Radio Interferometry: Theory, Techniques and Applications, IAU Coll. 131, ASP Conference Series, Vol. 19, 1991, T.J. Cornwell and R.A. Perley (eds.)

SOFTWARE SYSTEMS FOR THE RYLE TELESCOPE

DJ TITTERINGTON
Mullard Radio Astronomy Observatory, Cavendish Laboratory, Madingley Road, Cambridge CB3 0HE, UK

ABSTRACT An overview of the software systems developed for the broad-banded Ryle Telescope.

THE RYLE TELESCOPE

The Ryle Telescope, at the Mullard Radio Astronomy Observatory, Cambridge, is currently completing commissioning trials. It is based on the 8-antenna East-West array of the Cambridge 5km telescope, using cooled GaAs FET receivers operating at 5 and 15 GHz, and a completely new broad-band digital correlator system, giving a bandwidth of 350MHz divided into five 70MHz sub-bands, each of 7 channels. This instrument will now begin programmes for the study of the microwave background via the Sunyaev-Zeldovich Effect, and for the mapping of extended low surface brightness structure, especially nearby galaxies.

THE COMPUTER

MRAO has been using Norsk-Data mini-computers for telescope control for over 10 years. These systems have the advantages of a dual processor architecture; a 16-bit frontend processor runs the operating system and provides fast context switching for real-time operations, leaving the 32-bit backend free to perform the compute-intensive data reduction tasks. The Ryle Telescope is controlled by a model ND-5400 system (roughly equivalent in processing power to a VAX/780). The proprietary operating system, Sintran, contains good software tools for scheduling and synchronising real-time applications.

SOFTWARE OVERVIEW

The software described here deals with the control of the telescope, acquisition of data from the digital correlator and subsequent data reduction (software for control of the embedded micro-processors within the correlator system, written by Nick Dillon and Michael Jones, is not covered here).

https://doi.org/10.1017/S0252921100013208 Published online by Cambridge University Press

It is all written in Fortran and runs on a single computer. In this sense the system is highly integrated; one machine deals with the complete process of scheduling observations, controlling the telescope and receivers during the observation, receiving data and performing all the analysis on the final maps. Observers can typically receive their first map within 15 minutes of the end of an observation, and can complete the full data reduction within a few hours. The same software is used for reduction of observations from the other major instrument at MRAO, the Cambridge Low Frequency Synthesis Telescope.

There are three main components:

- The real-time system deals with observation control, from the queueing of an observation request to the production of a *sample file* containing raw visibility data.
- The POSTMORTEM package deals with all aspects of visibility data evaluation, calibration and map-making.
- The ANMAP package provides a wide range of map analysis and display facilities.

All telescope software is written in ANSI standard Fortran 77 - with the usual extensions. This means that most of it is highly portable, and there are plans to install the data reduction packages on UNIX workstations in the near future.

What about AIPS?

The POSTMORTEM and ANMAP packages together provide similar functionality for radio data reduction as the AIPS package. They were developed for the MRAO telescopes in the days when AIPS was not available on a computer near Cambridge, and are more closely integrated to local needs. AIPS is now available on UNIX workstations whose processing power will allow self-calibration of Ryle Telescope data, and this will be exploited in the future. Both UV and image data can readily be transferred to AIPS using the standard FITS format.

The Real Time system

The scheduling of observations and control of the telescope is achieved by a set of independent but synchronised real-time processes, each of which performs a specific function. They communicate via a common data area - the control tables - which contains a full parametric description of the telescope at any instant. The control tables are at the heart of the real-time system: they are 'live' during the observation and can be accessed and updated by each of the real-time processes, and are then stored on disc with the visibility data at the end of the run to provide a complete record for postmortem analysis.

The observer's interface is an interactive program, TELCON, run at any terminal, which adds observation requests to the track queue (for routine observing, only the coordinates of the pointing direction need be specified, but control over all aspects of telescope configuration and observing style is available). No further operator interaction is required during the 12 hours of the observing run, although many monitoring facilities are available at the observer's terminal.

There are separate processes within the real-time system to perform the following functions:

https://doi.org/10.1017/S0252921100013208 Published online by Cambridge University Press

- o STCLOCK is the master process and keeps sidereal time, driven by interrupts from an external 100KHz sidereal time source.
- o INIT prepares the control tables for a new observation, by reading parameters from the track queue and initialising the other processes.
- o AEDRIVE computes new antenna pointing coordinates, applies bending and refraction corrections, and transmits these to the aerial drives.
- o PC computes new delay settings for each antenna every second, and transmits these to the path compensation system.
- o PHASE provides data for the fringe rotator micro-processors, updating every second.
- o CORR initialises the correlator at the beginning of the run, and reads data from the correlator after every integration period. Correlated data is produced for 8 different delay values in each of the 5 sub-bands, for every aerial pair, giving a typical data rate of 10 Kbytes every 12 seconds.
- o AMPHI performs initial processing, including the time-dependent FFT which generates the frequency channels within each sub-band, preliminary antenna calibration and phase correction; and integrates further before writing each 36 second sample of raw visibility data to disc.
- o PURGE is triggered at the end of the observing period to tidy up and close down the run.

The end product of the observation is the *sample file*, containing a copy of the control tables together with the complex visibility data, and some items of instrumental monitoring data recorded during the run. A sample file for a full observing run using the complete telescope will contain 8 Mbytes of data. This is the raw material for subsequent processing by the POSTMORTEM package.

POSTMORTEM

The POSTMORTEM package contains facilities for evaluating and reducing the visibility data. It is usually employed after the end of the observing run, but can equally well operate on 'live' data — the same software can be used for monitoring telescope performance and instrumental fault-finding during test observations. POSTMORTEM is a command driven interactive program, with the ability to launch detached processes for compute-intensive operations. It contains sub-systems for:

- o Monitoring telescope performance
- o Displaying and editing the visibility data
- o Performing calibrations
- o Removing flux from interfering bright sources
- o Map-making

Logical sample files

An important part of the design is the concept of *logical sample files*. The raw data files are too large to allow edited copies to exist on disc, and all editing and transforming operations, such as interference rejection, spacing selection or shifting the phase centre, are only applied to the data when it is being processed for display or map-making. The complete definition of the required operations can itself be stored and recalled as a named logical sample file,

https://doi.org/10.1017/S0252921100013208 Published online by Cambridge University Press

leaving the raw data unchanged on disc. Using this technique, the visibility data may be displayed and analysed with complete flexibility.

Editing
Any combination of baselines, frequencies or samples in time may be selected or excluded from further processing. The data can be smoothed, phase rotated and merged over frequency or baseline before display, to isolate interference or instrumental problems.

Calibration
Observations of calibration sources are made regularly, and are compared with point source model visibilities to provide estimated amplitude and phase corrections for each baseline. These corrections can then be applied to subsequent data simply by selecting the calibration as part of the logical sample file definition.

Map-making
Maps are made in a fairly conventional way, by gridding suitable weighted visibilities on to the aperture plane, multiplying the resultant aperture by a grading function and transforming into the map plane by FFT. Several choices for weighting factors and grading functions are available. The map file is the final product of the POSTMORTEM package. All subsequent analysis is the domain of ANMAP.

ANMAP

The ANMAP package deals with map display and analysis. It has similar functionality to AIPS, but with more emphasis on interactive graphics and flexibility for the production of complex diagrams for publication. A more complete treatment of polarisation and spectral index maps is also provided, including the production of error maps. ANMAP is an interactive command driven program, with sub-systems for:

- Displaying maps, using contours and greyscales
- CLEANing maps
- Combining images, making spectral index maps
- Convolving and reprojection
- Analysing images - finding fluxes, noise analysis, etc.
- Synchrotron spectra analysis

Authors

The real-time system was developed by Guy Pooley and David Titterington.

The current version of the POSTMORTEM program was designed and written by Nick Rees for the analysis of the 38MHz survey from the Cambridge Low Frequency Synthesis Telescope, with additional material for the Ryle Telescope by Paul Alexander, David Titterington and Peter Warner.

ANMAP was designed and written by Paul Alexander, with additional material from Nick Rees, David Titterington and Elizabeth Waldram.

https://doi.org/10.1017/S0252921100013208 Published online by Cambridge University Press