Design Considerations for a New Single-dish Data Analysis System

Harvey Liszt
Jay Lockman
Dan Stinebring

14 November 1984

In this document we discuss how to fulfill NRAO's need for a new single-dish data analysis system. The purpose of this program is to manage one-dimensional arrays, such as spectra or time series (and their associated headers) that are the principle data from single-dish telescopes. The program should contain features for the editing, reduction, correction, or modification of the data, should contain analysis and display features, and should produce a variety of machine-readable output, ranging from tables to n-dimensional cubes.

Ground Rules

a) One system for the entire observatory.
b) Well maintained.
c) Documented internally and externally.
d) Portable.
e) Output in a form easily accesible to AIPS.
f) Users are protected from each other.
g) Mathematically correct.

Central Features

Lotus 1-2-3 type features:

a) Screen oriented interaction.
b) Data-base managment facilities.
c) Spread-sheet capabilities (on data and header).

d) Flexible and varied displays.
e) Symbolic manipulation of data.
f) History files.
g) Various levels of prompting.
h) Batch and interactive modes.

i) Built-in Utilities.
j) Easy addition of user-written subroutines.

Details

a) The command language must be simple and correspond as far as possible to ordinary mathematical usage. It should not require that the user keep track of details like the location of disk data sets, the state of internal pointers, etc.
b) Direct, easy access to every bit of data. e.g. easy to change the date, time, position or value of the 47th point.
c) Access to data by all qualifiers and logical combinations of qualifiers stored in the header information (not just by scan number as in the present systems).
d) Can create new arrays from functions acting on other arrays. e.g. create T(v) vs. declination, or create a table of Gaussian-fit components vs. date.
e) Access to user-defined auxillary arrays and operations, including user-written subroutines that are linked into the system.
f) Functions smart enough to permit algebraic manipulation and redefinition of arrays through simple statements of commonly used operations such as +, -, *, /, etc.
g) Works with any $y(x_i)$ even when the x_i are not uniformly spaced or not one-dimensional. e.g. $T(ra_i, dec_i)$ where $0 < i < 4096+$.
h) Has interpolation and merge features. e.g. can add spectra with different Δv, center velocity, and total number of channels or merge several 256 channel spectra to form a 1024 channel spectrum.
i) Contains utilities (precession, coordinate conversion) which can be used while in the midst of data manipulation.

Discussion

The four general steps in single-dish data reduction are (1) access of the data, (2) application of functions to it, (3) display, and (4) output of a finished product. The current NRAO software is most seriously deficient in the first and fourth areas, largely because data sets have grown beyond what the TSPOWER/CONDARE programs can handle gracefully.

(1) Access The first step in data reduction is to access the data -- there must be a unique way to specify the array or arrays that we are interested in. In the current programs this means specifying a scan number and, possibly, selecting a "receiver" from others with the same scan number. This is often not the most logical or convenient way to access data, and it is responsible for many of our problems. For example, we may really want to average all spectra on "Orion A" from a large set of data that were acquired over days, or years. We currently, however, are forced to average scans 1234, 1236, 2439, 8762 . . . We should be able to access data by simply specifying any item in the header, or logical (Boolean) combinations of arithmetic operations on header items. In contrast to maps, one-dimensional data sets can have a large ratio of header information to data, and it is common for single-dish data sets to contain a thousand independent 1-d arrays. Editing the data set to remove bad data points depends heavily on the ease of access and the ease of data flagging. A lot of attention should be paid to interactive data editing.

(2) Functions We have received many suggestions for new functions. Most could be included within the current programs. More important is the clumsiness of many of the current functions. To add two spectra, for example, we currently must move the first into register A and the second into register B; "ADD" then sums them and places the result into register C. In TPOWER/SPOWER the registers A, B, and C are chosen from three arrays, "Work", "Temp" and "Hold"; in the Arecibo ANALYZ program the registers are in a stack. There is no reason why the user should be this involved in the internal bookkeeping of a program. The "Average" function should look like T = < all 'Orion A' fsky=5.303 except scans 243-321>. The resulting average should have its own header and history and be stored, retreivable, and funtionally indistinguishable from data direct from the telescope.

(3) Display We see many needs for improvement in the area of display. As an example, the display of multiple spectra, either on a spatial grid or in a data cube should be possible. High quality contour plotting and/or gray scale plots are essential. A user should be able to annotate a plot freely, as well as place symbols such as arrows and channel flags in arbitrary places on the plot. Multiple window graphics should be seriously considered, allowing the user to display and manipulate more than one plot at once. Many of the insights in data

analysis come from flexible and powerful graphics displays. A large part of the design effort of this system should go into providing interactive graphics of the highest quality.

(4) Output Right now, there are only two outputs for processed data: a "Keep" tape, or hard copies of spectra or of the green-screen contents. Note that both of these remove the finished product from the system. We need instead a way to keep reduced data within the system, as suggested above in the data averaging example, complete with its history. Sometimes reduced data do not look like a spectrum -- there may be a table of Gaussian components, a moment map, a single number. These should also be accessible to the functions of a program. For example, given an output of Gaussian components we may wish to plot the central velocity vs. position, or the peak intensity vs. time. All this is just to say that there many stages of "output", most intermediate, and many in need of further processing. The "output" problem feeds directly into the "access" problem.

How do we get a new system?

1) Steal it.
2) Buy it. There are several commercial or industry (Bell Labs) systems that do many of the things we want. Commercial software can be very cheap compared to that written in house (what would it cost us to write Lotus?). But we probably could not distribute the code to other users, and there may even be limiatations on changes we could make to it.
3) Modify or extend existing programs. This means either AIPS, TPOWER/CONDARE, ANALYZ, or some of the university-developed programs like the UMass/CIT package or the Texas system. The modifications are major.
4) Write a new system from scratch. This could be big bucks

It is absolutely essential that the right answer to this question be found. We do not know the answer, but we believe that the success of this project hinges on how this question is pursued and what conclusion is arrived at. We strongly recommend discussions with the AIPS group on this question. Not only have they created astronomy's only portable (and maintainable) data analysis system, but they have developed a level and style of documentation (from the Cookbook to the online aids) that is far ahead of other systems we have seen. It would be attractive to recommend AIPS outright as a base for the new system, except that we are concerned about a) the suitability of the user interface, b) the efficiency of the catalog structure for lots of small data items, and c) the difficulty of adding user-written subroutines. We have looked at IRAF and are not enthusiastic about it, mainly because of the arcane command language.

Summary

A new NRAO single-dish analysis package is needed. Work on it should be started now. Within the next few years we will see multi-beam spectral line and continuum receivers coming on line, a new 300' back end, and a phased-array processor at the VLA. This wave of data will paralyze the current programs. We are already operating under a patchwork of programs, so the time is especially ripe for a new system.

We recommend that a single-dish software design group be formed immediately and that a project manager be appointed. The project manager should be able to spend the majority of his or her time on this project over the next two to three years. The design group should include at least one AIPS member and single-dish oriented members of the scientific staff. This project is as important to the continuing health of NRAO single-dish work as AIPS is to aperture-synthesis work; manpower and management support should reflect this importance.

The NRAO has developed and distributed an outstanding synthesis reduction package. The new single-dish package should meet the high standards set by AIPS, but reflect the advances in interactive data analysis made since AIPS was designed.

Appendix

The following example is only a guide to what we have in mind. The formats, commands and output are negotiable. Only the ability to do the processing is important.

Editing and averaging a set of spectra. We start with the data loaded into the system. User's entries are upper case, machine response is underlined. Our comments are in parentheses. For clarity we use a machine prompt '>'. This carries no implications whatsoever.

```
>CATALOG 'ORION A' = 'ORION A'        (user makes a catalog called
                                       `orion a' that contains all the data
                                       labeled `Orion A')
```

catalog 'orion a' created. 232 'orion a' entries with defaults.

```
>LIST CAT 'ORION A'
seq  scan   ra(1950)     dec(1950)    freq     tsys   rcvr    date
1    3412   18 30 18.4   -28 29 66    1.2936   12.4   1       1/1/84
2    3412   18 31 18.4   -28 29 66    1.2936   12.3   2       1/1/84
3    3413   18 32 18.4   -28 39 64    1.2936   12.4   1       1/1/84
.    .      .            .            .        .      .       .
.    .      .            .            .        .      .       .
.    .      .            .            .        .      .       .
 (and so on)

>COLUMN 3 = 'ELEVATION'
>COLUMN 4 = 'LST'
>COLUMN 5 = 'INTEGRATION TIME'
>RECATALOG 'ORION  A'                 (I want different items listed
                                         in the catalog. 'Orion A' is
                                         still the sorting item)
```

catalog 'orion a' has been recreated. it contains 232 items.

>LIST CAT

seq	scan	elev(deg)	LST	tint(s)	tsys	rcvr	date
1	3412	1.0275	14:00:00.2	20.0	12.4	1	1/1/84
2	3412	1.0275	14:00:00.2	20.0	12.3	2	1/1/84
3	3413	1.1067	14:00:21.2	20.0	12.4	1	1/1/84.
.	.	.	.	.	.	.	.
:	.	.	.	.	.	.	.
:	.	.	.	.	.	.	.

(and so on)

>GRAPH TSYS VS ELEV
(up comes a terminal display of the system temp. vs elevation)

>UNCATALOG CROSSHAIR
(a crosshair is displayed on the screen along with instructions. When you press the "space" bar, for example, the data set belonging to the point where the crosshair is located is removed from the catalog. When you are done you hit "escape" and the machine responds):

18 entries have been removed from catalog 'orion a'

>TELL WEIGHT (what is the current weighting function for averaging)

weight=tint/(tsys**2)

>WEIGHT=WEIGHT/(ELEV**2) (change the weighting function)

weight=tint/(tsys**2)/(elev**2)

>T=< CATALOG 'ORION A'> (the array T now contains the weighted average of the spectra in the `orion a' catalog)