On line simulation in ALMA Software

Robert Lucas

Summary

- Need for hardware simulation for software testing
- Requirements for on-line simulation
- A Shared Simulator
- Limitations
- Current capabilities
- Further work

Need for on-line simulation

- Main motivation is to test software
  - At the subsystem level: simulate the hardware for which specific software (drivers) are written or other pieces of software
    - *Not the subject of this talk*
  - At the system level: use all the software, without any hardware
    - This is done to integrate all the subsystems together (integrated tests).
    - This is mainly done in the process of software integration.
    - We have multiple test systems (STEs) to test integrated software on the different development sites (ESO, NRAO, NAOJ, ALMA/Chile)

Requirements for on-line simulation

- Perform an end-to-end simulation:
  - Put Scheduling Blocks (SB) in the Archive using the Observing Tool (OT)
  - Drive the system from the Operator interface (OMC)
  - Schedule a SB
  - Execute it, including On-line Calibration and Quick-Look
  - Check the resulting data (ASDM) in the Archive
- To do this we need to:
  - Simulate all hardware devices by their own simulators (e.g. antenna mounts, local oscillators

Shared Simulator

- In order to get meaningful data, we need, e.g. :
  - The total power detectors know what the antenna is doing (on-source or off-source?)
  - The correlator knows what the interferometry specific hardware is doing (delay and fringe tracking).
- In principle do this fast enough to simulate the actual observing process
  - In practice do this on a timescale of 48ms (1 TE).
- We have a centralized software component (the Shared Simulator)
  - This component communicates with the individual hardware simulators so that those that produce data do this in a consistent way.

Example 1

- To calculate the total power output of a detector, the shared simulator would:
  - Get the current antenna pointing direction
    - Talk to mount simulator
  - Get the actual sources in the beam at current time
    - Have a model of the sky
  - Compute the expected detector output
    - Know the current attenuator setup
    - Know the LO1, LO2 frequencies for that baseband
    - Know to which receiver band that baseband is connected (USB, LSB)
    - Know the receiver gain and noise at this frequency
    - Know the atmosphere properties at this frequency

Example 2

- To calculate the correlator output in a baseband/baseline (*ab*), the shared simulator would need to:
  - Get the current antenna pointing directions for antennas *a* and *b*
    - Talk to mount simulators, focus control simulators
  - Get the actual sources in the beam at current time
    - Have a model of the sky
    - Compute the visibilities for sources in the primary beam
  - Compute the expected correlation function for baseline *ab*
    - Know the current attenuator setup
    - Know the LO1, LO2 frequencies for that baseband, and the phase commands and phase rates applied to those LOs
    - Know the delays applied at various stages (sampler clock, correlator)
    - Know to which receiver band that baseband is connected (USB, LSB)
    - Know the receiver gain and noise at this frequency
    - Know the atmosphere properties (absorption, added path) at this frequency

Limitations

- This can become very complex, given the complexity of ALMA hardware.
- All devices in the data path can have an effect on the correlator output
- These effects can be very subtle (there might even be some that we do not anticipate fully at this time, hopefully a few...)
- Sending very often a full set of correlation functions to the correlator simulators can be very heavy in terms of data rate...
- Thus we needed to simplify a lot; fortunately the scope of the simulation is limited.
- To start with, some of the control software was not yet developed (e.g. LO control), so we needed to bypass those.

Actual scope?

- Provide a simulation realistic enough that we could read the data through data reduction software, and see that it is not grossly wrong
- The goal is to check that the software is:
  - Not crashing
  - Going to the end of the Scheduling Block
  - Provide the expected values in the various columns of the ASDM tables
  - Even more: we see the source, we see fringes, the phase is not drifting…
- There more efficient ways of doing data simulations that are capable of producing results suitable to fully qualify the observing strategies and data reduction algorithms of ALMA .
  - *That's of course the rest of this workshop.*

Current simulation capabilities

- Simulate 2 point sources in the sky (e.g. 1 object, 1 calibrator), with flat spectrum
- Assume rectangular bandpasses
- Flat-spectrum system temperatures
- No atmospheric fluctuations
- Describe correlation functions by a few parameters (amplitude, phase, delay, phase rate, gaussian noise level), that the correlator simulator uses to generate the lags from a simple formula, including random Gaussian noise.

Other functionality

- Switch on/off the use of ‘actual’ antenna pointing directions (assume perfect pointing, focus tracking)
- Switch on/off use of LO tracking, delay tracking (assume it works, i.e. replace by ideal values)
- Actually most if not all of the simulations we’ve made in the last few years have been using these ‘off’ modes...

Further work?

- May be justified by the unavailability of a working ALMA interferometer after ATF closes, and until OSF interferometry is achieved.
- May also be justified by the improved simulation capabilities in Control Software (simulating more than 2 antennas in the same computer) which means that we may need only two computers to run such a simulation (Simulation FBT)
- Use full implemented functionality
- Possibilities: add WVR receivers, simulating e.g. the pathlength fluctuations by a simple waveform.
- Having a casa script to check the final product of a simulation run would be useful.