External JIVE & NRAO VLBI for CASA development meeting

Date: 9 June 2016, 2-3PM
Present: Jeff Kern, George Moellenbrock, Des Small, Ilse van Bemmel

After in-depth discussion with George on Tuesday and this morning, we have a meeting to discuss the overall timeline and implementation of VLBI tasks in CASA. This meeting focuses on the fringe fitting algorithm being developed in JIVE by Des. After additional discussion it is decided to name the task *fringecal* in CASA.

The status of the algorithm is that the prototype in Python in working and verification against AIPS indicates the functionality is good, but some more debugging needs to be done. This is best done in Python. We plan to be finished by summer, at which time the code needs to be ported to C++ for production and inclusion in CASA. This requires the following work to be done:

1. Provide table support for fringe fitting solutions

Des and George have started working on a skeleton FringeJones class, which includes the class itself, a rudimentary test pogram, (Solvable) VisCal factories and a connection to the build system. This class provides the socket for the C++ code that Des is developing. (As of 18 June this has been checked into the repository by George).

2. Timeline: port the prototype to C++

The current timeline for CASA is to freeze per August 1 and release version 4.7 in September. We cannot make that deadline for the C++ version, but it will be possible to have code ready for checking in as soon as the release is complete. This coincides with a migration of CASA from svn to a git (bitbucket) repository, so it seems natural to hold off with checking in too much code until that is complete.
The plan is to aim for implementation of the new FRING task in the March 2017 release, which will coincide with the work that Mark Kettenis does. The new release will probably be 5.0, and would be the first CASA release with VLBI data processing capacity.

3. Develop actual CASA task and if necessary adjust CASA tools

Writing the actual CASA task involves writing an XML definitions file, which is a fairly basic task. We can design the required parameters at JIVE, and suggest to use AIPS inputs as the basis. We will avoid using ambiguous parameters such as 'APARM' and 'DPARM'. George can translate this into the CASA XML format. The binding to the Python layer of CASA will be nearly identical to the existing gaincal task, and should also be easy to implement.

4. Full CASA integration (plotting, flagging)

Inspection of solutions and flagging is done with the CASA task plotms. Currently this doesn't know how to handle FRING solutions. Pam Ford is the NRAO person working on plotms and is already thinking about how to select and plot specific types of data. Adding delay, rate, and phase solutions out of a FRING task would be a natural addition. This is anticipated to be implemented for the CASA 5.0 release, and may or may not be available for commissioning during the development period. The alternative task plotcal will be phased out soon and should no longer be used.

5. Documentation writing

Documents need to be written in HTML format and will be integrated with the CASA plone based documentation (under development). The documentation will then be maintained with the rest of the CASA documents.

6. Testing, verification and benchmarking

There are basic functionality tests for CASA code, George will show Des how to work with them. Google Tests is the platform for this. There are also Python functional tests that need to be included. Functional verification for the C++ production code against AIPS will continue in JIVE as done for the current prototype. This will include some benchmarking. A well known issue with CASA is I/O, should we run into issues there, we can contact the NRAO developers for assistance.
Currently CASA is based on Python 2.7, and there is no intention to move to Python 3 any time soon, since there are too many interdependencies to solve at the moment. For developers wishing to build their own CASA, there is a RedHat RPM of build dependencies available.

7. Future ideas

We briefly discussed parallelization of the calibration process. There are already ideas about this for gain calibration, which work with MMS (Multi Measurement Sets). An MMS is essentially a stand-alone part of a complete MS that is calibrated individually. This is MPI based and in the long term could be useful for FRING.

Implementation of alternative fringe fitting algorithms and new algorithm development should be possible, but is something for the longer term future. First we need to sell the basic functionality, and the NRAO synthesis school of 2018 could provide a great platform for this.

Actions

1. Write table framework (Des & George)
2. Get access to git repository for Des (Jeff, George)
3. Define CASA task parameters (Ilse, Des)
4. Define CASA XML definitions file for task (Des, George)
5. Bind to Python layer (George)
6. Integrate into plotms (George will notify Pam)
7. C++ testing (Des, George)
8. Python based testing (Des, George)
9. Verification of C++ code against AIPS (Ilse, Des)
10. Benchmarking of completed task against AIPS (Ilse, Des)