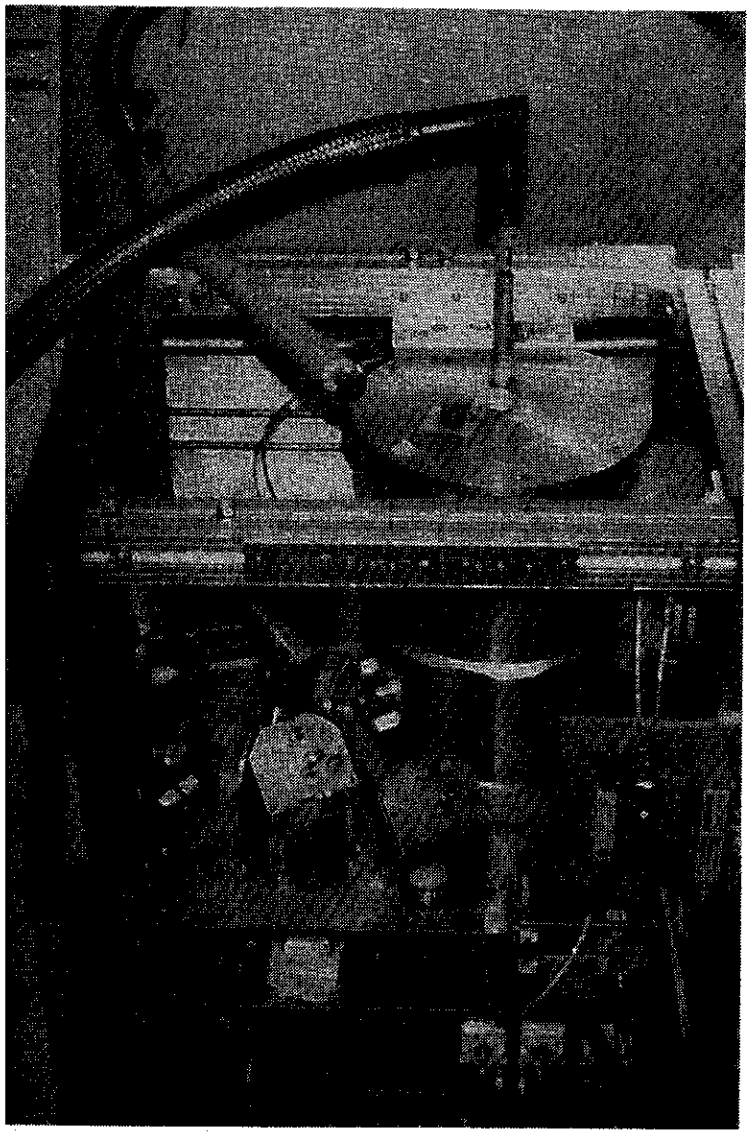
PROTO STAR

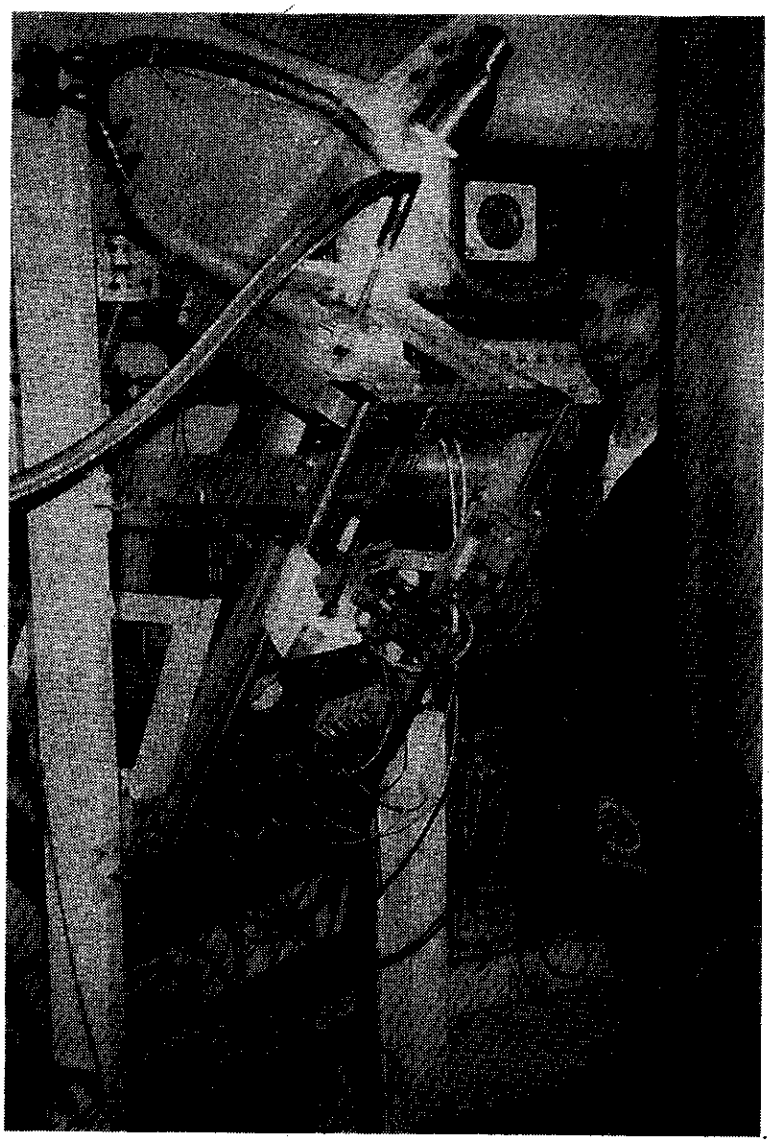
NEWSLETTER
of the
James Clerk Maxwell
Telescope

ISSN 0267-1247

Number 9, March 1990

Kulia I Ka Nu'u





Two views of Receiver C in the JCMT receiver cabin – and one of Dr Glenn White who, with Dr Rachael Padman, writes in this issue about the successful commissioning of the receiver at 492 GHz.

SOFTWARE NEWS

The following article by Rachael Padman about SPECX is the first, I hope, in a regular series about the principal software packages used in the processing of JCMT data. If you have a favourite program other than SPECX (POPS perhaps?) I would be glad to hear from you too.

Editor

SPECX CORNER

HISTORY

Since SPECX is, for better or worse, the reduction program operating "on-line" at the JCMT, and for many users is still the only program they have capable of providing data reduction facilities for their GSD spectra, I thought it might be sensible to spend some time discussing the history and philosophy of the program, and in this issue at least to dwell a bit on the current state of the program and its likely development over the next one to two years.

SPECX originated at MRAO as a data reduction package for spectra taken with the MRAO/RAL/QMC/UKC receivers at UKIRT -- the first version was written in an extreme hurry over Christmas of 1979 to allow me to get some spectra plotted for a talk in early January (having arrived back from the telescope with the first ever UKIRT submm data only 10 days previously). The only plotting device was a Benson pen plotter! Philosophically SPECX developed first as a synthesis of Colin Spratling's PDP-11 program MANIP, written at QMC as the on-line analysis program for the receiver, and Paul Rayner's program SPCRED, then in use on the Cyber 7600 at CSIRO in Sydney, whence I lately come. Incidentally, the name arose as these things do in a very innocent way; it was originally called SPEC, but then latter versions became SPEC1, SPEC2 etc. In order to avoid having to do global substititions all the time I changed it to a generic name, and as SPECN was less euphonious than SPECX, SPECX it became. Since SPECX was written on a NORD computer, it seemed appropriate to use the Nord style of command processing, and the minimum-matched multi-word command names have survived until now (about the only part of the program that has).

All this preamble is by way of establishing that the early use was for intensive reduction of a very few, very hard-won spectra -- what was needed was a fast response to commands, most of which did very little computing (hence the choice of a monolith), while there was never going to be any amount of actual data involved (the mapping software wasn't sketched out until 1982, and even then a big map consisted of 9 spectra). It was then transferred to the VAX, improved and proliferated during my 2-year stay in Berkeley, and had become quite a useful package by the time I returned to MRAO in 1984.

So how (and why!) did it get to be the JCMT package. The simple answer is that once it had been decided to use the GSD data format (GSD stands for Global Section Datafile by the way, and is not to be confused with GSDD, which is the General Single Dish Data format that describes what it is we keep in the GSD files - got it?), SPECX was the only program we had that we understood well enough to modify to read GSD data. We did have a copy of POPS from Kitt Peak, but that was a fairly old product by that time and would have been very hard for us to modify, while packages like CLAS, Figaro etc either did not exist or existed only in embryo form, and were not obviously suitable for our needs. In any case, there was a lot of other software to be written for JCMT, and since it only took a few days to modify SPECX to read the GSD files that was what I did, and that gave us a way of looking at the first spectra. Since then it has just been a matter of momentum: it is always easier to keep modifying something you have than to make a cold start with something you don't have yet... I will come back at the end to discuss the various options we now have with regard to both SPECX and other packages.

Current state and version 6.0

The JCMT represents a quantum leap in the UK's mm-wave data-gathering ability. We now regularly collect rather big spectral-line maps (requiring several Mbytes of storage for the cube), containing perhaps a few hundred to a thousand spectra, and it is I guess fairly obvious to most people the biggest single deficiency of SPECX is its command language. The current form (as of V5.4) is very intensely geared to reduction of a limited number of spectra, and was designed to be user-friendly in the sense that the command names and prompts are as self-explanatory as possible (I recognize that I have strayed into dangerous territory here - perhaps I should add the rider "in my opinion"). In V5.4 I attempted to eliminate most of the nastiest bugs (of which I was aware), and also provided an experimental DO command to allow automatic processing of larger numbers of spectra. Still, something more was called for.

I have over this last Christmas therefore written a new command language geared toward this particular program and using the same basic syntax as already existed. In fact nearly all the code was already there in the general terminal i/o routines that look after the command files, prompting the terminal and translating your input, and except for a little tidying up this has remained unchanged. What I have added is a layer of new commands, plus a symbol table for both predefined and user-defined variables, together with a general expression evaluator to allow you to do arithmetic. All the data and header can now be accessed directly via statements like

```
do m 1 no-map-pts
from-map m
do n 1 npts(1)
data(n) = 1.5*data(n)*2.718↑tau
enddo
add-to-map
enddo

int-time = 3500
```

and

```
set-map-size 100, 100*(ny*dy)/(nx*dx)
(actually equivalent to s-ma-siz 100 0 !)
```

which (I hope) are fairly self-explanatory (farewell to that awful ED-SPEC-HEAD command!) Also included are a structure IF/ELSEIF/ELSE/ENDIF with relational expressions, ASK, DECLARE and PRINT commands and a RETURN command. There is no claim that SCL (the new command language) is the ultimate in such products, and I make no excuse for re-inventing the wheel thus, but it was fun, very educational (for me), and I think will make life easier, particularly for people with big data sets. For my point of view it means that I can continue to give people elsewhere copies of SPECX to play with WITHOUT having to give them all of ADAM and the Starlink environment, which makes life easier for non-UK users.

V6.0 also incorporates a number of more minor bug fixes and alterations, most of which are aimed at standardizing the prompts for particular commands. While it is nice in interactive mode to be prompted for other information that might be required (which file to read from if more than one seems to be OK, for example), this is disaster in a batching or command-file environment, so a lot of these niceties have had to go. Mistakes now generate errors that cause the command or command file to be abandoned - no more of this "sorry I didn't like that, do you want to try again?", although it may be possible in a later version to put some of this back for interactive input only. I hope that V6.0 will be available at JCMT by the middle of the year (the real problem is documenting all these changes), and on Starlink (in GKS form) not long after.

There is one more major set of improvements required before the program really settles down, and this is to allow error and frequency arrays to be carried around with the data. AOS frequency scales are intrinsically always slightly non-linear, while the TABLES mode

of operation for RxC will generate data which don't even pretend to be regularly sampled. The same set of changes will, I hope, include rather more data in the map file for each spectrum, to allow the use of a sensible average command on map-file data, and will change from an actual stack of spectra to a stack of pointers, thus allowing spectra of arbitrary length to be processed. These improvements will I hope all be implemented before next Christmas, and should be available shortly after that.

Whither?

Or why not switch to Figaro? Starlink does employ one programmer (not me), whose workload is dictated by the Radiastronomy SIG, to work on (amongst other things) JCMT related matters. Chris Flatters during his tenure of this job worked mainly on AIPS things, while Paul Harrison has been working full time on tranplanting the JCMT-specific functions from NOD2 to Figaro, and is just about ready to begin testing this seriously. It was the SIG's opinion that this was a much higher priority than Figaro-ing SPECX, while it seems likely that the main thrust of Paul's efforts will now have to switch back to longer wavelength work again for a while at least. In other words - the main reason is manpower. It would in fact be relatively easy to ADAM-ize the main functionality of SPECX, since I think even the order of the arguments in the GENLIB terminal input routines is the same as those in the Starlink routines (no, I don't know which were written first). This is thus one option, and would probably be somewhat easier than going the full hog and putting everything we need into an existing program.

The availability of JCMT FITS tapes (at least I believe they are available), and the recent definition of a binary FITS, suggest that another set of options now includes running other packages, such as IRAM's CLAS, which should then find it relatively easy to import our data. Personally I think that it would be a good idea to have other packages available in this way, on Starlink at any rate, but I am not so sure there is room for more than one "on-line" package at the telescope, and I think this an area that merits more debate (which package should that be, and what do we expect from it?). So I'm ending this first SPECX column by suggesting as a suitable subject for "letters to the editor", the requirements for the next spectral line package.

Most of the sensible documents on this subject so far in fact originate at NRAO, which has recently been undergoing a similar sort of soul-searching. I will send my copies of these to Alex McLachlan at ROE, from whom those interested in pursuing the subject further may obtain analog copies. In the meantime, although I am not offering to act as an on-line debugging and information facility, I would like to hear from users of SPECX regarding both bugs and new facilities you'd desparately love to see, and if this generates sufficient interest we could discuss these in future editions of this column (which I hope will be somewhat shorter, now that the preliminaries have been disposed of).

Rachael Padman
MRAO