Spectral Line Data Reduction at the Large Millimeter Telescope

F. Peter Schloerb
University of Massachusetts Amherst

July 23, 2019

CHANGE HISTORY:

April 30, 2019 – Initial Draft
July 23, 2019 – Expanded OTF Mapping Section; Appendix C

Contents

APPENDIX C OTF Mapping Simulations

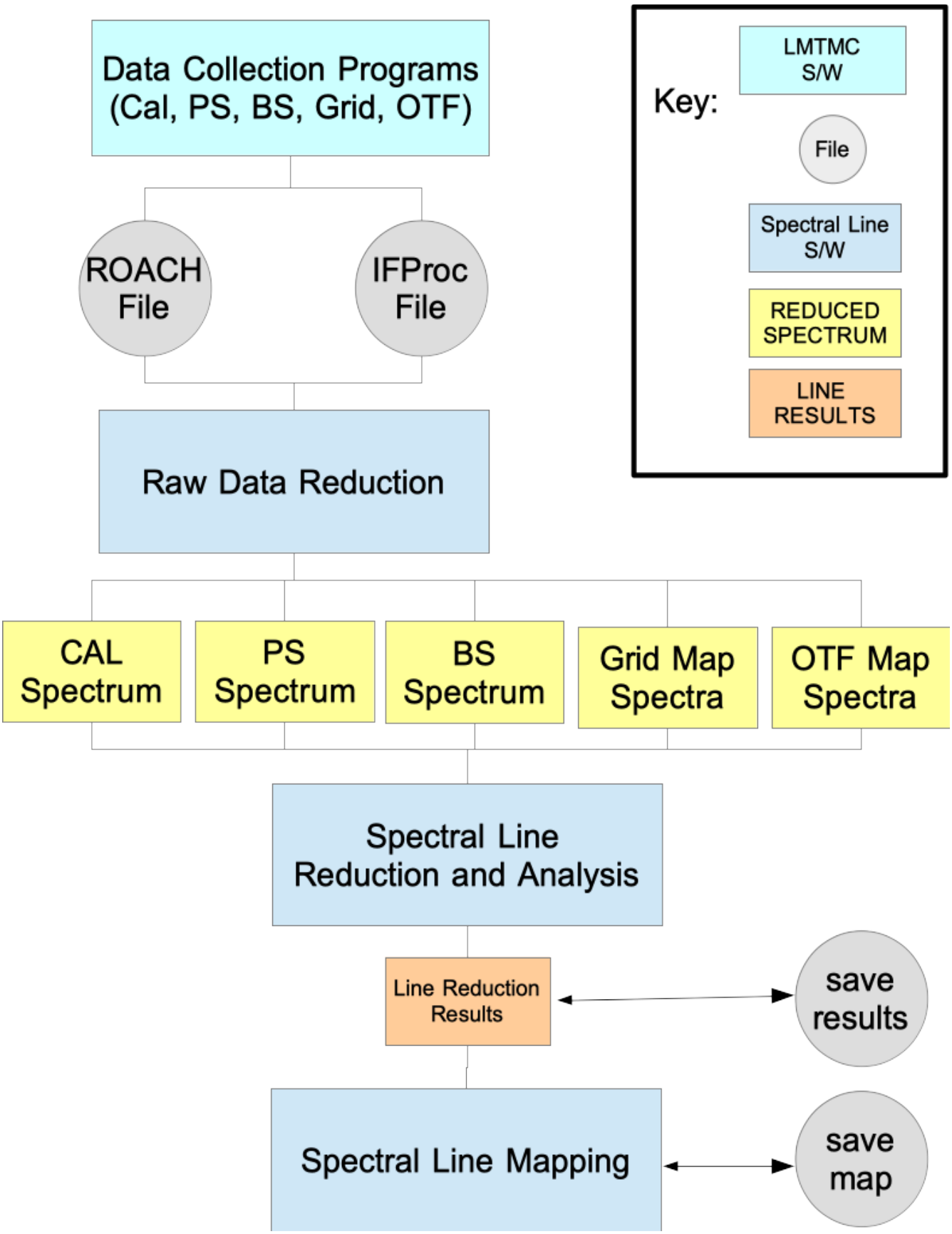


Figure 1 – Overview of Spectral Line Data Reduction Software for LMT

1. Introduction

The Large Millimeter Telescope Alfonso Serrano has a number of receivers which are capable of obtaining spectral line data. These receivers make use of a single spectrometer system, known as WARES. In its present configuration, this spectrometer can analyze 16 inputs and operates in a number of modes.

My goal for this document is to lay out a summary of the end-to-end process of spectral line data reduction leading to the creation of advanced data products for the LMT. Figure 1 presents an overview diagram with the major components and data types for the software involved in collecting and reducing LMT spectral line data. The document seeks to describe: (1) the production of the raw LMT spectral line data, including use of basic instrumentation, data collection programs, and data formats; (2) the necessary steps and features of reduction software for the raw data files that result from LMT observations; (3) necessary steps and features of data reduction/analysis software for spectral line reduction and analysis after the raw data reduction step; and (4) spectral line mapping software needed to create maps from sets of spectra.

2. Production of Raw Spectral Line Data at the LMT

2.1 Spectral Line Front Ends

At this time, there are two front end instruments on the telescope that are used with the WARES spectrometer: (1) SEQUOIA; and (2) the MSIP 1mm Receiver. We also anticipate a 16-element array receiver for the 1.3mm window (OMAyA). So I will describe the use of that system with the spectrometer as well.

We will ignore two spectral line instruments that do not currently use WARES: the Redshift Search Receiver and the Band 4 Receiver provided by colleagues at NAOJ. The RSR is an ultra-wideband receiver/spectrometer system which covers the 74-111 GHz band at low (30 MHz) resolution. The B4R receiver is a guest instrument on the LMT. It is an ALMA band 4 (2mm) receiver provided with its own special purpose spectrometer. LMT may chose to connect these receivers to WARES in the future.

2.1.1 SEQUOIA Description

SEQUOIA is a 16-element focal plane array for the 85-115 GHz region. The array can be tuned to cover either the 85-100 GHz band or the 100-115+ GHz band. The full 15 GHz of either band is available for processing. The 16 inputs of the spectrometer are used to analyze the data from the 16 pixels of the array.

2.1.2 MSIP 1mm Receiver Description

The 1mm receiver is a “single pixel” receiver. It is a dual (circular) polarization, sideband separation receiver so that it produces 4 IF outputs for analysis (one for each polarization and each sideband). The receiver has a tunable LO system to

cover 210-270 GHz. The IF bandwidth is nominally 8 GHz in each sideband covering IF frequencies from 4-12 GHz. The spectrometer is then configured to analyze four outputs from the receiver with one dual polarization pair from each sideband of the receiver.

2.1.3 OMAyA Description

This receiver consists of 8 dual-polarization pixels covering the one millimeter window from 210-270 GHz. Each pixel produces four 4-12 GHz IF outputs --- one for each sideband in each polarization --- as with the MSIP 1mm Receiver. The 16 spectrometer inputs would then nominally be configured to look at a single spectral line at 8 points on the sky in two polarizations.

2.2 IF Processor and Spectrometer

IF signals from the spectral line front end instruments are downconverted to baseband for input to the WARES spectrometer. This IF processing is done in two stages. The *IF Processor* uses a second local oscillator to carry out an initial conversion of the receiver IF to baseband with bandwidth of approximately 1 GHz. The *Baseband Processor* then provides a second stage of filtering and amplification of signals, including band pass filtering corresponding to the modes of the spectrometer.

TABLE 1 – WARES SPECTROMETER MODES AND DATA REQUIREMENTS

	Mode		
Property	**Wide**	**Intermediate**	**Narrow**
Bandwidth (MHz)	800	400	200
N Channels	2048	4096	8192
Chan. Spacing(kHz)	391	98	24
Data per Spectrum (kB)	8	16	32
SEQUOIA Data Rate (MB/s) [1]	1.25	2.5	5
SEQUOIA 5-minute observation (GB)	0.37	0.73	1.46
SEQUOIA Data per Night (GB) [2]	24.41	48.83	97.66

Note [1] – Assumes 16 pixels and dump time of 0.1s
Note [2] – Assumes 20,000 s per night (~5.5h)

Inputs to the IF processor system can work with two possible second LO sources to allow two independent spectral lines to be tracked. For SEQUOIA, this is not done since there are only 16 spectrometers available. However for the MSIP 1mm receiver, this capability is used to independently track lines in the upper and lower sidebands of the receiver. OMAyA would nominally be configured to work with all spectrometers looking at the same spectral line, although of course there are other possible options.

The WARES spectrometer can analyze 16 IF outputs in one of three modes. The modes are summarized in Table 1 above. The 16 spectrometer inputs are handled by 4 ROACH2 boards, with each board therefore producing four spectra. The four spectra from each ROACH board are managed separately and recorded together as a single file (the ROACH file) for each observation.

A set of spectrometers serving different receivers (pixels) but having the same LO settings and covering the same frequencies is called a *bank* of spectrometers. The 16 available spectrometers are set up as a single bank for SEQUOIA observations. Observations with the MSIP 1mm receiver cover two separate lines (currently one in USB and one in LSB) and therefore are currently analyzed in two banks of two spectrometers each.

2.3 Data Collection Programs

The LMT system provides several programs to coordinate telescope motions and data acquisition.

- Position Switching – This is a standard way to obtain spectra by switching between a “Main” and “Reference” position on the sky.

- Beam Switching – This is a variation on position switching using a receiver with multiple pixels. The “Main” and “Reference” positions on the sky are calculated so that the receiver is always pointing at the source. This is most useful for point sources.

- Grid Mapping – A grid map is a sequence of spectra taken on a regular grid of sky points. In this procedure, the telescope tracks a specific position in the grid as the “Main” position in a position switched spectrum. The procedure allows the user to define a single integration on the “Reference” position to be used for all points or allows the user to interleave additional “Reference” spectra into the observation.

- OTF Mapping – In this procedure the telescope is scanned across the sky to sample the emission. The samples are then “gridded” into a map.

The above data collection programs can be used for both calibration and science purposes. For example, spectral line pointing and focus measurements are carried out using the mapping routines and position/beam switching routines.

2.4 Data Format

LMT data are recorded in the netCDF-4 format, though we limit usage to features in netCDF-3. The spectrometer records a time series of the four spectra collected by a given ROACH board. Spectra are dumped to the file (the ROACH file) approximately every t_{DUMP} seconds. (Typical values for t_{DUMP} are 0.1s.) Each dump is tagged with the time of the dump so that it may be used with a second file (the IFProc file) which contains time-tagged information about the telescope's position

and other useful ancillary data needed to properly reduce the observation. The names of the ROACH and IFProc files each contain a unique ObsNum which is used to identify the observation.

Reduction of spectral line data requires the two types of files (ROACH and IFProc) to be read so that the time-tagged spectra can be associated with sky position data that are needed to construct reduced spectra according to the type of observation being carried out. A time series of telescope positions is recorded in the IFProc file is used to establish telescope position as a function of time.

The spectrometer tags the time when the data are read into the file, which of course is not the same as the time of the actual collection of the data. We derive the time that the dump of the spectrum was initiated by using a measurement of the time taken to read out the data from the spectrometer. Then, nominally, the center time of the integration would be earlier than the ending time by ½ the integration time. This additional time correction can be derived to an accuracy of a few milliseconds from observations of sources on the sky. We find that, for 100 msec integrations, the additional correction to the end time of the integration is approximately -30 msec. This value is then used to look up the position information from the IFProc file and derive the position of the spectrum.

The final result of reading the spectra from the data file is a two-dimensional dynamic spectrum containing the spectra read out at each dump time during the observation. We define $R_{i,j}$ to be the number of counts in spectral channel i from spectrum j in the file. Each spectrum then has an array of values of the necessary position data to reduce the data properly.

The position data that is interpolated for each spectrum includes sky position data (usually the “x” and “y” positions in a map and the parallactic angle) as well as a flag (BufPos) which is a code used by the data collection program to indicate whether, e.g., the spectrum is at the “Main” or “Reference” position in a PS observation or the “Hot” or “Sky” position in a calibration measurement. The defined values of BufPos are given in the table below.

TABLE 2 – KEY TO THE BUFPOS POSITION VARIABLE

Position	BufPos Value
On Source (“Main”)	0
Off Source (“Reference”)	1
Calibration Sky Position	2
Calibration Hot Position	3
Grid Position in a Grid Map	100 + position number

2.5 Data Rates and Data Volumes

Table 1 provided data rates and example volumes for each mode of the WARES spectrometer. A night of SEQUOIA observations can produce up to 100 GBytes of

data in its highest spectral resolution mode under the assumptions given there. Even a single observation of a few minutes results in a fairly large file on the order of a GByte.

3. Raw Data Reduction Procedures

3.1 Review of an Observation Data File

A single data file contains a large number of spectra. It is important to be able to review the spectra for quality and for the purpose of finding "bad" data which should be excluded from the next reduction steps.

The LMT's basic data type is a dynamic spectrum which is actually two-dimensional. For assessment of data quality and to allow possible editing of "bad" data, it is useful to be able to see all the individual spectra in an observation in some way. Thus, to examine the raw data, it is useful to be able present 2D plots that show all the samples. I note that this is sometimes difficult to do effectively in cases where there are thousands of spectral points (e.g. 8192, 4092, or 2048) but only a few hundred spectra.

Another useful view for assessing data quality might be to show intermediate sets of spectra within the observation, such as the individual position switched spectra in the observation or the averaged Reference spectra in a map.

3.2 Editing and Flagging Data

After review of the data, it may be desirable to flag portions of the observation file data which should be eliminated from further reduction steps. A means should be provided to do this.

3.3 Reduction of a Calibration Observation

For chopper wheel calibration we may derive the system temperature by finding all spectra on the Calibration Hot Position (hereafter "HOT") and all spectra obtained with the receiver looking at the Calibration Sky Position (hereafter "SKY"). From the BufPos array, we may find the set of indices j_H which indicate the receiver on the HOT load. Similarly, we can find the set of indices j_S which indicate the receiver on the SKY position. Given these two sets of indices, we then construct a spectrum of the system temperature for each channel i of the spectrometer:

$$T_{sys}(i) = T_{amb} \frac{\overline{R_{i,j_S}}}{\overline{R_{i,j_H}} - \overline{R_{i,j_S}}}$$

where T_{amb} is the temperature of the HOT, and $\overline{R_{i,j}}$ is the median of spectral channel i over all indices j_H or j_S.

3.4 Reduction of a PS or BS Spectrum

A PS (or BS) spectrum is obtained by taking the difference of spectra on the "Main" position and the "Ref" position. In this case, we use the BufPos values to identify all spectra obtained on the Main and Reference positions respectively. If $MAIN_i$ is the value of channel i estimated from the Main spectrum and REF_i is the value of channel i estimated from the Reference spectrum, then we can form the final, calibrated spectrum, *S(i)*:

$$S(i) = T_{sys}(i) \frac{MAIN_i - REF_i}{REF_i}$$

There are a number of ways to estimate $MAIN_i$ and REF_i. Typically, given a number of Main-Reference pairs one estimates the value $(MAIN_i - REF_i)/REF_i$ for each pair and then averages the result for all pairs to get a final spectrum. The fact that the LMT's data format has a spectrum for each dump time means that the reduction of the spectrum can be more sophisticated than this, however, even allowing for individual "bad" dumps or Main-Ref pairs to be identified and removed from the final spectrum.

The reduction of a BS spectrum follows the same procedure. In this case we estimate two spectra corresponding to the two receiver beams involved in the switching. One of these beams will have the sense of Main and Reference flagged in the BufPos array reversed and so we must account for that when its spectrum computed. Ultimately, we average the two spectra to obtain the final BS spectrum.

3.5 Reduction of Spectra in a Grid Map

In a Grid Map, the user specifies a regular grid of points to be observed. The telescope dwells on each point on the grid and also spends time observing a reference position at times interleaved with the map positions. The BufPos array is used to identify where the spectra obtained are located on the grid. Grid points are identified with values beginning at 100, where the value 100 is the first point on the grid, the value 101 is the second point on the grid, etc.

A common way to carry out this observation, e.g. for pointing maps, is to observe the Reference position before and after the grid points are observed. In this way, an average reference spectrum may be computed from all the individual reference spectra and then this spectrum can be removed from the average spectrum of each grid point.

3.6 Reduction of Spectra for an OTF Map

OTF maps are made by scanning the antenna over the source to be mapped. The data are obtained by interleaving observations of the Reference position with the individual on-source spectra that will be used to make the map. The Reference position data could be used in a couple of ways. As with the common

grid map strategy, Reference observation before and after the scanned map can be averaged and used as a single Reference for all spectra in the map. On the other hand, when Reference spectra are interleaved with the scans, say every row or every few rows, then the map data can use a Reference that is based on the average of the two Reference observations that bracket the map points. Finally, in some cases where the source is small compared to the map, it is possible to use the median of all spectra obtained as a way to estimate the background spectra.

For any case, the reduction program needs to identify all the Reference spectra in the file, using the BufPos value, and then compute averages according to the options described above. Given a Reference spectrum appropriate for a period of observation within the map, one may then remove the Reference spectrum and produce an individual calibrated spectrum for each individual spectrum in the file.

4. Spectral Line Reduction and Analysis Procedures

The preceding section described reduction of the raw spectrometer data to form a calibrated spectrum with intensity as a function of spectrometer channel. After this initial step, there are many next steps to be taken with the spectrum itself. It is important to note that there are spectral line reduction packages in astronomy that might be used for the purposes enumerated below. Possibly the only feature that must be done with special LMT software involves setting the frequency / velocity scale of the spectra (the “X” axis). This section provides a broad overview of the important steps: (1) Set the “X-axis”; (2) View Data; (3) Edit/Flag Data; (4) Scale Spectra; (5) Baseline Spectra; (6) Smooth Spectra; (7) Resample Spectra; (8) Average Spectra; and (9) Spectral Line Estimation.

4.1 Set the “X” axis

Spectral line data commonly present the intensity as a function of frequency or velocity. In both cases, because of the doppler shifts involved in the Earth's motions and the Sun's motion, it is common to set provide frequency or velocity with respect to some specific frame of reference, such as the topocentric frame, the frame of the solar system bary center, the local standard of rest (LSR), etc.

When Observers set up an observation, they may specify one of these frames and a velocity to be tracked. An Observer may also specify a frequency offset for the spectrum so that the spectral line being tracked can be placed at different positions in the IF band analyzed by the spectrometer. The specific data entered are:

- Line Rest Frequency (value in GHz)
- Reference Frame of Velocity (Options: LSR, Barycenter, Topocentric)
- Source Velocity in Reference Frame (value in km/s)
- Velocity Definition (Options: Optical, Radio, and Relativistic)
- Frequency Offset (value in GHz)

Frequencies may also be set up by providing the line rest frequency and a redshift. Given the above information, the system computes the sky frequency that is to be tracked and saves the relevant values of all necessary LO frequencies in the IFProc file.

APPENDIX A provides an explanation and the relevant formulae for establishing the frequency or velocity scale of a spectrum.

4.2 View Data

A single spectrum is one-dimensional and so an easy way to view the spectrum is to make a simple plot of the intensity versus whatever “X-Axis” is specified. Thus, the ability to make labeled plots with specified ranges in X and Y and appropriate labeling is useful. These sorts of graphs are straightforward in modern plotting packages.

4.3 Edit/Flag Data

If “bad” data can be identified within a set of spectra, then it is useful to be able to flag the data so that it will not be included in subsequent analysis. It should be possible to easily remove portions of an observation.

There is also often a need to flag regions within a spectrum with bad channels or bad baseline characteristics. Ideally a spectrum with a flagged region would be able to be averaged with other spectra which had good data in that region. (Some thinking is required to figure out how to do this properly since integration time and noise would now be different across the full spectrum.)

4.4 Scale Spectra

Spectra that are calibrated using the chopper wheel technique will be in units of antenna temperature (Kelvin) on the T_A^* scale. There is sometimes a need to convert antenna temperatures to another scale to facilitate comparison with data from other telescopes or to make comparisons to physical models. Thus, there is a need to rescale the spectrum by some value.

One common rescaling of T_A^* is to divide by the main beam efficiency to obtain a *main beam antenna temperature*, T_{MB}. Another useful rescaling might include correction of data for changes in the gain of the antenna with elevation, using some standard *gain curve* for the LMT to accomplish this. (Notably, with the LMT's active surface this correction is expected to be small.) Finally, it may be useful to scale spectra from units of T_A^* into Jansky. This is especially for point sources since it allows direct comparison to results from other telescopes. All of the above scalings imply knowledge of the LMT's efficiency versus frequency and elevation angle.

4.5 Baseline Spectra

A baseline fit uses a region within the spectrum (away from the position of the spectral line) to fit a polynomial function to approximate the baseline. Commonly, one specifies one or more regions for the fit and the order of the polynomial function to be used in the fit.

In addition to determining and subtracting the derived baseline function from the spectrum, we also determine the rms of the fit of the baseline function over the fit regions as a way to estimate the noise in the spectrum.

4.6 Smooth Spectra

Spectral line data are sometimes smoothed over velocity (or frequency) in order to improve the signal-to-noise ratio of the spectral line. Typically, one specifies the smoothing function and its resolution and then the program performs a convolution of the spectrum with the function. Typical smoothing functions might be a boxcar with some number of channels or a triangular function like a Hanning or Hamming window.

4.7 Resample (Generate) Spectra

When averaging spectra, it is necessary to have the spectral line in the same channels in all arrays to be averaged. Lines are often moved within the spectrometer during a long integration to: (1) demonstrate that a suspected feature moves as expected if the frequency offset is changed; and (2) improve the quality of the spectrum by averaging out any systematic baseline errors within the spectrometer. If two spectra have the line at different channels in the spectrometer, then it is necessary to resample the spectra to make a new spectrum with the lines aligned before averaging.

Another use of resampling comes after smoothing. The convolution of the spectrum with a smoothing function leads to a new spectrum where channels within the width of the smoothing filter are no longer independent measurements of the spectrum. Thus, after smoothing, it is a common practice to resample the spectrum with a channel spacing that matches the resolution of the filter.

4.8 Average Spectra

Once a set of spectra are aligned, they may be averaged to obtain a new spectrum with a longer integration time. Spectra should generally be weighted so that very noisy spectra are not given the same weight as less noisy spectra. Usually, when making a weighted least squares estimate, one would wish to weight by the inverse of the variance of the data. Thus, common practice is to weight each spectrum according to its rms or system temperature by either $(1/\mathrm{rms})^2$ or $(1/\mathrm{Tsys})^2$.

4.9 Spectral Line Estimation

Once a final spectrum is available, we can estimate properties of the spectral line(s) being observed. Usually one provides a region within the spectrum for the the measurement.

Estimation of the moments of the line is one common practice to derive, e.g., the peak temperature observed in the region, the integrated area under the line, or the first moment of the line over velocity (frequency).

More sophisticated fitting of the line profile, using, for example, a Gaussian line shape or some other model, is also possible. Sometimes, in such fits, it is useful to fit multiple lines in the same spectrum or multiple Gaussian components to a single line.

4.10 Mapping with Focal Plane Arrays

Derivation of the position of an observation with a single pixel receiver is straightforward. However, with a focal plane array, we require additional information about the locations of the beams on the sky in order to locate the individual spectra in the final map. The geometry of the SEQUOIA focal plane array is described in APPENDIX B.

4.11 Estimation of Basic Pointing Data

Determination of pointing offsets and other properties of the antenna beam is a regular undertaking at the telescope. Thus, given a set of reduced spectra at different positions on the sky, it is useful to be able to fit a gaussian profile to derive the strength of the source, pointing of the telescope, and shape of the beam.

4.12 Saving Spectra

At the completion of the spectral line reduction process, it will be useful to save spectra and derived line results. This will be especially true for large mapping projects which generally involve many individual observations to be reduced and tagged with the sky position of the actual observation. One might choose to estimate parameters of the line being mapped at this point as well.

Thus, when the reduction step is complete, it should be possible to save any intermediate results, either the line parameters or the spectra themselves, for further processing. This requires specification of a file type (or types) for reduced spectra.

It is useful to note that with these intermediate results --- a set of spectra with tagged positions --- it is possible to use other software packages to construct maps of the data on regular grids of sky position. Alternatively, LMT may choose to write our own software for the next steps. We will describe these important

steps in the next section.

5. Spectral Line Mapping

5.1 OTF Mapping Tutorial Overview

5.1.1 Introduction

For most maps at the LMT we use the technique of “on-the-fly” (OTF) mapping. In this procedure, the antenna is scanned over the source and data are collected at a high rate. The scanning pattern results in an irregular sampling of the positions on the sky, and so to make a final, science-grade map, we must use the OTF data to estimate the value of the map at specific, regularly spaced, grid points that also account for spatially redundant information.

One might imagine that using a “gridding” procedure to make the map would be an obvious step. However, gridding procedures, such as those in python, are actually interpolation algorithms which seek to interpolate between the data points closest to the grid points to make the estimate. This works well in cases of high signal-to-noise ratio, but in the typical case of low signal-to-noise astronomy data, we'd like to make use of more data samples to find an estimate of the map at the specified grid points.

A method for estimating values on a regular grid using OTF mapping has been developed and is in common use in the OTF reduction applications in use at all single dish telescopes and at ALMA for single dish mapping. The technique is described in a paper by Mangum, Emerson, and Griesen (A&A 474 679-687 2007) as well as in other papers describing OTF systems (see e.g. Sawada et al. PASJ 60 445-455 2008 and FCRAO internal memos by M. Brewer). The basic principles of the method are straightforward, but different groups describe the problem in different ways which can make matters confusing. Thus, I offer this short tutorial. A set of simulations of the procedures are presented in Appendix D.

5.1.2 OTF Mapping Basics

Let's call the true brightness distribution of a source on the sky *I(x,y)*, where *x* and *y* are the sky coordinates. When we make a map with the LMT, we do not measure *I*, but the convolution of *I* with the beam pattern of the telescope, which I will call *B*. Lets call the result I_{obs} and then we may write the convolution:

$$I_{obs}(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} B(x-x',y-y')\,I(x,y)\,dx'\,dy' \qquad <1>$$

I'll write this another way using the convolution operation “*” to make life easier:

$$I_{obs} = B * I \qquad <2>$$

From the convolution theorem I know that if I have two functions *f* and *g* then

$$FT(f * g) = FT(f) \cdot FT(g) \quad <3>$$

where *FT* denotes a Fourier Transform and we represent multiplication with the symbol "·" to distinguish it from the convolution operation. Therefore, for our mapping problem, we can write:

$$FT(I_{obs}) = FT(B) \cdot FT(I) \quad <4>$$

The Fourier Transform of the map we can make, I_{obs}, is the product of the Fourier Transform of the true map and the Fourier Transform of our beam pattern *B*. Now the beam power pattern itself is related to the Fourier Transform of the electric field in the antenna aperture, and to make a long story short, FT(*B)* has a maximum spatial frequency of D/λ. Since FT(*B)* multiplies the spatial frequency information of our true map, *I*, it follows that the observed map, I_{obs}, can have no spatial frequency information at frequencies higher than D/λ.

5.1.3 Receiver Noise in the Map

Noise from the receiver will cause the signal we observe to fluctuate rapidly so that if we scan across the source we will see changes in signal that are due to receiver noise and not due to real spatial brightness variations in the source itself. Therefore, if I take the Fourier Transform of the observed map to see its spatial frequency information, I would typically find power apparent at spatial frequencies greater than D/λ. We know that the source brightness can't really change on very fine spatial scales (high spatial frequencies) in our map; these fluctuations are not an actual property of the source. Therefore, we'd like to use that information to eliminate contributions of noise at impossible spatial frequencies from the map.

An easy way to remove the effect of noise at impossible spatial frequencies is to: (1) take the Fourier Transform of our data map, FT(I_{obs}); (2) multiply FT(I_{obs}) by a function (we'll call it Π) with value 1 for all spatial frequencies less than D/λ and 0 for all spatial frequencies greater than D/λ; and then (3) Fourier Transform that result back to get a new estimate of I_{obs} without any noise which we know is not a real property of the source map. Using the convolution theorem, we can see:

$$FT(FT(I obs) \cdot \Pi) = I_{obs} * FT(\Pi) \quad <5>$$

and the operation we described is just a convolution of the Fourier Transform of the Π function with our observed map I_{obs}. The Π function just has unit value out to some radius (D/ λ) and its Fourier Transform is well known:

$$FT(\Pi) = \pi\left(\frac{D}{\lambda}\right)^2 \cdot 2\,\frac{J_1(2\pi r\, D/\lambda)}{2\pi r\, D/\lambda} \quad <6>$$

where r is the offset in sky position. If I remove the normalization term and consider r to be scaled so that r = r' λ/D, then our convolution function becomes

$$C(r') = 2\frac{J_1(2\pi r')}{2\pi r'} = 2\,jinc(2\pi r') \qquad <7>$$

5.1.4 Resampling the Map

If I convolve my observed map, I_{obs}, with the function *C* defined above, then I know that I have not removed any features from the map that could be physically present. Now I just need to be sure to sample the map, according to the Nyquist sampling theorem, at a rate that corresponds to twice the maximum spatial frequency that is possible in the data. This corresponds to a spatial frequency of 2D/λ, or a spacing of samples on the sky of λ/2D. I can sample on a finer grid than this as well, but it will not change the spatial frequency content (or resolution) of the map.

So, the above arguments may be used to explain OTF mapping procedure. We convolve the actual, irregularly spaced, data with function C and then sample the result on a new, regular, grid with grid spacing of λ/2D in order to preserve all the spatial frequency information that it is possible to have. In so doing, we are usually combining several data points to make an estimate of I_{obs} at each grid point. This can lead to an improved estimate of the map value of that grid point over what would have been obtained from a single sample.

5.1.5 Nearest Neighbor Gridding

The *jinc* filter approach sounds complicated, and people often wonder whether a simpler approach will work. A common alternative approach to making maps from a set of data points is to divide the sky into "cells" and average all data points that fall within a cell. What happens when you do this?

Following the above reasoning, it should be apparent that the nearest neighbor gridding procedure is actually a convolution of your map data with a box function with unit amplitude within the map cell. In spatial frequency this means multiplying the spatial frequencies of the data map by the Fourier Transform of the box. Let c be the width of the cell in units of λ/D, and u and v be the spatial frequencies in units of D/λ, then the Fourier Transform of the cell box, $FT_{cell}(u,v)$, may be be written:

$$FT_{cell}(u,v) = \frac{\sin(\pi c u)}{\pi c u} \cdot \frac{\sin(\pi c v)}{\pi c v} \qquad <8>$$

Thus, this procedure multiplies the above Fourier Transform of the cell box by the Fourier Transform of the map of the data. Since the this function has values at spatial frequencies higher than D/λ, the resulting map will include noise at spatial frequencies than can't exist. Therefore, the OTF filtering approach has been developed, and in this approach, we define a filter that eliminates the high spatial frequencies effectively. We will do more comparisons of the "nearest neighbor" and OTF filter approaches after dealing with some practical issues in optimizing

the OTF filter.

5.1.6 Dealing with Practical Problems

The main problem with the simplified OTF filter algorithm outlined above is that the *jinc* function (like the *sinc* function of the nearest neighbor algorithm) has lobes with significant values to great distances from the convolution point. Figure 1 shows the *jinc* function sampled out to 12 times λ/D. The spatial frequency response of the *jinc* convolution with three cutoff values (RMAX) of 3 λ/D, 6 λ/D, and 12 λ/D is shown in the right-hand panel. It is clear that the sidelobes of the *jinc* function lead to a nonuniform spatial frequency response.

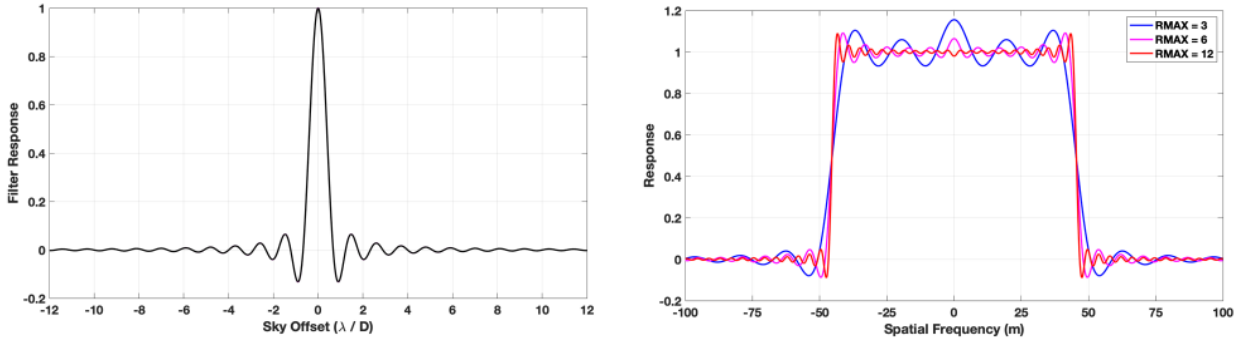


Figure 1 – (left) plot of the jinc function assuming uniform weight of spatial frequencies out to the diameter of the antenna. (right) Spatial frequency response for jinc truncated at different maximum distances (in units of λ/D).

Since It is impractical to carry out the convolution to infinite distance, practical filters have been developed to reduce the *jinc* sidelobes to a low level by multiplying them with a gaussian and/or some other function so that the sidelobe level is close to zero by the time the maximum distance for the convolution is reached, usually a few times λ/D. The equation (9) below gives a common function used in the FCRAO OTF mapping software. Figure 2 illustrates what the additional terms do as the original *jinc* is successively multiplied by a gaussian and then a second *jinc* chosen to reach its first zero at precisely the convolution cutoff radius, RMAX.

$$c(r') = 2\ jinc\left(\frac{2\pi r'}{a}\right)\ \exp\left(-\left(\frac{2r'}{b}\right)^{c}\right)\ 2\ jinc\left(\frac{3.831706\,r'}{RMAX}\right) \qquad <9>$$

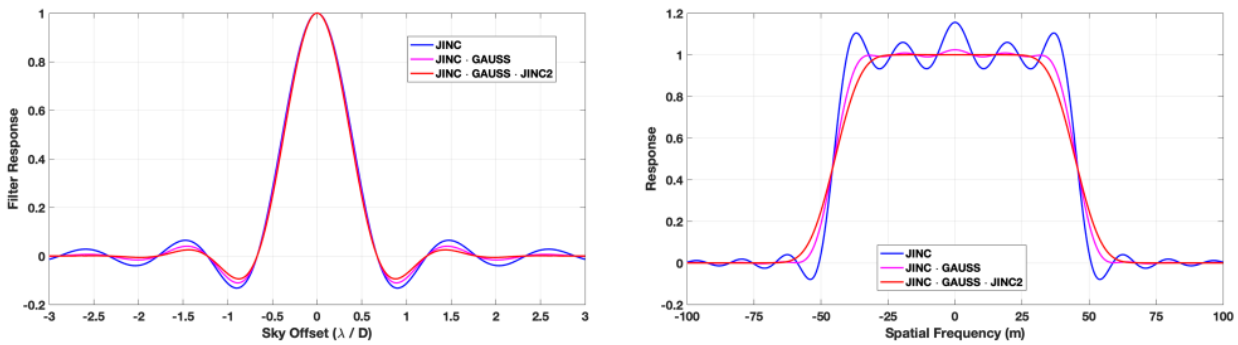


Figure 2 – (left) plot of jinc function (blue); product of jinc and gaussian function (magenta); and product of jinc, gaussian, and a second jinc scaled so that it's first zero is at the cutoff of the convolution function. In this case, the cutoff is at RMAX=3. (right) Spatial frequency response of the three filters. For these plots, the otf "a" parameter is 1.1, the otf "b" parameter is 4.75, and the otf "c" parameter is 2.

Each successive multiplication of the gaussian and the second *jinc* function has the benefit of reducing the first *jinc's* sidelobes and thereby reducing ripples in the spatial frequency response of the filter. However, each new function convolves the spatial frequency response with its Fourier Transform. These additional convolutions lower the response to the highest spatial frequencies and increase the response to noise that lies just outside the maximum spatial frequency that is possible in the map. Usually, a lower response at the highest spatial frequency is justified since the antenna illumination is tapered and the contribution from the edges of the dish is downweighted anyway. The response from just outside the maximum spatial frequency is tolerated since the process does succeed in removing responses at even higher spatial frequencies that would exist without the additional application of the gaussian and second jinc function.

5.2 Data Processing Steps

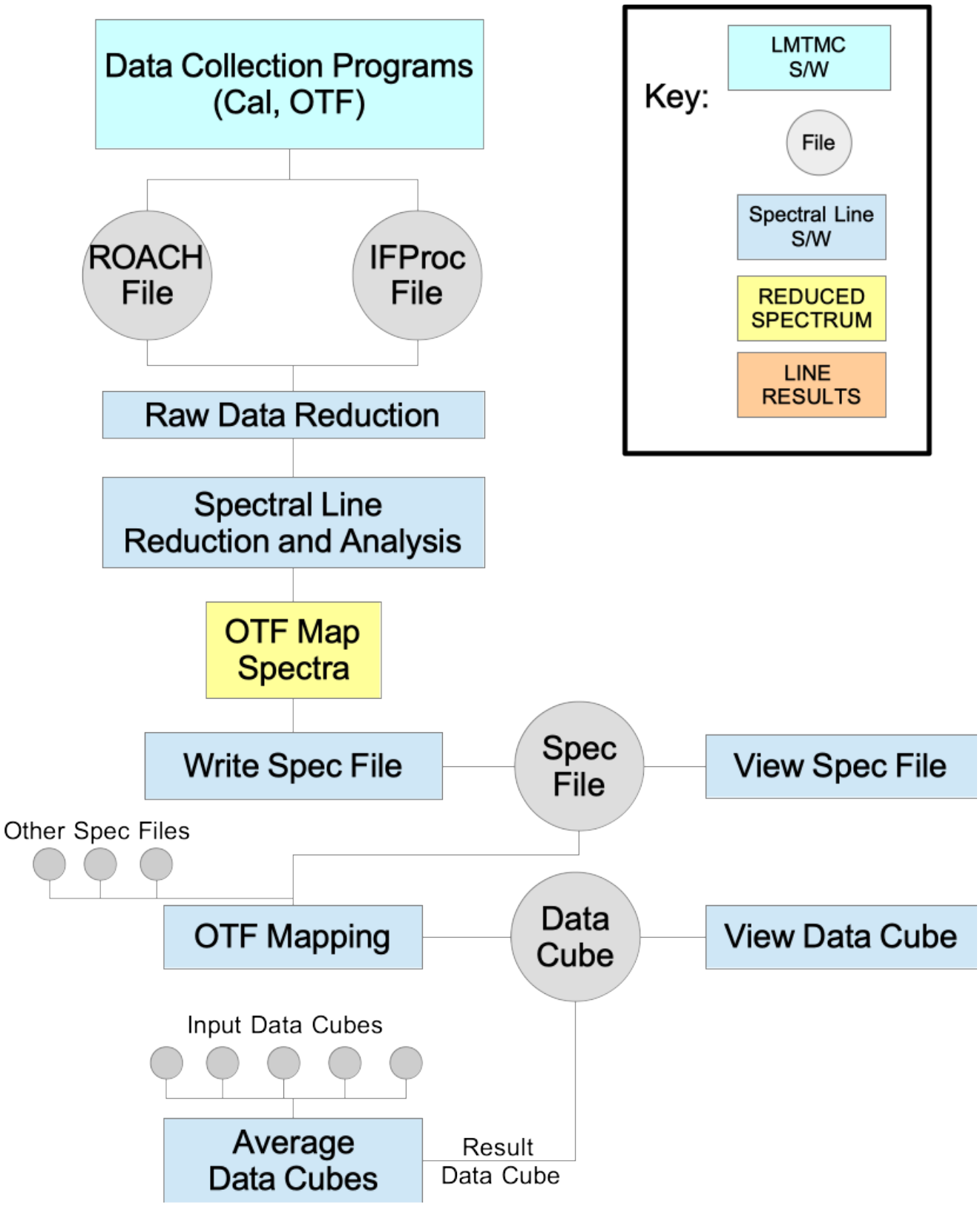
5.2.1 Overview

An overview of the process to create an OTF map is given in Figure __. The initial steps follow the procedure to reduce the raw IFProc and Roach files to create individual spectra for each "dump" of the spectrometer. Then the spectra may require baseline removal or line fitting to establish the properties to be mapped in addition to the values of the spectral channels.

In the LMT system, all spectra in an individual observation are taken with the same configuration of the spectrometer, so we may store all data in a two-dimensional data structure with one dimension representing the sequence of the observation and the second dimension representing the spectrum itself. Each spectrum in the sequence has timing and other information that will be required to make the map. All spectra correspond to a single set of velocities or frequencies depending on the processing at the "line" stage. The set of all spectra corresponding to a single observation can be written to a "SpecFile" for later analysis.

Next in the procedure is to read all reduced spectra from the SpecFIle and use the OTF filter to accumulate them into a regular, three-dimensional grid. The result of this procedure is to write the final data cube of information into a new file for viewing and analysis.

Finally, given multiple maps of the same region, it is useful to be able to combine the data cube from each map into a new data cube to improve the signal-to-noise ratio of the map.


Data Collection Programs (Cal, OTF)
ROACH File
IFProc File
Raw Data Reduction
Spectral Line Reduction and Analysis
OTF Map Spectra
Write Spec File
Spec File
View Spec File
Other Spec Files
OTF Mapping
Data Cube
View Data Cube
Input Data Cubes
Average Data Cubes
Result Data Cube
Key:
LMTMC S/W
File
Spectral Line S/W
REDUCED SPECTRUM
LINE RESULTS

5.2.2 Raw Data Reduction

Raw data files for an OTF map contain all spectra obtained during the map, including reference spectra and spectra which scan the map. The first step in the reduction involves reducing the raw data files to produce final, calibrated spectra to be processed into the final map product: a *data cube.* Section 3.6 describes the basic reduction process.

Use of the raw data for mapping requires certain properties of each spectra to be recorded:

Pixel ID
System Temperature
RMS
Map position in the “X” and “Y” coordinates.
Frequency data that may be needed for future line reduction steps

5.2.3 Intermediate Spectral Line File

The reduced spectra could, in principle, be filtered and added to the data cube one-by-one as they are reduced. However, in implementing a prototype of the procedure, I have found that it is useful to save an intermediate file with all spectra to be used in the map.

There are a number of options for the specific format of this intermediate data product. Currently, in the prototype software, the spectra are written in a netCDF file.

Creation of the intermediate file allows examination of the results in order to determine whether data need to be filtered in some way before inclusion into the data cube. For example, it may be useful to identify bad pixels in the array or set limits on the desired rms for individual spectra in order to eliminate bad spectra from the creation of the final data cube.

5.2.4 OTF Filtering Step

The OTF filtering step is described in detail in section 5.1. Each spectrum is assigned a weight for each desired map grid point, and all spectra that contribute to a grid point are averaged together. The weighting of a particular spectrum may depend on:

- RMS of the spectrum
- System Temperature of the receiver at the time of data acquisition
- Distance from the grid point, using the OTF filter weighting described in section 5.1

5.2.5 Writing the OTF Map Result to a File

The OTF data are accumulated into a data cube, which is then written into an output file for further analysis. The next section discusses data cubes, their properties, and the requirements for operating on them.

5.3 Dealing with Data Cubes

The result of the OTF mapping operation is the construction of a three-dimensional set of data with two of the dimensions representing position on the sky and the third dimension holding the set of spectra in the map. This data structure is called a *data cube*.

5.3.1 Data Cube Basics

5.3.1.1 Definitions

A data cube contains data which depend on three independent variables. At the LMT, a common use for data cubes is to hold spectral line maps, where the first two dimensions correspond to position on the sky and the third dimension of the cube holds the spectrum at each map point.

Figure 1 illustrates the concept of a Data Cube. We have individual elements of data arranged in three dimensions and indexed according to indices i, j, and k.

If `T` is a data cube in Python, we can refer to any individual data element of `T`, i,j,k, as `T[i,j,k]`, and using slices of the cube, we can refer to portions of the cube using the colon operator.

We may refer to planes cut through the data cube:

* j-k plane at index i - `T[i]` or `T[i,:,:]`
* i-j plane at index k - `T[:,:,k]`
* i-k plane at index j - `T[:,j,:]`

We may also reference 1D arrays within the cube:

* 1D array along axis k at location i,j - `T[i,j,:]`
* 1D array along axis j at location i,k - `T[i,:,k]`
* 1D array along axis i at location j,k - `T[:,j,k]`

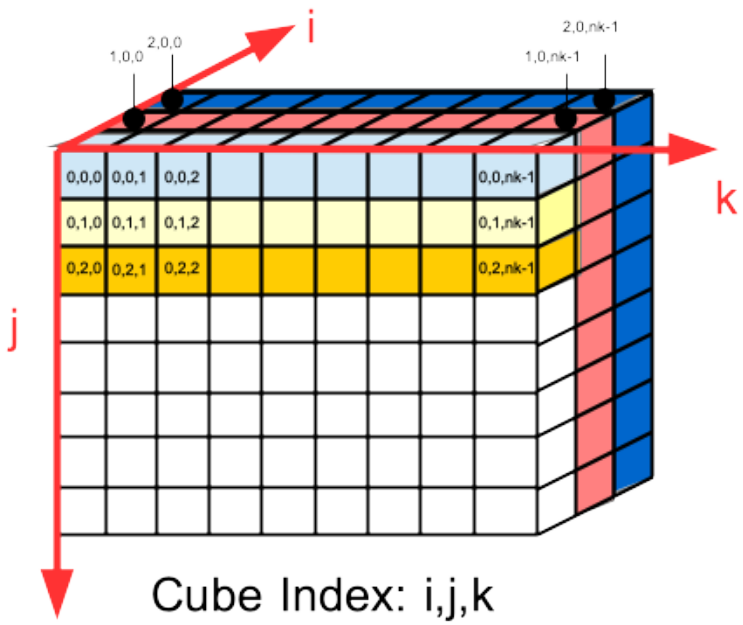


FIGURE 1 - Data Cube Illustration. The data cube is a three-dimensional set of data values. In astronomy the three dimensions are usually sky position (e.g. Right Ascension and Declination) and a third dimension for spectral information. A particular value in the data cube can be retrieved with three indices: i, j, k.

5.3.1.2 Data Cubes in Memory

Although the "data cube" is logically considered to be a 3-dimensional entity, it is actually stored in memory as a 1-D array. Figure 2 shows the correspondence between the data cube shown in figure 1 and the 1D array.

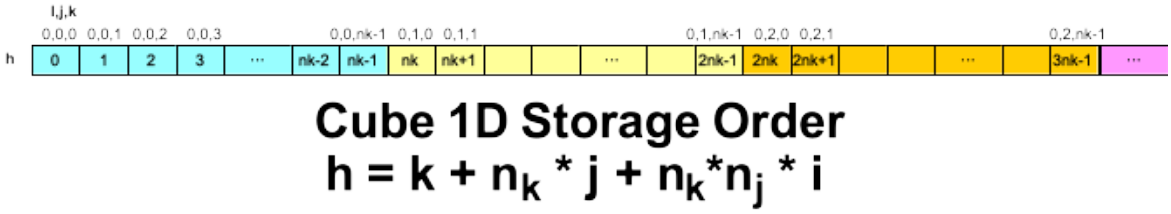


Figure 2 – One-dimensional storage of data elements in the Data Cube shown in Figure 1.

There is a simple indexing scheme that relates the two approaches. If axes i, j, k have dimensions n_i, n_j, and n_k, then the index to a particular element i,j,k in the 1D array is

$$h = k + jn_k + in_jn_k$$

In this scheme it may be noted that the k-axis data are stored consecutively for each row in the cube. I call this the "fast" axis for the cube since any row along this dimension can be read from a continuous block of memory. On the other hand, retrieval of consecutive elements along the j-axis or i-axis requires jumping in increments of n_k or n_k*n_j, respectively. I refer to these axes as the

"intermediate" and "slow" axes of the cube. The indices in our python data cube `T[i,j,k]` are then understood to be: i=slow; j=intermediate; k=fast.

5.3.1.3 FITS Conventions

A FITS Data Cube is stored in a particular order in memory. FITS axis 1 is the "fast" axis with values stored consecutively in memory. Axis 2 is the "intermediate" axis. In this case axis 1 and axis 2 form a 2D plane of data. Axis 3 (the 'slow' axis) then distinguishes these planes.

Comparing FITS convention (axis 1=fast; axis 2=intermediate; axis 3=slow) to Python (first index=slow; second index=intermediate; third index = fast) shows that there is some potential for confusion here.

5.3.1.4 Conventions in Astronomy

In many astronomical uses of images and cubes, the position is given using a longitude-like coordinate (e.g. Right Ascension) and a latitude-like coordinate (e.g. Declination). In radio astronomy, the third dimension is often given in velocity with respect to some reference frame (e.g. Local Standard of Rest). The third dimension can also be provided in frequency, again with respect to some reference frame (e.g. at the Observatory).

A common convention for programs using data cubes (e.g. astropy and glue) is to make longitude-like coordinate the fast axis of the cube and the latitude-like coordinate the intermediate axis. That means that the slow axis will be the spectral dimension of the cube.

5.3.1.5 Reordering the Cube

I note that the original data cubes written at the FCRAO do not follow the above convention commonly assumed by more modern programs. Moreover, one might imagine cases where cubes are originally made in a “non-standard” order for other reasons. If the axes are not in this order, then we might need to reorder them for programs that assume a particular order.

5.3.2 Parameter Estimation

Given a data cube of spectra, it should be possible to make estimates of the “line parameters” (peak value, integrated intensity, mean velocity, etc.) for all spectra in the cube in order to create images of the results.

5.4 Basic Data Cube Visualization

This is a big topic. In a simple package, there are a number of standard visualizations of the results that are useful:

- Maps – 2D representation of a property, either a single channel value or a

derived line parameter, over the sky.

- Channel Maps – A set of Maps with data for each channel of the spectrum
- “SV” Maps – 2D representation of the cube in which one of the dimensions is the spectral dimension and the other is the spatial dimension.
- Spectrum at a point in the map – plot of a spectrum at a designated point.

5.5 Writing Data Cubes

Once spectra are in data cube format, there are many existing analysis packages that could be used for visualization of the data and derivation of results. Thus, at this stage, it is important to be able to read and write data in this format. The most common format for this purpose is FITS.

APPENDIX A

Setting Frequency and Velocity Scales

A.1 BACKGROUND INFORMATION

A heterodyne receiver operates by converting a signal at high frequencies, which can be difficult to process directly, to much lower frequencies where further processing and signal detection are more straightforward. A common strategy is called "mixing" where in a device called a "mixer" converts a band of high frequencies into an equivalent bandwidth at a lower frequency. The mixer uses a single frequency tone, called the "Local Oscillator" (LO) to make the conversion. The output of the mixer is called the "Intermediate Frequency" (IF).

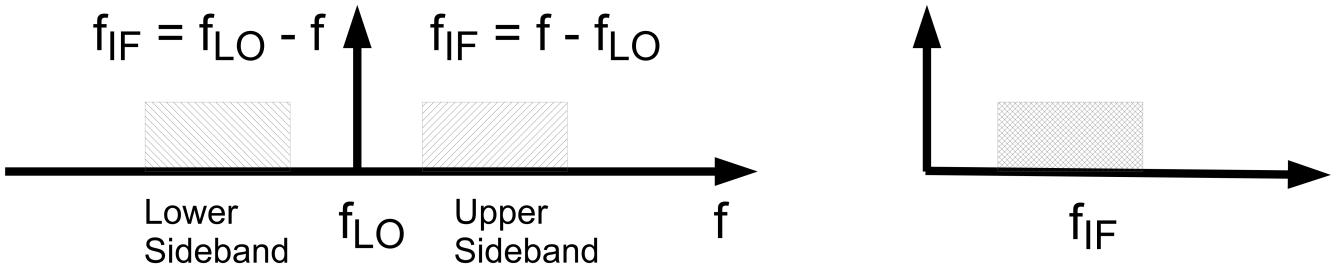


Figure A.1 – Conversion of frequency by a mixer. (left) original frequency, f, is converted to a new range of frequency known as the IF frequency. The IF is the difference between the original frequency and the LO frequency. Signals above and below the LO are both converted so that, unless some special filtering is done, the IF frequency band (right) has contributions from both the "upper sideband" range of frequencies and the "lower sideband" range of frequencies.

In an actual system, there may be several stages of mixing from higher to lower frequencies. Figure A.2 illustrates the situation when a second mixer is used to convert a portion of the signal in the first IF to a new frequency range, known as the second IF.

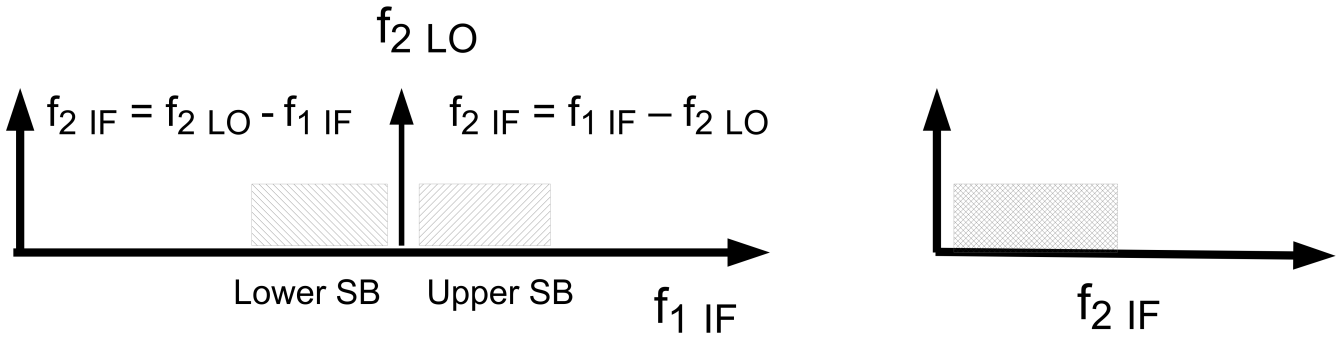


Figure A.2 – Conversion of frequency by a second mixer after in initial mixing resulting in conversion to a First IF band. The left figure shows the second LO ($f_{2\ LO}$) frequency within the first IF ($f_{1\ IF}$) band. The result of this mixing operation is to convert the first IF frequencies in the upper and lower sideband into a new range of frequencies, the second IF shown in the right hand panel. In general, frequencies from both upper sideband and lower sideband are combined in the new IF, but for the initial LMT processors, only the lower sideband frequencies are passed to the second IF.

In the initial LMT system there are three stages of frequency conversion, amplification, and filtering:

- The *Receiver* uses a “first” local oscillator to convert the original millimeter-wave frequency signal to a “first” IF frequency band.

- The *IF Processor* uses a “second” local oscillator to mix the first IF produced by the receiver down to “baseband”, which is defined to be a range of frequencies starting at 0 and extending to cover the full bandwidth for processing. In the present system, the bandwidth is approximately 1 GHz. The mixer for this stage is a single sideband mixer which rejects the upper sideband in the mixing process and only produces an output for the lower sideband.

- The *Baseband Processor* amplifies and filters the output of the IF Processor to provide the input to our spectrometer. The initial Baseband Processor produces frequency bands of 0-800 MHz, 0-400 MHz, and 0-200 MHz.

For *spectral line observations*, the IF Processor and Baseband Processor are totally passive devices. We simply set the frequencies and power levels. Data collection is carried out by the Spectrometer which analyzes the output signal from the Baseband Processor to produce a spectrum.

A.2 Definitions for Spectral Lines

In order to document the procedure for setting observing frequencies, there are several frequencies which need a brief definition:

- *Spectral Line Rest Frequency* – Each spectral line occurs at a specific frequency. This is called the rest frequency since it is the frequency of the line measured in the lab, where the relative velocity of source and observer is 0 (source is at rest).

- *Source Velocity* – Astronomical sources are in motion with respect to the telescope, and the Doppler shift that results from this relative motion must be accounted for in order to place the desired spectral line within the band of the spectrometer. Source velocity is given as the velocity of the source with respect to a particular coordinate system. Often this is with respect to the IAU-defined Local Standard of Rest system (LSR).

- *Observer Velocity* – This accounts for the motion of the observer with respect to a specific reference frame. Commonly, we reckon LMT's velocity with respect to the Local Standard of Rest. A number of motions are considered in this calculation, including rotation of the Earth, Earth's motion around the Sun, and Sun's motion with respect to the LSR.

- *Sky Frequency* – The Sky Frequency is the actual frequency being observed in the frame of reference of the receiver. Thus, a spectral line from a source

with Source Velocity, V, will occur at a Sky Frequency that accounts for Doppler Shifts due to the Source Velocity and the Observer Velocity.

A.3 Doppler Steering Description

The sky frequency of a specific spectral line from a specific source will vary during the time that an observation is being made. We often wish to maintain the frequency alignment of the spectrometer on a specific frequency in the source rest frame during the observation, and so it is necessary to continually update the sky frequency to account for changes in the observer velocity. The sky frequency can be updated in two ways. First we may continually update the First LO value so that the sky frequency is updated to the correct value. A second method, which is used in the SEQUOIA system, is to leave the First LO fixed in frequency and then update the Second LO to track the correct sky frequency.

There are several conventions in Astronomy for using the radial velocity of a source to set the sky frequency of a spectral line: (1) the *optical* definition, which is the preferred definition; (2) the *radio* definition, which is the definition formerly in use at the FCRAO 14m telescope; and (3) a relativistic definition, which accounts for special relativity and is sometimes adopted for very high velocities. Finally, we may wish to track a spectral line by simply entering its redshift. So this gives us four options. For the following, I have relied on a paper by Lindgren and Dravins (A&A 401 1185 2003) and on web pages from IRAM 30m and GBT describing what is done at those places.

Given measurements of observed and rest frequencies of spectral lines we may define the dimensionless redshift parameter, z, in terms of wavelength or frequency:

$$z = \frac{\lambda_{obs} - \lambda_{rest}}{\lambda_{rest}} = \frac{\nu_{rest} - \nu_{obs}}{\nu_{obs}}$$

Using the “optical definition”, the redshift can be converted to a conventional velocity according to:

$$V_{optical} = c\ z$$

Using the “radio definition”, the redshift is converted to velocity with a different formula:

$$V_{radio} = \frac{c\,z}{1+z} = c\left(\frac{\nu_{rest} - \nu_{obs}}{\nu_{rest}}\right)$$

The fact that these are not identical means that some care in definitions is called for. At the IRAM 30m, they use the optical definition for all calculations to set the

sky frequency to be observed. At the GBT, they allow either definition to be used. LMT follows the GBT approach and allows the observer to pick the value most appropriate to his/her project. Thus, an observer must specify:

- **velocity of source** – this is the number that will be used for the calculation of sky frequency with respect to choice of frame and according to definition selected.

- **reference frame for the source velocity** – This selects the rest frame for the above velocity. Options are topocentric, solar system barycenter, and local standard of rest.

    - Topocentric Definition: no corrections for motion of observer with respect to solar system barycenter or for other motions. Any velocity entered is used to set the sky frequency directly.

    - Solar System Barycenter Definition: correction for motion of the observer with respect to solar system barycenter, but no other corrections are applied. This correction is computed in the LMT system from the JPL ephemeris.

    - LSR Definition: present definition of LSR is 20 km/s toward RA=271 degrees; DEC=30 degrees in J2000 wrt the solar system bary center. This is the original definition of Gordon (1975) updated to the J2000 system. I note that the LSR correction is computed routinely by the LMT system.

- **velocity definition** – This picks the formula to be used to calculate sky frequency from the total radial velocity, V, with respect to the receiver (topocentric system) . Options here would include optical definition, radio definition, and relativistic definition. I would pick, as default, the optical definition since that is what is done at the IRAM 30m. Here are the sky frequency calculations for total radial velocity V according to the different definitions:

$$\nu_{sky}(V) = \nu_{rest}\left(\frac{1}{1+V/c}\right) \quad OPTICAL$$

$$\nu_{sky}(V) = \nu_{rest}\left(1-\frac{V}{c}\right) \quad RADIO$$

$$\nu_{sky}(V) = \nu_{rest}\left(\frac{\sqrt{1-V^2/c^2}}{1+V/c}\right) \quad RELATIVISTIC$$

The equation for sky frequency in a system with two LO's :

$$\nu_{sky} = \nu_{1\mathrm{LO}} + SB_{1\mathrm{LO}}\left(\nu_{2\mathrm{LO}} + SB_{2\mathrm{LO}}\,\nu_{2\mathrm{IF}}\right)$$

where $SB_{1\,LO}$, $SB_{2\,LO}$ are +1 for USB and -1 for LSB. Our goal with SEQUOIA is to set the second LO, $\nu_{2\,LO}$, so that the value of ν_{sky} computed by the system falls at a particular frequency in the second IF, $\nu_{2\,IF}$. I note that $SB_{2\,LO}$ is always -1 in the LMT system at this time, so that sky frequency *decreases* with 2nd IF frequency (channel number) if we are observing the USB of the 1st LO.

Many of the relevant frequency parameters are supplied in the IFProc Header. These are summarized in Table A.1.

SETTING THE FREQUENCY AXIS

Sky Frequency Scale

With the spectrometer operating in a mode with *N* channels and a bandwidth of *B* MHz, we define the reference channel of the spectrometer, C_0, and the channel spacing, d ν / d chan, given by

$$C_0 = \frac{N-1}{2}$$

$$\frac{d\nu}{d\,chan} = SB\ \left(\frac{B}{N}\right)$$

where SB = 1 is the observation is in the lower sideband of the first LO and SB = -1 if the observation is in the upper sideband of the first LO. The frequency in the sky (or topocentric) frame of channel i is then:

$$\nu(i) = (i - C_0)\frac{d\nu}{d\,chan} + \nu_{sky}$$

where ν_{sky} is the sky frequency calculated

Nominal Velocity Scale

The sky frequency is calculated to provide to track a desired velocity given in the specific frame of rest at channel C_0 in the spectrometer. Unless other offsets are also given, this desired velocity in this frame is provided as the “source velocity”, V_{source} . The channel spacing in velocity units is related to the channel spacing in frequency units:

$$\frac{dV}{d\,chan} = -\left(\frac{c}{\nu_{rest}}\right)\ \frac{d\nu}{d\,chan} \qquad RADIO\ DEFINITION$$

$$V(i) = (i - C_0) * \frac{dV}{d\,chan} + V_{source}$$

Frequency and Velocity Scales in Other Frames of Reference

Each observation records two useful velocities for conversion between the different velocity frames that can be specified.

- V_{OBS} – is the line of sight velocity of the LSR in the topocentric system.
- V_{BARY} – is the line of sight velocity of the LSR in the solar system barycenter frame.

Note that in the above, V_{OBS}-V_{BARY} is then the line of sight velocity of the solar system barycenter in the topocentric frame.

In addition, it is important to consider any frequency offsets given by the Observer when calculating the frequency or velocity scale.

In the sections below, I summarize the formulae to convert between possible frequency or velocity scales:

- Frequency Scales
  - Sky Frequency – frequency in topocentric frame
  - LSR Frequency – frequency in frame of LSR.
  - Barycenter Frequency – frequency in the frame of solar system barycenter.
  - Source Frequency – frequency in the frame of the source assuming input velocity of source (line appears at line rest frequency)
- Velocity Scales
  - Topocentric Velocity – velocity in topocentric frame
  - LSR Velocity – velocity in LSR frame
  - Barycenter Velocity – velocity in Barycenter frame
  - Source Velocity – velocity in frame of the source assuming input velocity of source (line appears at 0 velocity)

The exact formulae depend on the frame used to track the source, so I collect formulae below for the cases of LSR frame and Barycenter frame. In each case, we include a term to account for a possible added frequency offset ν_{off} provided by the observer.

$$V_{off} = -\frac{c}{\nu_{rest}} \nu_{off}$$

Data Collected by tracking in the LSR frame:

$$V_{LSR}(i) = (i - C_0)\frac{dV}{d\,chan} + V_{source} + V_{off}$$

$$V_{BARY}(i) = (i - C_0)\frac{dV}{d\,chan} + V_{source} + V_{BARY} + V_{off}$$

$$V_{SKY}(i) = (i - C_0)\frac{dV}{d\,chan} + V_{source} + V_{OBS} + V_{off}$$

$$V_{source}(i) = (i - C_0)\frac{dV}{d\,chan} + V_{off}$$

$$\nu_{sky}(i) = (i - C_0)\frac{d\nu}{d\,chan} + \nu_{sky}$$

$$\nu_{LSR}(i) = (i - C_0)\frac{d\nu}{d\,chan} + \nu_{sky} + \frac{\nu_{rest}}{c}V_{OBS}$$

$$\nu_{BARY}(i) = (i - C_0)\frac{d\nu}{d\,chan} + \nu_{sky} + \frac{\nu_{rest}}{c}(V_{OBS} - V_{BARY})$$

$$\nu_{source}(i) = (i - C_0)\frac{d\nu}{d\,chan} + \nu_{sky} + \frac{\nu_{rest}}{c}(V_{OBS} + V_{source})$$

Data Collected by tracking in the Solar System Barycenter frame:

$$V_{LSR}(i) = (i - C_0)\frac{dV}{d\,chan} + V_{source} - V_{BARY} + V_{off}$$

$$V_{BARY}(i) = (i - C_0)\frac{dV}{d\,chan} + V_{source} + V_{off}$$

$$V_{SKY}(i) = (i - C_0)\frac{dV}{d\,chan} + V_{source} + V_{OBS} - V_{BARY} + V_{off}$$

$$V_{source}(i) = (i - C_0)\frac{dV}{d\,chan} + V_{off}$$

$$\nu_{sky}(i) = (i - C_0)\frac{d\nu}{d\,chan} + \nu_{sky}$$

$$\nu_{LSR}(i) = (i - C_0)\frac{d\nu}{d\,chan} + \nu_{sky} + \frac{\nu_{rest}}{c}V_{OBS}$$

$$\nu_{BARY}(i) = (i - C_0)\frac{d\nu}{d\,chan} + \nu_{sky} + \frac{\nu_{rest}}{c}(V_{OBS} - V_{BARY})$$

$$\nu_{source}(i) = (i - C_0)\frac{d\nu}{d\,chan} + \nu_{sky} + \frac{\nu_{rest}}{c}(V_{OBS} - V_{BARY} + V_{source})$$

TABLE A.1 – IFProc HEADER PARAMETERS FOR FREQUENCIES

Parameter		IFProc Header Variable Name (RX = name of receiver in use)	Units	Array Dimension
Line Rest Freq.	ν_{rest}	Header.RX.LineFreq	GHz	2
Source Velocity	ν_{source}	Header.Source.Velocity	km/s	1
Source Velocity Frame		Header.Source.VelSys	"VLSR" or "BARY"	1
Doppler Track Flag		Header.RX.DopplerTrack	1 or 0	1
LSR Velocity wrt Observatory	V_{OBS}	Header.Sky.ObsVel	km/s	1
LSR Velocity wrt SS Barycenter	V_{BARY}	Header.Sky.BaryVel	km/s	1
Sky Frequency	ν_{sky}	Header.RX.SkyFreq	GHz	2
1st LO Frequency		Header.RX.Lo1Freq	GHz	1
2nd LO Frequency		Header.RX.Lo2Freq	GHz	2
1st IF Frequency		Header.RX.If1Freq	GHz	2
2nd IF Frequency		Header.RX.If2Freq	GHz	2
Synthesizer Harmonic		Header.RX.SynthHarm	GHz	2
Synthesizer Frequency		Header.RX.SynthFreq	GHz	2
		Header.RX.SideBand1LoType		2
		Header.RX.SideBand2LoType		2
1st LO Sideband		Header.RX.SideBand1Lo	USB or LSB	2
2nd LO Sideband		Header.RX.SideBand2Lo	USB or LSB	2
Velocity Definition		Header.RX.VelocityDefinition	Optical or Radio	
Line Offset	ν_{off}	Header.RX.LineOffset	GHz	
Redshift (Z)		Header.RX.LineRedshift	Z	2

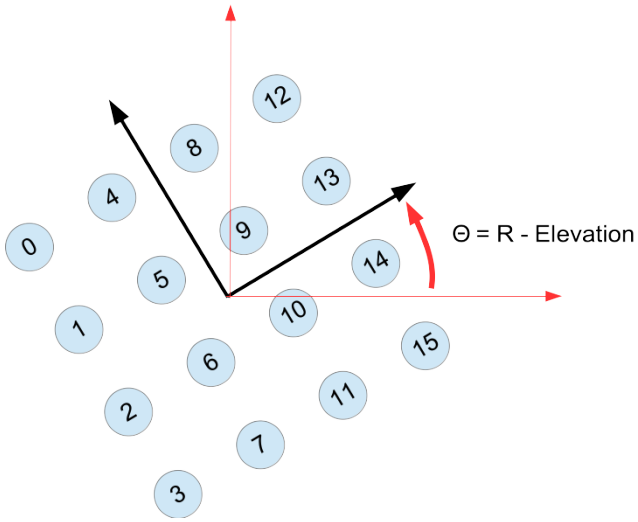
APPENDIX B

GEOMETRY OF MAPPING WITH FOCAL PLANE ARRAYS

Introduction

Maps are constructed by pointing the telescope to different positions on the sky or scanning the receiver beams over the sky. The telescope positions that are recorded during spectral line mapping refer to a single location on the sky. Therefore, for reconstructing a map taken with a focal plane array, we must find the position offsets of the individual beams from the point that is being tracked. Nominally, when making a map, one would instruct the telescope to track the geometric center of the array. However, it is also possible to have the telescope track any of the array's individual pixels.

Description of Array Geometry



The SEQUOIA array is a 4x4 regular grid. The spacing (S) of the pixels is fixed with a value that has been determined to be 27.9 arcsec. The grid orientation on the sky varies with the elevation angle (El). At zero degrees elevation, the “X” axis of the array is rotated counter clockwise by 46 degrees. We refer to the rotation angle of the grid at zero degrees elevation as R.

The angle, Θ, between the “X” axis of the array and the positive azimuth direction on the sky is given by

$$\theta = R - El$$

Beam Offsets

Given the above description of the grid geometry, it is possible to solve for the offsets of the beams with respect to the center. Each beam is given a position within the grid in units of the grid spacing with respect to the geometric center.

Let GXi, GYi be the grid location of pixel i (e.g. GXi=-1.5, GYi=+1.5 for pixel 0), then

$$DAZ_i = S \left(GX_i \cos(R-El) - GY_i \sin(R-El) \right)$$
$$DEL_i = S \left(GX_i \sin(R-El) + GY_i \cos(R-El) \right)$$

where

$$DAZ_i = \Delta Az_i \cos(El)$$
$$DEL_i = \Delta El_i$$

It is not unusual to track a given pixel in the grid (often pixel 10). In this case, the telescope position that is recorded is that of pixel 10 and all other pixel locations are offset from that position by

$$DAZ'_i = DAZ_i - DAZ_{10}$$
$$DEL'_i = DEL_i - DEL_{10}$$

Mapping in the RA/Dec system places a further requirement on the position calculation. The Azimuth and Elevation offsets on the sky must be rotated by the parallactic angle of the source. For parallactic angle P, the RA and Dec offsets

$$DRA_i = -DAZ_i \cos(P) + DEL_i \sin(P)$$
$$DDEC_i = DAZ_i \sin(P) + DEL_i \cos(P)$$

where

$$DRA_i = \Delta\alpha \cos(\delta)$$
$$DDEC_i = \Delta\delta$$

The source parallactic angle is provided in the IFProc file header or can be computed from first principles based on position of the source on the celestial sphere and time of observation.

APPENDIX C

OTF SIMULATIONS

In order to demonstrate the use of the convolution for noise averaging and data "gridding", as described in section 5.1, I've carried out a set of Monte Carlo simulations of the map of a point source. The resolution (λ/D) is 14", corresponding to the 50m LMT at a wavelength of approximately 3.4mm. The source is a 16.1" HPBW Gaussian, corresponding to the beam size expected for the 10 db taper as with SEQUOIA. The map data are sampled a 1" intervals along the scan (X) direction in the map, with rows spaced by 6.65" (0.95 times Nyquist spacing). The map cells are at the Nyquist scale of λ/(2D), corresponding to 7". Figure C.1 (left) shows one of the trial maps in the Monte Carlo simulation using the nominal OTF parameters. The right hand panel shows the result of using the nearest neighbor algorithm.

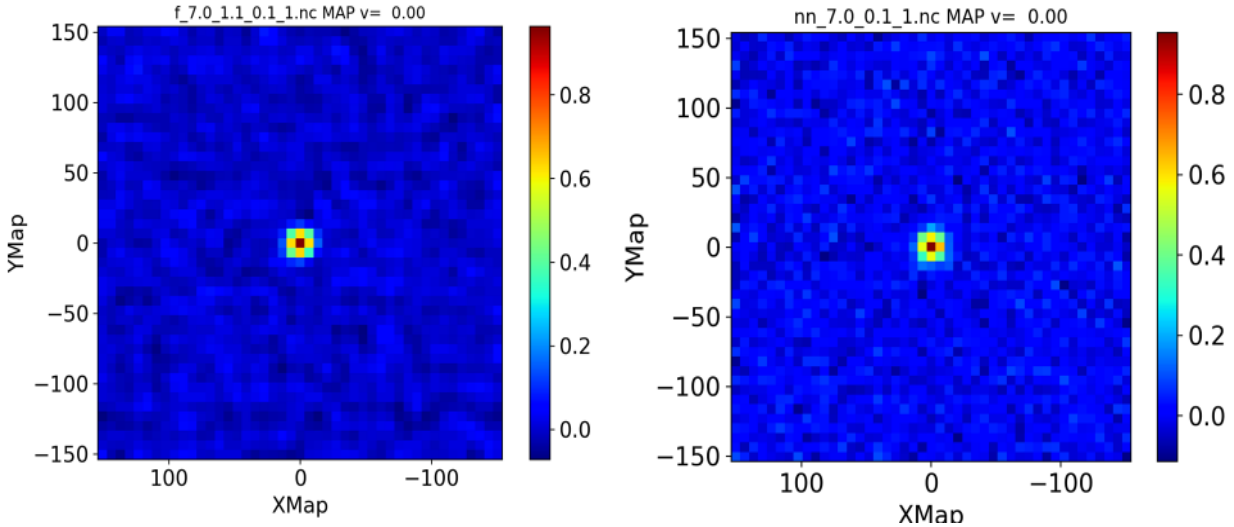


Figure C.1 – Trial map observing a point source with 16.1" HPBW given resolution (λ/D) of 14". The cell dimension is λ/2D = 7". Noise level in an individual simulated spectrum is 0.1 of map peak amplitude. The left panel gives the map constructed with the nominal OTF filter. The rms of a channel in this map is 0.0237. The right map is constructed using the nearest neighbor algorithm. The rms of a channel in this map is 0.0380.

The simulated data are convolved with the filter described by Equation 9 of section 5.1. For this experiment, we explore the effect of the OTF "a" parameter and maintain nominal values of the b (4.75) and c (2.0) parameters. We adopt RMAX of 3 for all cases. Figure C.2 shows the filter and spatial frequency plots for three different values of OTF "a". It is notable that the value a=0.8 corresponds to a case where the filter response is uniform over nearly the entire aperture. However, this case includes a significant response from beyond the aperture. The case a=1.4 limits the spatial frequency response by about a factor of 2, but has the virtue of having no response to noise at spatial frequencies outside of the aperture. The case a=1.1 is the nominal value used at FCRAO for many years, which represents a compromise between these cases.

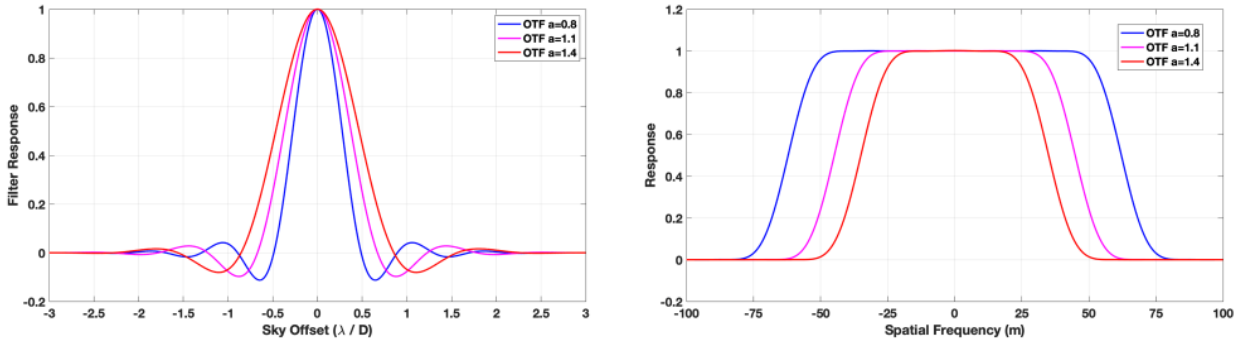


Figure C.2 – (left) filter function for simulations assuming three different values of the "a" parameter. (right) Spatial Frequency response of the three filters.

Ten Monte Carlo trials were performed for values of OTF "a" between 0.8 and 1.4. The data for the trials consist of a 64-channel spectrum with rms noise of 0.1K. The amplitude of the simulated line in the spectrum follows a gaussian model for the source with a HPBW of 16.1 arcsec, corresponding to what is seen for a point source at LMT at 86 GHz. For each trial, we may derive an rms for the map and the properties of a gaussian fit to the model source. Figure C.3 shows the result of the simulations. The left hand panel shows that as the width of the filter function increases the map rms decreases, which is to be expected since greater numbers of data points are included with the wider filter. Broadening the filter does have other effects, however. The center panel of Figure C.3 shows that the peak amplitude of the derived point source is decreased as filter width increases. The peak amplitude is estimated in three different ways here. The first method (Peak Pixel) is just to take the derived value of a pixel centered on the source. In the second method, we perform a least squares fit to a gaussian to derive the amplitude, assuming that the position and half power width of the gaussian are known (constrained fit). Finally, all parameters of the gaussian are determined (unconstrained fit) to estimate the amplitude of the source. The right hand panel shows the derived half power width of the gaussian fits for the unconstrained fit. In this case, the derived width increases with the increase in the width of the filter.

Figure C.4 looks at two other properties of the simulation. In the left hand panel, we compute the integrated flux under the unconstrained gaussian fit (presented here as the product of the amplitude and the two half power widths normalized by the value for the simulated source). The plot shows a trend of a few percent as the width of the filter is increased, though the magnitude of the bias is much reduced compared to the changes seen in the amplitude values alone. The right panel shows the signal-to-noise ratio in the map reckoned in two ways: (1) the Peak case is the ratio of the Peak pixel value divided by the map rms; and (2) the Amplitude case is the ratio of the amplitude to its error in the least squares fit. It is notable that the SNR increases in both cases as the filter width is increased.

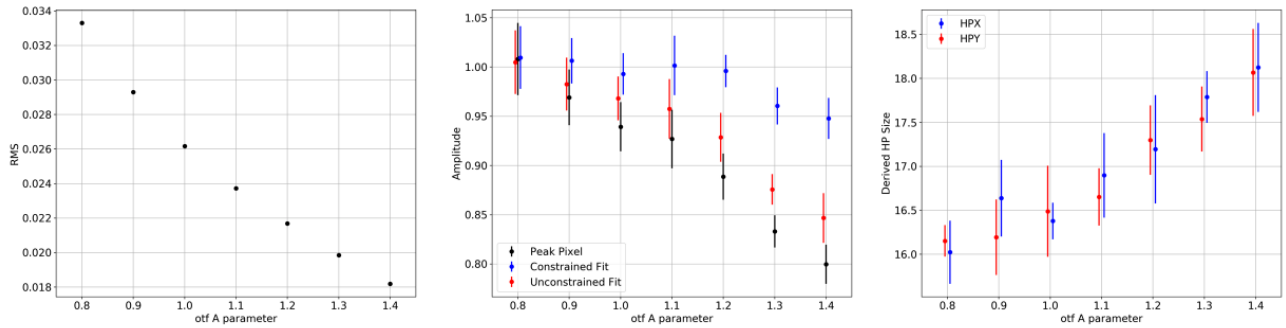


Figure C.3 – (left) Map rms as a function of the OTF "a" parameter. Note that larger values, corresponding to integration over more data points, lead to smaller rms. (center) Amplitude of the simulated source as a function of the OTF "a" parameter. Amplitude is found under three algorithms: (1) Peak pixel value; (2) least squares gaussian fit assuming position and beam HPBW set to nominal value (constrained fit); and (3) least squares gaussian fit allowing position and HPBW to be free parameters (unconstrained fit). Error bars represent the standard deviation of all trials. (right) Derived source HPBW as a function of OTF "a" parameter. The simulated source has an HPBW of 16.1" Error bars shown represent the standard deviation of all trials.

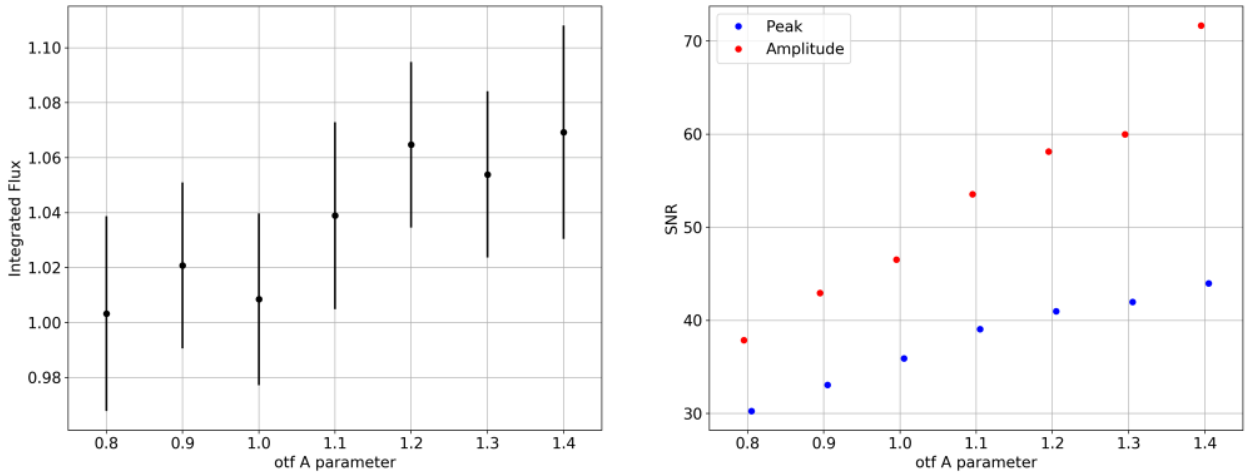


Figure C.4 – (left) Results for integral of the flux over the gaussian beam for the unconstrained fit. Error bars show the standard deviation of the Monte Carlos results. We note that the bias in the integrated flux with "a" is much less than that seen in the determination of the amplitude alone. (right) Signal-to-Noise ratio (SNR) versus the OTF "a" parameter. The Peak estimate is the peak pixel value divided by the rms for the map. The Amplitude estimate uses the average amplitude normalized by the average amplitude error of the trials.

Finally, it is straightforward to compare results of applying the OTF filter convolution with the nearest neighbor convolution in order to assess the possible differences. Figure C.5 shows a comparison of the filter functions and their spatial frequency response. It is clear that the nearest neighbor function has significant response at spatial frequencies beyond the extent of the antenna.

Table C.1 shows the results of parameters derived from 10 Monte Carlo trials of map simulations. The Map RMS, which is derived from regions of the map with no signal, shows a smaller value for the OTF filtered map, as might be expected since

it allows more data to be combined for each grip point.

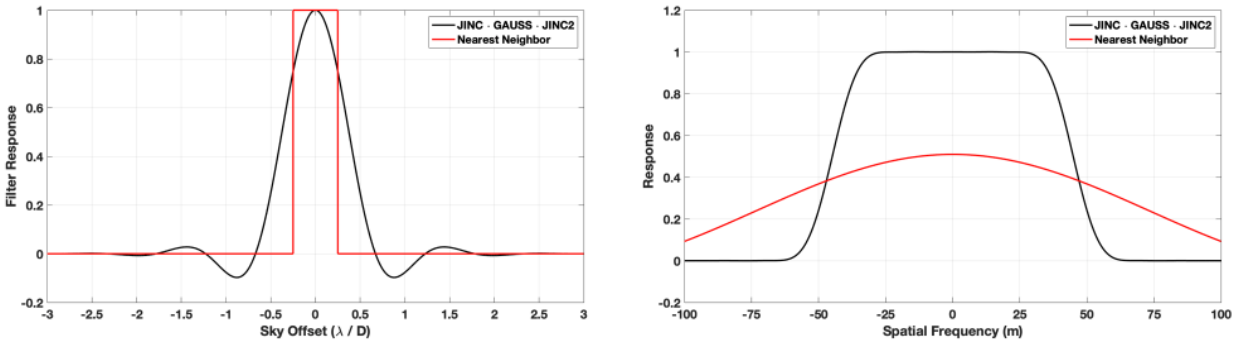


Figure C.5– (left) comparison of the OTF filter and the nearest neighbor (Nyquist sized box) filter functions. (right) Spatial frequency response of the filters.

Estimation of the signal-to-noise ratio in the map can be subtle. We derive the signal from various possible fits to the source amplitude, including Peak pixel flux, result of constrained model fit, and result of unconstrained model fit. The “noise” is estimated in two ways: (1) from the formal fit errors of the least squares fit; and (2) from the standard deviation of the parameter derived from the Monte Carlo trials.

Simulation Results

The map rms should be related in a simple way to the number of points per map cell. In this case, the simulation used 7” cells with samples along a row spaced by 1”. Rows were spaced by 0.95 * 7”, so most cells would have had samples from just one row. We'd therefore expect ~7 points in each cell. Nominally, the rms of the cell should be 1/sqrt(7) less than the 0.1 rms of an individual sample, corresponding to 0.0378. This is very close to the nearest-neighbor value (0.0380), as one might expect.

The map rms with the OTF filter is less than that seen for the nearest-neighbor filter. This may be understood since the OTF filter uses more spectra than the nearest neighbor filter to compute a grid point. We can use the improvement in the map rms to compute an effective increase in the integration time for our map of 2.6. This is pretty close to values predicted in the Mangum et al paper of approximately 3, so we see the expected improvement in the noise level with the filter.

The OTF filter has an effect on the value of the peak pixel in the map. The OTF filter gives a peak pixel value that is 96% of the peak pixel value of the nearest neighbor map. However, when you compute signal-to-noise ratio, you find that the SNR is still much better with the OTF filter value (39 compared to 25).

Another way to look at the SNR for the peak pixel is to consider the scatter of the peak pixel values in the 10 Monte Carlo trials. In this case, we see a bit more

scatter in the OTF result, which reduces the SNR. Comparison to nearest-neighbor still gives a better result: 31 compared to 24.

In a real case, estimation of the source flux night involve a fit of a gaussian to the map. I've carried this out using two assumptions. In the first case, we use a gaussian with a known half power width, corresponding to the expected size for a point source. This is what I call the “constrained fit” in the table. Using the constrained fit returns the correct flux for the source. Then we can use two approaches to estimate the noise. In the first approach, I just use the error in the amplitude returned by the fit routine. In the second approach, I use the standard deviation of the results of the amplitude fit. SNR's reckoned from the fit results are better for the OTF filter, but using the other approach gives a better SNR for the nearest-neighbor approach. *(I don't understand this result!)*

Finally we can fit the source using a model where the HPW of the gaussian is unconstrained. This has the advantage of producing a result for the HPW of our source. The two filters give identical results for the amplitude and beam size. The amplitude is about 4% lower than the known amplitude. The derived beam sizes, in both cases, are about 5% larger indicating no practical difference in resolution for the two methods. Once again SNR based on fit errors is better (54 compared to 33) in favor of the OTF filter. When one considers the standard deviation of the fit amplitudes in the MC trials, however, there is no clear SNR advantage.

Acknowledgements

Thanks to Mark Heyer for review and comments.

PARAMETER	Nominal OTF	Nearest Neighbor
MAP RMS	0.0237	0.0380
Peak Pixel		
Average Value	0.927	0.965
Map Error	0.024	0.038
SNR (Amp/Error)	39	25
Value STD from MC	0.030	0.040
SNR (MC Simulation)	31	24
Constrained Fit		
Average Amp	1.002	1.003
Amp Error	0.013	0.022
SNR (Amp/Error)	75	46
Amp STD from MC	0.030	0.014
SNR (MC Simulation)	33	72
Unconstrained Fit		
Amp	0.958	0.958
Amp Error	0.018	0.029
SNR (Amp/Error)	54	33
Amp STD from MC	0.030	0.030
SNR (MC Simulation)	32	32
HPX	16.90	16.86
HPX STD from MC	0.48	0.46
HPY	16.65	16.73
HPY STD from MC	0.33	0.44

Table C.1 – Comparison of results for Nearest Neighbor convolution and OTF Filter approach in Monte Carlo Simulation. The OTF filter uses the nominal FCRAO parameters a=1.1, b=4.75, c=2.0, RMAX=3.0.