MOPSIC – An Extended Version of MOPSI

Version 1.0

Markus Nielbock

June 14, 2004

Contents

Chapter 1

Introduction

After the success of MOPSI, its developer Robert Zylka started to merge this software with the command interpreter of GILDAS into the successor now called MOPSIC. This simplified its use, but made the change of part of its syntax unavoidable. Furthermore, the skynoise reduction has been improved considerably being now more sensitive to faint extended emission that previously had escaped detection. In the following, I will explain the changes necessary to convert old MOPSI procedures into MOPSIC style. Additionally, I will mention important changes to the philosophy of the data reduction.

I assume, the reader is already familiar with the usage of MOPSI. A comprehensive summary can be found in the old SEST/SIMBA user manual.

Chapter 2

General changes

2.1 GILDAS commands

MOPSIC can access the command set of the GILDAS software by indicating the package where the command resides (see example in Sect. 2.2).

2.2 Abbreviations

All commands need at least the first three letters in order be recognised by the command interpreter. Sometimes, even this is not enough to avoid confusion with other commands. MOPSIC will announce this with an error message.

```
MOPSIC> ext
E-INTER,  Ambiguous command could be :
GREG2\EXTREMA       MOPSIC\EXTINCT      MOPSIC\EXTRACT      MOPSIC\EXTREMA
IMAGE\EXTENT
```

2.3 Scrolling

It is now possible to scroll the command history up and down using the cursor keys. Instead, the command `reexecute` has disappeared.

2.4 File names

The extension for scripts to be recognised by MOPSIC is changed to `.mopsic`. All the other file extensions are the same as in MOPSI.

2.5 Help facility

The GILDAS help is initiated with `help`, while the MOPSIC help function is accessed with `?`.

Chapter 3

Default settings

After reducing and combining data from many observers during the SIMBA sessions from 2001 until 2003, we are able to provide a database containing calibrations and opacities for almost the whole lifetime of SIMBA. There is also a set of bad channels that should be deleted for various observing periods. They reside in the standard directory where MOPSI and MOPSIC have been installed and can all be accessed via options of the different commands as described in the following sections.

3.1 Bad channels

The file containing the bad channels of SIMBA is called `SEST_SIMBA.DRC`. Every line consists of the keyword `JD`, the Julian date until the entry is valid, followed by the numbers of channels to be deleted. A typical section of such a file is:

```
JD 2452425.00 38 39 40
JD 2452499.06 2 12
JD 2452500.00 2 14
JD 2452500.03 2 12 14
JD 2452500.08 2 12 14 32
```

The command to erase the bad channels using this file is:

```
del rc default
```

I found out that sometimes it might be necessary to repeat this command in order to function.

3.2 Calibration

We compiled calibration factors in a consistent manner using Uranus calibration maps for the whole period of SIMBA. If the user wishes to access them, it can be done with:

```
calibrate default
calibrate
```

The file containing all calibration values is called `SEST_SIMBA.CAL`. Every line consists of the Julian date until the calibration is valid followed by the calibration factor in mJy/count. A typical section of the file is:

```
2452079.5516 86
2452079.5581 86
2452079.5646 86
2452083.4547 86
```

3.3 Opacity correction

We created an almost complete database of opacity values determined from skydip observations. It can be used to automatically correct for the atmospheric extinction. The referring file is called SEST_SIMBA.TAUS. Every line lists the scan number of the skynoise measurement, the UT date of the observation, its Julian date and the zenith opacity value. Here is a section from this file:

```
0945 16/06/01 2452076.69792 63609 0.096
0973 16/06/01 2452076.84722 76544 0.103
1003 16/06/01 2452077.40278 38276 0.119
1029 17/06/01 2452077.53681 49888 0.112
1043 17/06/01 2452077.60069 55423 0.141
1047 17/06/01 2452077.65972 60537 0.116
```

It can be used with:

```
tau def 20000
correct ext
```

Chapter 4

From MOPSI to MOPSIC

4.1 File lists

File lists have to be read in with the command `in-list <list>`. The `list` command can only be used to display the current file list. Consequently, it must be initialised with `init in-list`. The command `replace` to replace the output list with the input list must be written in a full word.

```
init in-list
in-list fits.LIST
list
...
replace in-l out-l
```

4.2 Buffers

Only one standard buffer is available at the start of the programme. It is the backup buffer. As usual, data can be stored with the command `store`. However, to recover the content of the backup buffer one has to use the command `take`; `old` is not available anymore.

One can define new buffers in order to store data. For this, it must have the suitable dimensions for the data set to fit in. This can be achieved with the command `define` (see help pages `? define` and `help define`). The syntax is: `define <type> <name> <size>` A typical example is `define double buffer like data`. This creates a new data buffer of variable type "double" called "buffer" with the same dimension as the current data. The data in the working memory can now be put there with `store buffer`. It can be retrieved again with `take buffer`. After its usage, the buffer must be erased with `del var buffer`. Example:

```
def double mem1 like data
store mem1
...
take mem1
del var mem1
```

4.3 FOR-Loops

Both the variable for the last entry in a file list and the running index have changed. Instead of "last" one has to write "nobsin". Hence, the loop definition is written as `for i 1 to nobsin`. When indicating a file to be read from the previously loaded file list, the running index has to be put in quotes like `read 'i'`. The typical header of a loop should read something like:

```
for i 1 to nobsin
read 'i'
```

4.4 Skynoise filtering

The section in reduction scripts defining the skynoise reduction was previously called with `snf` (Sky Noise Filtering). This command is still valid, but it should be replaced by the synonymous command `csf` (Correlated Skynoise Filtering) and is indicated as such with the prompt. Most commands should now be explicitly written in full words to avoid possible ambiguities. The command `no` for switching off actions during the skynoise reduction has been replaced with the command `disable` because of discrepancies with the GILDAS command set. Possible example:

```
csf iter 6 0
!> 5 rms not_in_data < -5 rms not_in_data
disable > disable <
range 1 900 best 6
!cn av
cn med
disable source
!source model.gdf
!sb
run
```

4.5 Combine and Convert

The subroutines of `combine` and `convert` do no longer use the command `no`, either. It has also been replaced with `disable`. Furthermore, most commands have to be written in full words, too. The command `quit` was replaced with `exit`. Example:

```
combine
extent 600 600
disable weight
...
exit
```

4.6 Integration

The `integrate` command is no longer provided in its interactive form. All integration parameters must now be listed along with the command in one line. Example:

```
integrate factor 1 b lev 0 1e10
```

4.7 Corrections

The command `correct` used in the gain-elevation and opacity correction (see Sect. 4.10) must be written in a full word. Example:

```
correct ge
```

4.8 Bad channels

There is a default list of bad channels (see Sect. 3.1). The user can also delete channels manually as usual with the command `del rc <number>` or use his own bad channel file after inspecting the data set. Its syntax is as described in Sect. 3.1 and can be accessed via `del rc <file>`. Examples:

```
del rc 2 12
del rc bad.drc
```

4.9 Calibration

Calibration of the data can be achieved either with the default set of calibration values (see Sect. 3.2) or using your own calibration. This can be done as in MOPSI just by multiplying the data with the calibration factor in mJy/count or providing a calibration file similar to the default calibration. The syntax is the same as in Sect. 3.2. The calibration is done with `calibrate <file>`. Example:

```
calibrate factors.cal
calibrate
```

4.10 Opacities

We suggest to use the standard opacity correction as described in Sect. 3.3. Apart from that, the correction for atmospheric extinction has not changed from the procedure in MOPSI using an own opacity file, usually called `tau.TAUS`. Example:

```
tau tau 2000
correct ext
```

Chapter 5

SIMBA data reduction

5.1 Baseline fitting

We found that one should not apply any baseline fitting per subscan (e.g. `base l 0`) to the data before reducing the skynoise with CSF or SNF. Instead, it is suggested only to use the correction on the whole data set (e.g. `base t 4`). However, if possible, the fitting should only be conducted on an area excluded with a previously defined polygon to avoid the influence of any real positive emission. A fit per subscan can be used after the skynoise reduction sequence. Example:

```
pol base.pol
base r out sys eq
base t 4
...
base t 4
store weight rms2 out sys eq
csf
...
base l 3
store weight rms2 out sys eq
```

5.2 Example MOPSIC reduction script

```
init gfit
init histo
init spike

sys "!rm *.btd"
init in-list
init out-l

in-lis fits
sort in-list
select ecar

for i 1 to nobsin
read 'i'
del dc
del rc default
del rc default
```

```
init spike

pol ecar_1
base r out sys eq
base t 5
!set mask
!> 5 rms
!< -5 rms
!interpol blank
!mask

!base t 5

set mask
deconv
mask

correct ge
tau default 20000
correct ext

calibrate def
calibrate

mask spike spike_buffer
base t 5
!store weight rms2 az < 1e9 > -1e9
store weight rms2 out sys eq

csf iter 6 0
! > 5 rms not_in_data < -5 rms not_in_data
disable > disable <
rang 1 900
best 6
cn av
!cn med
!disable source
source model.gdf
sb
run

base l 2
!store weight rms2 az < 1e9 > -1e9
store weight rms2 out sys eq

wr .btd
next

init spike
replace in-l out-l

read 1
convert command
extent 1200 1200
```

```
pix 4 4
run all
exit

file out ecar.gdf
write
close out
```