

LOFLI

LOFAR Lightning Imaging

Notes & software description

Olaf Scholten

Kapteyn Institute & KVI, University of Groningen, The Netherlands

O.Scholten@rug.nl

Source: Nts_LofarImaging; Last edited: 2021-05-25 21:55:59

Guide to the guide for V19

This guide is structured according to the chronological order of the main steps that are usually followed in converting LOFAR TBB data to an image of a lightning flash and to each a chapter is devoted. An outline of the basic imaging method can be found in [Scholten et al., 2021]. Edition v19 contains also the routines for full interferometry. At this time the paper is still being written.

In each chapter contains a description of the essential ingredients of the code, examples of input lines, examples of generated output, as well as examples of shell commands for running the code in Linux.

Table 1: Naming conventions.

MainDir	The main working folder containing the various FlashFolders, the “/Utilities”, and the “/program” folders.
FlashFolder	one for each flash, named with the tag of the flash as for example in: <code>FlashFolder=''/MainDir/18D-1'</code>
ArchiveDir	The path to the archive containing the .h5 time traces from all antennas for a flash. If these reside on a remote site it is advised to use a SSHFS link through a local directory, as for example: <code>ArchiveDir=''/home/olaf/kaptdat/lightning_data/2017/D20170929T202255.000Z'</code>
ProgramDir	The folder containing the sources and executables of the programs. <code>FlashFolder=''/MainDir/program'</code>
UtilDir	Folder containing useful scripts <code>UtilDir=''/MainDir/Utilities'</code>

Table 2: Required packages. Make sure that the script “compile.sh” (in “FlashFolder”) contains the right links to the FFTPACK library.

GLE	Principle plotting utility [Pugmire et al., 2015]. Download from: sourceforge
fftpack	Basic FFT routines [Swarztrauber, 1982] Download from: netlib; the double precision version is used.
hdf5-fortran	Allows random access to compressed data files with .h5 extension. Download from: Linux and Windows

Contents

1 Installation cookbook	4
2 Introduction	4
3 Setup & RFI mitigation	5
3.1 Additional details	5
3.2 Figures and print-out	6
4 The ‘explore’ option	7
4.1 Figures and print-out	8
5 The ‘calibrate’ option	8
5.1 First calibration stage	9
5.1.1 Discussion	10
5.2 Second calibration stage	11
5.2.1 Discussion	13
5.3 Third calibration stage	13
5.4 Figures and print-out	14
5.4.1 Curtain plot	14
5.4.2 Cross correlation plot	14
5.4.3 Print output	15
6 Impulsive imaging or source finding	16
6.1 details	18
6.2 output	18
7 Plotting flashes	18
8 Interferometric imaging	22
8.1 Input detailed	22
8.2 Output	24
8.2.1 Intensity plot	24
8.3 Selective plotting	25
8.4 Method of Time Resolved Interferometric 3D Imaging	26
8.4.1 Space & time grids and summing time traces	28
8.4.2 Antenna weighting	28
8.4.3 Antenna calibration	29

8.4.4	Determining the position of maximal intensity	30
8.4.5	Source intensity and polarization	30
9	E-field Interferometric imaging	31
9.1	Basics	31
9.1.1	Polarization dependent weights	32
9.1.2	Relative X- & Y-dipole timing calibration	34
9.1.3	The Jones matrix	34
A	Details of the LOFLI code	34
A.1	Data reading	35
A.1.1	Subroutine AntennaRead	35
A.2	Candidate pulse selection	35
A.2.1	Subroutine DualPeakFind	35
A.3	Cross correlation	36
A.3.1	Subroutine BuildCC	36
A.3.2	Subroutine GetCorrSingAnt	36
A.3.3	Subroutine CrossCorr	37
A.3.4	Subroutine ReImAtMax	37
A.4	Fitting	37
A.5	Source search	37
A.5.1	Subroutine SourceTryal	38
A.5.2	Subroutine SourceFind	38
A.5.3	Subroutine SourceFitCycle	38
A.5.4	Subroutine CleanPeak	38
A.5.5	Subroutine FindStatCall	38
B	Kalman filter	39
B.1	For sources	40
B.1.1	Notation	40
B.1.2	Linear	40
B.1.3	Realistic, non linear	41
C	Tables and such	42

September 15, 2021

1 Installation cookbook

- Unpack the zip file, keeping the directory structure. This should give something looking like Table 3.
- Install GLE from sourceforge, follow the instructions [Pugmire et al., 2015]. The shortcut “gle” should start running the gle program.
- Install hdf5-fortran see .h5 extension. Download from: Linux and Windows
- Install “FFTPACK”, most probably in the directory with the same name. Source code is included in this distribution.
- Edit the file “ShortCuts.sh” to make sure that especially “LIBRARY” points to the binary, compiled, version of the FFTPACk library. You may use an absolute path, or a relative path. Note that all relative paths should start from the “FlashFolder” sub-folder of the “MainDir”.
- Edit the files “Utilities/compile.sh” to make sure that especially “prefix” and “H5TOOL_BIN” are correct for your implementation of the h5-tools. Note that all relative paths should start from the “FlashFolder” sub-folder of the “MainDir”.
- Edit the files “Utilities/FlashID.sh” to make sure that especially “ArchiveDir” points to the data archive following your convention for the flashes. Possibly also modify “Utilities/list.ssv” to make the proper link between the archive file-naming convention of a flash and your local convention.

Table 3: Directory structure after complete installation should be like this.

MainDir	Likely named “LOFLI-v19”
Utilities	containing general utilities scripts, in particular used for making images.
program	containing source codes and executables.
copysetup	containing templates for installing the Flash-directories by running “NewFlash.sh”.
Antennafields	containing information on antenna positions and such.
FFTPACK	containing the Fast Fourier transform library.

2 Introduction

The LOFAR data files are written in HDF5 format and require the installation of HDF5 for Fortran, see the download page. There is also a Windows installation possible.

Running the LOFLI package assumes a particular sub-directory structure as well as some particular files. The script “`NewFlash.sh`” creates the correct directory structure, copies the necessary files from the “`copysetup`” directory to the required new “FlashFolder” (requires some editing of the shell code), starts to run the RFI-Mitigation code in batch-mode, and starts an exploratory imaging run, see Section 4, of the data after the RFI-mitigation has finished. When setup correctly “FlashFolder”

has two subfolders named "Book" and "files". The first will store the calibration files which are essential for imaging and the latter files needed for intermediate checking and it may be considered as a scratch directory. These directories as well as necessary script files are copied to their appropriate location by running the script "`NewFlash.sh`" in the MainDir.

Best practise is to name the FlashFolder (as set in script "`NewFlash.sh`") with the tag of the flash, say "18D-1". The translation from the shorter internal flash label used in the lightning group to the UTC-label used by LOFAR is generated by the script "`FlashID.sh`" that resides in the Utilities directory. This script reads the file "`list.ssv`" and make sure this is updated for your system, where the first entry is the short-hand notation used for the FlashFolder name and the third entry refers to the flash identifier used in the (LOFAR) archive.

In the script "`FlashID.sh`" also `ArchiveDir=/home/olaf/kaptdata/lightning_data/` needs to be set to the proper place where the raw LOFAR-data reside. Note that there are no spaces allowed around the equal signs. The ArchiveDir should point to the directory that contains the TBB data from LOFAR as .H5 files. It is very handy to make a sshfs logical connection between the remote computer containing the archive and a local folder if the archive is not on your local system already.

3 Setup & RFI mitigation

The purpose of "`RFI_Mitigation.sh`" is to pre-process the data-files for a single flash and in the process determine the frequency notch filters to mitigate RFI. The RFI filter data is written to "`Book/RFI_Filters.upt`" of the main flash-working directory 'FlashFolder'. The file names to be processed are read from the file "`directory.out`" (should be in the FlashFolder) as was created by the script "`NewFlash.sh`". The script "`NewFlash.sh`" should have automatically started the running of "`RFI_Mitigation.sh`".

The files will be processed in the order as listed in "`directory.out`" where the • **second** • station in the directory listing will be used as the reference station later in imaging (to change this replace the "2" on the line "\${ProgramDir} \${Prog} \${AntennaFieldsDir} 2" by the appropriate number in "`RFI_Mitigation.sh`"). Be careful to set here a station that is in a rather dense part of the array, preferably be it CS002.

Additional axillary information needed for the imaging runs is written to "`Book/LOFAR_H5files_Structure.dat`".

The script "`RFI_Mitigation.sh`" takes a while to run. At the end it will submit the script `Explore.sh`.

3.1 Additional details

All files in the directory listing given in "'`directory.out`'" are scanned. If 9 of the first 100 data-chunks of a specific antenna are non-zero an RFI analysis is made otherwise the antenna is omitted from further analysis. The amplitudes of the 9 frequency spectra are summed. In the range of 25 to 79 MHz. A running (inclined) straight line fit is made over 2 MHz. When the amplitude is larger than the average (from the line-fit) that frequency is filtered out. In addition the power in this frequency range is determined as the sum of the square amplitude (in frequency). This power will later be used to normalize spectra. The antenna positions are obtained from the files in directory "`AntennaFields/`" (in "MainDir") and written to the auxiliary file.

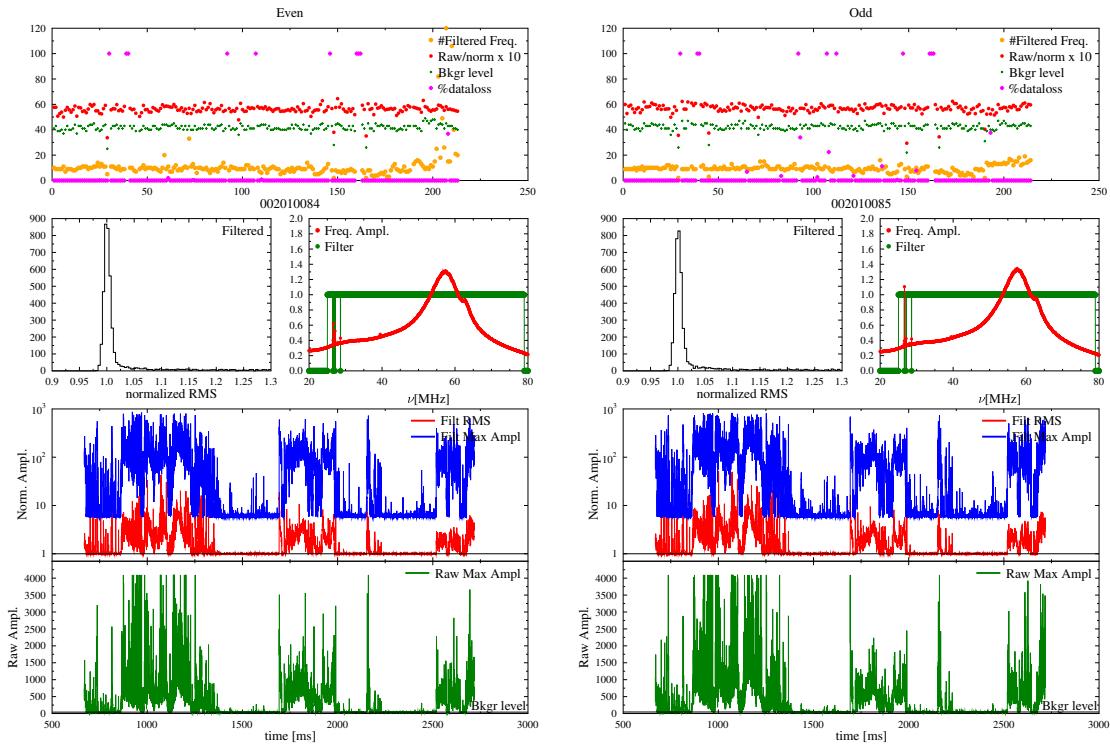


Figure 1: RFI plot results result for a typical flash, 20A-7 in this case.

3.2 Figures and print-out

In Fig. 1 a typical plot is shown as generated by the RFI-mitigation program. The results are presented separately for Even- (odd-) numbered antennas officially known as Y- (X-)dipoles. The top panel shows for each antenna (over 200 even and odd antennas for this case). For all values it is important that it does not vary too much, where outliers need to be scrutinized.

orange The number of notch filters to mitigate RFI within the specified frequency band (30-80 MHz).

The RFI filters are determined per antenna and taken the same for all data-chunks.

red The normalization factor between raw data and the noise-normalization.

green For every chunk of data (0.3 ms) the maximum amplitude is determined. The minimum value for all chunks of these maxima/chunk, a measure of the background, is shown.

magenta The percentage of data chunks that have too much data loss to be considered in the analysis.

The panel “Filtered” gives the Distribution of mean power (square of the complex amplitude) after filtering per chunk for all chunks in the reference station. The panel next to it gives the frequency filter (for the reference station) in green and the antenna gain in red.

The bottom panels give, again for the reference antenna, for each chunk, the RMS power after RFI filtering (the same data as used in making the plot in panel “filtered”), and the maximum amplitude (blue: after filtering, green: before filtering). This gives a good indication of the time-frames with lightning activity. Note that for this example the reference antenna had no data loss, if it had, the blue line would go to the bottom of the panel for the chunks with data loss.

Part a typical output (the .out file) that contains useful info for further imaging (the non-diagnostic part).

```

1 time range for which there are data [ms] 669.0022399999992 2717.0073600000001
2 !!!!!!! For 22 Antennas RFI-mitigation failed, out of a total of 427 !!!!!!!
3 5094 5095 7093 11088 11089 11090 11091 13088 32086 32087 103092 103093 121091 125088
4 MinAmp statistics: 41.883950617283951 41.991680539934350
5 statistics powr=sq(raw/Norm): 31.767085498017074 31.995666167967695
6 !!!!!!! Bad antenna based on Raw Amplitude range 5092 25 32.866010868572779 50.901890365995122
7 !!!!!!! Bad antenna based on powr 5092 11
8 !!!!!!! Bad antenna based on Raw Amplitude range 5093 26 32.866010868572779 50.901890365995122

```

This summary lists the antennas that are -most likely- better be omitted from subsequent analysis. Given is the Station Antenna ID (SAI) number that is used internally in the analysis code. The translation table from station ID to station Mnemonic is

1	2	3	4	5	6	7	11	13	17	21	24	26	28	30	32	103	106
CS001	CS002	CS003	CS004	CS005	CS006	CS007	CS011	CS013	CS017	CS021	CS024	CS026	CS028	CS030	CS032	CS103	RS106
121	125	128	130	141	142	145	146	147	150	161	166	167	169	181	183	188	189
CS201	RS205	RS208	RS210	CS301	CS302	RS305	RS306	RS307	RS310	CS401	RS406	RS407	RS409	CS501	RS503	RS508	RS509

The (most often) long line with numbers are the SAI of stations where the RFI-suppression filter could not be determined due to too much data loss. The following lines give the SAI of the antennas that are suspect because the power-normalization was outside reasonable bounds (red dots in top panels of Fig. 1), or because the amplitude range was off (green dots in Fig. 1).

4 The ‘explore’ option

For the time-calibration we will use a boot-strap method that necessitates the use of some strong pulses for this particular flash. To locate these the program needs to run the script `Explore.sh` where the input options are read from the file "Explore.in" in the FlashFolder. Typical input lines are

```

1 &Parameters
2 Explore=.true.
3 Calibrations= "Calibrations_ZERO.dat"
4 ! ExcludedStat= "RS210", "RS310", "RS208", "RS409"
5 SignFlp_SAI= 21078, 125028, 142092, 142093, 145032, 145033, 166015
6 PolFlp_SAI= 021092, 30028, 32092, 106044, 106064, 125028
7 145032, 146092, 166014, 181048, 181092, 189076
8 BadAnt_SAI= 003065, 021078, 021079, 021093, 026030, 32016, 32017, 32048, 32049
9 101045, 125029, 150065, 188051, 188094, 188095
10 31086, 31088, 31071
11 &end
12 - - - - -

```

where `Calibrations= "Calibrations_ZERO.dat"` point to a calibration file in the sub-folder Book that was copied from an earlier flash calibration. All text on a line after an exclamation mark is considered comment and not used. The following lines are obtained from experience with other flashes, where:

`ExcludedStat=` list the stations that should be excluded from the searches.

`SignFlp_SAI=` list the antenna IDs where the signal is reversed.

`PolFlp_SAI=` list the antenna IDs where the polarization is reversed.

`BadAnt_SAI=` list the antenna IDs where the signal is bad.

Note that none of these lines in the namelist have a comma at the end, for the other input lines this is optional. Several keywords may appear on a single line if separated by a comma. Anything after an

exclamation mark (within the namelist) is treated as a comment. A complete list of namelist keywords is given in Table 4. For the exploration run the details are not important since the aim is just to get a general impression.

The program will produce a file in the FlashFolder "Explore.out" that has diagnostic information. The script will also produce a plot "Map_Explore.pdf" that gives an overview of the image at 30 ms time steps.

For the exploratory search of the flash only the antennas within 2.5 km from the core are used.

4.1 Figures and print-out

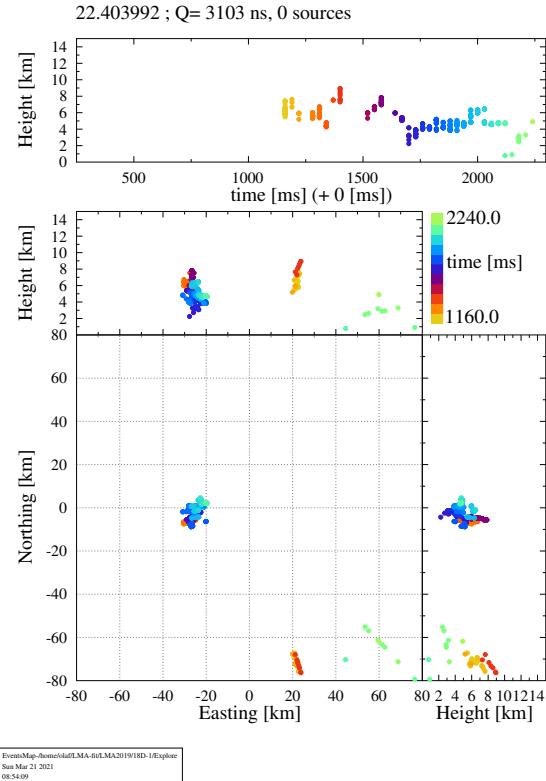


Figure 2: Exploratory view of Flash 18D1, showing the areas and periods of lightning activity.

At the end of the “explore” run a plot is made that should resemble Fig. 2 giving a rough idea of the spatial and temporal extent of the flash.

5 The ‘calibrate’ option

This option is used to perform the time-calibration of all antennas and antenna stations.

The calibration is run from the script "Calibrate.sh" where the input is set in the file "Calibrate.in". The calibration script has to be run many times over, first to obtain sources of sufficient quality (stand alone, little interference with others, similar structure for all antennas, visible in even and odd numbered antennas) using calibration files from another flash, followed by runs where the antenna calibration is fixed at the nanosecond level. The scripts need to be run in the FlashFolder.

The general philosophy is to select a few of the strongest pulses in different parts of the lightning discharge. These should cover the spatial extent of the flash reasonably well. The exploration map,

see Section 4, is a good tool to select times at which to search for the good sources for time calibration. Pulses located in a few independently selected chunks of data on the reference antenna should be fitted simultaneously, i.e. fitting simultaneously all source locations as well as station and antenna timings.

5.1 First calibration stage

For the first run the typical content of "Calibrate.in" is like

```

1 &Parameters
2 ! CurtainPlot=.true. ,
3 ! XcorelationPlot=.true. ,
4 ! FullAntFitPrn=.true.
5 ! FitIncremental=.false. ,
6 AntennaRange=5 ,
7 FitRange_Samples=100 ,
8 Dual=.false.
9 Calibrations = "Calibrations_ZERO.dat"
10 ! WriteCalib=.true.
11 PeakNr_dim=30
12 PolFlp_SAI= 021092, 30028, 32092, 106044, 106064, 125028
13 145032, 146092, 166014, 181048, 181092, 189076
14 BadAnt_SAI= 003065, 021078, 021079, 021093, 026030, 32016, 32017, 32048, 32049,
15 101045, 125029, 150065, 188051, 188094, 188095
16 &end
17
18 1200 , 8196.54 -15215.28 2838.11 , 0 0 ! StartTime_ms , SourceGuess(N,E,h)
19 1250 , 10319.85 -15302.11 4817.81 , 0 0 ! StartTime_ms , SourceGuess(N,E,h)
20 1285 , 9279.98 -19032.77 2994.75 , 0 0 ! StartTime_ms , SourceGuess(N,E,h)
21 - - - - -
22 only !

```

Some of the parameters were explained already in Section 4, all are discussed in the following and labelled by line number. The order of the parameters in the namelist input is arbitrary. All input on the line after an exclamation sign (!) is ignored and may be used for comments, or storing parameter one may want to use on some other run.

- 2 "! CurtainPlot=.true.": No curtain plots (see Section 5.4.1) are made since the default for this variable is .false..
- 3 "! XcorelationPlot=.true.": No plots of the cross correlations (see Section 5.4.2) are made since the default for this variable is .false..
- 4 "! FullAntFitPrn=.true.": No extensive printout giving the time offsets per antenna per source (see Section 5.4.3) is made since the default for this variable is .false..
- 5 "! FitIncremental=.false.": Since the default for this variable is .true. antennas within an increasing range around the core will be included in the source locating fits, where the source locations of the previous fit are used as the guess locations for the following iterative fit in which more distant antennas are included.
- 6 " AntennaRange=5": All antennas within a distance of 5 km will be included in the calculations.

- 7 " FitRange_Samples=100": The maximal range, in samples (of 5 ns), that is searched for a pulse around the predicted location (based on the pulse location guess). The range may be made smaller if the error on the guess location is small. In a first run the value for `FitRange_Samples` should be relatively large to be able to find peaks also for antennas that are very poorly calibrated. Later in the calibration process it will be tuned down.
- 8 " Dual=.false.": The peaks in even and odd numbered antennas will be searched for independently. In a later round the two will be coupled.
- 9 'Calibrations = "Calibrations_ZERO.dat)": The default timing calibrations will be used. If there is a better guess available it is advised to use it.
- 10 "! WriteCalib=.true.": The obtained new antenna calibrations (if calculated) will not be written to an updated calibration-data file.
- 11 " PeakNr_dim=30": The maximum number of pulses that can be fitted is 30 which, at this stage, should be taken larger than ($8 \times \#$ time-slots), where the time-slots are given in following lines.
- 12 " PolFlp_SAI= 021092": The antenna SAI identifiers (see Section 4.1) of the even numbered antennas where apparently even and odd have been swapped in the field.
- 14 " BadAnt_SAI= 003065, ": The antenna SAI identifiers (see Section 4.1) of the antennas that do not have good data.
- 18 ... 20 " 1200 , 8196.54 -15215.28 2838.11 ": Three lines (in this example, but the number is free) specifying the time-slots where pulses should be searched as well as the general expected location of the sources. separating commas are not necessary. These times and locations are obtained from the earlier Explore run. It is recommended to stay below 5 entries since otherwise one may run into memory problems. It is possible to try different sections of the flash in separate runs.
- 1200** The data chunk of 0.3 ms for the reference antenna, starting at 1200 ms, will be searched for the strongest peaks.
- 8196.54 -15215.28 2838.11** The position in [m] using (N,E,h) that is taken as a first guess for the source positions. The location should be in the right quadrant, but not more precise.
- 21 " - - - - - - - - - ": A closing line as long as it does not contain a line with number, but only text.
- 22 "only !": No search for antenna calibrations is done at this stage.

Running the script "`Calibrate.sh`" for this input file goes fast and will crate the files "`Calibrate.out`" with much diagnostics concerning the fit results for all sources as well as a series of plots, see Section 5.4.

5.1.1 Discussion

At this first go for the calibration the automatic pulse-finding algorithm is used to locate the strongest pulses in each to the `ChunkNr_dim` blocks of data (chunks in the language of LOFLI) starting at the times specified in the lines following the namelist input (starting with "`&Parameters`" and ending with "`&end`"). Any empty lines are skipped. The namelist input specifies the values of pre-defined control parameters where a complete list is given in Table 4.

The antennas are limited to those within a range of `AntennaRange=5` km from the core since the experience is that usually, due to non-optimal calibration, the fitting procedure starts to derail for antennas at larger distances, making the results useless. It is a matter of trial and error to learn what is best for this particular flash. To learn what was going on the output or this run, "Calibrate.out", should be scrutinized to find the sources and their deviations. At this stage one should pay particular attention to stations that show a large deviation. These need to be excluded (for all sources or for particular sources) for the time being (see following sections). It is in this respect also instructive to inspect the produced plots (all of them, quite a number) showing the shape of the cross correlation for each used antenna. This generally shows in the blink of an eye what went on during fitting.

If all spectra look crazy, decrease `AntennaRange` and try again. If the results are reasonable, you may want to increase the value or proceed.

5.2 Second calibration stage

For a limited value for `AntennaRange` sources positions for many of the pulses have been found. The source locations are not very reliable yet, but the peak in the cross correlation for these sources looks healthy (for you to judge!). The objective of this stage is to obtain the station-timing calibrations. Antenna timings will be dealt with in the last, third, stage. All good things come in three!

At this second stage the automatic pulse finding is by-passed and the locations of the pulses in the reference antennas (one for even and one for odd polarization) is specified explicitly as well as a first guess for their location. A typical content of "Calibrate.in" is

```

1 &Parameters
2 CurtainPlot=.true. ,
3 XcorelationPlot=.true. ,
4 ! FullAntFitPrn=.true.
5 FitIncremental=.false. ,
6 AntennaRange=50 ,
7 FitRange_Samples=70 ,
8 Dual = .true. ,
9 Calibrations= "Calibrations202001241344.dat"
10 WriteCalib=.true.
11 SignFlp_SAI= 21078, 125028, 142092, 142093, 145032, 145033, 166015
12 PolFlp_SAI= 021092, 30028, 32092, 106044, 106064, 125028
13 145032, 146092, 166014, 181048, 181092, 189076
14 BadAnt_SAI= 003065, 021078, 021079, 021093, 026030, 32016, 32017, 32048, 32049
15 101045, 125029, 150065, 188051, 188094, 188095
16 &end
17
18 1200 , 8196.54 -15215.28 2838.11 , 0 0 ! StartTime_ms , SourceGuess(N,E,h)
19 1250 , 10319.85 -15302.11 4817.81 , 0 0 ! StartTime_ms , SourceGuess(N,E,h)
20 1285 , 9279.98 -19032.77 2994.75 , 100 0 ! StartTime_ms , SourceGuess(N,E,h)
21
22 1 0 1 24633 -43585.46, 5624.30, 4043.00, -0.15 RMS[ns]= 1.38, sqrt(chi^2/df)=
23 exclude RS106
24 2 0 1 27403 -45564.77, 7630.14, 5605.47, -0.17 RMS[ns]= 5.47, sqrt(chi^2/df)=
25 exclude RS307
26 3 0 1 17753 -45524.56, 7567.17, 5828.33, 0.28 RMS[ns]= 1.32, sqrt(chi^2/df)=
27 4 0 1 13914 -45509.36, 7592.17, 5808.88, -0.19 RMS[ns]= 4.01, sqrt(chi^2/df)=

```

```

28 exclude RS307 RS508
29 .
30 .
31 .
32 16 1 3 5631 -44100.44, 6050.22, 7115.12, -0.21 RMS [ns]= 1.88, sqrt(chi^2/df)= 1
33 17 1 3 22867 -44192.72, 6681.31, 7684.49, -0.04 RMS [ns]= 1.78, sqrt(chi^2/df)= 1
34 exclude RS310 RS407 RS409
35 18 1 3 7640 -44054.82, 6066.37, 7106.88, 0.10 RMS [ns]= 5.68, sqrt(chi^2/df)= 1
36 exclude RS307
37
38 -----
39 only CS026 CS028 CS030 CS032 CS103 RS106 CS201 RS205 RS208 RS210 CS301 CS302 !
40 only RS305 RS306 RS307 RS310 CS401 RS406 RS407 RS409 CS501 RS503 RS508 RS509 !
41 abut CS001 CS002 CS003 !
42

```

The preamble, the namelist input, as well as the specification of the data-blocks (chunks in the language of LOFLI) is similar to what was used earlier, however now a list of pulses is given. This pulse information is (mostly) cut-and-paste from "Calibrate.out", so don't worry about typing, however the formatting, i.e. spaces and so, is important.

`FitIncremental=.false.` is important at this stage. All stations within a distance of, say `AntennaRange=70` [km] are taken into account where the cross correlations are calculated over an interval of `FitRange_Samples=50` [samples]. The precise values for these parameters is again a matter of trial and error. At the end of this stage you want to reach `AntennaRange=100` and `FitRange_Samples=20` while starting with the values from the first stage, see Section 5.1.

In this stage it is recommended to use `Dual = .true.` to combine the calibrations for odd and even polarizations. With this option the sources for pulses allocated to the same sample numbers in the even and odd polarized reference antennas are taken identical.

A new and updated calibration table will be produced with `WriteCalib=.true.` while the old one remains. In the output "Calibrate.out", towards the end, you will find the name. If the result from this run gains your approval, you can used the updated table for a following run by specifying its name as in `Calibrations= "Calibrations202001241344.dat"`, for example.

The important new ingredient here is the list of pulse positions, source locations and stations that are excluded for a particular source. An undetermined number of lines is being read until there is a clash in format. The line "-----" does so. A line starting with single or double digits gives for each pulse in order:

- 1) A sequential number (ignored on read in).
- 2) The polarity (0 or 1). For each chunk polarity 0 sources are followed by polarity 1 sources.
- 3) The chunk number (1,2, or 3 in the present case). Sources should be ordered according to chunk number.
- 4) The sample number for the pulse location in the reference antenna (integer).
- 5) The coordinates for the source location as (N,E,h,dt).

The remainder of this line is ignored upon reading. The line may be followed by a line starting with "exclude" listing the stations that should not be taken into account for this source when fitting.

When `Dual = .true.`, two pulses in the same chunk having the same sample number are treated as coming from the same source location during fitting. For these lines the spaces matter! With cut-and-paste this section in the right format can be obtained from the output (.out) file.

The first line following "-----" specifies which station timings should be searched

for. Option ‘only’ implies what you would guess. the list is closed with an exclamation mark. The option ‘abut’ (standing for ‘all but’) also does what you think it should do when you have deciphered the meaning. Note that the options are case sensitive and should not be preceded by a space.

All remaining lines are ignored. An excellent place to store reminders or potentially useful input lines.

5.2.1 Discussion

The objective at this stage is to use several excellent-quality sources, spread over a large volume of a flash, to calibrate the station timing. The reasoning is that over the duration of the flash all pulses in a single antenna will have the same time-shift due to calibration errors, leaving their relative arrival times unchanged. By fitting the location of sources distributed over a large volume, the relative arrival times are sufficient to fix their positions. The arrival times thus fix the antenna calibrations.

To obtain the source positions we use the fact that the relative timing for antennas in a single station have an accuracy of better than 5 ns. Also the relative timing of the core stations is known to this level. At the end we do want to improve on this, however.

The source locations found for an inner circle of antennas will yield a prediction when the pulses arrive in the antennas in the next circle out losing accuracy for larger distances. Thus by increasing `AntennaRange` by not too much the pulse is likely to lie in the next ring of antennas at the predicted time within `FitRange_Samples` and will thus be located in the search algorithm. If the pulse is further away, it will not be located in this particular antenna. In other antennas the situation might be better, and thus the obtained source position is improved, while this particular antenna, for this particular pulse is marked as ‘excluded’ in the output. In a next round with the same settings, but updated source locations, the pulse may be found. This indicates the delicate balance between the various variables.

At all stages it is VERY STRONGLY recommended to look at the curtain, see Section 5.4.1, and the cross-correlation, see Section 5.4.2, plots to decide if a station should continue to be excluded, or was included while the cross correlation was a mess. At the end one prefers to have a minimal number of excluded stations.

During this process it is recommended to perform an intermediate calibration of (some of) the station timing, using the options only or abut. This will stabilize the results for following rounds.

The output file “`Calibrate.out`” contains a part that resembles the sample input. Certainly after each calibration table update the source locations should be updated, but it is better to do so more frequently.

Occasionally you may find that when making curtainplots, see Section ?? one station or even one antenna is completely off. In such a case the calibration file should be edited to correct this (it should be self evident what to do). Once the offset is reduced to less than `FitRange_Samples` the program can be used to reduce the error.

At the end of the day all station should be involved in the fitting process with at least ten sources distributed over the flash with an RMS error of about 2 ns. During the fitting process the error assigned to the extracted peak in the correlation spectrum is 1 ns.

Do not despair, after a few weeks of struggling you are sure to have some vague idea of the logic behind this procedure.

5.3 Third calibration stage

In the third calibration stage the individual antennas will be calibrated. The same procedure will be followed as for the second stage, only now `Fit_AntOffset=.true.` is set in the namelist input.

Since fitting all antennas simultaneously requires too much memory, this is done in different batches. At the end of the day a χ^2 of order unity should be obtained, or an RMS deviation of 1 ns.

5.4 Figures and print-out

5.4.1 Curtain plot

When you feel completely lost (but also if you are not), it is instructive to switch on `CurtainPlot=.true.` to make a -so named- curtain plot showing the spectra for all antennas, sorted per peak (sorry, only for peaks listed for the even numbered antennas) and separate for even and odd antenna numbers. In this case the fifth entry is read on the line just before the list of sources, like (omitting all non essential lines)

```
1 1285 , 9279.98 -19032.77 2994.75 , 100 0 ! ! StartTime_ms , SourceGuess(N,E,h) , wind
```

This entry, 100 in the present case, gives the width of the time-window, the number of time samples before and after the lined-up part, that will be shown in the figures

Many figures named like "CuP-02.pdf" will be created showing all spectra where the antenna timings are adjusted such that if these pulses indeed comes from a source at the position of source #02, all peaks would be lined-up perfectly, see Fig. 3. The time traces for the same station and polarity are overlayed.

The pulses in all antennas can be included in the calibration calculation if the pulses line-up nicely. To see this better the time-window can be adjusted. The pulse amplitudes have been normalized for each spectrum to the maximum, say unity.

The structure of the pulses in all odd numbered antennas should resemble each other. The same for the even numbered antennas. The pulse for in the odd may be different from that in the even ones. The noise level should hardly be visible on the plotted scale.

If one condition is not obeyed the antenna should be excluded for that particular source and that polarity of the antenna.

5.4.2 Cross correlation plot

In Fig. 4 a typical plot is shown of the cross correlations. The peak-time of these is used for calculating the arrival times differences that are fitted to obtain source positions. We usually work with the absolute magnitude as this yields better convergence than working with the real part. The real part has many oscillations which creates many local minima in a chi-square search, even though the position of the maximum can be determined more accurately. There is an option to plot the real parts.

For each antenna the Hilbert envelope is shown. The labels on the side helps to distinguish the different stations. Results are given in separate plots for even and odd numbered antennas. When all lines look alike, the result is good. Antennas, and stations should be excluded when the structure differs from that for the other antennas. The results for excluded stations is given by the dotted curves. It can be seen that for the case in Fig. 4 most of them show a different structure. On the basis of these plots it may be decided to include a previously excluded station again in the fit. The paradigm is: the more the merrier, i.e. generally the results for the antenna calibrations are better if more stations are included, however, if for some of these station the pulse is unclear it could be that pulses from two different sources happen to arrive simultaneously in a particular antenna, the result improves if the -obviously- erroneous result is excluded. The art is to find the optimum between these conflicting requirements.

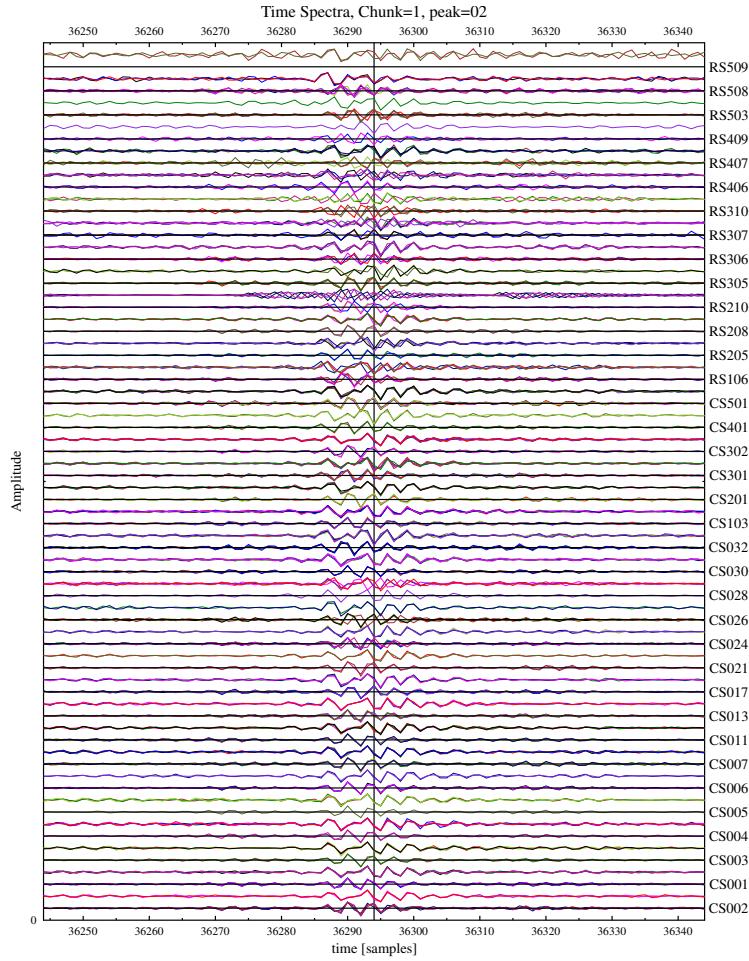


Figure 3: A curtain plot for the second source for a typical flash.

5.4.3 Print output

Part a typical output (the .out file) that contains the most useful info for further processing is:

```

1 i_Peak= 9 0, PeakPos= 16250, Chi^2/DegrF= 2.51 2.51, source position: -2141.69, -22721.36, 3935.67, RefAntTimeErr: 0.706
2 Stat Nr = CS001; CS002; CS003; CS004; CS005; CS006; CS007; CS011; CS013; CS017; CS021; CS024; CS026; CS028; CS030; CS032; CS103; RS106; CS201; RS208; RS210; CS301; CS302;
3 Dropped#= 0/ 5; 0/ 4; 0/ 4; 0/ 3; 0/ 2; 0/ 3; 0/ 4; 0/ 3; 0/ 6; 0/ 4; 0/ 2; 0/ 4; 0/ 5; 0/ 4; 0/ 4; 0/ 5; 0/ 2; 0/ 4; 0/ 3; 0/ 4; 0/ 5;
4 RMS [ns]= 1.2; 0.6; 0.6; 0.8; 0.2; 0.4; 0.8; 1.0; 1.3; 0.7; 0.4; 0.5; 1.0; 0.9; 0.8; 0.5; 1.3; 0.9; 1.3; 2.4; 2.7; 0.4; 0.8;
5 Avrg[ns]= 1.0; -0.3; 0.5; 0.8; 0.1; -0.3; 0.5; 1.0; 1.0; 0.5; 0.4; 0.3; 0.8; 0.4; 0.8; 0.3; 1.1; 0.5; 1.3; -2.3; -2.3; 0.3; 0.7;
6 RMS(Jac)= 0.06; 0.00; 0.02; 0.01; 0.01; 0.02; 0.05; 0.01; 0.05; 0.12; 0.02; 0.16; 0.13; 0.04; 0.04; 0.58; 0.05; 2.77; 4.06; 0.15; 0.21;

```

There is such a section for each peak. the ones towards the end of the output are for the last fits. This list for each station 1) the number of antennas that are dropped all together, because that one showed an unreasonably large difference; 2)the contribution to the RMS, 3) the mean time difference. If this matches (in absolute magnitude) the RMS for this station, then all antennas show about the same difference, if not, one of the antennas differs considerably. the last line shows the importance of this station for pinning down the source location.

Towards the end the following can be found

```

1
2   6 1 2  18841 -72112.37,  21167.93,  5397.60,    0.00;   1.63,   1.67 RS406 RS509
3 exclude RS406 RS509
4   7 0 3  27662 -6112.90, -28762.40,  6490.86,  -0.39;   1.80,   1.80
5   8 1 3  27662 -6112.90, -28762.40,  6490.86,  -0.37;   2.38,   2.50 RS208 RS210 RS310
6 exclude RS208 RS210 RS310
7   9 0 4  16250 -2141.69, -22721.36,  3935.67,   0.71;   1.58,   1.58

```

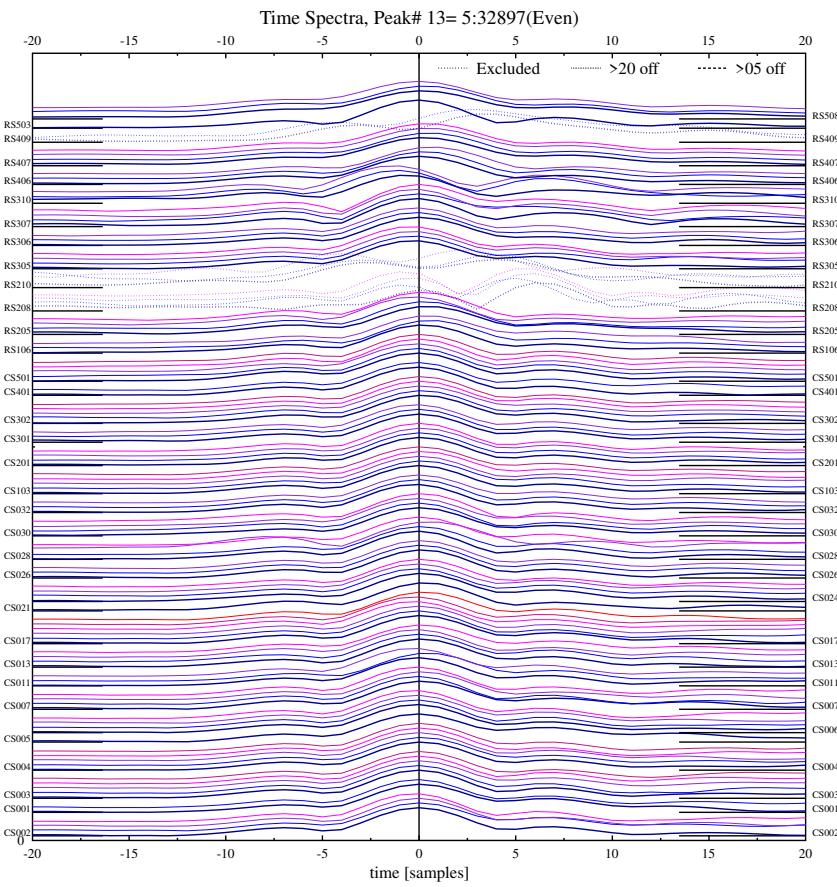


Figure 4: A plot of the absolute cross correlations for source #13 for a typical flash.

which follows the same formatting as the list needed in the input. With cut-and-paste it can thus be used conveniently. Per pulse it lists the fitted position and the value of the $\text{sqrt}(\chi\text{-square})$ (last numbers) for this source. For a good result these numbers should not exceed 2 by much (units are nanoseconds). if the number is large, it should be searched which station is responsible for this, either by checking the listing for the pulse (see previous discussion) and/or checking the curtain and/or the cross correlation plots. This is the most tedious part of the calibration runs.

When the option " `FullAntFitPrn=.true.`" is switched on there is for each peak and each antenna, ordered per station, a list of the time differences between determined arrival times and calculated ones, i.e. the pull values for calculating the chi-square. On the basis of this information is may be decided to label a particular antenna as 'bad'.

6 Impulsive imaging or source finding

Source finding is performed by the script "`Imaging.sh`" residing in the FlashFolder. The running of the script is controlled by "`Imaging.in`", which typically resembles

```

1 &Parameters
2 ImagingRun=.true.
3 Dual=.true.
4 ChiSq_lim= 50.

```

```

5 NoiseLevel = 300.
6 AntennaRange=100
7 FitRange_Samples=70 ! 50
8 ! RealCorrelation=.true.
9 Calibrations= "Calibrations201912042045.dat"
10 BadAnt_SAI= 003065, 021078, 021079, 021093, 026030, 32016, 32017, 32048, 32049
11 101045, 125029, 150065, 188051, 188094, 188095
12 OutFileLabel="-s"
13 &end
14
15 1165 , -45500. 7700. 5300.0 , 101.7 300 ! StartTime_ms , (N,E,h), t_init [ms], t_fin [ms]

```

While in imaging mode only very few parameters need to be specified and often the default settings will suffice.

- 2 "ImagingRun=.true.": Run in imaging mode.
- 3 " Dual=.false.": The peaks in even and odd numbered antennas will be searched for independently. In a later round the two will be coupled.
" Dual=.true.": In searching for the source locations the peak positions in both even- and odd-numbered antennas are used.
- 4 "ChiSq_lim= 50.": Sources with a chi-square better than 50 will be stored.
- 5 "NoiseLevel = 300.": Only for pulses with an amplitude in the reference antenna greater than 300 the source location will be searched.
- 6 " AntennaRange=100": All antennas within a distance of 100 km from the reference antenna will be included in the calculations.
- 7 " FitRange_Samples=70" The maximal range, in samples (of 5 ns), that is searched for a pulse around the predicted location (based on the pulse location guess). The range may be made smaller if the error on the guess location is small. In a first run the value for FitRange_Samples should be relatively large to be able to find peaks also for antennas that are very poorly calibrated. Later in the calibration process it will be tuned down.
- 8 ' ! RealCorrelation=.true.': The Hilbert envelopes of the cross correlation functions is used to determine arrival-time differences in the imaging procedure.
- 9 ' Calibrations = "Calibrations201912042045.dat"': The name of the calibration file created at the end of the time calibration procedure.
- 10 " BadAnt_SAI= 003065, ": The antenna SAI identifiers (see Section 4.1) of the antennas that do not have good data.
- 12 ' OutFileLabel="-s" ': An additional label for the output files, including the list of sources.
- 15 " 1165 , -45500. 7700. 5300.0 , 101.7 300 ": Specification of the starting time, approximate location of the flash, and the time frame where sources should be searched.

The last line in the input specifies

- 1) The start time, $T_0 = 1165$ (in ms), of the flash. The output of the RFI-mitigation and also that of

the Explore runs should give you a reasonable estimate of when the flash started.

2–4) The general direction towards the flash (N,E,h) in m.

5) The time after T_0 (in ms) to proceed with source finding. It may be negative.

6) The time after T_0 (in ms) to stop with source finding.

6.1 details

The results for the source locations are written to `Srcs18-dblxxxx.csv` or otherwise the files `Srcs18-oddxxxx.csv` and `Srcs18-evenxxxx.csv` in the FlashFolder. These .csv files are plain text files and contain some header lines with some general information followed by the specific data of the sources that are found and pass some very crude criteria. In particular $\chi^2 < \text{ChiSq_lim}$. The data for the sources is organized in columns where:

col1: Unique label composed from the chunk number (after T_0) and the pulse number in the chunk.

col2-4: distances northward, eastward, upward [m].

col5: Time at the source [s].

col6: χ^2 , if this had units, they would be [ns²] .

col7-9: 1 std error in north, east, vertical direction [m].

col10: Number of antennas that have contributed to imaging this source.

col11: Total number of antennas for which data are available for this chunk.

See later sections for the details of the source-finding procedure including the criteria for excluding antennas.

6.2 output

The most important output is the list of source locations with their quality identifiers. Additionally a plot is made of the peak-finding statistics like shown in Fig. 5.

7 Plotting flashes

For plotting it is advised to use the script `FlashImage.sh` that reads the input from `FlashImage.in` specifying values for the quality indicators as well as the windows for which the source locations and timings will be plotted. The script `FlashImage.bat` is for running under windows.

```

1 "Srcs18-evenS" 5. 2. 3.0 55   18D1e-i   -30.00 -27.  -7.5 -5.0  4.5 7.0  0. 30.  NoBox
2 0.1 0.5 0.1      ""    1.  1.   1   0.5                                ! MaxTrackDi
3 ======
```

The first line gives in order:

1) The name of the .csv file containing the sources data as created by an imaging run.

2) The cut σ_0^h on σh

3) The height h_0 . All sources higher than h_0 are plotted when $\sigma h < \sigma_0^h$. Sources at lower altitudes h will be plotted when $\sigma h < \sigma_0^h \times (h_0/h)$.

4) RMS cut value, in [ns].

5) N_{ex} , the maximum number of excluded stations.

6) naming of the .pdf file containing the image.

7 – 14) limiting values for east, north, height (all in [km]) and time (in [ms] after t_0

15) if "NoBox" is specified, no high-lighting box will be drawn, otherwise the coordinates of the box will be read from the files for the specified plot.

The second line gives the parameters for calculating track that will be indicated in the image of

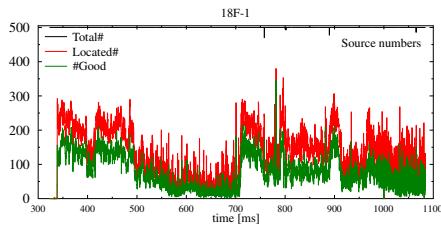


Figure 5: Statistics of the source-finding procedure for the impulsive imager. For each time frame the number of located sources is given with a chi-square below the maximum specified.

the sources. In order:

- 1) maximal distance for a source to be removed from the track-head to be assigned to a track.
- 2) Weighting of additional points to determine the position of the track-head.
- 3) Width of gaussian time window to smooth the track.
- 4) Name of file containing predefined track.
- 5) Binning window for determining track statistics such as velocity.
- 6) Factor to scale height in the calculation of relative distances.
- 7) Integer. The maximum number of tracks to be generated.
- 8) Factor used to scale intensity-dependence of the size of the dots when plotting source positions.

The result is displayed in Fig. 6 for the image of the flash for the selected area. The normalized pulse powers distributions $N(I)$ are fitted with a modified exponential,

$$N(I) = \mathcal{N}_e e^{-\alpha_e I - \gamma_e/I^2}, \quad (1)$$

as well as with a modified powerlaw, T

$$N(I) = \mathcal{N} I^{-\alpha} e^{-\gamma/I}, \quad (2)$$

where I is expressed in units of [GB]. The last factor, dependent on γ , suppresses the distribution at small amplitudes to good agreement with the data. The values for the fitted values for the normalization \mathcal{N} , the power α , and the small-intensity suppression factor γ are given in the output file (with extension .out).

If the 7th number on the second line in the input, "FlashImage.in", is positive, a graph like Fig. 7 is made for each track. The second panel from the top shows the pulse density along the flash as

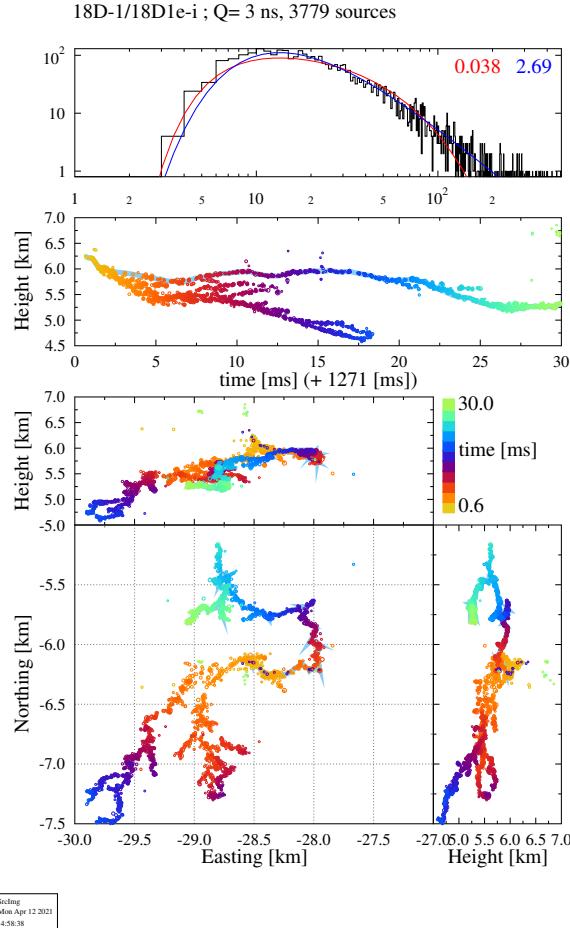


Figure 6: Typical image for the Impulsive Imager as created by running “FlashImage.bat”. Light blue band is the reconstructed track (starting from the latest point and tracing back). Top panel give pulse power statistics where the modified exponential is plotted in red and the modified power law plotted in blue.

function of time for various width of a gaussian smoothing function. The top panel shows the fourier decomposition of this plot. The third from the top gives the velocity along the track. The blue line for each source along the track, the red line for an average leader-tip location. The bottom three panels show the spread of the sources in the three directions from the propagating tip of the leader, in blue as a scatter plot v.s. time of the source (top and right scales), in red as a histogram (bottom and left scales).

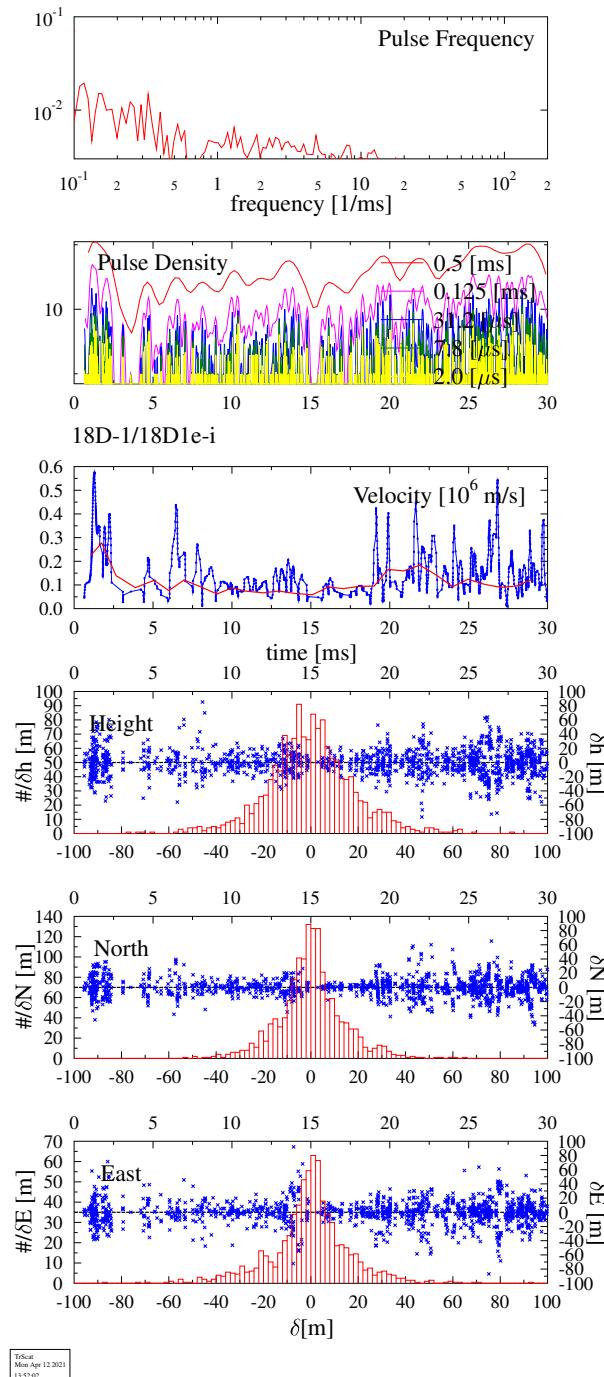


Figure 7: Typical image for the Impulsive Imager when showing the statistics along a track.

8 Interferometric imaging

Interferometry can be performed working only with the signals measured in the antennas, i.e. without unfolding the antenna function, as is done in this section, and which should be referred to as Signal Interferometry (SI). More sophisticated is to convert the measured signal in the X- and Y- dipoles to the radiation electric field at the antenna and use this for interferometry. For future reference the latter will be named E-field Interferometry (EI) and will be discussed in the following chapter.

Source finding is performed by the script "`Interferometry.sh`" residing in the FlashFolder. The running of the script is controlled by "`Interferometry.in`", which typically resembles:

```

1 &Parameters
2 Interferometry=.true.
3 IntfPhaseCheck=.false.
4 ! CurtainPlot=.true.
5 TimeBase=669.
6 Calibrations = "Calibrations202012211039.dat" ! updated for v18 old peak 30 removed, chi^2= 3
7 BadAnt_SAI= 007093, 005092, 005093, 011088, 011089, 011090, 011091, 013088, 013089, 032086,
8 103092, 103093, 121091, 125088, 125089, 125092, 125093, 130092, 130093, 130094, 13
9 145084, 145085, 145088, 145089, 166094, 166095
10 SignFlp_SAI= 142092, 142093 !, 106090, 106091, 106092, 106093
11 ExcludedStat = "RS208", "RS508", "RS307"
12 OutFileLabel="XYZ"
13 AntennaRange=50. ! [km]
14 IntfSmoothWin=20
15 ChainRun=0
16 NoiseLevel=0.01 &end
17 245.1 , -25.292 37.5200 5.660 ! StartTime_ms=278.5, (N,E,h)CenLoc
18 P 50 .005, 50 .03 , 9 50. ! N_phi, d_N, N_theta, d_E, N_R, d_h[m]
19 4000 60040 100 1.0 ! Ini [samples] starting loc & window for summing power & slice number &

```

This script produces interferometric images (see Section 8.4 for the procedures followed) and puts the results in several data files in the subdirectory "`files`". It also prepares the command files (in "`GLE-plotsXYZ.sh`") for running the GLE scripts [Pugmire et al., 2015] to produce "`InterfContourXYZ.pdf`", "`XYZInfImgBar_0.pdf`", "`XYZInfImaMx_0.pdf`", "`InterfTrackXYZ.pdf`" and also equivalent files with extension 1 for the odd numbered antennas (officially X-dipoles). It should be noted that the interferometry option runs for even and odd numbered antennas in two separate cycles and produces independent images. The interferometric hypercube is also shown in the plot of the source locations.

It is recommended to subsequently run the script "`InterfSrcSel.sh`" using "`InterfSrcSel.in`" as input to produce the plots that are zoomed in on the region of interest, see Section 8.3.

8.1 Input detailed

The lines in the namelist "`&Parameters`" input specify:

2 "`Interferometry=.true.`": Switch to interferometry mode.

3 "`IntfPhaseCheck=.false.`": Do not print for each antenna the amplitude, phase, and angle to the source, done for the central pixel. This can be helpful to check if there are certain stations that are off in certain ways, but the info is difficult to interpret. Make sure that the direction

for the central pixel is such that it is perfect for the largest source in the selected time trace. When commented, this information will appear in the output.

- 4 "CurtainPlot=.true."!: When un-commented, a curtain plot (see Section ??) will be made for the time-traces lined up for the central pixel.
- 5 "TimeBase=669.": This time is added to the relative time specified in the input.
- 6 "Calibrations =": The name of the file containing calibration and RFI information.
- 7 "BadAnt_SAI)": These antennas are excluded from the analysis.
- 10 "SignFlp_SAI)": The amplitude for this antenna is multiplied by minus unity.
- 11 "ExcludedStat)": Mnemonics of the stations that will be excluded from interferometry. The exclusion is usually based on unexpected phases seen when running with "IntfPhaseCheck=.true.".
- 12 "OutFileLabel)": Additional label used for the output files, including the plots.
- 13 "AntennaRange)": The maximal distance (in [km]) from the reference station of the antenna stations that are included in interferometry.
- 14 "IntfSmoothWin)": Integer. The width of the sampling function used for Time Resolved Imaging (in [samples]). This number should be of the order of the impulse-response time, about 20 samples.
- 15 "ChainRun)": Integer. If =0 the present run will not spawn any children.
If positive another run is generated with identical parameters except
 - The central location of the hypercube (line 17) which will be pinned down to the most intense spot for the present time interval.
 - The start time (line 17) will be increased by 0.3 ms. If the total window (line 19) equals 60040 (line 19, where $40=2\times\text{IntfSmoothWin}$) this will generate image data that can later be stitched together to generate (using Section 8.3) a continuous image along a negative leader.
 - The absolute value of "ChainRun" is decreased by unity.
 - The present value of "ChainRun" is worked into "OutFileLabel" as well as the naming of the spawned scripts.

If "ChainRun=" negative, the same will be done, only backward in time.
Chaining jobs works only on Linux systems.

- 16 "NoiseLevel)": Only sources with a coherent intensity exceeding this will be imaged.

The following lines specify:

245.1 , ... Entry#1) Starting time in the trace (in the reference antenna, which differs from the time of the sources) of the data-chunk that will be used. Time is taken relative to "TimeBase".
Entry#2-4) Coordinates of the central pixel in (N,E,h) notation. The units may be [km].

P 50 .005, ... Entry#1) If capital "P" or "C" a Polar or Cartesian grid is forced. If anything else (including a space) a reasonable choice is made based on the specified grid structure. A Polar grid (recommended) is specified in the order: Azimuth angle ϕ , elevation angle θ_e , distance to reference antenna R . A Cartesian grid as: Northing N , Easting E , altitude h . Internally both are used in parallel.

Entry#2, 4, 6) Number of grid points.

Entry#3, 5, 7) Grid spacing.

4000 ... Entry#1) The first sample number used for interferometric imaging of the trace. The program needs select some previous channels as a buffer to be able to image the indicated part of the chunk in the reference antenna. The first sample thus be larger than zero. Error messages are generated and image is truncated when the value is too small.

Entry#2) The length of the window (in [samples]) used for imaging. 60k samples corresponds to 0.3 ms. The additional 40 ($=2 \times$ "IntfSmoothWin") is to accommodate for the beginning and end smoothing windows when performing Time Resolved Imaging (see Section 8.4).

8.2 Output

The produced .dat files are plain text files and contain some header lines with some general information followed by the specific data of the sources that are found and pass some very crude criteria. The files have a format that is suitable for the plotting script "SourcesPlot.gle". The naming of these data files is as in "XYZIntfSpecPowMx_1.dat", where "XYZ" is set by the user through "OutFileLabel=XYZ" when running the interferometry option; "IntfSpecPow" is fixed for these kind of files; "Mx" implies that these source positions were determined using the quadratic maximum search while for "Bar" the barycentric procedure was used, see Section 8.4.4; "_0.dat" ("_1.dat") denote the use of even (odd) numbered antennas (Y- and X-dipoles).

This file also contains the coordinates of the corners of the hypercube used in the interferometry calculation.

8.2.1 Intensity plot

A typical interferometric Intensity plot is shown in Fig. 8. These are produced separately for even-(left side of Fig. 8) and odd- (right side) numbered antennas. The top shows in red the spectrum in the reference antenna while green shows the coherent sum of all antennas in the field, beamed towards the central pixel of the hypercube. The green spectrum is normalized by the number of antennas (N_a , i.e. the green and red should fall on top of each other if the signal would be completely coherent for this direction. If the signal is random noise it should be at a fraction of $1/N_a$. This signal power is rebinned over 5 samples. With black lines the zoom-in window is indicated. The zoomed-in spectrum is shown below it, using the same convention but without re-sampling. The other figures show the intensity along a plane through the hypercube, top left at $dR=0$, the bottom two at $d\theta = 0$ and $d\phi = 0$. Step sizes and ranges are specified in the input line starting with 'P'.

The located sources are given in Fig. 9 as well as the hypercube that was used for imaging. Sources close to the borders of this hypercube should not be trusted. An image like this is made for even and odd antennas, as well as for the Max and Barycenter methods for the position of the interferometric maximum.

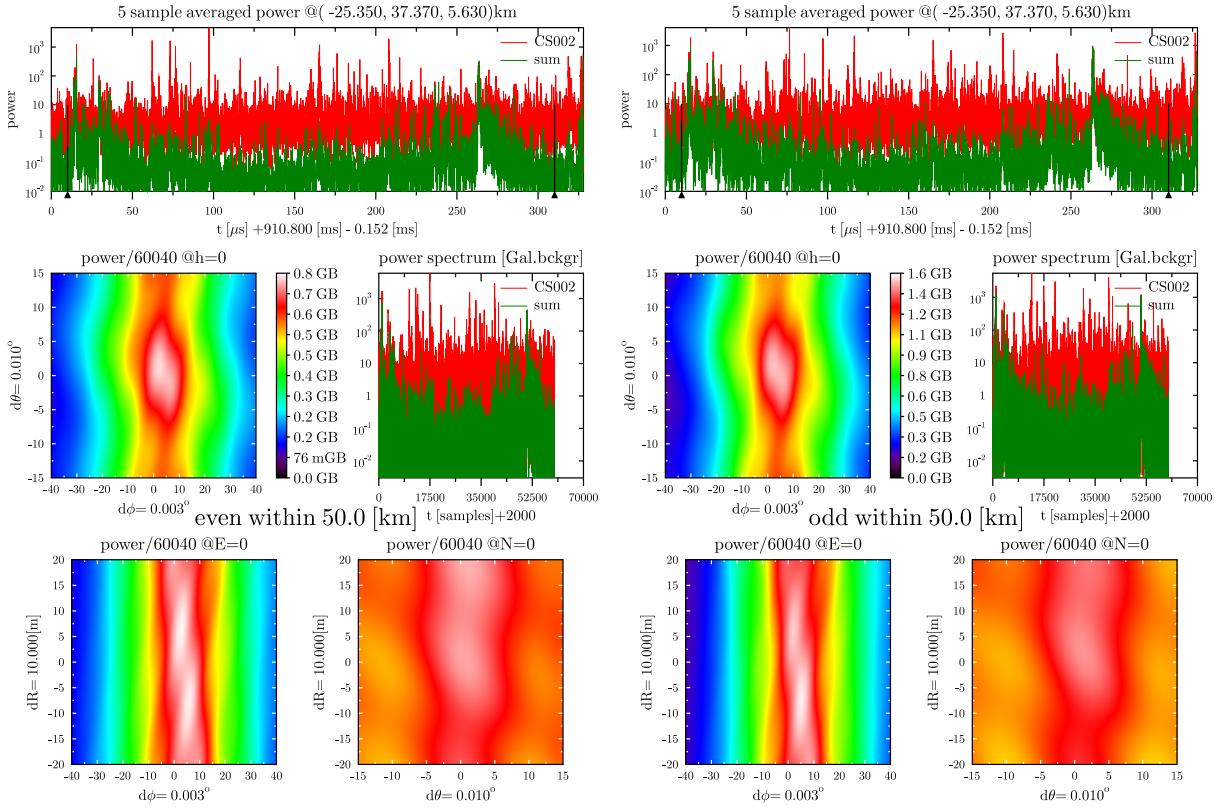


Figure 8: Typical image for the TRID Imager as created by running “InterfSrcSel.sh”.

8.3 Selective plotting

The script "InterfSrcSel.sh" using "InterfSrcSel.in" as input allows to produce plots that are zoomed in on the region of interest. This script plots the position of the sources using the GLE-script, "SourcesPlot.gle" and the distribution of sources intensities using "SourcesPlot.gle". These are run via spawned scripts on Windows as well as Linux machines. A typical input resembles:

```

1 &Parameters
2 OutFileLabel="XYZ",           ! Normal Negative Leader (a)
3 DataFile=   "SE20A7-Nh-", "SE20A7-Nh:-10", "SE20A7-Nh:-09", "SE20A7-Nh:-08", "SE20A7-Nh:-07",
4   "SE20A7-Nh:-05"
5 SMPowCut = 3.,             ! power of pulse included in the plot
6 AmpltPlot=0.1              ! dependence of dot sizes on pulse power
7 MaxAmplFitPercent=0.001    ! Maximal percentile amplitudes to be included in fits of power-s
8 ZoomClip = .true.          ! clip sources outside plotting region
9 xmin=37.15 , xmax=37.6, ymin=-25.45, ymax=-25.15, zmin=5.35, zmax=5.8, tmin=237.0 , tmax=245.
10 &End

```

The lines in the namelist "&Parameters" input specify:

2 "OutFileLabel": Additional label used for the output files, including the plots.

3 "DataFile": Distinctive "XYZ" labels of the data files discussed in Section 8.2. The data of all these files will be sorted in time and used for plotting and producing the intensity distributions.

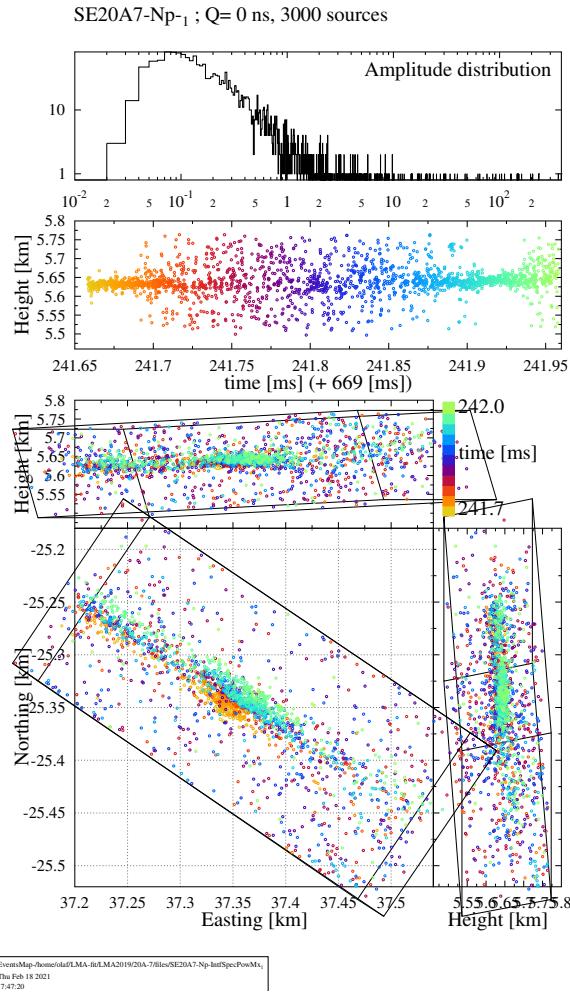


Figure 9: Typical image for the TRID Imager as created by running “InterfSrcSel.sh”.

- 5 "SMPowCut": Only stronger sources are used for plotting.
 - 6 "AmpltPlot": A factor used for scaling the dot-size when plotting. If zero all dots are the same size.
 - 7 "MaxAmplFitPercent": Percentile cut for the sources included in fitting a power-law to the distribution. Stronger sources are excluded.
 - 8 "ZoomClip": Clip source locations to the plotting volume.
 - 9 "xmin": The bounding boxes used in space and time for the plots.

The plots will be made for the ‘Mx’ and ‘Bar’ files as well as for X- and Y-dipoles, all resembling Fig. 10. This allows for appending seamless the results of different interferometry runs.

8.4 Method of Time Resolved Interferometric 3D Imaging

This section is adapted from the draft of the manuscript “Time resolved 3Dinterferometric imaging of a section of a negative leader with LOFAR”

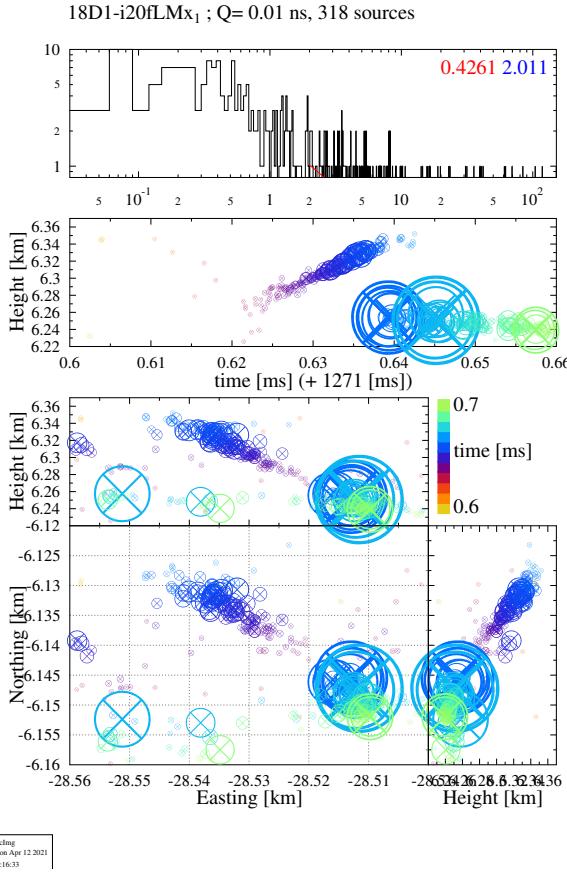


Figure 10: Typical image for the TRID Imager as created by running “InterfSrcSel.sh”.

The basic implementation of beam-forming imaging is relatively straightforward. The part of the atmosphere where the lightning is to be imaged is divided up in voxels. For each voxel the time traces of the antennas are summed while accounting for the difference in travel time from this voxel to each antenna. This yields for each voxel a time trace which is the coherent sum of all antennas, as discussed in Section 8.4.1. Adding the traces for the different voxels is always performed for a fixed time span in one particular ‘reference’ antenna, usually one from the core of LOFAR. Thus we can find the source location of a particular structure seen in the trace of the reference antenna.

The time trace of each voxel is cut in slices of a fixed duration. The intensity is determined for all slices resulting in many voxelated intensity profile, one such profile for each time slice corresponding to a fixed time slice in the reference antenna, much like what is shown in Fig. ?? but for much shorter time spans. The length of the time slices determines our final time resolution since any detail within a slice is summed over. For each time slice the maximum of this voxelated intensity profile is determined and used as the source location, as discussed in more detail in Section 8.4.4. Plotting the position of all sources will result in the beam-formed (or interferometric) image of a segment of the flash in space and time. In the following the different steps involved in this procedure are discussed in more detail.

The main selling points of LOFAR are that we can reach 1) a time resolution of 100 ns, the pulse response width as determined by the band width (30-80 MHz) 2) a spatial resolution of the order of

1 m, determined by the time calibration (order 1 ns) and the antenna baselines (up to 100 km), 3) a sensitivity of two orders of magnitude below the noise level of a single antenna, determined by the number of antennas that are summed (or the order of 100 – 200). In addition we make true three dimensional images.

8.4.1 Space & time grids and summing time traces

The part of the atmosphere that is of interest is divided up into voxels on a regular grid. We have found that this is most conveniently done in polar coordinates with the center at the reference antenna in the dense LOFAR-core. In this way we can more easily account for the fact that our resolution in the radial direction is much worse than in the two transverse directions. The optimal grid-spacing is dependent on the areal spread of the antenna locations. For the example discussed in this work we include antennas within 50 km distance from the core where the antennas are distributed on an irregular logarithmic grid with a dense core that is designed to be optimal for astrophysical applications, see [van Haarlem et al., 2013]. The voxel grid is chosen such that the maximal time shift for any antenna for two bordering voxels is about 1 ns, our time calibration accuracy. A typical grid spacing of 0.003° in the azimuth ($\hat{\phi}$) angle, 0.01° in elevation ($\hat{\theta}_e$) angle, and 10 m in the radial (\hat{R}) direction is reasonable. For a source at 50 km distance this implies a grid of about 1 m transverse to the line of sight (from the reference antenna) and 10 m along the line of sight. A typical grid spans over $(60 \times 30 \times 20)$ voxels in $(\hat{\phi}, \hat{\theta}_e, \hat{R})$.

A section of the time trace in the reference antenna is selected with typically a length of 0.3 ms. For each voxel the relative time shifts of the antennas are calculated with respect to that of the reference antenna to account for the travel time differences of a signal from the voxel. These time-shifted traces are summed to yield the coherent (or beam formed) time trace for that voxel. All coherent voxel time traces are thus evaluated at identical times for the reference antenna.

The actual time shifting and adding of the time traces is done in Fourier space by applying frequency-dependent phase shifting to each antenna trace before summing them. After summation, the traces are transformed back to the time domain. The antenna time traces used are taken longer than the required 0.3 ms for the final analysis to account for the maximal relative time shift over the full voxelated volume. We thus obtain for each voxel a coherently summed time trace of 0.3 ms length where only in the trailing times there will be partial coherence.

The intensity may be integrated over the full time trace of 0.3 ms which washes out all time dependence over the integration range. To image the dynamics in the flash we want to use a much better time resolution. The smallest time step where subsequent time-frames can be considered to be independent is the dictated by the width of the impulse response function. The impulse-response time equals the full width at half maximum of the narrow peaks in the spectrum, about 100 ns. We thus cut the time trace in slices of 100 ns and the voxelated intensity profile is determined for each slice. The 100 ns can also be regarded as a compromise, taking it shorter would make the imaging more sensitive to noise fluctuations, taking is longer would increase the chance that multiple sources appear simultaneously and are confused. Since 100 ns is close to the impulse-response time of the system we name this Time-Resolved Interferometric 3D (TRID) imaging.

8.4.2 Antenna weighting

Since the array of antennas has a large extent compared to the distance of the source region, the received signal strength varies greatly over the array. If this is not taken into account one would effectively use the antennas that are near the source region only and thus not use the full imaging power of the array. To improve the performance we thus include weighting factors for the antennas. In Interferometry applications for astronomy there is a considerable amount of experience with the pro's and

con's of different weighting schemes, see[Briggs et al., 1999, Yatawatta, 2014] for some comprehensive reviews.

For this work we have chosen for a weighting scheme that compensates the signal strength due to distance to the source in lowest order, caps the weight for antennas far from the source where the signal to noise ratio becomes unfavorable, and takes into account the fact that the largest density of antennas is at the core.

The amplitude of an emitted signal drops inversely proportional to distance, $R_{as} = \sqrt{D_{as}^2 + h_s^2}$ where D_{as} is the distance from a certain antenna to the source in the horizontal plane and h_s denotes the altitude of the source. It should be noted that the Dutch LOFAR antennas are all (to a very good approximation) in a horizontal plane as that part of The Netherlands is rather flat. The measured signal strength is also dependent on the antenna gain which depends on the azimuth, ϕ , and elevation, θ_e , angles of the source with respect to the antenna. The antenna gain vanishes for sources at the horizon and thus (since it is an analytic function of the angles) depends on θ_e as $\sin(\theta_e)$. For almost all cases of interest the sources are at small elevation angles for which we can assume $\sin(\theta_e) \approx h_s/R_{as}$ where the antenna position enters in R_{as} only.

The antenna gain depends also on ϕ as well as on the polarization angle of the signal. Since an analysis of the polarization of the radiation falls outside the scope of this work as it would make the analysis considerably more complicated, we have ignored this dependence. This could be done since the antennas are mostly in a relatively narrow cone (full opening angle less than 60°) from the source. We have performed some checks of the phase stability of the signal from selected isolated and strong sources that resemble point sources that indicates that ignoring the ϕ dependence appears not to be a safe assumption.

Combining the $1/R_{as}$ drop of signal strength with the antenna gain proportionality to $\sin(\theta_e)$, we obtain an antenna weighting factor

$$W_a = \begin{cases} R_{as}^2/R_{rs}^2 & \text{if } R_{as}^2/R_{rs}^2 < 1.2 \\ 1.2 & \text{if } R_{as}^2/R_{rs}^2 \geq 1.2 \end{cases} . \quad (3)$$

where the weight is normalized to unity for the reference antenna at a distance of R_{rs} . In addition the weight is capped to a maximum of 1.2 for distant stations where the signal to noise ratio is getting worse.

8.4.3 Antenna calibration

The antenna timings are calibrated for each flash following the procedure outlined in [Scholten et al., 2021]. Per flash 20 – 30 bright stand-alone pulses are selected from the whole flash where care is taken that their source locations roughly cover the extent of the flash. For all these sources their location as well as the antenna timings are searched in a simultaneous fit using the source location search algorithm discussed in [Scholten et al., 2021].

Since for interferometry the stability of the phase of the signal over all antennas is important, we perform sometimes an additional check of this phase for a single distinct pulse in the spectrum selected for TRID imaging. If this phase is off by more than 90° the antenna will not be used. Usually this eliminates less than 1% of all antennas. This may also result in eliminating a whole station from the analysis for a particular flash.

The gains of the antennas are calibrated by normalizing the noise level to unity. We select for this normalization only the parts of the recorded trace for which there is no lightning activity detected. Since this noise level is largely due to extra terrestrial sources, i.e. radiation produced by the Milky Way, we name this the Galactic Background [GB] level.

8.4.4 Determining the position of maximal intensity

The voxelated coherent intensity is determined for subsequent time slices of 100 ns where it is straightforward to determine the voxel with maximal intensity. To reach an inter-voxel accuracy we have implemented two interpolation procedures, quadratic and barycentric interpolation.

Quadratic The intensity of the voxels around the voxel with the maximum intensity are fitted by a paraboloid,

$$I_p(\vec{x}) = I_0 + \sum_i \left[\frac{1}{2} A_i x_i^2 + B_i x_i \right] + \sum_{i,j} R_{i,j} x_i x_j , \quad (4)$$

where x_i is the i^{th} grid coordinate, $i = 1, 2, 3$. The coefficients I_0 , A_i , and $R_{i,j}$ are fitted to the grid points bordering the maximum. The inter-voxel maximum \vec{x}_p is taken at the point where the paraboloid reaches its maximum.

Barycentric The barycentric maximum is calculated as

$$\vec{x}_b = \sum_{\vec{x}} [(I(\vec{x}) - I_{th}) \vec{x}] / \sum_{\vec{x}} [(I(\vec{x}) - I_{th})] , \quad (5)$$

where the sum runs over all voxels with an intensity exceeding the threshold value I_{th} . This threshold is taken as $I_0/1.2$ where I_0 is the maximum intensity of voxels in the grid or the largest value of a voxel on the outer surface of the grid, whichever is larger.

Each of the two interpolation procedures has advantages and disadvantages. When the atmosphere is noisy, with many active sources in a small area, the barycentric interpolation will yield some weighted average position, while the quadratic interpolation yields the position of the strongest source. It is, however, observed that the details of intensity surface are complicated with small ripples (order 10% in intensity of distances of 20 m) probably remnants of side beams. For a coarse grid the quadratic interpolation will thus be unstable, but never give a result that is off by more than the grid spacing. Using barycentric interpolation these ripples are efficiently averaged yielding a properly interpolated maximum. The physics case shown in Section ?? uses a fine grid and for this reason the quadratic interpolation is used.

8.4.5 Source intensity and polarization

The coherent or interferometric intensity is calculated as the intensity of the signal from the source as received by the reference antenna and is expressed in units of [GB] (see Section 8.4.3). It has thus tacitly assumed that the azimuth-angle dependence of the antenna gains is limited and is effectively averaged-out. The elevation-angle dependence is to a large extent accounted for by the weighting factors (see Section 8.4.2) when calculating the coherent intensity.

All LOFAR antennas used in this work are inverted v-shape dipoles where X- (Y-) dipoles corresponding to odd (even) numbered antennas are oriented in the NE-SW (NW-SE) plane. Our analysis is performed separately for X- and Y-dipoles since they differ significantly in their sensitivity to polarized radiation. The antenna function (the Jones matrix, specifying for each dipole and polarization the gain depending on angle and frequency) has to be used [van Haarlem et al., 2013] to convert the measured intensity to the absolute intensity of the source. The absolute calibration of the LOFAR antennas is performed in [Mulrey et al., 2019]. This complicated task simplifies considerably if it is assumed that the source is point-like (flat frequency spectrum over our frequency range) and linearly polarized, however this falls outside the scope of the present work.

We observe that the received coherent power averaged over 0.3 ms for a flash located at the NE of the array is about twice as large in the Y-dipoles as in X where for a flash at the SE side of the core it is the other way around. This is due to the azimuth-angle dependence of the antenna gain. For sources in the NE horizontally polarized radiation is recorded almost exclusively in the Y-dipoles with a gain (in power) that is about a factor two larger than that for vertical polarization recorded exclusively by the X-dipoles in this configuration.

9 E-field Interferometric imaging

In this section the measured signal in the X- and Y- dipoles is converted to the radiation electric field at the antenna and to be used for interferometry. For future reference this will be named E-field Interferometry (EI).

9.1 Basics

To convert the measured signal the Jones matrix is used, J

$$\vec{E}_a(\hat{r}_{as}) = J^{-1}(\hat{r}_{as}) \vec{S}_a , \quad (6)$$

where subscript a refers to a particular antenna, $\vec{E}_a(\hat{r}_{as})$ is a 3 component vector giving the radiation electric field at the position of the antenna, $\vec{S}_a(\hat{r}_{as})$ is a two component vector where the two components are the measured signals in the X- and Y-dipoles. The arrival direction is specified by \hat{r}_{as} with $\vec{r}_{as} = \vec{r}_a - \vec{r}_s$ where \vec{r}_a points to the antenna and likewise \vec{r}_s to the source. The unit vector \hat{r}_{as} thus points from the source to the antenna. The radiation field obeys $\vec{E}_a(\hat{r}_{as}) \cdot \hat{r}_{as} = 0$. We also introduce the distance from the source to the antenna, $R_{as} = \sqrt{\vec{r}_{as} \cdot \vec{r}_{as}}$.

The electric field at the antenna is written in terms of \vec{I} , the source current moment as

$$\vec{E}_{as} = \frac{\vec{I} - (\vec{I} \cdot \hat{r}_{as}) \hat{r}_{as}}{R_{as}} , \quad (7)$$

which obeys, by construction, $\vec{E}_{as} \cdot \hat{r}_{as} = 0$ and properly falls-off with distance to the source.

To reconstruct \vec{I}_s from the fields determined at the various antenna positions, $\vec{E}_a(\hat{r}_{as})$ we minimize,

$$\chi_E^2 = \sum_a \left(\vec{E}_a(\hat{r}_{as}) - \vec{E}_{as} \right)^2 \vec{w}_a = \sum_{a,i} \left(E_{a,i} - \frac{I_i - (\vec{I} \cdot \hat{r}_{as}) \hat{r}_{as,i}}{R_{as}} \right)^2 w_{a,i} \quad (8)$$

$$= \sum_{a,i} \left[E_{a,i}^2 w_{a,i} - 2 E_{a,i} I_i w_{a,i} + 2 E_{a,i} E_{a,i} w_{a,i} \left(\vec{I} \cdot \hat{r}_{as} \right) + I_i w_{a,i} I_i - \left(\vec{I} \cdot \hat{r}_{as} \right)^2 \hat{r}_{as,i}^2 w_{a,i} \right] \quad (9)$$

with respect to the components I_i , $i = 1, 2, 3$. Note that a proper time translation is understood. Weights $w_{a,i}$ have been introduced here and should be taken equal to the inverse-square-error in $E_{a,i}$, but are taken equal to unity when searching for the maximum intensity location. Taking $w_{a,i} \neq w_a$ i.e. not i -independent gives serious problems as $\sum_i E_{a,i} \hat{r}_{as,i} w_{a,i} \neq 0$. Excluding i -dependent weights gives the simpler

$$\chi_E^2 = \sum_a w_a \left[\vec{E}_a \cdot \vec{E}_a - 2 \vec{E}_a \cdot \vec{I} / R_{as} + \vec{I} \cdot \vec{I} / R_{as}^2 - \left(\vec{I} \cdot \hat{r}_{as} \right)^2 / R_{as}^2 \right] , \quad (10)$$

Minimal χ_E^2 gives us the three conditions,

$$0 = \partial \chi_E^2 / \partial I_i = 2 \sum_a \left[-E_{a,i}(\hat{r}_{as}) + I_i / R_{as} - \hat{r}_{as,i} \left(\vec{I} \cdot \hat{r}_{as} \right) / R_{as} \right] w_a / R_{as} . \quad (11)$$

This equation we rewrite as,

$$A\vec{I} = \vec{F}, \quad (12)$$

with

$$F_i = \sum_a E_{a,i}(\hat{r}_{as}) w_a / R_{as}, \quad (13)$$

the coherent sum of the fields over all antennas, and

$$A_{ij} = \sum_a (\delta_{i,j} - \hat{r}_{as,i} \hat{r}_{as,j}) w_a / R_{as}^2, \quad (14)$$

which is positive definite and but not symmetric because of the weighting factors. The matrix can easily be inverted.

The current moment for a pixel can thus be written as

$$\vec{I} = A^{-1} \vec{F} = \sum_a A^{-1} \vec{E}_a(\hat{r}_{as}) w_a / R_{as}, \quad (15)$$

which is probably more efficient from a calculational point.

Minor issue: one eigenvalue of matrix A (for the case the weighting factors do not depend on i) tends to be very small, i.e. the inverse blows-up! Best way to deal with this situation is to realize that the array does not have enough sensitivity to reconstruct the current-moment in the direction of the small eigenvalue.

Thus we rewrite

$$A_{ij} = \sum_{i=1}^3 \vec{\varepsilon}_i \alpha_i \vec{\varepsilon}_i^T, \quad (16)$$

resulting in

$$\vec{I} = A^{-1} \vec{F} = \sum_{i=1,2} \vec{\varepsilon}_i \alpha_i^{-1} \sum_a \vec{\varepsilon}_i^T \vec{E}_a(\hat{r}_{as}) w_a / R_{as}, \quad (17)$$

where only the large eigenvalues are kept.

9.1.1 Polarization dependent weights

To reconstruct the time-dependent current \vec{I}_s from the (time dependent) fields determined at the various antenna positions, $\vec{E}_a(\hat{r}_{as})$ we minimize (where k runs over two orthogonal transverse polarizations \hat{r}_{ak} that are antenna and direction dependent with $\hat{r}_{ak} \cdot \hat{r}_{as} = 0$),

$$\chi_E^2 = \sum_{a,k,t} \left(E_{ak} - \vec{E}_{as} \cdot \hat{r}_{ak} \right)^2 w_{ak} = \sum_{a,k,t} \left(E_{ak} - (\vec{I} \cdot \hat{r}_{ak}) / R_{as} \right)^2 w_{ak} \quad (18)$$

$$= \sum_{a,k,t} w_{ak} \left[E_{ak}^2 - 2 E_{ak} (\vec{I} \cdot \hat{r}_{ak}) / R_{as} + (\vec{I} \cdot \hat{r}_{ak})^2 / R_{as}^2 \right], \quad (19)$$

with respect to the components I_i , $i = 1, 2, 3$ and where the model electric field is written as in Eq. (7). It should be noted that $\vec{E}_{ak} = E_{ak} \hat{r}_{ak}$ as well as \vec{E}_{as} , or equivalently \vec{I} , are time dependent in Eq. (19) where a time translation, proportional to signal travel, time is understood. Transverse-polarization

dependent weights w_{ak} have been introduced and should be taken equal to the inverse-square-error in $E_{a,k}$, but are taken equal to unity when searching for the maximum intensity location. Minimizing χ_E^2 gives us the three conditions,

$$0 = \frac{\partial \chi_E^2}{\partial I_i}|_t = 2 \sum_{a,k} w_{ak} \left[-E_{ak}^* \hat{r}_{ak,i} + \hat{r}_{ak,i} (\vec{I}^* \cdot \hat{r}_{ak}) / R_{as} \right] / R_{as}. \quad (20)$$

This equation we rewrite as,

$$A\vec{I} = \vec{F}, \quad (21)$$

with

$$F_i = \sum_{a,k} E_{ak} \hat{r}_{ak,i} w_{ak} / R_{as}, \quad (22)$$

the coherent sum of the fields over all antennas, and

$$A_{ij} = \sum_{a,k} (\hat{r}_{ak,i} \hat{r}_{ak,j}) w_{ak} / R_{as}^2, \quad (23)$$

which is positive definite and symmetric. While the currents and the electric fields are time-dependent, A_{ij} is not. The matrix can easily be inverted.

We need the Hessian matrix,

$$\frac{\partial^2 \chi_E^2}{\partial I_i \partial I_j} = 2A_{ij}, \quad (24)$$

for estimating errors in the current moment. The variance in \vec{I} is now given by the inverse of the Hessian. In particular we use $\sigma(I)_i = A_{i,i}^{-1} \times \chi^2 / \text{DoF}$. It is not clear as yet what the proper expression is for the error in the Stokes parameters

At the minimum of the chi-square where Eq. (21) applies the value is given by

$$\chi_E^2 = \sum_{a,k} \left(E_{ak} - \vec{E}_{as} \cdot \hat{r}_{ak} \right)^2 w_{ak} \quad (25)$$

$$= \sum_{a,k} w_{ak} \left[E_{ak}^2 - (E_{ak}^* (\vec{I} \cdot \hat{r}_{ak}) + E_{ak} (\vec{I}^* \cdot \hat{r}_{ak})) / R_{as} + (\vec{I} \cdot \hat{r}_{ak})^2 / R_{as}^2 \right] \quad (26)$$

$$\neq \left[\sum_{a,k} w_{ak} E_{ak}^2 \right] - \Re(\vec{F} \cdot \vec{I}^*) \quad (27)$$

$$= \sum_{a,k} \left[w_{ak} E_{ak}^2 - E_{ak} w_{ak} (\hat{r}_{ak} \cdot \vec{I}^*) / R_{as} \right] \quad (28)$$

$$= \sum_{a,k} \left[(\sqrt{w_{ak}} E_{ak})^2 - (\sqrt{w_{ak}} E_{ak}) (\hat{r}_{ak} \cdot \vec{I}^*) \sqrt{w_{ak}} / R_{as} \right] \quad (29)$$

$$\chi_E^2 = \sum_{a,k} \left[\sqrt{w_{ak}} E_{ak} - (\hat{r}_{ak} \cdot \vec{I}) \sqrt{w_{ak}} / R_{as} \right]^2, \quad (30)$$

which should be positive. Note that $(\vec{F} \cdot \vec{I}^*)$ has a vanishing Imaginary part and that the expression includes an implicit summation over time.

The correlation matrix (where the diagonal elements are the error) is the inverse of the Hessian. To obtain the error in the stokes parameters we still need to integrate over time. Not sure how to work in the complex part of the current moment or Stokes.

9.1.2 Relative X- & Y-dipole timing calibration

The basic idea for the relative timing calibration of a dipole pair is that the circular polarization of a signal is determined by the relative time-offset of the signals in the two polarization directions (take these t=zenith- or p=azimuth-angle) of the signal from a source. When averaging over many sources one expect the circular polarization to average to zero unless there is a systematic timing offset. The sign of the time-offset depends however on the in- or out-of phase oscillation of the t- and p-polarizations. This relative phase is dependent on the angle of the net linear polarization, 45° is in phase, -45° is out of phase. Depending on this angle the pulse needs a sign-change to add coherently regarding the time shift.

To implement this we calculate for a particular antenna pair the cross correlation for the t- and p-polarizations,

$$X_j(\tau) = U_j(\tau) + i V_j(\tau) = I_{j,t} \bigotimes I_{j,p}^*|_\tau , \quad (31)$$

for all calibration pulses j in this antenna where $U_j(\tau)$ and $V_j(\tau)$ are real, the asterisk denotes complex conjugation and \bigotimes the convolution of two traces. $\tau = 0$ corresponds to no additional delay and thus $U_j(0)$ and $V_j(0)$ are the usual U and V Stokes parameters measuring polarization at 45° and circular polarizations, respectively. Statistically one would expect U and V to have a random spread around zero for many pulses, even when there is a timing offset. As the next step we construct

$$X_s(\tau) = \sum_j X_j(\tau) Sng[U_j(0)]/I_j , \quad (32)$$

where the sum runs over all calibration pulses, $Sng[]$ denotes the sign, and I_j is the stokes I or pulse intensity. Due to the sign function the stokes U all add coherently. To avoid domination of one or two strong pulses, which would spoil the statistical average, the stokes parameters are normalized. Working with normalized stokes parameters has the additional advantage that pulses that are strongly (linearly) polarized in the t or p direction, and thus have no information on the relative time offset, give a negligible contribution. For a finite (relatively small) time offset, $V_s(0) \neq 0$ where the sign depends on which of the two polarizations is delayed. The delay, τ_0 , is taken where $U_s(\tau_0) = \max$ since this is numerically simplest and has been verified to be within 0.5 ns of a time where $U_s(\tau) = 0$.

Repeated application of this procedure shows convergence after the first step.

9.1.3 The Jones matrix

The angular dependence of the Jones matrix is parameterized as a sum over spherical harmonics to allow for a smooth and analytical interpolation at near-horizon angles,

$$J_{d,i}(\nu; \theta, \phi) = \sum_{j,m} A_{j,m}^{(i)}(\nu) Ch_m(\phi) Lgdr_j(\cos \theta) \quad (33)$$

where $d = (X, Y)$ denotes the dipole, $i = (t, p)$ the polarization of the electric field, ν the frequency, and $A_{j,m}^{(i)}(\nu)$ the functions that parameterize the Jones matrix. The sum runs over $j = (1, 3, 5, 7)$ for the Legendre polynomials ($Lgdr$) and $m = (1, 3, 5)$ Chebyshev polynomials. The X- and Y- dipoles are assumed to be identical, only rotated over 90°. The functions $A_{j,m}^{(i)}(\nu)$ are determined by (analytic) fitting tabulated values of the Jones matrix.

A Details of the LOFLI code

The code used for source finding and antenna calibration.

A.1 Data reading

Data read requires that the files have been pre-processed to determine the RFI-mitigation parameters for each antenna using "program RFI_mitigation". In this process of RFI mitigation the files "RFI_Filters.uft" and "LOFAR_H5files_Structure.dat" are produced which are required here as input.

A.1.1 Subroutine AntennaRead

All data files in "directory.out" are looped-over. Info on the structure of the data files and concerning RFI-mitigation parameters is read from the auxiliary files. Information on pole-flip and bad antennas is supplied. Files that contain data from antennas that have already been read are skipped. Antenna and station calibration in the data-files are combined with calibration tables that have been generated in earlier calibration runs (see Section ??). The calibrations delays are combined with the delay calculated from "SourceGuess", a guessed source position that ideally should correspond to the center of the lightning flash. This should guarantee that pulses are roughly lined-up. The appropriate chunk of data is read-in. The Chunk-length equals "Time_dim". If the number of zeros in this chunk is disproportionately large, (number-of-2/number-of-0 < 0.66) this Chunk is skipped. Otherwise the data are multiplied by a Hann window ("HW_size"=32) and the trace is fourier transformed, RFI-filtered, phase-shifted according to the expected delay (w.r.t. the LOFAR-core), and transformed back to the time domain. Spectra are stored in one big array for all antennas and the various time-periods (data chunks).

The spectra are normalized by dividing by the $\sqrt{(\text{power})}$ determined are the preparatory-stage, see Section 3.

A.2 Candidate pulse selection

Candidate pulses are selected in subroutine DualPeakFind. First pulses are selected separately for the even and odd polarized antennas. In the following phase the pulses that are close to each other will be labeled 'dual'.

In the Dual mode the first even/odd antenna pair at the same location are chosen as reference antennas.

A.2.1 Subroutine DualPeakFind

The Hilbert-envelope of the spectrum of the reference dipole for the even or the odd numbered antennas is searched for the highest value. This search is done for the samples more than "EdgeOffset" samples away from either beginning or the end of the chunk. The length of a chunk is equal to "Time_dim"=32768. "EdgeOffset"=7000 is chosen such that two sources that are separated by 5km will have their pulses not further than "EdgeOffset" samples apart,

$$\text{EdgeOffset} = 5 \text{ [km/c/sample]} = 5 \cdot 10^3 / (3 \cdot 10^8 \times 5 \cdot 10^{-9}) = 5 \cdot 10^3 / 1.5 = 7000 \text{ [samples]}. \quad (34)$$

Once the peak is found it is checked where the closest local minimum is to the left ("W1") and right ("Wu") of the peak. In addition the condition is imposed that this minimum is below a fraction (1/2) of the peak value. If within a distance of "W_low"=6 samples from the peak the spectrum has been zeroed (because previously a peak has been found) the part of the spectrum between "W1" and "Wu" is zeroed and the peak-position is not stored. Otherwise the peak is considered genuine and is stored and the region of the spectrum from "Separation" below till "Separation" above the found peak-position is zeroed. This process is repeated till "PeakS_dim" peaks have been found. "Separation"=20 is

used as zero padding in the calculation of the cross correlations.

Peaks are kept in order of their peak-value in the reference antenna, separately for even and odd numbered antennas.

In Dual mode, the obtained peaks are searched for those that occur in odd as well as even antennas. First even and odd peaks are sorted according to sample number and two peaks are considered to come from the same source when their distance (in samples) is less than the minimum of "W1" and "Wu" where "Wu" of the earlier peak and "W1" of the later peaks are used.

After scanning through the all pulses the pulses are ordered according to pulse strength.

- In peak-finding it turns out that too often the peaks with the largest amplitude do not pass the quality selection criteria. •

A.3 Cross correlation

The calculation of cross correlation is central to imaging.

A.3.1 Subroutine BuildCC

For all antennas and all active (=open) data-chunks the cross correlations are calculated in Subr. A.3.2 for all active sources. When calibrating, the number of active data-chunks and the of active sources can be any. When performing source-finding there is only a single active data-chunk and only one or two active sources (two for the case of double polarity fits). The building of the cross-correlation spectra is done first for the even antennas followed by the odd numbered ones in the next cycle. There is an option to write the phases of the cross correlations to file, set by "PlotCCPhase".

A.3.2 Subroutine GetCorrSingAnt

The cross-correlation spectra are calculated for all peaks for one particular antenna and one data-chunk. Note that each peak is assigned to either odd or even antenna numbers. (I know, ... this is a bit clumsy and wastes resources.) For this antenna the re-calibration corrections are retrieved as resulted from a previous calculation. The main calibration results (station and antenna delays) have already been included at the stage of reading-in the data, see Subr. A.1.1. For each antenna the arrival-time difference is calculated for the signal from the actual source location and the 'raw-source'-location. The latter has been used in Subr. A.1.1 to obtain a rough outline of the traces and this should roughly correspond to the center of the imaged area. In the first call of a cycle (separate cycles for even and odd antennas) the reference spectrum is stored of length "Tref_dim" [samples] symmetrically around the specified peak position in "PeakPos(i_Peak)" taking into account the reference-antenna-shift parameter for this peak.

This routine is also used for calculating relative timing between pulses in antennas from a single station which may be remote. In that case the pulse position on the first antenna may be rather different from that of the original reference antenna. In this case the logical "PulsPosCore" and the time-shift between the core-center and the first antenna is corrected for, i.e. it is considered that the peak position is given for an antenna at the core.

For a non-reference antenna a section of length "T2_dim"="Tref_dim"+2 "Safety" [samples] is taken, symmetrically around the calculated peak position, based on the given source location.

The cross correlation is calculated with the reference spectrum (padded with zeros), see Subr. A.3.3, such that the zero in the cross correlation time (difference) spectrum corresponds to the expected time delay with the reference antenna for the present source position. The peak position in the cross correlation spectrum is obtained in Subr. A.3.4, which is later minimized in the chi-square fitting routine by optimizing the source location. During fitting the cross correlation spectra are not recalculated. Depending on the logical "RealCorrelation" the real part or the Hilbert-envelope of

the cross correlation is used.

A.3.3 Subroutine CrossCorr

The cross correlation between two spectra is calculated by multiplying the Complex conjugate of the fourier transform (FFT) of the reference with the FFT of the other while applying a frequency-dependent phase-shift corresponding to the specified time-off-set between the two. Before the FFT a Hann-window with a fall-off width of "HW_size"=5 is applied to both time-spectra.

A.3.4 Subroutine ReImAtMax

The Hilbert transform of the cross correlation is multiplied by a parabola normalized to unity at the location where maximum in the correlation is expected, based on the previously determined source location. The zero crossings of the parabola are set `SearchRange*SearchRangeFallOff` samples out from the maximum. Default is `SearchRangeFallOff=4` but is an input parameter. `SearchRange` is calculated in subroutine 'SearchWin' as the quadratic sum of the expected timing error based on the covariance matrix and a fixed error `Sigma_AntT=2` samples, which is an input parameter.

The time of the maximum in the correlation function is calculated using a spline interpolation. This is done for the Hilbert envelope as well as for the real and imaginary parts of the cross correlation. In addition the real & imaginary value of the cross correlation is returned. Only the part of the cross correlation between $\pm \text{"Safety"}$ from zero is searched.

The default value for the error-bar is taken to be "`error`"=1 [ns]=0.2 [samples]=1 [ns] ($=e_j$ in Eq. (36)). If the max is at either end of the searched time-range, "`error`"=200 [samples], if further than "`Safety`"/4 from zero, "`error`"=20 [samples]. The error is like-wise increased when the maximum is more than twice the 'SearchWin' removed from the expected position. Large error implies small weight in the fit and in this case the antenna is counted as 'excluded'.

If the 'shape' of the cross correlation, defined as the ratio of the peak height with the integral (over the calculated range, differs more than a factor "`CCShapeCut`" from the same quantity for the self-correlation, the weight is decreased and the antenna is also counting as excluded.

A.4 Fitting

The fitting is performed using the routine "`NL2SOL`"[Dennis et al., 1981] that uses a Levenberg–Marquardt algorithm optimizing the value of χ^2 as defined in Eq. (36). Fit-parameters can be some or all coordinates of the peaks, possibly in combination with either antenna timings of selected stations or station timings of selected stations. An analytic expression for the Jacobian matrix is programmed. If "`Doble`" is set, identical source location are used for sources with the same "`PeakPos`" for even and odd antenna numbers. If the logical "`CalcHessian`" is set (done only for source finding) the covariance matrix is calculated for converged fits, defined as

$$\text{Cov} = \chi^2/nfd \times H^{-1}, \quad (35)$$

where nfd =number of degrees of freedom corrected for the number of free parameters, and H is the Hessian, the second derivative of the χ^2 w.r.t. the parameters. The square roots of the diagonal matrix elements as kept as $\sigma(i)$.

A.5 Source search

A search for source locations (for the candidate pulses found earlier) starts with a grid search involving the stations at or near the Superterp, followed by a chi-square search.

A.5.1 Subroutine SourceTryal

First RMS values are calculated for a 16 sources distributed a circle with diameter of 50 km. For the direction where the RMS has a minimum a finer grid is searched. For each search only those antennas are included in the calculation of the RMS where for the complete grid the calculated position of the pulse falls within the window for which the cross correlation is calculated. For this reason `FitRange_Samples` should be at least equal to 70.

Once an approximate location is found, this is fed into the chi-square fitting machine, starting with the small circle antennas around the Superterp.

A.5.2 Subroutine SourceFind

The source finding can be done using pulses in either even or odd numbered antennas. There is also an option to find sources that produce pulses on all antennas. The different options are selected through the range of the loop over "`i_eo`". Note that imaging for even and odd independently can be done simultaneously. Because of the zero-ing of the pulses these options are not compatible with finding sources for all antennas.

The search for the source position proceeds in steps of increasing distance to the reference station, starting using all antennas in a station. The searches are performed in distance steps of 0.5, 1.05, 2.5, 5, 10, 20, 30, and 50 [km] where the last distance includes all Dutch stations, using Subr. A.5.3.

During the chi-square fitting the co-variance matrix is calculated. For a following fitting round this is needed to calculate the guess for the arrival window. After the last run the found source location will be written to file when the source obeys certain rather loose quality conditions. In that case the peaks are zero-ed in all antennas, using Subr. A.5.4.

At this stage the quality conditions are: 1) distance to the core is less than `Dist_Lim`=100 km, and 2) the fraction of included antennas is greater than `EffAntNr_lim`=0.8, and 3) $\sigma(h)$ is less than 990 (with an even larger value for lower heights than 1 km, and 4) RMS² is less than `ChiSq_lim`, an input parameter.

A.5.3 Subroutine SourceFitCycle

The initial start search location is specified by "`SourceGuess`". For each following search the previously found location is used as first guess. For distance <0.5 [km] only x is fitted • **should be changed to azimuth angle** •, for <1.0 [km] only x & y, for <5 [km] (x, y, z), and otherwise all, including a timing offset. For the largest distance also the Hessian is calculated. The source location is obtained by minimizing

$$\chi^2 = \sum_j \left(\frac{\delta t_j^o - \delta t_j^s}{e_j} \right)^2 , \quad (36)$$

where t_j^o is the pulse-arrival-time difference as determined from the cross correlation (see Subr. A.3.1), t_j^s is the pulse-arrival-time difference as calculated from the source location (see Subr. A.2.1), and e_j is the assumed error in the pulse-arrival-time difference with a nominal value of 1 [ns] (see Subr. A.3.2). The minimization routine (`nl2sol`) uses the Levenberg–Marquardt algorithm (see Section A.4).

A.5.4 Subroutine CleanPeak

In all spectra a section of length "`Tref_dim`" [samples], which is considered as the pulse-length, is set to zero, using a Hann-window with a fall-off width of "`HW_size`"=5 [samples]. The window is centered at the peak-position calculated from the (just found) source location.

A.5.5 Subroutine FindStatCall

Station and antenna calibration-data are distinguished. It is verified that the mean delay of

all antennas in a station equals zero. Two types of station calibrations are used, main ones specified file="StationCalibrations.dat" in the main directory containing station-delays as given by ASTRON, generated using "CalibrationTablesGenerator.f90". Secondary station and antenna calibrations are created by previous runs of Subr. A.5.5 and stored in a file that is quoted in the output and can be specified in the input with the parameter "Calibrations".

It is recommended to zero the individual antenna delays in the Calibrations data file. This is the case for the calibration file with name ending in ".ZERO"

The relation between station numbers and station names is:

```

11 !      2      3      4      5      6      7      11     13     17     21     26     30     32
12 !    CS002  CS003  CS004  CS005  CS006  CS007  CS011  CS013  CS017  CS021  CS026  CS030  CS032
13 !    101    103    106    121    125    128    141
14 !    CS101  CS103  RS106  CS201  RS205  RS208  CS301
15 !    142    145    146    147    150    161    166    167    169    181    183    188    189
16 !    CS302  RS305  RS306  RS307  RS310  CS401  RS406  RS407  RS409  CS501  RS503  RS508  RS509

```

The calculated station delays are written as a table in the output file and also to file where the filename is given.

B Kalman filter

These notes build on the Master thesis of Alex Pel, in particular eqs. (4.37–42).

Rewriting the equations, introducing the column state vector for antenna a after k iterations where this iteration includes the measured pulse arrival time for this antenna

$$x_k^a = (t_k^s + R_{k,a}/v, \vec{x}_k^s)^T, \quad (37)$$

where t_k^s and \vec{x}_k^s are the time and the position of the source emitting the pulse and $R_{k,a} = |\vec{a} - \vec{x}_k^s|$ where \vec{a} denotes the position of the antenna. v is the propagation velocity of radio waves. Iteration $(k-1)$ includes all data up to those of antenna $(a-1)$ and not that of antenna a .

We thus have now

$$x_{k-1}^a = (t_{k-1}^s + R_{(k-1),a}/v, \vec{x}_{k-1}^s)^T \quad (38)$$

as the prediction for the state vector for antenna a using only the measurements of antennas $1 \cdots a-1$.

We need to introduce the covariance matrix C (called P by Alex Pel). The predicted covariance matrix C_{k-1}^a for the state vector x_{k-1}^a is related to the covariance matrix C_{k-1}^{a-1} as

$$C_{k-1}^a = J_{k-1} C_{k-1}^{a-1} (J_{k-1})^T \quad (39)$$

where

$$J_{k-1} = \begin{bmatrix} 1 & \vec{d}_{k-1} \\ 0 & \mathbb{1} \end{bmatrix} \quad (40)$$

is the propagator for the error matrix from $(a-1)$ to a with

$$\vec{d}_{k-1} = (1/v) \frac{\partial(R_{k-1,a} - R_{k-1,a-1})}{\partial \vec{x}_{k-1}^s}. \quad (41)$$

It is useful to introduce the time-projection operator

$$H = (1, 0, 0, 0). \quad (42)$$

Denoting the measured pulse arrival time in antenna a as t_m^a with error σ , the equations for the Kalman filter can be written as

$$x_k^a = x_{k-1}^a + K^a (t_m^a - H x_{k-1}^a) \quad (43)$$

$$K^a = C_{k-1}^a H^T (H C_{k-1}^a H^T + \sigma^2) \quad (44)$$

$$C_k^a = (I - K^a H) C_{k-1}^a. \quad (45)$$

In the program we use for the calculation for antenna a , iteration k using the jacobian Eq. (40) with

$$\text{Der}(i) = (1/v) \frac{\vec{x}_{k-1} - \vec{a}}{D_{k-1,a}} - \frac{\vec{x}_{k-1} - (\vec{a} - \vec{1})}{D_{k-1,a-1}} \quad (46)$$

$$\text{Cov}'(i, j, k) = \text{Jac}(i, m) \text{Cov}(m, n, k-1) \text{Jac}(j, n) \quad (47)$$

$$\text{Kal}(i, k) = \text{Cov}'(i, 0, k) (\text{Cov}'(0, 0, k) + 1/\sigma)^{-1} \quad (48)$$

$$\text{Cov}(i, j, k) = \text{Cov}'(i, j, k) - \text{Kal}(i, k) \text{Cov}'(0, j, k) \quad (49)$$

$$x(0, k) = t^s(k) + R_{k,a}/v = (1 - \text{Kal}(0, k)) (t^s(k-1) + R_{k-1,a}/v) + \text{Kal}(0, k) t_m^a \quad (50)$$

$$\vec{x}(k) = \vec{x}^s(k) = x^s(i, k-1) + \text{Kal}(i, k) (t_m^a - (t^s(k-1) + R_{k-1,a}/v)), \quad (51)$$

where R is the difference in distance from the source to the antenna ($=D$) and from the source to the core. t_s is thus the arrival time of the pulse at the core. The equation for the source time can be rewritten to

$$t^s(k) = t^s(k-1) + (R_{k-1,a} - R_{k,a})/v + \text{Kal}(0, k) [t_m^a - (t^s(k-1) + R_{k-1,a}/v)]. \quad (52)$$

Note, the way the correlation spectrum is calculated, the peak position is already the shift from the expected value, i.e. the quantity in square brackets.

B.1 For sources

B.1.1 Notation

We will denote the source vector after iteration k as

$$\vec{S}_k = [t_k^c, \vec{x}_k^s]^T, \quad (53)$$

where t^c is the arrival time of the pulse at the core (=reference antenna) located at \vec{c} . Iteration $(k-1)$ includes all data up to those of antenna $(a-1)$ and not that of antenna a .

B.1.2 Linear

Assume the observation, z , is a linear function of the state vector \vec{S}_k then the prediction based on previous measurements is

$$z_{k-1}^a = \mathbf{F}_a^T \vec{S}_{k-1} \quad (54)$$

while the measurement gives z_m^a which deviates from the true value is z_t^a statistically where the true value is derived from the true state vector \vec{S}_t . The new state vector is now written as

$$\vec{S}_k = \mathbf{A}_k \vec{S}_{k-1} + \mathbf{K}_k z_m^a, \quad (55)$$

introducing the Kalman gain \mathbf{K} . By imposing that the expectation value of the new and predicted state vector, $E(\vec{S}_k - \vec{S}_t)$, vanishes (see Alex chapter 4) we derive that

$$\mathbf{A}_k = \mathbb{1} - \mathbf{K}_k \mathbf{F}_a^T . \quad (56)$$

Requiring in addition that the optimum value is obtained for the covariance $\mathbf{C}_k = E((\vec{S}_k - \vec{S}_t)(\vec{S}_k - \vec{S}_t)^T)$ by setting the derivative of the diagonal to the Kalman gain to zero, we get

$$\mathbf{K}_k = \mathbf{C}_{k-1} \mathbf{F}_a (\mathbf{F}_a^T \mathbf{C}_{k-1} \mathbf{F}_a + \sigma_m^2) . \quad (57)$$

In addition we have

$$\mathbf{C}_k = (\mathbb{1} - \mathbf{K}_k \mathbf{F}_a^T) \mathbf{C}_{k-1} . \quad (58)$$

B.1.3 Realistic, non linear

The arrival time of the pulse in antenna at position \vec{a} is

$$t_k^a = R d_{k,a} / v - t_k^c = f(\vec{S}_k) , \quad (59)$$

where $R d_{k,a} = |\vec{a} - \vec{x}_k^s| - |\vec{c} - \vec{x}_k^s|$ where \vec{a} denotes the position of the antenna and \vec{c} that of the reference antenna. We need to linearize the prediction for the pulse arrival in antenna a given all previous $(k-1)$ measurements. The true source position is \vec{S}_t and the error in iteration $k-1$ is $\vec{\epsilon}_{k-1} = \vec{S}_t \vec{S}_{k-1}$. Keeping linearity in this error we write

$$t_{k-1}^a = f_{k-1}^a(\vec{S}_{k-1}) + \mathbf{F}_{k-1,a}^T \vec{\epsilon}_{k-1} . \quad (60)$$

with the Jacobian

$$\mathbf{F}_{k-1,a}^T = \left[-1, (1/v) \frac{\partial R d_{k-1,a}}{\partial \vec{x}_{k-1}^s} \right] . \quad (61)$$

The Kalman filter now reads

$$\sigma_{k-1}^2 = \mathbf{F}_{k-1,a}^T \mathbf{C}_{k-1} \mathbf{F}_{k-1,a} \quad (62)$$

$$\mathbf{K}_k = \mathbf{C}_{k-1} \mathbf{F}_{k-1,a} (\sigma_{k-1}^2 + \sigma_m^2)^{-1} \quad (63)$$

$$\vec{S}_k = \vec{S}_{k-1} + \mathbf{K}_k (t_m^a - f_{k-1}^a(\vec{S}_{k-1})) \quad (64)$$

$$\mathbf{C}_k = (\mathbb{1} - \mathbf{K}_k \mathbf{F}_{k-1,a}^T) \mathbf{C}_{k-1} , \quad (65)$$

with σ_{k-1}^2 the estimated error in the arrival time of the pulse in antenna a based on previous data while the measurement is denoted with subscript m . Redefine the Kalman weight to obtain

$$\mathbf{K}_k = \mathbf{C}_{k-1} \mathbf{F}_{k-1,a} \quad (66)$$

$$\sigma_k^2 = \mathbf{F}_{k-1,a}^T \mathbf{K}_k + \sigma_m^2 \quad (67)$$

$$\vec{S}_k = \vec{S}_{k-1} + \mathbf{K}_k (t_m^a - f_{k-1}^a(\vec{S}_{k-1})) / \sigma_k^2 \quad (68)$$

$$\mathbf{C}_k = \mathbf{C}_{k-1} - \mathbf{K}_k \mathbf{K}_k^T / \sigma_k^2 . \quad (69)$$

C Tables and such

Namelist input for LOFLI. Note that none of the lines giving an array of values in the namelist that continues on the following line should have a comma at the end, for the other input lines this is optional. Several keywords may appear on a single line if separated by a comma. Anything after an exclamation mark (within the namelist) is treated as a comment. A complete list of namelist keywords is given in Table 4.

A typical example (however not for LOFLI!) of an input is as follows.

```

1  &ShPars IntegrateCurrent=-0.01 , nu_min=0 , nu_max=8000000
2    nu_min=30 , nu_max=80
3  Intensity_Weight=.false.
4  SAMPLINGTIME=5      ! in [ns]
5  StParRange = -11    ! in down-sampled sample times for calculation of Stokes parameters
6  F_lim=1.
7  JOT = 14.77, zen_B=22.19 , azi_B=-90.  ! direction magnetic field at LOFAR (49.5 mu T)
8  ! JOT = 16.8, zen_B=27 , azi_B=-90.  ! direction magnetic field at GRAND (56 mu T)
9  x_0=200, X_max=690. , GROUNDELEVEL= 0.0
10 energy_sh=2.e7, zen_sh=20
11   &end
12
13 0. 0.  ! shift_x [m], shift_y [m], alpha_vB [deg]
14 step
15 8.0 51.4198405287 103.495733281
16 5.0 35.0 -90.0
17 3.0 21.2132034356 -45.0
18
19 5, 7, 11, 12 0 0
20 data\coreas_456_set1_37
21
22 grid 25
23 dist 50
24 dist 75
25 dist 250
26 theta 90.
27 !-----
```

References

- [Briggs et al., 1999] Briggs, D. S., Schwab, F. R., and Sramek, R. A. (1999). Imaging. In Taylor, G. B., Carilli, C. L., and Perley, R. A., editors, *Synthesis Imaging in Radio Astronomy II*, volume 180 of *Astronomical Society of the Pacific Conference Series*, page 127.
- [Dennis et al., 1981] Dennis, J., Gay, D., and Welsch, R. (1981). Algorithm 573: An adaptive nonlinear least-squares algorithm. *ACM Trans. Math. Softw.*, 7(3):367–383.
- [Mulrey et al., 2019] Mulrey, K., Bonardi, A., Buitink, S., Corstanje, A., Falcke, H., Hare, B., Hoerandel, J., Huege, T., Mitra, P., Nelles, A., Rachen, J., Rossetto, L., Schellart, P., Scholten, O., ter

- Veen, S., Thoudam, S., Trinh, T., and Winchen, T. (2019). Calibration of the lofar low-band antennas using the galaxy and a model of the signal chain. *Astroparticle Physics*, 111:1 – 11.
- [Pugmire et al., 2015] Pugmire, C., Mundt, S. M., LaBella, V. P., and Struyf, J. (2015). Graphics layout engine gle 4.2.5 user manual. https://en.wikipedia.org/wiki/Graphics_Layout_Engine.
- [Scholten et al., 2021] Scholten, O., Hare, B. M., Dwyer, J., Sterpka, C., Kolmasova, I., Santolik, O., Lan, R., Uhlir, L., Buitink, S., Corstanje, A., Falcke, H., Huege, T., Hoerandel, J. R., Krampah, G. K., Mitra, P., Mulrey, K., Nelles, A., Pandya, H., Pel, A., Rachen, J. P., Trinh, T. N. G., Veen, S. t., Thoudam, S., and Winchen, T. (2021). The initial stage of cloud lightning imaged in high-resolution. *Journal of Geophysical Research: Atmospheres*, 126(4):e2020JD033126. e2020JD033126 2020JD033126.
- [Swarztrauber, 1982] Swarztrauber, P. (1982). Fortran subroutines for the fast fourier transform. <https://en.wikipedia.org/wiki/FFTPACK>.
- [van Haarlem et al., 2013] van Haarlem, M. P. et al. (2013). LOFAR: The LOw-Frequency ARray. *A&A*, 556:A2.
- [Yatawatta, 2014] Yatawatta, S. (2014). Adaptive weighting in radio interferometric imaging. *Monthly Notices of the Royal Astronomical Society*, 444(1):790–796.

Table 4: Possible `ParameterNames` in the `&Parameters` namelist of LOFLI (closed off with `&end`) and their default values.

Parameter	Default	Description
options		
<code>CurtainPlot</code>	<code>= .false.</code>	Produce a curtain plot where the pulses are aligned for a source at <code>sourceguess</code> , see Section ??.
<code>ImagingRun</code>	<code>= .false.</code>	Search for all sources, see Section 6.
<code>Explore</code>	<code>= .false.</code>	Explore the structure of the flash, see Section 4.
<code>Dual</code>	<code>= .true.</code>	Image odd and even polarization simultaneously, see and Section 6.
<code>FitIncremental</code>	<code>= .true.</code>	When calibrating this may be on or off, see .
<code>RealCorrelation</code>	<code>= .false.</code>	At some point it would be nice to use the real part of the correlation function instead of its Hilbert envelope, see Subr. A.3.4.
Calibration		
<code>Calibrations</code>	<code>= —</code>	Specify the name of the user-made calibration file, see Section 5.2.
<code>Fit_AntOffset</code>	<code>= .false.</code>	When set the delays for the individual antennas are searched for.
<code>WriteCalib</code>	<code>= .false.</code>	Write the user-made calibration data to file.
Antennas		
<code>SignFlp_SAI</code>	<code>= —</code>	Single Antenna ID's of those where the polarity is wrong, see Section 4.
<code>PolFlp_SAI</code>	<code>= —</code>	Single Antenna ID's of those where even-odd should be interchanged.
<code>BadAnt_SAI</code>	<code>= —</code>	Single Antenna ID's of those that should be excluded from further processing.
<code>ExcludedStat</code>	<code>= —</code>	Names of complete stations that should be omitted from the analysis.
Imaging Control		
<code>ChunkNr_dim</code>	<code>= 1</code>	Number of chunks to be used for the calibration run, see .
<code>PeakNr_dim</code>	<code>= —</code>	maximum number of pulses in the reference antennas that can be taken into account. Make sure that this is not too low, see Section 5.1
<code>FitRange_Samples</code>	<code>= —</code>	total number of channels for which the cross correlations are calculated. The values for 'Safety', see SubRefGCSA is equated to this.
<code>AntennaRange</code>	<code>= 100</code>	Maximal distance, in [km], from the core for which the antennas are included in the analysis, see Section 5.2.
<code>CCShapeCut_lim</code>	<code>= 0.6</code>	Maximal value for the shape parameter, see Subr. A.3.4.
<code>ChiSq_lim</code>	<code>= 50</code>	Maximal value for the RMS ² for sources to be written to file, see SubRefSF.
<code>EffAntNr_lim</code>	<code>= .8</code>	minimal fraction of antennas to be retained during source finding, see SubRefSF.
<code>Sigma_AntT</code>	<code>= 2</code>	Intrinsic size for the search window (in [samples]) in addition to what is obtained from the covariance matrix, see Subr. A.3.4
<code>SearchRangeFallOff</code>	<code>= 4.</code>	Used in the calculation of the search window, see RIAM.
Other		
<code>Diagnostics</code>	<code>= .false.</code>	output control
<code>FullAntFitPrn</code>	<code>= .false.</code>	output control
<code>OutFileLabel</code>	<code>= " "</code>	possible special label for the output files for this run.