

Guide to JCMT Receiver H3 Holography Data Analysis Software

John Richer
jsr@mrao.cam.ac.uk
Yuri Dabrowski

January 2, 2004

Contents

1	Introduction	2
2	Data Reduction User's Guide	3
2.1	Getting started	3
2.2	Environment	3
2.2.1	Initialisation file	3
2.2.2	Command history	3
2.3	Input and Output File Locations	4
2.4	Basic data reduction commands	4
2.4.1	holo-datadir	4
2.4.2	holo-showpars	5
2.4.3	holo-defaults	7
2.4.4	holo-setpar	7
2.4.5	holo-list	8
2.4.6	holo-reduce	9
2.4.7	holo-load <data-spec> <reduction-subsec>	9
2.5	Automatic customisation of parameters	10
2.5.1	The Rxh3 database file	10
2.5.2	The holo-custom procedure	10
2.6	Inspecting, Plotting and Comparing Data and Results	10

Chapter 1

Introduction

Chapter 2

Data Reduction User's Guide

2.1 Getting started

Holography data from the JCMT's RXH3 holography receiver are reduced using commands in the **TDL** scripting language. This is the Tcl Data Language written in C++ by John Richer, which provides an extension to the **TCL** scripting language which allows efficient manipulation of arbitrary size numerical arrays. The system has been developed in a UNIX (both LINUX and Solaris) environment and should port easily to other UNIX variants.

2.2 Environment

Various environment variables should be set before invoking the **TDL** executable:

- `TCL_LIBRARY` points to the location of the **TCL** run time files
- `TCLX_LIBRARY` points to the location of the run time files needed by the TclX extension to **TCL**.

Both of these variables are set if the **TDL** configuration script has been invoked. This is typically called `config/init.csh` in the **TDL** distribution.

For RXH3 data handling, the following variables should be set:

- `RXH3_DATA` should point to the directory containing the data files. If not set, the programme uses the current directory.
- `RXH3_RED` should point to the directory where the results of the reduction will be written. If not set, the programme uses the location `rxh3red`, i.e. a subdirectory of the current working directory.

Finally,

- `TDL_PRINTER` can be set to indicate the printer to be used for printing plot files.

2.2.1 Initialisation file

When **TDL** is invoked, it reads commands from the `.tdlrc` file in the user's home directory, if this file exists. This is a way for the user to customise any aspect of the **TDL** programme at run time.

2.2.2 Command history

Commands entered from the keyboard are written to the `.tdlhist` file in the user's home directory.

2.3 Input and Output File Locations

The code reduces standard RXH3 data files, whose format is defined elsewhere in this manual. The naming convention uniquely identifies when the file was taken: for example, the file

```
rxh3-20020615-234633.fits
```

Acontains an observation which started on 15 June 2002 at 233446. Not that the time and date are local times i.e. Hawaiian standard time. The 15 character identification tag 20020615-234633 can also be used in the programme to identify the file.

The programme looks for data files which match this filename pattern in the directory specified by the environment variable `RXH3_DATA`. This can be specified before running the code, but can also be changed within the programme using the `holo-datadir` command. Its default value is the current directory (`.`).

The output files from a reduction are written to a directory whose name is built from a rootname specified by the `RXH3_RED` environment variable. Again, this is usually set prior to running the programme, but can also be set using the `holo-setpar` command. Its default value is `rxh3red`. A subdirectory within this directory is created for each file reduced, using the 15-character date-time id of the file. All reductions of this file are stored in this directory tree. Finally, each individual reduction is written to a subdirectory within this directory. The default name for this is `default`, so that if necessary several different reductions of the same data can be made with different parameters, and the results stored for comparison. Continuing our example above, suppose for example we do a standard reduction of the file above, and we have set the output directory using

```
setenv RXH3_RED /home/jsr/rxh3red
```

The output files from this run will be written to

```
/home/jsr/rxh3red/20020615-234633/default
```

`noi` Subsequent reductions of the same file will overwrite the results in this directory unless the parameter `out.subdirspec` is changed before the re-reduction. e.g.

```
holo-setpar out.subdirspec nocal
```

would cause the next reduction to be written to

```
/home/jsr/rxh3red/20020615-234633/nocal
```

2.4 Basic data reduction commands

There are a small number of commands for reducing and manipulating RXH3 data. Each of these is in fact a high level **TCL** script, which in turn calls various lower-level scripts to perform its actions. Some of these commands takes UNIX-style options. These commands can be listed using the `list-comands` command:

```
tdl> list-comands -match holo-*
```

```
Compiled commands::
```

```
Tcl procedures::
```

```
holo-datadir : Set name of directory containing data files
```

```
holo-defaults: Reset all parameters to default values
```

```
holo-list : List Rxh3 data files
```

```
holo-load : Load a previous reduction into memory
```

```
holo-reduce : Reduce holography map
```

```
holo-setpar : Set Holo parameter
```

```
holo-showpars: List Rxh3 parameters
```

```
tdl>
```

2.4.1 holo-datadir

Define the location of the data files (if not set through the environment). The procedures takes a single argument which is the directory containing the data files. After setting the directory, the programme searches for data files and reports the number found. It also loads the databse file if it exists — see 2.5.1

```
tdl> holo-datadir /data/jsr/rxh3/data/  
Data directory /data/jsr/rxh3/data/ contains 384 data files  
Reading database file: /export/home/jsr/soft/tdl/src/./prms/rxh3db.dat  
tdl>
```

2.4.2 holo-showpars

Lists the numerous parameters controlling the data reduction. It takes a single optional parameter which is a wildcard specification for the parameters to be listed, and an optional flag `-brief` which generates only a list of the parameters:

```
tdl> holo-showpars  
con.cal_degree = 0  
Calibration degree (>0:poly, <0:spline, 0:no cal)  
con.cal_spline_smooth = 0.0  
Spline smoothing parameter  
con.correct_for_reference_offset = 1  
Correct for reference phase? (1/0)  
con.correct_for_sec_diff = 1  
Correct for secondary diffraction? (1/0)  
con.dft = 0  
Also calculate apertures via slow DFTs? (1/0)  
con.dft_extent = 8.0  
Half-size of maps for DFTs (metres)  
con.dmV_dMHz = 29.0  
80GHz PLL voltage gradient in mV per MHz  
con.far_field = {650 12 0}  
Far field beam estimates required  
con.far_perfect_zernikes = {0.0 4 0} {0.0 3 3}  
Far field beam: zernikes for 'perfect' aperture  
con.fdecipher_method = 1  
Method choice for frequency deciphering  
con.grid_cell = 0  
Grid cell size in arcsec (0 = auto)  
con.grid_even = 1  
Use an even number of pixels in maps? (1/0)  
con.grid_npts = 0  
Number of pixels on side of maps (0 = auto)  
con.grid_power_of_two = 1  
Make grid have power of two pixels  
con.gridder_extent = 3.0  
Extent of gridding function in units of grid cell  
con.gridder_size = 1.5  
FWHM of gridding function in units of grid cell  
con.hilo_method = 2  
Method choice for combining hi and lo channels  
con.nclip_per_cal_row = 5  
Number of points to ignore at cal scan extremes  
con.nclip_per_data_row = 5  
Number of points to ignore at data scan extremes  
con.pan.edge_factor = 0.03  
Panel fitting edge scaling factor  
con.pan.min_weight = 0.1  
Panel fitting min weight  
con.receiver_defocus = 0.0
```

```

Error in focus due to receiver misplacement (mm)
con.shadow_scale = 1.25
    Scaling factor for shadow
con.shadow_shift = 0.0
    Additional shadow in metres
con.subtract_zeropt = 1
    Subtract A/D zero point values? (1/0)
con.zernike_max_order = 7
    Max order of Zernike fit to aperture
con.zernikes_to_fit = 0 1 2 4
    List of zernike terms to fit and subtract
data.comment = (no comment)
    Comment string for this data file
data.directory = /data/jsr//rxh3/data
    Data directory
data.filename =
    Data filename
data.pointing_lag_x = 0
    Azimuth hysteresis parameter (arcsec)
data.pointing_lag_y = 0
    Elevation hysteresis parameter (arcsec)
data.pointing_offset_x = 0
    Azimuth pointing correction (arcsec)
data.pointing_offset_y = 0
    Elevation pointing correction (arcsec)
data.process_freqs = 0
    List of frequencies to reduce (or 'all')
data.secondary_defocus_offset = 2.5
    Correction to nominal defocus (mm)
data.trim_box =
    extent of box outside of which to trim data
data.trim_pixels =
    list of pixel ranges to be set bad
err.cal_max_damp = 1.02
    Max amplitude cal change before raising warning (ratio)
err.cal_max_dphi = 10.0
    Max phase cal change before raising warning (degrees)
err.tracking_max_range = 5.0
    Max tracking range before raising warning (arcsec)
err.tracking_max_rms = 2.0
    Max tracking rms before raising warning (arcsec)
geom.distance_source = 695.0
    Distance to source (metres)
geom.focal_bigf = 180.0
    Effective focal length (metres)
geom.focal_littlef = 5.4
    Primary focal length (metres)
geom.mirror_position_x = +1.850
    X offset of reference mirror (metres)
geom.mirror_position_y = +3.861
    Y offset of reference mirror (metres)
geom.mirror_position_z = -0.240
    Z offset of reference mirror (metres)
geom.primary_diameter = 15.0

```

```

    Primary mirror diameter (metres)
geom.secondary_shadow = 1.0
    Secondary mirror shadow diameter (metres)
out.aperture_as_text = 0
    Save final average aperture as text files
out.graph = 3
    Graphical output level (0=none, 5=everything)
out.loglevel = 8
    Level of detail in log messages (0-9)
out.lut = heat
    Colour look up table name
out.plot_device = gif
    Name of device for plotting (gif/psl)
out.plot_height = 767
    Height of output plots (pixels)
out.plot_width = 1023
    Width of output plots (pixels)
out.save_results = 1
    Save results to fits files
out.subdirspec = d
    Output dir ('d'->default, 'u'->unique, or name)
out.thumbnails = 1
    Attempt to display thumbnail images? (1/0)
out.topdir = /home/jsr/public_html/rxh3/red
    Top-level directory for output files
tdl>

```

Using the optional argument and parameter we can get a shorter listing:

```

out.aperture_as_text = 0
out.graph             = 3
out.loglevel          = 8
out.lut               = heat
out.plot_device       = gif
out.plot_height       = 767
out.plot_width        = 1023
out.save_results      = 1
out.subdirspec        = d
out.thumbnails        = 1
out.topdir            = /home/jsr/public_html/rxh3/red
tdl>

```

2.4.3 holo-defaults

This procedure resets all the holography parameters to their default values. In addition, it then runs the user-defined procedure `holo-custom`, if it exists, which can be set up to overrule any default values if the user so chooses.

2.4.4 holo-setpar

Set individual parameters controlling the data reduction. It takes two arguments — the parameter and its new value. There are many parameters which affect the reduction, although the default set are designed to produce good results in most circumstances. Here we change the output produced by a reduction:

```

tdl> holo-setpar out.loglevel 5
tdl> holo-setpar out.thumbnails 1

```

Here we change the frequencies which are chosen in the reduction, and ask for two far-field beam pattern estimates to be produced at 900 and 460 GHz — note the format of the list, and of the list of lists:

```
tdl> holo-setpar data.process_freqs {0 3 5 11 23}
tdl> holo-setpar con.far_field {{900 15 0} {460 10 0}}
```

2.4.5 holo-list

This command makes a listing of the data files in the directory currently set. With no arguments, it just lists the files in data-time order, and gives each a unique integer identification. With the `-full` argument, it also lists details of each file, printing out the values of various key parameters. The procedure can also be supplied with an optional wildcard parameter to restrict the listing:

```
tdl> holo-list
Data directory /data/jsr//rxh3/data contains 384 data files
Reading database file: /export/home/jsr/soft/tdl/src/./prms/rxh3db.dat
id  file
0   20000503-152752
1   20000503-160637
2   20000503-164214
3   20000503-165342
4   20000503-165928
5   20000510-155525
6   20000510-155757
7   20000510-160211
8   20000510-161528
9   20000510-161915
10  20000510-162449
11  20000510-163222
12  20000510-163736
[... listing truncated]
```

With arguments:

```
tdl> holo-list -full 20020602*
Data directory /data/jsr//rxh3/data contains 384 data files
Reading database file: /export/home/jsr/soft/tdl/src/./prms/rxh3db.dat
id  file                freq1  nf  rows npts    dz  rlen  vel  cell  sxper  tmux  tmuy  ncal
352 20020602-102205 80.32212 5   121 146600  31. 4840 400  40   10.0 29318 -5100 20
353 20020602-112531 80.32032 24  121 366500  31. 4840 400  40    4.0 29318 -5100 20
354 20020602-122052 80.32032 24  121 366500  31. 4840 400  40    4.0 29318 -5100 20
355 20020602-153034 80.32032 24  121 364400  31. 4840 400  40    4    29318 -5100 20
356 20020602-163048 80.32032 24  121 364400  31. 4840 400  40    4    29318 -5100 20
tdl>
```

In addition, the various reductions of the data that have taken place can be listed with the `-red` option being passed:

```
tdl> holo-list -full -red 20020602*
Data directory /data/jsr//rxh3/data contains 384 data files
Reading database file: /export/home/jsr/soft/tdl/src/./prms/rxh3db.dat
id  file                freq1  nf  rows npts    dz  rlen  vel  cell  sxper  tmux  tmuy  ncal  Reductions
352 20020602-102205 80.32212 5   121 146600  31. 4840 400  40   10.0 29318 -5100 20  default
353 20020602-112531 80.32032 24  121 366500  31. 4840 400  40    4.0 29318 -5100 20  default
354 20020602-122052 80.32032 24  121 366500  31. 4840 400  40    4.0 29318 -5100 20  default all_freqs
355 20020602-153034 80.32032 24  121 364400  31. 4840 400  40    4    29318 -5100 20  default
356 20020602-163048 80.32032 24  121 364400  31. 4840 400  40    4    29318 -5100 20  default
tdl>
```

This shows that all the listed files have been reduced once, and that the file observed at 12052 has been reduced twice.

2.4.6 holo-reduce

This is the main command to read in RXH3 data and reduce a holography map, and is typically used once the data reduction directory and reduction parameters have been set. It takes a single parameter which specifies the file to be reduced: this may be either (a) the full file name, or (b) a wildcard expression specifying (uniquely) the date-time of the observation, or (c) the integer index of the data file as produced by `holo-list`. For example, after inspecting the data files:

```
tdl> holo-list -full 20020615-234*
Data directory /data/jsr/rxh3/data contains 384 data files
Reading database file: /export/home/jsr/soft/tdl/src/./prms/rxh3db.dat
id  file                freq1   nf   rows npts    dz  rlen  vel   cell  sxper  tmux  tmuy  ncal
380 20020615-234239 160.7   1   1    300   31.  600  200   0    10   29318 -5100 0
381 20020615-234302 160.7   1   1    300   31.  600  200   0    10   29318 -5100 0
382 20020615-234326 160.7   1   1    300   31.  600  200   0    10   29318 -5100 0
383 20020615-234633 80.322  6   241 289200 31.  4820 400   20   10.0  29318 -5100 0
```

the following commands are equivalent:

```
tdl> holo-reduce /data/jsr/rxh3/data/rxh3-20020615-234633.fits
tdl> holo-reduce rxh3-20020615-234633.fits
tdl> holo-reduce 20020615-2346*
tdl> holo-reduce 383
```

```
holo-reduce -readonly -scale
```

If `-readonly` is specified, the raw data are read into the `holo_data` variable but no reduction is done. If `-scale` is specified as well as `-readonly`, the data are read in and scaled somewhat — angles are converted to arcsec and times to seconds elapsed from the map start. These scalings make data plotting somewhat easier.

```
holo-reduce -prompt <prompt-level>
```

If the prompt option is given a non-zero value then the procedure will prompt for the reduction parameters, overriding any set earlier. A prompt value of 1 prompts for most of the critical parameters, a value of 2 or 3 also prompts for basic parameters including telescope geometry and error settings — these should rarely need changing.

2.4.7 holo-load <data-spec> <reduction-subsec>

This command loads into memory the results of a previous reduction which has been stored to disk. At the end of a reduction, the results are left resident in memory in the **TCL** array names `holo`, but this routine is useful for loading a previous reduction and making quantitative comparisons. See 2.6 for further details.

The first parameter is the data specification, and is the same as that used in `holo-reduce`. The second parameter is optional and specifies which of the reductions of this dataset to load, if there is more than one. If omitted, the reduction named `default` is loaded.

The option `-name <name>` can be used to load the reduction into a specific **TCL** array name. If omitted, the routine generates its own name (which is of the form `holo383`).

Here we load the default reduction of filespec 354 into a **TCL** array named `fone`:

```
tdl> holo-load 354 -name fone
Data directory /data/jsr/rxh3/data contains 384 data files
Reading database file: /export/home/jsr/soft/tdl/src/./prms/rxh3db.dat
Resolved file ID 354 = rxh3-20020602-122052.fits
Loading reduction 20020602-122052 (default) from
/home/jsr/public_html/rxh3/red/20020602-122052/default into array ::fone
Script = /home/jsr/public_html/rxh3/red/20020602-122052/default/load.tcl
tdl>
```

Here we load the reduction called `all_freqs` of filespec 354 into a **TCL** array whose name is generated automatically as `::holo354`:

```
tdl> holo-load 354 all_freqs
Data directory /data/jsr/rxh3/data contains 384 data files
Reading database file: /export/home/jsr/soft/tdl/src/./prms/rxh3db.dat
Resolved file ID 354 = rxh3-20020602-122052.fits
Loading reduction 20020602-122052 (all_freqs) from
```

```
    /home/jsr/public_html/rxh3/red/20020602-122052/all_freqs into array ::holo354
    Script = /home/jsr/public_html/rxh3/red/20020602-122052/all_freqs/load.tcl
tdl>
```

Note that the `::` in front of the variable name simply means it is a global variable — this is standard **TCL** terminology.

2.5 Automatic customisation of parameters

2.5.1 The Rxh3 database file

2.5.2 The `holo-custom` procedure

2.6 Inspecting, Plotting and Comparing Data and Results