

# MeqTrees and Focal Plane Arrays

A.G. Willis

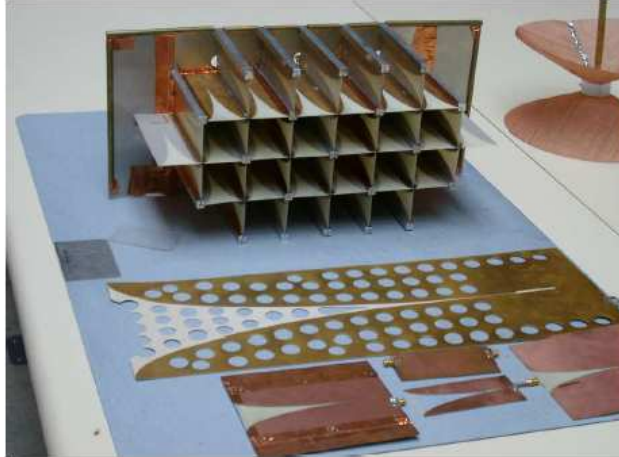
National Research Council of Canada  
Herzberg Institute of Astrophysics  
Penticton, BC, Canada

O.M. Smirnov, M. Mevius, S. Yatawatta  
ASTRON, The Netherlands

# The Problem

- Focal plane arrays used successfully for many years on single dish telescopes
- Now people would like to use them on synthesis arrays for rapid wide-field imaging
- Issues with focal plane arrays and synthesis telescopes
  - Off-axis beams can have ugly shapes
  - Focal plane arrays mounted on an AzEl telescope rotate on the sky
- Adapt Vivaldi FPA 311 MHz Beams calculated by W. Brisken (EVLA Memo 69) for our simulations

# Some Actual Vivaldi Arrays



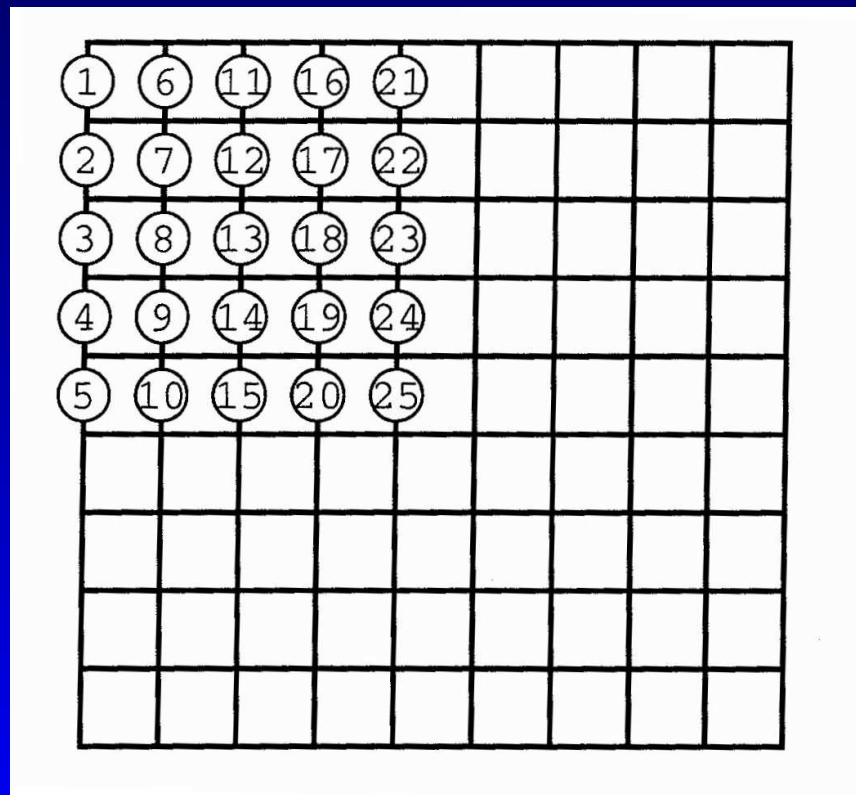
Vivaldi elements and array designed and built by Ed Reid at DRAO.



UMass Antenna Lab designed, fabricated and tested the above 8x9x2 array. Support was from ASTRON. Picture courtesy Dan Schaubert.

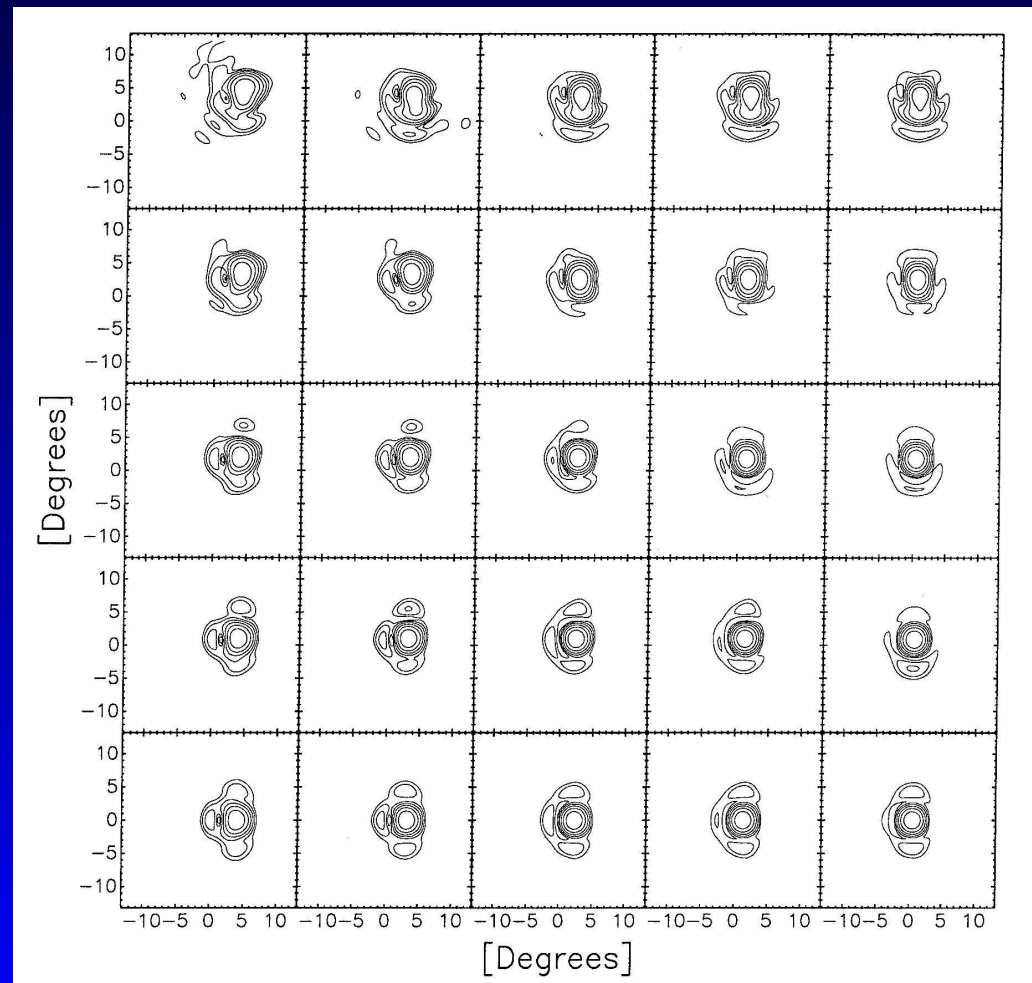
# Brisken Beam Locations

- Vivaldi array has 25 unique beams. We use symmetry / mirroring to fill up the array.
- 90 beams in horizontal direction
- 90 beams in vertical direction



# Brisken Beam Shapes

- Beam shapes are good when near boresight (lower right)
- Beam shapes are poor when far from boresight (upper left)



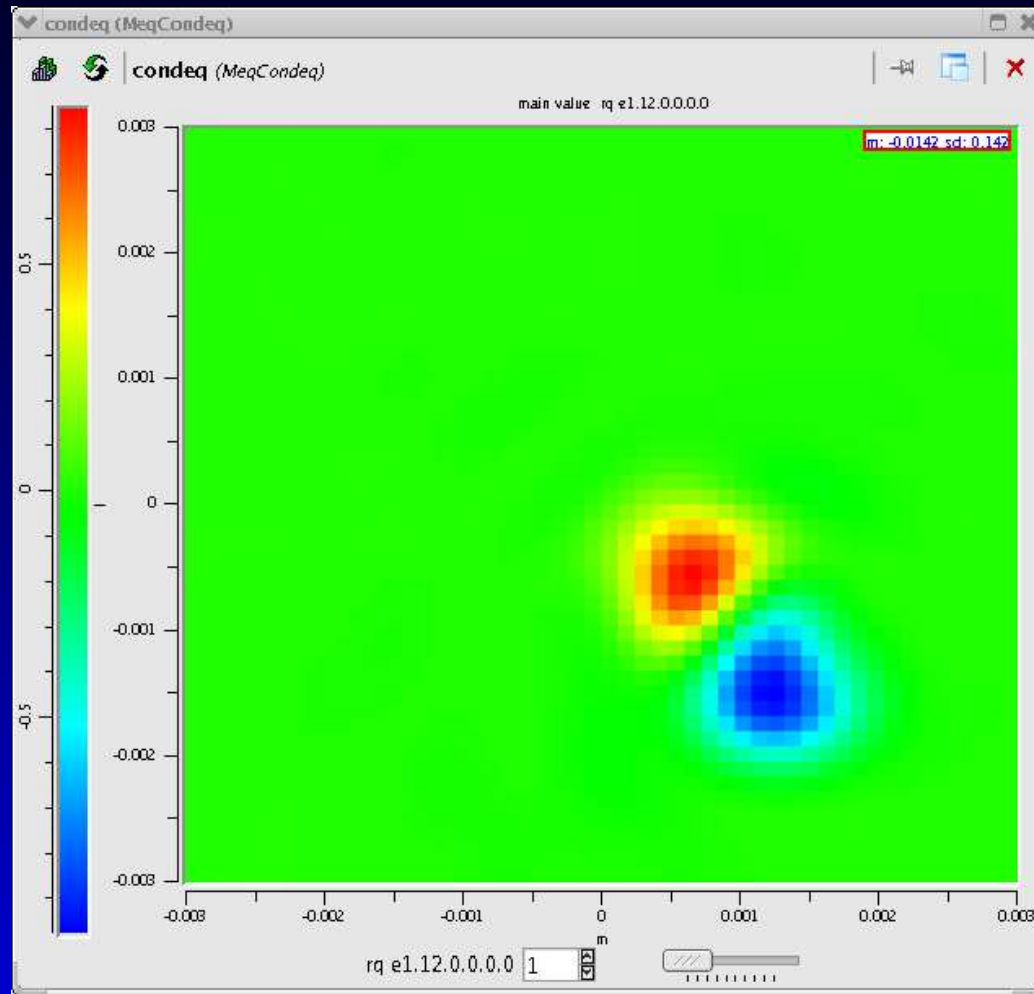
# Simulated Telescope

- 30 antenna system with about 7 km aperture
- antenna positions provided by Tim Cornwell
- dish sizes about 10 to 12 m, HPBW about 0.8 deg, frequency is 1400 MHz
- FPA on each dish has 180 beams on sky: beams scaled from Brisken 311 MHz beams
- obtain positions of individual FPA beam peaks from Gaussian fitting

# Obtaining Beam Parameters

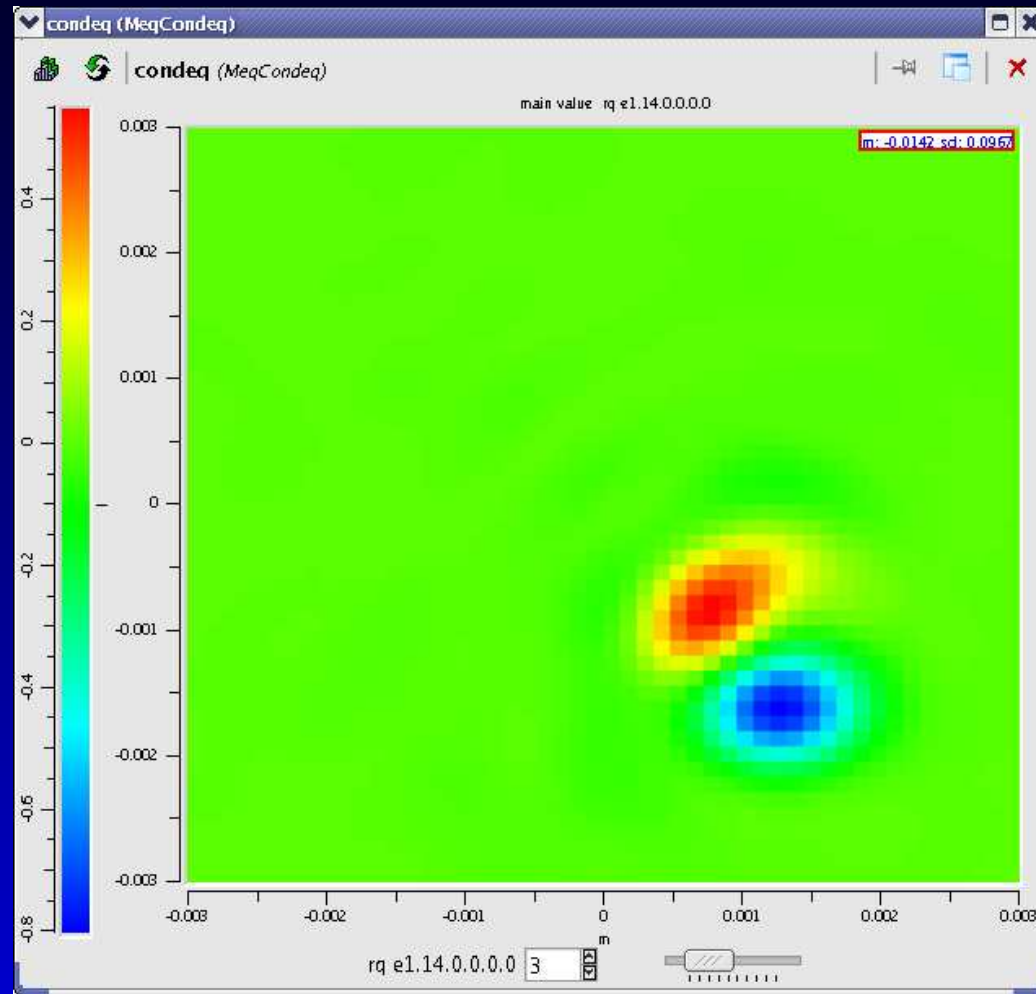
- Use MeqTrees solver to solve for positions of beam peaks and store in an aips++ table for later use.
- Condeq node shows difference between current guess for solution and what's actually observed.
- If we iterate toward a good solution, condeq node should go toward zero.

# Beam Position - First Guess

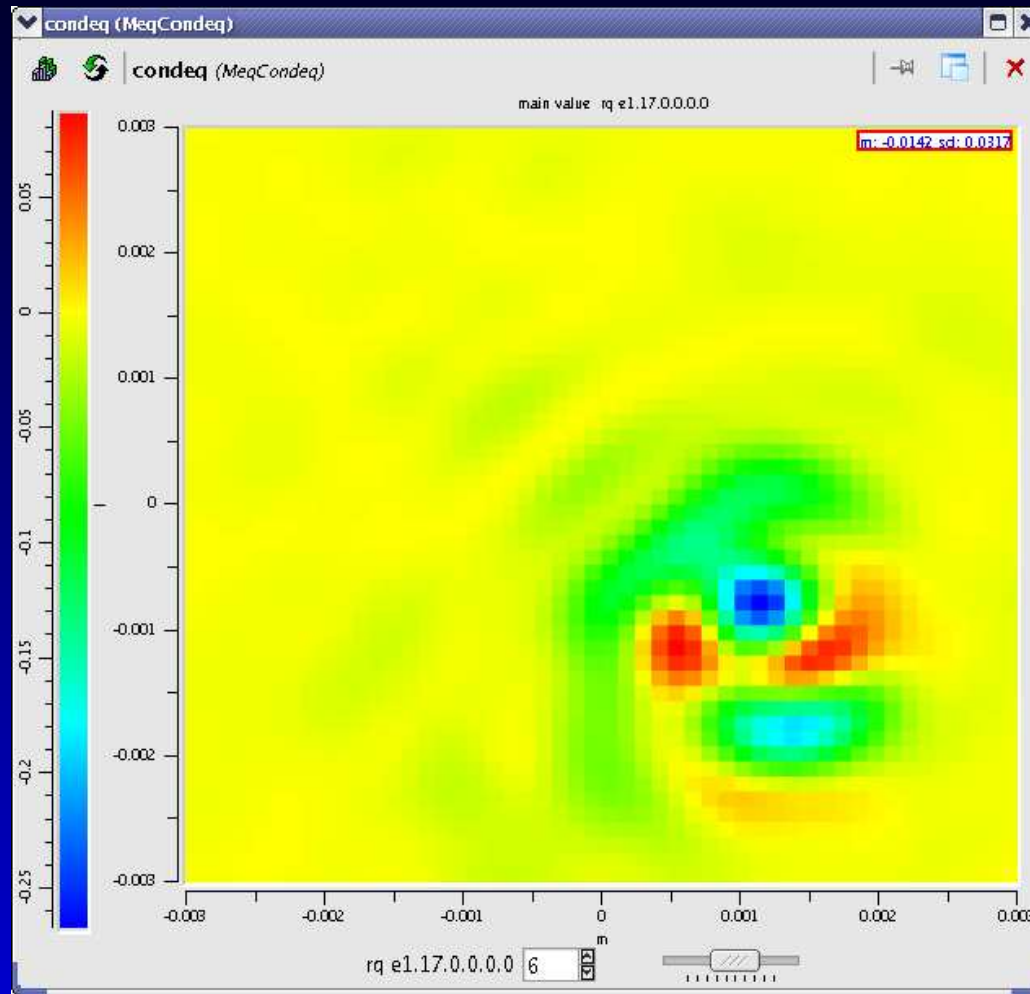




# Beam Position - Second Guess



# Final Solution



# Beamformer Algorithm

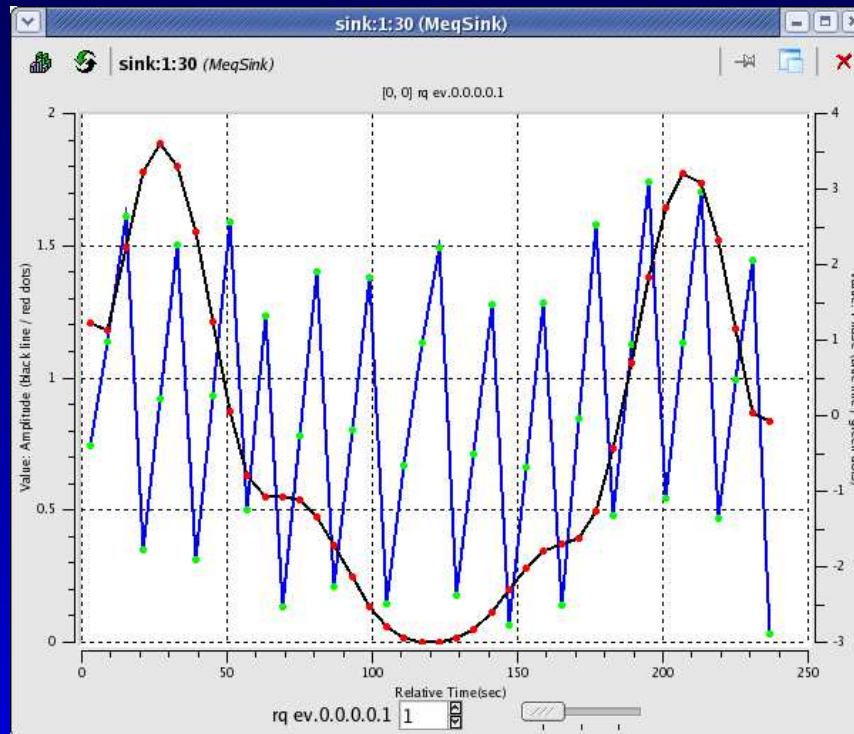
- Once we have parameters for individual beams -
  - Combine individual beams together with different weights (beamformer) to get optimized ‘phased’ array beam
  - Current simulated beamformer just weights individual beams inversely proportional to distance of their peaks from the ‘phased’ beam position
  - Improved algorithm obviously needed to get proper beam forming weights (future work)

# AzEl Telescope Experiment

- Simulated telescope array tracking centre set at -45 deg DEC
- Observe from -0.8 to 0.8 hr HA - integration period is 6 sec
- Focal plane array 'phased' beam is offset 3 deg in DEC from the tracking centre (equates to an offset of roughly 3x HPBW of a individual primary beam)
- If we have AzEl telescope, focal plane array beams move relative to sky
- Focal plane array beamformer weights must be updated on a regular basis to keep 'phased' beam at a constant RA, DEC

# 240 sec of data, 40 Visibilities

- The difference between visibilities when beamformer weights are updated every integration vs what happens when weights are updated every 240 sec



- The left hand scale indicates the percentage difference in amplitude (red dots connected by black line)

# MeqTrees Browser

The screenshot displays the MeqTrees Browser interface, titled "meqserver.0.5849.0 - executing". The interface is divided into several panels:

- Trees:** A hierarchical tree view on the left showing the structure of the simulation, including nodes like "Forest state", "All nodes (294)", and "By class (25)".
- Tabbed Tools:** A central panel showing the execution of a script named "MG\_AGW\_project5.py". The code includes a shebang line, a copyright notice for ASTRON (2006), and a license statement (GNU GPL).
- Gridded Viewers:** A right-hand panel containing four plots:
  - sink:1:30 (MeqSink):** A line plot showing Amplitude (black line/red dots) and Phase (blue line/green dots) versus Relative Time (sec). The amplitude shows a complex oscillatory pattern.
  - sink:12:25 (MeqSink):** A similar line plot for a different sink, showing a different oscillatory pattern.
  - image:1 (MeqFITSImage):** A 2D heatmap showing the intensity distribution of the first image. The main value is  $m: 0.0276$  with a standard deviation of  $sd: 0.102$ .
  - image:25 (MeqFITSImage):** A 2D heatmap showing the intensity distribution of the 25th image. The main value is  $m: 0.0147$  with a standard deviation of  $sd: 0.0778$ .

The status bar at the bottom indicates the execution progress: "executing 0:04:40 ETC 0:51:24 8% tile 2, t/s 80-119, r/t 0:08:00 to 0:11:54; avg 3.21 sec/ts VSZ:874M RSS:721M".

# Conclusions

- Amplitude error less than 0.5 percent within about 1 minute of central mean.
- The plot suggests that beamformer updates once per minute might be adequate, but further study needed to confirm this.
- Note: one important result from this analysis -
  - Simulations are incestuous - error in a calculation can lead to offsetting result elsewhere.
  - We really need at least two completely independent simulation packages with independent approaches to problem.
- future work - proper beamformer, polarization, etc etc ...

# Questions?

- Email: [tony.willis@nrc.ca](mailto:tony.willis@nrc.ca)