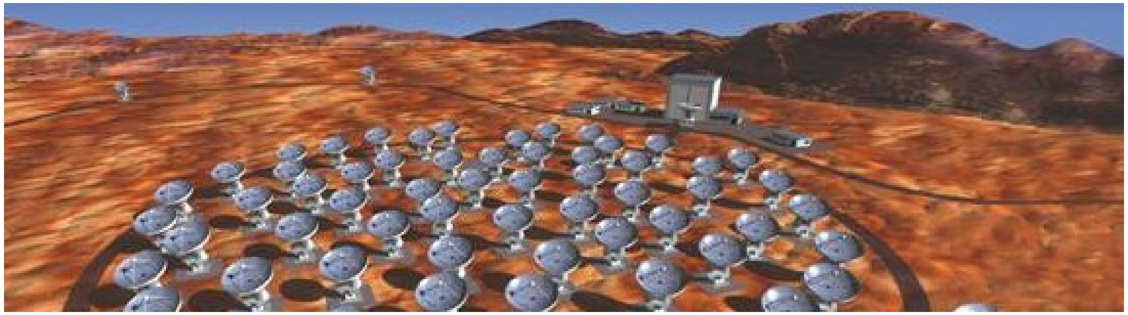


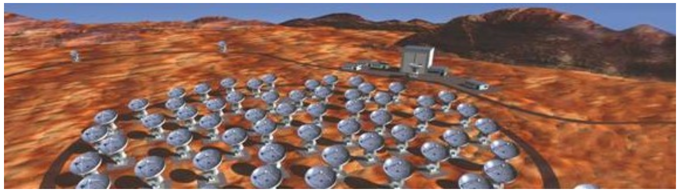
Calibration with CASA

Philippe Salomé

LERMA, Observatoire de Paris

Credits:

(Frédéric Gueth, George Moellenbrock, Wouter Vlemmings)

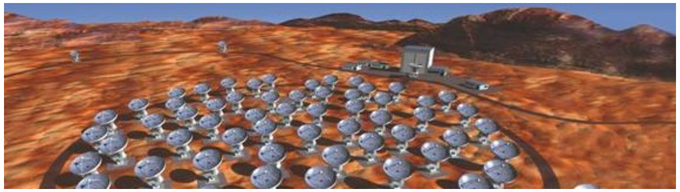


Calibration
On-line

Source of possible problems that may need flagging if not done automatically

- **Real-time calibrations:**
  - **Atmospheric absorption** (hot/cold measurements)
    The data are already in correct K scale
  - **Atmospheric phase fluctuations** (WVR)
    TelCal software (pointing focus, delays...)

- It will be possible to re-do some of these calibrations off-line but this should be reserved to special cases

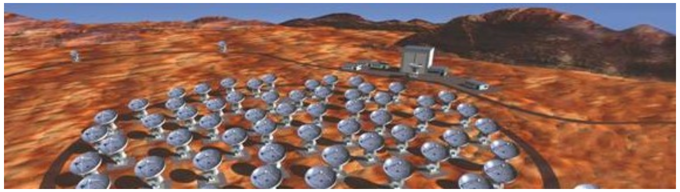


Calibration principles

Offline Calibration : your job

- Two main Calibrations for ALMA:
  - Freq.-dependent response of the system **bandpass**
  - Time-dependent response of the system **gaincal**

- Absolute flux scale must also be calibrated → **fluxscale**

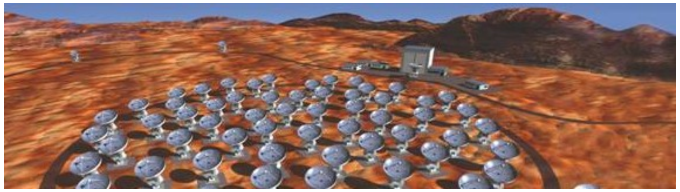


Calibration principles – on sky

- The whole calibration relies on models = **sources with known structure / spectra**

- In practice: continuum **point sources = quasars**
- Strong source is used to calibrate bandpass
- Nearby source regularly observed is used to calibrate gain temporal dependence
- Standard observing mode includes these observations

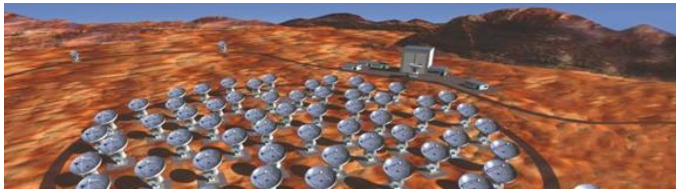
- Taking into account source structure/spectra is also possible: more complex models



Calibration principles - reduction

1. Select the appropriate data (field, spw, time,…)
2. Possibly apply existing calibration
3. Solve for calibration (bandpass, gaincal)
4. The calibration result is stored in a **new MS**
5. Go to 1. for next calibration

1. Apply calibration to science targets (= apply calibration MS to target MS)
   - **transfer among spectral windows**
   - **transfer among time = interpolation**



Data Calibration
Steps

Millemeter Interferometers

- **Bandpass** (amplitude and Phase vs frequency)

CASA: bandcal()

- **Phase** vs time

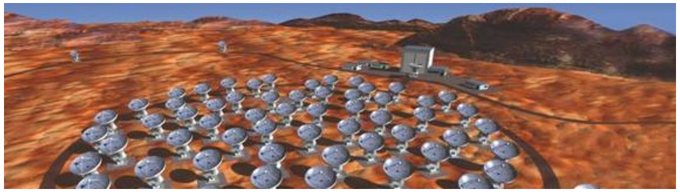
  CASA: gaincal()

- **Flux** scale

  CASA: setjy(), fluxscale()

- **Amplitude** vs time

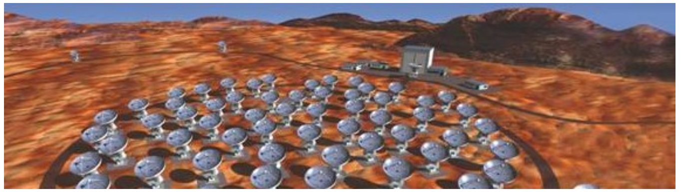
  CASA: gaincal()



Bandpass
Problems

Frequency dependence of the interferometer response arises from :

- Receivers intrinsic response
- Delay offsets (slope on phase)
- Cables attenuation
- Antenna chromatism
- Atmospheric lines (O2…)
- …

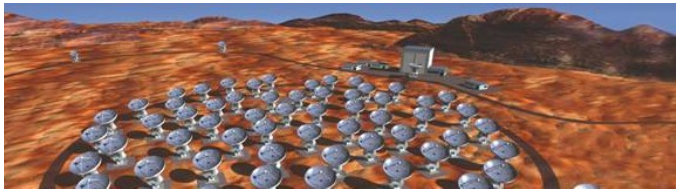


Bandpass
Method

A strong quasar is observed at the beginning of each project

- **Phase vs frequency should be zero** (point source)
- **Amplitude vs frequency should be constant** (continuum source)

Potential problem : spectral index of quasars over large bandwidth

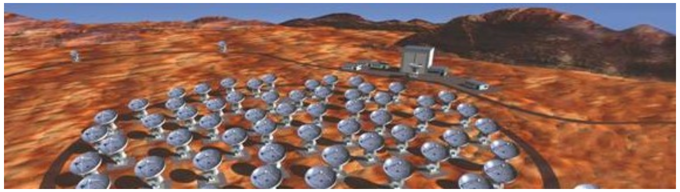


Bandpass
Method in practice

- Time average over one scan (to improve SNR)
- Time average over several scans (then need a phase calibration first of these scans)

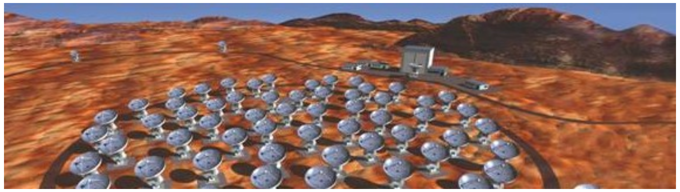
- **Solve for antenna based gains**
- **Fit amplitude and phase vs frequency** (polynoms)

- Assume bandpass is constant with time
- Must be recalibrated if receiver is retuned



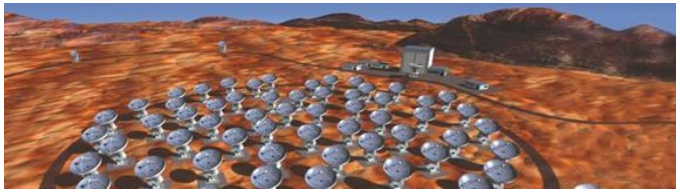
bandpass()

```
#  bandpass :: Calculates a bandpass calibration solution
vis                 = 'ngc5921.demo.ms' #  Nome of input visibility file
caltable            = 'ngc5921.demo.bcal' #  Name of output gain calibration
                                        #   table
field               =        '0'        #  Select field using field id(s) or
                                        #   field name(s)
spw                 =         ''        #  Select spectral window/channels
selectdata          =      False        #  Other data selection parameters
solint              =      'inf'        #  Solution interval
combine             =     'scan'        #  Data axes which to combine for solve
                                        #   (scan, spw, and/or field)
refant              =       '15'        #  Reference antenna name
minblperant         =          4        #  Minimum baselines _per antenna_
                                        #   required for solve
solnorm             =      False        #  Normalize average solution amplitudes
                                            #   to 1.0 (G, T only)
```



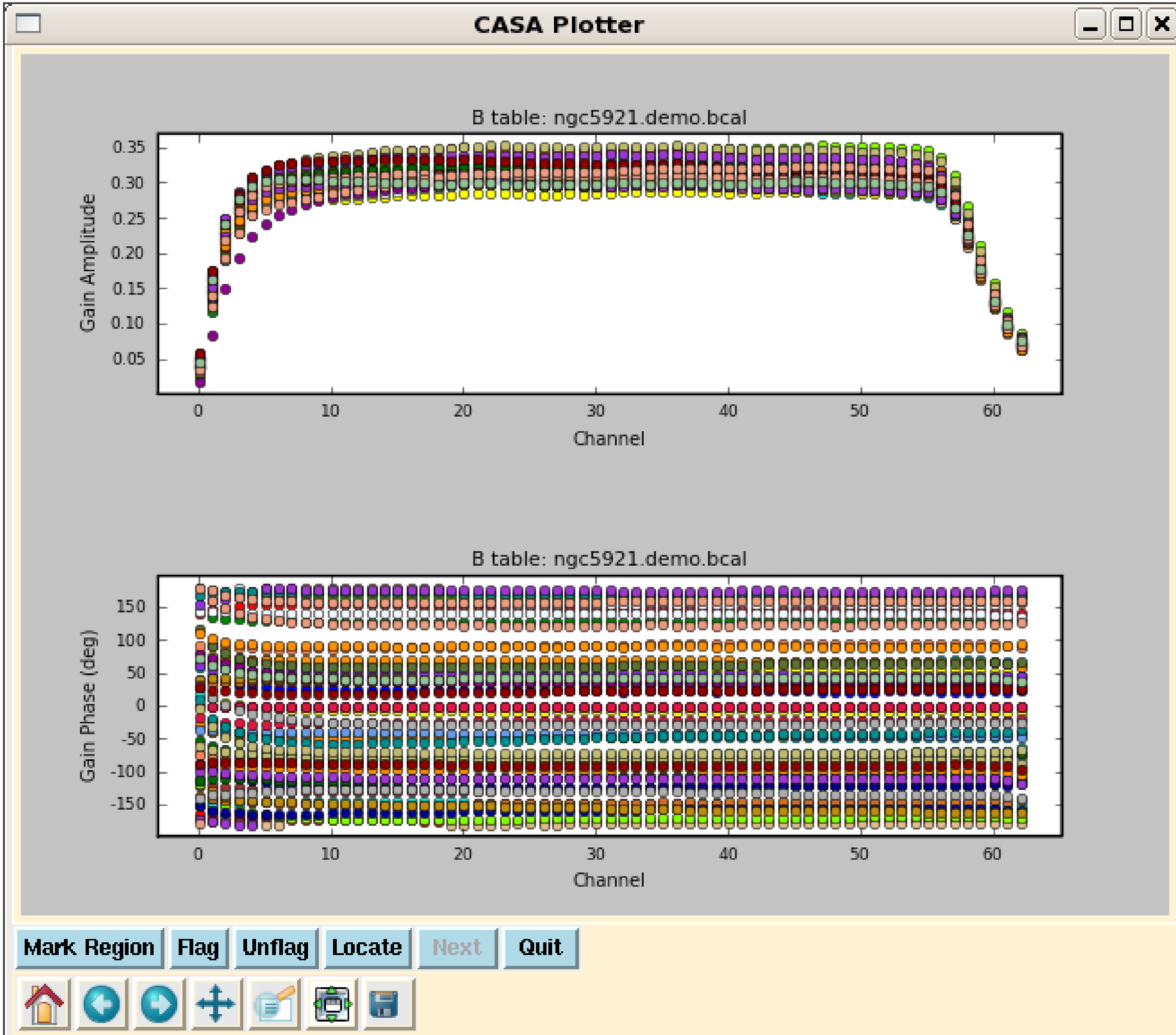
bandpass()

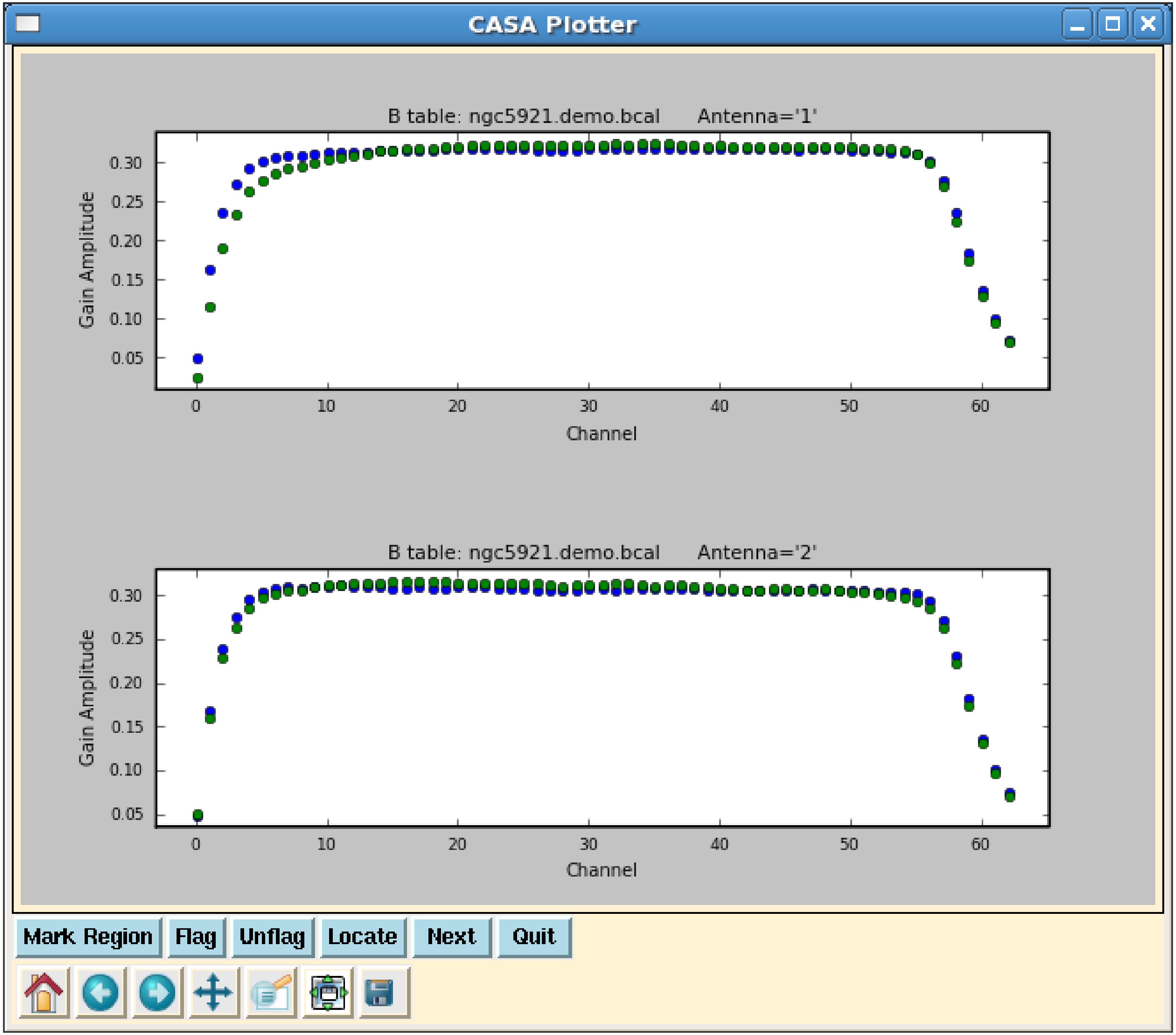
```
bandtype        =      'B'       #  Type of bandpass solution (B or
                                 #   BPOLY)
     fillgaps   =        0       #  Fill flagged solution channels by
                                 #   interpolation
append          =    False       #  Append solutions to the (existing)
                                 #   table
gaintable       =       ''       #  Gain calibration table(s) to apply on
                                                                 #   the fly
gainfield       =       ''       #  Select a subset of calibrators from
                                 #   gaintable(s)
interp          =       ''       #  Interpolation mode (in time) to use
                                 #   for each gaintable
spwmap          =       []       #  Spectral windows combinations to form
                                                         #   for gaintables(s)
gaincurve       =    False       #  Apply internal VLA antenna gain curve
                                         #   correction
opacity         =      0.0       #  Opacity correction to apply (nepers)
parang          =    False       #  Apply parallactic angle correction
async           =    False       #  If true the taskname must be started
                                 #   using bandpass(...)
```

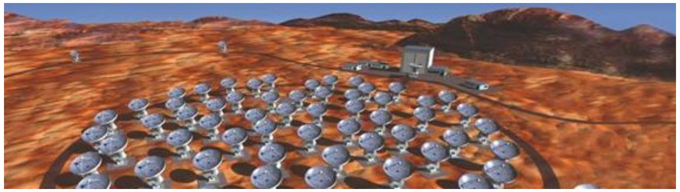


plotcal()

```
overplot          =      False        #  Overplot solutions on existing
                                      #   display
clearpanel        =     'Auto'        #  Specify if old plots are cleared or
                                      #   not
iteration         =         ''        #  Iterate plots on
                                      #   antenna,time,spw,field
plotrange         =         []        #  plot axes ranges:
                                      #   [xmin,xmax,ymin,ymax]
showflags         =      False        #  If true, show flagged solutions
plotsymbol        =        'o'        #  pylab plot symbol
plotcolor         =     'blue'        #  initial plotting color
markersize        =        5.0        #  Size of plotted marks
fontsize          =       10.0        #  Font size for labels
showgui           =       True        #  Show plot on gui
figfile           =         ''        #  ''= no plot hardcopy, otherwise
                                      #   supply name
async             =      False        #  If true the taskname must be started
                                      #   using plotcal(...)
```


CASA Plotter
B table: ngc5921.demo.bcal
Gain Amplitude
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0
10
20
30
40
50
60
Channel
B table: ngc5921.demo.bcal
Gain Phase (deg)
150
100
50
0
-50
-100
-150
0
10
20
30
40
50
60
Channel
Mark Region
Flag
Unflag
Locate
Next
Quit


CASA Plotter
B table: ngc5921.demo.bcal Antenna='1'
Gain Amplitude
Channel
B table: ngc5921.demo.bcal Antenna='2'
Gain Amplitude
Channel
Mark Region
Flag
Unflag
Locate
Next
Quit

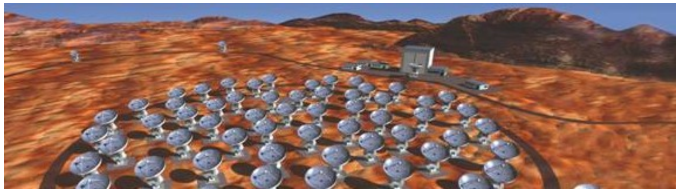


Gaincal – Phase
Problems

Short-term time variation of the phase caused by the atmosphere (< 1min : real time correction by water radiometer), 1min-1h not corrected but if fast switching)

Long-term time variation

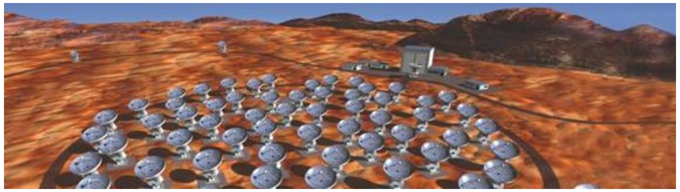
- Antenna position error (period 24h)
- Atmosphere (up to 1h) – **offline calibration (gaincal)**
- Antenna/electronics drifts



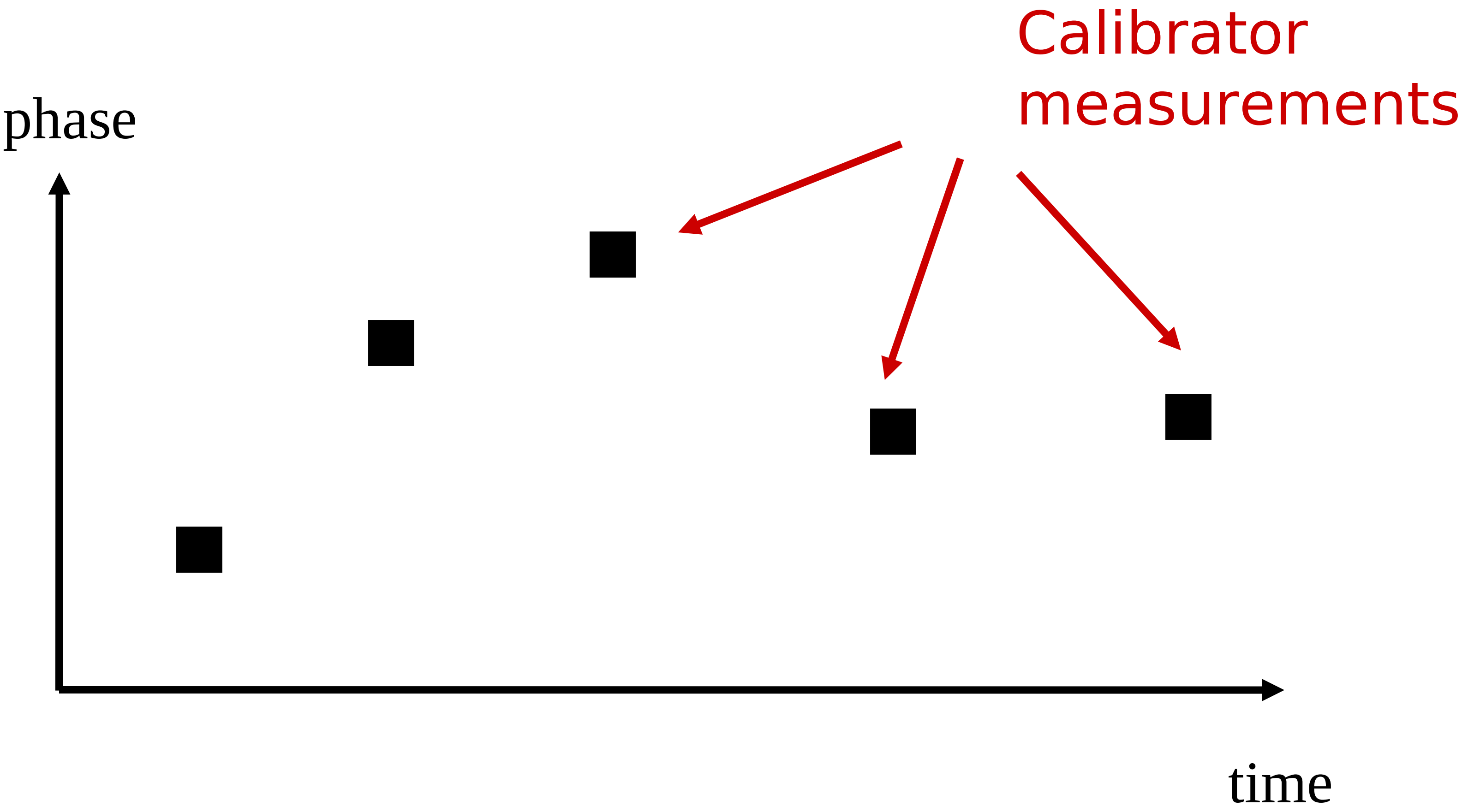
Gaincal – Phase
Method

A point source (quasar) is observed every ~20min

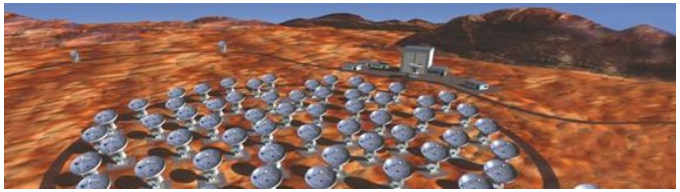
- **Phase should be zero** (point source)
- Solve for antenna-based gains
- **Fit as a function of time** (spline)
- Better: use of two calibrators
- Use previous calibration (bandpass)



Gaincal – Phase
Method

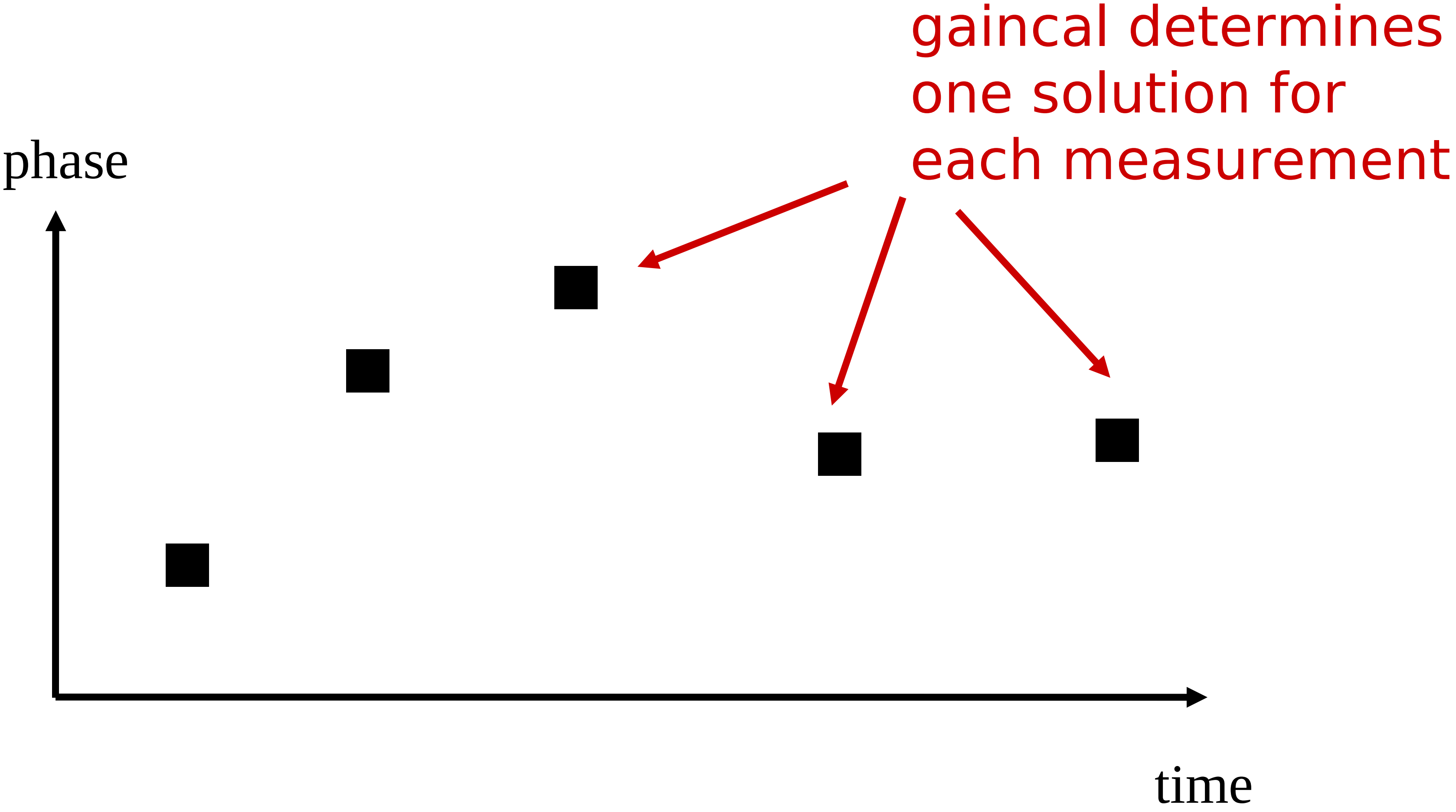


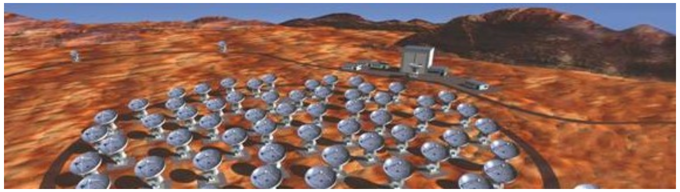
Astrophysical target is observed between the calibrators



Gaincal – Phase

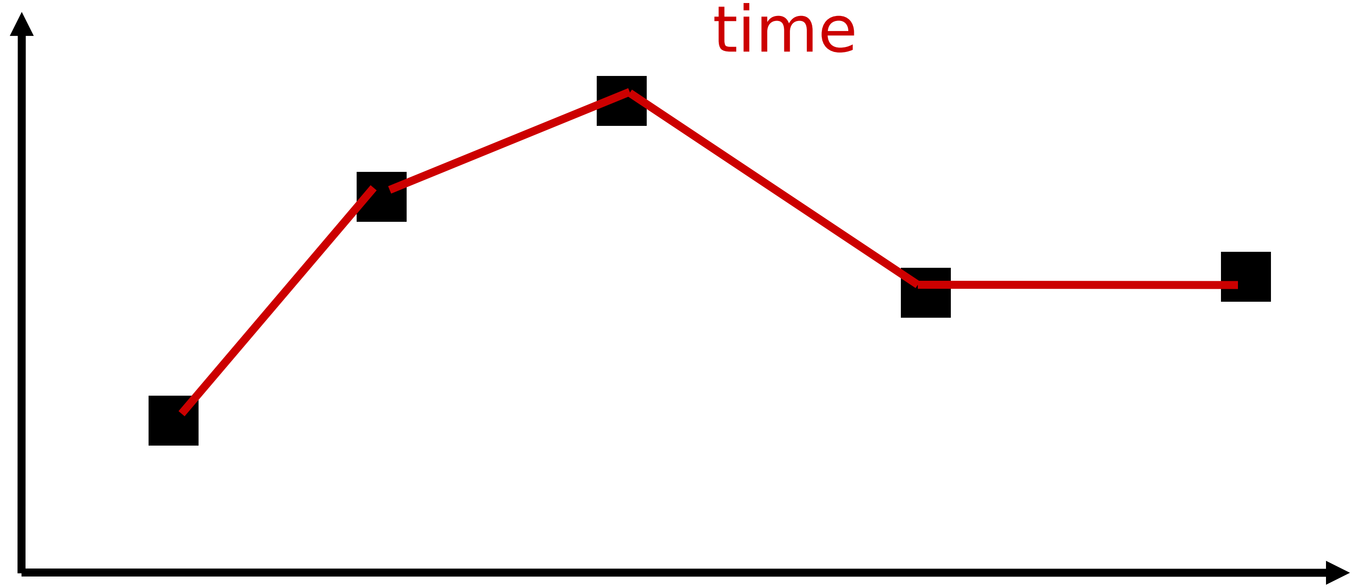
Method



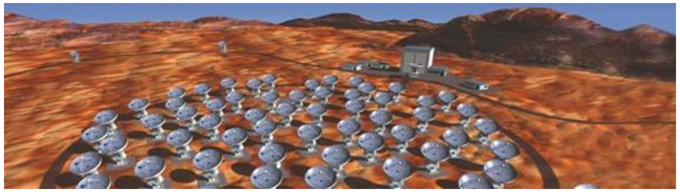


Gaincal – Phase
Method

applycal interpolates solution as a function of time

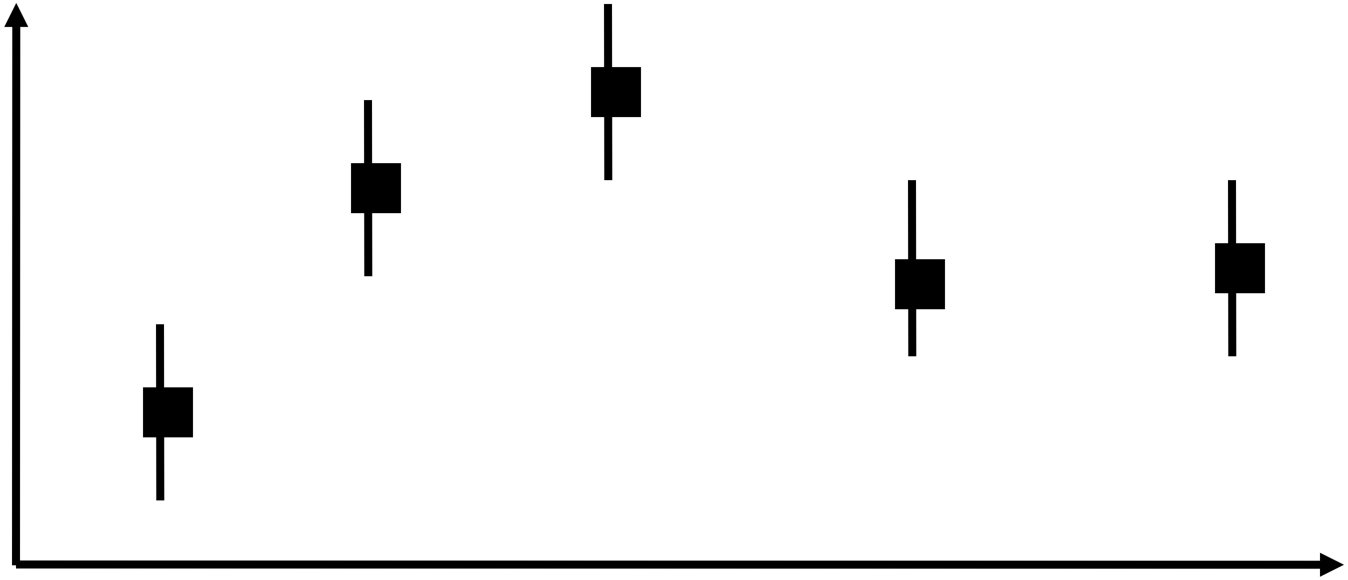


This assumes
- excellent SNR for each point
- no atmospheric phase



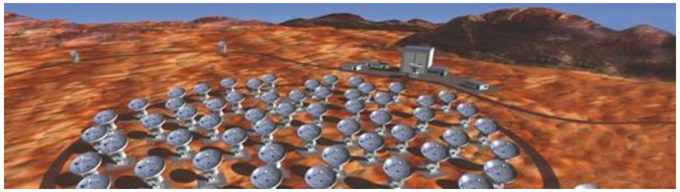
Gaincal – Phase
Method

phase

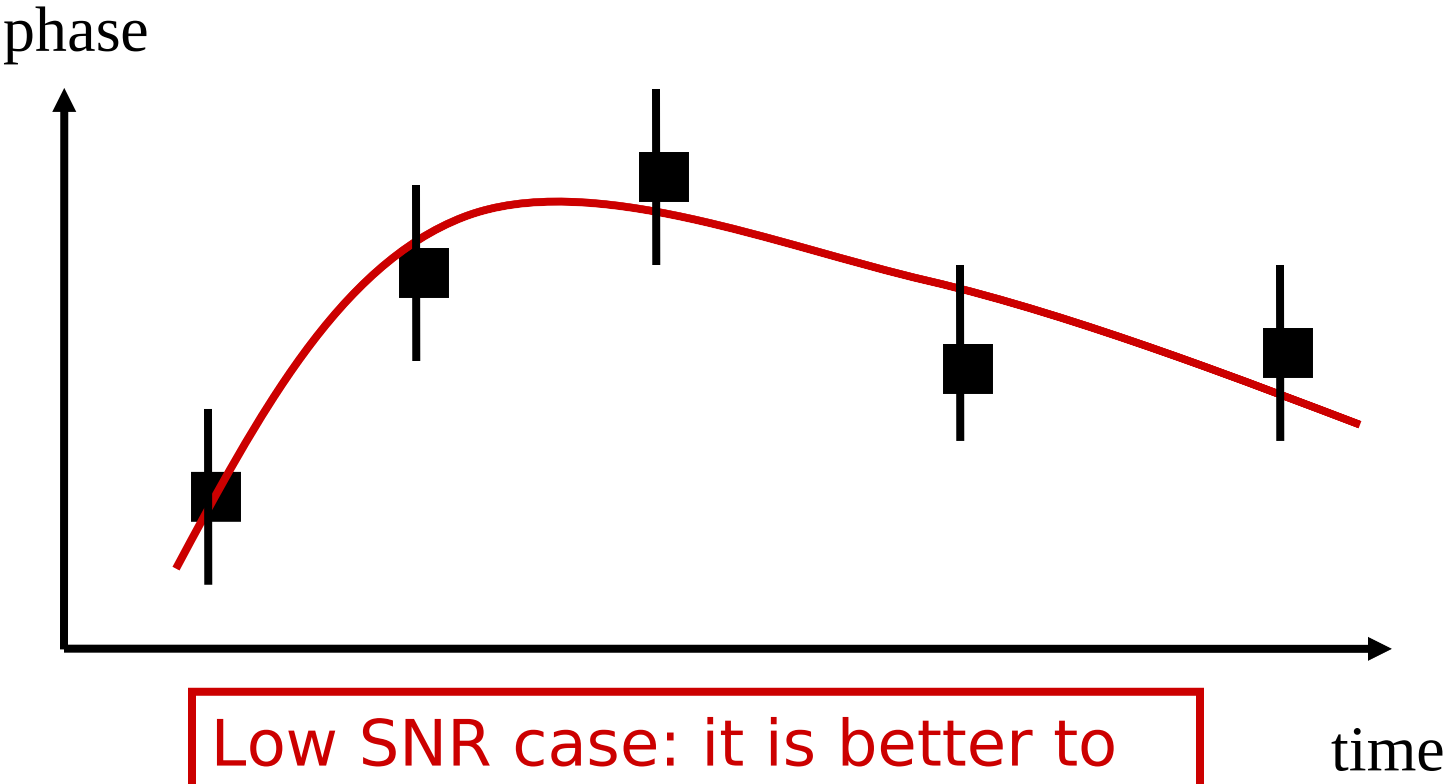


time

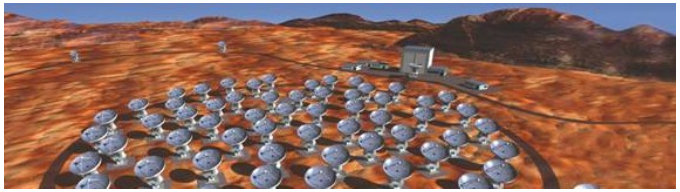
Low SNR case = millimeter case: each point has an error bar (thermal noise)



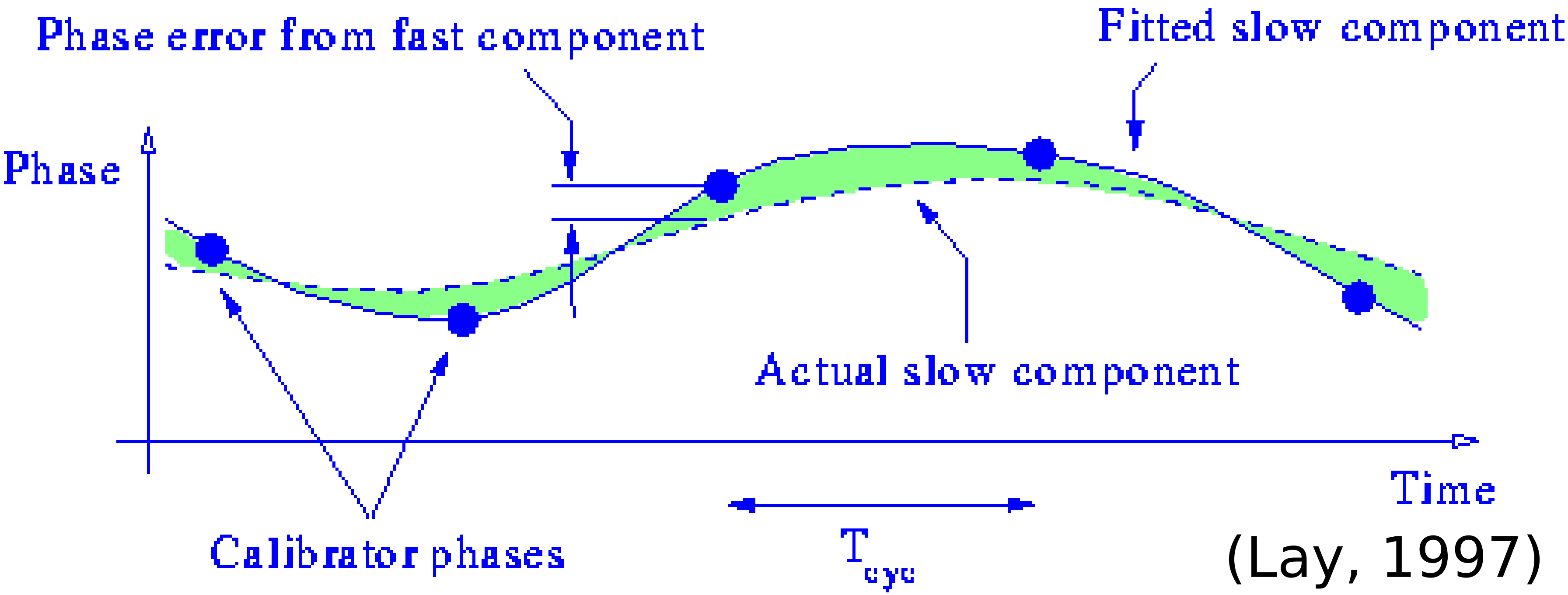
Gaincal – Phase
Method



Low SNR case: it is better to to fit a global, smoother solution

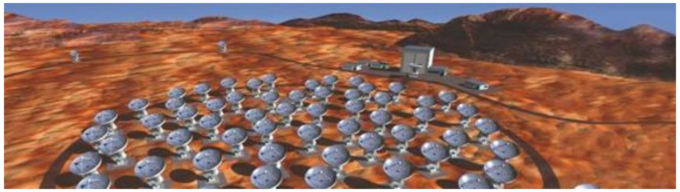


Gaincal – Phase
Method

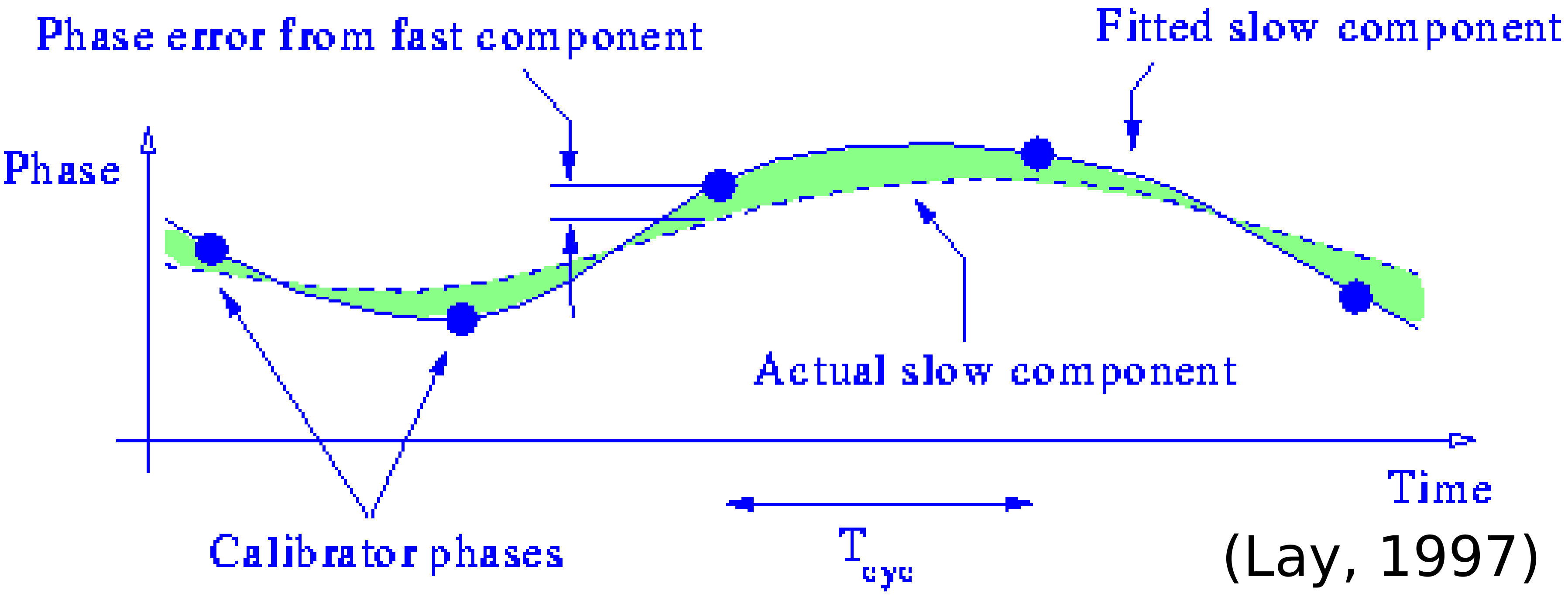


(Lay, 1997)

Phase is sampled at intervals Tc → fit is sensitive to errors due to the presence of the fast component (<2Tc), which can be large

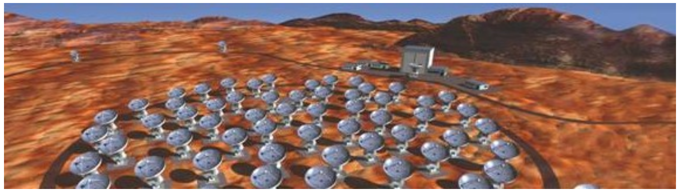


Gaincal – Phase
Method



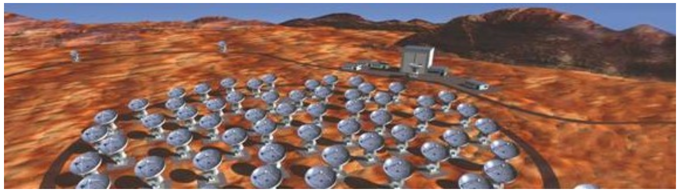
(Lay, 1997)

A solution going exactly through all points includes short-timescale noise aliased into the slow component



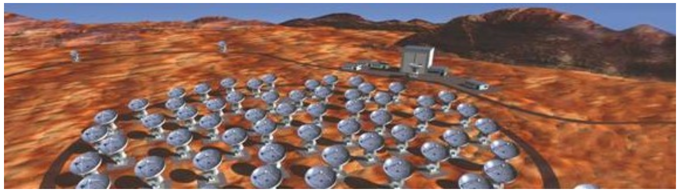
gaincal()

```
#  gaincal :: Determine temporal gains from calibrator observations
vis                 = 'ngc5921.demo.ms' #  Nome of input visibility file
caltable            = 'ngc5921.demo.gcal' #  Name of output gain calibration
                                        #   table
field               =      '0,1'        #  Select field using field id(s) or
                                        #   field name(s)
spw                 =   '0:6~56'        #  Select spectral window/channels
selectdata          =      False        #  Other data selection parameters
solint              =      'inf'        #  Solution interval (see help)
combine             =         ''        #  Data axes which to combine for solve
                                        #   (scan, spw, and/or field)
preavg              =       -1.0        #  Pre-averaging interval (sec)
refant              =       '15'        #  Reference antenna name
minblperant         =          4        #  Minimum baselines _per antenna_
                                        #   required for solve
minsnr              =        1.0        #  Reject solutions below this SNR
solnorm             =      False        #  Normalize average solution amplitudes
                                                #   to 1.0 (G, T only)
```



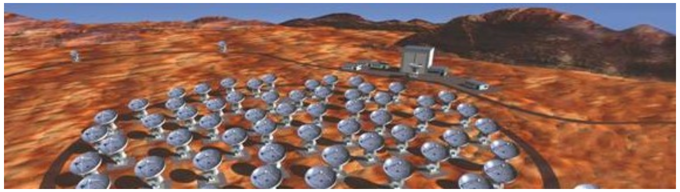
fluxscale – Flux Problems

- **All quasars have varying fluxes** (several 10% in a few months) and spectral indexes – Case of several configuration observations separated by months
- **Cannot rely on a priory antenna efficiency** to measure their flux (decorrelation...)
- Need to measure **known fluxes** against monitored Planets, strong, strong quasars...be careful if source are resolved
- **Difficult** part of the calibration



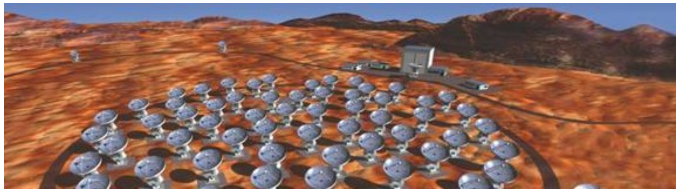
setjy()

```
#  setjy ::
vis                  = 'ngc5921.demo.ms' #  Name of input visibility file (MS)
field                = '1331+305*'       #  Field name list or field ids list
spw                  =         ''        #  Spectral window identifier (list)
modimage             =         ''        #  File location for field model
fluxdensity          =         -1        #  Specified flux density [I,Q,U,V]; -1
                                         #   will lookup values
standard             = 'Perley-Taylor 99' #  Flux density standard
async                =      False        #  If true the taskname must be started
                                         #   using setjy(...)
```



fluxscale()

```
#  fluxscale :: Bootstrap the flux density scale from standard calibrators
vis                 = 'ngc5921.demo.ms' #  Name of input visibility file (MS)
caltable            = 'ngc5921.demo.gcal' #  Name of input calibration table
fluxtable           = 'ngc5921.demo.fluxscale' #  Name of output, flux-scaled
                                        #   calibration table
reference           =    '1331*'        #  Reference field name(s) (transfer
                                        #   flux scale FROM)
transfer            =    '1445*'        #  Transfer field name(s) (transfer
   flux                                                            #   scale
   TO), '' -> all
append              =      False        #  Append solutions?
refspwmap           =       [-1]        #  Scale across spectral window
                                        #   boundaries.  See help fluxscale
async               =      False        #  If true the taskname must be started
                                        #   using fluxscale(...)
```

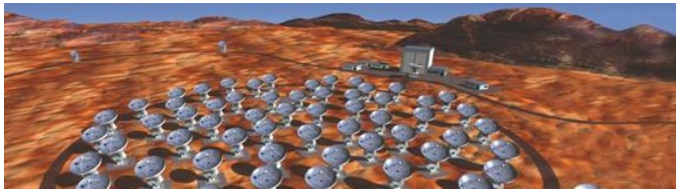


Gaincal – Amplitude
Problems

Temperature (K) → Flux (Jansky)

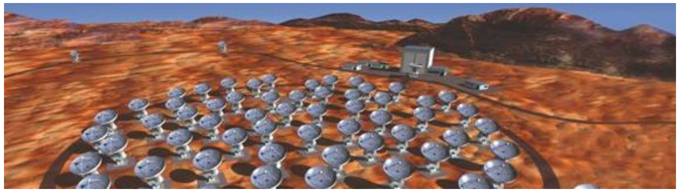
- Scaling by **antenna efficiency** (Jy/K)
- **Not enough** for mm-interferometers because
  - Amplitude loss due to decorrelation
  - Variation of the antenna gain (pointing, focus)

- Need **amplitude referencing to a point source** (quasar) to calibrate the **time variation** of the antenna efficiency (cf phase calib)



gaincal()

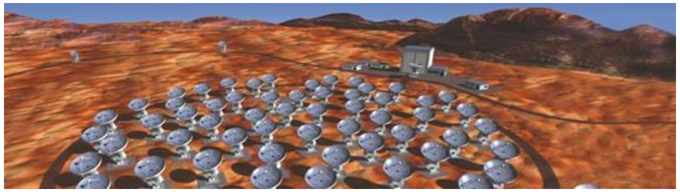
```
#  gaincal :: Determine temporal gains from calibrator observations
vis                 = 'ngc5921.demo.ms' #  Nome of input visibility file
caltable            = 'ngc5921.demo.gcal' #  Name of output gain calibration
                                        #   table
field               =      '0,1'        #  Select field using field id(s) or
                                        #   field name(s)
spw                 =   '0:6~56'        #  Select spectral window/channels
selectdata          =      False        #  Other data selection parameters
solint              =      'inf'        #  Solution interval (see help)
combine             =         ''        #  Data axes which to combine for solve
                                        #   (scan, spw, and/or field)
preavg              =       -1.0        #  Pre-averaging interval (sec)
refant              =       '15'        #  Reference antenna name
minblperant         =          4        #  Minimum baselines _per antenna_
                                        #   required for solve
minsnr              =        1.0        #  Reject solutions below this SNR
solnorm             =      False        #  Normalize average solution amplitudes
                                                #   to 1.0 (G, T only)
```



gaincal()

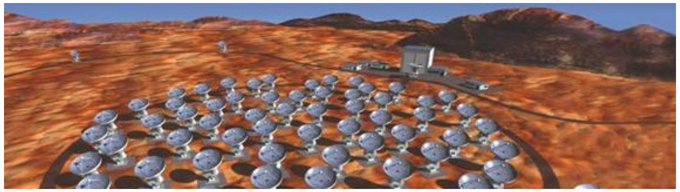
Use 'a' only here...

```
gaintype     =          'G'        #  Type of gain solution (G, T, or
                                   #   GSPLINE)
calmode      =         'ap'        #  Type of solution" ('ap', 'p', 'a')
append       =        False        #  Append solutions to the (existing)
                                   #   table
gaintable    = 'ngc5921.demo.bcal' #  Gain calibration table(s) to apply
                                   #   on the fly
gainfield    =           ''        #  Select a subset of calibrators from
                                   #   gaintable(s)
interp       =    'nearest'        #  Interpolation mode (in time) to use
                                   #   for each gaintable
spwmap       =           []        #  Spectral windows combinations to form
                                   #   for gaintables(s)
gaincurve    =        False        #  Apply internal VLA antenna gain curve
                               #   correction
opacity      =          0.0        #  Opacity correction to apply (nepers)
parang       =        False        #  Apply parallactic angle correction
async        =        False        #  If true the taskname must be started
                                   #   using gaincal(...)
```



plotcal()

```
#   plotcal :: An all-purpose plotter for calibration results
caltable            = 'ngc5921.demo.bcal' #  Name of input calibration table
xaxis               =          ''       #  Value to plot along x axis(time,chan
                                                #
   ,freq,antenna,amp,phase,real,imag,sn
      #   r)
yaxis               =     'phase'       #  Value to plot along y axis
                                        #   (amp,phase,real,imag,snr,antenna)
poln                =          ''       #  Antenna polarization to plot
                                        #   (RL,R,L,XY,X,Y,/)
field               =         '0'       #  field names or index of calibrators:
                                        #   ''==>all
antenna             =          ''       #  antenna/baselines: ''==>all, antenna
                                        #   = '3,VA04'
spw                 =          ''       #  spectral window:channels: ''==>all,
                                        #   spw='1:5~57'
timerange           =          ''       #  time range: ''==>all
subplot             =         212       #  Panel number on display screen (yxn)
```

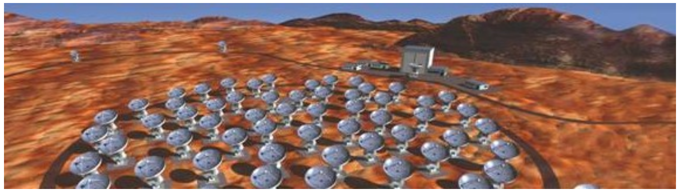


Calibration principles - reduction

1. Select the appropriate data (field, spw, time,…)
2. Possibly apply existing calibration
3. Solve for calibration (bandpass, gaincal)
4. The calibration result is stored in a **new MS**
5. Go to 1. for next calibration

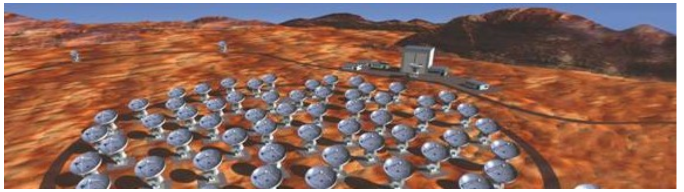
Apply calibration to science targets (= apply calibration MS to target MS)

- **transfer among spectral windows**
- **transfer among time = interpolation**



applycal()

```
#  applycal :: Apply calibrations solutions(s) to data
vis                 = 'ngc5921.demo.ms' #  Nome of input visibility file
field               =       '0'         #  Select field using field id(s) or
                                        #   field name(s)
spw                 =        ''         #  Select spectral window/channels
selectdata          =     False         #  Other data selection parameters
Gaintable           = ['ngc5921.demo.fluxscale', 'ngc5921.demo.bcal'] #  Gain ca
                                           #   libration table(s) to apply on the
                                        #   fly
gainfield           = ['0', '*']        #  Select a subset of calibrators from
                                        #   gaintable(s)
interp              = ['linear', 'nearest'] #  Interpolation mode (in time) to
                                        #   use for each gaintable
spwmap              =        []         #  Spectral windows combinations to form
                                           #   for gaintables(s)
gaincurve           =     False         #  Apply internal VLA antenna gain curve
                                                   #   correction
opacity             =       0.0         #  Opacity correction to apply (nepers)
parang              =     False         #  Apply parallactic angle correction
calwt               =      True         #  Calibrate weights along with data for
                                           #   all relevant calibrations
async               =     False         #  If true the taskname must be started
                                        #   using applycal(...)
```



To summarize...

…open data file…

…select data…

bandpass( )

plotcal ( )

…select data…

gaincal( )

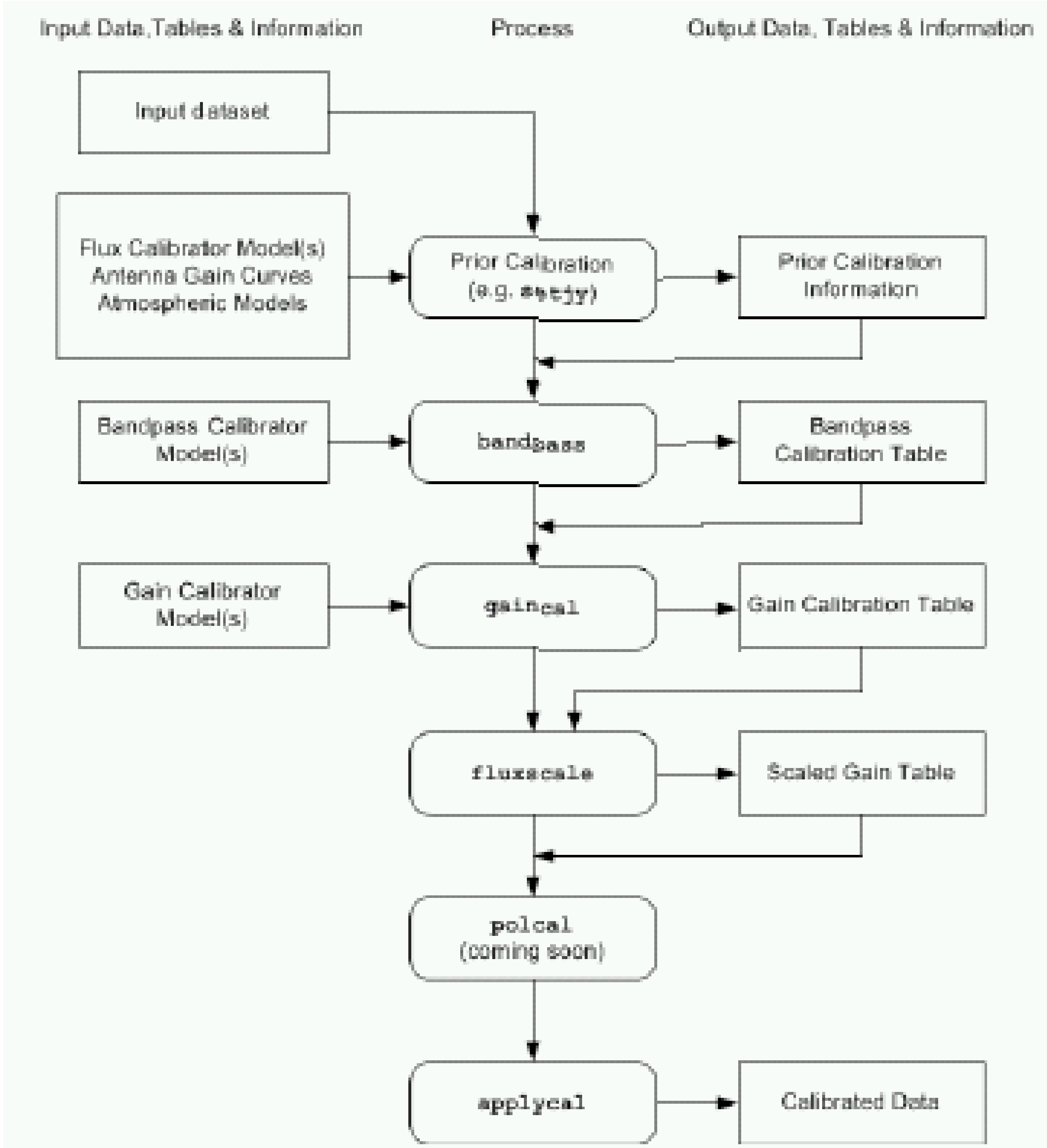
fluxscale( )

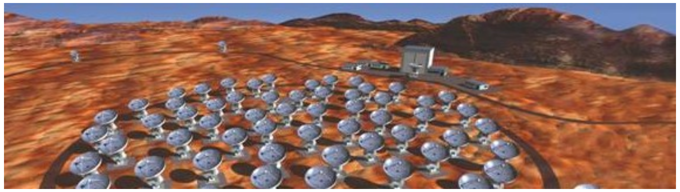
plotcal ( )

…select calibration and target…

applycal( )

… imaging …


Input Data,Tables & Information
Process
Output Data, Tables & Information
Input dataset
Flux Calibrator Model(s)
Antenna Gain Curves
Atmospheric Models
Prior Calibration
(e.g. setjy)
Prior Calibration
Information
Bandpass Calibrator
Model(s)
bandpass
Bandpass
Calibration Table
Gain Calibrator
Model(s)
gaincal
Gain Calibration Table
fluxscale
Scaled Gain Table
polcal
(coming soon)
applycal
Calibrated Data



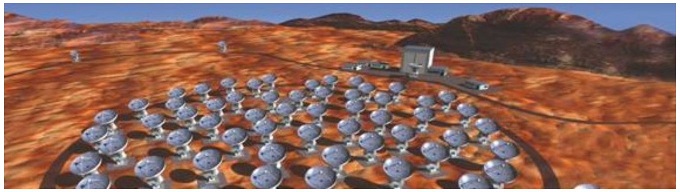
More ...

Exercices

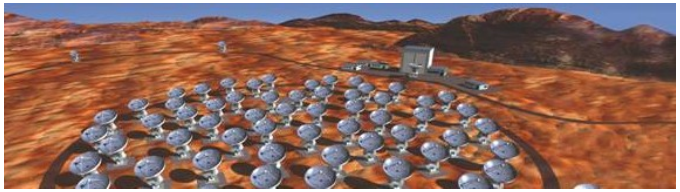
- Go through the ngc5921_demo.py script (first part until the clean() task)

References

- NRAO Lectures
http://www.cv.nrao.edu/course/astr534/ERA.shtml
- IRAM schools (2010 Oct. 4th -8th , Grenoble, France)
http://www.iram.fr/IRAMFR/IS/school.htm



The end !



Exercices

- Go through and play with the ngc3921_demo.py script

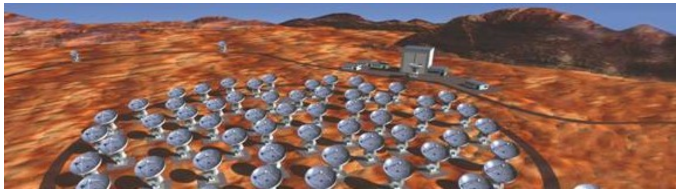
Try to plug a more complex calibration scheme:

preliminary gaincal on 1331+305

bandpass

final gaincal on 1331+305 & 1445+099

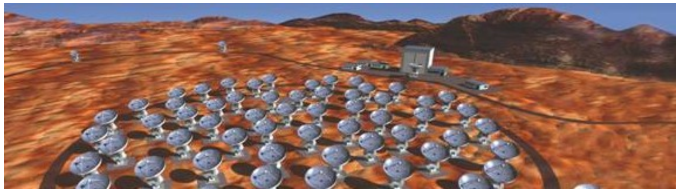
ngc4826 script



Formalism

- **Observation:** $V_{obs} = G V_{true} + N$
  - V_{true} = true visibilities = FT(sky)
  - V_{obs} = observed visibilities
  - Usually, G can be decomposed into antenna-based terms: $G = G_{ij} = G_i \times G_j^*$

- *Calibration:* $V_{corr} = G'^{-1} V_{obs}$

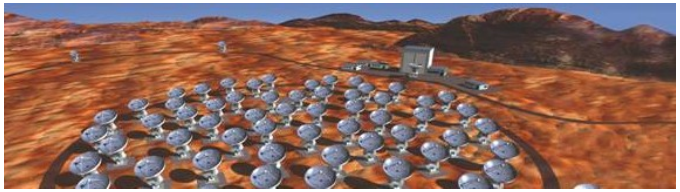


Formalism

- **Observation:** $V_{obs} = GV_{true} + N$
  - V_{true} = true visibilities = FT(sky)

V are complex numbers

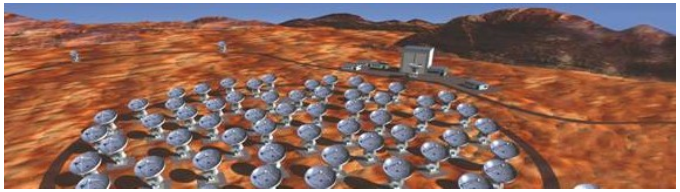
G are matrixes



Formalism

$$V_{obs} = M^{f}MBGDPETF\ V_{true} + Noise$$

- F – ionospheric Faraday rotation
- T – tropospheric effects
- E – collecting area
- D – instrumental polarization
- P – parallactic angle
- G – electronic gain
- B – bandpass
- M – baseline-based gain
- M^f – baseline-based bandpass
- N – additive noise (thermal, RFI)



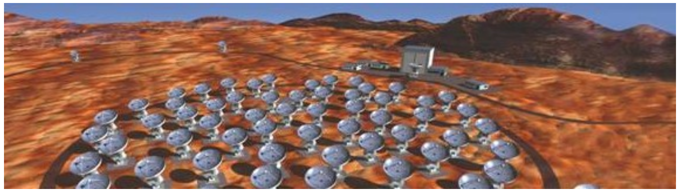
Formalism

$$V_{obs} = M^{f}MBGDPETF\ V_{true} + Noise$$

- F – ionospheric Faraday rotation
- T – tropospheric effects
- E – collecting area
- D – instrumental polarization
- P – parallactic angle
- G – electronic gain
- B – bandpass
- M – baseline-based gain
- M^f – baseline-based bandpass
- N – additive noise (thermal, RFI)

Antenna-based effects

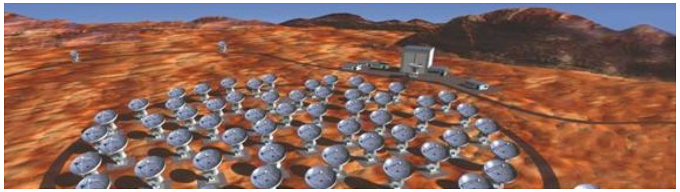
Baseline-based effects (bad!)



Formalism

$$V_{obs} = M^f MBGDPETF \, V_{true} + Noise$$

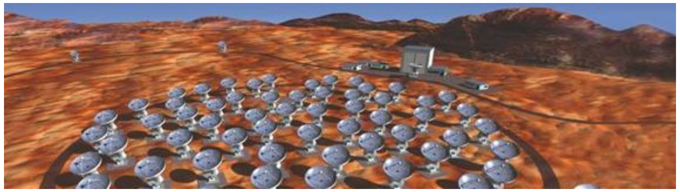
- F – ionospheric Faraday rotation → not for ALMA
- T – tropospheric effects → real-time calibration
- E – collecting area → included in imaging
- D – instrumental polarization → polarimetry
- P – parallactic angle → polarimetry
- **G – electronic gain → main calibration for ALMA**
- **B – bandpass → main calibration for ALMA**
- M – baseline-based gain → if strong decorrelation
- M^f – baseline-based bandpass → should not happen
- N – additive noise (thermal, RFI)



Calibration principles

- Calibration of Vobs = J V
  - **Select data** so that expected visibilities Vmodel are known (eg, point source)
  - Apply already known calibration (eg, gaincal)
  - Solve for J (eg, bandpass) **for each antenna**

- In general, calibrations do not commute
  - **Use orthogonality, timescale, source properties,… to separate**



Calibration principles

- Calibr
  - **Sel** cted visibilities Vm oint source)
  - App oration (eg, gaincal)
  - Sol **for each antenna**

$J = J_{ij} = J_i \times J_j^*$

N unknown

N(N-1)/2 measurements

- In general, calibrations do not commute
  - **Use orthogonality, timescale, source properties,… to separate**