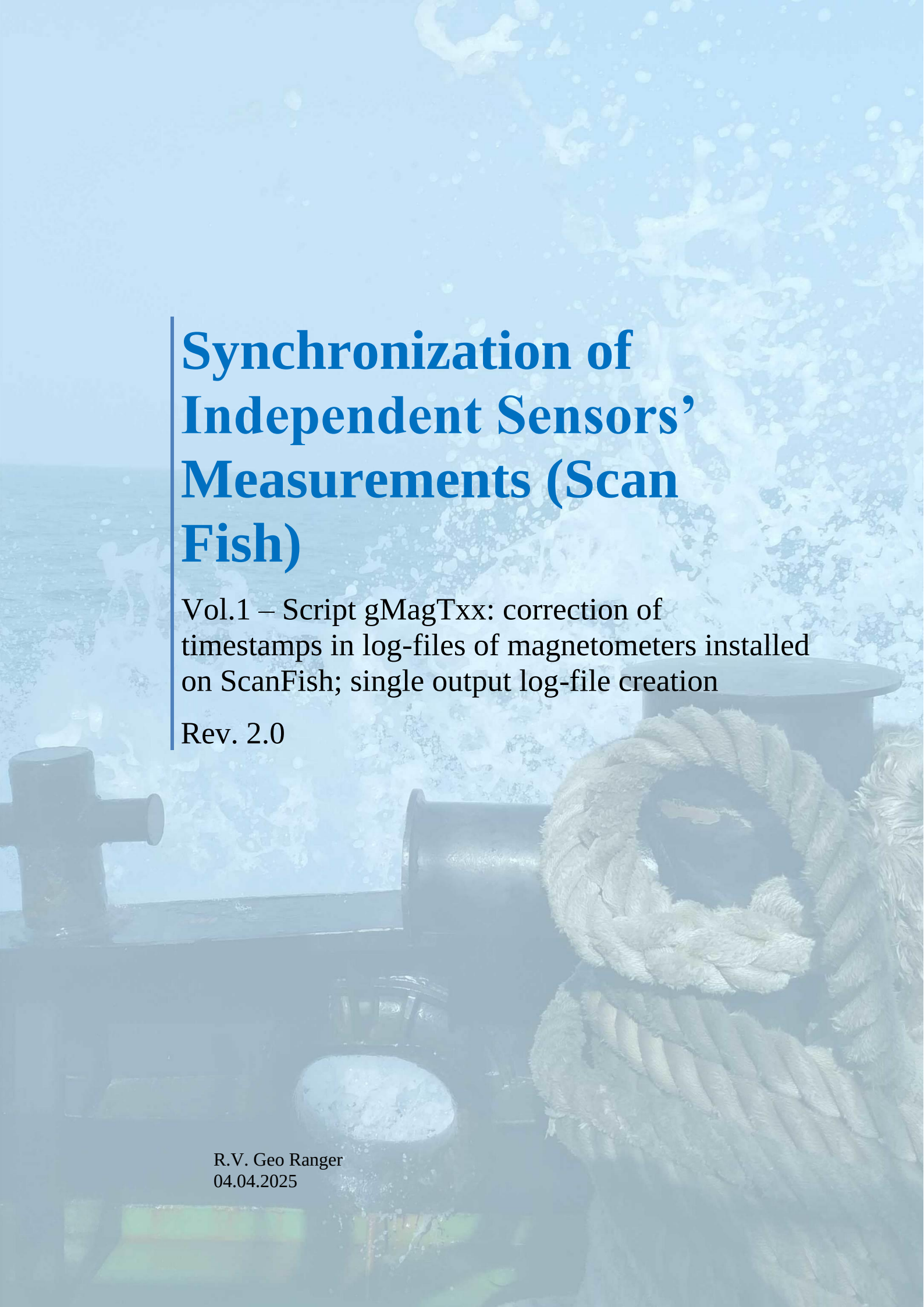


Synchronization of Independent Sensors' Measurements (Scan Fish)

Vol.1 – Script gMagTxx: correction of timestamps in log-files of magnetometers installed on ScanFish; single output log-file creation

Rev. 2.0

R.V. Geo Ranger
04.04.2025

SEA

Content

History of changes

Version	Date	Description of changes
1.0	2024-12-20	Document created for MagT10 version // Ivan D.
1.1	2024-12-31	Added changes related to the MagT11 version // Ivan D.
1.2	2025-01-19	Added minor corrections to the text // Ivan D.
2.0	2025-04-04	Added GUI and "schedule running", changed parameters of commands and names of parameters, renamed the script to MagT13 // Ivan D.

Quick start

QuickStart provides an example of running the script to create a single log-file for all magnetometers from the log-files of individual magnetometers.

0) Place the log-files with the individual magnetometer records to the "root folder" (see Appendix 3 for a description of the log-file format).

1) Install GNU Octave (the installation process is described in Section 3.1).

2) Enter the command in the Octave / MatLab command window to start the Simplified GUI:

{'GUI'};MagT13;

Press "Enter" to start the command. The GUI window will open.

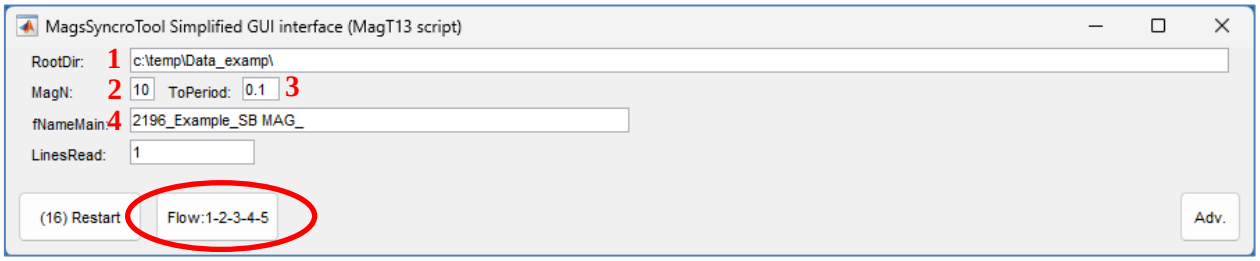
3) Adjust the following parameters in the GUI window if necessary:

1 -- the path to the root folder where the log-files with magnetometer records are located;

2 -- the number of magnetometers placed on ScanFish;

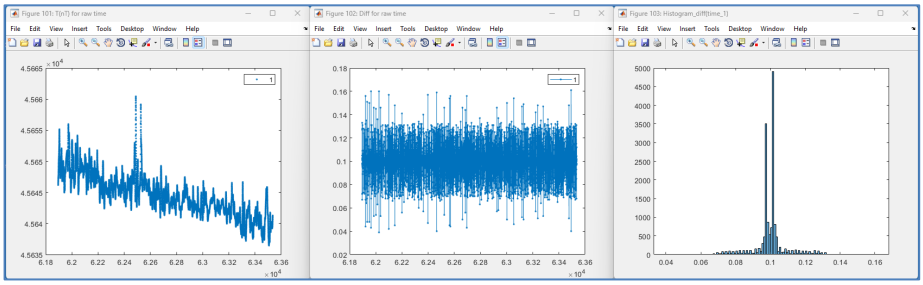
3 -- period of magnetometer measurements in seconds;

4 – the "main part" of the file name (files such as 2196_Example_PS MAG_01.txt, 2196_Example_PS MAG_02.txt, etc. have the following "main part": 2196_Example_PS MAG_).



4) Click on the "Flow:1-2-3-4-5" button to start the "working command sequence".

5) When loading magnetometers and correcting timestamps, three graphical windows are drawn for each magnetometer:



To start processing the next magnetometer one should press the "Backspace" key each time. The command window will display messages regarding processing progress.

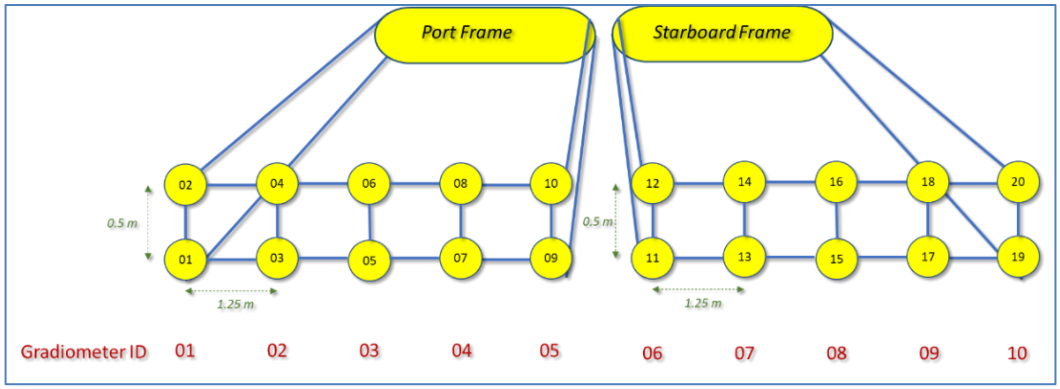
6) When the script has finished, a single log-file containing the records of all magnetometers will appear in the root folder.

Note: Use the "Adv." button to switch to the Advanced GUI mode.

1. Introduction

1.1 Annotation

The paper discusses the processing of marine magnetometry data using an array of magnetic sensors installed on ScanFish.



The document also considers working with the MagT13 script, designed for controlling delays in the ScanFish-QINSy data channel and applying corrections for delays in the data channel. The following issues are discussed:

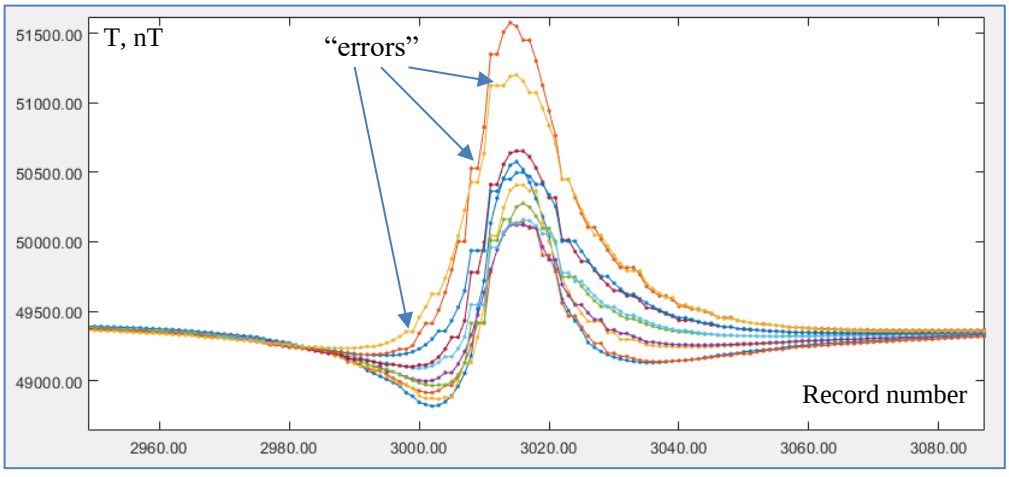
-- installation of GNU Octave required to run the script;

-- description of the script commands and their “working sequences”;

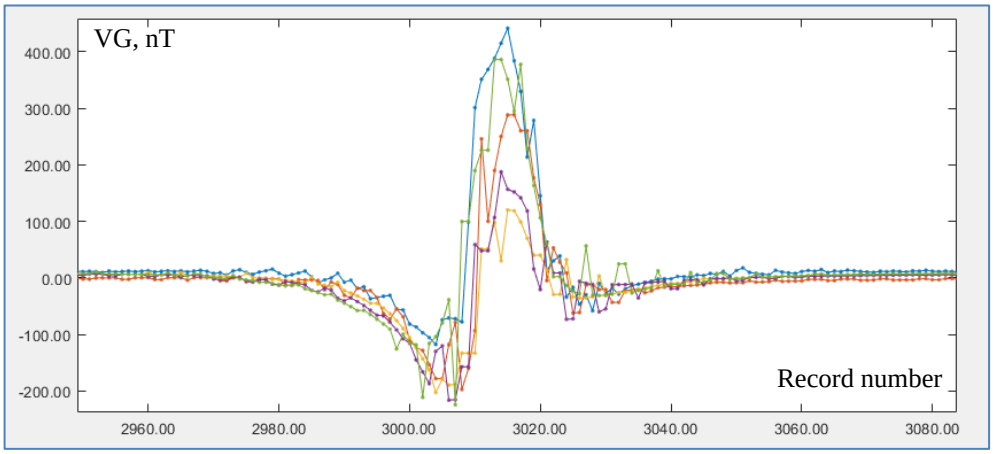
-- an example of working with the script through the graphical user interface (GUI) with the test record of magnetometers.

1.2 Problem statement

Delays in the ScanFish-QINSy data transmission channel introduce an additional type of measurement inaccuracy associated with the measurement time-referencing errors (synchronisation errors). This inaccuracy is significantly increased by saving measurements of magnetometers (digital messages from ScanFish) into a single log-file when using measurements of one magnetometer as a trigger. The artefacts arising due to synchronisation errors are clearly visible in the areas with fast magnetic field changes (figure below).



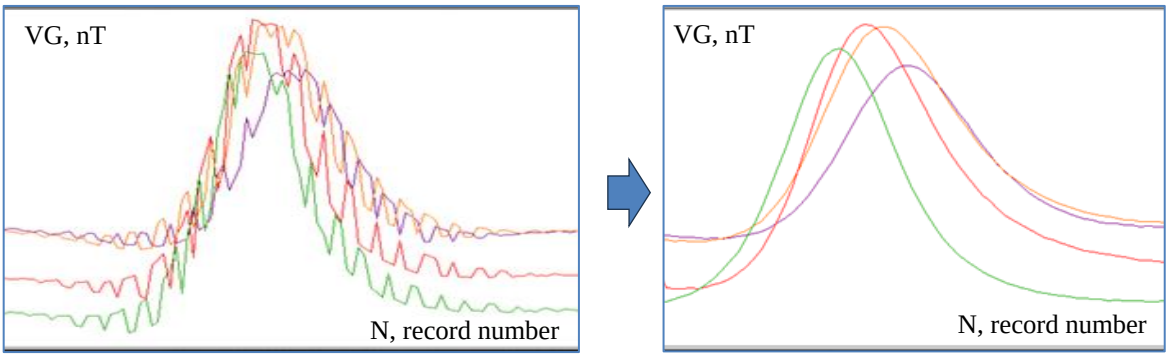
These artifacts are clearly visible as a “saw effect” (high-frequency meander) in the magnetic field gradient plots (the difference between the readings of two magnetic sensors).



In order to minimize synchronization errors and enhance data quality (including the removal of the “saw effect”) three conditions must be met:

1) the measurements of each magnetometer (digital messages from ScanFish) must be recorded into a separate log-file without using a magnetometer-trigger (QINSy will record the exact time of receiving digital messages containing measurements);
2) the settings for the digital data transmission channel must be adjusted to reduce delays in the data transmission channel;
3) corrections for time delays should be introduced in the magnetometer records to bring the time of “receiving digital messages from the magnetometers” closer to the time of “taking measurements by the magnetometers”.

Fulfilment of these conditions allows to significantly reduce synchronisation errors and improve the quality of magnetometric survey data; including removal of the “saw effect” in magnetic field gradient plots).



The paper discusses the script implementing the third condition – applying corrections for time delays in magnetometer records and interpolating magnetometer measurements to the common timestamps.

1.3 Discussion

Why should timestamp correction be done at all?

Let us consider an analogy – if there are spikes in magnetic field measurements, there is no doubt about the need to apply the procedure of spike removal. "Removing spikes" from magnetic field measurements to reduce their influence on the results of subsequent processing is an obvious thing to do. The procedure of "synchronisation error correction" has the same purpose as the procedure of spike removal, however, it does not process the values of the measured magnetic field, but those of the timestamps.

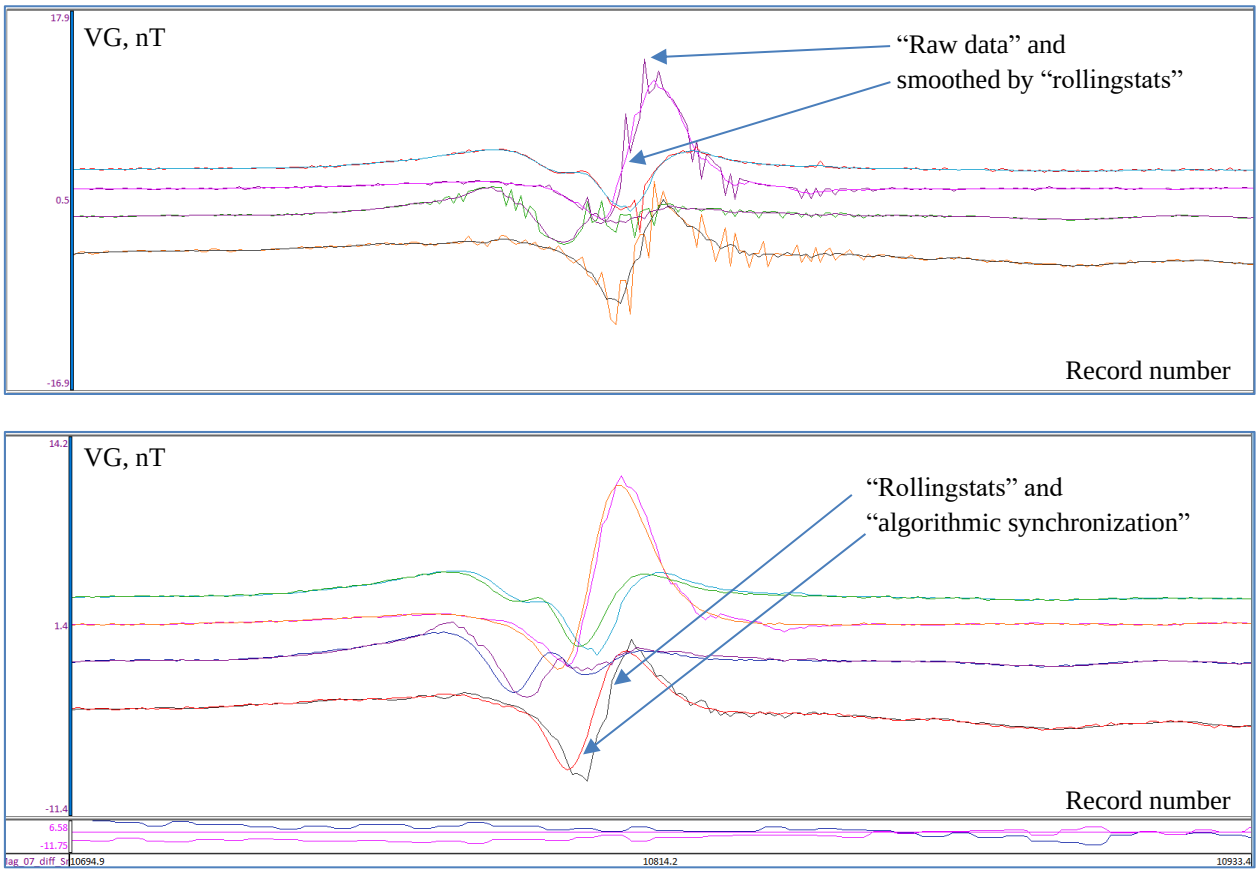
On-line should provide data for processing without synchronisation errors, processing should not be doing it.

The statement is correct in the sense that online should collect the highest quality data possible. However, in the particular case – online cannot remove synchronization errors without changing the ScanFish electronics. Therefore, synchronisation errors must either be corrected during the data processing phase or must be purposefully ignored if the SoW allows. However, even so, a tool is needed to monitor the magnitude of synchronisation errors, to confirm "they are within acceptable limits".

Can the "saw effect" caused by synchronisation errors be removed by magnetic field smoothing procedures?

The use of smoothing procedures can reduce high-frequency noise and the "saw effect" in magnetic field gradient plots, but this "synchronisation error removal" is not correct. The analogy above with spikes in magnetic field measurements is relevant here – if spikes are present in magnetic field measurements, it is inappropriate to apply a high-pass filter to such a type of data in order to "reduce the amplitude of spikes". When filtering, the error, enclosed in the spike, being "smeared" by the filter on adjacent measurements. There is a special procedure for spike removing that, most often, recognizes a measurement with an outlier and excludes it from the data set. This means that different processing procedures specifically designed for this type of error, are used to remove errors of different nature and with different characteristics. Qualitative removal of synchronization errors also requires a special tool designed for this specific purpose.

To confirm the thesis on the necessity of using a special tool, the first figure below compares magnetic field graphs for raw data (illustrating the "saw effect") and those smoothed using "rollingstats" in Oasis Montaj (rollingstats.statistic="6"; rollingstats.width="5"; rollingstats.shrink="1"). The second figure compares the graphs smoothed using "rollingstats" and those obtained by removing synchronization errors (algorithmic synchronization). It is clear that removing synchronisation errors gives a better result.

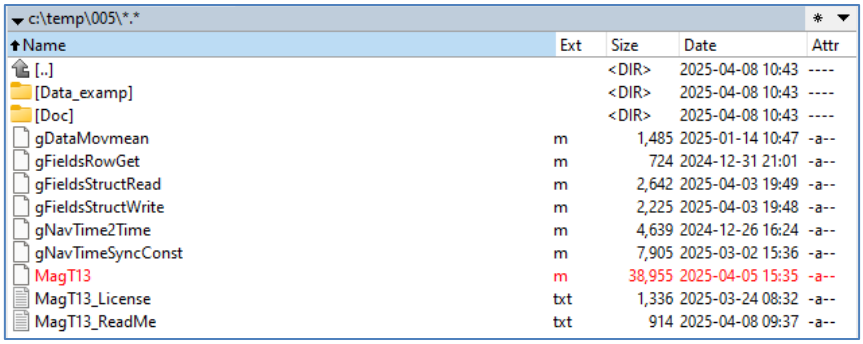


In a strict sense, the "raw" magnetic field records shown in the figure above can be regarded as "records which are complicated in the anomaly region by high-frequency noise exceeding the allowable SoW error". The application of "rollingstats" is then a method of "masking/smoothing high frequency noise" measured by magnetic sensors. Being aware of the delays in the data channel and applying methods of their correction show that high-frequency noise ("saw effect") is not caused by noise on magnetic sensors. The data with corrected timestamps are free of high-frequency noise and meet the SoW requirement.

2. Preparing to work with the MagT13 script

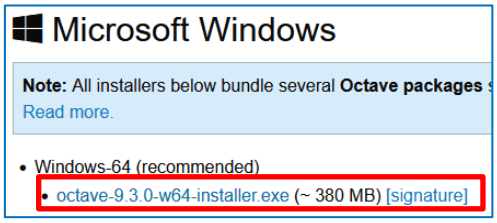
2.1 GNU Octave installation

1) Unpack the MagT13.7z archive into the selected folder. The path to the folder should be short for convenience; the example uses c:\Temp\005\.

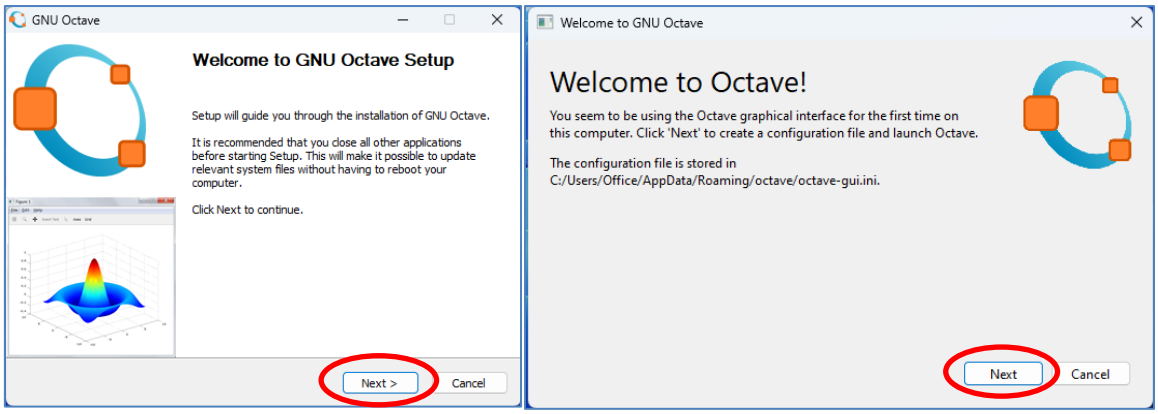


After unpacking the archive, the selected folder will contain files with the script code, files with the functions required for the script operation, the folder with the script documentation and the folder with the test record of magnetometers.

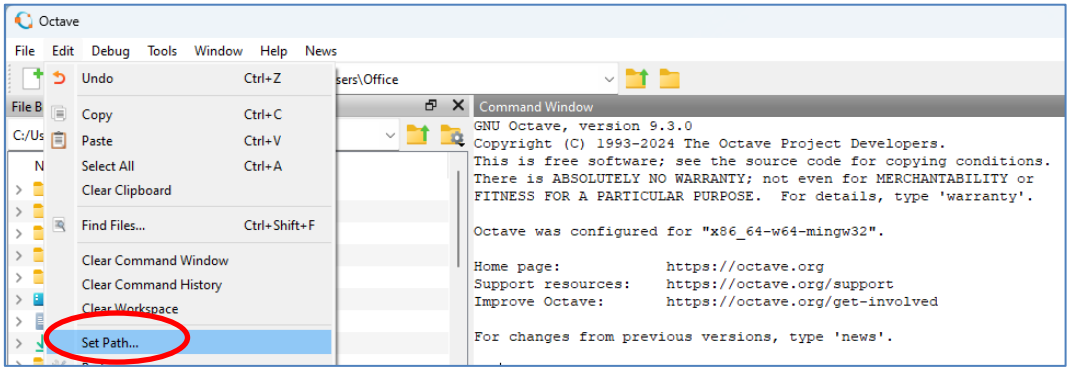
2) Download the GNU Octave distribution from https://www.octave.org/download.

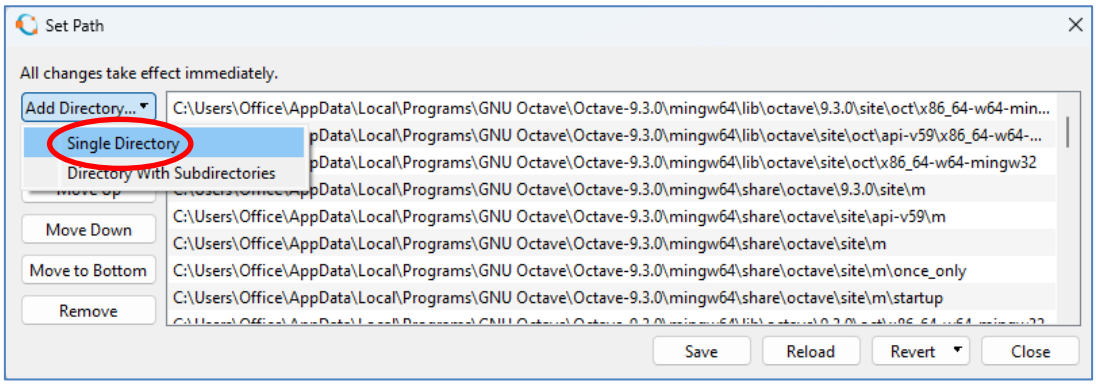


3) Install Octave using all default settings.

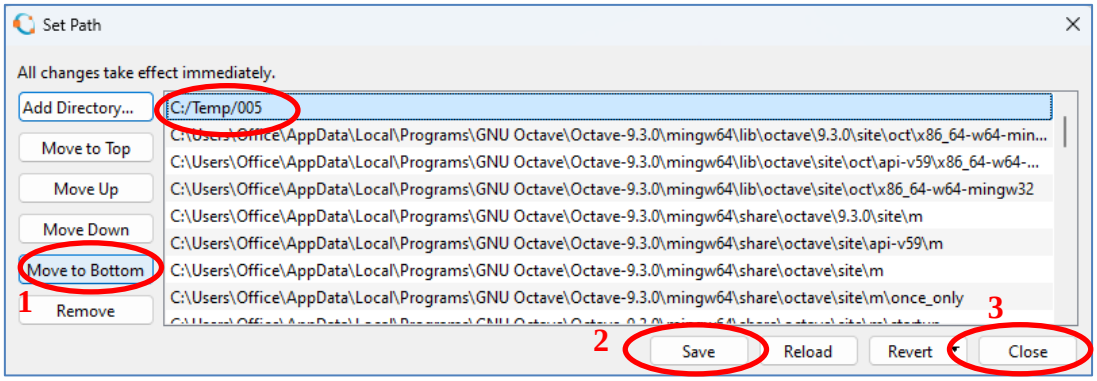


4) Add the path to the folder with the script files to the Octave settings.

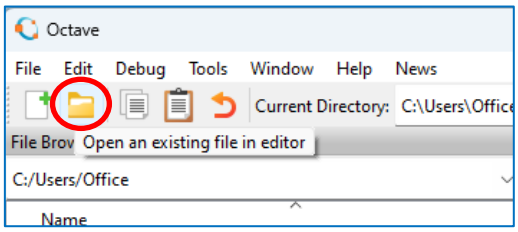




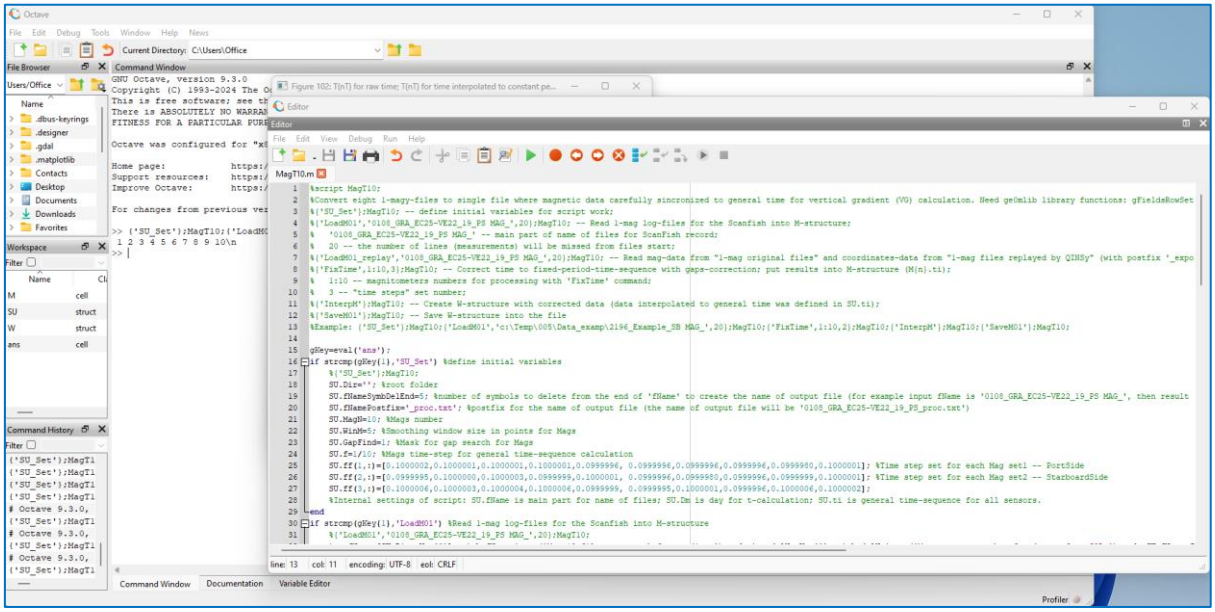
Highlight the row with the entered path and move it to the bottom of the list.



5) Open the MagT13.m file in the Octave script editor.



Use the mouse to grab the editor window in the “editor” label area and drag it outside the main Octave window.



6) Restart Octave. To do so, close the main window (but not the script editor window!) using the cross in the upper right corner and start Octave using the GUI icon that was added to the Windows desktop when you installed Octave.

7) The GNU Octave is installed and ready to go.

Note

In the script editor window, it is possible to correct values of command parameters displayed in the GUI window “by default”. This is convenient, for example, when setting the root folder in GUI or when changing the number of magnetometers installed on ScanFish.

```
gKey=eval('ans');
if any(strcmp(gKey{1},{'GUIa','GUI'})) %create Setup values for GUI
    %{'GUI'}:MagT13:
    GUI.gui.RootDir='c:\temp\Data_examp\';
    GUI.gui.MagN='10';
    GUI.gui.ToPeriod='0.1';
    GUI.gui.GapAccept='3';
    GUI.gui.fNameMain='2196_Example_SB MAG_';
    GUI.gui.LinesRead='1';
    GUI.gui.nnMagFix='1:10';
    GUI.gui.T1Period='0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1';
    GUI.gui.WinM='5';
    GUI.gui.T1CalcNum='2500';
    GUI.gui.T0InitPoint='';
    GUI.gui.T0InitLen='25';
    GUI.gui.TimeErrRob='2.72';
    GUI.gui.T0Low='{''rob2'',3,[2.72 1.81 1.36 1.3]}';
    GUI.gui.BMagInterp='1';
    GUI.gui.fNameDelE='5';
    GUI.gui.fNamePostfix='_proc.txt';
    GUI.gui.fFlagStat='0';
    GUI.gui.fNameMain_rp='2196_Example_SB MAG_';
    GUI.gui.LinesRead_rp='1';
    switch gKey{1}
        case 'GUIa'
            GUI.fh=figure(100);set(GUI.fh,'MenuBar','none','ToolBar','none','Na
            uicontrol('style','pushbutton','units','normalized','position',[0.0
```

2.2 General principles of working with the GUI

To start the Advanced GUI, type the command in the Octave / MatLab command window

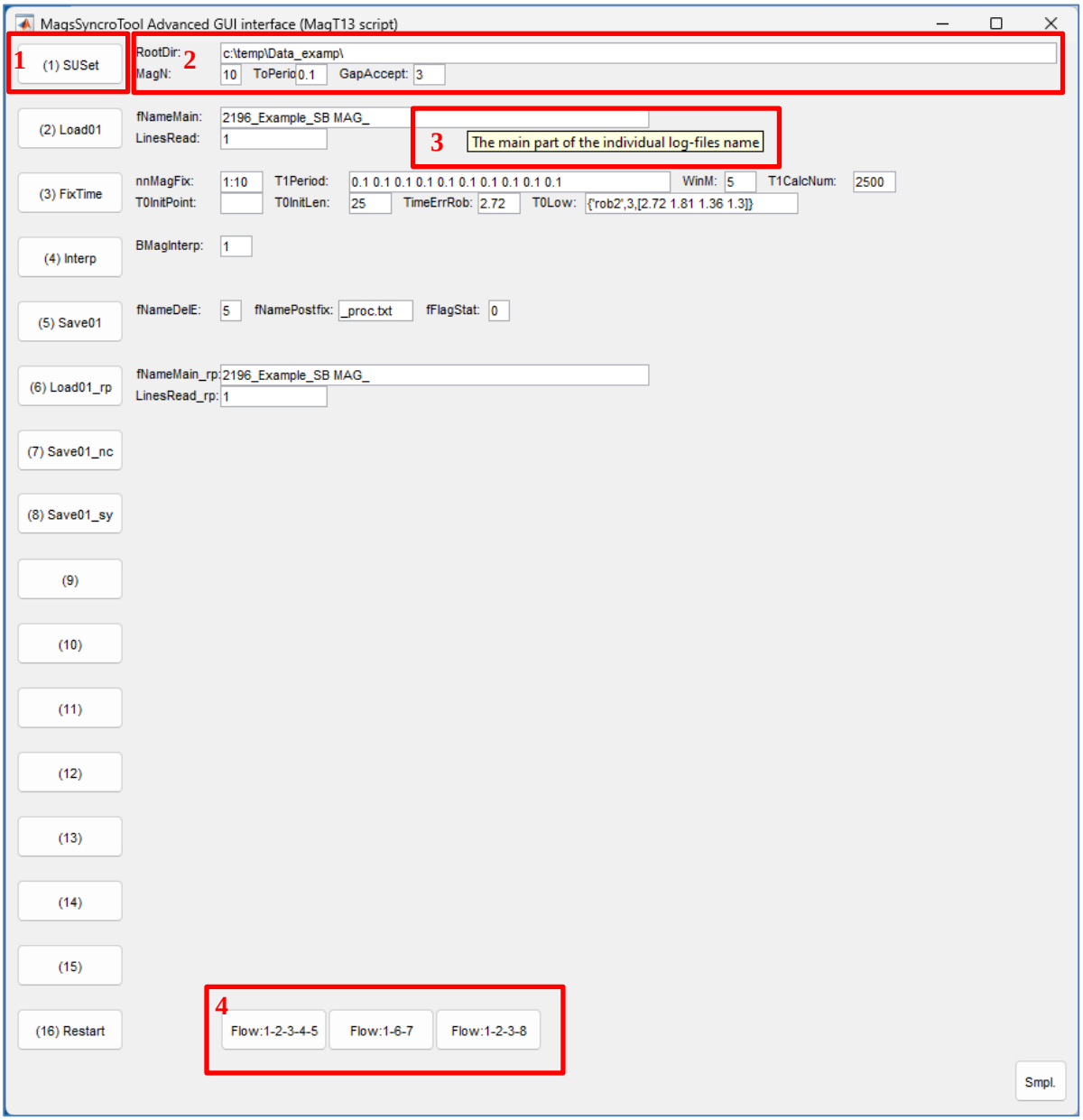
{'GUIa'};MagT13;

Press “Enter” to start the command. The GUI window will appear on the monitor, including the following elements:

-- Buttons (1), with each one corresponding to a specific script command (pressing the button starts the command);

-- Areas with parameters for each script command (2). Parameters can be changed if necessary; holding the mouse cursor over a graphical element to enter a parameter will display a brief help about this parameter (3);

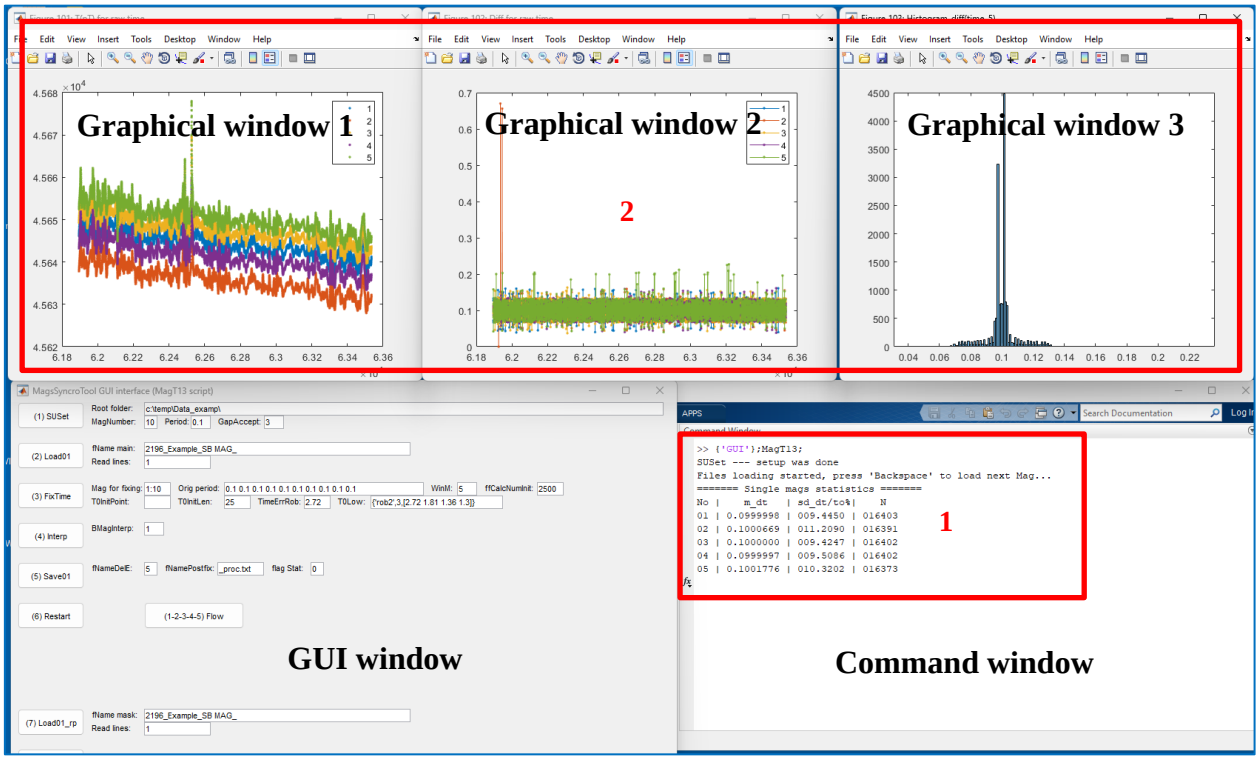
-- Buttons for executing the “working sequences” of commands (4). Pressing a “working sequence button” (e.g. Flow:1-2-3-4-5) is equivalent to sequentially pressing the corresponding buttons (e.g. 1, 2, 3, 4, 5).



After clicking each button, the corresponding command is executed. While doing so, the command window displays messages about the completion of the command execution or about the progress of the executing command. For example, when loading files with magnetometer data (Load01 button), the command window displays data statistics for each file (1).

Some commands take a relatively long time to execute; the script always displays a message indicating the start and the end of such commands. If you click the button again without waiting for the command completion message, the command will be sent for execution a second time after completion.

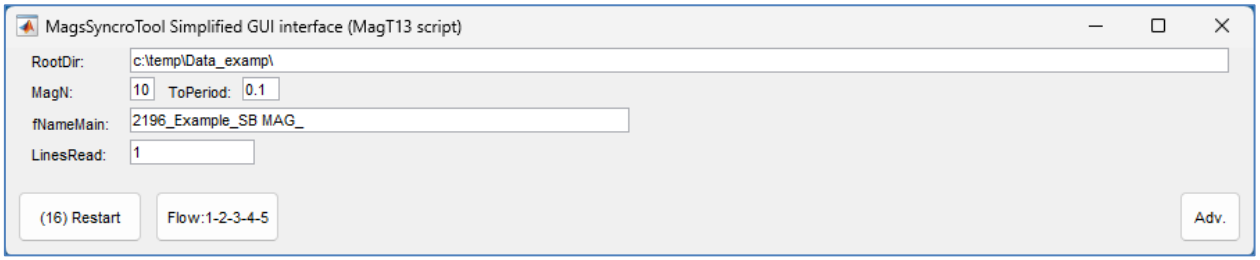
When executing the command, the computer screen may display graphical windows designed to control calculations. For example, three graphical windows (2) are displayed when loading files with magnetometer data (Load01 button); while doing so, the script stops and one should press the “Backspace” key to continue (loading the next magnetometer). Following the completion of the command execution, some of the graphical windows will be closed, while others will remain open.



To start the Simplified GUI, type the command in the Octave / MatLab command window

{'GUI'};MagT13;

or click the “Smpl.” button. Following this, the GUI will be restarted in the mode of access to the basic settings only, and all changed settings will be reset to their “default” values.



2.3 General principles of working with the script in command mode

The script is written to recognise and execute the *script commands*. Several *script command parameters* can be entered for each script command. Commands can be executed by clicking a button in the GUI or by typing the *execute command* and its parameters directly in the Octave / MatLab command window. Example of the execution command:

{'Load01','2196_Example_PS MAG_',20};MagT13;

where:

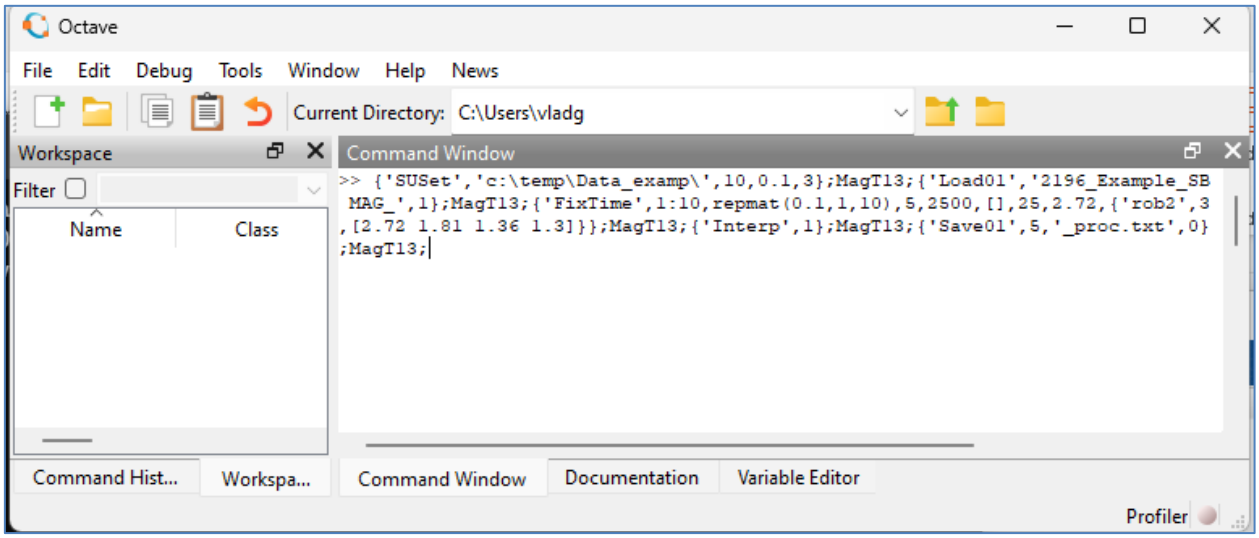
'Load01' – the name of the script command executing a certain section of the script code;

'2196_Example_PS MAG_' – the first parameter of the script command;

20 – the second parameter of the script command;

MagT13 – the name of the script that processes the command.

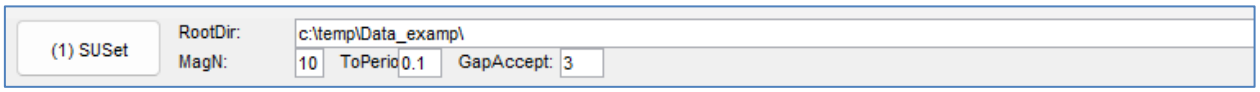
Data processing using the script is performed as a sequential execution of several script commands with specified parameters, i.e. “working sequence” of the commands. The latter can be executed in a single line entered in the Octave / MatLab command window; an example is shown in the screenshot below.



3. Description of MagT13 script commands and Advanced GUI buttons

3.1 Script commands

SUSet command // button (1) SUSet – define initial settings for the script operation.



Example of the command:

{'SUSet','c:\temp\Data_examp\',10,0.1,3};MagT13;

where:

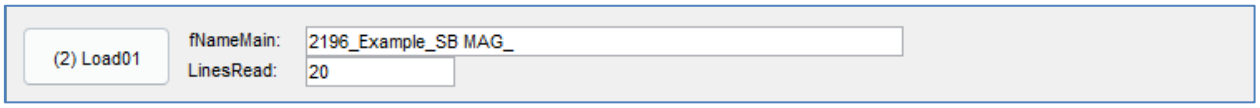
-- RootDir='c:\temp\Data_examp\' – The path to the root folder that contains the log-files of magnetometers. When loading files, the path is automatically substituted to the beginning of the "main part of the file name" specified in the Load01 command. Saving files using the Save01 command is also done in the specified path.

-- MagN=10 – The number of magnetometers installed on ScanFish;

-- ToPeriod=0.1 – The period of magnetometer measurements, specified during the equipment setup. The script utilises it in order to generate a series of timestamps to which the measurement values from all sensors, recorded in log-files, will be interpolated.

-- GapAccept=3 –The number of "missing measurements" for which interpolation to common timestamps is allowed. A value of 3 implies that no more than 2 missing measurements can be interpolated. Then, if 3 measurements are missing, they are assigned the value of NaN; this forms a "gap" in the data.

Load01 command // button (2) Load01 – load log-files with records of individual magnetometers (the format of log-files is given in Appendix 3).



Example of the command:

{'Load01','2196_Example_SB MAG_',20};MagT13;

where:

-- fNameMain='2196_Example_SB MAG_' – The *main part* of the individual log-file name (does not include the magnetometer's number postfix and file extension);

-- LinesRead=20 – Range of line numbers in the magnetometer log-file which are used for calculations. If the range is set to a single value, measurements from the specified line number to the last line in the file (e.g. from 20th line to the end of the file) are used. If the range is specified by two values, the measurements recorded in the lines whose numbers lie between these values are used.

Note: all the lines are first read from the file, then the specified range of lines "passes for use"; therefore, all the lines must contain data in the specified format.

The LinesRead parameter allows measurements at the beginning of the log-file to be missed. This may be necessary if the timestamps for the first counts contain a significant error.

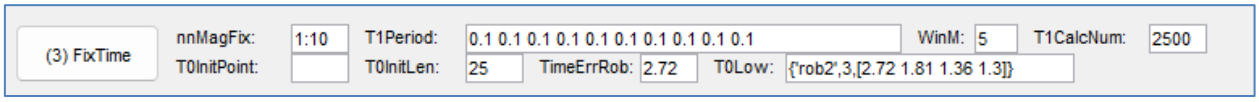
The full file name of the magnetometer record has two characters with the magnetometer number and a *.txt extension at the end; examples of full file names are:

2196_Example_PS MAG_01.txt

2196_Example_PS MAG_02.txt

For the example above, the main part of the file name is 2196_Example_PS MAG_.

FixTime command // button (3) FixTime – perform the correction for the measurement timestamps of the specified magnetometers.



Example of the command:

{'FixTime',1:10,repmat(0.1,1,10),5,2500,[],25,2.72,{'rob2',3,[2.72 1.81 1.36 1.3]}};MagT13;

where:

-- nnMagFix=1:10 – Numbers of magnetometers to be processed by the FixTime command (1:10 means "from the 1st to the 10th", 8 means "the 8th magnetometer");

-- T1Period=[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1] – "Approximate actual periods" of logging for each magnetometer (actual period T1Period differs from the one specified for the equipment in 6-7 decimal places, due to electronics errors). These periods are used to automatically calculate the "exact actual logging periods" of magnetometers installed on ScanFish (example of the selected values: [0.1000002, 0.1000001, 0.1000001, 0.1000001, 0.0999996, 0.0999996, 0.0999996, 0.0999996, 0.0999980, 0.1000001]). Manual correction of "approximate actual logging periods" of magnetometer is described in Appendix 1.

-- WinM=5 – Width of the smoothing window with an odd number of points (5, 7, 9, 11) for averaging the time of the magnetic field measurements. Smoothing is used when searching for "lost" values in a series of magnetometer measurements. The recommended window width is 5;

-- T1CalcNum=2500 – The size of the half window used to automatically correct the "approximate actual periods". When NaN is set, automatic correction is disabled;

-- T0InitPoint=[] – The number of the point whose time value is used as the "zero mark" for the magnetic field record. If the value is set as "empty", the point in the middle of the magnetic field record is taken. If automatic zero-mark correction is enabled, the position of the point will be "selected", relative to the specified point, in the range from minus T0InitLen to plus T0InitLen (see the setting below). Manual correction of the "zero mark" position for magnetometer records is described in Appendix 2.

-- T0InitLen=25 – The size of half of the window used for automatic zero-mark correction (if empty, then set to 15+WinM). When NaN is set, automatic zero-mark correction is disabled.

-- TimeErrRob=2.72 – Robust stigma coefficient used to estimate T0 and T1 in the first and second approximation models when adjusting the timestamps;

-- T0Low={'rob2',3,[2.72 1.81 1.36 1.3]} – Name and parameters of the model that implements the rule “a measurement cannot be received through a data channel before it has been performed”.

In some cases, it is possible that timestamps with duplicate time may be added into the file. This can happen when the file is replayed in QINSy, or when QINSy fails. In this case, a message listing the numbers of points with duplicate time is printed in the Octave command window. A value equal to 1/100 of the period is added to the repeating timestamp to avoid an error when executing the “InterpM” script command (the error is generated by the “interp1” function).

If there is a measurement skip in the time stamps with a length longer than that set in GapAccept (SUSet command), a message is printed in the command window listing the point numbers followed by the measurement skip.

Interp command // button (4) Interp – interpolate data (magnetic field measurements, magnetometer coordinates, etc.) to a common time series of measurements. In this case, the timestamps from the log-file are used for interpolating the coordinates (since QINSy determines the value of coordinates for this very moment of time), while the “corrected sequences” of timestamps are used for interpolating the magnetometer measurements.

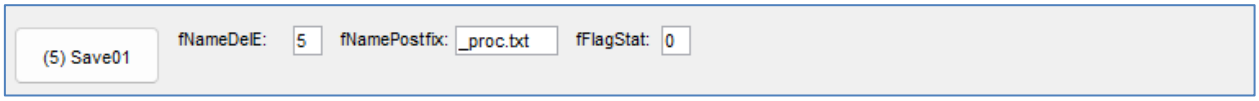
Example of the command:

{'Interp',1};MagT13;

where:

-- BMagInterp=1 – The number of the magnetometer whose log-file is used to obtain the coordinates of ScanFish and other “common” sensors in the interpolated data.

Save01 command // button (5) Save01 – save the data of all magnetometers, recalculated to a common measurement time series, in the single file (the format of log-files is given in Appendix 3). The name of the output file is formed from the “main part of the file name” with deleting the several characters on the right and adding a postfix.



Example of the command:

{'Save01',5,'_proc.txt',0};MagT13;

where:

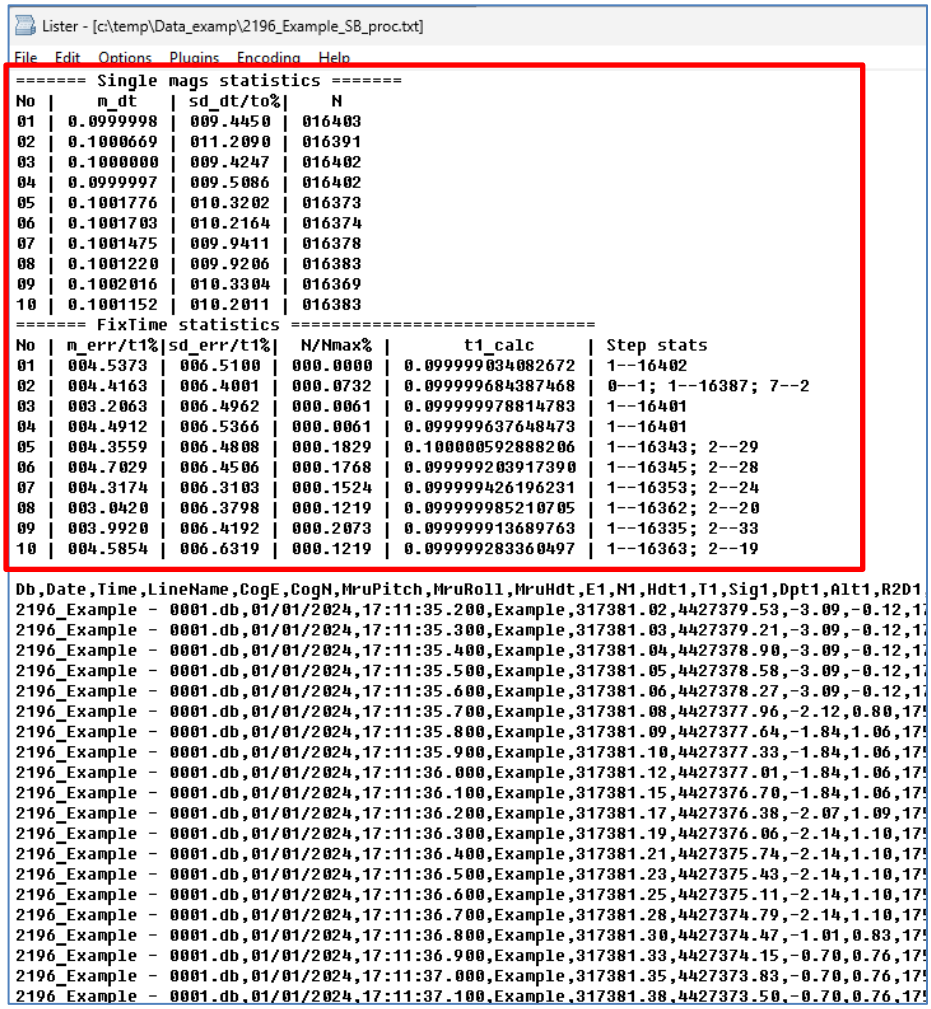
-- fNameDelE=5 – Number of characters removed from the end of the “main part of the file name” to create the name of output file;

-- fNamePostfix='_proc.txt' – Postfix and extension added to the end of the saved file name. For the values given in the example, from the name of the main part “2196_Example_SB MAG_” 5 characters from

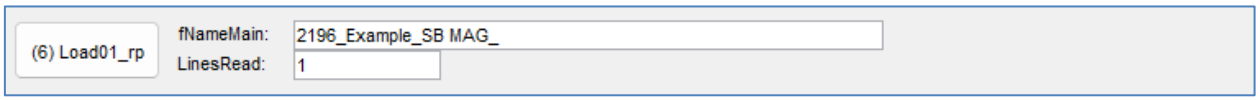
the end are removed and the postfix "_proc.txt" is added. As a result, the name of the saved file is "2196_Example_SB_proc.txt".

-- fFlagStat=0 – Flag for recording timestamp statistics in the header of the saved file (0 – do not record statistics, 1 – record statistics).

An example of statistics in the file with processing results is shown in the figure below.



Load01_rp command // button (6) Load01 – load the magnetometer log-files and coordinate log-files replayed in QINSy (with the postfix '_export') and inject the coordinates to the magnetometer data.



Example of the command:

{'Load01_rp','2196_Example_SB MAG_',20};MagT13;

The description of the parameters "2196_Example_SB MAG_" and 20 is the same as for the Load01 command parameters. Example of file names processed by the Load01_rp command:

2196_Example_PS MAG_01.txt

2196_Example_PS MAG_01_export.txt

2196_Example_PS MAG_02.txt

2196_Example_PS MAG_02_export.txt

The coordinate files are named the same as the magnetometer files, but the postfix "_export" is added to their names. The command was added because when exporting from QINSy, numerous gaps in the magnetometer log-files appear in the magnetometer coordinate data. Therefore, the export is performed twice: with linking to the time of receiving the magnetometer data (the files have gaps in the coordinates); with linking to the time of receiving the coordinates from USBL (the files have gaps in the magnetometer data). Then, based on the timestamps, the magnetometer coordinates are linearly interpolated to the timestamps of the magnetometer data.

The command can be used, for example, if erroneous data are received from USBL. In this case, files with "correct" coordinates are exported from QINSy and then, using the Load01_rp command, the coordinates are "injected" to the magnetometer data.

Save01_nc command // button (7) Save01_nc – save magnetometer data in the format of the single-sensor log-files, read by Load01 or Load01_rp commands, with the postfix "_nc" added to the output file name.

Example of the command:

{'Save01_nc'};MagT13;

The command can be used, for example, to save magnetometer data in the "initial data" format, after injecting coordinates replayed by QINSy, which is done with the Load01_rp command.

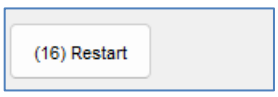
Command Save01_sy // button (8) Save01_sy – save the data of all magnetometers in the format of the original logs, after synchronisation by the FixTime command, with the postfix "_sy" added to the output file name. Before saving, the command performs interpolating the coordinates to the timestamps corrected by the FixTime command.

Example of the command:

{'Save01_sy'};MagT13;

The command can be used, for example, to save magnetometer data in the "initial data" format after correcting the timestamps by the FixTime command.

Button (16) Restart – when clicking the button, all graphical windows are closed, all variables with processed data are deleted from the computer RAM and the GUI is restarted.



Example of the command:

```
clear all;close all;clc;{'GUI'};MagT13;
```

The button can be used to clear the computer RAM after processing the profile with magnetometry data.

3.2 Script command sequences in the GUI

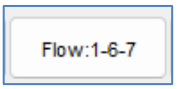
Button Flow:1-2-3-4-5

When clicking the button, a single log-file for all magnetometers is combined from the log-files of the individual magnetometers. While doing so, the timestamps are being corrected to bring them from the "moments of arrival of the digital message with the magnetometer measurement in QINSy" to the moments of "making the measurement". Following the corrections made, the magnetometer data and coordinates are interpolated to a single sequence of timestamps.

After clicking the button, the script sequentially performs the following actions:

-- Defining the script settings with the SUSet command;
-- Loading the magnetometer log-files with the Load01 command;
-- Correcting the timestamps with the FixTime command;
-- Interpolating data to the common timestamps with the Interp command;
-- Saving the data from all magnetometers to a general log-file with the common timestamps using the Save01 command.

Button Flow:1-6-7



When clicking the button, a new navigation is "injected" to the magnetometer data, which are then saved in the format of the original logs.

After clicking the button, the script sequentially performs the following actions:

-- Defining the script settings with the SUSet command;
-- Loading the magnetometer log-files and replayed log-files with navigation using the Load01_rp command, after which the navigation is injected to the magnetometer data based on timestamps;
-- Saving the magnetometer data to separate log-files in the format of the original logs with the Save01_nc command.

Button Flow:1-2-3-8

When clicking the button, the timestamps of the magnetometer data are being corrected, after which the magnetometer data are saved in separate log-files. While doing so, the data of different magnetometers are not interpolated to single timestamps, but are saved on the disk in an unchanged form (each magnetometer's measurements will correspond to its own series of timestamps).

After clicking the button, the script sequentially performs the following actions:

-- Defining the script settings with the SUSet command;

-- Loading the magnetometer log-files with the Load01 command;

-- Correcting the timestamps with the FixTime command;

-- Interpolating the data from other sensors to the magnetometer timestamps, after which all data are saved to separate log-files with the Save01_sy command.

4. Example of executing the script commands in Advanced GUI – recalculating magnetometer data to a common sequence of timestamps

To obtain a single log-file for all magnetometers from log-files of individual ones, you can issue commands to the script using one of the following options:

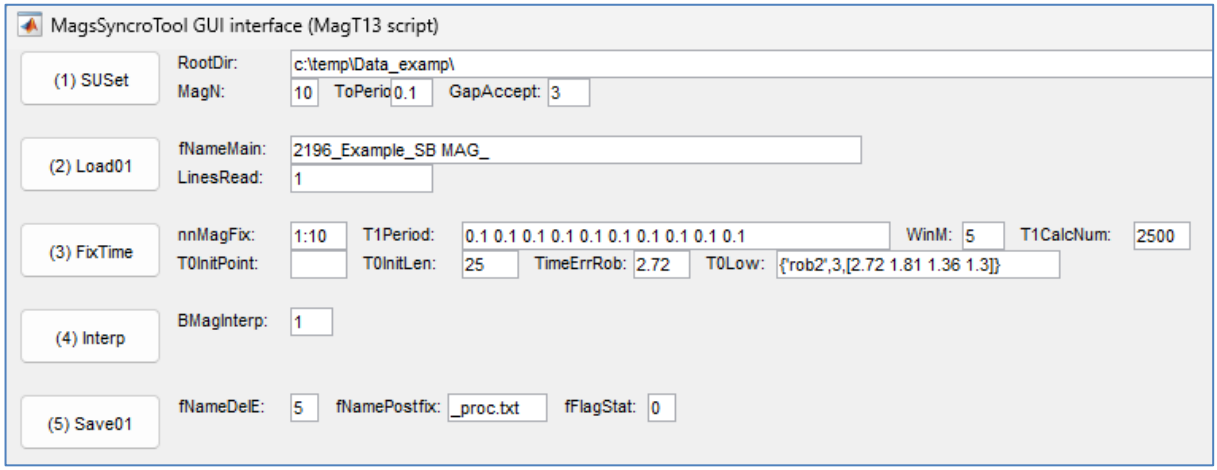
-- sequentially execute the commands SUSet - Load01 - FixTime - Interp - Save01 in the Matlab / Octave command window (with adding appropriate values of command parameters);

-- sequentially press the buttons (1) SUSet - (2) Load01 - (3) FixTime - (4) Interp - (5) Save01 in the GUI;

-- press the button to execute the working sequence of commands “Flow:1-2-3-4-5” in the GUI.

Below shows how we use the second method. To do so, let us start the GUI with the command

```
{'GUI'};MagT13;
```



4.1 Button (1) SUSet

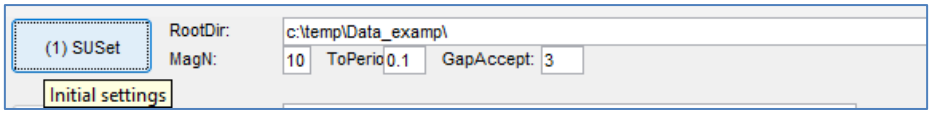
By clicking the (1) SUSet button, the SUSet command is being executed, which sets the initial settings for the script operation:

-- path to the root folder;

-- number of magnetometers installed on ScanFish;

-- period of magnetometers measurements, defined when setting up the equipment;

-- number of missed measurements, whose linear interpolation is acceptable.

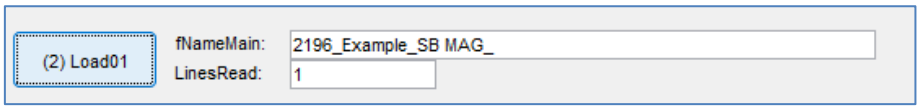


The command window displays a report on the execution of the SUSet command.

```
>> {'GUI'};MagT13;
SUSet --- setup was done
>>
```

4.2 Button (2) Load01

The command sequentially loads log-files for all magnetometers installed on ScanFish. Command parameters: main part of the magnetometers’ log-file name; range of processed lines from log-files.

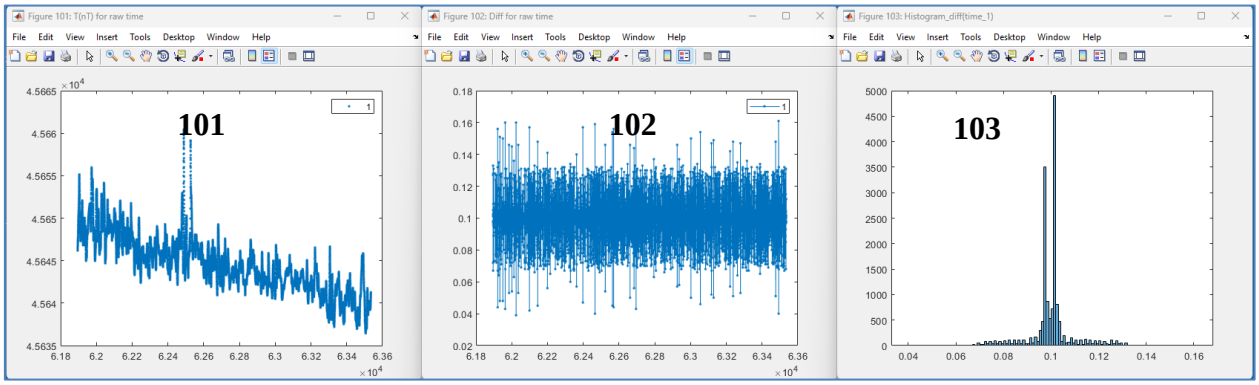


When loading the log-file of the first magnetometer, three graphical windows are opened (Figures 101, 102, 103):

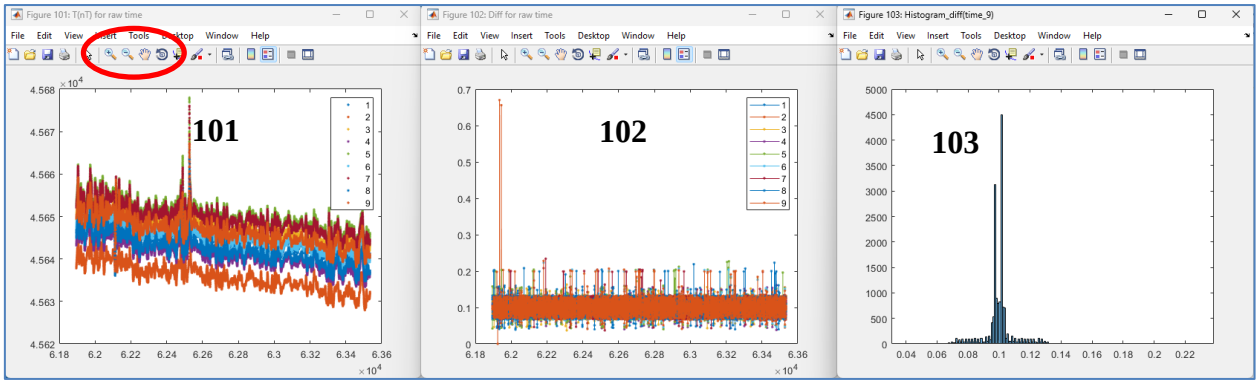
-- Figure 101 – Magnetic field graph for the first magnetometer. Magnetic field value is defined along the vertical axis and the number of seconds in a day is defined along the horizontal axis;

-- Figure 102 – Plots of the time-step between measurements, which is calculated as the time of the next measurement minus the time of the previous one for the first magnetometer. The vertical axis shows the time between adjacent measurements; the horizontal axis shows the number of seconds in day. Figure 102 illustrates the most points in the graphs located in the region of 0.1 second (the period of measurements of magnetometers set in hardware);

-- Figure 103 – Histogram of the step between measurements for the data of the first magnetometer. The horizontal axis shows the time step between adjacent measurements that occurs in the data sample. The vertical axis shows the number of measurements falling within a certain interval of “time step” values.



To start loading the log-file of the second magnetometer one should press the “Backspace” key. Graphical windows for the nine loaded magnetometers are shown below. Figure 101 and 102 “accumulate” the graphs of all previously loaded magnetometers. Figure 103 displays only the histogram for the last loaded magnetometer.



Here and below, you can use the tools in the graphical window panel to zoom in, zoom out, or shift the required area within the graphical window.

Simultaneously with drawing graphs, timestamp statistics are printed in the command window (see screenshot below).

```
>> {'GUI'};MagT13;
SUSet --- setup was done
Files loading started, press 'Backspace' to load next Mag...
======= Single mags statistics =======
No |    m_dt    | sd_dt/to%|    N
01 | 0.0999998 | 009.4450 | 016403
02 | 0.1000669 | 011.2090 | 016391
03 | 0.1000000 | 009.4247 | 016402
04 | 0.0999997 | 009.5086 | 016402
05 | 0.1001776 | 010.3202 | 016373
06 | 0.1001703 | 010.2164 | 016374
07 | 0.1001475 | 009.9411 | 016378
08 | 0.1001220 | 009.9206 | 016383
09 | 0.1002016 | 010.3304 | 016369
10 | 0.1001152 | 010.2011 | 016383
Load01 --- c:\temp\Data_examp\2196_Example_SB MAG_ loaded
>>
```

The statistics include the following columns:

-- sensor number;

-- arithmetic mean for the measurement step (the value is close to the hardware measurement period ToPeriod);

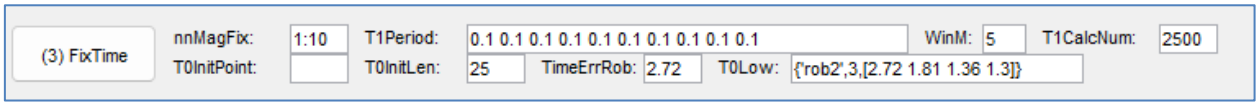
-- standard deviation for the measurement step divided by ToPeriod and multiplied by 100% (shows the approximate variation in the arrival time of the digital message from the magnetometer to QINSy);

-- number of magnetometer measurements in the file.

When completing the Load01 command in the command window, the following message is displayed: Load01 --- [name of file] loaded. Graphical window 103 will be closed.

4.3 Button (3) FixTime

The command sequentially corrects timestamps of data from the loaded magnetometer log-files. The command has a large number of parameters; under normal circumstances they do not need to be changed. The document, explaining the operation of the timestamp correction algorithm, describes these parameters in detail.

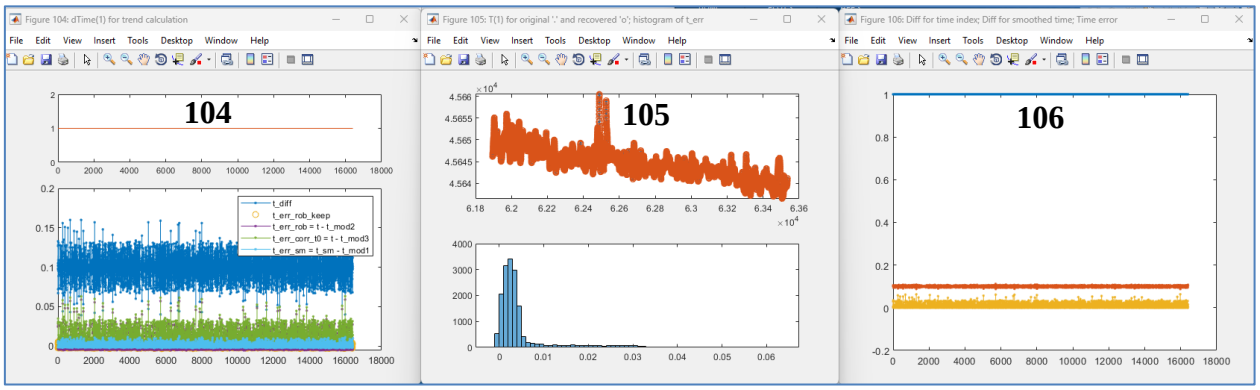


After correcting the timestamps of the first magnetometer, three graphical windows open (Figures 104, 105, 106):

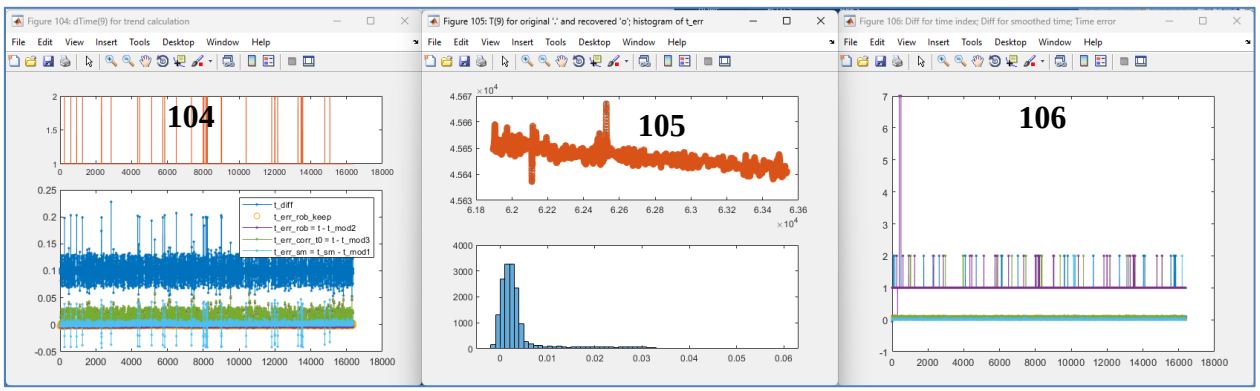
-- Figure 104 – The results of period selection and missing value search for the current magnetometer (the horizontal line with a value of 1 shown in the screenshot below means no "lost" measurements for the processed magnetometer);

-- Figure 105 – The graph of the magnetic field for the current magnetometer with initial timestamps, represented as points, and after correcting the timestamps, represented as circles, as well as a histogram of the correction values applied to the timestamps;

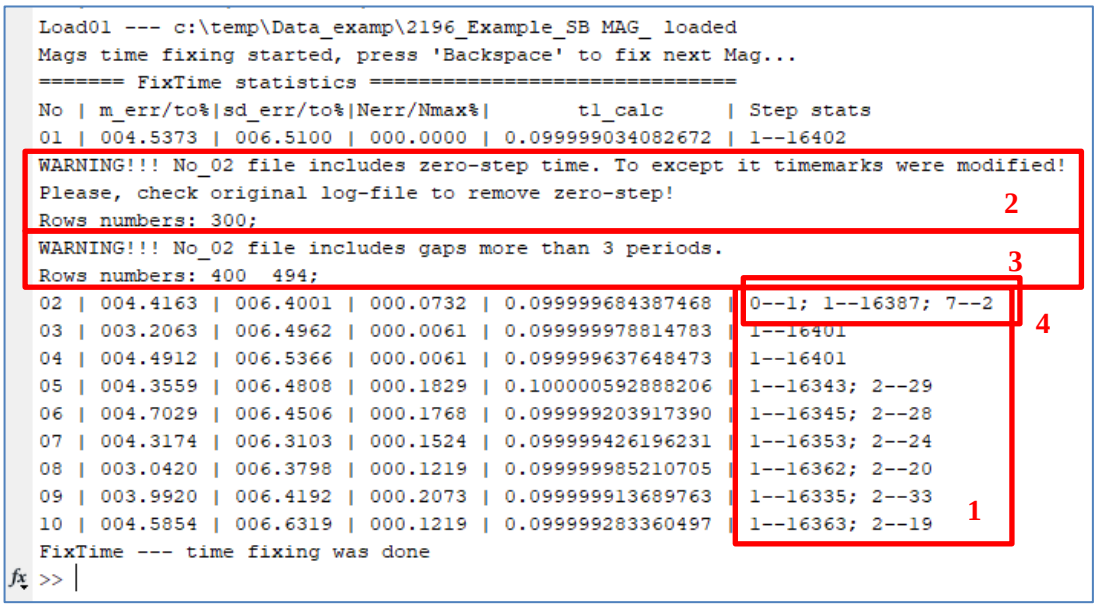
-- Figure 106 – The "cumulative" graphs showing the results of period calculation and missing value search for the timestamps of all previously processed magnetometers.



To start correcting the timestamps of the second magnetometer one should press the "Backspace" key. The graphical windows after correcting the time of nine magnetometers are shown below. Figures 104 and 105 display the results for the last loaded magnetometer only. Figure 106 "accumulate" the graphs of all previously loaded magnetometers.



Simultaneously with drawing graphs, the command window prints statistics (see screenshot below) on deviation of “raw” timestamps from the series with uniform step, created as a result of delay correction.



The statistics include the following columns:

-- the sensor number;

-- the error magnitude in relation to the measurement step ToPeriod in percent (calculated as the arithmetic mean of the time correction value divided by ToPeriod and multiplied by 100%);

-- the variation of the arrival time of the digital message from the magnetometer to QINSy in relation to the ToPeriod in percent (calculated as the standard deviation of the time correction value divided by ToPeriod and multiplied by 100%);

-- the percentage of lost measurements in the record (the number of lost magnetometer measurements is divided by the maximum number of measurements from all magnetometers, and then multiplied by 100%; the result is then subtracted from 100%);

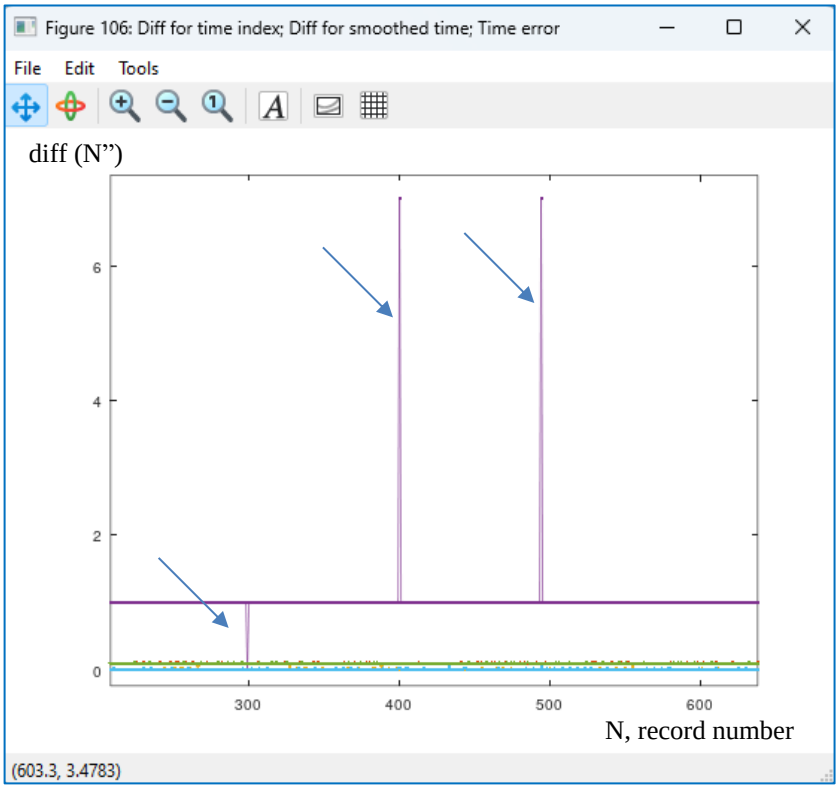
-- the “exact actual period” of the T1Period magnetometer, determined as a result of calculations;

-- the statistics on the lost magnetometer measurements (1; the number of periods between the adjacent measurements, i.e. the size of the “gap” and, with a hyphen, the number of such gaps in the record, are printed).

The magnetometer number two, in the 2196_Example_SB MAG_02.txt file, had several lines intentionally deleted in order to create two “gaps”, and a copy of a single line was added to create measurements with “repeating time”. These “features” were detected during the calculation of statistics, and a warning is printed in the command window for each of them (2, 3). The warning indicates the number of the line in the file where the “gap” or “repeated value” is located (red arrows). Also, the “step value between adjacent measurements” is printed in the statistics: “0--1” means a repeating time; “7--2” means two gaps of 6 magnetometer measurements (4).

These features are displayed not only in the command window, but also Figure 106 represents them as spikes in the plots of the “difference between measurement numbers, including missing measurements”. A zoomed part of Figure 106 for the second magnetometer data is shown below. The graph displays that

one value of the measurement number is repeated (the difference is 0), and in two cases the difference is 7 instead of 1 (in these cases 6 measurements are “lost”).



For a more detailed description of the graphical windows displayed by the TimeFix command, refer to the separate chapter on the analysis of the algorithm for correcting timestamps.

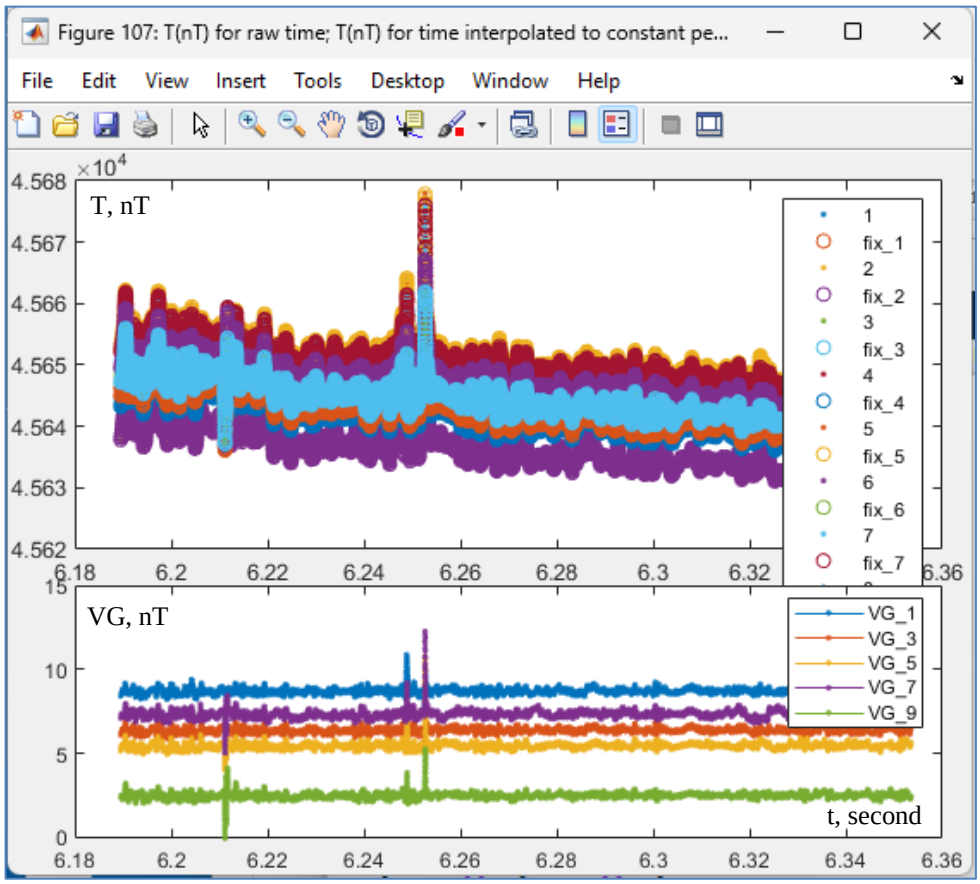
When the TimeFix command is completed, the command window displays the following message: FixTime --- time fixing has been done. Graphical windows 104 and 105 will be closed.

4.4 Button (4) Interp

The command interpolates measurements of magnetometers and other sensors to a series of common timestamps. The common time series is generated with the ToPeriod measurement period specified for the hardware.

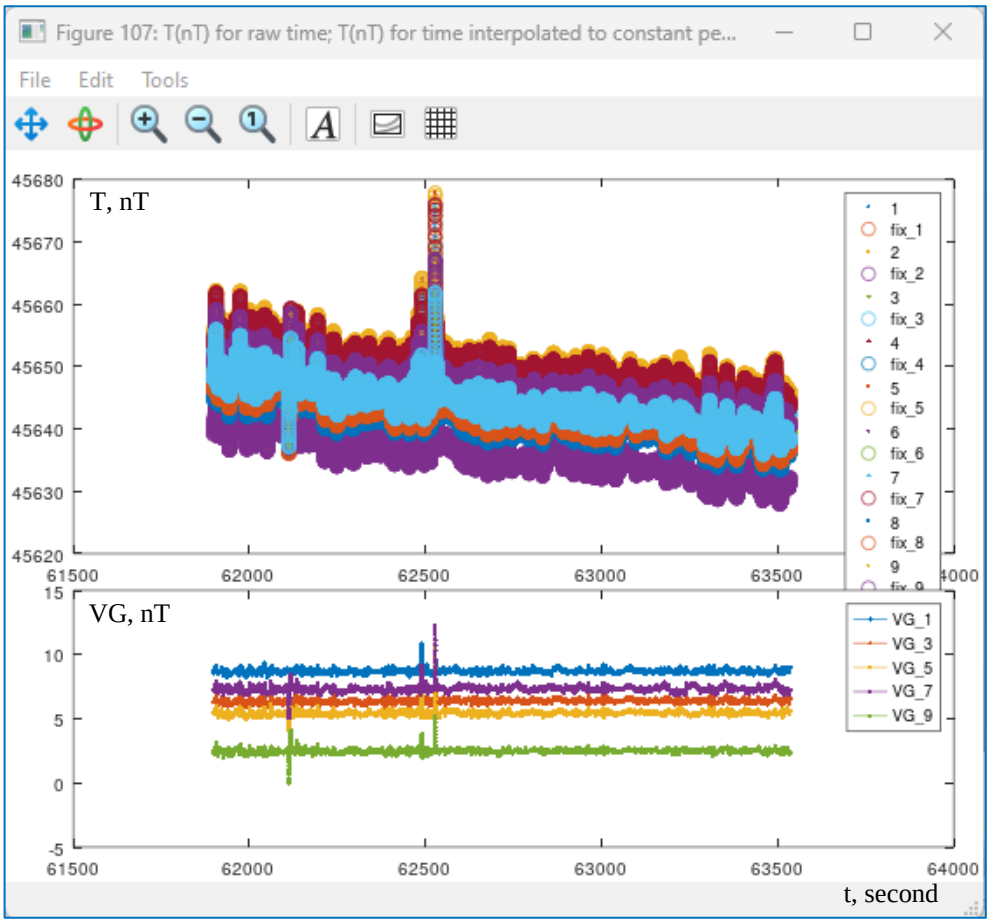
The log-files of magnetometers contain measurements from the same “sensors”; for example, ship stern coordinates, ship heading, etc. In the command parameters one should, select the magnetometer whose records will be used to interpolate such “sensors” into a common data series.

After performing the interpolation, the Figure 107 graphical window opens (the MatLab window view is shown below).

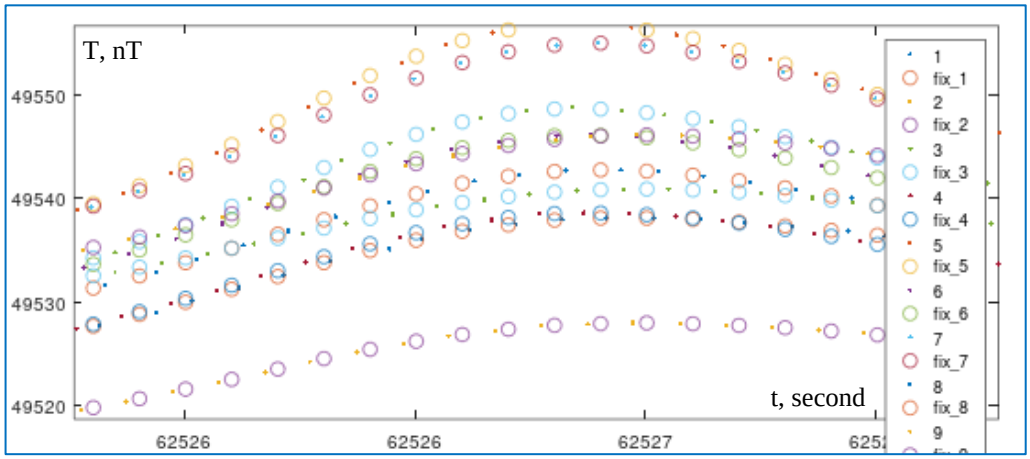


The horizontal axis of the graphs shows the second in day. The upper part of the window has magnetic field plots for “initial timestamps” and magnetic field plots for “corrected timestamps and interpolated magnetic field” drawn (after correcting the time and interpolating the measurements to a single series of timestamps). The lower part of the window has vertical magnetic field gradient plots (the difference between the ScanFish sensor measurements) plotted using the magnetic field with corrected timestamps.

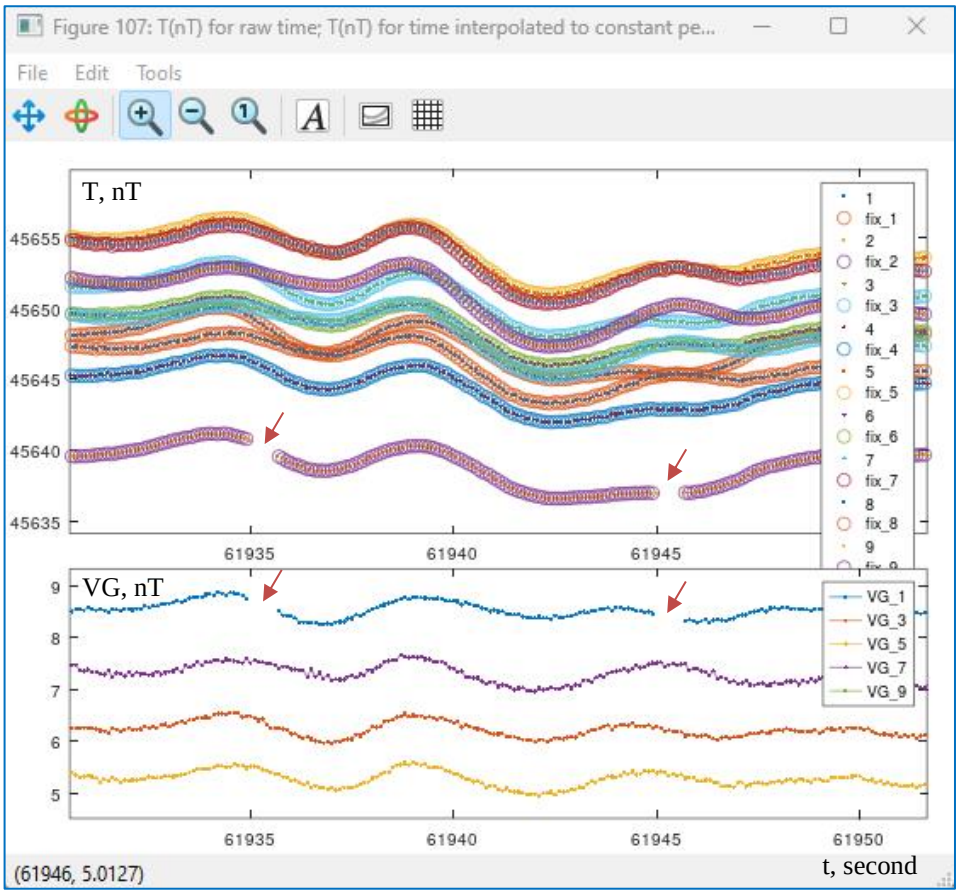
Below is the view of the Figure 107 window in Octave.



Graphs for “initial timestamps” are drawn as points, while for “corrected timestamps and interpolated magnetic field” – as circles. The graphs facilitate to trace the mutual arrangement of points and circles (repeatability of the graph shape for each magnetometer).



In the case of consecutive missing measurements exceeding the number specified in GapAccept (when executing the SUSet command), the interpolation results are replaced by NaN, and the graph displays the gaps. The graphs with two gaps are shown below in the file 2196_Example_SB MAG_02.txt.



When completing the Interp command, the following message is displayed in the command window: Interp --- interpolation has been done.

```
09 | 003.9920 | 006.4192 | 000.2073 | 0.099999913689763 | 1--16335; 2--33
10 | 004.5854 | 006.6319 | 000.1219 | 0.099999283360497 | 1--16363; 2--19
FixTime --- time fixing was done
Interp --- interpolation was done
>>
```

4.5 Button (5) Save01

The command saves the data of all magnetometers, recalculated to a general series of timestamps, as a single file. The command uses the following parameters:

-- number of characters removed from the end of the “main part of the file name”;

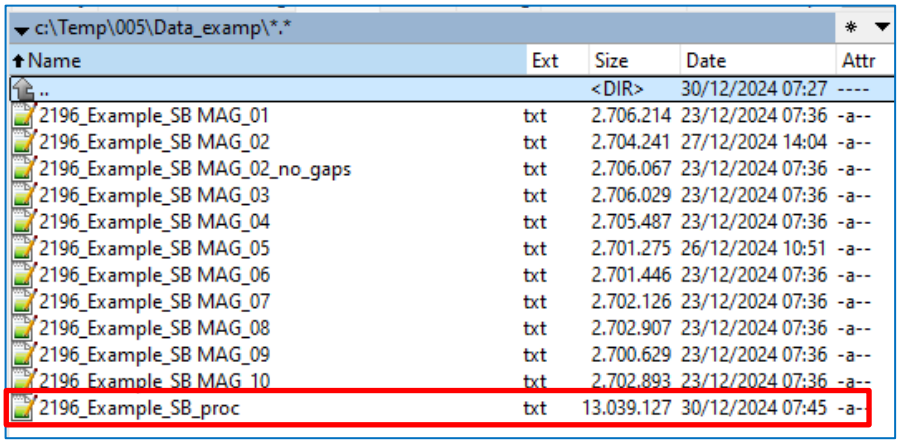
-- postfix and extension added to the end of the name of the file to be saved;

-- flag for recording timestamp statistics in the header of the file to be saved.

At the beginning of recording, the command window displays the following message: Output file is saving, please wait… In MatLab, file recording takes several seconds, while in Octave it takes from 10 seconds to several minutes; during this process, it is recommended not to press any buttons in GUI to avoid “unexpected execution of the commands” after completing the file recording. When the file recording is completed, the following message appears in the command window: Save01 --- output file was saved.

```
Interp --- interpolation was done
Output file is saving, please wait...
Save01 --- output file was saved
>>
```

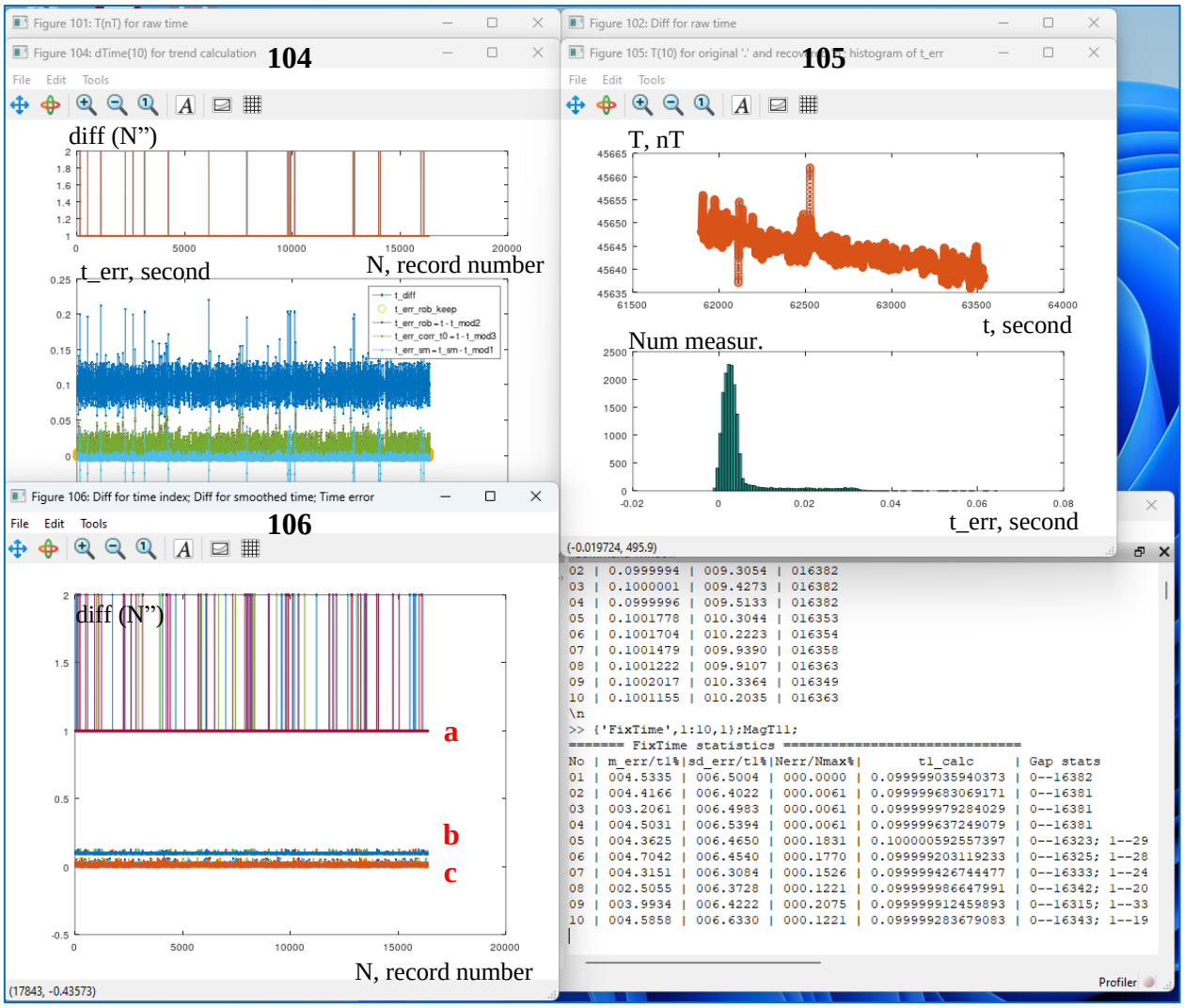
Executing the working sequence of commands to obtain a single log-file for all magnetometers is completed. A single log-file with measurements, reduced to a single time series of observations, will appear in the folder with log-files of magnetometers.



5. Description of the graphical windows displayed by the FixTime command

When running the FixTime script command, three graphical windows are displayed. Figures 104 and 105 are redrawn for each "current" magnetometer processed by the FixTime command. Figure 106 accumulates the graphs for all the processed magnetometers.

In Figures 104 and 106, the horizontal axis shows the magnetometer measurement number without lost points (it is equal to the line number in the log-file).

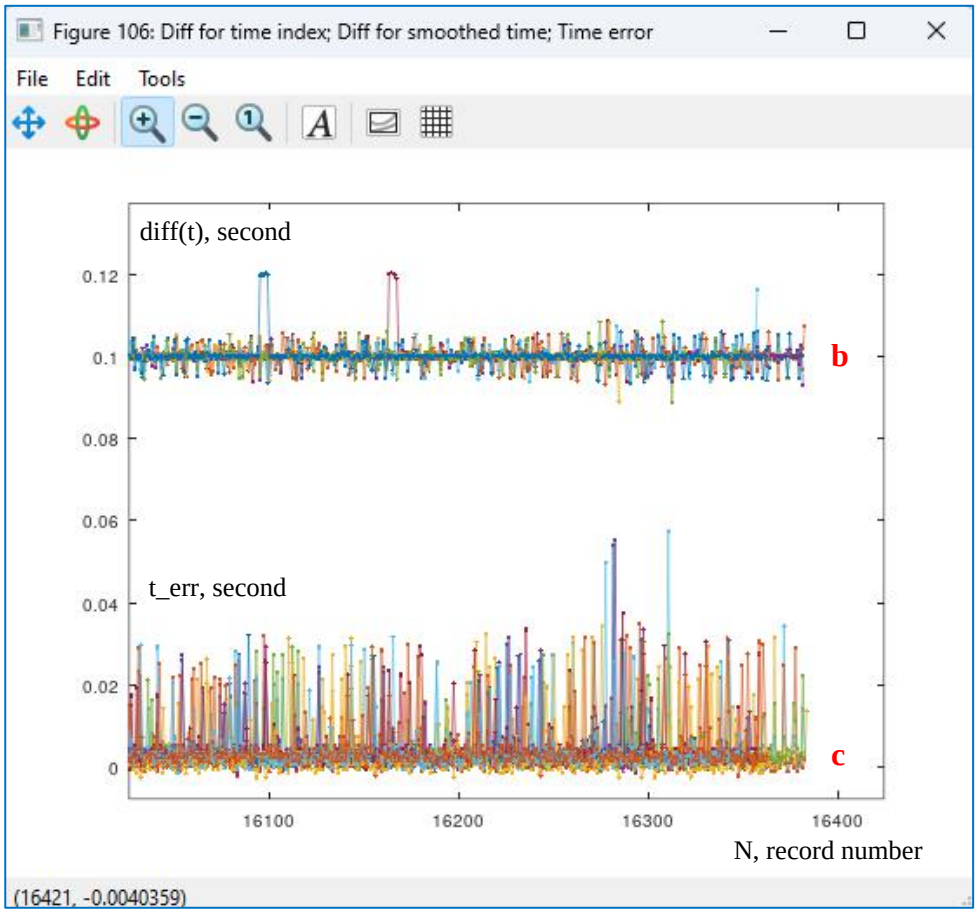


The graphs in Figure 106 show the results of searching for the "lost" values and bringing the measurements of each magnetometer to a constant step (to the constant magnetometer period). The graphs are categorised into three groups (signed as "a", "b", "c"). The "a" group displays the difference between the measurement numbers (subsequent minus previous) at a constant step (missing measurements are also assigned numbers). For example, for measurements with numbers 102, 103, 104, 106, 107, a graph is drawn with values 1, 1, 2, 1. This graph reveals that one point is missing (the difference value equals 2). Provided the ScanFish-QINSy digital channel is configured correctly, the number of "skips/gaps in a row" in the data should not exceed one measurement, i.e. the values in the graphs of the first group should not exceed 2. The difference value greater than 2 occurs rarely; for example, as a result of a short-term "hang-up" of QINSy or interruption of data flow from ScanFish.

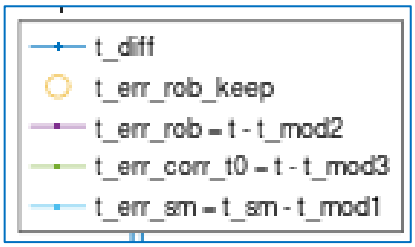
The “b” graph group shows the time difference between the subsequent and previous points, similar to the graphs in Figure 102; however, while in Figure 102 the difference was calculated for the “raw” time, for the “b” group in Figure 106 the difference is calculated for the time smoothed using a sliding window (the width of the smoothing window is set in the WinM in the SUSet command).

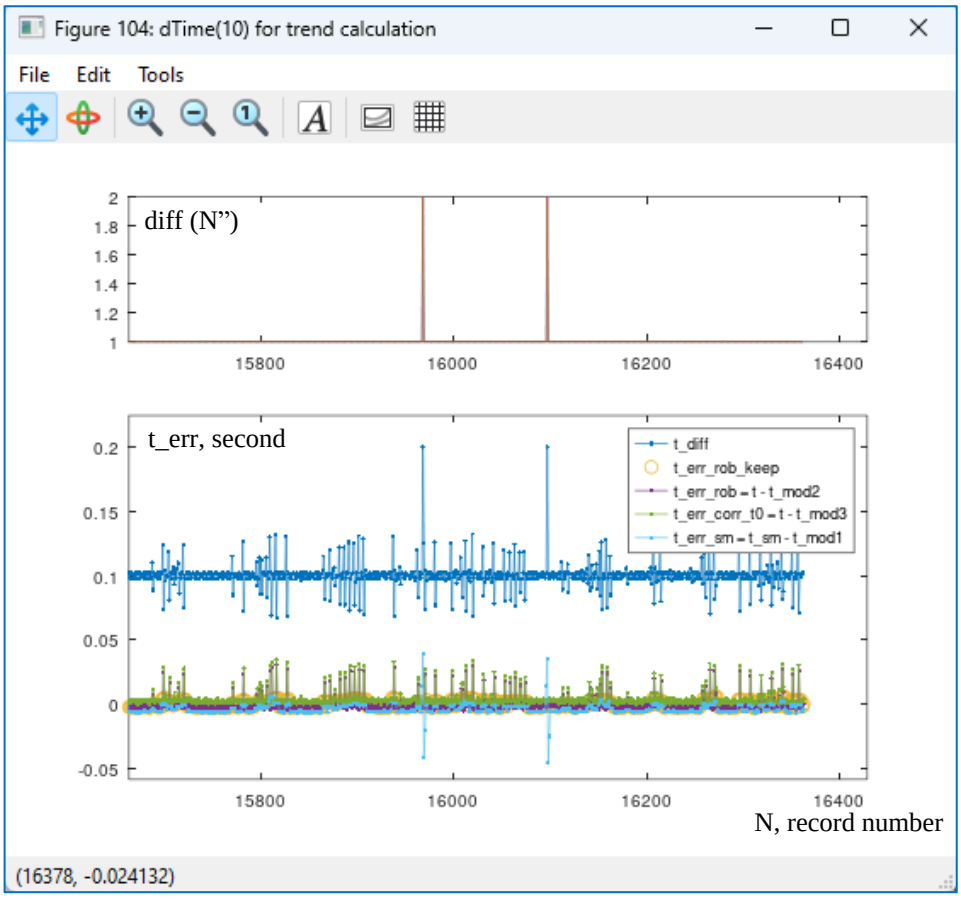
A zoomed part of graphs for “b” and “c” groups is shown below. Most of the graph points are concentrated in the proximity of the magnetometer measurement period value of 0.1 second. The rectangular meander in the plots marks the “lost” measurements.

The “c” graph group shows the difference between the “raw time” of magnetometer measurements and the “corrected” time, which has the timestamps brought to a constant step.



The graphs in Figure 104 are the main quality control (QC) tool. Their captioned legend is shown below for reference.





Legend designations:

t_diff. The time step between adjacent measurements of the magnetometer (same as shown in Figure 101).

t_err_sm. The error magnitude calculated as the difference between the smoothed time (the width of the smoothing window is set in the WinM) and the modelled time (first approximation model) of the magnetometer measurements.

t_err_rob_keep. The **t_err_sm** sample measurements after removing the elements associated with missing magnetometer measurements.

t_err_rob. The error magnitude calculated as the difference between the "raw" time of the magnetometer measurements and the modelled time (second approximation model).

t_err_corr_t0. The error magnitude calculated as the difference between the "raw" time of magnetometer measurements and the modelled time (third approximation model). The third approximation has an additional correction (shift) to the time, which is introduced to meet the following condition: "a digital measurement message cannot be received by QINSy before it is sent by the magnetometer".

Consider the enlarged section of the graphs in Figure 104 above. As one can see, the "lost" measurements correspond to "peaks in both directions" in the t_err_sm graph. As the width of the averaging sliding window increases, the width of the zones encompassing these peaks increases as well. Therefore, it is recommended to keep the averaging window as narrow as possible. However, given a significant "time shift for measurements" (delays in the data channel), it may be necessary to increase the averaging window to 7, 9, or even 11 points.

The upper part of Figure 104 shows the difference between the measurement numbers (subsequent minus previous) at a constant step; missing measurements are also assigned numbers. The same differences (but with “accumulation” for all processed magnetometers) are shown in Figure 106.

To understand the behavior of the graphs in Figures 104 and 106 in detail, one should study the description of the algorithm operation. To perform a quality control of the measurement time, it is sufficient to follow two rules:

1) The difference graph in the upper part of Figure 104 should not exceed the value of 2 (if it exceeds this value, one needs to understand the reasons of the “skipping” in the magnetometer data flow);
2) The trend direction for the t_err_sm graph (indicated by the arrows in the figure below) should be close to the horizontal. This means a good automatic selection of the “approximate actual period” of magnetometers relative to the values set in T1Period (parameters of the FixTime command). If the position of t_err_sm is close to the horizontal (with the graph shift of no more than 0.02 sec at the end, with the measurement period of 0.1 sec), the algorithm finding the position of the “missing” values and calculating the “exact actual period” works well.

Appendix 1 describes the manual estimation of “approximate actual periods” used in early versions of the script.

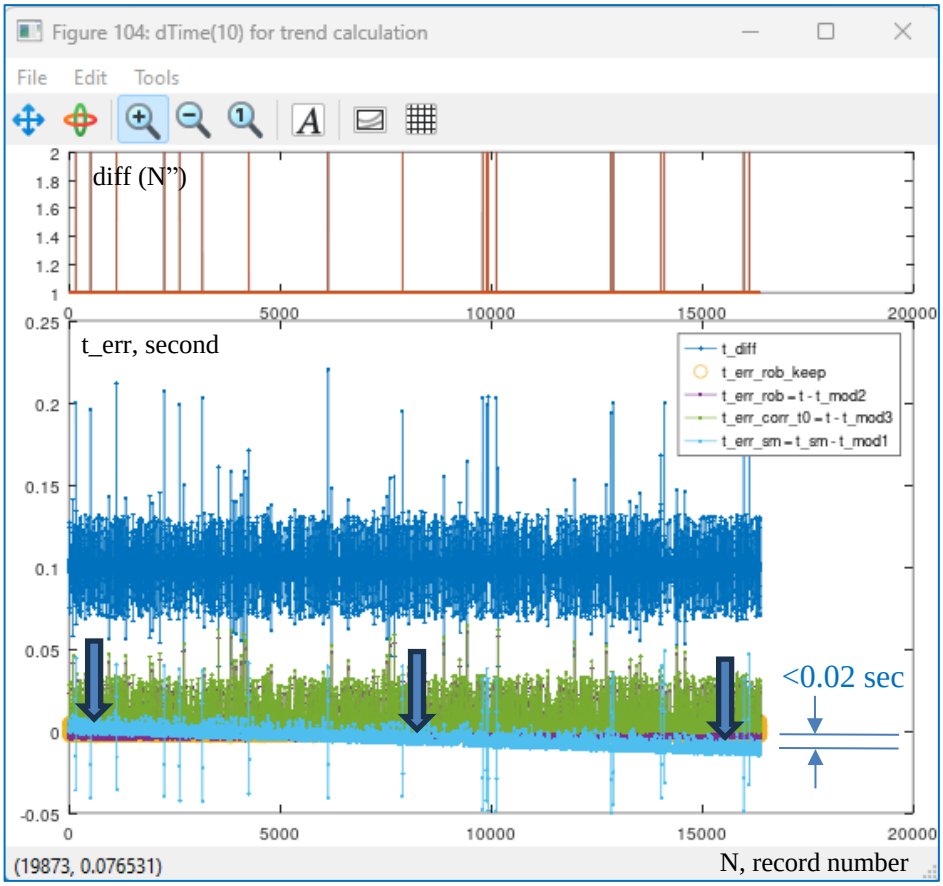
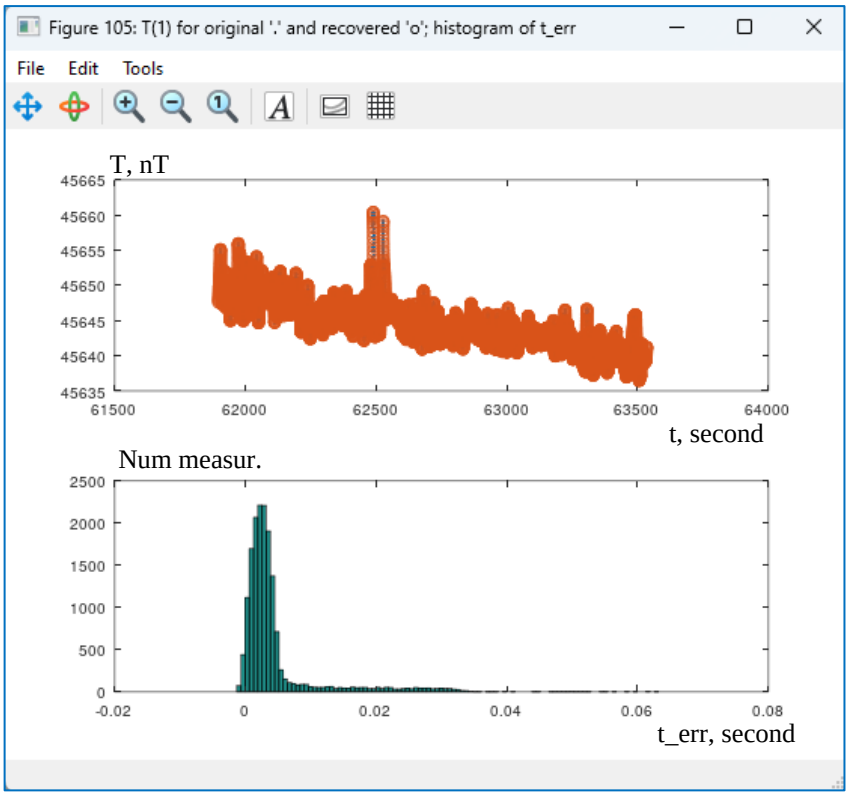
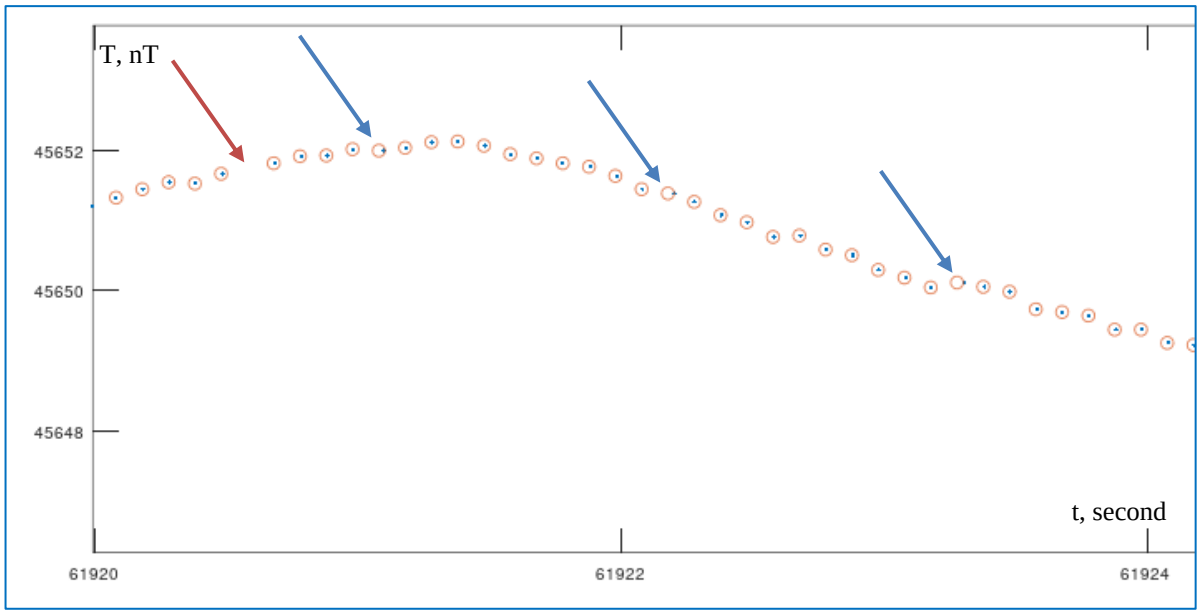


Figure 105 shows two plots of the magnetic field: the first displays the original timestamps (dots), while the second one shows the corrected timestamps (circles). The figure also presents the frequency histogram, which illustrates the difference between the original and corrected timestamps.



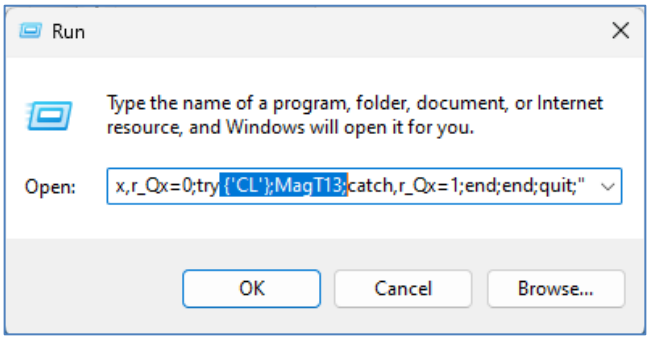
When zooming a section in the magnetic field plots, one can see the gaps corresponding to the “lost measurements” (the red arrow in the figure below) and measurements with delays in raw timestamps (the blue arrows below).



6. Running the script on schedule (MagT13_run)

There is a function for “automatic data processing”, which runs the script from the command line. Appendix 4 shows the Task Scheduler settings for Windows to run the script on a schedule.

When running the script from the command line, the {'CL'};MagT13; key is used.



The function code analyzes files in the specified folder, finds log-files which contain no “processed file”, and then performs their processing.

The code which starts the processing from the command line in MatLab:

matlab -r "r_Qx=1;while r_Qx,r_Qx=0;try {'CL'};MagT13;catch,r_Qx=1;end;end;quit;"

The code which starts the processing from the command line in GNU Octave (one must set the correct path to the octave.exe file):

c:\Program Files\GNU Octave\Octave-9.2.0\mingw64\bin\octave --eval "r_Qx=1;while r_Qx,r_Qx=0;try {'CL'};MagT13;catch,r_Qx=1;end;end;quit;"

All parameters of the script commands are set in the MagT13_run function code (the screenshot below shows the lines with parameters) and **must be corrected before running**.

```
59 -        GUI.gui.RootDir='c:\temp\Data_examp\';
60 -        GUI.gui.MagN='10';
61 -        GUI.gui.ToPeriod='0.1';
62 -        GUI.gui.GapAccept='3';
63 -        GUI.gui.fNameMain='2196_Example_SB MAG_';
64 -        GUI.gui.LinesRead='1';
65 -        GUI.gui.nnMagFix='1:10';
66 -        GUI.gui.T1Period='0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1';
67 -        GUI.gui.WinM='5';
68 -        GUI.gui.T1CalcNum='2500';
69 -        GUI.gui.T0InitPoint='';
70 -        GUI.gui.T0InitLen='25';
71 -        GUI.gui.TimeErrRob='2.72';
72 -        GUI.gui.T0Low='{''rob2'',3,[2.72 1.81 1.36 1.3]}';
73 -        GUI.gui.BMagInterp='1';
74 -        GUI.gui.fNameDelE='5';
75 -        GUI.gui.fNamePostfix='_proc.txt';
76 -        GUI.gui.fFlagStat='0';
77 -        GUI.gui.QcBitR='0';
78 -        GUI.gui.QcTr='{''poly'',1}';
79 -        GUI.gui.QcExtr='{''linear'',''extrap''}';
80 -        GUI.gui.QcSmoo='{''gDataMovmeanD'',101}';
81 -        GUI.gui.Qcm='1';
82 -        GUI.gui.QsBitR='0';
83 -        GUI.gui.QsTr='{''const'',0}';
84 -        GUI.gui.QsExtr='{''linear'',''extrap''}';
85 -        GUI.gui.QsSmoo='{''none''}';
86 -        GUI.gui.Qsm='T';
```

Here are some details about the operation of the MagT13 function with {'CL'} key:

1) Identification of the “log-file with single magnetometer record” is done by the last two characters in the name (2197_Example_SB MAG_01.txt). In this case, only the presence of the file of the first magnetometer (characters “01”) is taken into account.
2) Identification of the “log-file with merged data” is done based on the postfix (2196_Example_SB_proc.txt).
3) The next survey line files to be processed require an empty zero-volume “with processing result” file to be created before processing starts. This is done to ensure that if the script gives an error (“cannot process” the line), it will not be accepted for processing when the script restarts. If there is a processing error, the script proceeds with the processing of the next survey line.

Conclusion

MagT13 script facilitates to perform various operations with magnetometer log-files:

-- injecting a corrected navigation to magnetometer measurements;

-- correcting timestamps of magnetometer measurements;

-- interpolating coordinates and magnetometer measurements to a general time series and saving data to a single log-file.

The main purpose of the script is correcting the timestamps of the magnetometers (recalculate them from “receiving time of the QINSy digital message” to “time of measuring by the magnetometer”) and creating a single log-file from the log-files of individual magnetometers, containing the readings of all sensors, which are recalculated to a general sequence of timestamps. The calculations performed by the script in this case consist of two steps:

1) Determining the exact period and the initial time shift of the measurements from each magnetometer and creating a “corrected” sequence of timestamps with a constant time step for the measurements from each magnetometer (TimeFix command);
2) Creating a sequence of timestamps common to all sensors and interpolating the data from all magnetometers and all coordinates to the common sequence of timestamps (Interp command).

The timestamped “measurement sequences”, provided as input to the TimeFix command, must possess certain properties:

1) Magnetic sensors must send measurements to the digital channel with a period that is extremely constant in time (the algorithm is designed for a constant period; failure to meet this condition requires algorithm modification associated with estimating the “trend of the measurement period” in time).
2) The “lost measurement” event should be sufficiently rare to ensure that a statistically significant number of timestamps contains no “lost measurements” within the sliding window (WinM) during averaging;
3) “Lost measurements” should not occur consecutively: the number of “lost measurements” within the sliding window must not exceed one. In some cases, violation of this rule may cause the measurements adjacent to the “lost” ones to be assigned to timestamps with a one-period error.
4) The spread of measurement timestamps should not exceed half of the measurement period, which is achieved by optimizing the data-transmit channel settings.

The corrected sequence of timestamps has the “minimum data delay in the measurement channel” remaining uncompensated for each magnetometer and defined as the minimum time required to transmit a digital message from the moment of measuring to the moment when QINSy receives the data. This delay is estimated to be approximately 0.05 seconds if the data channel is well set up.

Appendix 1: Manual correction of the measuring period initial value for magnetometers

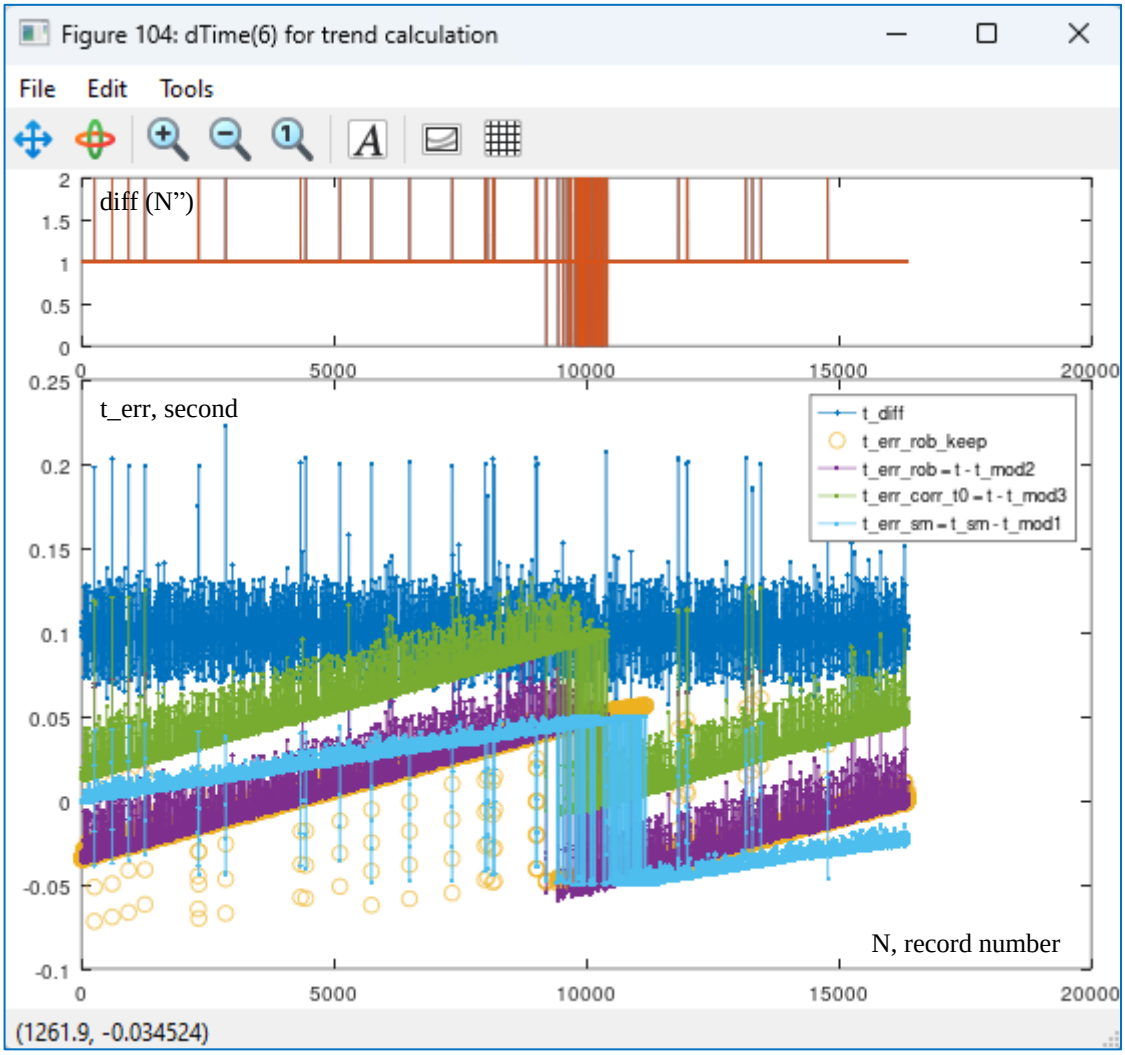
In the MagT10 version of the script, the values of “approximate actual periods” T1Period had to be set with sufficiently high accuracy, which required manual correction. In the next versions the selection of “approximate actual periods” started to be performed automatically; the values of these periods are currently set equal to the period defined in the equipment settings (e.g. T1Period=[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1]).

Nevertheless, the ability to manually correct the “approximate actual periods” has been retained. The following step by step description provides an example of manual correction.

1) To switch to the manual correction mode, set T1CalcNum to NaN (FixTime parameters).

2) To work with the 2196_Example_PS MAG_ test files, replace the T1Period value with [0.0999995, 0.1000000, 0.1000003, 0.0999999, 0.1000001, 0.0999946, 0.0999980, 0.0999996, 0.0999999, 0.1000001].

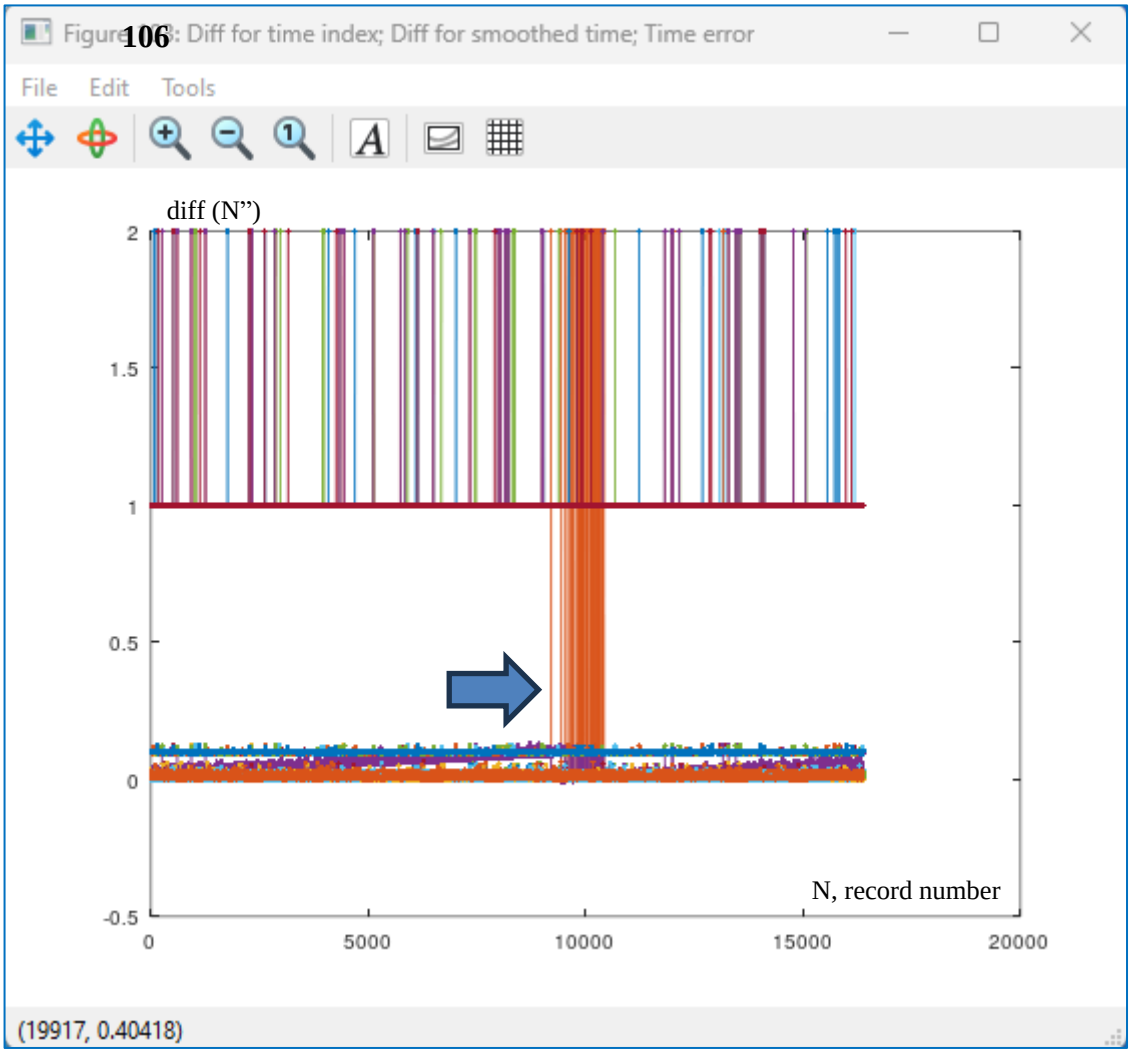
3) Run FixTime and then process magnetometers up to and including the 6th magnetometer. When the 6th magnetometer is processed, the Figure 104 window will display the figure shown below.



The period of 0.0999946 seconds, which we introduced for the 6th magnetometer, led to the accumulation of error due to the incorrect value of the “approximate actual period”. As a result, the

accumulated time shift error exceeded half of the measurement period. Eventually, the subsequent measurements "jumped one step forward" relative to the model with a uniform step. This is illustrated in the blue t_err_sm plot as the graph "jumping" from an error value of 0.05 seconds to -0.05 seconds. **The results of calculations performed by the script will be erroneous with such a view of the t_err_sm graph**. An additional indicator of erroneous calculations will be the appearance in the command window of a line showing the numbers of measurements for which the time is "duplicated" (has the same value).

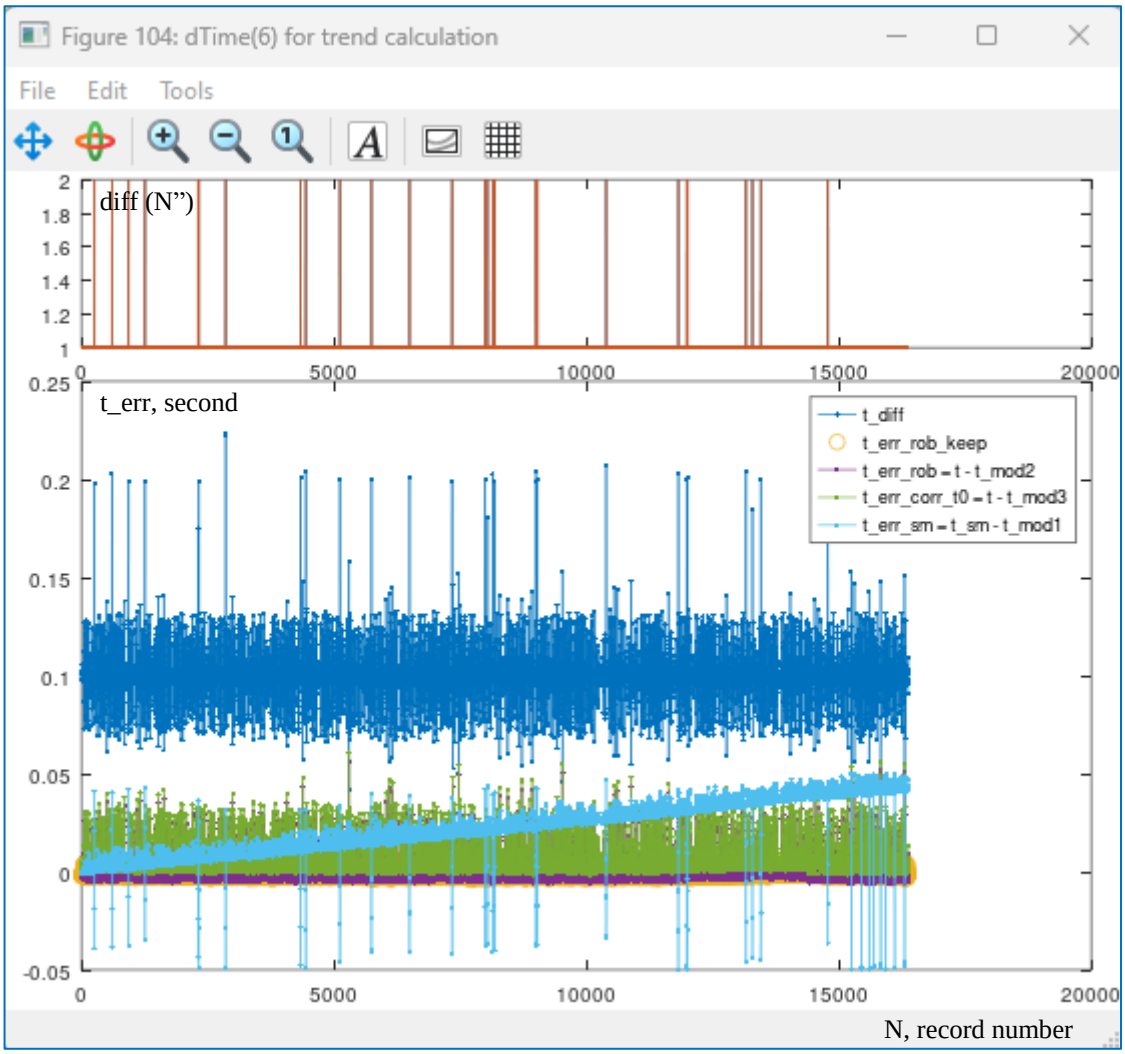
Figure 106 shows the characteristic graph displaying two-sided spikes associated with a zero step between the adjacent measurements. Although the script has saved the results of calculations to a file, **these results are incorrect!** Therefore, if a strong t_err_sm slope appears in Figure 104, one should interrupt the script execution using the Ctrl+C key combination and correct the initial values of the periods.



4) Complete or interrupt executing the FixTime command.

5) Change the value of T1Period. Since the graph in Figure 104 "went up", we must increase the measurement period to compensate for the error. Change the period value of the 6th magnetometer to 0.099996.

6) Run the FixTime command and then process the magnetometers up to and including the 6th magnetometer. The Figure 104 window will display the figure shown below.



The figure shows that the t_err_sm graph is still "going up", i.e. the period value has not been increased sufficiently.

7) Complete or interrupt executing the FixTime command.

8) Change the period value of the 6th magnetometer to 0.099999**6**. Run the FixTime command and then process the magnetometers up to and including the 6th magnetometer. The Figure 104 window will display the figure shown below. The deviation of the trend in the t_err_sm plot from 0 does not exceed 0.01 second, indicating that the period selection for the 6th magnetometer is performed with sufficient accuracy.

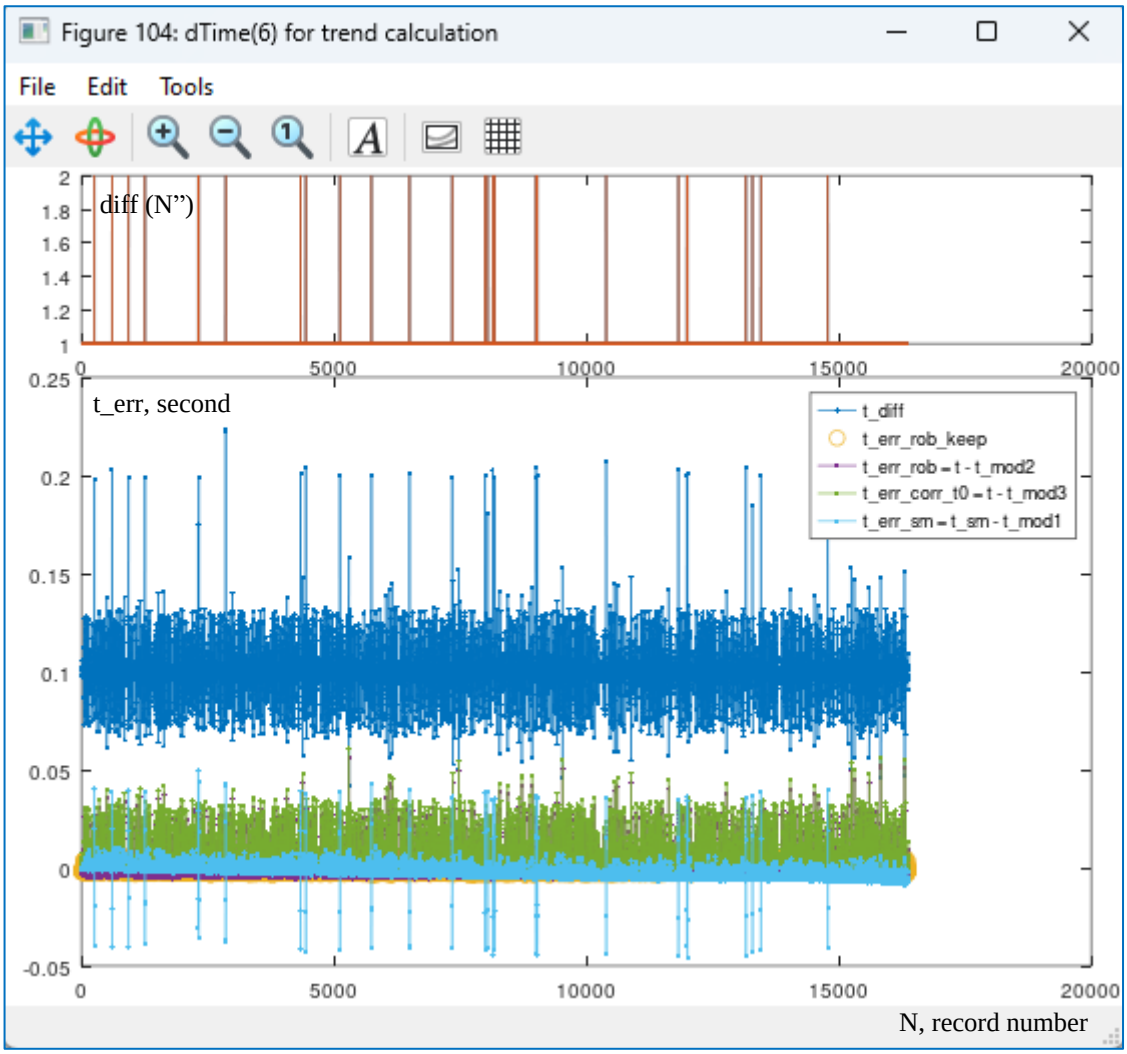
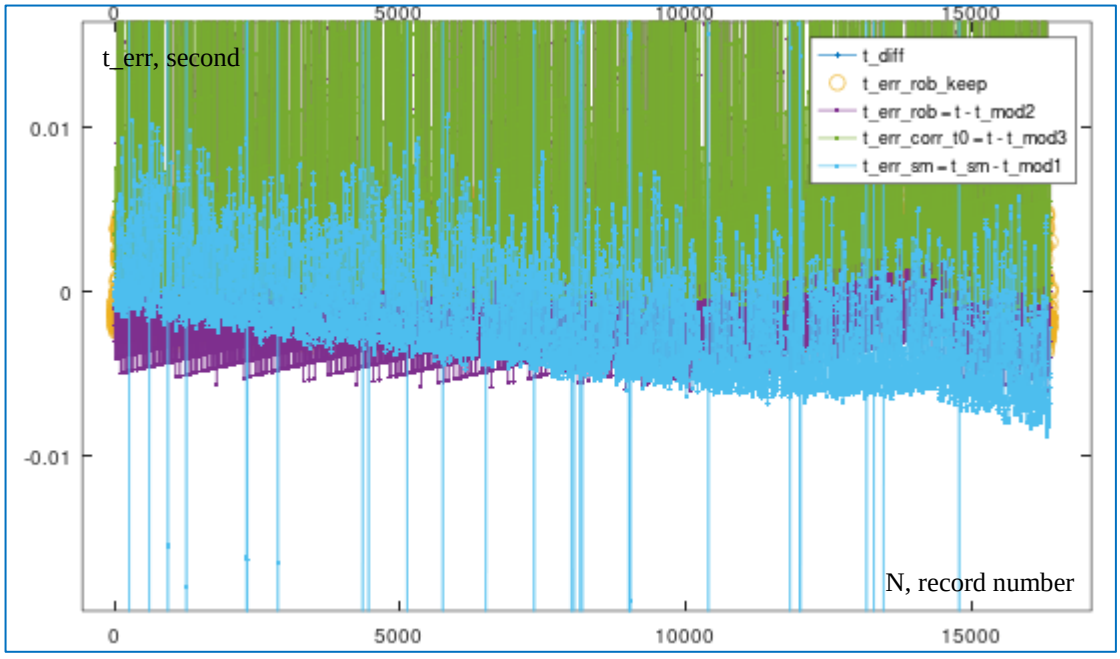


Figure 104: dTime(6) for trend calculation
File
Edit
Tools
diff (N")
t_err, second
t_diff
t_err_rob_keep
t_err_rob = t - t_mod2
t_err_corr_t0 = t - t_mod3
t_err_sm = t_sm - t_mod1
N, record number


t_err, second
t_diff
t_err_rob_keep
t_err_rob = t - t_mod2
t_err_corr_t0 = t - t_mod3
t_err_sm = t_sm - t_mod1
N, record number

9) Displaying the changed magnetometer period does not require processing all "previous" magnetometers each time. It is possible to set nnMagFix value equal to 6 and process the 6th magnetometer with FixTime command. By separately changing the period for each magnetometer using the method described above, you can quickly perform initial selection of accurate periods for all magnetometers installed on ScanFish.

As the recording duration of measurements increases, the "approximate actual periods" of magnetometers should be set with increasing accuracy. Given recording time of 4 hours with 10 Hz frequency, it is sufficient to set the period to the sixth digit accuracy.

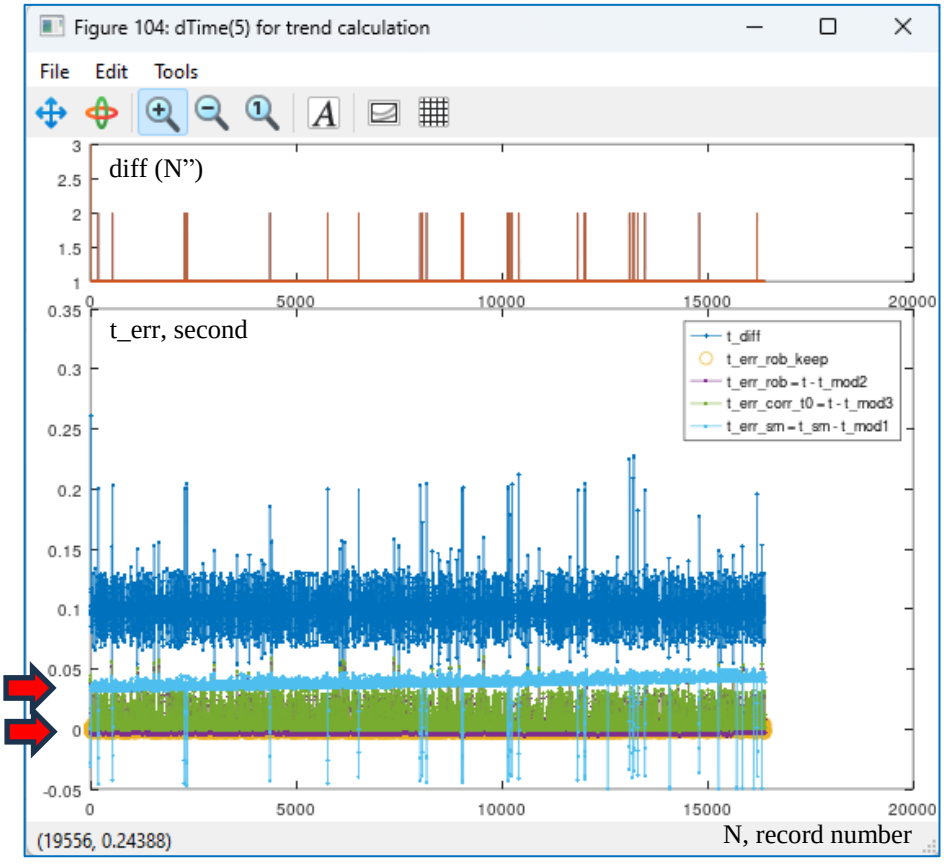
Period correction is required mainly in the following cases: 1) at the beginning of works, when mobilising; 2) when replacing the magnetometer on ScanFish; 3) when increasing the length of survey lines (the period value is "refined"); 4) when changing the survey conditions, when the internal "operating mode" of the magnetometer changes.

Appendix 2: Manual correction of the “zero-timestamp shift” value for the magnetometer

If the point of “the initial shift of the zero-timestamp estimation” falls within the proximity of the “lost measurement” (to a spike of the difference between “the arrival time of adjacent measurements”), the algorithm for estimating the time shifts may begin to malfunction. In the MagT10 script version, such cases required manual correction. The algorithm was modified to automatically correct the position of the “the initial shift of the zero-timestamp estimation” point. Nevertheless, there may be rare cases when the algorithm fails, and manual change of the script parameters is required. The following section describes the case of a “zero-timestamp point” on a spike of period difference (in the “synchronization error” graphs) and how to perform a manual correction.

1) In the Load01 settings, set the LinesRead=1. Execute the Load01 command.

2) Change the value of T0InitPoint to 5 and the value of T0InitLen to NaN in the TimeFix settings (to disable automatic correction of the “initial shift estimation” point position). Run the FixTime command and then process magnetometers up to and including the 5th magnetometer. When the 5th magnetometer is processed, the Figure 104 window will display the figure shown below.



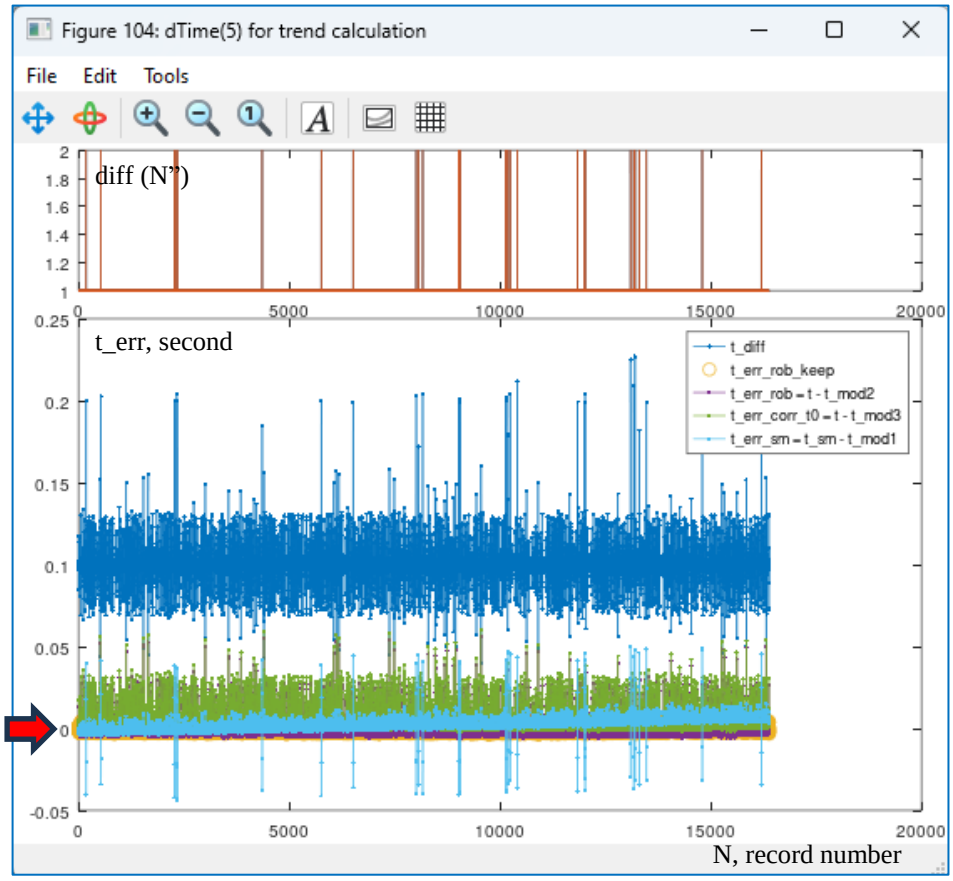
The initial data appeared to be such that when determining the “initial zero-time shift” for the magnetometer number five, the “zero timestamp” fell within the “lost measurement” region. This resulted in a significant shift of the t_err_sm plot relative to zero. This shift can cause the algorithm to malfunction (incorrectly determining the position of the missed measurements). Furthermore, even a minor slope of the t_err_sm graph can result in points in the graph “jumping” from values of 0.05 to values of -0.05 seconds.

The described "shift" of the t_err_sm graph should always be removed. In order to remove the shift, it is sufficient to skip a few points at the beginning of the record.

3) Complete or interrupt executing the FixTime command.

4) In the Load01 settings, set the LinesRead=5. Run the Load01 command.

5) Run the FixTime command and then process magnetometers up to and including the 5th magnetometer. The Figure 104 window will display the figure shown below. The shift relative to zero has disappeared.



Appendix 3: Magnetometer log-file formats

Format of individual magnetometer log-files loaded by commands Load01, Load01_rp, Load01_nc, Load01_sy

The following example shows the header and two lines of the file with a record of a single magnetometer:

Storage Db,Date,Time,[Mainline] Name,CoG Easting,CoG Northing,3049_SB scnafish MRU Pitch,3049_SB scnafish MRU Roll,3040_SB HDT Fram Value,SB MAG_01_PL Easting,SB MAG_01_PL Northing,3040_SB HDT Fram Value,Magnetic Field 3041_MAG01-SB [nT] Value,Magnetic Field 3041_MAG01-SB [nT] Quality,Mag Depth Value,Mag Altitude Value,SB Scanfish CoG Range to Node,SB Scanfish CoG Range to Node

0268_131 - 0001.db,01/02/2099,20:06:29.183,L131,439994.22,5899995.08,-5.37,-0.77,326.83,439997.94,5899992.29,326.83,49376.53,1382.00,32.15,3.01,186.84,190.14

0268_131 - 0001.db,01/02/2099,20:06:29.285,L131,439994.01,5899995.38,-5.37,-0.77,326.73,439997.75,5899992.59,326.73,49376.62,1382.00,32.15,2.77,186.84,190.14

This file contains the following information:

Storage Db; 0268_131 - 0001.db – name of the QINSy DB file to which the magnetometer 1Mag-log file corresponds;

Date; 01/02/2099 – date of the data reception from the magnetometer to QINSy;

Time; 20:06:29.285 – time of the data reception from the magnetometer to QINSy;

[Mainline] Name; L131 – name of the survey line (the same for the entire survey line);

CoG Easting; 439994.01 – Easting coordinate for the vessel CoG, meters;

CoG Northing; 5899995.38 – Northing coordinate for the vessel CoG, meters;

3049_SB scnafish MRU Pitch; -5.37 – Pitch value for ScanFish MRU, degrees;

3049_SB scnafish MRU Roll; 0.77 – Roll value for MRU ScanFish, degrees;

3040_SB HDT Fram Value; 326.73 – Heading value for ScanFish MRU, degrees;

SB MAG_01_PL Easting; 439997.75 – Easting coordinate for magnetic sensor, meters;

SB MAG_01_PL Northing; 5899992.59 – Northing coordinate for the magnetic sensor, meters;

3040_SB HDT Fram Value; 326.73 – Heading value for ScanFish MRU, degrees (repeated);

Magnetic Field 3041_MAG01-SB [nT] Value; 49376.62 – the value of the total magnetic induction vector modulus measured by the magnetic sensor, nanotesla;

Magnetic Field 3041_MAG01-SB [nT] Quality; 1382.00 – Signal Strength value for magnetic sensor measurement, conventional units;

Mag Depth Value; 32.15 – depth sensor reading, meters;

Mag Altitude Value; 2.77 – altimeter reading, meters;

SB Scanfish CoG Range to Node; 186.84 – distance from vessel CoG to Scanfish CoG in horizontal plane, meters;

SB Scanfish CoG Range to Node; 190.14 – distance from vessel CoG to CoG Scanfish in 3D, meters.

Format of a single log-file for all magnetometers, saved by Save01 command

The following example shows the header and two lines of the file with a record of ten magnetometers:

Db,Date,Time,LineName,CogE,CogN,MruPitch,MruRoll,MruHdt,E1,N1,Hdt1,T1,Sig1,Dpt1,Alt1,R2D1,R3D1,E2,N2,Hdt2,T2,Sig2,Dpt2,Alt2,R2D2,R3D2,E3,N3,Hdt3,T3,Sig3,Dpt3,Alt3,R2D3,R3D3,E4,N4,Hdt4,T4,Sig4,Dpt4,Alt4,R2D4,R3D4,E5,N5,Hdt5,T5,Sig5,Dpt5,Alt5,R2D5,R3D5,E6,N6,Hdt6,T6,Sig6,Dpt6,Alt6,R2D6,R3D6,E7,N7,Hdt7,T7,Sig7,Dpt7,Alt7,R2D7,R3D7,E8,N8,Hdt8,T8,Sig8,Dpt8,Alt8,R2D8,R3D8,E9,N9,Hdt9,T9,Sig9,Dpt9,Alt9,R2D9,R3D9,E10,N10,Hdt10,T10,Sig10,Dpt10,Alt10,R2D10,R3D10

0432_s027 - 0001.db,17/02/2099,08:25:16.100,Ls027,429991.88,5899994.18,0.28,3.49,104.57,429996.96,5899991.88,104.57,49389.17,1382.00,34.23,4.60,212.21,215.45,429996.94,5899991.92,104.57,49385.45,1426.00,32.50,5.10,212.21,215.45,429996.62,5899990.68,104.57,49389.61,1328.00,35.36,4.54,212.21,215.45,429996.61,5899990.73,104.57,49386.31,1631.00,16.38,5.04,212.21,215.45,429996.28,5899999.48,104.57,49386.35,1362.00,34.30,4.60,212.21,215.45,429996.27,5899999.53,104.57,49387.58,1675.00,15.86,5.10,212.21,215.45,429995.94,5899998.28,104.57,49391.11,1331.00,33.63,4.78,212.21,215.45,429995.93,5899998.33,104.57,49389.39,1548.00,32.00,5.28,212.21,215.45,429995.60,5899997.09,104.57,49392.41,1294.00,33.44,4.83,212.21,215.45,429995.59,5899997.12,104.57,49391.47,1216.00,-2.27,5.33,212.21,215.45

0432_s027 - 0001.db,17/02/2099,08:25:16.200,Ls027,429992.11,5899994.16,0.28,3.49,104.57,429997.16,5899991.83,104.57,49389.12,1384.00,34.23,4.58,212.21,215.45,429997.15,5899991.87,104.57,49385.47,1426.00,32.50,5.08,212.21,215.45,429996.82,5899990.62,104.57,49389.68,1328.00,35.36,4.54,212.21,215.45,429996.81,5899990.67,104.57,49386.37,1638.00,16.43,5.04,212.21,215.45,429996.48,5899999.42,104.57,49386.46,1360.00,34.30,4.50,212.21,215.45,429996.47,5899999.47,104.57,49387.72,1670.00,15.86,5.00,212.21,215.45,429996.14,5899998.23,104.57,49391.12,1326.00,33.63,4.69,212.21,215.45,429996.13,5899998.27,104.57,49389.45,1548.00,31.93,5.28,212.21,215.45,429995.81,5899997.03,104.57,49392.52,1296.00,33.44,4.79,212.21,215.45,429995.80,5899997.07,104.57,49391.50,1221.00,-2.27,5.33,212.21,215.45

The file contains the following data:

Db; 0432_s027 - 0001.db – the name of the QINSy DB-file corresponding to the survey line (the DB-file name is taken from the 1Mag-log file for the first magnetometer, recorded for the first measurement);

Date; 17/02/2099 – the date of the data reception from the magnetometer to QINSy, recalculated to the common measurement time series;

Time; 08:25:16.100 – the time of the data reception from the magnetometer to QINSy, recalculated to the common measurement time series;

LineName; Ls027 – the name of the survey line (the same for the entire survey line);

CogE; 429991.88 – Easting coordinate for the vessel CoG, meters;

CogN; 5899994.18 – Northing coordinate for the vessel CoG, meters;

MruPitch; -0.28 – Pitch value for ScanFish MRU, degrees;

MruRoll; 3.49 – Roll value for ScanFish MRU, degrees;

MruHdt; 104.57 – Heading value for ScanFish MRU, degrees;

E1; 429996.96 – Easting coordinate for the first magnetic sensor, meters;

N1; 5899991.88 – Northing coordinate for the first magnetic sensor, meters;

Hdt1; 104.57 – Heading value for ScanFish MRU, degrees (repeated);

T1; 49389.17 – the value of the total magnetic induction vector modulus measured by the first magnetic sensor, nanotesla;

Sig1; 1382.00 – the Signal Strength value for the first magnetic sensor measurement, conventional units;

Dpt1; 34.23 – depth sensor reading for the first magnetometer, meters;

Alt1; 4.60 – altimeter reading for the first magnetometer, meters;

R2D1; 212.21 – distance from the vessel CoG to the Scanfish CoG in the horizontal plane, meters;

R3D1; 215.45 – distance from vessel CoG to the Scanfish CoG in 3D, meters;

……..

E10; 429995.59 – Easting coordinate for the magnetic sensor N10, meters;

N10; 5899997.12 – Northing coordinate for the magnetic sensor N10, meters;

Hdt10; 104.57 – Heading value measured by the ScanFish MRU, degrees (repeated);

T10; 49391.47 – the value of the total magnetic induction vector modulus measured by the magnetic sensor N10, nanotesla;

Sig10; 1216.00 – the Signal Strength value measured by the magnetic sensor N10, conventional units;

Dpt10; -2.27 – depth sensor value measured by the magnetic sensor N10, meters;

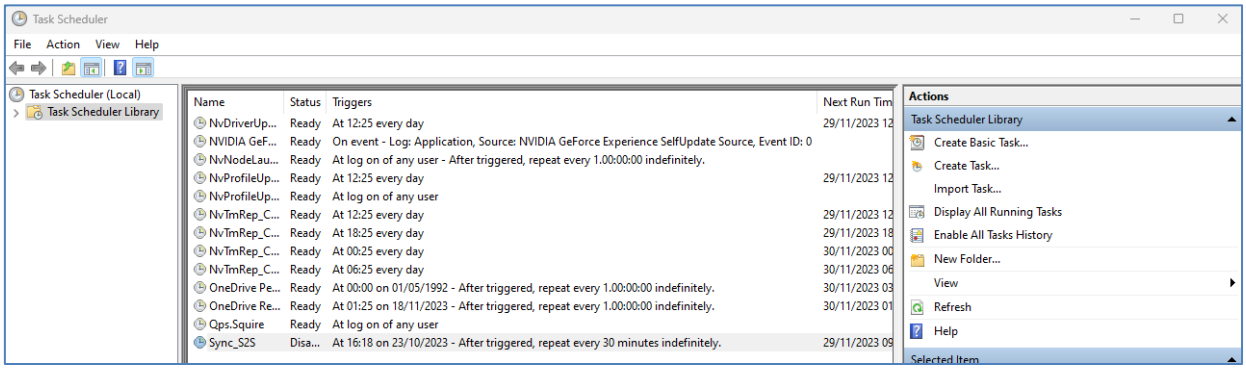
Alt10; 5.33 – altimeter value measured by the magnetic sensor N10, meters;

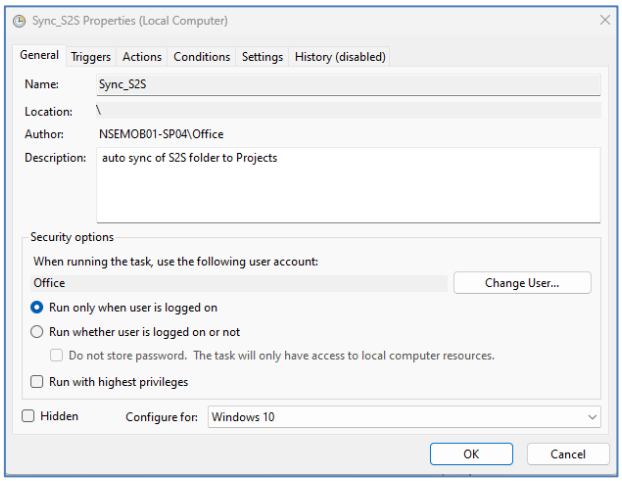
R2D10; 212.21 – distance from the vessel CoG to the Scanfish CoG in the horizontal plane, meters;

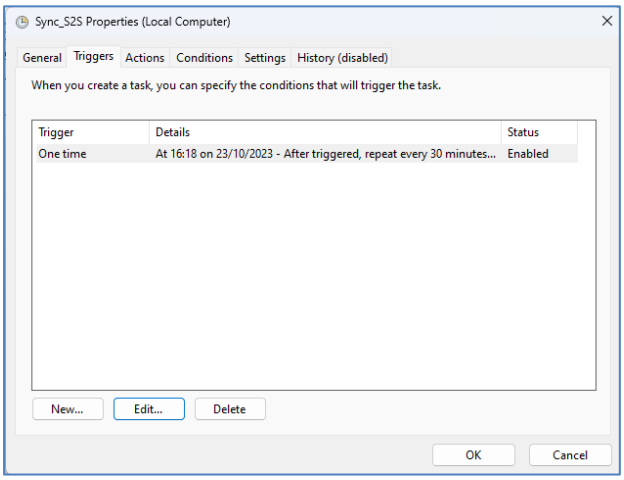
R3D10; 215.45 – distance from the vessel CoG to the Scanfish CoG in 3D, meters.

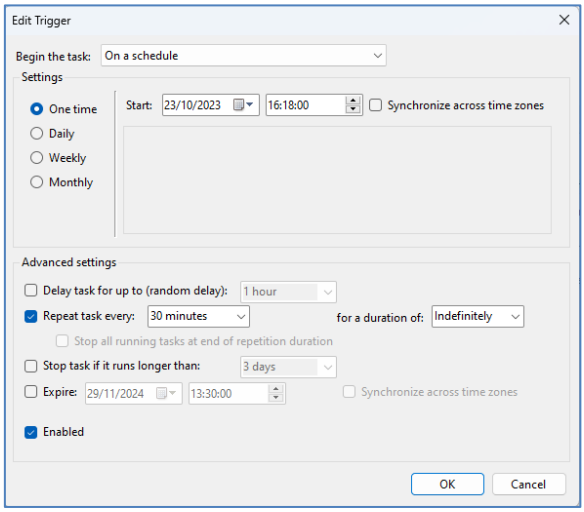
Appendix 4: Customizing Task Scheduler for Windows

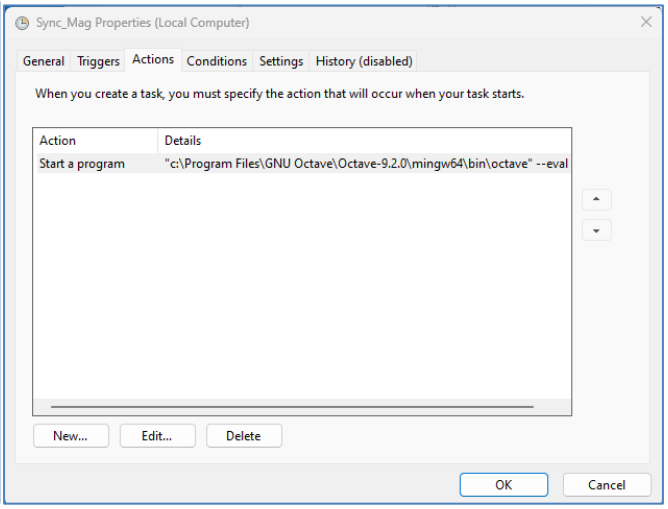
Task Scheduler for Windows can be used to prepare files automatically, with entering the corrections “for time delays in the data link”. The settings are shown below. In the “Program\script” settings you should enter the path to the executable file for MatLab or Octave. In the “Add arguments” setting – additional keys to run the gMagT13 script.

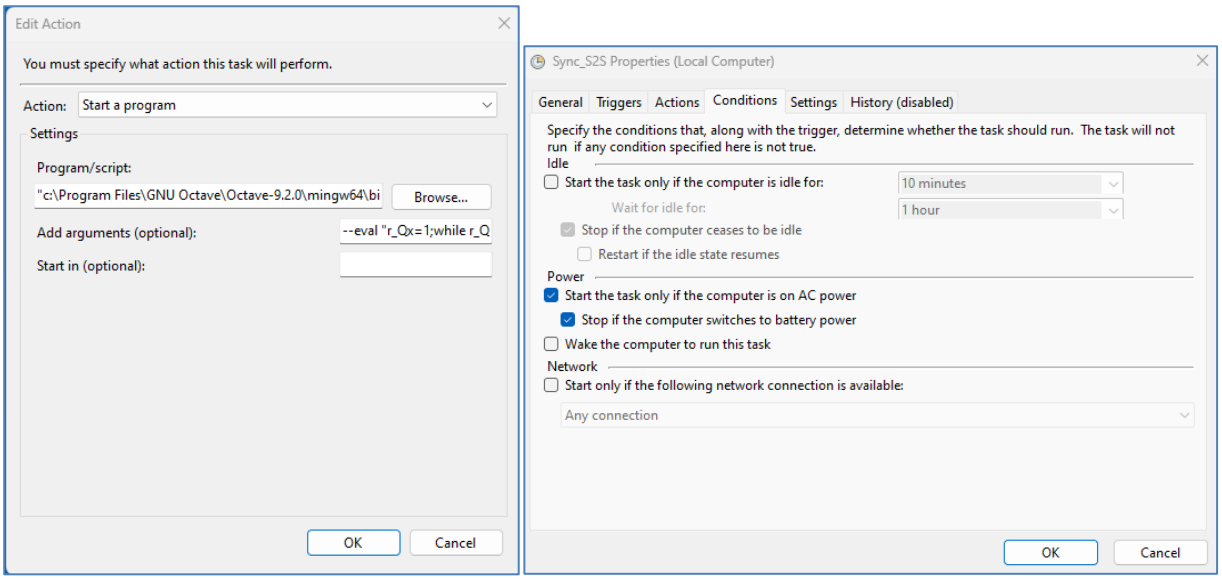












Settings for Task Scheduler to run the processing in MatLab:

-- program/script: "matlab"

-- add arguments: -r "r_Qx=1;while r_Qx,r_Qx=0;try {'CL'};MagT13;catch,r_Qx=1;end;end;quit;"

Settings for Task Scheduler to run the processing in GNU Octave (the correct path to the octave.exe file must be set):

-- program/script: "c:\Program Files\GNU Octave\Octave-9.2.0\mingw64\bin\octave"

-- add arguments: --eval "r_Qx=1;while r_Qx,r_Qx=0;try {'CL'};MagT13;catch,r_Qx=1;end;end;quit;"