CORONAL DIAGNOSTIC SPECTROMETER
SOHO

CDS SOFTWARE NOTE No. 34

Version 4 14 November 2003

Applying Calibration to VDS Data

William T. Thompson
NASA Goddard Space Flight Center
Code 682.3
Greenbelt, MD 20771

William.T.Thompson.1@gsfc.nasa.gov

1 Introduction

The calibration measurements of the VDS detector are described in CDS software note #13. Here we describe the software which applies that calibration information to the data taken by the CDS instrument.

2 Reading the data

The VDS calibration software is designed to work with the data structure used by the CDS quicklook software (software note #20). The data is input in that format, and output in the same format with the appropriate calibrations applied.

To read a data file into the appropriate quicklook format, use the routine READCDSFITS. For example,

```
IDL> DATA = READCDSFITS('s100r0.fits')
```

This will produce a data structure with an organization like the following:

```
IDL> HELP, /STRUCTURE, DATA
** Structure <4010f808>, 46 tags, length=1444856, refs=1:
   QL_ID           DOUBLE       8.2016871e+08
   QL_NO           INT                 1
   HEADER          STRUCT    -> HDR Array(1)
   DETDATA         INT       Array(300, 20, 120)
   DETDESC         STRUCT    -> NIS3 Array(3)
   BACKGROUND      STRUCT    -> <Anonymous> Array(1)
   DEL_TIMEDATA    DOUBLE    Array(20)
   DEL_TIMEDESC    STRUCT    -> AUX1 Array(1)
   INS_XDATA       FLOAT     Array(20)
   INS_XDESC       STRUCT    -> AUX1 Array(1)
          ⋮
```

The tag DETDESC is itself a structure which describes the properties of the data windows, and has an organization along the following lines:

```
IDL> HELP, /STRUCTURE, DATA.DETDESC
** Structure NIS3, 20 tags, length=232:
   IXSTART         INT       Array(3)
   IXSTOP          INT       Array(3)
   LABEL           STRING    'WW_321_29'
   UNITS           STRING    'ADC'
   MAX             FLOAT           3659.00
   MIN             FLOAT           490.000
   MISSING         INT             -1
```

```
AXES            STRING    Array(3)
ORIGIN          DOUBLE    Array(3)
SPACING         DOUBLE    Array(3)
ROTATION        DOUBLE    Array(3)
DETX            LONG               168
BINX            LONG                 1
DETY            LONG               522
BINY            LONG                 1
WAVELENGTH      DOUBLE           321.25400
WAVEMIN         DOUBLE           317.63300
WAVEMAX         DOUBLE           324.87600
WAVEBAND        LONG                 1
GR_ORDER        LONG                 1
```

Raw data from the VDS detector is always in units of "ADC" which stands for Analog to Digital Conversion units. Unlike the GIS detectors, the VDS is not a photon counting device, so describing the data as "counts" is inappropriate.

As the various calibration steps are applied to the data, the value of units will change accordingly.

3 Removing the CCD readout bias

After the data has been read in, the next step in the VDS calibration is to remove the CCD readout bias. This is done through the routine VDS_DEBIAS. After applying VDS_DEBIAS, the units of the data are "DEBIASED-ADC".

There are two ways in which the bias can be removed.

3.1 Automatic bias removal

If the raster was designed to include the VDS background windows, then automatic bias removal can be used. The command to do this is

```
VDS_DEBIAS, DATA
```

The BACKGROUND tag in the data structure is used to determine the biases in each of the four CCD quadrants.

Automatic bias removal is recommended.

3.2 Manual bias removal

One can bypass the automatic bias removal feature of VDS_DEBIAS by supplying the biases as a second, optional variable, e.g.

```
VDS_DEBIAS, DATA, BIASES
```

The BIASES parameter can take one of two forms. It either be an array of 4 elements giving the bias in the A, B, C, and D quadrants of the CCD, or it can be a two dimensional array where the first dimension is the number of exposures, and the second dimension is 4 for the four quadrants.

4 Applying the VDS calibration

After removing the CCD bias, the next step in the VDS calibration is to apply the routine VDS_CALIB. The format of this command is

```
VDS_CALIB, DATA  [, PROGVOL ]
```

where the optional parameter PROGVOL is the value of programable voltage #15 used for the observation. For example,

```
VDS_CALIB, DATA, 100
```

If PROGVOL is not passed, then it is extracted from the HEADER tag of the data structure. Thus,

```
VDS_CALIB, DATA, DATA.HEADER.VDS_PMCP
```

is equivalent to simply saying

```
VDS_CALIB, DATA
```

The latter is recommended—under normal circumstances one should *never* need to pass the PROGVOL parameter.

After applying VDS_CALIB, all the calibration parameters specific to the VDS detector, except quantum efficiency, have been applied to the data. The units of the data, as reflected by the units tag, are set to "PHOTON-EVENTS/PIXEL/SEC". The term "PHOTON-EVENTS" refers to the photons which have actually been detected by the instrument, as opposed to the photons which impinge on the detector. This is because the quantum efficiency, which varies with wavelength, has not yet been applied to the data.

5 Full calibration

The final step in calibrating the data taken with the VDS detector is to convert the photon events calculated by VDS_CALIB into absolute units. There are several factors which influence this step of the calibration:

1. The quantum efficiency of the VDS detector, as given in software note #13.

2. The wavelength calibration of the NIS spectrograph, which determines how to apply the quantum efficiency parameters. This will depend on the slit used to make the observation.

3. The effective area of the CDS telecope and NIS spectrograph.

These factors are applied by an additional calibration routine, called NIS_CALIB, based on the end-to-end calibration of the entire CDS instrument performed at RAL.

There are a number of keywords that can be passed to NIS_CALIB, to control the units that the data is returned in. The default units are "photons/cm^2/sec/arcsec^2", which can be modified by the following keywords:

ERGS: If set, then the data is returned in units of ergs instead of photons.

STERADIANS: If set, then the data is returned as per steradian instead of per arcsecond^2.

ANGSTROMS If set, then the data is returned as per Ångstrom. This keyword is ignored when the data has been summed over wavelength (SUMLINE or SUMWIN compression) or when the instrument is operated in spectroheliogram mode with the 90x240 arcsecond slit.

It's important to realize that the units (without /ANGSTROM) only apply when all the data in the line profile is added together. Otherwise, the units are per spectral pixel.

5.1 Applying the calibration by hand

It is possible to apply the calibration of NIS_CALIB by hand to partially analyzed data, so long as VDS_CALIB is properly applied. There are a series of GET_ routines which retrieve the various calibration parameters:

5.1.1 Absolute calibration

The routine GET_EFFICIENCY returns the absolute efficiency of either the NIS1 or NIS2 spectrograph. This number is a combination of the following efficiencies:

1. The reflectivity of the primary mirror.
2. The reflectivity of the secondary mirror.
3. The reflectivity of the scan mirror.
4. The efficiency of the grating.
5. The quantum efficiency of the VDS detector.

For example,

```
GET_EFFICIENCY, '10-JAN-1997', 'N1', EFF1
```

returns the efficiency of the NIS1 spectrograph on 10 January 1997.

5.1.2 Wavelength dependence

The wavelength dependence of the instrument response is returned by the routine GET_WAVE_EFF. This returns the parameters of a polynomial describing the shape of the response as a function of wavelength. This polynomial will pass through unity at a wavelength corresponding to the reference point of the efficiency returned by GET_EFFICIENCY. Thus, the product of these two quantities will be the efficiency as a function of wavelength.

For example,

```
WAVE1 = PIX2WAVE('N1', INDGEN(1024))
GET_WAVE_EFF, '10-JAN-1997', 'N1', C1
EFF1 = EFF1(0) * POLY(WAVE1, C1)
```

returns the wavelength dependence over the entire NIS1 range. The "`(0)`" in the above is to force the output of GET_EFFICIENCY to a scalar, instead of a single-element array.

It's also possible to get the efficiency at one or more wavelengths directly, using GET_EFFICIENCY with the /WAVELENGTH keyword. For example, if one wanted the efficiency at 584.3 Å, one could enter

```
GET_EFFICIENCY, '10-JAN-1997', 'N2', E584, WAVELENGTH=584.3
```

One can also pass in arrays of wavelengths. The above example of the use of GET_WAVE_EFF could be simplified to

```
WAVE1 = PIX2WAVE('N1', INDGEN(1024))
GET_EFFICIENCY, '10-JAN-1997', 'N1', EFF1, WAVE=WAVE1
```

5.1.3 Used telescope aperture

The section of the telescope aperture which contributes to an observation depends both on the spectral band, and on the position of the scan mirror. As the scan mirror moves from one end to the other, the two gratings trade off on which is getting the most light. The routine GET_EFF_AREA returns the aperture area as a function of mirror position. For example,

```
GET_EFF_AREA, '10-JAN-1997', 'N1', EFF_AREA1
```

This routine always returns an array of 256 elements, even though the valid scan mirror positions only go from 68 to 188. This makes it easier to extract the right value from the array, simply by referencing it with the scan mirror position. The units are cm^2.

5.1.4 Angular factors

The angular width of a CDS pixel depends on which slit is used. For slit 4 it is 2 arcseconds, and for slit 5 it is 4 arcseconds. The pixel height is 1.68 arcseconds. For 90×240 arcsec slit 6 data, it is best to use 1.68 arcsec in both directions.

The conversion factor from arcsec^2 to steradians is simply $(\pi/(180 \times 60^2))^2 \simeq 2.35 \times 10^{-11}$.

5.1.5 Other conversions

The conversion from photons to ergs is simply $hc/\lambda \simeq 1.986 \times 10^{-8}/\lambda$, where λ is in Ångstroms.

The conversion to per Ångstrom is accomplished simply by dividing the data by the pixel width in Ångstroms.

6 Calibration updates

The calibration of the CDS instrument has been an ongoing process. Occasionally, the calibration is re-evaluated, and the databases are updated. As of this writing, the calibration has gone through three versions:

Version	Date	Explanation
0	16-Sep-1996	Lab calibration.
1	22-Nov-1996	Small correction made to lab calibration for NIS-2.
2	23-Dec-1998	Substantially revised calibration, based on in-flight comparison with sounding rocket irradiance data, and on re-analyzed preflight lab data.

When data has been already calibrated with a previous version of the calibration, it's possible to calculate correction factors by passing the DATE_USED keyword to the GET_EFFICIENCY and GET_WAVE_EFF routines.

For example, suppose one had applied NIS_CALIB on 1 December 1998 to the observation s3300r00.fits, had fitted line profiles to these data, and had saved the results. A few weeks later, the calibration was substantially revised. Rather than reanalyze all the data from scratch, it's possible to calculate correction coefficients. The correction coefficient for the He I 584 Å line could be determined with the commands

```
GET_EFFICIENCY, '26-Jun-1996', 'N2', NEW_EFF, WAVE=584.3
GET_EFFICIENCY, '26-Jun-1996', 'N2', OLD_EFF, WAVE=584.3, DATE='1-Dec-1998'
```

The results are then corrected by multiplying by OLD_EFF/NEW_EFF.

6.1 Burn-in calibration updates

Unfortunately, the calibration is not the whole story. Also important is the detector flat field correction. While the correction for burn-in by the narrow slits is well characterized, a number of revisions have been made to the corrections for the burn-in of the detector from the use of the 90×240 arc second slit #6. This correction is strongest in the vicinity of strong lines such as He I 584 Å, but can grow to be significant in other regions of the spectrum. The routine CDS_SLIT6_BURNIN can be used to track this burn-in correction, and how the correction has changed with time. For example, the command

```
PRINT, CDS_SLIT6_BURNIN('1999-10-01', 584.3)
```

would return the burn-in correction for the selected observation date and wavelength, based on the most current burn-in calibration. One can extract the same information for a previous calibration by adding the keyword ANALYSIS_DATE, e.g.

```
PRINT, CDS_SLIT6_BURNIN('1999-10-01', 584.3, ANALYSIS_DATE='1999-10-10')
```