



**Warning and Mitigation Technologies for Travelling
Ionospheric Disturbances Effects**

TechTIDE

User Manual

HF-Interferometry method

Grant agreement no: 776011

The TechTIDE has received funding from the European Commission Horizon 2020
research and innovation programme (2017 – 2020)



Table of Contents

Document Information.....	4
1. Some information about the HF-Interferometry method	5
2. Installation of the code	6
3. Contents of the compressed file	6
4. Compilation of the code.....	8
5. Executing the code	9
5.1. Version for Near Real Time operation.....	9
5.2. Version for User Defined Time Intervals	10
6. Description of the input files.....	11
6.1. stations.txt	11
6.2. STA_YYYYMMDDHHmm.txt.....	12
7. Description of the output files	14
7.1. STA_YYYYMMDDHHmmR.dat file.....	14
7.2. TID_HFI_YYYYMMDDHHmm.log file.....	15

Index of Tables

Table 1. List of files to run the HF interferometry method with a short description.....	6
Table 2. List of output files generated by the HF interferometry method with a short description of its respective content.	14



Index of Figures

Figure 1. Geographical distribution of the HF sensors in Europe (left) and South Africa (right).	5
Figure 2. Sample of a screen shot showing the setting time of a query for a given run of the main program.....	10
Figure 3. Sample of a screen shot showing the setting time of a query for a given run of the main program.....	10
Figure 4. Information provided by default in the input file <i>stations.txt</i>	12
Figure 5. Example of a portion of a given intermediate input file <i>STA_YYMMDDHHMM.txt</i> which refers to the measurements of the ionosonde in Pruhonice, PQ052, for the 24-h time interval ending at 04:30:30 UT of 16 of July of 2017. Note that measurements were recorded with a sampling time of 15 minutes.....	13
Figure 6. Example of a portion of a given intermediate output file <i>STA_YYMMDDHHmmR.dat</i> which refers to the measurements of the ionosonde in Pruhonice, PQ052, for the 6-h time interval ending at 04:30:30 UT of 16 of July of 2017.....	15
Figure 7. Example of an output file <i>TID_HFI_YYMMDDHHmm.log</i> which refers to the observations for the 6-h time interval ending at 04:30:30 UT of 16 of July of 2017.....	17



Document Information

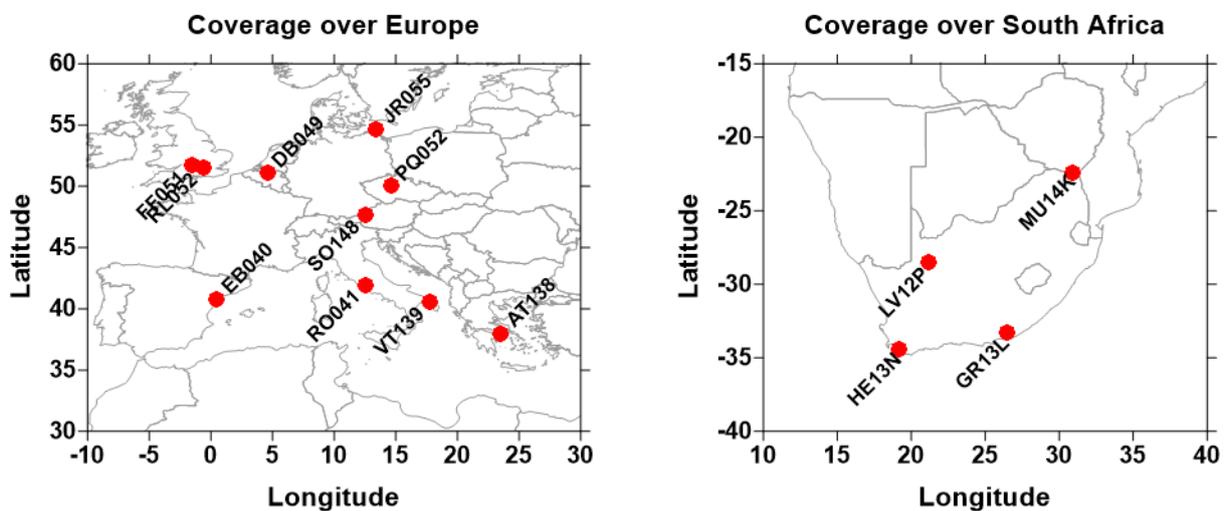
Title:	HF-Interferometry User Manual
Date of Publication:	22 January 2019
Author(s):	Antoni Segarra, Estefania Blanch, David Altadill
Work Package no.:	2
Work Package title:	TID Identification Methodologies
Work Package leader:	OE
Dissemination level:	Public, Open Access, distributed under the terms of Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0)

1. Some information about the HF-Interferometry method

HF Interferometry is a method to identify LSTIDs for the monostatic measurements of a given network of HF sensors (Ionosondes). The spatial distribution of the network should be dense enough to detect LSTID; i.e, distance between measuring sites no larger than 1000 km. The method detects quasi-periodic oscillations of ionospheric characteristics, identifies coherent oscillation activity at different measuring sites of the network and sets bounds to time intervals for which such activity occurs into a given region. The disturbance potentially associated to TID in the last 6-h interval will be related to the de-trended ionospheric characteristics after removing the main daily harmonics. The dominant period of oscillation and amplitude of the LSTID are obtained by spectral analysis. This allows for identification of TID activity from Digisonde Networks. The vector velocity of propagation is estimated by the measured time delays of the disturbance of a given ionospheric characteristic at different sensor sites and assuming a plane wave propagation. Due to the geographical distribution of Digisonde sites within Europe and South Africa (some of the sensors are separated by about 1000 km from one to the other), only identification of LSTIDs is feasible in principle, which are associated with auroral and geomagnetic activity, directly related to Space Weather.

The HF interferometry method uses the Maximum Usable Frequency (MUF) obtained from 10 European Digisondes and 4 Digisondes from South Africa (see Figure 1). It uses near real time data from the GIRO DIDBase Fast Chars (<http://giro.uml.edu/didbase/scaled.php>).

Figure 1. Geographical distribution of the HF sensors in Europe (left) and South Africa (right).



References

Altadill, David; Belehaki, Anna; Blanch, Estefania; Borries, Claudia; Buresova, Dalia; Galkin, Ivan; Haralambous, Haris; Juan Zornoza, José Miguel; Kutiev, Ivan; Oikonomou, Christina; Sanz Subirana, Jaume; Tzagouri, Ioanna. Report on TID algorithms. TechTIDE project deliverable D2.2; doi 10.5281/zenodo.2510350.

2. Installation of the code

The following steps should be executed for installing the code:

1. This code is designed to work under a Linux platform.
2. Create a folder where the code will be installed (hereafter, “main folder”).
3. De-compress the downloaded tar file in the main folder.
4. All executables and input file *stations.txt* must be in the main folder.

3. Contents of the compressed file

The tar file contains the files listed in Table 1.

Table 1. List of files to run the HF interferometry method with a short description.

File	Folder	Type	Short description
main_HFI_NRT_TechTIDE.sh	Main folder	Executable	Main file for near real-time process. Bash file that download data, create folders and call the programs.
main_HFI_INT_TechTIDE.sh	Main folder	Executable	Version of main_HFI_NRT_TechTIDE.sh to be used for a user defined time interval
stations.txt	Main folder	Input file	List of ionosonde stations involved in the process.
data_size.out	Main folder	Executable	Check the number of data for every initial raw data file
data_size.for	Main folder	Source code	FORTRAN source code of data_size.out
calculateR.out	Main folder	Executable	Remove possible spikes in the raw data files and calculate the residual values
calculateR.for	Main folder	Source code	FORTRAN source code of calculateR.out
JD.for	Main folder	Source code	FORTRAN source code to convert a date time into Julian Date. It is needed for calculateR.out and data_size.out
DATE.for	Main folder	Source code	FORTRAN source code to convert a Julian Date into a date

File	Folder	Type	Short description
			time. It is needed for calculateR.out and data_size.out
LinFit.for	Main folder	Source code	FORTRAN source code to de-trend raw data. It is needed for calculateR.out
period.out	Main folder	Executable	Calculate period, amplitude, power and confidence level
period.for	Main folder	Source code	FORTRAN source code of period.out
CPGASTT.for	Main folder	Source code	FORTRAN source code needed for period.out
BuscaMaxP.for	Main folder	Source code	FORTRAN source code to estimate the dominant periodic components of time series. It is needed for period.out
ConEsPor.for	Main folder	Source code	FORTRAN source code to calculate the spectral energy contribution of the dominant periodic components of the time series. It is needed for period.out
Ordena.for	Main folder	Source code	FORTRAN source code to find the dominant periodic components of time series. It is needed for period.out
checkp.out	Main folder	Executable	Finds the dominant period at the different stations.
checkp.for	Main folder	Source code	FORTRAN source code of checkp.out
correl.out	Main folder	Executable	Calculate maximum correlation between time series, time delays, velocity and azimuth of LSTID
correl.for	Main folder	Source code	FORTRAN source code of correl.out
warning_level.out	Main folder	Executable	Provides final output file (*.log) with result of LSTID and establish a warning level.

File	Folder	Type	Short description
warning_level.for	Main folder	Source code	FORTRAN source code of warning_level.out
STA_YYYYMMDDHHmm.txt	YYYYMMDD	Intermediate input data file	Input data: 24-h raw data file downloaded from the GIRO DIDBase Fast Chars
STA_YYYYMMDDHHmmR.dat	YYYYMMDD	Intermediate output data file	Intermediate product: 6-h of background and de-trended (residual) values obtained with calulateR.out
TID_HFI_YYYYMMDDHHmm.log	YYYYMMDD	Output data file	Final product file which contains the final results

The input file *stations.txt* provides information of the ionospheric sensors whose data will be used by the method. The bash file *main_HFI_NRT_TechTIDE.sh* is the main executable to run the HF-Interferometry code in near real time. Finally, the bash file *main_HFI_INT_TechTIDE.sh* is the main executable to run the HF-Interferometry code for a particular period of time defined by the user. Files highlighted in light orange in Table 1 can be modified by the users to adapt to their needs. However, some restrictions apply as noted in the following sections.

4. Compilation of the code

If the provided BIN-file for Linux system does not work, you can re-compile the source codes using the following commands:

```
gfortran data_size.for DATE.for JD.for -o data_size.out
```

```
gfortran calculateR.for DATE.for JD.for LinFit.for -o calculateR.out
```

```
gfortran period.for CPGASTT.for BuscaMaxP.for ConEsPor.for Ordena.for -o period.out
```

```
gfortran checkp.for -o checkp.out
```

```
gfortran correl.for -o correl.out -llapack -lblas
```

```
gfortran warning_level.for -o warning_level.out
```

You may have to install *gfortran* and *tcsh* in your linux machine. You can install it using the following commands: “*sudo apt-get install gfortran*” and “*sudo apt-get install tcsh*”.

Note that for re-compile *correl.for* you need the libraries *-llapack* and *-lblas*. You can install it using the command “*sudo apt-get install libblas-dev liblapack-dev*”.



5. Executing the code

The main program that controls the whole method is a Linux bash file written in *tcsh* language. This program downloads data, creates folders, call the different programs, stores the results, check results and write the final result.

We provide two versions of the bash file: one is designed to work in near-real time, and the other is designed to work for user defined time intervals.

The main program of each version reads the measuring sites from the *stations.txt* input file and downloads the corresponding raw data files from the GIRO DIDBase Fast Chars (<http://giro.uml.edu/didbase/scaled.php>). These files are stored in a specific folder that identifies the day of the measurements as described in the following sections.

The code we provide is applied to two different networks (European and South African).

5.1. Version for Near Real Time operation

The code is executed by running the following command:

```
./main_HFI_NRT_TechTIDE.sh
```

Every time you run the code, the *main_HFI_NRT_TechTIDE.sh* reads the date and time of the computer machine using the Linux program *date* and it creates, if it doesn't exist, a folder named as YYYYMMDD. The Linux program *date* converts your computer time to Universal Time. YYYYMMDD is an 8 characters' string that identifies the day of the measurements, where YYYY, MM, and DD mean year, month and day respectively. For every run, the main program downloads one single file for each of the measuring sensors listed in the input file *stations.txt*. Each of these files are named as follows *STA_YYYYMMDDHHmm.txt* and it contains the raw data of the 24-h interval just before the query time. STA is a 5 characters' string that identifies the measuring sensor by its URSI code and YYYYMMDDHHmm is a 12 characters' string that identifies the time of the query, where YYYY, MM, DD, HH and mm mean year, month, day, hour and minute respectively. Once the data has been downloaded from the DIDBase, the bash file calls the rest of the programs and obtains the intermediate and final results which are also stored in the aforementioned folder.

To run the method systematically and automatically, it is necessary to include the above command in a task scheduler, i.e. *crontab* file, and the user can specify when to run the main program and with what cadence. We have scheduled the code to run with a cadence of 5 minutes. In order to make sure that the "real-time" data of all the measuring sites has arrived and are available at the GIRO DIDBase, we add 4 minute delay to the starting running time respect to the query date. For example, to process data query at 00:00 UT, we schedule the main program *main_HFI_NRT_TechTIDE.sh* to start at minute 00:04 UT. That is why we ask for date "-4 minutes" in the *\$timestamp* variable of the main program (see Figure 2) in order to set the query time. The user must update the main program to change this delay. In addition, we wait for 30 seconds extra time to launch the main program (second line in *main_HFI_NRT_TechTIDE.sh*: `sleep 30`).

Figure 2. Sample of a screen shot showing the setting time of a query for a given run of the main program.

```
#!/usr/bin/tcsh
sleep 30

#Taking date and time from the machine
#variables must have 2 digits, i.e. 00,05,... for be introduced in the website

#Get the unique timestamp to produce everything from:
set timestamp=`date -ud "-4 minutes" +%s`
#Produce today's data-time parameters:
set year=`date -ud @$timestamp +%Y`
set month=`date -ud @$timestamp +%m`
set day=`date -ud @$timestamp +%d`
set hour=`date -ud @$timestamp +%H`
set minute=`date -ud @$timestamp +%M`
#Produce yesterday's date parameters:
set timestamp_1=`date -ud "$year$month$day -1 day" +%s`
set year_1=`date -ud @$timestamp_1 +%Y`
set month_1=`date -ud @$timestamp_1 +%m`
set day_1=`date -ud @$timestamp_1 +%d`

#creating folder
mkdir -p $year$month$day
cp stations.txt $year$month$day
cd $year$month$day

#selecting the network to work with, 1 means Europe, 2 means South Africa
#we start with Europe
set network = 1
#the while-structure allows us to change the network
while ( $network < 3 )
if ( $network == 1 ) then
set stas = ( "AT138" "DB049" "EB040" "FF051" "JR055" "PQ052" "RL052" "R0041" "S0148" "VT139" )
```

5.2. Version for User Defined Time Intervals

This version is designed to apply the method for a user defined time interval and make retrospective analyses. The code is executed by running the following command:

`./main_HFI_INT_TechTIDE.sh "initial date-time" "final date-time"`

The main program of this version ask to the user for the date and time boundaries of the time interval to be analyzed and reads the query time in the command line (Figure 3). Date-time has to be written in the format YYYY-MM-DD HH:mm.

Figure 3. Sample of a screen shot showing the setting time of a query for a given run of the main program.

```
asegarr@DESKTOP-ASEGARRA:/mnt/c/test/soft_TIDE$ ./main_HFI_INT_TechTIDE.sh "2017-07-16 00:00" "2017-07-16 23:59"
```

Similarly to the Near Real Time version, the version for User Defined Time Intervals creates, if it doesn't exist, a folder named as YYYYMMDD. YYYYMMDD is an 8 characters' string that

identifies the day of the measurements, where YYYY, MM, and DD mean year, month and day respectively. For every run, the main program downloads one single file for each of the measuring sensors listed in the input file *stations.txt*. Each of these files are named as follows *STA_YYYYMMDDHHmm.txt* and it contains the raw data of the 24-h interval just before the query time. STA is a 5 characters' string that identifies the measuring sensor by its URSI code and YYYYMMDDHHmm is a 12 characters' string that identifies the time of the query, where YYYY, MM, DD, HH and mm mean year, month, day, hour and minute respectively. Once the data has been downloaded from the DIDBase, the bash file calls the rest of the programs and obtains the intermediate and final results which are also stored in the aforementioned folder.

By default, the version for User Defined Time Intervals runs with a time sampling of 5 minutes, but can be user defined. The end of the bash file reads $@seconds = \$seconds + 300$ and the user can adjust 300 (300 seconds; i.e. 5 minutes) to a desired time frequency of analysis, e.g. 900 (900 seconds; i.e., 15 minutes). However, the user must respect the blank spaces in the instruction of the bash file.

Once the data has been downloaded from the DIDBase, both versions of the method for Near Real Time operation and for User Defined Time Interval are exactly the same and no other distinction between them will be noticed from now on.

Note:

As we commented above, the code is applied to 2 different networks of stations, Europe and South Africa. The recent work with ionosondes of South Africa network shows that these ionosondes often experience troubles to provide their data to DIDBase, accumulating a significant time delay. This might result with a scarcity of near-real time data of South African stations into DIDBase, and consequently, the detection LSTID in near real time might not be possible for South Africa network.

6. Description of the input files

6.1. *stations.txt*

The file *stations.txt* is the only input file of the HF-Interferometry code and it must be stored in the main folder. The input file *stations.txt* provides information of the ionospheric sensors (ionosondes) whose data will be used by the HF-Interferometry method. By default, *stations.txt* have the information of ten ionosondes located in Europe and four in South Africa (Figure 4). The user can modify the input file by adding or removing ionosondes. However, it is recommended to verify whether the sensors reported in the input file have data available in the GIRO DIDBase Fast Chars. Note that to get results, be sure to have at least three measuring sensors available and separated by 1000 km or less. Data availability can be consulted at the "Digital Ionogram Data Base" (<http://ulcar.uml.edu/DIDBase/>)

Each line in the input file has the following format:

STA, ALATI, ALONG, NT, ST

Where:

- *STA* is a 5 characters' string that identifies the ionosonde by its URSI code
- *ALATI*, *ALONG* are the latitude and longitude positions of the ionosonde.
- *NT* is a number that indicates a network (or region) where a given ionosonde stands. Currently *NT* = 1 means the European network and *NT* = 2 means the South African network.
- *ST* enumerates the ionosondes of the respective ionosonde networks.

Figure 4. Information provided by default in the input file *stations.txt*.

STA	ALATI	ALONG	NT	ST	FORMAT(A5,1X,2(F7.2),2I4)
AT138	38.00	23.50	1	1	
DB049	50.10	04.60	1	2	
EB040	40.80	00.50	1	3	
FF051	51.70	-1.50	1	4	
JR055	54.60	13.40	1	5	
PQ052	50.00	14.60	1	6	
RL052	51.50	-0.60	1	7	
RO041	41.90	12.50	1	8	
SO148	47.63	16.72	1	9	
VT139	40.60	17.80	1	10	
GR13L	-33.3	26.50	2	1	
HE13N	-34.4	19.22	2	2	
LV12P	-28.5	21.20	2	3	
MU12K	-22.4	30.88	2	4	

As noticed above, the user can modify the file *stations.txt* by removing or adding ionosondes, whose data will feed the analysis. However, the user must apply the same format as shown in Figure 4 to any change introduced in the file *stations.txt*. Any change applied to the file *stations.txt* imply that the vector defined in the last command line of the bash file shown in Figure 2 must be modified accordingly:

```
set stas = ( "AT138" "DB049" "EB040" "FF051" "JR055" "PQ052" "RL052" "RO041" "SO148" "VT139" )
```

This command line sets the ionosondes to be used by the method. Note that all ionosondes set in the above command line must belong to the same network. Therefore, the main code must be applied to a set of ionosondes belonging to a single network.

6.2. *STA_YYYYMMDDHHmm.txt*

These intermediate input data file contain the raw data of foF2 and MUF(3000)F2, hereafter *MUFD*, as measured by the ionosondes (identified by *STA*) for the 24-h interval just before the query time. These data files are downloaded from the GIRO DIDBase Fast Chars (<http://giro.uml.edu/didbase/scaled.php>) and are stored it in the YYYYMMDD folder.

Each of these files are text files which reads two messages (Figure 5). The first message consist of a header of 17 lines providing information about the ionosonde sensor, time interval and data selection. The second message contains as many lines as measurements available within the requested time interval. The lines of the message with data measurements read the following format:

Time, CS, foF2, QD, MUFD, QD

Where,

- *Time* is a 24 characters' string providing year, month, day, hour, minute and second of the measurement.
- *CS* is the Autoscaling Confidence Score (from 0 to 100, 999 if manual scaling, -1 if unknown)
- *foF2* is the measurement of the critical frequency of the F2 layer expressed in MHz
- *MUFD* is the measurement of the maximum usable frequency, expressed in MHz, for a radio transmission of one hop in the F2 layer and for ground distance D, where D is set to 3000 km
- *QD* are a qualitative and descriptive letters related to the previous measurement if apply. Q and D read / for auto-scaled measurements.

Figure 5. Example of a portion of a given intermediate input file *STA_YYMMDDHHMM.txt* which refers to the measurements of the ionosonde in Pruhonice, PQ052, for the 24-h time interval ending at 04:30:30 UT of 16 of July of 2017. Note that measurements were recorded with a sampling time of 15 minutes.

```

PQ052_201707160430.txt
# Global Ionospheric Radio Observatory
# GIRO Tabulated Ionospheric Characteristics, Version 1.0 Revision B
# Generated by DIOBGetValues on 2018-05-18T08:19:46.445Z
#
# Location: GEO 50.0N 14.6E, URSI-Code PQ052 PRUHONICE
# Instrument: Ionosonde, Model: DPS-4D
#
# Query for measurement intervals of time:
# 2017-07-15T04:30:30.000Z - 2017-07-16T04:30:30.000Z
#
# Data Selection:
# CS is Autoscaling Confidence Score (from 0 to 100, 999 if manual scaling, -1 if unknown)
# foF2 [MHz] - F2 layer critical frequency
# MUFD [MHz] - Maximum usable frequency for ground distance D
# Distance D for MUF calculations: 3000 km
#
#Time          CS  foF2 QD  MUFD QD
2017-07-15T04:45:00.000Z  85  4.125 // 12.88 //
2017-07-15T05:00:00.000Z  80  4.488 // 13.75 //
2017-07-15T05:15:00.000Z  80  4.563 // 14.63 //
2017-07-15T05:30:00.000Z  90  4.538 // 14.74 //
2017-07-15T05:45:00.000Z  90  4.688 // 14.59 //
2017-07-15T06:00:00.000Z  95  4.588 // 14.87 //
2017-07-15T06:15:00.000Z  90  4.525 // 15.14 //

```

7. Description of the output files

The HF-Interferometry method provides two types of output files that are automatically stored into the corresponding YYYYMMDD folder described in earlier sections (Table 2).

According to the selected running cadence (X minutes) and the number of selected ionosonde sensors (Y), the program provides, every X minutes, Y number of intermediate output files *STA_YYYYMMDDHHmmR.dat* and 1 output file *TID_HFI_YYYYMMDDHHmm.log*.

Table 2. List of output files generated by the HF interferometry method with a short description of its respective content.

OUTPUT FILE	EXTRACTED CHARACTERISTICS
<i>STA_YYYYMMDDHHmmR.dat</i>	time, de-trended or residual values of foF2, background of foF2, de-trended or residual values of MUF _D , and background of MUF _D
<i>TID_HFI_YYYYMMDDHHmm.log</i>	network, station ID, flag, station, time, latitude, longitude, number of periods, period ID, period, power, amplitude, spectral contribution, velocity, azimuth, traffic light, quality index, range index

7.1. *STA_YYYYMMDDHHmmR.dat* file

The intermediate output files are named as *STA_YYMMDDHHmmR.dat*. STA is a 5 characters' string that identifies the measuring sensor by its URSI code, YYYYMMDDHHmm is a 12 characters' string that identifies the time of the query, where YYYY, MM, DD, HH and mm mean year, month, day, hour and minute respectively. Finally, R refer to the de-trended or residual data. Each of these files contains the respective data of a given ionosonde for a 6-hs time interval ending at the time of the query. The data are reconstructed at 1-min sampling; i.e., each data file has 360 values, and each line of the intermediate output file reads with the following format (Figure 6):

Time, res_foF2, back_foF2, res_MUF, back_MUF

Where,

- *Time* changes from 1 to 360 and refer to of the 6-h interval ending with the time of the query.
- *res_foF2, back_foF2* are residual and background values of foF2 respectively. We refer the user to D2.1 and D2.2 for further details.
- *res_MUF, back_MUF* are residual and background values of MUF_D respectively. We refer the user to D2.1 and D2.2 for further details.

Figure 6. Example of a portion of a given intermediate output file *STA_YYMMDDHHmmR.dat* which refers to the measurements of the ionosonde in Pruhonice, PQ052, for the 6-h time interval ending at 04:30:30 UT of 16 of July of 2017.

PQ052_201707160430R.dat				
1	-0.13	4.20	-0.27	13.18
2	-0.13	4.18	-0.27	13.14
3	-0.12	4.17	-0.27	13.10
4	-0.11	4.16	-0.26	13.07
5	-0.10	4.15	-0.25	13.03
6	-0.09	4.14	-0.23	12.99
7	-0.07	4.13	-0.21	12.96
8	-0.05	4.12	-0.19	12.93
9	-0.03	4.11	-0.16	12.89
10	-0.01	4.10	-0.13	12.86
11	0.01	4.09	-0.10	12.83
12	0.03	4.08	-0.07	12.80
13	0.05	4.08	-0.03	12.76
14	0.07	4.07	-0.00	12.73
15	0.09	4.06	0.03	12.70
16	0.10	4.05	0.06	12.67
17	0.12	4.04	0.09	12.64
18	0.13	4.03	0.11	12.61
19	0.13	4.02	0.14	12.59
20	0.14	4.02	0.16	12.56

7.2. *TID_HFI_YYYYMMDDHHmm.log* file

The output file *TID_HFI_YYYYMMDDHHmm.log* provides the final results of the HF-Interferometry method, where *YYYYMMDDHHmm* is a 12 characters' string that identifies the time of the query, and *YYYY*, *MM*, *DD*, *HH*, and *mm* mean year, month, day, hour and minute respectively. This file contains as many lines as the number of stations set for the analysis for both networks. Each line of the output file reads the following format (Figure 7):

NT, ST, FL, STA, DATE&TIME, ALATI, ALONG, NW, IW, PERIO, POWER, AMPLI, SPCONT, VEL, AZI, TrL, IQ, IR

Where,

- *NT* indicates the network or the region subject of analysis. Currently, 1 and 2 correspond to the European and South African networks respectively.
- *ST* enumerates the sensors of the respective ionosonde networks.
- *FL* is a flag that indicates if a given sensor provides enough data for the analysis. 0 means that the sensor provides less than 80% of the expected data for the interval of analysis and this data is no longer analyzed. 1 means that sensor provides enough data for further analyses.
- *STA* is a 5 characters' string that identifies the ionosonde by its URSI code

- *DATE&TIME* is a 12 characters' string that identifies the date and time of the query as described above.
- *ALATI*, *ALONG* are latitude and longitude of the respective ionosonde.
- *NW*, *IW* are period indicators. The method searches for the two main periodicities in the time series. *NW* specifies how many periodicities are significant in the time series (0, 1 or 2). *IW*, specifies which periodicity is the most coherent for all the stations of the network (0, 1 or 2). The most coherent period will be considered as the dominant period of the disturbance for further analysis. In case of stations with coherent period but bad correlation (less than 3 stations with correlation greater than 0.5) we set the *IW* value to 9, in order to identify these cases.
- *PERIO*, *POWER*, *AMPLI*, *SPCONT*, *VEL*, *AZI* are the period (minutes), spectral power, amplitude (MHz), spectral contribution (%), velocity (m/s) and azimuth ($^{\circ}$ from true North) of the perturbation respectively. We refer the user to D2.1 and D2.2 for further details.
- *TrL* is a warning level indicator: 0 means no data, 1 means quiet, 2 means moderate activity and 3 means severe activity.
- *IQ*, *IR* are quality index. *IQ* indicates how many sensors of the network have enough data ($FL = 1$) to contribute to the calculations in relation to the total number of sensors of a given network. *IR*, is a range index which indicates how many sensors have a cross-correlation maximum value larger than 0.5 with the reference sensor in relation to the total number of sensors that contribute to the calculations ($FL = 1$) and that periods are coherent ($IW \neq 0$). For the European network, if there are less than 3 sensors having a cross-correlation maximum larger than 0.5 with the reference sensor, we do not observe a coherent disturbance in the network (No TID) and set a value equal to -999 to the velocity and azimuth. However, for the South African network, due to the short number of sensors, we only request 2 sensors having a cross-correlation maximum larger than 0.5 with the reference sensor to observe a coherent disturbance in the network.

Figure 7. Example of an output file *TID_HFI_YYMMDDHHmm.log* which refers to the observations for the 6-h time interval ending at 04:30:30 UT of 16 of July of 2017.

TID_HFI_201707160430.log

NT	ST	FL	STA	DATE&TIME	ALATI	ALONG	NW	IW	PERIO	POWER	AMPLI	SPCONT	VEL	AZI	TrL	IQ	IR
1	1	1	AT138	201707160430	38.00	23.50	2	1	112	28.938	0.3056	14.9628	1260	276	2	70	60
1	2	1	DB049	201707160430	50.10	4.60	2	9	0	0.000	0.0000	0.0000	-999	-999	1	0	0
1	3	1	EB040	201707160430	40.80	0.50	2	1	88	9.926	0.4981	12.2630	967	299	2	70	80
1	4	1	FF051	201707160430	51.70	-1.50	2	1	141	132.943	0.2516	88.1191	975	320	3	70	60
1	5	1	JR055	201707160430	54.60	13.40	2	1	146	75.257	0.1795	79.9900	734	325	3	70	60
1	6	1	PQ052	201707160430	50.00	14.60	2	9	0	0.000	0.0000	0.0000	-999	-999	1	0	0
1	7	1	RL052	201707160430	51.50	-0.60	2	1	146	134.188	0.3502	89.2878	878	316	3	70	80
1	8	0	R0041	201707160430	41.90	12.50	0	0	0	0.000	0.0000	0.0000	-999	-999	0	0	0
1	9	0	S0148	201707160430	47.63	16.72	0	0	0	0.000	0.0000	0.0000	-999	-999	0	0	0
1	10	0	VT139	201707160430	40.60	17.80	0	0	0	0.000	0.0000	0.0000	-999	-999	0	0	0
2	1	1	GR13L	201707160430	-33.30	26.50	2	1	101	29.101	0.4120	13.2075	337	260	2	100	66
2	2	1	HE13N	201707160430	-34.40	19.22	2	1	146	57.381	0.1746	72.1181	234	236	3	100	50
2	3	1	LV12P	201707160430	-28.50	21.20	2	9	0	0.000	0.0000	0.0000	-999	-999	1	0	0
2	4	1	MU12K	201707160430	-22.40	30.88	2	1	149	71.518	0.4570	84.1500	969	240	3	100	50

The HF Interferometry method creates also two “dummy” files named *previous_file_1.log* and *previous_file_2.log* (1 refers to the European network and 2 refers to the South African network) which contains the same results as in file *TID_HFI_YYYYMMDDHHmm.log* but for the previous instant of the latest query time. This file is needed to check for the temporal stability of the velocity and azimuth of a given disturbance.