



# Self-calibration

Steven T. Myers

9th VLA Data Reduction Workshop



# Introduction

This talk is based on the VLA self-calibration CASA guide: [casaguides.nrao.edu/index.php?title=VLA\\_Selfcalibration\\_Tutorial](http://casaguides.nrao.edu/index.php?title=VLA_Selfcalibration_Tutorial)

Please see the *casaguides* wiki page for links to download the data, the complete set of CASA commands and a more in-depth discussion

Acknowledgement: Much of this material is from J. Marvil and the casaguide.

# Goals

- what is self-calibration (“selfcal”)?
- learn the basic selfcal procedure (for continuum data)
- understand the parameters and how to choose them
- recognize common pitfalls that cause selfcal to fail
- quantify how well selfcal has worked
- more advanced options...

Self-Calibration  
Part 1

# Theorycrafting

# What is Self-Calibration?

- Self-calibration (“selfcal”) is a procedure that uses imaging of the target source itself to provide a model for the calibration of the interferometer data
- This process is necessary in many (most?) cases where “high” dynamic range (>500:1?) is required and is limited by unmodeled residual calibration errors left over after application of the standard calibration transfer from other sources.
- The self-calibration process is in most cases identical to that used for standard calibration, with some modifications to control biases inherent to using the target itself.
- “Standard Calibration” is just self-calibration on the Calibrator source using a model, then transfer to the targets!

# What is Self-Calibration?

- The errors to be corrected by selfcal may include effects from:
  - Timescales shorter than the time between calibrator observations
  - Correction for differences in line of sight between calibrator and target
  - Effects other than delays, bandpasses, and gains (such as ionospheric dispersion, Faraday rotation, etc.)
  - Direction-dependent effects (e.g. ionosphere, primary beam, pointing error)
  - Other effects (higher-order polarization leakage, etc.)

# General Calibration Process

- In calibration, we solve for various terms in the Measurement Equation to make the data “look like” the model:
  - For CASA, this follows the convention that Model x Calibration = Data, e.g. for correlated visibilities

$$V_{\text{obs}} = \mathbf{J} \times V_{\text{model}}$$

- If data is “low” compared to prediction from model, the calibration amplitude will also be “low”
- The “Jones Matrix” J is made up of terms describing different effects, e.g. for most cases the solvable terms are

$$\mathbf{J} = \mathbf{K B G D E X} \quad (\text{these are antenna based})$$

- See Synthesis Imaging Workshop lecture on “Calibration” for more

# Limitations of Self-Calibration

- The use of Selfcal comes with some possible side-effects
- Loss of astrometry (for phase selfcal)
  - A position shift in the image plane leaves the closure phase unchanged. Astrometric positions are in general fixed to the centroid of the model emission.
- Loss of flux density scale (for amplitude selfcal)
  - Multiplication of all gain amplitudes by a single factor leaves closure amplitude unchanged. In practice a normalization of gain amplitudes is used to avoid overall flux density changes.
- Loss of fidelity (artifacts and image errors)
  - This should be minimal for a large array like the VLA with careful and deliberate selfcal procedures (see below)



Self-Calibration  
Part 2

# Practical Application

# Calibration in CASA

- 1) Set model visibilities (column: **MODEL\_DATA**) for the field being calibrated— **setjy**, **ft**, **tclean**  
(if no model is defined, a 1 Jy point source is assumed)
- 2) Compare observed visibilities (column: **DATA**) to the model— **gaincal**, **bandpass**, **polcal**, **blcal**, **fringefit**
- 3) Calibration tables are the solutions that make **DATA** more closely resemble **MODEL\_DATA**

**Both standard and self-calibration  
are consistent with this definition**

# Some Assumptions

- There is measurable “continuum” emission in your target
  - You can average the channels (per spw) to get sufficient Signal-to-Noise ratio to solve for the gain (or bandpass) terms.
- The continuum spectral windows cover the full data range
  - Helpful, though not necessary. Sometimes even the narrower windows for lines are wide enough to detect emission.
- What about bright spectral lines? (masers etc.)
  - You can solve using narrow-band data in channels with bright spectral line emission, and apply to full dataset (e.g. to detect faint continuum and lines), but you may need to adjust some things during applycal. Warning - bright lines can complicate the “continuum” also! Ask a line expert for help!

# Calibration in CASA

## Standard calibration:

- input fields are marked as calibrators
- models from setjy or default 1 Jy (with “fluxscale” rescaling)
- solutions applied to target fields

## Self calibration:

- input fields are marked as target
- models are typically from tclean
- solutions applied to same fields
- Iterative!

# When Selfcal might help

There are artifacts from the source of interest due to calibration errors

There are artifacts from a background source due to direction-independent calibration errors

Multiplicative errors encompassed by the measurement equation terms handled by the solvers

## **Typical sources of these errors:**

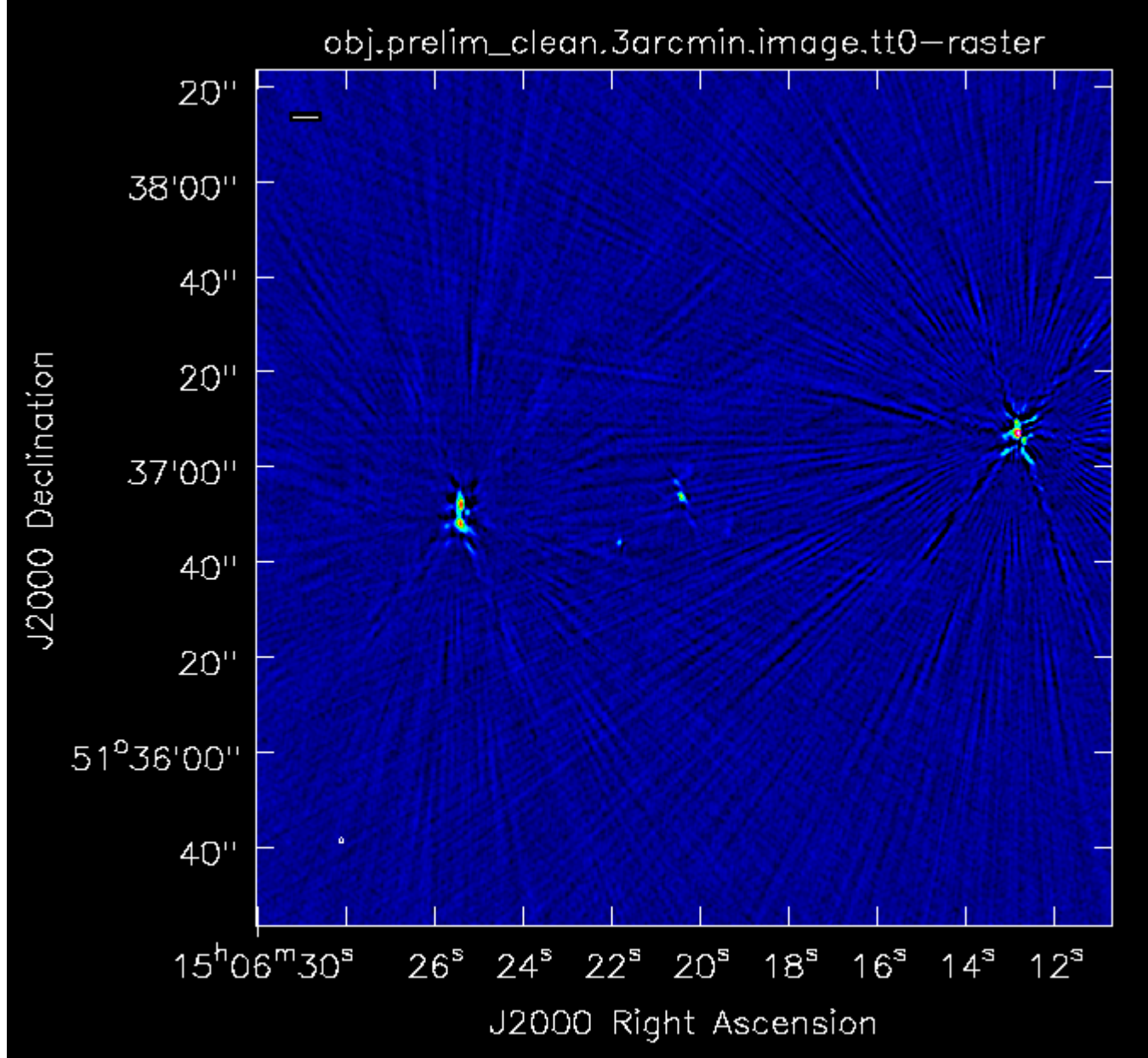
- **atmospheric density fluctuations**
- **antenna-based gain changes**

# When Selfcal will not help

- Errors in creating the image (e.g., ignoring the W-term)
- Errors in deconvolution (e.g., ignoring wide-band effects)
- Insufficient UV coverage (e.g., missing short spacings, snapshot synthesis)
- Unflagged RFI in the target visibilities
- Source(s) are too weak to achieve sufficient signal-to-noise in the self-calibration solutions
- There is a bright outlying source with artifacts due to direction-dependent errors (caveat: direction-dependent selfcal such as “peeling” can help these cases)

# The Observation

- Target is  $z \sim 1$  galaxy cluster with high radio activity
- B-configuration (project 17B-197; October 2017)
- C-band continuum (8 bit, 2 GHz, 16 SPWs)
- 3 target scans of  $\sim 10$  minutes each
- Multiple radio sources, peak flux  $\sim 7$  mJy
- Calibrated with the VLA CASA pipeline
- Strong artifacts remained after cleaning





# Phase Stability

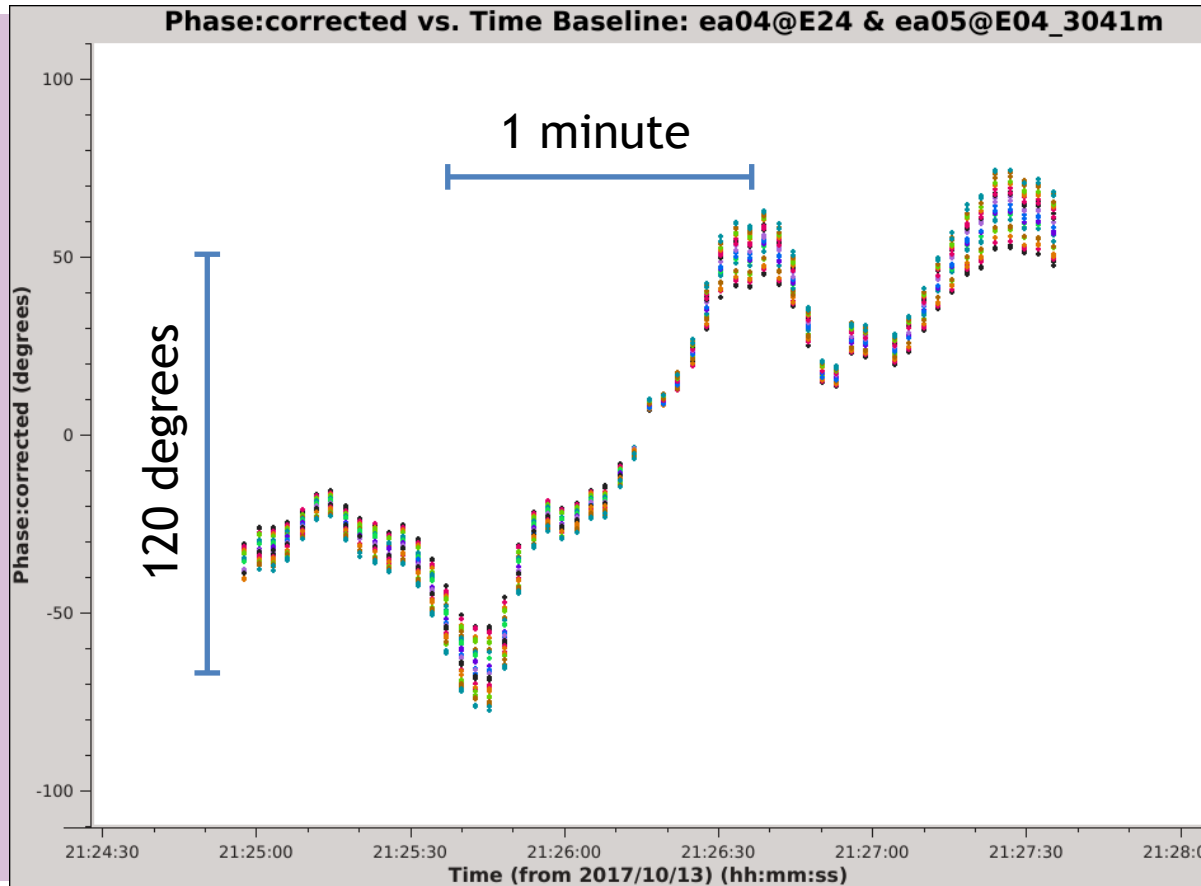
Our first goal with self-calibration will be to improve the image by correcting phase errors in the target visibilities

Before we do this, it may be insightful to learn more about the observing conditions (i.e., phase stability)

One way to do this is to identify a long scan on a bright calibrator (e.g., bandpass) and plot the phase vs. time for a single baseline

# Phase Stability

```
plotms(vis='17B-197.sb  
34290063.eb34589992.58  
039.86119096065.ms',  
xaxis='time',  
yaxis='phase',  
ydatacolumn=  
'corrected',  
field='2',  
antenna='ea05',  
correlation='RR,LL',  
avgchannel='64',  
iteraxis='baseline',  
coloraxis='spw')
```



(Target scans are 10 minutes long)

# Initial Inspection

The atmospheric phase stability is not good

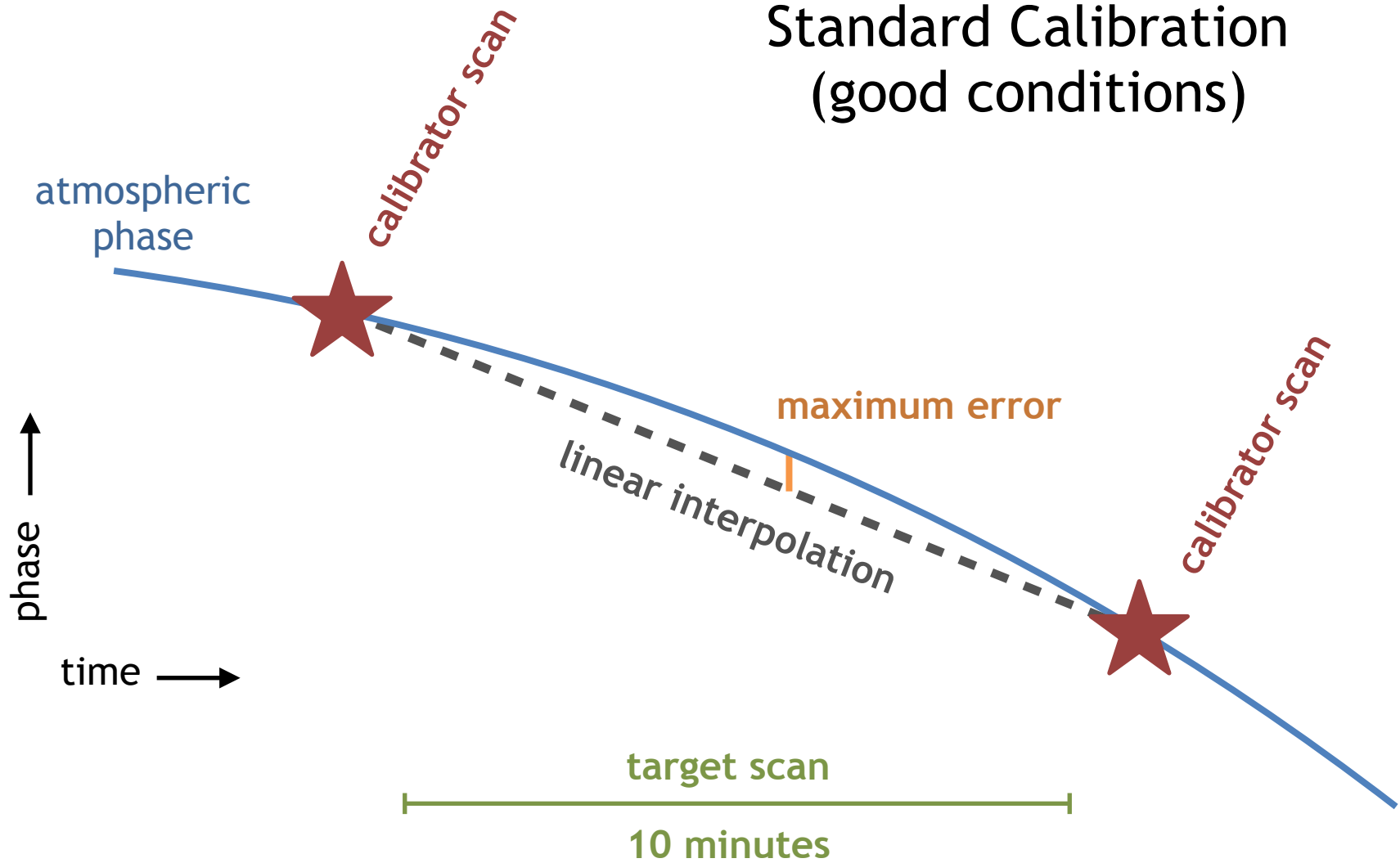
The time it takes for the visibility phase to change by a radian, i.e., the *coherence time*, is  $\approx 1$  minute

This is much shorter than the calibrator cycle time of 10 minutes

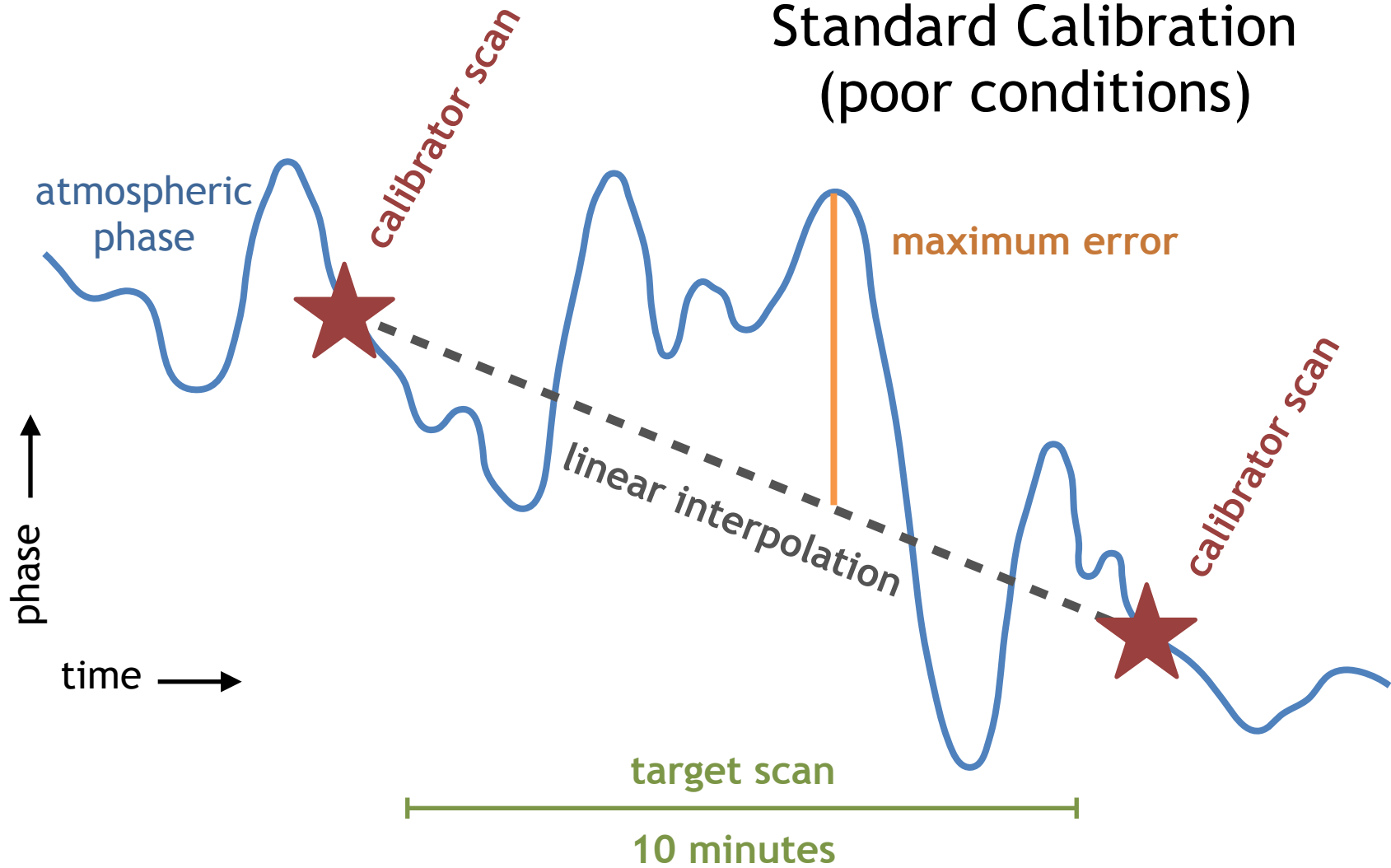
Standard calibration will likely leave similarly large residual phase errors in the target data

We should aim for self-calibration solutions  $\approx 1$  minute if possible

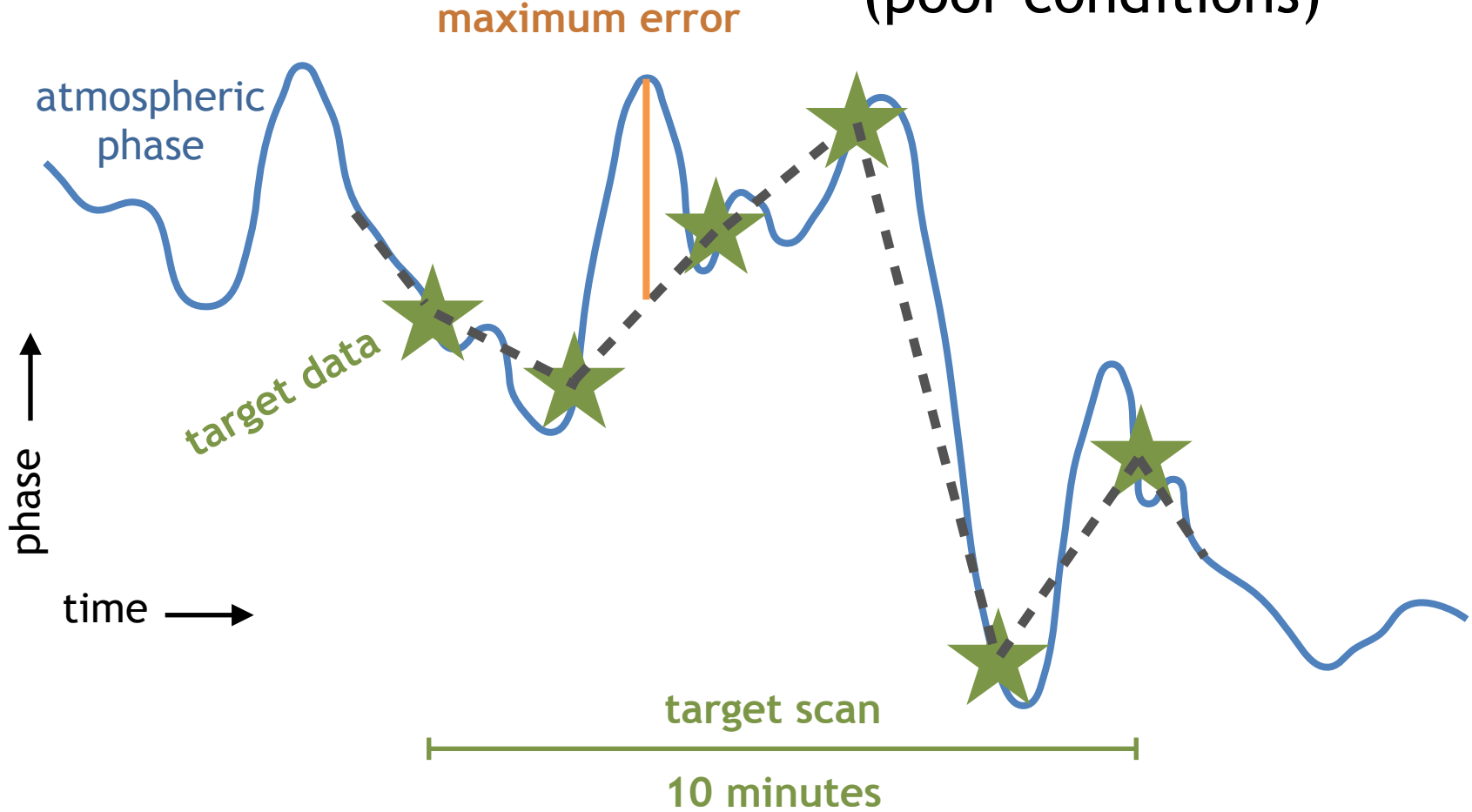
# Standard Calibration (good conditions)



# Standard Calibration (poor conditions)



# Self Calibration (poor conditions)



# Preparing the Data

At the end of the standard calibration procedure, the **applycal** task is run to create the **CORRECTED\_DATA** column

To prepare the data for self-calibration, we will copy the corrected visibilities to a new measurement set **\*\* selfcal will not work without this step \*\***

Specifically, the task **split** will take the **CORRECTED\_DATA** column of the input MS and place it in the **DATA** column of the output MS

You should also use data selection to copy only the target data of interest, e.g., field(s), SPWs, correlations. It is likely best to work with a single band at a time.

```
split(vis='17B-197.sb34290063.eb34589992.58039.861  
19096065.ms', datacolumn='corrected', field='1',  
correlation='RR,LL', outputvis='obj.ms')
```

# Selfcal Requires an Initial Model

Now that we have prepared a measurement set to start the self-calibration process, the first step is to create a model of our target field

This is typically done by imaging the target visibilities, but if you have prior knowledge of the field you may be able to use that instead

In this demonstration, we will run the task **tclean** to image and deconvolve the target field



# Creating the Initial Model

It is important to ensure that proper imaging techniques are used (e.g., wide-band, wide-field) so as not to create additional image errors and thus model errors

We want to model all sources that contribute significantly to the visibilities, even those far away from the target of interest

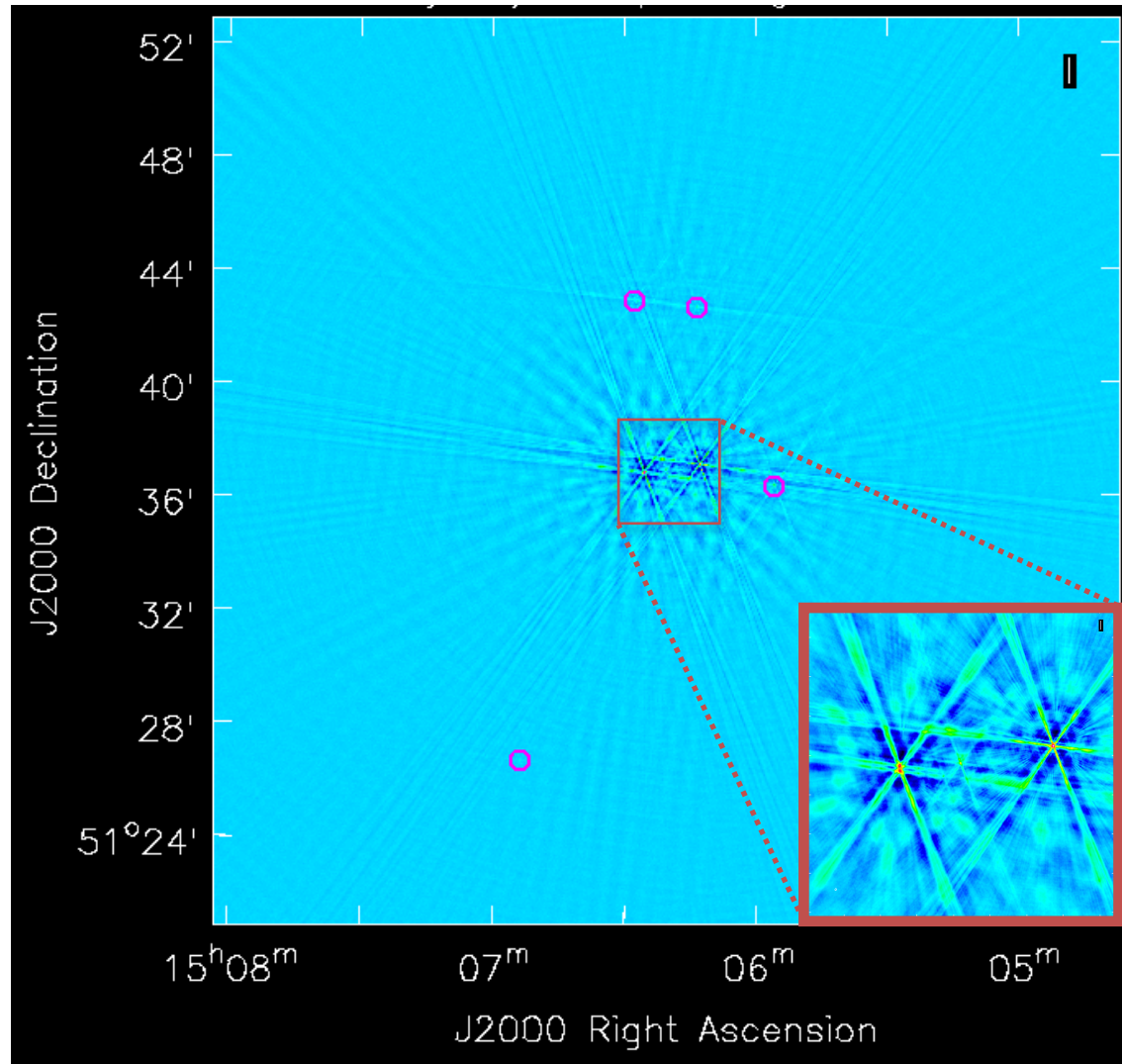
We want to clean conservatively to avoid artifacts (e.g., sidelobes, negative regions) becoming part of the model

# Choosing the Field-of-View

We create a dirty image of the entire primary beam and its first sidelobe to see where the bright sources are

We see two bright sources near the center of the image

Four much fainter sources are circled in magenta



# Handling Outlier Sources

Outlying sources can be modeled by cleaning the large image or by placing an outlier field on each source.

Alternatively, they can be UV-subtracted if they are not of scientific interest (you will need to split out a new MS from uv-subtracted data)

Another option is to simply ignore them if they are too weak to contribute to the calibration solutions

We conclude that the outlying sources do not need to be modeled because they are much fainter than the two bright sources in the center

# Running tclean

By studying the large field-of-view image we were able to make a decision about the outlying sources and to choose an appropriate image size to use going forward

We will now run `tclean` again to save the initial model. We choose the following key parameters:

`savemodel='modelcolumn'` -- To enable writing the `MODEL_DATA` column to the MS. **\*\* selfcal will not work without this step \*\***

`weighting='briggs'; robust=0` -- to suppress PSF sidelobes and thereby reduce certain types of artifacts

`interactive=True` -- to avoid artifacts by careful masking and by deciding when to stop cleaning

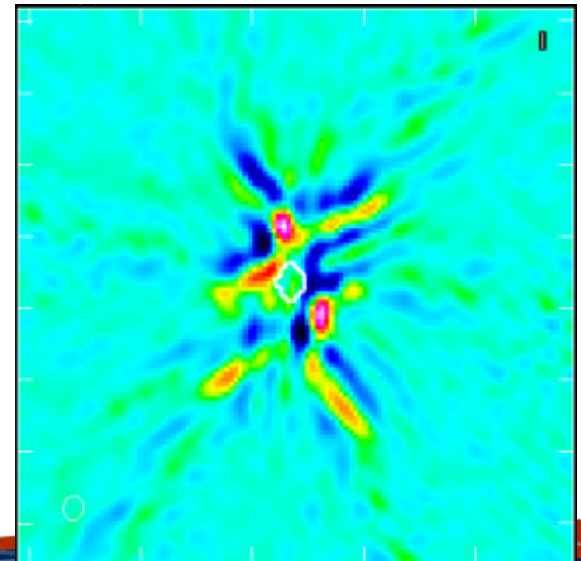
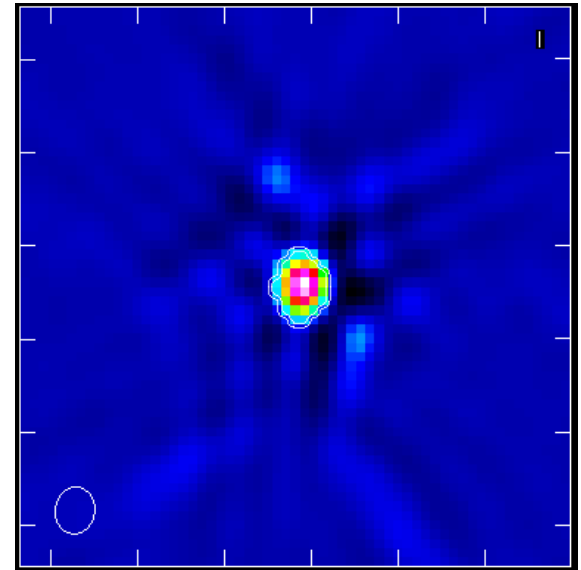
# Interactive Masking

This is one of the two bright sources in the center of the field.

We draw tight masks around the emission that we believe is real

This is the same source after cleaning for 3 major cycles

We are suspicious that the residual structures are artifacts so we do not include them in the mask



# The Initial Image

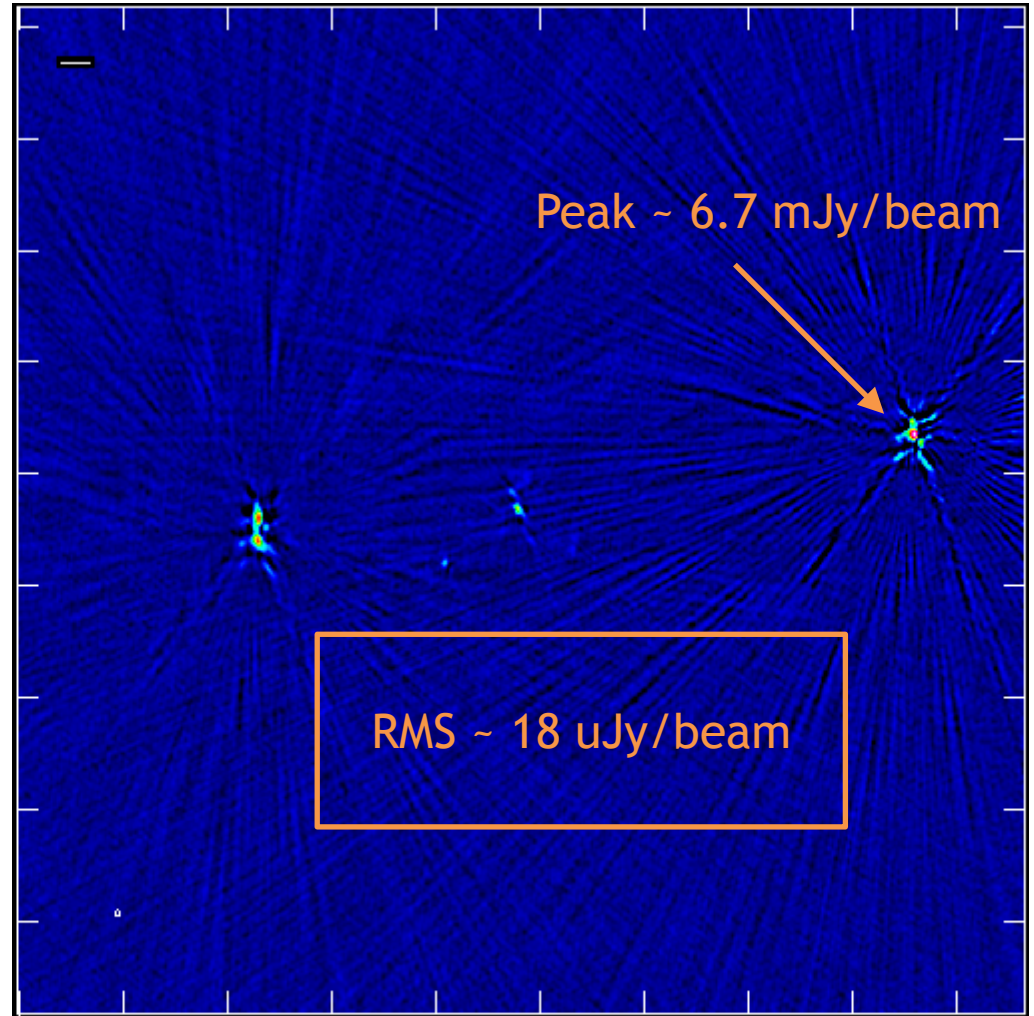
This is the image we get after cleaning

The sensitivity is not thermal noise-limited

Artifacts from the bright sources dominate the noise floor

We measure a **dynamic range** of:

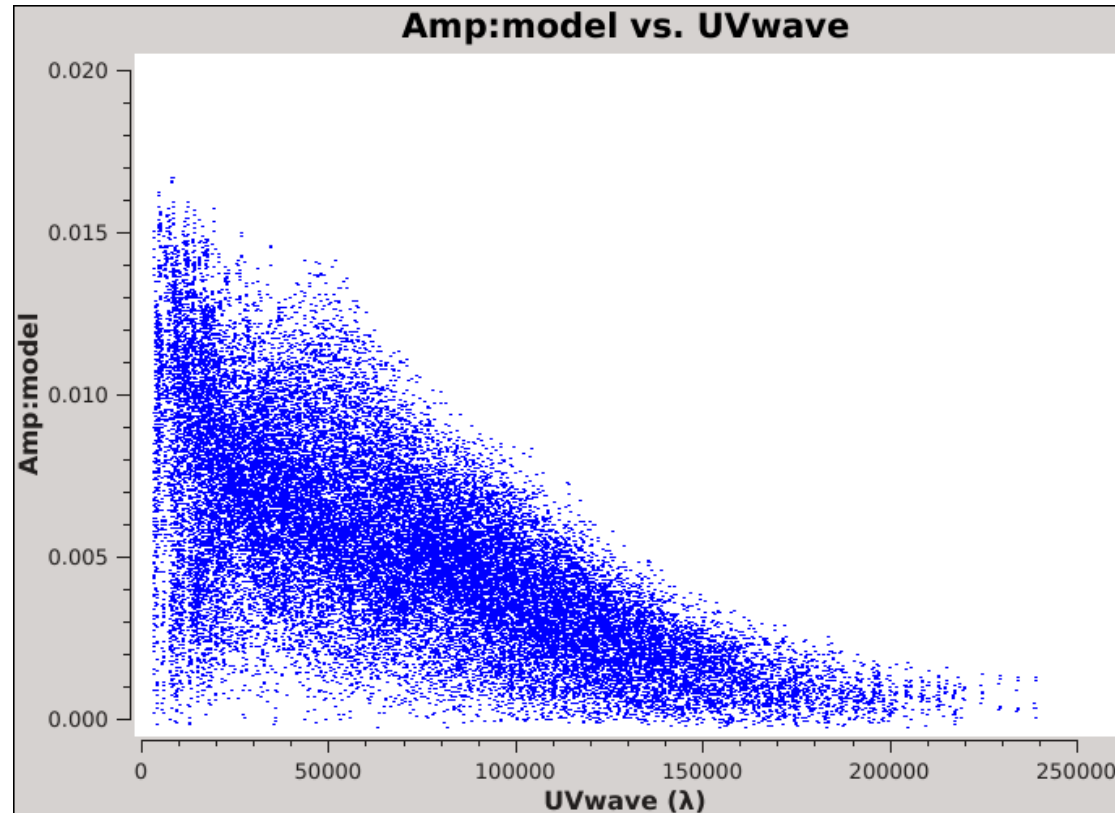
Peak / RMS ~ 370



# Verifying the Initial Model

The initial model now exists in two forms: the model image used during deconvolution, and the model visibilities in the **MODEL\_DATA** column of the measurement set

It is recommended to verify that the later has been filled properly



```
plotms(vis='obj.ms', xaxis='UVwave', yaxis='amp',  
ydatacolumn='model', avgchannel='64', avgtime='300')
```

# Choosing Parameters

Now that we have filled the **MODEL\_DATA** column with visibilities we are ready to solve for calibration solutions (i.e., compare with **DATA**)

It is recommended to begin with **gaincal** in phase-only mode because phase corrections can boost the amplitude by reducing decorrelation

We will have several **gaincal** parameters to consider, in particular:

- solution interval
- data combination (i.e., pol, spw)
- solution signal-to-noise ratio



# Choosing Parameters

**type of solutions:** set **calmode** = 'p' for phase-only solutions (recommended for the first rounds), or 'ap' for amplitude+phase

**solution interval:** as short as possible while maintaining an acceptable signal-to-noise. **solint** = 'int', 'inf' or time quantity. Can start out longer and reduce in further iterations as more flux density is recovered.

**data combination:** only if demonstrated to show the same underlying pattern, and only if needed to achieve the desired signal-to-noise. **gaintype** = 'T' will combine polarizations, 'G' will solve each polarization separately. **combine** = 'spw' will combine all selected SPWs

**minimum signal-to-noise:** default **minsnr** = 3.0, but can be raised to reject noisy solutions; can be set to zero for inspection purposes

# Initial Solutions

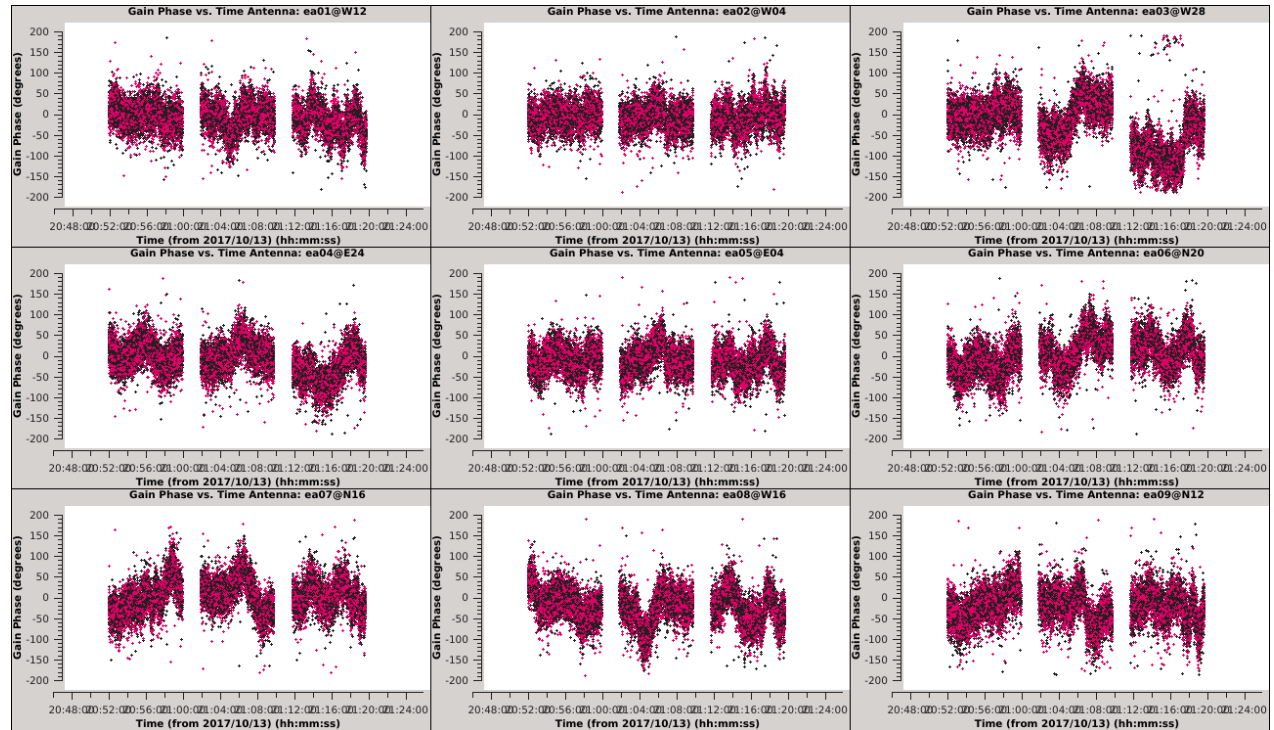
Our first calibration table will use the highest time interval and will not flag any solutions for low signal-to-noise.

This will be a temporary table that we will inspect to help determine the optimal **gaincal** parameters.

```
gaincal(vis='obj.ms', caltable='selfcal_initial.tb',  
solint='int', refant='ea24', calmode='p', minsnr=0,  
gaintype='G')
```

# Inspecting Solutions

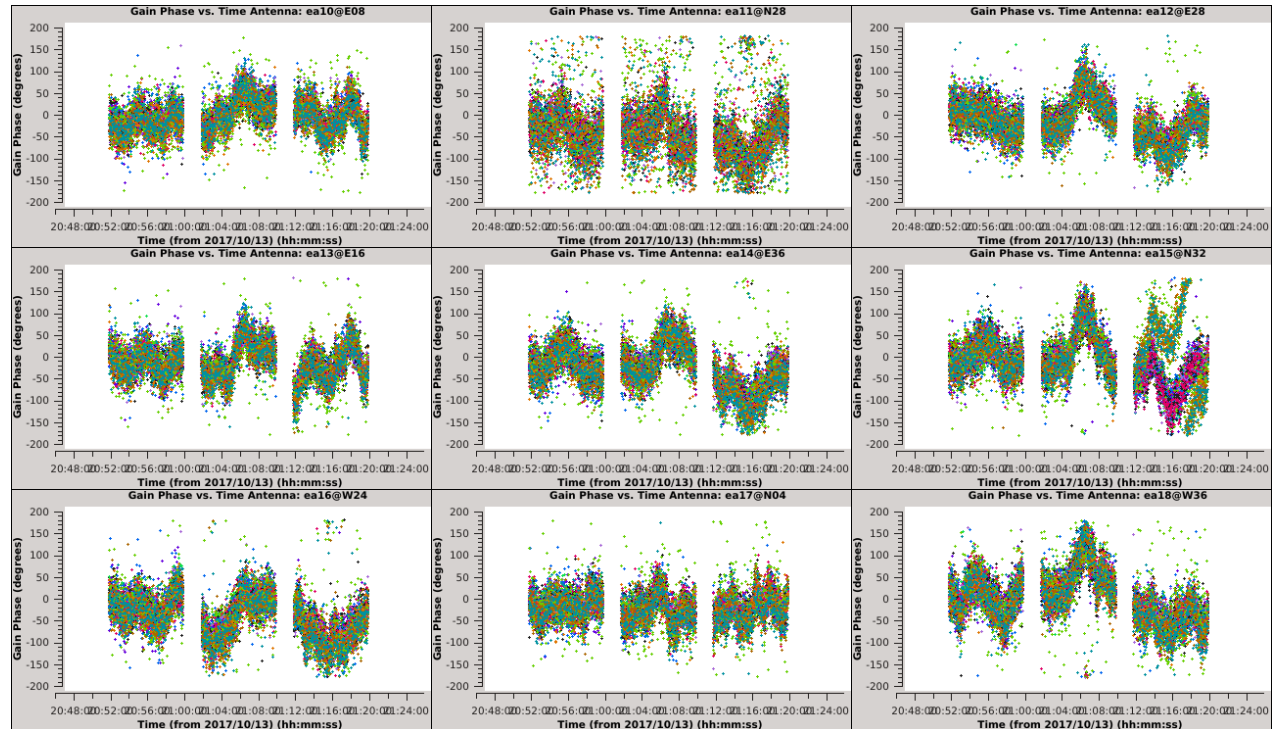
Both polarizations (red and black) show the same pattern, and therefore could be combined if necessary



```
plotms(vis='selfcal_initial.tb', xaxis='time',  
yaxis='phase', iteraxis='antenna', gridrows=3,  
gridcols=3, coloraxis='corr')
```

# Inspecting Solutions

All SPWs (colored) do not always show the same pattern, e.g., antenna ea15

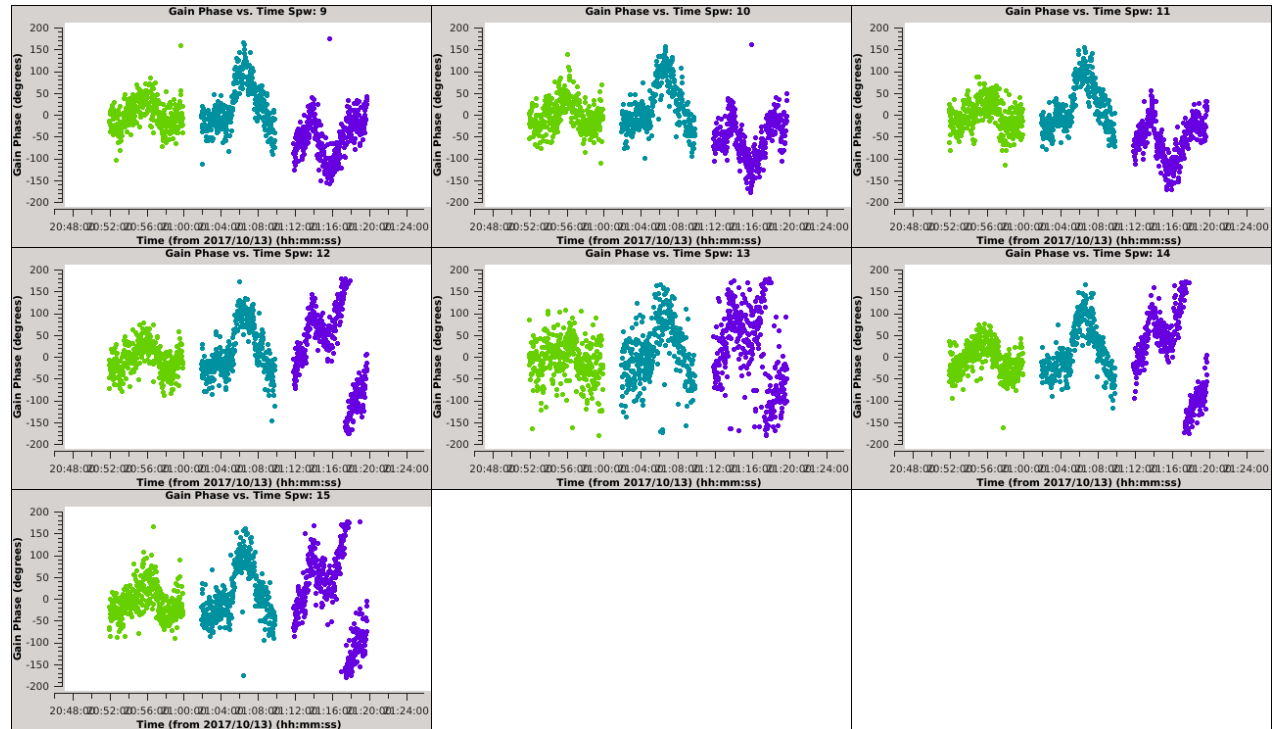


```
plotms(vis='selfcal_initial.tb', xaxis='time',  
yaxis='phase', iteraxis='antenna', gridrows=3,  
gridcols=3, coloraxis='spw')
```

# Inspecting Solutions

The last four SPWs show a phase wrap ( $2\pi$ ) while the other SPWs do not.

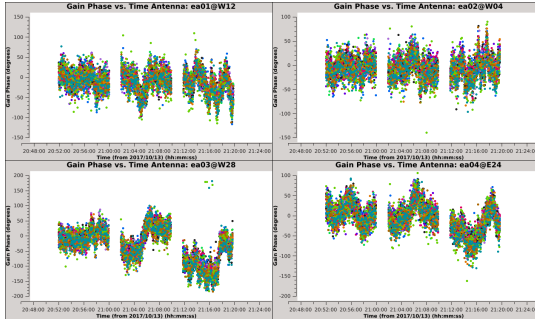
This means that combining SPWs will not be trivial



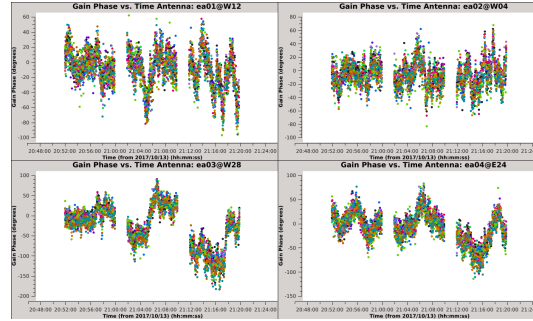
```
plotms(vis='selfcal_initial.tb', xaxis='time',  
yaxis='phase', antenna='ea15', iteraxis='spw',  
gridrows=3, gridcols=3, coloraxis='scan')
```

# Comparing Solution Intervals

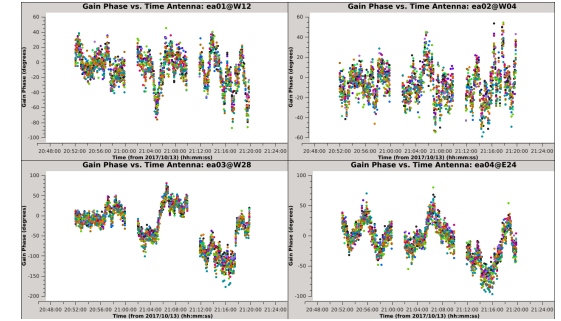
3s



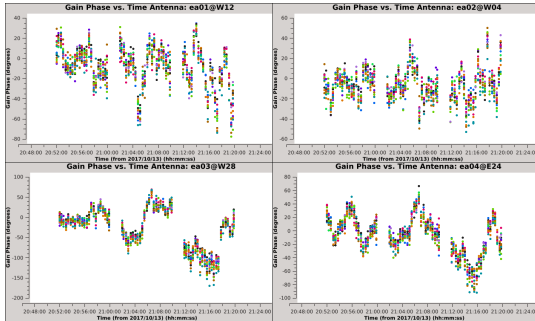
6s



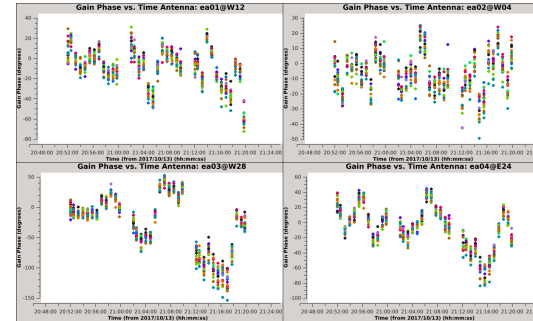
12s



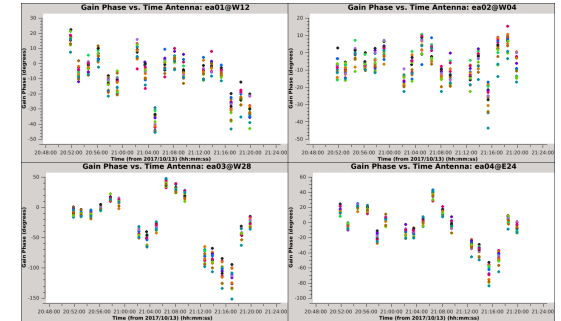
24s



48s



96s



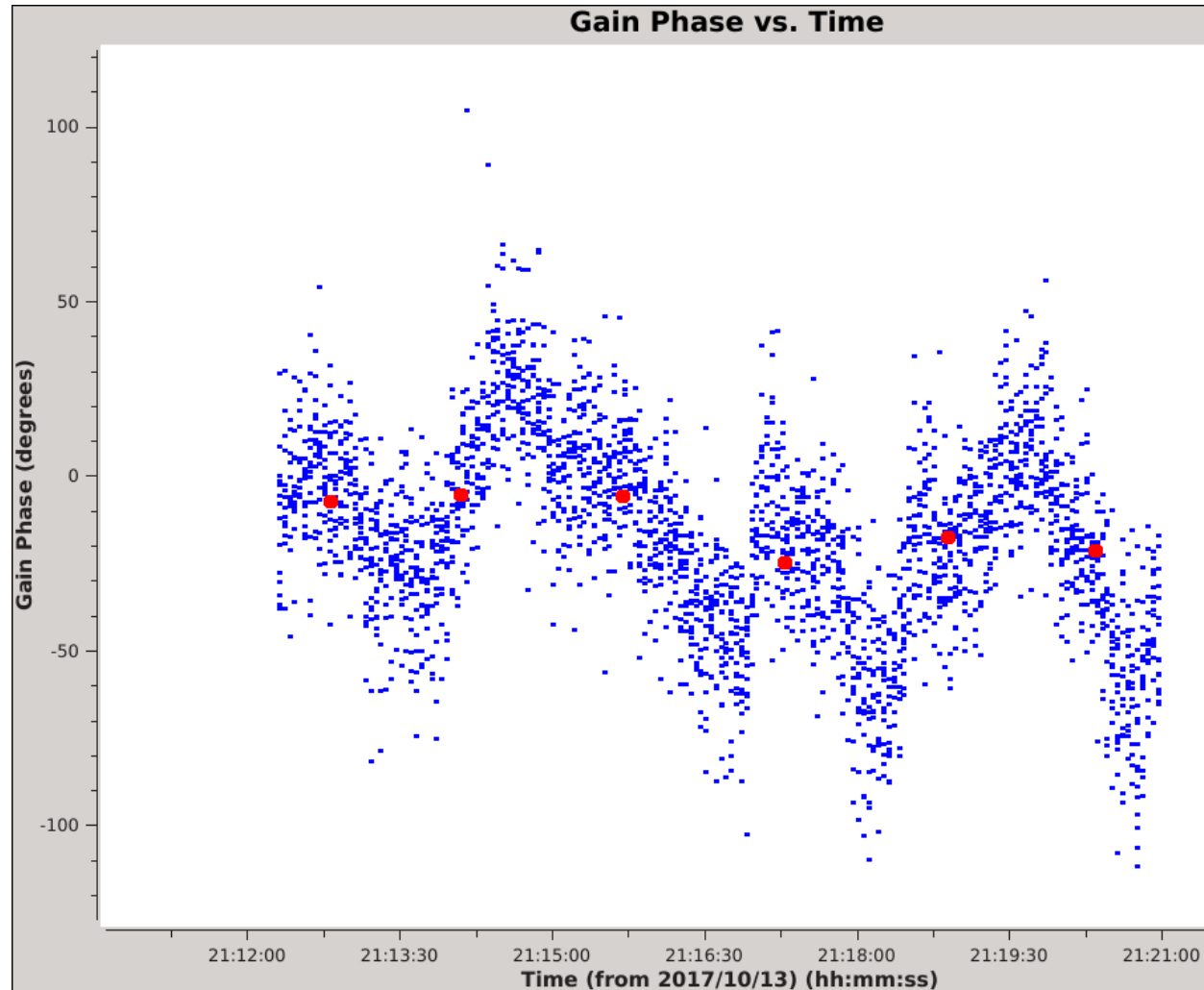
```
for solint in ['3s', '6s', '12s', '24s', '48s', '96s']:  
    gaincal(vis='obj.ms', solint=solint, refant='ea24',  
           caltable='selfcal_combine_pol_solint_'+solint+'.tb',  
           calmode='p', gaintype='T', minsnr=0)
```

# Inspecting Solutions

Comparison of 3s (blue) to 96s (red) solution intervals.

The 3s solutions show significant structure but also a lot of noise.

The 96s solutions are unable to sample the large, fast phase changes.

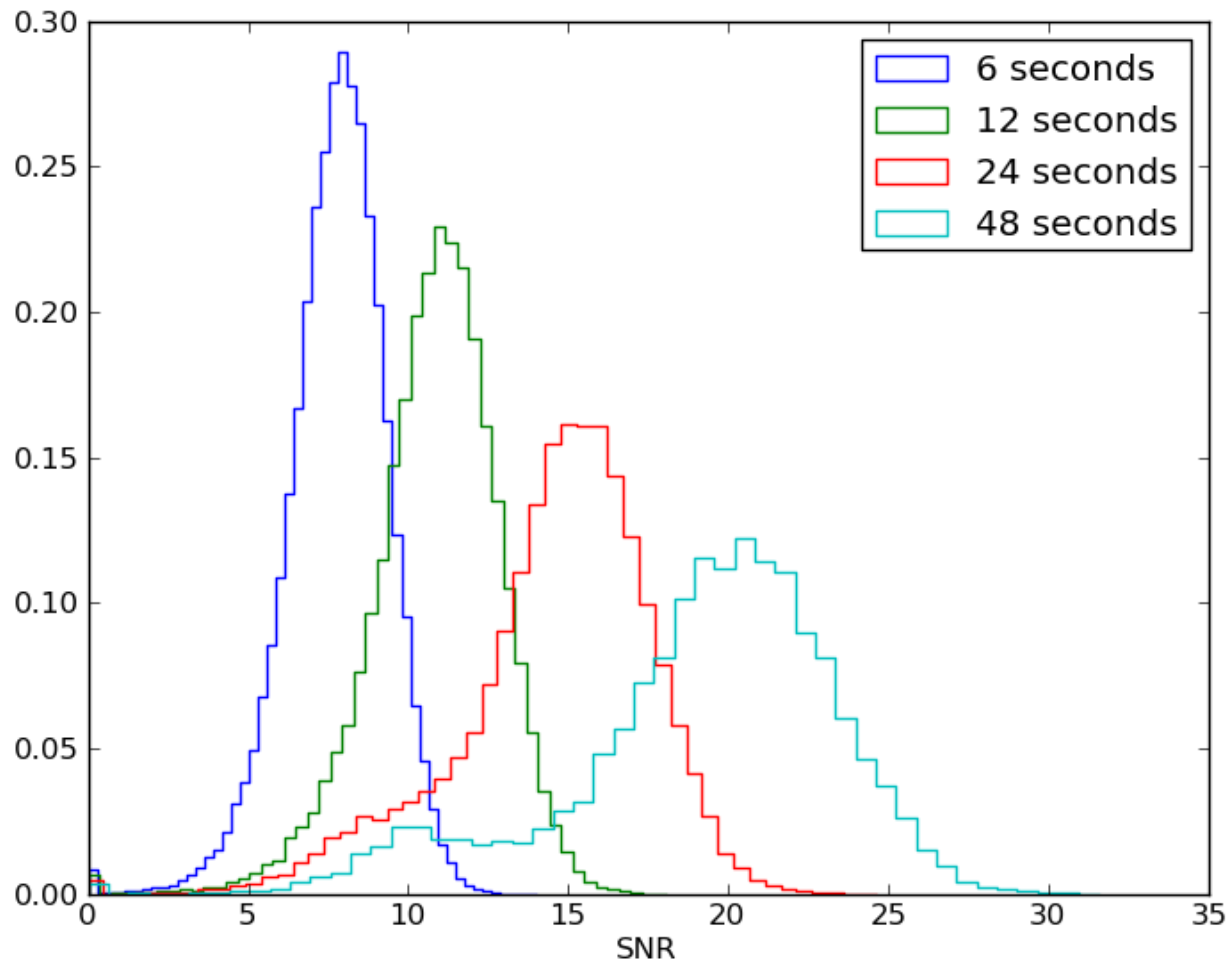


# Choosing Parameters

We suggest that the optimal parameters will use the shortest possible interval for which the signal-to-noise is also sufficient.

Low SNR solutions can bias the results, so we adopt a conservative value of  $\text{minsnr}=6$

This leads us to choose the 12s solutions since <2% of solutions will be flagged



See Tutorial for instructions to make this plot



# The Final Solutions

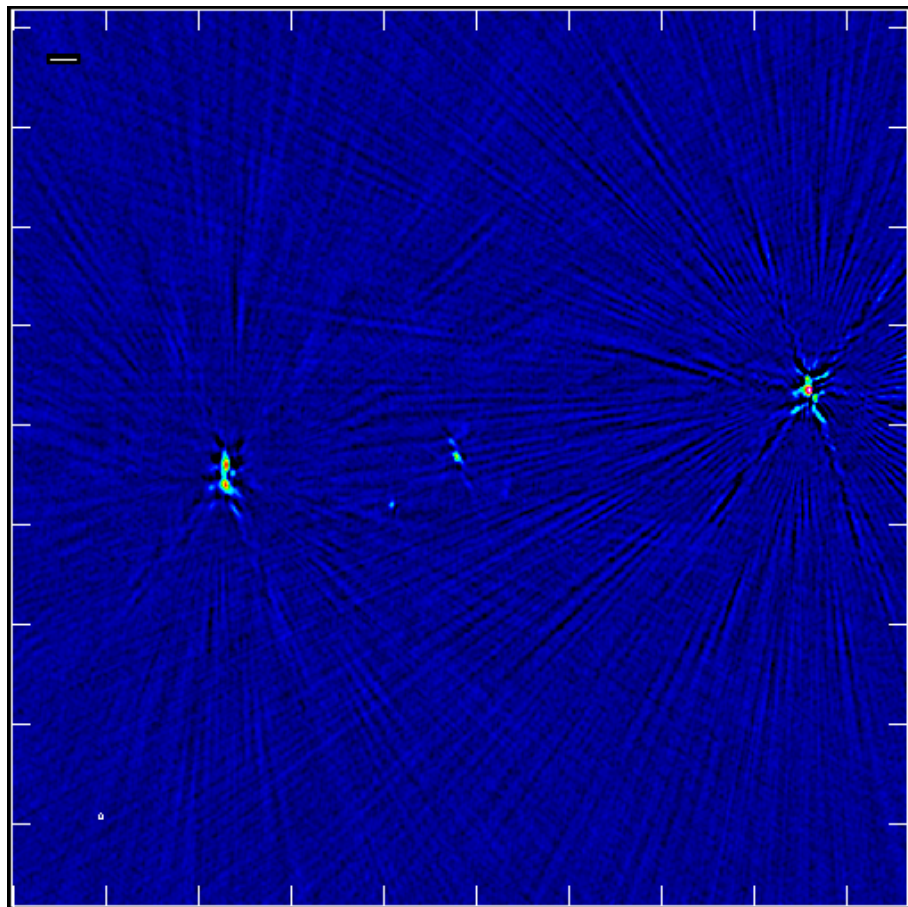
We generate the final calibration table by imposing the **minsnr** of 6 and then applying the solutions.

Then we re-run **tclean** using the same parameters except we set **datacolumn='corrected'** to use the self-calibrated visibilities

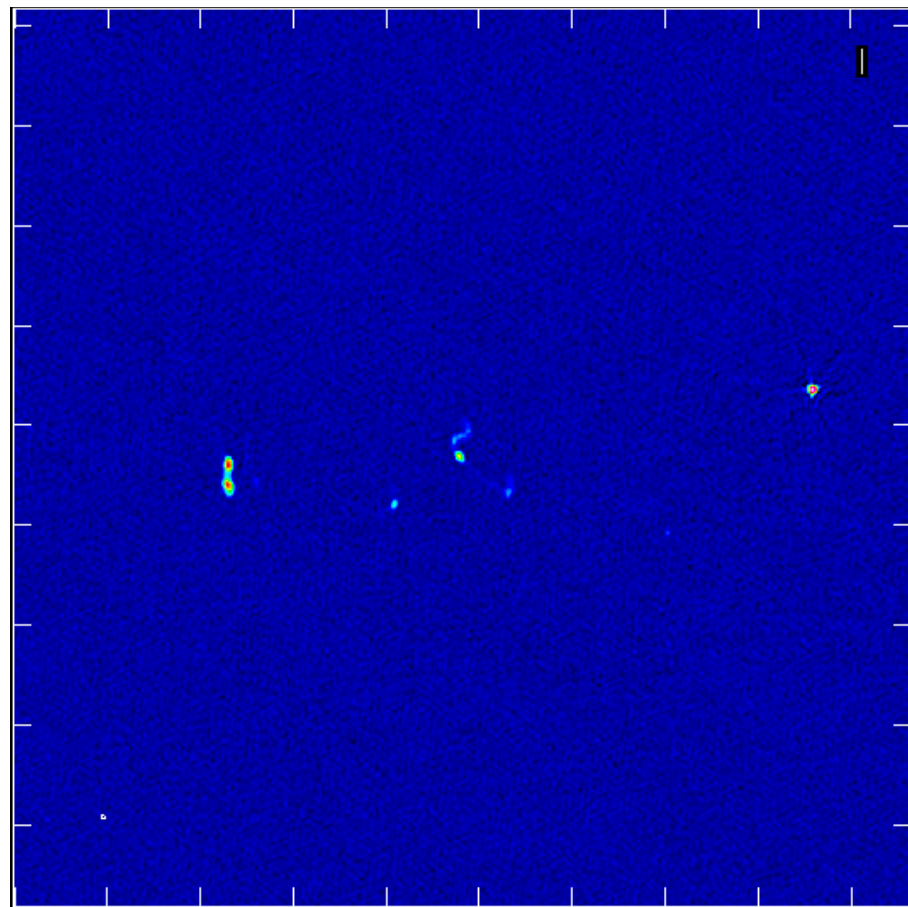
```
gaincal(vis='obj.ms', solint='12s', refant='ea24',  
caltable='selfcal_combine_pol_solint_12_minsnr_6.tb',  
calmode='p', gaintype='T', minsnr=6)
```

```
applycal(vis='obj.ms', gaintable='selfcal_combine_pol_  
solint_12_minsnr_6.tb')
```

Before selfcal



After selfcal

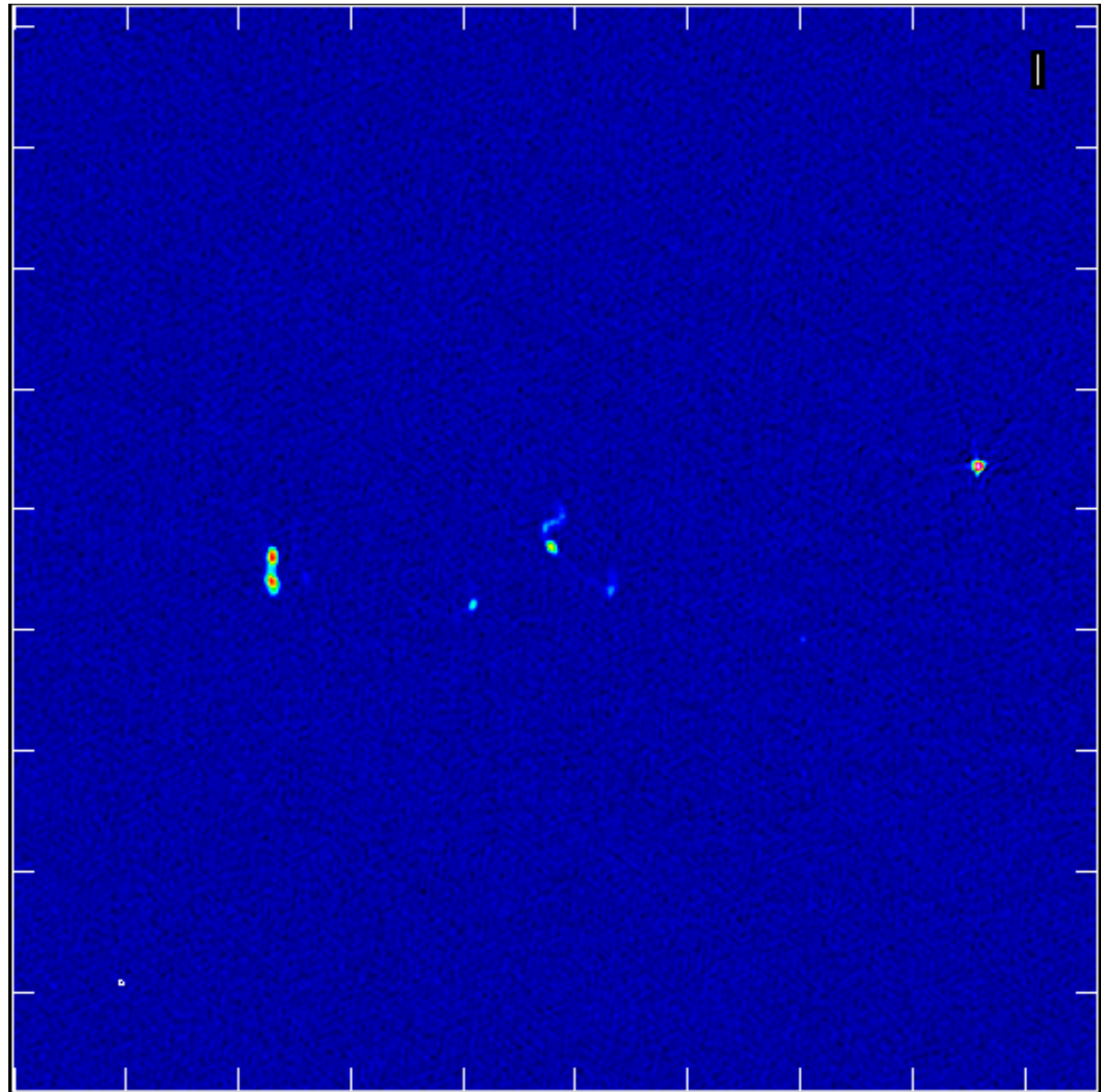


Peak flux has **increased** from 6.7 to 9.1 mJy

Background RMS has **decreased** from 18.2 to 8.2  $\mu$ Jy and noise looks more Gaussian

Dynamic range has **improved** by a factor of 3, from 370 to 1100!

Science target (central source with bent tail) is more clearly visible



# Self-Calibration Part 3

# The Beyond

# Next Steps

**Decide to stop:** the new image may now be good enough to meet your science goals, so there may be no need to go further

**Modify Image Parameters:** the new image may now be limited by issues that self-cal can not improve but can be sidestepped

**Iterate on Selfcal Parameters:** it's easy to undo selfcal and try again, just remember to restore any flags written by applycal

**Additional Rounds of Phase-only Self-calibration:** sometimes it may take several iterations to reach a satisfactory result, improving model each time

**Amplitude Self-calibration:** after phase-only selfcal, you may want to try including amplitude corrections if you suspect issues. It will be important to clean deeply, carefully iterate, and/or normalize the amplitude solutions in order to preserve the flux-density scale.

# Considerations

**Iteration:** by default, each iteration of gaincal/applycal will solve based on the model and what is in DATA column . It is also possible to include past calibration tables in gaincal and applycal as “bootstrapping” (e.g. slowly decreasing the solution interval), though watch for excessive flagging between runs (revert if needed).

**Including Amplitude Self-calibration:** when moving on to amplitude self-calibration, you will most likely want to start by using the best phase-only table, either by specifying during solve and apply, or by writing out the CORRECTED column to a new MS after the phase-only calibration. It is customary to solve using **calmode** = ‘ap’ to pick up any corrections to the phase resulting from a changed (hopefully improved) model. It is likely that you will be using longer solution intervals compared to those for phase-only (particularly at lower frequencies) but they may end up being comparably short when fluctuations are dominated by the troposphere (e.g. K, Ka, Q).

# Advanced Options

**Bandpass Self-calibration:** running the **bandpass** task instead of **gaincal** will trade frequency resolution for signal-to-noise. The solutions can be a function of time or just a single solution interval

**Delay Self-calibration:** many phase errors are actually delay errors, SNR can be gained by solving for delays per baseband, but this is best done using a Global Fringe Fitter (e.g. as used in VLBI processing). Note that “delay clunking” on short timescales falls under this.

**Direction-dependent Self-calibration:** “peeling” is possible in CASA: solve for gains at the location of sources in outer primary beam, apply, uv-subtract, split residual to new MS, repeat. Scripting the peeling process is highly recommended (also see source subtraction casaguide). There is also experimental code for pointing self-cal. Other more advanced techniques require 3rd party software.

**Baseline-based Self-calibration:** to reach the highest dynamic ranges (e.g.  $>10^6:1$ ) it often becomes necessary to solve per baseline (“M”)! Great care and restraint must be used...

# Delay Clunking - example

At ultimate dynamic range ( $10^6:1$  and beyond!) you have to deal with a number of esoteric effects, such as “delay clunking” due to bit changes in the delays applied in the correlator. This particular effect introduces a time-variable delay in each spectral window that cycles from  $\pm 11$ deg at edge of each spw.

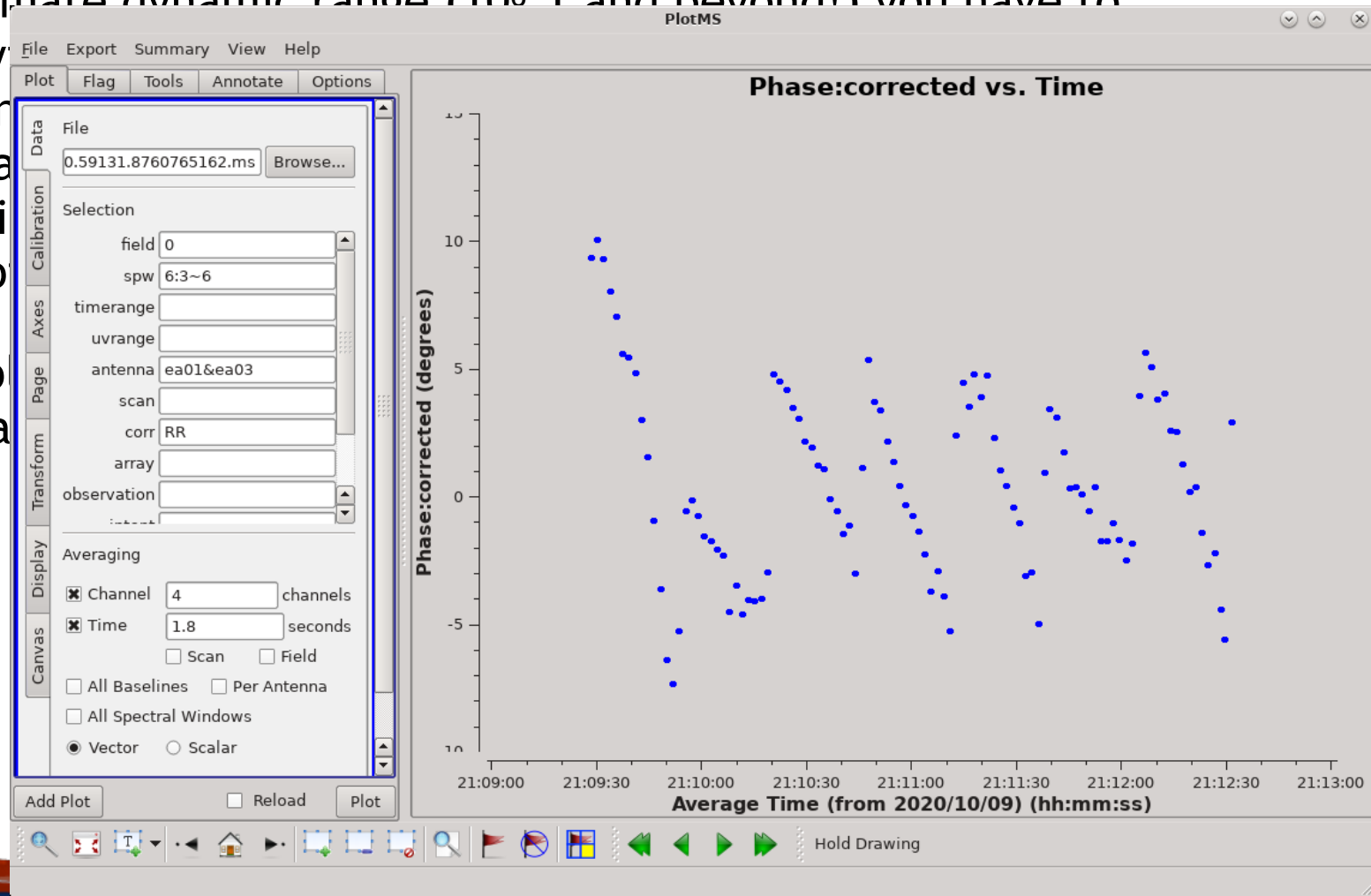
Example from random VLASS observation of 3C286 as calibrator in S-band:



# Delay Clunking - example

At ultimate dynamic range (106:1 and beyond!) you have to deal with clunking correlations, delay in edge of

Example calibration



# References

**CASA Guides:**  
[casaguides.nrao.edu](http://casaguides.nrao.edu)

**CASA Docs:**  
[casa.nrao.edu/casadocs](http://casa.nrao.edu/casadocs)

**Synthesis Imaging in Radio Astronomy II:**  
[ui.adsabs.harvard.edu/abs/1999ASPC..180.....T](http://ui.adsabs.harvard.edu/abs/1999ASPC..180.....T)

Acknowledgement: Much of this material is from J. Marvil and the casaguide.



**[www.nrao.edu](http://www.nrao.edu)**  
**[science.nrao.edu](http://science.nrao.edu)**  
**[public.nrao.edu](http://public.nrao.edu)**

*The National Radio Astronomy Observatory is a facility of the National  
Science Foundation  
operated under cooperative agreement by Associated Universities, Inc.*