21cm Instrument Simulation Software and Foreground Subtraction

Dave McGinnis

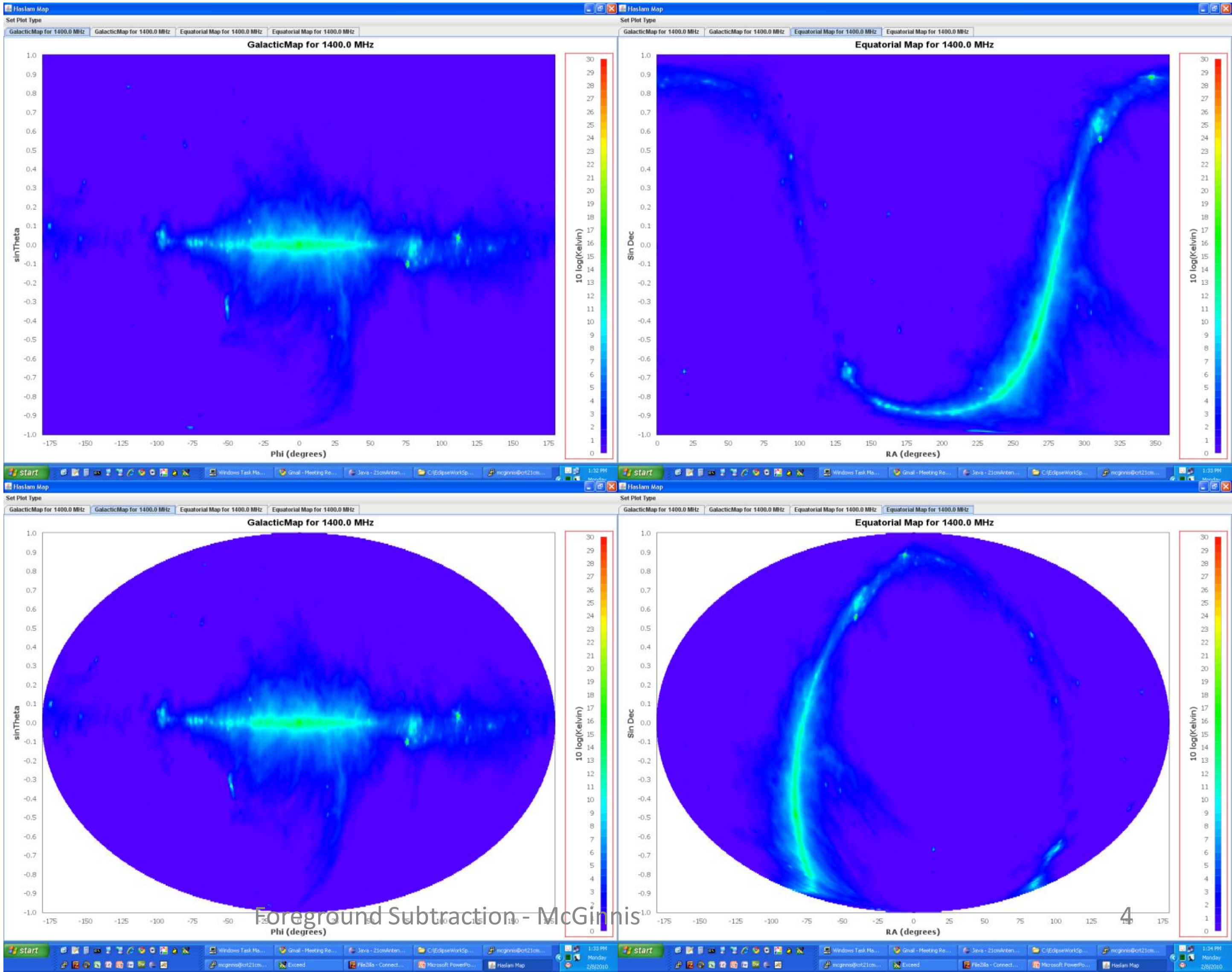
SKY SIMULATION SOFTWARE

Code Suite

- 20 Java classes organized into 7 packages
- Major Packages
  - Sky Map Generator
  - Cylinder Visibility Simulator
  - Cylinder Visibility Modeler
  - Sky Reconstructor

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



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:

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

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

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

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:

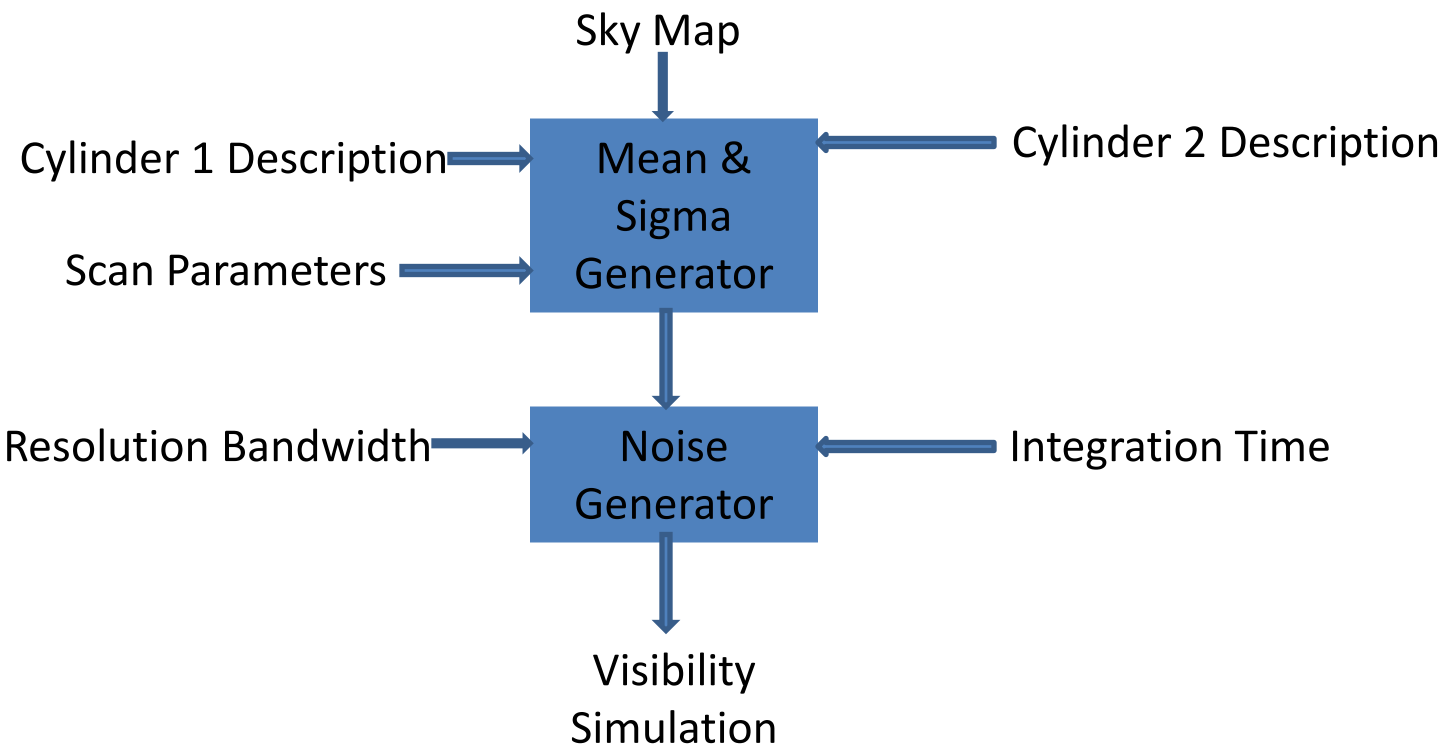
$$p_{n,m} = p_{z_{n,m}} + \sum_q \Delta\Omega_{\mathrm{q}}\mathrm{f_q}\mathrm{a_{q,n,m}}e^{-j\vec{\beta}_q\cdot\vec{r}_{n,m}} \quad (6)$$

Cylinder Amplitude

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

http://projects-docdb.fnal.gov/cgi-bin/ShowDocument?docid=778

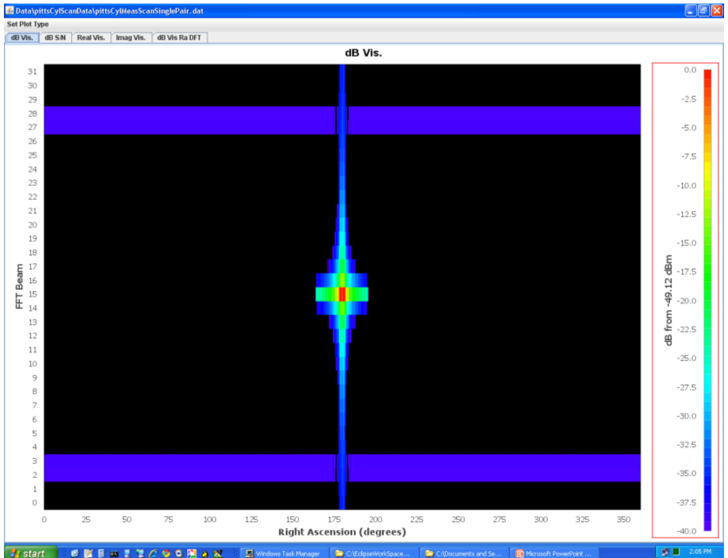
Cylinder Visibility Simulator

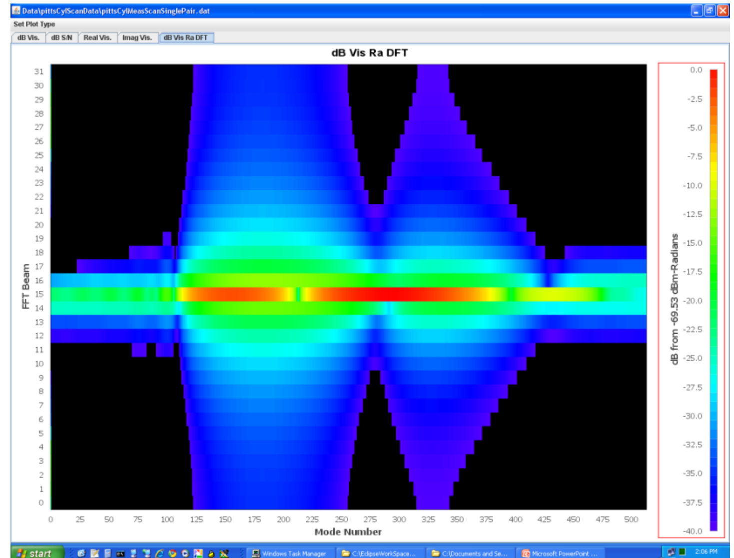


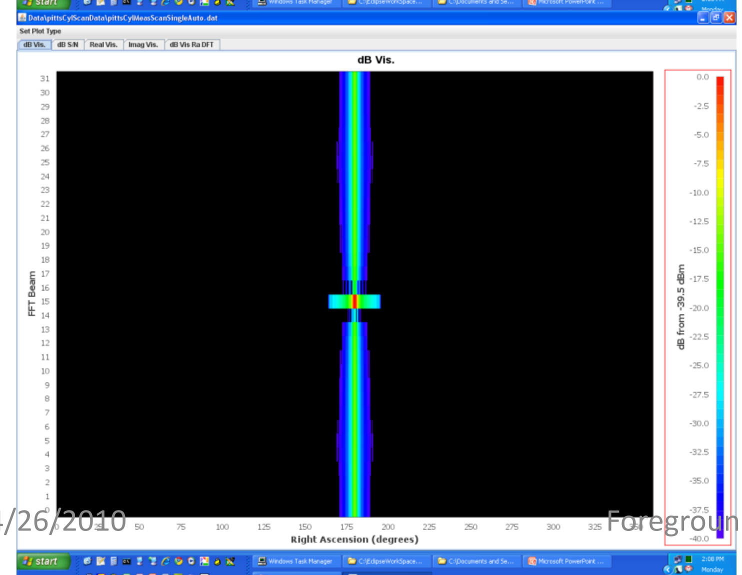
Cylinder Description XML Format

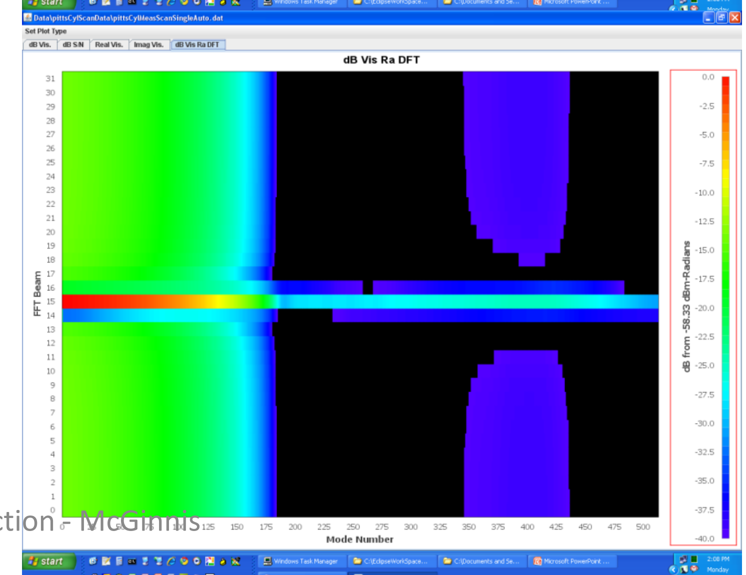

```xml
<?xml version="1.0" encoding="UTF-8" standalone="no" ?><!--blah--><!--
--><Cylinder>
    <Feed ImagGain="0" RealGain="1" effWidth="7" lengthLocation="0" nCosThetaResponsePower="1" noiseTemp="75" powerEfficiency="0.5" widthLocation="0" />
    <Feed ImagGain="0" RealGain="1" effWidth="7" lengthLocation="0.145" 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 Visbility due to a Point Source

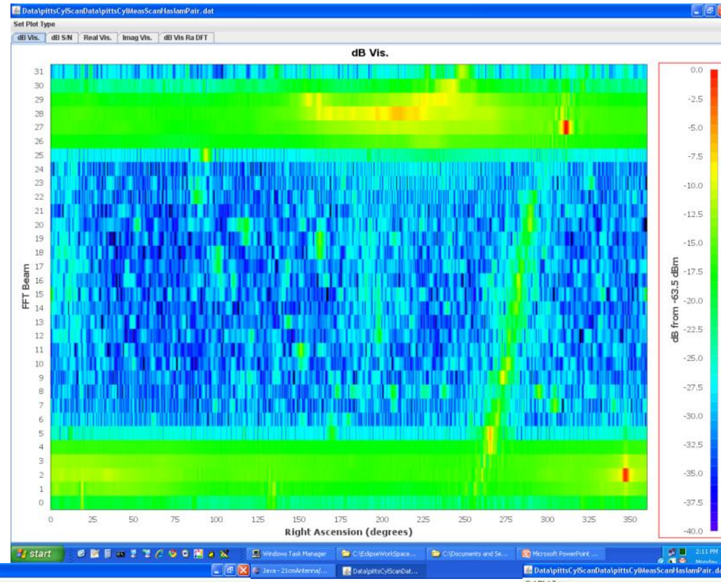




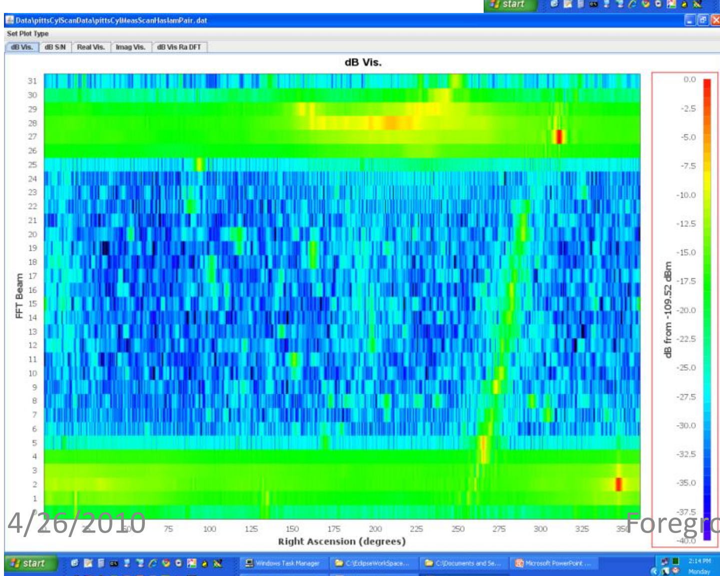




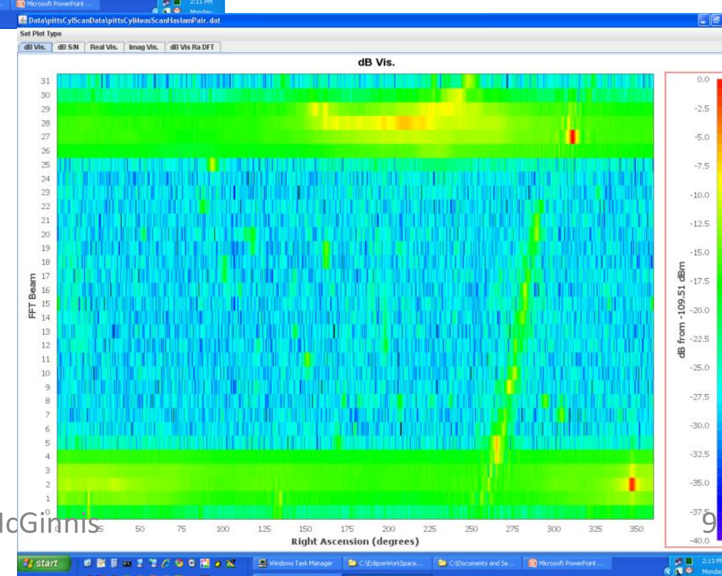
Pittsburgh Cylinders Visibility 25 MHz Res. BW



100 day integration



1 day integration



Cylinder Modeler

Sky Reconstruction from Cylinder Visibilities

Dave McGinnis
February 4, 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:

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

Where **φ** is the time of the day (in units of angle), **λ** is the wavelength and **T** is the power flux of the sky. The cylinder pair Fourier area is defined as

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

where the subscripts **c1**, **c2** indicate cylinder 1 and cylinder 2, respectively. The Fourier root area of a cylinder is defined as

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

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

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

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

Sky Expansion

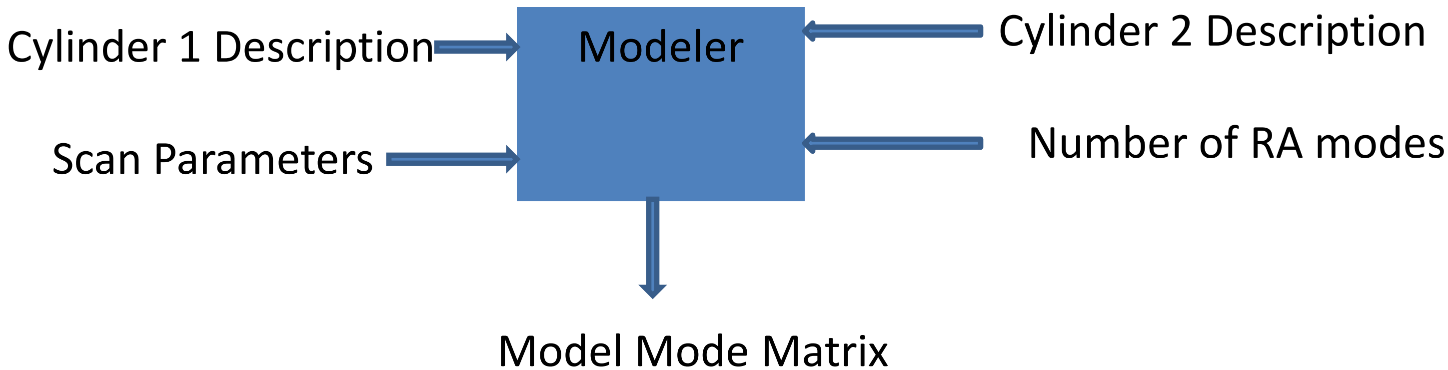
Since the sky is periodic in right ascension, it can be expanded in a Fourier series:

$$T(\theta,\phi) = \sum_l \chi_l(\theta)\left(\tilde{T}_{dc_{l,0}} + \sum_m \tilde{T}_{c_{l,m}}\cos(m\phi) + \sum_m \tilde{T}_{s_{l,m}}\sin(m\phi)\right) \qquad (5)$$

Substituting Equation 5 into Equation 1,

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

Cylinder Visibility Modeler



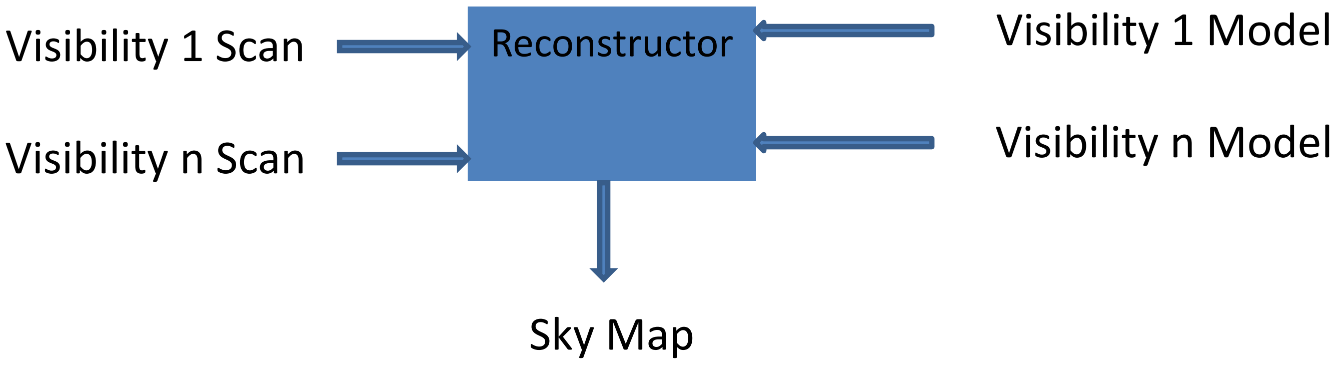
$$\tilde{V}^{(R)}{}_{k,m} = \sum_{l} \hat{A}^{(R)}{}_{k,l,m}\tilde{T}_{l,m}$$

$$\tilde{V}^{(R)}{}_{k,m} = \frac{2}{N}\sum_{n} Re\{v_k(\varphi_n)\}\, e^{-jm\varphi_n}$$

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

$$T(\theta,\phi) = \sum_{l} \chi_l(\theta)\left(\tilde{T}_{dc_{l,0}} + Re\left\{\sum_{m} \tilde{T}_{c_{l,m}} e^{jm\phi}\right\}\right)$$

Sky Reconstruction



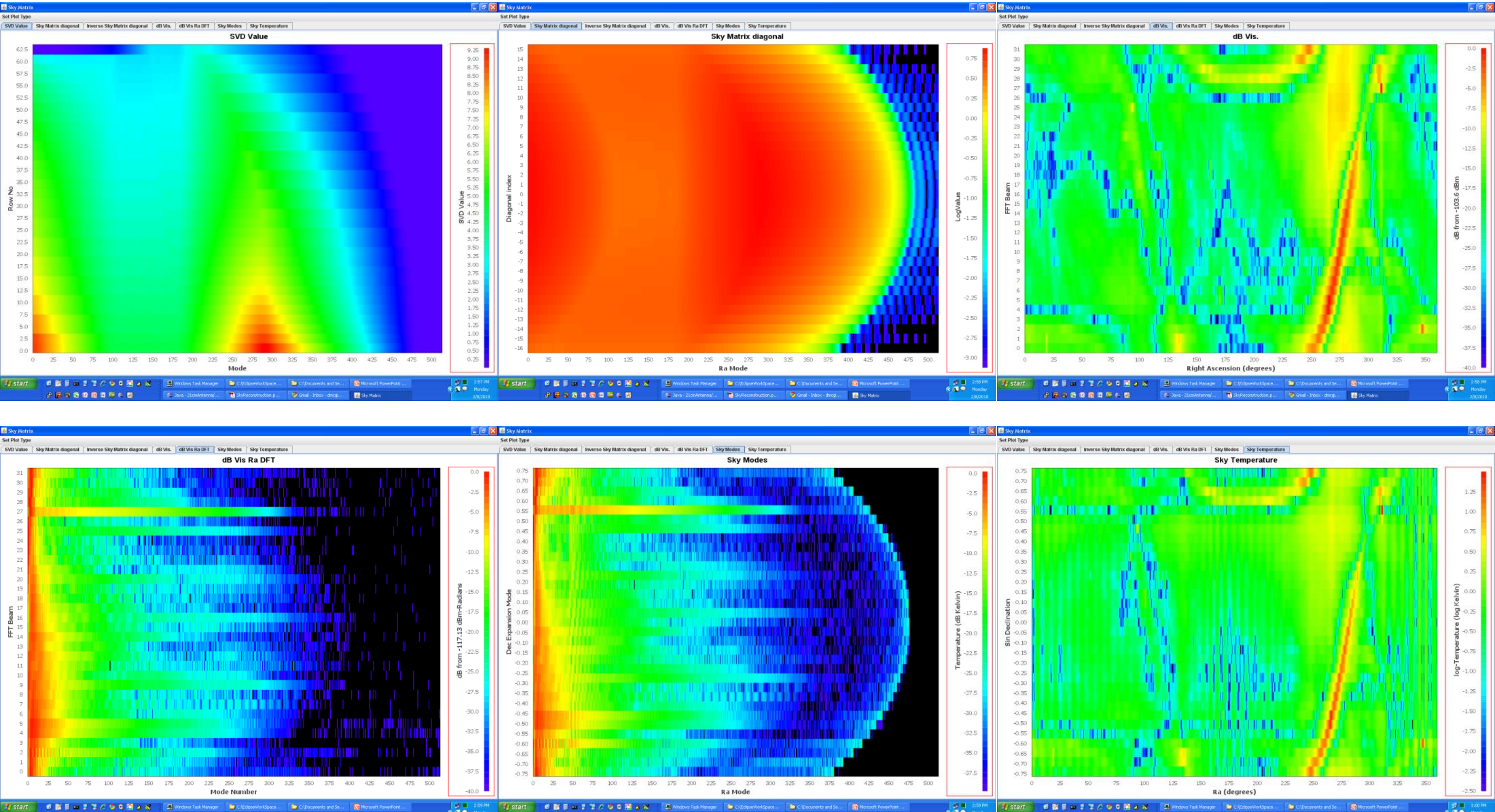
$$\tilde{V}^{(R)}{}_{k,m} = \sum_{l} \hat{A}^{(R)}{}_{k,l,m} \tilde{T}_{l,m}$$

$$\tilde{V}^{(R)}{}_{k,m} = \frac{2}{N} \sum_{n} Re\{v_k(\varphi_n)\}\, e^{-jm\varphi_n}$$

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

$$T(\theta,\phi) = \sum_{l} \chi_l(\theta) \left(\tilde{T}_{dc\,l,0} + Re\left\{ \sum_{m} \tilde{T}_{c\,l,m} e^{jm\phi} \right\} \right)$$

Pair and Auto Pittsburgh Cylinder Haslam Map Reconstruction

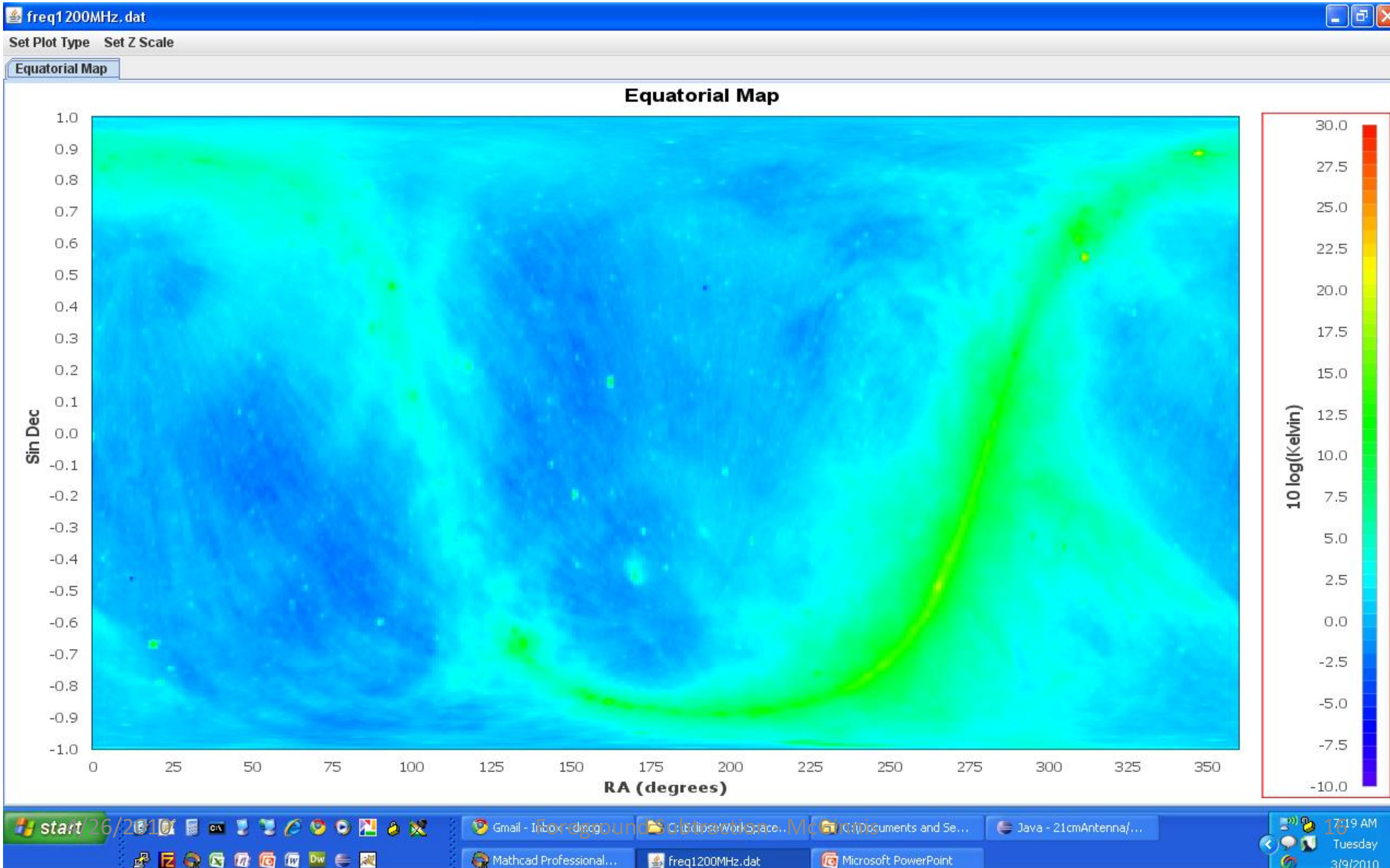


FOREGROUND SUBTRACTION FLUCTUATING SKY PATCH

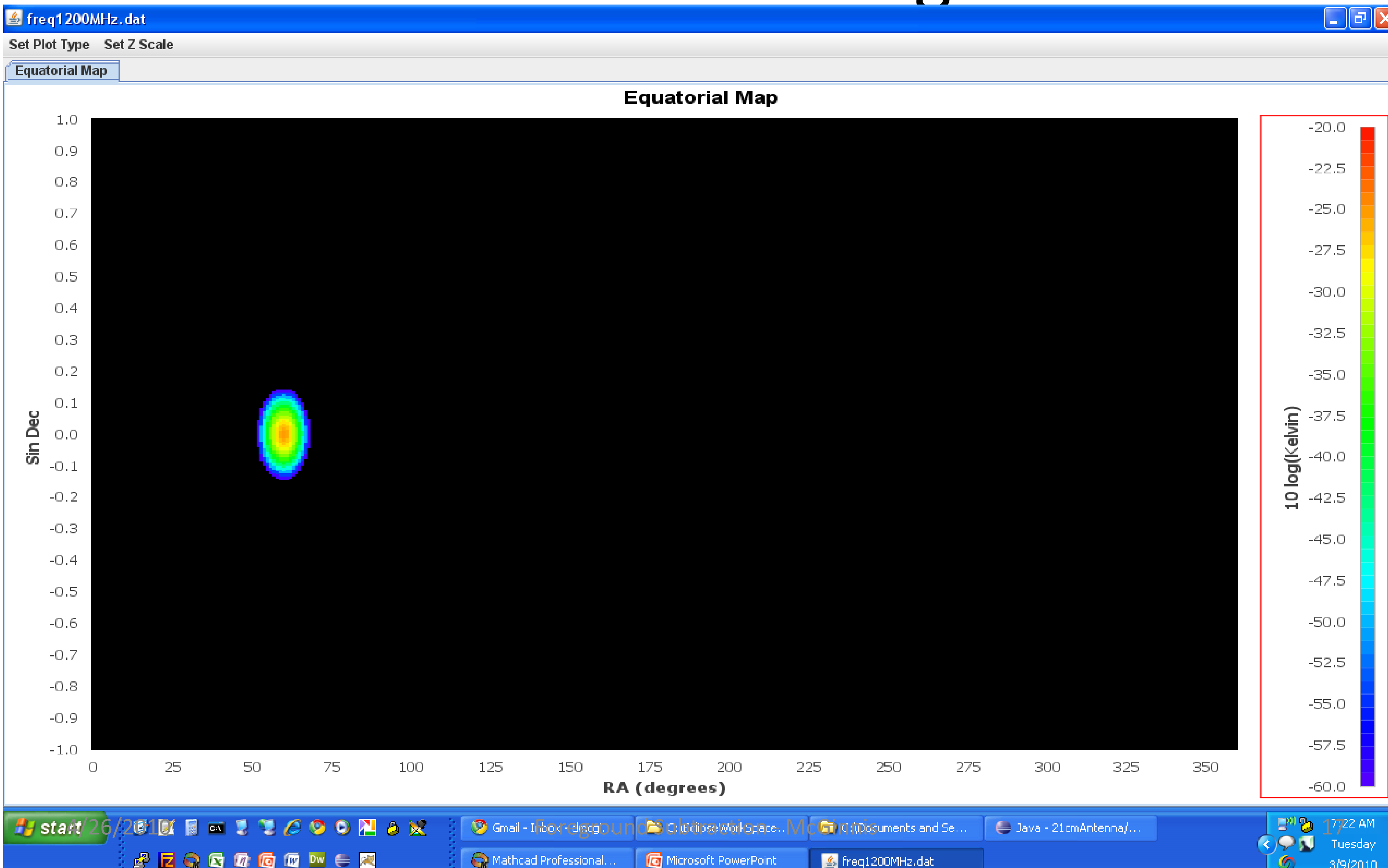
Sky Model Subtraction Algorithm

- Take cylinder visibility data and subtract a simulation of a 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 smoothed pixel trace from the difference map pixel by pixel
- Further FFT filter in frequency each the remaining pixel trace

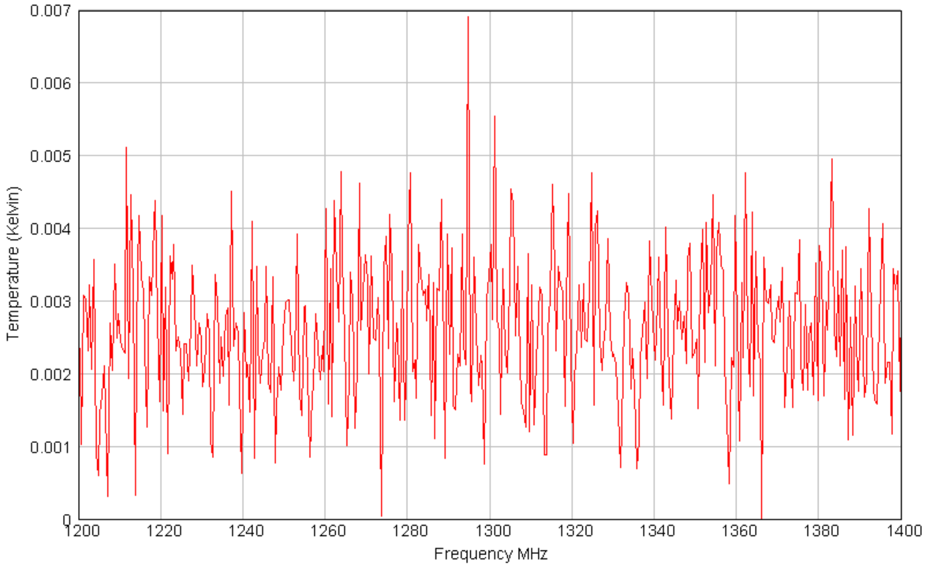
Angelica Sky Map

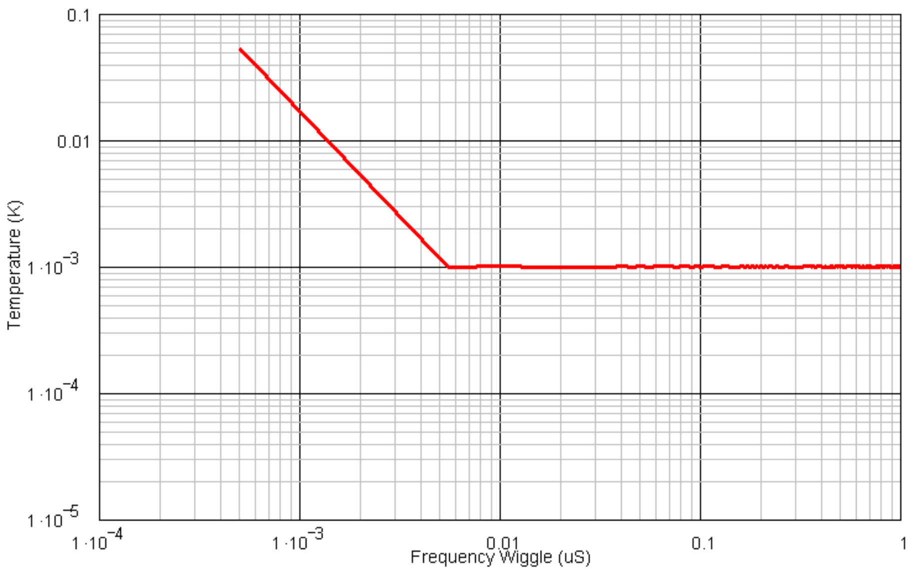


Freq. Fluctuation Patch
r.m.s radius = 3 degrees

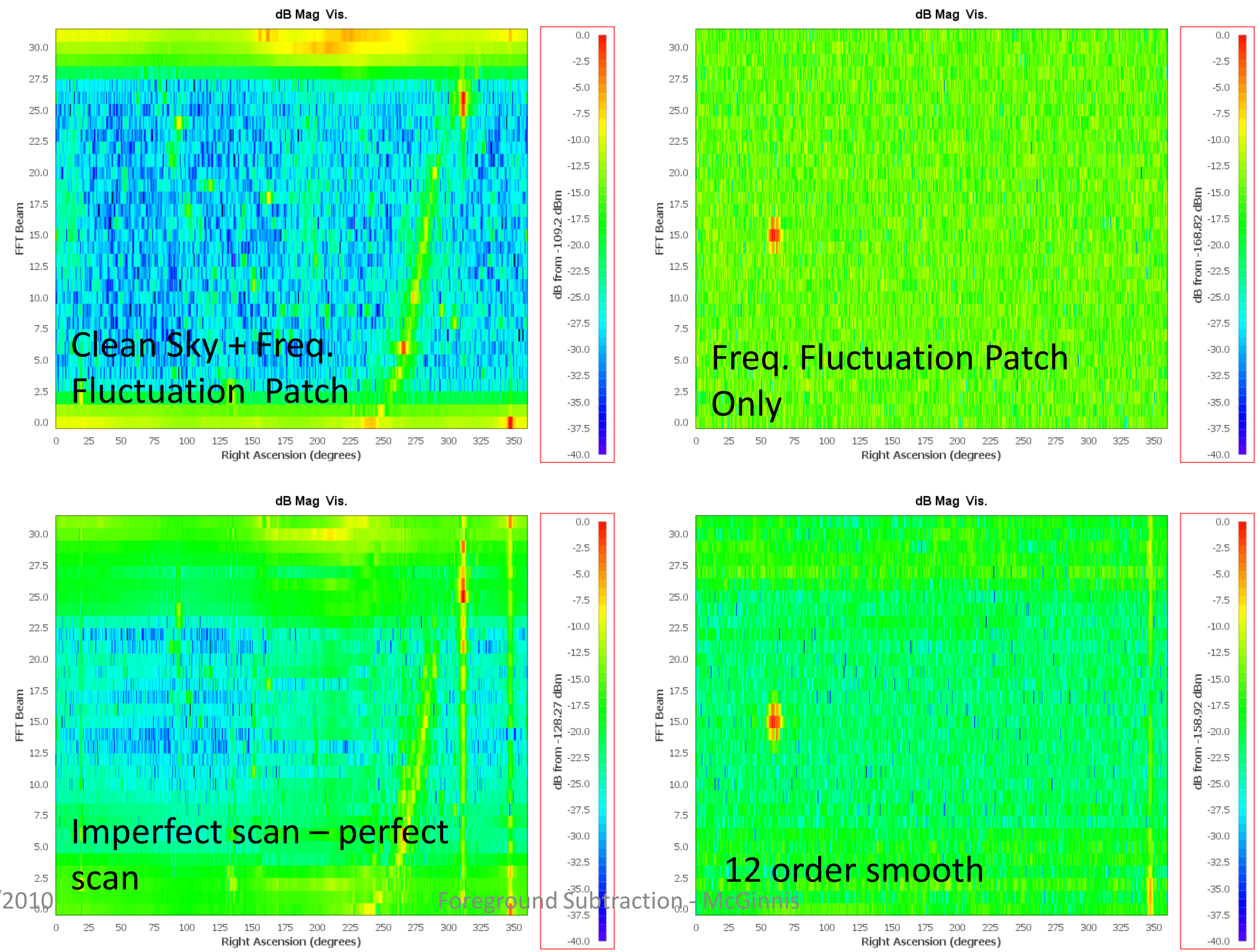


Freq. Fluctuation Patch Temperature vs Frequency



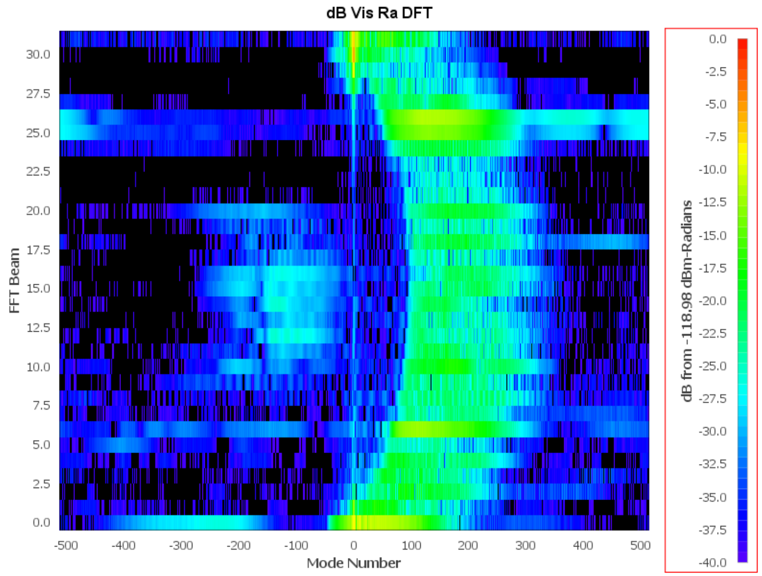


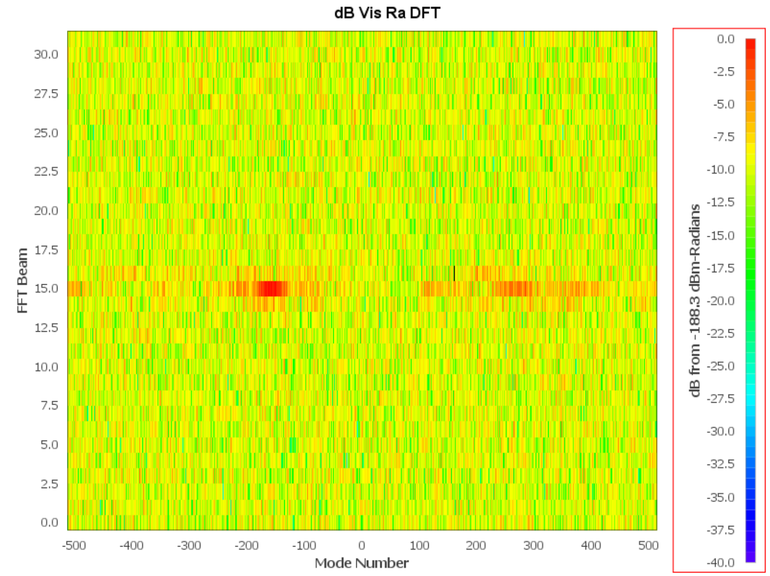
Pittsburgh Cylinder Simulations Sky Scan



RA DFT of Pittsburgh Cylinder Simulations

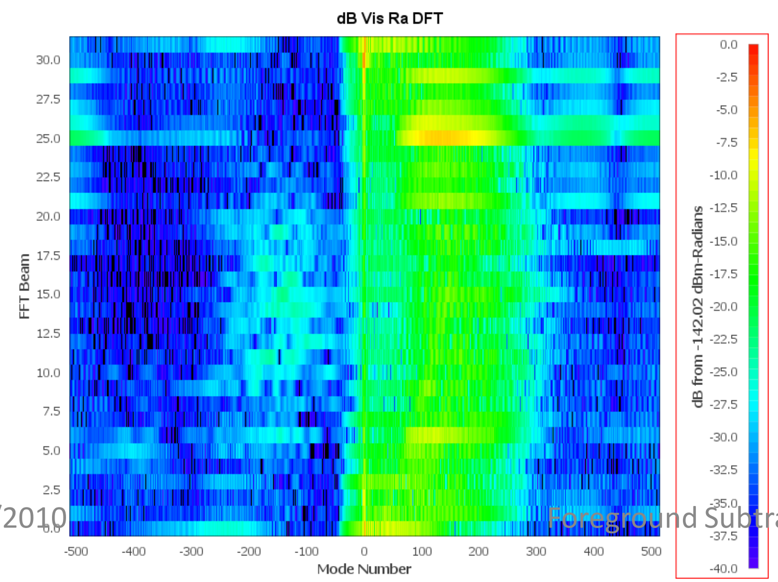
Clean Sky + Freq. Fluctuation Patch

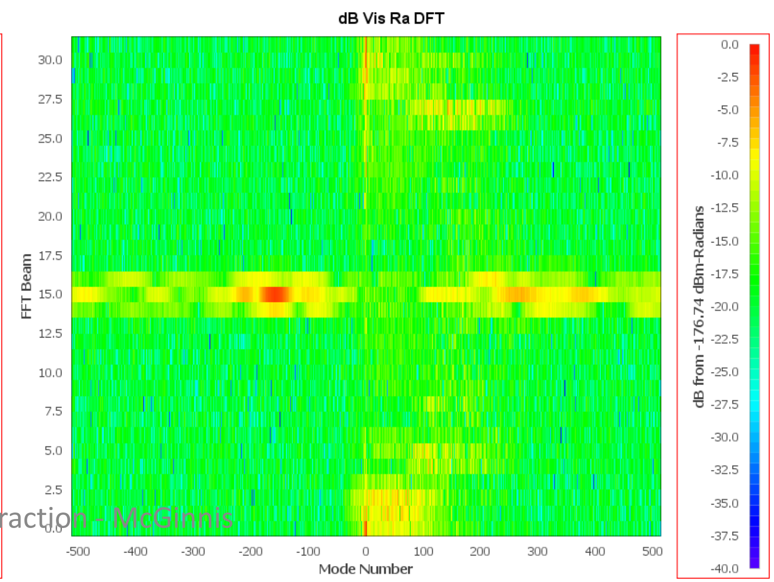




Freq. Fluctuation Patch Only

Imperfect scan – perfect scan

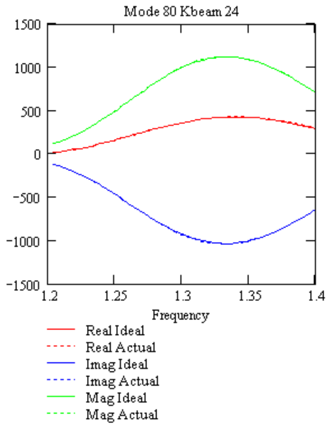


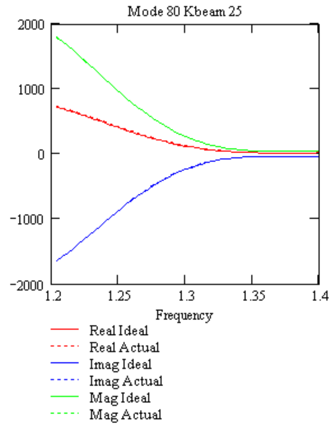


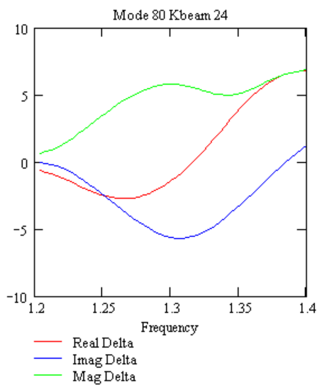
12 order smooth

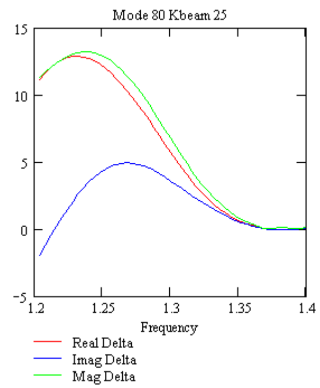
Mode Mixing Smoothness

"Hot Pixel" track before Sky Subtraction

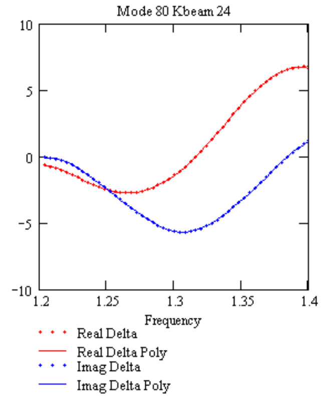


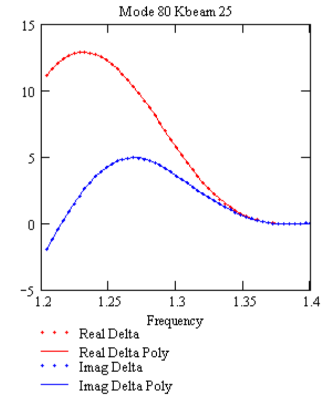


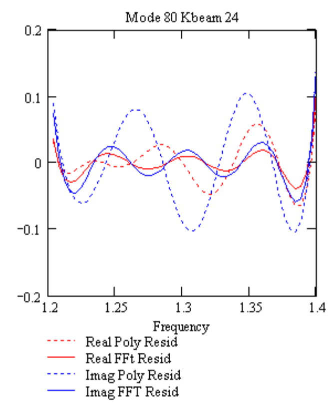


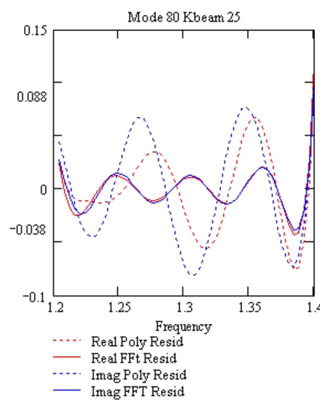


"Hot Pixel" track after Sky Subtraction





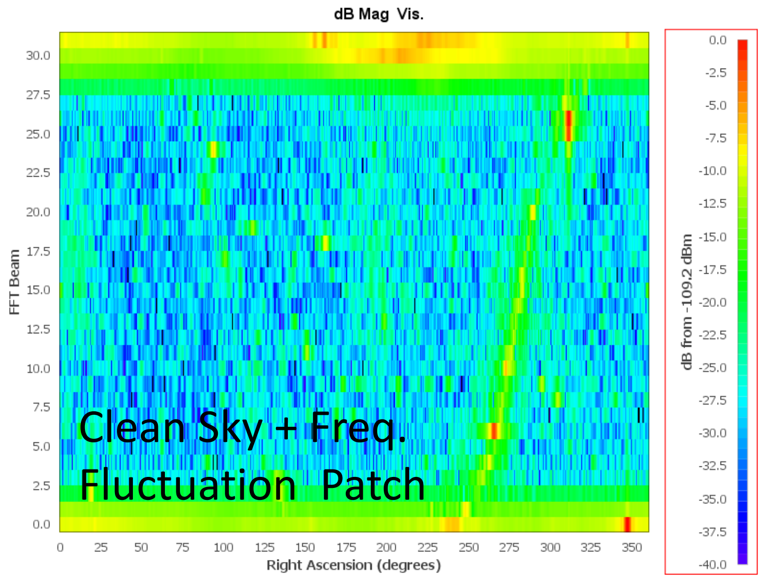


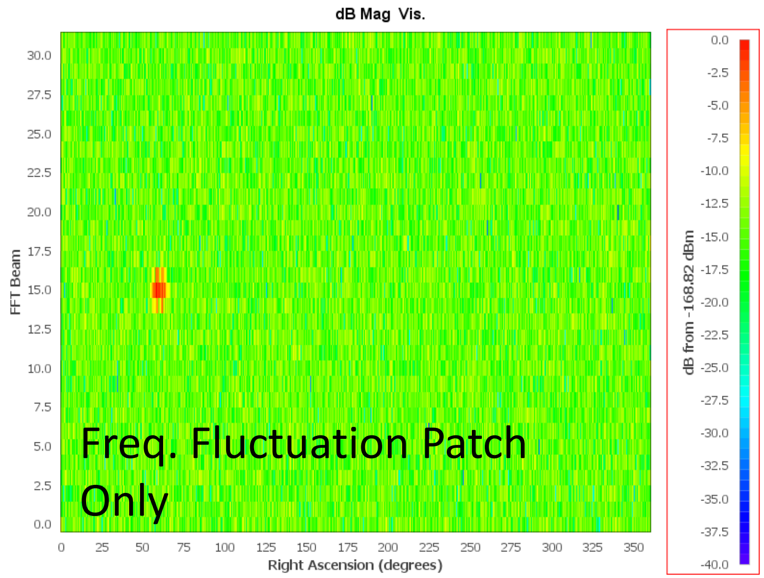


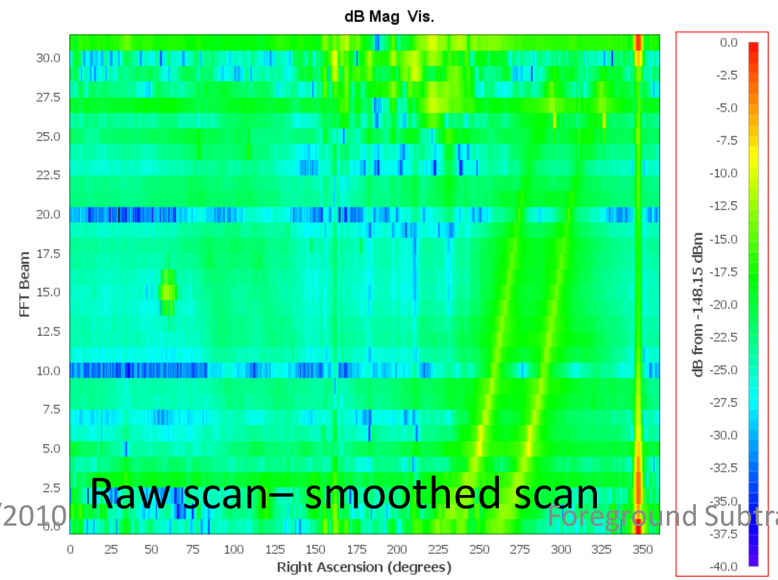
Smoothed Sky Subtraction Algorithm

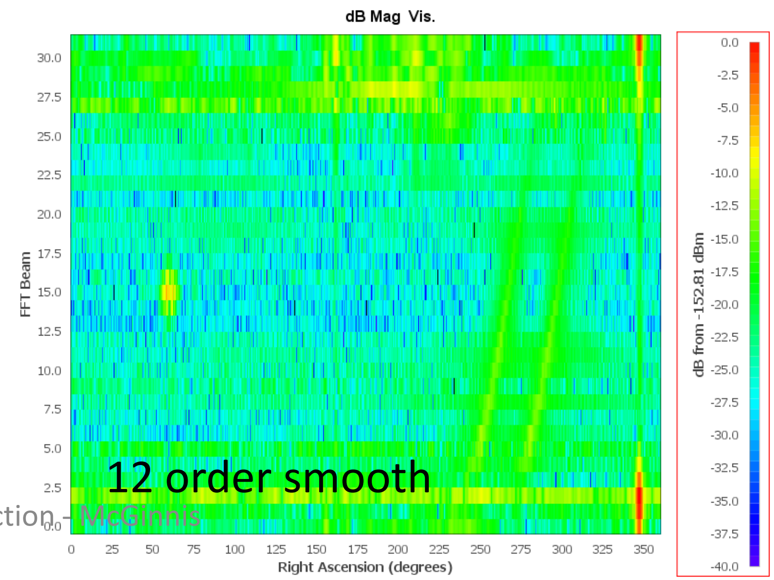
- Take cylinder visibility data smooth it along the frequency axis using a N order polynomial for each pixel
- Subtract the smoothed map from the raw map producing a difference map
- From the difference map, fit each visibility spectrum "pixel" as a nth order polynomial in frequency
- Subtract the smoothed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

Pittsburgh Cylinder Simulations Sky Scan





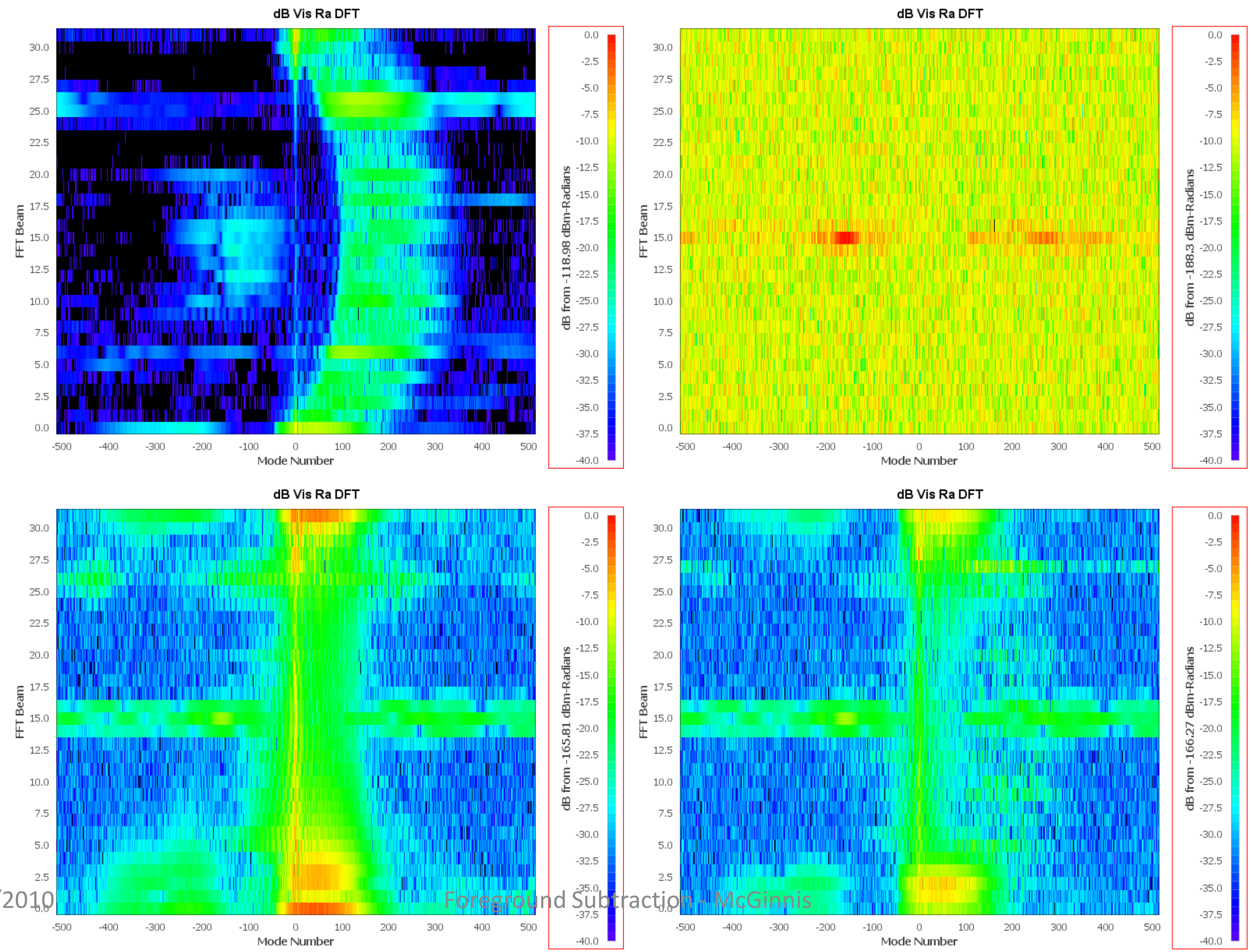




RA DFT of Pittsburgh Cylinder Simulations

Clean Sky + Freq. Fluctuation Patch

Freq. Fluctuation Patch Only



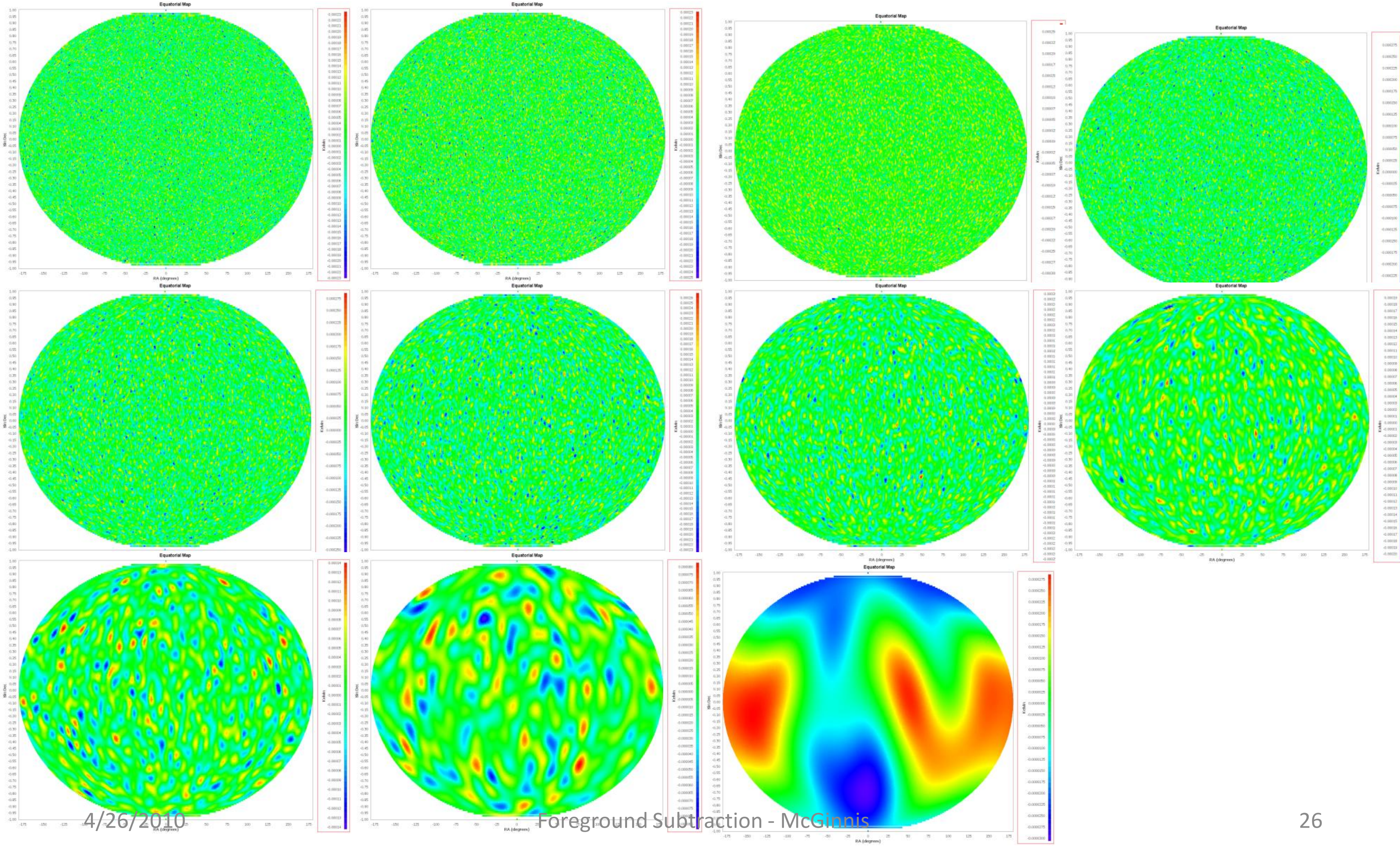
Raw scan – smoothed scan

12 order smooth

Foreground removal using BAO Simulations

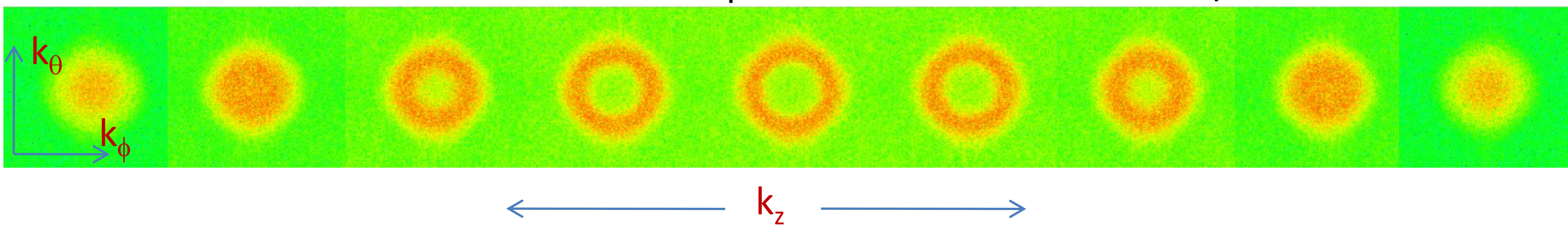
- For simplicity - "use high resolution telescope model"
  - Pittsburgh telescope cannot resolve first BAO peak
- Use BAO simulations of the first peak from Nick Gnedin
  - 1000 frequency points from 400-1400MHz
  - Nside=128

BAO Signal First Peak from 400-1400MHz

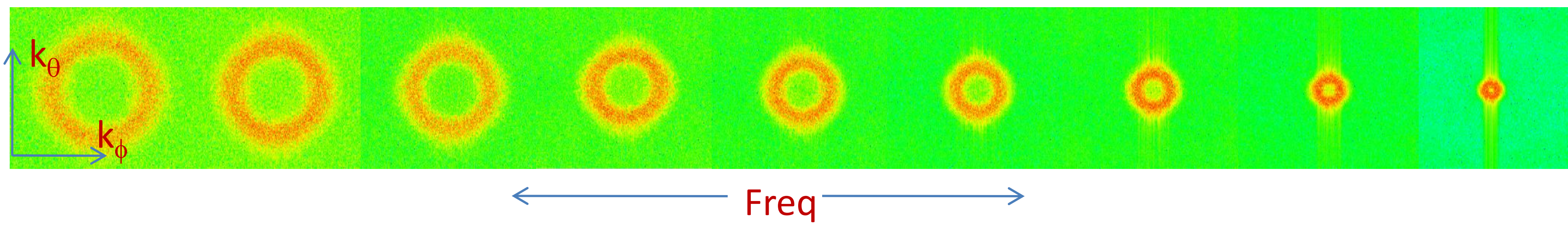


BAO First Peak 3-D K space

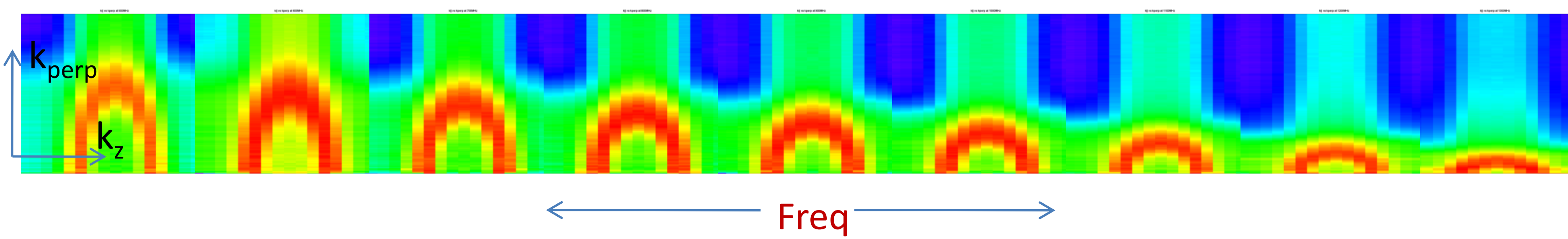
BAO First Peak in 3-D k-Space at 750 MHz – ResBw = 1/128 MHz



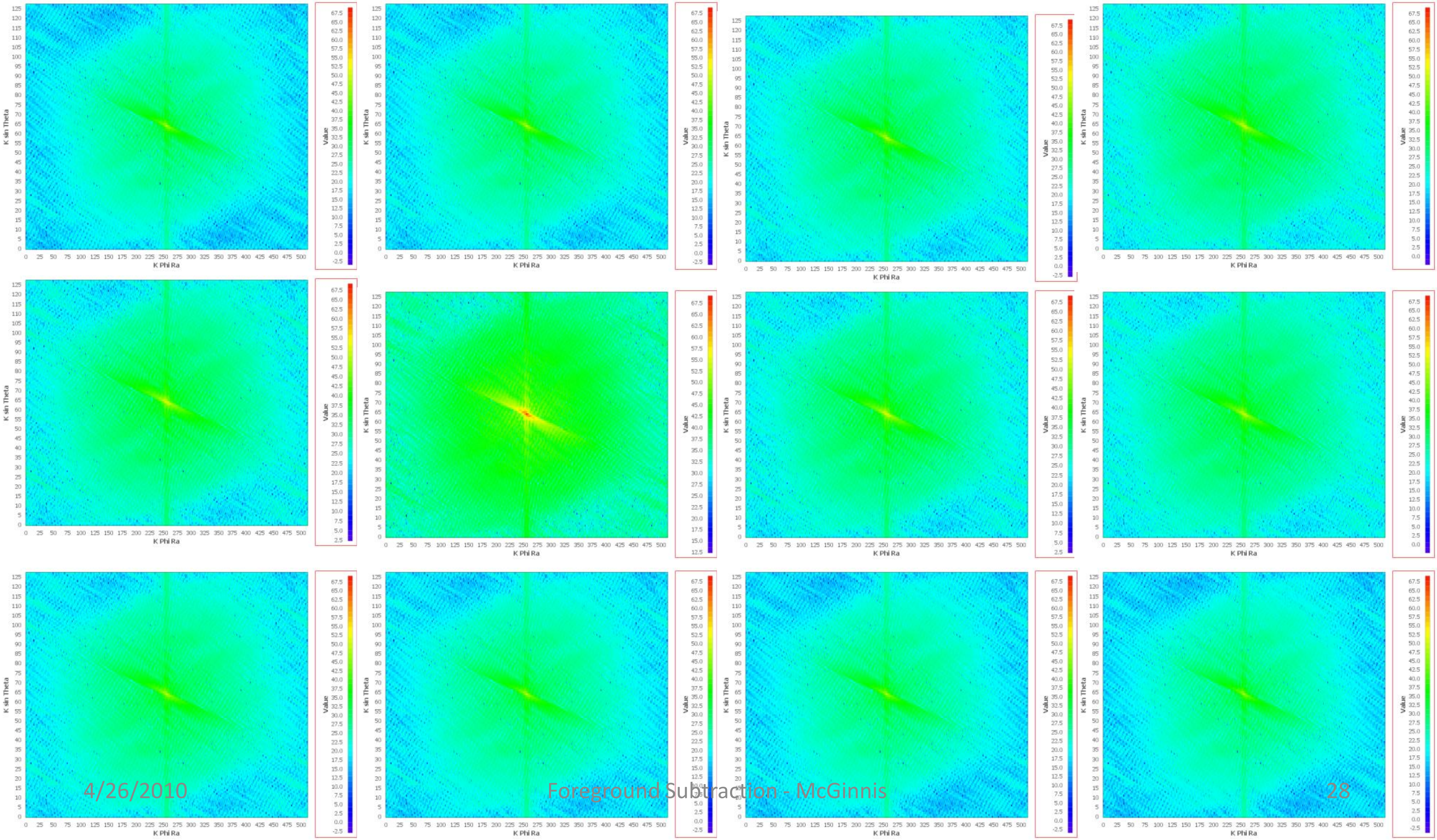
BAO First Peak from 500-1300MHz; Kperp at “k| | = 0 ”; ResBW = 1/128 MHz



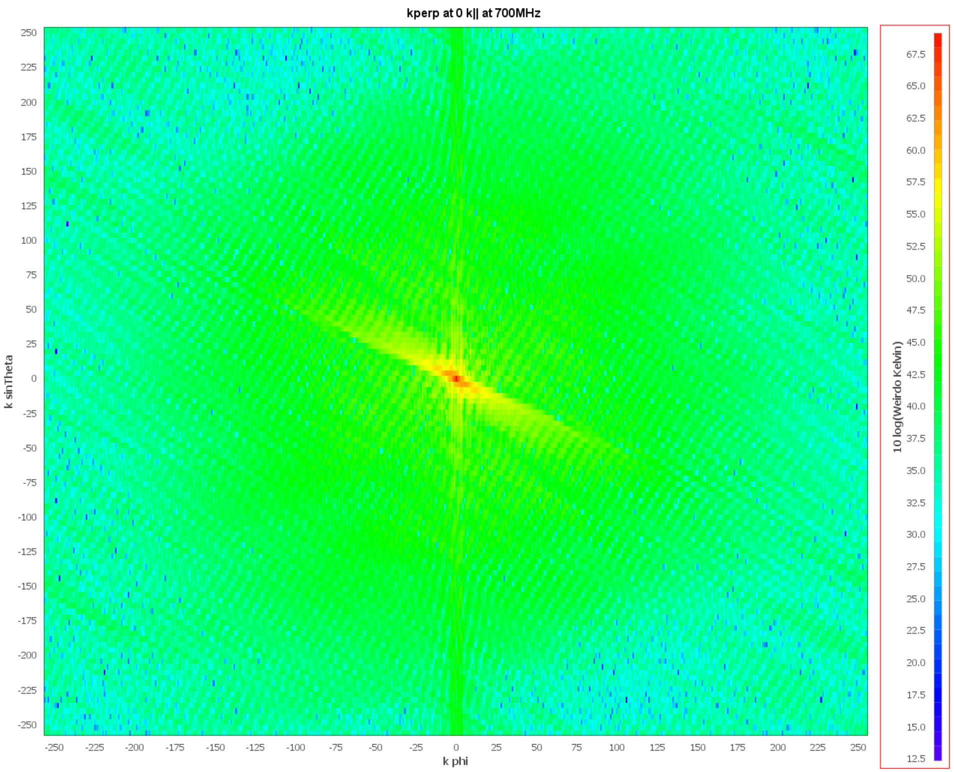
BAO First Peak from 500-1300MHz; kperp vs “k| |” ;ResBW = 1/128 MHz



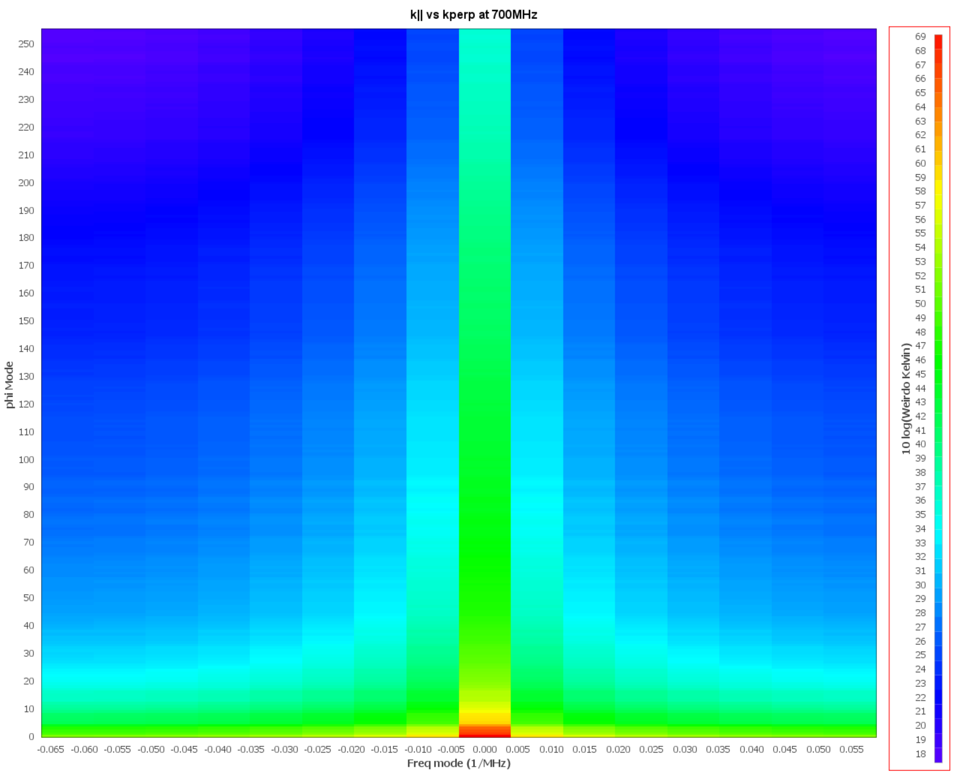
BAO + Smooth Sky
ResBW = 1/128 MHz



BAO + Smooth Sky at 700 MHz
ResBW = 1/128 MHz



kperp at k| | = 0 slice

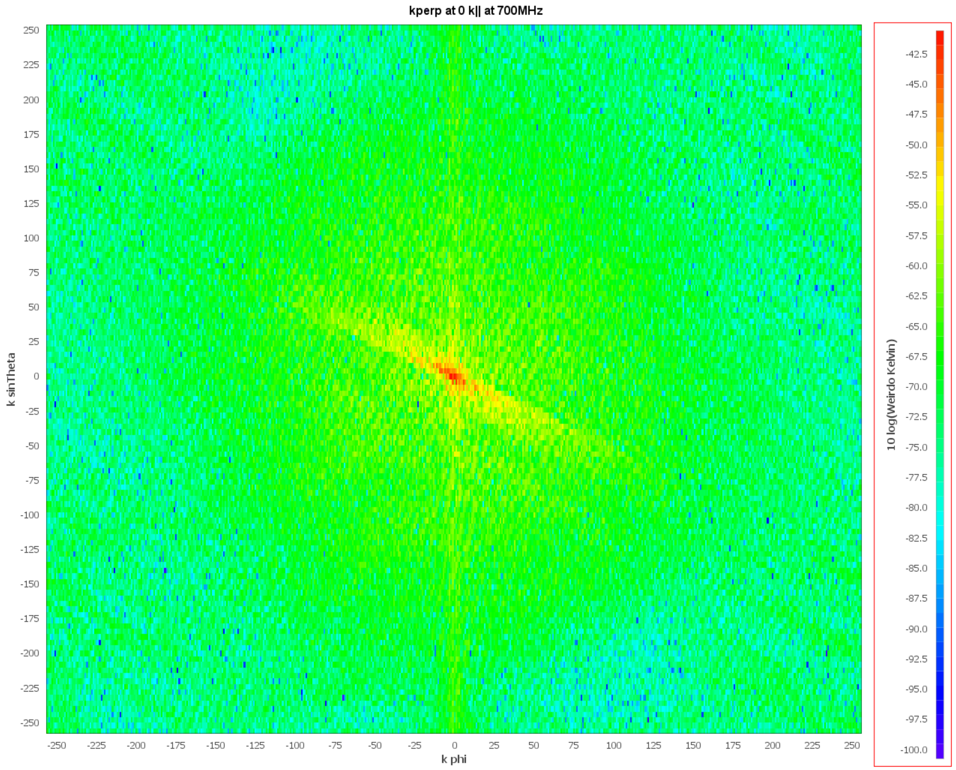


kperp vs “k| |”

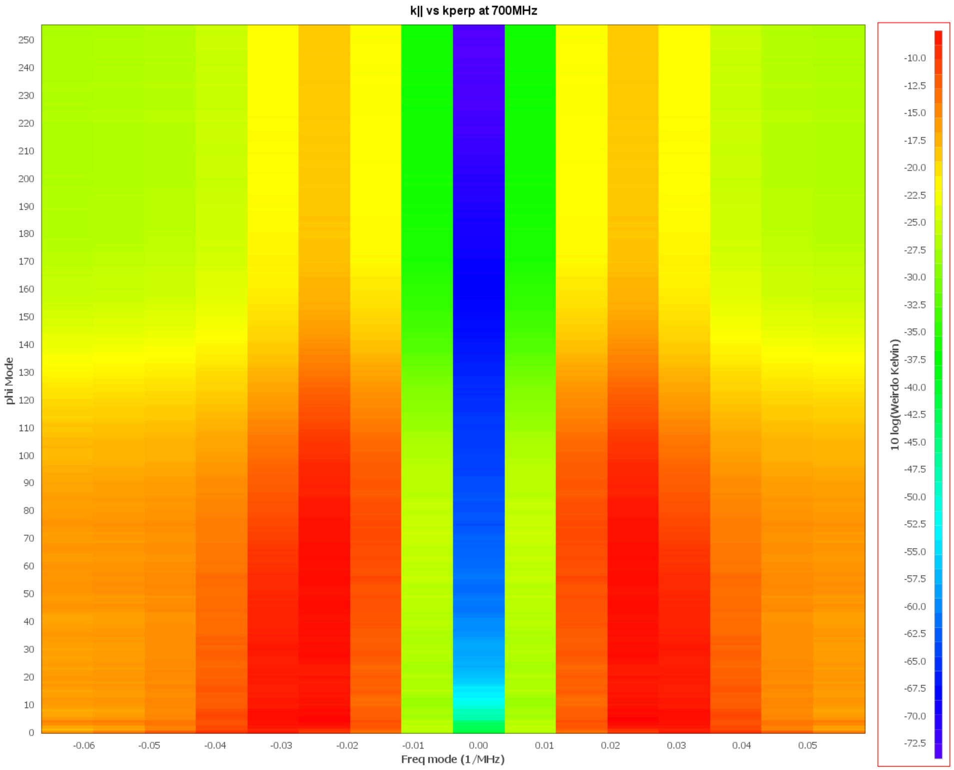
Smoothed Sky Algorithm in Reconstructed K-space

- Work in reconstructed sky transverse kspace
  - Removes 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 smoothed kspace from raw kspace
- Fourier transform along frequency axis
- Look the transverse kspace slices at high k|| mode number.

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



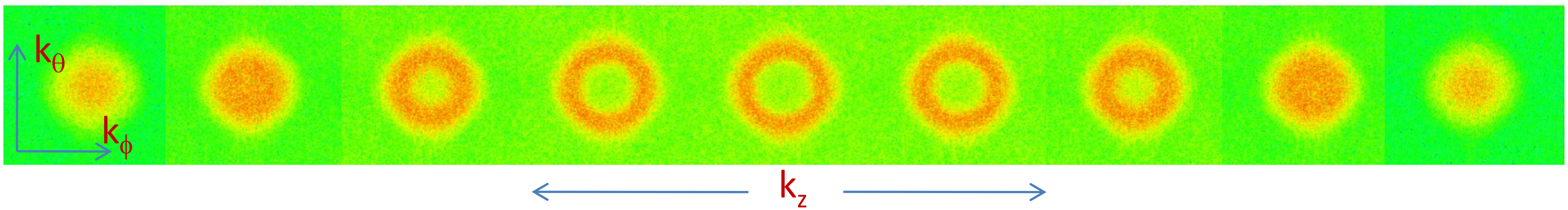
kperp at k| | = 0 slice



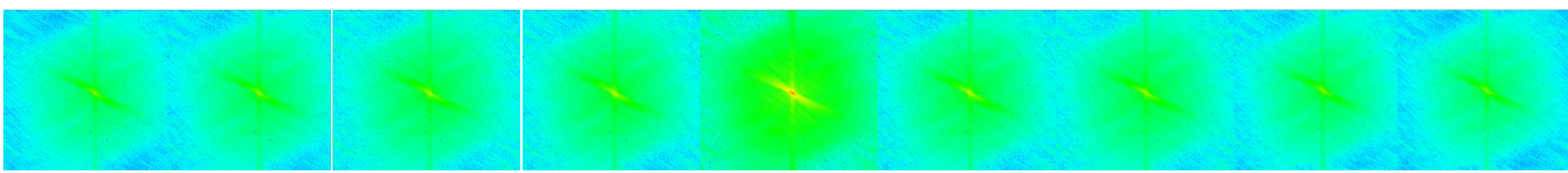
kperp vs “k| |”

Foreground Removal (Fermilab)

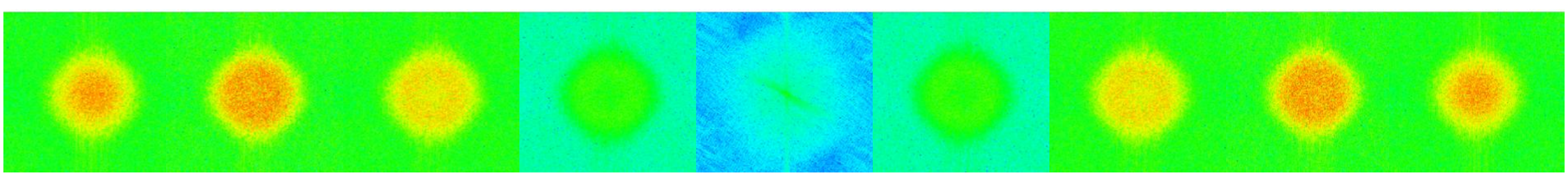
BAO First Peak in 3-D k-Space (Gnedin)



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
  - Smoothed sky subtraction algorithm on the raw data cube
  - 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

- Add 2nd and 3rd BAO peaks
- Try "smooth" cuts of large foregrounds
- Try pattern recognition of BAO sphere
- Examine the effects of noise
- Examine the effects of calibration errors