Practice Work Session (3h)
MIDI data reduction

O. Chesneau,
Observatoire de la Côte d'Azur (OCA)

Euro Summer School
Observation and Data reduction with the Very Large Telescope Interferometer
4-16 June 2006, Goutelas (France)

Abstract

This Practice work aims at initiating beginners to the MIDI data reduction. Several independent packages can be used that provide good results. We will concentrate on the MIA (MIDI Interactive Analysis) for this introduction though several EWS commands are also shown. Note that these packages were developed in a commercial langage IDL (Interactive Data Langage).

1 Introduction

The tutorials for MIA and EWS packages can be found in internet at the following addresses: http://www.mpia-hd.mpg.de/MIDISOFT/, in section Analysis contains the MIA tutorial or in: http://www.strw.leidenuniv.nl/ koehler/MIA+EWS-Manual/ in which the MIA and EWS complete manuals can be found. A large part of this tutorial comes from the MIA website and was written by Rainier Kohler and Thorsten Ratska.

2 How to deal with this practical session

You start the MIA+EWS reduction tool by typing *IDL* in a shell.

Several compiling lines should appear :

```
Compiling fbt...
Compiling oir...
Compiling atv...
Compiling utilities...
compiling midiUtilities
compiling fringeUtilities
compiling midiFringe
Compiling MIA...
Compiling Heidelberg routines
```

Finally, you get a prompt like :

```
MIA+EWS>
```

Every MIA+EWS command will be typed here. Note you can switch to the shell by typing $ and go back to *IDL* by typing *exit*.

For the purpose of this session, all the data required are stored in $HOME/MIDIDATA and $HOME/MIDIDATA2. These data are from AB Auriga and several calibrators.

Some display glitches may happen with the small screen resolution and MIA+EWS : if you need to move a part of a window outside the screen (for example to have access to files not displayed), hold the [ALT] key and the left mouse button. You are now able to drag the windows. If needed resize the window.

3 Choosing data

3.1 Making a log of Observations

In a first step we want to look what files have beeen received. It's not easy to find tools reading binary fits-tables headers and thus MIA provides two methods. On the one hand a simple IDL procedure that is called make_midi_log. Just type at the IDL prompt

```
make\_midi\_log,'path',LOGBOOK='target/filename.log'
```

where 'path' is the directory where the downloaded files are located. The parameter LOGBOOK specifies the output file. In the default version the procedure reads the' keywords providing informations about the observed object, the observing mode, the filter, the shutter, the integration time, and the number of exposures.

Q: Log the MIDATA and MIDIDATA2 directory and read the result with your favorite text editor. To what each column is related ?. Do you understand each parameters ?

3.2 Choosing files with midigui

Another way to get that informations is "Gorgonzola":

```
  files=midigui(dir='path')
```

After a few seconds a window pops up and lists all MIDI-files in the specified directory (this screenshot has been reduced in size and readability, but note with the laptop in the school with limited resolution, this can be tricky to get all the files.).

You have to recognize the three type of data recorded:
Acquisition files: image of the source for both telescopes. Contains sequences of typically 1000 frames, containing each two windows of 69x62 pixels (about 98mas per pixels with UTs).
Tracking files: contains the dispersed fringes (a spectral dispersion is always used in MIDI). These files must be processed with dedicated MIA or EWS commands.
Dispersed photometric files ('Default Chop'): The files are the photometry of each telescope (one file per telescope) that are processed for the visibility extraction or for calibrating the spectrum and flux of the source.

You might have noticed that the number of files is reduced from twelve to eight. "**Gorgonzola**" recognizes files that belong together in one dataset and displays only the first file. If you now select one or more files by marking them in the first column and pressing the "*SELECT*" button in the upper part of the panel the names of the files are written into the array files that was specified in the calling command of "Gorgonzola".

A similar routine called with the command

```
 files=midiguis(dir='path')
```

is preferred when handling large numbers of MIDI-files. After the user selected his files, the routine writes a binary file to the directory 'path'. In this file all the displayed informations are stored, i.e. next time you start midiguis within 'path' the routine quickly reads this file instead of processing all the fits-headers again. Of course, this will not work if you don't have permission to write to the data directory.

If you know which data you want and are too impatient to wait for the GUI to come up, midifile_search can be used. For example, to find all the fringe-track and photometry files of ABAUR in the current directory, type

```
files=midifile_search("*.fits",OBJECT="ABAUR",/FTRACK,/PHOTOMETRY)
```

To check what is stored in the variable files just type

```
print,files
```

or

```
print,files[i]
```

to print the files of dataset i.

Q: Select all the AB Auriga obs_fringes_track_fourier datasets and print the corresponding files. Same for HD98292_ACQ_UT_COARSE_CHOP datasets.

4 Inspecting data

The commands described in this section are low level commands to extract data directly from MIDI Fits files. These commands are presented here for an introduction purpose in order that you get familiar with the kind of raw data one can get with MIDI. However, we stress that this commands are not necessarily used during the data reduction.

Now it is time to see what MIDI files contain. Select all acquisitions files related to HD102461 and store them in "calfile".

To read the first acquisition data, type

```
acq=oirgetdata(calfile[0])
```

This will take a few seconds and the data is read into the variable acq.

Note that raw MIDI data is stored in 16-bit signed integers. Since the maximum counts can be higher than the maximum such variables can hold, an offset of 32768 is subtracted from the counts before they are written to the file. This means that you have to add the offset again when you read the data in order to get the correct counts. Processed data, e.g. the mean and RMS of a data cube, are generally stored in floats, therefore there's no need to add any offsets.

With

```
tvscl,acq[100].data1
```

The 100th frame of beam B (data1) is displayed. Since each real pixel on the detector is shown as one pixel on your monitor, it's a good idea to magnify the image, e.g. by a factor 5:

```
tvsclm,acq[100].data1,5
```

Try also e.g. a contour plot with

```
contour,acq[100].data1
```

Q: Look at the acquisition images for beam A (replace data1 by data2) for each acquisition dataset.

5 Photometry

5.1 Undispersed Photometry

In this section we show how to extract the photometry of the source from Acquisition Files. As shown above the flux from the calibrator is still hidden in the overwhelming background that dominates observations in the mid-infrared. To overcome this, the background has to be determined immediately after a certain number of frames with the source have been taken. This is done by observing with the telescope a nearby part of the sky ("chopping"). Whether a exposure is taken on the sky or on the source is given in the fitsheader belonging to that frame.

A routine called chop_nod_phot makes use of this keyword. The routine is started by typing :

```
chop_nod_phot,calfile[0]
```

Options

- THRESHOLD=x.xx (by default 0.1)
- /SILENT, tells routine to be non interactive
- /REVERSE, flags for object and sky are reversed

This routine subtracts in both beams the sky frames from the acquisition frames containing the source, combines them and calculates the relative positions of the source by fitting two-dimensional gaussians.

After asking the user whether the dataset looks satisfactory, the routine continues with the aperture photometry for radii from 1 to 10 pixels (i = 1..10). The sky annuli are defined with an inner radius i+6 and an outer radius i+8. The calculated flux is given for both beams.

When the convergence defined as (Flux[i]-Flux[i-1])/Flux[i] reaches a certain threshold (by default 0.1) in one of the two beams, the photon flux is given for this radius j (tick in the plot) and the following one (j+1).

The results are prompted on the shell window: the position of the object and its FWHM and the value and position of the max of the image. Note that the direction of the North is indicated as the MIDI Field-Of-View rotates during observations. The North directions must be checked upon when one suspects that the source is spatially resolved.

You must accept or reject the fits. If accepted, the photometric results are displayed (for two radius).

Q: Try to look at the undispersed photometry for HD102461.

5.2 Dispersed Photometry: getting a spectrum

A similar routine exists for dispersed data. It is called chop_nod_disp. To start it, e.g. for the photometry of beam A, after selecting the first dispersed photometric dataset (default_chop & AOPEN), just type :

```
  chop_nod_disp,calfile[0]
```

Options

- WIDTH=w, width of 1D-gaussian relative to width of fit (by default 1.0)
- TRACE_ORDER=o, order of polynomial used to fit position (by default 2)
- FWHM_ORDER=o, order of polynomial used to fit FWHM (by default 1)
- BEFORE=b, number of frames to skip before chop (by default 1)
- AFTER=a, number of frames to skip after chop (by default 1)
- /SILENT , tells routine to be non interactive
- /REVERSE , flags for object and sky are reversed (obsolete)

The chop_nod_disp command creates in the PHOTOMETRY directory an ascii file containing the extracted fluxed in count Units.

Q: Do the same with the other beam.

You have to calibrate these spectra and some utilities, note directly inserted in the MIA package also exist to help for the extraction of calibrated spectra.

The command to read these files is readphot,'photometric-file',output

The output is an array containing at least two spectra (High_Sens mode) or more (Sci_Phot mode).

```
readphot, calfile[0], hd102461
```

and in the screen, one gets the following information looking like (here for another object):

```
;PHOTOMETRY/Phot_2004-12-30T04:53:03.500.txt
;TARGET : hd25604
;DISPERSION : GRISM
;MODE : HIGH_SENS
;SHUTTER : AOPEN
;AIRMASS :  2.06900
```

These utilities are available in the form of a library containing useful routines and template spectra extracted from the ISO database of mid-IR calibrators: http://www.iso.esac.esa.int/users/expl_lib/ISO/wwwcal/isopr

A part of these calibrators, are also MIDI calibrator for interferometric observation. It has been proved very useful to observe on the these calibrators for calibrating simultaneously spectra and visibilities. Note however that the Cohen calibrators are rather faint (5-15 Jy).

```
wave=wavecalPrism(size(spectrum))
```

or

```
wave=wavecalGrism(size(spectrum))
```

are getting respectively the working wavelength for the Prism or the Grism .

and

```
cohen,'K0IIIb_hd37160.midi',wave,template
```

is getting the Cohen template of K0IIIb_hd37160. for the wavelength *wave*

If your calibrator is also a primary calibrator from the ISO database so you can compute directly the spectrum of you object: Targ_{Jy}=$\mathrm{Template}_{ADUs}$*Targ_{ADUs}/Cal_{ADUs}

We stress that the target and calibrator spectra have to be recorded at close airmass (i.e. typically ±0.2). An airmass correction is difficult to be performed with existing MIDI mode of operation. It would imply recording spectra of the same source at different airmass and would prevent the efficiency of MIDI for recording fringes, its primary goal.

If the target is NOT a primary calibrator, the best is to use as reference the IRAS flux of the object and use a Cohen template of similar spectral type for the continuum correction: Targ_{Jy}=$\mathrm{Template}_{ADUs}$*F12μm_{Cal}*Targ_{ADUs}/(Cal_{ADUs}*F12$\mu m_{template}$)

Q: Try to reduce the dispersed photometry for HD102461.

6 Extracting visibilities

6.1 Working with the MIA GUI

As a first step start "Gorgonzola" by typing

```
  calfile=midigui(dir='path')
```

where 'path' is the directory where the interferometric and photometric data are stored. Now select the data set for the fringe tracking (OBS_FRINGE_TRACK_FOURIER) and the two photometric measurements (OBS_PHOTOMETRY_CHOP) by marking the first column and pressing the button "SELECT". Then the names of the files are stored for this example in the array calfile. It is important that the data set containing the fringes is stored as the first entry in this array. If the photometric observations have been performed before the fringe-tracking was done, please rearrange the array by typing:

```
  calfile=SHIFT(calfile,1)
```

Now the main routine of the MIA software package xmdv can be started with:

```
  cal=xmdv(calfile)
```

There are several options you can give after calfile:

- MASKFILE=filename uses the specified file as mask to extract the photometric flux and the fringe signal. This can be the mask of
- photometry A or B or any other mask you found somewhere.
- MASKWIDTH=w create a mask that is wider by a factor w than the default.

```
cal -> gui,/true
```

to get the Gui.

Q: Play with HD102461 again.

6.2 Instrumental visibilities

In a next step we want to determine the instrumental visibility, i.e. the visibility of an unresolved point source, from a calibrator object of known diameter with the routine

```
  insvis = x->instruvisi(diameter,NAME=name)
```

where diameter is the diameter of the calibrator in mas. If the object is listed in the calibrator database, i.e. in 'heidelberg/calibrators.pro', the diameter can be omitted. The name that is used to find the object in the database is either read from the MIDI data files (can be checked with object = cal-¿get_objectname()) or can be given with the parameter NAME. The instrumental visibility is calculated by dividing the raw visibility with the expected visibility of the calibrator. When /VISPLOT is specified one gets a plot of the instrumental visibility, i.e. insvis[2,*] vs. insvis[0,*]. The parameter /PLOT launches the display for the wavelength-binned spectral power of the fringes.

HD102461 should be in the database. Thus the output of the command

```
  insvis = cal->instruvisi()
```

look like (here for another object)
Looking up hd10380 in calibrator database
Found: 2.85000 mas
object: hd10380
Baseline: U1 - U3
ESO's baseline len.: 72.7130
Date of observation: 2003-06-15T09:37:19.000
Local Sid Time: 80914.915
Right Ascension: 25.357838
Declination: 5.4878200
Our baseline length: 72.7233, p.a.: 7.63431
Diameter 2.85000mas = 1.38172e-08rad
lambda x_bessel visibility
12.9299 0.244145 0.992568
....

6.3 Calibrating visibilities

To calculate the calibrated visibility of the science object, i.e. to divide the raw visibiliy of the science object by the instrumental visibility, use the routine

```
  calvis = sci->calibratedvisi(cal,diameter,NAME=name,/PLOT,/VISPLOT)
```

where diameter is the diameter of the calibrator in mas. Alternatively, the diameter can be read from the database ('heidelberg/calibrators.pro') by using the name of the calibrator that is either set automatically from the fits header (can be checked with object = cal-¿get_objectname()) or manually by using the parameter NAME. The advantage of the routine calibratedvisi() over calling instruvisi() manually and dividing the raw visibility of the science target by the result is the adjustment of the lambda-binning for the visibility of the calibrator, i.e. calibratedvisi()

ensures that the same lambda-binning used for the science target is applied to the calibrator. The option /VISPLOT leads to a plot of the calibrated visibility. /PLOT launches the display for the wavelength-binned spectral power of the fringes for both the raw visibility of the science target and the now lambda-bins-adjusted instrumental visibility.

```
  calvis = sci->calibratedvisi(cal,/VISPLOT)
```

The wavelength and the calibrated visibility are stored in calvis[0,*] and calvis[2,*].

You are now able to plot the visibility versus the wavelength with the IDL command :

```
plot,calvis[0,*],calvis[2,*]
```

Q: Follow this procedure with HD98292 as a science object (do not forget to reduce photometry).

7 Conclusion

Reduce now the AB Auriga data, using HD37160 for the calibrator object.