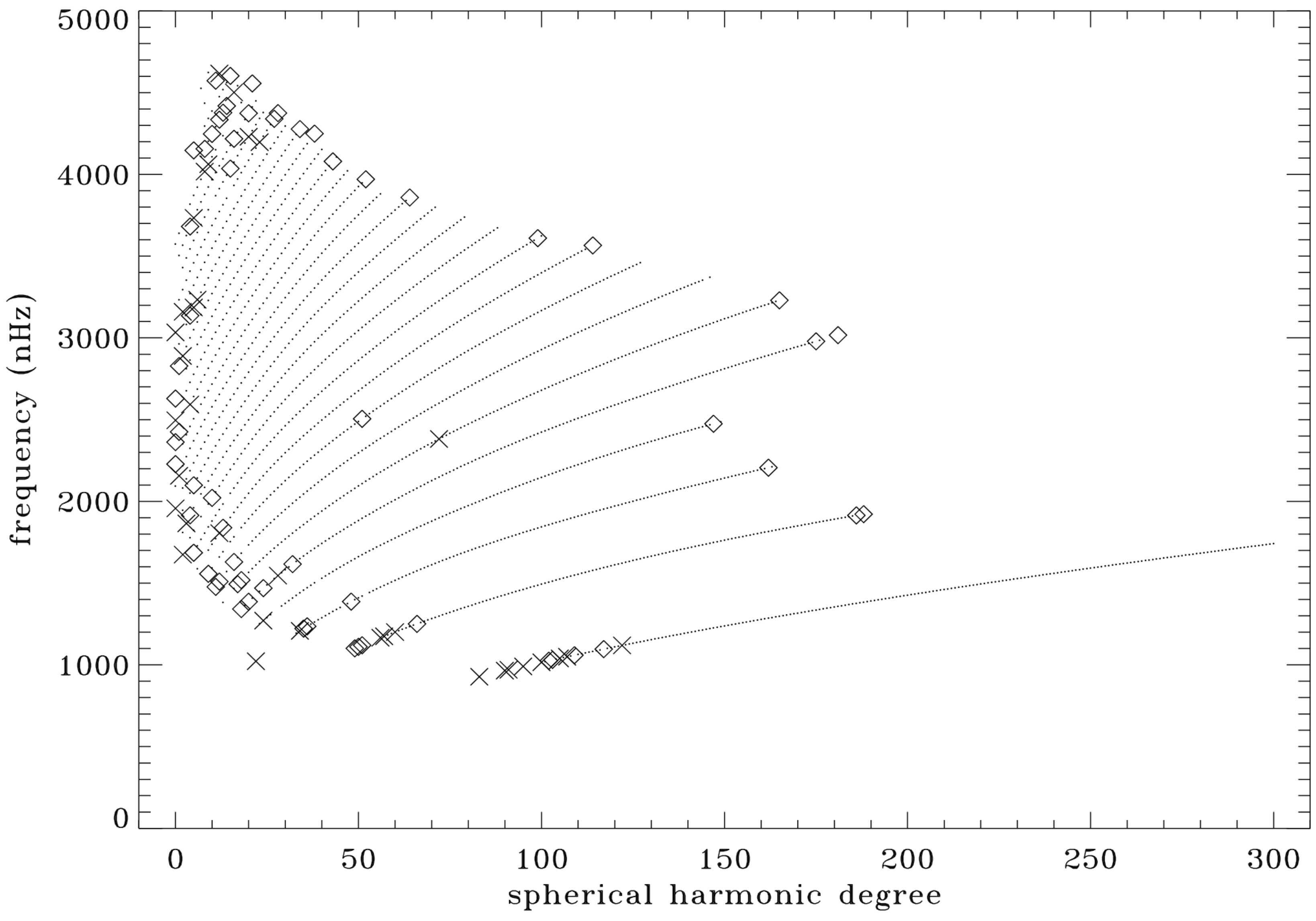
Extending the Medium-l Program to HMI

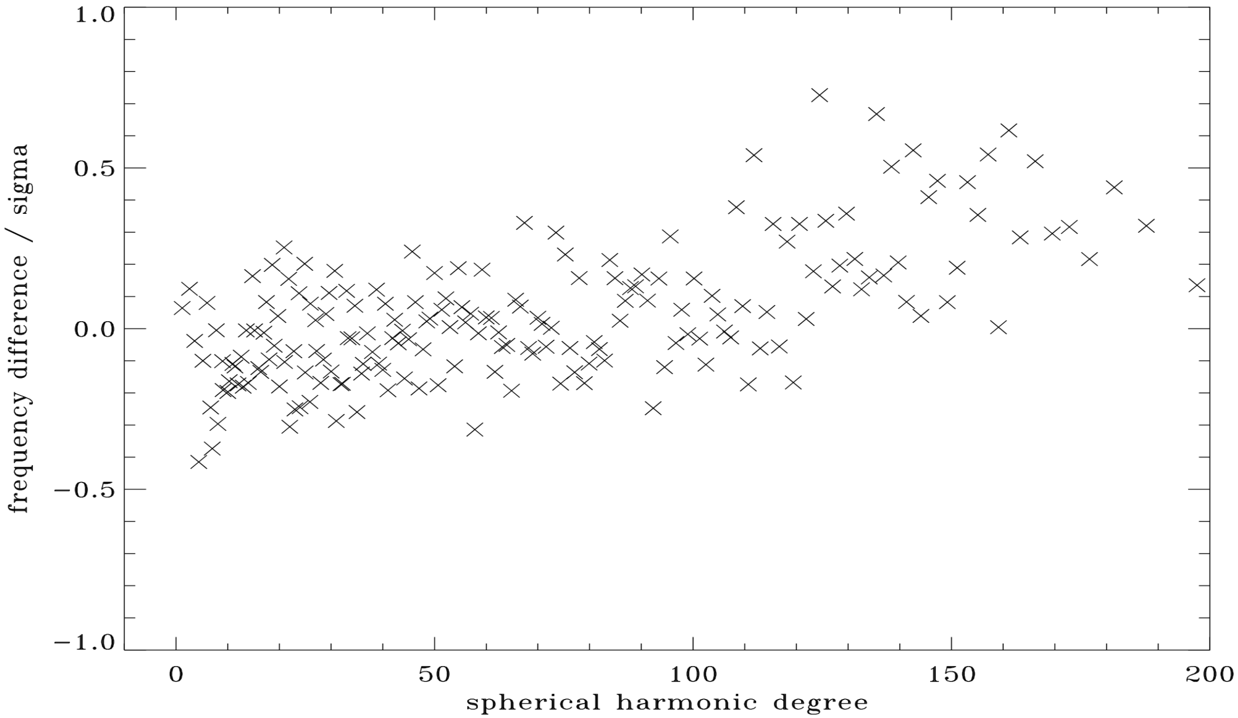
Tim Larson, Jesper Schou
Stanford University

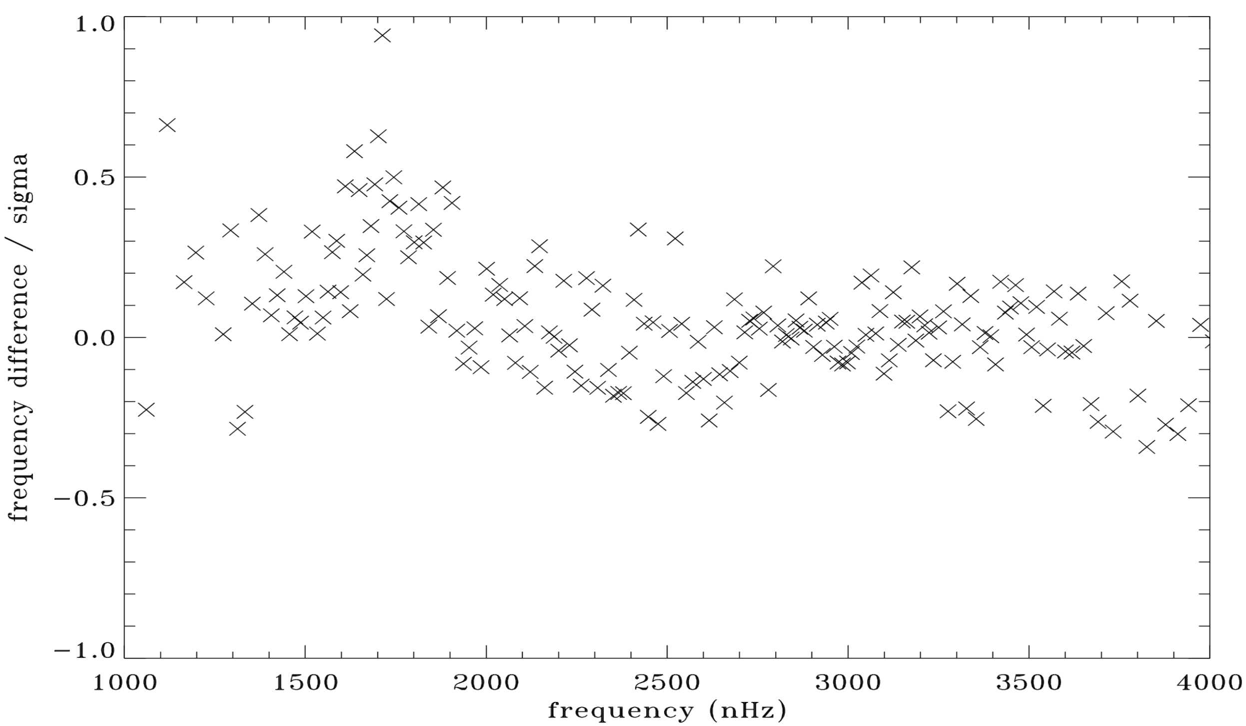
tplarson@sun.stanford.edu

As we approach a full year of regular observations from HMI, the MDI project draws to a close. In this poster we discuss a continuation of the MDI Medium-l Program using data from HMI. While agreement between the two instruments is generally quite good, HMI provides an opportunity to finally unravel some of the systematic errors we found in the analysis of MDI data. To that end, we recompute the leakage matrices with different resolutions, apodizations, and point spread functions and compare the resulting mode parameters obtained during the last MDI Dynamics run with contemporaneous results from HMI.

The l-nu diagram below shows the mode coverage for the last MDI dynamics run. Dots represent modes fitted in both MDI and HMI data, X's were fitted only for MDI, and diamonds were fitted only for HMI.







The plots to the left show frequency differences between HMI and MDI in units of standard deviation. The sense of subtraction is HMI – MDI. Points have been binned by a factor of 10 for clarity.

The top plot seems to show a positive slope with a zero crossing in the middle, implying that for low l HMI systematically sees lower frequencies than MDI and for high l it sees higher frequencies.

Does the bottom plot have a feature around 1700 nHz? We do not yet know its origin.

Overall, however, agreement is quite good. Out of 1988 common modes, only 6 differed by more than 2 sigma. 122 modes differed by more than 1 sigma.

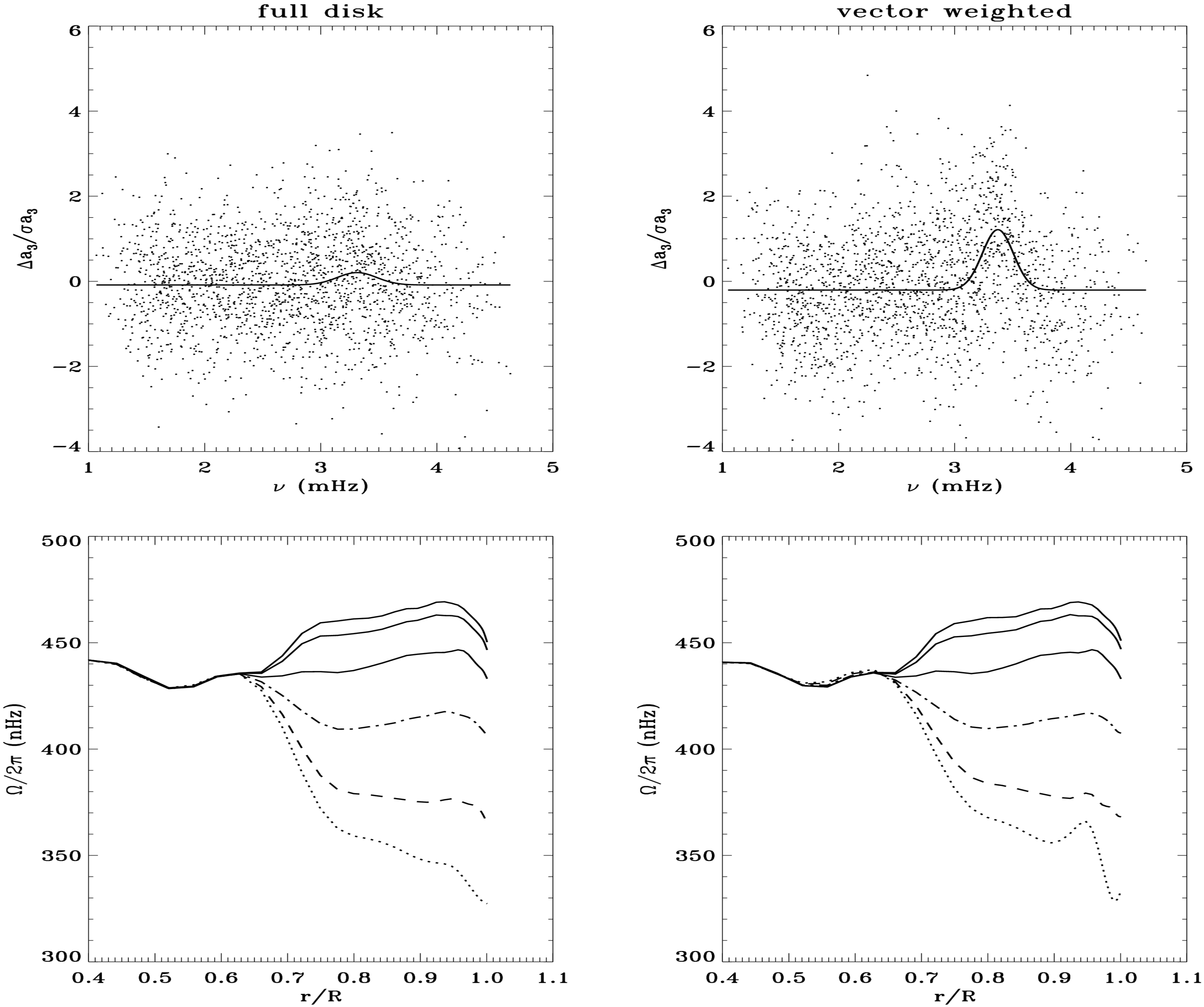
So what are we worried about? The comparison plots above are all comparing HMI with MDI full disk data. Unfortunately, the usual input to the MDI Medium-l Program is not full disk data but rather vector weighted data, so called because it has been convolved with a gaussian "vector". It is also subsampled by a factor of 5 and highly apodized. Although the full disk data is generally of higher quality, it is only available a few months out of the year. Because we have the vector weighted data for almost the entirety of the MDI mission, that is the dataset that we would like to have continuity with HMI.

Unfortunately, and as the plots below show, we are not even able to bring the two MDI datasets into agreement with each other. The top plots below show the normalized residuals for one of the a-coefficients. If the model is a good fit to the data, we would expect for these to be normally distributed around zero. The feature seen in the vector weighted data at 3.4 mHz is an unexplained deviation from this, but it is almost completely absent in the full disk data.

The bottom plots below show rotation profiles obtained from RLS inversions. Again, the vector weighted data show a spurious feature, the polar jet. And again, it is absent in the full disk data.

Our previous investigations have revealed that both of these features depend more strongly on the apodization of the data rather than its resolution. We therefore began to suspect that there could be errors in the leakage matrix.

The Problems



How to Make a Leakage Matrix

We begin by generating fake spherical harmonic images with a given P-angle, B-angle, observer distance, and CCD offsets. For all of our previous work, a value of zero was used for all of these. After projecting onto the line of sight, we have the option of convolving the image with a point spread function (PSF). If we are calculating the images at high enough resolution, we can also bin them during this step. Because the PSF of MDI is not well known, the full disk images are left as they are. For vector weighted images we convolve with a gaussian, but in the past we have not subsampled the images. Rather we have simulated the effect of the interpolation done in the spherical harmonic transform by convolving with the cubic convolution kernel used in the interpolation. The reasons for doing this will be made clearer below.

Once we have the fake images, they are run through a spherical harmonic decomposition using exactly the same pipeline as used for real data, which is to say they are remapped to a regular grid in longitude and sin(latitude), apodized, and then an inner product is taken between the result and a target spherical harmonic mask. The results have only to be retabulated to yield the leakage matrix. Because of the computational burden, we usually only calculate a subset of all the leaks in this manner, and interpolate the rest.

This process must be done separately for the vertical and horizontal components of the leakage matrix, and in principle for the real and imaginary part of each one of these. One might think that we would have to calculate how the real parts leak into the imaginary parts and vice verse, but because of our assumed geometry these leaks are all identically zero.

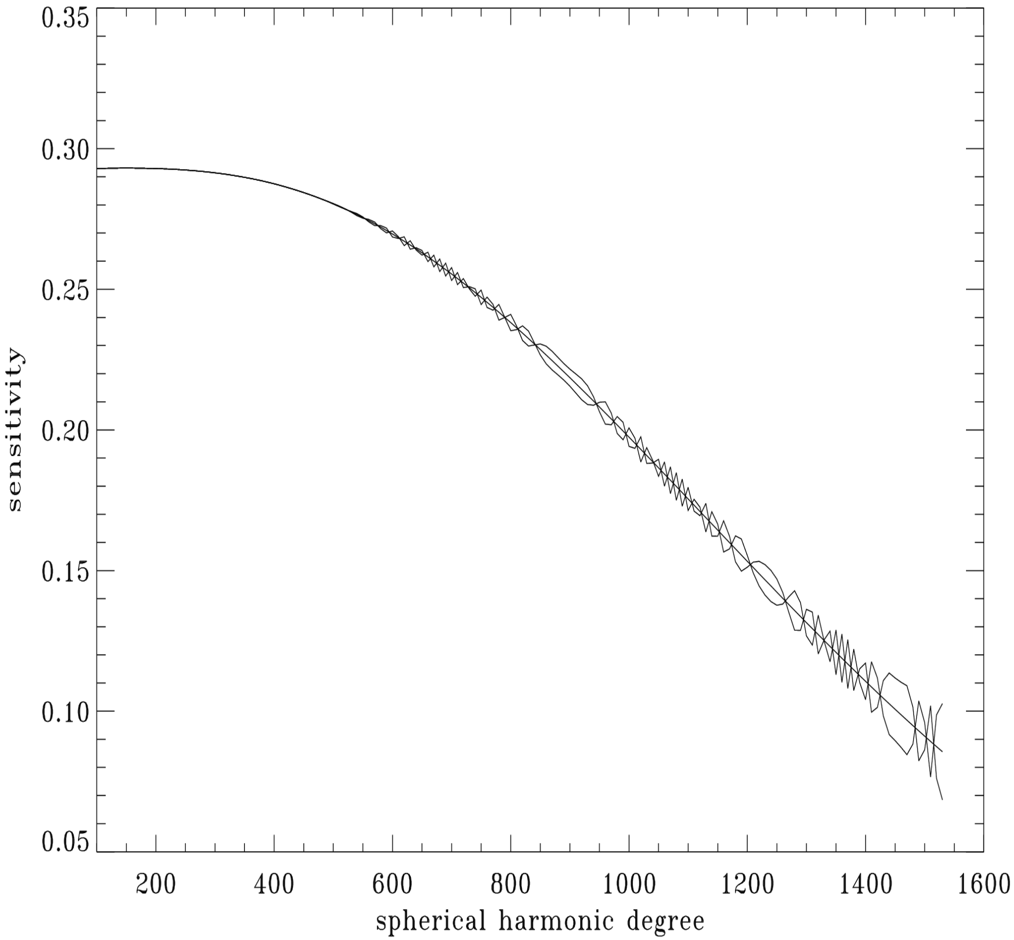
Variations

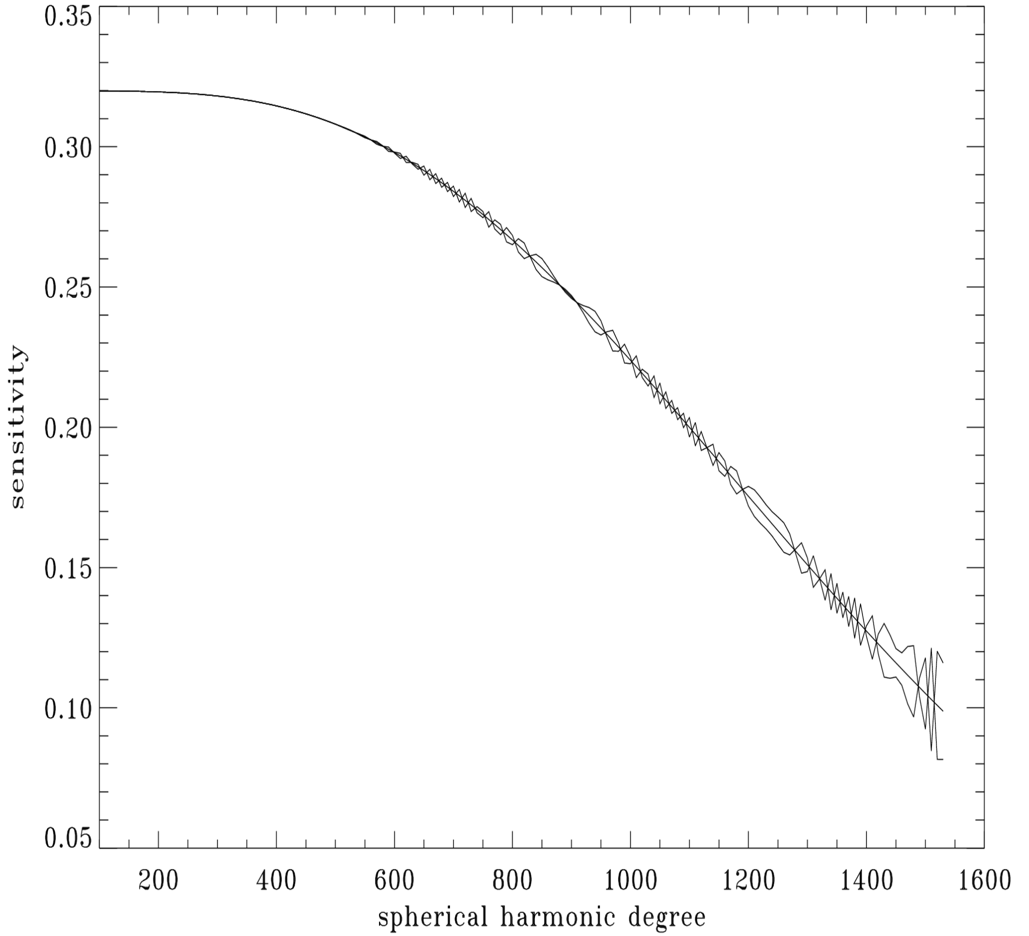
To investigate the effect of our various assumptions, we generated many new full disk leakage matrices (resolution 1024x1024). For the sake of brevity, all the results shown below are for the real part of the vertical component only. Furthermore, we show only the values for $\Delta l=\Delta m=0$, in other words the sensitivity to the target mode. The leaks were not interpolated.

We first tried varying the x and y offsets of the CCD. The result is shown below. As one can see, the effect is negligible in the medium-l range. As l increases, we violate the Nyquist theorem worse and worse, especially near the limb. The effect is that the leaks calculated have an increased sensitivity to how the pixels line up with the fake spherical harmonics. But we know that the pixel coordinates of image center varies from image to image, so at the outset it would seem impossible to calculate meaningful leaks at high l.

This is why the vector weighted leakage matrix has been made using full resolution images, because with subsampled images the Nyquist theorem is violated at much lower values of l. On the other hand, we can in principal improve the high l leaks by calculating the fake images at higher resolution.

Left panel shows the effect of offsets in x0 for m=l. Right panel shows the effect of offsets in y0 for m=0. Plotted are 0 offset, 0.5 pixel offset, and their average.



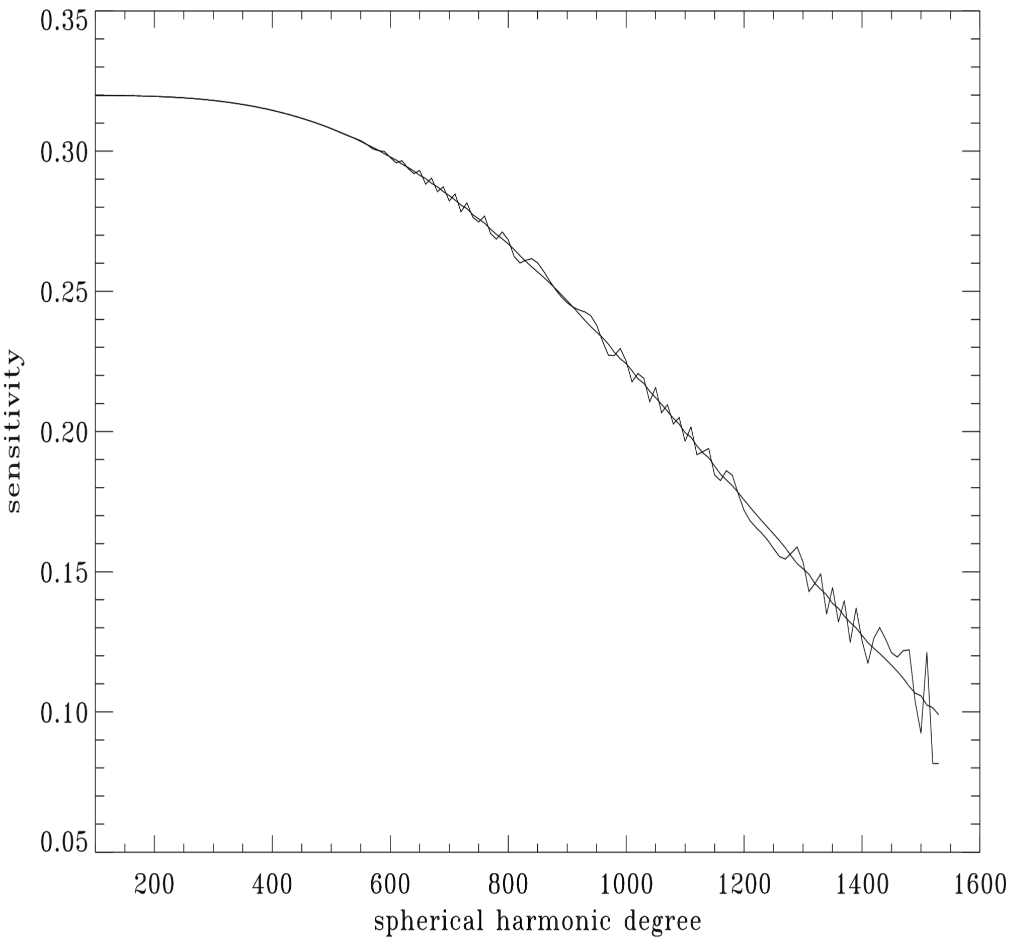


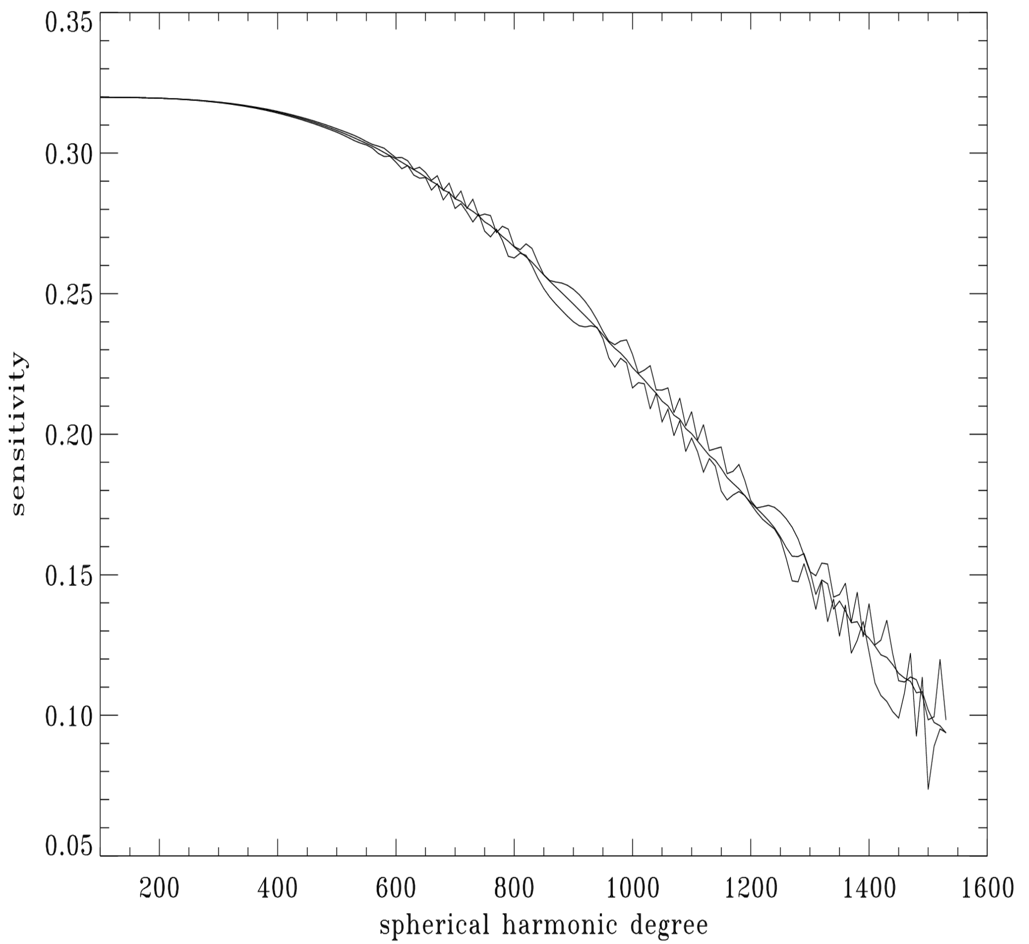
The best one might hope for would be to calculate the average over the possible values of x0 and y0. As it turns out, the variation of the leakage matrix with pixel offset seems to vary sinusoidally with a period of 1 pixel. Therefore the average of the sinusoid can be found by averaging the values at 0 pixel and 0.5 pixel offsets, as shown above. In the future, we hope to use a leakage matrix averaged in this fashion.

We also tried varying the P-angle and observer distance. As we increased the P-angle, the variation at high l became more smooth. Observer distance is different from the parameters discussed so far, in that it varies in a known way with time. (The B-angle also has this property, but have not yet tried varying it.) So in principle we could have a time-varying leakage matrix as well to account for it, although the best we could do would be to take the average over the time period we are analyzing. Since observer distance varies by ±1.7%, the average shown in the plot below is over almost the whole range.

Left panel shows effect of changing the p-angle for m=0. Plotted are 0 degrees (as in right plot above) and 0.5 degrees. The latter is smooth with no averaging. The effect on m=l was negligible.

Right panel shows the effect of changing the observer distance, or equivalently the radius of the solar image, for m=0. Plotted are 0.984 AU (top line), 1.016 AU (bottom line), and an average over these and 7 points in between. The average is smooth only until about l=1300. The effect on m=l was similar.





Discussion and Future Work

We also tried computing the fake images at 4096x4096 resolution and then rebinning them by a factor of 4. The leakge matrix remained smooth to only slightly higher l, and the sensitivities were slightly reduced. When convolving with a gaussian, the resulting leakage matrix is smooth, but the calculated sensitivities fall to zero near l=300, as expected.

There are still many things to try, most notably varying the B-angle. We also hope to create fake images near the normal vector weighted resolution and average together the leaks for different pixel offsets. In principle this should result in better fits to the data. We have also developed the ability to create a facsimile of the MDI medium-l data from HMI data. There is still the effect of the PSF to investigate. Lastly, we need to vary the apodization in order to find a clue as to the origin of the systematic errors mentioned above.