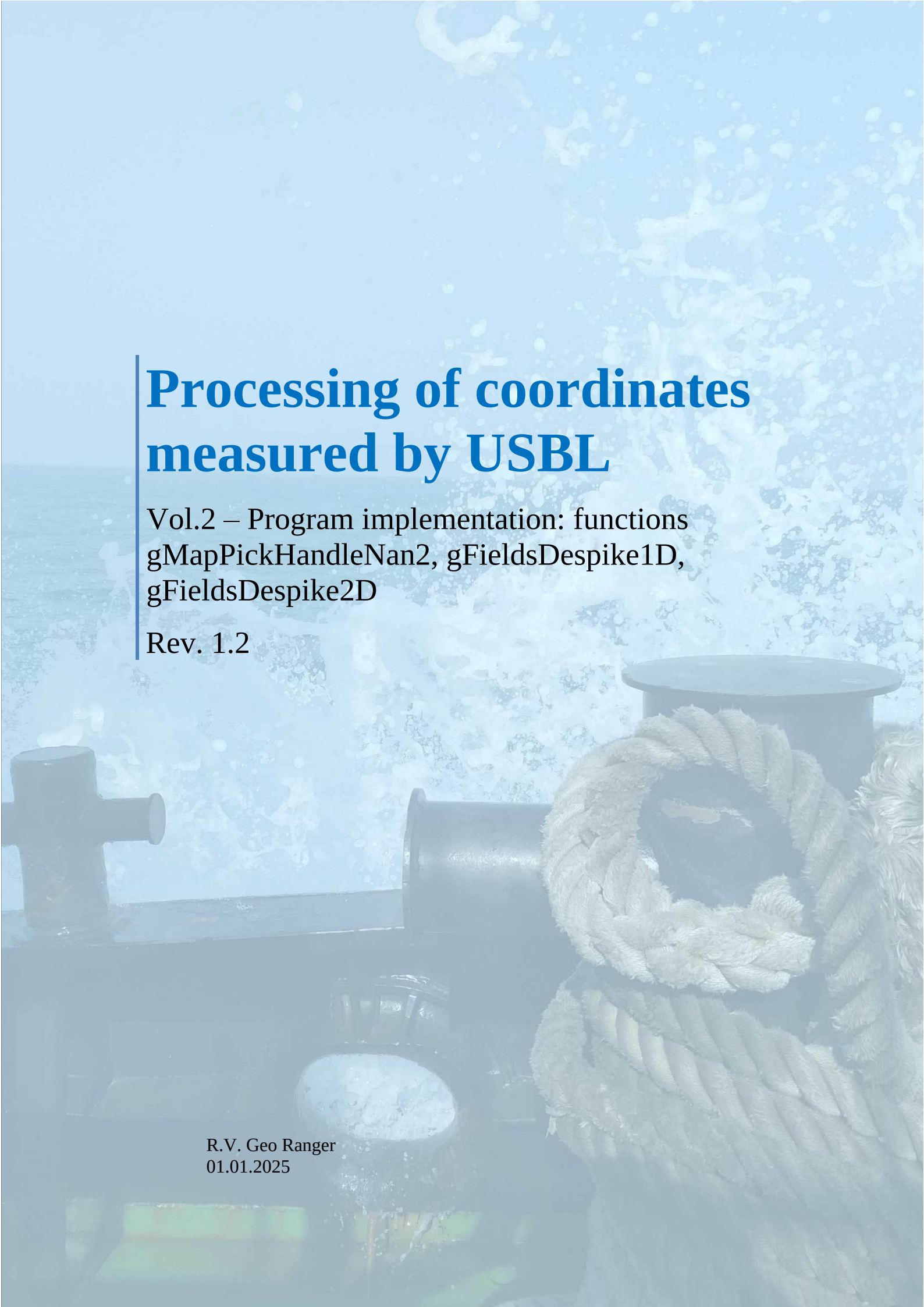




# Processing of coordinates measured by USBL

Vol.2 – Program implementation: functions  
gMapPickHandleNan2, gFieldsDespike1D,  
gFieldsDespike2D

Rev. 1.2

R.V. Geo Ranger  
01.01.2025

## Contents

|                                                                                                                    |           |
|--------------------------------------------------------------------------------------------------------------------|-----------|
| <b>Quick start: processing the coordinates of the SSS positioned by USBL (working with SssCoord01 script).....</b> | <b>4</b>  |
| <b>Introduction .....</b>                                                                                          | <b>7</b>  |
| <b>1 Graph for preliminary measurement processing .....</b>                                                        | <b>8</b>  |
| 1.1 Verifying the time stamps' representativeness.....                                                             | 8         |
| 1.2 Spikes deleting .....                                                                                          | 9         |
| 1.3 Analysis of deviations from the theoretical model.....                                                         | 12        |
| 1.4 Averaging/smoothing of measurements .....                                                                      | 12        |
| 1.5 Elimination of systematic errors.....                                                                          | 13        |
| 1.6 Summarized graph of measurements preprocessing .....                                                           | 15        |
| <b>2 Practical realization of the preprocessing graph.....</b>                                                     | <b>16</b> |
| 2.1 Graphical interface for spikes removing (gMapPickHandleNan2) .....                                             | 16        |
| 2.2 Preprocessing of scalar measurements (gFieldsDespike1D).....                                                   | 18        |
| 2.3 Preprocessing of 2D coordinates (gFieldsDespike2D).....                                                        | 19        |
| <b>3 Script example for SSS coordinates and heading processing .....</b>                                           | <b>24</b> |
| 3.1 Reading navigation files .....                                                                                 | 24        |
| 3.2 Processing .....                                                                                               | 24        |
| 3.3 Saving the processing results .....                                                                            | 29        |
| <b>Conclusions .....</b>                                                                                           | <b>30</b> |
| <b>Appendix 1. Script SssCoord01 (MatLab) .....</b>                                                                | <b>31</b> |

## History of changes

| Version | Date       | Description of changes           |
|---------|------------|----------------------------------|
| 1.0     | 2023-03-31 | Document created // Ivan D.      |
| 1.1     | 2024-04-22 | The text edited // Ivan D.       |
| 1.2     | 2025-01-01 | The grammar corrected // Ivan D. |

## List of figures

|                                                                                                                                                                                                                                                  |    |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Figure 1.1 The appearance of stretch zones on the sonar mosaic due to the lack of decimal places for the timestamp in navigation files containing SSS coordinates .....                                                                          | 8  |
| Figure 1.2 Example of SSS coordinates processing; (a) SSS track in EN coordinates; (b) a cross-profile deviation component vs a linear trend; (c) smoothed series without deleting low-spikes; (d) smoothed series with low-spikes deleted ..... | 10 |
| Figure 1.3 "Leaking" of the vessel's roll into the towed fish coordinates positioned by USBL .....                                                                                                                                               | 14 |
| Figure 1.4 Measurement summarized preprocessing graph .....                                                                                                                                                                                      | 15 |
| Figure 2.1 Tools for deleting measurements from a series (gMapPickHandleNan2 function) .....                                                                                                                                                     | 16 |
| Figure 2.2 Setting a marker with a label (gMapPickHandleNan2 function) .....                                                                                                                                                                     | 17 |
| Figure 2.3 Timestamp control window (gFieldsDespike1D function) .....                                                                                                                                                                            | 18 |
| Figure 2.4 Window with SSS heading processing results (gFieldsDespike1D function) .....                                                                                                                                                          | 19 |
| Figure 2.5 Window with SSS track in EN coordinates (gMapPickHandleNan2 function) .....                                                                                                                                                           | 20 |
| Figure 2.6 Window with SSS track in EN-coordinates (a), deviation along the linear trend TrDL (b) and deviation across the linear trend TrDC (c) .....                                                                                           | 22 |
| Figure 2.7 Deviation along the linear trend TrDL and deviation across the linear trend TrDC after deletion of spikes and smoothing .....                                                                                                         | 23 |
| Figure 2.8 Window with the results of the SSS track processing (the figure on the left is scaled) .....                                                                                                                                          | 23 |
| Figure 3.1 Timestamp control window (gFieldsDespike1D function) .....                                                                                                                                                                            | 25 |
| Figure 3.2 2D-Coordinate processing windows for SSS (gFieldsDespike2D function) .....                                                                                                                                                            | 26 |
| Figure 3.3 SSS heading processing windows (gFieldsDespike1D function) .....                                                                                                                                                                      | 27 |
| Figure 3.4 Windows with the results of the SSS coordinates and heading processing for the current sequence of profiles .....                                                                                                                     | 28 |

## List of abbreviations and acronyms

|             |                                                           |
|-------------|-----------------------------------------------------------|
| SSS         | side-scan sonar                                           |
| SBP         | sub-bottom profiler                                       |
| BTTP        | bearing to tow-point                                      |
| MatLab      | programming environment                                   |
| MBES        | multibeam echosounder                                     |
| NaN         | Not-a-Number                                              |
| NavInjector | SonarWiz utility for injecting coordinates into SSS files |
| QINSy       | software package for navigation                           |
| SonarWiz    | SSS data processing software                              |
| USBL        | ultra-short baseline hydro-acoustic positioning system    |

## Quick start: processing the coordinates of the SSS positioned by USBL (working with SssCoord01 script)

1) Script command: reading the navigation files with SSS coordinates and the heading from a specified folder.

```
{'NavQinsyLog_ReadTxt','e:\tmp\'};SssCoord01;
```

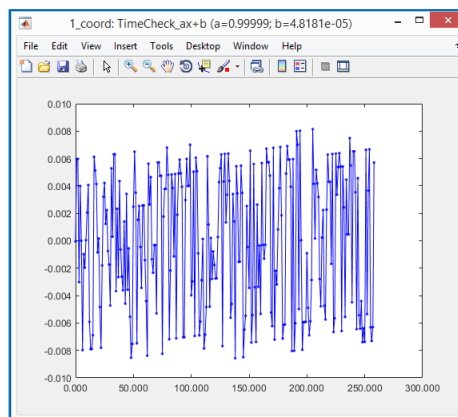
2) Script command: to process the heading and coordinates.

```
{'tXYAng_Edit',[],0,[40 40]};SssCoord01;
```

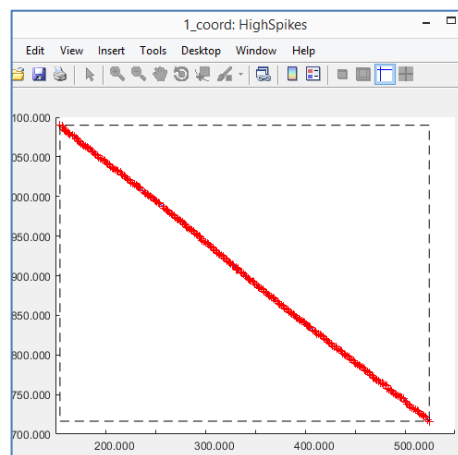
Smoothing window width for coordinates – 40, for a SSS heading – 40.

The command generates interactive and visual control windows 2.1-2.9.

2.1) Visual control window of timestamps with linear trend deleted (it is sufficient to close any "visual control window" by pressing any key).



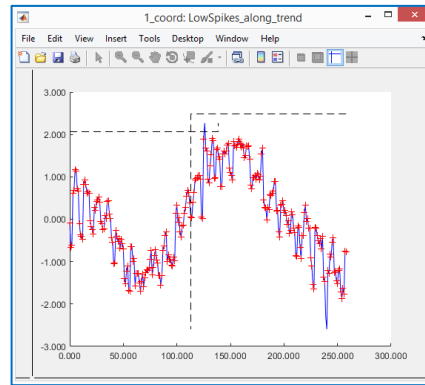
2.2) Interactive window for deleting **high spikes** coordinates.



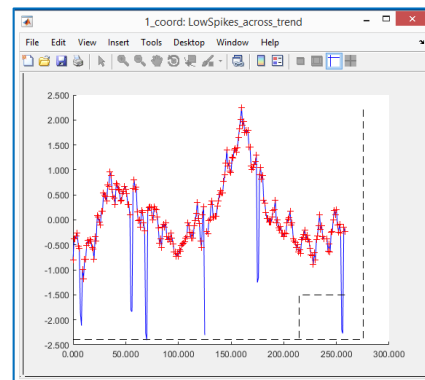
Basic keys for editing all interactive windows:

- the left mouse button – set limiter 1 to select points of the measurement series;
- the right mouse button – set limiter 2 to select points of the measurement series;
- the mouse scroll wheel – "delete" the points of the measurement series between the limiters;
- key "z" – Zoom mode (zoom in or out), exit the mode by pressing any key;
- key "x" – Pan mode (movement), exit the mode by pressing any key;
- key "q" – exit from the measurement deletion mode and close the window.

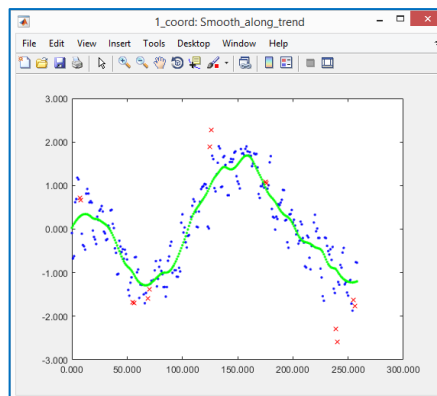
2.3) Interactive window to delete **low spikes** coordinates (plot of **deviations along the profile**).



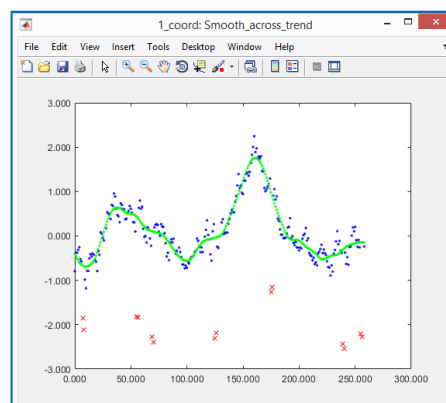
2.4) Interactive window to delete low spikes coordinates (plot of **deviations across the profile**).



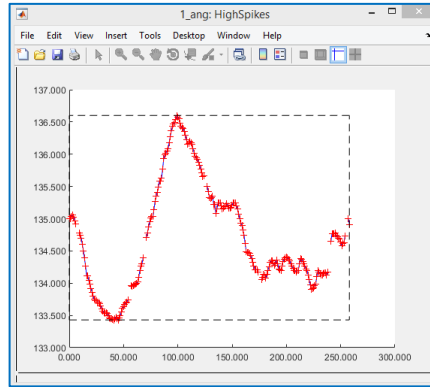
2.5) Visual control window for coordinates smoothing (deviations along the profile).



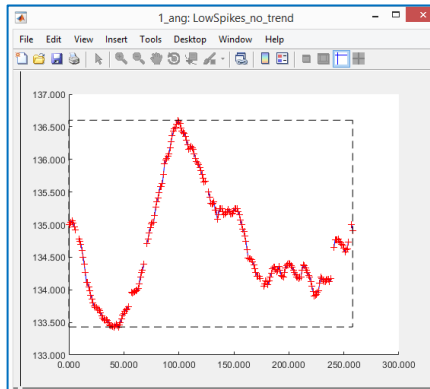
2.6) Visual control window for coordinates smoothing (deviations across the profile).



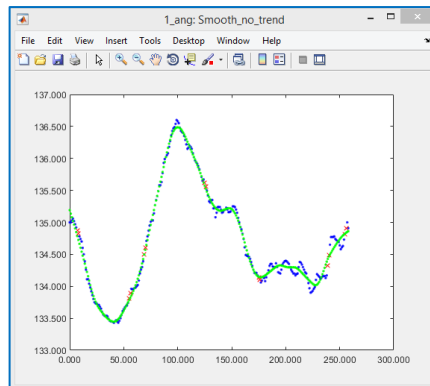
2.7) Interactive window to delete high spikes for the SSS heading.



2.8) Interactive window to delete low spikes for the SSS heading.



2.9) Visual control window for SSS heading smoothing.



3) Script command: to save navigation files with SSS coordinates and heading after processing.

`{'NavCODA_Write', 'e:\tmp\out\',[]};SssCoord01;`

## Introduction

The document describes a generalized pre-processing graph of measurements received from a time-referenced sensor. Such a device can be a depth sensor, an altimeter, a magnetic field sensor, i.e., a one-dimensional scalar quantity measured along the time axis. However, the main focus is made on coordinate processing, which is a two-dimensional scalar quantity defined along the time axis.

The description of the processing graph is done on the example of processing the coordinates of a towed SSS, whose spatial reference is obtained using the underwater positioning system (USBL). The time reference allows to move from observing the coordinates to considering the towed device velocity, and also assume that "the towed fish moves translationally along a smooth trajectory with no abrupt changes in its position". This is the assumption that was used to create the basis of the processing graph. It was then generalized to include the case of "random time-referenced measurements" (i.e., measurements that include both coordinates, and any other physical quantity).

Chapter 1 examines the problems of measurement preliminary processing (removing spikes, filtering, etc.) and methods to solve them. A generalized pre-processing graph has been presented.

Chapter 2 discusses the measurement preprocessing graph implementation. There is a description of the graph (`gFieldsDespike1D` function) intended for processing the scalar measurement series with time reference; for example, temperature sensor measurements, pressure/depth sensor measurements, towed device altimeter readings. There is also a description of the graph (`gFieldsDespike2D` function) intended to process the 2D-coordinates' measurement series with time reference; for example, the coordinates of a towed device positioned with USBL.

Chapter 3 describes a script intended for processing the SSS coordinates and heading (data import, processing SSS coordinates and heading, data export).

## 1 Graph for preliminary measurement processing

The pre-processing of measurement series includes the following tasks:

- Checking the representativeness of timestamps;
- Eliminating spikes (gross deviations);
- Averaging/smoothing the measurements;
- Removing systematic errors.

Let's discuss each task in more detail and then present the structure of the summarized pre-processing graph. In the context of spike deleting, we will address the issues of analyzing the measurement deviation from the theoretical model.

### 1.1 Verifying the time stamps' representativeness

Most of the measurements in marine geophysical surveys are time-referenced with a constant (magnetometry, SSS, SBP) or near-constant (USBL) increments. Before processing, we need to ensure that there are no artifacts (deviations from the general pattern) in the timestamps. The occurrence of such artifacts is almost always associated with the hardware or format configuration errors in the data transfer flow. Usually, the artifacts' effects can lead to both distortions in the data processed and misinterpretations regarding the nature of those distortions. An example of a timing error and its consequences are shown in Figure 1.1.

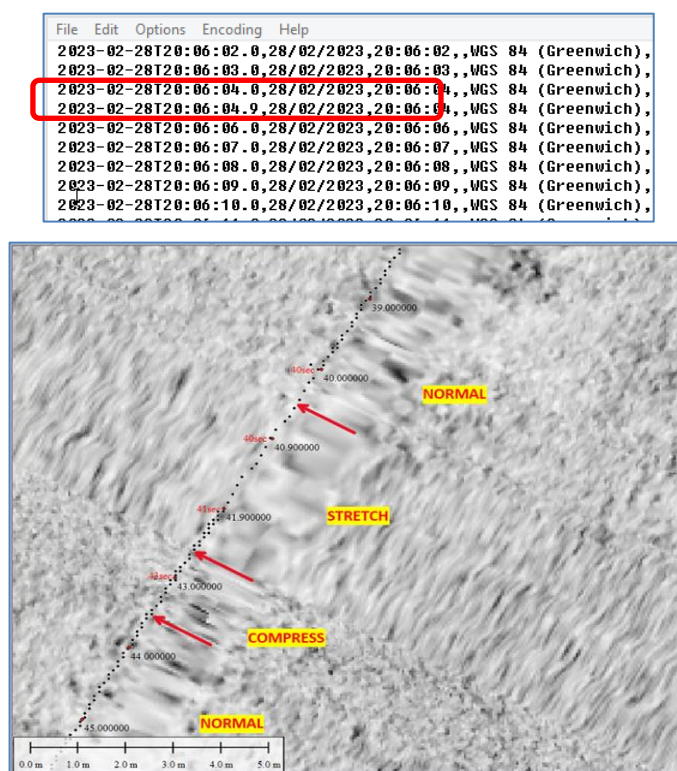


Figure 1.1 The appearance of stretch zones on the sonar mosaic due to the lack of decimal places for the timestamp in navigation files containing SSS coordinates

The upper part of Figure 1.1 shows a fragment of the navigation file with SSS coordinates injected into the SSS data files. Due to an operator mistake when setting up QINSy, the timestamp is displayed without a decimal point. This leads to periodic artifacts in the measurement timestamps, namely, the repetition of the time value (20:06:04, 20:06:04) followed by the second count (20:06:06) skipping. Due to the artifacts described above, "stretch" zones appear on the sonar mosaic (a small number of SSS pings between coordinate points with a second repetition), alternate with subsequent "compression" zones. Such zones are shown in the lower part of Figure 1.1.

It is possible to process timestamps that contain artifacts: in the example given, we could remove the duplicate seconds and calculate the values of the removed timestamps using linear interpolation. However, the best way is to find and eliminate the causes of the artifacts.

## 1.2 Spikes deleting

**Spikes** are caused by external factors unrelated to the value measured. Such spikes can be visually recognized by a sharp and significant deviation from the general pattern of the measured values (for example, a single 50-meter spike in the coordinates of the towed fish; high-amplitude noise in the MBES data when the vessel's thrusters are turned on). It is impossible to single out a "useful measurement component" against the background of an additive one with a spike; the most acceptable decision is to remove such a measurement from the series, thus eliminating its impact on subsequent processing.

In a first approximation, spikes can be viewed as a short-period component of the signal. Therefore, to detect and delete them, it seems reasonable to first remove the **long-period trend**, which can mask relatively low-amplitude spikes. If the values of the processed measurement series are used to calculate the long-period trend, then the algorithm with its multi-step removal of spikes works well:

- 1) **The highest** amplitude **spikes**, which are clearly seen in visualizing the processed signal, are deleted.

Such spikes can affect the calculated trend and greatly distort its shape;

- 2) A long-period trend is calculated for a series of measurements using the deleted high-amplitude spikes, and then subtracted from the series of measurements. To better visualize the spikes, transformations can be calculated before the trend and the original data. An example of transformations named "**deviations from the trend along the movement**" and "**deviations from the trend across the movement**" is given in the text below (Chapter 2.3);
- 3) For a series of measurements with a removed long-period trend, the spikes are reanalyzed and re-deleted, which makes it possible to delete **low-spikes** with a relatively small amplitude (previously they were masked by a long-period trend).

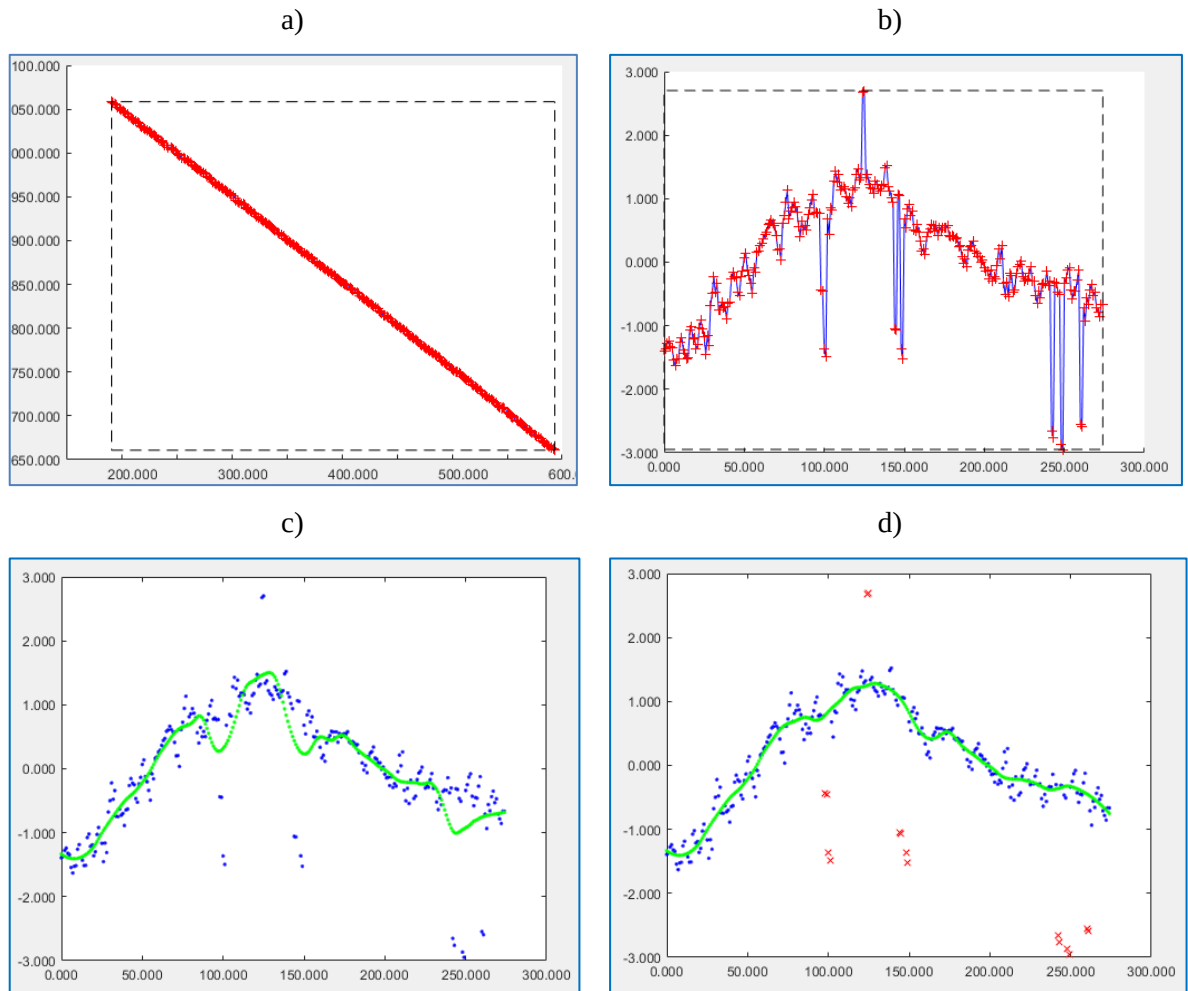


Figure 1.2 Example of SSS coordinates processing; (a) SSS track in EN coordinates; (b) a cross-profile deviation component vs a linear trend; (c) smoothed series without deleting low-spikes; (d) smoothed series with low-spikes deleted

Figure 1.2 shows an example of the real SSS data processing. Figure 1.2, a illustrates the SSS track in rectangular coordinates. For Figure 1.2, b, the coordinate system is rotated and the linear trend is removed to best visualize the low-spikes that are almost indistinguishable in the EN-coordinates. Figure 1.2, c-d shows a comparison of data smoothed in the presence of low-spikes and with deleted low-spikes. As we can see from the figures, low-spikes cause smooth shifts in coordinates with an amplitude of up to 1 meter. On sonograms, this will show up as a visually indistinguishable smooth shift for sonar pings; objects in the low-spike impact zone will be distorted and shifted by up to 1 meter.

Speaking of "removing spikes", it is important to specify the action of "**deleting a measurement** from a series". This could be done in the following way:

- Delete the entire measurement with its timestamp, and continue processing without the one deleted timestamp. The problem may arise when measurements of multiple sensors are referenced to the same timestamp. For example, for magnetic surveying the measurements of a magnetic sensor, a depth sensor and an altimeter are referenced to the same timestamp. Then, to delete the timestamp due to "bad" (substandard) altimeter measurements may result in deleting "good" (standard) magnetic field measurements. On top of that, the timestamp proper contains useful information about the

measurements period. For the reasons given above, it was decided not to resort to the step "delete with a timestamp";

- We can delete the values of a measurement by replacing it with NaN (Not-a-Number, the special state for floating point numbers). However, additional nuances may occur in mathematical processing; for example, when calculating transformations in a sliding window (filtering, averaging), which may include the NaN-value in the calculations, this will bring about assigning the NaN-value to the result of the calculations. That is, if we delete one point from the coordinate measurements and then average in a sliding window of 50 points, we will get a "data gap" (a sequence filled with NaN) with a size of 99 points. Therefore, it is necessary to take into account NaN values in all calculations (for example, eliminate them before calculations) and adapt all algorithms to the "variable step measurement series";
- We have chosen the deletion method when "bad" values are marked, and after marking is completed, they are "recreated" using interpolation. Probably, it would be more correct to speak about **replacing measurements** with interpolated values rather than deleting them. However, by analogy with "removing the high-frequency component" (when, actually, the measurement values are also recalculated and replaced), we will use the words "delete" or "remove". The information about the values being "bad" is kept in a special 32-bit mask. In this case, the algorithms for a constant measurement step can be used during further processing, assuming that "bad" data is a relatively rare event (indeed, with a large amount of "bad" data, the entire series of measurements becomes substandard and should not be accepted for processing), while interpolation gives a sufficiently accurate approximation. If necessary, we can replace the "bad" values with NaN and use algorithms for measurements with a variable step. It's a good idea to mark "created" (interpolated) data when visualizing the values for a processor operator.

### 1.3 Analysis of deviations from the theoretical model

Given a mathematical model that theoretically calculates the processed measurements, it is interesting to "**model the series**" instead of "calculating the trend based on the processed series". In this case, the difference between the modeled and measured series will show us the deviations of the measurements from the theoretical model. If the theoretical model describes the measured series in a sufficiently accurate way, then the spikes (for the spikes deletion procedure) will be more clearly visible. We also get a tool for estimating "**deviations from the theoretical model**", which will enable us to identify and analyze measurement deviations from the theoretical model in medium and long periods. We also get a tool to estimate the error of the theoretical model (if it is not accurate).

To pre-process towed fish coordinates using USBL data, such a model can be a calculation of towed fish coordinates based on cable counter measurements, towed fish depth sensor measurements, and information about sea current to calculate towed fish drift. Rough calculations based on survey line and vessel's diametric plane direction can be used instead of information about the current. The results of these calculations make it possible to analyze the difference between the theoretical position of the towed fish and the measured position. The analysis covers issues such as "the stability of the towed fish motion during winch operation" or "the presence of systematic USBL errors" (an example is given in the text in Figure 1.3).

### 1.4 Averaging/smoothing of measurements

Averaging/smoothing of measurements is done to reduce the noise (similar to high frequency component). When describing the reason for applying averaging without mathematical rigor, we assume the random measurement errors to be normally distributed (additive Gaussian noise with zero mathematical expectation). While the measured values within the "sliding averaging window" vary slightly (or linearly) compared to the spread of the high-frequency noise component. Therefore, the measurement error will decrease by  $\sqrt{n}$  when averaged over  $n$  measurements (three times for a window of 10 measurements). The processor defines this visually as "a good reduction in the high-frequency noise level with an acceptable distortion of the useful signal".

In practice, if the noise does not fit the additive Gaussian model, we have two effects:

- The measurement error is "smeared" (redistributed) within the averaging window, while the measurements with a relatively large error will become "more accurate" whereas the other ones become "less accurate". All measurements within the averaging window can turn into "measurements with a relatively large errors", if spikes are not removed in advance.
- A series of measurements become "smooth" with no significant deviations between the adjacent measurements. This property may be specific to the measured quantity, while smoothing may be an important condition for further processing. For example, noise with a spread of 0.5-1 meter in the towed SSS coordinates is physically unreal (the movement of an outboard towed fish follows the laws of hydrodynamics and its own inertia, which limits sharp changes in position and heading). If the noise persists, it will lead to fragmentation and distortion of small objects on sonograms.

## 1.5 Elimination of systematic errors

Systematic errors must be analyzed individually for each type of measurement. Such an analysis should have an idea about the mathematical model of measurements, calibrations and errors of measuring devices. Most often, systematic errors occur due to incorrect calibration; less often, errors are associated with unaccounted factors in the measurement model (i.e., model error).

Regarding the determination of the towed fish coordinates based on USBL, shows that such errors are almost always caused by incorrect calibrations of the systems associated with the positioning of the SSS. Let's consider a few examples:

- Constant summand to the SSS pings rotation angle, given the SSS heading is determined as BTTP (direction from USBL transponder to SSS tow point). In this case, the misalignment angle for the USBL in the horizontal plane is likely to be incorrectly determined. At any distance of the towed fish from the vessel, the corrections will be constant and equal to the misalignment angle error. A change of the towing cable rotation angle in the area of the towed fish can also cause the error due to the impact of the current. Inside the water column, the cable is not stretched in a straight line, but bent, which can be considered as a "model error". However, in this case the cable is always bent to one direction in opposite courses.
- Constant summand to the SSS ping rotation angle, if the SSS heading is determined by the internal SSS compass. In this case, the compass may not be calibrated correctly. It is also possible that the magnetic declination or a value similar to the Gaussian convergence in the area of operation may be incorrectly accounted for.
- The summands for the SSS position along and across the survey line are constant for the survey profile (to compensate for them, corrections are usually applied to the sonogram position in SonarWiz). Such summands can be linked to the USBL misalignment angle miscalculation. If the angle in the horizontal plane (yaw) is incorrect and the towed fish is approximately the same distance from the vessel (typical of most surveys), the corrections will be "approximately constant" (their value can be calculated by a right triangle, where the hypotenuse is the distance of the SSS from the USBL transceiver).
- "Leaking" of the vessel motion (roll, pitch) into the towed SSS coordinates (Figure 1.3). This is caused by improper estimation of the USBL misalignment angles. In this case, a part of roll/pitch is projected onto yaw, which leads to oscillations on the SSS track synchronized with the roll/pitch. Thus, the oscillations exhibit a pronounced period (approximately 5-10 seconds) corresponding to that of the vessel's roll.

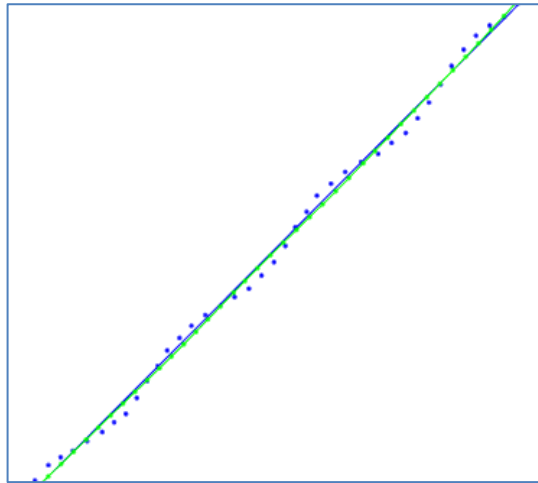


Figure 1.3 "Leaking" of the vessel's roll into the towed fish coordinates positioned by USBL

As mentioned above, there is no universal method for removing a systematic error. Anyway, it should be analyzed, its causes identified and individual methods of its removal or reduction considered. Rare exceptions are possible to reduce the effect of systematic error in "general processing". An example is the effect of the vessel motion (roll, pitch) on the SSS coordinates. In the first approximation, such an error can be considered as a "high-frequency noise" and removed by increasing the width of the averaging sliding window. However, such error removal is cosmetic and does not eliminate all the consequences of incorrect USBL calibration – it will also be necessary to enter a coordinate shift along and across the profile direction, and if the length of the towing cable changes significantly, the entered coordinate shifts will be different for different sections of the survey profile.

Methods for removing systematic errors will not be considered in this paper. The situation with the recognition and removal of errors in the misalignment angles of USBL, when assessing the position of the towed SSS, will be considered in a separate document.

## 1.6 Summarized graph of measurements preprocessing

A block diagram of the measurement preprocessing graph is shown in Figure 1.4. Each block corresponds to one of the operations described above.

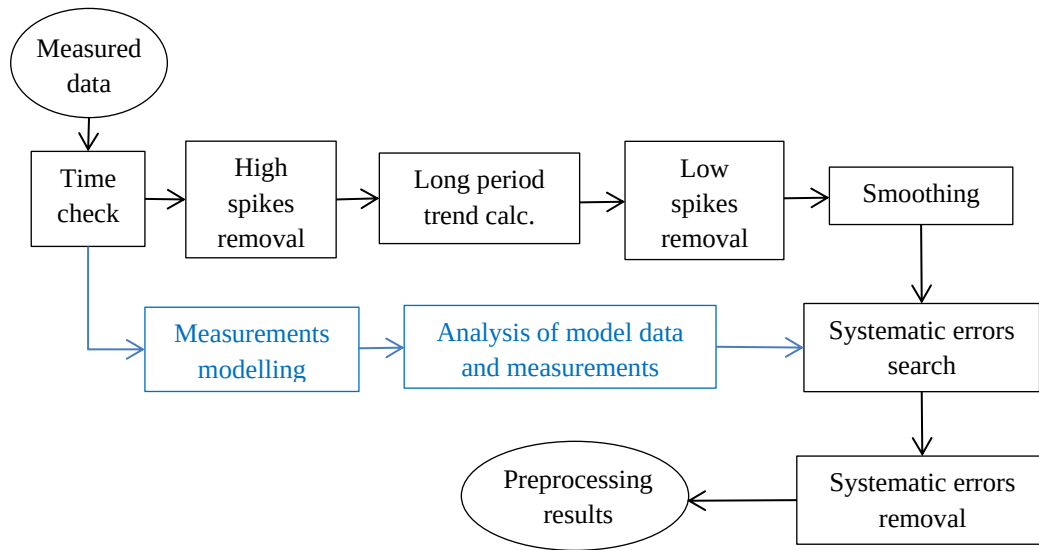


Figure 1.4 Measurement summarized preprocessing graph

Measured data – initial data; a series of time-referenced measurements for which analysis and pre-processing is performed;

Time check – checking the correctness of timestamps;

High spikes removal – deleting of high-spikes from a number of measurements;

Long period trend calc. – calculation of a long-period trend for a number of measurements after deleting high-spikes;

Low spikes removal – subtracting a long-period trend from a series of measurements, identifying and deleting low-spikes;

Smoothing – smoothing/averaging the series values (spikes were deleted, long-period trend not removed);

Measurement modeling – a preprocessing method based not on "long-period trend calculation", but on modeling the measured signal based on other independent data. The graph block corresponds to the execution of such modelling;

Analysis of model data and measurements – based on a comparison of the model and measured series, spikes are deleted, data are smoothed, and data differences are analyzed for long and medium periods;

Systematic errors search – identification of systematic errors and their nature. In this case, the results of preliminary processing are used for search of data discrepancies caused by systematic errors (a search for data discrepancies with modeling results can be performed as well);

Systematic errors removal – removal of systematic errors; removal methods are determined after systematic errors are identified. These can be "formal" methods that do not affect the source of error (e.g., sonogram shift and high-frequency component filtering shown in Figure 1.3) or

methods that remove the source of error (e.g., correct recalibration of USBL for towed fish positioning);

Preprocessing results – measurements preprocessing results after systematic errors removal.

## 2 Practical realization of the preprocessing graph

For the practical realization of the measurement preprocessing graph described in the first chapter, three functions are created in the MatLab system:

- function `gMapPickHandleNan2` provides the operator with a graphical interface for spikes deleting;
- function `gFieldsDespike1D` is designed to process some scalar values measured in time axis (temperature sensor measurements, pressure/depth sensor measurements, towed fish altimeter measurements, etc.). The processing includes the following graph steps (Figure 1.4): Time check, High spikes removal, Long period trend calc., Low spikes removal, Smoothing;
- function `gFieldsDespike2D` is designed to process coordinates on a 2D-plane (used to process towed fish coordinates obtained from USBL data). Processing includes the following graph steps (Figure 1.4): Time check, High spikes removal, Long period trend calc., Low spikes removal, Smoothing.

The text below describes the purpose of the functions, the user interface and the mathematics methods implemented inside the function. A detailed description of the functions input-output, which is necessary for programming, can be obtained from the functions help-topics.

### 2.1 Graphical interface for spikes removing (`gMapPickHandleNan2`)

The `gMapPickHandleNan2` function provides the user with an interactive graphical interface to delete spikes. After starting the function, a plot is displayed in the XY axes (a measurement series of a scalar value along the time axis or towed fish track coordinates in EN-coordinates). Two tools are available to the operator to delete selected measurements from a series, as shown in Figure 2.1.

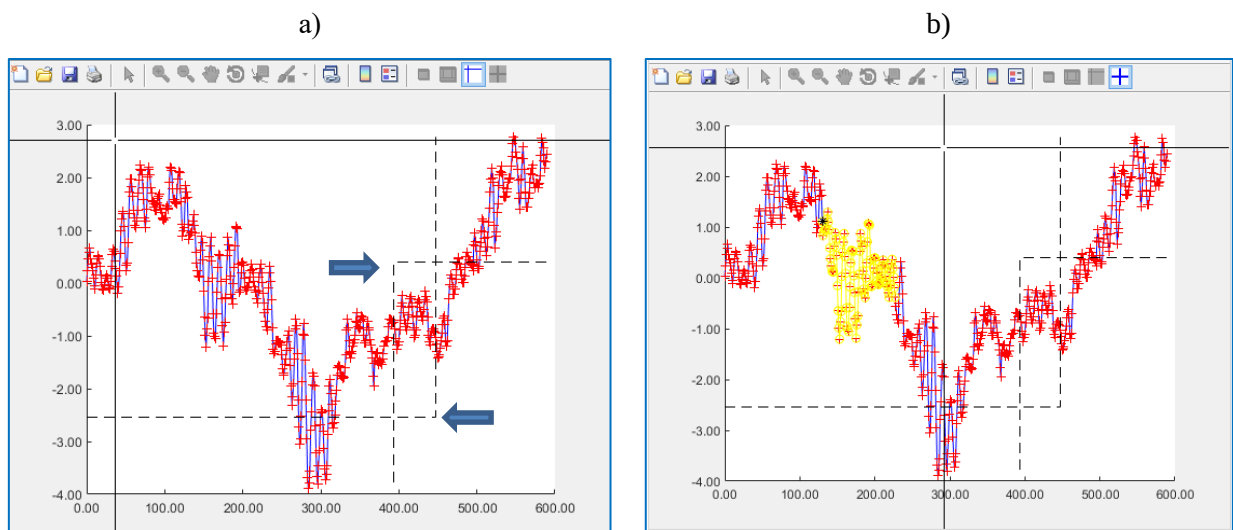


Figure 2.1 Tools for deleting measurements from a series (`gMapPickHandleNan2` function)

The first tool (Figure 2.1, a) allows to set dotted borders by clicking the left and right mouse buttons; click points are shown by arrows. Clicking the mouse wheel deletes all points within the dotted rectangle from the measurement series (marked as deleted). The second tool (Figure 2.1, b) allows us to delete a sequence of points by selecting the first and the last point of the sequence with the left and right mouse buttons (the selected sequence is highlighted in yellow). Deletion is done by clicking the mouse wheel. The second tool is also useful for selecting and deleting sections of "scattering" towed fish track coordinates in the EN axes (we can select parts of the track "tied in knots"). Switching between modes is done by keys "1" and "2".

Below is a description of all the controls:

- left mouse button – setting limiter 1 to select points of a measurement series;
- right mouse button – setting limiter 2 to select points of a measurement series;
- mouse wheel – deleting (marking) measurement points between limiters 1 and 2;
- key "1" – activating the first tool;
- key "2" – activating the second tool;
- key "a" – cancelling the last erasure of the deletion chain for the current tool;
- key "s" – applying the last erasure of the deletion chain for the current tool;
- key "z" – Zoom mode, the exit from the mode by pressing any key;
- key "x" – Pan mode (movement), the exit from the mode by pressing any key;
- key "c" – a marker setting mode, when each marker is signed with a label and a value along the XY axes and a measurement number inside the series (Figure 2.2);
- key "q" – exit from the measurement deleting mode and closing the window.

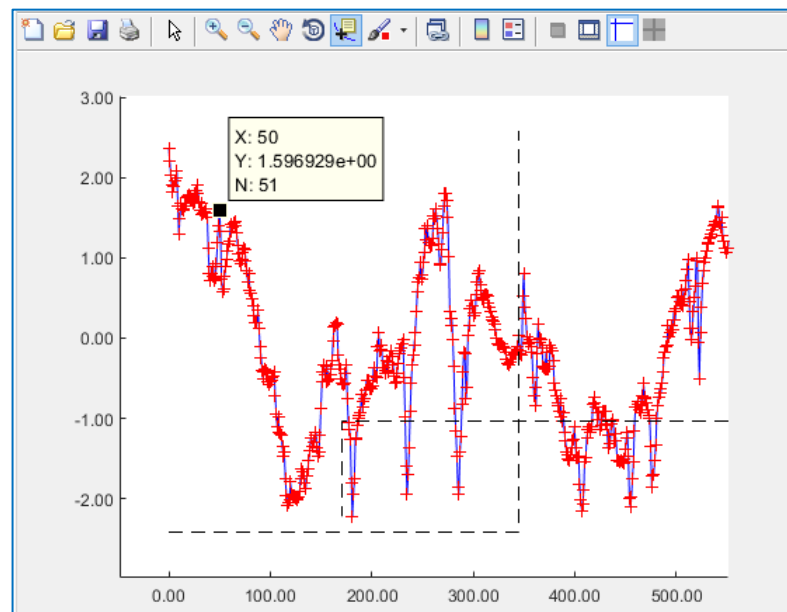


Figure 2.2 Setting a marker with a label (gMapPickHandleNan2 function)

## 2.2 Preprocessing of scalar measurements (gFieldsDespike1D)

The `gFieldsDespike1D` function is designed to process a series of scalar measurements over time, such as pressure/depth sensor measurements, towed altimeter measurements. The function header is shown below:

```
function [Us,QMask]=gFieldsDespike1D(t,U,QMask,QBit,QBitReset,Tr,IntP,SmoothP,f)
```

Processing includes the following graph steps (Figure 1.4): Time check, High spikes removal, Long period trend calc., Low spikes removal, Smoothing.

**Time check.** Timestamps `t` are numbered from 1 to `n`; then a linear trend is calculated as given by the equation  $y=ax+b$ . The calculation is done by the least squares method using the `polyfit` function (help in MatLab is available by the `help polyfit` command). The result is displayed as a graph that shows the "irregularity" of the time step `t` between measurements (Figure 2.3). The value of the coefficient "a" (Figure 2.3, signed in the window name) shows the average time-step size.

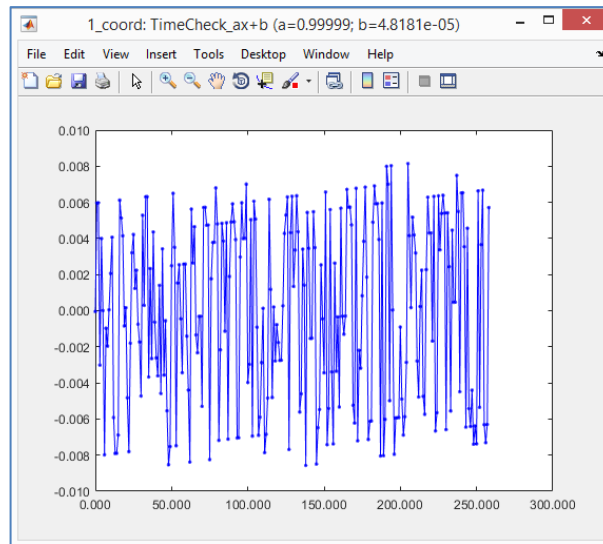


Figure 2.3 Timestamp control window (gFieldsDespike1D function)

**High spikes removal.** The value of the `U` series is plotted along the time axis `t` in the `gMapPickHandleNan2` function window (Figure 2.1) for interactive deleting of high-spikes by the operator.

**Long period trend calc.** After deleting high-spikes from a series of measurements, a long-period trend is calculated. The trend type is set in the `Tr` input parameter. The following calculations can also be done:

- const – no trend is calculated yet; all trend values are assumed to be equal to the specified value;
- poly – a polynomial trend of a given degree is calculated; for example, a linear trend given by the equation  $y=ax+b$ . The calculation is performed using the least squares method using the `polyfit` function (help in MatLab is available by the `help polyfit` command);
- smooth – a trend is calculated by averaging in a sliding window using the `smooth` function (help in MatLab is available by the `help smooth` command); moving, lowess, loess, sgolay, rlowess, rloess smoothing methods are available.

-- user-defined function – the trend is calculated using a MatLab function written by the operator. This option is provided to set the function that calculates the "modeling series" and the subsequent use of the modeling results as a trend.

**Low spikes removal.** A subtraction is performed from an observation series of the long-period trend calculated in the previous step. The resulting series is used to identify and delete low-spikes in the [gMapPickHandleNan2](#) function window.

**Smoothing.** For the initial measurement series all deleted spikes are replaced by the interpolation results. Interpolation is done by the [interp1](#) function (help in MatLab is available by the [help interp1](#) command). The interpolation parameters are set in the input parameter [IntP](#). The following interpolation methods can be used: nearest, next, previous, linear, spline, pchip, makima, cubic. Next, the series is smoothed, in which the spikes values were replaced by interpolated ones. Smoothing parameters are set in the [SmoothP](#) input parameter. Smoothing is performed using the [smooth](#) function (help in MatLab is available by the [help smooth](#) command). Such smoothing methods as moving, lowess, loess, sgolay, rlowess, rloess are also available.

As a result of the calculations, we get a smoothed measurement series [Us](#) (with previously deleted and interpolated spike values) and a [QMask](#) mask marking the deleted values (one bit of the 32-bit [QMask](#) mask specified in the [QBit](#) parameter).

The results of the function are displayed in the window with the number specified in the [f](#) parameter. Figure 2.4 shows an example of the SSS heading processing results window. Blue color – initial data series [U](#); red color – deleted measurements (marked as spikes); green color – the [Us](#) processing result given by the function.

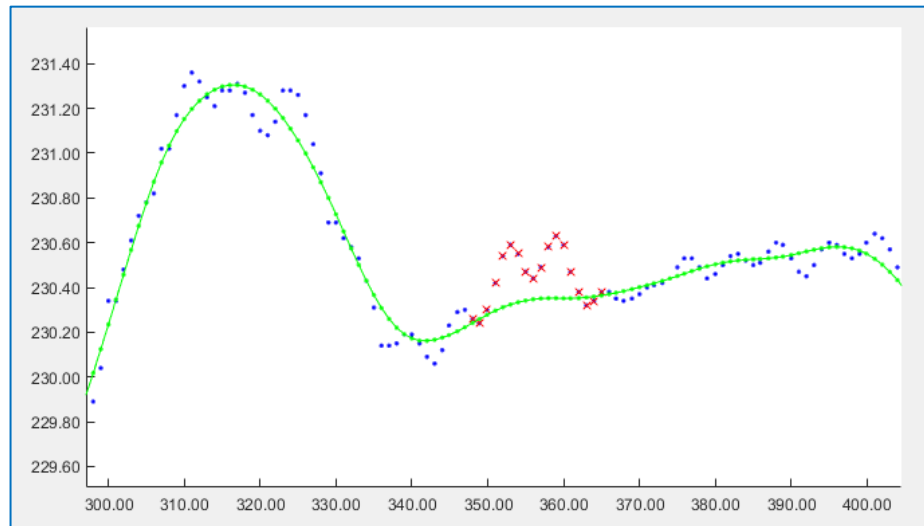


Figure 2.4 Window with SSS heading processing results ([gFieldsDespike1D](#) function)

### 2.3 Preprocessing of 2D coordinates ([gFieldsDespike2D](#))

The [gFieldsDespike2D](#) function is designed to preprocess coordinates on a 2D plane (used to process towed fish coordinates obtained from USBL data). The function header looks as follows:

```
function [Es,Ns,QMask]=gFieldsDespike2D(t,E,N,QMask,QBit,QBitReset,Tr,IntP,SmoothP,f)
```

Processing includes the following graph steps (Figure 1.4): Time check, High spikes removal, Long period trend calc., Low spikes removal, Smoothing.

**Time check.** Timestamps are numbered from 1 to  $n$ ; then a linear trend is calculated given by the equation  $y=ax+b$ . The calculation is done by the least squares method with the [polyfit](#) function (help in MatLab is available by the [help polyfit](#) command). The result is plotted as a graph that shows the "irregularity" of the time step between measurements (Figure 2.3). The value of the coefficient "a" (Figure 2.3, signed in the window name) shows the average step size.

**High spikes removal.** The  $[E,N]$  coordinate value is plotted in EN axes in the [gMapPickHandleNan2](#) function window (Figure 2.1) for interactive removal of high-spikes by an operator (Figure 2.5 does not have any high spikes to demonstrate in this example).

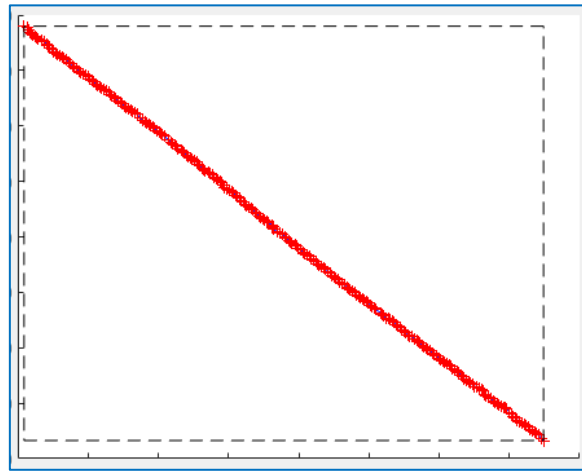


Figure 2.5 Window with SSS track in EN coordinates ([gMapPickHandleNan2](#) function)

**Long period trend calc.** After deleting high-spikes, a long-period trend is calculated separately for the E and N coordinates. In fact, two trends are calculated in the time-Easting and time-Northing axes. The trend type is set in the [Tr](#) input parameter. The following types of calculations are possible:

- const – trends are not calculated, all trend values are taken equal to the specified value;
- poly – polynomial trends of a given degree are calculated; for example, linear trends given by the equation  $y=kx+b$ . The calculation is performed by the least squares method with the [polyfit](#) function (help in MatLab is available by the [help polyfit](#) command);
- smooth – trends are calculated by averaging in a sliding window using the [smooth](#) function (help in MatLab is available by the [help smooth](#) command). Such smoothing methods as moving, lowess, loess, sgolay, rlowess, rloess are also available.
- user defined function – trends for E and N are calculated using a MatLab function written by the operator. This option is provided to specify the function that calculates the "modeling series" and the subsequent use of the simulation/modeling results as a trend. For example, the towed fish track can be calculated by the Layback model based on the following input data: tow-point coordinates, tow cable length, depth (pressure) sensor measurements, vessel heading, and its diametric plane direction.

**Low spikes removal.** The long-period trends calculated in the previous step are subtracted from the E and N coordinates.

Subsequent calculations focus on transforming the EN-coordinates to those in the coordinate system tangential directed to the long-period trend at each point. This is done to better visualize low-spikes and also introduce the concepts of "*deviations from the trend along the displacement*" and "*deviations from the trend across the displacement*". In the first approximation, the "along" deviation corresponds to a change in the distance from the towed fish to the USBL transceiver, while the "across" deviation corresponds to a change in the bearing angle from the USBL transceiver to the towed fish. To obtain the "along" and "across" coordinates, the following calculations are performed:

- The angle of the tangent to each trend point is calculated (the angle changes smoothly, since the trend is a smooth function);
- The difference values (between the initial EN series and the long-period trends) are calculated along the tangent ("deviation from the trend along the displacement" [TrDL](#)) and across the tangent ("deviation from the trend across the displacement" [TrDC](#)).

Based on the series obtained such as "along the displacement" and "across the displacement" coordinates, low-spikes are identified and deleted using the [gMapPickHandleNan2](#) function. An example of SSS track in EN-coordinates as well as deviations along and across the trend (calculated as a linear function – a first-degree polynomial) is shown in [Figure 2.6](#). The figure shows distinct spikes for the deviation "across the trend". These spikes are indistinguishable in the EN coordinate axes, even when the track is scaled in the spike area. The artifacts are most pronounced when used with the time axis and in the along and across trend/displacement coordinate axes.

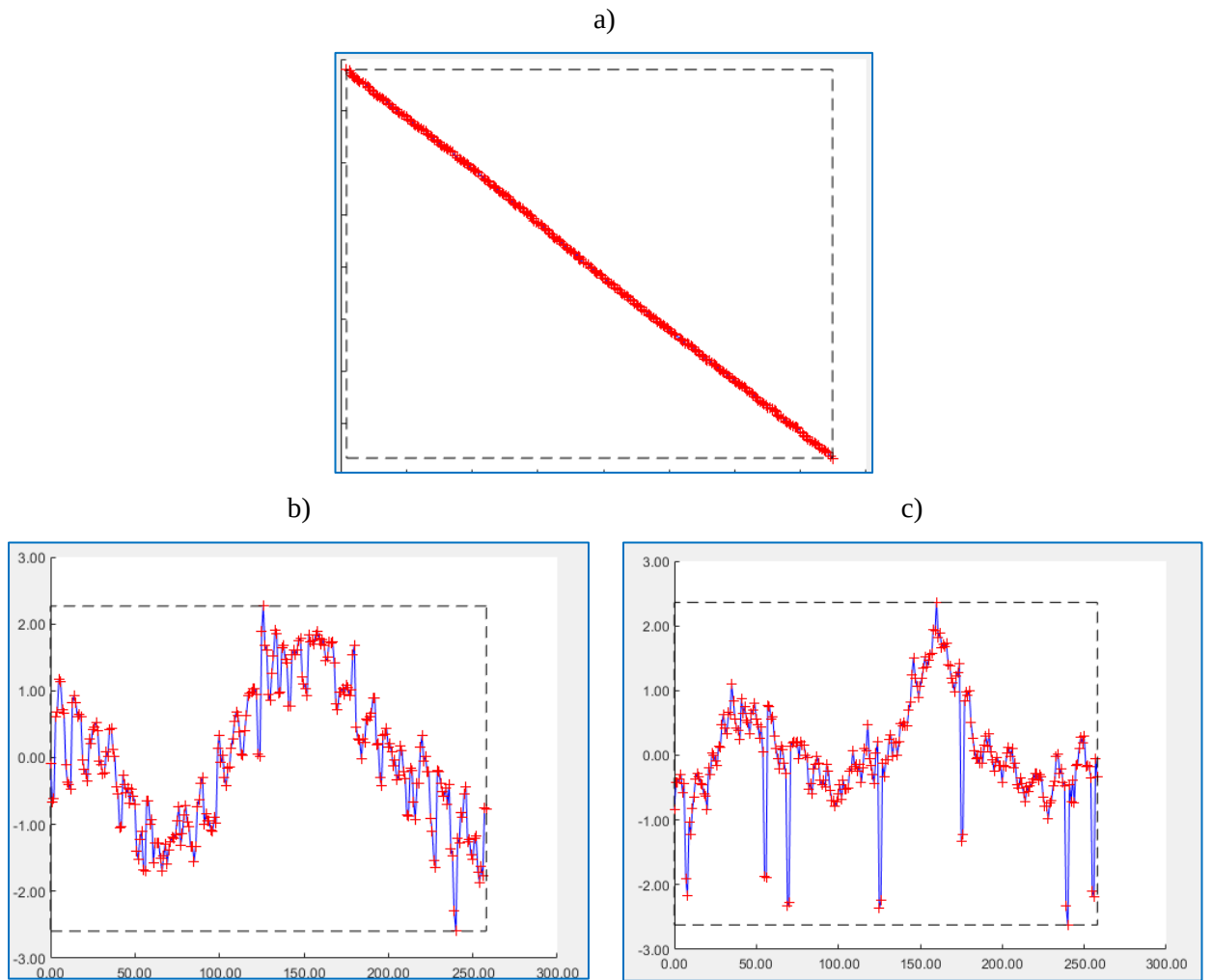


Figure 2.6 Window with SSS track in EN-coordinates (a), deviation along the linear trend TrDL (b) and deviation across the linear trend TrDC (c)

In the first approximation, the linear trend calculation for coordinates (Figure 2.6, a) can be considered as creating a model of the towed fish movement along the survey line at a constant speed (uniform step on the plane for a uniform time-step). Then the plots in Figure 2.6, b-c illustrate the measured deviation values from the "theoretical model" at a constant speed. The long-period component when "moving along" the profile (Figure 2.6, b) correlates with a change in the towed fish speed. The long-period component responsible for "movement across" the profile (Figure 2.6, c) is related to the deviation of the towed fish from the profile line.

**Smoothing.** For the initial measurement series E and N, all deleted spikes are replaced by the interpolation results. Interpolation is performed by the `interp1` function (help in MatLab is available by the `help interp1` command). The interpolation parameters are set in the input parameter `IntP`; possible methods are: nearest, next, previous, linear, spline, pchip, makima, cubic. Next, the series is smoothed, in which the spike values were replaced by the interpolated ones. Smoothing parameters are set in the `SmoothP` input parameter. Smoothing is performed using the `smooth` function (help in MatLab available by the `help smooth` command). Such smoothing methods as: moving, lowess, loess, sgolay, rlowess, rloess are also available.

The smoothed and recalculated series along and across the trend are shown in green in Figure 2.7.

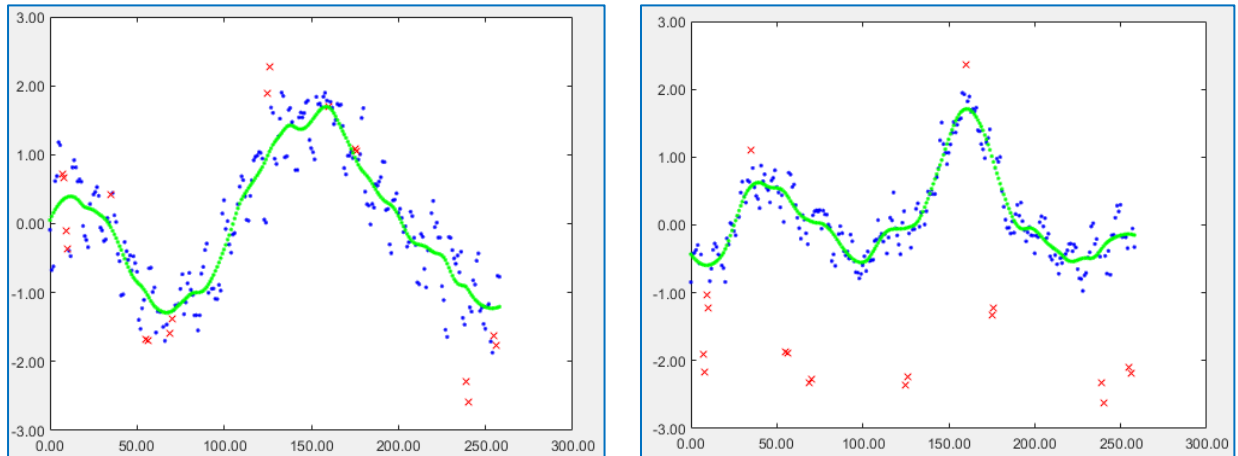


Figure 2.7 Deviation along the linear trend  $TrDL$  and deviation across the linear trend  $TrDC$  after deletion of spikes and smoothing

As a result of the calculations, we obtain the smoothed measurement series  $Es$  and  $Ns$  (with previously deleted and interpolated spike values) and a  $QMask$  mask marking the deleted spikes (one bit of the 32-bit  $QMask$  mask, specified in the  $QBit$  parameter).

The results of the function are displayed in the window with the number specified in the  $f$  parameter. Figure 2.8 shows an example of a window with the results of the towed SSS coordinates under processing. Blue color – initial data  $E, N$ ; red color – deleted data (marked as spikes); green color – the result of  $Es, Ns$  processing, returned by the function.

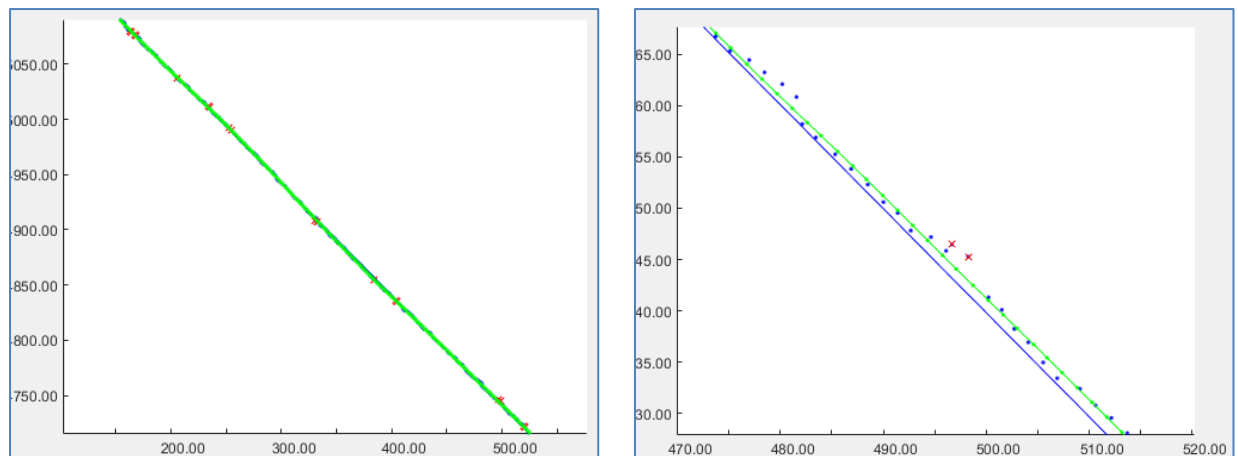


Figure 2.8 Window with the results of the SSS track processing (the figure on the left is scaled)

### 3 Script example for SSS coordinates and heading processing

Script SssCoord01 ([Appendix 1. Script SssCoord01 \(MatLab\)](#)) is written for navigation files with SSS coordinates and heading processing. The files are generated by QINSy (with a time-step of about 1 second) and contain information about the SSS track. The script uses the following data from navigation files: date and time; SSS coordinates calculated from USBL data; SSS heading calculated as the direction from the USBL transponder to the tow point (BTTP).

A brief description of the script operation:

- 1) [NavQinsyLog\\_ReadTxt](#) command – the operator specifies the name of the directory from which all navigation files generated by QINSy are read;
- 2) [tXYAng\\_Edit](#) command – the operator performs timestamp control, SSS coordinates processing, SSS heading processing;
- 3) [NavCODA\\_Write](#) command – the operator writes the processing results to the files in CODA format for further navigation injection to SSS-files using NavInjector (SonarWiz).

#### 3.1 Reading navigation files

Navigation files are generated by QINSy (with a time-step of about 1 second) and contain a sequence of lines with data in the following format: YY-MM-DDThh:mm:ss, DD/MM/YY, hh:mm:ss.sss, JobN, DatumName, ProjName, UtmNum, VesselName, LineName, LineLen, DbName, FixNum, GpsE, GpsN, Height, Head, Sog, Cog, Dcc, Kp, UsblX, UsblY, UsblZ, Text1, Range2Nod, Bearing2Nod, Text2, Text3, DtmDepth.

The script command to read the navigation files is as follows:

```
{'NavQinsyLog_ReadTxt','e:\tmp\'};SssCoord01;
```

where 'e:\tmp\' – the path to the folder containing the navigation files to be processed.

#### 3.2 Processing

Under procession, calculations are made using the following data:

DD/MM/YY – date;

hh:mm:ss.sss – time;

GpsE – easting;

GpsN – northing;

Bearing2Nod – the direction calculated by QINSy from the USBL transponder to the SSS tow point.

The script command to process an SSS track is as follows:

```
{'tXYAng_Edit',[],0,[40 40]};SssCoord01;
```

where

[] – key, which means that all previously loaded data (files) will be processed (numbers of the files to be processed can be entered with a sequence of numbers);

0 – the key, which means that if the data is reprocessed, the mask for deleted spikes will be kept (you won't have to manually delete them again); the key is introduced for multiple processing runs, when selecting the size of the smoothing filter window;

[40 40] – size of the smoothing filter window for coordinates (first value) and for angles (second value).

After starting the "perform pre-processing" command, the windows are opened sequentially for processing and the data processing control by the position of the SSS.

### Timestamp control

The first window to open after running the command is the timestamp control window (Figure 3.1). This window does not allow you to perform any actions, as it is opened only for visual control. Given the artifacts in the timestamps, it is necessary to analyze the causes of the artifact occurrence, since such data are not qualified as standard.

The timestep control window displays a graph of timestamps with a linear trend removed. To plot it, the timestamps are numbered from 1 to  $n$ ; then a linear trend is calculated, given by the  $y=ax+b$  equation. The graph shows the "irregularity" of the time-step between measurements; the value of the coefficient "a" (indicated in the window name) indicates the average value of the time-step.

The window is closed by pressing any key.

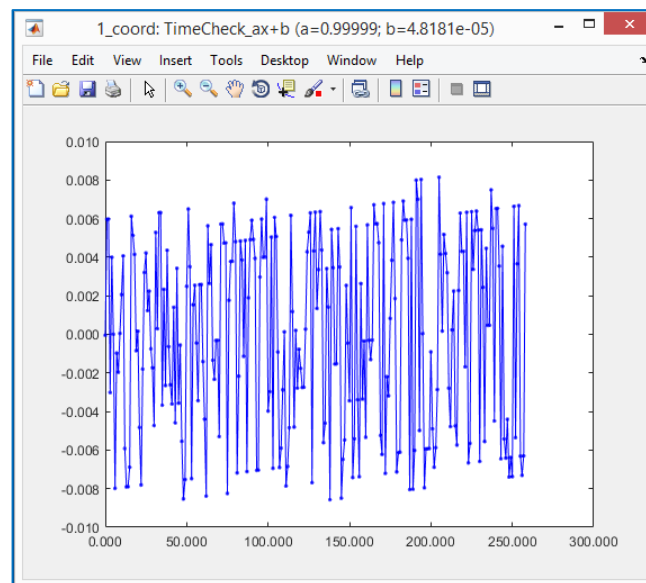
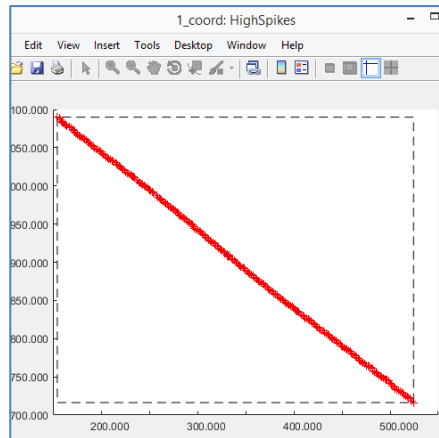


Figure 3.1 Timestamp control window (gFieldsDespike1D function)

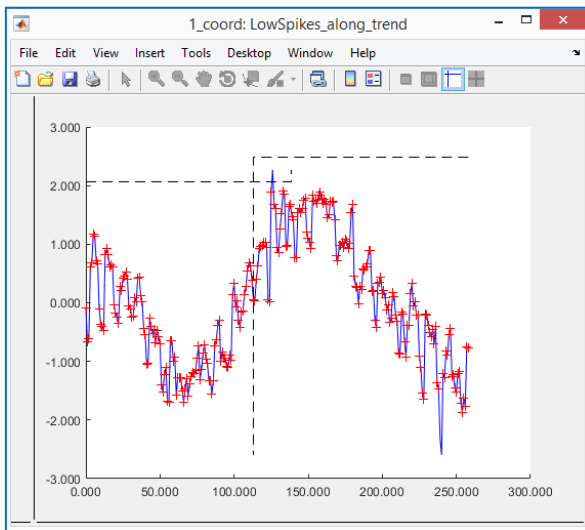
## EN-coordinates processing

The sequence of the windows for processing 2D-coordinates can be seen in [Figure 3.2](#).

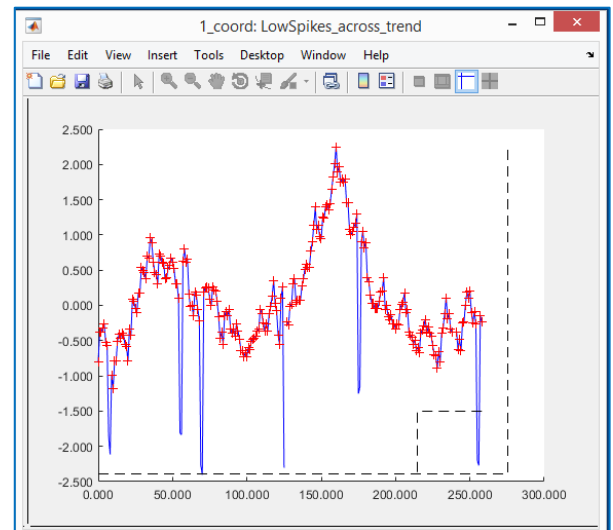
a)



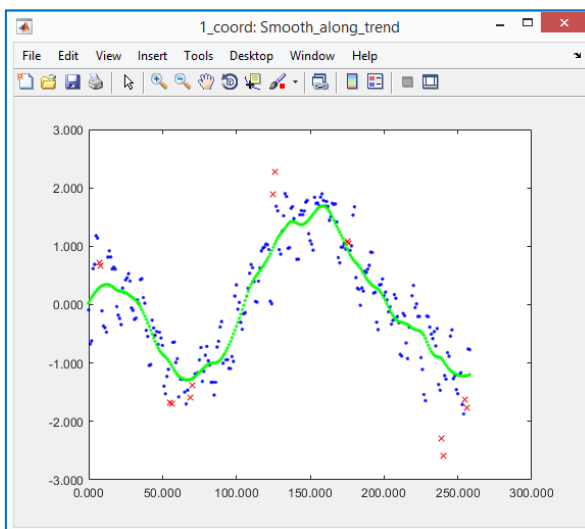
b)



c)



d)



e)

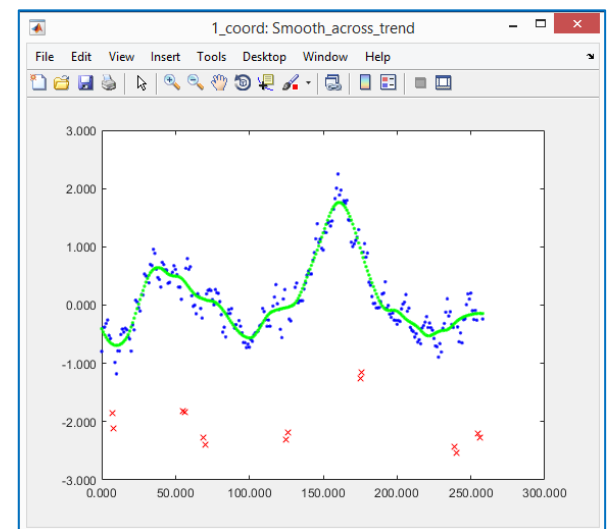


Figure 3.2 2D-Coordinate processing windows for SSS (gFieldsDespike2D function)

Figure 3.2, a shows the first interactive window for high-spikes deleting. Since the survey profiles are assumed to be straight lines (the SSS moves in a straight line), a linear polynomial trend is calculated. Figure 3.2, b-c shows two interactive windows for low-spikes deleting (the linear trend is removed, the coordinate system is rotated along and across the linear trend). The interpolation for deleted spikes is also calculated as linear. After deleting points in the "along the trend" window, they are no longer displayed in the "across the trend" window.

The window width for smoothing/averaging is set in the script command. Figure 3.2, c-d shows two windows that allow us to control the smoothing result. Green color displays coordinate transformations (along and across the trend) after smoothing; deleted spikes are shown in red.

### SSS heading processing (BTTP)

After editing the coordinates, the edit windows for heading are displayed (Figure 3.3).

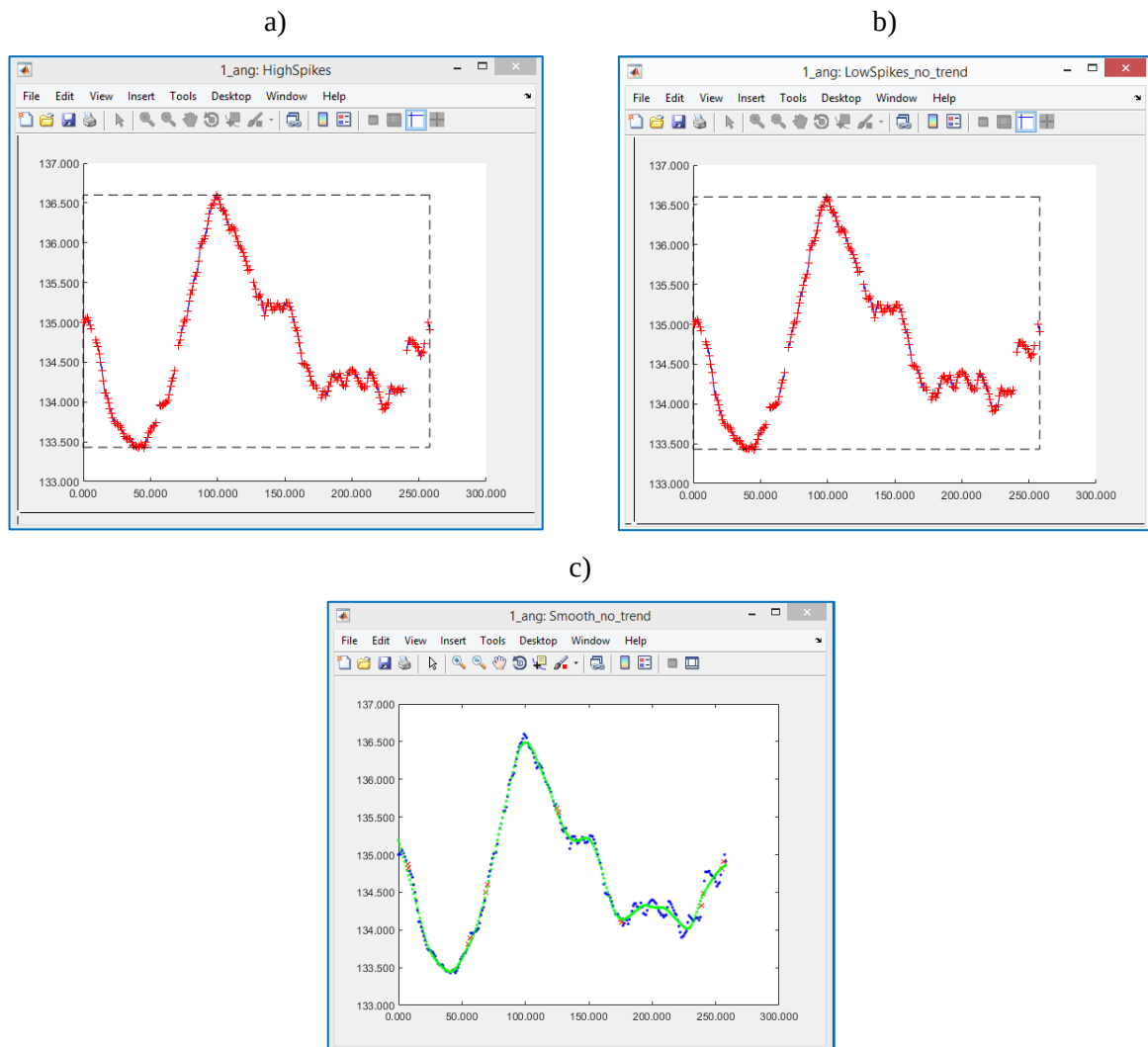


Figure 3.3 SSS heading processing windows (gFieldsDespike1D function)

The timestamp control window is not displayed since the timestamps are similar to those of the coordinates. Spikes previously deleted in coordinates are also removed for the heading and are not displayed.

It would be more correct not to edit for the SSS heading, but to recalculate on the basis of the smoothed coordinates. To implement this, we would need the tow point position data, as well as, possibly, editing and smoothing/filtering for the tow point coordinates. Actually, swaying of the tow point arm is compensated by the towing cable hydrodynamics. So if these movements are not smoothed out, they will cause fluctuations in the towed fish heading. However, since in this case the tow point coordinate data cannot be displayed in the navigation Log-files, a simplified processing procedure is used – spikes are removed and heading values smoothed.

When processing the heading in the script, the trend value is set constant and equal to zero. Therefore, the windows for high and low spikes deleting are similar (Figure 3.3, a-b). Heading values for the deleted spikes are calculated by using linear interpolation.

The window width for smoothing/averaging is set in the script command. Figure 3.3, c shows a window that allows us to control the smoothing result. The green color shows the results of heading smoothing.

### Summary display of graphs for processed files

Under processing, two additional windows (numbered 100 and 101; Figure 3.4) open. They display plots of the SSS position in the plan (EN-coordinates) and SSS heading plots for all processed profiles. When re-processing the same profile with different parameters (for example, smoothing), the windows will show the results for the same profile, thus allowing to compare its processing with different parameters.

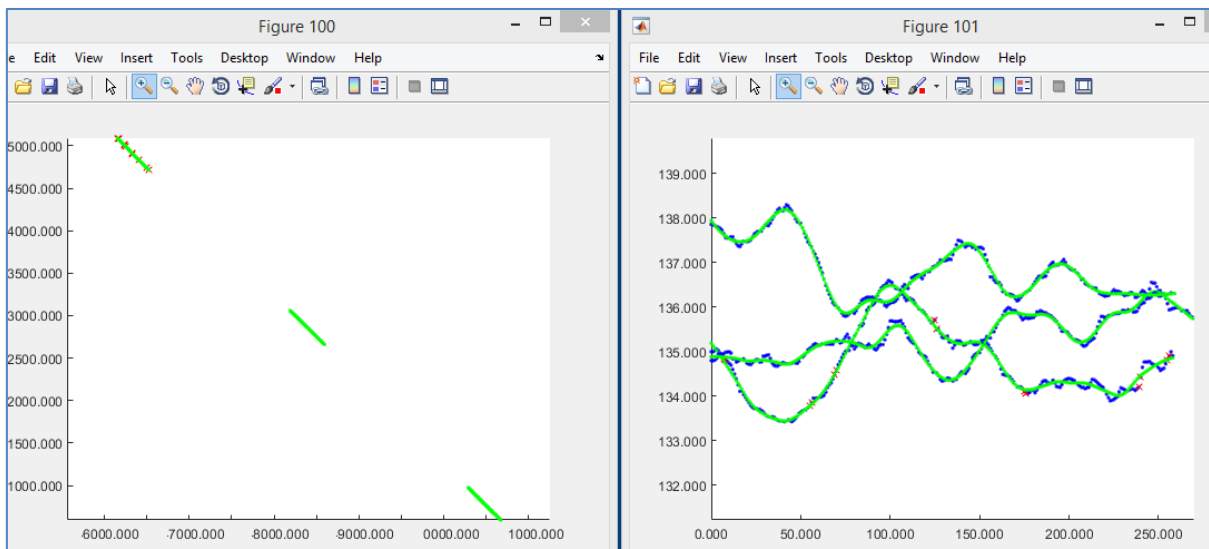


Figure 3.4 Windows with the results of the SSS coordinates and heading processing for the current sequence of profiles

### 3.3 Saving the processing results

The script command to save the processing results is as follows:

```
{'NavCODA_Write', 'e:\tmp\out\',[]};SssCoord01;
```

where

'e:\tmp\out' – the path to the folder for the files to be processed and the results to be written;

[] – a key which means that all previously read data will be written.

Processing results are saved in CODA format, which contains the following data: UnixTime, EventNumber, Easting, Northing, Heading, KP, DCC.

## Conclusions

The document considers a preprocessing graph of measurements received from a certain sensor with time reference. It also contains the description of the functions `gMapPickHandleNan2`, `gFieldsDespike1D`, `gFieldsDespike2D`, which practically implement the processing graph.

As an example, the document considers the processing of coordinates and heading of the SSS positioned using USBL. The script (`SssCoord01`) given as an example, provides the SSS processor with quality and speed to perform the processing of coordinates and prepare files with coordinates for injection using NavInjector (`SonarWiz`).

### Improved processing of towed fish coordinates

To process the USBL data, it is more appropriate not to use the [E,N] coordinates: they are incomplete without the vertical axis data. It would make more sense to recalculate with the applied vessel roll corrections from the initially measured USBL values – the distance to transponder and two bearing angles. When using the specified values, the data are considered exactly in those axes in which they were physically measured; this brings the relative "independence" of the components under consideration and the possibility of their separate processing with the `gFieldsDespike1D` function.

The use of distance to transponder and bearing angles makes it easier to recognize and remove systematic errors. For example, incorrect USBL misalignment angles will not affect the distance to the transponder. However, the effect of the vessel roll will cause roll to "leak" into the USBL transponder bearing angles (as was the case with the position in Figure 1.3). Most likely, this will lead to the effect of the SSS harmonic "yaw" in the direction perpendicular to the transceiver-transponder direction, and, more generally, to movements along a helical trajectory. In this case, the error in the misalignment angles will be smaller while the positioning accuracy will become higher by correcting the misalignment angles and bringing the SSS track to a straight line. However, such calculations are associated with the recalculation of the entire vessel navigation and require the use of appropriate mathematical methods and data flows.

## Appendix 1. Script SssCoord01 (MatLab)

```
%script SssCoord01;
%Script for SSS navigation pre-processing.
%==== There are following script's commands:
%NavQinsyLog_ReadTxt -- QinsyLog navigation read from folder;
%tXYAng_Edit -- Check time, edit XY-navigation and SSS-heading using files numbers;
%NavCODA_Write -- Navigation write using read files numbers;
%==== Files with navigation >>>
%YY-MM-DDThh:mm:ss, DD/MM/YY, hh:mm:ss.sss,JobN,DatumName, ProjName,UtmNum,VesselName,LineName, LineLen,DbName,
FixNum,GpsE, GpsN, Height, Head, Sog, Cog, Dcc, Kp, UsblX,UsblY,UsblZ,Text1,Range2Nod,Bearing2Nod,Text2,Text3,DtmDepth
%2023-03-11T05:11:22.0,11/03/2023,05:11:22.003,,WGS 84 (Greenwich), (WGS 84),31,GEO,5183_MBS_C11_1,159.61,0553-5183_MBS_C11_1 -
0001.db,25764,449999.46,6109999.91,-25.07,135.00,2.04,135.17,-3.04,-258.31,2.65,-109.72,24.85,9999,76.10,314.97,SSSdpt,SSSalt,
%==== CODA-formatted navigation >>>
%UnixTime, EventNumber, Easting, Northing, Heading, KP, DCC
%1662968246.19 10003 699999.29 6499922.52 174.39 1.667 -31.77 * * * *
%1662968247.19 10004 699999.46 6499921.10 174.37 1.665 -31.60 * * * *
%1662968248.19 10006 699999.70 6499919.64 174.36 1.664 -31.49 * * * *

gKey=ans;
%===Navigation Edit=====
if strcmp(gKey{1},'NavQinsyLog_ReadTxt') %QinsyLog navigation read from folder
    %{'NavQinsyLog_ReadTxt','d:\temp\coord\'};SssCoord01;
    try rootD1=gKey{2};catch,rootD1=input('QinsyLog folder name=');end
    dz=dir(rootD1);dz([dz.isdir])=[];fName={dz.name};for nn=numel(fName):-1:1,if ~strcmp(fName{nn} (end-
3:end),'.txt'),fName(nn)=[];end;end;LL=numel(fName);

FF=repmat(struct('fName','', 'YY',[], 'MM',[], 'DD',[], 'hh',[], 'mm',[], 'ss',[], 'JobN','', 'DatumName','', 'ProjName','', 'UtmNum',[], 'Vessel
Name','', 'LineName','', 'LineLen',[], 'DbName','', 'FixNum',[], 'GpsE',[], 'GpsN',[], 'Height',[], 'Head',[], 'Sog',[], 'Cog',[], 'Dcc',[], 'Kp',
[], 'UsblX',[], 'UsblY',[], 'UsblZ',[], 'Text1','', 'Range2Nod',[], 'Bearing2Nod',[], 'Text2','', 'Text3','', 'DtmDepth',[],LL,1);

NV=repmat(struct('fName','', 'UnixTime',[], 't',[], 'EventN',[], 'GpsERaw',[], 'GpsNRaw',[], 'GpsE',[], 'GpsN',[], 'HeadRaw',[], 'Head',[], 'Kp'
,[], 'Dcc',[], 'QMask',[],LL,1);
for nn=1:LL,
    fId=fopen([rootD1 fName{nn}], 'r');C=textscan(fId, '%f-%f-
%fT%f: %f: %f%f/%f/%f%f: %f: %f%s%s%s%f%s%f%f%f%f%f%f%f%f%f%f%f%f%f%f%f', 'Delimiter', ',', 'MultipleDelimsAsOne', 0, 'EndOfLine'
, '\r\n');fclose(fId);

FF(nn)=struct('fName',fName{nn}, 'YY',C{1}, 'MM',C{2}, 'DD',C{3}, 'hh',C{4}, 'mm',C{5}, 'ss',C{12}, 'JobN',C{13}, 'DatumName',C{14}, 'Pro
jName',C{15}, 'UtmNum',C{16}, 'VesselName',C{17}, 'LineName',C{18}, 'LineLen',C{19}, 'DbName',C{20}, 'FixNum',C{21}, 'GpsE',C{22}, 'GpsN',
C{23}, 'Height',C{24}, 'Head',C{25}, 'Sog',C{26}, 'Cog',C{27}, 'Dcc',C{28}, 'Kp',C{29}, 'UsblX',C{30}, 'UsblY',C{31}, 'UsblZ',C{32}, '
Text1',C{33}, 'Range2Nod',C{34}, 'Bearing2Nod',C{35}, 'Text2',C{36}, 'Text3',C{37}, 'DtmDepth',C{38});
```

```

L=~isnan(FF(nn).GpsE);Ut=gNavTime2Time('DxSd2Ut',datenum(FF(nn).YY(L),FF(nn).MM(L),FF(nn).DD(L)),FF(nn).hh(L).*3600+FF(nn).mm(L).*60+FF(nn).ss(L));
    NV(nn)=struct('fName',fName{nn},'UnixTime',Ut,'t',Ut-
    Ut(1),'EventN',FF(nn).FixNum(L),'GpsERaw',FF(nn).GpsE(L),'GpsNRaw',FF(nn).GpsN(L),'GpsE',FF(nn).GpsE(L),'GpsN',FF(nn).GpsN(L),'HeadRaw',
    ',FF(nn).Head(L),'Head',FF(nn).Head(L),'Kp',FF(nn).Kp(L),'Dcc',FF(nn).Dcc(L),'QMask',uint32(zeros(1,numel(Ut))));
    NV(nn).HeadRaw=unwrap(NV(nn).HeadRaw./180.*pi)./pi*180;NV(nn).Head=NV(nn).HeadRaw;
    end;
    clearvars rootD1 dz fName nn LL fId C L Ut FF
end;
if strcmp(gKey{1},'tXYAng_Edit') %Edit XY-navigation and Angle using read files numbers
    %{'tXYAng_Edit',[],0,[40 40]};SssCoord01;
    try NN=gKey{2};catch,NN=input('File numbers for edit=');end;if isempty(NN),NN=1:numel(NV);end; try
QBitReset=gKey{3};catch,QBitReset=input('QBitReset=');end; try SmoothPn=gKey{4};catch,SmoothPn=input('SmoothPn=');end; QBit=[1 2 3];
    for nn=NN,

[NV(nn).GpsE,NV(nn).GpsN,NV(nn).QMask]=gFieldsDespikes2D(NV(nn).t,NV(nn).GpsERaw,NV(nn).GpsNRaw,NV(nn).QMask,QBit(1),QBitReset',{'poly',
1},{'linear','extrap'},{SmoothPn(1),'loess'},100,[num2str(nn) '_coord: ']);
    NV(nn).QMask=bitset(NV(nn).QMask,QBit(2),bitand(NV(nn).QMask,2.^(QBit(1)-1))); %set to nan Angle for XY were cut

[NV(nn).Head,NV(nn).QMask]=gFieldsDespikes1D(NV(nn).t,NV(nn).HeadRaw,NV(nn).QMask,QBit(2),QBitReset',{'const',0},{'linear','extrap'},{SmoothPn(2),'loess'},101,[num2str(nn) '_ang: '],0);
    end;
    clearvars NN QBit QBitReset SmoothPn nn
end;
if strcmp(gKey{1},'NavCODA_Write') %Navigation write using read files numbers
    %{'NavCODA_Write','d:\temp\coord\out\',[[]];SssCoord01;
    try rootD1=gKey{2};catch,rootD1=input('NavCODA output folder name=');end; try NN=gKey{3};catch,NN=input('File numbers for
export=');end;if isempty(NN),NN=1:numel(NV);end;
    for nn=NN,
        z1=NV(nn).Head;z1(NV(nn).Head<0)=360+z1(NV(nn).Head<0);z2=zeros(size(z1));
        fid=fopen([rootD1 NV(nn).fName(1:end-4) '_nc.txt'],'w');fprintf(fid,'%0.2f %0.0f %0.2f %0.2f %0.2f %0.3f %0.2f * * *
*\r\n',[NV(nn).UnixTime;NV(nn).EventN;NV(nn).GpsE;NV(nn).GpsN;z1;z2;z2]);fclose(fid);
        end;
        disp('Processing exported.');
```

```
clearvars rootD1 NN nn z1 z2 fid
end;

%mail@ge0milib.com 30/03/2023
```