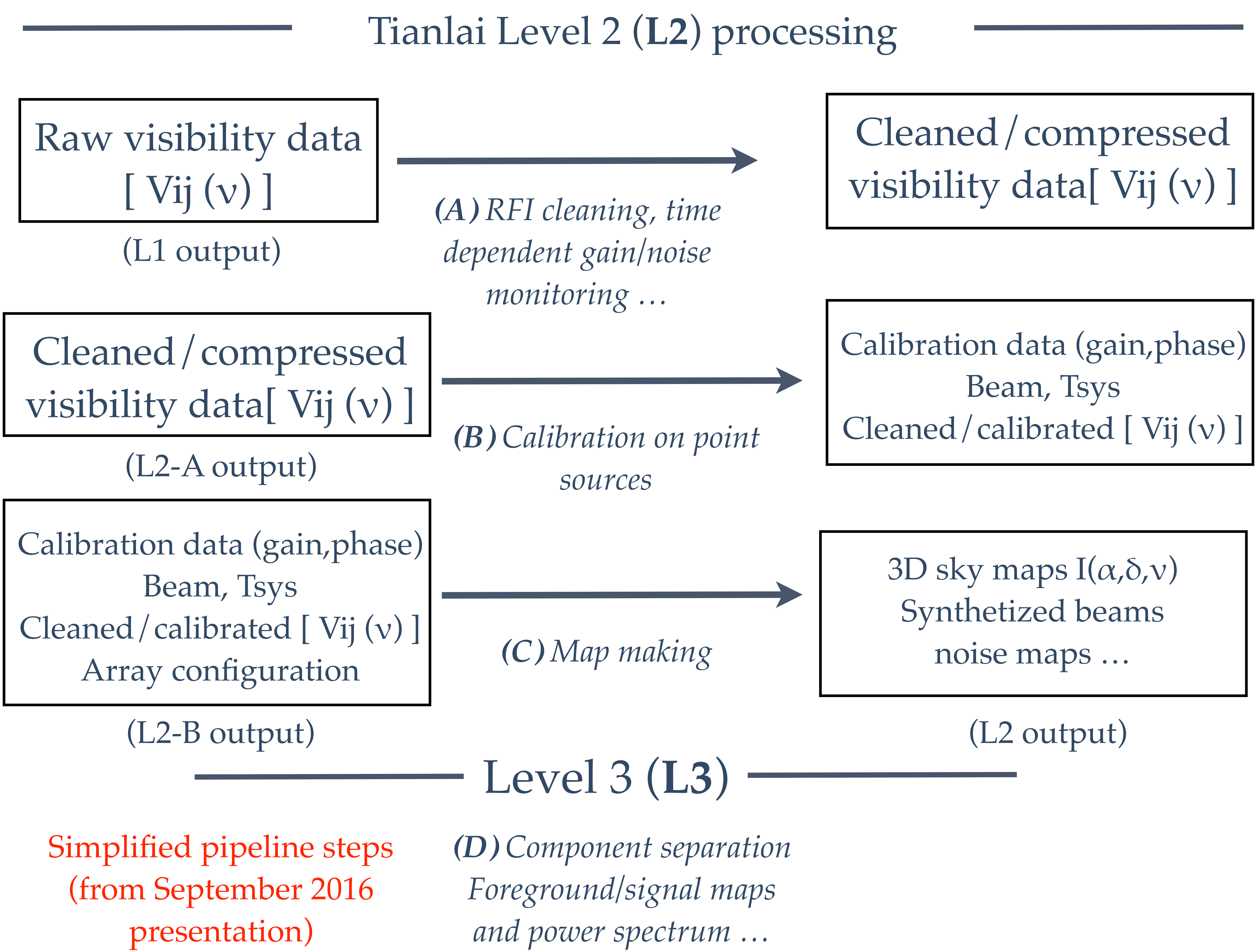
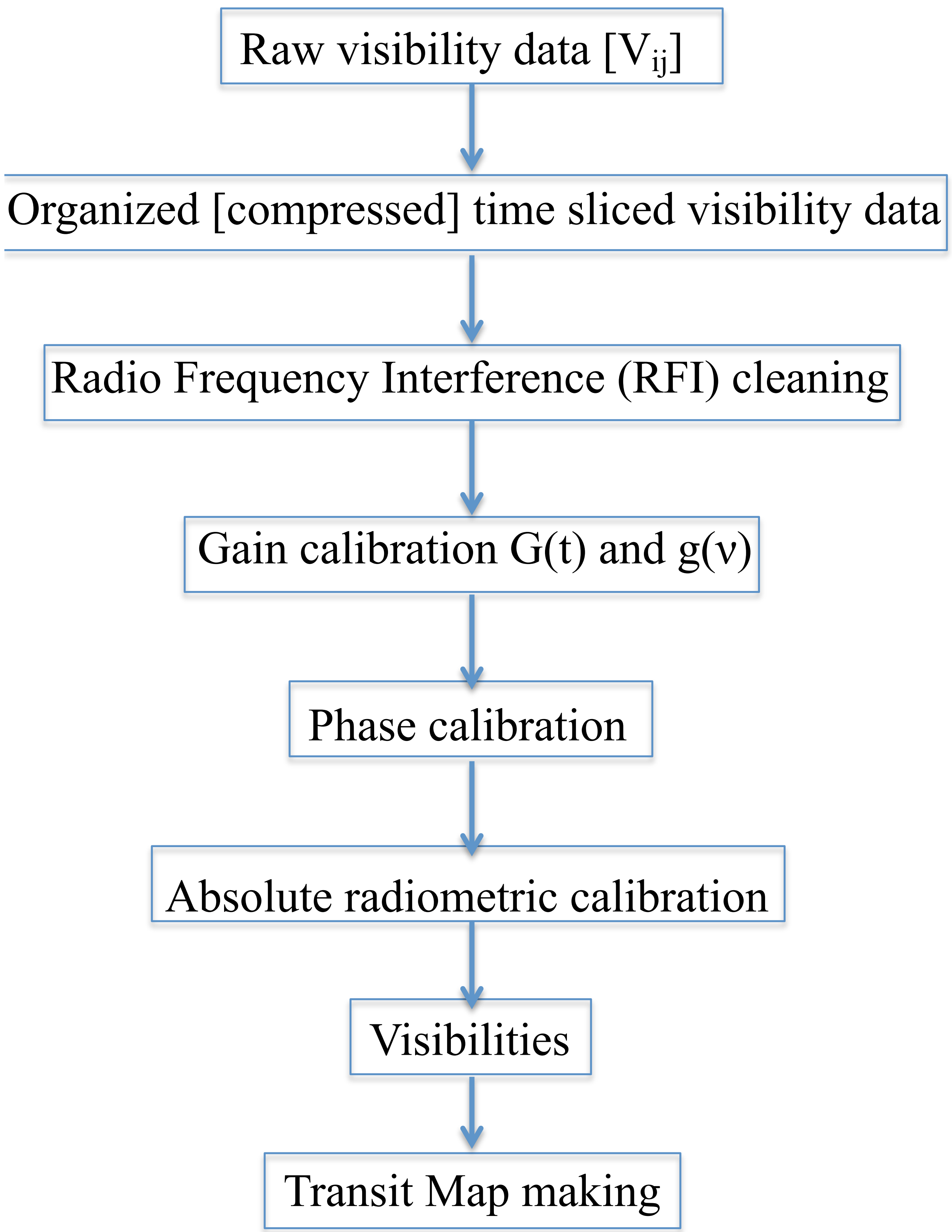
Analyse PAON-4
Brain storming

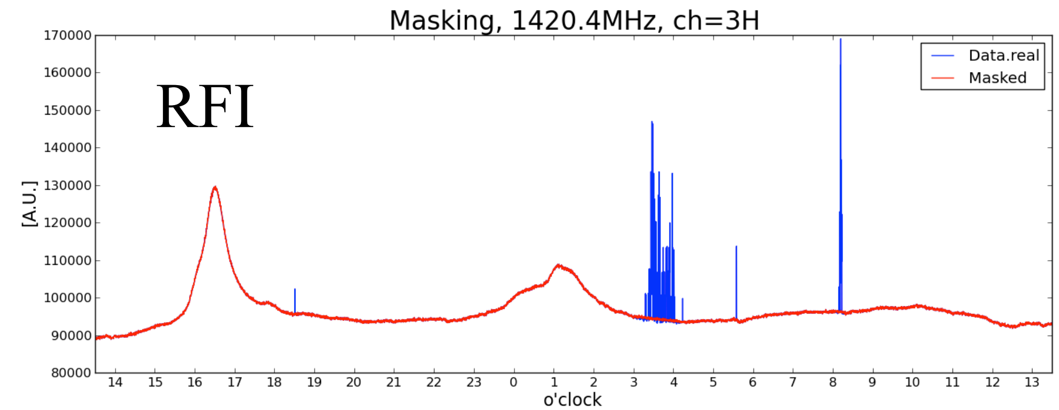
Septembre 2017

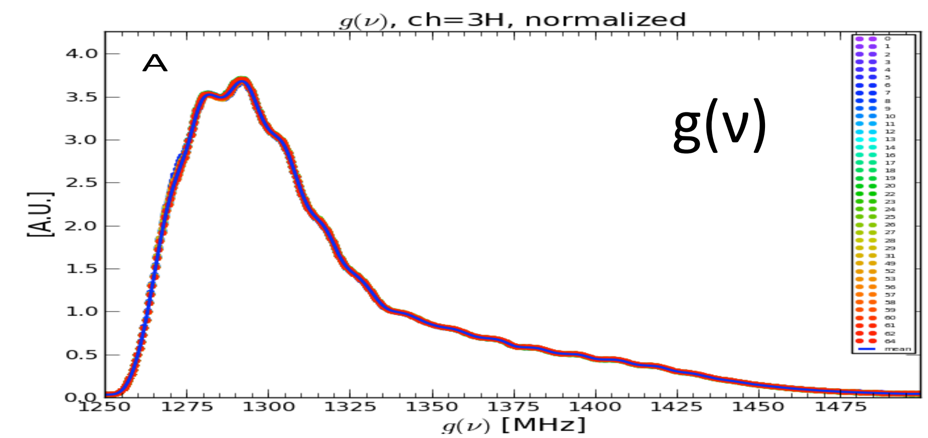
Chaîne d'Analyse

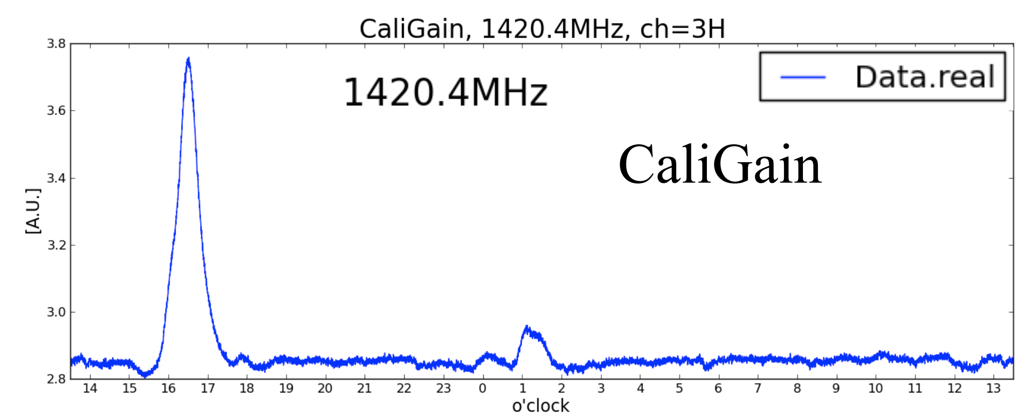

Tianlai Level 2 (L2) processing
Raw visibility data [Vij (ν)]
(L1 output)
(A) RFI cleaning, time dependent gain/noise monitoring …
Cleaned / compressed visibility data[Vij (ν)]
Cleaned / compressed visibility data[Vij (ν)]
(L2-A output)
(B) Calibration on point sources
Calibration data (gain,phase) Beam, Tsys Cleaned / calibrated [Vij (ν)]
Calibration data (gain,phase) Beam, Tsys Cleaned / calibrated [Vij (ν)] Array configuration
(L2-B output)
(C) Map making
3D sky maps I(α,δ,ν) Synthetized beams noise maps …
(L2 output)
Level 3 (L3)
Simplified pipeline steps (from September 2016 presentation)
(D) Component separation Foreground/signal maps and power spectrum …

Data cleaning and calibration procedure









Slide from Jiao's PhD defence
June 2017

IV.1

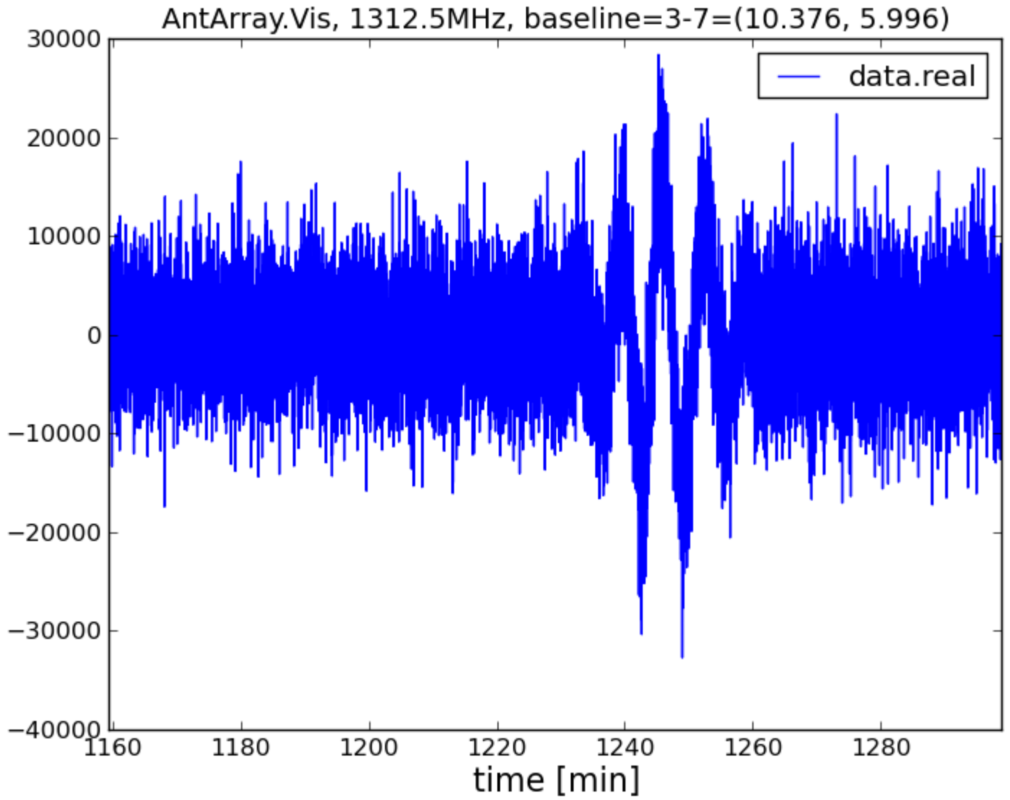
Data

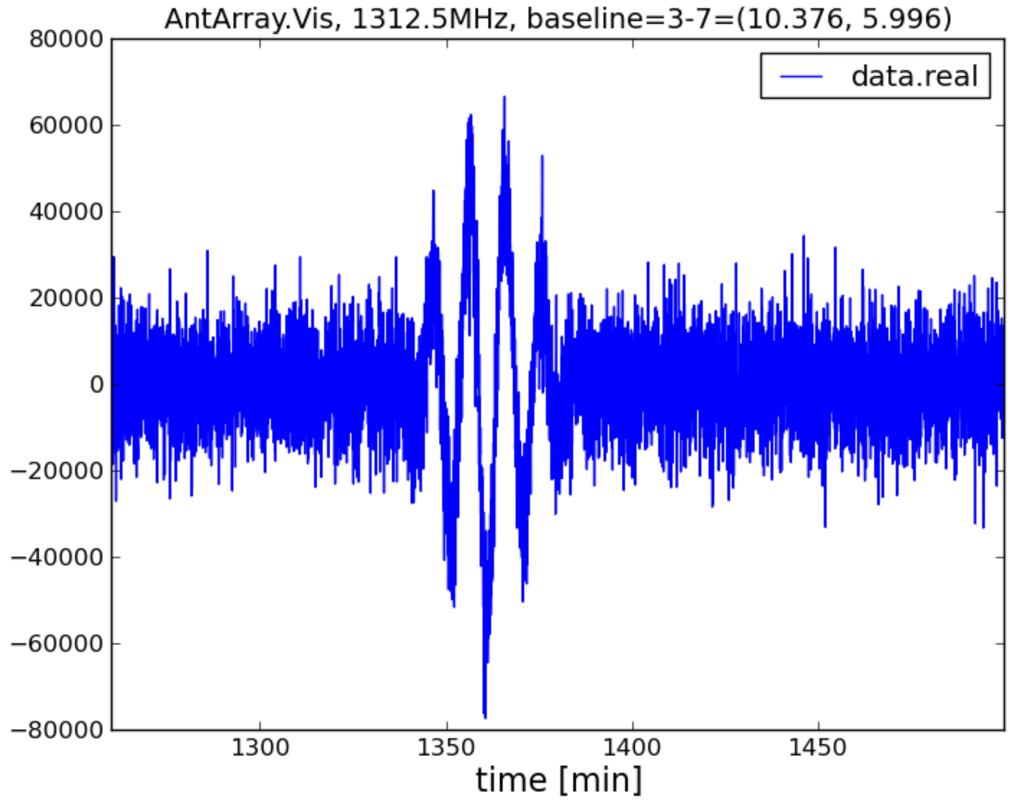
- CygA6sep16

~1s

CasA29sep16

~2.2s



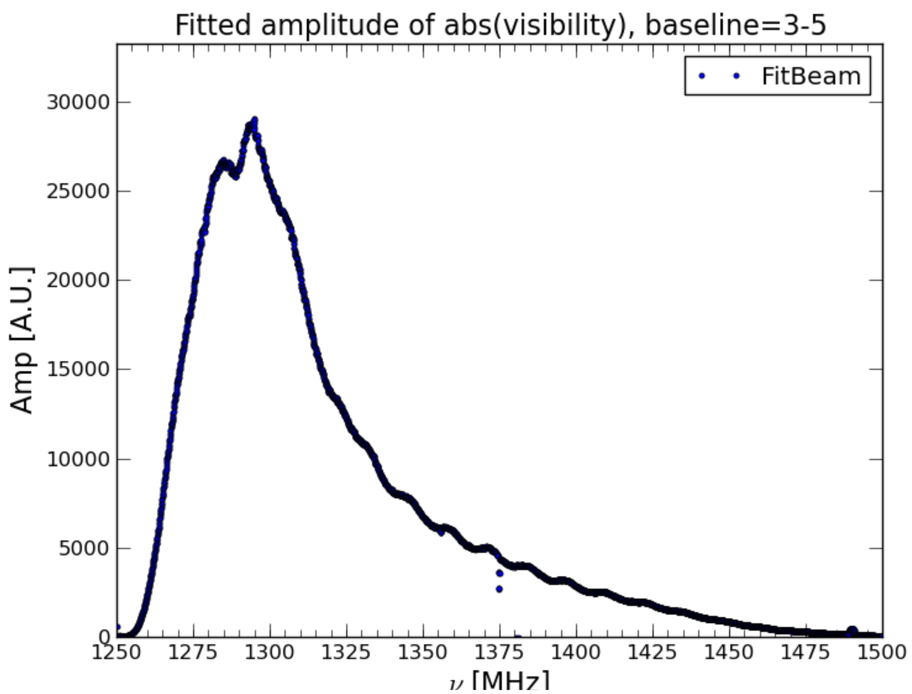


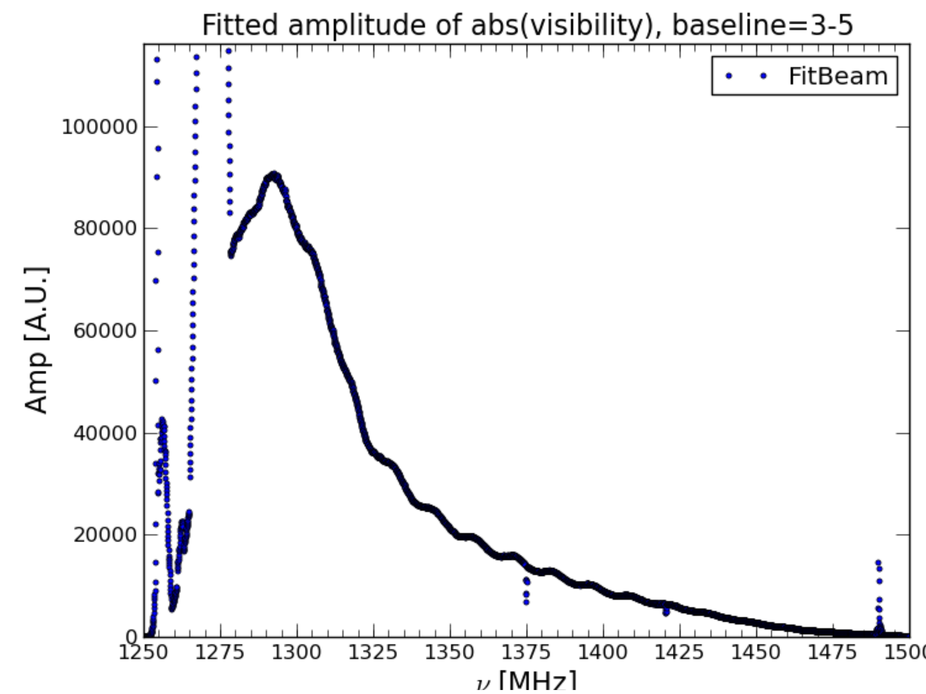
Slide Qizhi Huang analysis
Nov 2016

CygA6sep 16

CasA29sep16

g(nu)

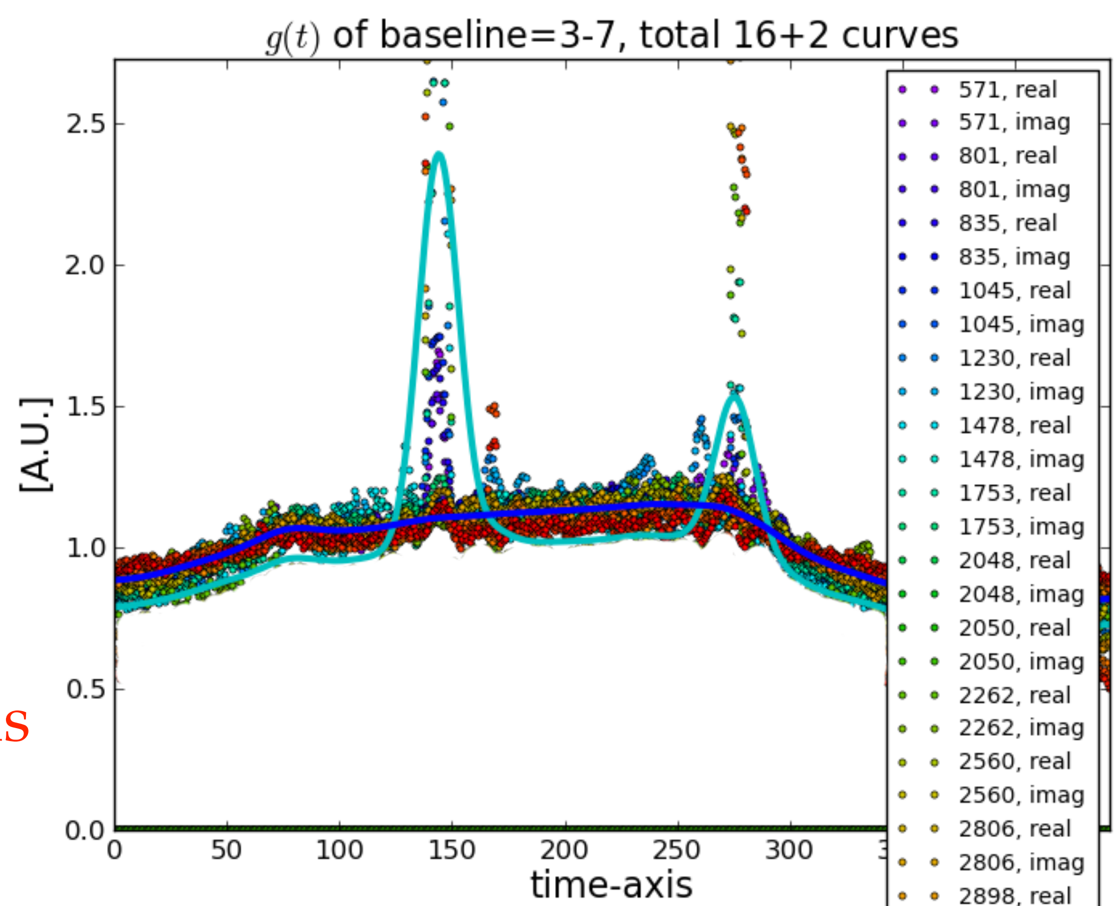




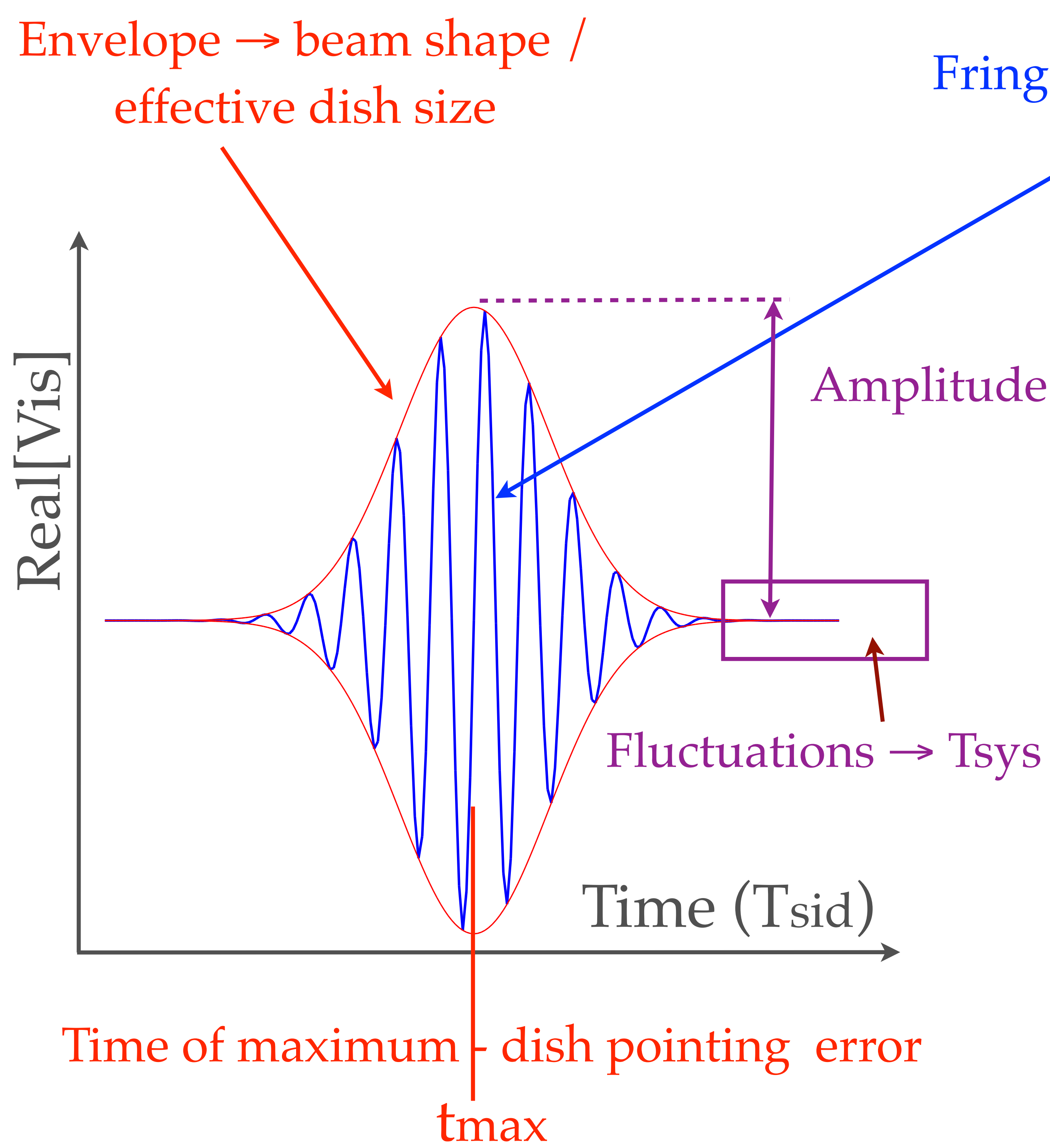
G(t)

CygA6sep 16

G(t)

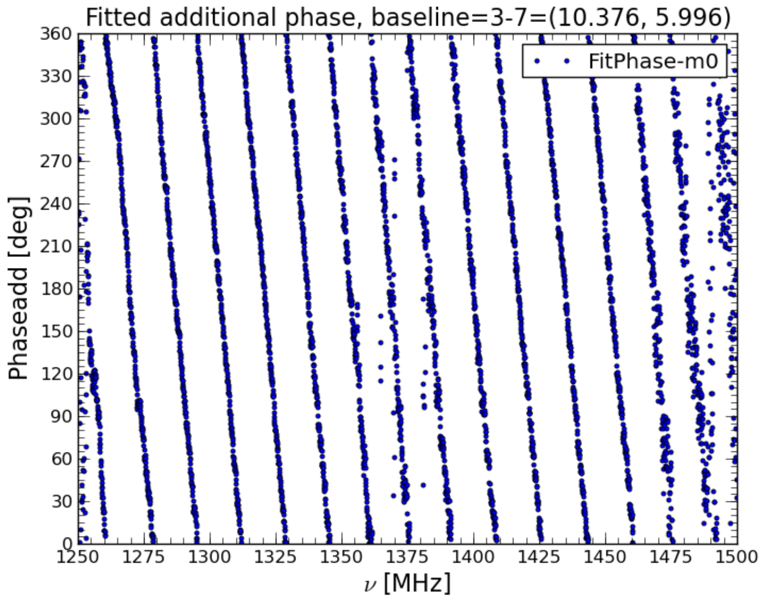


Slide Qizhi Huang analysis
Nov 2016

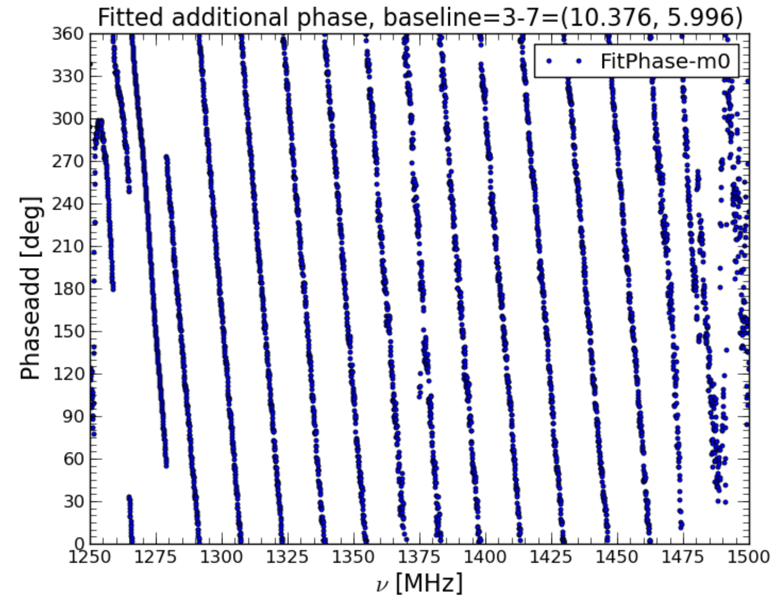


- Determine g(ν) using auto-correlation + filetring
- Determine phase difference using fringes
- Determine / check beam and array geometry using the fringes
- Determine gain (fringe amplitude) & Tsys (fluctuations before / after transit

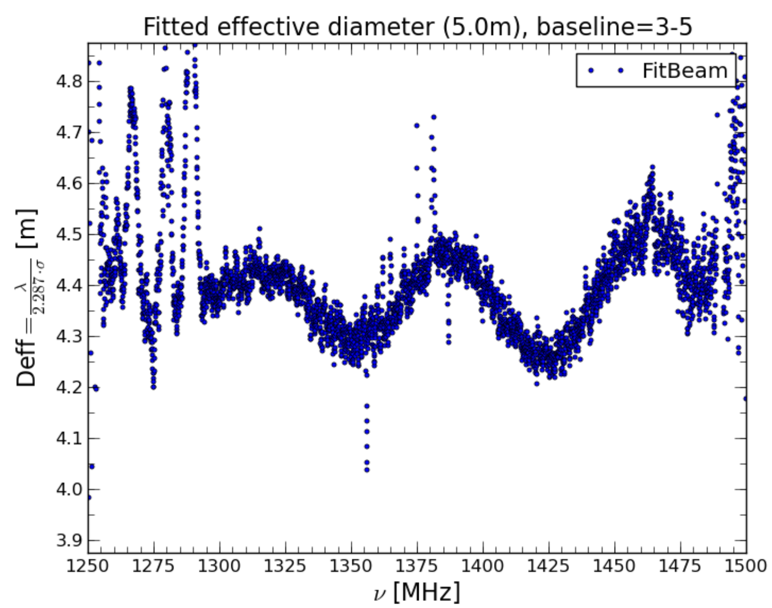
CygA6sep 16



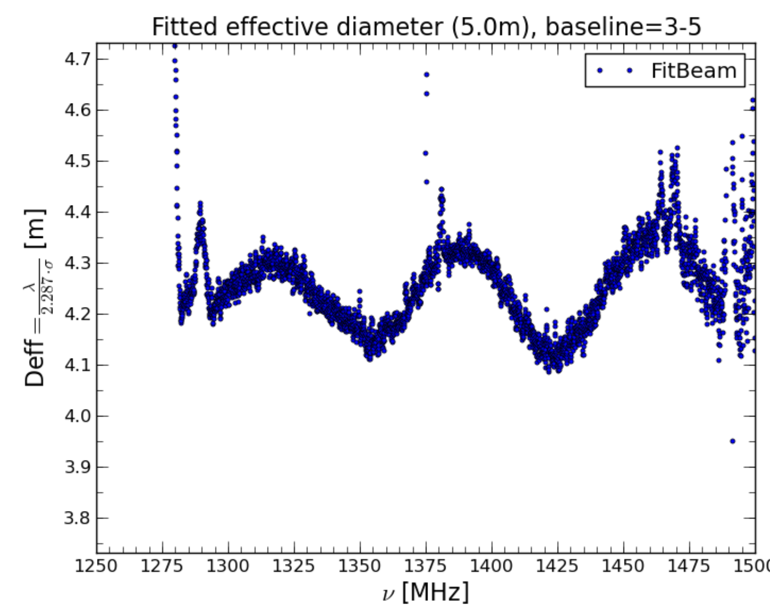
CasA29sep16



CygA6sep 16



CasA29sep16



Slide Qizhi Huang analysis
Nov 2016

Le code

gitlab.in2p3.fr

@baoradio

Software, document, tools for the BAORadio project (see http://bao.lal.in2p3.fr and http://groups.lal.in2p3.fr/bao21cm/)

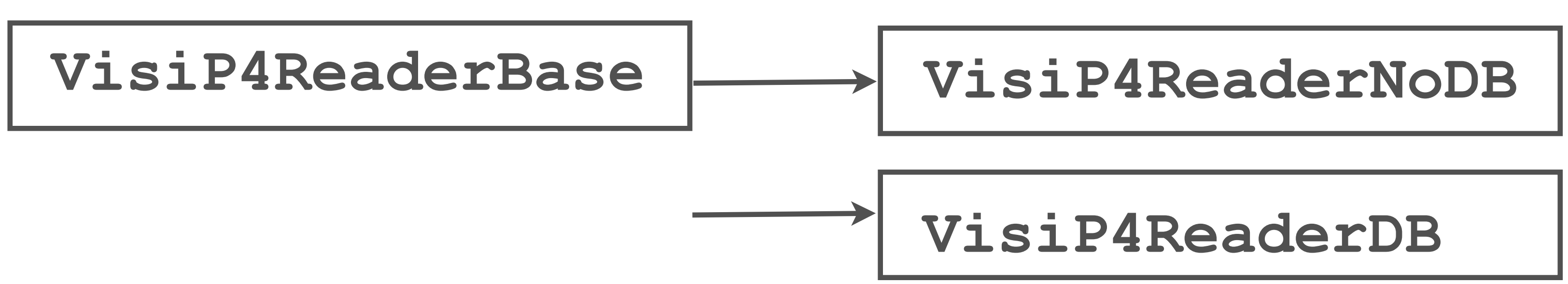
Global

All Projects

Filter by name | Last updated | New Project

- C **ControlAcq**
  usefull shell and piapp (.pic) scripts for BAORadio (FAN, HICluster, OptX21) and PAON data acquisition and on-line monitoring
- D **Documentation**
  Documentations (manuals, howto, help) for BAORadio (FAN,HICluster, OptX21) and PAON acquisition and data analysis.
- T **TAcq**
  Acquisition and processing programs for BAORadio (CRT, HICluster, OptX21, PAON)
- A **AnaPAON4**
  PAON4 visibility analysis software
- L **LHCb-find-datas**
- T **TAcqBoardCtrl**
- T **TAcqWebServer**
- A **AlteraDriver**
- **UtilPrgs**

- TAcq / : Software d’acquisition et programmes d’analyse NRT / Amas
- AnaPAON4 : programmes d’analyse
  - Classes VisiP4Reader , classes utilitaires
  - Calcul gain g(nu)
  - petit programmes utilitaires
- JSkyMap



JSkyMap : Map making software (II)
(J. Zhang PhD)

- Do not yet handle polarisation, but extension is rather easy
- Except for the computation of polarised beam responses…
- The code is rather simple, built around few classes, but relies on the SOPHYA class library (http://www.sophya.org)
- Main classes used in JSkyMap :
  - BeamTP , BeamLM and BeamVis
  - SphCoordTrans , PseudoInverse<T>
  - JSphSkyMap
  - JSkyMap and BeamUV for planar geometry
  - Some utility functions

GIT repo: https://gitlab.in2p3.fr/SCosmoTools/JSkyMap

JSphSkyMap class

```
JSphSkyMap(int lmax=512, int m=256);
```

Computing visibility array from an input map

```
/* ==== compute visibilities in spherical geometry
    -Input :  Input sky map
              BeamLM list
              lmax (optional, if zero, use the value already in the class)
    -Output : The visibility matrix, with X-index corresponding to m-modes (0<=m<=SizeX())
              and Y-index corresponding to the beams ( SizeY() = 2*beams.size() )
              the factor two comes from the fact that visibilities for positive and negative m-modes
              are written as two separate rows of the array  */
TArray< complex<double> > ComputeVisibilities(
       SphereHEALPix<double> const& inmap,
       vector< BeamLM > const& beams, int lmax=0);
```

Computing map from a visibility array

```
/* ==== reconstructing map from a single set of visibilities
  -Input :
     visarr : one array of m-mode visibilities
     beams: list of beams
     wnoisecov == true , use the noise covariance matrix when inverting the A matrix to extract alm
     compcovar == true : Compute the error covariance matrix on estimated stky alm and save it to the PPF file
     compainva == true ; save A, Ainv Ainv * A matrices
     nthreads : nb of computing threads
   -Output :
     the returned JSphSkyMap object  */
JSphSkyMap ReconstructFromVisibilityArray(
     TArray<complex<double> >  & visarr,
     vector< BeamLM > const& beams,
     bool wnoisecov, bool compcovar, bool compainva, int nthreads=1);
```

JSkyMap : some utiliy programs

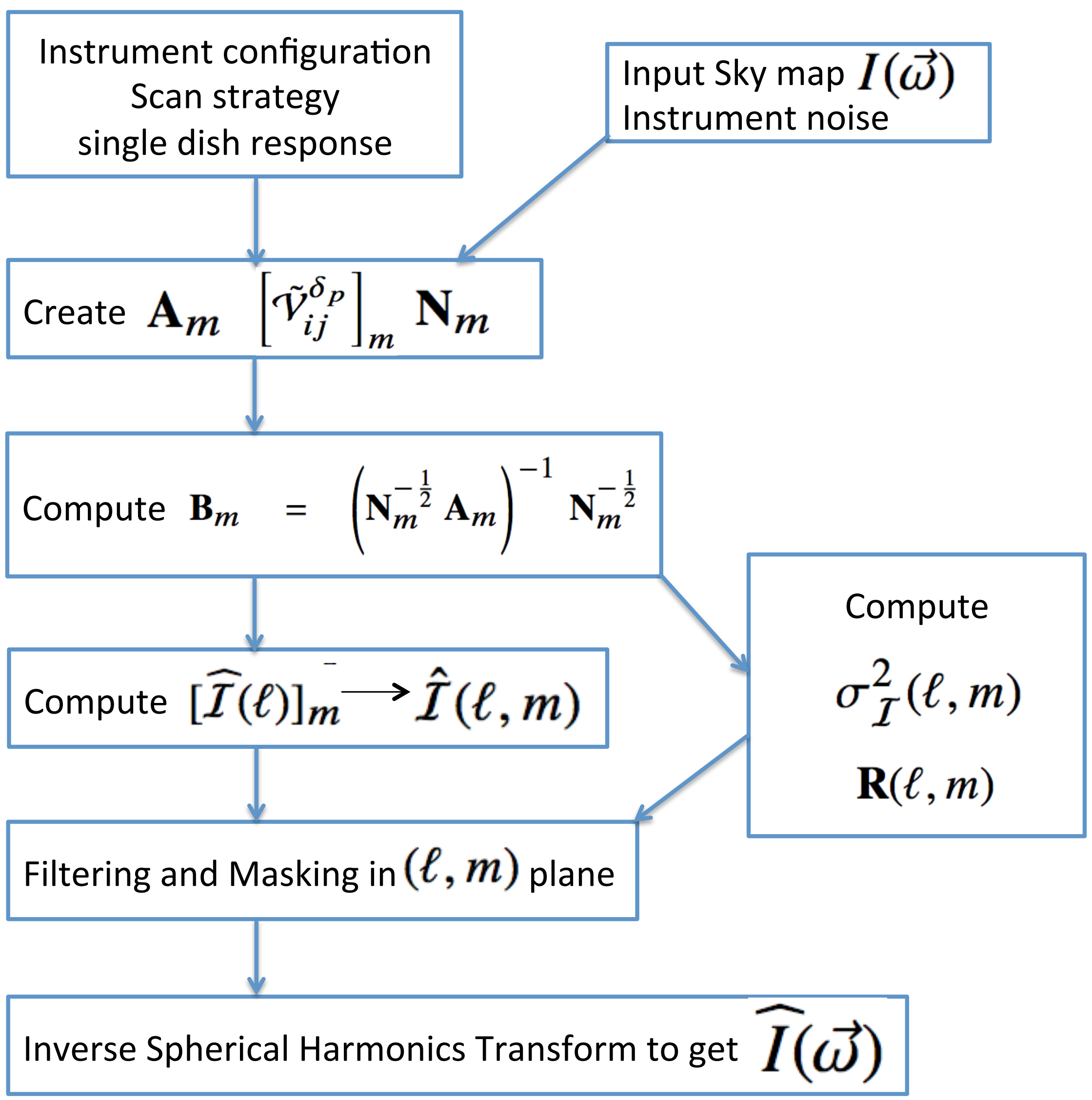
Computing visibility array from an input map (map2vis.cc)

```
 ------ map2vis.cc : Computing Visibility array from an input map and a set of
baselines -------
 map2vis/usage: map2vis InMapPPF_File OutPPF_File elevation baseline1  [baseline2
baseline3 ...]
  o elevation : elevation angle in degree (offset with respect to zenith in NS plane,
+ toward N
  o baselineS : baselineX,baselineY,baselineZ
```

Computing map from a visibility array (vis2map.cc)

```
------ vis2map.cc : reconstructing map from a set of visibility arrays -------
vis2map/usage: vis2map OutputMapPPF elevations VisiPPF1 [ VisiPPF2 ... ]
 o elevations : comma separated elevation angle values in degree
       (NS plane, + toward N)
 o VisiPPFS : Input visibility arrays
```

JSkyMap : Map making software (III)



Non polarised

$$\tilde{\mathcal{V}}_{ij}(m) = \sum_{\ell=|m|}^{+\ell_{\max}} (-1)^m \mathcal{I}(\ell,m)\mathcal{L}_{ij}(\ell,-m)$$

$$\tilde{\mathcal{V}}^*_{ij}(-m) = \sum_{\ell=|m|}^{+\ell_{\max}} \mathcal{I}(\ell,m)\mathcal{L}^*_{ij}(\ell,m)$$

$$\left[\tilde{\mathcal{V}}\right]_m = \mathbf{L}_m \times [\mathcal{I}(\ell)]_m + [\tilde{n}]_m$$

Polarised

$$\tilde{\mathcal{V}}_{p_ip_j}(m) = \sum_{\ell=|m|}^{+\ell_{max}} \sum_{\mathcal{X}} (-1)^m \mathcal{L}^X_{p_ip_j;l,-m}\mathcal{X}_{\ell m}$$

$$\tilde{\mathcal{V}}^*_{p_ip_j}(-m) = \sum_{\ell=|m|}^{+\ell_{max}} \sum_{\mathcal{X}} \mathcal{L}^{X*}_{p_ip_j;l,m}\mathcal{X}_{\ell m}$$

$$\mathcal{X} = \mathcal{I},\mathcal{E},\mathcal{B},\mathcal{V}.$$

$$\mathcal{V}_{p_i,p_j} = \left[\mathcal{V}^{xx}_{ij};\mathcal{V}^{yy}_{ij};\mathcal{V}^{xy}_{ij};\mathcal{V}^{yx}_{ij}\right]$$

$$p_i = \{(i,x),(i,y)\} \qquad p_j = \{(j,x),(j,y)\}$$

JSkyMap : Visibility array organisation