

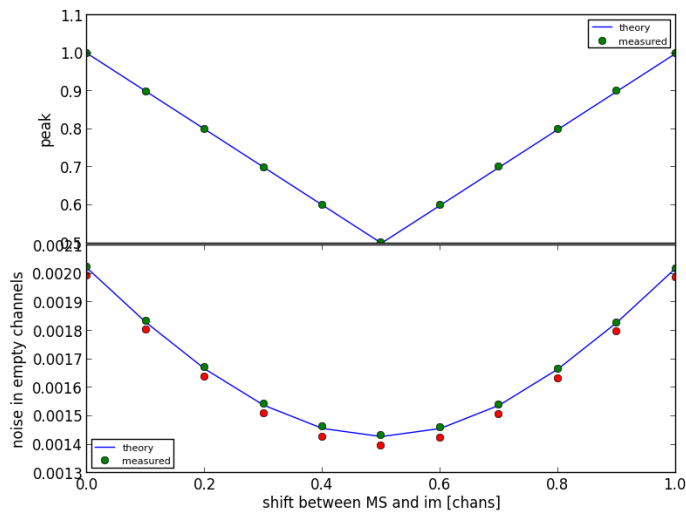
Correlated and uncorrelated Noise in Visibilities and Images

This document is intended to clarify how noise is correlated in casa when regridding operations like mstransform and tclean are used. It is a report on tests done, and the scripts used are available upon request (attached to CAS-9852 or ask Remy Indebetouw)

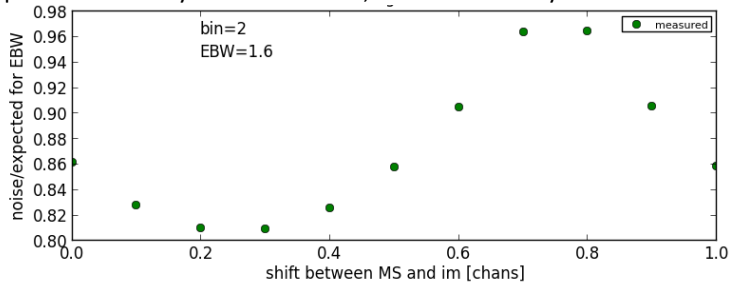
TL/DR:

If one takes independent channels, and interpolates between them, one obtains correlated channels. An output channel exactly between two input channels will have $\sqrt{2}$ less noise than the input channels, whereas an output channel precisely aligned with the input channels will have the same noise as the input channels. Thus, noise in an image can vary from channel to channel, in the worst case between the expected noise level and $1/\sqrt{2}$ of that level.

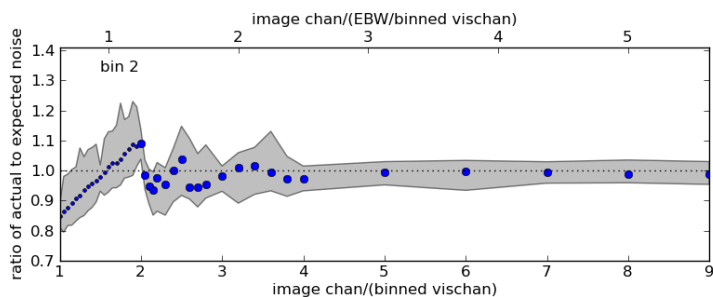
This plot shows the amplitude (top panel) and noise (bottom channel) in an image channel. The image channel is the same width as the (independent) visibility channels, but it is shifted by the amount shown on the x axis:



This plot again shows the noise in an image channel, as a function of shift between the image and visibility channels, but now for simulated ALMA data in which the visibilities have been online-averaged by 2 channels (the Cycle 5 default). That online averaging leaves the visibility channels moderately correlated, so the interpolation effects on noise are not as pronounced as for completely independent visibility channels above, but still varies by 20%.



Finally, this plot shows the per-channel noise in an image created from simulated ALMA visibilities with online averaging of 2 channels, but as a function of image channel width. The greatest variation of image channel noise is for image channels equal to the visibility channels ($x=1$), in which the noise varies between the expected (radiometer equation or sensitivity calculator) noise level and 20% lower than that. The channel-to-channel noise level variation is also high when the image channels are slightly less than twice the visibility channels – noise in the image channels in that case varies between expected and 20% higher.



Noise in a single channel (visibility or image)

In the faint-source limit, the noise in the real or imaginary part of a visibility is $\sigma = \frac{kT_s}{A\eta} \sqrt{\frac{2}{\Delta\nu\Delta\tau}}$ where T_s is the system temperature, A the antenna area, η the efficiency (antenna+correlator), $\Delta\nu$ the IF bandwidth, $\Delta\tau$ the averaging (integration) time. The noise in the visibility amplitude is not Gaussian, but in the zero-signal limit has a standard deviation of 0.655σ - see Thompson, Moran, and Swenson Fig. 6.9

The noise in an image derived from those visibilities using natural weighting is $\sigma / \sqrt{n_{\text{baselines}} n_{\text{integrations}} n_{\text{polarizations}}}$

This can be verified with a single channel synthetic observation:

```
myia=ia.newimagefromshape(outfile="zero.1chim",shape=[128,128,1,1],overwrite=True)
mycs=myia.coordsys()
mycs.setdirection(refcode="J2000",refval="10:00:00 -35.00.00",refpix=[64,64])
mycs.setrestfrequency("299.792458GHz")
mycs.setreferencepixel(type="spectral",value=0.)
mycs.setreferencevalue(type="spectral",value="299.792458GHz")
kms1=1./2.99792e5*299.792e6 # in kHz
mycs.setincrement(value="0.2arcsec 0.2arcsec "+str(kms1)+"kHz")
myia.setcoordsys(mycs.torecord())
myia.done()

simobserve(project="zero.1ch.5",skymodel="zero.1chim",totaltime="1000s",
  antennalist="alma;1arcsec",thermalnoise="",mapsize="1arcsec")
shutil.copytree("zero.1ch.5/zero.1ch.5.alma_1arcsec.ms",
  "zero.1ch.5/zero.1ch.5.alma_1arcsec.noisy.ms")
sm.openfromms("zero.1ch.5/zero.1ch.5.alma_1arcsec.noisy.ms")
sm.setnoise(mode="simplenoise",simplenoise="1Jy")
sm.corrupt()
sm.done()
visstat("zero.1ch.5/zero.1ch.5.alma_1arcsec.noisy.ms")
# 'stddev': 0.6551718813400643,
visstat("zero.1ch.5/zero.1ch.5.alma_1arcsec.noisy.ms", axis="real")
# 'rms': 0.9975540637969971,
# 'stddev': 0.9975547137370306,

tclean("zero.1ch.5/zero.1ch.5.alma_1arcsec.noisy.ms",
  imagename="zero.1ch.5/zero.1ch.5.noisy.mfs",specmode="mfs",imsize=128,
  cell="0.2arcsec",outframe="LSRK",restfreq="299.792458GHz",niter=0)
# rms noise in the image ~ 0.00195, expected = 1/sqrt(49*50/2 *100 *2) = 0.002
```

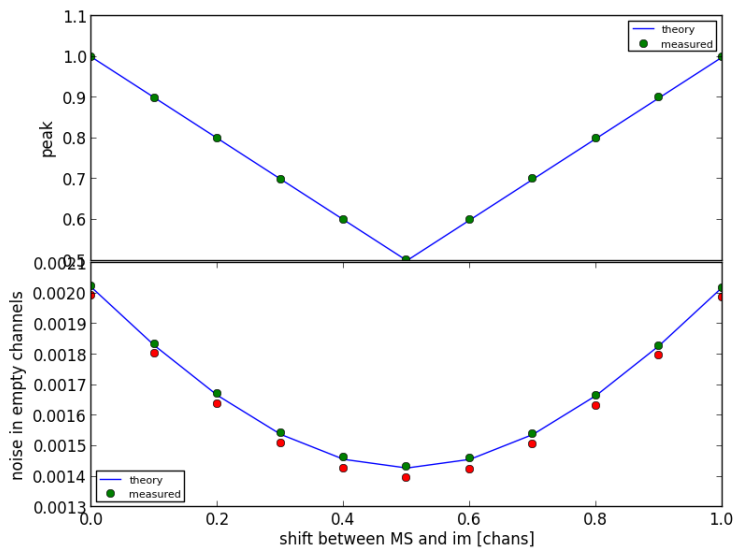
In a cube however, the noise in an image can be affected by interpolation between independent channels.

Case 1: image and visibility channels of the same width, but shifted:

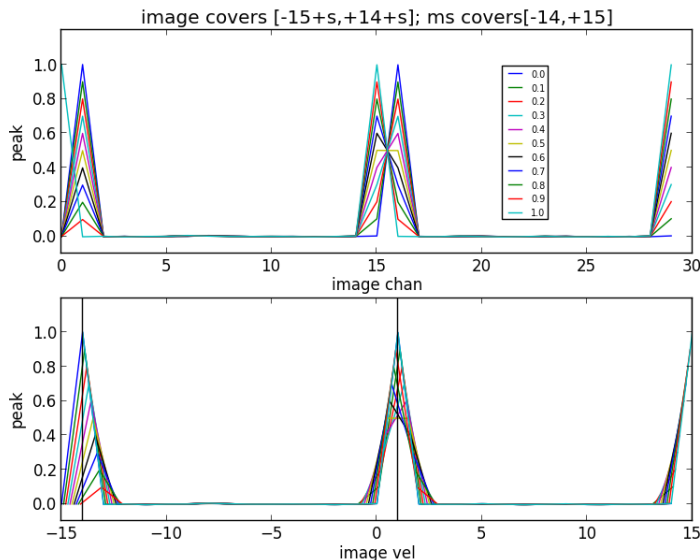
A simulated signal-free MS created in LSRK with 30 1km/s channels, and 1Jy of simplenoise added to the real and imaginary parts of the visibility as in the code above, then tcleaned into an image with 1km/s channels in LSRK, should have no interpolation in between the visibilities and image. Indeed, in this case, the noise per image channel is ~ 0.002 Jy/bm.

However, if the image channels are shifted relative to the visibility channels by a shift “s” expressed as a fractional channel, then each image channel is interpolated from two visibility channels. $\text{image} = s \cdot \text{vis}(i) + (1-s) \cdot \text{vis}(i+1)$. If the visibility channels are drawn from independent Normal distributions of rms σ , then the image will also be from a Normal distribution, with rms = $\sigma \sqrt{s^2 + (1-s)^2}$.

The script “shiftgrid.py” creates a simulated MS with 1km/s channels in LSRK, with a single channel, amp=1 point source in the center of the 30 channels (vel=+1km/s), and simplenoise of magnitude 1Jy added to the real and imaginary parts of the visibilities. The MS is in LSRK. Images are created in tclean with image channels shifted by successive values of “s” relative to the MS. The amplitude in the central image channel, and the noise away from the central channel, in the zero-signal parts of the cube, follow the two formula above, plotted as solid lines:

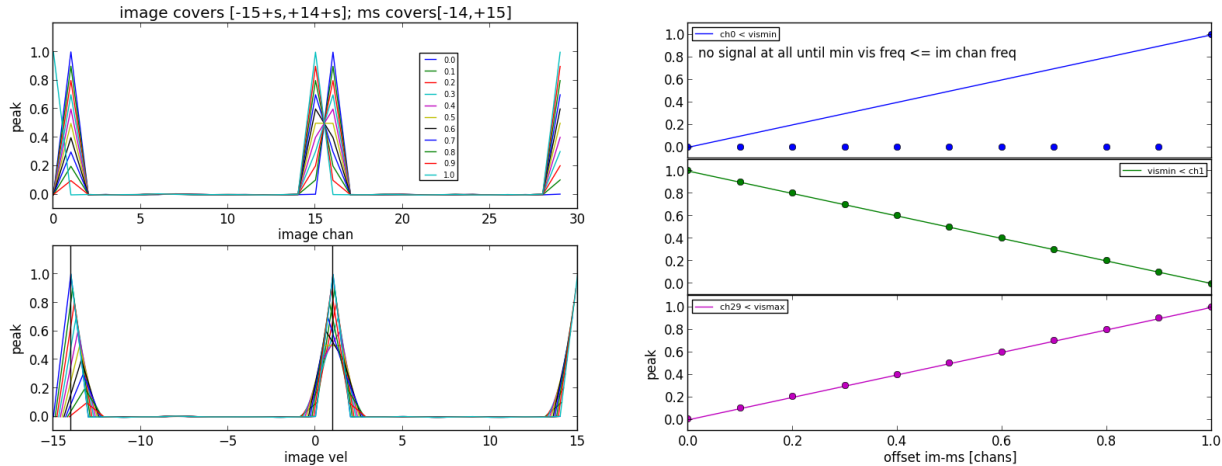


The script also puts a point source in the end channels, i.e. at -14km/s and +15km/s in the MS – those edge channels will be discussed on the next page. Spectra of the image for different shifts 0.0 to 1.0 look like this:

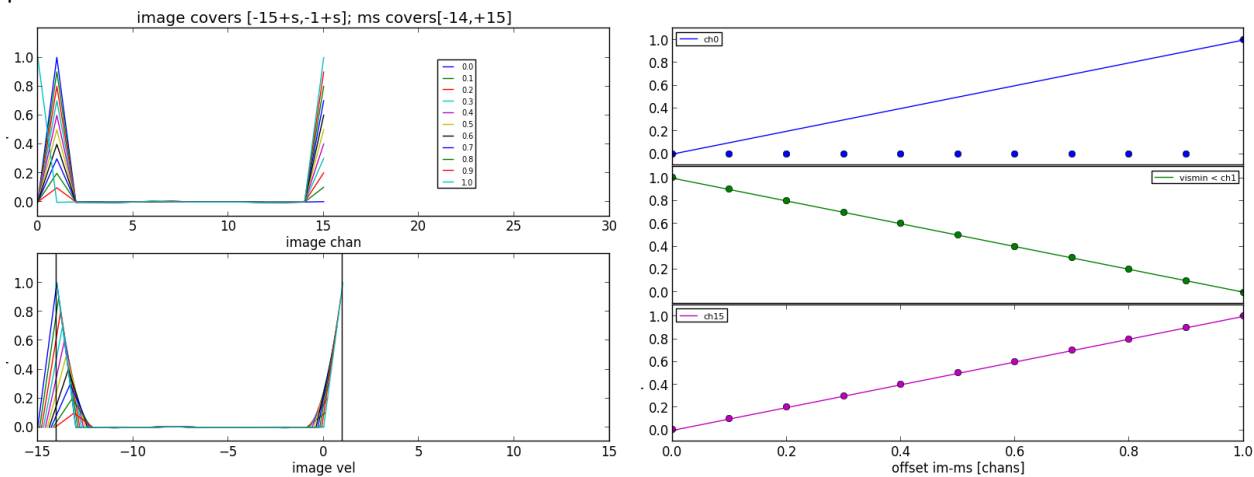


Edge Effects

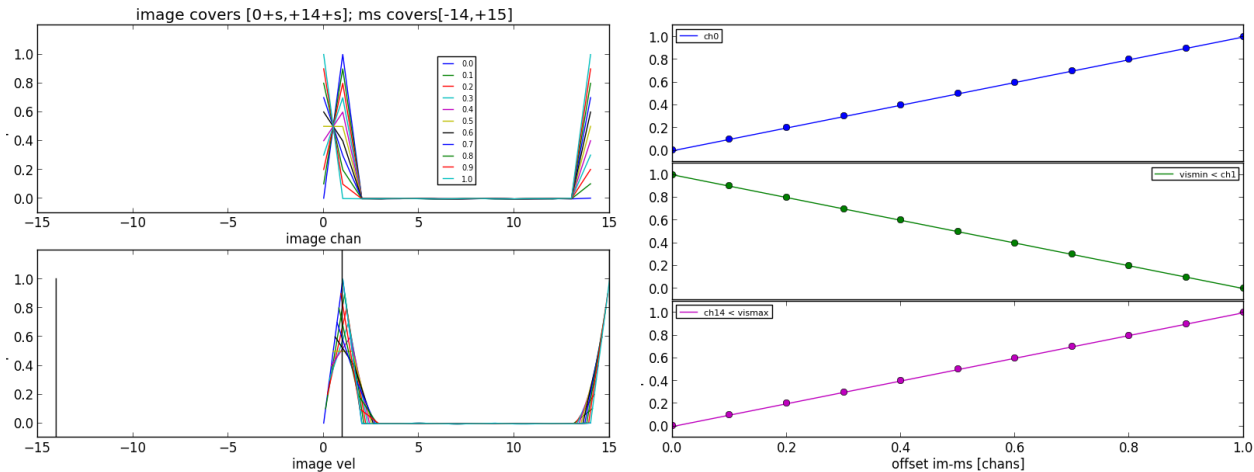
The image is requested from $-15+s$ km/s to $+14+s$ km/s, so for $s < 1$, the first image channel is at a velocity less than the lowest velocity in the MS i.e. the image extends “beyond” the visibilities, whereas the last image channel is the opposite – the visibilities extend to higher velocities than the image. In the first row, image ch1 and ch29, which have visibilities extending beyond the image channel, have the expected flux density. However, ch0 gets no flux until $s=1$, whereas it should be getting the interpolated partial flux from the MS channel zero – image ch0 should look like ch15 as a fn of s .



If one selects to image only half of the MS, ch15 behaves like ch29 did above – the asymmetric behavior is not a result of channel preselection.

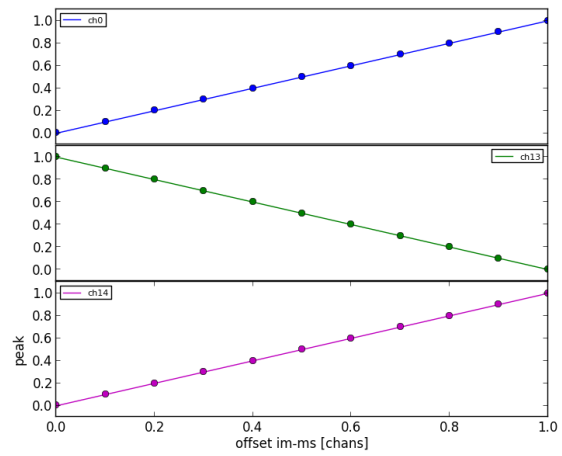
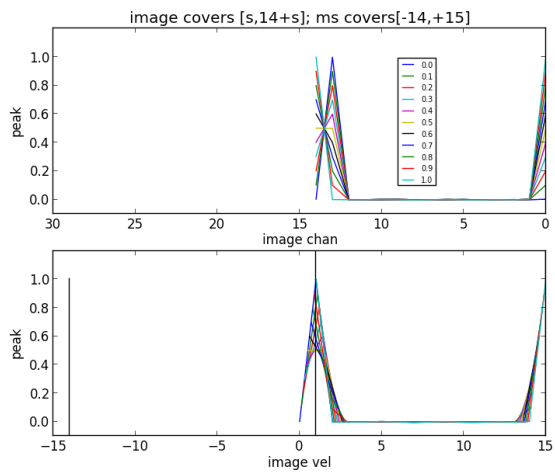
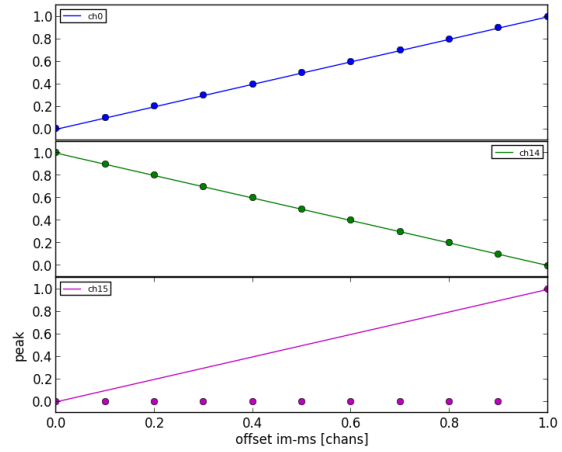
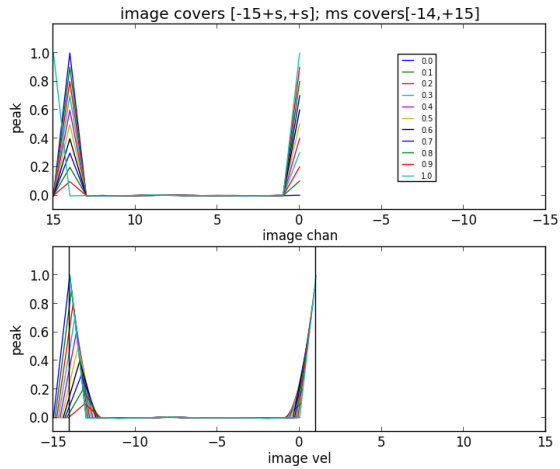


If one selects the other half of the ms, ch0 behaves as I would have expected ch0 to behave in the first example/full spw.



Since an argument could be made either way that one should or should not extrapolate past the end of the MS, CASA tclean’s behavior of not doing so is reasonable.

Asking for images increasing in frequency instead of velocity, i.e. in the same order as the MS, has similar results – everything works as expected except arguably the channel of an image which is at a higher frequency than the highest MS frequency: ch15 in the first row of images:



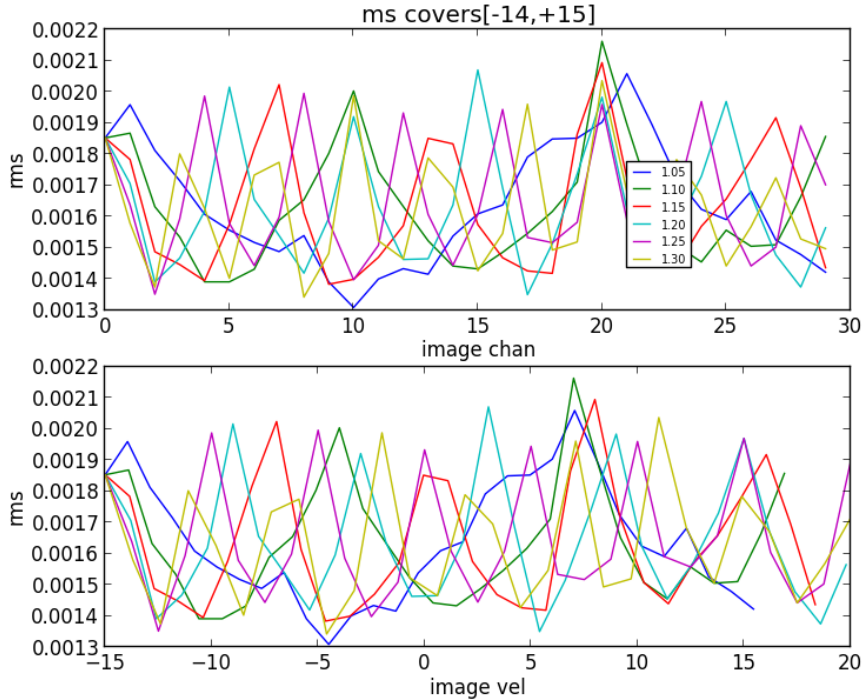
Case 2: image and visibility channels of different widths:

Now consider 1km/s visibility channels, and image channels that are slightly larger. The first image channel will contain data from the first visibility channel, and have its same noise distribution. The second image channel will be interpolated from the second and third visibility channels $\text{image_2} = s \cdot \text{vis_2} + (1-s) \cdot \text{vis_3}$, noise in $\text{im_2} = (\text{noise in vis}) \cdot \sqrt{s^2 + (1-s)^2}$.

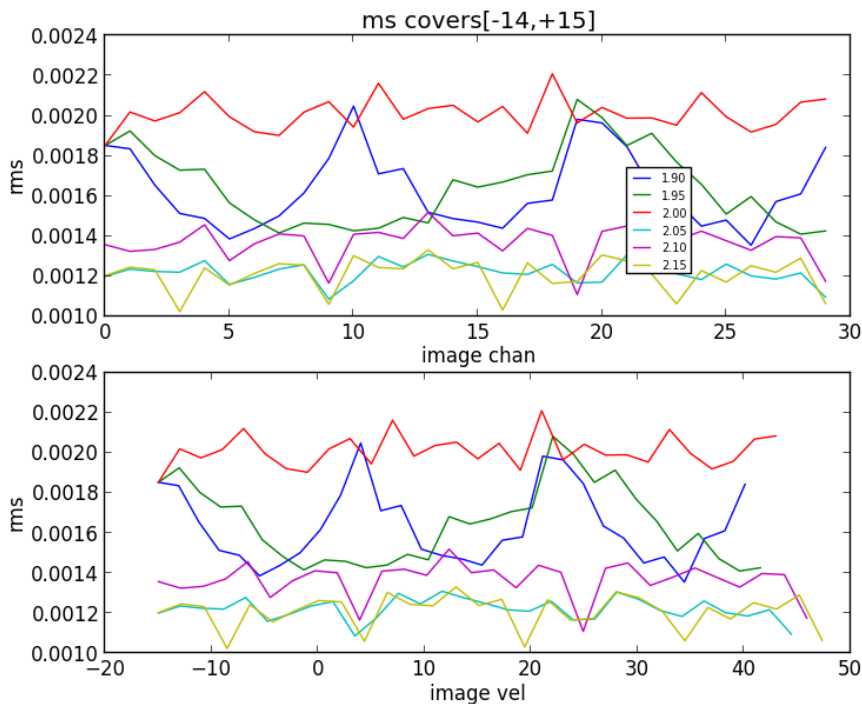
For vis channel widths w_v (1km/s here) and image channel widths w_i , $s_2 = (\frac{w_i - w_v}{w_v})$, $s_3 = 2(\frac{w_i - w_v}{w_v})$, etc.

Noise will decrease and increase again as s approaches 1.0, which occurs at the “beat frequency” between the two channel widths, i.e. after $w_i / (\frac{w_i}{w_v} - 1)$

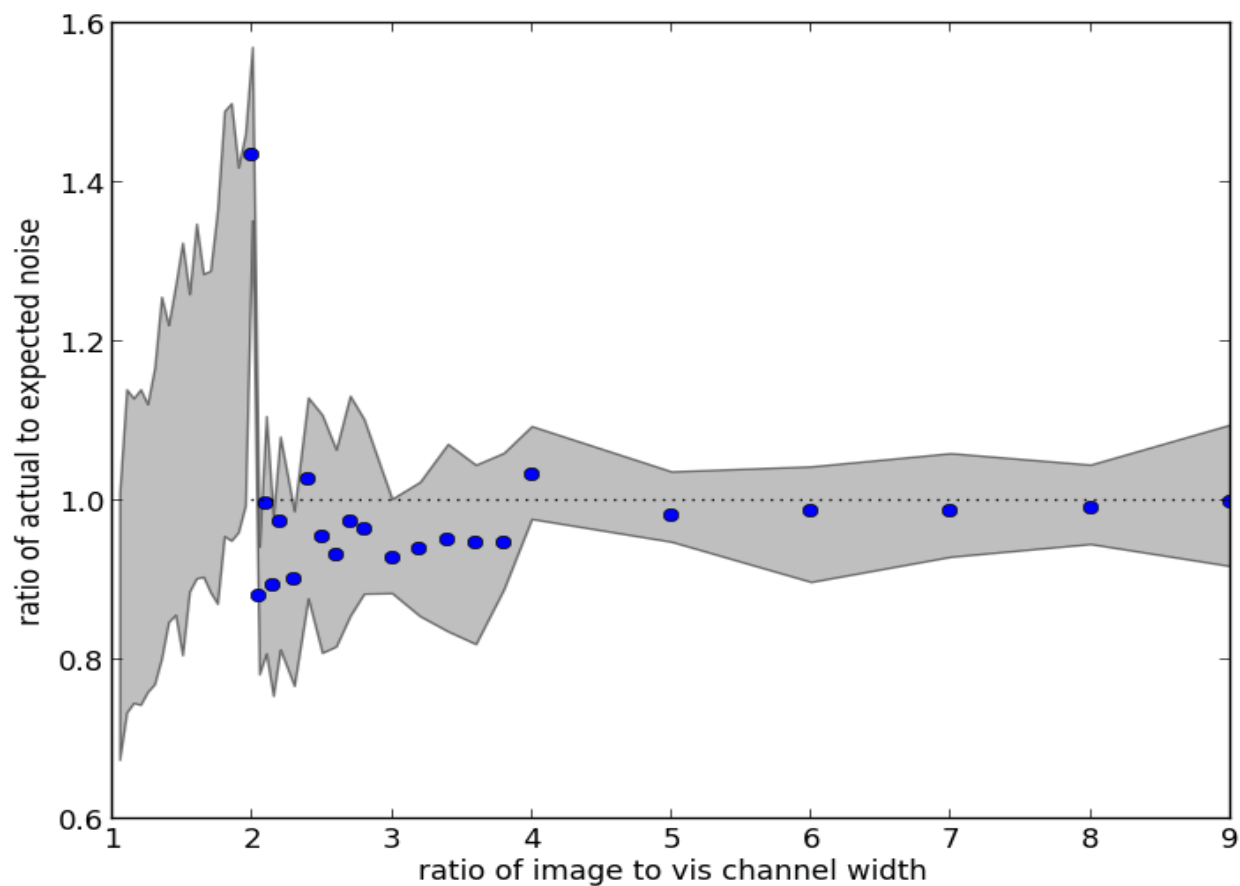
The script “growgrid.lt1.py” creates a 30ch blank MS with 1km/s LSRK channels and a series of images with 1.05, 1.1, ... km/s channels. The noise spectrum has a period of $1 / (\frac{w_i}{w_v} - 1)$ image channels, as expected:



As the image channels approach 2x vis channels, the period of the noise increases again, following $1 / (2 - \frac{w_i}{w_v})$. At exactly 2x, the rms is $\sqrt{2}$ times what it should be. At >2, the noise follows the expected $\sqrt{\text{width ratio}}$.

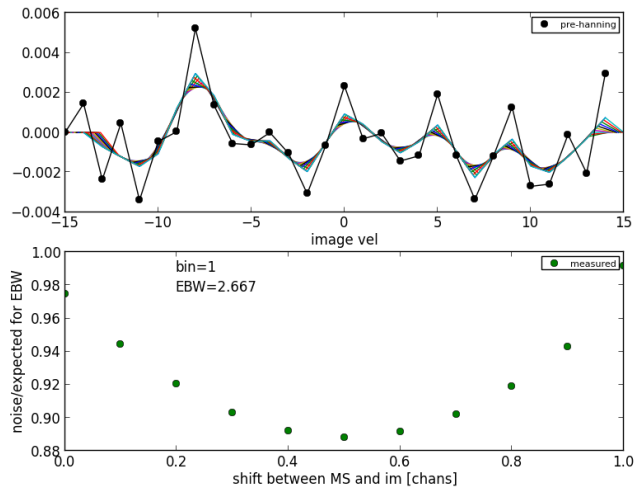


The bottom line, for completely independent visibility channels: The worst case is binning to image channels 2x the visibility channels, in which case the noise will vary from channel to channel in a “scallop” pattern, between the expected noise and $\sqrt{2}$ higher than expected noise:

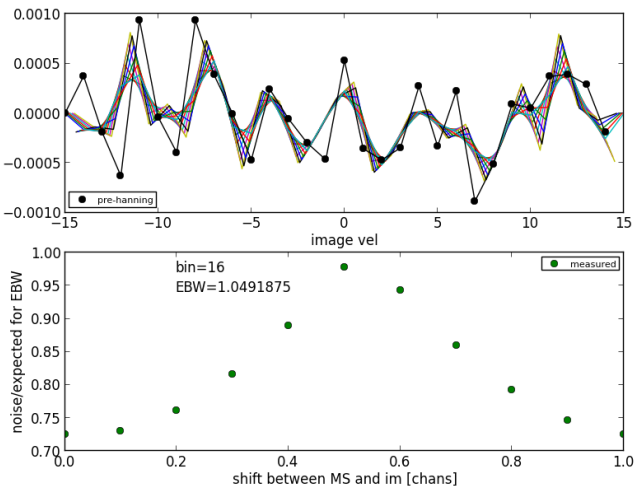
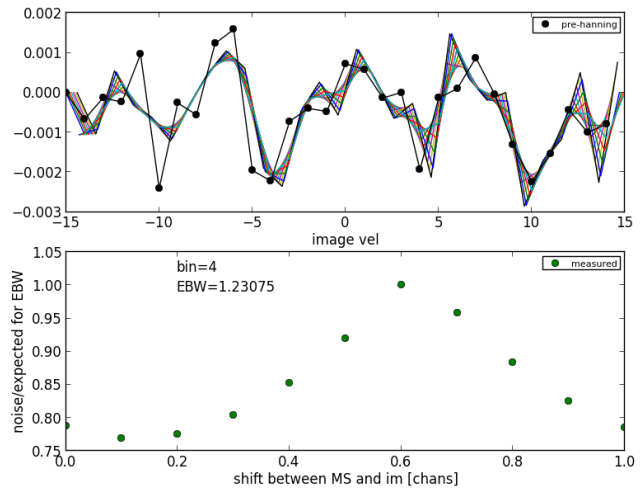
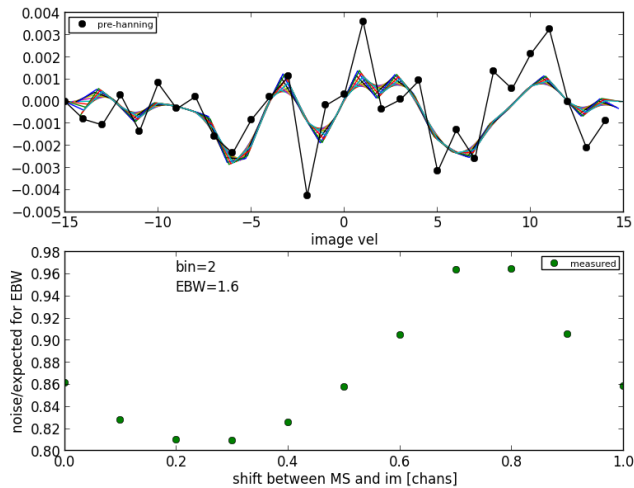


Synthetic ALMA data (visibility channels not independent)

As described in detail in Richard Hills' memo, ALMA data is tapered in the time domain to reduce ringing in the spectral domain. Currently ALMA does what amounts to Hanning smoothing in the spectral domain. Individual visibility channels are highly correlated, with effective bandwidth = $2.667 \times$ channel spacing. Clearly this will decrease the interpolation effects described above. To simulate it, I simulated independent-noise channels, applied hanning smooth, and then binned channels with `split(chanave=N)`. With no binning, 1km/s LSRK visibility channels gridded to 1km/s LSRK image channels, as a function of shift, shows the expected scallop shape in the noise(channel), but with only a 10% amplitude instead of the 40% for independent channels.



Binning the visibilities by factors of 2,4,16, and then imaging the subsequent channels to shifted channels of the same width: The magnitude of scallop noise variation increases as expected as the channels become more independent.



Finally, I imaged hanningsmoothed, optionally binned visibilities to image channels that were larger than the visibility channels and measured the noise as a function of the channel ratio.

The “expected” noise in this case is taken from CAS-8534, which is an empirical fit to Richard Hill’s calculations:

Expected noise = $N + 2.667 \cdot 0.42$ where $N = (\text{“online” binning factor}) \cdot (\text{ratio of image to vis channel} = \text{“offline” factor})$

I’ve removed the damping factor $(1-N/\text{spwchan})$ in CAS-8534 because it only makes at most a 4% difference to this result, and for these simulated data I would not expect any dependence on the spw size, since I’ve filtered in the frequency domain, not time domain.

As expected, the “scalloping” noise variation as a function of channel increases with more “online” binning.

