
ASTRON
Netherlands Institute for Radio Astronomy

LOFAR
software developments

Jasper Annyas

Head of Software Development and Operational Support
Radio Observatory, ASTRON

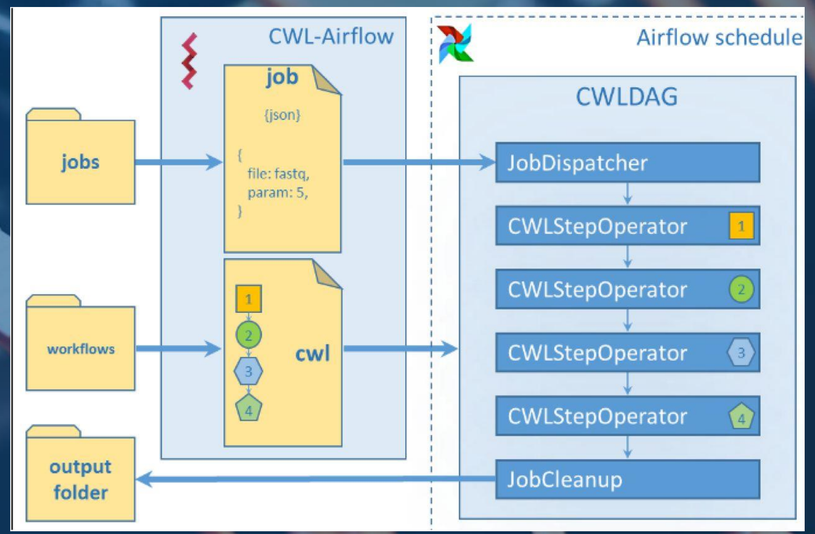
06-11-2019

Achievements: improve operations

- LOFAR software version 4.0 (Python 3, C++11, CasaCore 3)
- LOFAR software to Git
- Operational version of Station Monitor
- Installed new COBALT 2 cluster
- Automated server installation and maintenance with Cobbler and Ansible

Current activities

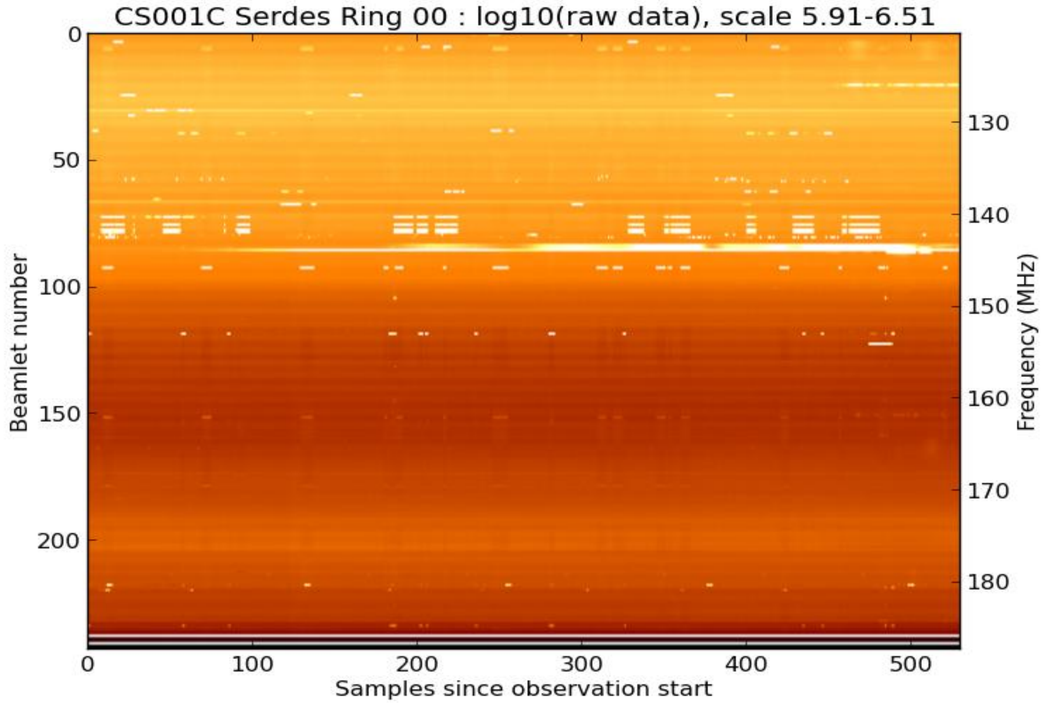
- ADDER (Data quality): New inspection plots and AI
- Holography
- LOFAR 2: Station design
- EOSC: Try out PreFactor in CWL
- SDF: Test PreFactor on CEP4



LOFAR visibility data quality

Current situation: station BST-plots

- https://proxy.lofar.eu/inspect/HTML/index.html
- For each station a dynamic spectrum is plotted showing visibility signal strength
- For 52 stations,
  that's 'only' 52 plots...

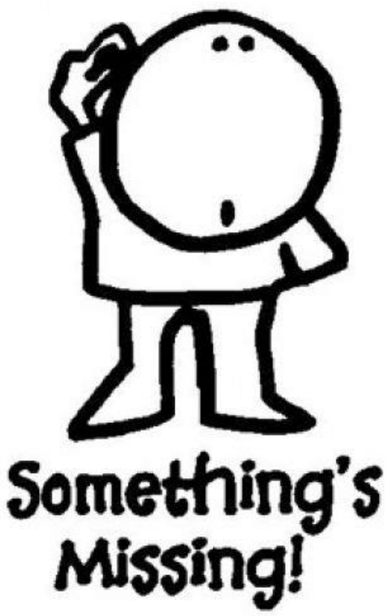


ASTRON
Netherlands Institute for Radio Astronomy

LOFAR visibility data quality

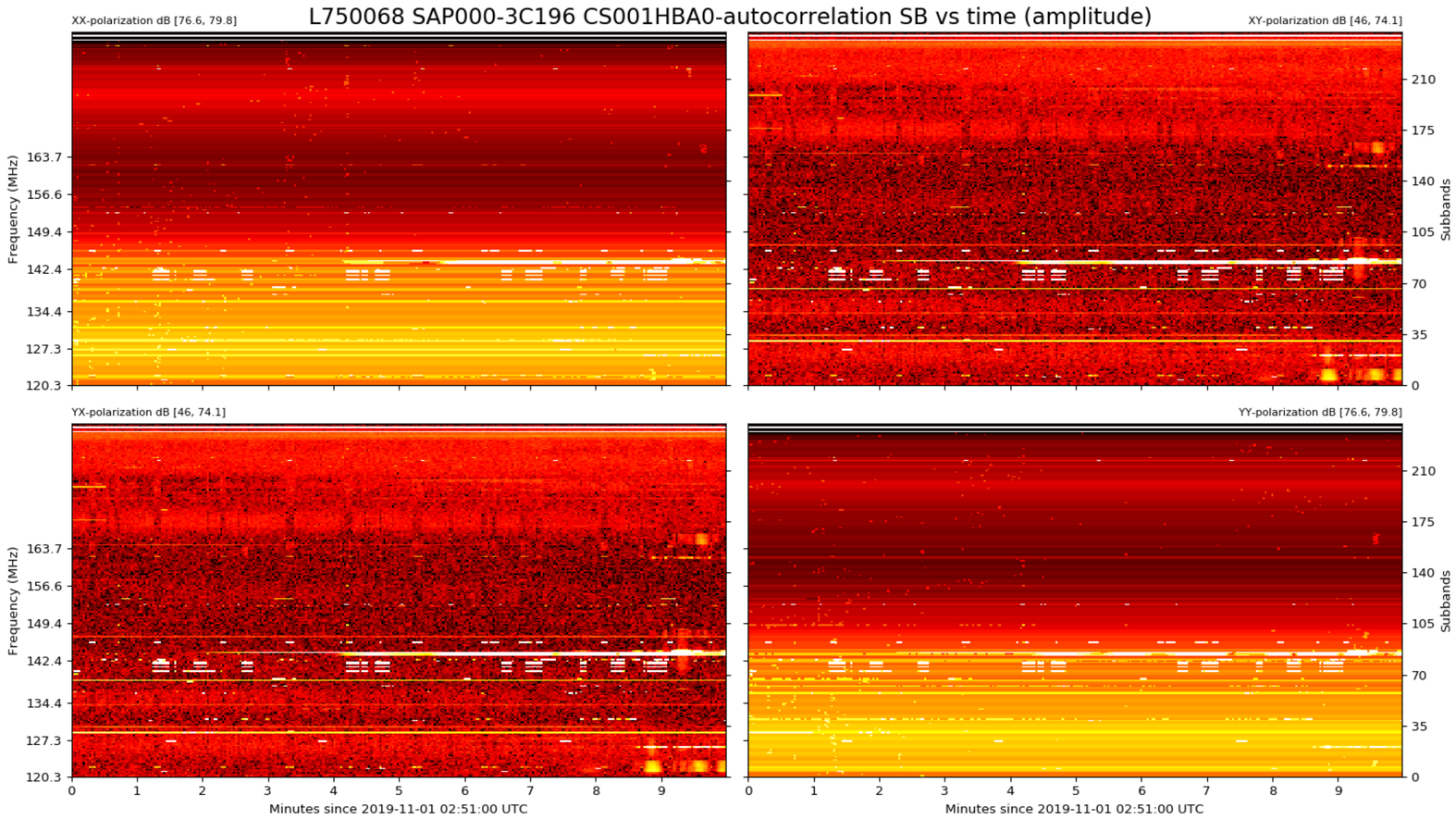
The current BST plots are missing:

- **Polarization (XX, XY, YX, YY)**
- **Phase of complex visibility values.**
- **Cross-correlations between stations (baselines)**
  → For 52 stations that results in 1326 baselines or extra plots to inspect.



ADDER: prototype data inspection

New auto-correlation plots

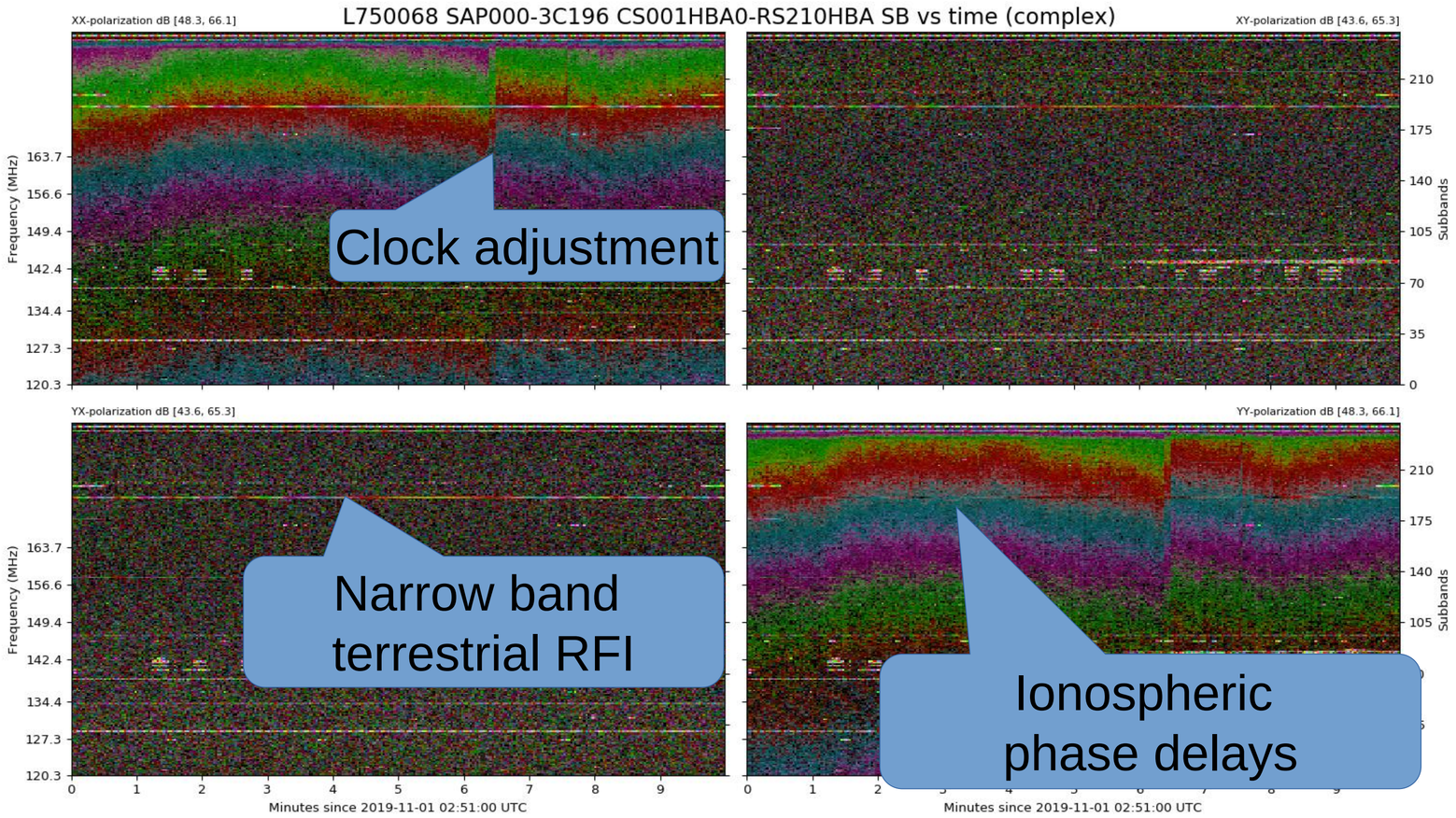


- All polarizations
- Better scaling of colormap
- Better annotation

ASTRON
Netherlands Institute for Radio Astronomy

ADDER: prototype data inspection

New cross-correlation plots



All polarizations
Complex colormap:

- color: phase
- intensity: amplitude

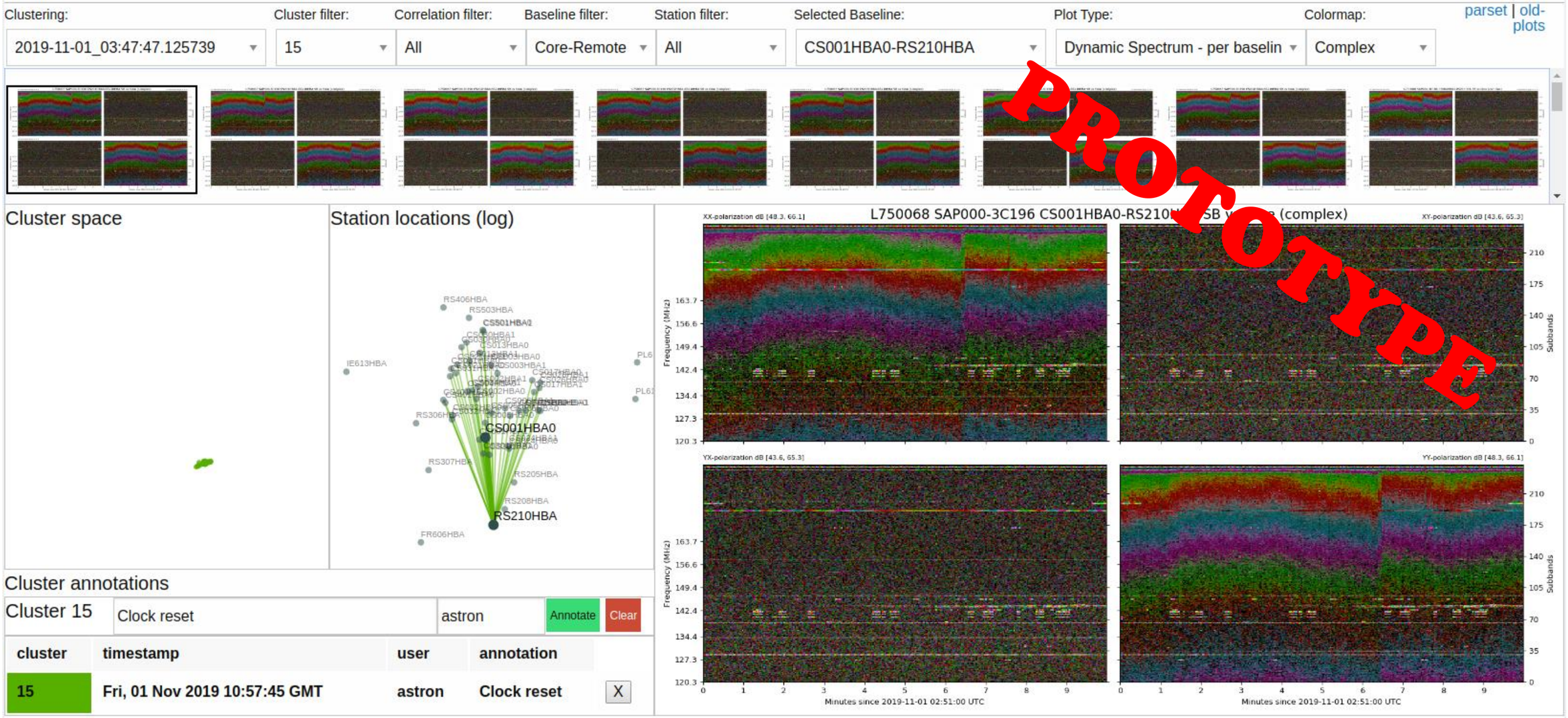
ASTRON
Netherlands Institute for Radio Astronomy

ADDER: Challenges

- **Too many plots to inspect visually**
  - N auto-correlations
  - N*(N-1)/2 cross-correlations
- **Identify and label issues (RFI, clock offset, Ionosphere, etc)**
  - Use Machine Learning to reduce ambiguity and manual work

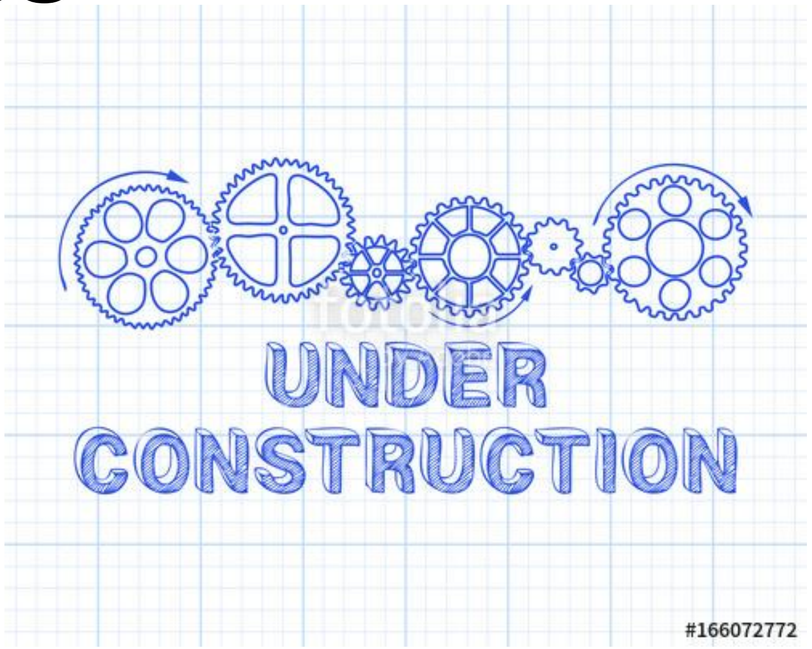
ASTRON
Netherlands Institute for Radio Astronomy

ADDER: Interactive UI to inspect & annotate



ASTRON
Netherlands Institute for Radio Astronomy

ADDER: Work in progress



- **Prototype available at:**
  - http://head01.cep4.control.lofar:8521/
  (Only inside astron/lofar domain)
- **Developers:**
  - Jorrit Schaap > ASTRON
  - Misha Mesarcik> NL eScience Center

StationMonitor: First part of MMIS

- **Stationtest**
  - Tests a specific station
  - Probe all the components of a station but only *on a specific time*
  - Scheduled weekly
- **RTSM**
  - Test a specific station during an observation
  - Probe all the components of a station used *during an observation,* therefore they cover a longer time span compared to the Stationtest
- **WinCC-OA**
  - Tracks the current status of the components (disabled, operational, beyond repair etc.)
  - Facilitates the decision what component are to be used during an observation

StationMonitor:

From a bird's eye to the ant's perspective

All LOFAR stations

Single station

Single Tile/Antenna

Available at:
https://proxy.lofar.eu/lofmonitor/

ASTRON
Netherlands Institute for Radio Astronomy

StationMonitor: All LOFAR stations

LOFAR Station monitor *v0.1*

Dashboard | Station Overview | Tiles

Station group: All stations | Errors only | Period: 1 wk | 2 wk | 3 wk | 4 wk | Error type: All

Station overview

Name	Station tests	Latest observations
CS001C	20 17 13 9	708614 1 | 708610 1 | 708914 3 | 708454 1 | 707454 1 | 708908 5
CS002C	20 15 44 53	708618 1 | 708610 2 | 708606 4 | 708602 3 | 708590 3 | 708582 1
CS003C	13 38 42 10	708622 1 | 708618 1 | 708610 1 | 708606 3 | 708602 3 | 708598 1
CS004C	10 10 13 13	708618 2 | 708610 2 | 708606 2 | 708602 3 | 708598 2 | 708590 3
CS005C	15 30 19 16	708914 2 | 708912 1 | 707454 2 | 708908 4 | 708430 1 | 708406 1
CS006C	20 45 19 22	708912 4 | 707454 1 | 708908 1 | 708350 1 | 708662 4 | 708658 1
CS007C	13 11 17 13	708912 4 | 708928 1 | 708406 1 | 708662 5 | 708108 3 | 708104 1
CS011C	8 7 6 7	708626 1 | 708618 1 | 708610 1 | 708606 1 | 708602 2 | 708598 1
CS013C	12 14 12 10	708912 7 | 708662 6 | 708108 1 | 708066 1 | 708866 1 | 707190 1
CS017C	12 9 11 19	708594 1 | 708912 4 | 707454 1 | 708928 1 | 708926 1 | 708422 1
CS021C	29 85 30 23	708626 1 | 708622 1 | 708594 1 | 708566 1 | 708546 2 | 708534 1
CS024C	8 10 32 25	708618 2 | 708614 1 | 708610 1 | 708606 1 | 708602 1 | 708598 1
CS026C	15 8 10 8	708614 1 | 708602 1 | 708566 2 | 708914 2 | 708912 4 | 707454 1

Latest observations

Observation	Start date	Stations	Stations with errors	Total errors
708626	2019-05-14 23:19	4	4	14
708622	2019-05-14 23:0			15
708618	2019-05-14 20:5			33
708614	2019-05-14 20:4			22
708610	2019-05-14 20:3			31
708606	2019-05-14 18:4			42
708602	2019-05-14 18:1			43
708598	2019-05-14 18:0			27
708594	2019-05-14 17:5			21
708590	2019-05-14 17:3			47
708586	2019-05-14 17:22	8	8	32
708582	2019-05-14 17:11	8	8	37
708578	2019-05-14 17:00	10	10	36

708626
Start: 2019-05-14 23:19
End: 2019-05-14 23:29
Mode: 5

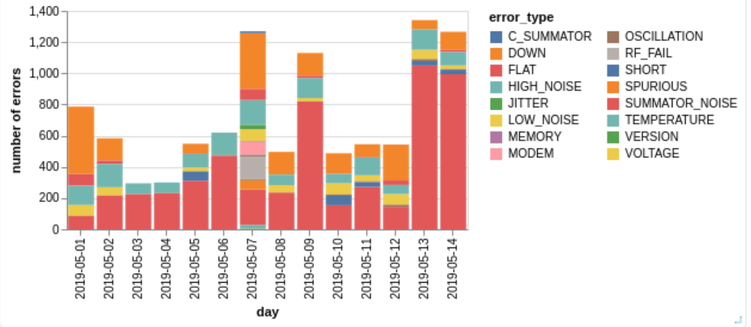
Station name	errors
CS021C	1
CS302C	6
CS011C	1

Station test summary

Date	Time	Station	Comp.	VS	SM	TE	SH	DW	CS	JI	VO	MY	HN	LN	MO	OS	FL	SP	RF
2019-05-07	21:00	CS001C	HBA		1				2	1			1		1			2	2
			LBH				1	3						1			1		
			LBL					2									1		
			RSP			1													
	21:00	CS002C	HBA							2			1		1				6
			LBH					1					1					1	1
			LBL					3						1					
			RSP			2													
	21:00	CS003C	HBA						1				4		1				3

Station statistics

Both test types | day | per error type



StationMonitor: Single station

LOFAR Station monitor *v0.1*

Dashboard | Station Overview | Tiles

CS002C | Type: BOTH ST-TEST RTSM | Period: 1 wk 2 wk 4 wk | 2019-May-01 | 2019-May-15 | Error type: All

HBA LBH LBL RSP

	0	3	4	6	10	13	14	16	18	19	24	26	30	31	32	33	34	35	36	37	38	39	40	41	42	43	45	46	47
RT 38 obs 6d2h			11%	45%			3%		27%				8%	3%			3%				3%		3%	45%					
RT 2019-05-14 22:59													1																
RT 2019-05-14 22:31				1					1																				
RT 2019-05-14 20:40				1					1				1											1					
RT 2019-05-14 20:19				1					1															1					
RT 2019-05-14 19:36				1					1															1					
RT 2019-05-14 19:11																								1					
RT 2019-05-14 19:00				1					1																				
RT 2019-05-14 18:27													1																
RT 2019-05-14 18:16														1															
RT 2019-05-14 16:31																								1					
RT 2019-05-14 08:19																								1					
RT 2019-05-13 10:51																								1					
RT 2019-05-13 10:40				1																				1					
RT 2019-05-13 10:29																								1					
RT 2019-05-13 09:58			1																										
RT 2019-05-13 09:36				1																				1					
RT 2019-05-13 07:52																								1					
RT 2019-05-13 06:57																								1					
RT 2019-05-13 06:46																								1					
RT 2019-05-13 02:51									1																				
RT 2019-05-13 01:00									1																				
RT 2019-05-11 21:03				1																									
RT 2019-05-11 04:57				1					1								1												
RT 2019-05-10 22:26			1																										
RT 2019-05-10 12:59									1																				
RT 2019-05-10 12:52																					1								
RT 2019-05-09 20:21			1	1																									
RT 2019-05-09 20:10							1																						
RT 2019-05-09 15:13				1																									
RT 2019-05-09 13:53				1																									

HBA, Antenna 18, Test 2019-05-14 22:31

RCU 36 (X)

SUMMATOR_NOISE

frequency-range: 100-200 MHz

percentage: 66.67%

count: 6

component_id: 18

StationMonitor: Future plans

Plans for additional functionality:

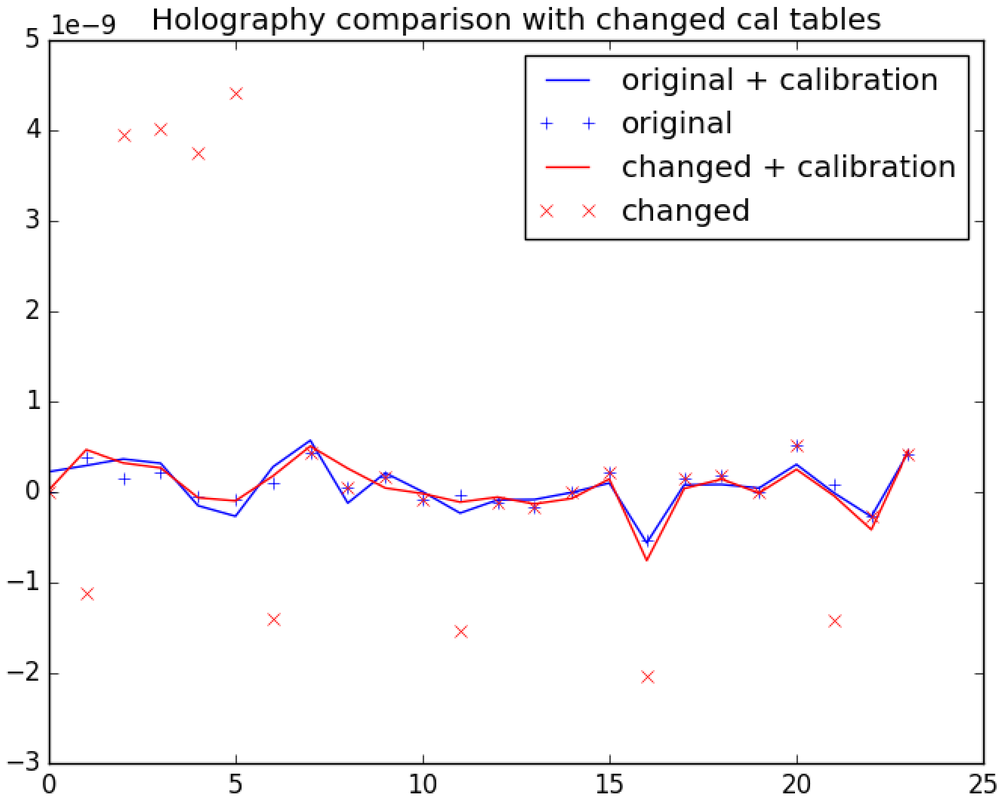
- Integration of the maintenance log to display the repairs on a specific component
- KPIs (Key Performance Indicators) under investigations to quantify the impact of a failure on an observation
- Overview of the failures per observation
- Long term statistics plots

Developers: Mattia Mancini and Reinoud Bokhorst
Coordinator: Ronald Roelfsema

ASTRON
Netherlands Institute for Radio Astronomy

HOLOGRAPHY

CS001HBA0 X-pol calibration



Crosses: Calibration table applied
Line: Calibration table after holography
Red: Artificial delays added for antennas 1,2,3,4,5,6,11,17,21

➢New station calibration mode
- “Image your antennas”
- Determine gains
- System monitoring (e.g. RSP clock distribution offsets in DE609)

➢Status HBA:
- Software mostly finished
- Can now produce corrected calibration tables
- Working on inspection plots
- Commissioning HBA Q4 2019 (ongoing)

➢Status LBA:
- Changes to station beamformer required
- Changes required to software to be investigated
- Work and commissioning to be planned

➢Developped by Sander ter Veen
and Mattia Mancini

Upcoming other items

Under development by our system administrators

- CentOS upgrade continues after the LCUs
- Cluster management: Kubernetes
- Security

And continue again with:

- LOFAR Efficiency Improvement; the new MoM and more
- Projects continued

STATION CONTROL AND COMMANDS IN LOCAL MODE

AUKE L. KLAZEMA

KLAZEMA@ASTRON.NL

ASTRON
Netherlands Institute for Radio Astronomy

LOFAR

ILT-TO F2F 2018 MEETING - OLSZTYN

OVERVIEW STATION

SPECS

LOW BAND

- LBA (LBH)
- 10 Mhz - 90 Mhz

HIGH BAND

- HBA antennas
- 110 Mhz - 240 Mhz

SAMPELING

- 160 Mhz
- 200 Mhz

HARDWARE FULL STATION

- 6 * 32 RCU
- 6 * 4 RSP
- 6 * 2 TBB
- 6 * 1 TDS
- 6 * 1 JTAG
- 6 * 1 SPU
- 1 LCU

HARDWARE ILT VS NL

- ILT 96 LBA
- NL 48 LBA inner + 48 LBA outer
- ILT 96 HBA
- Core 24 HBA0 and 24 HBA1
- Remote 48 HBA

SWLEVEL

1. Monitoring
2. RSP + TBB driver
3. Beam + Cal server

MAC_REVERT

list older installes versions

```
MAC_revert -s
```

activate specific version

```
MAC_revert -v <version name>
```

activate previous version

```
MAC_revert -p -i
```

CONTROL PROGRAMS

BEAMCTL

beamctl will setup a beam and manage tracking.

A maximum of 244/488 multiple beams can be created

BEAMCTL USE CASES

- LBA beams
- HBA beams
- Calibration information

LBA BEAM

```
beamctl --antennaset=LBA_INNER --rcus=0:191 --band=10_90 \
        --subbands=200:443 --beamlets=0:243 \
        --digdir=6.1234876806221052,1.026515399560464 8,J2000 &
```

band options: 10_90 10_70 30_70 30_90

HBA BEAM

```
beamctl --antennaset=HBA_DUAL --rcus=0:191 --band=110_190 \
        --subbands=200:443 --beamlets=0:243 \
        --anadir=6.1234876806221052,1.0265153995604648,J2000 \
        --digdir=6.1234876806221052,1.0265153995604648,J2000 &
```

HBA BEAM

```
beamctl --antennaset=HBA_DUAL --rcus=0:191 --band=110_190 \
        --subbands=200:443 --beamlets=0:243 \
        --digdir=6.1234876806221052,1.0265153995604648,J2000 &
```

HBA BEAM

```
beamctl --antennaset=HBA_DUAL --rcus=0:191 --band=110_190 \
        --anadir=6.1234876806221052,1.026515399560464 8,J2000 &
```

band options: 110_190 170_230 210_250

MULTIPLE BEAMCTL

list all running beamctl with it arguments and pid

```
pgrep -a beamctl
```

kill specific pid

```
kill <pid>
```

kill all beamctl

```
killall beamctl
```

DIRECTION FORMAT

```
digdir=lon,lat,dirtype,duration
```

```
anadir=lon,lat,dirtype,duration
```

```
beamctl --remotehost
```

RSPCTL

RCU CONTROL SETTINGS

show current RCU control settings

```
rspctl --rcu --select=0:95
```

set the RCU control settings (see usage for details)

```
rspctl --rcu=0x00000000 --select=0:95
```

RCU CONTROL SETTINGS HELPERS PART 1

set rcumode

```
rspctl --rcumode=0 --select=0:95
```

0 = off, 1 = LBL 10Mhz HPF, 2 = LBL 30Mhz HPF, 3 = LBH 10Mhz HPF, 4 = LBH 30Mhz HPF, 5 = HBA 110-190Mhz, 6 = HBA 170-230MHz, 7 = HBA 210-270Mhz

turn on/off pseudo random sequence generator

```
rspctl --rcuprsg=1 --select=0:95
```

enable or disable reset on RCU

```
rspctl --rcureset=1 --select=0:95
```

RCU CONTROL SETTINGS HELPERS PART 2

set RCU attenuation

```
rspctl --rcuattenuation=10 --select=0:95
```

set RCU delay

```
rspctl --rcudelay=99 --select=0:95
```

enable or disable RCU

```
rspctl --rcuenable=1 --select=0:95
```

RCU CONTROL SETTINGS HELPERS PART 3

enable or disable spectral inversion

```
rspctl --specinv=1 --select=0:95
```

set RCU mode like rcmode but it does more. Clock switch, inversion, enableling rcu, HBA delay

```
rspctl --mode=3 --select=0:95
```

0 = off, 1 = LBL 10Mhz HPF, 2 = LBL 30Mhz HPF, 3 = LBH 10Mhz HPF, 4 = LBH 30Mhz HPF, 5 = HBA 110-190Mhz, 6 = HBA 170-230MHz, 7 = HBA 210-270Mhz

SIGNAL PROCESSING PART 1

get weights as complex values

```
rspctl --weights --select=0:95
```

set weights as complex values

```
rspctl --weights=value,value.im --select=0:95 --beamlets=1
```

get weights as amplitude and angle

```
rspctl --aweights
```

set weights as amplitude and angle

```
rspctl --aweights=amplitude,angle --select=0:95 --beamlets=1
```

SIGNAL PROCESSING PART 2

get subbands

```
rspctl --subbands --select=0:95
```

set subbands

```
rspctl --subbands=0:39 --select=0:95
```

get subband for cross correlation

```
rspctl --xcsubband
```

set subband for cross correlation

```
rspctl --xcsubband=1
```

SIGNAL PROCESSING PART 3

get waveform generator settings

```
rspctl --wg --select=0:95
```

set waveform generator settings

```
rspctl --wg=freq --phase=phase --amplitude=amplitude \
       --select=0:95
```

STATUS INFO PART 1

get RSP board status

```
rspctl --status --select=0:23
```

get TDS board status

```
rspctl --tdstatus --select=0:23
```

get SPU board status

```
rspctl --spustatu
```

get RSP board status

```
rspctl --tdstatus --select=0:23
```

STATUS INFO PART 2

get version

```
rspctl --version --select=0:23
```

get HBA real delays

```
rspctl --realdelays --select=0:191
```

get status of all registers (updated 1s)

```
rspctl --regstate
```

get latency of ring and lanes

```
rspctl --latency
```

STATISTICS

subband statistics

```
rspctl --statistics=subband --select=0:95 --duration=sec \
       --integration=sec --directory=/localhome/user0
```

beamlet statistics

```
rspctl --statistics=beamlet --select=0:95 --duration=sec \
       --integration=sec --directory=/localhome/user0
```

cross correlation statistics

```
rspctl --xcstatistics --select=0,1 --duration=sec \
       --integration=sec --directory=/localhome/user0 \
       --xcangle
```

MISCELLANEOUS PART 1

set clock

```
rspctl --clock=200
```

clear FPGA registers on RSPboard

```
rspctl --rspclear --select=0:23
```

set HBA delays

```
rspctl --hbadelays=184,188,192,196,184,188,192,196,184,188,192
```

set Serdes splitter

```
rspctl --splitter=1
```

MISCELLANEOUS PART 2

set transient tbbmode

```
rspctl --tbbmode=transient
```

set subband tbbmode

```
rspctl --tbbmode=subbands,0:39
```

swap x and y

```
rspctl --swapxy=1 --select=0:191
```

set bitmode

```
rspctl --bitmode=8
```

bitmode options: 4 8 16

MISCELLANEOUS PART 2

set datastream to CEP

```
rspctl --datastream=0
```

SUBBAND DATA OUTPUT

enable or disable SDO

```
rspctl --sdoenable=1
```

set number bits per sample

```
rspctl --sdomode=4
```

options: 3,5,8,16

set sdo selection

```
rspctl --sdo= --select=0:191
```

TBBCTL

information on size of memory on selected boards

```
tbbctl --size --select=0:5
```

show rcu info for NON free rcu's

```
tbbctl --rcuinfo --select=0:95
```

get version information from selected boards

```
tbbctl --version --select=0:5
```

information on size of memory on selected boards

```
tbbctl --status --select=0:5
```

TBB RECORD MODES

record subbands

```
tbbctl --mode=subbands --select=0:243
```

record raw A/D converter data

```
rspctl --mode=transient
```

TBB ALLOCATION

allocate space for selected RCU's

```
tbbctl --alloc --select=0:95
```

free space for selected RCU's

```
tbbctl --free --select=0:95
```

TBB RECORDING

start writing data into the ring buffer

```
tbbctl --record --select=0:95
```

stop writing data into the ring buffer

```
tbbctl --stop --select=0:95
```

TBB TRIGGER SETUP

```
tbbctl --setup=level,start,stop,filter,window,mode --select=0:
```

- level : standard deviation trigger level
- start : number of samples above level needed
- stop : number of samples belown level needed
- filter: if filter is on or of (0/1)
- window: integration window length (0,8)
- mode : one_shot of continues

set coefficients of IIR filter

```
tbbctl --coef=f00,f01,f02,f03,f10,f11,f12,f13 --select=0:95
```

release the trigger

```
tbbctl --release --select=0:95
```

TBB TRIGGER LISTENING

listen to next trigger

```
tbbctl --listen=one_shot
```

listen to all incomming triggers

```
tbbctrl --listen=continues
```

TBB READING TO LOCAL FILE

stop recording before reading

```
tbbctl --stop
```

read data to local file but it is slow

```
tbbctl --readpage=rcunr,startpage,npages
```

TBB READING TO CEP

set CEP storage node

```
tbbctl --storage=node --select=0:95
```

stop recording before reading

```
tbbctl --stop
```

read data to CEP

```
tbbctl --read=rcunr,centertime,timebefore,timeafter
```

the next command stops, reads and restarts recording

```
tbbctl --readall=pages --select=0:95
```

MODE-357

```
rspctl --mode=3 --sel=0:31
sleep 2
rspctl --mode=5 --sel=32:63
sleep 2
rspctl --mode=7 --sel=64:95
```

TBB/RSP RAW RECORDING

One way is to dump udp packet coming from the TBB/RSP boards with a tool like wireshark or tcpdump. The jumbo udp packets contain the data that still needs to be extracted. The dumping tools might help you there.

For some TBB transient mode we already have datawriters that can extract the data to a file. For the subband mode a datawriter is in the making.