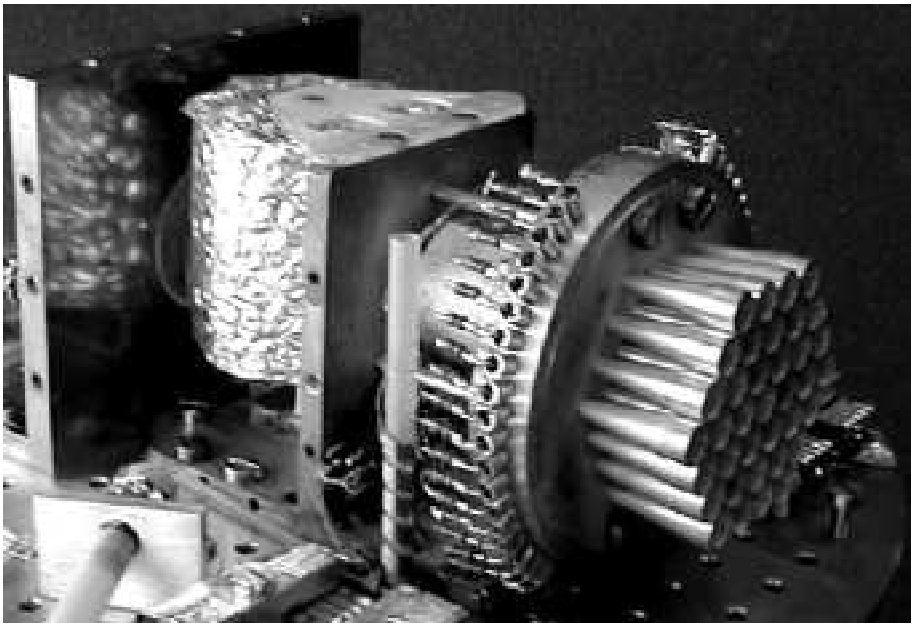
SIMBA

SEST Imaging Bolometer Array

An Observer's Handbook



Version 1.8

18th August 2002

Contents

1 Observing with SIMBA **1**
1.1 Observing Time Estimate . . . 1
1.1.1 Example . . . 2
1.2 Observation Planning – Map Layout . . . 3
1.2.1 Map Orientation with a Template . . . 3
1.2.2 Map Orientation without a Template . . . 6
1.3 General Observation Procedure . . . 7
1.4 Pointing . . . 11
1.5 Focus . . . 13
1.6 Skydip . . . 15
1.7 On–Off . . . 17
1.8 Fast Scanning . . . 18
1.9 The SIMBA FITS Format . . . 20
1.9.1 Data Archiving . . . 21

2 Data Reduction and Analysis **23**
2.1 State of Development . . . 23
2.2 General Concept . . . 23
2.3 Simbaread – The FITS file converter . . . 24
2.4 NIC . . . 25
2.4.1 Introduction . . . 25
2.4.2 Workspace . . . 25
2.4.3 Getting Started . . . 26
2.4.4 Initialisation . . . 27
2.4.5 Reading Data Files . . . 27
2.4.6 Data Reduction . . . 28
2.4.7 Visualisation . . . 30
2.4.8 Combining Maps . . . 31
2.4.9 Saving Data . . . 32
2.4.10 The quick.nic Routine . . . 32
2.5 MOPSI . . . 33
2.5.1 Preparing the Opacity File . . . 33
2.5.2 Getting Started . . . 33

2.5.3 Reading Data Files 34
2.5.4 On-Line Reduction 35
2.5.5 On-Line Reduction – Step by Step 38
2.5.6 Command Hierarchy 43
2.5.7 Command List 45
2.5.8 Data Display 46
2.5.9 Saving Data 50
2.5.10 Sophisticated Data Reduction 51
2.5.11 The Weak Compact Source 51
2.5.12 The Moderately Strong Compact Source 54
2.5.13 The Strong Compact Source 54
2.5.14 The Weak Extended Source 55
2.5.15 Combining Maps – Single Position 55
2.5.16 Combining Maps – Mosaic 56
2.6 Calibration 57
2.6.1 The Flux Density of the Calibrator 57
2.6.2 The System Intensity (Counts) of the Calibrator 58
2.6.3 Errorbeam Correction 59
2.7 Photometry 59

Chapter 1

Observing with SIMBA

The bolometer array is operated in different electronic configurations depending on the measuring mode. They can be distinguished according to the biasing and coupling of the bolometer channels, namely AC and DC. SIMBA observations can in principle be separated into calibration modes (pointing, focus, skydip) and scientific modes (on–off, mapping, fast scanning). The standard mapping mode is not supported, though. The fast scanning mode should always be chosen instead, because this is the only configuration, where all 37 channels are available. As the latter mode is used for creating maps, it will henceforth also be called mapping. The data is written into a fits file at the end of every observation and automatically transferred to the data reduction computer *Dorado*. The labelling of all SIMBA fits files begins with a B, followed by 5 to 6 digits and a small letter describing the type of observation (**p**ointing, **f**ocus, **s**kydip, **o**n–off, **m**apping).

1.1 Observing Time Estimate

The observing time needed depends on the aimed noise level σ and therefore on the atmospheric conditions during the observation run. The residual noise for one map can vary between $\sigma_{\rm map} = 30$ and 80 mJy/beam, while a value of 40 mJy/beam should be regarded as typical for reasonably good conditions during winter observing periods (June/July). The necessary number of maps $n_{\rm map}$ can be calculated according to:

$$n_{\rm map} = \left(\frac{\sigma_{\rm map}}{\sigma}\right)^2 \tag{1.1}$$

The total observation time $t_{\rm tot}$ to achieve σ is then:

$$t_{\rm tot} = n_{\rm map} \cdot t_{\rm map} + t_{\rm setup} \tag{1.2}$$

Consider $t_{\rm setup} = 2$ min, which is the time needed to send the observer's commands to the TCS (Telescope Control System) and the bolometer system. The observing time for one single map $t_{\rm map}$ can be calculated as follows.

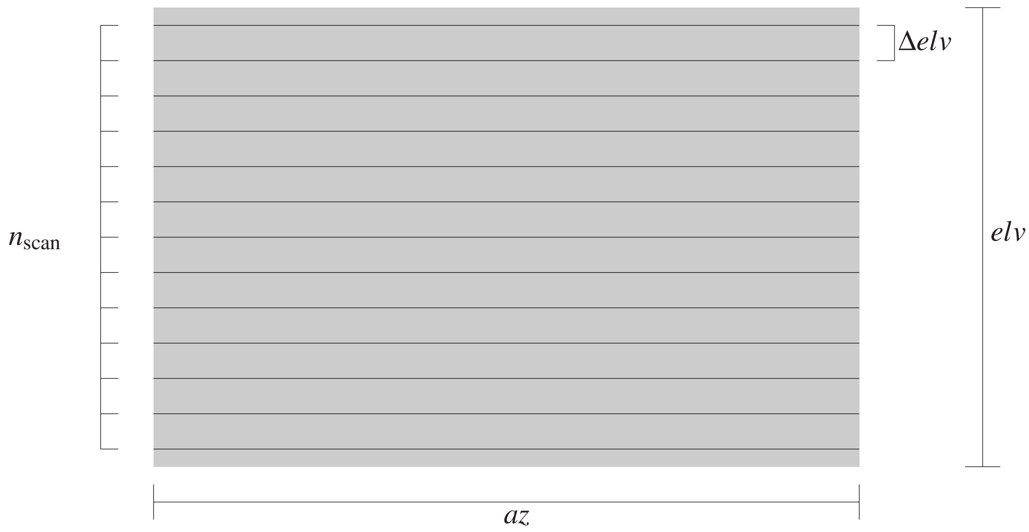


Figure 1.1 Schematic of a typical SIMBA map. The different parameters shown here determine the time needed to complete it.

A given map of dimensions $az \times elv$ taken with a scanning speed of v_{az} and a scan separation in elevation of Δelv consists of $n_{\text{scan}} = \frac{elv}{\Delta elv} + 1$ scans. However, n_{scan} should preferably be an odd number, in order to assure that the full beam always hits the map centre. The time for one single scan is $t_{\text{scan}} = \frac{az}{v_{az}}$. Considering a time for turning the telescope at the end of every scan $t_{\text{turn}} = 2$ s, the time needed for one map is:

$$t_{\text{map}} = (t_{\text{scan}} + t_{\text{turn}})\, n_{\text{scan}} \tag{1.3}$$

Or substituting the dependent variables:

$$t_{\text{map}} = \left(\frac{az}{v_{az}} + t_{\text{turn}}\right)\left(\frac{elv}{\Delta elv} + 1\right) \tag{1.4}$$

Our default values for the scanning speed and the scan separation are $v_{az} = 80''/s$ and $\Delta elv = 8''$, respectively. We recommend a minimum map-size, which can be calculated from the source extent (az_{src}, elv_{src}) and the size of the bolometer array ($az_{\text{bol}} \approx elv_{\text{bol}} \approx 4'$) according to:

$$az_{\text{min}} = az_{\text{src}} + 2\, az_{\text{bol}} \tag{1.5}$$
$$elv_{\text{min}} = elv_{\text{src}} + elv_{\text{bol}} \tag{1.6}$$

For point-like sources, this computes to a minimum map-size of $480'' \times 240''$.

1.1.1 Example

We want to observe a dust cloud with a diameter of $5'$ at a 3σ detection limit of 20 mJy/beam, hence a 1σ r.m.s. noise of about 7 mJy/beam. We expect a noise level of 50 mJy/beam for a single map.

The map-size is

$$az \times elv = 300'' + 2 \cdot 240'' \times 300'' + 240'' = 780'' \times 540''.$$

The minimum number of scans is

$$\frac{elv}{\Delta elv} + 1 = \frac{540''}{8''} + 1 = 68.5.$$

n_{scan} should therefore be 69. The time for a single scan is

$$t_{\text{scan}} = \frac{az}{v_{az}} = \frac{780''}{80''} \approx 12 \text{ s}.$$

The time for a complete map is then:

$$t_{\text{map}} = (t_{\text{scan}} + t_{\text{turn}})\, n_{\text{scan}} = (12 \text{ s} + 2 \text{ s}) \cdot 69 \approx 14 \text{ min}$$

In order to reach the desired noise level, we need 50 maps:

$$n_{\text{map}} = \left(\frac{\sigma_{\text{map}}}{\sigma}\right)^2 = \left(\frac{50 \text{ mJy/beam}}{7 \text{ mJy/beam}}\right)^2 = 51$$

The total observing time computes to:

$$t_{\text{tot}} = n_{\text{map}} \cdot t_{\text{map}} + t_{\text{setup}} = 51 \cdot 14 \text{ min} + 2 \text{ min} \approx 12 \text{ h}$$

One should consider additional time for pointing and focusing (~ 15 min each) as well as for skydips (~ 10 min). Pointing and focusing should be repeated always after a change from day to night time. Skydips should be repeated at least every 3 hours; more often during variable and bad weather conditions.

1.2 Observation Planning – Map Layout

As the telescope is set up in horizontal coordinates, the scan direction on the sky changes with the hour angle. This might at first sight complicate the observations, but in fact offers several advantages. Firstly, the spatial sampling can be optimised when co-adding single maps taken at different hour angles. This is very important, as even the hexagonal array design creates gaps in the spatial coverage of the observed area. Secondly, the fast scanning mapping mode may cause strip-like artefacts in the final map, especially at mediocre and variable weather conditions. Co-adding single maps taken at different hour angles can suppress this behaviour efficiently.

Therefore, it is desirable to know the map orientation on the sky when observing, in order to properly plan the measuring campaign. A suitable tool is provided within MOPSI, our preferred reduction software. It is therefore a good idea to get used with its basics first by studying Sect. 2.5.

1.2.1 Map Orientation with a Template

The orientation of the map can be calculated according to a template data file, which can be anything like FITS, GDF, etc. A typical case would be an optical image of a galaxy or a cloud or millimetre data from previous observations. The new *Digitised Sky Survey DSS2* can for instance be accessed via `http://archive.eso.org/dss/dss`.

The observation planning is carried out within a sub-programme of MOPSI called with the command `ae_obs` (azimuth–elevation observation). It possesses its own small set of commands, which can be listed by typing `help` within this new environment.

```
SHOWAEO> help
SHOW_Az_El_OBS calculates the position of an Az-El map or on-off in EQUATORIAL
system.
The plot of individual scans (rows) of planned map (on-off positions) can be
overlayed on intensity distribution map
SHOWAEO commands: (version 03.12.2000, R.W. Zylka)
@ COMMAND_FILE_NAME
beam_sep [VALUE] (in arc sec)
continue (with the COMMAND_FILE after PAUSE)
date [DD MMM YYYY  HH MM SS  or  now], now is default
extent [Az-EXTEND [El-EXTEND]] (in arc sec)
greg [any GreG command]
no array_plot
mode auto|bound|scan
offsets [X-OFF Y-OFF [/system NAME]] (in arc sec), equator. offsets are default
overhead_time [VALUE] (in seconds)
pause (action only with commandfiles)
pixels [Az-INC [El-INC]] (in arc sec)
plot array
projection [HH MM SS DD AM AS [MAP_NAME]], horizontal map is default
rc #|extremal|all: defines receiver channels to plot
rcp file_name: defines receiver channel parameters
quit
run: calculates time, elevation and ... of map to be observed
show: shows the HORIZONTAL map in EQUATORIAL system
sic [any SIC command]
skew [VALUE] (in arc sec)
st_start [H M S]
telescope [NAME]
velocity [VALUE] (in arcsec/sec)
```

However, only a small sample of these commands are actually necessary for our purposes. A typical routine to display a desired map could be the following:

```
!Mandatory buffer initialisations
init lis
init out-l
init gfit
init histo
init spike

!Load and display the template
cl
read dss2_ngc300.fits
plot scal log 8400 20000

!Call the sub-programme
ae_obs
```

```
 !Define pen behaviour for map overlay
 sic pen /dash 3 /col 0

 !Define mapping characteristics
 tel sest
 velo 80
 pix 8
 mode scan
 over 2
 beam 0

 !Define observation
 ext 1400 1400
 st 21

 !Display the map
 show

 !Reset pan behaviour to default
 sic pen /dash 1 /col 0

!Leave the sub-programme
quit
```

This routine can be saved into a batch file, e.g. called **ae.MOPSI**, and `@ae` carries it out. This routine is now explained in detail:

1. We need to initialise (`init`) a couple of memory buffers that might interfere with the calculations. This should always be done when calling a routine.
2. The template file is loaded and displayed. In this example, we have downloaded a FITS file from the *DSS2* database of the galaxy NGC 300 with R.A. $0^h54^m54^s$ and Decl. $-37°41'00''$, which is loaded with the `read` command. `cl` clears the graphical window, and the `plot` command displays it with the given scaling.
3. Enter the `ae_obs` sub-programme.
4. It is a good idea to redefine the plot behaviour in order to increase the visibility of the overlayed map schematic on top of the template image. This can be done with the GreG command `pen`. Dots instead of lines (`/dash 3`) and a suitable colour (`/col`) for the chosen colour table can be selected. See the GreG help function (`sic help pen`) for a more detailed explanation.
5. The mapping parameters must be defined. These are the telescope (`tel sest`), the scanning velocity in $''$/s (`velo 80`), the subscan offset in $''$, which is assumed to be identical to the pixel size (`pix 8`), the observing mode (`mode scan`), overheads in seconds (`over 2`) and the beam separation in $''$, which is 0 for SIMBA (`beam 0`).
6. The observation itself must be defined. At least the map size in $''$ (`ext`) and sidereal time of the start of the observation (`st`) have to be specified. MOPSI will then assume the observation to be

carried out on the image centre. If, however, the map shall be positioned somewhere else, its coordinates can be indicated with the `proj` command (e.g. `proj 0:55:26.8 -37 31 25 equ`). Do not forget the option `equ` for the equatorial coordinate system, as the default is horizontal.

7. `show` displays the map orientation as shown in Fig. 1.2, and gives some further information about the planned map in the text output.

```
SHOWAEO> Projection centre of the AZ-EL map for epoch 2002.59413
         (R.A.,Dec.) = ( 0 55  0.8650, -37 40  0.192)
SHOWAEO> This map will take about   57 minutes
         L.S.T. start: 21  0  0.0 (21.00000 hours)
         L.S.T. end:   21 57 27.5 (21.95764 hours)
         ELstart  40.94 ELend  52.65 ELmax  52.65 deg
```

   Several maps can be plotted on top of this by adding the observing parameters (`ext`, `st`, `proj`) for them together with the `show` command.

8. Restore the default setting of the plotting behaviour.

9. Leave the map planning environment with `quit`.

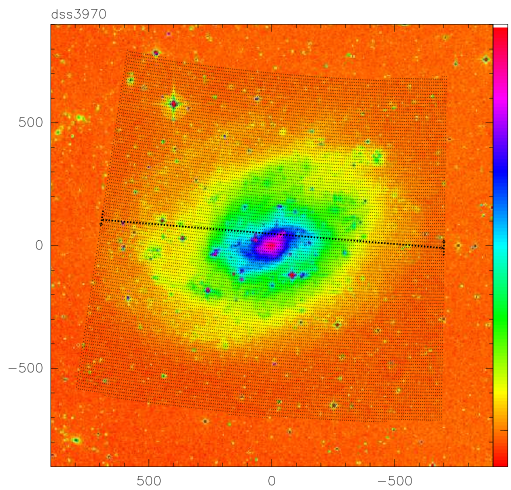


Figure 1.2 Example of a good placement of a SIMBA map. The background image is NGC 300 taken from the DSS2 online database.

1.2.2 Map Orientation without a Template

If a data template is not available, the underlying grid must be created first. This can be achieved by using the following commands within MOPSI.

```
sic lim <x-left> <x-right> <y-down> <y-up>
sic set box match
set angle sec
set system equ
```

```
set epoch 2000
proj <hh:mm:ss.s> <dd:mm:ss.s>
box
```

The grid boundaries are given in arcseconds, e.g. `sic lim 900 -900 -900 900` for a centred $1800'' \times 1800''$ area. The epoch can also be any other than 2000. The command `proj` defines the equatorial coordinates at the position 0,0 of the map, e.g. `proj 0:54:53.6 -37:41:00`. Then proceed with `ae_obs` as in Sect. 1.2.1.

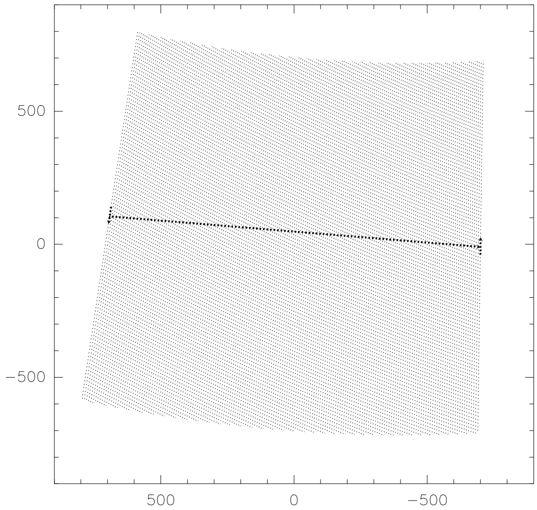


Figure 1.3 Example of a map without a template.

1.3 General Observation Procedure

The setup for bolometer observations is quite simple. All you have to do is to open the TCS display from the *Pegasus* GUI (see Fig. 1.4) and start the data synchroniser on Musca. Login to Musca for this and type **startSIMBA**. Only a few buttons are used during bolometer observations, which are all carried out via the SIMBA user interface. They can be accessed with the SIMBA button.

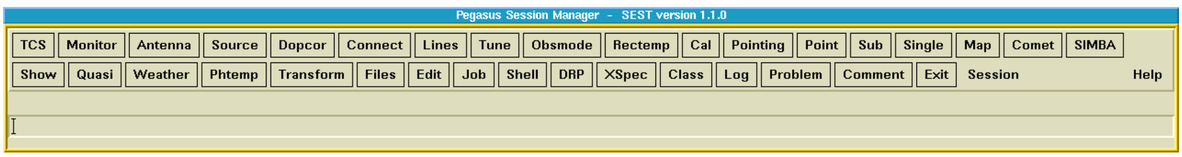


Figure 1.4 Topmost and main Graphical Use Interface (GUI) menu.

The SIMBA control window, which is typically left open throughout the observing session, appears (see Fig. 1.5) after clicking this button. The Connect button should be clicked at the beginning of every observation session. This sets the focus and pointing defaults for the bolometer. Likewise, Disconnect resets these values to zero. If desired, Transfer allows to copy the FITS files to Dorado, but by default this is done automatically anyway. Show opens a new window (see Fig, 1.6), which offers the opportunity to have a quick look at the data by accessing *simbaread* (see Sect. 2.3).

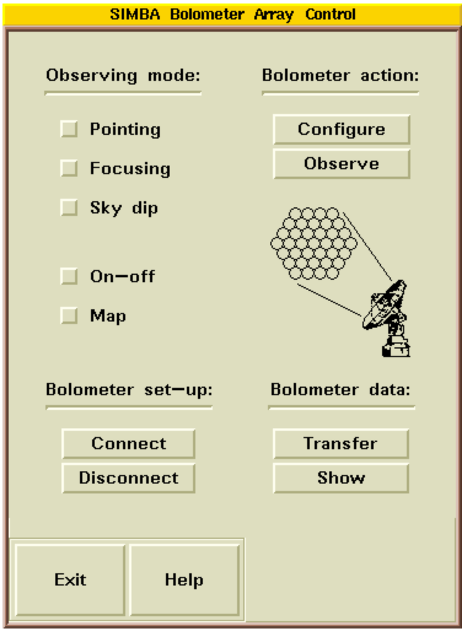


Figure 1.5 SIMBA user interface. All bolometer observations are carried out from here.

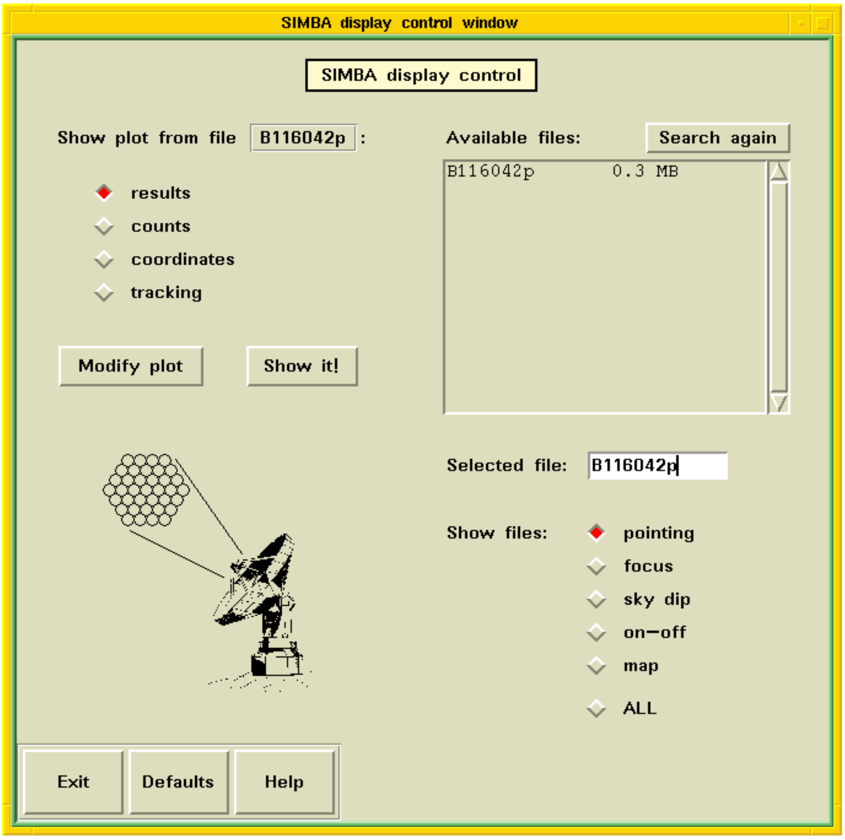


Figure 1.6 This window allows to have quick look at the data by calling the simbaread routine.

The new window gives a sorted list of all available data files. It is possible to restrict the listing to only one type of observation. The most recent observation is selected by default. The observer has four options of visualising the data. Selecting *Results* gives a brief and simple data analysis. *Counts* displays the time series of the collected data while *Coordinates* and *Tracking* show the coordinates and the tracking errors of the telescope. Show it! issues the display request while Modify plot gives the opportunity to alter the settings for the graphical display. The possibilities vary between the different observing modes. They mostly comprise choices for the displayed channels and whether the data should be shown with or without the skynoise reduction (see below). Consult the individual sections of the different observations modes for typical examples.

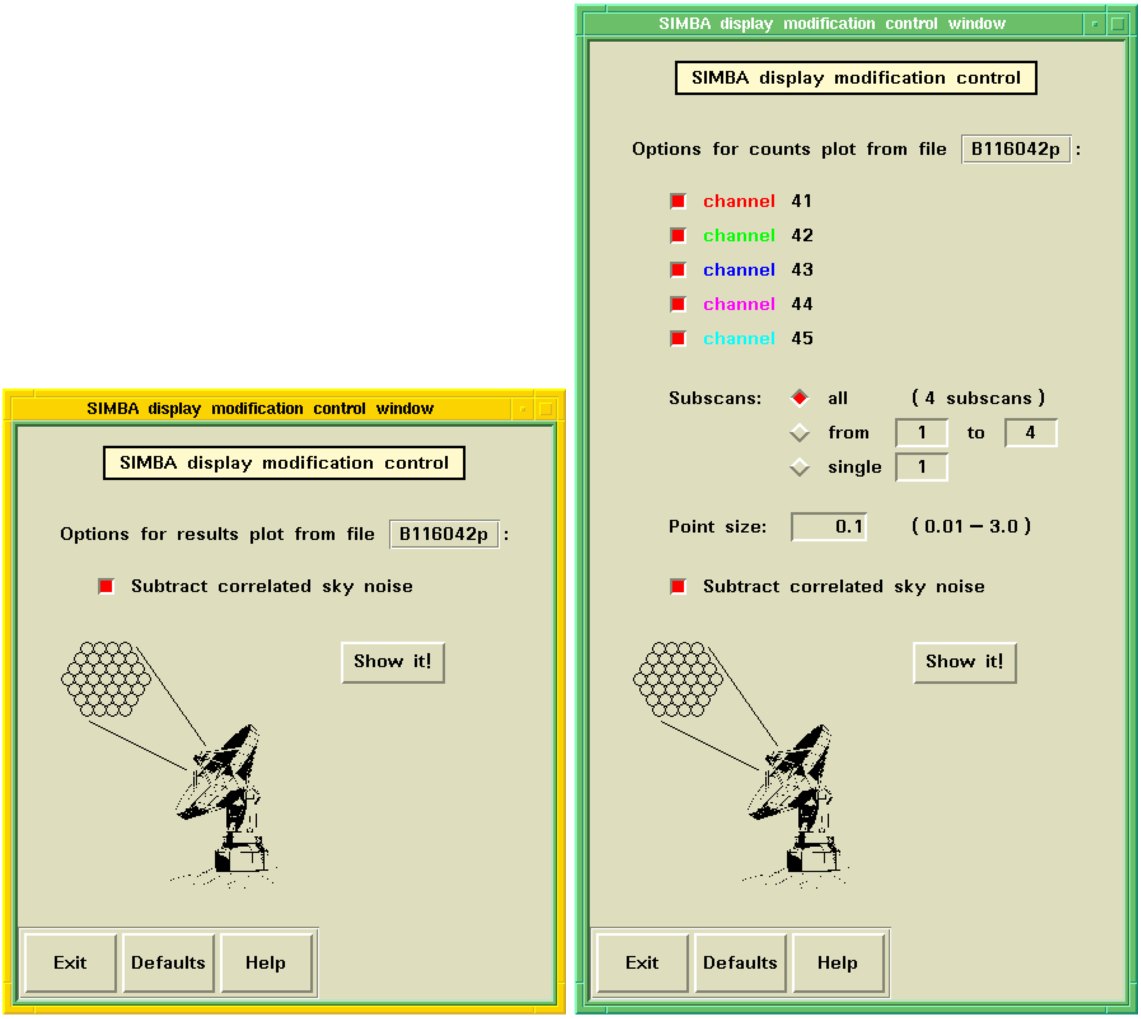


Figure 1.7 Two examples for the *Modify plot* option in the case of pointing measurements. The observer can choose the displayed channels and whether the data should be displayed with or without the skynoise reduction.

During the commissioning phase of SIMBA, we figured out that despite the various possibilities only minor configurations have to be done for the different observation modes. At this stage, the correct settings are sent automatically to the backend when starting a new measurement. However, the observer can also configure manually with the interface (see Fig. 1.8), which is displayed by clicking Configure.

Note that the Configure window shows you the biasing mode the Pegasus systems thinks that SIMBA is in, and not the recommended mode for a certain type of observation.

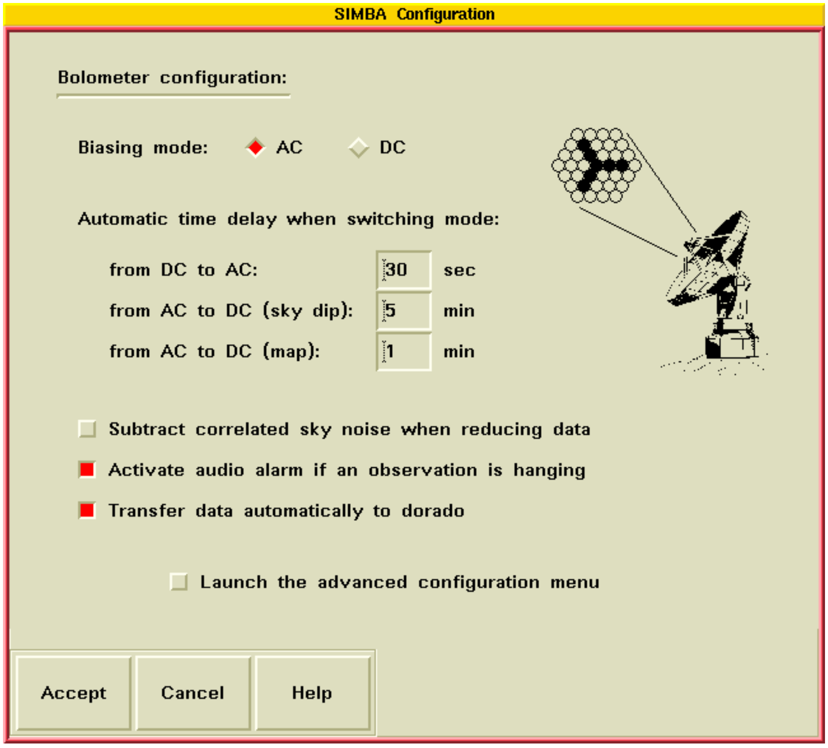


Figure 1.8 Simple configuration panel. The main purpose is to select the correct biasing mode (AC/DC) for the following observation.

Apart from the selection of the proper biasing mode (AC/DC), the observer might specify further settings: The electronics need a few minutes to stabilise, when switching between AC and DC biasing. Therefore, it is possible to indicate idle times before beginning the observation. This is particularly important for skydips, when changing from AC to DC biasing. It is necessary to wait at least 10 to 15 minutes after changing the biasing mode before starting the measurement. But this waiting time can be avoided, when a map is observed prior to the skydip, because the instabilities do not affect the mapping. This idle time is displayed as a countdown status window, which appears after issuing the observing command. It can be interrupted by pressing the Break button in this status window. There is an option to activate an experimental sky noise reduction algorithm. If used, the data reduction programme will try to determine the sky background fluctuations from the 4 AC channels that are not looking at the source, and then correct the data channel for these fluctuations. Pointing and focusing results will be calculated afterwards. This may improve these results, if the sky background is unstable. The sky subtraction algorithm is also applied to on-off measurements. However, the much more advanced and powerful procedures included in the off-line version of *simbaread* should be used for a proper reduction of on-offs. There is also a possibility to select, whether an audio alarm should be activated, if an observation seems to be hanging. Furthermore, one can indicate, whether the data should be automatically transferred to Dorado, which is the default setting. Finally, one has the opportunity to open the *advanced configuration menu*, which however is only meaningful for maintenance and testing purposes and is therefore not explained.

The Observe button must be pressed in order to start an observation. A configuration window appears according to the selected observing mode, where the observer can specify the parameters of the observation. One or two status windows appear on the right-hand side of the screen after starting a measurement, according to currently running measurement (see Fig. 1.9). They show the progress of the observations and also contain buttons that allow the observer to interact with the observation, i.e. abort or stop it in some way. Only buttons with black text are active, and one has to click again in a verification window before any action is taken. Depending on the status, there are up to three possibilities to stop an observation: Break interrupts the measurement after the next observation (loop, on-off, map scan) has finished. Abort stops the current measurement immediately and all data is lost. Stop also stops immediately, but saves all data taken so far.

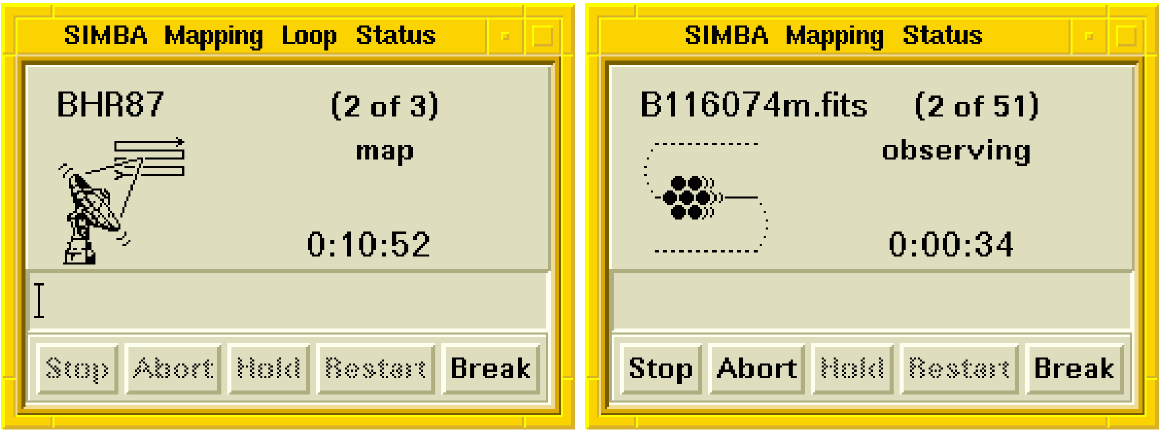


Figure 1.9 Status windows during observing. The left one shows the source name and the total amount of loops together with the elapsed time. The right window displays the name of the data file and the time that has passed during the very loop. Icons demonstrate the type and status of observation. If the measurement consists only of one loop, the left window is hidden behind the right one.

1.4 Pointing

The pointing measurement is necessary, in order to determine the appropriate offset correction to the telescope coordinates and align the telescope with the celestial coordinate system. The misalignment originates from the various optics in the receiver cabin, which redirect the telescope beam to the desired receiver. It is also slightly depending on the weather conditions. The user interface (see Fig. 1.10) for the pointing measurement gives the opportunity for indicating the quantity of iterations (default = 3), the length of the scans (default = 100$''$) and the scanning speed (default = 25$''$/sec). Some point-sources might be too weak for the pointing measurement. Therefore, it is recommended to test their strength with one single pointing without applying the result automatically.

The pointing is carried out in AC-bias mode with a DC coupling of the bolometer channels. 5 pairs of the array are combined to differentiating devices, meaning that always one member of a pair is charged inversely to the other. The difference signal is written to a new logical (non-physical) channel (see Fig. 1.11). Currently, we use channel 41 for pointing, which contains the differentiated signal of the physical channels 1 (+) and 10 (-). The resulting pointing position has been determined to coincide with channel 1.

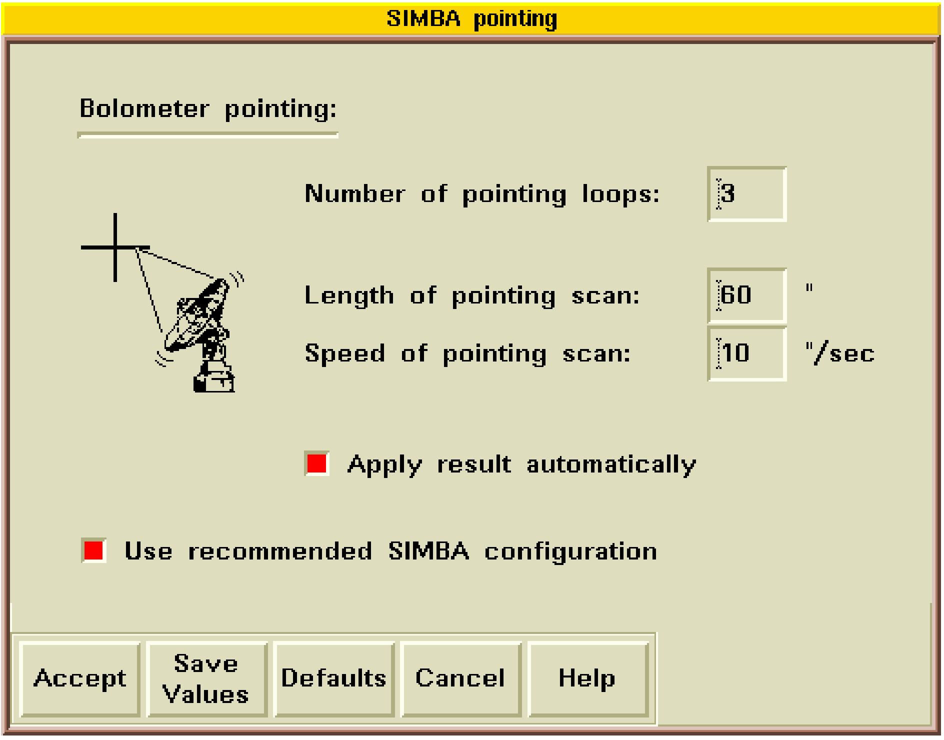


Figure 1.10 Graphical interface for pointing measurement.

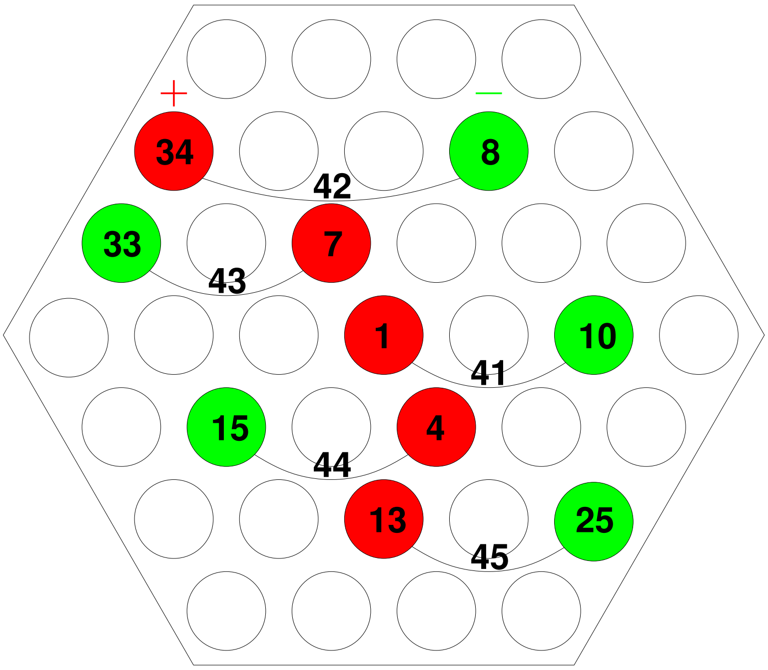


Figure 1.11 AC biased / DC coupled wiring of the bolometer channels. Each of the resulting logical channels 41 to 45 contains the difference signal of coupled physical channels. This configuration is used for pointing, focus and on–off measurements.

The pointing procedure consists of two scans each in azimuth and elevation. A simple Gaussian function is fitted to both scanning directions (see Fig. 1.12) by calling the *simbaread* programme (see Sect. 2.3). It calculates the offset correction and hands the values back to *Pegasus*. If the option for subtracting the correlated skynoise has been selected in the $\boxed{\text{Configure}}$ window, simbaread will try to subtract sky fluctuations from the data prior to calculating the offsets.

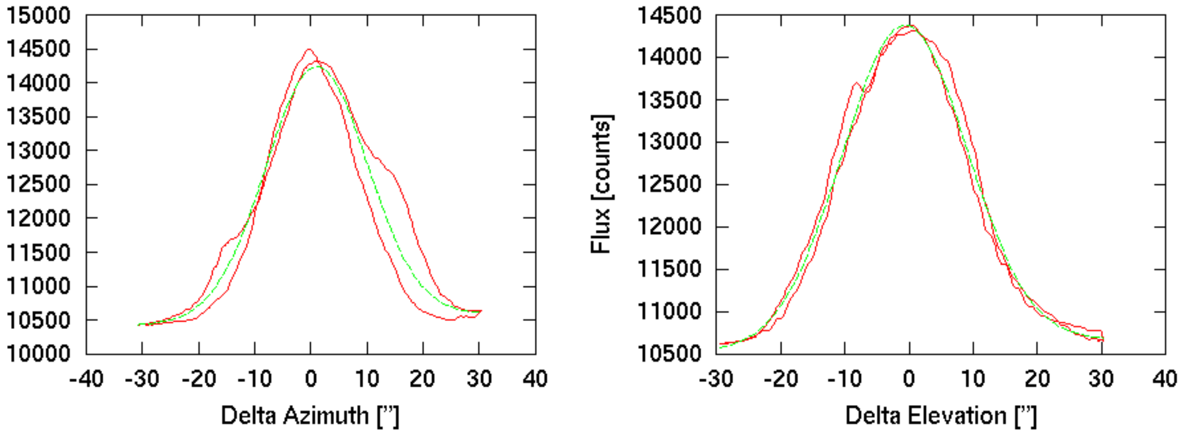


Figure 1.12 The pointing is carried out with two scans back and forth in azimuth and elevation each. The resulting spatial flux distribution is fitted with Gaussian functions, where the peak position gives the current pointing offset.

1.5 Focus

Focusing is necessary in order to, firstly, achieve the best spatial resolution and, secondly, provide the highest signal-to-noise ratio possible. As shown in Fig. 1.13, a focused telescope results always in the narrowest spatial flux distribution of a point-source with the highest peak flux. Therefore, simple on-off measurements (see Sect. 1.7) at different subreflector offsets help to derive the optimal offset value.

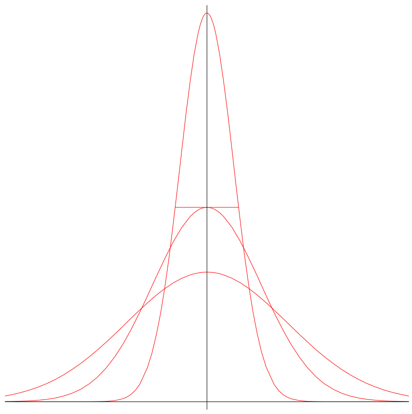


Figure 1.13 Defocusing not only broadens the flux distribution of a source, but also decreases its peak value. Hence, a focused telescope always yields the highest integrated flux of a point-source, when measured within the central beam.

Focusing is done by on-off measurements at three subreflector offsets, symmetrical around a central default value. The observer can insert the number of iterations (default = 3) and the central position for focusing. The current setting is automatically inserted as a default value. Finally, it is possible to indicate the relative positions for the offset measurements (default = 0.6 mm).

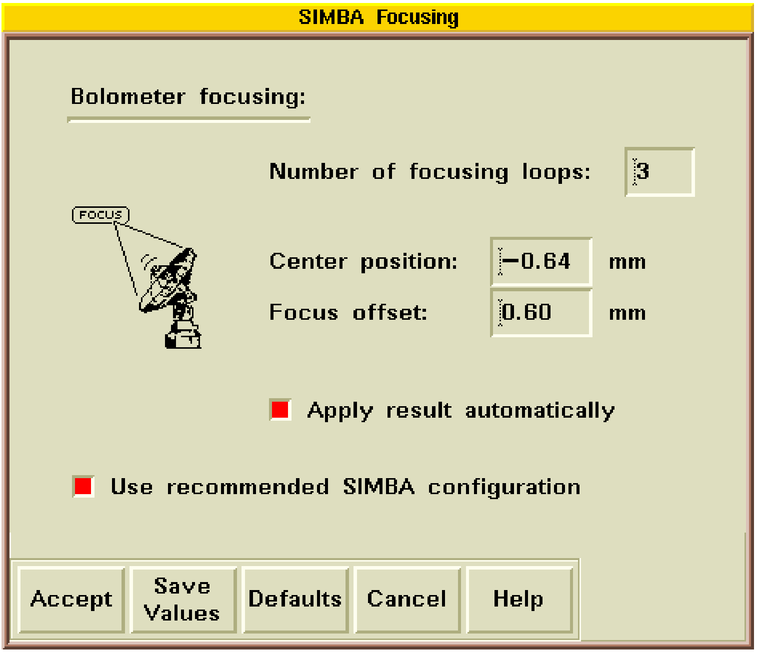


Figure 1.14 Graphical interface for focus measurement.

Every position is measured twice and the resulting flux values are fitted by a parabola, as depicted in Fig. 1.15. The apex is taken as the optimal subreflector offset.

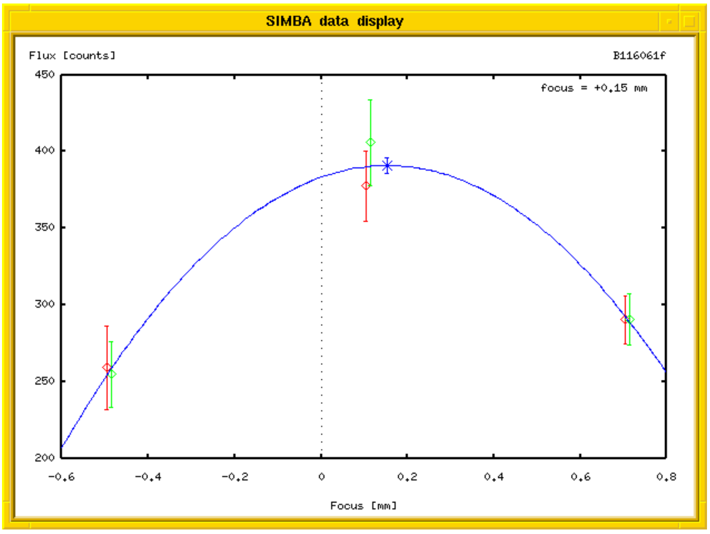


Figure 1.15 In total six on-off measurements of a point-source at three different subreflector positions are used to derive the optimal subreflector offset. The apex of a fitted parabola is taken as the correct focus value.

1.6 Skydip

Skydip measurements are necessary in order to be able to correct the observations for atmospheric extinction. In fact, not the sky extinction, but its emission is measured implying a black-body behaviour (emission = absorption). As visible from the user interface for skydips in Fig. 1.16, the observer can only choose the azimuth position, where the measurement is carried out.

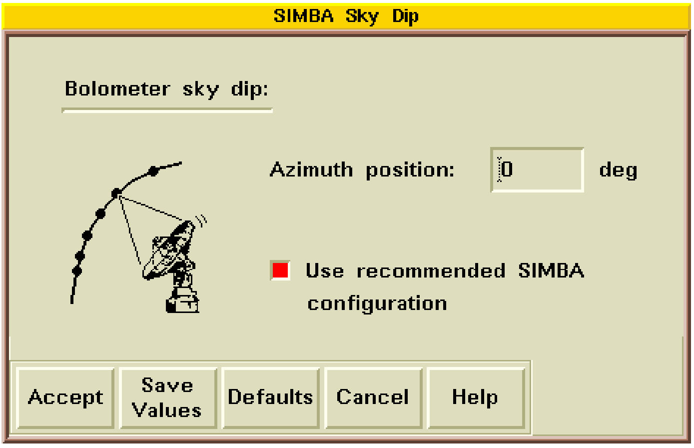


Figure 1.16 Graphical interface for skydip measurement.

The default value is the current telescope position, unless this position would take the telescope into the sun limit. The total power signal of the skydip measurements is taken with the three physical channels 1, 23 and 32 and stored into the logical channels 46 to 48 (see Fig. 1.17). However, we only use the central channel 48. The channels are charged DC-biased with a DC coupling. Since switching from AC biasing to this mode leads to a signal drift and the stabilising takes 10 to 15 minutes, we recommend to do a skydip after the first map, which has the same bias setting, but does not suffer from instabilities.

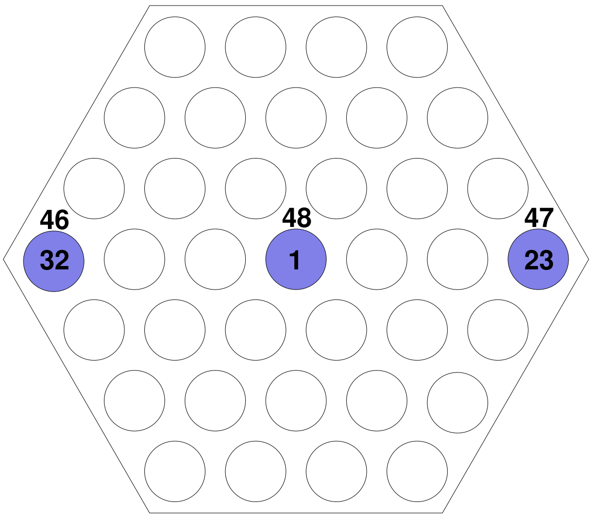


Figure 1.17 DC biased / DC coupled wiring of the bolometer channels. The resulting logical channels 46 to 48 are the total power signal of the physical channels. This configuration is used for skydips only.

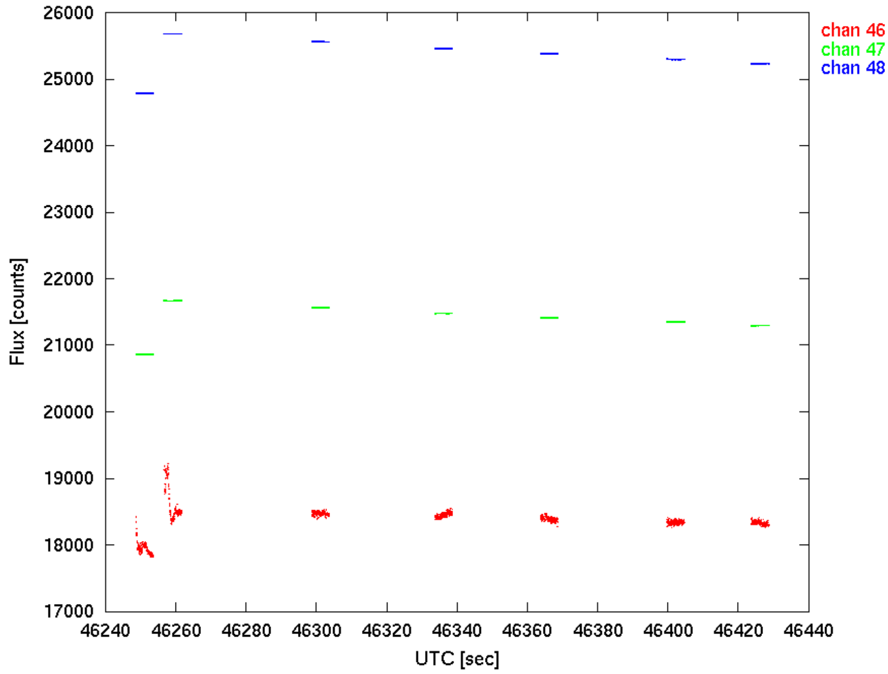


Figure 1.18 Visualisation of the signal during the skydip measurement. Only channel 48 is used.

We take measurements a 6 positions on the sky with different elevations. One additional calibration measurement is done beforehand by automatically putting an absorber of ambient temperature in the optical path of the bolometer array. The signal is visualised in Fig. 1.18. It is easy to see, how it changes with the different elevations.

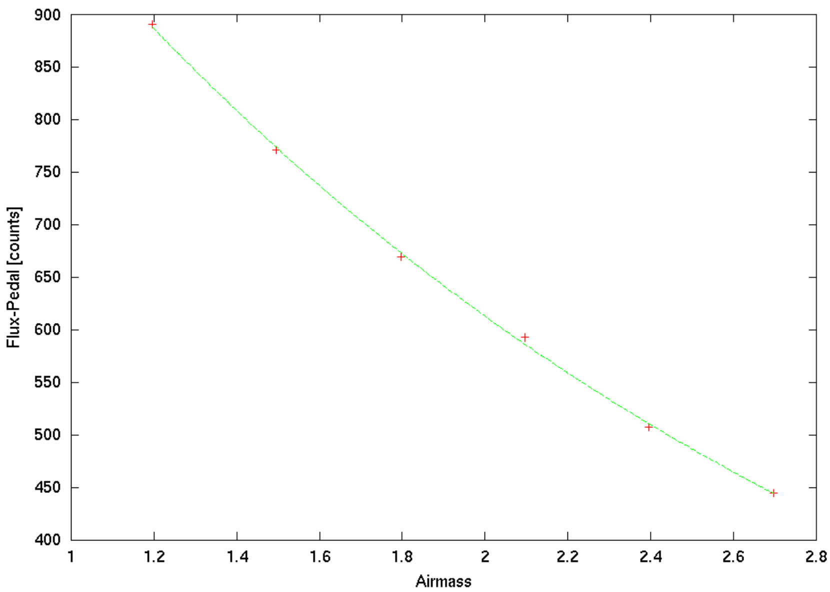


Figure 1.19 Visualisation of the dependency of sky measurement with increasing airmass. An exponential curve is fitted to the data in order to derive the zenith opacity τ.

The sky emission and absorption can be described by

$$S = S_0 \cdot \exp\left(-\frac{\tau}{\sin h}\right) \tag{1.7}$$

where S_0 is the unextincted flux, S the measured flux, τ the zenith opacity and $\frac{1}{\sin h}$ the airmass at the elevation h. The unknown τ can therefore be derived from the skydip measurement by fitting Eq. (1.7) to the data, as shown in Fig. 1.19. This value is used later during the data reduction when correcting for the atmospheric extinction. Opacities typically range between $\tau = 0.1$ and 0.3 at $\lambda = 1.2$ mm.

ATTENTION! Do not forget to go back to the source after the skydip with the Source window.

1.7 On–Off

The first scientific observing mode is the on–off measurement. An alternating series of integrations on the source and the sky background is made, which is called chopping. Both signals are subtracted from each other to reject the sky emission. The technique of a wobbling secondary mirror is used at most of the telescopes, but we do it electronically during bolometer observations. The paired channel wiring of Fig. 1.11 demonstrates, how two physical channels are combined to one differentiated logical channel. Channels 1 and 10 are used for the actual measurements, where channel 1 is on source and channel 10 is off source. The channel separation of about 88″ is therefore the fixed chopper throw. The differentiated, i.e. chopped, signal is stored in channel 41 (see Fig. 1.21). Nodding is achieved by moving the telescope away from the source and repeat the measuring cycle. The observer can freely specify the nodding offset, but we recommend to use the default value of −88″, i.e. the channel separation. This puts the source into the negative beam of the chopping measurement and effectively doubles the on-source time and consequently raises the final signal-to-noise ratio.

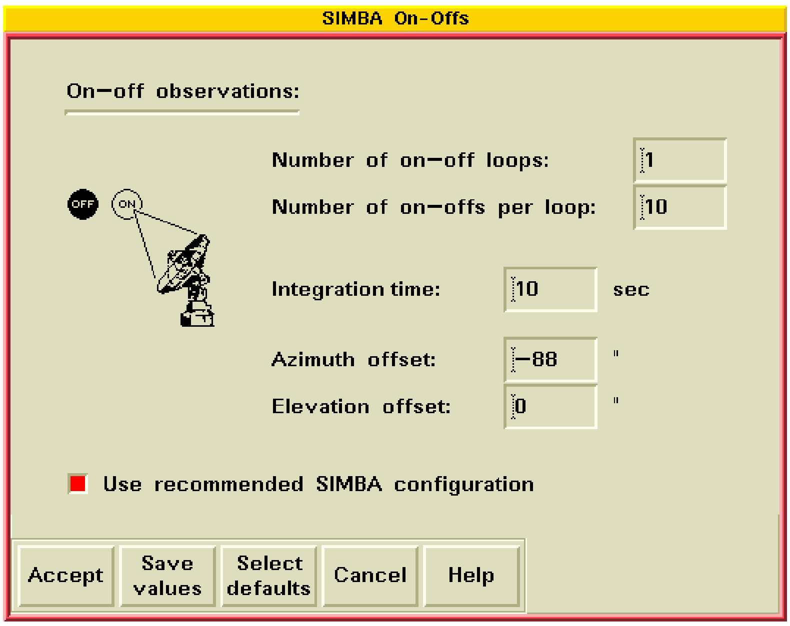


Figure 1.20 Graphical interface for on–off measurement.

The number of on-off loops is equivalent with the number of measurements and therfore single fits files to be created during the observation. In order to specify, how many noddings should be made in one measurement, the number of on-offs per loop must be given. The integration time is the time spent on source in one nodding cycle. Finally, as already mentioned, the nodding offsets can be altered in azimuth and elevation. However, we recommend to keep the default setting.

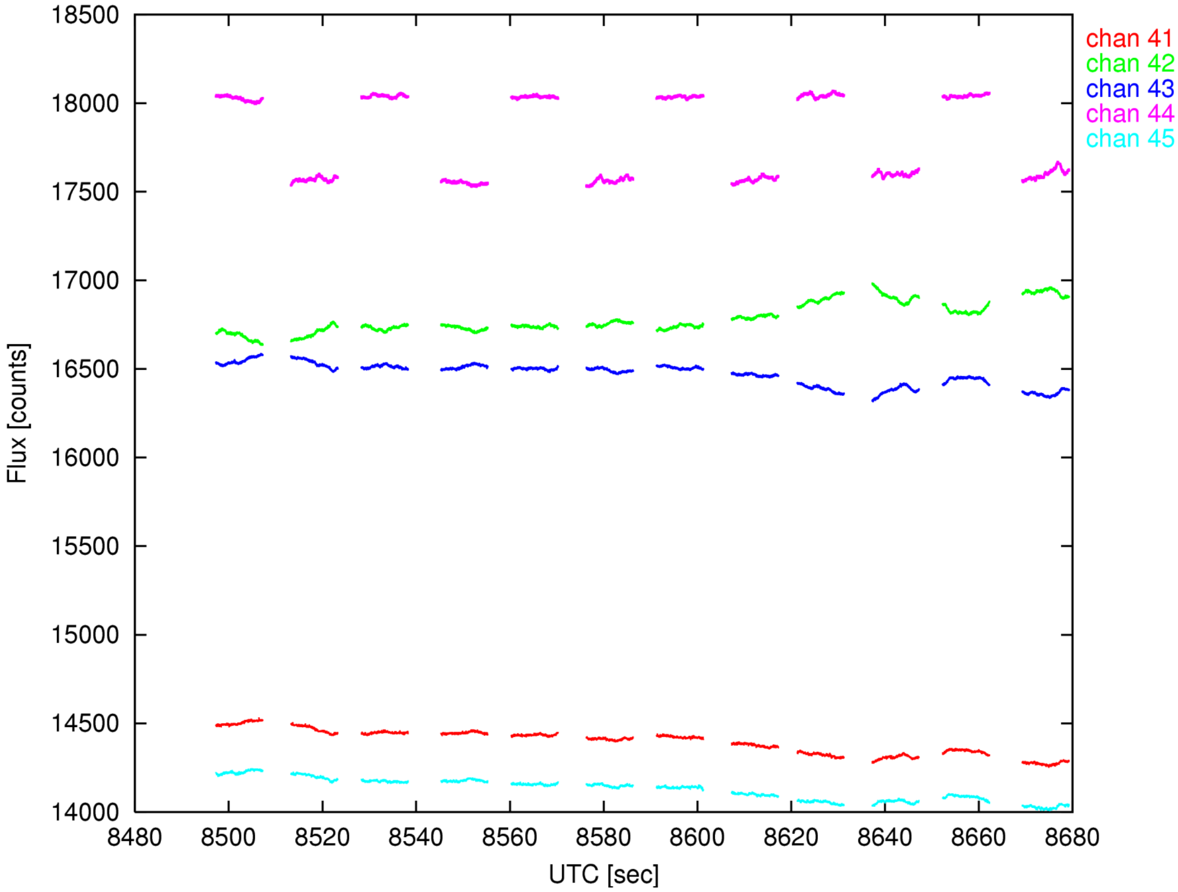


Figure 1.21 Time series of the data during on–off measurements. Note the change of intensity of channel 44, when the telescope switches from on- to off-position. The other channels only see the sky fluctuations, which is helpful in the rejection of the correlated sky noise.

In order to calculate the observing time needed, we give a preliminary sensitivity of 300 mJy$\cdot\sqrt{\text{s}}$. We want to stress that the time derived from this value has to be regarded as the time on source alone. The actual observing time would be higher by a factor of 3, considering the time spent on the off-position and additional overheads.

1.8 Fast Scanning

SIMBA has its greatest advantages in the newly introduced fast scanning mode. The large field of view (FOV) of about 6$'$ in diameter together with the high scanning speed of typically 80$''$ allows to create large maps with an average sensitivity of 30 – 40 mJy/beam in a small amount of time. Fast scanning is carried out with DC biasing and an AC coupling of the 37 channels (see Fig. 1.22).

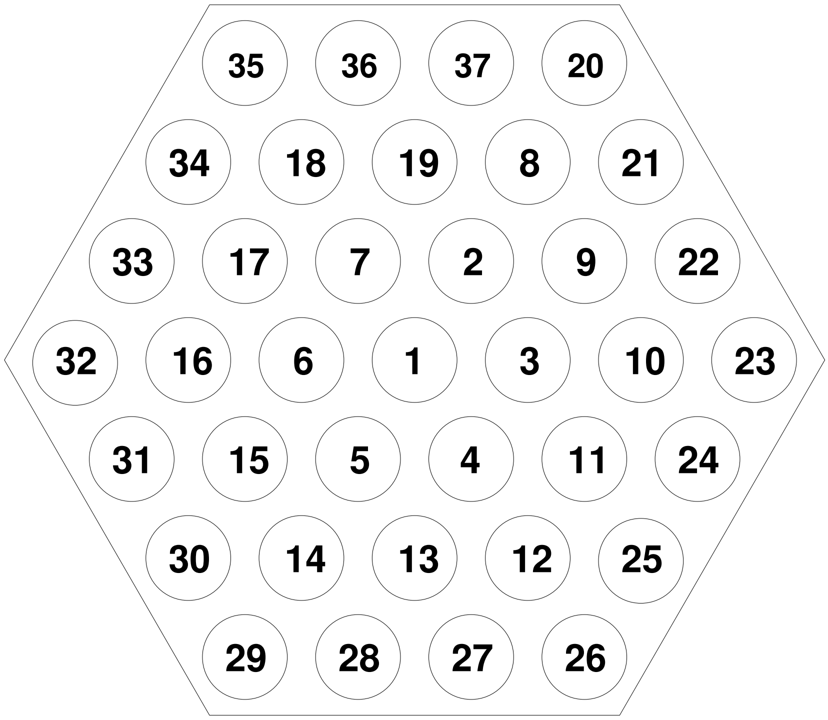


Figure 1.22 DC biased / AC coupled wiring of the bolometer channels. This configuration is used for the fast-scanning mapping mode.

The observer inserts the mapping parameters in the user interface, as showed in Fig. 1.23. The scan length is the size in the scanning direction, which is azimuth by default. The map width is the extent perpendicular to the scanning direction. Furthermore, the observer can insert a number of map repetitions. The underlying software calculates the number of necessary scans according to the scan separation. This is set to $8''$ by default, which yields a threefold beam sampling. An estimate of the total observing time is given at the bottom of this window after specifying the desired scanning speed. The map is also centred on the entered source position automatically. However, the observer can override this setting by specifying the map margins in horizontal coordinates relative to the source position. **We strongly suggest not to use RA/Dec offsets in the Source window**, as they are not written into the FITS header. We recommend to define new maps for every single position with proper coordinates instead. If, however, the observer still wishes to use offsets, the current *simbaread* version allows to add these offsets manually during the conversion process.

The telescope starts the observation after accepting the mapping parameters at the lower left corner of the map. It is carried out in parallel scans. The system has recently been upgraded to choose an arbitrary scanning direction. The scanning angle can be indicated in the user interface as well. Even pure elevation scans are available. This allows to cover a large region with several well orientated maps or correct for field rotation. The latter argument is especially important, when the same area should be mapped quite often to reduce the residual noise. Since the FOV rotates during the observing run because of the horizonal mounting of the telescope, normally only the central region of the final map would have the maximal signal-to-noise ratio. Yet, by avoiding the rotation, the total map has the same low residual noise. Unfortunately, the reduction software is not yet prepared to handle these data, thus we do not recommend to use this option at the moment.

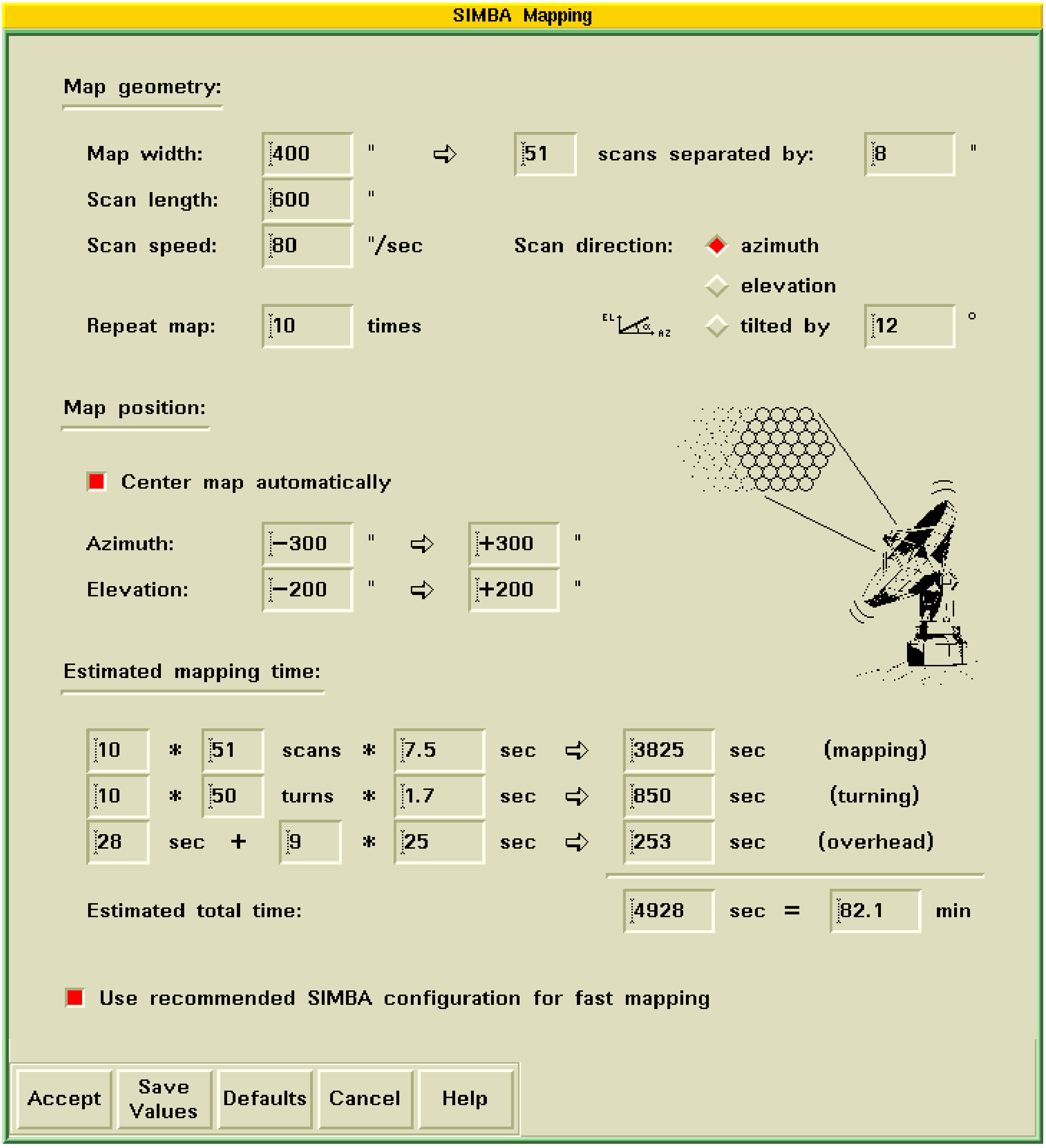


Figure 1.23 Graphical interface for fast scanning.

1.9 The SIMBA FITS Format

Data taken with SIMBA is stored in FITS files that can be read and reduced by NIC and MOPSI after conversion with the *simbaread* programme. The FITS files contain a general header and two or three extension blocks, each consisting of a header and a data block. The first extension block called *subscans*, includes timing information specifying, when different scans or on–offs start. The second extension block called *data* comprises all the data, stored in time order. All data taken in one observing loop is stored in one FITS file. FITS files can therefore rapidly become big. The data flow from SIMBA is roughly 1 MB per minute.

A list of ephemeris is calculated and used to control the telescope, when observing objects in the solar system. This ephemeris list is also added to the FITS file as a third extension table called *ephem*, which actually is located immediately after the main FITS header. The boolean keyword *PLANET* in the main FITS header indicates whether this table is present.

Apart from the inconvenience of having large files, it should be pretty safe to run long observations. The software has been designed to create a FITS file out of the data retrieved, even if the observation runs into trouble. In the unlikely event that the observing programme crashes without having created a FITS file, most of the data can still be recovered from temporary dump files that are created during the measurement. If this happens, a warning will appear when trying to start a new observation. The programme will then ask, if it should recover the old data into a FITS file, before starting the new loop.

1.9.1 Data Archiving

The computer used for the observations is called *Vela* and is not a very new HP work station with limited disk space. There are 3 computers called *Dorado, Eridanus* and *Sextans* that should be used for the data reduction. They are powerful PCs with RedHat Linux installed.

As soon as a FITS file has been created on *Vela*, it is automatically transferred to *Dorado* and put into the **Data** subdirectory of the user account as well as into the SEST data archive. *Eridanus* and *Sextans* are set up so that you will also see your data on these machines. The automatic transfer can be switched off in the configuration window of SIMBA, but this is not recommended. In that case, the $\boxed{\text{Transfer}}$ button has to be used in order to copy the data files to *Dorado*. Once a file has been sent to *Dorado*, it may take a few minutes before it appears in the observer's account. This is, because the archiving daemon is checking for new files only once a minute. All observations are be kept on *Vela* until the next day's maintenance, when a SEST staff member removes all SIMBA files. Any raw data left on *Vela*, which has not yet been transferred to *Dorado*, would then be sent there.

WARNING! The data files should **never** be transferred to *Dorado* during a running observation. This could cause a system crash because of transmission of time critical data on the control network during observations. The same is valid for *telnet* or *ftp* connections.

It is not recommended to try transferring the data to the observer's home institute via *ftp*, since SIMBA data typically consists of relatively large files and the La Silla internet connection is quite slow. We encourage observers, however, to bring DAT tapes or CD-ROMs for storing the data. Once the observing run is finished, a SEST staff member will prepare the DATs or CDs with all the **raw** FITS files from the SEST archive. Backups of other data, e.g. reduced maps, will be the observer's responsibility. The `save_data` script is available for this purpose. A second backup of the **raw** data will be stored at SEST. If the observer, for some reason, recognises the loss of some of the data, we can be contacted for providing a new copy of it. After his/her departure, all observer accounts will be cleared and restored for the next observer.

Chapter 2

Data Reduction and Analysis

2.1 State of Development

Since the observational technique of the fast scanning mode is still unique and totally new, a whole set of new reduction procedures had to be be established. This has begun in June 2001 with the NIC upgrade. Later on, also MOPSI was modified step by step and the development is still not complete. Although we think, that the currently available versions are now stable and functional, they still have to be regarded as preliminary and in an experimental state. Especially the reduction procedures themselves are subject to optimisation and will change in future. Therefore, this reduction manual can by no means be used as the final one.

The current versions of the computer programmes can be downloaded at:

`http://www.ls.eso.org/lasilla/Telescopes/SEST/html/telescope-instruments/simba/software/`

2.2 General Concept

Independent from the reduction software chosen, the whole reduction procedure is structured into several steps that are necessary to attain proper results.

1. Conversion	The raw FITS files must be modified to be read by the reduction software. *Simbaread* (see Sect. 2.3) has been created especially for this purpose.
2. Gain elevation correction	The telescope dish undergoes deformation from the optimal shape by gravitational influence. This distortion is elevation dependent and changes the effective intensity measured on the sky.
3. Opacity correction	The signal intensity is modified by atmospheric extinction. Therefore, the measured radiation must be corrected for the airmass it has passed on its way through the atmosphere. Skydips provide the zenith opacity, which are crucial for deriving the elevation dependent correction factors.
4. Baseline fitting	Several processes can cause baseline drifts, of which long term fluctuations in the sky background radiation and electronic instabilities

	are the most important causes. Additionally, all detectors have different zero point offsets. These influences can be removed or minimised by subtracting proper baselines.
5. Despiking	Eventually, spikes might be present in the data. They can originate from cosmic rays or electronic discharges. Such spikes must be eliminated as they can cause artificially high noise levels, mimic real emission or modify the actual measured flux of a source.
6. Deconvolution	The electronics of the pre-amplifier of the bolometer array behave like an electric oscillator. As a result, the fast scanning mode creates signal distributions with a negative dip. Therefore, they must be deconvolved with a filter function matching the electronic design.
7. Skynoise reduction	The atmospherical fluctuations contribute greatly to the map noise. Theoretically, all – but at least some – bolometer channels should experience the same changes in flux. This highly correlated noise should be eliminated from the data, which can drastically increase the signal-to-noise ratio.
8. Map creation	Finally, all channel maps are co-added and the coordinate system is transformed from horizontal to equatorial.
9. Calibration	The counts must be transferred into physical quantities. A planet map (e.g. Uranus, Mars) can be used for this, because their intensities are very well known and quite easy to obtain.

2.3 Simbaread – The FITS file converter

The converter has been programmed by Roland Lemke and Marcus Albrecht (Astronomisches Institut der Ruhr-Universität Bochum) in order to

1. convert the fits files created by *Pegasus* into fits files, which can be read by NIC (see Sect. 2.4) and MOPSI (see Sect. 2.5),
2. quickly determine crucial parameters from the measurement, e.g pointing offsets, focus corrections, opacity, tracking accuracy, map coverage,
3. create basic graphical visualisations from the observation.

You will find the directories **Data**, which stores the raw data-files, and **Work**, where *simbaread* puts the converted files by default. The single obvious difference to the user is, that the name of the final fits file only consists of 4 digits plus extension. The usage of *simbaread* is quite simple and is explained in Tab. 2.1. The designation of the data and work directory can by changed with the options **-D** (for the directory with the raw FITS files) and **-W** (for the work directory). The directory path is understood to originate in the user's home directory. However, absolute paths indicated with a leading slash (/) are also possible. Coordinate offsets in $''$ from the central RA/Dec position as used in the Source window can be applied manually with the **-r** option.

Only the option -f is necessary, in order to convert the FITS files. *Example:* simbaread -f B188500m. The filename can be written either with or without the file extension. The graphical output is created with GNUPLOT, which creates postscript files in the **Data** directory called <scan>.Counts.ps, <scan>.<obsmode>.ps and <scan>.coord.ps. Possible <obsmode> designations are *point*, *focus*, *sky* and *map*. Note that there is no *onoff*, because the final counts are printed with the verbose option.

Table 2.1 Options for using Simbaread.

simbaread	-v	verbose	gives some messages
	-d	debugging	gives a lot of messages
	-g <type>	graphics	creates postscript files, which contain • information about the results of the measurement • the time series of the counts • the coordinates and tracking errors possible types are: **c**oordinates, **f**lux, **r**esults
	-f <file>	filename	filename of the raw fits file
	-a <file>	append	append analysis results to a file; in case of skydips it creates a list of opacities, which can be read by MOPSI
	-D <directory>	data dir	override of the default data directory (subdirectory of the users' home directory)
	-W <directory>	work dir	override of the default work directory (subdirectory of the users' home directory)
	-c <channel>	analysis	sets the channel for the data analysis, e.g. for skydips (overrides the default)
	-r <RA> <Dec>	offsets	adds RA/Dec offsets in $''$ to the map centre coordinates

2.4 NIC

2.4.1 Introduction

NIC s a software package originally developed to reduce continuum data taken with the bolometer arrays of the MPIfR[1]. It is part of the GILDAS software package, which also contains reduction facilities like CLASS as well as the GreG graphical display. NIC is largely self-documented as paperback (available at SEST) or the online help function. This document is by no means trying to replace the above documents but rather acts as an introduction to the data reduction of SIMBA data. This document contains tips and scripts from several authors among SEST staff members and observers that have been visiting us.

2.4.2 Workspace

NIC is installed on the Linux machines called DORADO, ERIDANUS and SEXTANS. They can be accessed by using your username and the guest password. Several scripts have been prepared for the reducer's convenience:

ini.nic	initialises the program and sets default directories and file extensions
ini2.nic	applies zero-order corrections, which are necessary to obtain the correct intensities
quick.nic	produces a quick map of your data with minimal corrections and reduction.

[1] Max– Planck–Institut für Radioastronomie, Bonn, Germany

2.4.3 Getting Started

Just type **nic** in the **Work** directory. The startup messages show look something like this.

```
[green@dorado Work]$ nic
W-SIC_GTLGTR,  No local logical name table
I-SIC_GTLGTR,  Using user defined logical name table /data/observers/green/.gag.dico
I-SIC,  Normal keyboard mode
Default macro extension is .nic

 INPUT & GO : Procedures for multichannel display:

 - CONVERT        converts AZ-EL maps to RA-DEC maps
 - FLUX           computes fluxes
 - MAKPLAN        computes double beam maps for all channels
 - MAP            makes (color) contour maps for all or selected channels
 - ONOFF          produces a table of calibrated fluxes
 - QUICKLOOK      reduces scan by scan (use at the telescope)
 - REDUCE         reduces a scan (experimental ... modify at your will)
 - RESTORE        uses the EKH Algorithm to restore maps for all channels
 - SELECT         as SIGNAL, but uses cursor to select channels
 - SHIFT_AND_ADD  uses the homonymous technique to restore maps
 - SIGNAL         plots bolometer data for a selection of channels
 - TOTPOW         displays total power data for a sequence of scans

 Type: INPUT procedure-name for further information on a procedure
 Type: procedure-name, GO procedure-name or simply GO to (re)execute

Type INFO for recent modifications
NIC>
```

To open a graphical window to be able to visualize the data during reductions, type

```
NIC> device image white
```

or

```
NIC> dev i w
```

as an abbreviation. This will open a new graphical window with a white background. This window can be resized and the current plot will also be scaled according to new window dimensions after refreshing the plot with

```
NIC> zoo(m) ref(resh)
```

Since NIC uses GreG for visualisation, the plotting commands are equal to those of GreG.

2.4.4 Initialisation

Scripts in NIC are just plain text files which contain NIC commands. Each command should be in its own line. You can add comments to your scripts by starting it with exlamation mark (!). Scripts can also take command line input parameters. You refer to them with `&1, &2, &3`, etc. In the following example of a script, the user defines the file name as a command line input parameter. The script then reads in the corresponding file. The name of the script file in this case would be **read_data.nic**.

```
! This script (read_data.nic) reads in the file whose name user
! gives as a command line input parameter. You can execute this
! script in NIC by typing '@read_data 1234', where '1234' is the
! name of the file you want to read. It is assingned as a value
! for &1 variable.

signal &1                     ! Read in and display data
say "File &1 loaded"            ! This is typed after completion
```

As visible from the above example, the script can be called inside NIC with a preceeding @-character. This script would be called with

```
NIC> @data_read 1234
```

The script **ini.nic** has to be executed after starting NIC by typing

```
NIC> @ini
```

The initialisation script includes the following commands:

```
! Initialise
let do_check no             !
set opacity on              ! correct for the atmospheric opacity
sic log bolo_data: "./"     ! fits data directory
set ext fits                ! set default file extension
let sic$window no           ! no X widgets
let contour no              ! turn off contours
let do_map no               ! no map plotting after MAKPLAN, RESTORE...
set gain_channel on         ! apply the channel gain corrections
var * read                  ! have read access to all NIC variables
```

2.4.5 Reading Data Files

Only a single command is needed to read data files.

```
NIC> signal 1234
```

This displays the data for all 37 channels as a histogram as shown in Fig. 2.1. If this is the first scan to be read with NIC, the data file has the loaded twice.

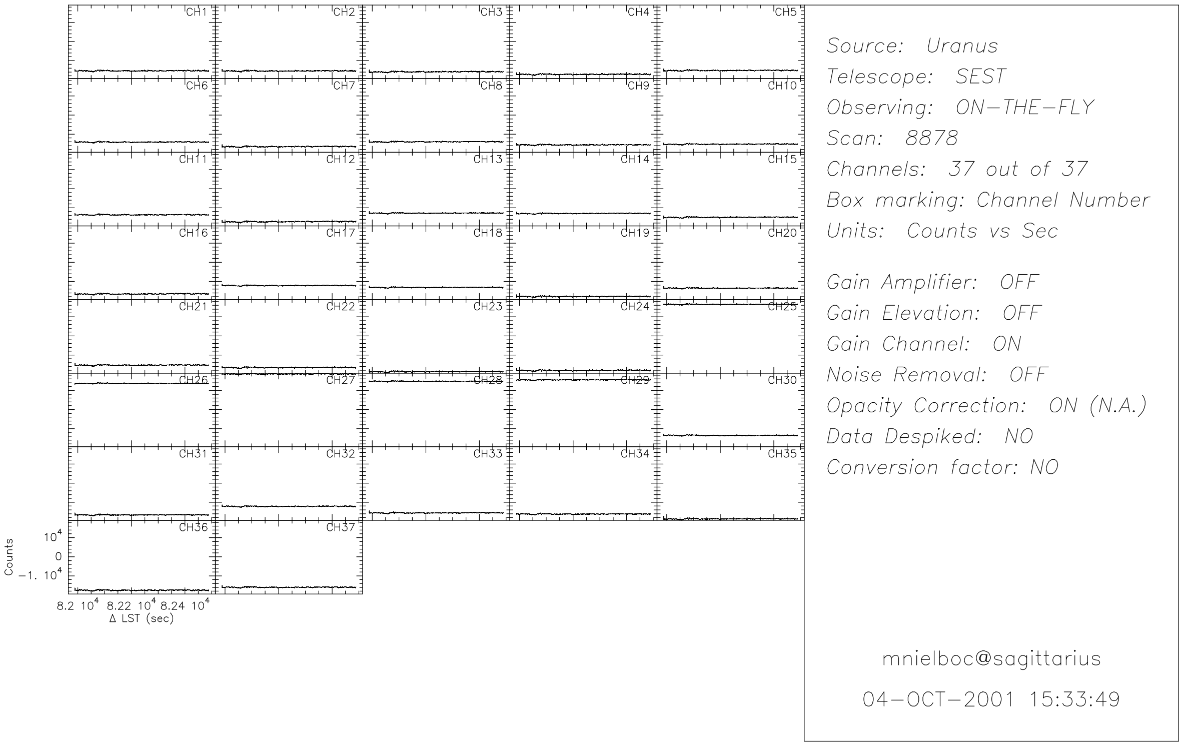


Figure 2.1 The *signal* command displays the data of all available channels as histograms. It is given as intensity vs. LST.

2.4.6 Data Reduction

The following short scripts are extractions from a script file called **quick.nic**, which can be used to have quick look of your data. The final reduction involves many iterations of the steps presented here. The default values used for different tasks are chosen, so that they are useful for majority of the observations.

First of all, the noisy channels (mostly 2 and 12) should be excluded from the reduction process. This can be done by flagging.

```
NIC> flag /chan 2 12
```

The beamsize can only be set after fitting a Gaussian to a pointsource. However, it can also be set manually by altering the referring variables.

```
NIC> var antenna write
NIC> let beam_az 23
NIC> let beam_el 23
NIC> var antenna read
```

After reducing the calibration data, the calibration factor in counts per Jansky can be automatically used in the reduction process of the scientific data.

```
NIC> set jansky on 9.09
```

The next step would be a proper baseline subtraction. An area of the data has to be defined for this, which can be done with the `support` command.

```
NIC> support 30 30 30 30
```

This defines the region outside a box in the centre of each channel map (in horizontal coordinates), where the corners have offset coordinates of 30″ in each direction relative to the map centre. Other definitions are possible, e.g. by defining a free polygon, which is only necessary for for advanced reduction procedures. The actual removal of improper baselines is done with the `baseline` command. Only a zero order fit should be performed this stage of reduction.

```
NIC> baseline /bysubscan /order 0
```

The option `bysubscan` assures a single baseline removal for every scan. Sometimes spikes show up in the data, which might originate from cosmic rays or electronic charges. The can be removed with the `despike` command. From the NIC online help:

```
NIC> help despike

        NIC\DESPIKE [/FUNCTION [Filter [Parameter]]] [/XRMS Threshold]
            [/WINDOW Window] [/ITERATION Number] [/CHANNEL List_of_channels]
            [/INTERPOLATE]

    Remove spikes (cosmic, microphonic,...) from data stored in the two-
    dimensional variable SIGNAL. DESPIKE replaces spikes with a blanking
    value (fixed to -99999.5) if option /INTERPOLATE is not used. Remove
    the baseline first if you wish to use /XRMS, since the RMS threshold
    is in units of the Root-Mean-Square of SIGNAL and not of the standard
    deviation from the mean.

           /FUNCTION          : available Filters are 1 (Default), 2 and 3
                                  1 = variable width (= Parameter) sin(x)/x
                                  2 = gradient filter
                                  3 = level cutting, no filtering
           /XRMS              : RMS threshold after filtering (Default=5)
           /WINDOW            : window in record units for RMS (Default is scan)
           /ITERATION         : number of iterations (Default = 1)
           /CHANNEL           : a list of channels (Default is all channels)
           /INTERPOLATE       : replace spikes by linear interpolation of
                                adjoining data (Default is no interpolation)
```

Despiking should be done very carefully, as especially strong sources might be destroyed. It is suggested, to raise the rms level at least to 10 and to constrain the despiking only to channels with obvious spikes. Interpolation of the blanked data should only be carried out after removing all obvious spikes.

```
NIC> despike /xrms 10 /iter 3 /chan 3 15 17 20
.
.
.
NIC> despike /xrms 10 /int /chan 3 15 17 20
```

Typically, most of the noise is correlated between all or most of the channels. The reason for this can be found in fluctuations of the sky emission. NIC possesses mainly two different skynoise reduction routines, but only one is able to deal with the fast scanning data.

```
NIC> baseline /sky
```

The older `baseline /noise` should only be used with extreme caution, as strong sources might be destroyed. Finally, the beam shape of the source must be deconvolved with a filter function, which matches the electronics of the readout hardware.

```
NIC> run fsdeconvolv /nowin
```

2.4.7 Visualisation

The data has been reduced at this stage, so the final map can be created. This is done with:

```
NIC> makplan
NIC> restore
NIC> convert
```

The `makplan` command transforms the data into channel maps in the coordinate system of the observations. `restore` usually creates single-beam maps from double-beam maps. Since SIMBA observations do not contain chopped data, this command does not do anything. Yet, `convert` needs the output, in order to create the final map by combining all channel maps and converting from horizontal to equatorial coordinates (see Fig. 2.2).

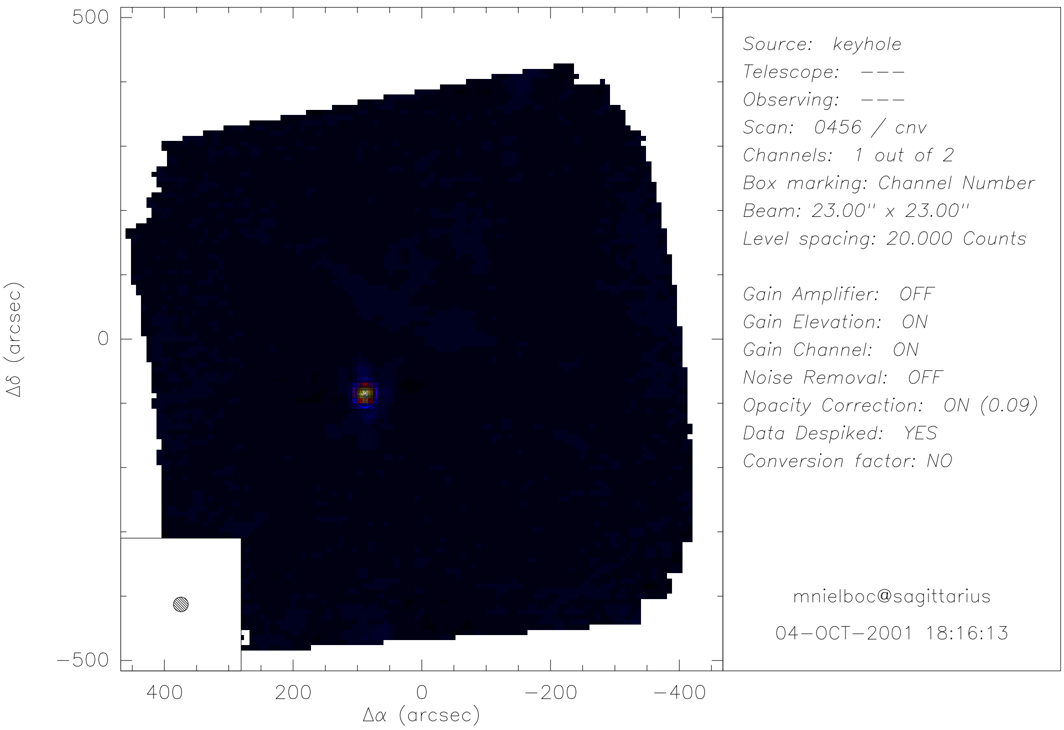


Figure 2.2 The final map after the `convert` command.

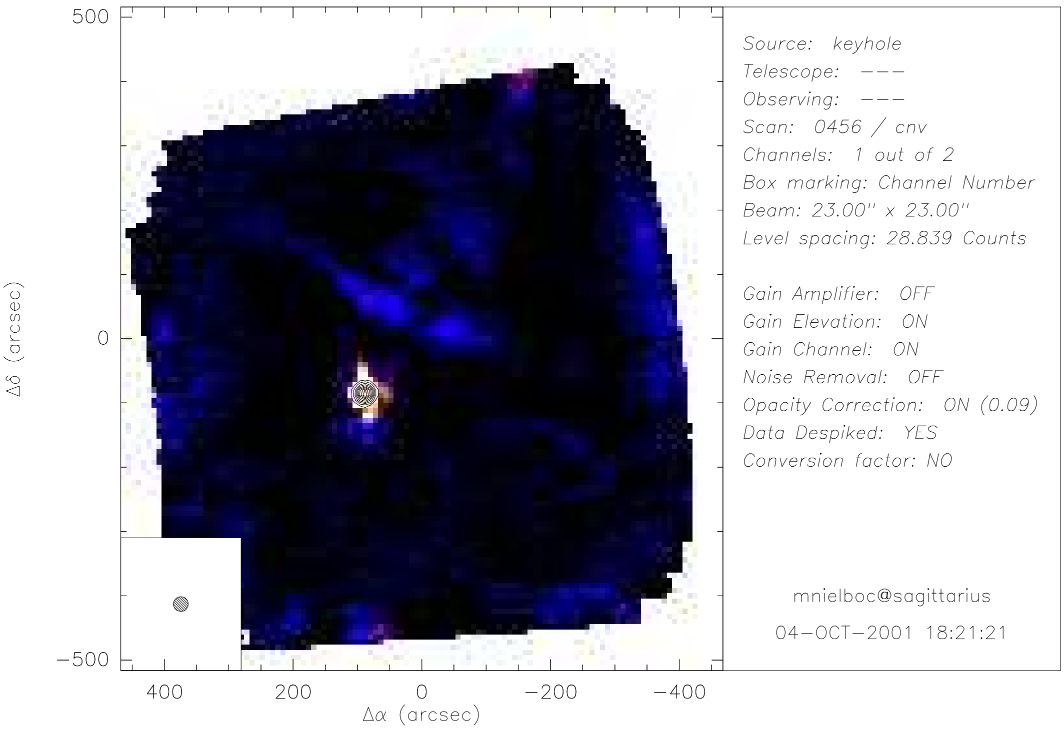


Figure 2.3 A combination of the `plot` command with `box` command allows to redefine the intensity scale manually.

Sometimes, as in this example, the automatic scaling might not be appropriate for the display. In that case, the map should be plotted with the normal `plot` command (see Fig. 2.3).

```
NIC> plot /scaling linear 0 20
NIC> box /unit s
```

2.4.8 Combining Maps

Multiple maps can be co-added to achieve a better signal-to-noise ratio or create a coverage of a larger area. This is done by assigning your scan numbers to the variable **SCAN_LIST** and then executing `convert` again. In the following the scan numbers of individual scans would be 1111, 1112 and 1113.

```
NIC> let SCAN_LIST "1111 1112 1113"
NIC> convert
```

Each map is weighted according to its weighting function, shifted to the common centre of reference, and added. Note, that the result is not displayed in the graphical window, but saved in a file as <sourcename>.cnv. The source name is identical with the labelling in the graphical display (e.g. keyhole in Fig. 2.3). In order show final map, the `map` command is used. The aforementioned combination of `plot` and `box` can be used again for changing the intensity scale. The result typically looks like the example in Fig. 2.4.

```
NIC> map <sourcename>
NIC> plot /scaling log 0.5 100
NIC> box /unit s
```

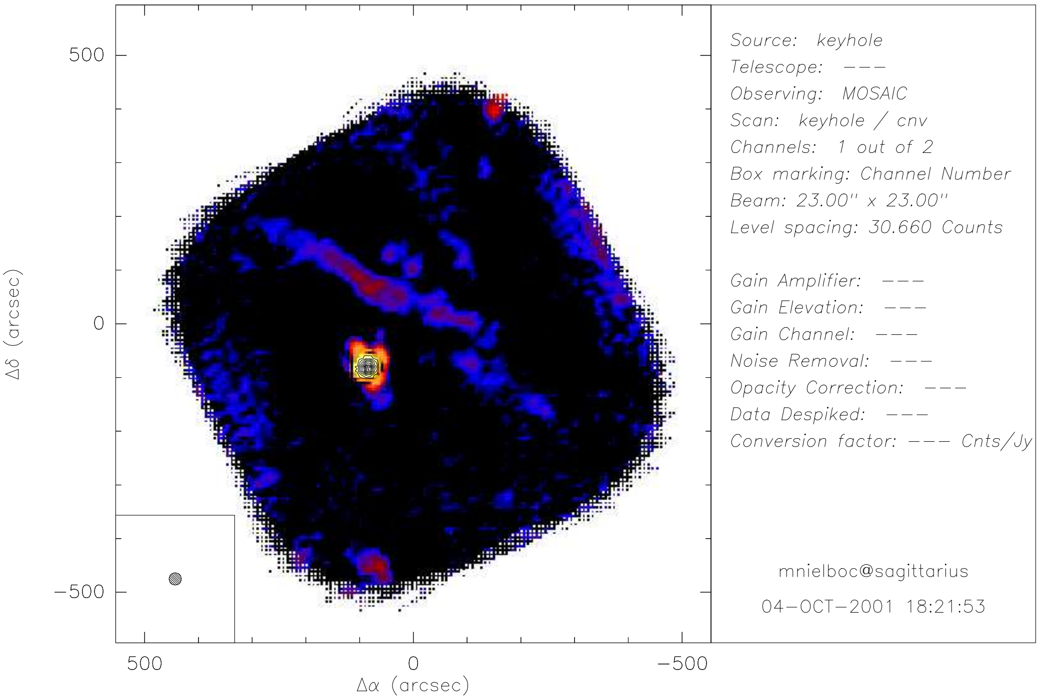


Figure 2.4 Example of a co-added map from six single observations.

2.4.9 Saving Data

The maps are stored as **cnv**, which always overwritten, when the same scan is reduced again. Therefore it is desirable to save the maps permanently. NIC uses the **gdf** format, which can be transformed to FITS with the GFITS package of the GILDAS software. The **gdf** format can also be read with MOPSI.

```
NIC> write image <filename>
```

2.4.10 The quick.nic Routine

```
!reduce
flag /chan 2
flag /chan 12
makplan
support 30 30 30 30
baseline /bysubscan /order 0
despike /xrms 10 /iter 3
despike /xrms 10 /int

baseline /sky
run fsdeconvolv /nowin
signal

makplan
restore
convert
```

2.5 MOPSI

Another very powerful tool to reduce the SIMBA data is MOPSI, a software package, which has been developed and is constantly upgraded by Robert Zylka (1. Physikalisches Institut, Universität zu Köln, Germany). MOPSI is an acronym for **M**apping **O**n-Off **P**ointing **S**kydip **I**maging. It usually provides much better results than NIC, because especially the skynoise reduction is superior to the procedures in NIC.

MOPSI was originally created for HP-UX. But currently, there are also precompiled binaries available for LINUX. As MOPSI uses the GREG graphics interface of the GILDAS software distribution, it has to be installed prior to start working with MOPSI. This also means that many commands related to graphical display are very similar or even identical to the syntax already known from NIC or other GILDAS packages. Furthermore, instead of having lots of additional procedures, which are called from large and widely spread directory trees as present for NIC, MOPSI incorporates everything in one single binary. This allows portability, but also requires typically more powerful computers, because everything is resident in the memory. However, the newest version provide variable memory access, which reduces the demand on computer resources. It is suggested to have about 256 MB RAM memory. The processor speed, on the other hand, is not a very limiting parameter. The most current versions can be downloaded from our web-site:

`http://www.ls.eso.org/lasilla/Telescopes/SEST/html/telescope-instruments/simba/software/`

2.5.1 Preparing the Opacity File

Before getting into the data reduction, it is good idea to prepare the opacity correction. MOPSI does this automatically, if there is a file, which contains suitable information about the opacities that can be extracted from all the skydip measurements. Such a file must have the extension TAUS, while the filename itself can be chosen freely except *skydip*. All the following reduction scripts will use `tau.TAUS`. This file contains at least 5 columns listing the **four-digit scan number**, the **date of the skydip measurement**, the **Julian Date**, the **sidereal time in seconds** as well as the derived **opacity**.

```
5952 11/08/02 2452498.343 46292 0.228
5994 12/08/02 2452498.549 64152 0.162
6027 12/08/02 2452498.728 79638 0.161
6031 12/08/02 2452498.842 3089 0.164
```

All skydip measurements must be listed in this file. Fortunately, *simbaread* is capable of creating this opacity file. The syntax for doing so is:

```
simbaread -a tau.TAUS -f B123456s.fits
```

After issuing the command for each of the skydip FITS files, one should erase the converted four-digit FITS files from the **Work** directory in order to avoid confusion with the other data. The newly created opacity file (tau.TAUS) must be copied from the **Data** to the **Work** folder.

2.5.2 Getting Started

It is a good idea, to have two terminals open while using MOPSI; one for typing in the MOPSI commands and another for e.g. editing batch-procedures. MOPSI is started by typing **mopsi** within the **Work** directory.

```
  MOPSI
  Version 14.08.2002

OPLM> Your LOG & MES files are MOPSI.LOG14 & MOPSI.MES14

      >>>> type  news  to get last informations <<<<

MOPSI>
```

New log- and message-files are created at the start of every MOPSI session, called **MOPSI.LOG** and **MOPSI.MES**, followed by the number of the new files (14 in this example). MOPSI.LOGxx contains the user command input and MOPSI.MESxx lists the messages and output of MOPSI. They can be of great use, e.g. to create own batch-procedures. It is safe, however, to delete them *before or after* using MOPSI.

The graphical display window can be opened afterwards with the GREG command **device image white** or simply **dev i w**. Such abbreviations are very common since the internal command parser recognises usually only the first three letters avoiding possible ambiguities. Therefore, all commands can typically be restricted to this or less.

2.5.3 Reading Data Files

Since the FITS data standard was not very common for radio, millimetre and submillimetre data, MOPSI supports a large variety of different data formats. Between them are NOD2, RGD, GDF and FITS. They are read in with **file in <filename.ext>** followed by the **read** command. For the most common types of data, also **read <filename.ext>** is possible. The correct data format and extension are necessary to use the latter procedure. The fundamental difference between these commands is that **read <filename.ext>** loads the file from an input directory, previously defined by **in-dir <directory>**, which by default is the current work directory.

Just like NIC, MOPSI cannot read the raw fits files. Therefore, they have to be converted with the simbaread software (see Sect. 2.3). The files are read in with **read xxxx.fits**, including the file extension. Yet, a more elegant way to do this, is to use file lists. Such a list is a file with the extension **LIST** containing the names of the fits files. This can be most easily achieved by issuing **ls *fits > fits.LIST** within the second LINUX window. This list is then transferred into MOPSI with **lis fits** (no extension). The file list can be checked by writing **lis**.

```
MOPSI> lis fits
LOLIS> Loading the LIST fits.LIST
       6 observations in the LIST
MOPSI> lis
POLIS> List of INPUT observations, page 1
  1 keyhole    1199.1 27/07/01      -1.0000 -0.0003 -1.00 m 0456.fits
  2 keyhole    1199.1 27/07/01      -1.0000 -0.0003 -1.00 m 0458.fits
  3 keyhole    1199.1 27/07/01      -1.0000 -0.0003 -1.00 m 0459.fits
  4 keyhole    1199.1 27/07/01      -1.0000 -0.0003 -1.00 m 0460.fits
  5 keyhole    1199.1 27/07/01      -1.0000 -0.0003 -1.00 m 0461.fits
  6 Uranus     1199.1 13/06/01      -1.0000 -0.0003 -1.00 m 0694.fits
  #  object    lambda   date       jul_date   LST    elev type   file
```

MOPSI always quotes the number in the file list (first column) together with the file name (last column). If further information is available, MOPSI also gives the wavelength, the date, the Julian date, the LST, the elevation and the type of the observation. Meaningless default values are quoted otherwise. The files are easily read in by indicating the number in the list, e.g. **read 3**.

```
MOPSI> read 3
SETIFN> File: 0459.fits
 nxyall 0 nxy 32560 nphas 16 ndatph 32560
REASIMBATAB> Reading the data
.
.
UTTOSTV> Date: 26-07-2001
REASIMBA> Source: keyhole
        40 RCs red in
READAT> Creating 40 PLACED OTF grids
 otfrg2> columns 0 change to 485 rows 76 planes 1 RCs 40 header 0
         xmn -278.153839 xmx  269.538452 xincas  1.23076928
 otfrg2> ndat 32560 ndatt 1302400 npti 1474400
REARCP> SIMBA   BOGLE  RCPCs 2001: Mars: 2910
REARCP> RC PARAMETERS of the 40rc-array successfully red in from:
        /usr/local/mopsi//SEST_2001a_40.rcp
APPRCG> Applying the RELATIVE RC GAINS (FLAT FIELD)
CALRCO> Calculating the Az-El offsets for OFF-BEAMS in the  40rc-array
        epoch 2001.56946, telescope SEST 15-m
```

Further information to the data files can be obtained by displaying the second page of the listing (**lis in-l 2**). The option *in-l* stands for *input list* and has to be specified because of two reasons. MOPSI also has an *output list*, containing all saved files. So confusion must be avoided. On the other hand, the number 2 would be misinterpreted as a list file *2.LIST*. This second page contains the map centre coordinates. Further information can be listed with increasing numbers or replacing the number by keywords like *proj, resolution, rms, offsets proj* <*hh:mm:ss.s dd:mm:ss.s*>.

```
MOPSI> lis in-l 2
POLIS> List of INPUT observations, page 2
  1 keyhole    1199.1 10 42 53.9994 -59 24  0.005 map 0456.fits
  2 keyhole    1199.1 10 42 53.9994 -59 24  0.005 map 0458.fits
  3 keyhole    1199.1 10 42 53.9994 -59 24  0.005 map 0459.fits
  4 keyhole    1199.1 10 42 53.9994 -59 24  0.005 map 0460.fits
  5 keyhole    1199.1 10 42 53.9994 -59 24  0.005 map 0461.fits
  6 Uranus     1199.1  0  0  0.0000   0  0  0.000 map 0694.fits
 #   object     lambda      projection centre     type   file
```

2.5.4 On-Line Reduction

The data reduction can begin as soon as the desired fits file is loaded. A simple on-line reduction routine is presented here, which can be used with hardly any intervention from the user. It might differ a bit from the routine, which is located in the observer's **Work** directory, but the general behaviour is the same. Every batch-procedure must be written in a file with the extension **MOPSI**, e.g. **reduce.MOPSI**.

It is executed by calling it with **@reduce**. Lines beginning with an exclamation mark are regarded as a remark and are not processed.

```
! This is reduce.MOPSI
! delete DC channels
del dc

! initialise a couple of data buffers
init gfit
init histo
init spike

! global baseline fit
base t 5

! gain-elevation correction
corr ge

! despiking (not for strong sources)
set mask
> 5 rms
< -5 rms
interpolate blanks
mask

! define baseline fit area
base r az < 1e9 > -1e9

! same for pointsource (optional)
!base r out 50 sys equa

! load a polygon for manual definition (optional)
!pol 8618
!base r out sys equa

! subtract a constant baseline
base l 0

! deconvolution
set mask
deconv
mask

! opacity correction (opacity file must exist)
tau tau 20000
corr ext
```

```
! higher order baseline fit
base l 1

! store channel rms weights
store weight rms2 az < 1e9 > -1e9

! skynoise reduction
snf
 it 5 0
 no chk
 > 5 rms < -5 rms
 r 1 900
 best 6
 llsf 1
 ffc avcn rc
 cn med
 cn cor
 cn cd
 cn w cfr 1 0.95 b
 no source
run

! create final map
convert
```

This procedure performs automatically baseline fitting, despiking, deconvolution, skynoise reduction and the creation of the final map and can be displayed with the `plot` command (see Fig. 2.5).

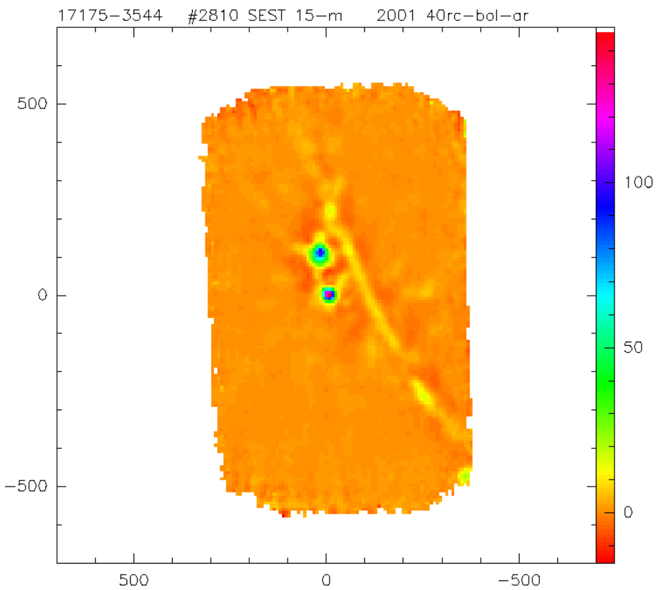


Figure 2.5 Typical final map processed with MOPSI.

The `plot` command alone displays the current data or map with automatic scaling. But it also provides the possibility to display the map either in linear or logarithmic scaling. The complete command might be e.g. `plot scal lin 0 20` or `plot scal log 5 100`. The numbers are the low and high scale margin. Note that MOPSI always remembers, whether the last plot was created in linear or logarithmic scaling and will apply the same, if only `plot` is commanded. If the map contains negative data, a logarithmic automatic scaling results in an empty image, while MOPSI complains about it. The colour table can be changed with **lut** <**table**>. See Tab. 2.2 for possible selections.

Table 2.2 List of colour-tables.

aips0	alex	backgr	blue	blulut
color	flag	gamma3	green	halley
heat	idl11	idl12	idl14	idl15
idl2	idl4	idl5	idl6	ikon
isophot	light	lnspiral	manycol	mousse
neg	null	pastel	pseudo1	pseudo2
rainbow1	rainbow2	rainbow3	rainbow4	rainbow
ramp	random1	random2	random3	random4
random	real	red	smooth	staircase
stairs8	stairs9	standard	white	

2.5.5 On-Line Reduction – Step by Step

This section intends to explain the various commands and actions during the on-line data reduction as proposed in Sect. 2.5.4.

`del dc`
The total power channels 38 to 40 are deleted. They are included in the raw data for information about the atmospheric conditions during the measurement.

`init gfit`
The buffer containing earlier Gaussian fits is emptied.

`init histo`
The buffer containing a list of previous commands is emptied. This should be done frequently to avoid a buffer overflow.

`init spike`
The buffer containing information about the blanked region is emptied and reset according to the current data. Only spike buffers that have the same dimension as the data (e.g. number of channels) can be used.

`base t 5`
This is a baseline fit to the whole time series of the data with a polynomial of the indicated order. This can be used as a first simple baseline fit.

`corr ge`
This is the correction for the gain-elevation variation of the telescope. Said variation is caused by the deformation of the main dish by its own weight. The fitted equation is $\cos(ele-64.^\circ8)^2+0.31\cdot\sin(ele-64.^\circ8)^2$.

```
set mask
```

This stores the currently blanked data in the spike buffer (see above). Together with `mask` it assures re-establishing the original blanked data after processes that modify it.

```
> 5 rms
```

Masks or blanks spikes. They are defined as data, which exceed values of 5 times the rms noise. Despiking can damage strong sources, i.e. clip off the peak of the intensity distribution. As a rule of thumb we suggest to skip despiking, when the source is already visible in the raw data of a single channel.

```
< -5 rms
```

The same for negative spikes.

```
interpolate blanks
```

Interpolates the blanks, which have been produced during the despiking. These gaps in the data are filled with values, which are interpolated from the neighbouring data points.

```
mask
```

Re-establishes the original blanked data (see above) from before the despiking. This is necessary, as the interpolation also affects blanked data, which have not previously been spikes, e.g. produced by tracking errors during the turning of the telescope at the end of a single subscan.

```
base r az < 1e9 > -1e9
```

This defines the area (base range) in the channel maps, where the pending baseline fit is carried out. The whole area is chosen for a first look by defining an azimuth range in arcseconds large enough to cover a complete subscan.

```
base r out 50 sys equ
```

This is an optional base range definition to the one before and especially useful for point-sources. It takes everything outside a radius specified in arcseconds (here 50). As this definition is referring to the centre of the final map in equatorial coordinates, the keywords `sys equ` must be present. If the base range should not be centred on the map, pro <hh:mm:ss.s> <dd:mm:ss> (projection centre) declares its coordinates in RA and DEC.

```
pol 8618
base r out sys equ
```

This is another possibility to define a base range, and is usually applied to extended sources. A polygon matching the edges of the source is loaded first. This polygon is usually created from a reduced map after a previous reduction stage with only a simple base range. The base range is then set to the map area outside this polygon. Since the polygon has been produced in equatorial coordinates, the keywords `sys equ` must be added. When reducing a mosaic of various maps, the polygon is understood relative to the final mosaic. In order to apply the correct bit of the polygon to each of the single maps, the keyword pro <hh:mm:ss.s> <dd:mm:ss> must be added representing the coordinates of the 0,0-position of the mosaic.

```
base l 0
```

This is a baseline fit with a polygon of the order 0, i.e. a constant value. It is applied to each of the single subscans in the map. Only 0 order baseline fits are allowed before the deconvolution, because others will cause wrong results (see deconvolution).

```
set mask
deconv
mask
```

The deconvolution of the bolometer signal restores the real source signal. The fastscanning technique makes it necessary to use AC coupling of the single bolometers in order to reduce the noise at low frequencies. This is achieved by a high pass filter within the bolometer electronics. The amplifier adds a low pass filter, which altogether produces a bandpass modifying the real signal in an unusual way. Therefore, the signal must be restored with a deconvolution procedure, which divides it by the band pass filter function (see Fig.2.6). The masking removes possible edge artefacts caused by the deconvolution.

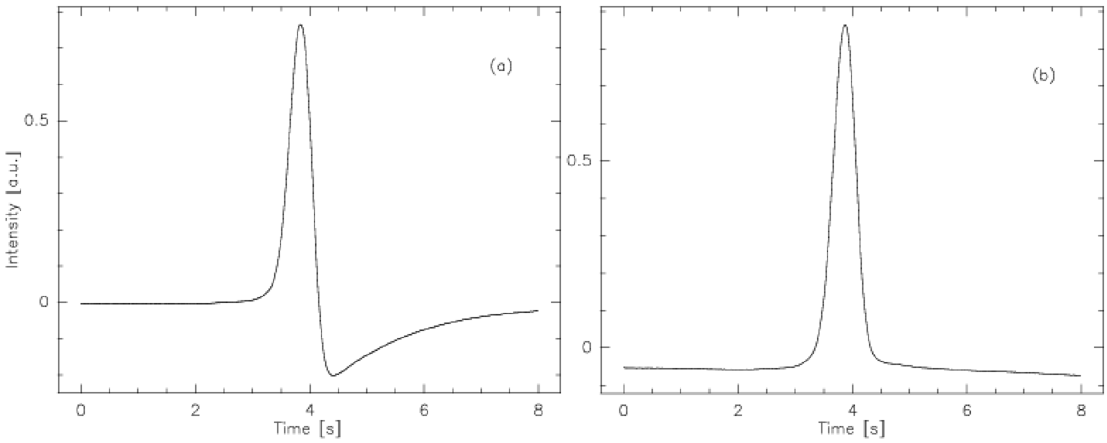


Figure 2.6 The AC coupling of the bolometer channels creates a band pass, which modifies the incoming signal, shown in (a). The deconvolution restores the original source signal (b) (Reichertz et al. 2001, A&A 379, 735).

```
tau tau 20000
```

In order to correct for the atmospheric opacity, the suitable skydip measurements have to be chosen. With the previously created opacity file (here: tau.TAUS), MOPSI does it automatically with this command. The two chronologically neighbouring skydip measurements are selected according to the Julian Date of the observation. In order to prevent an interpolation of the opacity over too large time scales, the search can be restricted to a time difference around the observed map in seconds (here: 20000).

```
corr ext
```

This is the actual correction procedure. It interpolates the opacity value from the selected skydip measurements for every data point in the map by multiplying it with the calculated correction factor.

```
base l 1
```

Another baseline fit, now of higher order. We suggest not to increase the order to more than 4, and only so, if the base range is properly defined. Otherwise, the final map will contain baseline artefacts. The order chosen here also sets the order of the baseline fit in the following skynoise reduction procedure.

```
store weight rms2 az < 1e9 > -1e9
```

The weight of each channel is stored according to its noise level. This is necessary for correctly applying the skynoise reduction. The region where the noise is calculated should be kept identical to the base range.

`snf ... run`
This encloses the skynoise reduction or filtering. Options are given as follows:

`it 5 0`
This is the number of iterations for the calculation of the correlated skynoise. There are two modes of skynoise filtering that can be chosen. However, usually only the first one is taken. Therefore, we let it iterate five times, while the second mode is never carried out. The whole process converges typically after 5 to 6 iterations.

`no chk`
There is no check for a model noise flatfield. A noise flatfield can be understood as a table of the detector noise for every channel of the array. This noise is regarded as purely detector dependent, and must therefore be excluded from the calculation of the correlated noise. A model of the noise flatfield can be included into the skynoise reduction. The noise flatfield is, however, normally simply calculated from the data. Hence: No model, no check.

`> 5 rms < -5 rms`
This despiking is similar to the one at the beginning of the script. It removes spikes that were previously hidden in the noise, but show up after the noise subtraction. In order to avoid despiking, replace this line with `no > no <`.

`r 1 900`
Every bolometer channel is corrected for correlated noise. This command defines a search range for correlating channels. Given are an inner and an outer radius in arcseconds from the channel that is going to be corrected. We define an inner radius of $1''$, as we want to exclude self-correlation. The outer radius is set to a high number that exceeds the array. The correlation is investigated in the bolometer channels lying in this range relative to current channel of investigation. The inner radius must be adjusted to the source geometry to achieve optimal results. $1''$ yields the best results, but should only be applied to weak compact sources. This value should be increased to exceed the inner ring of bolometers, e.g. $50''$ for moderately strong sources. Part of their flux can also penetrate the regime of the neighbouring channels of the actual main detector. The SNF procedure would interpret this as a perfect correlation and subtract the flux measured in the adjacent channels from the main channel. This must be avoided. For strong sources, the inner radius should either be increased to $100''$ (only the outer bolometer ring) or the whole skynoise reduction should be skipped. Modifying the number of iterations can help as well. For instance, only one iteration with an inner radius of $100''$ does not even change the flux of Uranus. Weak and extended sources are very difficult to handle and are subject to experimenting with the different options. However, a setting of `r 50 900` seems to be a good start.

`best 6`
This selects only the 6 channels with the best correlation to calculate the noise to be subtracted. For a setting of `r 1 900` this usually turns out to be the inner ring of bolometers around the channel to be corrected. This number should not be increased too much, since the correlation degrades from the centre to the edge of the array just because of the shape of the telescope beam.

`llsf 1`
This selects the mode to calculate the linear least square fit to the correlated noise. It should not be changed.

`ffc avcn rc`

This defines the correction of the noise flatfield. In this example, an average over the channels is calculated (`avcn`). The correction is applied to receivers regardless of the phases (`rc`). It should not be changed.

`cn med`

This calculates the correlated noise as a median over the correlating channels selected from the previous settings. Using the median is appropriate for the skynoise reduction without any source model. Taking the average (`cn ave`) instead would produce negative dips around the source. In the easiest case, the skynoise filtering investigates the surrounding 6 channels of the channel to correct for skynoise (Fig. 2.7 left). The calculation gives reasonable results for *average* and *median*, as long as this channel hits the source. But as soon as one of these neighbouring channels is going to be corrected, at least one of its surrounding channels covers the source and the averaging overestimates the noise level and local baseline (Fig. 2.7 right). MOPSI lowers the signal of the referring channel, which in the final map shows up as an array pattern or negative dips around the source. This can be avoided with the source model (see below), which informs MOPSI about its position and strength. The calculation can be changed to the superior `cn ave` then.

Figure 2.7 Every channel is corrected for correlated skynoise by inspecting the noise level in the surrounding channels. As long as the central channel covers the source, the surrounding channels detect the correct noise level (left). When one of the neighbouring channels is going to be corrected, suddenly one of its reference channels hits the source (right). The averaging would then give wrong results for the noise, unless the position and strength of the source is known.

`cn cor`

This command actually corrects the correlated noise.

`cn cd`

This tells the procedure to calculate the correlated noise from the correlating data only. One can also choose to take all the data with `cn ad`. The difference in the result is, however, still unclear.

`cn w cfr 1 0.95 b`

This determines the weighting of the correlated noise. We propose not to touch this.

`no source`

This switches off the calculation of the source distribution, i.e. all parts of the map are regarded equally. One should use this option for the first iterations, when the shape and position of the source is unclear. Later on, one can either let MOPSI calculate the source distribution by simply omitting this line, or provide a source model with `source model.gdf`. The model can be of any file format that MOPSI understands. In that case, remember to add `sb` for single beam observations. See Sect. 2.5.10 for information about creating a source model.

`convert`
This command puts all the single channel maps together according to their noise weight and performs the transformation into equatorial coordinates. One can modify the default settings for this procedure by entering the interactive version with `convert command` and concluding this command block with `quit`. Possible modifications among others are: pixel size in arcseconds (`pix <RA> <Dec>`), map extent in arcseconds (`ext <RA> <Dec>`), projection centre relative to the 0,0-position (`proj <hh:mm:ss.s> <dd:mm:ss>`), coordinate offset of the bottom left corner in arcseconds (`blc <ΔRA> <ΔDec>`). The interactive version is also used when co-adding single maps.

2.5.6 Command Hierarchy

The main programme MOPSI consists of input items, which I call commands, sequences and subprogrammes. Sequences can be easily characterised as interactive commands, which asks the user for the needed parameters to be executed. Sometimes, such parameters can also be given as options, which avoids the interactive character and lets sequences act like commands. The sequences *integrate* and *rms* are given as two examples for this.

Integrate is a sequence, which allows a flux measurement within a map. After initiating, it asks for parameters like the integration area and calibration factors.

```
MI_map> integrate

  INTG calculates the flux between two planes MINPLANE and MAXPLANE as:
       FLUX = (V-Vneg)*FACTOR / OMEGA
  within a box or a contour. Vneg is the volume of negative pixels within
  the integration area.
  OMEGA = HPBWx*HPBWy*pi/(4*log(2)) for main beam calibration
          pixel's source angle for pixel calibration
INTG> Define the integration boundary: box or contour ? box
INTG> Map dimension (cols x rows):     203 x     201
      Define the integration subfield (col. and row of BLC & TRC,
      0 0 0 0 = whole map): 0 0 0 0
      Subfield: 1 1 203 201
INTG> Enter MINPLANE and MAXPLANE intensities in counts,
      e.g. 0 &  0.0000000E+00 cnts: 0 1e9
INTG> Enter multiplicative cal. FACTOR (cnts to mJy) and KEY-WORD
      (b=cal. per. beam, p=per pixel): 1 p
INTG> Integration areas: (+) 6992, (-) 2675 pixels
      INTG in cnts 1.4823690461055E+03 +- 2.7606810377605E+02
      Using 1.00000E+00 mJy/cnts Flux =  1.4823690E+00 +- 2.7606810E-01 Jy
INTG> Enter bound, subfield, planes, factor, quit: quit
MI_map>
```

The sequence *rms*, which calculates the noise, has a mixed nature. It can act as a genuine sequence, but is often used as a command.

```
MI_map> rms
NOISE> Enter number of areas to be processed: 1
NOISE> Specify cols and rows of BLC & TRC of area: 0 0 0 0
```

```
NOISE> in area with BLC    1    1 TRC   29   31 pixels:
          rms 1.601897E-01 mean  7.068565E-02 sum  6.35464E+01  pixels      899
MI_map>
```

or:

```
MI_map> rms 0 0 0 0
MI_map> in area with BLC    1    1 TRC   29   31 pixels:
          rms 1.601897E-01 mean  7.068565E-02 sum  6.35464E+01  pixels      899
```

Sub-programmes, however, are separated parts of the whole MOPSI package, which fulfil special tasks with their own set of new and sometimes redefined commands and sequences. A typical representative of this class is *combine*, which is generally used to co-add single IR images. Note the change of the prompt, when entering a sub-programme.

```
MI_map> combine

COMBINE> Coordinate system for the output cube is EQUATORIAL
         with projection centre at
         RADEC50=  260.2224993  -35.7836110 ( 17 20 53.39982 -35 47  0.9996)
         and corresponds to
         LII/BII=  351.7652120    0.0590333 (351 45 54.7631   0  3 32.5199)
COMBINE> ext

COMBINE> COMBINE> Extent is  1392. x  1392. arc sec
COMBINE> blc

 COMBINE> Current offsets of BLC are  696. -696. arc sec
         they correspond to
         RADEC50=  260.4614007  -35.9767086 ( 17 21 50.73618 -35 58 36.1511)
COMBINE> don weigh
COMBINE> run
         The output cube requires 30625 pixels ( 175 x 175 x 1) + 100
cmbrun> run calibr findmxpix nrsearch nrcal chkmxint  mnint         mxint
         0    F       F         0       0       F     0.000000E+00  0.000000E+00
RETAB11> SIGNALS and WEIGHTS initialized
COMBINE> combine 17175_2.nod
cmbrun> run calibr findmxpix nrsearch nrcal chkmxint  mnint         mxint
         1    F       F         0       0       F     0.000000E+00  0.000000E+00
CMBREA> 17175_2.nod
GETNROBJ> 17175_2.nod has #   2 in the input list
        Title:
COMBINE> quit
COMBINE> Output cube calculated using    2 cubes

MI_map>
```

2.5.7 Command List

MOPSI has a large variety of commands, sequences and sub-programmes, although usually only a fraction of them is really necessary to reduce bolometer maps properly. A complete list is available by typing **help**. MOPSI is not case sensitive, when entering commands. Furthermore, command options, which are initiated with a slash "/" within the GILDAS software packages, can be omitted in MOPSI. However, when GREG commands are entered with a trailing *sic* or *greg*, the original GILDAS syntax has to be respected. Since the command parser within MOPSI usually respects only the first three letters of a command, abbreviations are very common. The user gets quickly accustomed to reduce his commands to a minimum.

```
MOPSI> help
MOPSI commands (version 12.02.2002, R.W. Zylka)
!            clear        find         lut          precession   SNF
@            close        fits or mt   map          procedure    sort
? or help    Coldens      flip         mask         projection   SPECTRA
+ or add     COMBINE      for          mean         put          spr
- or subtr   compare      gfits        mkdir        random       sqrt
* or mul     continue     greg         modify       rcg or ff    statistics
/ or divide  CONVERT      grid         Montecarlo   rco          store
^ or **      coordinates  gsmooth      mpr          rcp          STRIP
> or higher  copy or cp   Gsources     mt or fits   read         Subcube
< or lower   correct      gtvl         mul or *     recover      Subfield
= or equal   CORRELATE    h            msmooth      reexecute    subtract
accumulate   csmooth      ha           neglect      replace      sum
add or +     dbcorrect    help or ?    news         REPROJECT    system
AE_OBS       dbfilter     higher or >  next         restore      take
ALIGN        deconvolve   histogram    no           reverse      tau
append       delete       history      NOD or nod   rewind       telescope
apply        derotate     hpbw         noise        rg           td
atd          device       in-dir=i-d   norm         rgd          title
AVERAGE      directory    initialize   observatory  rgm          transpose
background   draw         INTEGRATE    old          rms          trend_surf
bad          divide or /  Interpolate  oo or o-o    rotate       tsmooth
base         EDIT         j            open         royalties    version
bin          equal or =   k            out-dir=o-d  save         wedge
box          example      let          outside      scan         within
bsmooth      exclude      levels       pause        seeing       write
btd          exit         LFITS        pen          select       xinc
calculate    expand       limits       phase        set          yinc
calibrate    exponent     list         place        shift        zpmag
cd           extremes     load         PLAIT        show
cdeconvolve  extract      log          PLANETS      sic
change       ff or rcg    lower or <   plot         simulate
check        fft          Lsources     polygon      SKYDIP
centre       file         lst          power_sp     smooth

Commands written in capitals: subprograms with its own language;
```

```
with first capital letter: old subprograms which ask questions.

HELP> Enter a command name, <CR> or quit to exit:
```

The number of options for any of these command items can be enormous and is not seldom dazzling. A list of options can be displayed with **help** <**command**> or by entering the command, if you already inside the help environment. An explanation is, however, very often not given. MOPSI provides examples for the use of some of the commands by typing **example**.

2.5.8 Data Display

Plotting the data can be done within MOPSI itself or by using GREG routines. It is generally advised to use MOPSI's graphical capabilities for data analysis and create nice publishable drawings with GREG. Both possibilities are described in this section.

PLOT

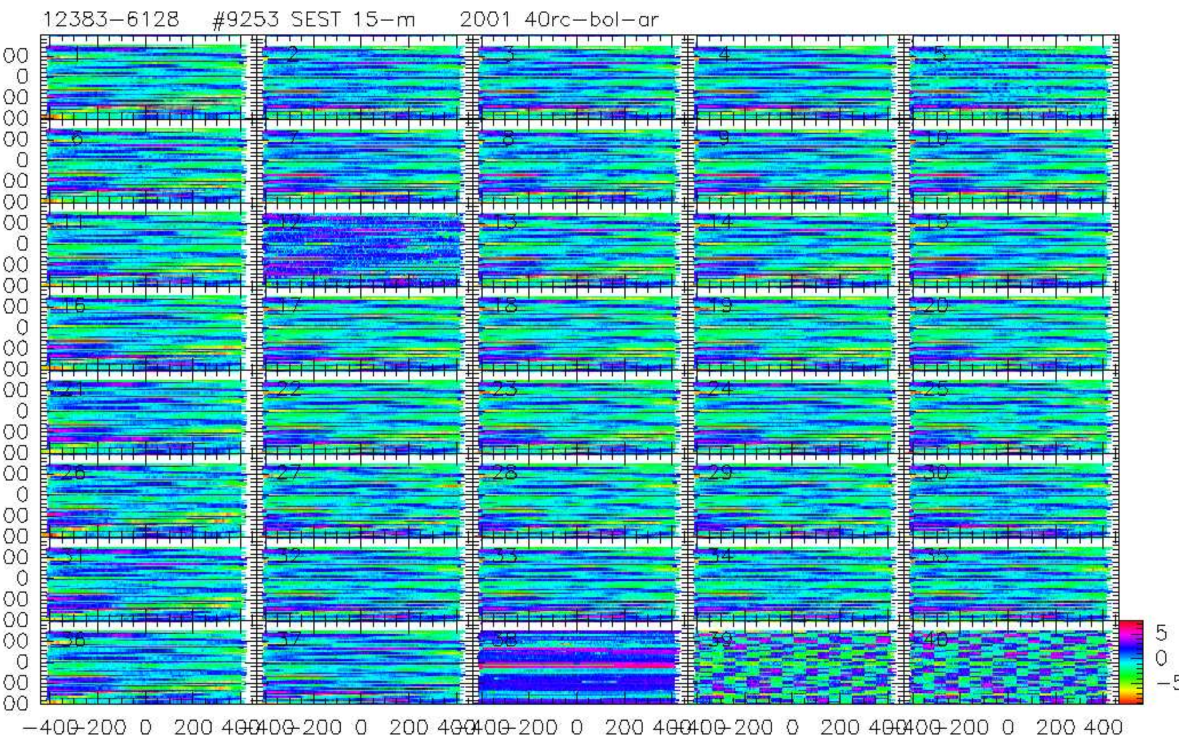


Figure 2.8 Azimuth-Elevation maps of all channels, which are read by MOPSI. See how differently the DC-biased and DC-coupled channels 38 to 40 and the noisy channels 2 and 12 behave from all the others. The character size, which is too large in this example, can be changed with **set char** <**size**>.

This is definitely the most commonly used command to display data in many different ways. Apart from the options already known from GREG like scaling and cut levels, the MOPSI version offers a much larger variety of possibilities.

```
MI_map> plot scal lin 0 35
```

Displays the current data with a linear scaling between the cut-levels 0 and 35. Note, that MOPSI remembers always the last scaling option. Therefore, every further *plot* command would be carried out with linear scaling until it is changed, e.g to logarithmic scaling.

```
MI_map> plot scal log 1 35
```

Cut levels are only recognised, when a scaling is given. Otherwise they will be set automatically. This can lead to empty plots, if the data contains negative values and the scaling is set to logarithmic. MOPSI will also complain about this. This behaviour originates from the GILDAS graphical library.

```
MI_map> plot
 Projection AZIMUTHAL     at angle     0.000000 from center
    Ra   11:12:59.500           Dec  -60:59:55.00
    Lii   291.625383             Bii    -0.531369
F-GTI_BITMAP,  LowCut is negative!
MI_map>
```

Since MOPSI can handle data cubes, the plot command not only displays single maps, but also e.g. az-ele maps of the single channels, similar to the *map* command in NIC after *makplan* (see Fig. 2.8).

```
MI_map> plo rc
```

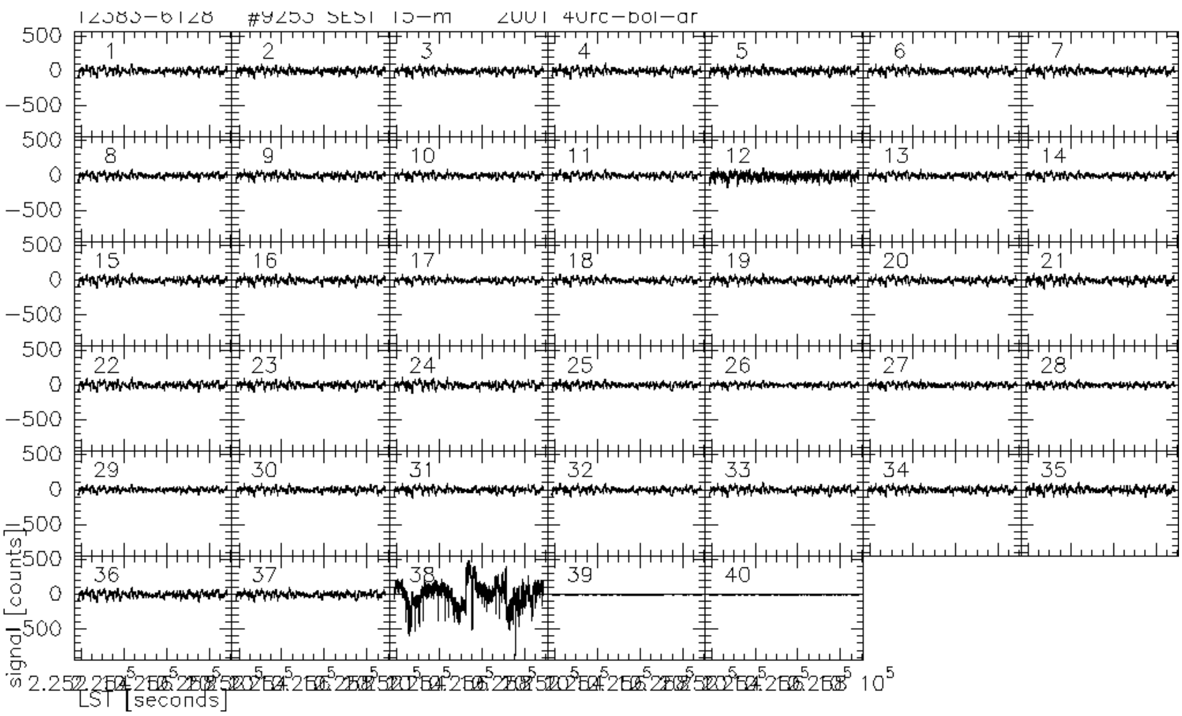


Figure 2.9 Time series of the data, which are shown as maps in Fig. 2.8.

Gives the time series of the data in every channel, similar to the *signal* command in NIC (see Fig. 2.9). It is also possible to restrict the display to smaller number of channels by adding the starting and ending channel number, e.g. **plot rc 5 24**. One single channel can be displayed, if only one number is mentioned.

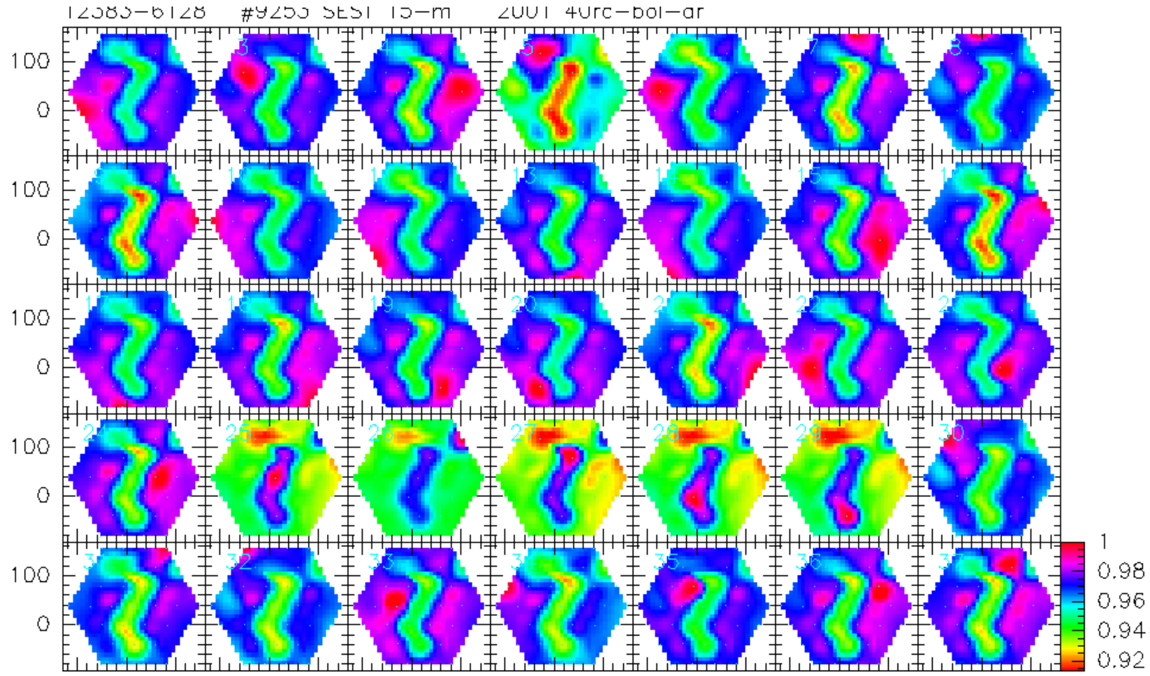


Figure 2.10 Correlation plot of the 37 bolometer channels. It is very diagnostic in detecting an imperfect decoupling of the channels, which are also used in AC-bias mode.

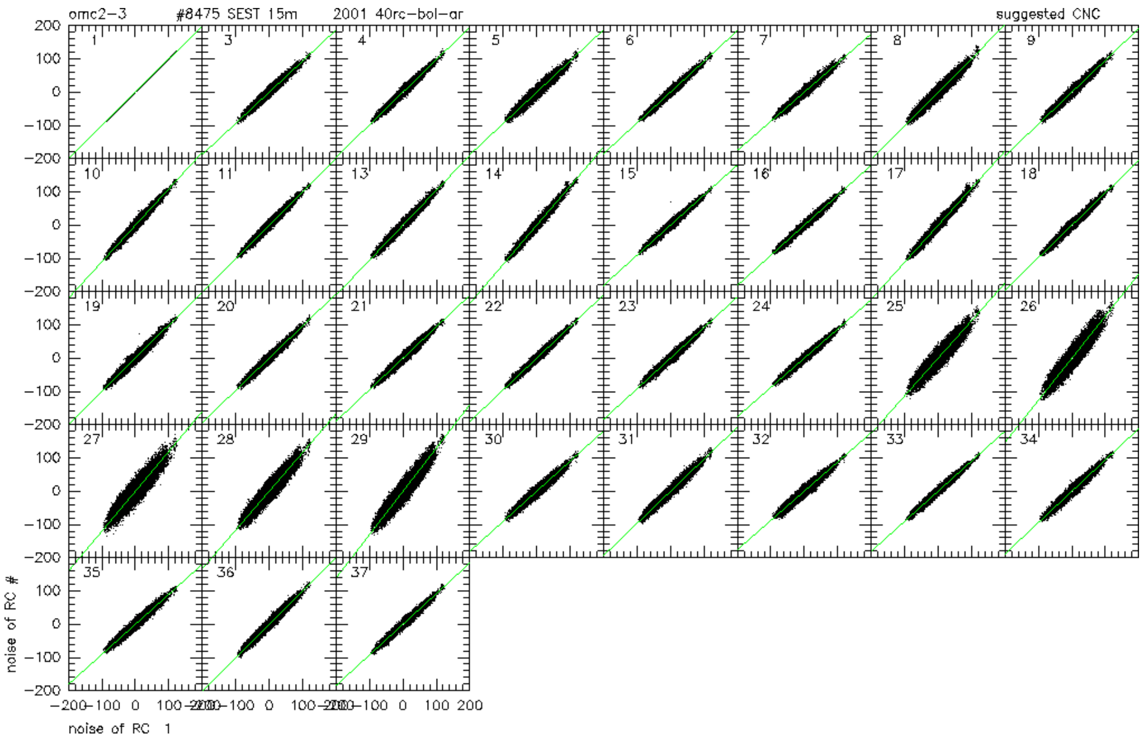


Figure 2.11 Noise correlation of all bolometer channels with channel 1 as a reference. Note that the distribution of the data points should be spherical for totally uncorrelated noise.

```
MI_map> plo corr
```

This is a very intriguing derivate of the *plot* command. It plots the correlation of every channel as a colour coded map of the array (see Fig. 2.10). It reveals the influence of the correlated skynoise as well as the coupling of the different channels. In this case, one can clearly see that obviously the channels, which are also used in AC-bias mode, are not totally decoupled in DC-bias mode.

Another possibility to plot the correlation between the channels is

```
MI_map> plo ff
```

(plot flatfield). As shown in Fig. 2.11, is displays the noise correlation of all channels to one reference channel, which be default is channel 1. Additionally, a linear least square fit is carried out for each receiver. This can be switched off with the option *lf -1*.

RGM

This command is used for creating contour plots. The contour levels have to be defined with the **levels** command.

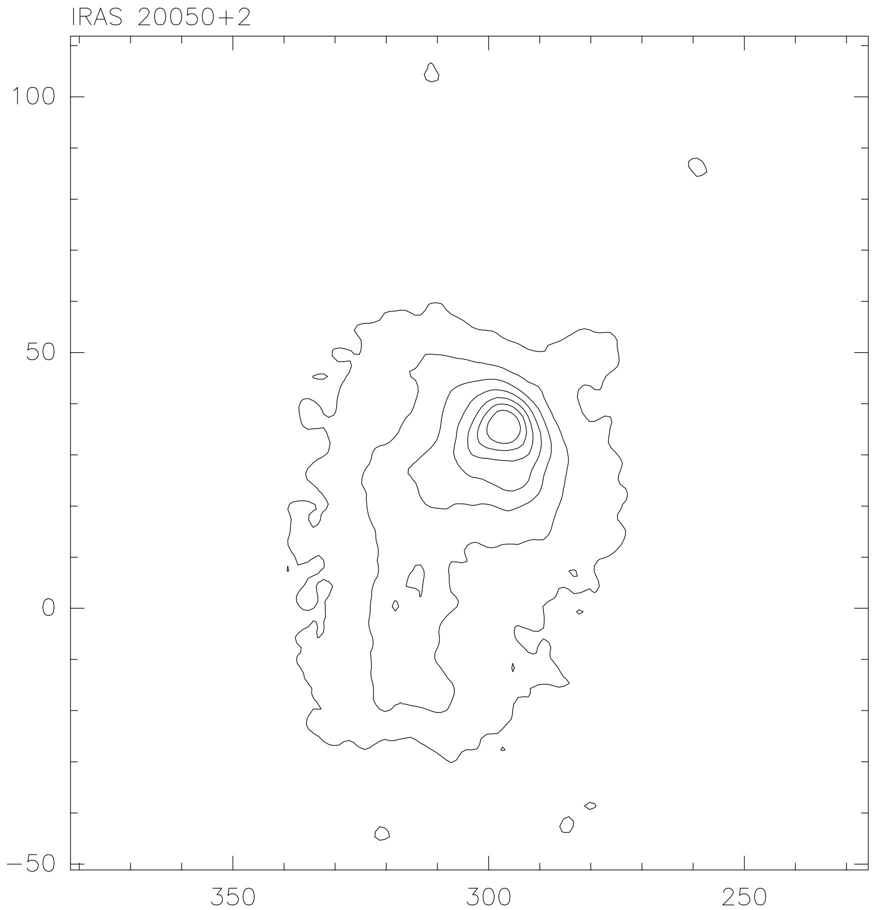


Figure 2.12 Contour plots are done with the GREG command rgm, after defining the contour levels first.

```
MI_map> lev 0.02 to 0.15 by 0.02
I-LEVELS,  Contour levels are :
    2.0000E-02    4.0000E-02    6.0000E-02    8.0000E-02    0.1000
    0.1200        0.1400
MI_map> rgm
 Projection AZIMUTHAL     at angle     0.000000 from center
    Ra   20:05:02.500           Dec   27:20:09.00
    Lii    65.780197              Bii    -2.612235
I-RGMAP,  Contouring checks the blanking value -1.000000E+08
I-RGMAP,  Pens  0 & 15 used for positive & negative levels
I-RGMAP,  Contour   1  2.000000E-02
I-RGMAP,  Contour   2  4.000000E-02
I-RGMAP,  Contour   3  6.000000E-02
I-RGMAP,  Contour   4  8.000000E-02
I-RGMAP,  Contour   5  0.100000
I-RGMAP,  Contour   6  0.120000
I-RGMAP,  Contour   7  0.140000
```

USING GENUINE GREG

In order to use the graphical command-set of the GreG package within GILDAS, the data has to be transferred to the GreG memory first. Furthermore, GreG uses coordinates in radians, while MOPSI is able to process any coordinate format, naturally arcseconds. This must be changed as well. The data is finally transferred to GreG with the **put** command.

```
extremes rad
put
sic lim /rgd
sic set box match
```

Any further graphical display in MOPSI can now be made with the genuine GreG command set by putting **sic** in front of any command line. This is particularly interesting for plotting the data with absolute coordinates. Please consult the online help function for further explanations by typing `sic help <command>`.

```
sic tick 1 5 15 60
sic plot /scal lin 0 100
sic box P P /abs
```

2.5.9 Saving Data

MOPSI provides storage capabilities, which allow to save data to both the memory and files. Especially putting the current data into a backup buffer is very convenient for testing some reduction procedures, without having to start the whole reduction process from the very beginning. In total, MOPSI has several buffers for different tasks. But the most important one is the backup buffer, which is the default. However, caution is advised, because it is used during the skynoise filtering. The spike-pointer buffer, which is used in the quick-look data reduction routine, contains blanked data, which surpass a predefined value for spikes or glitches. This buffer must be initialised (**init spike**) for every new data set.

The commands **store**, **store backup** and **store 1** are synonymous for putting the current data into the backup buffer. Mathematical operations like adding, subtracting, multiplying and dividing are also possible with it. The contents of the buffers are brought back into the working memory by the **take** command. The backup buffer again has exception, because **old** is identical to **take backup** and **take 1**. All buffers can be cleaned with **init <buffer>**.

Saving data as files can be achieved in several ways. The global command for doing so is **file out <filename>.ext**, followed by **write** and **close out**. Just like for reading files, the **file** command only directs the data-handling pointers *in* and *out* to a specified filename. **write** actually saves the data. The type of data is chosen by the extension. **close out** finally re-initialises the output pointer. A simplified version of saving can be applied, if the prefix of the final filename is already given, e.g. from the original file, which has been read in before. Then **write .ext** or **write +add .ext** can be used to save the data to *file.ext* or *fileadd.ext*, respectively. Again, the data format is defined by the extension. The possibility of adding characters to the file prefix can be very handy, to e.g. save different versions of the same data set.

A final remark: MOPSI CANNOT OVERWRITE EXISTING FILES. Any attempt will result in an error message. An exception are GDFs, which are created by GILDAS.

2.5.10 Sophisticated Data Reduction

The following sections demonstrate advanced procedures of the data reduction in two or three iterative steps. However, it is a good idea to reduce the data with the on-line script first, in order to decide on the appropriate steps. The on-line reduction procedure of Sect. 2.5.4 is used as a template. Therefore, it is assumed that the reader is already familiar with it. The three steps comprise:

1. a proper baseline definition
2. the creation of a source model
3. the final map

The whole reduction process must be modified, depending on the strength of the source, the extension of the low level emission and the source coverage. The changes mainly take place within the skynoise filtering and the baseline fitting. Examples for a weak compact source, a moderately strong compact source, a very strong compact source and a faint and extended source are given. We assume the case of a single map. For multiple maps please consult Sects. 2.5.15 and 2.5.16.

2.5.11 The Weak Compact Source

After reducing the data with the on-line procedure first, the map usually suffers from baseline artefacts close to the source, which are removed in the second iteration. In order to do so, a region for the baseline subtraction must be defined with the `polygon` command, matching the source geometry as shown in Fig. 2.13. Save the polygon to the disk with `write pol <name>`. A file called **<name>.pol** will be created, containing the pixel coordinates of the polygon vertices relative to the map. Note that MOPSI cannot overwrite files. It is always a good idea to label them with the scan number, e.g. `0946_base.pol`. Then modify the reduction procedure accordingly.

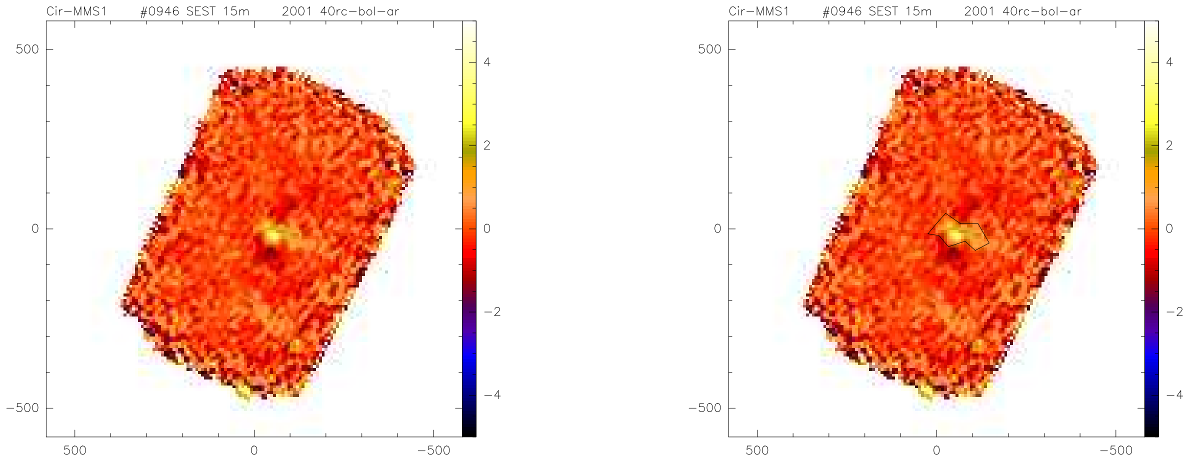


Figure 2.13 Reduction result after first iteration. The left image clearly shows baseline artefacts in scan direction close to the source. A new region for the baseline subtraction must be defined, as shown in the right image.

Replace

```
base r az < 1e9 > -1e9
```

with

```
pol 0946_base
base range out sys eq
```

and

```
store weight rms2 az < 1e9 > -1e9
```

with

```
store weight rms2 out sys eq
```

The option `sys eq` is needed, since the polygon has been defined in equatorial coordinates, while the channel maps are in horizontal coordinates. The execution of these modifications concludes the second iteration of the reduction process. Fig. 2.14 shows the improvement, which could be achieved.

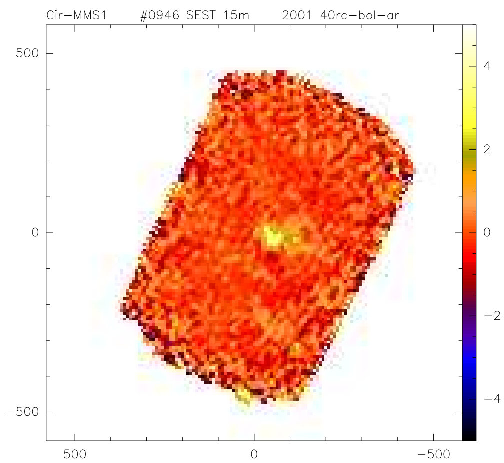


Figure 2.14 Reduction result after second iteration. The baseline artefacts have disappeared.

The final refinery can be achieved by adding a source model to the reduction process. The model takes care of differing noise and intensity levels between the sky and the source, and is included during the skynoise filtering. It can help suppressing the noise properly and supports the correct calculation of the correlated noise contribution. Its creation is simple. Just take the map of the recent reduction iteration and set everything but the source to zero (see Fig. 2.15). Draw a polygon not too close around the source and then use the command `neglect out`. Then save the image; any data format is appropriate.

```
file out 0946_model.gdf
write
close out
```

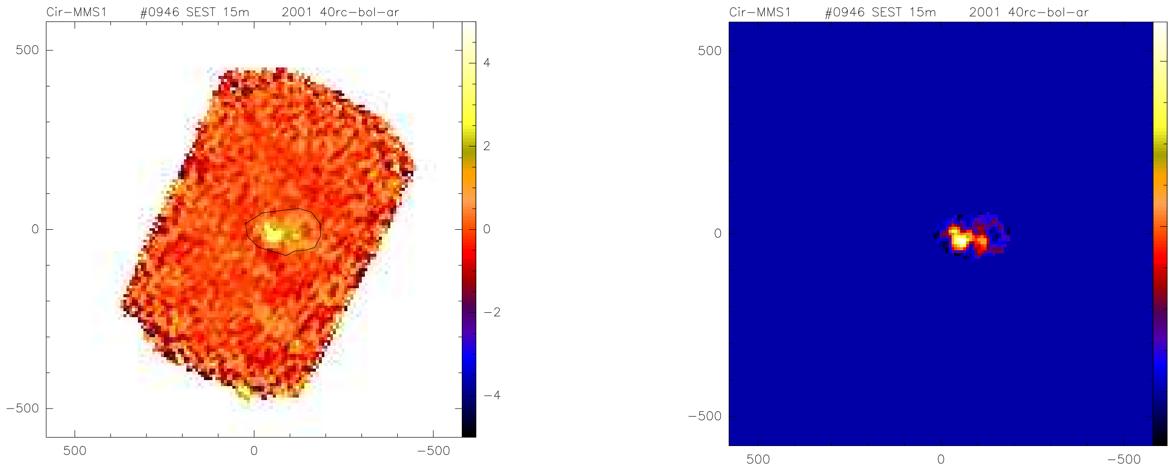


Figure 2.15 Creation of the source model. The left image shows a typical polygon, which is not too tight around the source. The `neglect out` command sets everything around this polygon to zero, as shown in the right image.

With the implementation of the source model, the calculation of the correlated contribution to the noise is changed (option `cn med` = median) to *average*, which yields better results.
Replace

```
cn med
```

with

```
cn av
```

and replace

```
no source
```

with

```
source 0946_model.gdf
sb
```

The option `sb` tells the skynoise filtering, that the data is given as single beam map, because it is not chopped. Start the last iteration and enjoy the look of the reduced map (Fig. 2.16).

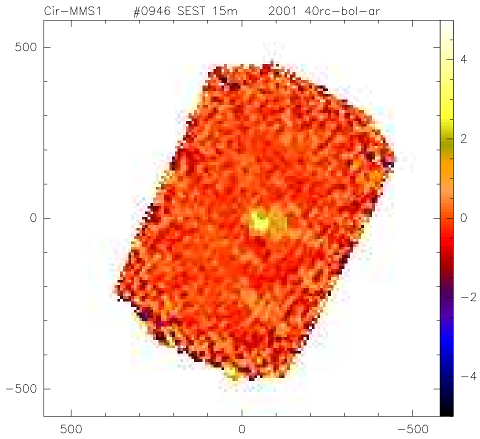


Figure 2.16 Final map after three iterations.

2.5.12 The Moderately Strong Compact Source

The necessary modifications affect the despiking and the skynoise reduction. The despiking can clip off the peaks of strong sources. As a rule of thumb, despiking should not be applied, if the source is already visible in the single channel maps of the raw data. One can, however, modify the spike threshold of 5 times the rms noise to a level, where the source is not affected anymore. The despiking in the skynoise filtering can be turned off by replacing `> 5 rms < -5 rms` with `no < no >`.

Improper settings in the skynoise reduction can reduce the intensity of a source. The search radius for correlating channels is particularly important, since the edges of the intensity distribution of strong sources can also penetrate neighbouring channels and simulate a perfect correlation (see Fig. 2.17), which will be subtracted from the main detecting channel.

Figure 2.17 Usually, the S/N ratio of weak sources is so low that the edges of the intensity distribution fade out into the noise before they reach neighbouring channels. The source is only detected in the central channel (left image). Stronger sources, however, can contribute to the detection in adjacent channels simulating a perfect correlation (right image).

This can be avoided by increasing the inner radius in `r 1 900` to $50''$, which would exclude the inner ring of bolometers around the main detecting channel. As a side effect, the efficiency of the skynoise reduction is diminished, which results in a higher residual noise.

2.5.13 The Strong Compact Source

The intensity distribution of very strong sources (e.g. Uranus) can even penetrate the second ring of bolometers. As explained later for the calibration (Sect. 2.6.2), we therefore suggest to reduce these data without any skynoise reduction at all. If a skynoise reduction is desired anyway, one should increase the inner radius up to 100″, which leaves only the outer ring of the bolometer array. Additionally, we propose to reduce the number of iterations in `it 5 0`. We have discovered that the intensity of Uranus is not affected when using `it 1 0` and `r 100 900`.

2.5.14 The Weak Extended Source

This class of sources is probably the most challenging one in terms of data reduction. We suggest to be highly cautious with the baseline fits as well as the skynoise reduction, here especially the search radius `r 1 900`. We suggest to use baseline fits without any curvature, i.e. orders < 2, as they can unintentionally remove any faint emission feature. Furthermore, extended emission can simulate a correlation in the detecting bolometer channels. Therefore, it is difficult to make any suggestions, but starting values of `base 1 0` and `r 50 900` usually give satisfactory results.

2.5.15 Combining Maps – Single Position

Multiple maps can be co-added to achieve a better signal-to-noise ratio or/and cover a larger area, i.e. when creating a mosaic (see Sect. 2.5.16). This is possible with the *combine* routine, but can only be used for infrared data. For millimetre data, we propose the interactive version of the *convert* command, which processes the reduced data, which are present as reduced single channel maps. This means, the whole data reduction should be carried out, but final maps should not be created before all the data is going to be co-added. They should be saved to disk as **btd files** instead (`wr .btd`). One can reduce each FITS file separately, create a btd-file and co-add them afterwards. In order to do so, create a file list of all btd-files (`ls *.btd > btd.LIST` outside MOPSI), read the list and just combine all of them:

```
list btd
read 1
convert command
run all
quit
```

Both the reduction and the co-adding can be included in one complete batch processing file, too. A loop over the list containing all FITS files must be defined for that.

```
init history
init gfit

! clear file lists
init lis
init out-l

! load file list
lis fits

! reduce all files in a loop
for i 1 to last
read i
del dc
init spike
```

```
! put the reduction without "convert" here
! as already explained for single maps

! write channel maps to disk and close the loop
wr .btd
next

! declare saved files as new input list
replace in-l out-l

! load the first data file as a reference
read 1

! co-adding, define map extent and bottom left corner (BLC) in arcsec,
! if the final map is larger than just one single map
convert command
run all
quit
```

The new commands are explained as follows:

`for i 1 to last ... next`
This defines a loop with the running index i, which selects all entries from the input list.

`read i`
This reads the FITS file with the number i in the input list.

`replace in-l out-l`
This prepares the co-adding by defining the output list containing all saved btd-files as the new input list.

`convert command ... quit`
This is the interactive version of the `convert` command.

`run all`
This combines all the single observations and creates the final map.

2.5.16 Combining Maps – Mosaic

The difference between a mosaic and the co-adding on a single position is that the single maps have usually different centre coordinates. This must be kept in mind, when defining base ranges for baseline fitting and tailoring the extent of the final map. The creation of a polygon as described in Sect. 2.5.10 is always relative to the projection centre, i.e. the 0,0-position of the underlying map. Such a polygon defined within the mosaic can even be larger than a single map. It is therefore necessary for the correct base range definition to indicate the projection centre to which the polygon has been created. This can be achieved with an additional option of the `base range` command.

```
pol mosaic
base r sys eq pro 10:20:30 -20:30:40
```

Another issue is the size of the final mosaic. By default, the final map will have the dimensions of the first map during the co-adding. This can be changed with additional parameters within the `convert command` environment. `ext <RA> <Dec>` sets the map extent in arcseconds with the 0,0-position of the first map of the conversion procedure. There are generally two ways of shifting the mosaic so that it fits into the selected size. One can either specify the projection centre of the mosaic (`proj <hh:mm:ss.s> <dd:mm:ss>`) or define the offset of the bottom left corner in arcseconds relative to the new 0,0-position (`blc <ΔRA> <ΔDec>`).

```
convert command
ext 2000 2000
proj 10:20:30 -20:30:40
run all
quit
```

or

```
convert command
ext 2000 2000
blc 1000 -1000
run all
quit
```

2.6 Calibration

The transformation from counts into physical quantities (e.g. Jansky) can be achieved by a measurement of a source with known intensity. Planets are usually a good choice, but secondary calibrators like quasars are also quite common. The calibration can be simplified when observing point-like objects. We therefore suggest to use Uranus or Mars. The calibration factor can be derived both from on-off measurements and maps. The observing mode should however kept the same as for the scientific data, since on-offs and maps are fundamentally different and sensitivity inconsistencies cannot be ruled out.

2.6.1 The Flux Density of the Calibrator

The use of secondary calibrators like quasars is tempting, since their constant distance implies also a fixed intensity. They are, however, unreliable, since quasars are known to be variable.

The calibration with planets, on the other hand, is slightly more complicated, as their distance varies with time, which affects the source strength. Their intrinsic emission, however, is known to be fairly stable and fundamentally dependent on the surface temperature, which can be derived from well established models. In rare cases though, weather phenomena on these calibration planets (e.g. sandstorms on Mars) can lead to wrong flux estimates.

The sub-programme PLANET in MOPSI is able to provide the appropriate temperatures and derive the desired flux values by applying models for the various planet atmospheres. The user only has to insert the topocentric distance in AU for the selected planets. This value can be interpolated from tables E14 – E42 of *The Astronomical Almanac*, which is available at the SEST. Further parameters (telescope, time of observation, frequency, beam size) are automatically added, if a data file has already been loaded.

```
MI_map> planet
PLAN> Which telescope ? (SEST 15m   )
SETDT> Local date [& time] (30-Jul-2002 16:46:02):
PLAN_REACHEB> The variable MOPSI_EPH is not defined
PLAN_REACHEB> Can't open ${MOPSI_EPH}/02.asc
              You have to specify the topocentric distance !
PLAN> Frequency in GHz (  250.0):
PLAN> Beam width (HPBW) in arc sec(   24.00):
PLAN> Which planet ? (all) uranus
PLAN> Uranus  topocentric distance in a.u. (19.799620): 19.052679
PLAN> Uranus  temperature in K ( 93.0):
PLAN> frequency  250.GHz       HPBW  24.arcsec
PLAN> planet  apparent  apparent   toopocentr. temp. flux dens. flux dens.
               diam.   solid angle  diistance         /beam      total
              [arcsec] [arcsec**2]   [a.u.]    [K]  [Jy/beam]     [Jy]
      Uranus    3.676     10.614     19.053    93.0    41.400      41.738
PLAN> exit
MI_map>
```

The relevant value is `flux dens./beam`, which is 41400 mJy/beam for the given date and time.

2.6.2 The System Intensity (Counts) of the Calibrator

If the calibrator is measured in on-off mode, the resulting value is already – by definition – in counts per beam. The calibration factor (in mJy per count and beam) can therefore be easily derived by dividing the physical flux density by the measured counts.

Calibrator maps should be reduced basically as usual, but **low order baseline fits are highly recommended** and **no despiking should be performed**. We additionally strongly suggest to **abstain from any skynoise reduction**. We therefore propose to create a special reduction script for calibrators, which only uses low order baselines and does neither include despiking nor the `snf` section. The resulting map will be very noisy, but this is not important for calibration purposes. Only in the unlikely case of a non-detection, a very weak skynoise reduction may be necessary. The `snf` section should then be edited as follows:

```
snf
 no chk
 no > no  <
 r 100 900
 best 6
 llsf 1
 ff avcn rc
 cn med
 cn cor
 cn cd
 cn w cfr 1 0.95 b
 no source
run
```

The system intensity of the calibrator is identical with its peak flux in the resulting map. It can be easily extracted with a Gaussian fit. The command `gfit cursor` activates the mouse cursor in the graphical window. The fit is carried out by clicking on the source. Typing `e` extis the fitting mode.

```
MI_map> gfit curs
GFIT> Fits in subf. with BLC:    77    46 TRC:    89    58 iter   5
      R.A.  23 59 59.974 Dec.   0  0 -0.77 corresponds to
       83.0483017 columns  51.9031715 rows    from BLC 1 1
      -0.386432081"  -0.774625242" from centre
  +/-  0.0795147046   0.0694560632
      FWHM   26.32" x  22.62" heigh     450.33 base     2.84 PA -76.73
  +/-        0.168"    0.186"           3.226         0.465     0.578
MI_map>
```

The height of the fitted intensity distribution (`heigh 450.33`) is the relevant value (in counts/beam). The calibration factor would then be the quotient of the physical flux density and the measured counts.

2.6.3 Errorbeam Correction

Like for every real antenna, a part of the intensity is distributed into side-lobes, and are therfore not contained in the main image of the source. As a result, the measured flux (counts) is always slightly smaller than the theoretical intensity, thereby overestimating the calibration factor. For the SEST, however, this is only a marginal and negligable effect well inside the normal observational errors. For strong sources like Uranus, this effect is only visible at intensity levels of $<$5% of the peak intensity, and closely distributed around the main beam. A correction is therefore not necessary.

If a correction is desired anyhow, please proceed as follows. First we define a beam filling factor $f = S_{tot}/S_{beam}$ (see MOPSI output in Sect. 2.6.1). Then integrate over the whole source (see Sect. 2.7). Use `1000 b` as a calibration factor during the integration procedure to get the final result in counts (given in Jy). Divide the total source flux by this value and the beam filling factor f to get the corrected calibration factor.

2.7 Photometry

MOPSI has not only been created for data reduction, but also for data analysis. This section gives a short introduction to deriving fluxes from the maps. At this point, it is not important, if the reduction has been carried out with NIC, MOPSI or any other software. MOPSI can read several different graphic formats. The native format for NIC files is **gdf**. Isolated point-sources are analysed fairly easy by aperture photometry, and an example for this is given in the current section. It is getting much more complicated, if such a source has been detected on top of an emission ridge or the source is extended. However, the technique would be similar. The difficulty would lie in defining proper baselines or matching apertures.

First of all, the map must be read. The general commands to do this are:

```
MI_map> file in source.gdf
MI_map> read
```

A typical map is shown in Fig. 2.18.

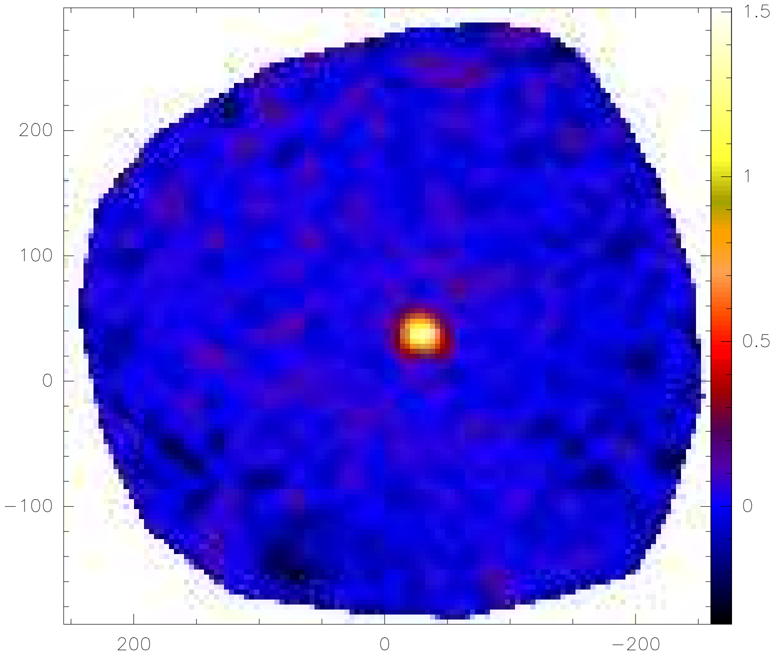


Figure 2.18 Typical millimetre map, not centred on the source.

In order to define an aperture, it is easier to have the source in the map centre, especially when the whole procedure is going to be automised with a script. Recentring can be achieved by applying a Gaussian fit to the source and the redefining the map centre to this fit. One should empty the buffer containing the previous fits first. The Gaussian is fitted with the **gfit** command.

```
MI_map> init gfit
MI_map> gfit curs
```

This implies, that MOPSI already knows the beam size of the map. It can be defined with:

```
MI_map> hpbw <beam>
```

On the other hand, it is also possible to define the fitting area in the command itself.

```
MI_map> init gfit
MI_map> gfit curs field <size>
```

The values should be given in the angular quantities of the map, which is usually arcseconds. The fit itself is carried out by clicking on the source in the graphical window. The fitting mode can be exited by typing **e** in the graphical window. The text window gives the fitting parameters like absolute coordinates, peak and base of the Gaussian. The map centre can be aligned to this fit by:

```
MI_map> coord ref gf
```

If there are more than one Gaussian fits in the buffer, the number of the fit must be appended. The complete list of all resident Gaussian fits can be displayed with **show gfit**. At the same time, the fits are plotted within the graphical window on top of the current plot. Afterwards, the baseline of the map should be redefined. The easiest way to do this, is to define a polygon omitting the source itself and fitting a new 2-dim baseline to the included area, as shown in Fig. 2.19.

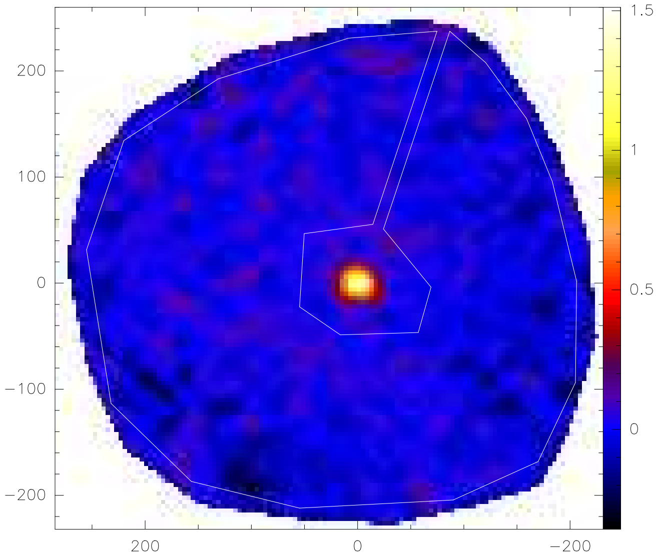


Figure 2.19 A polygon has been applied to the recentred map, omitting the source itself. A new baseline is fitted according to the included area.

```
MI_map> base range in
MI_map> base s 0
```

The command **base s 0** means that a zero order surface is subtracted from the map, in order the assure a flat baseline for the photometry. The next step is to define an aperture. The size itself depends, of course, on the source extension. A radius of 60″ has been chosen in this example (see Fig. 2.20). Smaller radii can be applied, but should never fall below the beamsize.

```
MI_map> mask out 60
```

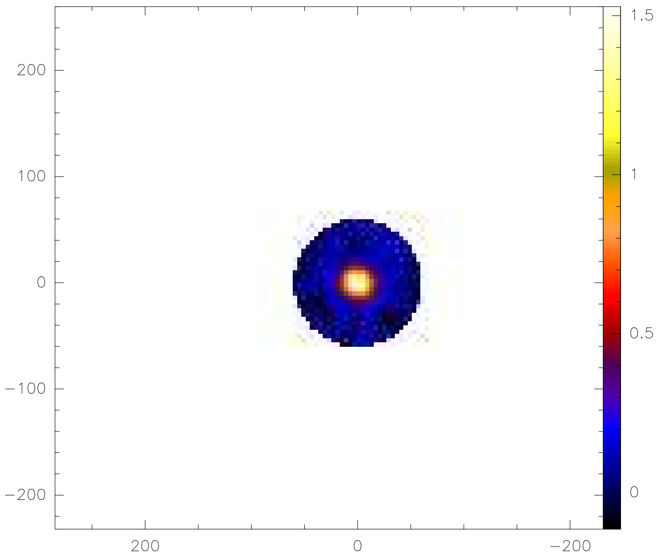


Figure 2.20 An aperture of 60″ in radius has been chosen for the photometry.

Elliptical apertures are also possible. The command must be altered to **mask out** <**horizontal half axis**> <**vertical half axis**>. Any other arbitrary apertures should be defined with a polygon and **mask out**.

The photometry itself is performed with the **integrate** procedure. It asks interactively for the different parameters. Nevertheless, they can also be written to a MOPSI script, in order to automise the measurement.

```
MI_map> integrate

  INTG calculates the flux between two planes MINPLANE and MAXPLANE as:
       FLUX = (V-Vneg)*FACTOR / OMEGA
  within a box or a contour. Vneg is the volume of negative pixels within
  the integration area.
  OMEGA = HPBWx*HPBWy*pi/(4*log(2)) for main beam calibration
          pixel's source angle for pixel calibration

INTG> Define the integration boundary: box or contour ? box
```

MOPSI wants to know, how the integration boundary should be defined. It can be set according to intensity margins (contour) or spatial extent (box), which is usually the best choice.

```
INTG> Map dimension (nx * ny):     129 *     123
      Define the integration subfield (col. and row of BLC & TRC,
      0 0 0 0 = whole map): 0 0 0 0
```

Since the map has already been correctly prepared and cut, it can be completely taken for the measurement.

```
      Subfield: 1 1 129 123
INTG> Enter MINPLANE and MAXPLANE intensities in counts,
      e.g. 0 &  0.0000000E+00 cnts: 0 1e10
```

All positive values should be integrated. Therefore, any range from 0 to a very high number is appropriate.

```
INTG> Enter multiplicative cal. FACTOR (cnts to mJy) and KEY-WORD
      (b=cal. per. beam, p=per pixel): 1000 b
```

MOPSI expects the calibration factor in mJy/count. Also, it must be clarified, if the calibration factor is understood as mJy/pixel or mJy/beam. If the map has already been calibrated previously (by multiplying the map with the calibration factor), the correct value would be 1000 in the case of Jy and 1 for calibration in mJy.

```
INTG> Integration areas: (+)  10064., (-)  1296. arcsec**2
      INTG in cnts*arcsec**2 1.5469180070354E+03 +- 4.3704073416222E+01
      Using 1.00000E+03 mJy/cnts and beam 2.24000E+01 x 2.24000E+01 arcsec
      Flux =  2.7208640E+00 +- 7.6870810E-02 Jy

INTG> Enter bound, subfield, planes, factor, quit: quit
```

MOPSI gives the result in Jy with a statistical error. The user can redefine some of the parameters and repeat the measurement, or leave the integration procedure with **quit**. Instead of typing all parameters for every photometry again, it is possible to use a script, which is called e.g. `int.MOPSI` as printed below. The measurement is automatically performed by typing `@int`.

```
! photometry script
integrate
 box
 0 0 0 0
 0 1e10
 1000 b
quit
```

Note that the result is depending on the chosen aperture. Its size should be reduced until a reasonable signal-to-noise ration has been achieved. But remember to never get below a radius of the order of the beamsize.