NOTE 201 – UPDATED AIPS++ SYNTHESIS DEVELOPMENT PLAN

Tim Cornwell

March 7 1997

Contents

1 Purpose

The purpose of this document is to lay out the steps needed in developing the synthesis capabilities inside AIPS++ from those recently delivered in the beta release AIPS++ V0.8, to those to be delivered in AIPS++ V2.0 in mid 1998. It replaces the previous Synthesis Development Plan (AIPS++ Notes 192).

I have split the developments required into those in the core synthesis system, and those needed by the individual AIPS++ sites. We plan to freeze the interfaces in the core system by late 1997. This means while not every planned capability needs to be implemented by that time, one example of each type of capability must have been tried out by that time.

The goal here is to specify the development of the core and to set a time-line for that development.

2 Current capabilities

The synthesis code is implemented using a formalism called the MeasurementEquation that is designed to allow plugging in of specialized calibration components, for both visibility and sky plane based effects. This has been extensively tested for visibility plane effects, not at all for image plane effects. The sky is modelled by a class SkyModel that has variants (ImageSkyModel) for a dirty image, two variants of CLEAN, and the NNLS algorithm. The inner loops of gridding, FFT, and cleaning have been re-written in Fortran to improve speed.

The user interface to the synthesis system is via a Glish object called `imager`. This wraps up a lot of functionality in one, probably too-complicated package. `imager` has extensive documentation (45-50 pages).

The user-level functionality is as follows:

- Filling from and writing to a UVFITS file,
- Filling from WSRT format,
- Full imaging, deconvolution, and self-calibration,
- Joint deconvolution of Stokes IQUV,
- Robust, uniform and natural weighting,
- Flexible windowing in the deconvolution,
- Non-Negative Least Squares Deconvolution,
- Flexible construction of models for self-calibration,
- A sophisticated multi-component model for gain effects,
- Graphics-based interactive editing,
- Flagging of gain solutions by antenna, spectral window, and time interval.

Thus the current system is quite capable for continuum self-calibration and imaging of polarization observations. It needs to be extended in many different directions: spectral-line, cross-calibration, mosaicing, VLBI processing, wide-field imaging, *etc.*. It needs a substantially improved user interface. Finally we need to perform a number of tests of the generality of the system before being confident that the interfaces can be frozen and opned out to other developers.

3 Required global additions

The following additions are needed for the core synthesis system, and serve to define the core system:

Improve documentation : We need a cookbook for synthesis

Tighten up current design : This is an ongoing task to simplify and tighten up the design.

Cleanup VisIter : VisIter needs some cleanup work, and also a non-MS based mode.

Evaluate role of `imager` : `imager` was written as a simple interface to the synthesis system. After some period of beta testing, we need to evaluate it's success (or not) and decide how else to present such functionality.

Change `imager` to DO V2 : `imager` is currently implemented using a deprecated version of the distributed object system. To gain the advantages of the new DO system (V2), `imager` must be re-written. It's not sensible to do this until beta-testing tells us how the functionality in `imager` should be packaged.

GUI interface to `imager` : The goal is to provide an easy-to-use interface to the `imager`. This is part of the overall GUI development for AIPS++, and is currently proceeding. I'm convinced that this has to be high priority for development, testing and updating during the entire beta release process.

MeasurementSet DO : Access to and plotting of MeasurementSets currently requires that they be loaded into Glish. This is certainly too slow for large datasets and so a distributed object interface must be developed, migrating the capabilities of visset into a MeasurementSet DO.

Add VisModels We need to add the capability to predict visibilities from VisModels, most particularly ComponentVisModels. Providing this capability is quite easy since it very naturally fits in the current framework.

Selector : We need a consistent mechanism for selecting data.

Component Sky and Vis Models : We need to be able to create, manipulate, edit, and solve for Sky and Vis Models that are composed of discrete components. Much of this work has been done but it needs finishing.

CompositeSkyModel This is a sum of ComponentSkyModels and ImageSkyModels.

Expand capabilities of deconvolution methods : a maximum entropy type method is needed, for both Stokes I and polarized emission. As part of his PhD work, Sanjay Bhatnagar is working on the development of Pixon-based algorithms in AIPS++.

Expand suite of calibration models : outstanding are a model for the ionosphere, and a model for frequency-dependent polarization leakage.

Test coupled solutions : Are coupled solutions for D and G possible, convenient, and efficient?

Simulation : A crude simulation capability has been developed but this needs further work and integration into the system. It also needs conversion to DO V2. We need to extend this to a test suite.

Miscellaneous minor `imager` improvements : A number of small additions are needed *e.g.* tapering, uv ranges, phase-shifting, tracking of true noise levels. These are fairly easy to do.

Non-symmetric sampling of the Fourier plane : Currently only those points with all 4 correlations are gridded when imaging all 4 Stokes parameters. The ME formalism is capable of dealing with non-symmetric coverage but the current implementation cannot. Also we need to make (XX,XY,YX,YY) and (LL, LR, RL, RR) images a standard product.

Optimization of Spectral line gridding and Fourier transformation : currently, one pass is made per channel. This can be improved by writing entirely new gridding code. This is very high priority and is planned for V0.9.

Doppler tracking : We will do this on the fly in the VisIter but the tools will be available elsewhere as well.

Disk-based FFTs : currently, FFTs are memory-based. This must be changed for large images. This is also needed elsewhere in the project, and is reasonably decoupled from the core synthesis code. We also need to deal with symmetries when present *e.g.* Hermitean symmetry if there are no image plane based effects.

Optimization of Gain solutions : Gain solutions are too slow for both continuum (4-5) and spectral-line (order of magnitude). Both can be improved by judicious averaging. This is very high priority and will be part of V0.9.

Formalization of calibration table conventions : The calibration table format must be extended and defined, using mechanisms similar to the MeasurementSet.

Full implementation of cross-calibration : Cross-calibration involves a large number of user-related issues concerned with helping users keep track of calibration information and versions. We also need better tools for handling calibration tables, including editing and interpolation. This all needs careful user testing.

Visualization and editing of visibility data : This area requires a lot of work but Jan's data visualization tool provides a good start. Some work on plotting can proceed using the binding of PGPLOT to Glish but visualization proper should wait until the image display library is ready. GMRT has expressed an interest in pursuing this general subject.

Multi-stage processing : We need to ensure that we can deal with the large data volumes that will come from *e.g.* VLBI. This will require multi-stage processing (pre- and post-fringe-fit), and averaging.

Multi-field and wide-field processing : The current implementation has hooks for multi-field processing that have not been tested.

Mosaicing : The ME formalism must be tested soon by extending it to deal with an image plane calibration effect such as mosaicing, preferrably including some self-calibration such as pointing correction determination. There seem to be two different concepts of mosaicing being used, differing in how model components are treated; we should support both. The core requirement should be a demonstration of mosaicing imaging and calibration. A fully worked out package should be developed after the core is finished.

Determination of instrumental parameters : We need to provide tools for determining instrumental quantities such as system temparatures, delay offsets, baselines, pointing offsets, *etc.* either from calibration tables or via specific Jones solvers. While many of the procedures are observatory-specific, the tools are generic. Holographical observations can in principle be reduced using the ME; we should do this.

Calibration of Tied arrays : Can the implementation cope with calibration of Tied arrays? That is, transfer the calibration from the elements of a Tied (or phased) array to the Tied array itself.

Extension to Single Dish processing : The MeasurementEquation and the MeasurementSet can both handle single dish observations. This should be one of the tests of the system. Probably OTF is the first case to test.

Freeze interfaces : The interfaces to the synthesis code must be frozen prior to allowing much work on variants of the basic capabilities. A reasonable goal for this work would be to have it all done by the time of the first release of a code development system, which means probably by the end of 1997.

4 Site-specific changes

Assuming that we implement the above global changes, then the following site-specific needs must be met:

4.1 Outstanding NFRA needs

1. Visualization of coherence data

2. Support for a library of ComponentSkyModels.

3. Redundancy calibration (I believe that this is mainly a WSRT concern).

4.2 Outstanding ATCA needs

1. Calibration solution for bandpass and gain using the parallel hand correlations.

2. Calibration solution for polarization leakage, gain and optionally source polarization using all four correlations.

3. Multi-frequency synthesis using the Sault algorithm.

4. Atmospheric corrections using results from water vapor radiometers.

5. We have decided not to use the Miriad model for spectral imaging (sources move, sidelobes are fixed). Instead we will continue to use the current model in AIPS++ (sidelobes move, sources are fixed). However, we may need to allow interconversion.

4.3 Outstanding BIMA needs

1. A filler for BIMA data.

2. Support for time-sliced fashion polarization measurements, measuring circular polarization (LR and RL) with a quarter-wave plate. See also Wright (1995a) for a discussion on some of the possibilities.

3. Deconvolution of mosaiced fields, including pointing corrections. Also adding single dish data to interferometry data.

4. VLBI: BIMA regularly participates in mm VLBI experiments. The phased array data are currently processed offline using standard VLBI techniques, and whenever AIPS++ will provide VLBI data processing, BIMA should be able to use them without any major problems.

5. Heterogenous array elements.

6. Unusual correlator modes: The correlator can be configured in many modes, and produces DSB data with a small (less than 8) number of windows with different settings of the IF. An interesting method to calibrate DSB data is to use the generally much slower varying gain ratio (phase difference and amplitude ratio).

7. Offline phase corrections, using total power measurements exemplify one of the many ways in which calibration needs to be flexible.

4.4 Outstanding VLA needs

1. A VLA filler.
2. Ionospheric corrections from GPS.
3. Automated flagging.
4. Wide-field imaging

4.5 Outstanding VLBI needs

VLBI development is the subject of a separate memo from Athol Kemball. Some elements of VLBI processing are planned as part of the development of the core system (*e.g.* fringe-fitting) but otherwise it is seen as being separate from the core and some fraction of it will proceed after the core has been defined and frozen.

5 Prerequisites

We need the following to be to be delivered from the rest of the project:

Measures support for Tables/MS : Needed in many different areas.

Image display library : To make progress on visualization of MeasurementSets. By mid-year 1997.

Support for Complex Images : We need to be able to display and manipulate Complex Images, probably by conversion to Float images, but I leave this open. We need this for various reasons, the most pressing being for *e.g.* (LL,LR,RL,RR) or (XX,XY,YX,YY) images, which I propose to make a standard product of the synthesis package.

Definition of standard Image : I propose that we define a standard Image (either StokesImage or DSSImage) derived from Image with a guaranteed minimum set of axes: Direction, Stokes, and Spectral. Such an object is vital to make the synthesis code more robust and less prone to assumptions, just as the MeasurementSet is needed in place of a fully general Table. Some of the functions now in StokesImageUtil can be moved into DSSImage.

Image support for FFTs : The FFT library should modify as required the image coordinate system, using it for directions as to what is to be transformed.

Convert LinearModel to use Lattices : We need this to avoid memory bloat.

Conversion of image projections : For wide-field imaging, we need to be able to convert from one Direction projection to another. This can limit the speed of wide-field imaging and will require some optimization.

Rationalization of the storage of data : Wim has a proposal addressing this issue.

6 Development plan

Here we cover the parts of the development plan appropriate to synthesis imaging. The goals of this development plan are to address pressing needs of the consortium partners and to push forward into new application areas. It is expected that this development plan will be revised from time to time to incorporate, for example, VLBI processing when appropriate.

Items in italics are not directly part of synthesis processing but are needed:

Target	**Duration**	**Due date**	**Who?**
Initial version MeasurementSet DO	1 week	March 97	Cornwell
Non-Solvable ComponentSkyModel	1 month	April 97	Marson
CompositeSkyModel	1 month	April 97	Marson
Optimization of gain solutions	1 month	April 97	Wieringa
Optimization of gridding	1 month	May 97	Cornwell
GUI Interface to `imager`	2 months	May 97	Glendenning
Non-symmetric sampling of Fourier plane	2 weeks	May 97	Cornwell
Change `imager` to DO V2	1 month	June 97	Cornwell
DSSImage	1 month	June 97	?
Support of Complex Images	1 month	June 97	?
Measures support in MS	1 month	July 97	Brouw
Formalization of calibration tables	3 months	August 97	Kemball
Simulation software	2 months	July 97	Marson
Disk-based FFTs	2 months	June 97	Brouw
Miscellaneous `imager` improvements	1 month	June 97	Cornwell
Mosaicing: imaging	2 months	July 97	Wieringa?
Image Projections	1 months	August 97	Brouw?
Full implementation of cross-calibration	3 months	October 97	Kemball
Wide-field imaging	2 months	October 97	Marson
Ionospheric Model	1 month	October 97	Noordam
Determination of instrumental parameters	3 months	October 97	?
Visualization and editing of visibility data	4 months	October 97	Noordam
Testing of the ME on single dish data	2 months	October 97	?
Fringe fitting and averaging	2 months	October 97	?
Calibration of Tied arrays	2 months	October 97	Noordam
Freeze interfaces	1 month	November 97	Cornwell

The main unknowns in the development plan are the impact of the beta release process, and of NRAO's commitment to OVLBI on the availability of personnel on the time scales described here.