Sunspot Band6 Calibration for CASA 4.7

1. *Overview*
2. *Before Starting the Calibration of Visibility Data*
3. *Unpack the Data*
4. *Confirm your version of CASA*
5. *Initial Inspection, A priori calibration*

   Flagging before creating Tsys and Tsys+Tant tables.

   Tsys calibration of the visibilities of the calibrators

   Tsys+Tant calibration of the visibilities of the Sun

   Flagging after Tsys and Tsys+Tant calibration
6. *Additional Data Inspection*
7. *Set up the Flux Calibration Model*
8. *Creating the Bandpass Calibration Table*
9. *Creating the Gain Calibration Table*
10. *Applying the Calibration Tables*
11. *Re-calculation of the direction*
12. *Alternative way of the Calibration*

1. Overview

The portion of the Sunspot_Band6 CASA Guide will cover the calibration of the raw visibility data. **To follow this guide you must have downloaded the file Sunspot_Band6_UncalibratedData.tgz from Sunspot_Band6#Obtaining the Data**[1].

Detail of the ALMA observations are provide at Sunspot_Band6[2].

To skip to the image synthesis portion of the guide, see

[1] Hyperlink to “Sunspot_Band6#Obtaining the Data”
[2] Hyperlink to “Sunspot_Band6”

Sunspot_Band6_Imaging_for_CASA_4.7[3].

This guide is designed for CASA 4.7.

From next, we will show all commands for the calibration. If you do not want to cut-&-paste the commands, you can use the script as described in section 12 "Alternative way of the Calibration".

---

2. Before Starting the Calibration of Visibility Data

The "Analysis Utilities" package must be used for the calibration of solar raw visibility data. Therefore, before starting the tutorial, you need to install the package to your data-anaylysis environment. The documents and software of the package can be obtained from the Analysis Utilities page.

---

3. Unpack the Data

Once the file Sunspot_Band6_UncalibratedData.tgz had been download, unpack it as follows:

```
# In a terminal outside CASA
tar -xvzf Sunspot_Band6_UncalibratedData.tgz

cd Sunspot_Band6_UncalibratedData

#Start CASA
casa
```

---

[3] Hyperlink to "Sunspot_Band6_Imaging_for_CASA_4.7"

4. Confirm your version of CASA

This guide has been written for CASA release 4.7. Please confirm your version before proceeding.

```
# In Casa
version = casadef.casa_version
print "You are using " + version
if (version < '4.7.0'):
    print "YOUR VERSION OF CASA IS TOO OLD FOR THIS GUIDE."
    print "PLEASE UPDATE IT BEFORE PROCEEDING."
else:
    print "Your version of CASA is appropriate for this guide."
```

We need to import some scripts we will use during the calibration.

```
#In Casa
import analysisUtils as aU
es = aU.stuffForScienceDataReduction()
execfile("SunRedUtil.py")
```

"SunRedUtil.py" is included in the file Sunspot_Band6_UncalibratedData.tgz

---

5. Initial Inspection, *A priori* calibration

We start by defining the directory name of the ASDM and some directory names of the Measurement Sets (MS) for the calibration.

```
#In Casa
asdm ='uid___A002_Xae00c5_X2a8d'
mso = asdm + '.ms'
mss = asdm + '_split.ms'
msc = mss + '.cal'
```

The raw data have been provided to you in the ASDM format. It is the native

format of the data produced by the ALMA observatory.
Before we can proceed to the calibration, we will need to convert those data to the CASA MS format. This is done simply with the task importasdm.

```
#In Casa
importasdm(asdm = asdm, vis = mso, asis='Antenna Station Receiver Source
CalAtmosphere CalWVR CorrelatorMode SBSummary CalDevice ')
```

The usual first step is then to get some basic information about the data. We do this using the task listobs, which will output a detailed summary of each dataset supplied.

```
#In Casa
listobs(mso, listfile = asdm + '_listobs.txt')
```

The output will be sent to the CASA logger, or saved in a text file. Here is a snippet extracted from the listobs output:

<<Insert the file "uid___A002_Xae00c5_X2a8d_listobs_ext.txt">>

The first section of the output describes the detail of each scan, and the second section (from line 34) shows the information of the observing targets. This second section shows that three targets with 151 fields were observed: The Sun, J1924-2914, nrao350(J1733-1304). From the section, the J1924-2914 was observed for the calibrations of pointing [2][4], sideband ratio [3], atmosphere [4], and bandpass [5]. nrao350 was observed for the calibration of pointing [6], atmosphere [7], flux [8], and phase [9,12,15,18]. The Sun was observed for scientific observations [11,14,17], the calibration of atmosphere [10,16] and the measurement of zero-signal level [1].
After a prior calibration, we never use the data of the scans for the calibration of pointing, atmosphere, sideband ratio, and the measurement of zero-signal level. To reduce the size of the dataset for bandpass and gain calibrations, we will extract only the data for the calibrations from the raw

[4] The numbers indicate the scan ID.

dataset. For the extraction, we define the scanIDs for the bandpass/gain calibrations and scientific observations, now.

```
#In Casa
sel_scans ='5,8,9,11,12,14,15,17,18'
```

The third section of the listobs output (from line 47) shows the information of the spectrum windows (Spw) in the dataset. From first and this sections, the scientific observations are done with the Spw 0~12, and the IDs of the Spw with 128 channels, which are used for image synthesis, are 5, 7, 9, and 11. Therefore, we will calibrate the data of SpwID 5, 7, 9, and 11 only. The data of spwID 0,1,2,3 are the data from the square-law detectors of the basebands. The data will be used for creating T_{sys}+T_{ant} tables, and are archived as the auto-correlation data in the dataset.
Thirty-one antennas were used for the dataset. Note that numbering in python always begins with "0", so the antennas have IDs 0-30. To see what the antenna configuration looked like at the time of this observation, we use the task plotants.

```
#In Casa
plotants(mso, figfile= asdm + '_antconf.png')
```

This will plot the antenna configuration on your screen (see Figure 1) as well as save it under the specified filename for future reference. We need to choose a reference antenna that is close to the center of the array (and is also stable and present for the entire observation). We will use the antenna DA41 as the reference antenna, and we define a variable to be used in the following command:

```
#In Casa
ref_ant ='DA41'
```

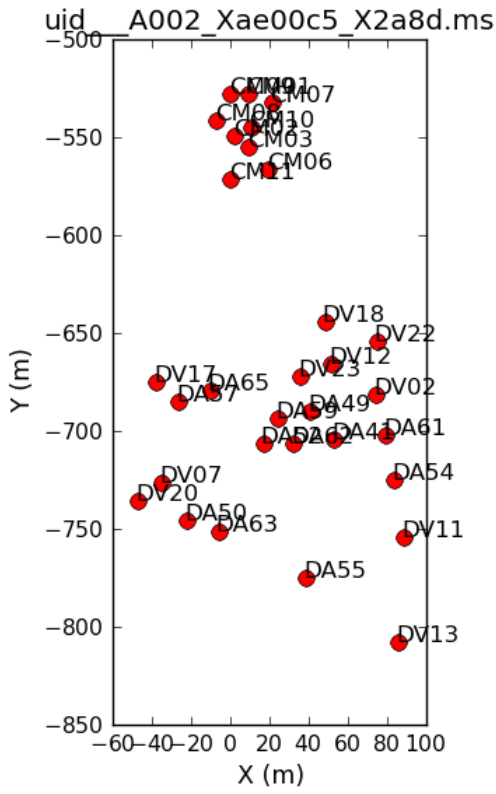

Figure 1. “plotants” output for the dataset uid___A002_Xae00c5_X2a8d

Flagging before creating Tsys and Tsys+Tant tables.

Some scans in the data were used by the online system for pointing and sideband ratio calibration. These scans are no longer needed, and we can flag them easily with flagdata by selecting on 'intent'.:

```
#In Casa
flagdata(vis = mso, mode = 'manual', intent = '*POINTING*,*SIDEBAND_RATIO*',
         flagbackup = F)
```

The averaged data of each spectrum window is not used, so we flagged the averaged data as a follow.

```
#In Casa
flagdata(vis = mso, mode = 'manual', spw = '6,8,10,12', flagbackup = F)
```

We will then store the current flagging state for each dataset using the flagmanager:

```
#In Casa
flagmanager(vis = mso, mode = 'save', versionname = 'priori1')
```

Tsys calibration of the visibilities of the calibrators

The Tsys calibration gives a first-order correction for the atmospheric opacity as a function of time and frequency, and associates weights with each visibility that persist through imaging. The MS dataset contains Tsys measurements; the task gencal is used to generate a calibration table.

```
#In Casa
gencal(vis = mso, caltable = mso + '.tsys', caltype = 'tsys')
flagdata(vis = mso + '.tsys', mode = 'manual',
         spw = '5:0~9;116~127,7:0~9;116~127,9:0~9;116~127,11:0~9;116~127',
         flagbackup = F)
es.checkCalTable(mso+'.tsys', msName=mso, interactive=False)
```

Since the dataset obtained with TDM, the data in the channels near the both edges of the spectrum window (~10 channels) are flagged. Then, the plots for checking are created by the subroutine of Analysis Utilities package (Figure 2). The Tsys of DA54 antennas are significant large, from the plot. In the later part, we will flag the data of the antenna.

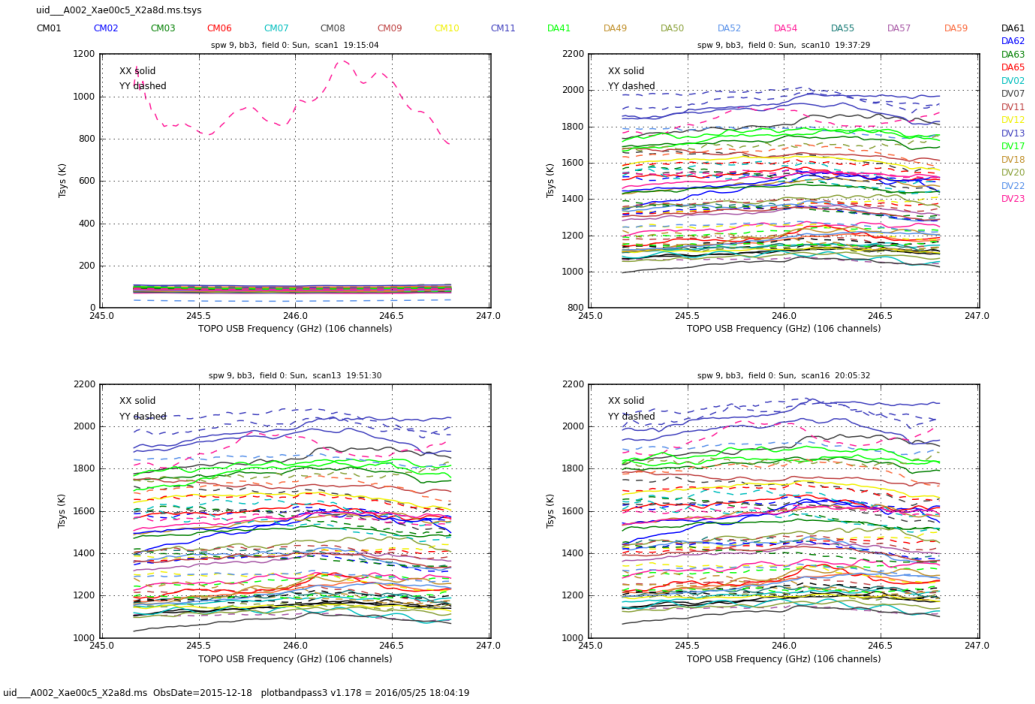

Figure 2. The plots of the Tsys of Spw=7.

We will apply the Tsys calibration table to the data of the calibrators with the task applycal, which reads the specified gain calibration tables, applies them to the (raw) data column, and writes the calibrated results into the corrected column. For non-solar observations, we also apply the WVR (Water Vapor Radiometer) calibration table to data. However, we must NOT apply the WVR table to the solar data, because the WVR receivers at the Sun occur the saturation.

We apply the Tsys calibration table to the data of the bandpass calibrator:

```
#In Casa
applycal(vis = mso, field = '1', spw = '5,7,9,11',
         gaintable = mso + '.tsys', gainfield = '1',
         interp = 'linear,linear', calwt = T, flagbackup = F)
```

In the observations, we do not the atmospheric calibration of the phase calibrator between the scientific scans, because a long suspension of scientific observations has a bad influence on science. Therefore, we apply the Tsys calibration table, which is created from the data of the atmosphere

calibration at the flux calibrator or the Sun, to the phase calibrator, as follows.

```
#In Casa
applycal(vis = mso, field = '2', spw = '5,7,9,11', gaintable = mso + '.tsys',
          gainfield = '2', interp = 'linear,linear', calwt = T,
          flagbackup = F)
```

You can use plotms to plot channel-averaged amplitudes as a function of time, comparing the DATA and CORRECTED columns after applying the Tsys correction. This way you can check that calibration has done what was expected, which is put the data onto the Kelvin temperature scale.

Tsys+Tant calibration of the visibilities of the Sun

The standard method of Tsys calibration cannot be apply to the data of the Sun, because the antenna temperature of the Sun cannot be neglected for estimating the system equivalent flux density (SEFD). To estimate correct SEFD at the Sun, the solar observing sequence includes some special measurements, like the measurement of zero-signal level. The subroutines for creating and applying Tsys+Tant calibration tables are prepared by the

```
#In Casa
sol_ampcal(mso, mso + '.tsys', exisTbl=F, outCSV=T)
```

ALMA solar development team. The subroutines are already imported at the section “4. Confirm your version of CASA”. For the Tsys+Tant calibration of the solar data, you will execute only the following command.

The process takes long time, about one night or day. If you have already carried out the process before and there is the Tsys+Tant calibration tables (The directory name of the table includes “tsystant”), you can skip the generating of the tables using the following command.

```
#In Casa
sol_ampcal(mso, mso + '.tsys', exisTbl=T, outCSV=F)
```

Flagging after Tsys and Tsys+Tant calibration

Since we completed the Tsys and Tsys+Tant calibration, the data that are not used the image synthesis are flagged. At first, the data of auto-correlation and atmosphere calibration are flagged as follows:

```
#In Casa
flagdata(vis = mso, mode = 'manual', autocorr = T, flagbackup = F)
flagdata(vis = mso, mode = 'manual', intent = '*ATMOSPHERE*',
          flagbackup = F)
```

Next is the spectrum windows that are not used in later:

```
#In Casa
flagdata(vis = mso, mode = 'manual', spw = '0~4', flagbackup = F)
flagdata(vis = mso, mode = 'manual', spw = '13~36', flagbackup = F)
```

The data in the channels near the both edges of the spectrum window (~10 channels) are flagged:

```
#In Casa
flagdata(vis = mso, mode = 'manual', flagbackup = F
          spw='5:0~9;116~127,7:0~9;116~127,9:0~9;116~127,11:0~9;116~127')
```

The some sub-scans at the start and end of the scientific scans are used to measure the sky with the setting of optimized for the Sun. Since the data are used only for estimating the antenna temperatures at the Sun, we flag the data:

```
#In Casa
msmd.open(mso)
sciScan = msmd.scansforintent('*OBSERVE_TARGET*')
msmd.done()

for i in range(0, len(sciScan)):
      subInf=aU.computeDurationOfScan(sciScan[i], vis=mso,
                                      returnSubscanTimes=True)
      subNum = subInf[1]
      flagdata(vis = mso, timerange = subInf[3][1], mode = 'manual',
               flagbackup = F)
      flagdata(vis = mso, timerange = subInf[3][2],mode = 'manual',
              flagbackup = F)
      flagdata(vis = mso, timerange = subInf[3][subNum-1], mode = 'manual',
              flagbackup = F)
      flagdata(vis = mso, timerange = subInf[3][subNum], mode = 'manual',
              flagbackup = F)
```

As mentioned at the section “Tsys calibration of the visibilities of the calibrators”, the Tsys values of DA54 antennas are significant large. Therefore, we flag the data of the antenna as follows:

```
#In Casa
flagdata(vis = mso, antenna = 'DA54', mode = 'manual', flagbackup = F)
```

Then, we store the current flagging state for each dataset using the flagmanager:

```
#In Casa
flagmanager(vis = mso, mode = 'save', versionname = 'priori2')
```

Now split out the CORRECTED data column, retaining spectral windows 5,7,9, and 11. This will get rid of the extraneous spectral windows.

```
#In Casa
split(vis = mso, outputvis = mss, datacolumn = 'corrected',
       scan =sel_scans, spw = '5,7,9,11', keepflags = T)
```

---

6. Additional Data Inspection

We will do some additional inspection.

The solar data frequently include zero values in some channels. To avoid their influence, we flag the data using the “clip” mode with “clipzeros=True” option.

```
#In Casa
flagdata(vis = mss, mode = 'clip', clipzeros = True, flagbackup = F)
```

The valleys appear in the plot of amplitude vs channel of the Spw #3. It shows the effect of the lines of the earth atmosphere. We need to flag the data.

```
#In Casa
flagdata(vis = mss, spw = '3:68~85', mode = 'manual', flagbackup = F)
flagdata(vis = mss, spw = '3:20~40', mode = 'manual', flagbackup = F)
```

Then, we store the current flagging state for each dataset using the

flagmanager:

```
#In Casa
flagmanager(vis = mso, mode = 'save', versionname = 'Initial')
```

---

7. Set up the Flux Calibration Model

It is very rare that a useful planet for the flux calibration is located near the Sun. Therefore, we usually use a quasar near the Sun as a flux calibrator. In this observations, there is no quasar that flux is enough for the observations with the Mixer De-tuning mode, except nrao350. Therefore, the flux calibrator is the same as the phase calibrator in this observation.
To fill the model data column for nrao350 with a model for the flux density, we obtain the flux density and spectrum index of the nro350 from the ALMA calibrator database.

```
#In Casa
intentSources=es.getIntentsAndSourceNames(mss)
ampCalId         =         intentSources['CALIBRATE_AMPLI']['id']         +
intentSources['CALIBRATE_FLUX']['id']
calFieldNames      =      intentSources['CALIBRATE_AMPLI']['name']      +
intentSources['CALIBRATE_FLUX']['name']

amp_cal_name=calFieldNames[1]
spwInfo=es.getSpwInfo(mss)
obs_freq="%fGHz"%(spwInfo[0]['refFreq']/1e9)

date=aU.getObservationStartDate(mss)
date_obs=date.split()[0]
spw1_flux=aU.getALMAFlux(sourcename=amp_cal_name,
                         date=date_obs,frequency=obs_freq)
```

Then, fill the model data column for nrao350 with a model

```
#In Casa
setjy(vis = mss, field = '2', spw = '0,1,2,3', standard = 'manual',
       fluxdensity = [spw1_flux['fluxDensity'], 0, 0, 0],
       spix= spw1_flux['spectralIndex'], reffreq = obs_freq)
```

---

8. Creating the Bandpass Calibration Table

The bandpass and gain calibrations are the same as the calibrations of non-solar data basically. Please refer to the other ALMA tutorials of the CASA guide too. The calibration method presented below is the same as that used in the QA2 process of the ALMA observatory.

At first, we run gaincal on the bandpass calibrator to determine phase-only

gain solutions. We will use solint='int' for the solution interval, which means that one gain solution will be determined for every integration time.

```
#In Casa
gaincal(vis = mss, caltable = mss + '.ap_pre_bandpass', field = '1',
        scan = '5', solint = 'int', refant = ref_ant, calmode = 'p')
```

The plots are created for the checking.

```
#In Casa
es.checkCalTable(mss+'.ap_pre_bandpass', msName=mss, interactive=False)
```

The plots look good. Next, create the bandpass calibration table:

```
#In Casa
bandpass(vis = mss, caltable = mss+'.bandpass', field = '1', scan = '5',
         solint = 'inf', refant = ref_ant, solnorm = True, bandtype = 'B',
         gaintable = mss+'.ap_pre_bandpass')
```

Check the bandpass calibration table using the plots created from the following command.

```
#In Casa
es.checkCalTable(mss+'.bandpass', msName=mss, interactive=False)
```

---

9. Creating the Gain Calibration Table

At first, we determine phase-only gain solutions of the calibrators, using the bandpass calibration table and solint='int' option, and create the plots for checking.

```
#In Casa
gaincal(vis = mss, caltable = mss + '.phase_int', field = '1~2',
         solint = 'int', refant = ref_ant,gaintype = 'G',calmode = 'p',
         minsnr = 2.0, gaintable = mss + '.bandpass')
es.checkCalTable(mss + '.phase_int', msName=mss, interactive=False)
```

Using the bandpass and phase calibration tables, we obtain the amplitude-only gain solutions on the scan time, **solint='inf'.**

```
#In Casa
gaincal(vis = mss, caltable = mss + '.ampli_inf', field = '1~2',
        solint = 'inf', refant = ref_ant, gaintype = 'T', calmode = 'a',
         minsnr = 2.0, gaintable = [mss + '.bandpass', mss + '.phase_int'])
es.checkCalTable(mss + '.ampli_inf', msName=mss, interactive=False)
```

Next we use the flux calibrator (whose flux density was set in setjy above) to derive the flux density of the other calibrators. Note that the flux table REPLACES the amp.gcal in terms of future application of the calibration to the data, i.e. the flux table contains both the amp.gcal and flux scaling. Unlike the gain calibration steps, this is not an incremental table.

```
#In Casa
fluxscaleDict = fluxscale(vis = mss, caltable = mss + '.ampli_inf',
                          fluxtable = mss + '.flux_inf', reference = '2')
es.fluxscale2(caltable = mss+'.ampli_inf', removeOutliers=True,
              msName=mss, writeToFile=True, preavg=10000)
```

Finally, we create the gain calibration table of phase-only gain solutions on the scan time, **solint='inf'.**

```
#In Casa
gaincal(vis = mss, caltable = mss+'.phase_inf', field = '1~2', solint = 'inf',
         refant = ref_ant, gaintype = 'G', calmode = 'p',
         minsnr = 2.0, gaintable = mss+'.bandpass')
  es.checkCalTable(mss+'.phase_inf', msName=mss, interactive=False)
```

---

10. Applying the Calibration Tables

We apply the calibration solutions to each source individually, using the **gainfield** parameter to specify which calibrator's solutions should be applied from each of the **gaintable** calibration tables.
Applying the tables to the bandpass, flux and phase calibrators:

```
#In Casa
for i in ['1', '2']:
     applycal(vis = mss,field = str(i),
          gaintable = [mss + '.bandpass', mss+'.phase_int', mss+'.flux_inf'],
          gainfield = ['', i, i], interp = 'linear,linear', calwt = T,
          flagbackup = F)
```

Next, applying them to the solar data:

```
#In Casa
applycal(vis = mss, field = '0, 3~150',
           gaintable = [mss+'.bandpass', mss+'.phase_inf', mss+'.flux_inf'],
           gainfield = ['', '2', ''], interp = 'linear,linear', calwt = T,
           flagbackup = F)
```

Finally, split out the CORRECTED data column

```
#In Casa
split(vis = mss, outputvis = msc, datacolumn = 'corrected', keepflags = T)
```

---

11. Re-calculation of the direction

During most solar observations, the antennas are tracking a structure on the Sun according to solar differential rotation. The image frame is fixed on the solar frame, but the frame is moving on the RA/Dec coordinate frame. If we do not do any measures, the pattern of pointing in the MOSAIC observation is shown in Figure 3, and the shape of the pattern is a rhombus though the correct shape is the square.

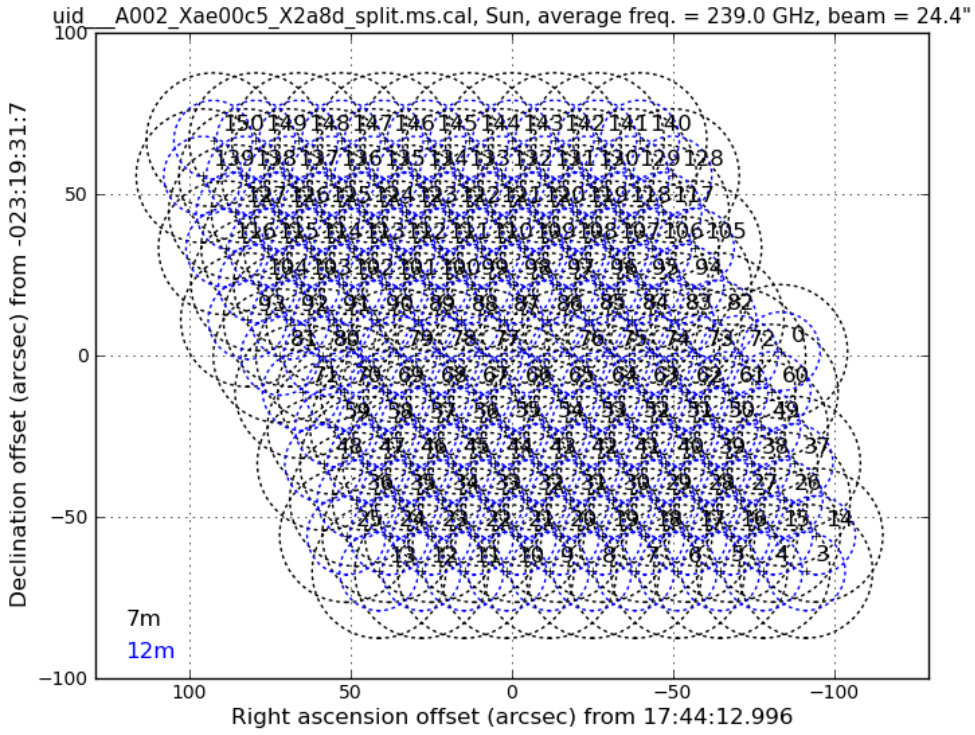

Figure 3. The pattern of mosaic BEFORE the re-calculation of the direction.

To correct the MOSAIC pattern, we re-calculate the coordinate of each field. At first, we modify the coordinate of the field “0” from the reference time

using fixplanets task. The reference time has to be the time when the antennas are directed to the field "0".

```
#In Casa
reftime = '2015/12/18/19:49:00'
fixplanets(msc, field ='0', fixuvw = T, refant=ref_ant, reftime = reftime)
```

We define that the modified coordinate of field '0' is as the reference coordinate, and re-calculate the coordinate of each field, as follows.

```
#In Casa
tb.open(msc+'/FIELD', nomodify=T)
phsCenOff = tb.getcol("PHASE_DIR")
tb.close()
refRaDec = aU.rad2radec(phsCenOff[0][0][0], phsCenOff[1][0][0],
                        prec=1, hmsdms=T, delimiter=' ')
for i in range(3, 151):
    raOff = phsCenOff[0][0][i] * 180./pi * 60. *60.
    deOff = phsCenOff[1][0][i] * 180./pi * 60. *60.
    offRaDec = aU.radec2deg(aU.radecOffsetToRadec(refRaDec, raOff,
                                                  deOff, prec=1))
    offRaDecF = 'J2000 ' + aU.deg2radec(offRaDec[0], offRaDec[1],
                                        prec=1, hmsdms=T, delimiter=' ')
    fixplanets(msc, field =str(i), fixuvw = T, direction = offRaDecF)

tb.open(msc+'/FIELD', nomodify=F)
tgt_refdir = tb.getcol("RefDir_Ref")
for id in range(3, len(tgt_refdir)):
    tb.putcell("RefDir_Ref", id, 21)
    tb.putcell("DelayDir_Ref", id, 21)
    tb.putcell("PhaseDir_Ref", id, 21)
tb.close()
```

Moreover, the direction in the pointing table has a bad influence to the coordinate system of the image synthesis. We erase the pointing table as follows:

```
#In Casa
tb.open(msc+'/POINTING', nomodify = False)
a = tb.rownumbers()
tb.removerows(a)
```

As a result, the pattern of the pointing in the MOSAIC observation becomes the map, as shown in Figure 4.

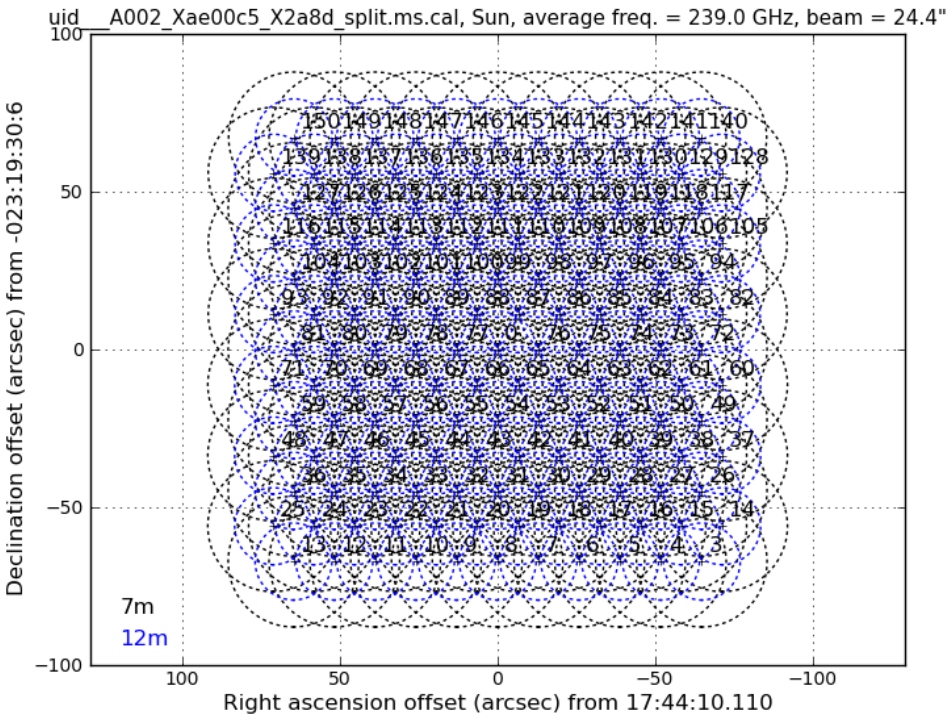

Figure 4. The pattern of mosaic AFTER the re-calculation of the direction.

12. Alternative way of the Calibration

If you do not want to cut-&-past all commands shown in above, you can use

the script included in the file Sunspot_Band6_UncalibratedData.tgz. The script can be executed as follows, after downloading the file and installing the Analysis Utilities package.
The script creates the calibrated visibility data, automatically.
Caution: You need to wait about one night (or day) for completing the script. Of course, the duration depends on the computer resources of your system.

```
# In a terminal outside CASA
tar -xvzf Sunspot_Band6_UncalibratedData.tgz

cd Sunspot_Band6_UncalibratedData

#Start CASA
casa

#In CASA
execfile("SunRedUtil.py")
execfile("SunspotBand6Cal.py")
```