

**MEASUREMENTS  
PRELIMINARY  
PROCESSING ON THE  
EXAMPLE OF THE  
TOWED SSS  
COORDINATES**

**FUNCTIONS GMAPPICKHANDLENAN2,  
GFIELDSESPIKE1D,  
GFIELDSESPIKE2D**

**REV. 202303**

Ivan V. Dmitriev  
30.03.2023

## *Contents*

<b>Quick start: processing coordinates of the SSS positioned by USBL (SssCoord01 script) .....</b>	<b>4</b>
<b>Introduction.....</b>	<b>7</b>
<b>1 Pre-processing graph.....</b>	<b>8</b>
1.1 Checking the timestamps 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 Generalized graph of measurements preprocessing .....	15
<b>2 Practical realization of the preprocessing graph.....</b>	<b>17</b>
2.1 Graphical interface for spikes deleting (gMapPickHandleNan2) .....	17
2.2 Preprocessing of scalar measurements (gFieldsDespike1D) .....	19
2.3 Preprocessing of 2D coordinates (gFieldsDespike2D) .....	21
<b>3 Script example for SSS coordinates and heading processing.....</b>	<b>25</b>
3.1 Reading of navigation files .....	25
3.2 Processing .....	25
3.3 Saving of processing results.....	30
<b>Conclusions.....</b>	<b>31</b>
<b>Appendix 1. Script SssCoord01 (MatLab).....</b>	<b>32</b>

## Figures list

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 with SSS coordinates .....	8
Figure 1.2 Example of processing SSS coordinates; (a) SSS track in EN coordinates; (b) a cross-profile deviation component relative to a linear trend; (c) smoothed series without deleting low-spikes; (d) smoothed series with low-spikes deleted .....	10
Figure 1.3 "Leakage" of the roll of a vessel into the coordinates of the towed fish, positioned by USBL .....	14
Figure 1.4 Measurement generalized preprocessing graph.....	15
Figure 2.1 Tools for removing measurements from a series (function gMapPickHandleNan2) .....	17
Figure 2.2 Setting a marker with a label (gMapPickHandleNan2 function).....	18
Figure 2.3 Timestamp control window (gFieldsDespike1D function) .....	19
Figure 2.4 Window with SSS heading processing results (gFieldsDespike1D function) .....	20
Figure 2.5 Window with SSS track in EN coordinates (gMapPickHandleNan2 function) .....	21
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).....	23
Figure 2.7 Deviation along the linear trend TrDL and deviation across the linear trend TrDC after deleting of spikes and smoothing.....	24
Figure 2.8 Window with the results of the SSS track processing (the figure on the left is scaled) .....	24
Figure 3.1 Timestamp control window (gFieldsDespike1D function) .....	26
Figure 3.2 2D-Coordinate processing windows for SSS (gFieldsDespike2D function).....	27
Figure 3.3 SSS heading processing windows (gFieldsDespike1D function).....	28
Figure 3.4 Windows with the results of processing SSS coordinates and heading for the current sequence of profiles .....	29

## List of abbreviations and acronyms

BTTP	bearing to tow-point
MatLab	programming environment
MBSE	multibeam echo sounder
NaN	not a numeric
NavInjector	SonarWiz utility for injecting coordinates into SSS files
QINSy	software package for navigation
SBP	subbottom profiler
SonarWiz	SSS data processing software
SSS	side scan sonar
USBL	ultra-short baseline hydro-acoustic positioning system

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

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

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

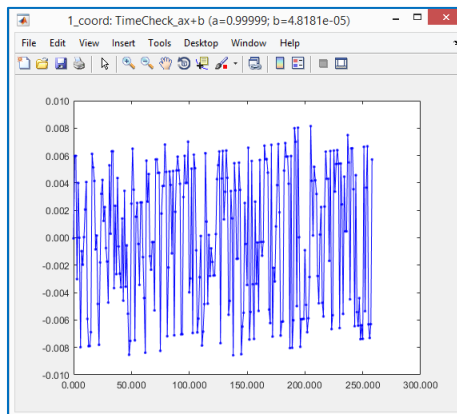
2) Script command: loaded data (heading and coordinates) processing.

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

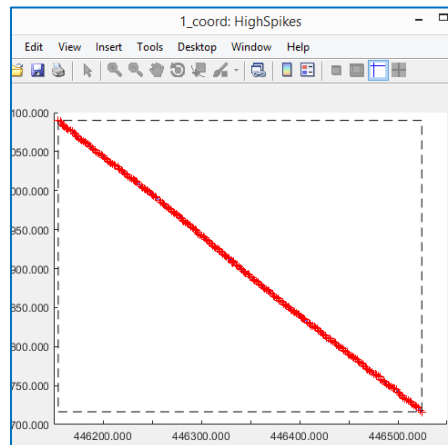
Smoothing window width for coordinates – 40, for a heading SSS – 40.

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

**2.1) Visual control window** of timestamps with linear trend removed (to close any "visual control window" – press any key).



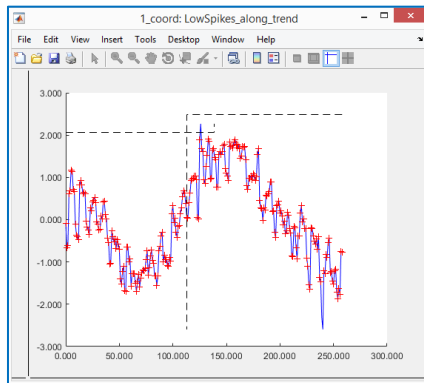
**2.2) Interactive window** for deleting *high spikes* coordinates.



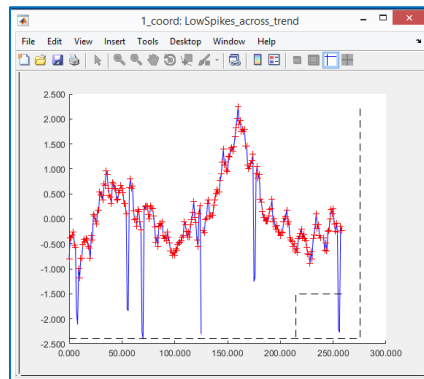
Basic keys for editing (for all interactive windows):

- left mouse button – setting limiter 1 for selecting points of a series of measurements;
- right mouse button – setting limiter 2 for selecting points of a series of measurements;
- mouse wheel – "delete" (mark) of measured points between the limiters;
- 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 "q" – the exit from the measurement deletion mode and closing of the window.

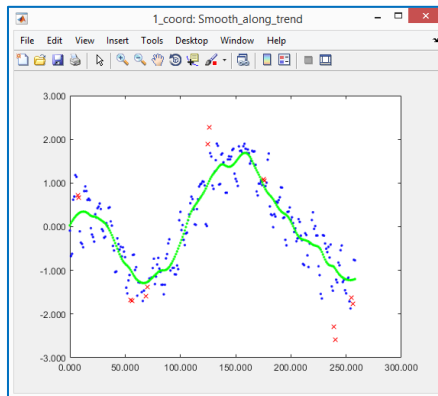
2.3) Interactive window for deleting *low spikes* coordinates (plot of deviations along the profile).



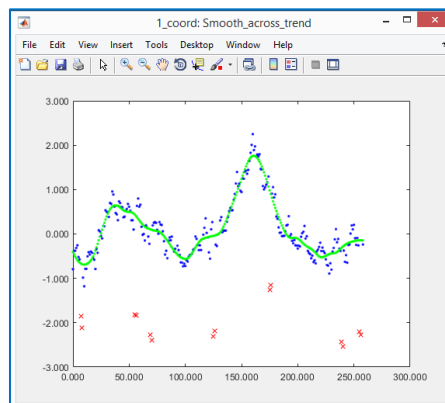
2.4) Interactive window for deleting low spikes coordinates (plot of deviations across the profile).



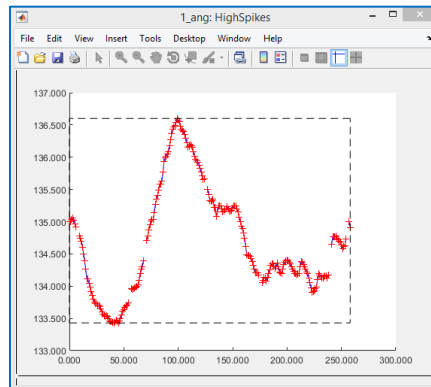
2.5) Visual control window of smoothing for coordinates (deviations along the profile).



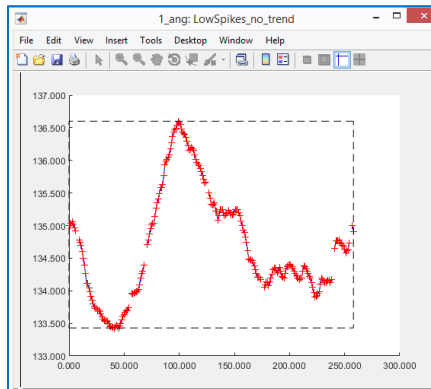
2.6) Visual control window of smoothing for coordinates (deviations across the profile).



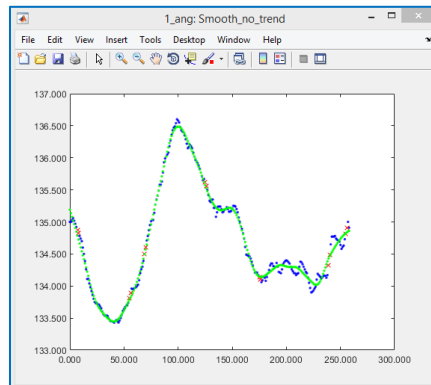
2.7) Interactive window for deleting high spikes for the SSS heading.



2.8) Interactive window for deleting low spikes for the SSS heading.



2.9) Visual control window of smoothing for the SSS heading.



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 pre-processing graph for measurements received from some sensor with time reference.

Initially, the text was prepared as a description of the coordinates processing of the towed SSS positioned by using USBL. The time reference allows us to switch from consideration of coordinates to consideration of the speed of the towed fish and to use the assumption that "the towed fish moves progressively along a smooth trajectory and without abrupt changes in its position"; this assumption was used as the basis for the processing graph. In generalizing the processing graph to the case of "arbitrary time-referenced measurements," the following points were added to the text:

- the first chapter describing the generalized preprocessing graph;
- description of series of scalar measurements preprocessing;
- description of timestamp representativeness control.

The first chapter deals with the problems of preliminary processing of measurements (spikes removal/disposal, filtering, etc.) and methods for solving them. A generalized preprocessing graph is presented.

The second chapter discusses the implementation of the measurement preprocessing graph. Describes a graph (gFieldsDespike1D function) intended for processing series of a scalar measurements with time reference; for example, temperature sensor measurements, pressure/depth sensor measurements, towed fish altimeter measurements. Describes a graph (gFieldsDespike2D function) intended for processing series of 2D-coordinates measurements with time reference; for example, the coordinates of a towed fish positioned with using USBL.

The third chapter describes a script designed to process the SSS position (data import, processing of SSS coordinates and heading, data export).

# 1 Pre-processing graph

Pre-processing of measurements series includes the following tasks:

- checking the representativeness of timestamps;
- spikes removal/disposal (gross deviations);
- averaging/smoothing of measurements;
- elimination of "systematic errors".

Let's consider each task in more detail, after which we present the structure of the generalized preprocessing graph. In the context of deleting spikes, we will touch the issues of analyzing the deviation of measurements upon from the theoretical model.

## 1.1 Checking the timestamps representativeness

Most measurements in marine geophysical surveys are performed with reference to time with a constant step (magnetometry, SSS, SBP) or time steps close to constant (USBL). Before starting processing, it is necessary to make sure that there are no artifacts in the timestamps. The reason of such artifacts is almost always associated with errors in the hardware settings or format settings for the data flow. The influence of these artifacts can lead to distortion in the processed data and incorrect conclusions about the nature of such distortions. An example of a timing error and its consequences is shown in Figure 1.1.

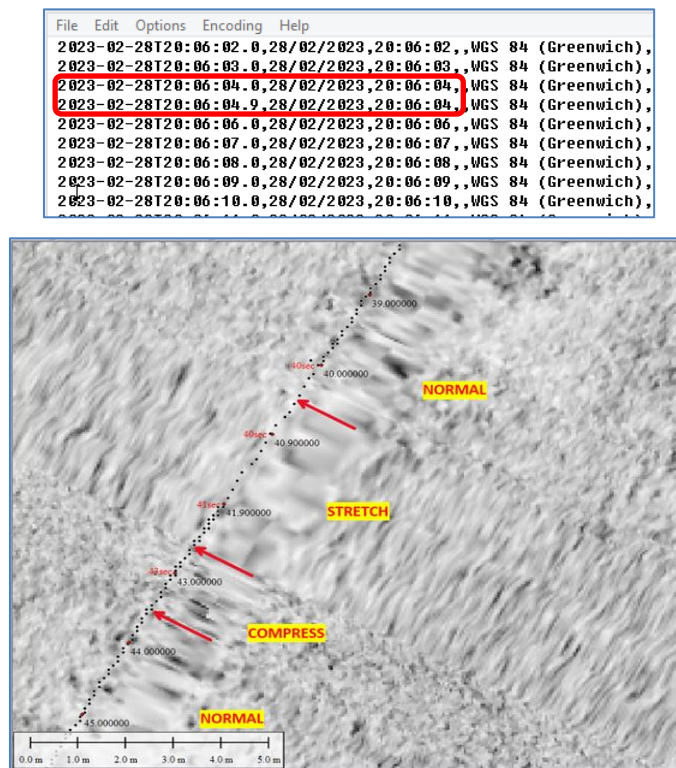


Figure 1.1 The appearance of stretch zones on the sonar mosaic due to the lack of decimal places for the timestamp in navigation files with SSS coordinates



The upper part of Figure [1.1](#) shows a fragment of the navigation file with SSS coordinates that were injected into files with SSS data. Due to an operator's error when setting up QINSy, the timestamp is recorded without decimal digits. This results in the periodic artifacts in the measurement time – the repetition of the time value (20:06:04, 20:06:04), followed by the skipping of the second count (20:06:06). Due to the described artifacts, "stretch" zones appear on the sonar mosaic (a few SSS pings between coordinate points with a repetition of a second), alternating with subsequent "compression" zones. Such zones are shown at the bottom of Figure [1.1](#).

It is possible to process timestamps with artifacts (for instance, for the given example, we can delete duplicate seconds and calculate the deleted timestamps using linear interpolation), but the best decision is to find and remove the causes of artifacts.

## 1.2 Spikes deleting

*Spikes* are caused by third-party causes not related to the measured values' physics. Such spikes are visually recognized by a sharp and significant deviation from the general pattern of the measured values (for example, a single spike in the coordinates of the towed fish by 50 meters; high-amplitude noise on the MBES data when the thrusters of a vessel are turned on). It is impossible to single out a "useful measurement component" against the background of an additive component with a spike; the most acceptable is to delete such a measurement from the series in order to exclude its influence on subsequent processing.

In the first approximation, spikes can be considered as a short-period component of the signal. Therefore, to identify and delete spikes, it seems reasonable to first remove the *long-period trend* first, which can mask relatively low-amplitude spikes. If the values of the processed series of measurements are used to calculate the long-period trend, then the algorithm with the spikes deleting in several stages works well:

- 1) The most high-amplitude *high-spikes* are being deleted, they are clearly visible on the visualizing of the processed signal. Such spikes can affect the calculated trend and greatly distort its shape;
- 2) For a series of measurements with the deleted high-amplitude spikes, a long-period trend is calculated, after which it is subtracted from the series of measurements. For better visualization of spikes, transformations can be calculated for the trend and the original data. An example of transformations – "*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 number of measurements with a removed long-period trend, 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).

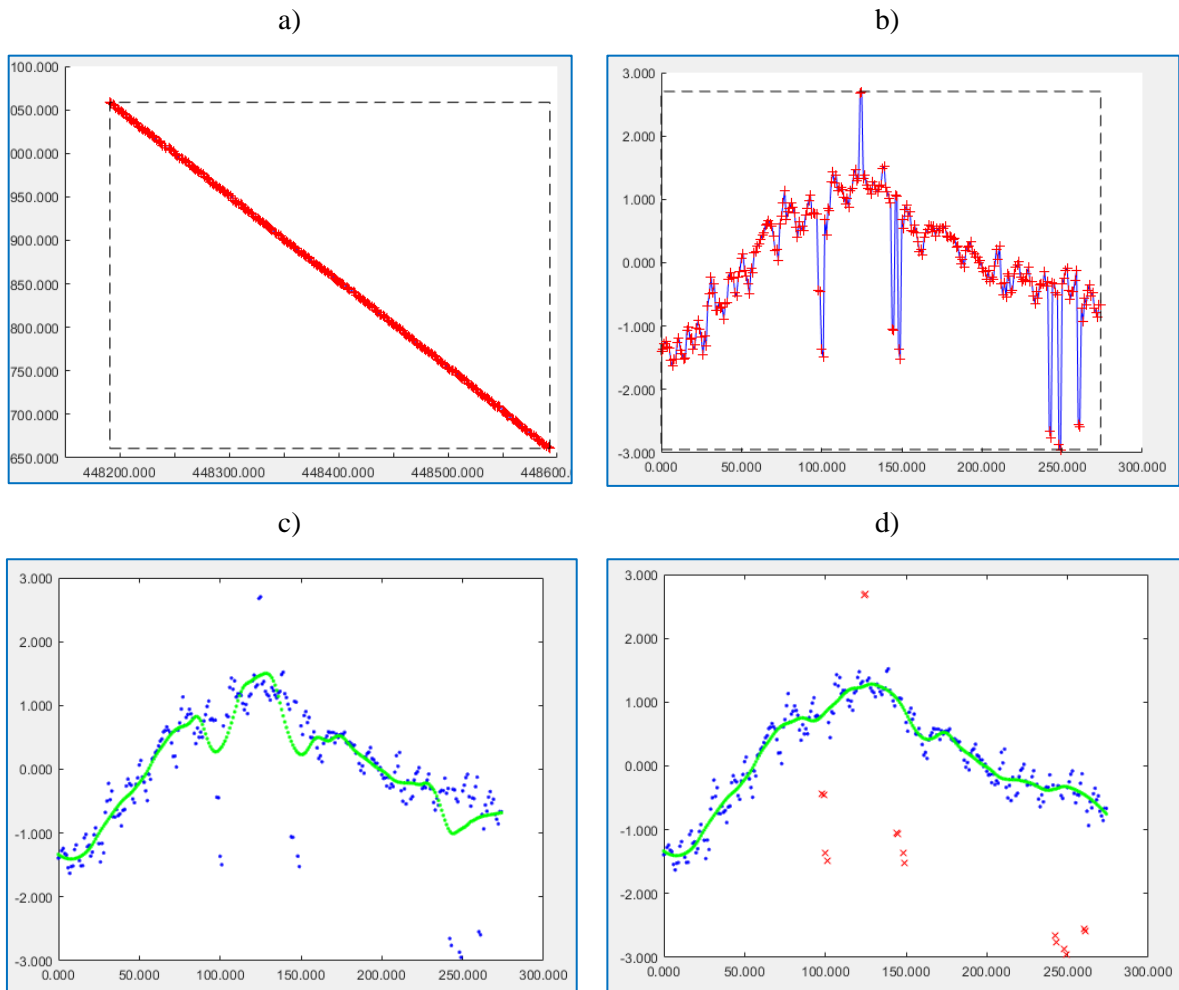


Figure 1.2 Example of processing SSS coordinates; (a) SSS track in EN coordinates; (b) a cross-profile deviation component relative to a linear trend; (c) smoothed series without deleting low-spikes; (d) smoothed series with low-spikes deleted

Figure 1.2 shows an example of processing production of the SSS data. Figure 1.2, a shows the SSS track in rectangular coordinates. For Figure 1.2, b, the coordinate system is rotated and the linear trend is removed in order to visualize low-spikes best 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 can be seen from the figures, low-spikes cause smooth shifts in coordinates with value of up to 1 meter. On sonograms, it will be shown up as a visually indistinguishable smooth shift for sonar pings; objects that fall into the zone of low-spikes influence will be distorted and shifted by up to 1 meter.

In the context of "spikes deleting", the action of "*deleting/removing a measurement* from a series" needs to be more clearly defined. Deleting a measurement from a series can be done in the following ways:

- Delete the measurement completely, including the timestamp, and proceed with further processing without the timestamp that was deleted. The problem arises when multiple sensor measurements

are linked to the same timestamp. For example, for magnetic surveying, measurements of a magnetic sensor, a depth sensor and an altimeter are linked to one timestamp. Deleting the timestamp due to "bad" (not conditioned/substandard) altimeter measurements entails deleting "good" (conditioned/standard) magnetic field measurements. In addition, the timestamp itself carries "good" information about the measurements period. For the reasons described, it was decided to disclaim "delete with a timestamp";

- We can delete the values of a dimension by replacing it with NaN (the special value not a numeric for floating point numbers). However, additional nuances arise here in mathematical processing; for example, when calculating transformations in a sliding window (filtering, averaging), including the NaN-value in the calculations entails assigning the NaN-value to the result of the calculations. That is, if we delete one point in the coordinate measurements (assign the NaN), and then average in a sliding window of 50 points, then we will get a "data gap" (a sequence filled with NaN) with a size of 99 points. Therefore, in all calculations, it is necessary to take into account NaN values (for example, eliminate them before calculations) and adapt all algorithms to the "variable step measurement series";
- We have chosen the deleting/removal method when "bad" values are marked, and after marking is completed, they are "created" using interpolation. Probably, it would be more correct to speak not about deleting/removal, but about *replacing measurements* with interpolated values, however, by analogy with "removing the high-frequency component" (when, in fact, the measurement values are also recalculated and replaced), we will use the words "delete or remove". The information that the values are "bad" is kept in a special 32-bit mask. In this case, during further processing, the algorithms for a constant measurement step can be used, 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) and interpolation gives a good 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 deleted/removed (interpolated) data when visualizing for a processor man.

### 1.3 Analysis of deviations from the theoretical model

In existence of a mathematical model that allows one to calculate the processed measurements theoretically, it is interesting to "*simulate/modeling the series*" instead of "calculating the trend based on the processed series". In this case, the difference (between the simulated/modeled and measured series) will show us the deviations of the measurements from the theoretical model. If the theoretical model describes the measured series well, then the spikes (for the spikes' delete procedure) will be displayed more clearly. In addition, we get a tool for assessing "*deviations from the theoretical model*" – it will be able to identify and analyze measurement deviations from the theoretical model in medium and long periods. We also get a tool for estimating the error of the theoretical model (if it is not accurate).

To pre-process towed fish coordinates from 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. Instead of information about the current, coarser calculations based on survey line direction and vessel's diametric plane direction can be used. Based on the results of such calculations, it is possible to analyze the difference between the theoretical position of the towed fish and the measured position; analyze such moments as "the stability of the towed fish during winch operation" or "the presence of systematic USBL errors" (an example is given further in the text in Figure 1.3).

### 1.4 Averaging/smoothing of measurements

Averaging/smoothing of measurements is performed to reduce the level of noise (similar to high frequency component). If we describe the reason for using averaging without mathematical rigor, then we assume that random measurement errors are normally distributed (additive Gaussian noise with zero mathematical expectation), and the measured values within the "sliding averaging window" varies weakly (and/or has an additional linearly) compared to the spread of high-frequency noise component. Therefore, the measurement error will decrease by a factor of  $\sqrt{n}$  when averaged over  $n$  measurements (three times for a window of 10 measurements). Visually, this is defined by the processor as "a good reduction in the level of high-frequency noise with an acceptable distortion of the useful signal."

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

- The measurement error will be "smeared" (redistributed) within the averaging window, while measurements with a relatively large error will become "more accurate", and the remaining measurements will become "less accurate". If spikes were not deleted beforehand, then all measurements within the averaging window can turn into "measurements with a relatively large error".
- A series of measurements become "smooth", without significant deviations between adjacent measurements. This property may be a characteristic of the measured quantity, and the reduction to

the smoothness may be an important condition for further processing. For example, noise with a spread of 0.5-1 meter in the coordinates of a towed SSS is not physically real (the movement of an outboard towed fish is due to the laws of hydrodynamics and its own inertia, which limits sharp changes in position and heading); the presence of this noise will lead to the effect of 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. For such an analysis, it is necessary to have an idea about the mathematical model of measurements, calibrations and errors of measuring fishes. Most often, systematic errors are due to incorrect calibration; less often, errors are associated with the presence of unaccounted factors in the measurement model (that is, with the error of the model).

In relation to the determination of the coordinates of the towed fish according to USBL, such errors are almost always caused by incorrect calibrations of the systems associated with the positioning of the SSS. Let's look at a few examples:

- Constant summand to SSS ping angle if SSS heading is defined as BTTP (direction from USBL transponder to SSS tow point). In this case, the misalignment angles for the USBL in the horizontal plane are likely to be incorrect. At any distance from the vessel of the towed fish, the corrections will be constant and equal to the installation angle error. Also, the error can be associated with a change of rotation angle of the towing cable in the area of the towed fish due to the influence 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 will always curve in one direction on opposite courses.
- Constant summand to the angle of rotation of the SSS pings, if the heading of the SSS is determined by the internal compass of the SSS. In this case, the compass may not be calibrated correctly. It is also possible to take into account the magnetic declination incorrectly or a value similar to the Gaussian convergence in the area of work.
- 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 made to the position of the sonogram in SonarWiz). Such summands can be related to incorrect determination of the USBL misalignment angles. If the angle in the horizontal plane (yaw) is incorrectly determined, then, provided that the towed unit is approximately the same distance from the vessel (which is typical for most surveys), the corrections will be "approximately constant" (their value can be calculated through a right triangle, where the hypotenuse is the distance of the SSS from the USBL).
- "Leakage" of the roll of a vessel (roll, pitch) into the towed SSS coordinates (Figure [1.3](#)). The reason may be incorrect definition of the USBL misalignment angles. In this case, part of the roll/pitch is projected onto the yaw, which is the reason of the appearance of oscillations on the SSS

track, synchronous with the roll/pitch. In this case, the oscillations have a pronounced period (approximately 5-10 seconds), corresponding to the period of the roll of the vessel.

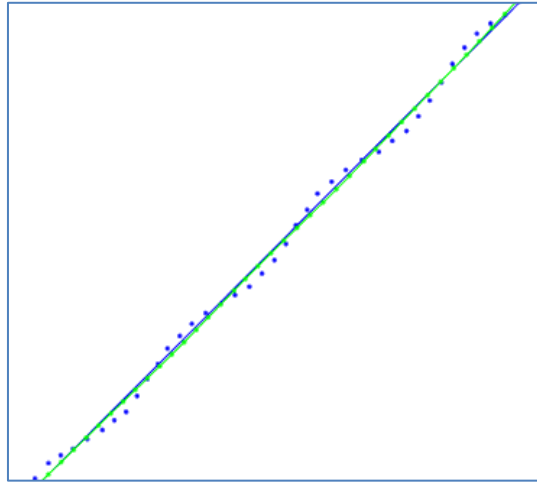


Figure 1.3 "Leakage" of the roll of a vessel into the coordinates of the towed fish, positioned by USBL

As mentioned above, there is no universal method for removing a systematic error. In each case, the error must be analyzed, its reasons 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 roll of a vessel (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 slice-window. However, such removal of the error is cosmetic and does not eliminate all the consequences of incorrect USBL calibration – it will also be necessary to introduce a coordinate shift along and across the profile direction, and with a significant change in the length of the towing cable, the introduced 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 Generalized 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.

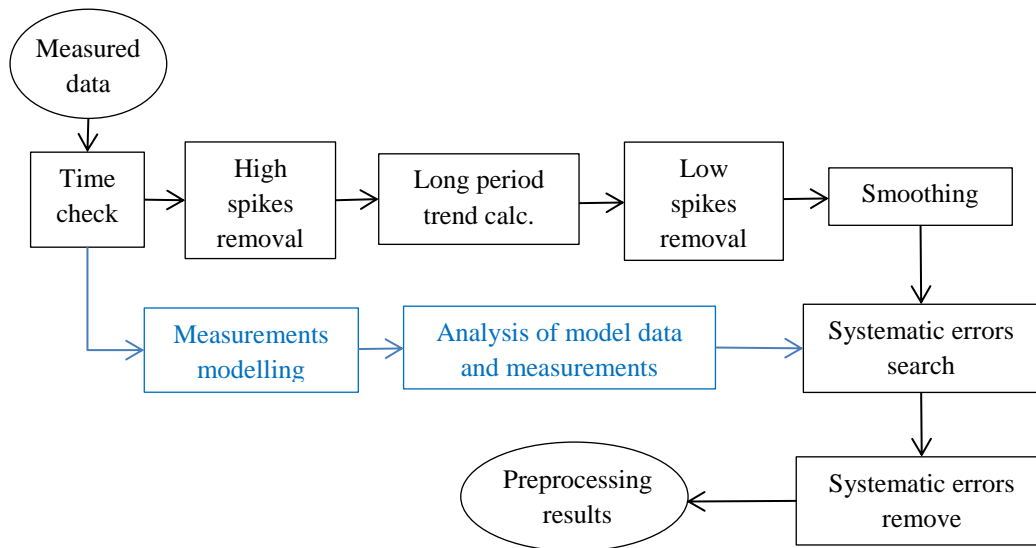


Figure 1.4 Measurement generalized preprocessing graph

Measured data – initial data; a series of time-linked measurements for which analysis and preprocessing is carried out;

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);

Measurements modeling – a preprocessing method based not on "calculating a long-period trend", but on modeling the data like measured based on other independent data. The graph block corresponds to doing such a simulation/modelling;

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

Systematic errors search – identification of systematic errors and the nature of systematic errors (results of primary preliminary processing are used for comparison with other data, comparison with modeling results);

Systematic errors removal – removal of systematic errors; removal methods are determined after systematic errors are identified. These can be "formal" methods that do not address the source

of error (eg, sonogram shift and high-frequency filtering shown in Figure 1.3) or methods that remove the source of error (eg, correct recalibration of USBL for towed fish positioning);  
Preprocessing results –preprocessing result of the measurements 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 remove, Long period trend calc., Low spikes remove, 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 remove, Long period trend calc., Low spikes remove, Smoothing.

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

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

The `gMapPickHandleNan2` function provides the operator with an interactive graphical interface for deleting spikes. After starting the function, a plot is displayed in the XY axes (a series of measurements 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 individual measurements from a series, shown in Figure 2.1.

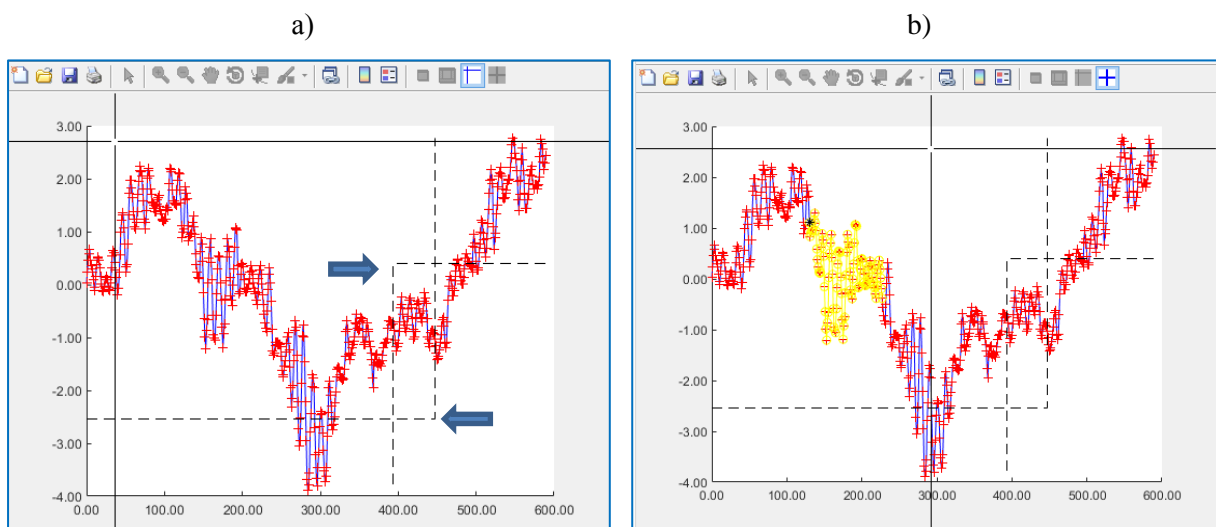


Figure 2.1 Tools for removing measurements from a series (function `gMapPickHandleNan2`)

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. By clicking on the mouse wheel, all points that fall inside the dotted rectangle are deleted from the measurement series (marked as deleted). The second tool (Figure 2.1, b) allows to delete a sequence of points by marking the first and last points of the sequence with the left and right mouse clicks (the selected sequence is marked in yellow); deletion is done by clicking on the mouse wheel. The second tool is useful for highlighting/deleting areas of "scattering" coordinates of the track of the towed fish in the EN axes (we can select parts of the track "tied into knots"). Switching between modes is carried out by the keys "1" and "2".

There is a description of all controls below:

- left mouse button – setting limiter 1 for selecting points of a series of measurements;
- right mouse button – setting limiter 2 for selecting points of a series of measurements;
- mouse wheel – deletion (marking) of measurements points between limiters 1 and 2;
- key "1" – activation of the first tool;
- key "2" – activation of the second tool;
- key "a" – cancel the last of the chain of deletions for the current tool;
- key "s" – apply the last of the chain of deletions 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" – marker setting mode, each marker is signed with a label with a value along the XY axes and a measurement number inside the series (Figure 2.2);
- key "q" – the exit from the measurement deletion mode and closing of the window.

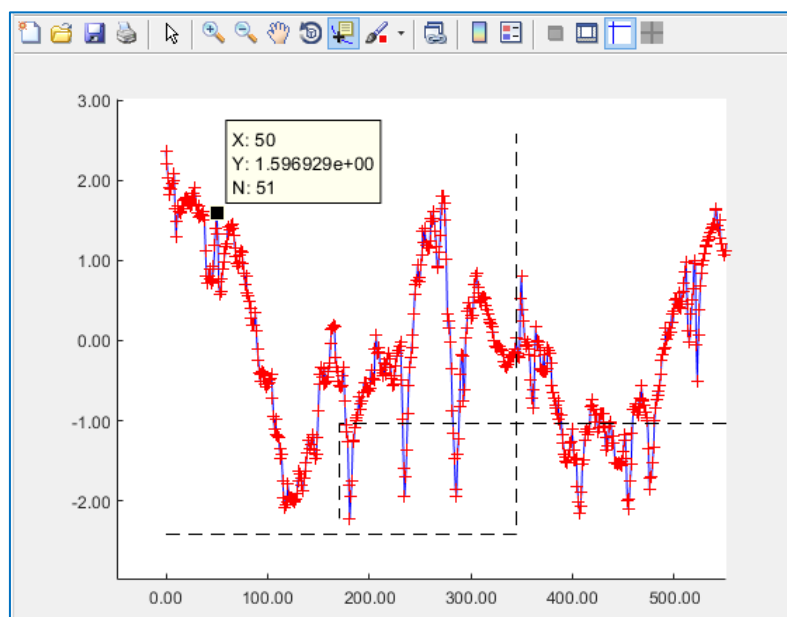


Figure 2.2 Setting a marker with a label (gMapPickHandleNan2 function)

## 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, given by the equation  $y=ax+b$ . The calculation is performed by the least squares method using the `polyfit` function (help in MatLab on the `help polyfit` command). The result is displayed as a plot 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.

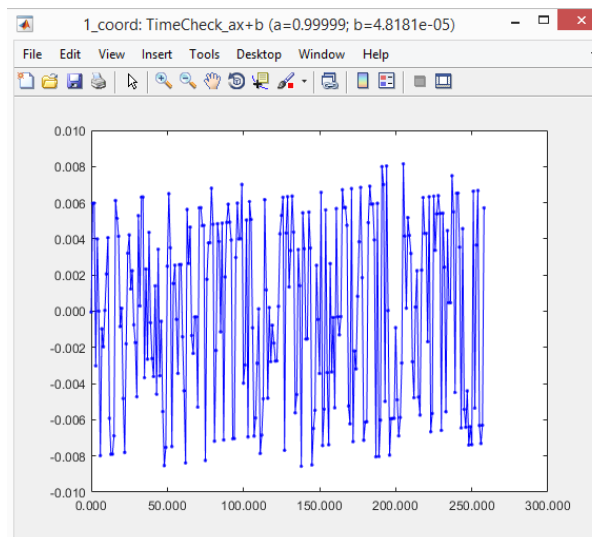


Figure 2.3 Timestamp control window (`gFieldsDespike1D` function)

**High spikes removal.** The value of the  $U$  series is plotted along the  $t$  time axis 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 input parameter  $Tr$ ; the following types of calculations are possible:

- const – trend is not calculated, all trend values are taken equal to the specified number;
- poly – 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 on the `help polyfit` command);

-- smooth – the trend is calculated by averaging in a sliding window using the [smooth](#) function (help in MatLab by command [help smooth](#)); 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 feature is provided for setting the function that calculates the "modeling series" and the subsequent use of the simulation/modeling results as a trend.

**Low spikes removal.** A subtraction is performed from a series of observations 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 original series of measurements, all deleted spikes are replaced by the interpolation results. Interpolation is performed by the [interp1](#) function (help in MatLab on the [help interp1](#) command). The interpolation parameters are set in the input parameter [IntP](#); possible methods: 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 on the [help smooth](#) command); moving, lowess, loess, sgolay, rlowess, rloess smoothing methods are available.

As a result of the calculations, we obtain a smoothed measurement series [Us](#) (with previously deleted and interpolated spikes 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. An example window with the results of SSS heading processing is shown in [Figure 2.4](#). Blue color – original data series [U](#); red color – deleted measurements (marked as spikes); green color – the result of processing [Us](#), issued by the function.

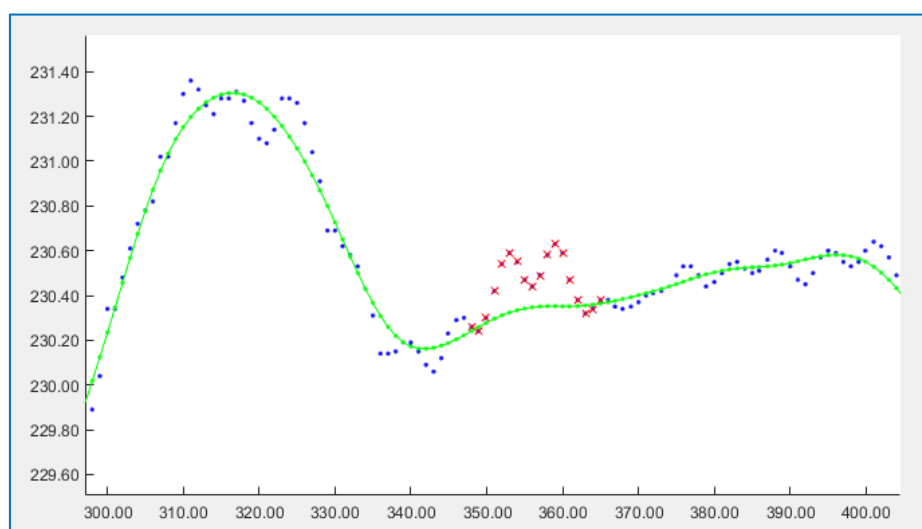


Figure 2.4 Window with SSS heading processing results (gFieldsDespik1D function)

### 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 is shown below:

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

Processing includes the following graph steps (Figure 1.4): 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 performed by the least squares method using the `polyfit` function (help in MatLab on 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 the EN axes, in the `gMapPickHandleNan2` function window (Figure 2.1) for interactive deleting of high-spikes by the operator (Figure 2.5; there are no high spikes for the given example).

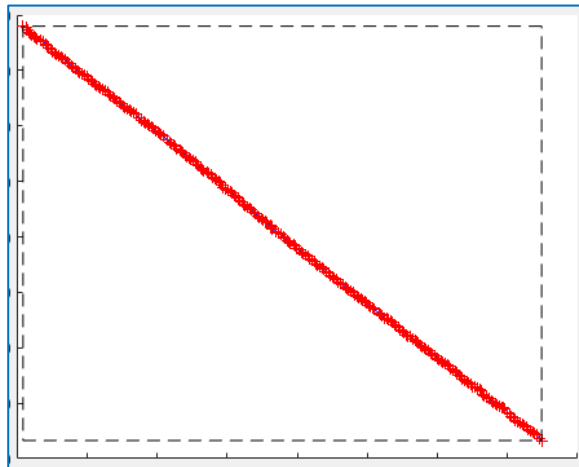


Figure 2.5 Window with SSS track in EN coordinates (`gMapPickHandleNan2` function)

**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-Easing and time-Northing axes. The trend type is set in the input parameter `Tr`; 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 using the least squares method using the `polyfit` function (help in MatLab on the `help polyfit` command);
- smooth – trends are calculated by averaging in a sliding window using the `smooth` function (help in MatLab by command `help smooth`); moving, lowess, loess, sgolay, rlowess, rloess smoothing methods are available.

-- user defined function – trends for E and N are calculated using a MatLab function written by the operator. This feature is provided for setting the function that calculates the "modeling series" and the subsequent use of the simulation/modeling results as a trend. For example, the track of a towed fish can be calculated using the following input data: coordinates of the tow-point, the length of the towing cable, the measurements of the depth (pressure) sensor, the heading of the vessel and the direction of its diametric plane.

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

Subsequent calculations are aimed at the transformation from the EN-coordinates to the coordinate system directed at each point tangentially to the long-period trend. This is done for better visualization of low-spikes – the concepts of "*deviations from the trend along the displacement*" and "*deviations from the trend across the displacement*" are introduced. In the first approximation, the "along" deviation corresponds to a change in the distance from the towed fish to the USBL head, and the "across" deviation corresponds to a change in the bearing angle to the towed fish from the USBL head. To move to the coordinates "along" and "across", the following calculations are performed:

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

Based on the obtained series of coordinates "along the displacement" and "across the displacement", 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 polynomial of the first degree) is shown in [Figure 2.6](#). The figure shows low-spikes for the deviation "across the trend". In the EN coordinate axes, these spikes are indistinguishable, even when the track is scaled in the spike area; most clearly these artifacts are manifested precisely with the use of the time axis and in the coordinate axes along and across the trend/movement.

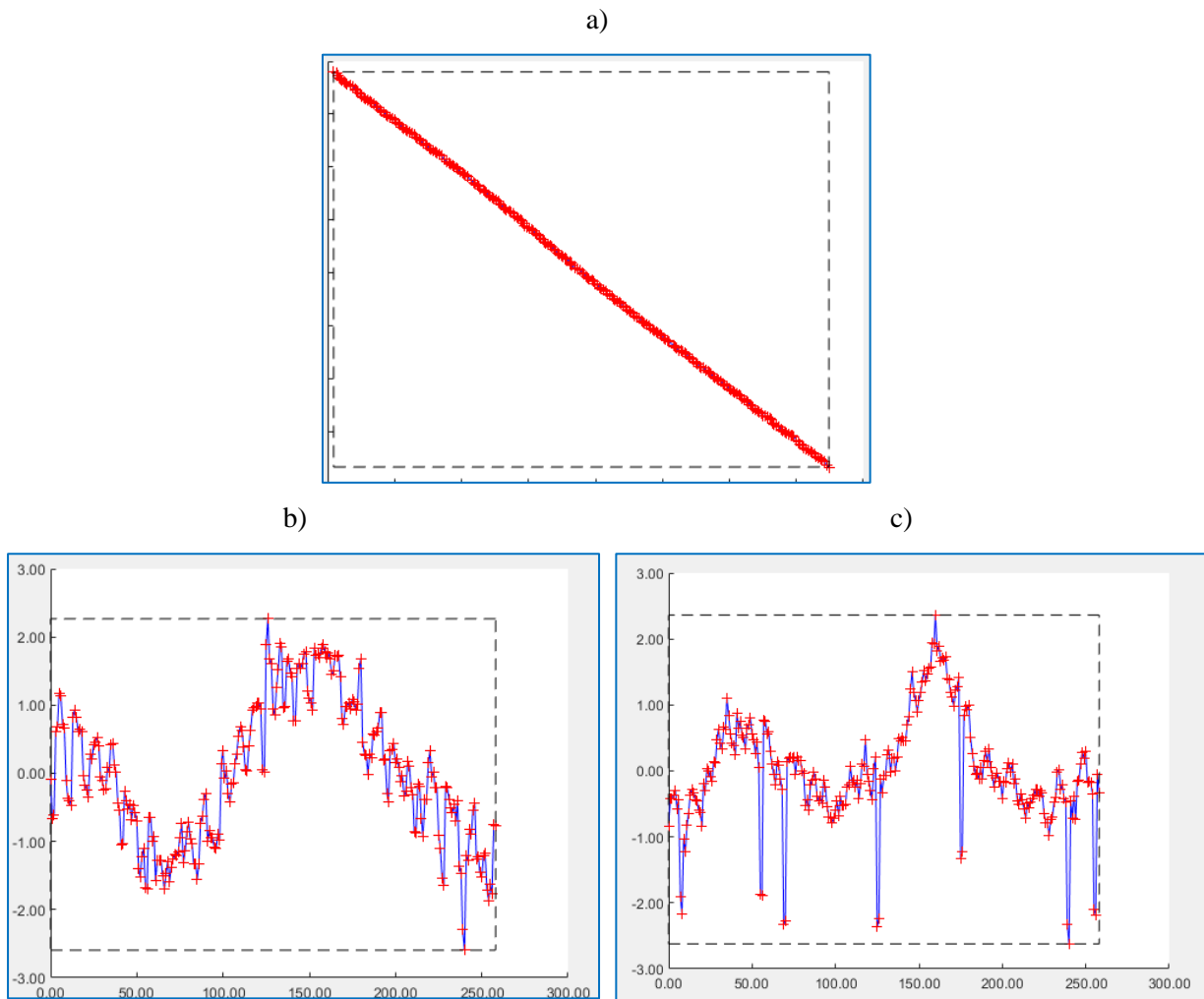


Figure 2.6 Window with SSS track in EN-coordinates (a), deviation along the linear trend  $TrDL$  (b) and deviation across the linear trend  $TrDC$  (c)

In the first approximation, the calculation of a linear trend for coordinates (Figure 2.6, a) can be considered as building a model of the movement of a towed fish 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 show the measured deviation values from the "theoretical model" for a constant speed. The long-period component when "moving along" the profile (Figure 2.6, b) is associated with a change in the speed of the towed fish; the long-period component for "movement across" the profile (Figure 2.6, c) is associated with the deviation of the towed fish from the profile line.

**Smoothing.** For the original measurement series E and N, all deleted spikes are replaced by the interpolation results. Interpolation is performed by the `interp1` function (help in MatLab on the `help interp1` command). The interpolation parameters are set in the input parameter `IntP`; possible methods: 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 on the

help smooth command); moving, lowess, loess, sgolay, rlowess, rioess smoothing methods are available.

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

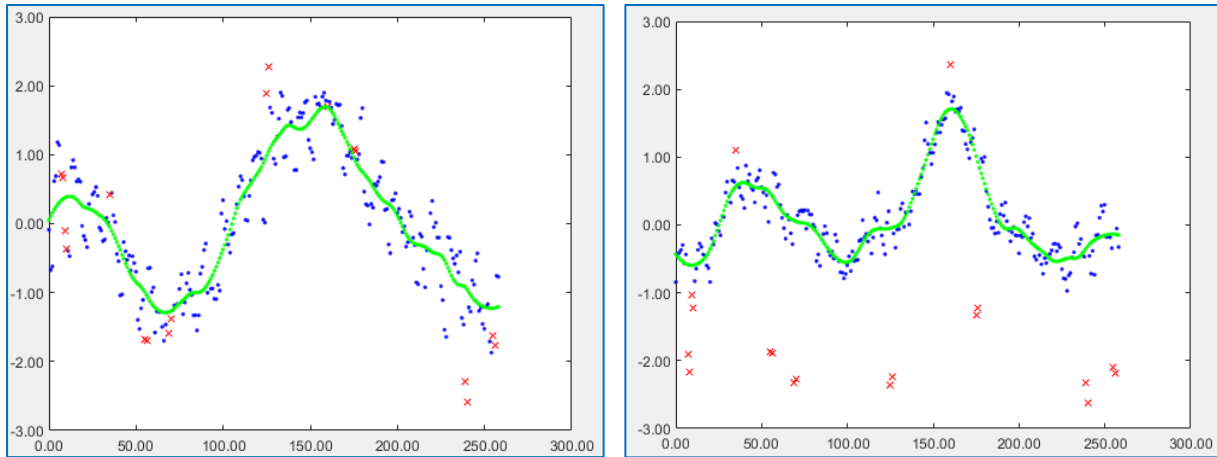


Figure 2.7 Deviation along the linear trend  $TrDL$  and deviation across the linear trend  $TrDC$  after deleting of spikes and smoothing

As a result of the calculations, we obtain smoothed measurement series  $E_s$  and  $N_s$  (with previously deleted and interpolated spikes 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. An example of a window with the results of processing the coordinates of a towed SSS is shown in Figure 2.8. Blue color – initial data  $E,N$ ; red color – deleted data (marked as spikes); green color – the result of  $E_s,N_s$  processing, returned by the function.

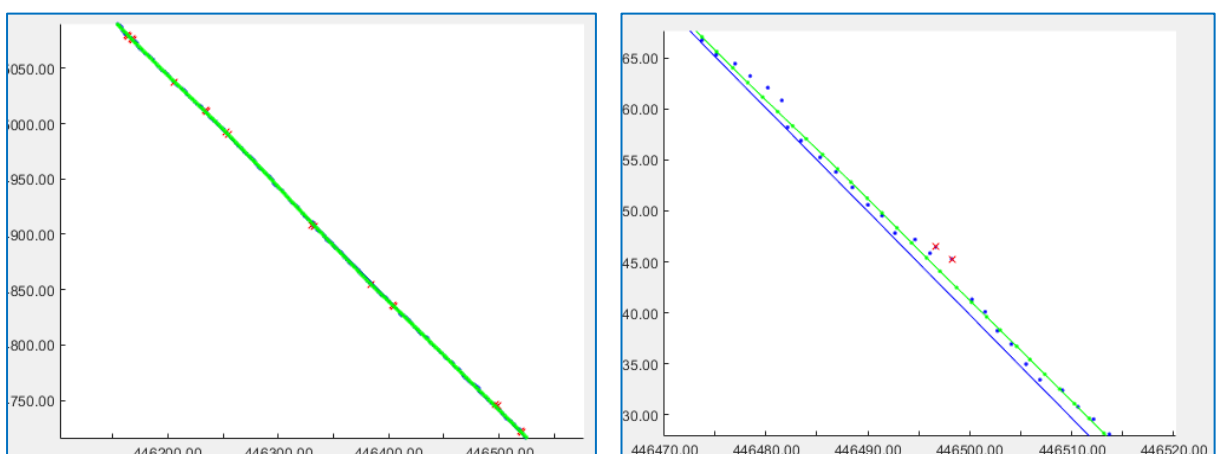


Figure 2.8 Window with the results of the SSS track processing (the figure on the left is scaled)



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

Script SssCoord01 (*Appendix 1. Script SssCoord01 (MatLab)*) is written for processing navigation files with SSS coordinates and heading. The files are generated by QINSy (with a time-step of about 1 second) and include 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).

Brief description of the script:

- 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, preliminary processing of SSS coordinates, preliminary processing of the SSS heading;
- 3) `NavCODA_Write` command – the operator writes the processing results to files in CODA format, for further navigation injection to SSS-files using NavInjector (SonarWiz).

#### 3.1 Reading of navigation files

Navigation files are generated by QINSy (in increments of approximately 1 second) and include a sequence of lines containing 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 for reading 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

During processing, calculations are performed 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 for processing a 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 processed files can be input as a sequence of numbers);

0 – the key, meaning 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 runs of processing, when selecting the smoothing filter window size;

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

After launching the "perform processing" command, windows are sequentially opened for processing and checking the data processing process 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, but is intended for visual control. If there are artifacts in the timestamps, it is necessary to start analyzing the reasons for the appearance of artifacts, since such data is substandard.

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

The window is closed by pressing any key.

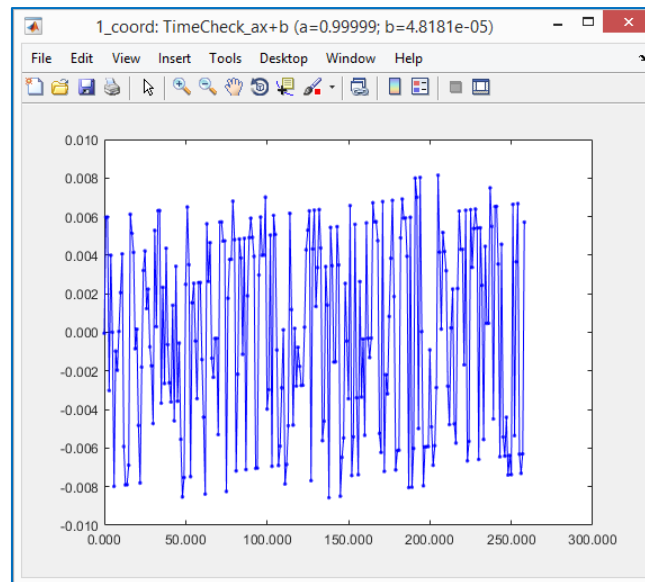
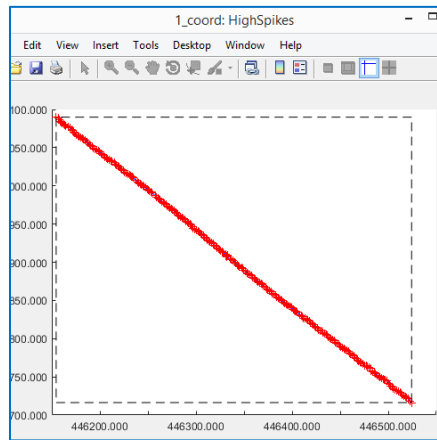


Figure 3.1 Timestamp control window (gFieldsDespike1D function)

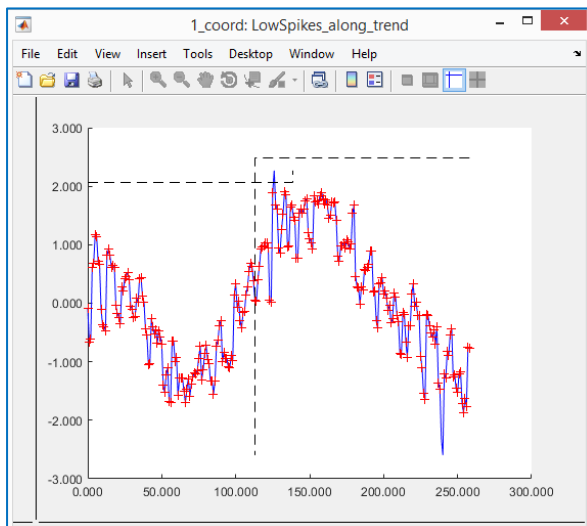
## EN-coordinates processing

The sequence of windows for processing 2D-coordinates is shown in [Figure 3.2](#).

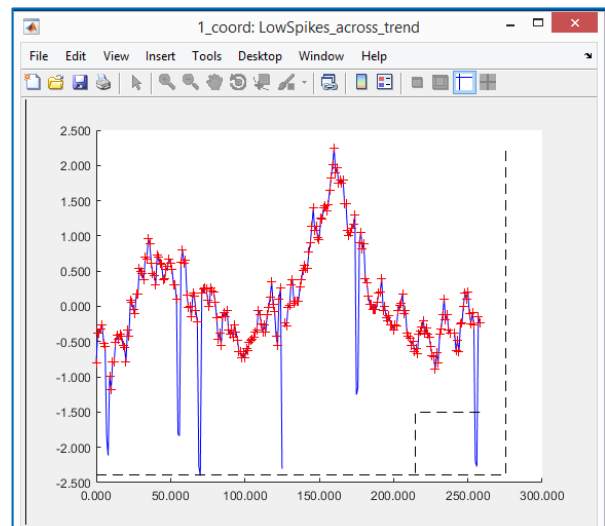
a)



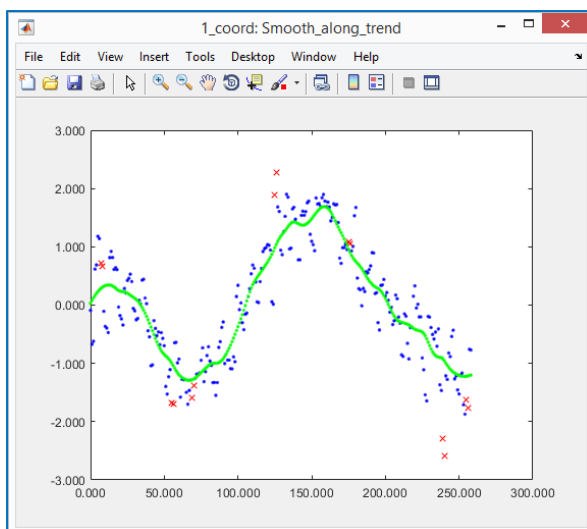
b)



c)



d)



e)

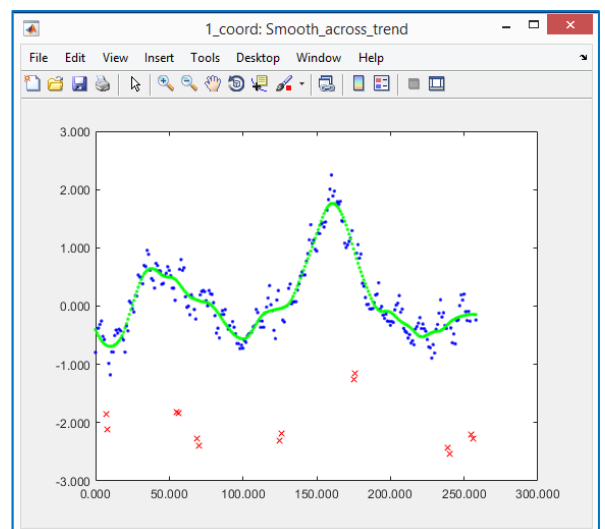


Figure 3.2 2D-Coordinate processing windows for SSS (gFieldsDespise2D function)

Figure 3.2, a shows the first interactive window for deleting high-spikes. 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 deleting low-spikes (the linear trend is removed, the coordinate system is rotated along-and-across the linear trend). Interpolation for deleted spikes is also calculated as linear. After deleting points in the "along the trend" window, they are not 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 you 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).

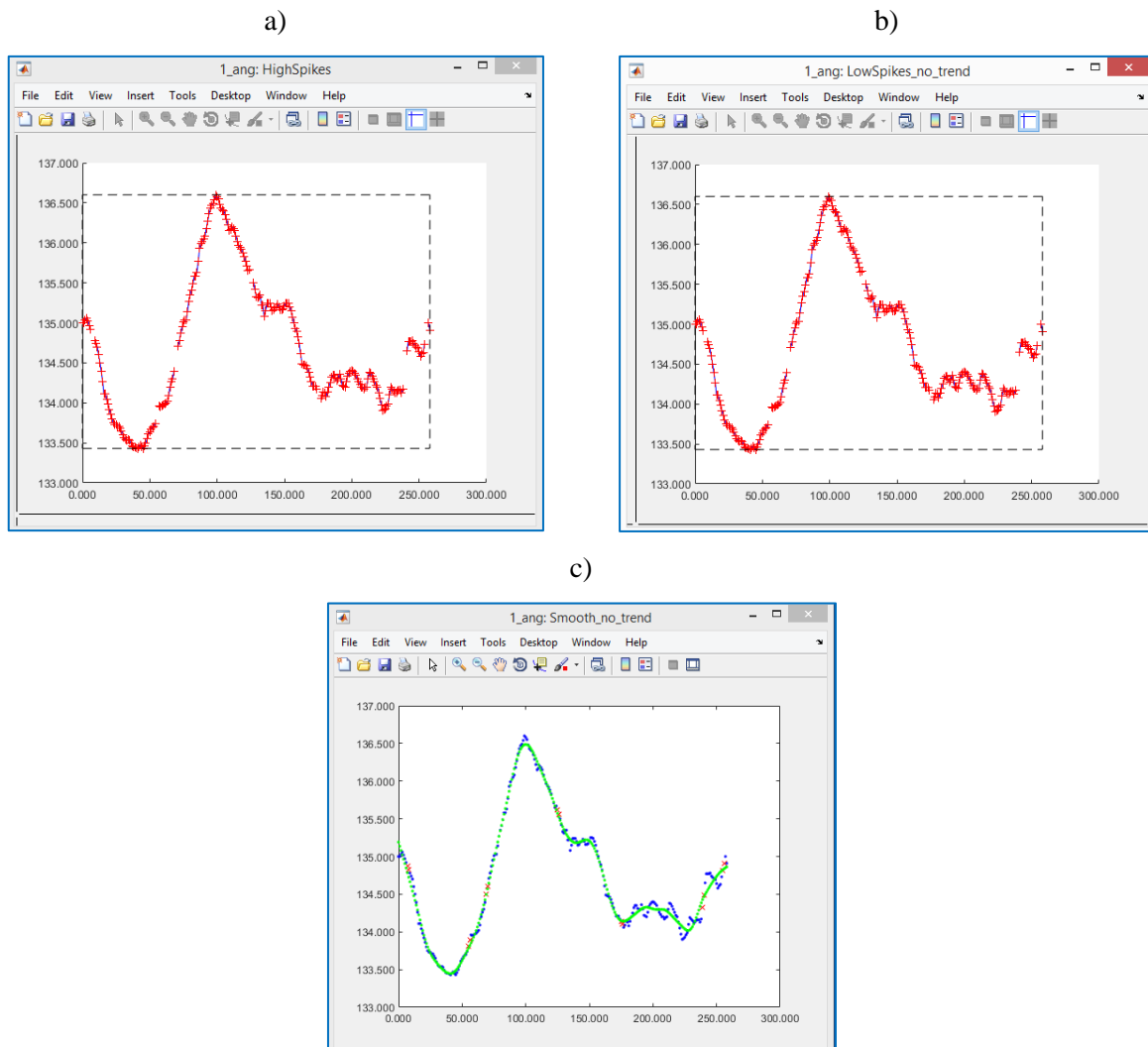


Figure 3.3 SSS heading processing windows (gFieldsDespike1D function)

The timestamp control window is not displayed because the timestamps are the same as the timestamps for the coordinates. Spikes deleted in coordinates previously are also deleted for the heading and are not displayed.

It would be more correct for the SSS heading not to edit it, but to recalculate it based on smoothed coordinates. To implement this, we would need data on the position of the tow point, as well as, possibly, editing and smoothing/filtering for the coordinates of the tow point (swaying of the tow point arm in reality is compensated by the hydrodynamics of the towing cable, and if these movements are not smoothed out, they will cause fluctuations in the heading of the towed fish). However, since there is no data on the coordinates of the towing point in the navigation Log-files, the processing is performed according to a simplified scheme – spikes are deleted, and the heading values are smoothed.

The value of the trend when processing the heading, in the script, is set constant and equal to zero. Therefore, the windows for deleting high and low spikes are the same (Figure 3.3, a-b). Heading values for spikes deleted 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 you to check the smoothing result. The green color shows the results of heading smoothing.

### Summary display of graphs for processed files

During processing, two additional windows (numbered 100 and 101; Figure 3.4) open, in which plots of the SSS position in the plan (EN-coordinates) and SSS heading plots for all processed profiles are displayed. When re-processing the same profile with different parameters (for example, smoothing), the windows will display the results for the same profile, which will allow to compare processing for it with different parameters.

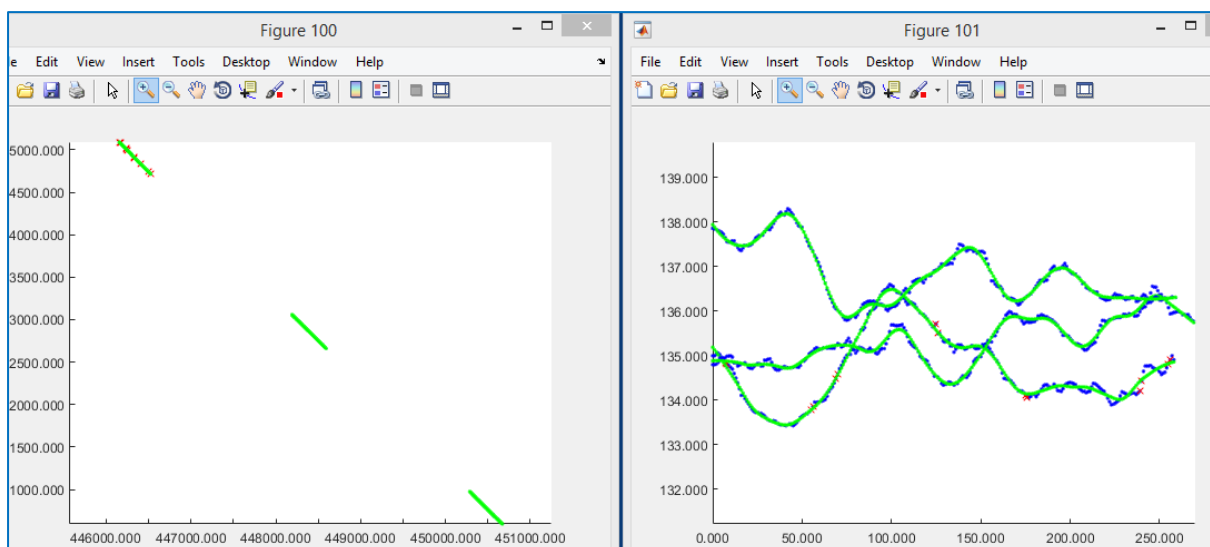


Figure 3.4 Windows with the results of processing SSS coordinates and heading for the current sequence of profiles

### 3.3 Saving of processing results

The script command for saving processing results is as follows:

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

where

'e:\tmp\out' – path to the folder where files with processing results will be written;

[] – a key meaning 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 some sensor with time reference. The description of functions `gMapPickHandleNan2`, `gFieldsDespike1D`, `gFieldsDespike2D`, which practically implement the processing graph, is given.

As an example, the processing of the coordinates and heading of the SSS, positioned using USBL, is considered. An example of a script (`SssCoord01`) is given, which allows the SSS processor to qualitatively and quickly perform the processing of coordinates and prepare files with coordinates for injecting by using NavInjector (SonarWiz).

## Evolution

For processing the USBL data, it is more appropriate to use not the [E,N] coordinates (which, moreover, are not complete without data on the vertical axis), but recalculations, with the introduced corrections for the roll of the vessel, from the initial 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 results in the relative "independence" of the components under consideration and the possibility of their separate processing using the `gFieldsDespike1D` function.

Among other things, the use of to-transponder distance 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, but the effect of the roll of the vessel will cause roll to "leak" into the USBL bearing angles of the transponder (as happened with the position in Figure 1.3). Most likely, this will lead to the effect of harmonic "yaw" of the SSS in the direction perpendicular to the head-transponder direction, and in a more general case, to movements along a helical trajectory. In this case, by correcting the setting angles and bringing the SSS track to a straight line, the error in the misalignment angles can be reduced and the positioning accuracy can be increased. However, such calculations are associated with the recalculation of the entire vessel navigation and require the use of an 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-
%f%f: %f: %f%f/%f/%f%f: %f: %f%s%s%f%s%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]=gFieldsDespike2D(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]=gFieldsDespike1D(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]);fclose(fid);
    end;
    disp('Processing exported. ');
    clearvars rootD1 NN nn z1 z2 fid
end;

%mail@ge0mllib.com 30/03/2023

```