Measurement Sets and the Sky Intensity

Russell O. Redman

1993 January 7

In this document we will discuss the mathematical nature of telescopic measurements of the intensity of radiation from the sky, focusing our attention upon intensitometers, that large class of instruments which perform multichannel measurements of the sky intensity. The discussion will summarize and expand upon the conclusions of the November meeting of the AIPS++ design team, making explicit connections between the mathematical structures and the object diagrams within AIPS++. Throughout this document, significant program objects (such as Measurement, MeasurementSet and MeasurementEquation) will be capitalized to distinguish them from more normal English uses of the same words.

Measurement Equations

We may represent the intensity of radiation from the sky as $\vec{S}(\nu, t, \alpha, \delta)$ where $\vec{S}$ is a Stokes vector (I, Q, U, V). In this expression we will refer to the independent variables as the "radiation coordinates", representing them collectively as the vector r with n-dimensional volume element dr. Technically, the polarization components are also radiation coordinates, and they will be included as such in the AIPS++ software, but their discrete nature makes them easier to deal with symbolically as vectors, with a summation over the components (implicit in a dot product) taking the place of an integral over dr. The set of functions $\vec{S}(r)$ which represent the sky intensity form a Hilbert space $S_I = \{\vec{S}(r), *\}$ with the product

$$\vec{F} * \vec{G} = \int \vec{F}(r) \cdot \vec{G}(r)\, dr. \tag{1}$$

After an intensity measurement has been calibrated its value may generally be represented as

$$s = \int \vec{P}(r) \cdot \vec{S}(r)\, dr = \vec{P} * \vec{S}. \tag{2}$$

Equation (2) is referred to as the measurement equation. The function $\vec{P}$ is the point spread function (PSF) of the measurement in radiation coordinates, and within the program will be called the MeasurementEquation. It defines explicitly such quantities as the center frequency of the channel, the start and stop times of the integration, and the polarization sensitivity of the receiver. Conceptually, we may consider a calibrated Measurement to consist of a pair $(s, \vec{P})$ giving the measured sample value and a precise description of how it was measured. More concretely, the calibrated Measurement consists of the measured value s with enough descriptive parameters to reconstruct the function $\vec{P}$.

A single, isolated measurement is rarely of much interest in astronomy. The instrumentation usually packages large, well-defined blocks of measurements, such as spectra or images, which move through the system as units. As our long discussions have revealed too clearly, it is important at this very abstract level of discussion to avoid assigning names to these blocks whose physical meanings are understood clearly, but differently, in different fields of astronomy. Let us therefore refer to a well-defined MeasurementSet which will travel through the data processing system as a unit by the term MeasurementBlock.

The Measurements in a MeasurementBlock $\{(s_i, \vec{P}_i)\}$ will normally have some regular structure imposed by the instrumentation, so that the $\vec{P}_i$ differ only in one or two parameters. It will often be useful to work with the entire set of functions $\{\vec{P}_i\}$ which for convenience will be refered to as the MeasurementEquation of the MeasurementBlock.

II - MeasurementBlocks and Projection Operators

After we have made a measurement, how much can we reconstruct of the sky intensity $\vec{S}(r)$? We will start this discussion with the observation in this section that any linear measurement of the intensity defines in a natural way a projection operator from the full Hilbert space S_I onto a finite dimensional subspace of S_I. We will show how to construct this projection and examine a few of its more important properties.

Consider a MeasurementBlock $B = \{(s_i, \vec{P}_i), i = 1, N\}$. Within the Hilbert space S_I of all possible sky intensities the functions $\vec{P}_i$ span a linear subspace $S_B = \{\sum F_i \vec{P}_i(r)\}$ which is optimally matched to the measurement process. Any function $\vec{F} = \sum F_j \vec{P}_j$ in S_B can be reconstructed from its measured values f_i by noting that

$$f_i = \vec{P}_i * \vec{F} = \sum \left(\vec{P}_i * \vec{P}_j\right) F_j = \sum P_{ij} F_j, \tag{3a}$$

where

$$[P_{ij}] = \left[\vec{P}_i * \vec{P}_j\right]. \tag{3b}$$

To avoid breaking the flow of the discussion, let us assume that the functions $\vec{P}_i$ are linearly independent (we will return to this point in a later section). We can then recover the original function $\vec{F}$ by solving this set of equations for the F_j.

Although the functions $\vec{P}_i$ span S_B, they are not the natural basis for S_B as a subspace representing the measurements. By choosing the correct basis for S_B we can arrange that the coefficients of the basis vectors are given directly by the measured samples; the discussion above shows that this natural basis is

$$\vec{p}_i = \sum P_{ij}^{-1} \vec{P}_j$$

so that for any $\vec{F}$ in S_B

$$f_i = \vec{P}_i * \vec{F} \quad \Longleftrightarrow \quad \vec{F} = \sum f_i \, \vec{p}_i. \tag{4}$$

With this machinery in place, the functions $\vec{P}_i$ in the MeasurementBlock define a projection P,

$$P : S_I \longrightarrow S_B \qquad \ni \qquad P[\vec{S}] = \sum \left(\vec{P}_i * \vec{S}\right) \vec{p}_i. \tag{5}$$

With only a slight abuse of notation, we may identify the MeasurementEquation of the MeasurementBlock with the projection operator P. We will use the term MeasurementEquation when we wish to emphasize the physical significance of the operator P and the term "projection" to emphasize the purely mathematical significance of P. The function $P[\vec{S}]$ will be referred to as the projected intensity.

A useful distinction should be made at this point between the equations on the left and right sides of the arrow in Equation (4). The left-hand equation describes how to derive measurements f_i from a theoretically known function $\vec{F}$. We use this when we wish to compare a theoretical model of the sky to an existing set of data. The right-hand equation describes how to reconstruct a function $\vec{F}$ from the measurements f_i and the basis vectors $\vec{p_i}$, but is agnostic about the origin of the measurements. The projection operator P uses the left-hand equation to sample a theoretical model sky, then reconstructs the projected intensity with the right-hand equation. For comparison, astronomical observations use a real telescope to sample the physical sky, yielding the raw samples s_i. One or more TelescopeModels convert the raw samples into calibrated samples f_i, and define the functions $\vec{p_i}$. If the original basis set $\vec{p_i}$ is not convenient, one or more MeasurementModels may be used to convert the measurements to a more appropriate basis. This process is sufficiently different from the construction of a projected intensity that it deserves a different name; the term "measured intensity" seems appropriate. Data reduction involves the construction of a **measured intensity** from the raw measurements. Data analysis involves comparison of the **projected intensity** derived from a **model sky** with the **measured intensity** from the **physical sky**. Figure 1 presents the normal infomation flow as a Rumbaugh functional diagram, with the physical sky seen through a real telescope and a model sky being the two actors, and the data store of calibrated measurements containing the representations within AIPS++ of the measured and projected intensity functions.

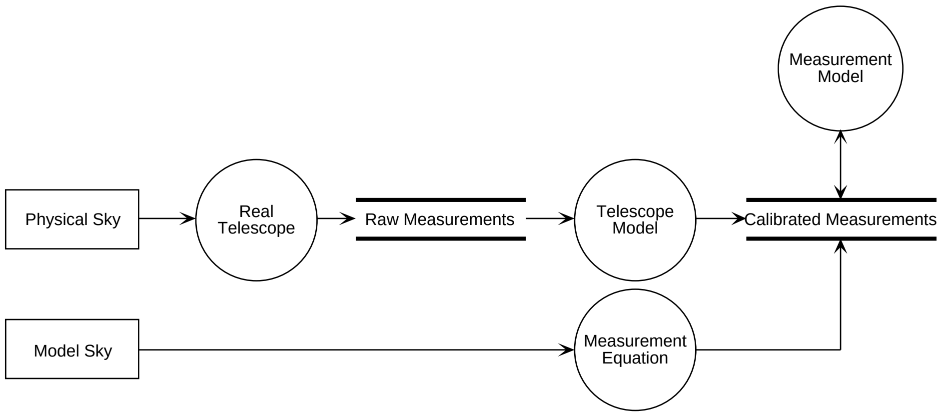

Figure 1 — The information flow from the physical or model sky into a store of calibrated measurements.

In the remainder of this section we will examine the utility of the projected intensity $P[\vec{S}]$, and of the projection P itself, in the data reduction process. In general, we will find that the projected intensity constructed directly from a telescope model often does a poor job of representing the sky intensity. The solution is to choose a set of basis functions which *will* represent the sky intensity well, then to model the measurement process by applying the projection P to these basis functions before fitting them to the data.

The most important (and obvious) property of the projected intensity $P[\vec{S}]$ is that the projection has discarded (set to zero) all the aspects of $\vec{S}$ which cannot be represented in S_B. Formally, if $\vec{S} = \sum S_i \vec{P_i} + \vec{S}_\perp$ where $S_\perp * \vec{P_i} = 0$ for all i, then $P[\vec{S}] = \sum S_i \vec{P_i} + 0 \cdot \vec{S}_\perp$. This offers a clue which can help us to recognise the projected intensity in more familiar objects.

I believe that interferometrists are already familiar with one example of the projected intensity in the form of a dirty map. For an interferometer the spatial part of the PSF $\vec{P}_i$ for each visibility sample is the product of the beam shapes of the two antennae times a cosine term due to the antenna separation. To the extent that the samples are well separated, the cosine terms will make the $\vec{P}_i$ from different samples nearly orthogonal so that the sum in Equation (5) becomes simply a Fourier transform of the visibilities, with the unmeasured components set to zero. The many flaws inherent in dirty maps are well known in interferometry and many complex procedures have been devised to find improved images. We may expect similar problems to arise for single-dish data.

Unlike interferometrists, spectroscopists will usually **not** be familiar with the notion of the projected intensity. This is at least partly because of the sloppy way in which we treat simple data structures such as spectra. Astronomers usually plot spectra either as histograms or a points joined by straight lines. No account at all is taken of the sensitivity profile of each channel. This simple-minded approach is normally quite acceptable as long as the channel width is small compared to the line width of the spectral features. For narrow features, however, it can provide a misleading representation of the data. In the past, unwary programmers (yes, even myself!) have often ignored the sensitivity profiles, and sabotaged important aspects of the data by writing inappropriate "shift-and-add" routines into their data reduction programs. In a later section we will discuss how to provide both histogram and dot-to-dot representations of the data correctly, with their virtues and drawbacks plainly visible.

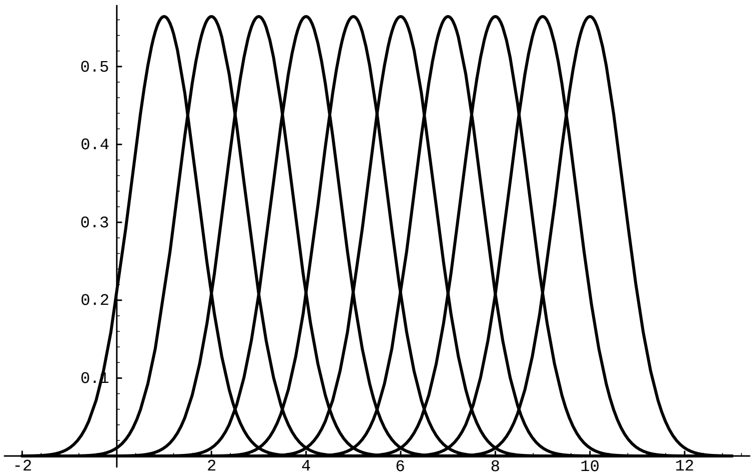

Figure 2 — The point spread functions $P_i(x) = e^{-(x-i)^2}$ for $i = 1 \ldots 10$

Although spectra are intrinsically simpler data structures than interferometer visibilities, a naive use of the projected intensity derived from the spectrometer can result in a spectrum just as ugly as any dirty map. Let us consider the simple case of a multi-channel spectrometer, with a bank of identical channels evenly spaced in frequency. For simplicity, let us restrict the radiation coordinates to a single real dimension x, and the intensity to a real valued function $I(x)$. Suppose we have a 10-channel instrument whose point spread functions are a set of Gaussians with unit standard deviation centered on the integers 1 through 10, as shown in Figure 2. (Gaussians were chosen for the PSF's simply because they are easy to integrate.)

It will be noticed that these functions are not orthogonal. As a consequence, the corresponding basis vectors $p_i(x)$ look distinctly different from the original $P_i(x)$, as seen in Figure (3).

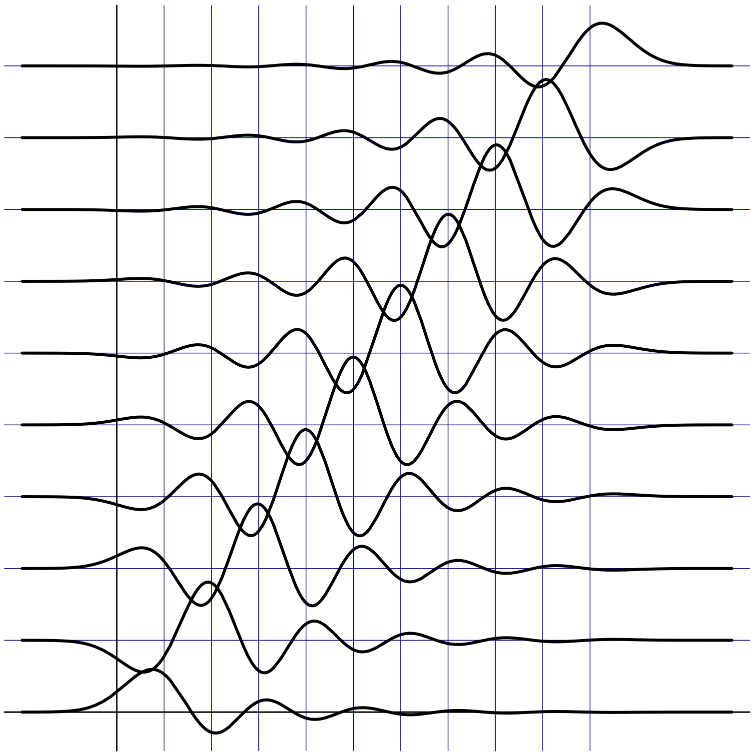

Figure 3 — The basis functions $p_i(x)$ corresponding to the $P_i(x)$ in Figure (2)

These basis functions have large sidelobes, extending across the entire spectrum. They do not necessarily peak at the nominal center of the channel, and are not usually zero at the centers of the other channels. These are all warning signs that this basis will not provide the most intuitive representation of the data.

As a simple test to see how well this basis can represent a commonly encountered intensity, we can apply the projection operator to the constant function $I(x) = 1$ whose measured values will just be the 10-tuple $\{1, 1, 1, 1, 1, 1, 1, 1, 1, 1\}$; the projected function $P[1]$ is shown in Figure (4).

$P[1]$ does not look very constant across the band, and few of us would accept it as a suitable representation of the data. There are large edge effects. Evaluating the function at the channel centers does not return anything like a constant function, and in fact Figure (4) shows that the channel centers are nearly the worst possible places to evaluate the function. It is possible to find other sums of the same 10 Gaussians which appear much flatter to the eye than $P[1]$, but they do not return a 10-tuple of 1's when sampled with the $[P_i(x)]$. In spite of its appearance, $P[1]$ is the unique function in S_P which returns exactly $\{1, 1, 1, 1, 1, 1, 1, 1, 1, 1\}$.

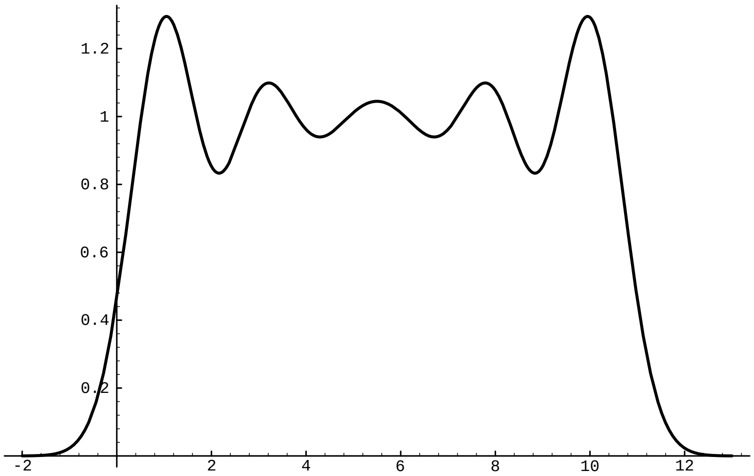

Figure 4 — The projection $P[1]$ onto the subspace S_C

It is clear that the projected intensity defined by an instrument is often not useful as a functional representation of the data. We may summarize the nature of the problem by recalling that for a MeasurementBlock B, *the projection P_B is optimally matched to the measurement process,* **not** *to the sky.* When we need a subspace of S_I to model the expected sky intensity, we should choose that subspace, call it S_P, and its associated projection P, openly and deliberately. This choice should be guided by our understanding of the physical origin of the intensity. Since the MeasurementBlock is related to the physical sky intensity $\vec{S}$ by $[s_i] = [\vec{P}_{B,i} * \vec{S}]$, we should compare any model sky intensity $\vec{I}$ in S_P to the measured intensity using the same measurement process $P_B[\vec{I}]$.

To illustrate this for a simple case, let us return to our 10 channel spectrometer. Suppose we know that $P[1]$ is an observation of a slowly varying source so that a suitable representation of the data might be a quadratic polynomial. The monomials 1, x, and x^2 project onto 1, x, and $1/2 + x^2$ respectively. Fitting these to the data, we would conclude correctly that the original sky intensity was well represented by the constant 1. We note that simply fitting a quadratic to the data would risk corrupting the final solution by mixing the constant and quadratic terms.

Imaging Models

Transformations from one representation of the data to another have been discussed by T. Cornwell (DRAFT: Recommendations for the AIPS++ Telescope Model, October 29, 1992) and by R. M. Hjellming (Some Thoughts on Telescope Data Handling in AIPS++, November 12, 1992) in the context of an imaging model. Although their notation is now officially obsolete, it is important to see how the current concepts mesh with their discussion. In their work the MeasurementEquation for a set of raw measurements is decomposed into two operations

$$A_{TM} A_{IM} I = Y \tag{6}$$

where I represents the desired image, A_{TM} is a telescope model which we hope will remove the corrupting effects of the telescope and instrumentation, A_{IM} is a MeasurementModel, referred to in their work as an "imaging model", which transforms an image of the desired form into the form of the calibrated data, and Y is the raw data. Since they were primarily concerned with the A_{TM} and A_{IM} operators, they did not concern themselves deeply with the nature of the image I, which was mostly treated as though it was just an array of numbers. However, each element in the array I has associated with it a start and stop time, a center frequency and frequency width, a beam center and a beam shape: in short, all of the properties of a measurement equation like Equation (2). Within the framework of the present document, the sky intensity $\vec{S}$ will be related to the image I by some projection $I = P[\vec{S}]$ (which may involve delta functions, all-time averages and other generalized functions) so that Equation (6) now reads

$$A_{TM} A_{IM} P[\vec{S}] = Y. \tag{7}$$

Note that the projection P need not be derived from a telescope model, but may be chosen quite arbitrarily to suit the nature of the astronomical source. Note also that moving from Equation (6) to Equation (7) entails a fundamental shift in the *mathematical* interpretation of the image I, which becomes a function in the subspace S_P of S_I. The purpose of the MeasurementModel A_{IM} is to simulate a measurement of this image for comparison with the calibrated data.

As we discussed in the first sections, each TelescopeModel has associated with it a natural representation of the data and an associated MeasurementEquation T. The MeasurementModel A_{IM} for the natural representation is trivial, so that

$$A_{TM} T[\vec{S}] = Y.$$

If we choose to represent the data in some other basis whose measurement model is P, the projection from S_P onto S_T which models the measurement process defines the non-trivial MeasurementModel

$$A_{IM} = \left[\vec{T}_i * \vec{P}_j \right]. \tag{8}$$

The documents by Cornwell and Hjellming give numerous examples of these imaging models without explicitly considering how they might be related to the projections T and P. The explicit representation of A_{IM} given in Equation (8) clarifies the mathematical content of their models, and allows them to be generalized consistently and unambiguously.

The operators A_{TM} and A_{IM} are presented using a matrix-like notation, but they can represent any invertible transformation, linear or nonlinear. Many important operations appear as linear transforms on the data with nonlinear internal parameters (the telescope model A_{TM} often has this form). It is not difficult to make the connection between linear and nonlinear operators (although nonlinear operations can be very difficult to impliment!). Consider a (possibly nonlinear) representation $\vec{R}(r; m) \in S_I$ where m is an array of parameters. This representation defines a manifold M in S_I whose coordinates are defined by the parameters m. For a linear representation, the manifold is a subspace and the coordinates m are provided by the coefficients of the basis vectors. Provided we take sufficient care about the invertibility of the operations, and do not allow ourselves to become trapped by existence and uniqueness problems, the discussions in the documents of Cornwell and Hjellming, and in this one as well, remain valid if one simply makes the substitutions

subspace .... manifold
basis .... coordinate patch
coefficient of a basis vector .... coordinate
change of basis .... coordinate transformation.

Bearing this in mind, we will continue to use a linear algebraic approach in this discussion.

How often will we need to consider a nontrivial MeasurementModel? The most obvious examples occur in interferometery, where the MeasurementModel sits at the core of the entire data reduction process. MeasurementModels, however, occur in a large number of circumstances, in spite of the fact that previous single dish data reduction programs have mostly ignored the entire concept.

The most glaring, although somewhat artificial, problem to be dealt with in an MeasurementModel is that the support in time of the projection associated with a MeasurementBlock is determined by the start and stop times of the integrations within the block. Different MeasurementBlocks will only rarely overlap in time. The natural basis of a set of MeasurementBlocks thus allows explicitly for the possibility of block-to-block changes in the intensity of a source, even if we expect the real timescale for variation to be $> 10^6$ years. In the past this problem has been circumvented by politely ignoring the start and stop times of the component integrations while averaging the data. In the current context, we would prefer to choose a representation of the sky intensity which was explicitly time-invariant over the period of the observing run and to construct (very simple) MeasurementModels to relate the time-invariant representation to the time-limited samples in each Measurement. This is mostly a matter of keeping the language consistent, and would only be reflected in code if we explicitly chose a time-variable representation.

It will usually be necessary to construct a new MeasurementModel for each MeasurementBlock in the MeasurementSet because the details of how the instrumentation sampled the sky intensity would have changed with time. Consider, for example, a spectrometer whose channel spacing is smaller than the diurnal change in radial velocity for an equatorial source. The Doppler correction due to the rotation of the earth is often neglected during data collection because it is small and because regular changes to the LO can detune a receiver. During data reduction, as a consequence, the spectrometer channels will not line up. Adding spectra together conventionally requires that the the spectra be brought into alignment, usually by some completely arbitrary interpolation method, and a decision must be made whether to keep or discard the end channels which do not overlap. The final representation of the spectrum thus depends upon the arbitrary choice of the first spectrum in the average, when the feature of interest may not even be visible above the noise. Narrow features are quite normally smeared across several channels by the interpolation algorithms. In the terms being developed here, the two MeasurementBlocks being added have different MeasurementEquations. Because the spectra have been treated as simple arrays of numbers, ignoring significant information in the MeasurementEquations, the data has been corrupted. A better data reduction procedure would first choose a representation for the data which includes enough basis functions to represent the data over the required frequency interval, explicitly defines the time-dependance and includes or ignores the end channels. Each MeasurementBlock would be related to the chosen representation by its own MeasurementModel. The final spectrum would be found by applying a suitable generalized inverse to the simultaneous MeasurementModels for all of the spectra included in the average.

More dramatic examples include frequency-switched spectra or beam-switched images, where the PSF of the measurement may be represented as the difference between two shifted copies of the fundamental PSF of the telescope. In these cases we explicitly do **not** want to use the natural bases of the MeasurementBlocks, requiring instead an estimate of the sky intensity with the effects of switching removed. Methods to accomodate frequency-switched data are often added as after-thoughts to spectral line data reduction programs. The same methods are often used (and may be better implimented) in continuum mapping programs. Both fit naturally into the formalism being developed for AIPS++.

The ultimate change of representation which can be implimented in a data reduction program is a generic "user specified" representation, with a finite but unspecified number of internal parameters whose meaning is defined only by its relationship to the calibrated data. To handle this kind of model may be outside the limits of practical programming, but would surely be a worthwhile goal. In the forward direction the problem is simple to state: for a representation whose associated projection is P, and a user model $\vec{M}(m) \in S_I$ with fixed parameters m, we must be able to evaluate the projected intensity $P[\vec{M}(m)]$, i.e. we must provide a means to evaluate the integrals $\int \vec{P}_i(r) \cdot \vec{M}(r; m)\, dr$. These integrals might be evaluated either numerically or symbolically. This would allow us to convert the external representation $\vec{M}$ into an internal representation which the program will be able to manipulate and display. The backwards problem would require the provision of some kind of generalized handles by which the program could manipulate the parameters m of the external representation, perhaps with some very general optimization algorithms to fit the model to the data.

Combining Data from Different MeasurementBlocks

The previous section skipped rather lightly over the problem of combining data from different MeasurementBlocks. In this section we will examine a semi-practical approach to this central problem of data **reduction**. In the process we will also handle the problem of MeasurementBlocks whose natural bases are not linearly independent.

The key is to choose a representation (defined by its measurement model P) for the sky intensity whose basis is known to be linearly independent. This is normally very easy. Any kind of image which covers the domain in radiation coordinates accessed by the measurements with at least the required resolution, and whose pixels do not completely overlap, will usually do the job. In this representation, let us denote the averaged intensity model as $I \in S_P$. For a MeasurementBlock b, we denote the data as y_b, the associated MeasurementEquation as P_b and the corresponding subspace in S_I as S_b. For each block b to be included in the average we can then contruct a MeasurementModel $A_b = \left[\vec{P}_{b,i} * \vec{P}_j\right]$. Within the subspace¡ S_b we can employ a penalty function N_b. The solution I can then be found by minimizing

$$N(I) = \sum N_b\big(y_b - P_b[I]\big). \tag{9}$$

Although this expression may not seem familiar, if the penalty functions N_b are just the l_2 norms weighted by the integration times, minimizing the RHS of Equation (9) becomes just a normal, weighted least squares, and if the channels align in all of the representations (i.e. $P \equiv P_b, \forall b$), we will get in each channel of I a weighted average of the measured signals in the corresponding channels of the contributing MeasurementBlocks. The extreme generality of Equation (9) encompasses more complex data reduction methods such as maximum likelihood or maximum entropy, which are commonly used in astronomy.

Minimization methods often become unwieldy and slow, but the form of Equation (9) is only a common language to frame the problem and does not commit us to any particular method of solution. Let us return to the task of aligning spectra to see how practical the problem is when approached from this direction. In the notation of the last paragraph, we will use the l_2 norm weighted by the integration time so that

$$\sum_b N_b\left[y_b - P_b[I]\right] = \sum_b \sum_i \Delta t_b \left(y_{b,i} - \sum_j \vec{P}_{b,i} * \vec{P}_j \; I_j \right)^2 .$$

Minimizing this for I_k gives the normal equations

$$\sum_j \left(\sum_b \sum_i \Delta t_b \; (\vec{P}_{b,i} * \vec{P}_k) \cdot (\vec{P}_{b,i} * \vec{P}_j) \right) \; I_j = \sum_b \sum_i \Delta t_b \; y_{b,i} \; \vec{P}_{b,i} * \vec{P}_k. \qquad (10)$$

There are, of course, better ways to solve least squares problems than the normal equations, but they will serve for the purpose of illustration. For most spectrometers the PSF's of each channel will be quite narrow, effectively overlapping with only one or two of its nearest neighbours on either side. Without much loss in accuracy, the matrix $[P_{b,ij}] = \left[\vec{P}_{b,i} * \vec{P}_j\right]$ may be approximated by a narrow band matrix and the normal equations will similarly have a band structure which may be constructed and solved without imposing difficult storage problems. The most time-consuming parts of the calculation are likely to be the products $\vec{P}_{b,i} * \vec{P}_j$, but even these can be made tractable with a little effort on the part of the software engineers in charge of the instrument. For most spectrometers these products will depend only upon the choice of representation P and the seperation $\nu_{b,i} - \nu_j$ between the centers of the channels. If this can be approximated to a tolerable accuracy by a rational function, we will still be able to compute the average spectrum efficiently.

Choosing a Good Basis

It has been emphasised several times that the astronomer should carefully and deliberately choose a representation for the sky intensity, i.e. a set of basis functions $\{\vec{P}_i\}$ which will be used to represent the intensity during the period of observations, $\vec{I} = \sum I_i \vec{P}_i$. What properties do we expect of a good basis set? These will, of course, depend upon the physical system under study, but it is easy to list several properties which are generally useful.

1) Orthogonality — if $\vec{P}_i * \vec{P}_j = \delta_{ij}$ many of the equations in this document are greatly simplified. Spectra are often plotted as histograms; letting $\{\nu_i\}$ be the set of channel center frequencies, the basis set for a histogram is a set of box functions

$$p_i(\nu) = \begin{cases} 1 & (\nu_{i-1} + \nu_i)/2 < \nu < (\nu_i + \nu_{i+1})/2 \\ 0 & \text{otherwise.} \end{cases}$$

Since these functions are orthogonal, the corresponding measurement equation is simply

$$P_i(\nu) = \begin{cases} (\nu_{i+1} - \nu_{i-1})/2 & (\nu_{i-1} + \nu_i)/2 < \nu < (\nu_i + \nu_{i+1})/2 \\ 0 & \text{otherwise.} \end{cases}$$

2) Local Support — If the basis is not orthogonal, it may still be possible to arrange that the support for each basis function overlaps the supports of only a small number of neighbouring functions. For example, another popular display for spectra plots the power in each channel as a point at the center frequency ν_i and joins the points with straight lines. The basis functions for this representation are the hat functions

$$h_i(\nu) = \begin{cases} \frac{\nu-\nu_{i-1}}{\nu_i-\nu_{i-1}} & \nu_{i-1} < \nu < \nu_i, \\ \frac{\nu_{i+1}-\nu}{\nu_{i+1}-\nu_i} & \nu_i < \nu < \nu_{i+1}, \\ 0 & \text{otherwise.} \end{cases}$$

These functions are not orthogonal, but the matrix $[p_{ij}] = [\vec{p}_i * \vec{p}_j]$ is tridigonal (and, of course, symmetric), with

$$\begin{aligned} p_{i\,i} =& (\nu_{i+1} - \nu_i)^2/3 + (\nu_i - \nu_{i-1})^2/3 \\ p_{i\,i-1} =& (\nu_i - \nu_{i-1})^2/6. \end{aligned}$$

For a simple system like this it is easy to calculate the projection functions on the fly. A typical example is shown in Figure 5.

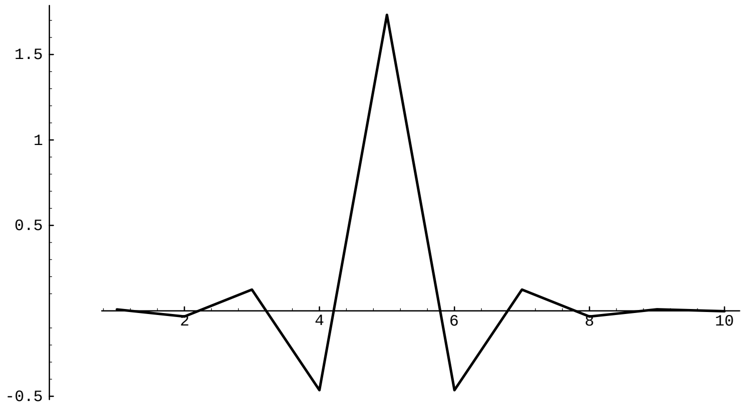

Figure 5 — The projection $H_5(x)$ corresponding to the hat function $h_5(x)$ with ten channels centered at $1 \ldots 10$.

It will be noted that if the basis has local support, the projection usually does not, and vice versa.

3) Finite Bandwidth — If our instrumentation, which always has a finite bandwidth, has the same PSF $\vec{P}$ for each channel but with a different center position, i.e. $\vec{P}_i = \vec{P}(r - r_i)$, we can view each MeasurementBlock as a discrete sample of the convolution of the sky $\vec{S}$ with the $\vec{P}$, $s_i = SP(r_i) = \int \vec{P}(r - r_i) \cdot \vec{S}(r)\, dr$. Since $\vec{P}$ has a finite bandwidth, the convolved sky intensity SP will as well, and it would be desireable if our final representation of the data was similarly band-limited. Indeed, reconstructing the "perfect sky" SP for a given telescope is a common way to express the immediate purpose of data calibration. Neither the box function basis nor the hat function basis preserves the band limitation of the original data, so more complicated bases might be needed if band-limitation is an important property of the image.

4) Channel-Center Interpolation — If we can arrange that $p_i(r_j) = \delta_{ij}$, then the projected intensity $p(r) = \sum s_i p_i(r)$ will interpolate between the points (s_i, r_i). This

makes the projected intensity much easier to understand. Both the box function basis and the hat function basis have this property, but the natural basis derived from a telescope model often does not (recall Figure 4). Note that Local Support is not the same as Channel Center Interpolation; the functions $\frac{\sin(2\pi x)}{2\pi x}$ which appear in the projection operator for an autocorrelator have Orthogonality, Finite Bandwidth and Channel Center Interpolation, but not Local Support.

5) Analytic Simplicity — The MeasurementModel relating an image whose MeasurementEquation is P to a MeasurementBlock taken with a telescopic MeasurementEquation T requires that we calculate the products $\vec{T_i} * \vec{P_j}$. Since we do not have any choice in the form of T, we would be well advised to choose P so that these products may be computed quickly. Simplicity here is almost surely a key to success.

6) Independent Noise — Any image derived from a measurement will include some noise, and it is usually desireable if the noise is distributed uniformly and independently among the channels of the final image. This will often happen naturally if the number of channels in the final image is similar to the number of channels in each of the MeasurementBlocks used to produce it. Often, however, it is useful to include far more channels in the final image than are justified by the data. In this case some non-linear process, such as maximum entropy or CLEAN, must be used to determine a unique answer and the noise properties of non-linear transformations can be quite unpredictable. In a similar vein, smoothing data can introduce strong correlations in the noise in adjacent channels. This can make it very difficult to determine when a weak feature is real and when it is just correlated noise. The noise in the final image is a property of both the MeasurementModel and of the algorithm used to solve the MeasurementModel; both should be chosen carefully if misleading answers are to be avoided.

7) Consistency — Corruption of the data, such as is caused by "shift-and-add" routines to align spectra, may be avoided by consistent use of the MeasurementEquations and MeasurementModels throughout the system. This principle can be extended to display as well. If the astronomer wishes to plot spectra as histograms, it would be desireable to reduce the spectra using the box function basis discussed above. The histograms in the display would then consistently reproduce the sky intensity computed in the program. This kind of consistency is logically desireable, but not strictly necessary for hardcopy outputs, which are only rarely re-entered into another program, but can be quite crucial for images which are partial results, to be retained for additional processing later.

Implications for AIPS++

The final output from the data reduction process, be it an image, a spectrum, a data cube, or whatever, is just another form of calibrated data, i.e. a MeasurementBlock. For exactly the same reasons that a Measurement should be viewed as the pair $(s, \vec{P})$, the finished data should be viewed as a pair (I, P) where I is an array containing the data, and P is the MeasurementEquation (projection operator) which defines the physical meaning of the data in I. In hardcopy output we normally display the data in the body of a figure and the physically important parts of the MeasurementEquation in the axes, labels, and captions.

The data structures and their relationships which we derive for AIPS++ are shown in Figure 6. This figure summarizes the conclusions of the design team at the Charlottesville meeting of November 16-18, 1992.

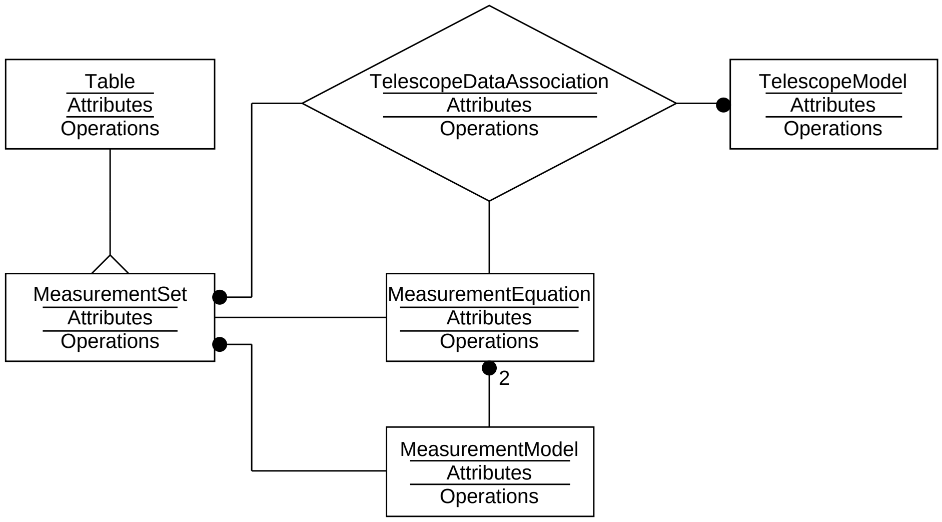

Figure 6 — The Object Model Diagram for the classes related to Measurements.

There was some discussion about whether a MeasurementSet "isa" Table containing data with extra properties or "hasa" Table containing the data. Although there was a strong feeling that a MeasurementSet "hasa" Table, this author prefers the "isa" relation. The Table in a MeasurementSet **must** include **all** of the physical parameters (possibly indirectly through other Tables) needed to construct the MeasurementEquations. Also, although MeasurementEquations figure very prominently in the analysis of data from intensitometers, they are much less useful in the other great class of instruments which record "events" in the radiation and instrumental coordinates, such as the arrival of an individual photon at a time t and a position (x, y) in a detector, or the position x_λ on a photographic plate of a spectral line of wavelength λ. This kind of data will often use the internal operations of the Table class directly, without much of the extra analysis needed to work with MeasurementEquations. A MeasurementSet will therefore need access to all of the operations which apply to Tables, and hiding the Table as a class member will add complexity to the code without improving the security of the operations or the clarity of the model.

Note that a TelescopeDataAssociation has associated with it a MeasurementEquation class which defines the natural basis for the MeasurementBlocks within the MeasurementSet. It is **not** necessary for a MeasurementEquation to have an associated TelescopeDataAssociation. Unfortunately, it is not possible to represent this in an ObjectMaker diagram, since the set of connectors allowed for an "association as class" includes a one-to-optional, but not an optional-to-one. Similarly a MeasurementSet does **not** require a TelescopeDataAssociation unless the MeasurementSet includes raw data to be calibrated. ObjectMaker does not provide an optional-to-many connector for "association as class" either.

Although the MeasurementEquation is officially a function of all of the radiation coordinates, it is most often true that the PSF factors into a product of terms for each independent coordinate. For these cases, it would be useful to provide subclasses of MeasurementEquation which provide the factored forms, with seperate member functions for each radiation coordinate.

A MeasurementModel is **defined** by its two associated MeasurementEquations. This association must dynamic, with the MeasurementModel constructor building the matrix relating the two MeasurementEquations on the fly. Initially this will probably have to be a restricted service, handling only a limitted selection of MeasurementEquations with simple offsets in frequency and space. Later, it may be useful to include formulae in the MeasurementEquations which could be passed to through a Parser and an Integrator to generate these formulae in real time for arbitrary MeasurementEquations. This would be extremely challenging, however, and may not be important for the first version of AIPS++.