CHARA Technical Report

No. 97 February 2014

The CLASSIC/CLIMB Data Reduction: The Software

Theo ten Brummelaar

ABSTRACT: This is one of two technical reports that describe the methods used to extract closure phase from CLIMB data and visibility amplitude from both the CLASSIC and CLIMB beam combiners. In this, the second technical report, I describe the pipeline software.

1. INTRODUCTION

There are almost as many CLASSIC data reduction programs as there are groups working with the data. This document describes three of these:

1. The first "CHARA sanctified" C/Gtk based pipeline known as `reduceir`. Reduceir has the ability to handle many types of visibility estimators, as well as separated fringe packets (SFP), and is at this time probably the most used package for CLASSIC data. This pipeline will cease being expanded, but will continue to be supported as far as the SFP stuff is concerned.

2. The second "CHARA sanctified" C based pipeline `redclassic`. This does most of what reduceir does, but does not use GTK and so is more portable. It doesn't do as many visibility estimators, nor does it handle SFP data or selfcal, but it is much more streamlined and much easier to use as a command line program in shell scripts.

3. The only "CHARA sanctified" CLIMB data reduction program called `redclimb`.

I will describe first those things all packages have in common, and I will then discuss each package separately. I will not attempt to cover all the possible switches and tricks, as that would take far too much space, but I will try and cover the most common data issues.

I will use the same example data throughout. These example data, and the data reduction software itself, can be found at the web page:

`http://www.chara.gsu.edu/˜theo/chara_reduction/index.html`

If you come across something not covered in this document, and there are many of those, feel free to talk to someone with more experience, or indeed contact me directly.

[1] Center for High Angular Resolution Astronomy, Georgia State University, Atlanta GA 30303-3083
Tel: (404) 651-2932, FAX: (404) 651-1389, Anonymous ftp: chara.gsu.edu, WWW: http://chara.gsu.edu

2. THE SOFTWARE

All of the data reduction software consists of command line programs with switches and flags in the style of most Unix like programs. Note, however, that unlike most Unix like programs in all cases these programs must not have a space between a flag and its argument. So for example in the CLIMB calibration program, you enter the calibrator diameter using the following syntax:

```
calibclimb -s0.419 OBJ CAL1
```

Note that there is no space between the `-s` flag and its argument `0.419`.

Both the CLASSIC and the CLIMB data reduction pipelines create a directory into which they place all of their output. The name of this directory is based on the name of the input FITS file. In all cases you can set the location of this directory by using the `-D` flag. The recommended method is to copy all the relevant data files into a directory in your own file system and to do all of the data reduction in that directory.

2.1. The INFO file

Many files are created in the output directory, and they will be described in the relevant sections below. By default, this directory will be created in the current directory, so please do not do this in the archives. It's better to copy the data files into a local directory.

The most important output file is called the *info-file.* This is a text file that contains keywords and data, very much like a FITS file header. The data reduction programs all use the info-file to keep track of the options used, where you are up to in the data reduction process, and the results of any calculations. If you run a data reduction step more than once it will check the info-file first and use the same options you used last time, unless you override these by using different flags when you invoke the program. This will result in several entries in the info-file with the same keyword. In all cases the value used by the software will be the last keyword in the info-file of any given type. Any text after a # character will be regarded as a comment and will be ignored. An example of part of an info-file is

```
# INFO file for data file 2011_04_05_HD_32630_climb_003.fit

# Information from FITs header:

COMBINER                CLIMB
DETECTOR                NIRO
CHARA                   53526
CAT3                    HD_32630
OBJECT                  CAL1
RA_DEC                  05 06 30.8928 +41 14 04.108
SEQUENCE_NUMBER         3
UT_DATE                 2011-04-05
UT_TIME                  03 22 04.364
TIME_FIRST_DATUM        12061401
DAY_OF_YEAR             94
UNREFAZ                 295.226281
```

```
UNREFEL                 51.347559
SCOPES                  S2 E1 W1
HOUR_ANG                 03 14 53.613
UV_POINT_12             74.371595396    -682.983674518  302.249811
UV_POINT_23             371.254859392   515.427262455   279.457228
UV_POINT_31             -445.626454788  167.556412063   209.450618
```

If you wish to be certain of avoiding any defaults in an info-file I recommend that you delete the entire output directory and begin reduction of that data file from scratch.

There are two programs that enable searching the info-files and these are `summaryir` and `extractir`. Both scan through all the directories in the current path and look for info-files. They then extract specific values from the info-file and print them out, sorted chronologically. In the case of `summaryir`, the values extracted are fixed and are most relevant to the CLASSIC data pipeline. If you wish to control which parts of the info-file you want displayed you must use the `extractir` program. The arguments for `extractir` are a list of the keywords you are interested in. So for example

```
{di-centos:1023} extractir V_NORM_12 T0_SCANS
# Name      Type Mod Julian Date  V_NORM_12 T0_SCANS
HD_32630    CAL1 55656.124145590  0.3681 0.2715 0.0191 0.0208
HD_31964    OBJ  55656.128190799  nan    nan    nan 0.0141
HD_32630    CAL1 55656.131780671  0.3218 0.1909 0.0134 0.0285
HD_31964    OBJ  55656.135458171  0.1115 0.0201 0.0014 0.0275
HD_32630    CAL1 55656.140328287  0.3646 0.2451 0.0173 0.0214
HD_31964    OBJ  55656.146489282  0.1083 0.0215 0.0015 0.0266
HD_32630    CAL1 55656.150196331  0.3441 0.2156 0.0152 0.0291
HD_32630    CAL1 55656.153616678  0.4108 0.2567 0.0182 0.0388
HD_31964    OBJ  55656.156611644  0.0820 0.0148 0.0010 0.0221
HD_32630    CAL1 55656.159629456  0.4323 0.2999 0.0211 0.0253
HD_31964    OBJ  55656.166859491  0.1141 0.0197 0.0015 0.0230
HD_32630    CAL1 55656.172361516  0.4680 0.3173 0.0239 0.0227
HD_31964    OBJ  55656.175319155  0.1276 0.0216 0.0016 0.0253
HD_32630    CAL1 55656.178373553  0.4066 0.2454 0.0173 0.0164
HD_32630    CAL1 55656.182141852  0.5070 0.3442 0.0234 0.0281
HD_31964    OBJ  55656.185112697  0.0856 0.0191 0.0013 0.0136
HD_32630    CAL1 55656.188372951  0.3838 0.2715 0.0191 0.0201
```

displays the V_NORM visibility amplitude estimate for the telescopes on beams 1 and 2, as well as the τ_0 estimate. The object name, type and modified Julian data are always displayed.

Since the data reduction programs all read and write to the info-file, if things go wrong, or you wish to start the reduction process again, it is best to remove the whole directory created by the data reduction programs, or move it to a backup directory, and start with a fresh info-file.

3. THE CLASSIC PIPELINES

As mentioned above, there are two CLASSIC reduction pipelines, `reduceir` and `redclassic`. I will deal with each of these separately. If you are not sure which to use, I recommend using redclassic for any visibility amplitude data. For other more complex data, like SFP data, you will need reduceir.

3.1. REDUCEIR - A GTK CLASSIC Data Pipeline

Reduceir consists of a series of standalone command line programs which together perform all the necessary operations in the reduction process. Each of these has numerous flags, many of which I will not discuss in this document. The GTK based GUI reduceir is really just a wrapper that calls these individual programs, most often of the same name as the heading in the GUI itself. In this document, I will only cover the GUI interface itself. Should you need to use the program directly you will need to study the available flags. Again, feel free to contact me directly if you wish to see examples of scripts files that use the lower level programs.

The reduceir program has the following flags:

```
(ctrscrut:1009) reduceir -h
usage: reduceir [-flags]
Flags:
-D[Dir]                 Directory for results (Basename)
-e[scroll_length]       Size of scroll window (200)
-f                      Toggle FITs file format (ON)
-R                      Toggle remote mode (OFF)
-h                      Print this message
```

The most commonly used of these would be the `-e`, which is useful for small screens when you need to reduce the vertical size of the GUI, and `-R`, which will reduce the size of all plots for when you are running the program remotely and the network connection is slow. A screen shot of the main panel of the GUI is given in Figure 1.

There are numerous places in the GUI to enter numbers, most of which will be filled out automatically by the software. These defaults can be changed by clicking and typing, but, except in the case of choosing the integration range as discussed in section 3.1.8, this is rarely necessary, and sometimes quite dangerous.

The basic steps of CLASSIC data reduction are

1. Selecting input files and output directories.

2. Breaking the data file up into shutter and data sequences.

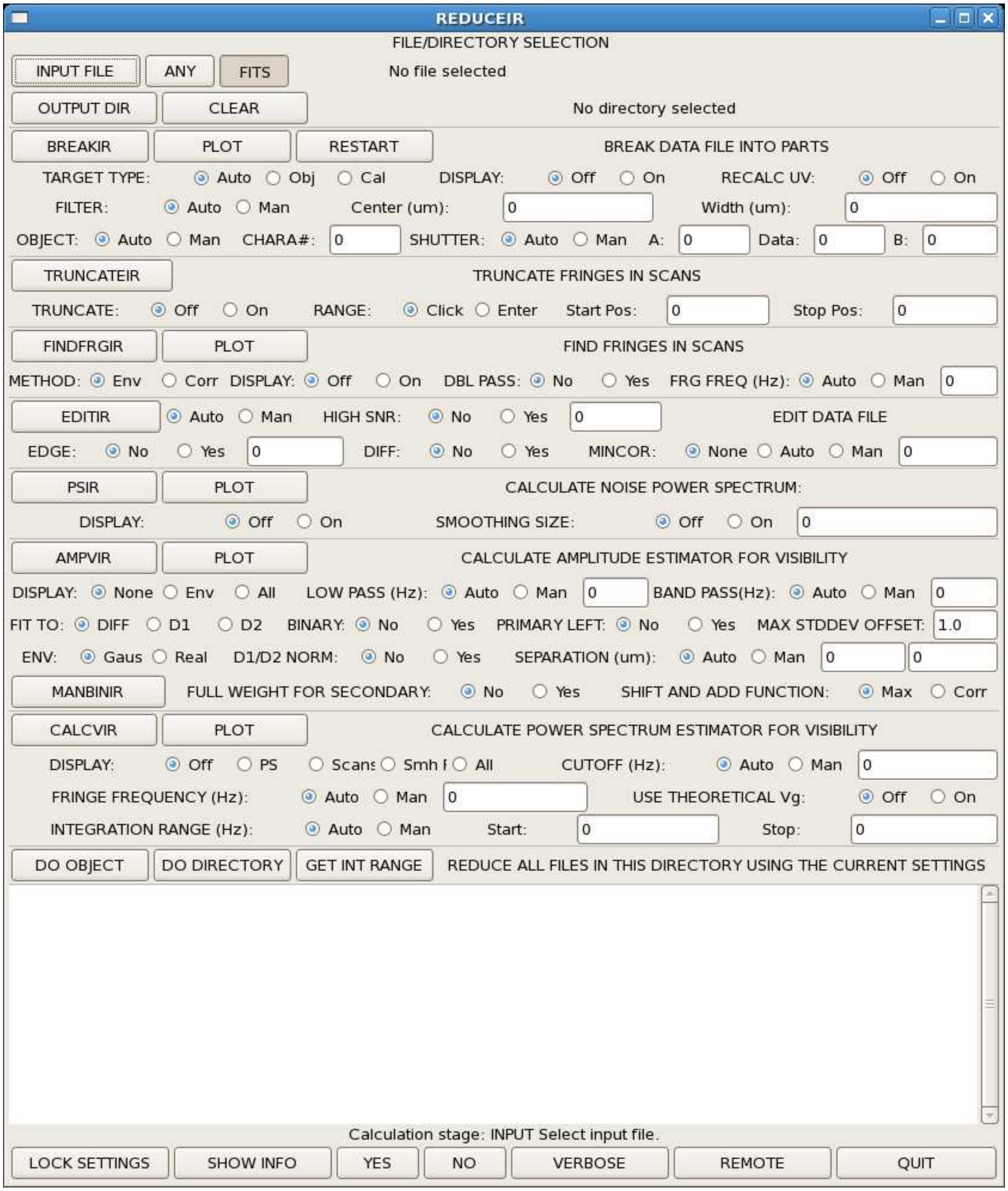
3. Optional truncation of the data scans. This is only done if there is a danger of saturation in the scan and is largely obsolete since the non-destructive read method has also become obsolete. I will not discuss truncation further in this document.

4. Identification and location of possible fringes in the data scans. This is used by the editing process that follows, and in particular automated editing.

5. Editing the data scans.

6. Calculating the background noise power spectra.

7. Calculating the fringe visibility magnitude in amplitude space.



FILE/DIRECTORY SELECTION

No file selected

No directory selected

BREAK DATA FILE INTO PARTS

TARGET TYPE: ● Auto ○ Obj ○ Cal DISPLAY: ● Off ○ On RECALC UV: ● Off ○ On

FILTER: ● Auto ○ Man Center (um): 0 Width (um): 0

OBJECT: ● Auto ○ Man CHARA#: 0 SHUTTER: ● Auto ○ Man A: 0 Data: 0 B: 0

TRUNCATE FRINGES IN SCANS

TRUNCATE: ● Off ○ On RANGE: ● Click ○ Enter Start Pos: 0 Stop Pos: 0

FIND FRINGES IN SCANS

METHOD: ● Env ○ Corr DISPLAY: ● Off ○ On DBL PASS: ● No ○ Yes FRG FREQ (Hz): ● Auto ○ Man 0

● Auto ○ Man HIGH SNR: ● No ○ Yes 0 EDIT DATA FILE

EDGE: ● No ○ Yes 0 DIFF: ● No ○ Yes MINCOR: ● None ○ Auto ○ Man 0

CALCULATE NOISE POWER SPECTRUM:

DISPLAY: ● Off ○ On SMOOTHING SIZE: ● Off ○ On 0

CALCULATE AMPLITUDE ESTIMATOR FOR VISIBILITY

DISPLAY: ● None ○ Env ○ All LOW PASS (Hz): ● Auto ○ Man 0 BAND PASS(Hz): ● Auto ○ Man 0

FIT TO: ● DIFF ○ D1 ○ D2 BINARY: ● No ○ Yes PRIMARY LEFT: ● No ○ Yes MAX STDDEV OFFSET: 1.0

ENV: ● Gaus ○ Real D1/D2 NORM: ● No ○ Yes SEPARATION (um): ● Auto ○ Man 0 0

FULL WEIGHT FOR SECONDARY: ● No ○ Yes SHIFT AND ADD FUNCTION: ● Max ○ Corr

CALCULATE POWER SPECTRUM ESTIMATOR FOR VISIBILITY

DISPLAY: ● Off ○ PS ○ Scans ○ Smh I ○ All CUTOFF (Hz): ● Auto ○ Man 0

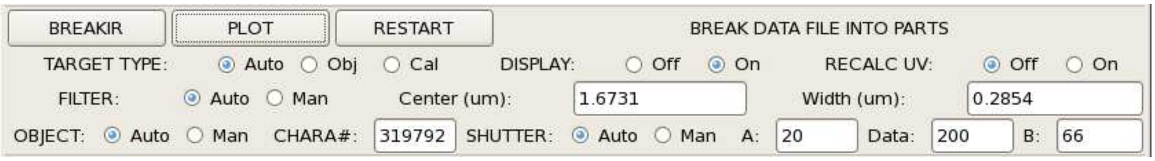
FRINGE FREQUENCY (Hz): ● Auto ○ Man 0 USE THEORETICAL Vg: ● Off ○ On

INTEGRATION RANGE (Hz): ● Auto ○ Man Start: 0 Stop: 0

REDUCE ALL FILES IN THIS DIRECTORY USING THE CURRENT SETTINGS

Calculation stage: INPUT Select input file.

FIGURE 1. A screen shot of the reduceir GTK based interface. Each operation in the reduction process is self contained and separated by horizontal lines and run by a button at the top left of each box.



BREAK DATA FILE INTO PARTS

TARGET TYPE: ● Auto ○ Obj ○ Cal DISPLAY: ○ Off ● On RECALC UV: ● Off ○ On

FILTER: ● Auto ○ Man Center (um): 1.6731 Width (um): 0.2854

OBJECT: ● Auto ○ Man CHARA#: 319792 SHUTTER: ● Auto ○ Man A: 20 Data: 200 B: 66

FIGURE 2. The BREAKIR panel once it has completed working on the example data file.

8. Calculating the fringe visibility magnitude in Fourier space.

I will describe each of these steps in separate sections.

3.1.1. FILE/DIRECTORY SELECTION: Loading a file

One begins the data reduction process by selecting an input file. Note that by default the `FITS` file type is selected, and unless your data was taken before 2008 there is no need to change this. Clicking on the `INPUT FILE` button will bring up file selection window in which you can select the data file you wish to work on. In the examples to follow we will use the file `2011_10_02_HD_220825_ird_002.fit`.

This section of the GUI also allows you to select a director other than the current one, or the one in which you invoked the GUI, for the location of the output directory.

3.1.2. BREAKIR: Breaking the Data File into Parts

The first step in the data reduction process breaks the file up into parts, in particular shutter sequences, background scans, and data scans. Under normal circumstances, you need only click on the `BREAKIR` button and wait for the process to complete. If no error message appears in the text box all is well. You will also notice that all of the other select buttons and text boxes will have been filled out or set to defaults. In rare circumstances you may have to override these defaults and I will go over these in the same order in which they appear in the GUI. Figure 2 shows what the panel should look like once it has completed working on the example file.

To the right of the main `BREAKIR` button is a button called `PLOT`, and you will find a similar button in most stages of data reduction. Clicking any of these buttons will bring up a small panel that will allow you to plot various parts of the data once this part of the process is complete. To the right of this is the `RESTART` button. Clicking this will delete everything you have done on this object and allow you to start the reduction from scratch. You can not undo this, so make sure you really wish to delete your work. The program will ask you to confirm that you really wish to do so, and you will find `YES` and `NO` buttons at the bottom of the GUI. This is a general mechanism in the program, that is, if it asks a question you will use these buttons to answer it.

On the second line of this panel there are three sets of selection buttons for target type, auto-display, and forcing a recalculation of the UV coordinates of the observation.

It is essential that each object is labeled as either an object or a calibrator[2]. This is normally done automatically, but you can override this by clicking on either the `Obj` or `Cal` buttons.

[2] Check stars are considered to be objects

Sometimes, the data file has not been marked as either an object or a calibrator and you will get an error message to that effect. In these cases you will be forced to select it manually.

Like many parts of the data reduction process, you can also choose to turn on displaying the data, but this is rarely used during this stage. The distinction between "display" and "'plot" is as follows. The "display" is generated automatically while the data reduction step takes place and is most often used for debugging, or checks of the data itself. The "plot" is an interactive panel that lets you select which parts of the data you wish to see and to alter the range of these plots. In many cases, a plot will be generated by default when you run one step of the data reduction process.

The last button on the second line of this section is a way of forcing the program to recalculate the UV coordinate of the observation. This dates back from when the data collection routine was not able to calculate UV coordinates and is now rarely necessary.

On the third line of this panel you are able to override the filter characteristics by changing the numbers in the text boxes and clicking on the `Man` button. If you click on `Auto`, the default, these boxes will be filled out based on the information in the data file.

Many stages of the reduction process contain these text boxes and selection buttons, and in the remaining sections I will not discuss them all, only those that are commonly used.

3.1.3. TRUNCATEIR: Truncation of Data Scans

If you have used the non-destructive mode while observing, there is a chance that some, or all, of you scans are saturated and this saturation needs to be removed from the scans. If you used destructive reads you can safely skip this step.

Clicking on the `TRUNCATEIR` button will put up a window like the one shown in Figure 3, although note that the two vertical lines shown in this figure only appear once you have used the mouse to select the area within the scan set you wish to use in the remainder of the data collection process. The data shown in this figure is not part of the example data but has been selected in order to show what saturation looks like, so you will not see this in the example data.

There are four lines plotted by the truncation sequence, two each for the two outputs of the detector. At the top the purple and blue lines show the mean of all scans for the two detector outputs. At the bottom the red and green show the minimum value in each scan location across all scans for the two detector outputs. In the example here, we used non-destructive reads H band. Saturation appears as a smooth drop in the average scans and a more sudden drop in the minimum scans. It is important to include only the part of the scans that contains no saturation - a single saturated scan can bias the result. For this reason, it is normally best to use the minimum scans as a guide as they will show saturation with the data even if it has occurred only as part of a single scan. Finally, note that the residual "reset" noise at the beginning of each scan has also been removed.

3.1.4. FINDFRGIR: Find the Fringes in the Scans.

This part of the reduction process attempts to identify and locate fringes in each scan. These values are then used by the editing process for automatically removing some scans. It is very rare that you need to do anything but click on the `FINDFRGIR` button.

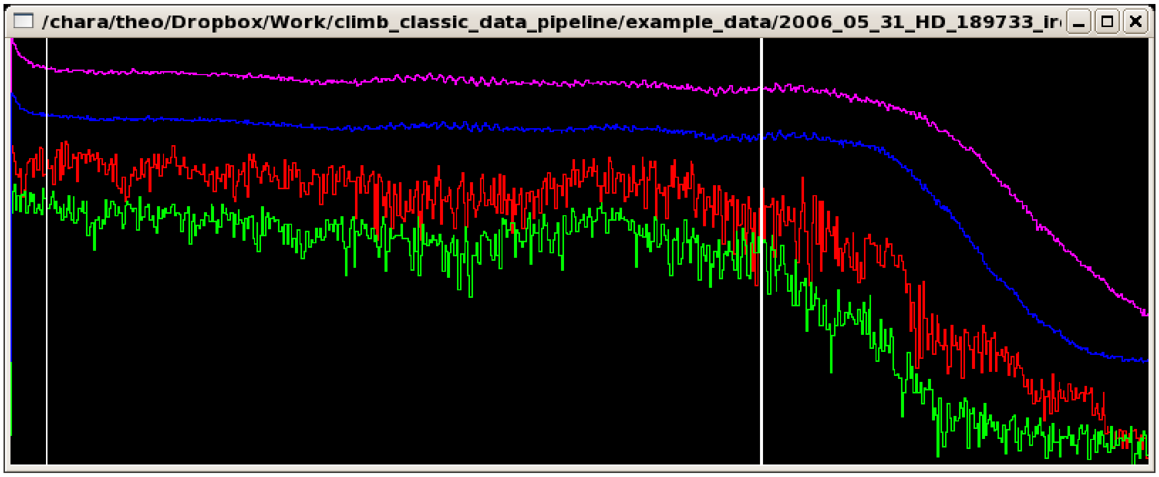

/chara/theo/Dropbox/Work/climb_classic_data_pipeline/example_data/2006_05_31_HD_189733_ir

FIGURE 3. Example of the truncation window. On the left we can see the residual "reset" noise, while on the right we can see the pixel saturate. The vertical white lines show where the user has clicked with the mouse to select the usable portion of the data.

3.1.5. EDITIR: Edit the Data.

Editing the data is probably one of the most difficult, and risky, elements of data reduction. The idea is to remove scans that do not contain fringes, for example when a telescope has lost the star or when the delay lines are not in the correct positions, without removing scans that contain fringes, albeit at very low SNR. If you don't remove enough bad scans you will bias the data towards a lower visibility, while if you remove too many you will do the reverse. The fringe editing method described in section 3.2 based on waterfall plots is much more reliable in this regard and is one reason why I recommend using that pipeline. In my opinion it is better to leave a scan in rather than remove it, as it's normally very clear when fringes have been lost.

Before discussing the data editing process, I should mention that there are four ways of editing the file automatically. The first, always on by default, is labeled `EDGE` and this automatically removes scans whose fringes are not wholly within the scan. By default the fringe center, as defined by the maximum of the fringe envelope, must be at least $\frac{3}{4}$ of the fringe envelope width away from either edge of the scan. The other automatic editing procedures are to include only the N fringes with the highest signal to noise (`HIGH SNR`), remove fringes for which the two detectors disagree on the fringe center location (`DIFF`), and include only those fringes whose SNR is above a specified minimum (`MINCOR`). These editing processes will be performed before the manual editing process begins.

You can also decide to use completely automatic editing by selecting the `Auto` button. In this case you will not be given the chance to manually edit the data at all. This is not a reliable method, and is only typically used when you wish to have a "quick look" at some data, in which case it is common to select a minimum SNR of 1.0.

If you are manually editing the data, once you do hit the `EDITIR` button you will be presented with a plot like that shown in Figure 4, along with a panel of buttons.

At the top of Figure 4 you can see that the fringes were lost at the end of the data sequence, most likely after the final shutter sequence. Note how the fringes suddenly stop - this is a common sign that there is a problem and this section of the data have been removed. The

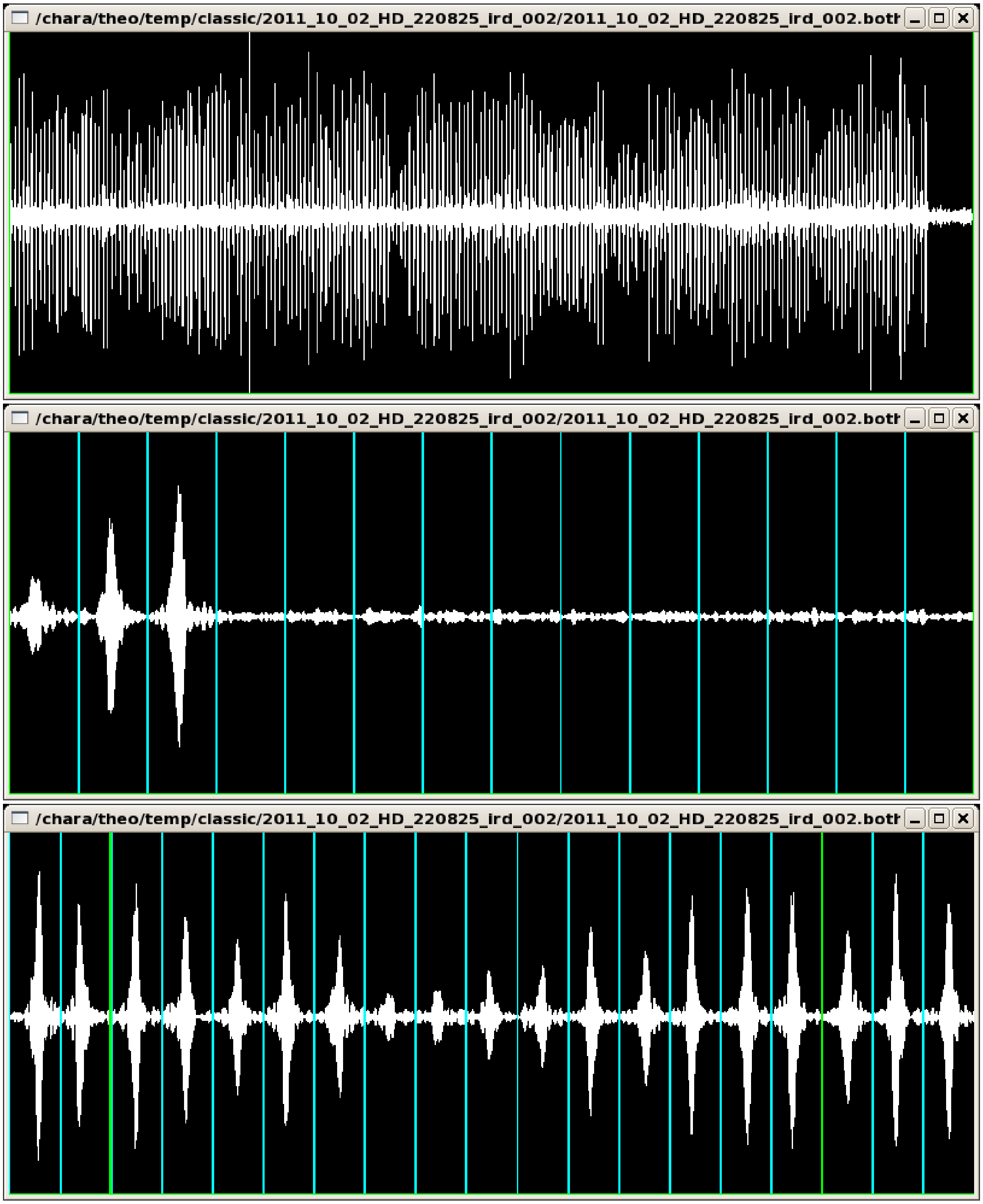

/chara/theo/temp/classic/2011_10_02_HD_220825_ird_002/2011_10_02_HD_220825_ird_002.both
/chara/theo/temp/classic/2011_10_02_HD_220825_ird_002/2011_10_02_HD_220825_ird_002.both
/chara/theo/temp/classic/2011_10_02_HD_220825_ird_002/2011_10_02_HD_220825_ird_002.both

FIGURE 4. Example of the fringe editing window. At the top we see all plots in the scan. The middle plot shows a zoom in of the end of the data sequences where fringes have been lost. The lower plot shows a zoom in of the middle of the data sequences where fringes are seen to go and come back.

middle of Figure 4 shows a zoom of this area. There are other places in the data sequence where fringes seem to have been lost, for example about one third of the way through the data sequence. However, if you zoom into these, as shown at the bottom of Figure 4, you see that they are in fact still there, just at very low SNR. There are times when the SNR is so low that you see no fringe at all, but fringes fading and coming back again like this is caused by seeing and these sections should not be removed from the data file.

The editing panel has numerous buttons:

- `REDRAW` - Will redraw the editing window.
- `SET CURSORS` - Allows you to click on the window and set the cursor positions. Note that they will be set at the nearest scan boundary.
- `MAX CURSORS` - Sets the cursors to include the full data set.
- `MAX LEFT` - Sets the left cursor at the beginning of the data set.
- `MAX RIGHT` - Sets the right cursor at the end of the data set.
- `ZOOM IN` - If the cursors have been set zooms in to that area of the data sequence. If they have not been set zooms in a little.
- `OUT` - Zooms out a little.
- `<<` - If you are zoomed in moves the current area towards the beginning of the data sequence.
- `>>` - If you are zoomed in moves the current area towards the end of the data sequence. These two buttons are handy for moving through the data sequence from one end to the other while you are zoomed in and checking for scans without fringes.
- `SHOW ALL` - Zooms out all the way.
- `SHOW SCAN ENDS` - Toggles the light blue lines seen in Figure 4 that show the scan boundaries.
- `DEL FRNG` - Deletes a single fringe scan using a single click.
- `DEL SECT` - Deletes all scans between the current cursor positions.
- `SAVE FILE` - Saves the file.
- `QUIT NO SAVE` - Quits without saving the file.
- `SAVE & QUIT` - Saves the file then quits.

Note that there is no “undo” command. If you decide you need to start again, you can exit without saving and hit the `EDITIR` button again. You will be given the option of editing the current version or the previous version. To really begin from scratch hit the `RESTART` button, but bare in mind you will start the whole data reduction process from the beginning.

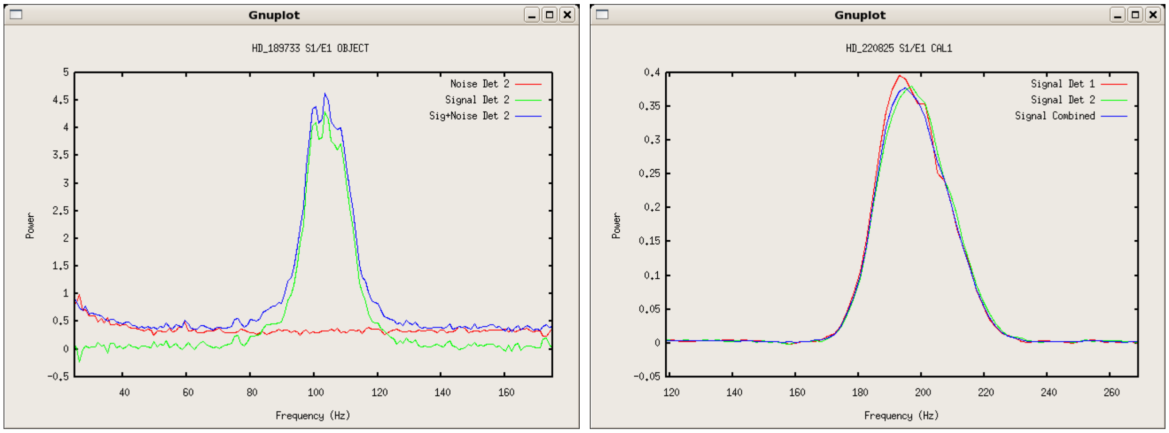


FIGURE 5. (Left) An example of a relatively low SNR power spectrum. In this case we show the power spectra of the same data as shown in Figure 3. This plot shows the raw signal plus noise, the noise estimate, and the estimated signal only power spectra for the second detector output. (Right) An example of a high SNR power spectra. These are for the example data used above. This plot shows the estimated signal power for both output detectors and the difference signal.

3.1.6. PSIR: Calculate the Noise Power Spectra.

In this step in the reduction process the mean power spectra of the fringes and the background noise are calculated. Apart from the display for debugging, the only option here is to turn on smoothing for the power spectra. This is sometimes useful for very low signal to noise data. In all other cases one need only hit the `PSIR` button. Once the calculation is complete the program will automatically display a plot of the noise subtracted power spectra, like those shown in Figure 5.

It is these plots of the power spectra that are your best tool for evaluating the quality of the data. For example, the left hand plot of Figure 5 shows one of the two detector outputs for the saturated, and truncated, H band data also shown in Figure 3. Here we can see the raw signal power spectrum, which also contains noise (blue line), the estimate of the noise power spectrum obtained either from the shutter sequences for old data, or the off fringe sequence for more recent data (red line), and the difference of the two, presumably the spectrum of just the signal (green line). Here you can see that the noise estimate nicely lies on the noise bed of the raw signal and the final signal only spectrum is close to zero either side of the signal peak. Note that you can see the effect of scintillation noise in the lower frequencies of these plots, and that it is not present in the final signal spectrum estimate. The right hand plot of Figure 5 shows a much higher SNR plot of the example data. In this case we have plotted only the signal estimators, but for the two detector outputs (red and green lines) as well as the difference signal (blue line). Note that they all are nicely overlapped, and all are close to zero away from the peak representing the fringe power.

Things to look out for here are spikes of noise, for example at 60 Hz or multiples thereof, non-zero power away from the fringe peak, odd shaped fringe peaks, and large differences between the two detectors and the difference signal. This last is less important, as the final calculation is almost always done using the difference signal, but it does indicate poor alignment of the beam combiner during the observation and you may wish to flag a data set like that to be considered as un-reliable when it comes time to calibrate the data.

Another important thing to think about here is the final integration range you will use for

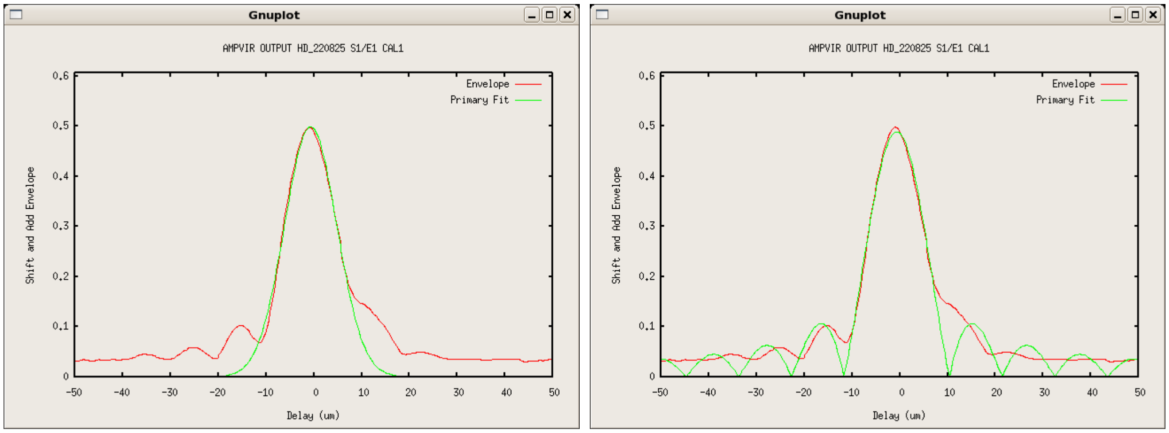


FIGURE 6. (Left) The mean fringe envelope of the high SNR data example (red) with a Gaussian fit over plotted (green). This is the default plot produced by the AMPVIR routine. (Right) The same data, except with a fitted theoretical envelope function.

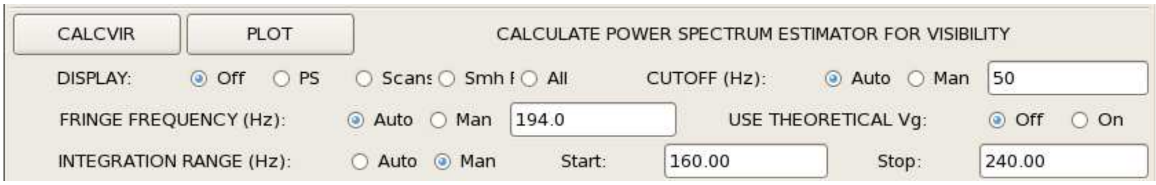
the power spectrum based visibility amplitude estimate. As the seeing changes the width of the fringe peak will change, getting narrower in good seeing and broader in poor seeing. It is always a good idea to use the same integration range on all data files in a bracketed data set, and so you must choose an integration range that will include all the power of all data sets in the bracket, while minimizing the addition of background noise. I will normally process several data sets in a bracket up to this points, some near the beginning, some near the end, and some in the middle of the bracket, and choose an integration range that covers all the data in the bracket. For example, a good range for the low SNR example in Figure 5 would be 70 to 130 Hz, while for the high SNR spectra I would use 160 to 240 Hz. As we shall see in section 3.1.8 the software is capable of choosing the integration range but it is best to manually choose one that best suits the bracket of data.

3.1.7. AMPVIR: Calculate the Time Domain Estimator.

The next step is to calculate a visibility estimator based in amplitude space, both by looking at the fringe envelope and by directly fitting a fringe function to the bandpass filtered data as described in the theory technical report. There are many options in this section of the reduction pipeline, but they are almost all for use when processing SFP data and this will not be covered here. By default, a plot of the mean fringe envelope will be displayed with either a Gaussian fit, the default, or a fitted fringe envelope function. Examples of these are shown in Figure 6. Normally one need only hit the `AMPVIR` button and wait for the process to complete.

The results of the process will be displayed in the text window, were in the case of this example file we get

```
Vamp: N = 219
Detector 1: V = 0.5457 StdDev= 0.0992 StdErr = 0.0067
Detector 2: V = 0.5663 StdDev= 0.0995 StdErr = 0.0067
Vfit: N = 216 V = 0.4846 StdDev= 0.0868 StdErr = 0.0059
Vcmb: N = 219 V = 0.5394 StdDev= 0.0949 StdErr = 0.0064
Fitted waveband: 1.6502 0.2437
```



DISPLAY: ● Off ○ PS ○ Scans ○ Smh ○ All CUTOFF (Hz): ● Auto ○ Man 50

FRINGE FREQUENCY (Hz): ● Auto ○ Man 194.0 USE THEORETICAL Vg: ● Off ○ On

INTEGRATION RANGE (Hz): ○ Auto ● Man Start: 160.00 Stop: 240.00

FIGURE 7. Example of setting the integration to range to the one selected earlier, in this case from 160 to 240 Hz. Note that we have selected the manual setting for the integration range.

Here we see that there where 219 scans included in the fringe envelope based reduction, while only 216 were included in the fringe fitting reduction. Note also that the various fringe visibility estimators do not agree, although they are all within one standard deviation. This, while not desirable, is not uncommon. Note that the two detector values are also envelope estimates and agree fairly well with the final combined envelope estimate. Furthermore, the fitted waveband is very close to what we would expect for H band data.

This amplitude estimator by its nature includes all the noise present in the signal. This gives the envelope estimate a tendency to over-estimate and the fringe fitting method to under-estimate the fringe signal. Unless one is processing SFP data these estimators are rarely used in the final calibration step. Nevertheless, this step is a useful one for assessing data quality and as a check on the final power spectrum based estimator described in section 3.1.8.

3.1.8. CALCVIR: Calculate the Frequency Domain Estimator.

In this, the final part of reduction of a single file, we obtain a visibility estimate using the frequency domain estimator, that is, an integration of the fringe peak in the power spectrum. In most cases, the only option you will use here is to manually set the integration range to the one you have chosen for this data bracket. This requires that you click on the `Man` selector button in the `INTEGRATION RANGE` area as shown in Figure 7.

Once you have selected the integration range and hit the `CALCVIR` button you will have the option of plotting out the results, as shown in Figure 8. Note how the last visibility estimates are very low. This is a sign that there is a problem in these scans and you may wish to go back to the editing stage and remove these scans. In this case these are the scans after the shutter sequence and I have left them in place just as an example. Ordinarily they would have been removed in the editing process. You will also see the final results printed in the text box, for example:

```
CALCVIR RESULTS:
Detector 1: Group Velocity = 328.1760 StdDev= 10.1800
Detector 2: Group Velocity = 329.7510 StdDev= 10.0300
Difference: Group Velocity = 328.8390 StdDev= 9.6100
Detector 1: t0(Lambda0) = 26.2 mS
Detector 2: t0(Lambda0) = 26.5 mS
Difference: t0(Lambda0) = 26.5 mS
Detector 1: t0(0.5um) = 6.1 mS
Detector 2: t0(0.5um) = 6.2 mS
Difference: t0(0.5um) = 6.2 mS
Detector 1: V^2= 0.3188 StdDev= 0.1265
```

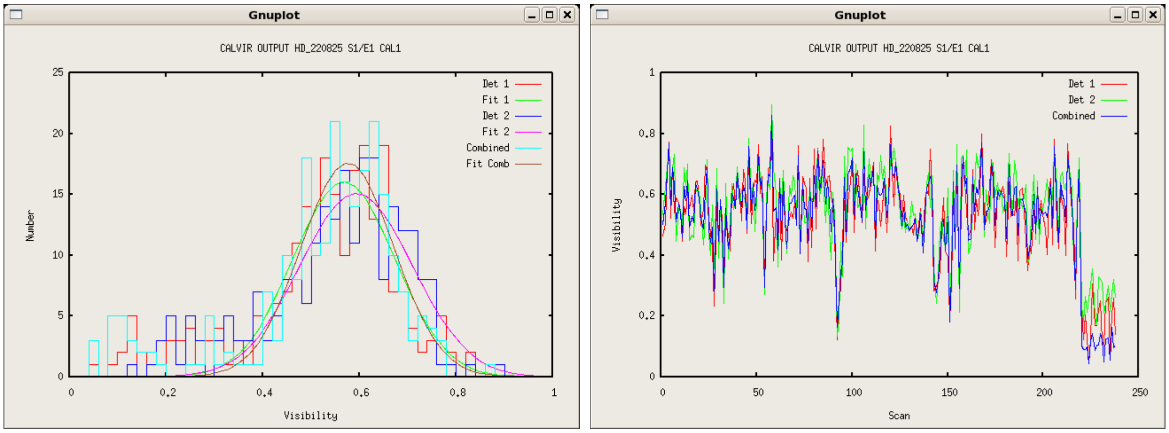


FIGURE 8. Examples of the possible plots of the results of running CALCVIR. (Left) Here we plot a histogram of the resulting visibility estimates, along with fitted Gaussian curves. (Right) A plot of the individual visibility estimates. Note how the last last few are very low compared to the others.

```
Detector 2: V^2= 0.3445 StdDev= 0.1333
Difference: V^2= 0.3235 StdDev= 0.1175
Detector 1: V= 0.5517 StdDev= 0.1198
Detector 2: V= 0.5736 StdDev= 0.1244
Difference: V= 0.5578 StdDev= 0.1113
Gaussian 1: V= 0.5531 StdDev= 0.1889
Gaussian 2: V= 0.5757 StdDev= 0.1916
Difference: V= 0.5592 StdDev= 0.1748
LogNorm  1: V= 0.5544 StdDev= 0.1070
LogNorm  2: V= 0.5768 StdDev= 0.1086
Difference: V= 0.5601 StdDev= 0.0993
```

Here there are estimates for the group velocity, the τ_0 for both the observing waveband and for 0.5μm, along with several visibility estimators. It is always a good idea to check that these basically agree, and are not more than one sigma away from the amplitude estimates discussed in the previous section.

3.1.9. Reducing all Object Files or Whole Directories.

Once you have completed one data file in a bracket, and selected the integration range you wish to use in the visibility calibration, it is now possible to process either all the files, or all data files in the current working directory. This can be done by either hitting the `DO OBJECT` or `DO DIRECTORY` buttons. This will lock in all the settings you have and move through all the same steps for the range of objects you have selected.

3.1.10. Other Buttons

You will have noticed a number of buttons on the bottom of the reduceir panel. These are:

- `LOCK SETTINGS` This locks out any changes to the settings, for example the integration range will no longer be calculated by the software. This is automatically set to on when you reduce all the files for an object or in a directory.

- SHOW INFO This will print the entire info file into the text box.
- YES and NO Sometimes you will be asked a question and these buttons can be used to respond.
- VERBOSE Puts the program into verbose mode in which it will print a great deal of information about what it is doing. This is normally only useful for debugging.
- REMOTE This allows you to turn remote mode on and off. All remote mode does is make the plot windows smaller for use over slow connections.
- QUIT I'm sure you can guess what this does.

3.2. REDCLASSIC - A Command Line CLASSIC Data Pipeline

Once the CLIMB data pipeline had been written we found that it was very useful to have a pipeline that could be run without the need for a GTK interface. This newer package does pretty much the same thing as the older one, except that it does not include fringe fitting or the SFP analysis tools. The fringe editing is also different. As we improve things, this newer package will be maintained, and I'd recommend using it if you can.

There is only a single command in this package, and it has the following command line structure:

```
{di-centos:1029} redclassic -h
-a              Toggle apodize for FFT (OFF)
-A              Use only shutter sequence A (OFF)
-b(mid,range)   Change bandpass filter (Auto or 200,0.25)
-B              Use only shutter sequence B (OFF)
-d[0,1,2,3]     Set display level(1)
-D[Dir]         Directory for results (Basename)
-e              Toggle edit scans (ON)
-h              Print this message
-i              Toggle manual integration range (OFF)
-I[start-stop]  Set integration range of data (AUTO)
-l[freq]        Use low pass instead of mean (5Hz).
-M              Toggle manual data selection in scan (OFF)
-O              Toggle Photometry Only (OFF)
-P[smooth_noise_size]   Change noise PS +-smooth size (3)
-R              Toggle remote mode (OFF)
-S[smooth_signal_size]  Change +-smooth size (1)
-t[start,stop]  Truncate scans (OFF)
-u              Toggle noise PS multiplier (OFF)
-U[freq]        Set DC suppression frequency (20.0 Hz)
-v              Toggle verbose mode (OFF)
-V              Print version number.
-w[lambda0,dlambda]     Set wavelength (2.1329,0.3489)
-z[pixmult]     Set pixel multiplier (2)
```

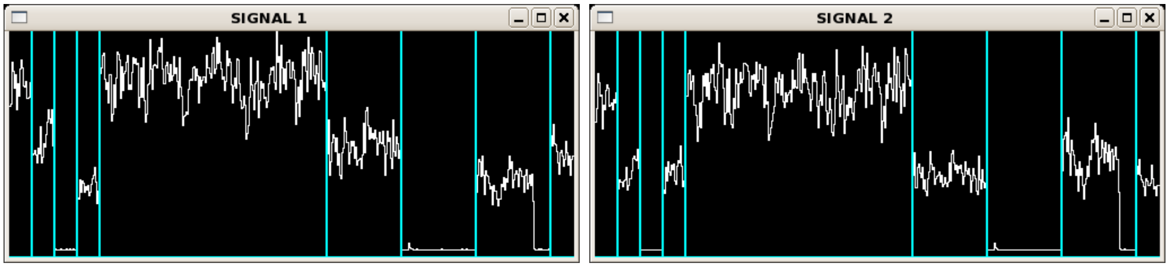


FIGURE 9. This is an example of the first plots normally made during the reduction of data with the redclassic program. These plots are placed vertically, rather than horizontally as shown here. There will also be a third, empty, window not show here. Note that these data were taken in 2011 and so still have the second shutter sequence. More recent data will show background scans instead.

Many of these flags will be discussed as we go along, and the remainder will be explained at the end of this section. The defaults are often all that are required, although it is common to use the `-i` and `-I` flags to control the integration range.

We begin by invoking the command with something like.

```
redclassic 2011_10_02_HD_220825_ird_002.fit
```

where we are reducing the same file as we did above. Unless you have changed the display level by using the `-d` flag, the first thing you will see will be a plot of the photometry as show in figure 9.

This plot of the photometry is a great aid in checking that the star was not lost in either telescope, which sometimes happens. In fact, the star was indeed lost in the second scope during the second shutter sequence in this file. Since these data were taken in 2011 the second shutter sequence was still in use. In this case, you may wish to use the '-A' flag and use only the first shutter sequence. In more recent data you will see the total counts suddenly drop.

Should you need to truncate the data, for example if you are using non-destructive reads, you can use the `-t` flag. If you supply numbers to this flag you can force the truncation points. If you do not, you will need to click the mouse in the data display exactly as described in section 3.1.3.

As before, the next step in the process is to edit the fringes, and very much the same rules apply. Instead of seeing a display of the filtered fringes themselves, this program shows a waterfall plot of the fringe envelopes, as shown in Figure 10. In this figure you can see that one section has been edited out of the data. The edits you do will be remembered from one run to the next. Once editing you will be asked what you would like to do:

```
Move on, zoom in, Zoom out, Edit, Redraw or Clear (m/z/Z/e/r/c)?
```

You will need to type a key and then the enter key. The advantage of the waterfall plot is that you can easily see when fringes have been lost due to delay errors as you can see them drift across. If the fringes don't drift off, but fade in an out you should not remove them as this is most likely caused by seeing and removing them will bias the measurement statistics. The "Clear" function removes all of your edits. Once you have finished editing you need

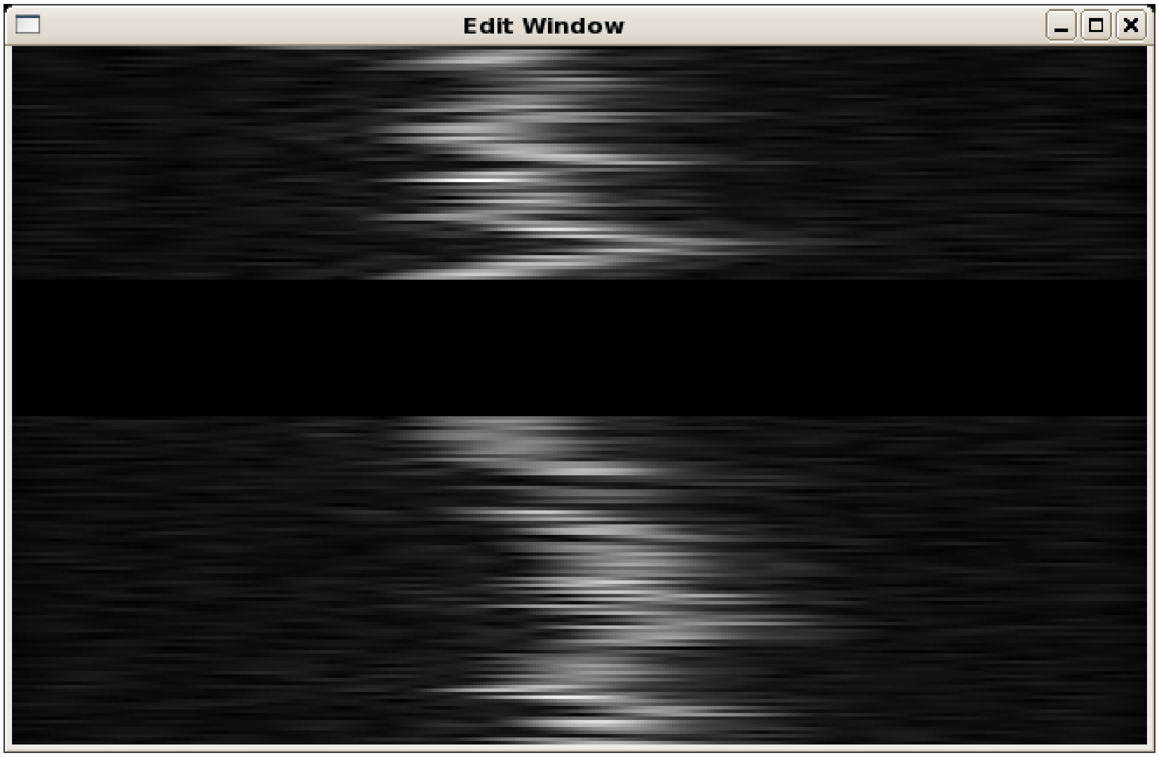


FIGURE 10. This is an example of the redclassic edit window, with an arbitrary section in the middle removed.

to type ‘m’ for “Move on”. Since your edits are saved from one run of the program to the next, once you have edited a file, you can skip this step by using the `-e` flag.

Once the data has been edited, it is time to select the integration range. If you do not use either the `-i` or `-I` flags, the program will pick the integration range automatically based on a Gaussian fit to the peak of the power spectrum. An example is given in Figure 11. Once you have selected the integration range the program will print out something like:

```
# To get this integration range use
# -I151.78-242.02
```

so you can force the same integration range onto other files in the same bracket of observations.

Once you have selected the integration range you will be presented with a plot of the noise subtracted power spectra of the two output channels, and the difference signal, as shown in Figure 12. This plot is a good diagnostic of how well the noise subtraction process has gone. The light blue line at the bottom is the zero point, and away from the fringe peak the data should be close to this zero point. If it is not, then somethings has gone wrong, for example during the shutter sequence, or the data themselves may not be very good.

There are a few things you can do to try and improve the noise subtraction. For example, the program by default tries to ensure that the noise estimate fits the high frequency noise well by working out a multiplier that best fits the data. If the high frequency noise power spectrum is particularly noisy, this can cause a problem, so it’s worth trying the reduction

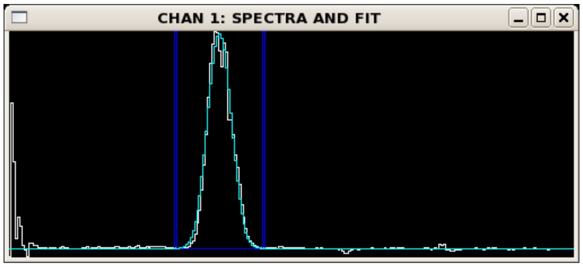


FIGURE 11. This is an example of the redclassic display of the selected integration range. Here the data are in white and the fitted Gaussian is in light blue.

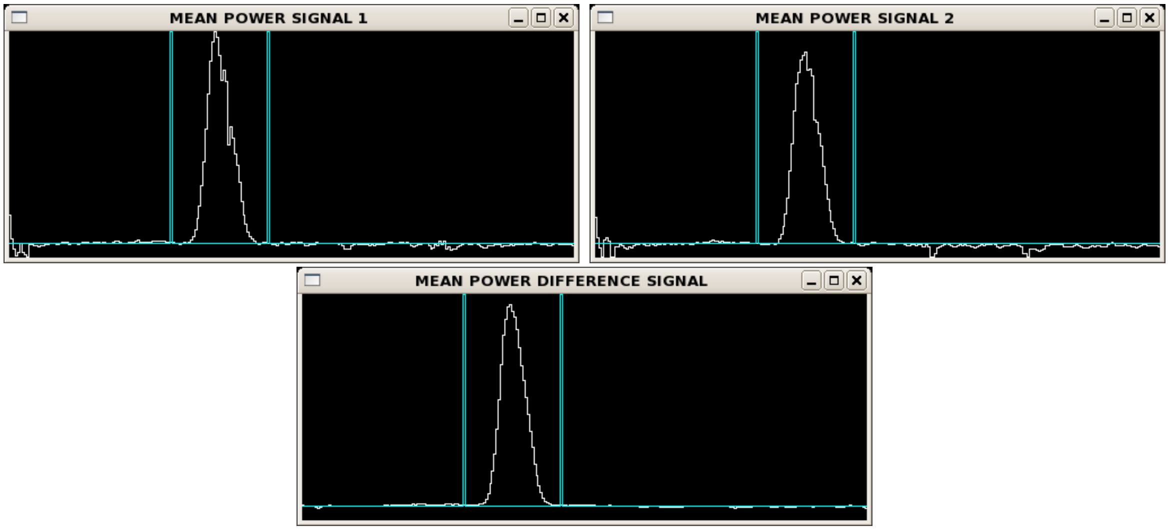


FIGURE 12. This is an example of the redclassic display of the mean noise subtracted power spectra for the two output signals and the difference signal. Here the data are in white and the fitted Gaussian is in light blue. Note that in order to save space the arrangement of these windows has been changed.

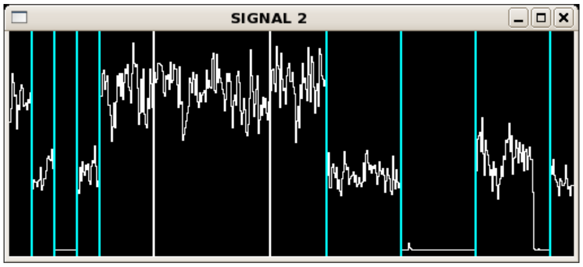


FIGURE 13. This is an example of the redclassic display for manually selecting the on and off fringe data. Here the two white vertical lines show where the user has clicked to include a smaller part of the data.

with this feature turned off by using the `-u` flag. Another thing to try is to increase the amount of smoothing done on the power spectrum by using the `-R` and/or `-S` flags. In the case of old data it's also possible to use only the first or second shutter sequences, and in newer data force the use of the first shutter sequence for noise estimates rather then the off fringe data at the end of the file. To do this you use the `-A` and/or `-B` flags. Finally in some cases where the seeing is particularly bad, there is a great deal of noise in the low frequencies. This can be suppressed using the `-V` flag.

Once you are satisfied that the noise subtraction is working correctly hit the enter key and the program will complete the data reduction process, creating an info-file in much the same way as described in section 3.1.

There are several other flags one can use, but in most cases they are not necessary. The only one left that you may need is the `-M` flag to turn on manually selecting the on and off fringe data. The selection of the on-fringe data is in fact redundant with fringe editing, but selecting the off-fringe data is not. In some cases the star is lost from one, or both, telescopes during the off-fringe process and it is important to use only the part of the off-fringe data that has both telescopes. As for selecting the integration range, this is a matter of clicking the mouse to surround the area you wish to use. An example of this is given in Figure 13.

3.3. Calibrating CLASSIC Data

With all the data for a sequence reduced, the final step is to calibrate the data and create an OIFITS file. This is done using the program `calibir`, which has the following flags:

```
{di-centos:1149} calibir -h
usage: calibir [-flags] {OBJ CAL1,CAL2,CAL3...}
Flags:
-b[beta]                Set intensity ratio (Cal/Obj) (1.0)
-B[Vis Type]            Set Vis estimator for Object and Calibrator (V_LOGNORM)
-c                      Use CHARA number for identifier (OFF)
-C[Cal Vis Type]        Set Vis estimator for Calibrator (V_LOGNORM)
-d                      Add change in calibrator to error (ON)
-f[oif]                 Set OIFITS filename (From object)
-F                      Toggle saving OIFITS file (OFF)
-h                      Print this message
-H                      Use HD number for identifier (OFF)
```

```
-i                        Use ID/Name for identifier (OFF)
-J[mjdmin,mjdmax]         Restrict MJD range (All)
-I[12|23|31]              Select CLIMB baseline (None)
-n                        Use standard error instead of standard deviation (ON)
-o                        Invert the sign of the UV coords (OFF)
-O[Obj Vis Type]          Set Vis estimator for Object (V_LOGNORM)
-r                        Print raw data (ON)
-s[diam1-err,diam2-err,...]     Size of calibrators in mas (0.0)
                          If error left out it is set to zero.
-S                        Self Calibrate mode (OFF)
-v                        Verbose mode (OFF)
```

If the data are correctly labeled as calibrator or object, and the calibrator is unresolved you can just type the command `calibir` and it will do its best to perform a calibration. By default the calibir program outputs text to let you know what it is doing. This text is many characters wide, and so in the example below it has been broken up into two parts. The first part is:

```
#  Name     HD Num CHARA# Type  MJD       T1 T2  U(m)   V(m)   BL(m)
C HD_220825 220825 319792 CAL1 55836.198 S1 E1 193.67 251.71 317.59
O HD_222603 222603 319897  OBJ 55836.202 S1 E1 197.70 252.28 320.52
C HD_220825 220825 319792 CAL1 55836.205 S1 E1 189.42 251.88 315.16
O HD_222603 222603 319897  OBJ 55836.209 S1 E1 193.86 252.53 318.36
C HD_220825 220825 319792 CAL1 55836.212 S1 E1 184.39 252.07 312.31
O HD_222603 222603 319897  OBJ 55836.216 S1 E1 188.92 252.82 315.61
C HD_220825 220825 319792 CAL1 55836.220 S1 E1 178.86 252.26 309.23
O HD_222603 222603 319897  OBJ 55836.237 S1 E1 173.41 253.55 307.18
O HD_222603 222603 319897  OBJ 55836.244 S1 E1 167.77 253.78 304.22
C HD_220825 220825 319792 CAL1 55836.254 S1 E1 148.91 253.02 293.59
O HD_222603 222603 319897  OBJ 55836.257 S1 E1 155.77 254.19 298.13
C HD_220825 220825 319792 CAL1 55836.260 S1 E1 142.40 253.15 290.45
O HD_222603 222603 319897  OBJ 55836.264 S1 E1 149.34 254.39 294.99
C HD_220825 220825 319792 CAL1 55836.267 S1 E1 135.02 253.29 287.03
O HD_222603 222603 319897  OBJ 55836.270 S1 E1 142.35 254.59 291.68
C HD_220825 220825 319792 CAL1 55836.274 S1 E1 127.93 253.41 283.87
```

and the second part is:

```
Vis    Stddev Scans Lambda dLambd  Vcal   Err     dT
0.5637 0.0048   200 1.6731 0.2854
0.4231 0.0044   200 1.6731 0.2854 0.9781 0.5341   9.55
0.3220 0.0086   200 1.6731 0.2854
0.4006 0.0052   200 1.6731 0.2854 0.9179 0.4938  10.48
0.5588 0.0060   200 1.6731 0.2854
0.4480 0.0046   200 1.6731 0.2854 0.7793 0.0422  10.67
0.5891 0.0050   200 1.6731 0.2854
0.4915 0.0047   201 1.6731 0.2854 0.8221 0.0254  48.66
0.4775 0.0063   201 1.6731 0.2854 0.7942 0.0258  48.66
0.6062 0.0063   202 1.6731 0.2854
0.4919 0.0048   201 1.6731 0.2854 0.8164 0.0137   9.27
```

```
0.5992 0.0061   201 1.6731 0.2854
0.5389 0.0056   205 1.6731 0.2854 0.8758 0.0462  10.12
0.6307 0.0059   200 1.6731 0.2854
0.5167 0.0033   200 1.6731 0.2854 0.8250 0.0132   9.39
0.6224 0.0054   201 1.6731 0.2854
```

On the very right hand side we can see the final visibility estimate, its error and the amount of time in minutes between measurements of the calibrator. The error is based on the formal errors in the measurement process along with the amount of change in the calibrator visibility. So, for example, in these data there was a large change in calibrator visibility on the second measurement, and so the first two calibrated visibilities have very large errors. One might consider removing this bad calibrator, which is done by removing the directory that contains the relevant info file. This results in

```
Vis    Stddev Scans Lambda dLambd  Vcal   Err     dT
0.5637 0.0048   200 1.6731 0.2854
0.4231 0.0044   200 1.6731 0.2854 0.7523 0.0114  20.03
0.4006 0.0052   200 1.6731 0.2854 0.7152 0.0126  20.03
0.5588 0.0060   200 1.6731 0.2854
```

for these points. The range of data included in the calibration can also be controlled using the `-J` flag to restrict the allowed range of MJD.

The program will always try and find the nearest calibrator measurements before and after each measurement of the object, even if they are different calibrators. If this process goes well we need only add the `-f` or `-F` flags in order to save these results in an OIFITS file.

Of course, things quite often go as well as we would like. Frequently, for example, the data has not been correctly tagged as calibrator or object, or if you have two calibrators you may wish to calibrate one against the other in order to check the data quality. You can tell calibir which stars are to be treated as object or calibrator by using the `-c`, `-H`, or `-i` flags. For example you could use

```
calibir -H 222603 220825
```

where the first number is the HD number of the object and it is followed by a list of calibrators. Furthermore, you may have estimates for the diameter of the calibrators which must be taken into account in the calibration process. You can tell calibir about these using the `-s` flag:

```
calibir -H -s0.306-0.012 222603 220825
```

where the size estimate and error are separated by a `-`. If there is more than one calibrator the remaining size estimates must follow use a comma as a separator:

```
calibir -H -s0.306-0.012,0.543-0.034 222603 220825 220945
```

Figure 14 shows the final fit to a uniform disk for these data using the program `oifud`. For this fit, the errors bars are adjusted to force the $\chi^2 = 1.0$.

By default, calibir creates a V^2 estimate based on the V_LOGNORM calculation in the reduction process. You may choose to save this directly as a V instead by using the `-2`

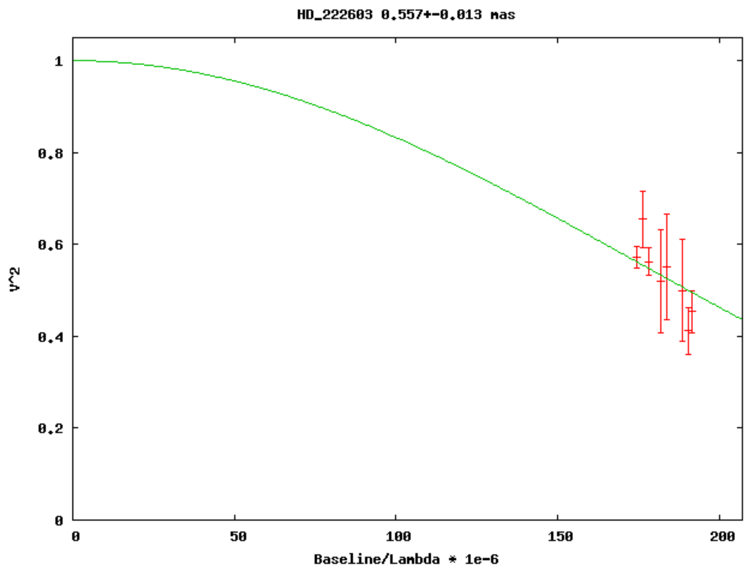


FIGURE 14. Output of the oifud uniform disk fit to the example CLASSIC data.

flag. It is also possible to select a different form of visibility estimator by using the `-B`, `-C`, or `-O` flags. It is not normally a good idea to use different types of visibility estimates for calibrators and objects so I recommend using the `-B` flag most of the time. Popular alternatives to the V_LOGNORM estimate are V2_SCANS and V_NORM.

4. REDCLIMB - A CLIMB DATA PIPELINE

The climb data reduction program is called `redclimb`, and at the time of writing this document there is no GTK based interface for this program. The climb reduction program has the following flags:

```
{di-centos:1018} redclimb
usage: redclimb [-flags] ir_datafile
Flags:
-a              Toggle apodize for FFT (OFF)
-A              Use only shutter sequence A (OFF)
-b              Toggle redo background and beams (OFF)
-B              Use only shutter sequence B (OFF)
-c              Toggle pixels must agree (ON)
-C              Toggle confirm (ON)
-d[0,1,2,3,4,5] Set display level(1)
-D[Dir]         Directory for results (Basename)
-e              Toggle edit scans (ON)
-E[weight]      Use fringe weight to edit for AMP (OFF)
-f[width_frac]  Envelope width fraction (0.35)
-F              Toggle filtering of signal (ON)
-g[0-1]         Fraction of Gaussian to include in CP (0.1)
```

```
-G                Toggle using AMP editing for CP (ON)
-h                Print this message
-H                Toggle use complete scan for closure (OFF)
-i                Toggle manual integration range (OFF)
-I[12-12,23-23,31-31]   Set integration range
-k                Toggle skip visibility calculation (OFF)
-l[freq]          Use lowpass filter signal for normalization (OFF)
-m                Toggle save means as text (ON)
-M                Toggle manual data selection in scan (OFF)
-n                Toggle use noise instead of signal (OFF)
-Nndata           Force data segment size (AUTOMATIC)
-o                Toggle skip overlap test (ON)
-O                Toggle photometry only (OFF)
-p                Toggle plot closure phases (ON)
-P[smooth_noise_size]   Change noise PS +-smooth size (4)
-r                Toggle save raw data as text (OFF)
-R                Toggle remote mode (OFF)
-s[n]             Scans to skip after shutter change (0)
-S[smooth_size]   Change +-smooth size (1)
-t[start,stop]    Truncate scans (OFF)
-u                Toggle noise PS multiplier (ON)
-U[freq]          Set DC suppression frequency (10.0 Hz)
-v                Toggle verbose mode (OFF)
-V                Print version number.
-w[weight]        Set minimum fringe weight for CP (0.5)
-W[width]         Set PS peak width for fit (10.0 Hz)
-x                Toggle use dither freqs for fringes (OFF)
-z[pixmult]       Set pixel multiplier (2)
-#[start,stop]    Set range of scan to include (ALL)
```

Many of these flags are the same as those for `redclassic` and have exactly the same purpose and these will not be discussed again in this section. As with classic data the process begins by typing a command like

```
redclimb 2013_05_02_HD_122364_climb1_001.fit
```

and like redclassic, unless you have changed the default display level by using the `-d` flag, the first thing the program does is to plot the photometric signals for the three outputs pixels. An example is shown in Figure 15. Note that in pixel only two beams are present and so the second shutter event contains no light in that pixel. If light from a scope is lost during one of the shutter events, or during the off-fringe scans, you can choose to use only the first or second shutter sequences by using the `-A` or `-B` flags, or edit the scans using the `-M` flag.

The second plot shown by default is the noise subtracted power spectra for the three output pixels as shown in Figure 16. Note that the DC part is suppressed, and that pixel 1 only sees a single baseline. The other two pixels should show three peaks for the three baselines. This is a good time to check how well the noise subtract process has worked, and all the same tricks and flags apply to redclimb as it did for redclassic. Since climb has fringes at rather low frequencies the noise subtraction is rarely as good as for classic, and the peaks

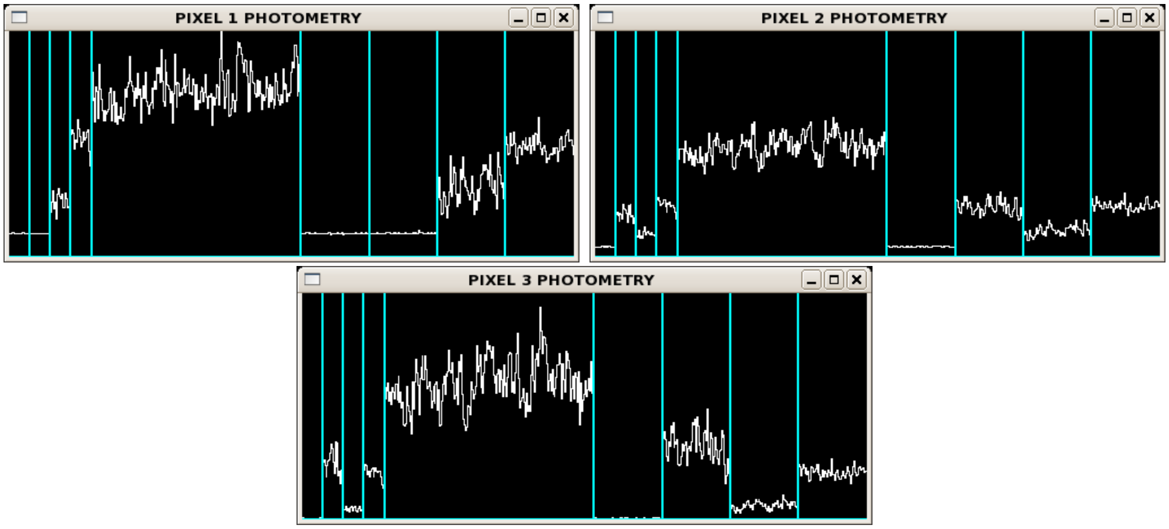


FIGURE 15. This is an example of the first plots normally made during the reduction of climb data with the redclimb program. Note that, like the example classic data, still have the second shutter sequence and that in order to save space the arrangement of these windows has been changed.

may be broad and therefore cause cross-talk between baselines. These sorts of faults should be noted and taken into consideration in the calibration process.

The third plot shows Gaussian fits to the three peaks, one for each baseline and one in each window, as shown in Figure 17. This is a very important step, because it is with these fits that the software establishes the true fringe frequencies, which will be used in the calculation of the closure phase. If this is wrong, the closure phase calculation will fail. In this case, the software got it correct, but in poor SNR data this sometimes fails. If this is the case, you can force it to use the frequencies chosen at the time of the observation by using the `-x` flag. The next step is fringe editing, which is very much like fringe editing for classic data, except there will now be three baselines for you to edit.

Next we choose the integration range, as shown in Figure 18, and again like redclassic the flags used to adjust these are `-i` to use a mouse click, or `-I` to set particular values. You can use both flags without danger, and if you use the `-i` flag the software will tell you the right `-I` flag to use to get the same integration range on later files. The three signals shown in this plot have been calculated using the algorithm described in the accompanying document “The CLASSIC/CLIMB Data Reduction: The Math”. In essence, the middle plot should only contain fringes for one baseline, while the top and bottom plots contain two baselines, but are each optimized for one of those two. If you see small amounts of power in the fringes that are not supposed to be seen it is a sign that the optical alignment was probably not very good. Picking the integration range, as you can see from Figure 18, is often a judgment call. I have found that it is sometimes necessary to use the `-i` flag on all data files and tailor each integration range to the file in question. Once the integration range has been selected the software will re-plot the power spectra, although this time normalized, although it will not look very different from the un-normalized plot.

The final step is to calculate closure phase, which will be plotted on a per scan basis as shown in Figure 19. Scans in which the three baselines do not have overlapping fringes, or the fringes are missing or edited out in one or more baselines, will not be included in

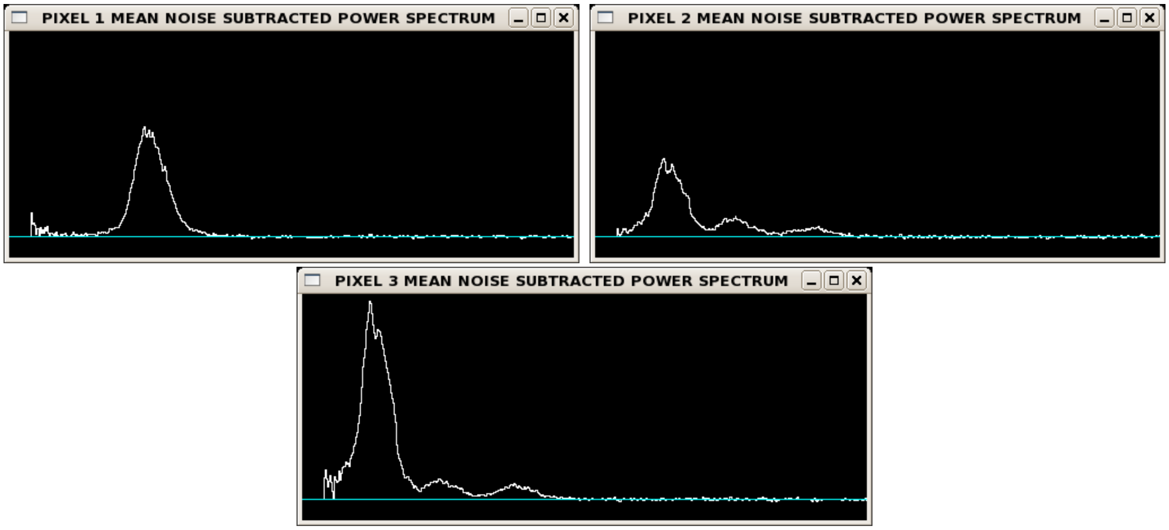

PIXEL 1 MEAN NOISE SUBTRACTED POWER SPECTRUM
PIXEL 2 MEAN NOISE SUBTRACTED POWER SPECTRUM
PIXEL 3 MEAN NOISE SUBTRACTED POWER SPECTRUM

FIGURE 16. This is the second set of plots shown by default, the noise subtracted power spectra. The light blue lines shows the zero point and all three plots use the same scale. Note and that in order to save space the arrangement of these windows has been changed.

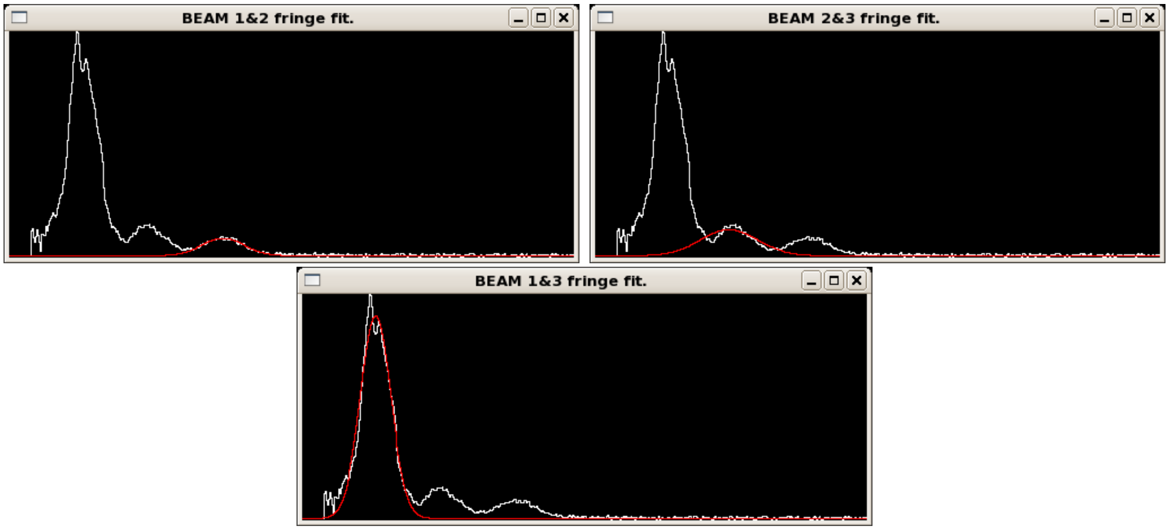


FIGURE 17. This is the third set of plots shown by default, the power spectra fitted by Gaussian. Here we see that the fits have correctly identified the correct positions of the three fringe peaks. Note and that in order to save space the arrangement of these windows has been changed.

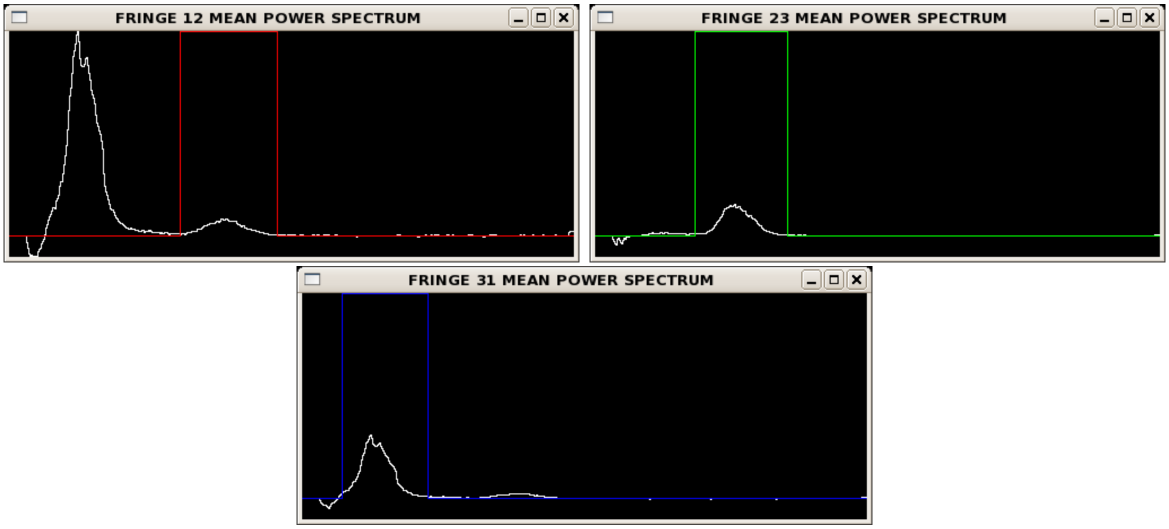

FRINGE 12 MEAN POWER SPECTRUM
FRINGE 23 MEAN POWER SPECTRUM
FRINGE 31 MEAN POWER SPECTRUM

FIGURE 18. Here we see the plot of the three combined and normalized power spectra and the range of the integrations. Note and that in order to save space the arrangement of these windows has been changed.

this plot. The final value of the closure phase will be a mean weighted by triple amplitude. If you find that too many scans are removed from this process it most likely means that the SNR is poor. You can force redclimb to include more scans by lowering the minimum fringe weight required using the `-w` flag, removing the overlap test with `-o`, or increase the amount of the fringe used for closure with either the `-g` or `-H` flags. Finally, you may see a warning that the segment size is not a multiple of 3. In those cases it is sometimes helpful to force a segment size using the `-B` flag.

4.1. Calibrating CLIMB Data

The final step is calibrating the data and creating an OIFITS file, and this is done using the `calibclimb` program, which has the following flags:

```
{di-centos:1015} calibclimb -h
usage: calibclimb [-flags] {OBJ CAL1,CAL2,CAL3...}
Flags:
-B[Vis Type]            Set Vis estimator for Object and Calibrator (V_LOGNORM)
-c                      Use CHARA number for identifier (OFF)
-C[Cal Vis Type]        Set Vis estimator for Calibrator (V_LOGNORM)
-d                      Add change in calibrator for error (ON)
-f[oiffile]             Set OIFITS filename (From object)
-F                      Toggle saving OIFITS file (OFF)
-h                      Print this message
-H                      Use HD number for identifier (OFF)
-i                      Use ID/Name for identifier (OFF)
-J[mjdmin,mjdmax]       Restrict MJD range (All)
-n                      Use standard error instead of standard deviation (ON)
-N                      Invert the sign of the closure phase (OFF)
-o                      Invert the sign of the UV coords (OFF)
```

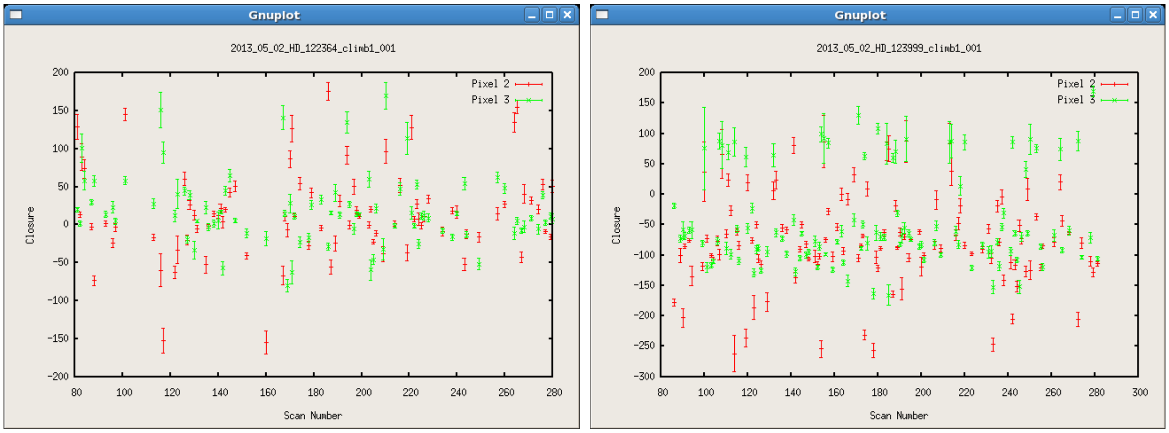


FIGURE 19. Here we see two examples of the final plots shown by redclimb, the closure phase estimates. On the left is the calibrator and on the right is the object. Each point is the result of a single scan. Apart from outliers, the calibrator closure sits near zero while the object closure is near -90 degrees.

```
-O[Obj Vis Type]        Set Vis estimator for Object (V_LOGNORM)
-p[0, 2 or 3]           Use both or only pixel 2 or 3 for closure (0-Both)
-r                      Print raw data (ON)
-s[diam1-err,diam2-err,...]     Size of calibrators in mas (0.0)
                        If error left out it is set to zero.
-v                      Verbose mode (OFF)
-V                      Print version
-2                      Output V^2 table based on V estimator (ON)

Visibility estimators available are:
V2_SCANS V_SCANS V_NORM V_LOGNORM
```

As you can see, many of these flags are the same as those in calibir, and indeed the calibration process for CLIMB data is not very different from CLASSIC data. In particular the calibration of the visibility amplitude data is pretty much exactly the same and all the information in section 3.3 holds for CLIMB data. The main difference is the addition of closure phases and triple amplitudes.

The output of calibclimb on the example data, here broken into four parts, should look something like this:

```
#  Name     HD Num CHARA# Type   MJD     T1 T2 T3
C HD_122364 122364 161870 CAL1 56414.205 S1 W2 E1
O HD_123999 123999 163170  OBJ 56414.215 S1 W2 E1
C HD_122364 122364 161870 CAL1 56414.225 S1 W2 E1
O HD_123999 123999 163170  OBJ 56414.234 S1 W2 E1
C HD_122364 122364 161870 CAL1 56414.244 S1 W2 E1
O HD_123999 123999 163170  OBJ 56414.253 S1 W2 E1
C HD_122364 122364 161870 CAL1 56414.263 S1 W2 E1
```

The second part:

```
BL12(m) BL23(m)  BL31(m) Vis12 Err12 Vis23 Err23 Vis31 Err31
202.280  205.595  329.839 0.448 0.012 0.474 0.010 0.470 0.007
205.217  205.909  328.956 0.152 0.003 0.366 0.007 0.156 0.003
202.351  213.378  330.663 0.432 0.011 0.424 0.009 0.415 0.007
205.344  213.463  330.450 0.132 0.004 0.344 0.007 0.136 0.003
202.934  218.577  330.066 0.386 0.011 0.447 0.009 0.433 0.006
205.847  218.449  330.575 0.131 0.003 0.354 0.006 0.126 0.002
203.958  221.317  328.357 0.387 0.009 0.417 0.008 0.379 0.007
```

The third part:

```
Pix2CP  ErrCP2  Pix3CP  ErrCP3 WL    dWL
  7.20    0.55    6.46    0.56 2.133 0.349
-85.43    0.49  -92.93    0.58 2.133 0.349
  0.15    0.51    3.54    0.52 2.133 0.349
-89.40    0.69  -95.61    0.66 2.133 0.349
  6.47    0.35    1.42    0.48 2.133 0.349
-85.20    0.94 -100.38    0.88 2.133 0.349
 -6.15    0.87  -13.05    0.79 2.133 0.349
```

The fourth, final, and most interesting part that contains the calibrated data

```
V12   Err   V23   Err   V31   Err      CP       Err  Cal_dT

0.346 0.016 0.814 0.093 0.354 0.044  -93.49    2.78  27.89

0.322 0.038 0.790 0.045 0.320 0.015  -95.42    2.48  27.56

0.339 0.010 0.820 0.060 0.310 0.042  -90.13    6.86  27.49
```

As we can see, the default in this case works quite well. Of course, one normally needs to include the calibrator diameter, and so the command to use here would be

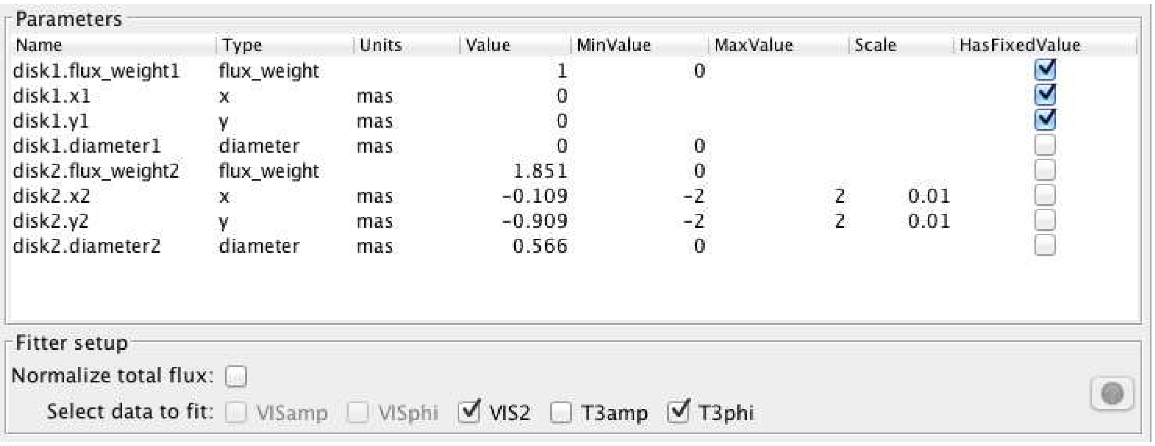
```
calibclimb -H -F -s0.495-0.010 123999 122364
```

The final OIFITS file will contain two tables, one of V^2 measures just like a CLASSIC output file, along with a table of triple amplitudes and closure phases. Note that the triple amplitude and V^2 results are redundant, and you should take this into account when fitting the data or forming images. I normally use only the V^2 data and closure phases.

Most of the flags for calibclimb are the same as those for calibir, with the addition of a one specifically used for closure phase calibration. This is the `-p` flag with which you can choose to only include the closure phase data from one of the output pixels rather than the mean of both. In some cases, when the optical alignment on the sky was not so good, one pixel or the other produces a much more reliable closure signal. This can be seen by checking the closure of the calibrator, which should always be close to zero.

For simple object likes this, I use LITPro[3] for modeling the data. Figure 20 shows the parameters I used to fit these data. The final fit produced by LITPro was

[3] See the JMMC web page www.jmmc.fr



Parameters

Name	Type	Units	Value	MinValue	MaxValue	Scale	HasFixedValue
disk1.flux_weight1	flux_weight		1	0			☑
disk1.x1	x	mas	0				☑
disk1.y1	y	mas	0				☑
disk1.diameter1	diameter	mas	0	0			□
disk2.flux_weight2	flux_weight		1.851	0			□
disk2.x2	x	mas	-0.109	-2	2	0.01	□
disk2.y2	y	mas	-0.909	-2	2	0.01	□
disk2.diameter2	diameter	mas	0.566	0			□

Fitter setup

Normalize total flux: □

Select data to fit: □ VISamp □ VISphi ☑ VIS2 □ T3amp ☑ T3phi

FIGURE 20. This is the configuration I used to fit a binary star to the example CLIMB data.

```
Final values and standard deviation for fitted parameters:
     diameter1 = 0        +/- 7.19e+05 mas (*)
     diameter2 = 0.56631  +/- 0.378    mas
  flux_weight2 = 1.8515   +/- 0.974
            x2 = -0.10854 +/- 0.0327   mas
            y2 = -0.90881 +/- 0.0642   mas
```

showing that both stars are largely unresolved, and the astrometry of this binary (12 Boo) is $\rho = 0.915 \pm 0.072$ milli-arcseconds and $\theta = 6.81 \pm + - 2.68$ degrees. Note that LITPro has made the star at the origin the secondary, that is the fainter star. This is not unusual and easy enough to reverse. It is unlikely that you will get exactly the same results, but they should agree within the error bars.

5. CONCLUSION

It is difficult, if not impossible, to cover every possibility or demonstrate the use of all the flags, but it is my hope that if you work through this document and the example data sets you will get a feel for how the data reduction process works. Should you find any bugs, have trouble making the software work, or have suggestions for improvements, please contact me.