21cm Instrument Simulation Software and Foreground Subtraction

Dave McGinnis

3-D Radio Maps

- With the 3-D radio intensity map, we will pull the unique BAO signal out of a map that is dominated by foregrounds.
- Foregrounds consist of galactic synchrotron emission, point radio sources, etc.
- Foreground subtraction will be one of the most difficult parts of the project and will dominate the design of the instrument
- To tackle foreground removal, we are developing a large package of software to simulate the telescope and test foreground removal

Sky Simulation Software

- Simulate telescope response
  - Include the effects of noise and telescope errors
  - Provide data stream that would mimic the real telescope
- Reconstruct the sky temperature from telescope response
- Foreground removal algorithms

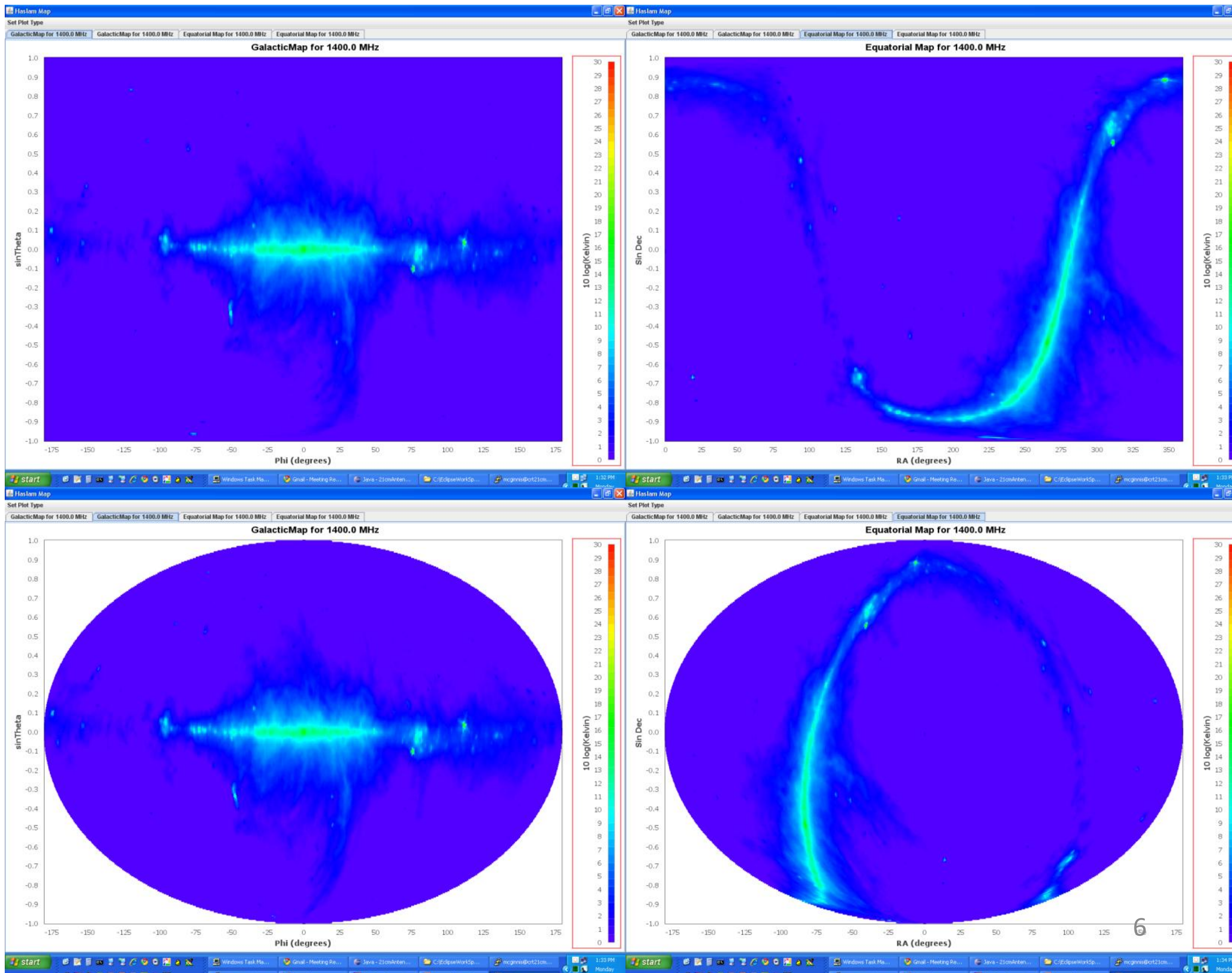
Code Suite

- 29 Java classes organized into 6 packages
- Major Packages
  - Sky Map Generator
  - Cylinder Visibility Simulator
  - Cylinder Visibility Modeler
  - Sky Reconstructor
  - Foreground Removal Algorithms

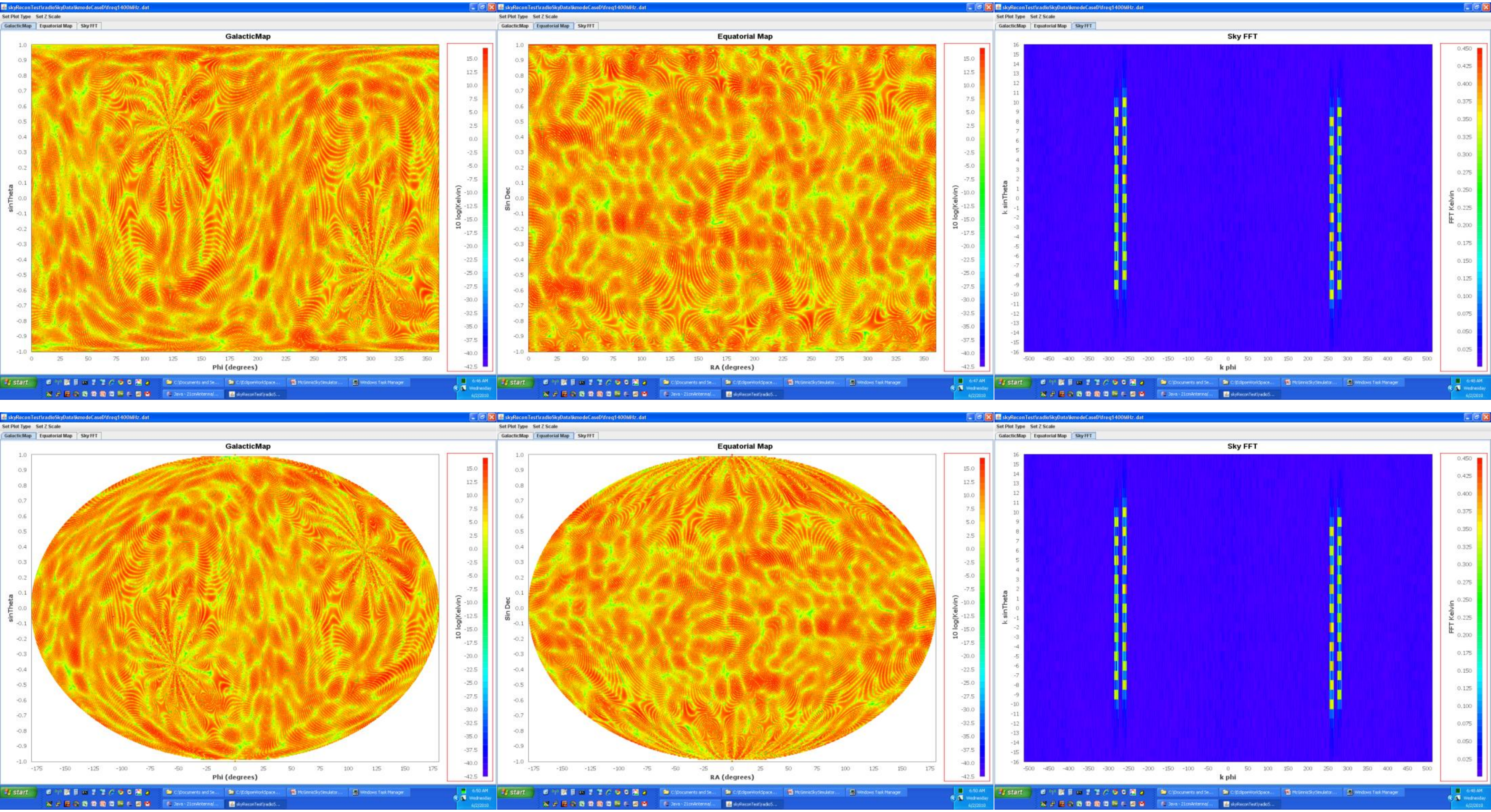
SKY MAP GENERATOR

Sky Map Generator Plotter for Haslam Sky Map at 1.4 GHz

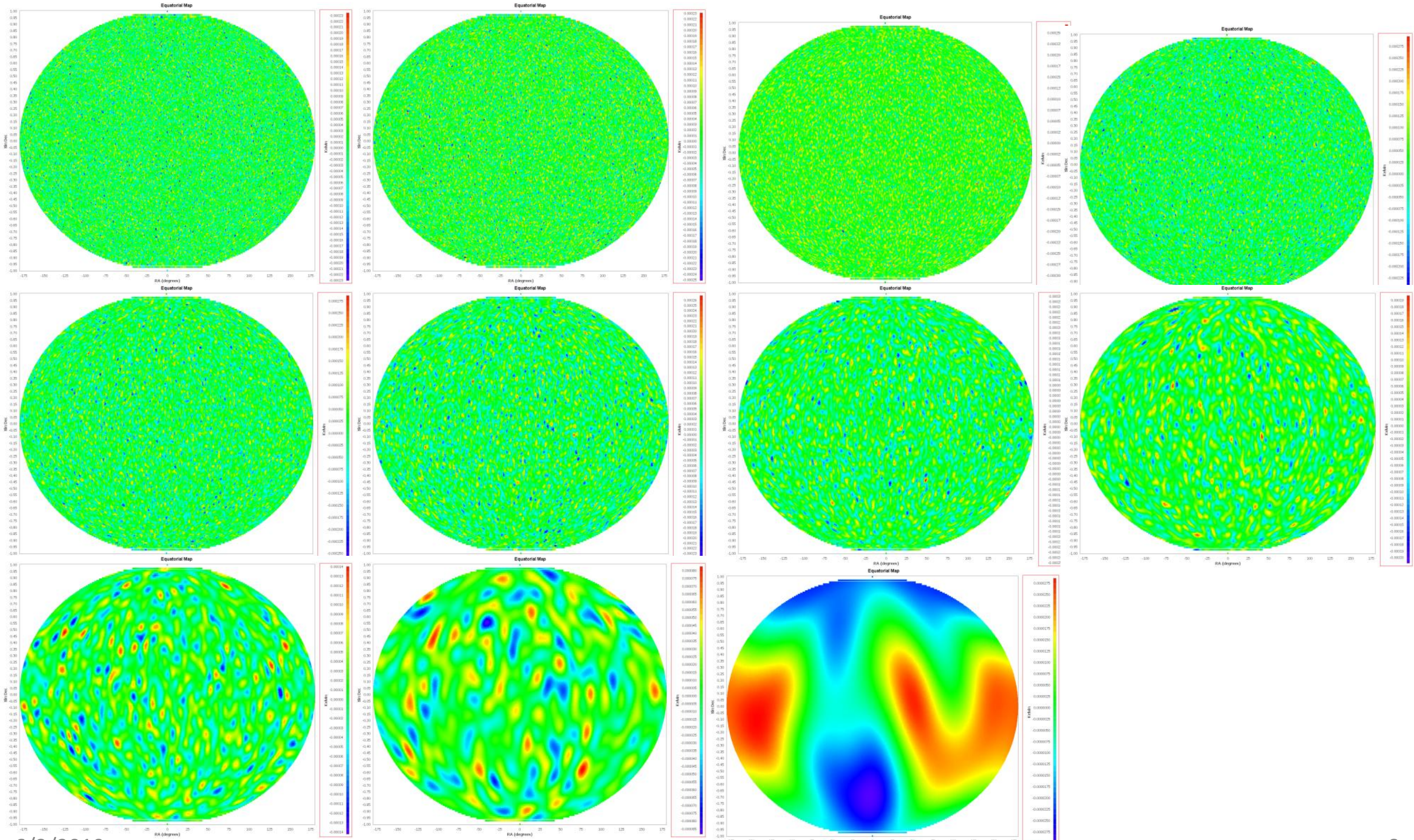
- Maps in Healpix format
  - Nside = 512
- Maps use MIT Angelica 10 parameter frequency fit
- Maps are about 100MB in size



Designer Skies



BAO Signal First Peak from 400-1400MHz



CYLINDER VISIBILITY SIMULATOR

Cylinder Visibility Formulation

Formulation of Cylinder Visibilities

Dave McGinnis
November 5, 2009

Feed Amplitude

The voltage received at a feed located at coordinate r is:

$$\frac{v(\vec{r})}{\sqrt{2R}} = \iint_{\Omega} f(\Omega) a(\Omega) e^{-j\vec{\beta}(\Omega)\cdot\vec{r}} d\Omega \quad (1)$$

The sky flux amplitude is:

$$|f(\Omega) d\Omega|^2 = \frac{kT_{sky}(\Omega)}{\lambda^2} d\Omega_{pow} \quad (2)$$

where $d\Omega_{pow}$ is the differential power solid angle area. The incoming wave vector is:

$$\vec{\beta}\big(\Omega(\theta,\phi)\big) = \frac{2\pi}{\lambda}\left(sin(\theta)\hat{x} + cos(\theta)sin(\phi)\hat{y}\right) \quad (3)$$

The collecting area of the feed is:

$$A(\Omega) = |a(\Omega)|^2 \quad (4)$$

The noise power generated by the feed amplifier is:

$$P_z = |p_z|^2 \quad (5)$$

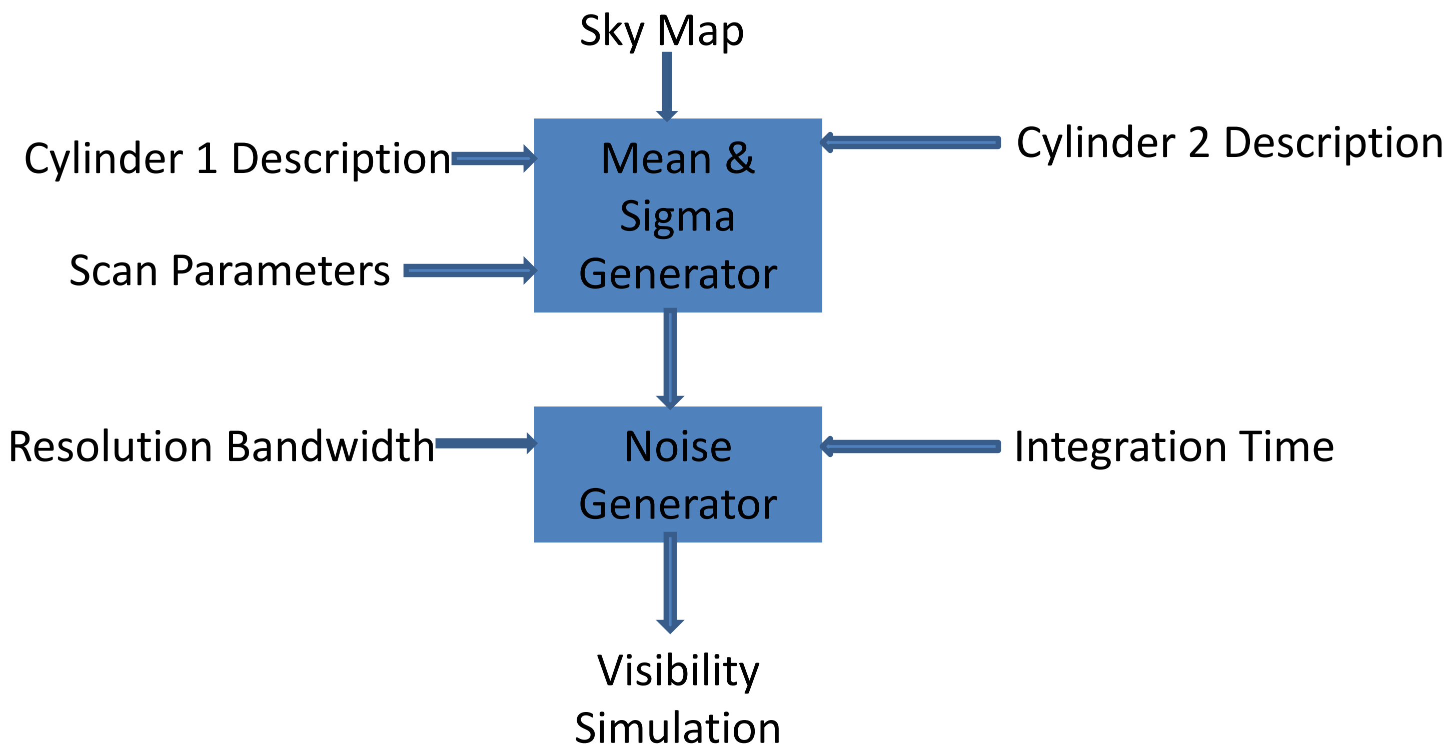
If the sky is pixelized into **q** pixels then, the signal amplitude at feed **n** of cylinder **m** is:

$$p_{n,m} = p_{z_{n,m}} + \sum_q \Delta\Omega_q f_q a_{q,n,m} e^{-j\vec{\beta}_q\cdot\vec{r}_{n,m}} \quad (6)$$

Cylinder Amplitude

A spatial Fourier transform will be taken of the cylinder feed voltage.

Cylinder Visibility Simulator



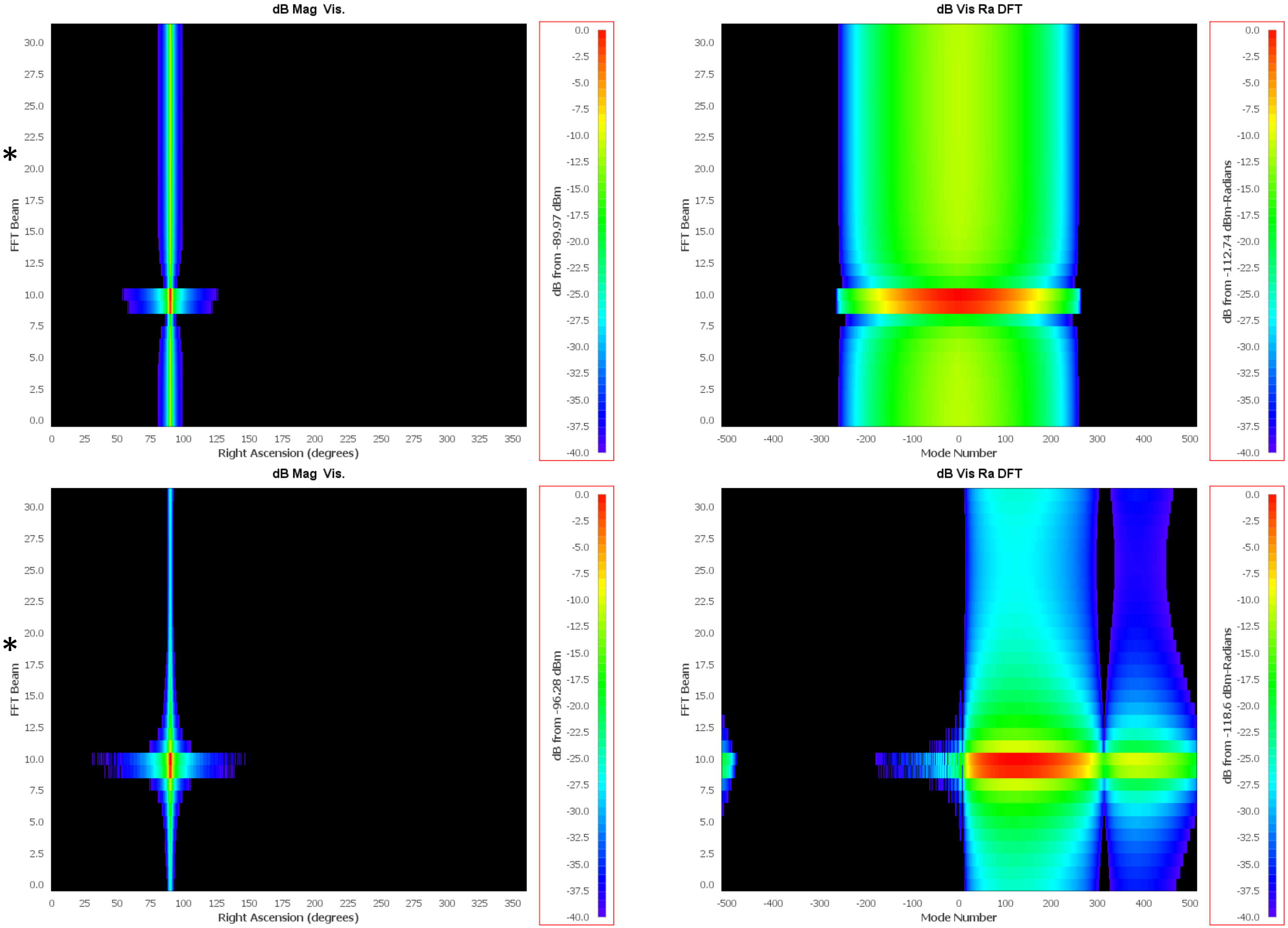
Cylinder Description XML Format


```xml
<?xml version="1.0" encoding="UTF-8" standalone="no" ?><!--blah--><!--
--><Cylinder>
    <Feed ImagGain="0" RealGain="1" effWidth="7" lengthLocation="0" nCosThetaResponsePower="1" noiseTemp="75" powerEfficiency="0.5" widthLocation="0"/>
    <Feed ImagGain="0" RealGain="1" effWidth="7" lengthLocation="0.145" nCosThetaResponsePower="1" noiseTemp="75" powerEfficiency="0.5" widthLocation="0"/>
    <Feed ImagGain="0" RealGain="1" effWidth="7" lengthLocation="0.29" nCosThetaResponsePower="1" noiseTemp="75" powerEfficiency="0.5" widthLocation="0"/>
    <Feed ImagGain="0" RealGain="1" effWidth="7" lengthLocation="0.435" nCosThetaResponsePower="1" noiseTemp="75" powerEfficiency="0.5" widthLocation="0"/>
    <Feed ImagGain="0" RealGain="1" effWidth="7" lengthLocation="0.58" nCosThetaResponsePower="1" noiseTemp="75" powerEfficiency="0.5" widthLocation="0"/>
    <Feed ImagGain="0" RealGain="1" effWidth="7" lengthLocation="0.725" nCosThetaResponsePower="1" noiseTemp="75" powerEfficiency="0.5" widthLocation="0"/>
    <Feed ImagGain="0" RealGain="1" effWidth="7" lengthLocation="0.87" nCosThetaResponsePower="1" noiseTemp="75" powerEfficiency="0.5" widthLocation="0"/>
    <Feed ImagGain="0" RealGain="1" effWidth="7" lengthLocation="1.015" nCosThetaResponsePower="1" noiseTemp="75" powerEfficiency="0.5" widthLocation="0"/>
    <Feed ImagGain="0" RealGain="1" effWidth="7" lengthLocation="1.16" nCosThetaResponsePower="1" noiseTemp="75" powerEfficiency="0.5" widthLocation="0"/>
    <Feed ImagGain="0" RealGain="1" effWidth="7" lengthLocation="1.305" nCosThetaResponsePower="1" noiseTemp="75" powerEfficiency="0.5" widthLocation="0"/>
    <Feed ImagGain="0" RealGain="1" effWidth="7" lengthLocation="1.45" nCosThetaResponsePower="1" noiseTemp="75" powerEfficiency="0.5" widthLocation="0"/>
    <Feed ImagGain="0" RealGain="1" effWidth="7" lengthLocation="1.595" nCosThetaResponsePower="1" noiseTemp="75" powerEfficiency="0.5" widthLocation="0"/>
    <Feed ImagGain="0" RealGain="1" effWidth="7" lengthLocation="1.74" nCosThetaResponsePower="1" noiseTemp="75" powerEfficiency="0.5" widthLocation="0"/>
    <Feed ImagGain="0" RealGain="1" effWidth="7" lengthLocation="1.885" nCosThetaResponsePower="1" noiseTemp="75" powerEfficiency="0.5" widthLocation="0"/>
    <Feed ImagGain="0" RealGain="1" effWidth="7" lengthLocation="2.03" nCosThetaResponsePower="1" noiseTemp="75" powerEfficiency="0.5" widthLocation="0"/>
    <Feed ImagGain="0" RealGain="1" effWidth="7" lengthLocation="2.175" nCosThetaResponsePower="1" noiseTemp="75" powerEfficiency="0.5" widthLocation="0"/>
    <Feed ImagGain="0" RealGain="1" effWidth="7" lengthLocation="2.32" nCosThetaResponsePower="1" noiseTemp="75" powerEfficiency="0.5" widthLocation="0"/>
    <Feed ImagGain="0" RealGain="1" effWidth="7" lengthLocation="2.465" nCosThetaResponsePower="1" noiseTemp="75" powerEfficiency="0.5" widthLocation="0"/>
    <Feed ImagGain="0" RealGain="1" effWidth="7" lengthLocation="2.61" nCosThetaResponsePower="1" noiseTemp="75" powerEfficiency="0.5" widthLocation="0"/>
    <Feed ImagGain="0" RealGain="1" effWidth="7" lengthLocation="2.755" nCosThetaResponsePower="1" noiseTemp="75" powerEfficiency="0.5" widthLocation="0"/>
    <Feed ImagGain="0" RealGain="1" effWidth="7" lengthLocation="2.9" nCosThetaResponsePower="1" noiseTemp="75" powerEfficiency="0.5" widthLocation="0"/>
    <Feed ImagGain="0" RealGain="1" effWidth="7" lengthLocation="3.045" nCosThetaResponsePower="1" noiseTemp="75" powerEfficiency="0.5" widthLocation="0"/>
    <Feed ImagGain="0" RealGain="1" effWidth="7" lengthLocation="3.19" nCosThetaResponsePower="1" noiseTemp="75" powerEfficiency="0.5" widthLocation="0"/>
    <Feed ImagGain="0" RealGain="1" effWidth="7" lengthLocation="3.335" nCosThetaResponsePower="1" noiseTemp="75" powerEfficiency="0.5" widthLocation="0"/>
    <Feed ImagGain="0" RealGain="1" effWidth="7" lengthLocation="3.48" nCosThetaResponsePower="1" noiseTemp="75" powerEfficiency="0.5" widthLocation="0"/>
    <Feed ImagGain="0" RealGain="1" effWidth="7" lengthLocation="3.625" nCosThetaResponsePower="1" noiseTemp="75" powerEfficiency="0.5" widthLocation="0"/>
    <Feed ImagGain="0" RealGain="1" effWidth="7" lengthLocation="3.77" nCosThetaResponsePower="1" noiseTemp="75" powerEfficiency="0.5" widthLocation="0"/>
    <Feed ImagGain="0" RealGain="1" effWidth="7" lengthLocation="3.915" nCosThetaResponsePower="1" noiseTemp="75" powerEfficiency="0.5" widthLocation="0"/>
    <Feed ImagGain="0" RealGain="1" effWidth="7" lengthLocation="4.06" nCosThetaResponsePower="1" noiseTemp="75" powerEfficiency="0.5" widthLocation="0"/>
    <Feed ImagGain="0" RealGain="1" effWidth="7" lengthLocation="4.205" nCosThetaResponsePower="1" noiseTemp="75" powerEfficiency="0.5" widthLocation="0"/>
    <Feed ImagGain="0" RealGain="1" effWidth="7" lengthLocation="4.35" nCosThetaResponsePower="1" noiseTemp="75" powerEfficiency="0.5" widthLocation="0"/>
    <Feed ImagGain="0" RealGain="1" effWidth="7" lengthLocation="4.495" nCosThetaResponsePower="1" noiseTemp="75" powerEfficiency="0.5" widthLocation="0"/>
</Cylinder>
```

Noiseless Pittsburgh Cylinder Visibility of a Point Source

C1 x C1*

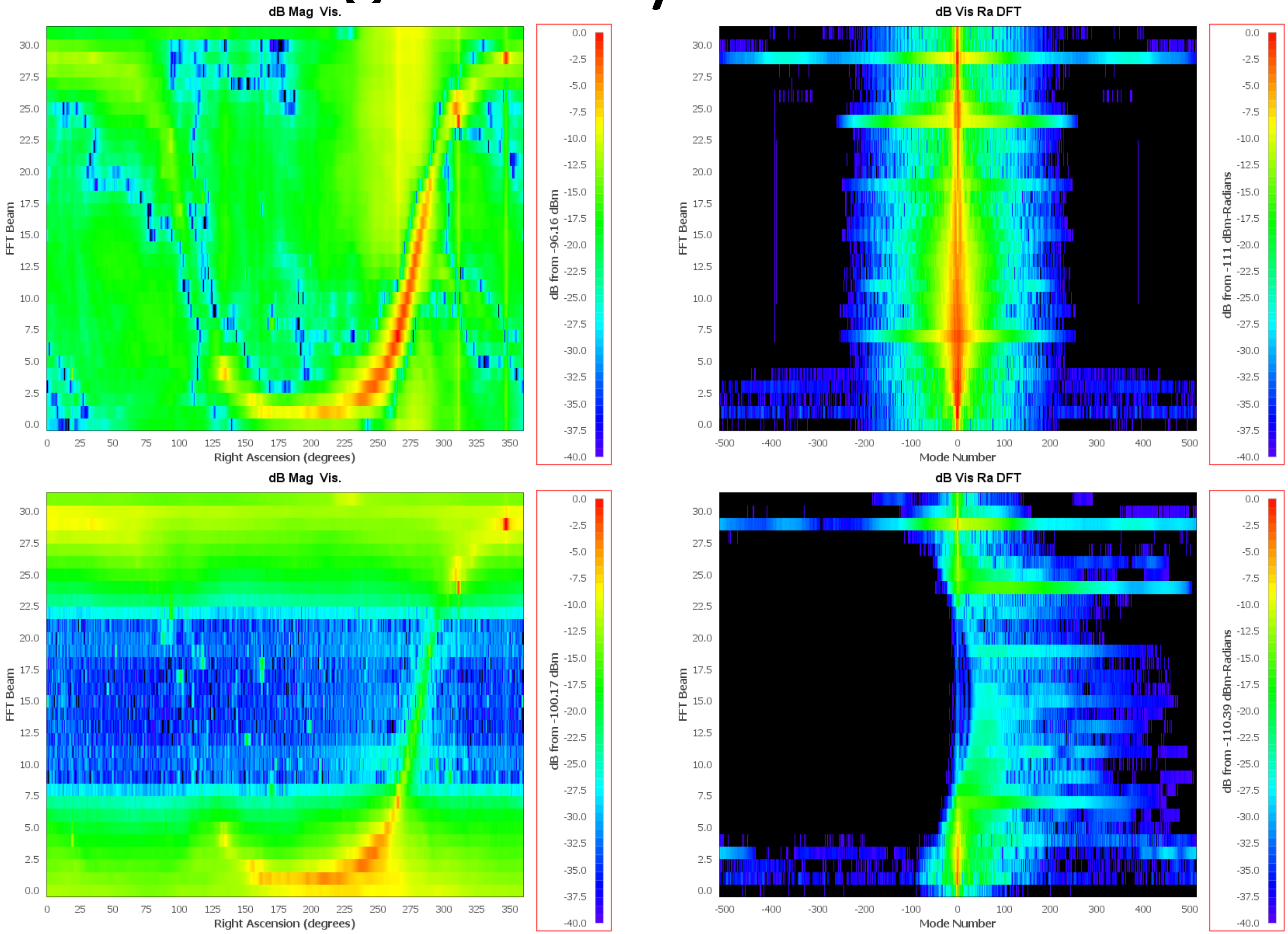
C1 x C2*



Noiseless Pittsburgh Cylinder Visibility of Angelica Sky at 1400MHz

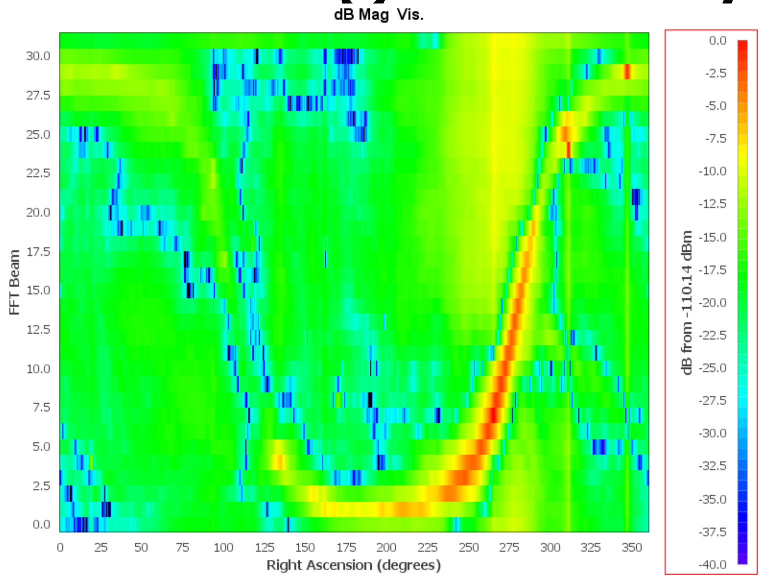
C1 x C1

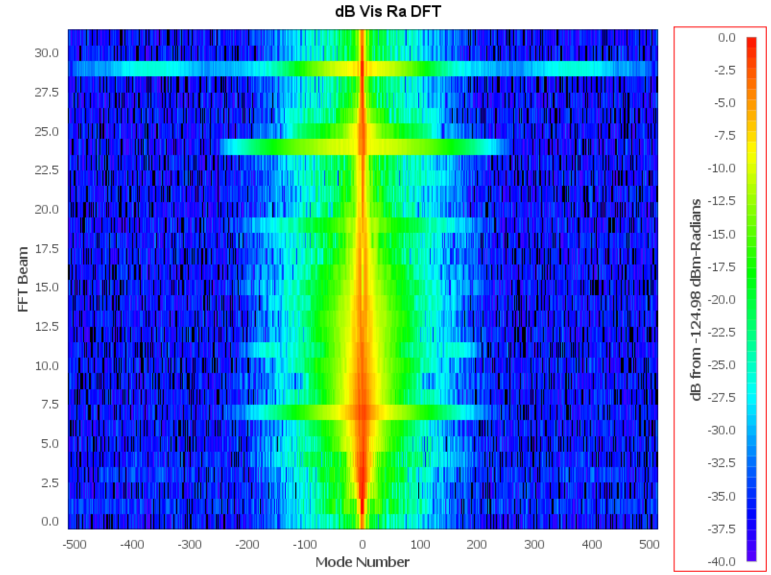
C1 x C2



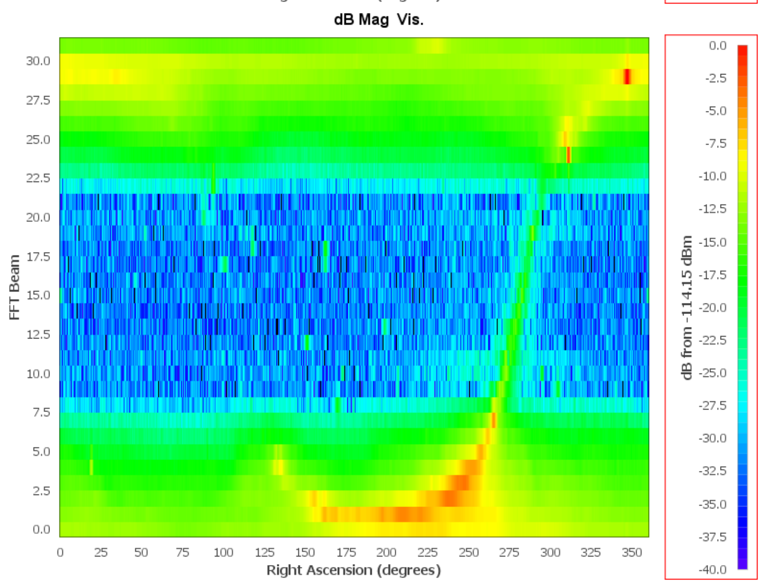
Noisy Pittsburgh Cylinder Visibility of Angelica Sky at 1400MHz

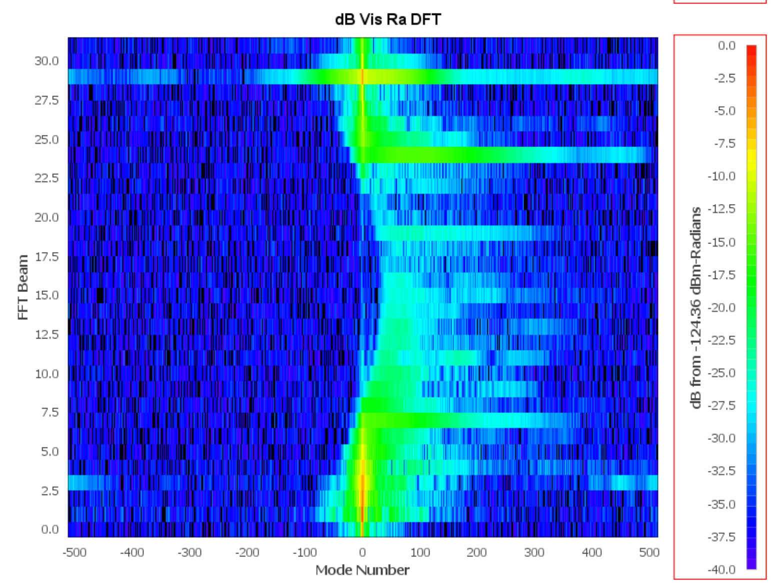
C1 x C1*





C1 x C2*



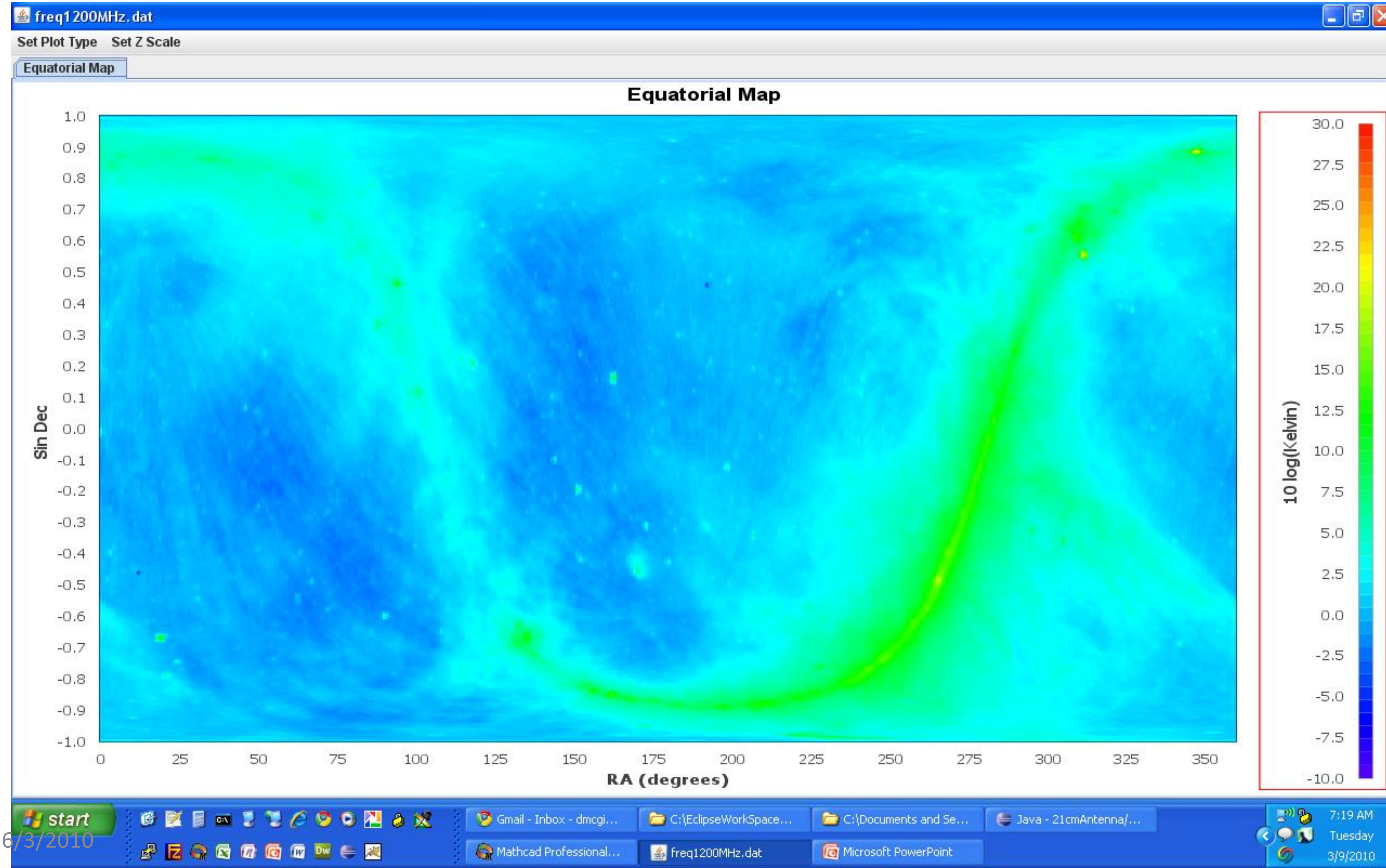


FORE-GROUND SUBTRACTION FLUCTUATING SKY PATCH

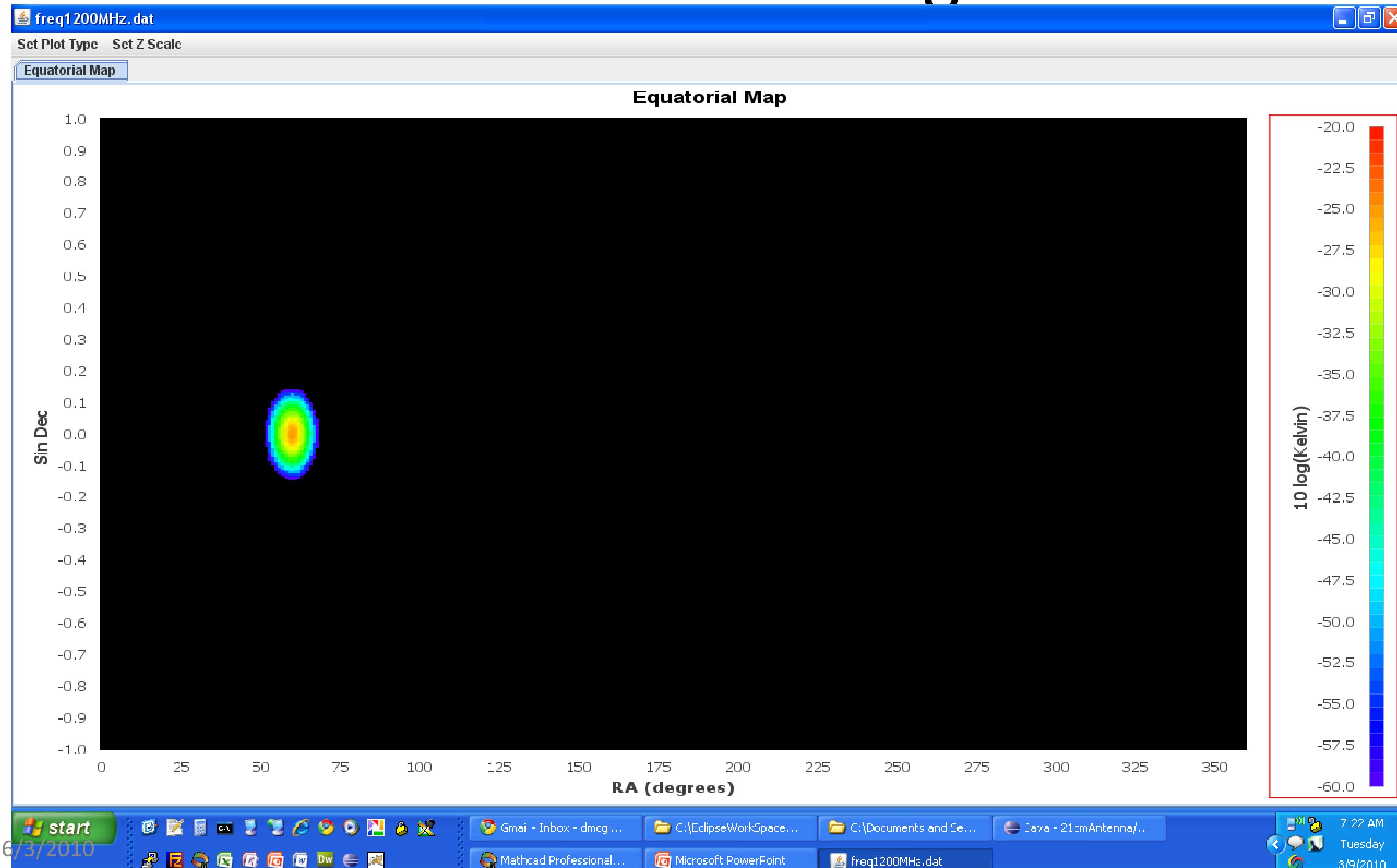
Sky Model Subtraction Algorithm

- Take cylinder visibility data and subtract a simulation of a frequency-smooth sky into a cylinder model
- From the sky difference map, fit each visibility spectrum “pixel” as a nth order polynomial in frequency
- Subtract the frequency-smoothed pixel trace from the difference map pixel by pixel
- Further FFT filter in frequency each the remaining pixel trace

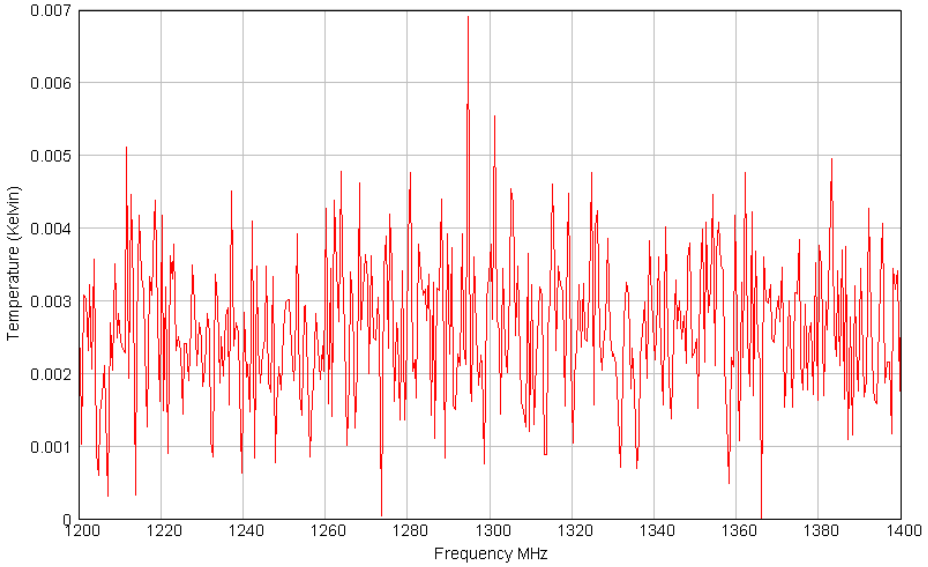
Angelica Sky Map

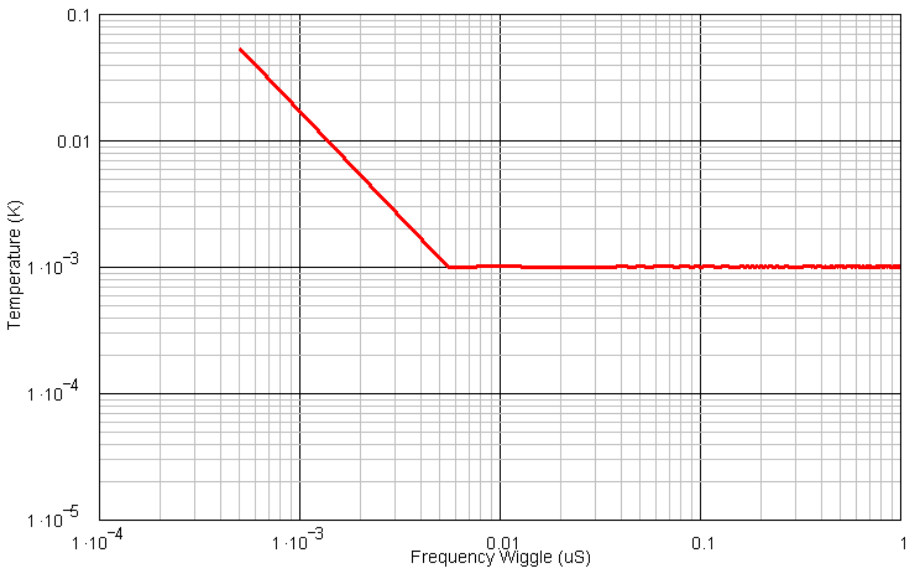


Freq. Fluctuation Patch
r.m.s radius = 3 degrees

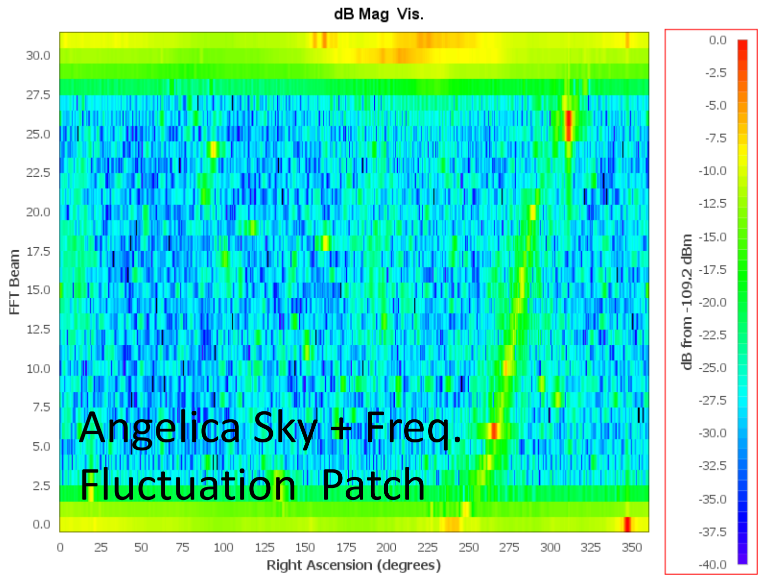


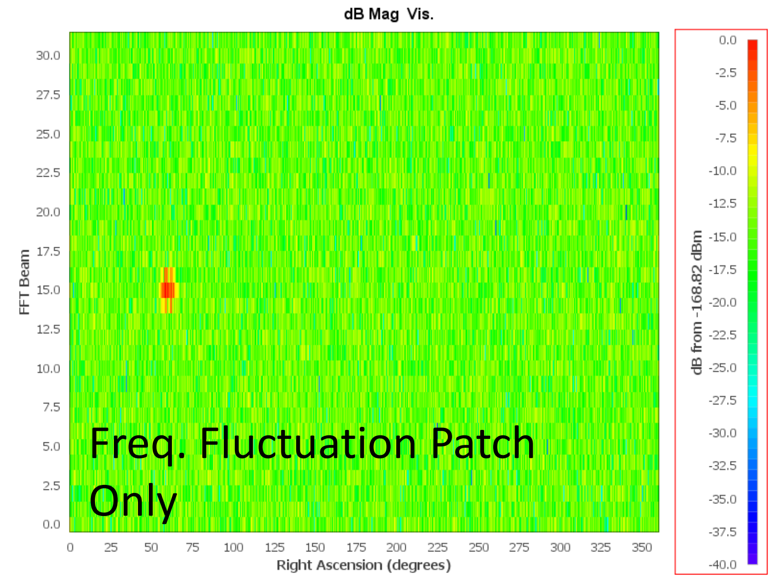
Freq. Fluctuation Patch Temperature vs Frequency

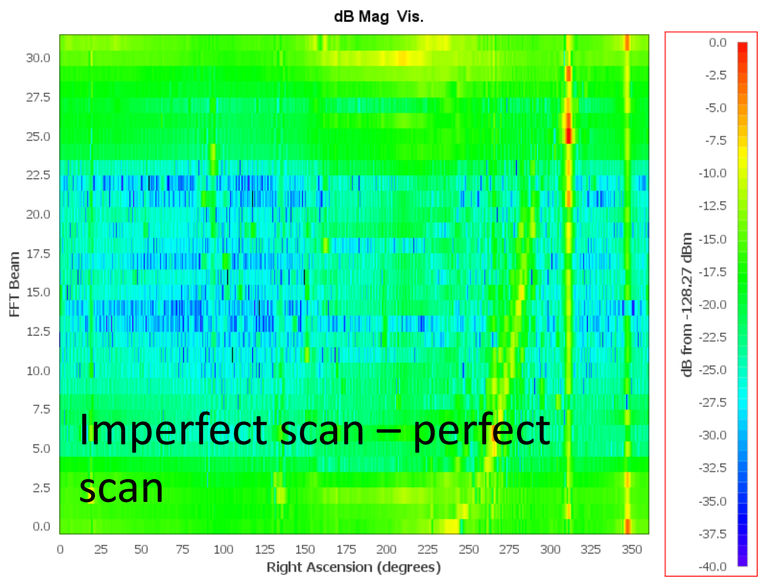


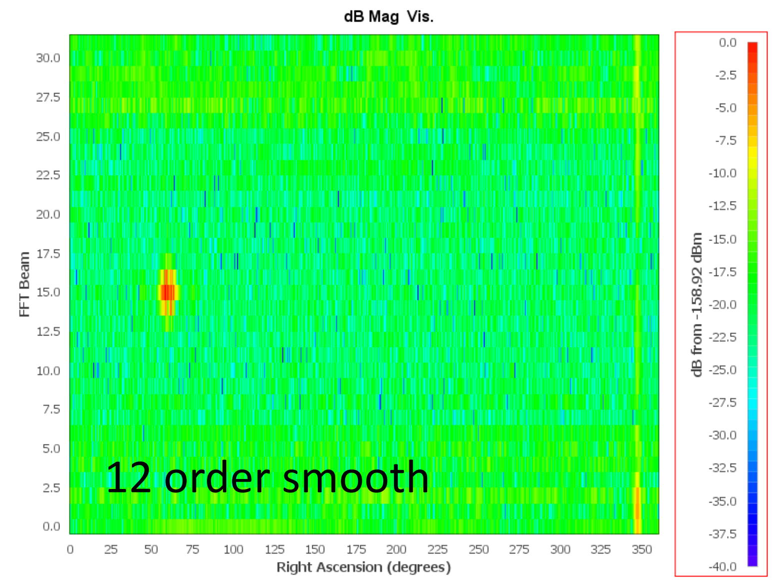


Pittsburgh Cylinder Simulations Sky Scan



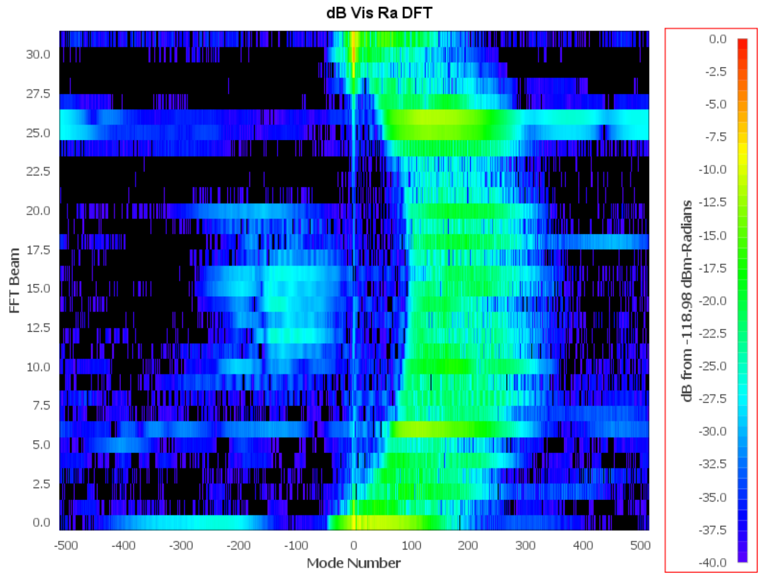


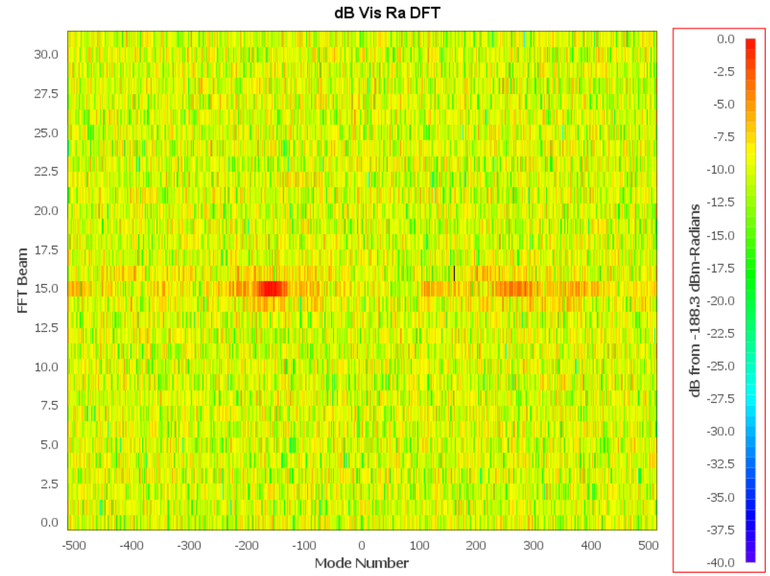




RA DFT of Pittsburgh Cylinder Simulations

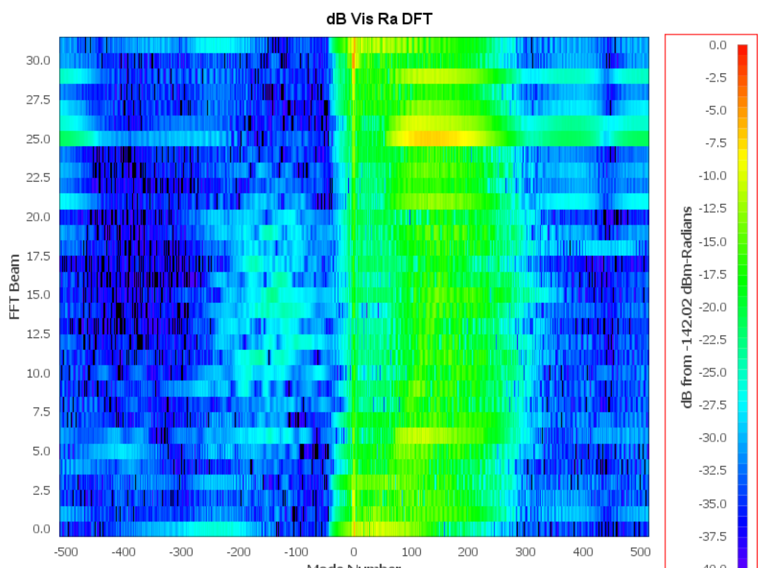
Angelica Sky + Freq. Fluctuation Patch

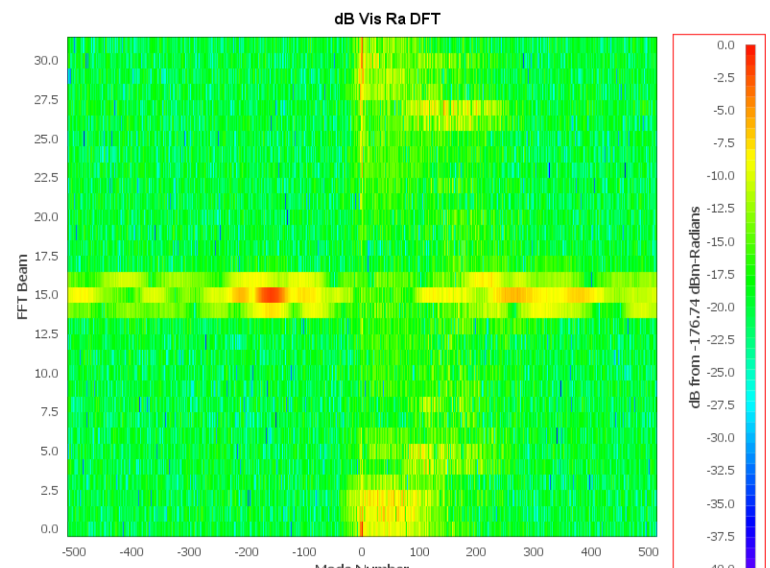




Freq. Fluctuation Patch Only

Imperfect scan – perfect scan



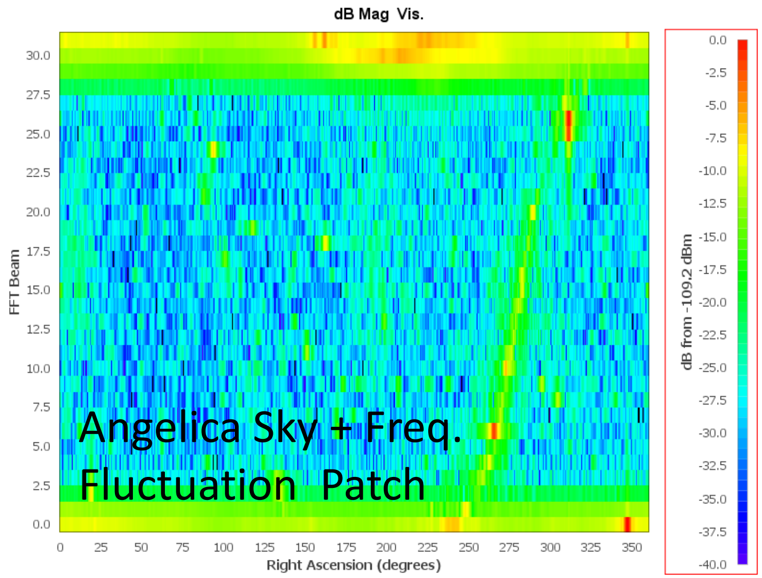


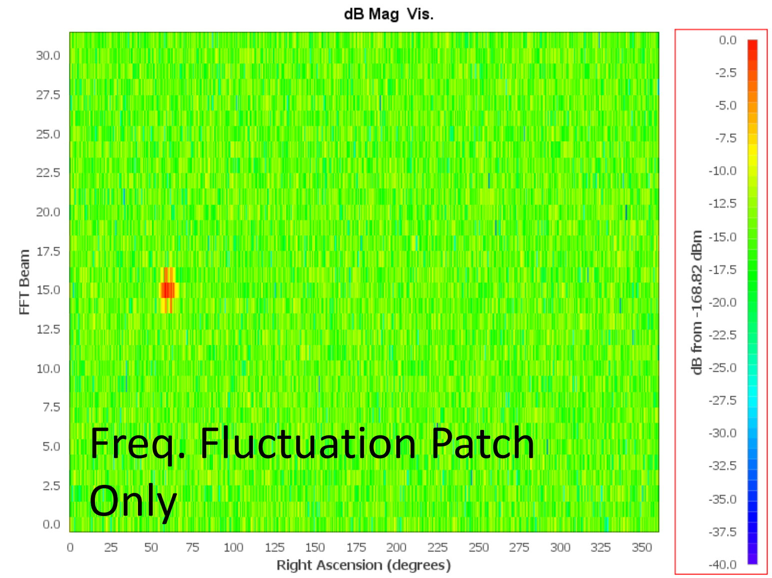
12 order smooth

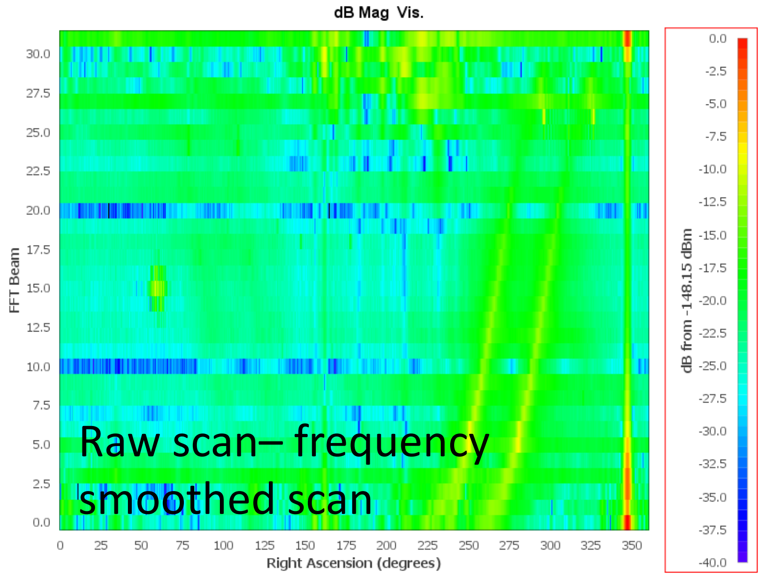
Frequency-Smoothed Sky Subtraction Algorithm

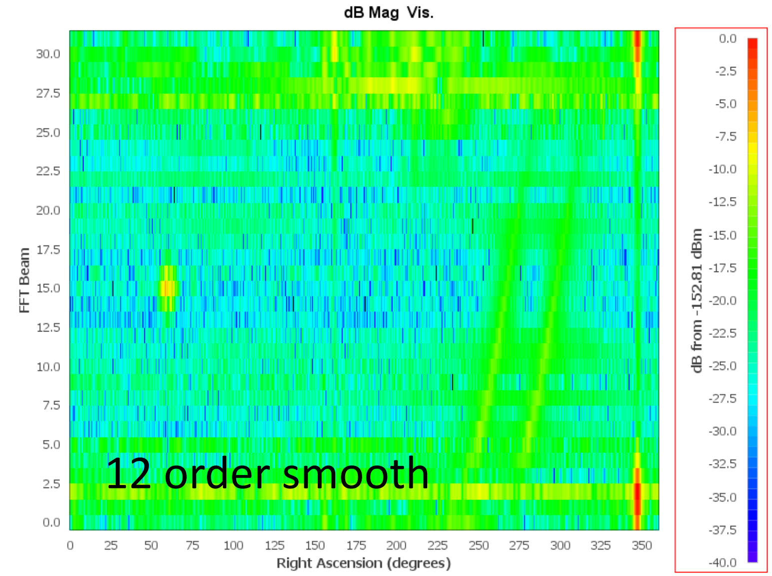
- Take cylinder visibility data smooth it along the frequency axis using a N order polynomial for each pixel
- Subtract the frequency-smoothed map from the raw map producing a difference map
- From the difference map, fit each visibility spectrum "pixel" as a nth order polynomial in frequency
- Subtract the frequency-smoothed pixel trace from the difference map pixel by pixel
- Further FFT filter in frequency each the remaining pixel trace

Fluctuating Sky Path Simulations



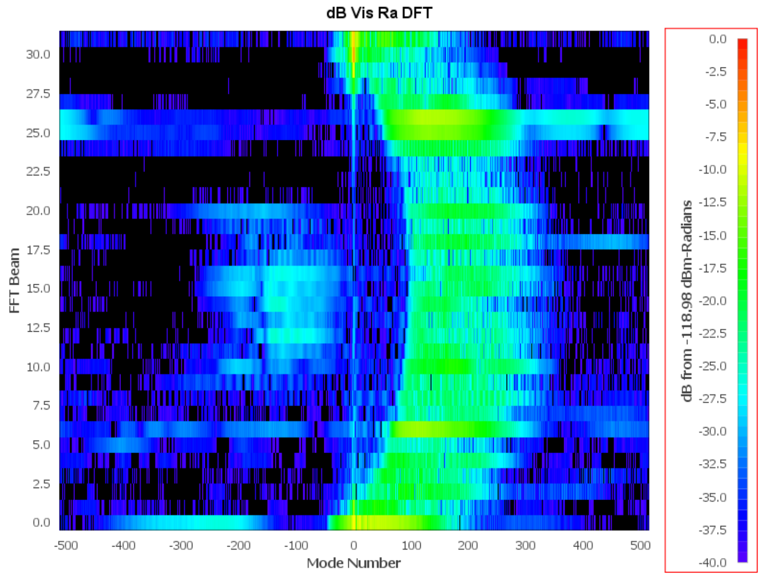


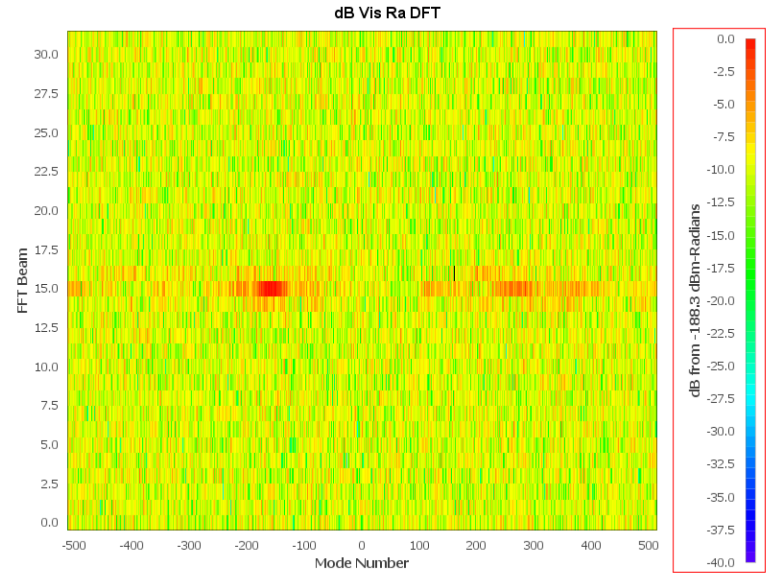




RA DFT of Pittsburgh Cylinder Simulations

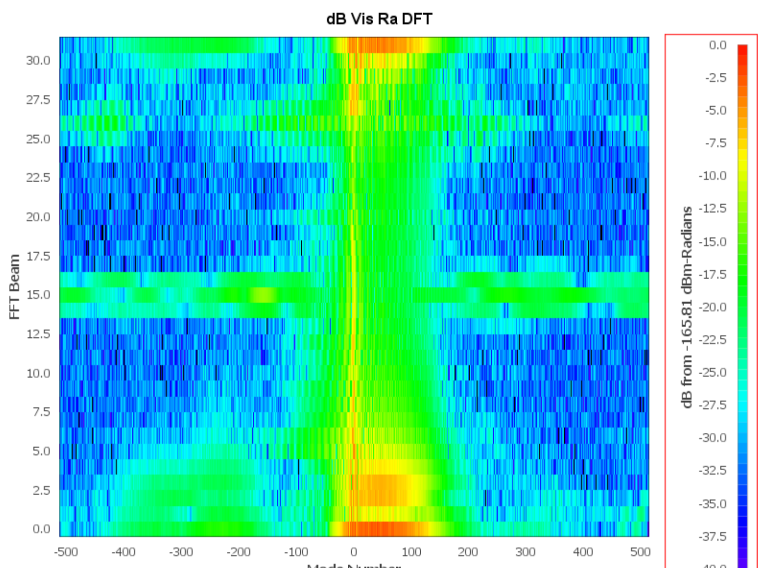
Angelica Sky + Freq. Fluctuation Patch

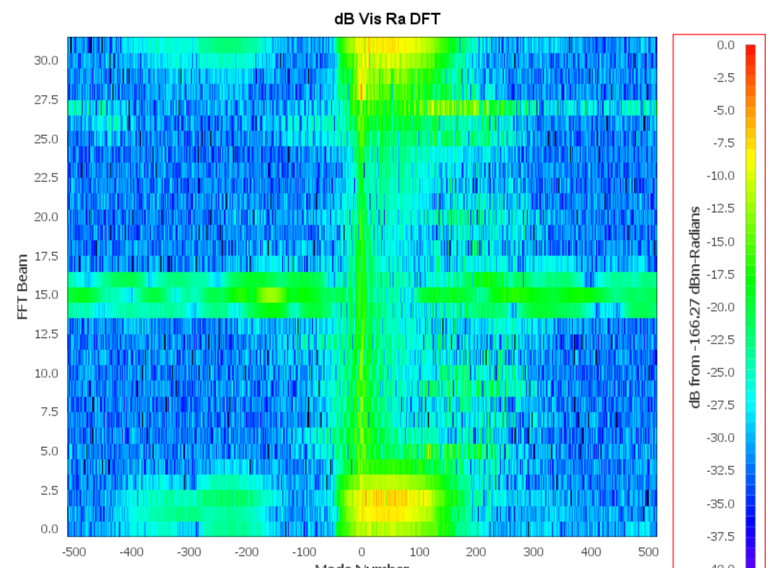




Freq. Fluctuation Patch Only

Raw scan – smoothed scan

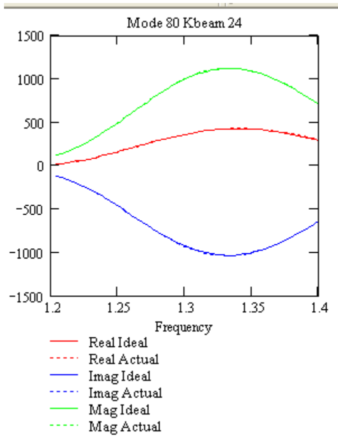


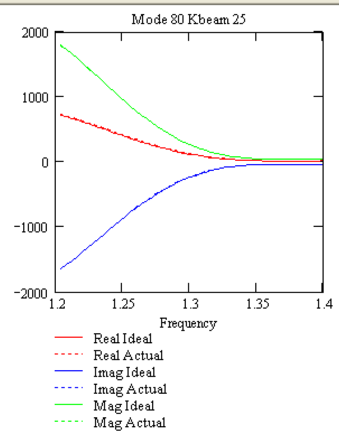


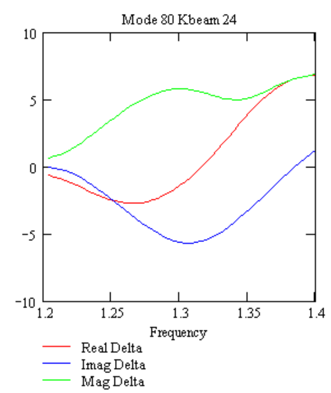
12 order smooth

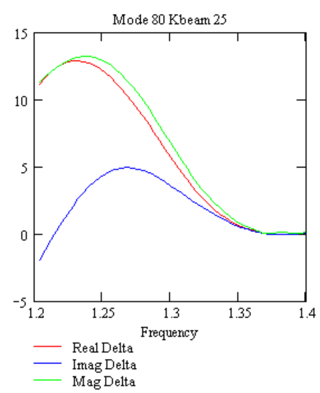
Mode Mixing Smoothness

“Hot Pixel” track before Sky Subtraction

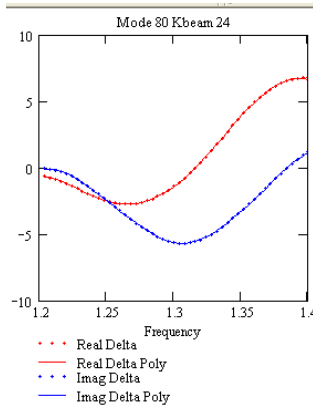


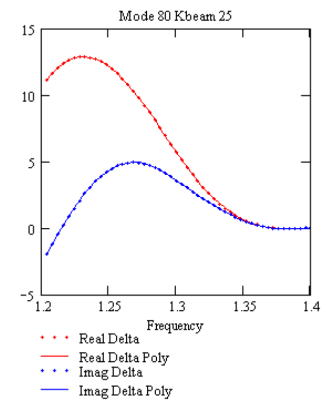


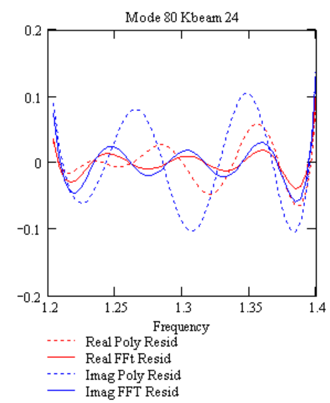


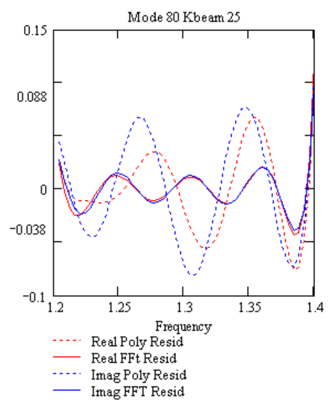


“Hot Pixel” track after Sky Subtraction









CYLINDER VISIBILITY MODELER AND K-SPACE SKY RECONSTRUCTION

Cylinder Modeler

Sky Reconstruction from Cylinder Visibilities

Dave McGinnis

June 1, 2010

Visibility

This note will consider the reconstruction of the sky from the measured visibilities from a pair of cylinder antenna arrays. It is assumed that the cylinders fixed and are oriented along the meridian. Each cylinder is populated with N feeds spaced uniformly along the length. The output voltage of each feed provides an input of a spatial Fourier transform along the cylinder length. The spatial Fourier transform forms N beams along the length of the cylinder.

For a pair of cylinders the visibility between cylinders is formed for each beam. As the sky drifts through the cylinder beam, the visibility for beam k is:

$$v_k(\varphi) = \int_{-\pi}^{\pi}\int_{-\pi}^{\pi} \frac{\tilde{A}_k(\theta,\phi)}{\lambda^2} T(\theta, \varphi - \phi) cos(\theta)\, d\theta d\phi \quad (1)$$

Where $\boldsymbol{\varphi}$ is the time of the day (in units of angle), $\boldsymbol{\lambda}$ is the wavelength and $\mathbf{T}$ is the power flux of the sky. The cylinder pair Fourier area is defined as

$$\tilde{A}_k(\theta,\phi) = \tilde{a}_{k,c1}(\theta,\phi)\left(\tilde{a}_{k,c2}(\theta,\phi)\right)^* \quad (2)$$

where the subscripts $\mathbf{c1}$, $\mathbf{c2}$ indicate cylinder 1 and cylinder 2, respectively. The Fourier root area of a cylinder is defined as

$$\tilde{a}_{k,c}(\theta,\phi) = \sum_n a_n(\theta,\phi) e^{-j\vec{\beta}(\theta,\phi)\cdot\vec{r}_{n,c}} e^{j2\pi k\frac{n}{N}} \quad (3)$$

Where $\mathbf{n}$ is the feed number, $\mathbf{r}_{\mathbf{n,c}}$ is the global location of the feed and $\boldsymbol{\beta}$ is the incoming wave vector:

$$\vec{\beta}(\theta,\phi) = \frac{2\pi}{\lambda}(sin(\theta)\hat{x} + cos(\theta)sin(\phi)\hat{y}) \quad (4)$$

It is assumed that the length of the cylinders is in the x direction.

Sky Expansion

Since the sky is periodic, it can be expanded in a Fourier series:

$$T_c(\theta,\phi) = \sum_l \sum_m \hat{T}_{m,l} e^{jl\pi sin(\theta)} e^{jm\phi} \quad (5)$$

Since T_c is a complex function but the sky temperature must be a real function, the sky temperature can be written as:

$$T(\theta,\phi) = T_c(\theta,\phi) + \left(T_c(\theta,\phi)\right)^* \quad (6)$$

$$T(\theta,\phi) = \sum_l \sum_m \hat{T}_{m,l} e^{jl\pi sin(\theta)} e^{jm\phi} + \sum_l \sum_m \left(\hat{T}_{m,l}\right)^* e^{-jl\pi sin(\theta)} e^{-jm\phi} \quad (7)$$

$$T(\theta,\phi) = \sum_l \sum_m \left(\hat{T}_{m,l} + \left(\hat{T}_{-m,-l}\right)^*\right) e^{jl\pi sin(\theta)} e^{jm\phi} \quad (8)$$

Substituting Equation 8 into Equation 1,

$$v_k(\varphi) = \sum_l \sum_m \left(\hat{T}_{m,l} + \left(\hat{T}_{-m,-l}\right)^*\right) e^{jm\varphi} \int_{-\pi}^{\pi}\int_{-\pi}^{\pi} \frac{\tilde{A}_k(\theta,\phi)}{\lambda^2} e^{jl\pi sin(\theta)} e^{-jm\phi} cos(\theta)\, d\theta d\phi \quad (9)$$

Fourier Transform of Cylinder Visibilities

Now assume that the visibility is measured at N discrete times during the day. The visibility will be periodic with a period of a day so we can expand the visibility into a Fourier series.

$$v_k(\varphi_n) = \sum_{m'} \tilde{V}_{k,m'} e^{jm\varphi_n} \quad (10)$$

where

$$\tilde{V}_{k,m'} = \frac{1}{N}\sum_n v_k(\varphi_n)\, e^{-jm\varphi_n} \quad (11)$$

Substituting Equations 9 into Equations 11,

$$\tilde{V}_{k,m} = \sum_l \hat{A}_{k,l,m}\left(\hat{T}_{m,l} + \left(\hat{T}_{-m,-l}\right)^*\right) \quad (12)$$

where:

$$\hat{A}_{k,l,m} = \int_{-\pi}^{\pi}\int_{-\pi}^{\pi} \frac{\tilde{A}_k(\theta,\phi)}{\lambda^2} e^{jl\pi sin(\theta)} e^{-jm\phi} cos(\theta)\, d\theta d\phi \quad (13)$$

The discrete approximation for Equation 13 is:

$$\hat{A}_{k,l,m} \approx \frac{4\pi}{N_p N_q} \sum_{q=0}^{N_q} \sum_{p=0}^{N_p} \frac{\tilde{A}_k(\theta_p,\phi_q)}{\lambda^2} e^{jl(\pi sin(\theta))_p} e^{-jm\phi_q} \quad (14)$$

The matrix form of Equation 13 is:

$$[\tilde{V}_m] = [\hat{A}_m][\tilde{T}_m] \quad (15)$$

Multiple Visibilities

There can be a number of combination of cylinder visibilities that have a non-zero component a given right ascension mode m. However, the sky mode temperature at mode m must be unique.

$$[\tilde{V}_m]^{(1)} = [\hat{A}_m]^{(1)}[\tilde{T}_m] \quad (16)$$

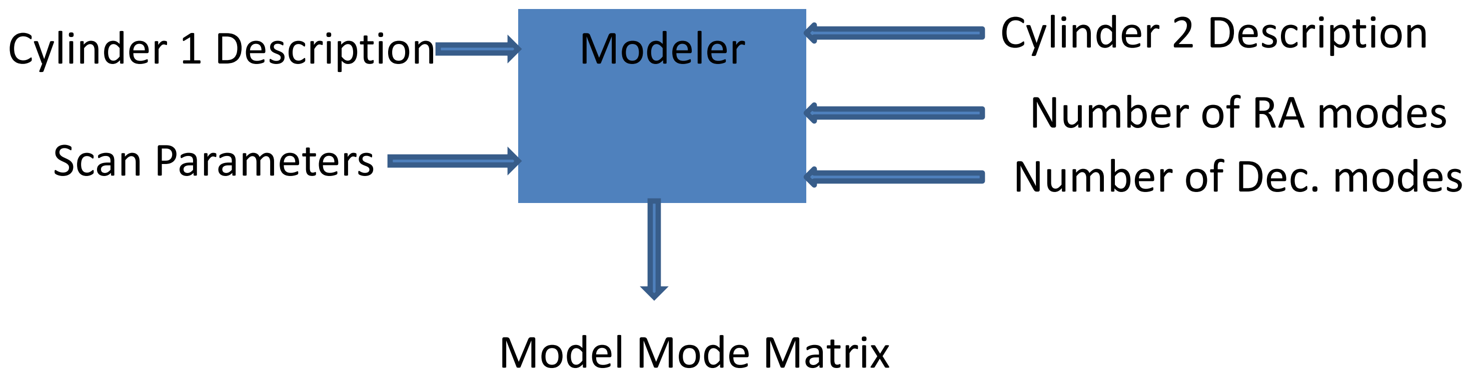
$$[\tilde{V}_m]^{(2)} = [\hat{A}_m]^{(2)}[\tilde{T}_m] \quad (17)$$

To satisfy both equations in a least squares fit, equations 16 and 17 can be combined:

$$\left\{\left\{[\hat{A}_m]^{(1)}\right\}^T\right\}^* [\tilde{V}_m]^{(1)} + \left\{\left\{[\hat{A}_m]^{(2)}\right\}^T\right\}^* [\tilde{V}_m]^{(2)} = \left(\left\{\left\{[\hat{A}_m]^{(1)}\right\}^T\right\}^* + \left\{\left\{[\hat{A}_m]^{(2)}\right\}^T\right\}^*\right)[\tilde{T}_m] \quad (18)$$

http://projects-docdb.fnal.gov/cgi-bin/ShowDocument?docid=838

Cylinder Visibility Modeler



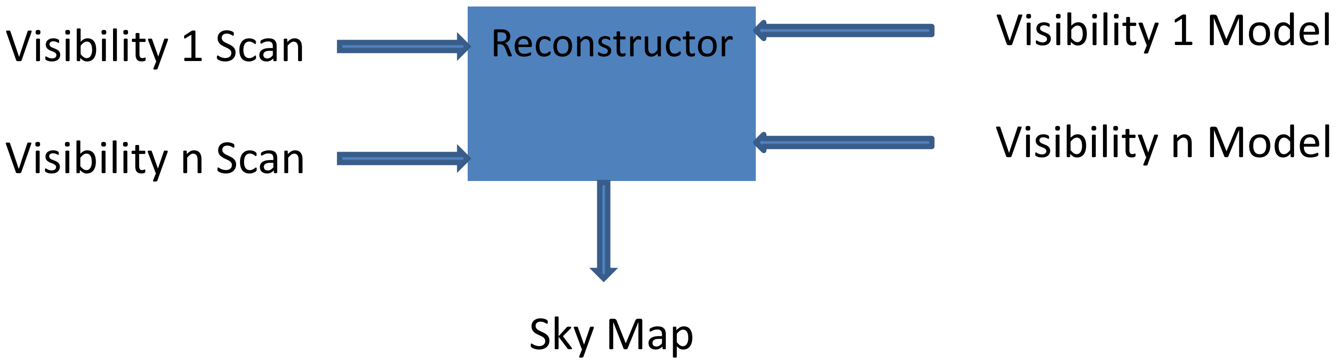
$$\tilde{V}_{k,m'} = \frac{1}{N}\sum_{n} v_k(\varphi_n)\ e^{-jm\varphi_n}$$

$$T(\theta,\phi) = \sum_{l}\sum_{m}\left(\hat{T}_{m,l} + \left(\hat{T}_{-m,-l}\right)^*\right)e^{jl\pi sin(\theta)}e^{jm\phi}$$

$$\hat{A}_{k,l,m} = \int_{-\pi}^{\pi}\int_{-\pi}^{\pi}\frac{\tilde{A}_k(\theta,\phi)}{\lambda^2}e^{jl\pi sin(\theta)}e^{-jm\phi}cos(\theta)\,d\theta d\phi$$

$$\tilde{V}_{k,m} = \sum_{l}\hat{A}_{k,l,m}\left(\hat{T}_{m,l} + \left(\hat{T}_{-m,-l}\right)^*\right)$$

Sky Reconstruction



$$[\tilde{V}_m]^{(1)} = [\hat{A}_m]^{(1)}[\tilde{T}_m]$$

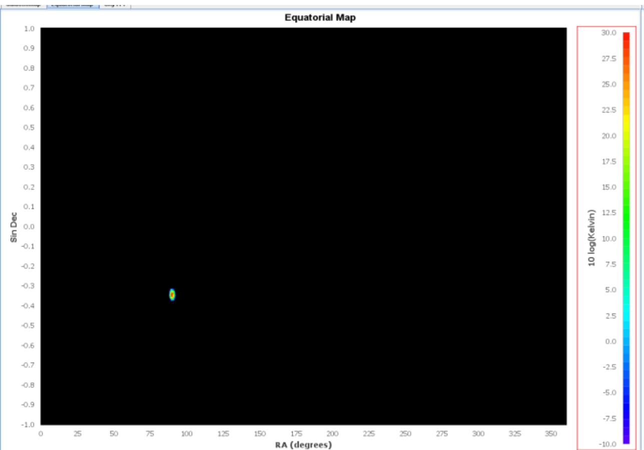
$$[\tilde{V}_m]^{(2)} = [\hat{A}_m]^{(2)}[\tilde{T}_m]$$

$$\left\{\left\{[\hat{A}_m]^{(1)}\right\}^T\right\}^* [\tilde{V}_m]^{(1)} + \left\{\left\{[\hat{A}_m]^{(2)}\right\}^T\right\}^* [\tilde{V}_m]^{(2)} = \left(\left\{\left\{[\hat{A}_m]^{(1)}\right\}^T\right\}^* + \left\{\left\{[\hat{A}_m]^{(2)}\right\}^T\right\}^*\right)[\tilde{T}_m]$$

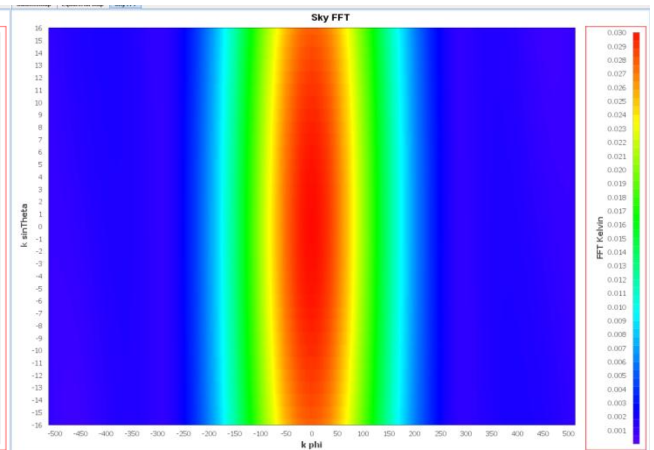
Point Source Sky Map Reconstruction

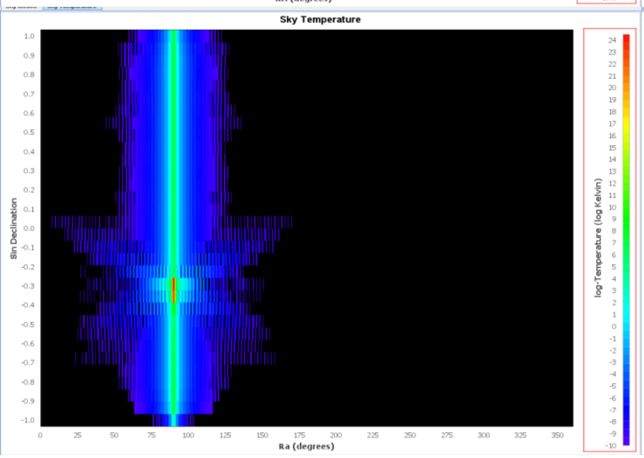
(Pittsburgh Cylinders)

Input Sky

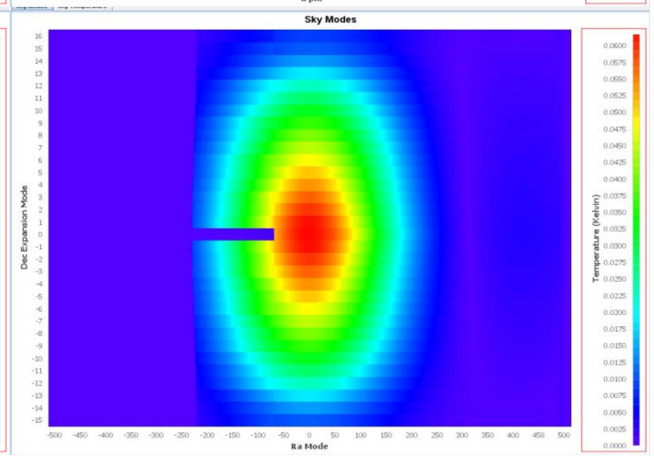


Input K-Space

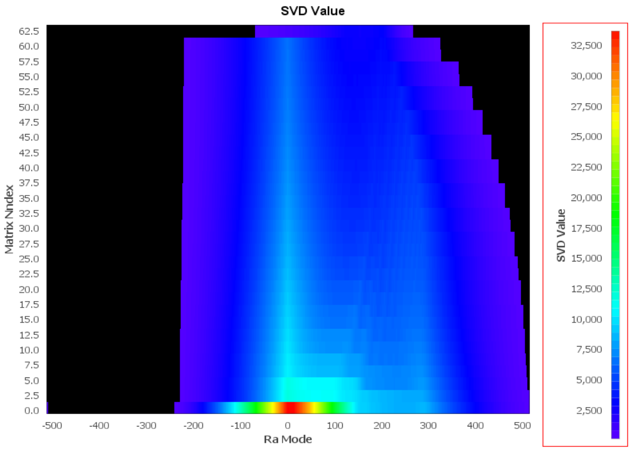




Output Sky



Output K-Space

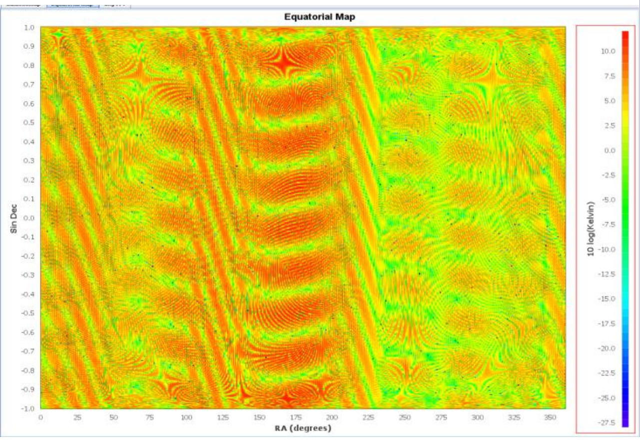


SVD Values

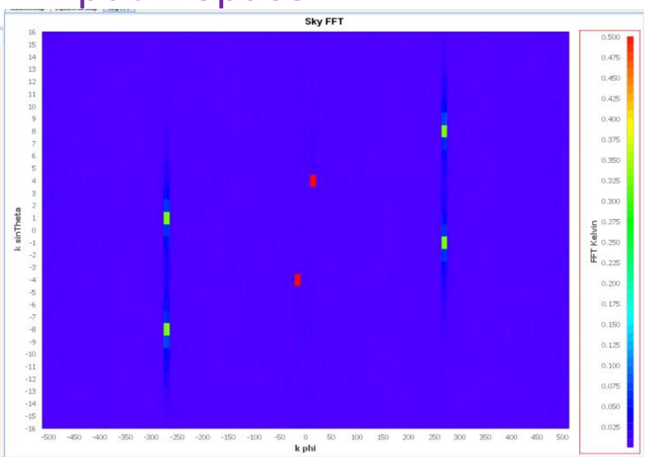
K-Wave Sky Map Reconstruction

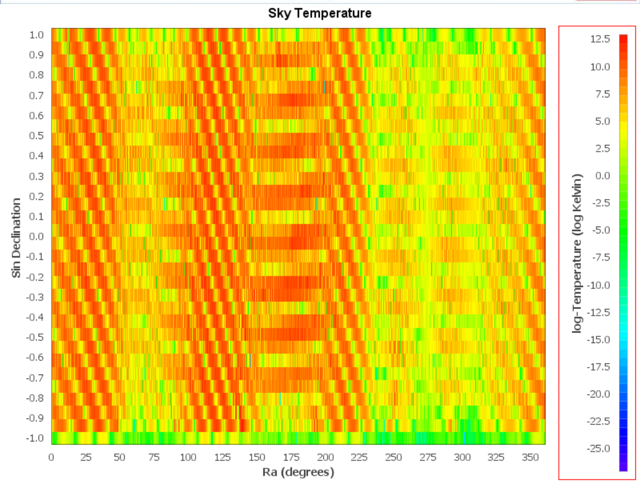
(Pittsburgh Cylinders)

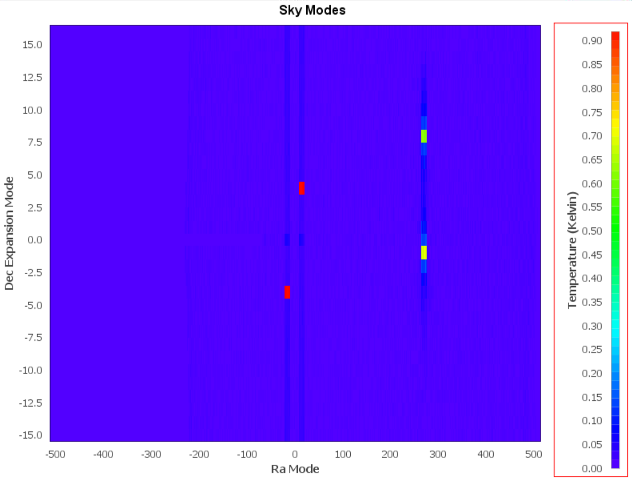
Input Sky

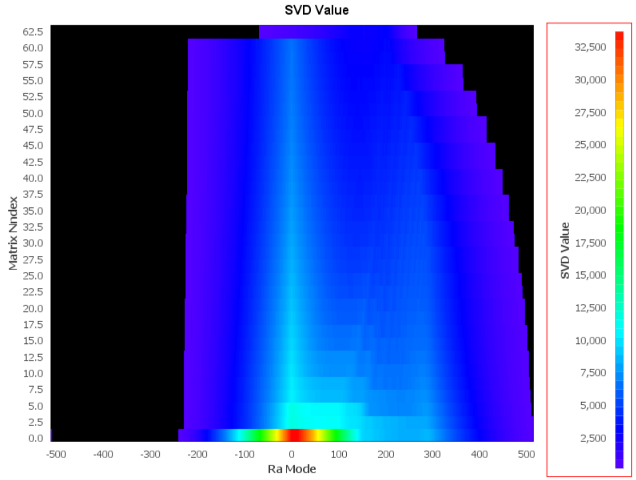


Input K-Space









Output Sky

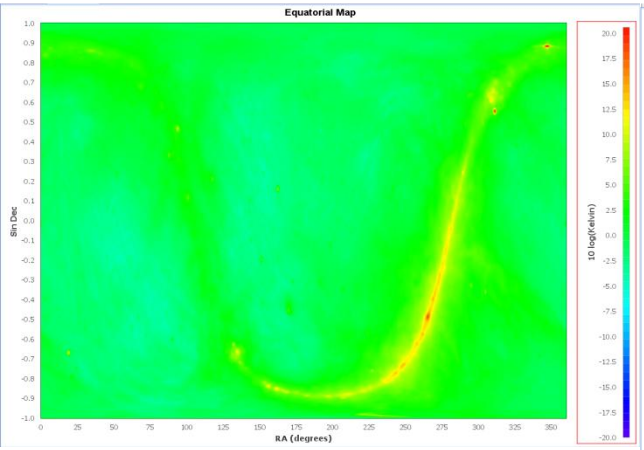
Output K-Space

SVD Values

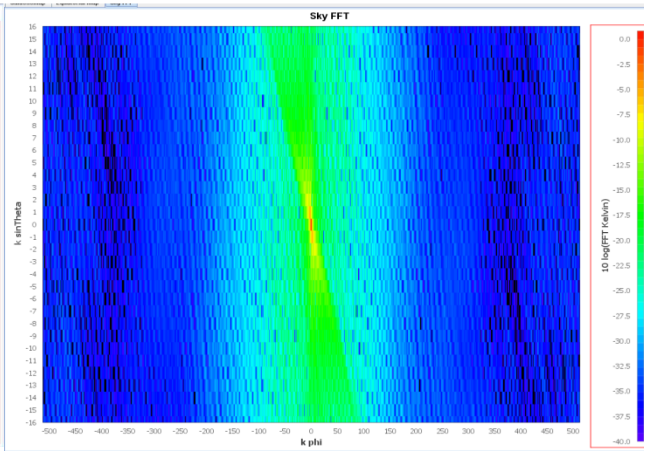
Angelica Sky Map Reconstruction

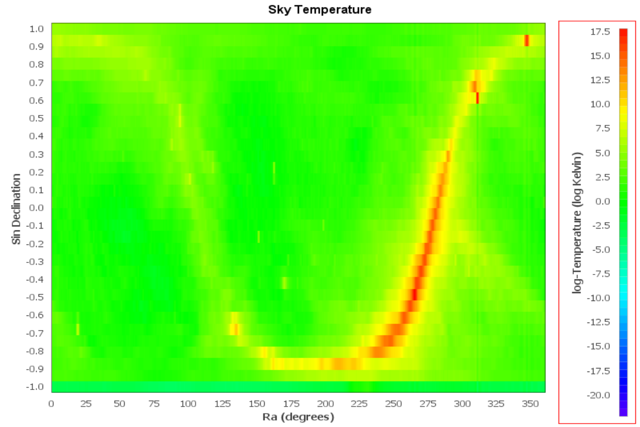
(Pittsburgh Cylinders)

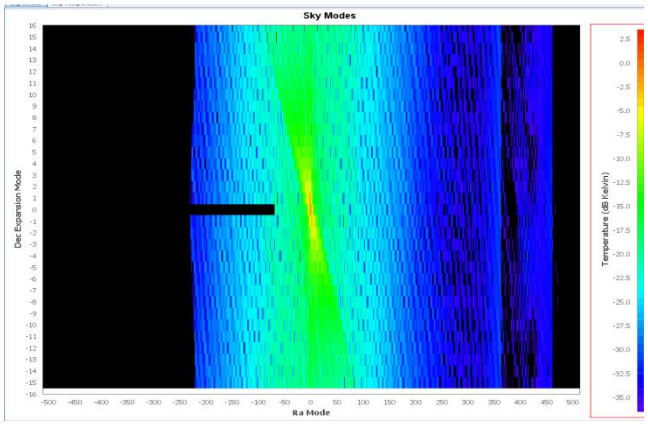
Input Sky

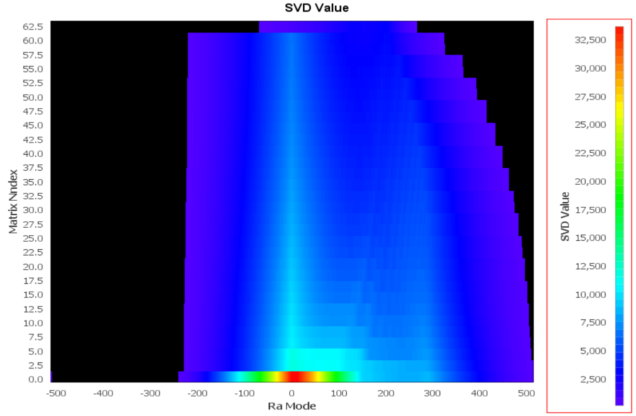


Input K-Space





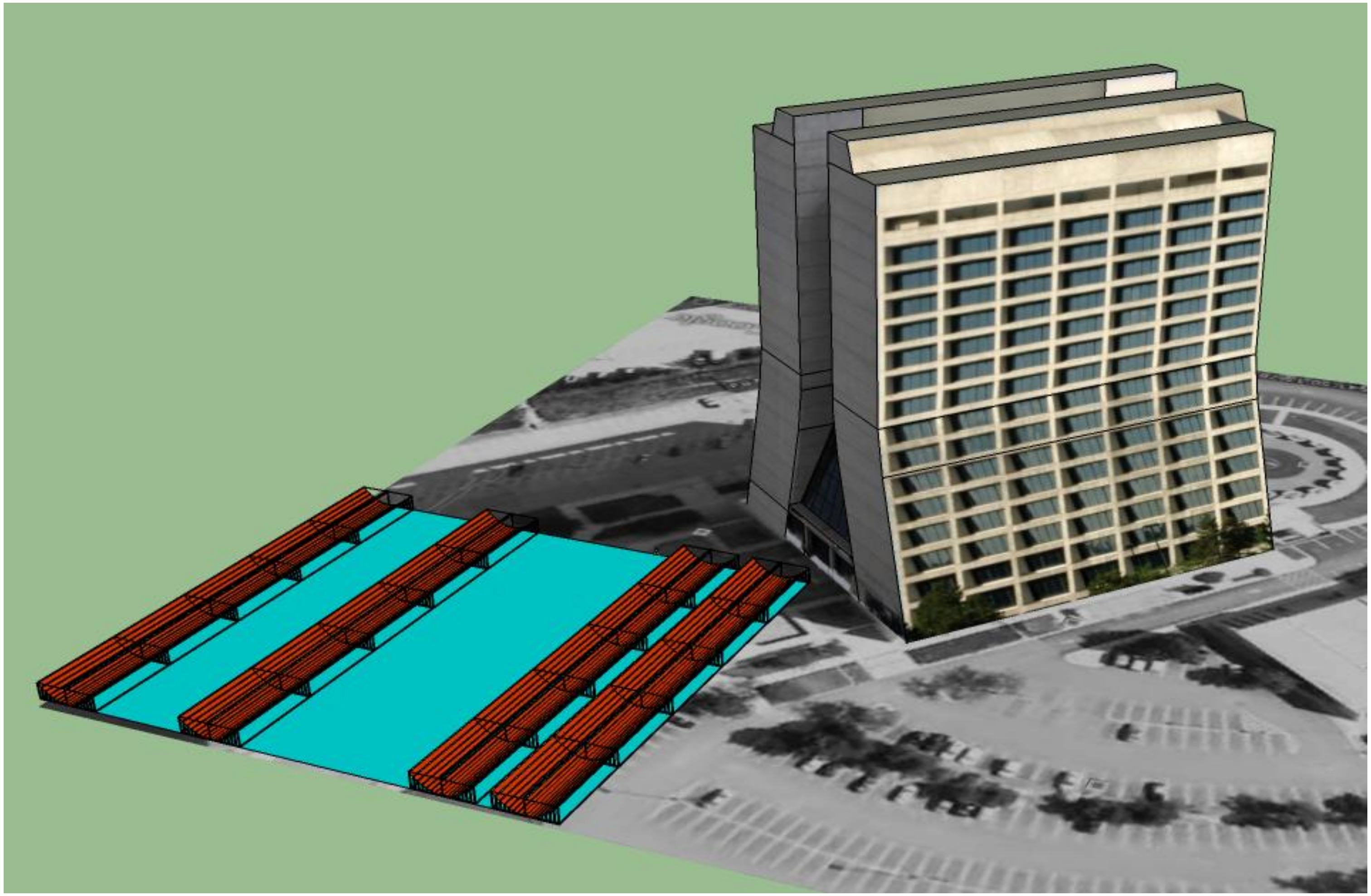




Output Sky

Output K-Space

SVD Values



NEXT STEP – FULL SCALE TELESCOPE SIMULATION AND RECONSTRUCTION

Foreground Removal using BAO Simulations

- We need ~ 128 feeds/cylinder to resolve the first BAO peak
- The simulation size increases ~ cubically with the number of feeds
  - 256 feeds and 4 cylinders requires ~500 times more computing than 32 feeds and 2 cylinders
  - In addition, the simulation has to be done over a number of frequency slices (>100 slices)
    - To simulate a telescope with 256 feeds/cylinder and 4 cylinders (7 cylinder combinations) at 100 frequency slices requires 1.8 years of cpu time.
    - To reconstruct the telescope, , the telescope model matrices requires a total of 1.500 Terabytes of disk space.

FNAL-KICP Joint Cluster

- Since the last review, we have moved the simulation software to the FNAL-KICP cluster which has >1200 cpu's available.
  - Using 112 cpu's, it takes about 140 hours to simulate the 256 feed telescope.
    - This has been done.
  - We are currently having issues with the large storage space required for the telescope model matrices.
    - Space is not an issue. There is 97 Terabytes available with 12 Terabytes available to a single user.
    - Accessing the data rapidly is the issue. – it can be solved.

FNAL-KICP Joint Cluster

Fermi National Accelerator Laboratory

FNAL - KICP Joint Cluster

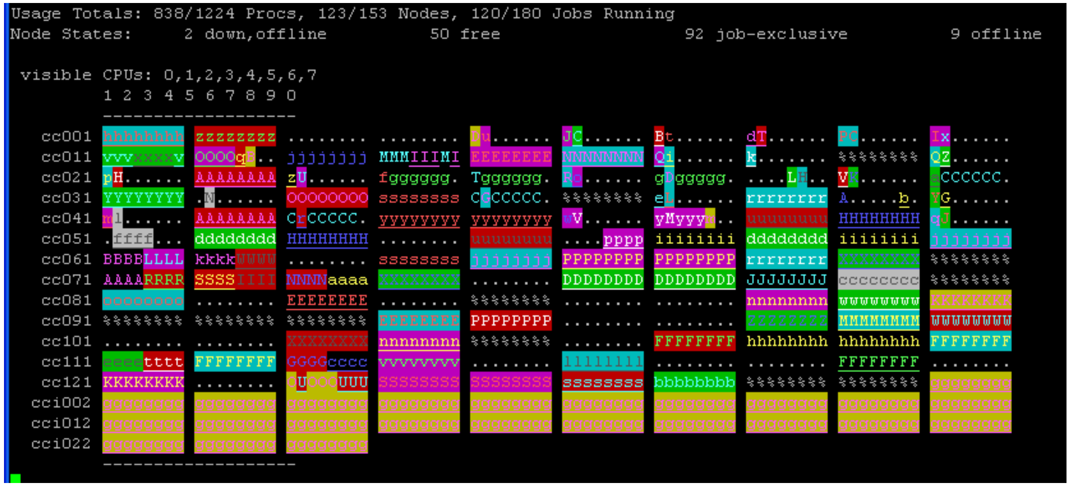
Hardware Details

Component	Properties	Total
Master node	8-way, 16GB, fulla	**2**
Quad-socket dual-core nodes	8-way, 128GB, cc001	**1**
Quad-socket dual-core nodes	8-way, 32GB, cc002-004	**3**
Dual-socket quad-core nodes	8-way, 16GB, cc005-...	**149**
Total # of cores		**1240**
Infiniband nodes		**24**

- Quad-socket nodes contain 2.2GHz AMD Opteron 8214 processors; they are good for shared memory parallel jobs or for jobs that require very large amount of memory per node.
- Dual-socket nodes contain 2.0GHz AMD Opetron 2350 processors; they are suitable for all kinds of jobs (serial, embarrasingly parallel, shared or distributed memory). These nodes are available under the PBS queue normal.

Operating System

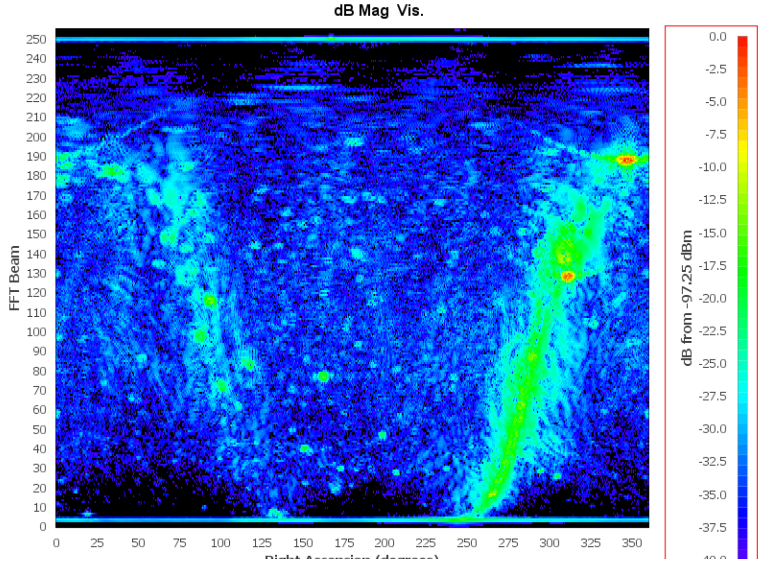
Scientific Linux, 64 bit

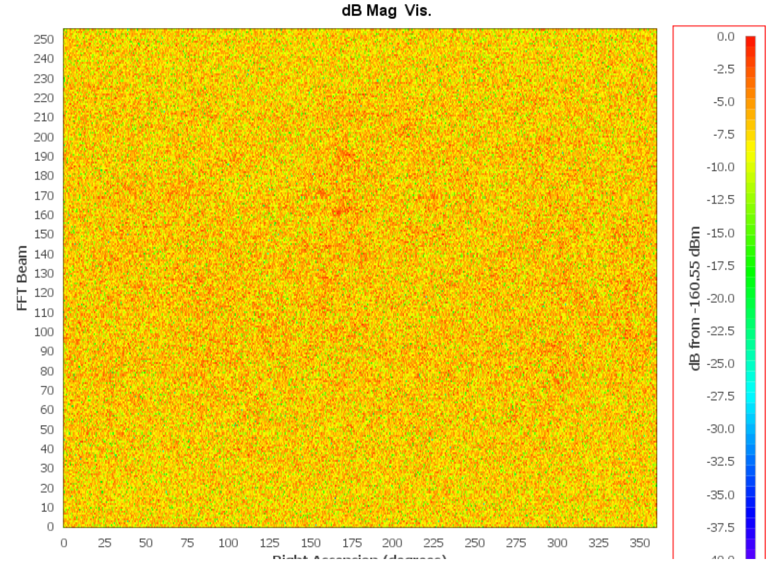


C1-C7 "Visibility" for 256 feeds

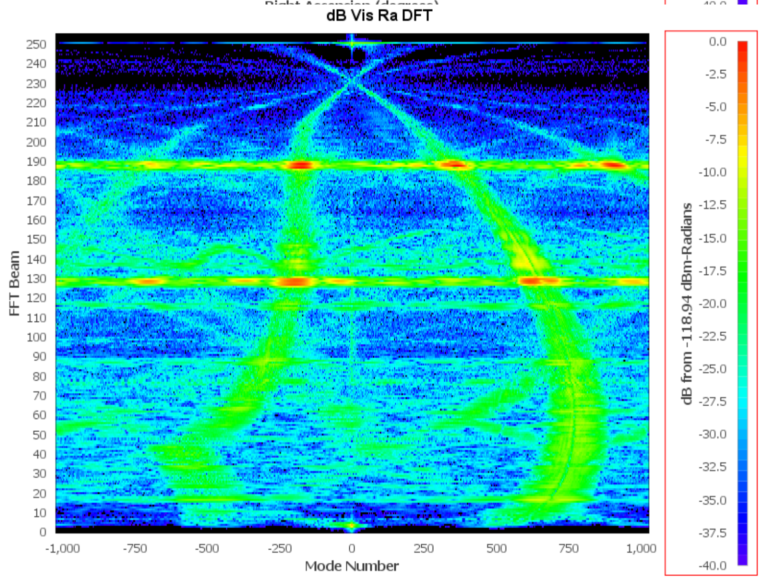
BAO 1st Peak + Angelica | **BAO 1st Peak**

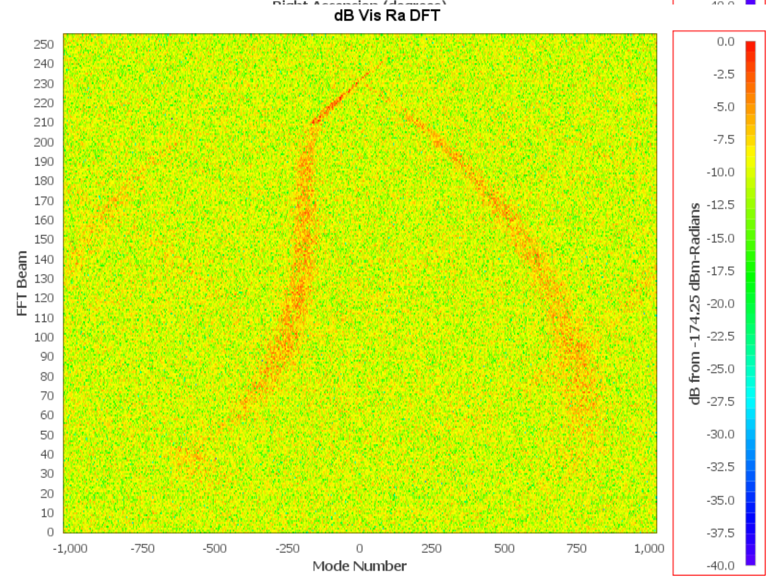
Ra Scan





Ra FFT

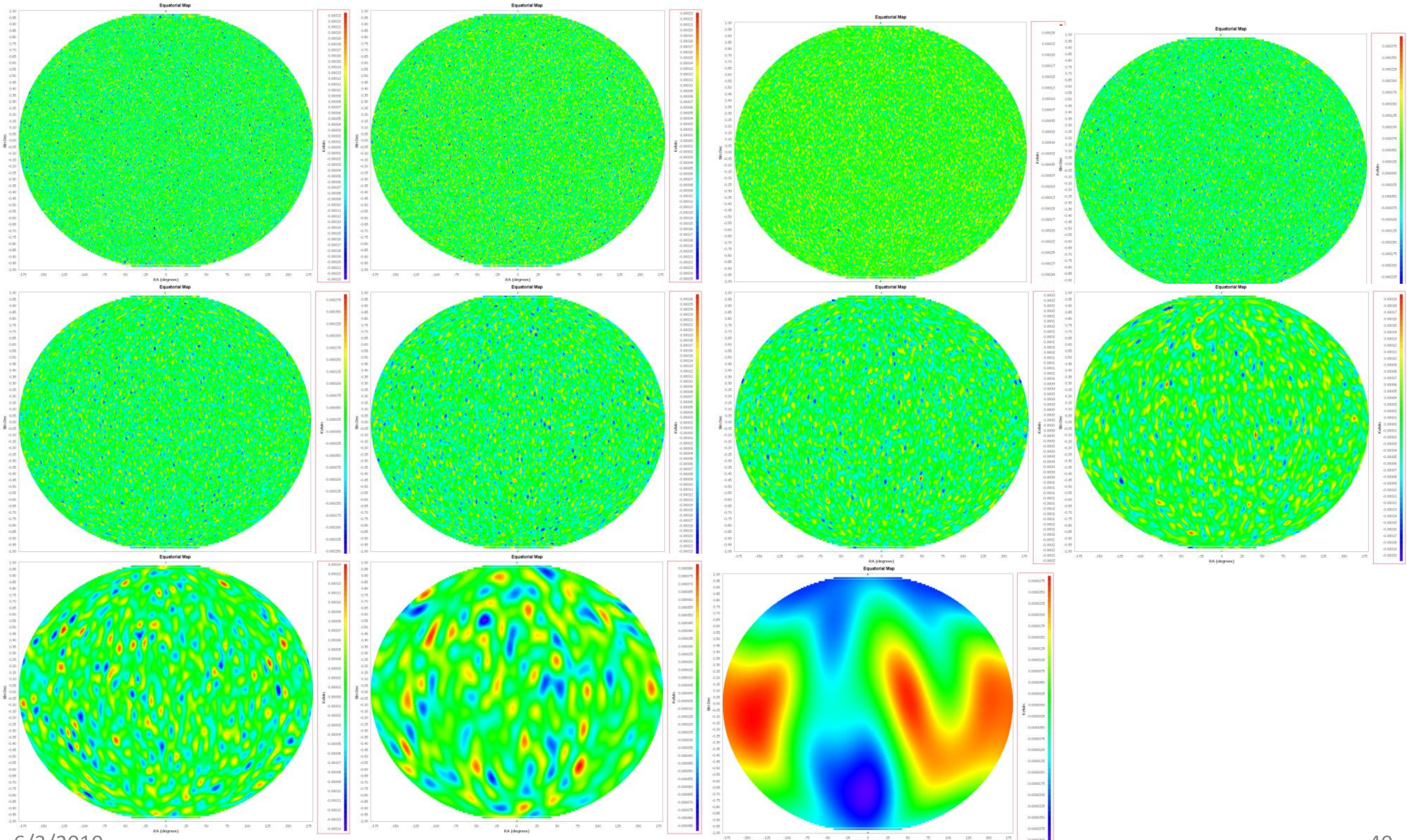




Intermediate Step to Foreground Removal

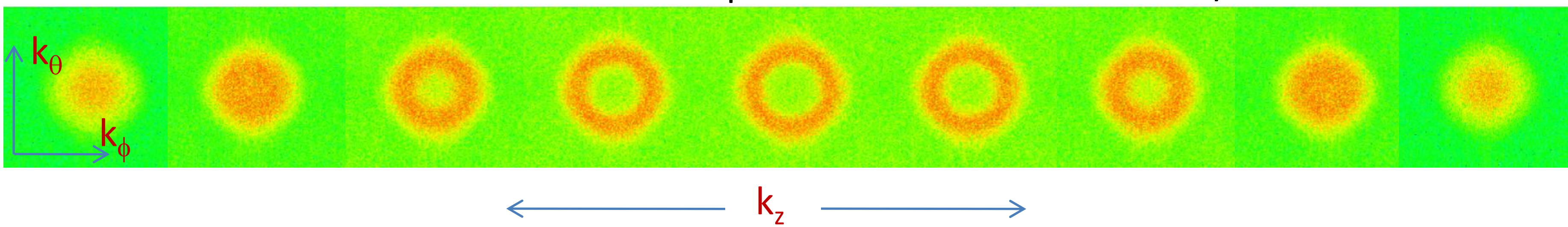
- Separate issues of telescope modeling and sky reconstruction from foreground removal algorithms.
  - Assume that the sky reconstruction will remove issues with mode mixing.
- Use sky maps with frequency-smooth foregrounds overlaid on top of BAO signal to test foreground removal algorithms
- Use BAO simulations of the first peak from Nick Gnedin
  - 1000 frequency points from 400-1400MHz
  - Nside=128

BAO Signal First Peak from 400-1400MHz

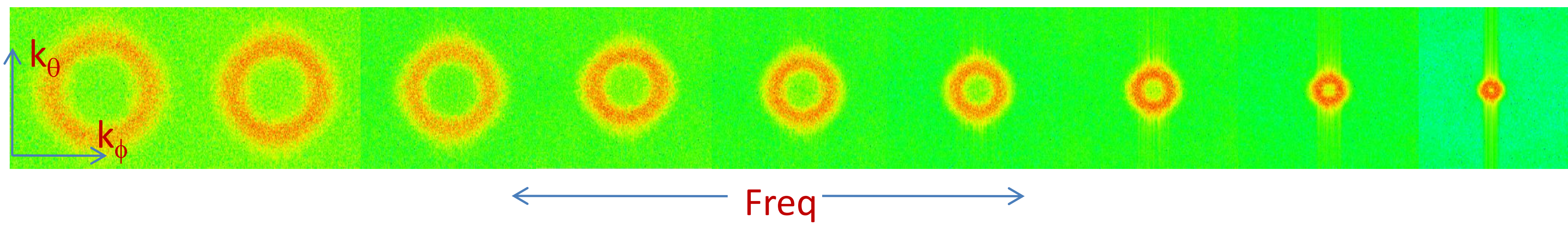


BAO First Peak 3-D K space

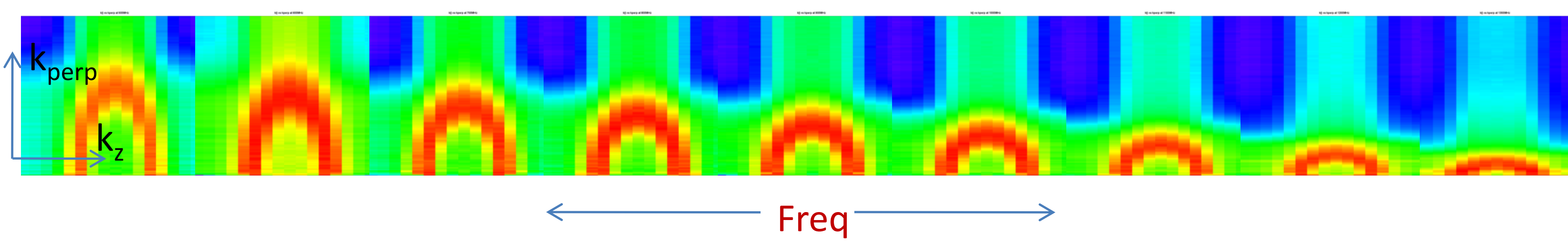
BAO First Peak in 3-D k-Space at 750 MHz – ResBw = 1/128 MHz



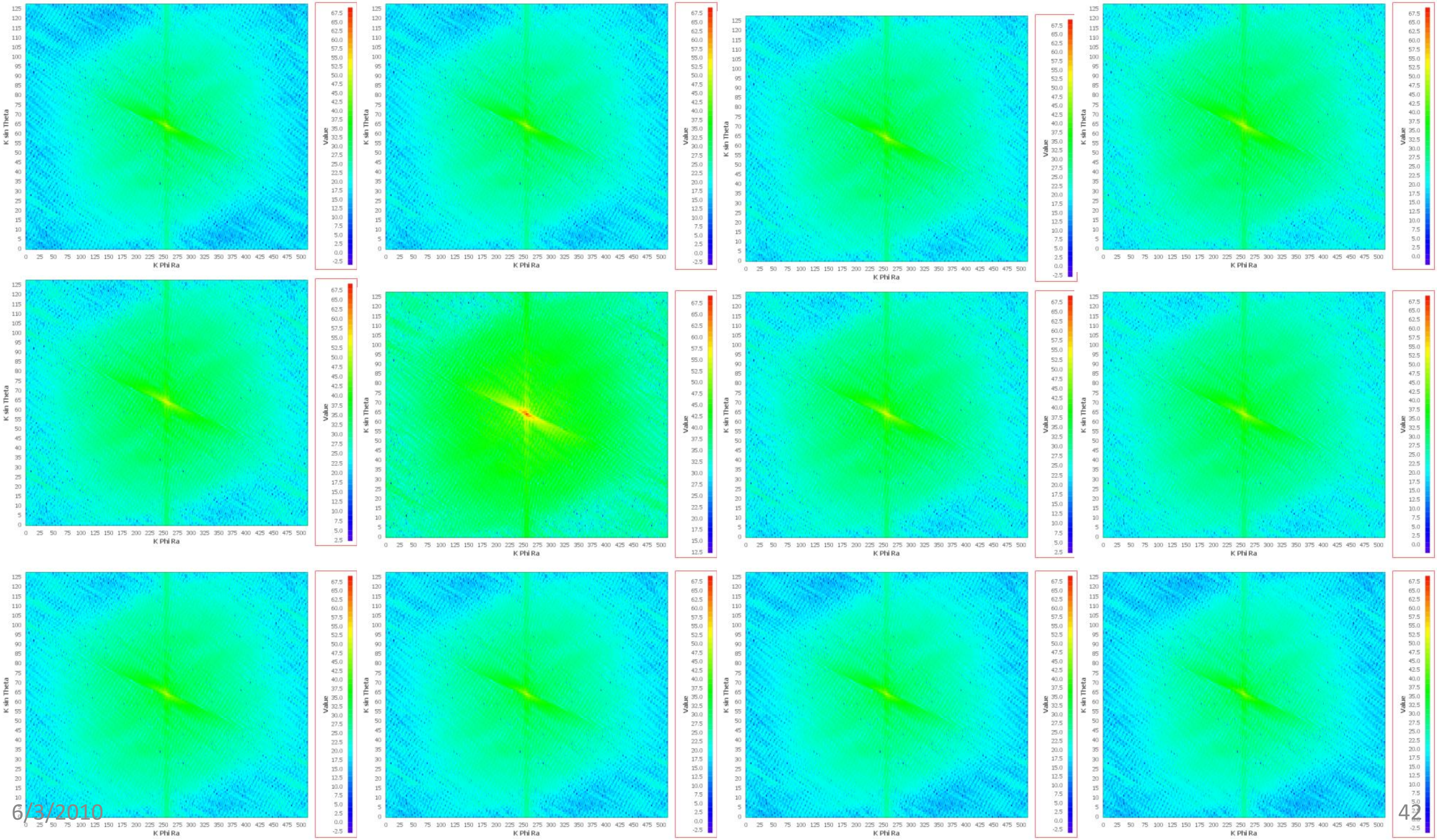
BAO First Peak from 500-1300MHz; Kperp at “k| | = 0 ”; ResBW = 1/128 MHz



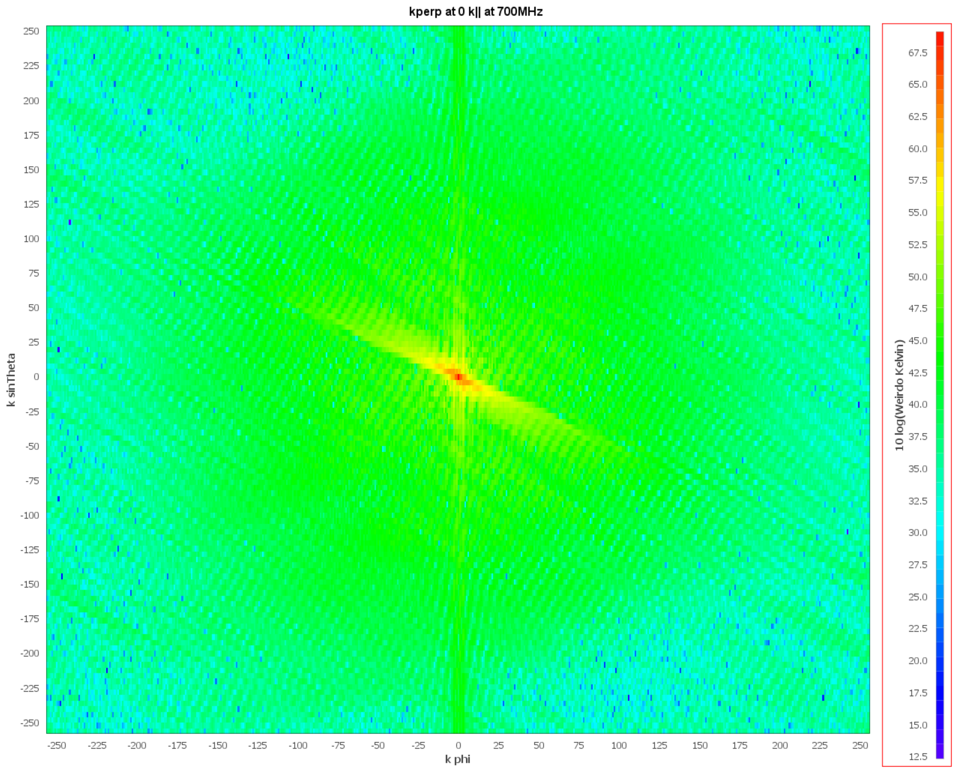
BAO First Peak from 500-1300MHz; kperp vs “k| |” ;ResBW = 1/128 MHz



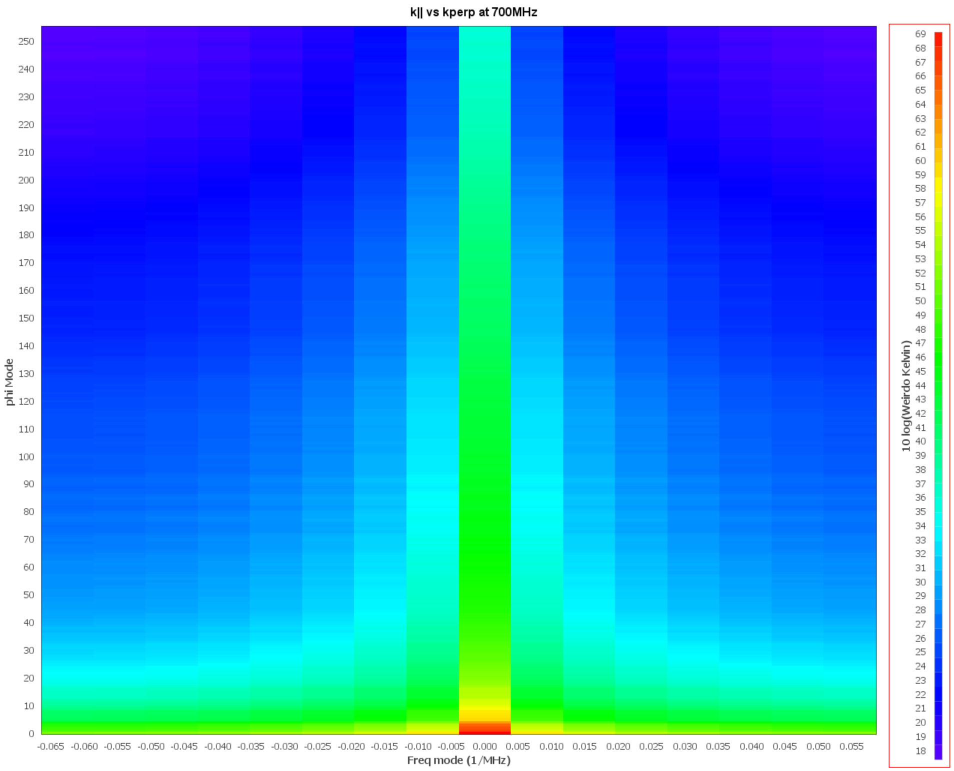
BAO + Angelica Sky
ResBW = 1/128 MHz



BAO + Angelica at 700 MHz
ResBW = 1/128 MHz



kperp at k| | = 0 slice

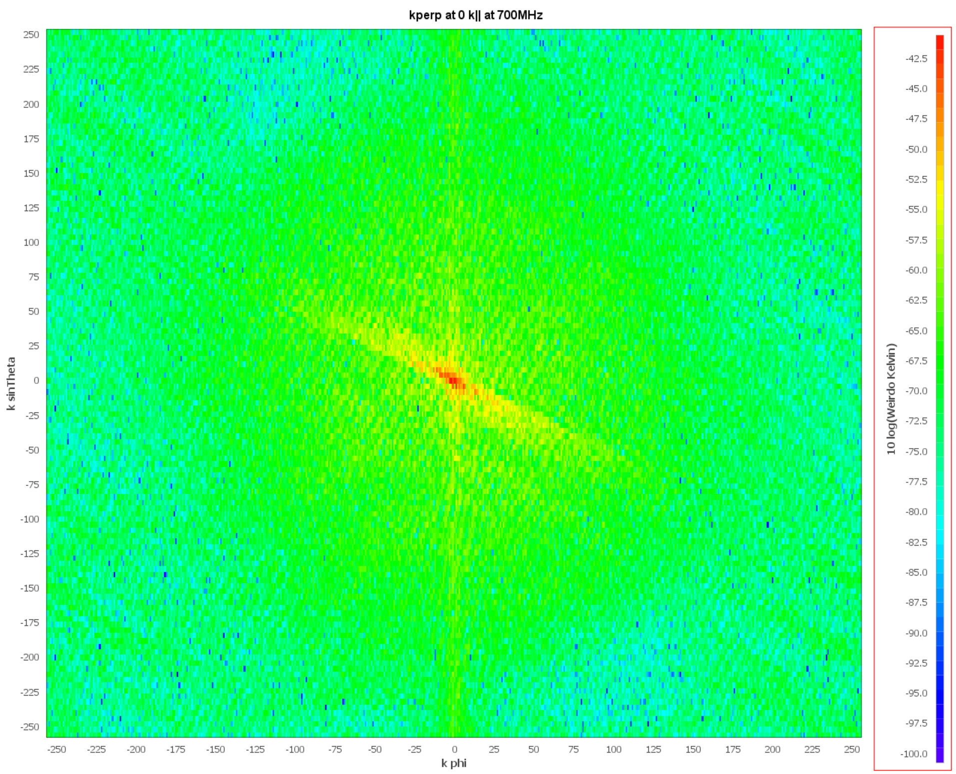


kperp vs “k| |”

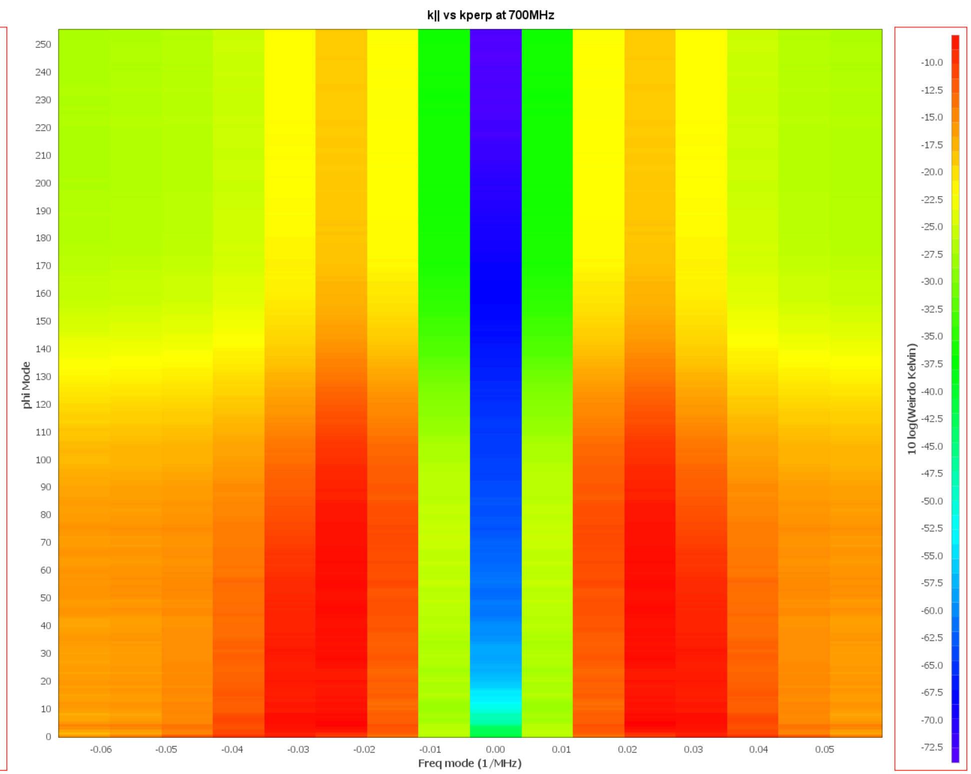
Frequency-Smoothed Sky Algorithm in Reconstructed K-space

- Work in reconstructed sky transverse k-space
  - Addresses using up polynomial fitting "horsepower" on mode mixing
- Smooth in frequency by fitting an N order polynomial along frequency axis for each transverse k-space pixel
- Subtract frequency smoothed k-space from raw k-space
- Fourier transform along frequency axis
- Look the transverse k-space slices at high k|| mode number.

BAO + Angelica Sky at 700 MHz with Foreground Removal
ResBW = 1/128 MHz



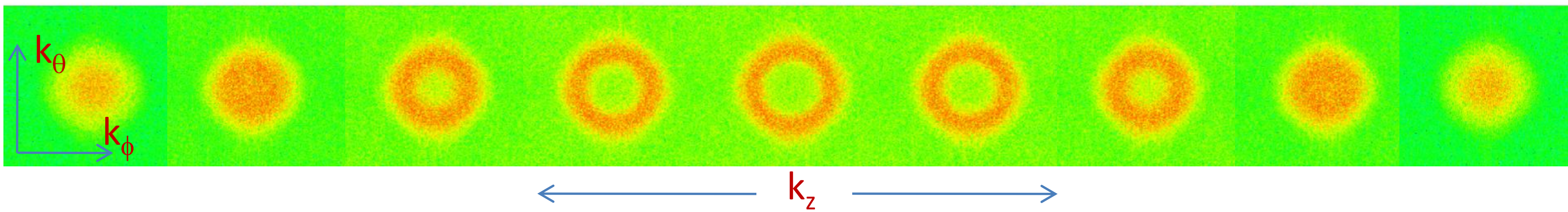
kperp at k| | = 0 slice



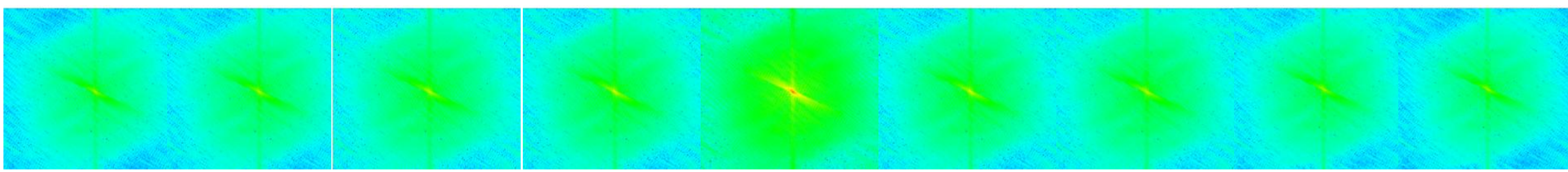
kperp vs “k| |”

Foreground Removal (Fermilab)

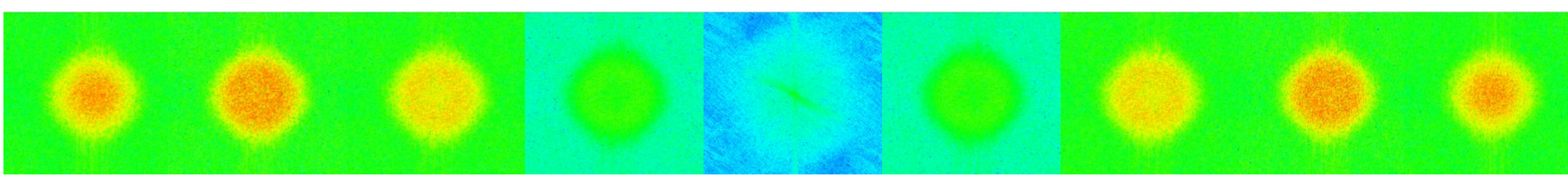
BAO First Peak in 3-D k-Space (Gnedin)



BAO First Peak and Foreground in 3-D k-Space



BAO First Peak and Foreground with Foreground Removal in 3-D k-Space



Summary

- We have developed fairly sophisticated
  - Instrument modeling software
  - Sky Reconstruction software
  - BAO and foreground sky maps
- We have began initial tests of foreground removal algorithms
  - Sky model subtraction algorithm on the raw data cube
  - Frequency-smoothed sky subtraction algorithm on the raw data cube
  - Frequency-smoothed sky subtraction algorithm in reconstructed k-space
- Initial results look promising
  - Can remove 5 orders of magnitude of foreground on a raw data cube
  - Can see the first BAO peak behind foregrounds in reconstructed k-space (6 orders of magnitude reduction)

Future Work

- Get large scale sky reconstruction software working.
- Test foreground removal with k-space reconstruction of a 256 feed/ cylinder telescope
  – Examine the effects of calibration errors
  – Examine the effects of noise
- Add 2nd and 3rd BAO peaks (i.e complete LSS)
- Test other algorithms

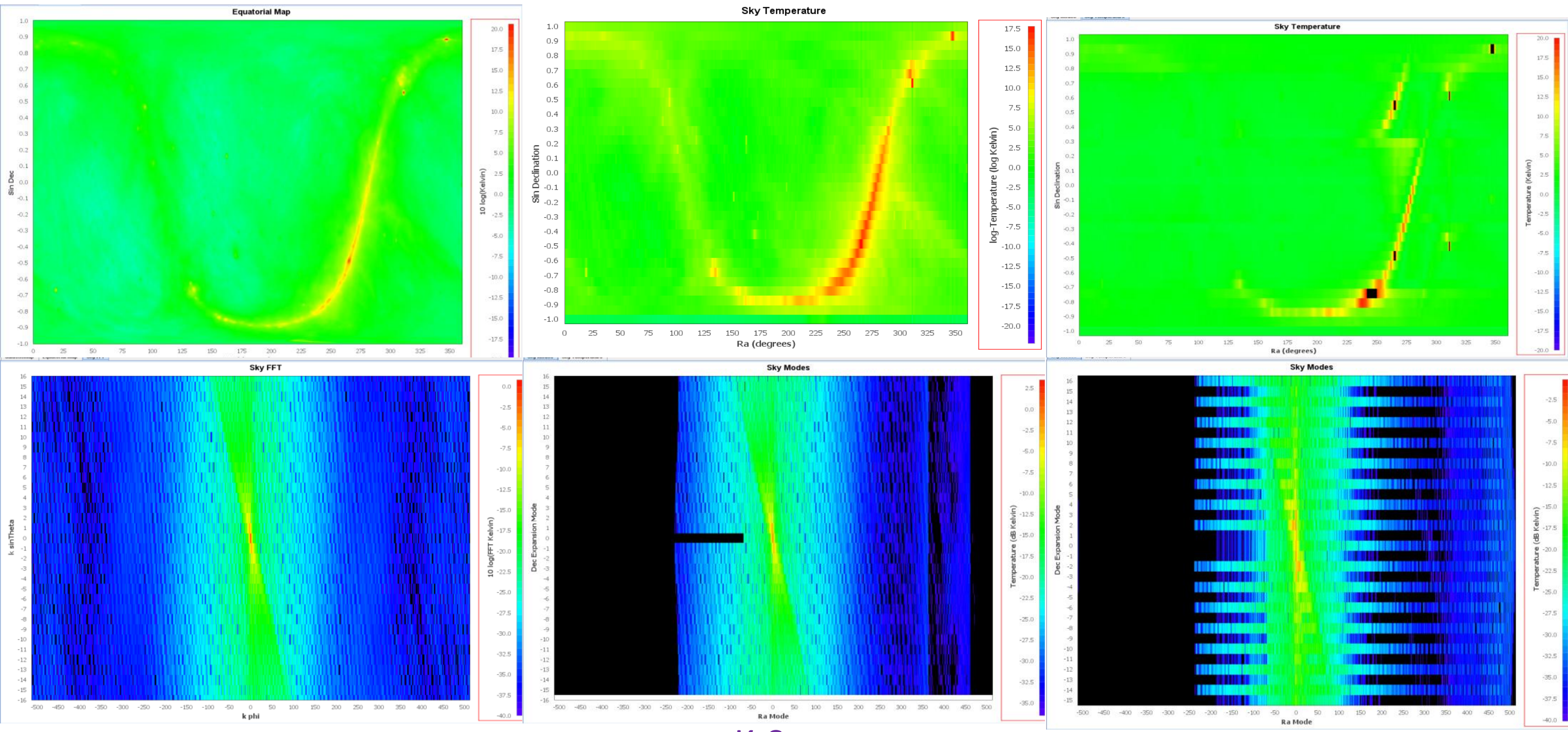
BACKUP SLIDES

Grating Lobes

(Pittsburgh Cylinders)

Input Sky | d/λ=0.5 | d/λ=1.0

Sky Map



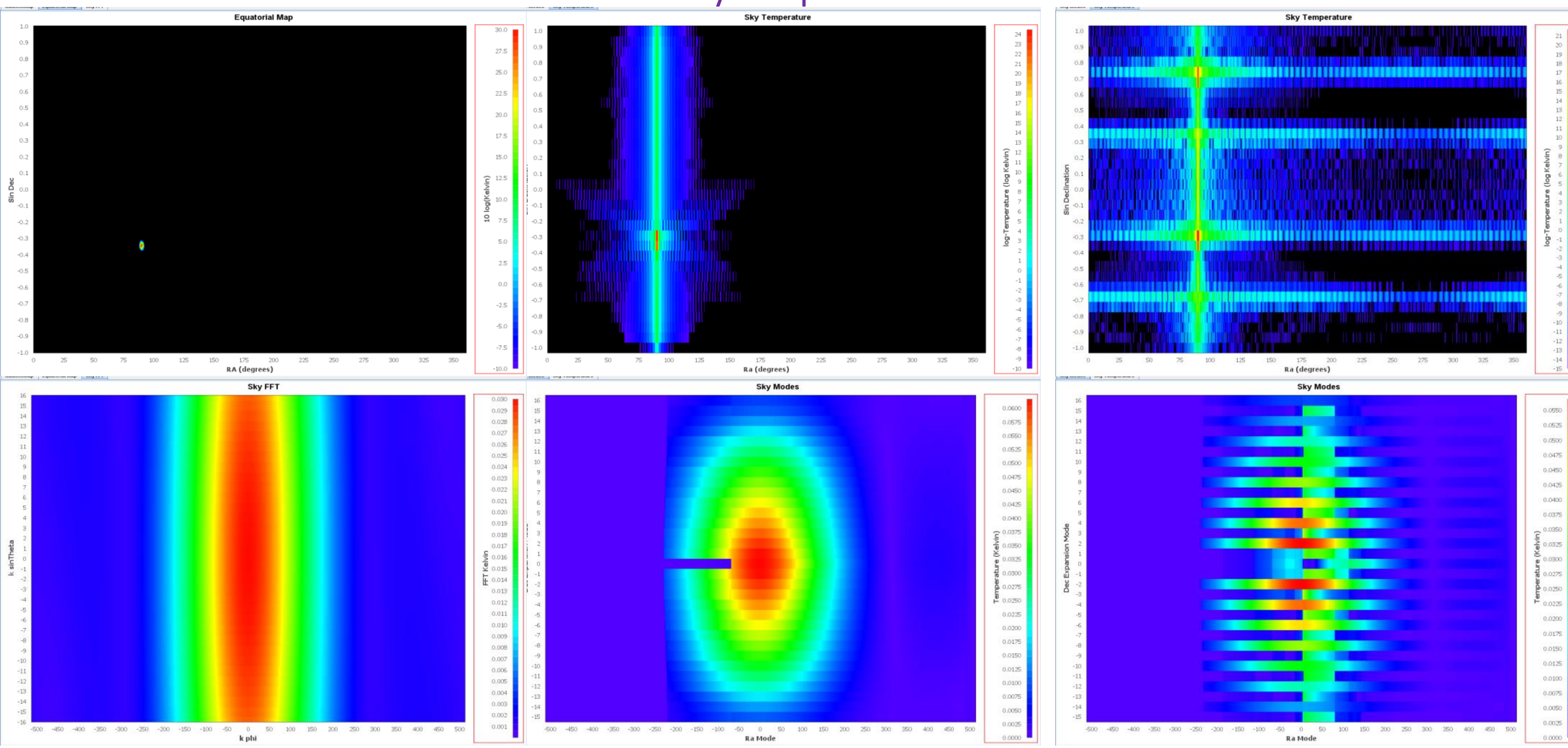
K-Space

Grating Lobes with a Point Source

(Pittsburgh Cylinders)

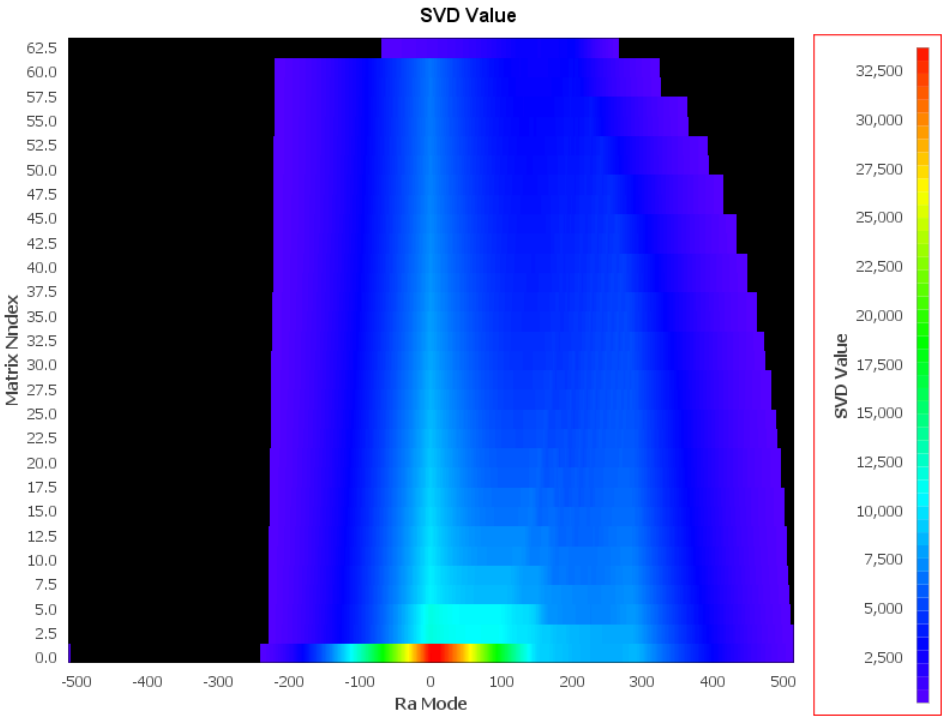
Input Sky | $d/\lambda=0.5$ | $d/\lambda=1.0$

Sky Map

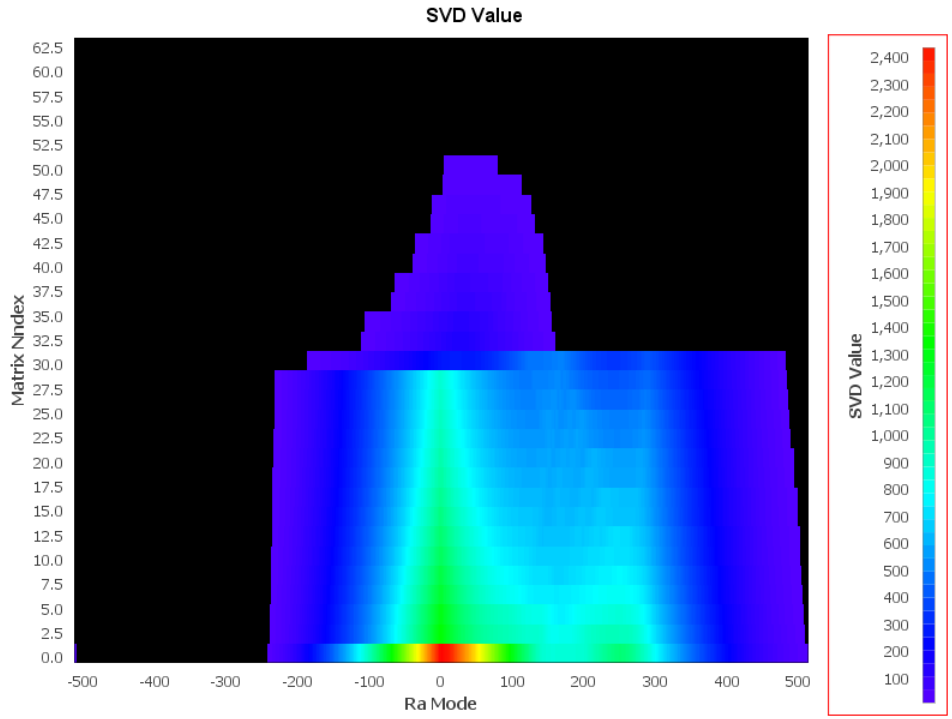


K-Space

SVD Values of Telescope Model Matrix



d/λ=0.5



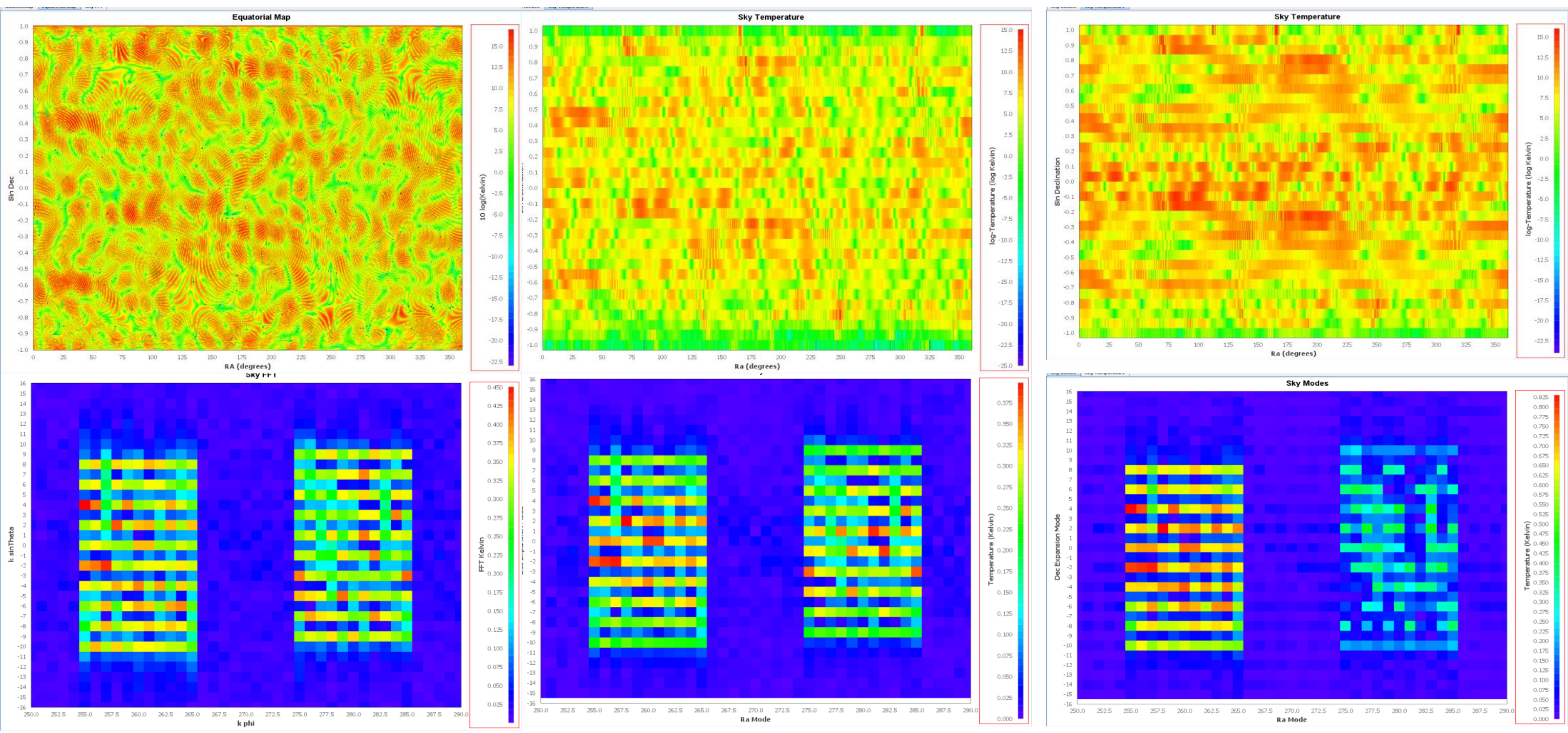
d/λ=1.0

Grating Lobes with Sine Wave Sources

(Pittsburgh Cylinders)

Input Sky | $d/\lambda=0.5$ | $d/\lambda=1.0$

Sky Map



K-Space