-15T00:00:36‘
  - SCLK LONG 377179303
  - PSC LONG 3528
  - SCALER ULONG Array[23]
  - T_BGO FLOAT 9.97000
  - T_CZT1 FLOAT 19.9800
  - T_CZT2 FLOAT 20.4200
  - T_CZT3 FLOAT 21.2900
  - T_CZT FLOAT 20.5633
  - T_CZT4 FLOAT 23.9000
  - T_INTERFACE FLOAT 13.0100
- The SOH structure contains data from the –RDG and –SOH-SCL tables found in the AUX subdirectory. Since the sampling time TELSOH=35s was selected to be smaller than the science accumulation time (TELREADOUT=70s) there are more (2x) SOH entries
- The -SOH-SCL structure contains data for all 23 scalers sampled on a fine time scale, which is a useful diagnostic
- Only the temperatures from the –RDG table are included in the structure

Temperature plot



- IDL> p=plot(smooth(soh.t_bgo,20), yrange=[0,12], xtitle='index', ytitle='BGO Temperature ($\deg$C)')
- The BGO temperature sensor gives readings that are representative of GRaND's scintillators
- As can be seen, the internal temperature is stable over long periods of time (about a week is shown here)

Summary

- The IDL functions read the EDR science and housekeeping data into structures that can be manipulated in IDL
- The routines are intended for example only and do not process the data beyond Level 1A
- Additional information needed to reduce and analyze the data can be found in the documentation accompanying the archive and references that follow

Bibliography

- Prettyman et al. (2011), Dawn's Gamma Ray and Neutron Detector. *Space Science Reviews* **163** 371-459, http://dx.doi.org/10.1007/s11214-011-9862-0

- Prettyman et al. (2012), Elemental mapping by Dawn reveals exogenic H in Vesta's regolith. *Science* **388** 242-246, http://dx.doi.org/10.1126/science.1225354