BIMA Memoranda Series #16

Calibrating BIMA Data with MIRIAD: A Cookbook

Christine Wilson
University of Maryland
August 26, 1991

Abstract

This cookbook is designed to help the new or slightly experienced user calibrate BIMA data using the MIRIAD data reduction package. Some detailed guidelines are given for editing the data and estimating the quality of the data. Some familiarity with MIRIAD is assumed, such as how to run MIRIAD programs and how the observations from a single track on an object are organized into the different MIRIAD datasets. Advanced topics such as `selfcal` and `MINT` are not covered here.

Preliminary Housekeeping

Your data from a particular observing session (called a 'track') consists of at least three MIRIAD datasets: the program source, the phase calibrator, and the passband/amplitude calibrator. You may have more than one phase calibrator, you probably have separate amplitude and passband calibrators (except in C array), and you may have more than one amplitude calibrator. In addition, if your observing session crossed 0 hours UT, you will have separate datasets for each day. Since you will typically have 3-5 different tracks for a given program source, it is most convenient to keep the observations from different tracks in separate sub-directories. Thus the first thing to do is to locate all the datasets from a particular track and move them into the sub-directory in which you want to work. In addition, the datasets may need to be converted from RALINT format to MIRIAD format. Since where and how the datasets are stored is site specific, no details are given here; see the MIRIAD User's Guide or your local MIRIAD expert for help in locating your data.

Listing the data

The first thing to do is to use `listobs` to generate a listing of all the observations taken during a single track. This includes your phase, amplitude, and passband calibrators, as well as your source observations. The program `listobs` will accept wild cards as part of its input. So, if you were observing in HCN, you could run `listobs` with `in=`**hcn* and `log=date.log`. This will create an ASCII file called `date.log` that contains the output from `listobs`.

Examine the file `date.log`. Always look at the system temperatures, as this gives you a first indication of how good your data is. Look carefully at any high system temperatures: was the observation taken at low elevation? If so, then the high temperature is probably not something to worry about. If not, you may have had unsettled weather. For ^{12}CO (115 GHz), reasonable system temperatures in summer near the zenith are 1000 K, while for HCN (89 GHz) they drop as low as 300 K. The system temperatures get worse rapidly as you get closer to 115 GHz: ^{13}CO and C^{18}O (110 GHz) have system temperatures that are a factor of two better than ^{12}CO. Your system temperatures will increase as the source drops in elevation, to as much as 4000 K and 1000 K for ^{12}CO and HCN at 20^o elevation, respectively.

Look at the end of `date.log` to see if the focus was changed during the track. If so, there will probably be a phase jump on one or more baselines when the focus was changed. Also look at the top of the file where the baseline distance in wavelengths is specified; it is handy to know which baseline was the shortest.

If this is your first data on a new source, it is wise to check the source parameters near the top of `date.log`, especially the velocity and the correlator setup, to make sure you set up the observing command file correctly.

Correcting the baselines

If necessary, the next thing to do is to make any baseline corrections to the data. The baseline is usually updated after each move, but sometimes small changes are necessary, especially near the start of a configuration when things may still be settling down. The baseline correction is done using the program `uvedit`. Use `vis=`**hcn* to correct all your datasets at once. The parameter `dantpos` is used to specify the baseline correction. Set `dantpos=A1,X1,Y1,Z1,A2,X2,Y2,Z2,A3,X3,Y3,Z3`, where the input values are the antenna number and the three equatorial coordinate offsets in units of nanoseconds. The values for these offsets can usually be obtained from Rick Forster; you may also be able to find them in the $MIRCAT or miriad: subdirectory. The names of the output datasets from `uvedit` will be the names of the input datasets with "_c" appended to the end.

Making the amplitude and phase calibration dataset

The next step is to create the dataset which you will use to determine the calibration of phase and amplitude as a function of time. This is done using the program `calmake`. The easiest way to do this is to create a file called, e.g., `cal.sources`, which simply contains a list of all the calibrator datasets except the narrowband passband observations. The inputs you need for `calmake` are `vis=@cal.sources`, `gcal=cal_date` (the output dataset), and `auto=true`. The last parameter automatically inserts breakpoints when the source changes, such as when you changed from the passband to the phase calibrator. It is handy to create your dataset with these breakpoints inserted automatically, since they are a lot easier to take out than to put them in later by hand if they turn out to be necessary.

Sample file: `cal.sources`

> 3c273whcn_10jun
> 2037p511hcn_10jun
> venushcn_10jun

While `calmake` is running, it will tell you, among other things, what fluxes it is using for the quasars, when they were measured, and what the flux is on each baseline for the planets. If any of your quasars are not found in the flux table, or if you think the value in the table is wrong, you can specify the flux yourself by specifying, e.g., `fluxes=3c454,10.0`. `calmake` will also give you a warning message if the visibility for a planet on any baseline falls below 30% of the true flux. Generally, if less than 30% of the true flux of the planet is detected on a given baseline, it will not be a reliable amplitude calibrator on that baseline.

Obtaining the phase and amplitude calibration

How good is your data?

The first step in obtaining the phase and amplitude calibration is to use the program `calflag` to inspect and edit your calibration dataset. Use `gcal=cal_date` (the output from `calmake`), and set `device` to your appropriate display device (e.g., on a Sun, `device=/sun`).

Inspecting your data using `calflag` can give you an idea of how good your data is. First of all, look at the phase scatter: if it is larger than 4 radians peak to peak your data is probably worthless. Good data typically has a phase scatter of less than 1-2 radians peak to peak. Second, estimate the gains on your amplitude calibrator: for good ^{12}CO data they should be about 300-350 Jy/K, while for lower frequencies they should be more like 200 Jy/K. (We will look at the exact gains after we fit the calibration data.) Remember to look at the scale on the plots: the upper and lower sideband are almost never at the same scale, so what looks like bad phase scatter may just be due to a larger scale plot.

Editing your data

First, inspect the breakpoints: are they all necessary? If at all possible, we want to eliminate unnecessary breakpoints. However, if a breakpoint is required on one baseline, I generally leave it in on all three baselines. If you had a focus change, look carefully at the phases across the change to see if they continue smoothly and if so, remove the breakpoint. Focus changes almost always cause a jump in the phase, so these breakpoints are usually needed. If you changed phase calibrators during the track, look to see if the phases continue smoothly across the change of calibrator and if so, remove the breakpoint. A jump in phase between two different phase calibrators may imply an error in the baseline. In addition, be aware that if you have to leave in the breakpoint between the two phase calibrators, you will lose any data between the two calibrator observations. Leave in breakpoints which separate the observations of amplitude and passband calibrators from any other observations: we will want to fit each of these calibrators separately.

Next, you will want to go through and edit the data by flagging points that don't cluster together well. For example, you probably want to remove all points with gains larger than 1000 Jy/K. In addition, if you have a group of 4-6 observations where the phase or amplitude of 1-2 points are significantly different from the rest, you should probably also remove the discrepant points. If you had some high amplitude points that forced the amplitude scale to be very small, you will probably want to look at the data again once these points are taken out. The only way to get `calflag` to enlarge the scale is to exit (saving your flags) and then run it again. You can also fix the amplitude scale by setting `ampmax=1000`.

It sometimes occurs that you had bad weather or some other problem and you want to throw out a large chunk of the data before you even start. One way to do this is to use `uvflag` to flag data as bad in the individual quasar datasets before you create the calibration dataset. For example, say by looking at your output from `listobs` you see that you want to throw out the first hour of your calibrator data because it was raining and the system temperatures were extremely large. You can also use the program `uvlist` with `options=full,variables` to look at the data with higher time resolution than provided by `listobs`.

Use the program `uvflag` with `select=time(`*start,stop*`)` and `flagval=flag` to flag the points in the given time range as bad. You can check that you've done it correctly using the program `uvlist`. Use `select=time(`*start,stop*`)` and `recnum=20000`; the visibility points will be printed to your terminal. Data which have been flagged bad show up with an asterisk next to them. If you use the same start and stop times as you used in `uvflag`, all the data should show the asterisk. If you then use `select=time(`*stop,stop+1*`)`, you should see some unflagged data. Another way to see the result of this editing is that some points will show up as flagged in `calflag` when you first look at your calibration dataset, even before you have done any interactive editing. If the section of bad data has good calibrator data on either side of it, you also need to insert a breakpoint somewhere in the time range of the bad data. This will result in your source data in this time range being flagged as bad when the calibration is applied. Without it, `calfit` will extrapolate the fit and happily calibrate your source data in the bad region.

Obtaining the calibration

The program `calfit` is used to calculate the phase and amplitude calibration. The amplitude should almost always be fit to zeroth order; for phase I generally start with second order and increase or decrease it if necessary. You should normally use the lowest order polynomial that adequately fits the data. Fit the polynomial with the most breakpoints first; this will usually be the phase. The inputs to `calfit` are `gcal=cal_date`, `order=0,2`, and `mode=phase`. An additional complication is that we want to force the phase fit to the passband calibrator to be linear, otherwise you can often end up with ridiculous looking fits for this source due to the small time range covered. The input `taver=`*tgap,ttot*, which controls the vector averaging of the data, can be used for this purpose. I generally start with `taver=20,30`; the default is `taver=0,1000` (no vector averaging). If you find you are not forcing a linear fit to the passband calibrator, increase *ttot* until you do. When you manage to force a linear fit, you will get a warning message that "fit was done to a lesser degree as requested". For one passband or amplitude calibrator, there should be 6 of these messages (2 sidebands x 3 baselines); if you have two calibrators with a short total observation time you should see 12 messages, etc.

You should then fit for the amplitude calibration. With the inputs listed above, use `mode=amp, taver=20,30,` and `source=`*amplitude_calibrator_name*. Unless your quasar is very strong (> 10 Jy) and your weather was very good, you will not want to even try calculating gains from your phase or passband calibrators, since amplitude bias can easily be a significant problem for sources weaker than 10 Jy. Your source is probably suffering from amplitude bias if the fit from `calfit` lies significantly above the mean distribution of the amplitudes when you look at the fit with `calflag`. `calfit` will tell you the fitted gains on each of the baselines; check that the gains are reasonable (200-350 Jy/K).

Inspecting the fit

Now run `calflag` again with the same inputs you used initially. The fits will appear overlaid on the plots of the data. Check that the phase fits track the data reasonably well and that the amplitude fit is near individual gain measurements for the calibrators. Note that `calflag` sometimes shows a 2π difference between the phase fits and the data; this is a known but not yet fixed bug, so don't panic, the calibration is correct.

If you don't like the fits or just want to try some other combination of parameters, you may need to delete the old fits using the program `delhd` before you can make new ones. Specify `in=cal_date/pdata` to delete the polynomial fits.

Making the passband calibration dataset

The next step is to create the dataset which you will use to determine the passband calibration. This is done using the program `passmake`. If you have observed in the "wide-wide-wide" mode, the only inputs you need for `passmake` are `vis=`*wideband_passband_dataset* and `pcal=pass_date` (the output dataset). If you have observed in the "wide-narrow-wide mode", then you must specify `vis=`*narrowband_passband_dataset* and use the phase calibration you fit in the previous step, by setting `gcal=cal_date`. Do not fit higher than first order in amplitude and second order in phase; `order=0,0` is often sufficient for narrow passbands.

Obtaining the passband calibration

The passband calibration is made using `passfit`. Use `pcal=pass_date`, `drop=4` (to drop 4 channels at ends of bands and near corfs), `order=1,1`, and `chave=8`. You can improve the signal-to-noise by using the `chave` parameter; in general it should not be set to any larger than 16.

When you run `passfit`, you will get plots of the phase and amplitude versus channel for each of the three baselines successively. Estimate the peak to peak scatter in the amplitude and divide by 4-5; the inverse of this is your signal to noise ratio. If the signal to noise ratio is < 5, you definitely should be channel averaging; if it is < 3, the data should not be used for passband calibration. It is generally better to apply no passband calibration on any given baseline than to apply one from low signal-to-noise data. The amplitude fit should be within 20% of 1, and the phase fit should be within 30^o of zero (the center of the plot). If you are using a first order fit to amplitude, look at all the fits carefully to make sure a few high or low points at the ends are not forcing an unnecessarily steep slope. If so, it might be better to do a zeroth order fit in amplitude. Alternatively, you can interactively flag a few points as bad and refit the data. You can also try specifying `drop=6` or `8`.

If you don't have enough signal to noise in your narrowband observations, you can use the wideband mode observations of your passband calibrator to solve for the passband, provided your narrowband windows fall entirely within a single corf in the wideband mode. Rerun `passmake` with `vis=`*wideband_passband_dataset* and no `gcal` and then proceed to fit this new dataset using `passfit`.

Applying the calibration

The phase, amplitude, and passband calibrations are applied to the source data using `calapply`. Use `vis=`*program_source_dataset*, `gcal=cal_date, pcal=pass_date, out=` *source_cal,* and `flagbad=false,true`. Specifying `flagbad=false,true` tells the program to apply the amplitude calibration outside the range to which it was fit (i.e., apply it to the whole dataset and not just the time range of the amplitude calibrator), while the phase calibration is only applied over the time range(s) for which it was fit.

It is possible to apply different order passband fits on different baselines, but the procedure is somewhat complicated. This might be necessary if you were using a planet for your passband calibration and it turned out to be heavily resolved out on one baseline. As an example, suppose you want to apply the passband calibration to only the two baselines involving antenna 2. You would first run `calapply` with standard inputs but using `select=antenna(2)` and `out=temp2`. You would then run `calapply` with the same input for `vis`, no `pcal`, `out=temp13`, and `select=antennae(1)(3)`. Finally, you would use the program `uvcat` to combine the `temp2` and `temp13` datasets.

Checking the calibration

You can check whether the phase calibration produced satisfactory results by using `calapply` to apply the phase calibration to the phase calibrator(s). Do not specify `pcal` and use `flagbad=false,false` if you used the `taver` parameter when you ran `calfit`. This allows extrapolation of the fitted polynomials in phase as well as amplitude; without it you will lose data off the end. Then run `calmake` on the output dataset(s) from `calapply`, and use `calflag` to look at the output from `calmake`. If you don't change `units`, the source amplitude should come out near 1, while the phase should be near zero.

A similar series of steps can be used to check the passband calibration. For data taken in "wide-wide-wide" mode, use `calapply` to calibrate your passband dataset by specifying `pcal=pass_date` and no `gcal`. Then run `passmake` on the output from `calapply` and use `passfit` to look at the calibrated data. Again, the amplitudes should be near 1 and the phase near zero. If you interactively flagged a few bad data points in your original fit, you might see a slight residual slope, since this new dataset will not have those points flagged. Re-flagging the bad points should remove any residual slope.

You can also estimate the fluxes of all the quasars you observed. Use the program `uvaver` with `vis=`*calibrated_quasar_dataset*, `line=wide,2,1,1,1` and `out=temp`; this will select out only the wideband data. Then run `uvaver` with `vis=temp, interval=1000`, and `out=temp1` to time-average all the wideband measurements together. Then run `uvlist` with `vis=temp1` and `recnum=20000` to get a listing of calibrated fluxes on the three baselines and two sidebands. On the line for each baseline, lower sideband is listed first. In calculating the average flux for the quasar over all baselines and sidebands, it is nice to limit yourself to observations where the aperture efficiency is greater than 40% (aperture efficiency = 94/gain (Jy/K)). However, this is not usually possible for the CO transitions. To estimate the aperture efficiency, look at the gains using `callist` with `gcal=cal_date` and `poly=true`. "AUW? 3" indicates the fit to amplitude in the upper sideband on the third baseline; "PLW? 3" indicates the fit to the phase in the lower sideband on the third baseline.

Summary of Calibration Steps

1. Creating a summary listing of your data

`listobs in=`**hcn* `log=date.log`
Include all datasets; inspect system temperatures; note focus changes.

2. Correcting for baseline errors (if necessary)

If baseline corrections are required, run
`uvedit vis=`**hcn* `dantpos=A1,X1,Y1,Z1,...`
Output_dataset_name = input_dataset_name_c

3. Removing bad data in a selected time interval (if necessary)

If large sections of data are bad, run
`uvflag vis=`*calibrator_dataset* `select=time(`*start,stop*`) flagval=flag`
Insert a breakpoint roughly in the middle of the bad section using `calflag`.

Check flagging was done correctly using
`uvlist vis=`*calibrator_dataset* `select=time(`*start,stop*`) recnum=20000`

4. Making the calibration dataset

`calmake vis=@cal.sources gcal=cal_date auto=true`
Include all phase, amplitude calibrator datasets, plus wideband passband datasets.

5. Editing the calibration dataset

`calflag gcal=cal_date device=...`
Check phase scatter, gains; clear all unnecessary breakpoints; flag high amplitude points, points with large scatter in amplitude or phase.

6. Creating the polynomial calibration fits

`calfit gcal=cal_date order=0,2 mode=phase taver=20,30`
Make sure you have forced a first order fit to the passband and amplitude calibrators; use higher or lower order fit as necessary

`calfit gcal=cal_date order=0,2 mode=amp taver=20,30`
`source=`*amplitude_calibrator_name*
Do not use sources weaker than 10 Jy to try to calculate gains.

7. Checking the calibration fits

`calflag vis=cal_date device=...`
Inspect quality of fits; before making new ones, delete old fit using
`delhd in=cal_date/pdata`

8. Creating the passband dataset

For "wide-narrow-wide" passband observations, use
`passmake vis=`*narrowband_passband_dataset* `gcal=cal_date pcal=pass_date`

For "wide-wide-wide" passband observations, use
`passmake vis=`*wideband_passband_dataset* `pcal=pass_date`

9. Obtaining the passband fit

`passfit pcal=pass_date drop=4 order=1,1 chave=8`
Estimate S/N ratio in amplitude; do not use data with S/N <3 to fit passband; amplitude fit within 20% of 1, phase within 30^o of 0; check slopes of amplitude fits are reasonable and if not, use zeroth order; in wideband mode second order fit to phase may be necessary

10. Calibrating your data

`calapply vis=`*source_dataset* `gcal=cal_date pcal=pass_date flagbad=false,true out=source_cal`

11. Checking your calibration

To check the phase calibration, for each phase calibrator
`calapply vis=`*quasar_dataset* `gcal=cal_date flagbad=false,false out=`*quasar*`_cal`
`calmake vis=`*quasar1*`_cal,`*quasar2*`_cal,... gcal=caltest`
`calflag gcal=caltest device=?`
Check that amplitudes are near 1, phases near 0.

To check the passband calibration in "wide-wide-wide" mode
`calapply vis=`*passband_quasar/planet_dataset* `pcal=pass_date out=`*quasar/planet*`_pcal`
`passmake vis=`*quasar/planet*`_pcal pcal=passtest`
`passfit pcal=passtest drop=4 order=1,1 chave=8`
Check that amplitudes are near 1, phases near 0.

To get the calibrated fluxes for your quasars
`uvaver vis=`*quasar*`_cal line=wide,2,1,1,1 out=temp`
`uvaver vis=temp interval=1000 out=temp1`
`uvlist vis=temp1 recnum=20000`
Calculate average gains for all baselines; perhaps select only baselines where aperture efficiency (94/gain) is >0.4.