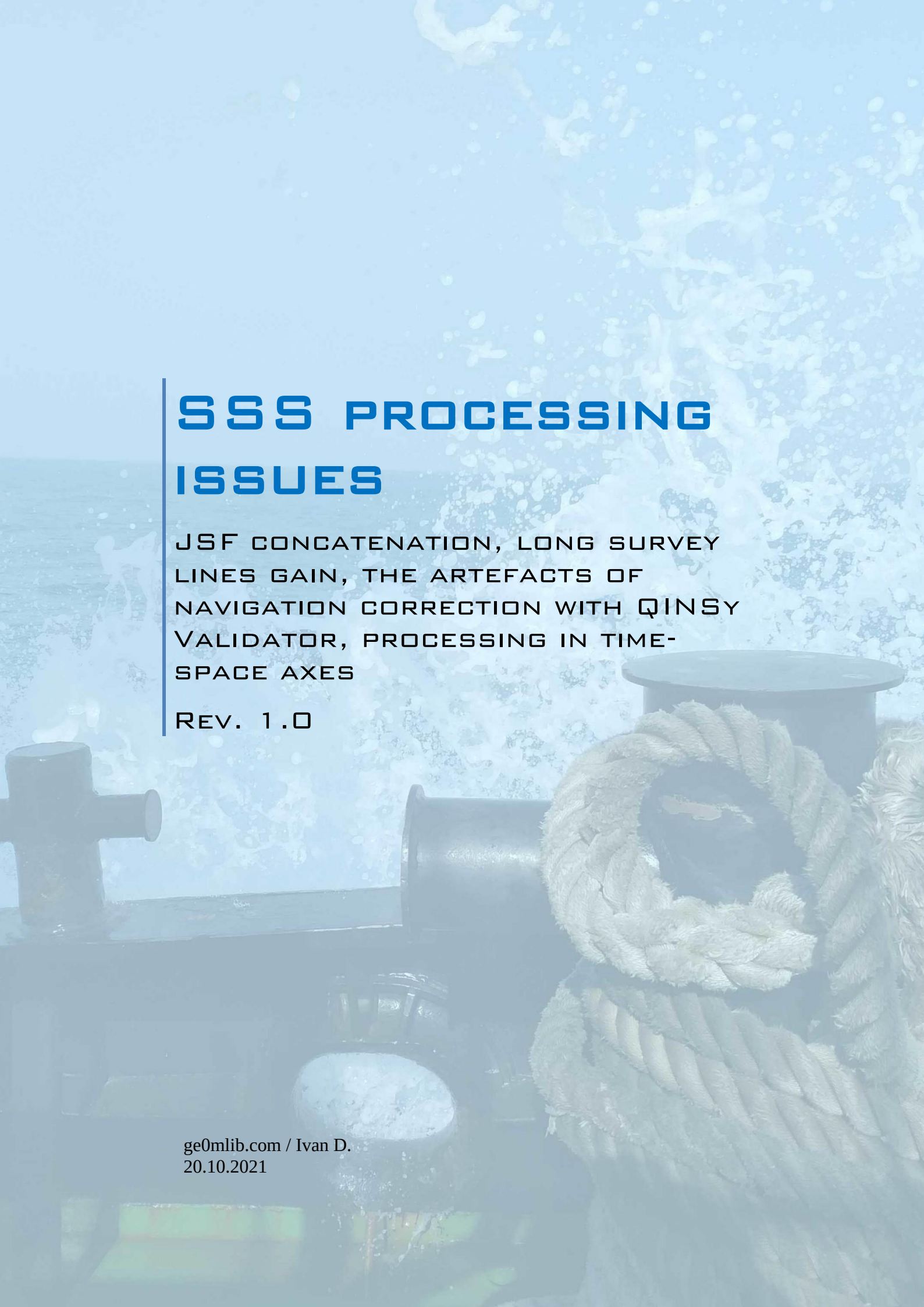


The background of the slide is a photograph of a ship's deck. In the foreground, there is a large, thick coil of light-colored rope. To the left, a searchlight is visible, and various metal fixtures and pipes are scattered across the deck. The background shows a blue sky and a body of water with white foam from a wake.

SSS PROCESSING ISSUES

JSF CONCATENATION, LONG SURVEY
LINES GAIN, THE ARTEFACTS OF
NAVIGATION CORRECTION WITH QINSY
VALIDATOR, PROCESSING IN TIME-
SPACE AXES

REV. 1.0

Contents

1	Overview	4
2	JSF-files gain method for a long lines	4
3	The features of SSS navigation data processed by QINSy Validator and NavInjectorPro....	7
3.1	Heading 0.00 at the last line of a file exported from QINSy Validator	7
3.2	“Navigation gap exceeds” value in NavInjectorPro settings	7
3.3	Stretching of the sonogram, localized in the location of a gap, occurring after QINSy Validator processing	9
3.4	“Stretches” on sonograms when strong filters are used (occurring after QINSy Validator processing)	10
3.5	Sonogram stretching and ZEdit (SonarWiz)	18
4	SSS navigation view in time-space axes	20
5	QINSy predictions view in time-space axes	22
6	QINSy extrapolation to 1-second step	26
7	MatLab script for coordinates processing in time-space axes	27
	Appendix 1. MatLab code	28

Figures list

Figure 2.1 Junction place for files “15.002” and “15.003”	4
Figure 2.2 Enable Auto TVG	4
Figure 2.3 Junction place for files “15.002” and “15.003” with EGN and AutoTVG applied	5
Figure 2.4 Junction place for files “15.002” and “15.003” which were concatenated to file 15_All; EGN and AutoTVG applied.....	5
Figure 2.5 File 15_All: (a) with EGN applied, (b) with EGN and AutoTVG applied.....	6
Figure 2.6 JSF-files concatenation (sections of line) using Total Commander	6
Figure 3.1 Heading values 0.00 at the last line of file created by QINSy Validator (processed navigation export)	7
Figure 3.2 NavInjectorPro settings	8
Figure 3.3 The NavInjectorPro processing result for the C_04 survey line; (a) navigation injected to parts (file was cut to “parts of file”, the 1500 m of some parts is not drawn), (b) navigation injected to concatenated file, (c) zoomed sonogram with “raw SSS position” drew to see gaps location (blue line in nadir zone)	8
Figure 3.4 Stretching in the gap area	9
Figure 3.5 Navigation with gaps for line C_04; blue is the raw data, red is processed by Validator	9
Figure 3.6 Navigation processing for the gap: (a) “ <i>specific artifacts</i> ” are not removed; (b) the points with “ <i>specific artifacts</i> ” are carefully deleted from navigation file.....	10
Figure 3.7 Stretching for small gaps (from 0.5 to 1.5 meters)	11
Figure 3.8 Navigation data processed by Validator (line A1_212) with a strong filter; points density (axes E and N).....	12
Figure 3.9 Navigation data processed by Validator (line A1_212) with a strong filter (x-axis time and y-axis N)	12
Figure 3.10 “Smoothed gaps filling effect”; example 1	13
Figure 3.11 “Smoothed gaps filling effect”; example 2.....	14
Figure 3.12 “Smoothed gaps filling effect”; example 3.....	15
Figure 3.13 “Smoothed gaps filling effect”; example 4.....	16
Figure 3.14 Strong “smoothed gaps filling effect”	17
Figure 3.15 Smoothed spikes for navigation (time-distance axes)	17
Figure 3.16 Sonogram’s stretching zone for a strong filter.....	19
Figure 3.17 SonarWiz’s ZEdit (navigation edit) window for stretching zone (blue ellipses).....	19
Figure 4.1 Survey line coordinates in Time-and-Distance axes (“across the modeled profile” component)	21
Figure 5.1 LT-view with QINSy predictions for line C_04; red crosses is raw data points	22
Figure 5.2 Linear “predicted data” shapes for line C_04.....	23
Figure 5.3 LT-view for line C_04; “linear extrapolation” by QINSy was removed.....	23
Figure 5.4 LT-view for line C_04; linear “predicted data” by QINSy was removed (cyan circles), the “coordinate’s oscillations” were smoothed (blue points).....	24
Figure 5.5 Coordinates in EN-axes for line C_04 with “oscillations of coordinate” were smoothed (blue points)	24
Figure 5.6 LT-view for line C_04; blue points are processed coordinated and interpolated coordinates	25
Figure 5.7 LT-view for line C_04 processed by Validator (red points) and processed with QINSy “predictions” removed (blue points); the tips step for vertical axis is 5 meters.....	25
Figure 6.1 Points extrapolated by QINSy (blue) between coordinates from USBL (red)	26

1 Overview

The document discusses three questions of an SSS processing. The first chapter describes Jsf-files concatenation and sonogram processing to create “uniform gain” for a long survey line. The second chapter describes specific features of navigation data processing with QINSy Validator and SonarWiz’s NavInjectorPro. The third chapter describes navigation data processing based on time-space axes (the MatLab’s script is used).

Warning!!! The text below refers to the specific processing sequence in the Validator. Perhaps it is possible to carry out processing without the appearance of the described artifacts (stretch zones, smoothed gaps filling, strong smoothed gaps filling effect).

2 JSF-files gain method for a long lines

When Discoverer software is recors SSS-file with a big volume, it can cut this file to several parts (usually about 800 Mb for each file, with extension 001, 002, etc.). When processing such “parts of file” difficulties may arise. For example, if we apply a different shift and angle of a pings rotation to them, then a “hole” can form between them. Furthermore, if we apply EGN for “parts of file”, the result can be differ for each part and place of junction will not look good (*Figure 2.1*).

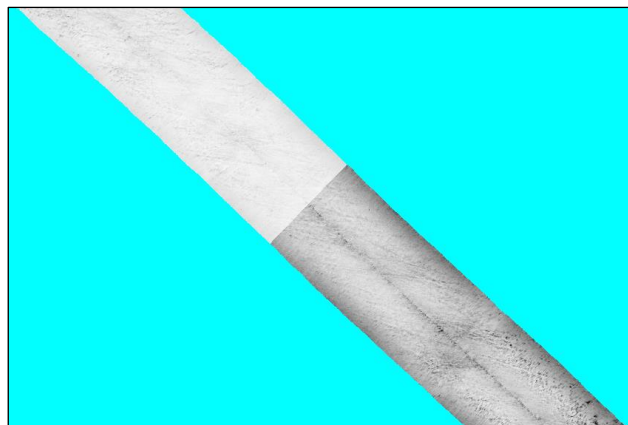


Figure 2.1 Junction place for files “15.002” and “15.003”

We can try to apply AutoTVG (*Figure 2.2*), to remove a long-period “trend of color”, but the result does not looks fine too (*Figure 2.3*).

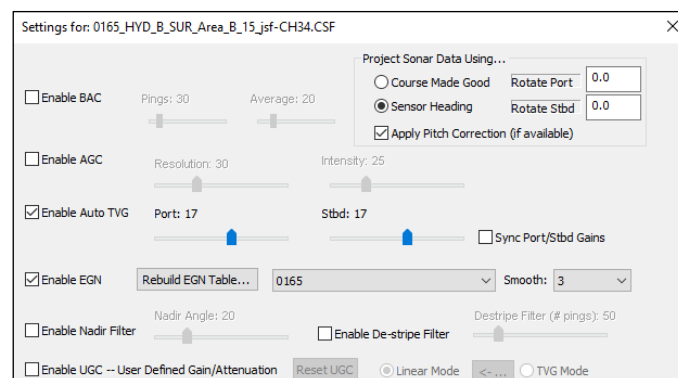


Figure 2.2 Enable Auto TVG

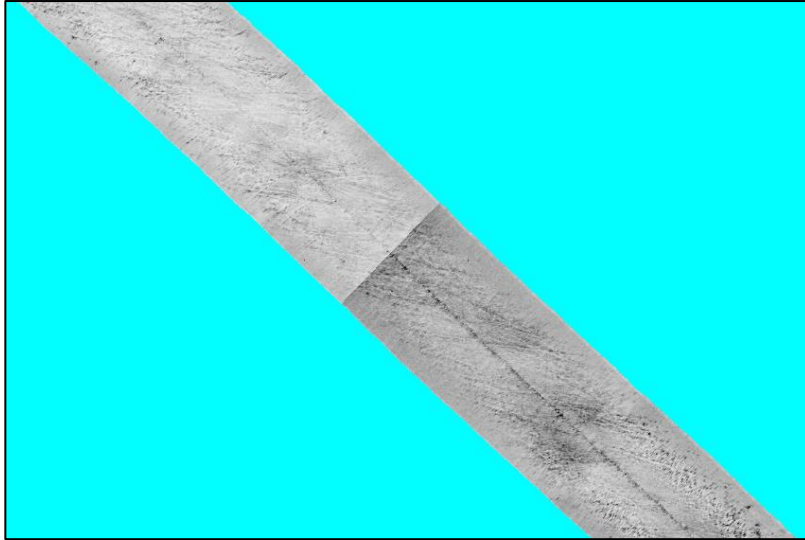


Figure 2.3 Junction place for files “15.002” and “15.003” with EGN and AutoTVG applied

The best result takes place if we concatenate parts of the line and apply EGN and AutoTVG to the whole line (**Figure 2.4**).

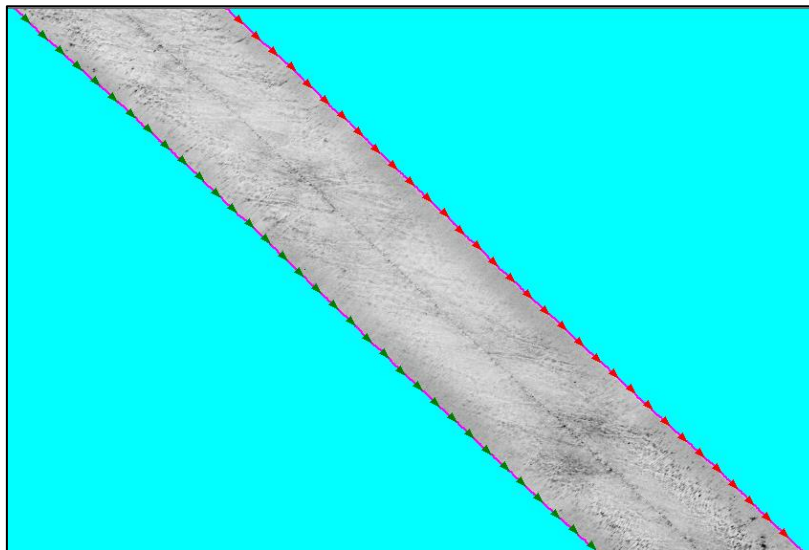


Figure 2.4 Junction place for files “15.002” and “15.003” which were concatenated to file 15_All; EGN and AutoTVG applied

The concatenated line with EGN applied is shown in **Figure 2.5, a** (the line length is about 50 km). The central part of the line has a light color. This color is the result of real environmental acoustic properties, it can be useful for “sediments type detection”, but it is not a good palette for “local features and objects detection” task. For the “local features and objects detection” task we can remove a long-period “trend of color” using Auto TVG. The result with Auto TVG applied is shown in **Figure 2.5, b**. That is, to solve different tasks, it can be necessary to use different types of amplification.

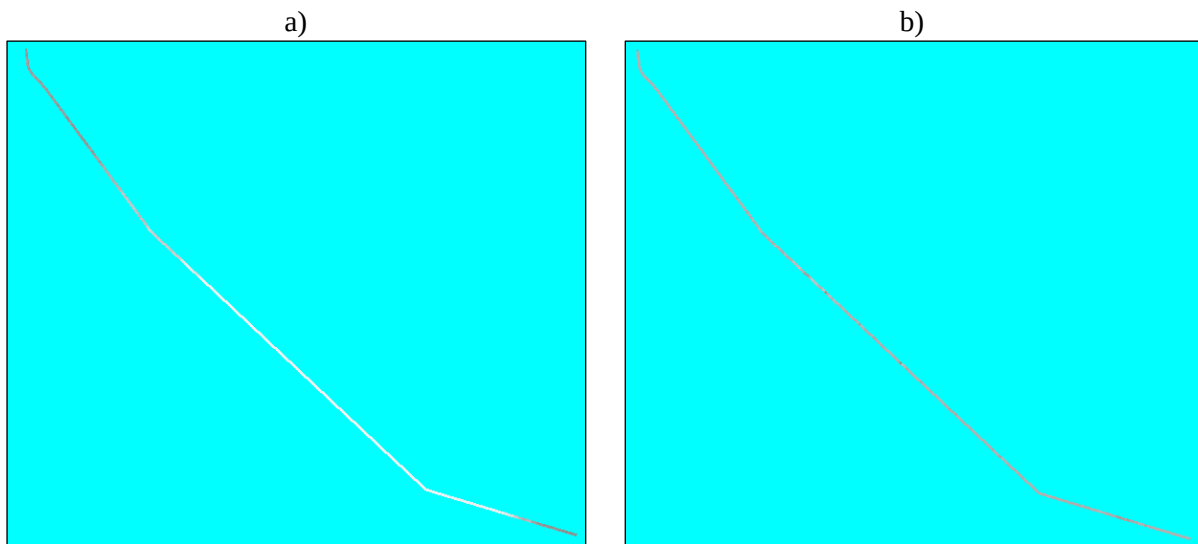


Figure 2.5 File 15_All: (a) with EGN applied, (b) with EGN and AutoTVG applied

JSF-files (line's sections) concatenation method

The JSF-format structure lets concatenate files if written second file's data since first. Any tool can be used for this; the Total Commander using example is shown in **Figure 2.6**.

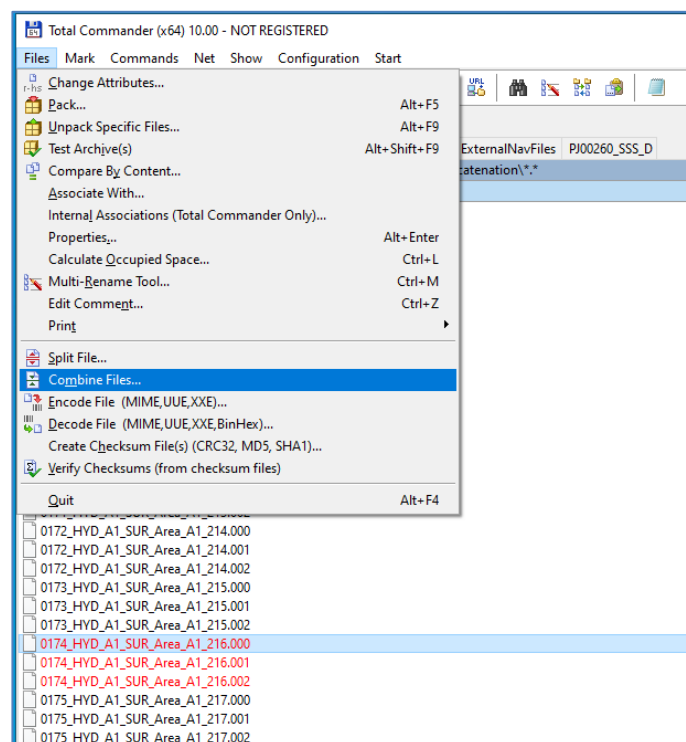


Figure 2.6 JSF-files concatenation (sections of line) using Total Commander

There are following steps:

- Change extension to 000 for a first file,
- Select first file and use “Combine Files” command from menu (**Figure 2.6**),
- Change extension of the concatenated file from 000 to JSF.

3 The features of SSS navigation data processed by QINSy Validator and NavInjectorPro

QINSy Validator software is often used to process SSS navigation on board. The procedure usually looks like this:

- 1) navigation data is loaded from the QINSy database into the QINSy Validator software,
- 2) spikes are removed manually and data is averaged/filtered using the Validator tools,
- 3) the processed navigation data is saved in a text file,
- 4) the navigation data from the text file is encapsulated into a file with SSS data (JSF, XTF or same) using the SonarWiz NavInjectorPro.

There are “pitfalls” in processing the navigation in QINSy Validator, which are described below.

3.1 Heading 0.00 at the last line of a file exported from QINSy Validator

The last line of navigation files (were processed in QINSy Validator) includes value 0.00 for Heading (Figure 8) every now and then. Probably it is some calculation method features. In any case, all files need to be controlled and value 0.00 must be changed to value from the previous line to the exclude false value. Otherwise, the false value will distort the mosaic built from the sonogram by the end of the survey line.

0163 - HYD_C_SUR_Area_C_04 - 0002 - Raw.txt - Notepad

File Edit Format View Help

1632855248.27	585	3.97	74.95	183.06	19.569	-9.51	* * * *
1632855248.49	585	1.95	74.54	183.08	19.569	-9.47	* * * *
1632855248.71	585	3.92	4.09	183.10	19.568	-9.43	* * * *
1632855248.92	585	3.89	3.65	183.12	19.568	-9.39	* * * *
1632855249.15	585	3.85	3.18	183.14	19.567	-9.34	* * * *
1632855249.36	585	3.82	3.66	183.18	19.567	-9.29	* * * *
1632855249.47	585	3.83	2.35	183.23	19.566	-9.29	* * * *
1632855249.69	585	3.78	1.89	183.22	19.566	-9.22	* * * *
1632855249.91	585	3.72	1.43	183.22	19.565	-9.15	* * * *
1632855250.13	585	3.67	1.00	183.20	19.565	-9.09	* * * *
1632855250.34	585	3.62	0.58	183.17	19.565	-9.02	* * * *
1632855250.46	585	3.60	0.38	183.15	19.564	-9.00	* * * *
1632855250.68	585	3.56	0.98	183.12	19.564	-8.95	* * * *
1632855250.89	585	3.53	0.61	183.07	19.564	-8.91	* * * *
1632855251.11	585	3.51	0.24	183.02	19.563	-8.88	* * * *
1632855251.33	585	3.51	0.86	182.96	19.563	-8.86	* * * *
1632855251.55	585	3.51	0.49	182.90	19.563	-8.85	* * * *
1632855251.66	585	3.52	0.31	0.00	19.562	-8.85	* * * *

<

Ln 1, Col 1

Figure 3.1 Heading values 0.00 at the last line of file created by QINSy Validator (processed navigation export)

3.2 “Navigation gap exceeds” value in NavInjectorPro settings

The NavInjectorPro settings include “Navigation gap exceeds” value (**Figure 3.2**). If the navigation data contain gap exceed the specified value (for example, as a result of a failure of the USB), then following navigation for the file does not generate and sonogram is not drawn in mosaic (see **Figure 3.3**).

Select Navigation Fields to Use

☐ Update Ship Position Only (P)
 ☒ Update Fish Position Only (U)
 Client: FGSI

☒ Interpolate navigation data to sonar time. (Unchecked inserts navigation closest to sonar time and may insert duplicate positions)

☒ Do not generate a file if navigation gap exceeds 80 Seconds of Navigation Time

Log File Entry Threshold Time (sec)
 (Make a log file entry if interpolation exceeds this time) 600

☒ Only write output file when replacement data available. Leaving this unchecked exports every original record regardless of whether navigation was injected into it. With this option checked only records that have been injected are output to file.

Set the check box next to any additional data items to transfer from the navigation file to the sonar file.

- ☒ EventNumber Column(2)
- ☒ FishHeadingDeg Column(5)

OK Cancel Select All Fields

Figure 3.2 NavInjectorPro settings

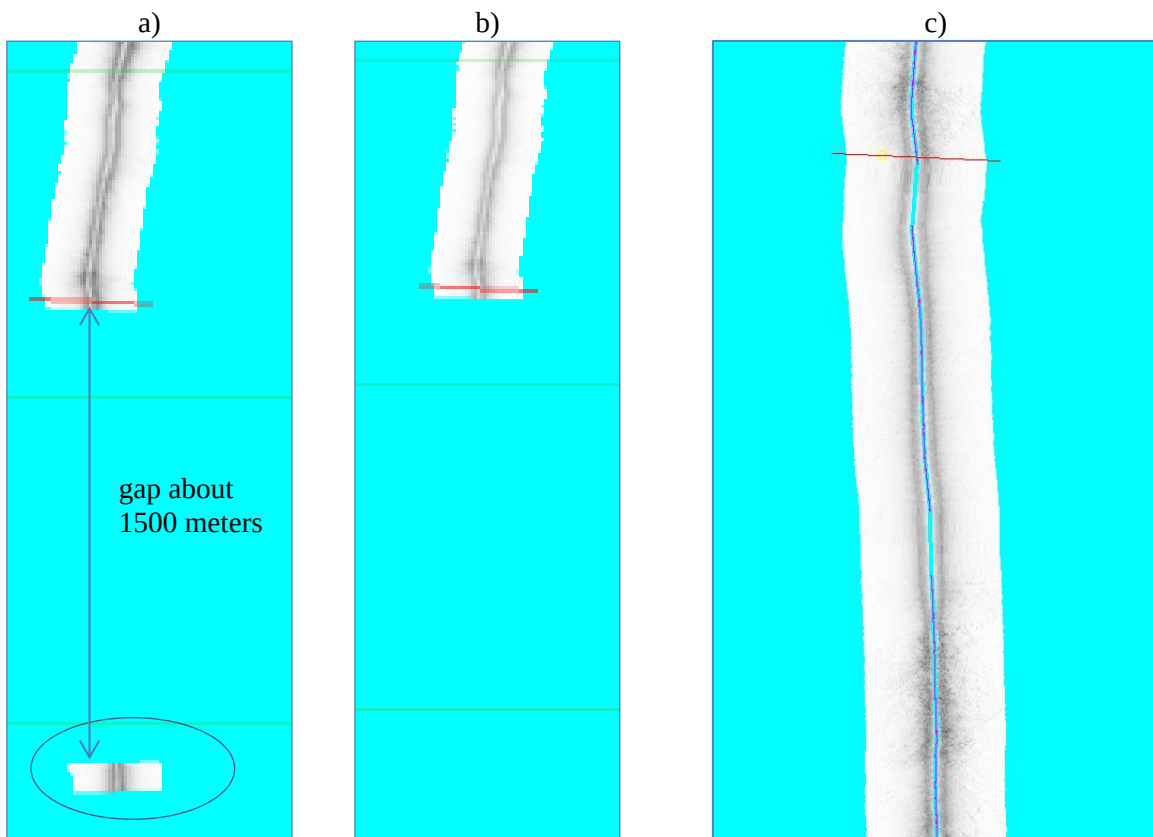


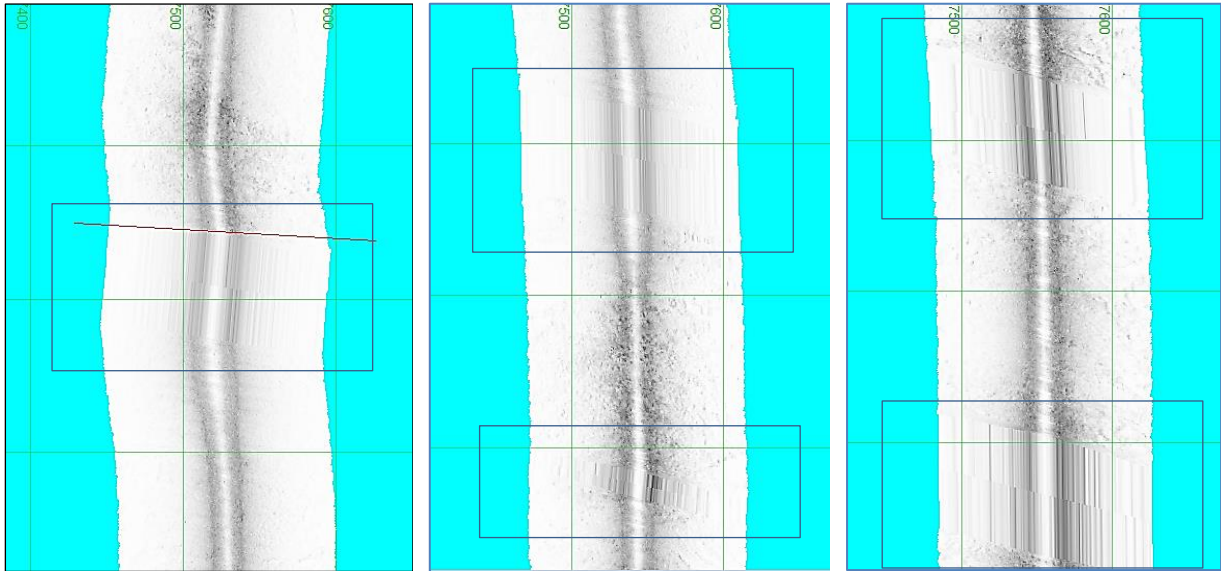
Figure 3.3 The NavInjectorPro processing result for the C_04 survey line; (a) navigation injected to parts (file was cut to “parts of file”, the 1500 m of some parts is not drawn), (b) navigation injected to concatenated file, (c) zoomed sonogram with “raw SSS position” drew to see gaps location (blue line in nadir zone)

3.3 Stretching of the sonogram, localized in the location of a gap, occurring after QINSy

Validator processing

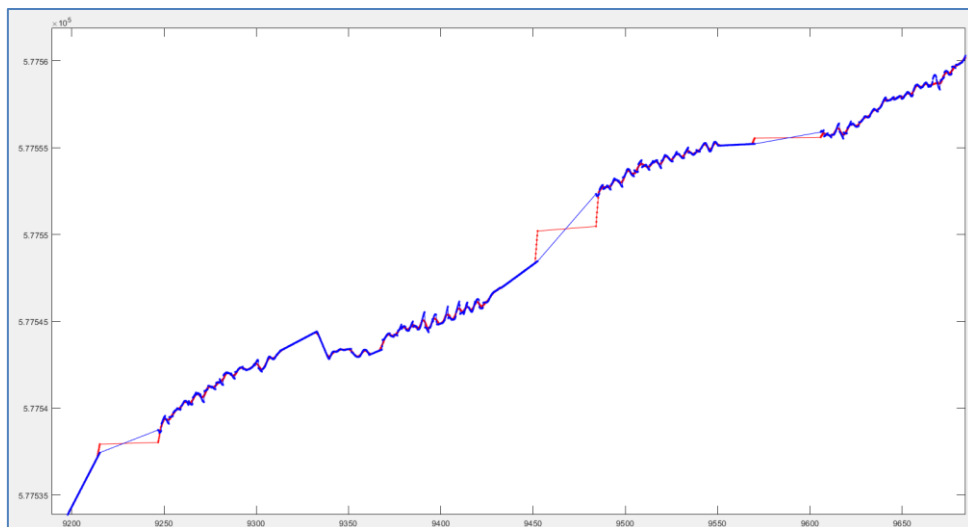
Warning!!! The text below refers to the specific processing sequence in the Validator. Perhaps it is possible to carry out processing without the appearance of the described artifacts (stretch zones, smoothed gaps filling, strong smoothed gaps filling effect).

In the areas where gaps are located (missed navigation data), after processing with NavInjectorPro using, a “stretch zones” appear similar to shown in the [Figure 3.4](#).



[Figure 3.4](#) Stretching in the gap area

To understand the stretch-zones nature, we need to see the navigation data at the plot in [Figure 3.5](#). The blue line is the raw navigation; the red line is the navigation processed by Validator. The blue line without points is the navigation gap without USBL data. The horizontal axis of [Figure 3.5](#) is the time in seconds, the vertical axis is Easting (the axes are chosen for the best gaps visualization). We see the **specific artifacts** drawn red line in the gap area – the Easting changes quickly for a small time, after that – quick time changes and small changes for Easting ([Figure 3.5](#)).



[Figure 3.5](#) Navigation with gaps for line C_04; blue is the raw data, red is processed by Validator

Described artifacts formed stretching zones (a lot of SSS pings in a stretch-zone flank area and a big pings density in the center of a stretch-zone). Probably, the described artifacts are the result of smoothing in Validator (the filter window effect in the border of a gap).

Figure 3.6 shows the different processing navigation:

- a) The navigation injected using NavInjectorPro, which includes the artifact points. The stretching zone is presented on mosaic;
- b) The artifact points were found and deleted from nav-file. The navigation is injected using NavInjectorPro, without the artifact points. The stretching zone is absent (NavInjectorPro performs linear interpolation for “missing” points).

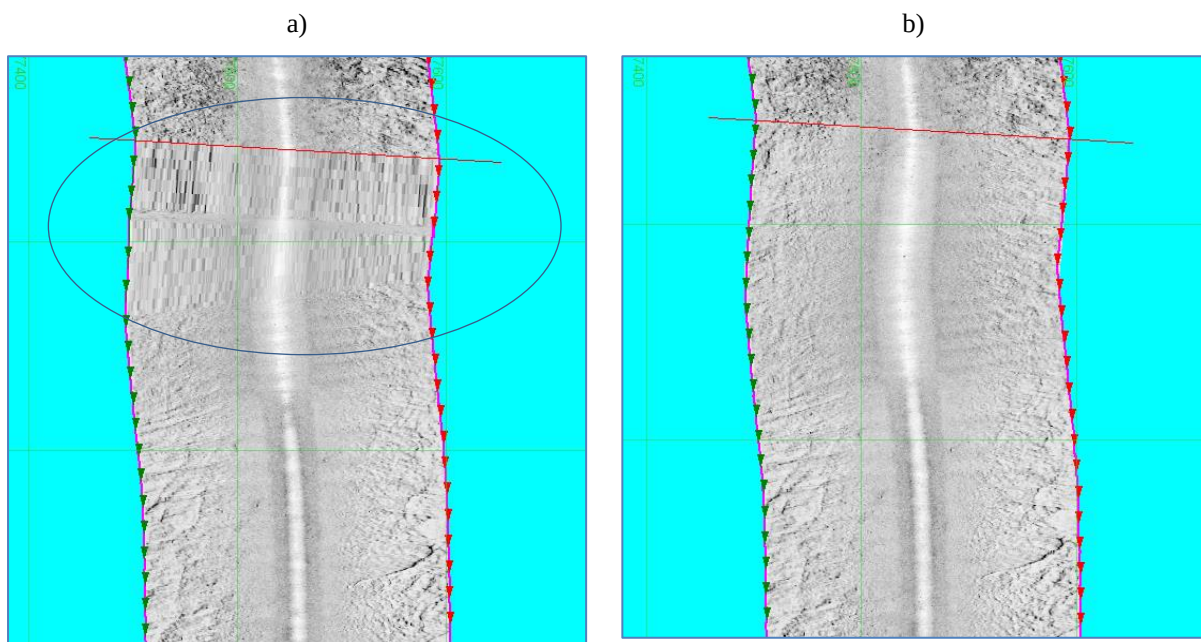


Figure 3.6 Navigation processing for the gap: (a) “**specific artifacts**” are not removed; (b) the points with “**specific artifacts**” are carefully deleted from navigation file

An additional feature in a navigation data is an oscillation with amplitude from 0.2 to 1.5 meters (see *Figure 3.5*, blue line oscillation). Probably, there is a pitch/roll result – the angles were not fully compensated by USBL. In any case, this feature is not a true SSS track. It will be better to smooth oscillation and create an even track. To realize this, we need to use relatively strong smoothed filters.

3.4 “Stretches” on sonograms when strong filters are used (occurring after QINSy Validator processing)

On the *Figure 3.7* is shown a Validator processing result with gaps mentioned above. There are shot gaps about 1 meter, but each of them form small “stretch zone” for the sonogram.

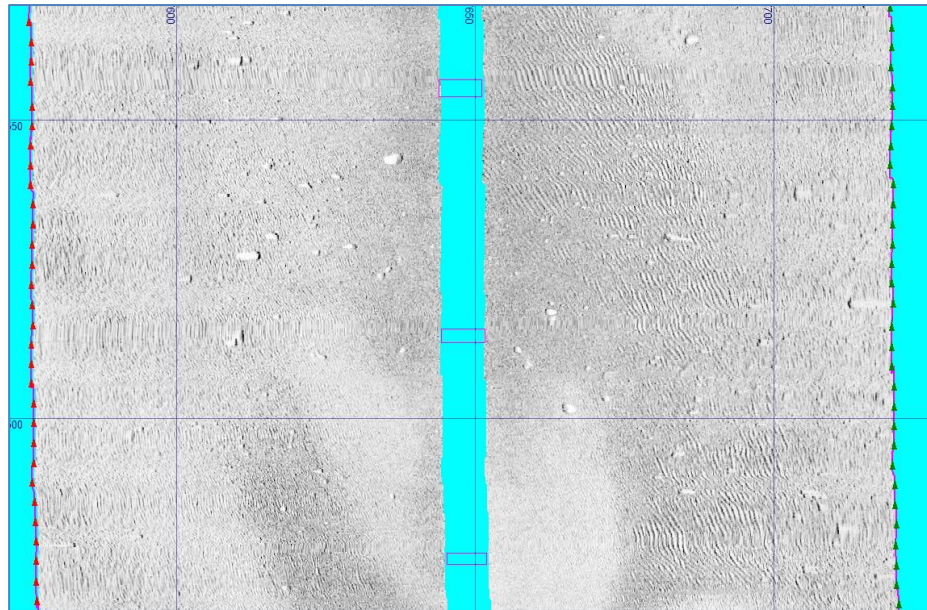


Figure 3.7 Stretching for small gaps (from 0.5 to 1.5 meters)

We can try to use a relatively strong filter in Validator software by following reasons:

- 1) to remove small “stretch zones” showed in **Figure 3.7**,
- 2) to smooth oscillations for navigation data showed in **Figure 3.5**, as blue line oscillations.

If we use a “strong filter”, then for the “big” gaps (means a gap size is the same to the filter window size) we will have “smoothed gaps filling”. It looks like “changes of pings density”, but it doesn't look so bad and sometimes is difficult to be detected visually on the mosaic.

“Smoothed gaps filling” is shown in **Figure 3.8** for N&E axes. “Smoothed gaps filling” for the y-axis along-line-distance (with linear component removed) and x-axis time is shown in **Figure 3.9**.

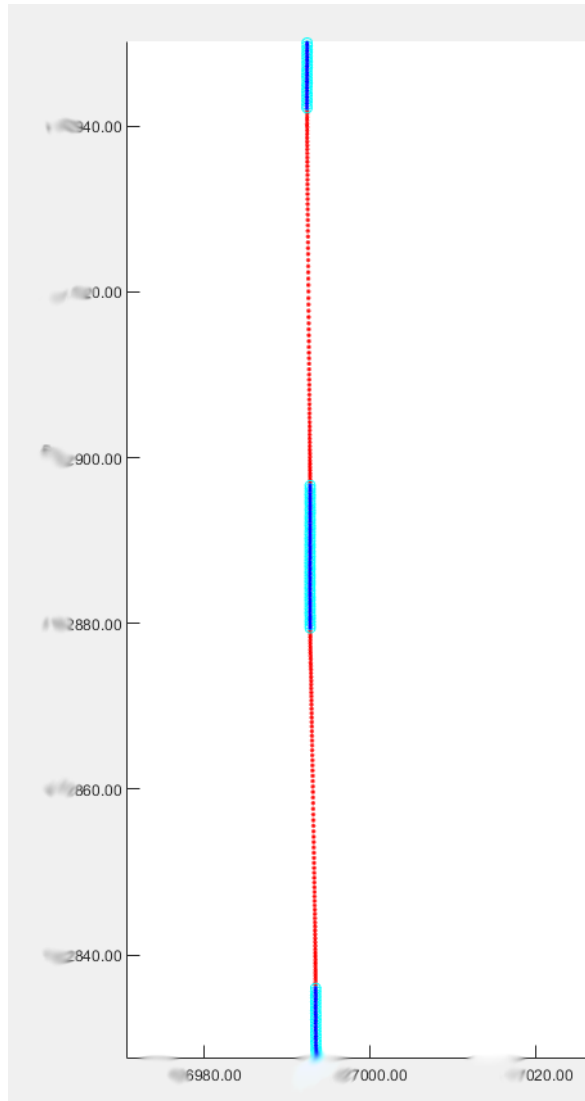


Figure 3.8 Navigation data processed by Validator (line A1_212) with a strong filter; points density (axes E and N)

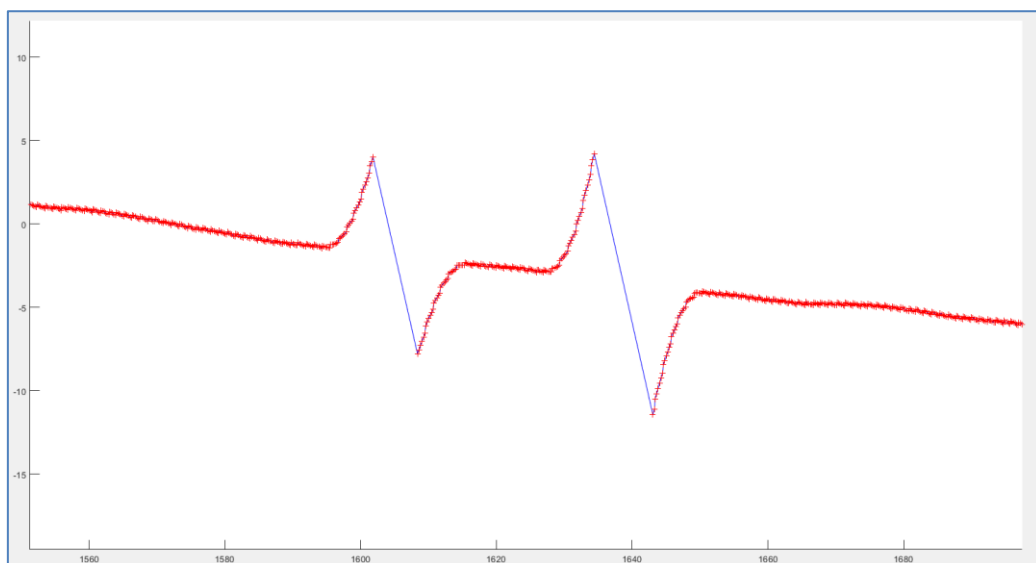


Figure 3.9 Navigation data processed by Validator (line A1_212) with a strong filter (x-axis time and y-axis N)

The “smoothed gaps filling” is shown in *Figure 3.10*, *Figure 3.11*, *Figure 3.12* and *Figure 3.13*. The first part of pictures shows a sonogram with linear interpolation for gaps; the second part shows a “smoothed gaps filling” effect; there is a stretching in the gap area after strong filters applied in Validator. The effect is not so rigid than in *Figure 3.4*, but absolute shifting is big too, it can be up to 10 meters (*Figure 3.10*).

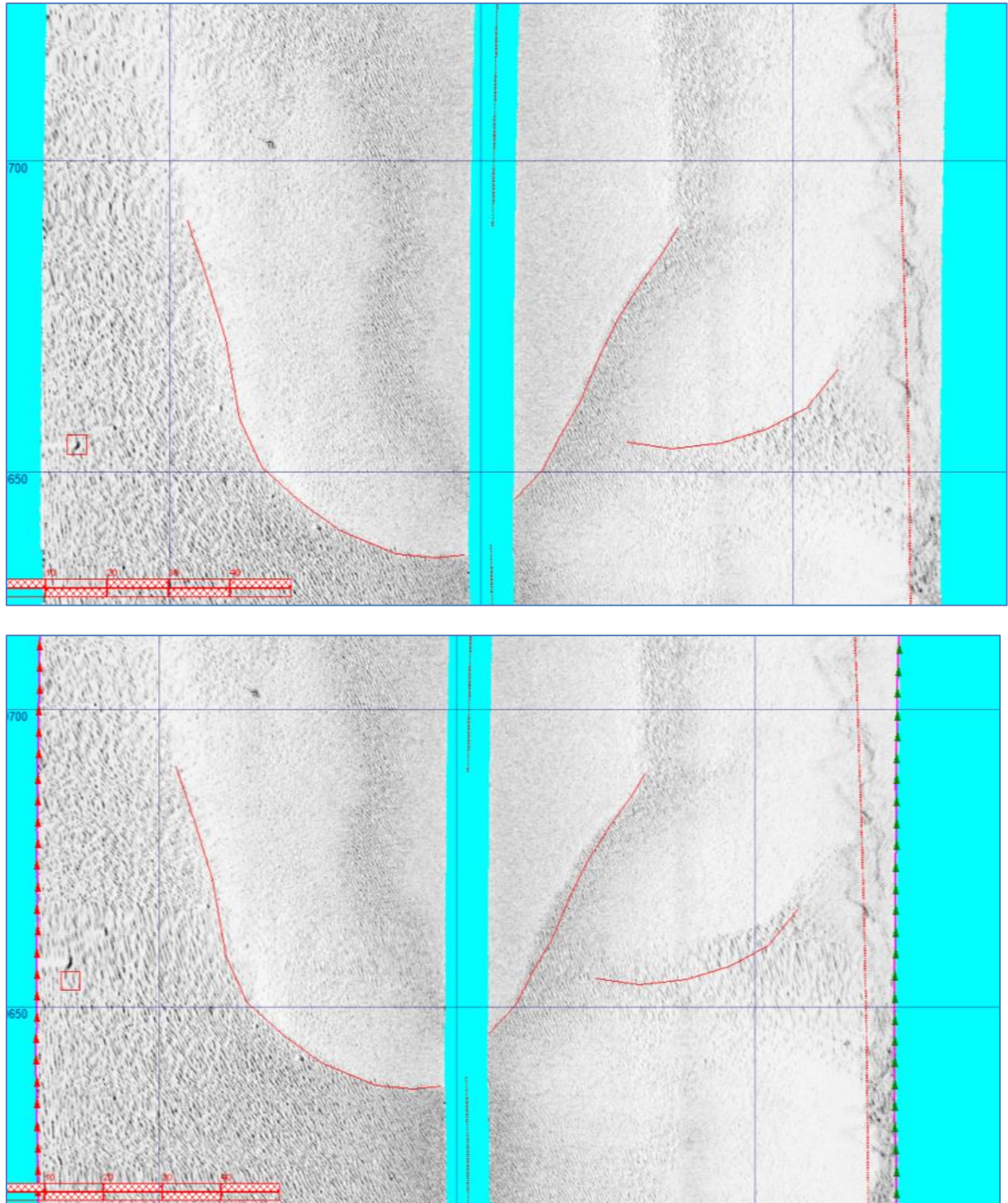


Figure 3.10 “Smoothed gaps filling effect”; example 1

In the central parts of the gap we can see the linear feature with high-density of pings (marked in *Figure 3.11*, blue ellipse). There are same features in *Figure 3.12* and *Figure 3.13*.

To simplify gaps identity, the DWG-files with pings locations were prepared and put in sonograms as an overlay layer (see red dots in the nadir zone for sonograms in *Figure 3.10*, *Figure 3.11*, *Figure 3.12* and *Figure 3.13*).

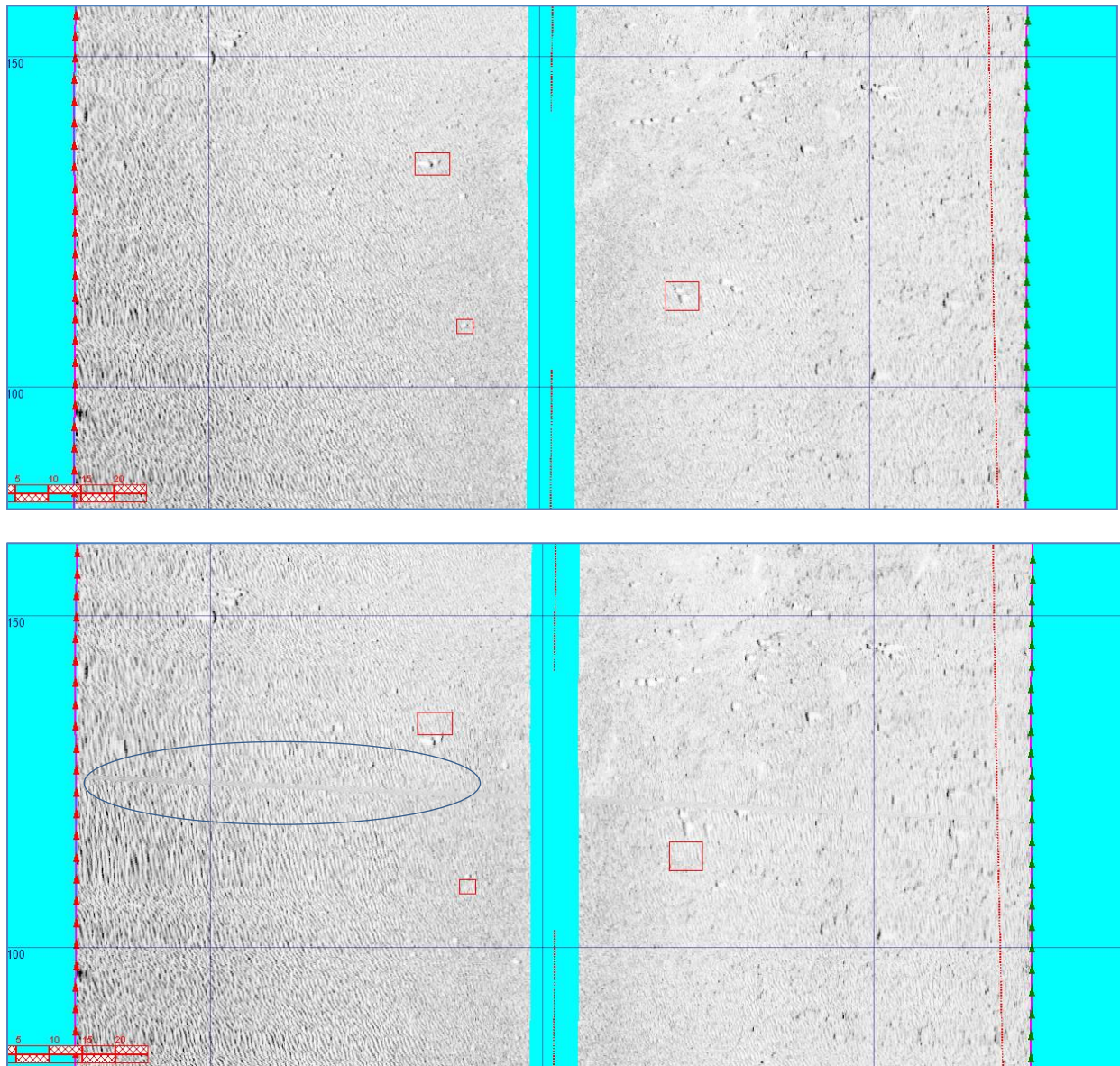


Figure 3.11 “Smoothed gaps filling effect”; example 2

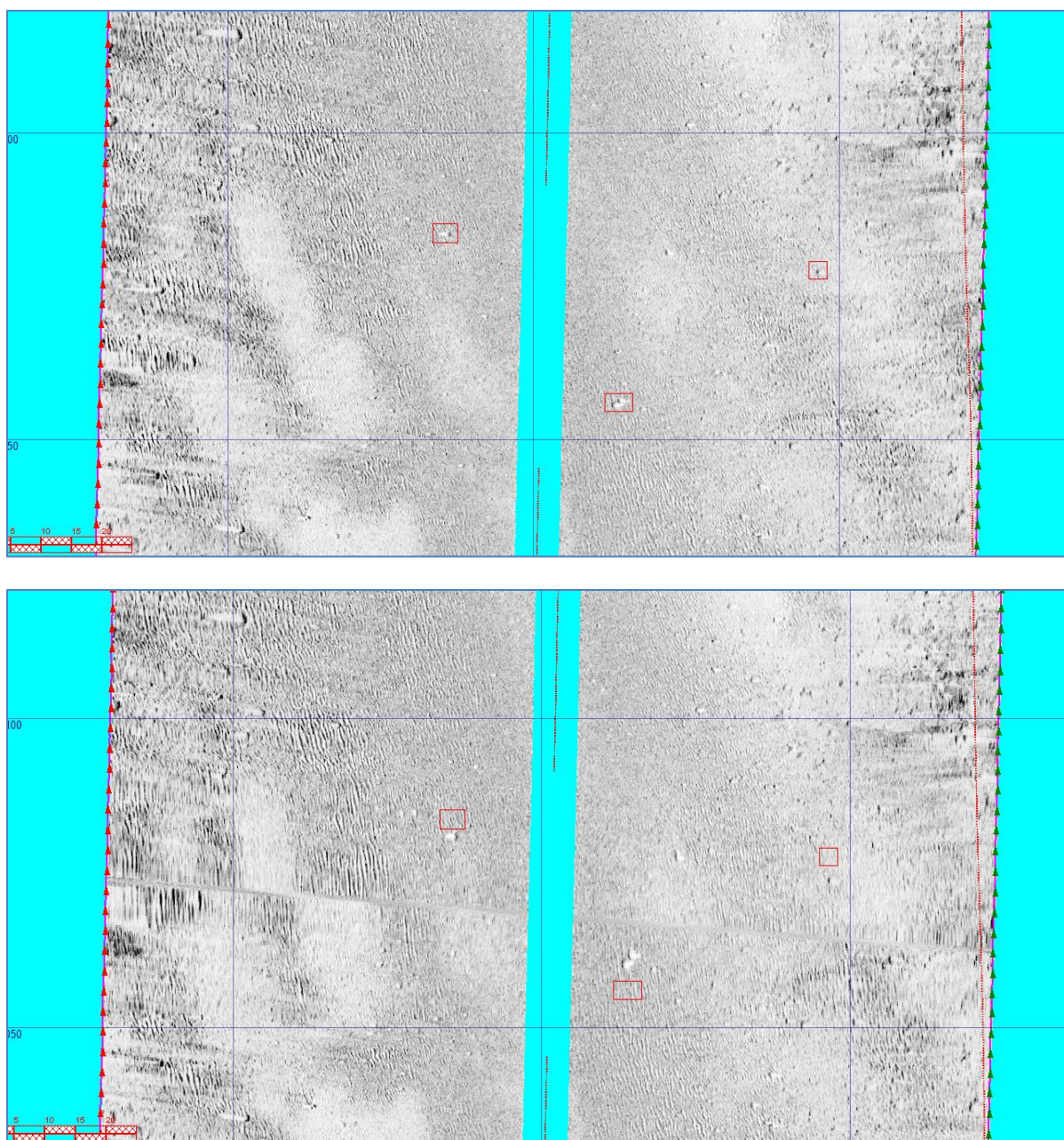


Figure 3.12 “Smoothed gaps filling effect”; example 3

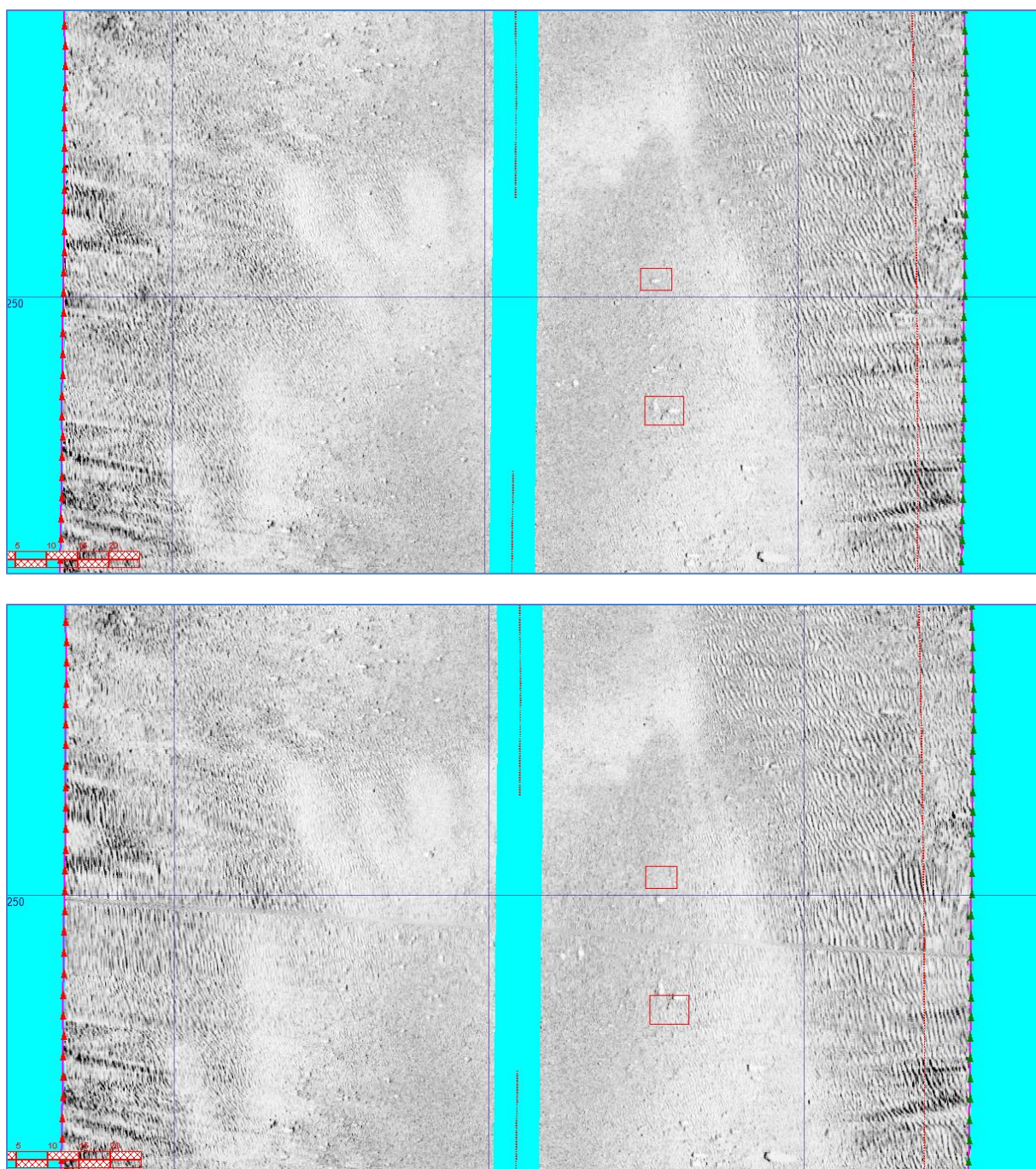


Figure 3.13 “Smoothed gaps filling effect”; example 4

Sometimes Validator creates output files with a strong “smoothed gaps filling effect” (*Figure 3.14, a*). There are a tens points with changes of coordinates about 30 meters. The strong effect view for sonogram is shown in *Figure 3.14, b*.

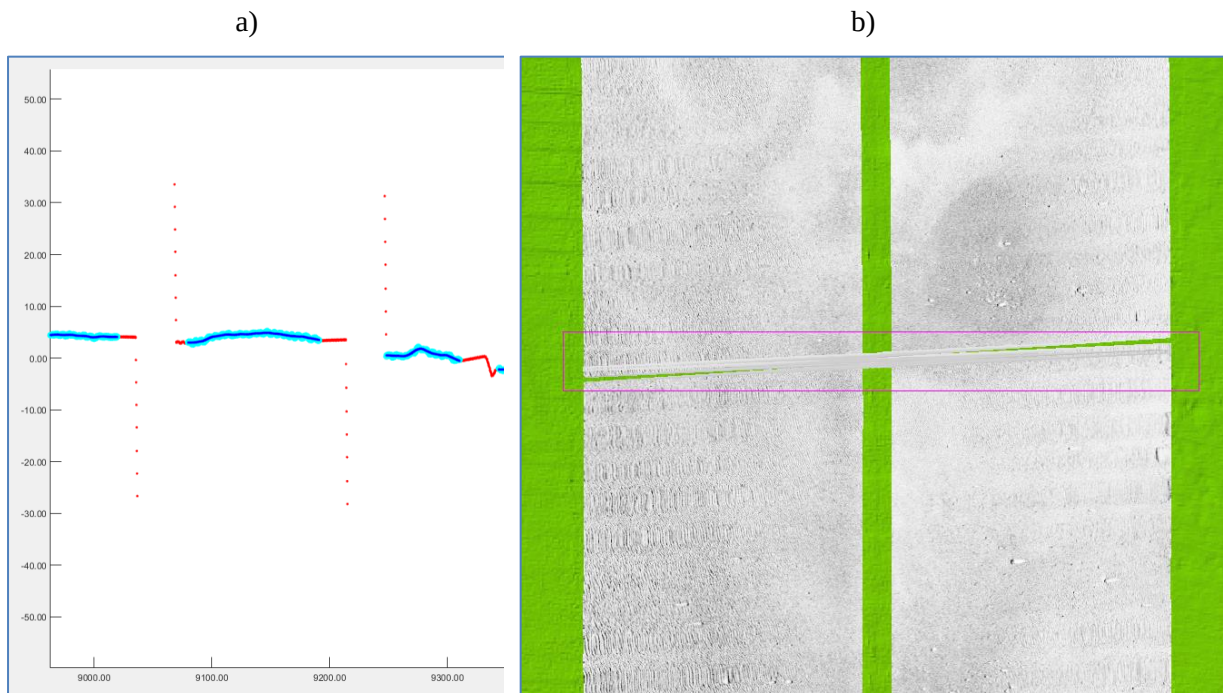


Figure 3.14 Strong “smoothed gaps filling effect”

Another “bad feature” for strong filter is smoothing for not removed spikes (“gross errors”), like shown in *Figure 3.15*. These errors became difficult-visible in sonograms like “smoothed gaps filling”.

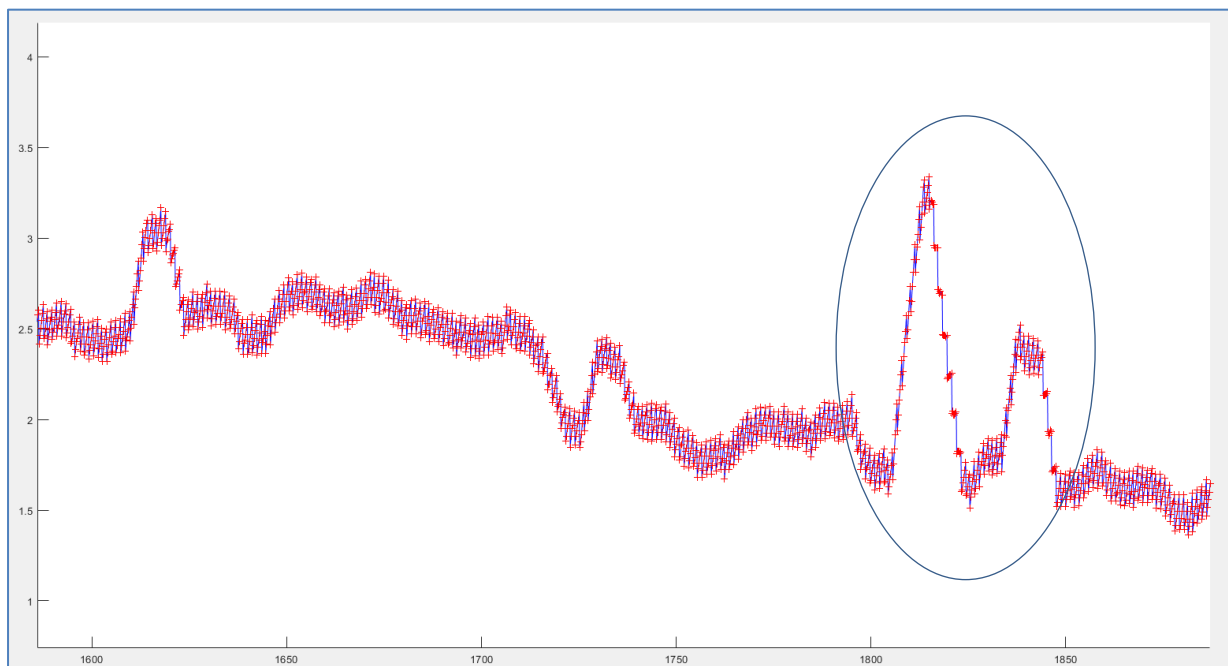


Figure 3.15 Smoothed spikes for navigation (time-distance axes)

3.5 Sonogram stretching and ZEdit (SonarWiz)

The stretching zone view ([Figure 3.16](#)) in ZEdit (SonarWiz's utility for navigation editing) is shown in [Figure 3.17](#). The stretch location mark using blue ellipse. The horizontal part in the ellipse center is a high-density-pings area in sonogram. The horizontal axis is a "ping number", the vertical axis is in Northing (the survey line direction is N to S, so for high-density-pings area the N values is factually not changed).

In the upper part of [Figure 3.17](#), Northing and Easting axes are presented; the pings density changes are recognized. It is difficult to detect and edit stretch zones by using ZEdit. The simplest way is detect ping number for a beginning and an end of the gap using sonogram, after that delete/smooth fit points for selected pings range using ZEdit.

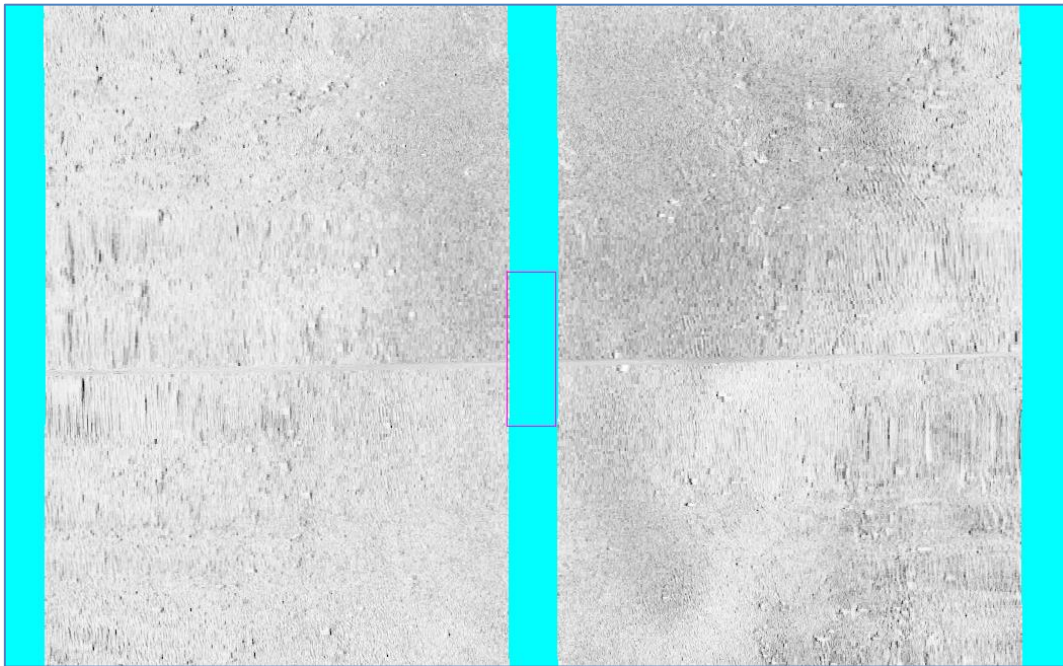


Figure 3.16 Sonogram's stretching zone for a strong filter

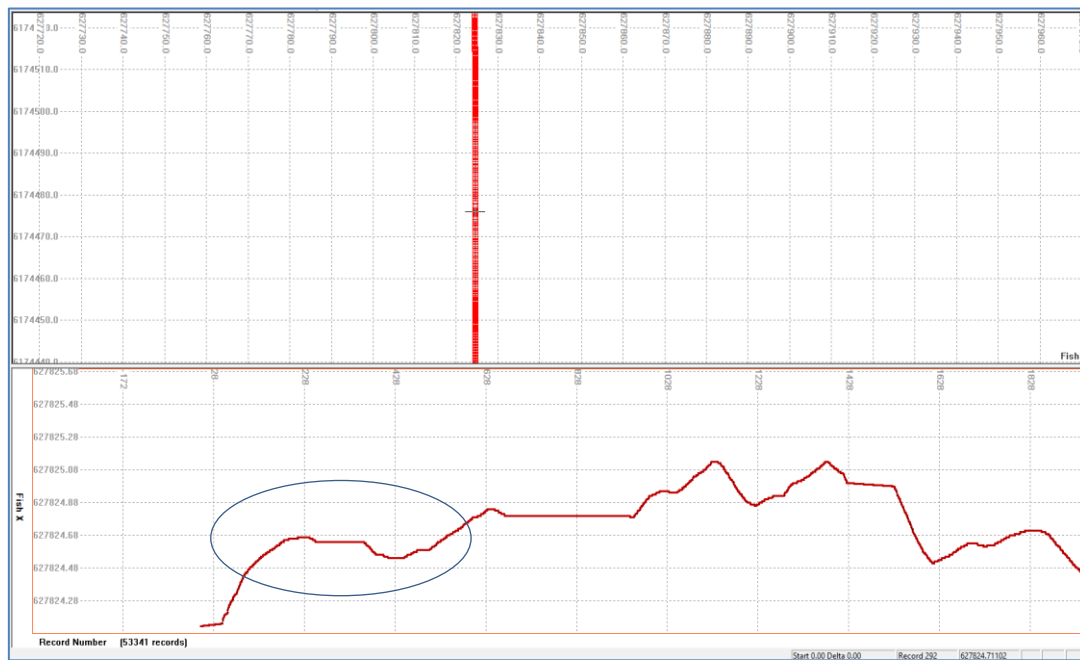


Figure 3.17 SonarWiz's ZEdit (navigation edit) window for stretching zone (blue ellipses)

4 SSS navigation view in time-space axes

As can be seen from [Figure 3.17](#), the display in the EN-axes does not highlight the position of the stretch zones, spikes and other “specific” features. More usable is the display in the coordinate-time axes (LT-axes), which allows you to correlate the distance increment with the elapsed time. Such representation is associated with the constancy of the speed of the SSS; the SSS has some inertia and cannot change its position instantly. The spike in coordinates will correspond to a relatively large increment of the “distance moved for the time step”, which will be clearly seen on the graph. The example of a view in LT-axes is shown in [Figure 4.1](#).

There was the following evolution in LT-view:

- 1) To display the distance increment, we tried to use the calculated increment along the E and N axes. However, it turned out to be more meaningful to calculate the increment “along a survey line” and “across the survey line” – the first is related to the speed of the SSS towing, and the second is related to the movements of the SSS across the line of motion (sway-like);
- 2) Initially, the calculations were performed for the survey line (profile) as a line (means shape). The direction of the line was calculated, then the increments along and across the line were calculated. Later there was a need for calculations for “non-line” profiles.
- 3) First, a “planned a curved profile” was set/define, the SSS position points (measured coordinates) were projected on the profile curve, and for each point the direction “along the profile” and “across the profile” was calculated.
- 4) When “planning a curved profile” it turned out to be expedient to use not “as planned” but as modeled “as surveyed” profile. The calculation of the “modeled as surveyed profile” made it possible to change the method of calculation -- from calculating the increments for the measured coordinates to calculating the difference between the “modeled relatively uniformly distributed coordinates” and the measured coordinates.
- 5) As a result, LT-view is calculated according to the following rules: on the basis of navigation data, the “modeled profile” is built with a relatively uniform distribution of modeled survey points along the profile; for each modeled point, the direction “along the modeled profile” and “across the modeled profile” is calculated; the difference between the coordinates of the measured and modeled points is calculated ([Figure 4.1, b](#)); the “along the modeled profile” difference is related to the speed of the SSS towing, the “across the modeled profile” difference is related to the movements of the SSS across the line of motion (sway-like).

The survey line coordinates view in LT-axes is shown in [Figure 4.1, a](#); there is “across the modeled profile” component only. The “modelled points coordinates” are a strongly smoothed measured coordinates; in consequence, the long-period trend is removed for “across” and “along” coordinates. **Spikes in coordinates stand out well for LT-view.** The plots with spikes are removed (light blue) and coordinates are smoothed (blue) and are shown in [Figure 4.1, b](#) (if we will use a



smooth filter without spikes-removing, then the spikes will apply additional error in the smoothed coordinate's values). The *Figure 4.1, c* shows the plot in Time-and-Easting axes “recovered” from *Figure 4.1, b* (a linear trend is removed for the best visualization).

The examples for stretching zones LT-view is shown in *Figure 3.5*, *Figure 3.9* and *Figure 3.14*. The stretching zones in SSS coordinates stand out better for LT-view, then for EN-axes.

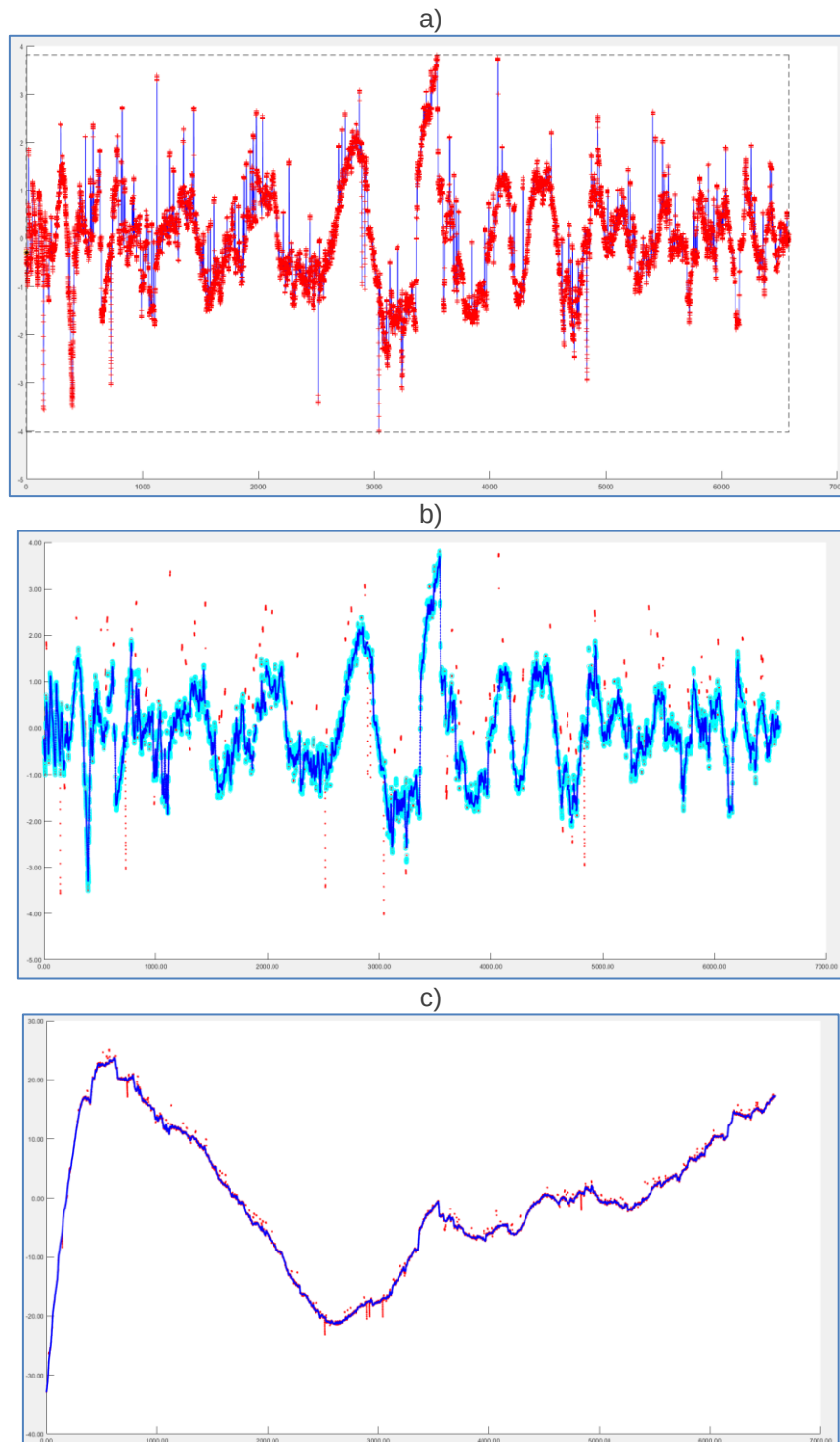


Figure 4.1 Survey line coordinates in Time-and-Distance axes (“across the modeled profile” component)

Three methods for calculating “model coordinates” have been tried:

- 1) Approximation of the measured coordinates (in the time-Y or time-T axes) using a polynomial of the N-degree; the calculation of model coordinates for given times “along” the polynomial curve;
- 2) Strong smoothing of the measured coordinates, as a result of which we calculate the “long-term trend” curve with a quasi-uniform measurement step;
- 3) Use of the layback calculation model (based on the position of the towing point and the heading of the vessel).

All three models make it possible to calculate a “long-term trend” relatively to the measured SSS coordinates and give approximately the same result. Models 1 and 2 require a preliminary removal of strong spikes, since the “measured SSS coordinates” are used for the calculation. Model 3 requires an additional data (position of the towing point, vessel’s heading). Last model is preferable.

5 QINSy predictions view in time-space axes

The navigation data LT-view for line C_04 is shown in [Figure 5.1](#). The horizontal axis is the time in seconds; the vertical axis – Easting with linear trend was removed. There is the same survey area/time plot than in [Figure 3.5](#), but without linear-trend for easting.

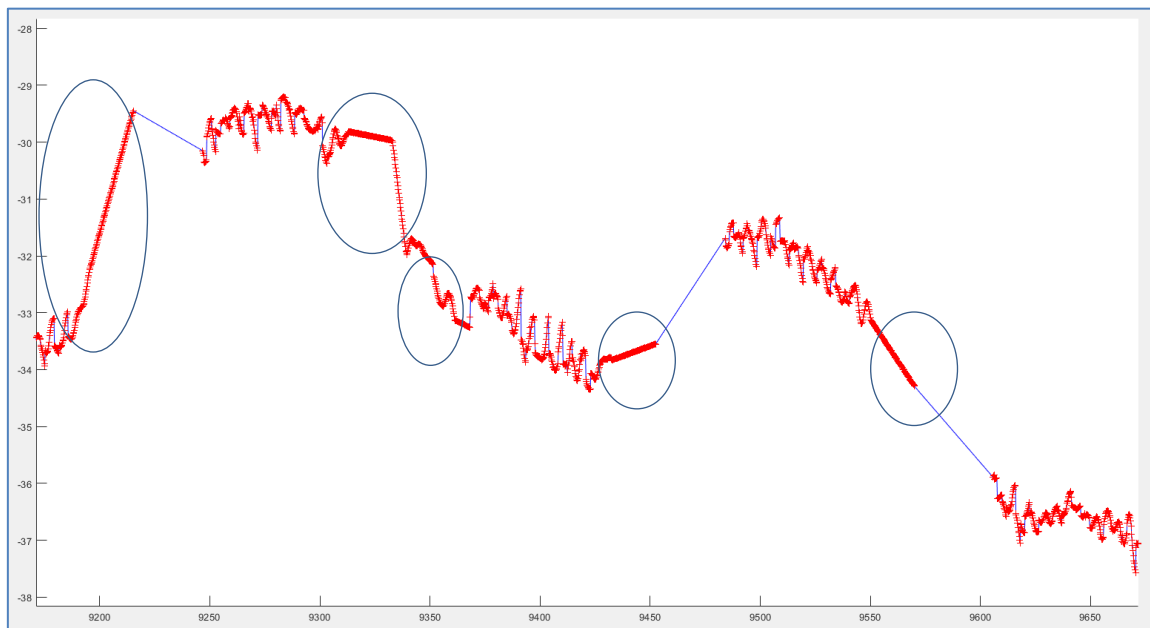


Figure 5.1 LT-view with QINSy predictions for line C_04; red crosses is raw data points

There are following features for [Figure 5.1](#):

- 1) Blue lines without red crosses are the “gaps were detected”. The gaps time is about 30 seconds by horizontal axis;
- 2) The parts with an “oscillation of the coordinates” about 1 meter peak-to-peak is a “good navigation data”. The oscillation is probably the result of un-compensated rolling and pitching of the vessel (for

the shot tow-cable it can be a true result of stern moving, but for a long tow-cable length it lies and need to be smoothed).

3) The linear parts marked in round are “linear extrapolation” by QINSy on-line navigation software. In other words, this is the result of “using prediction settings” for QINSy. These coordinates are not real data, but are the result of extrapolation and the work of Kalman filters. A number of the same “extrapolation shapes” is shown in [Figure 5.2](#).

The extrapolation results shown above are not “raw data”, so it is reasonable to treat them as “noise” or “no record” events. If we remove “linear extrapolation data” (linear-predicted data), the gap length for [Figure 5.1](#) will be increased and we can see plots only for measured data ([Figure 5.3](#)).

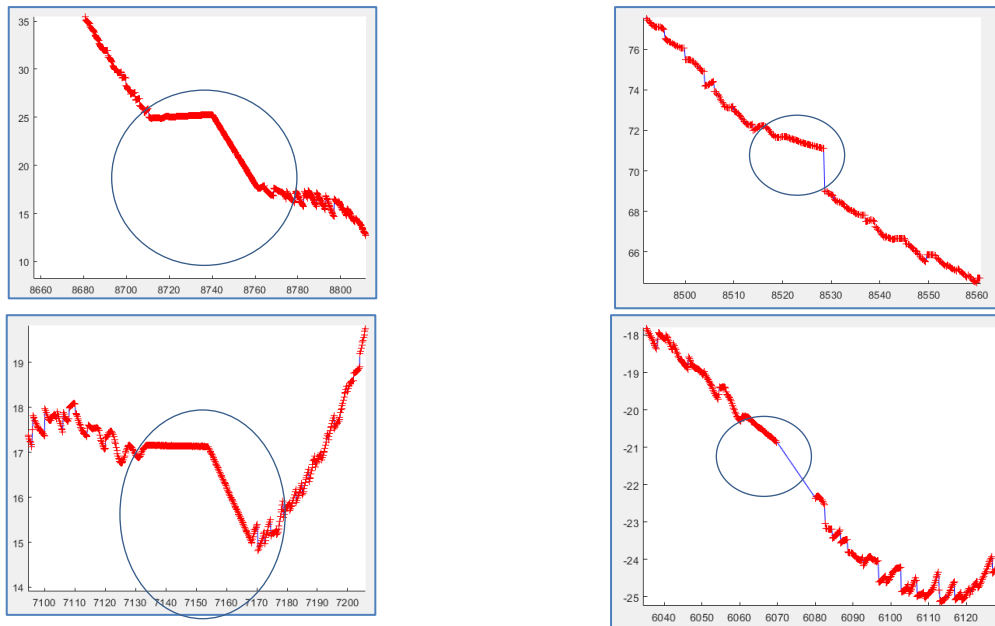


Figure 5.2 Linear “predicted data” shapes for line C_04

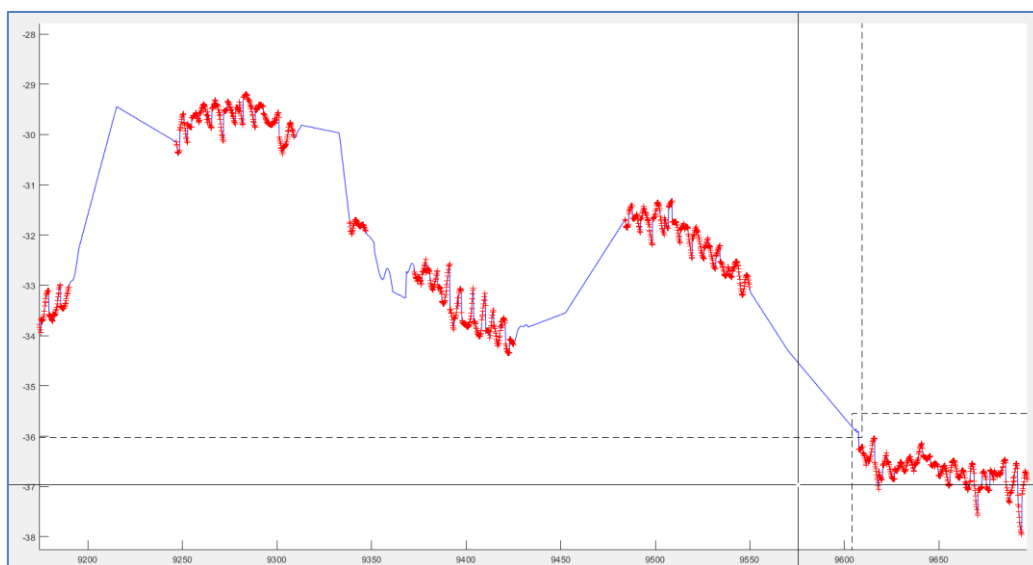


Figure 5.3 LT-view for line C_04; “linear extrapolation” by QINSy was removed

The following processing steps, after removal of a non-measured data, are smoothing for oscillations (*Figure 5.4, Figure 5.5*) and interpolation for missing values (*Figure 5.6*).

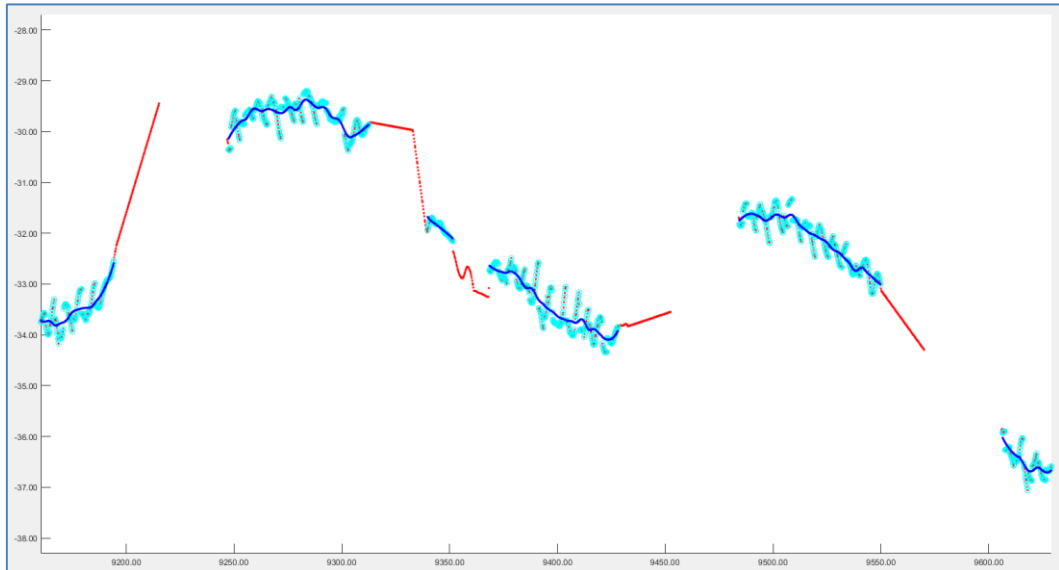


Figure 5.4 LT-view for line C_04; linear “predicted data” by QINSy was removed (cyan circles), the “coordinate’s oscillations” were smoothed (blue points)

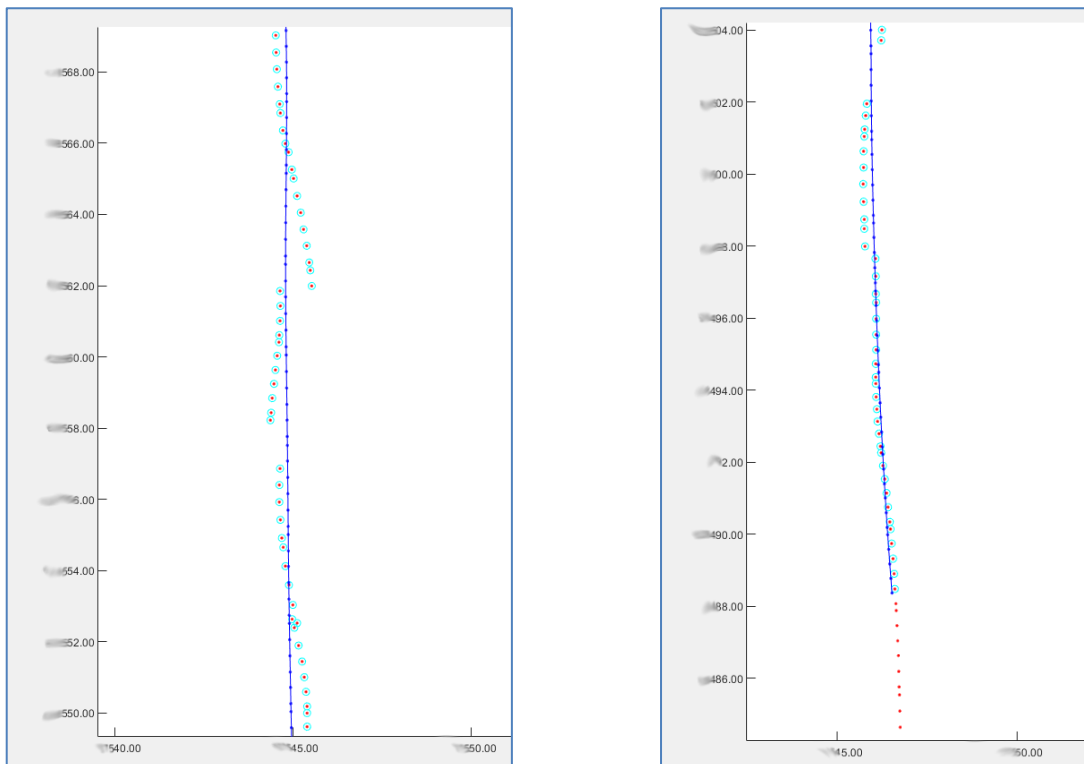


Figure 5.5 Coordinates in EN-axes for line C_04 with “oscillations of coordinate” were smoothed (blue points)

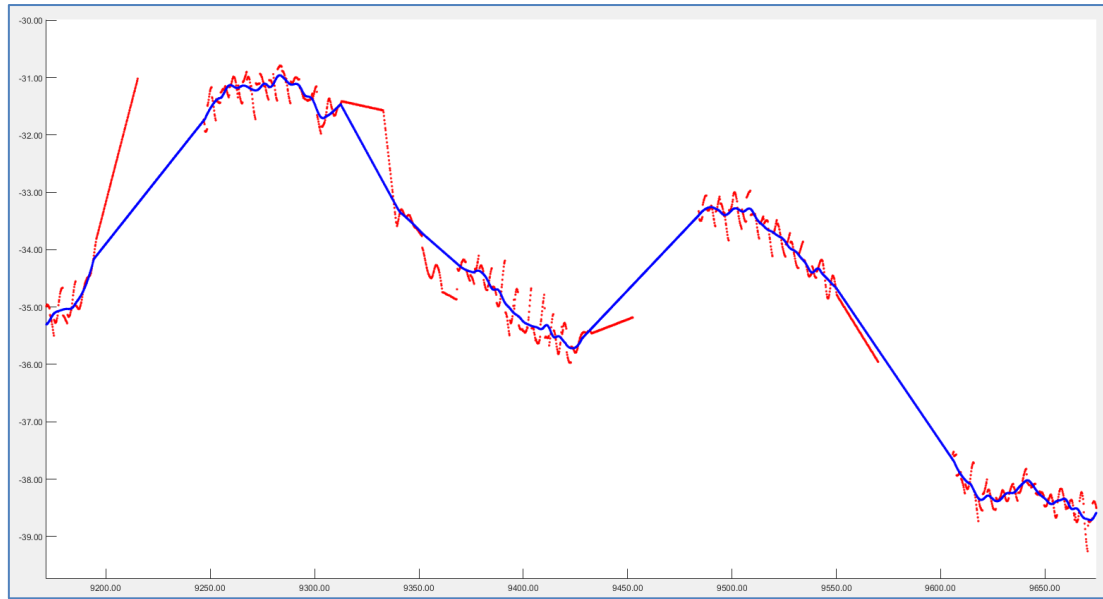


Figure 5.6 LT-view for line C_04; blue points are processed coordinated and interpolated coordinates

The linear interpolation applied (**Figure 5.6**) means, that the track of a vessel inside the gaps must be approximate as a linear. This assumption can be checked with features comparison in sonograms and MBES-data.

The comparisons of coordinates processed by Validator for line C_04 and processed by described steps are shown in **Figure 5.7**. The horizontal axis is the time in seconds; the vertical axis is Easting.

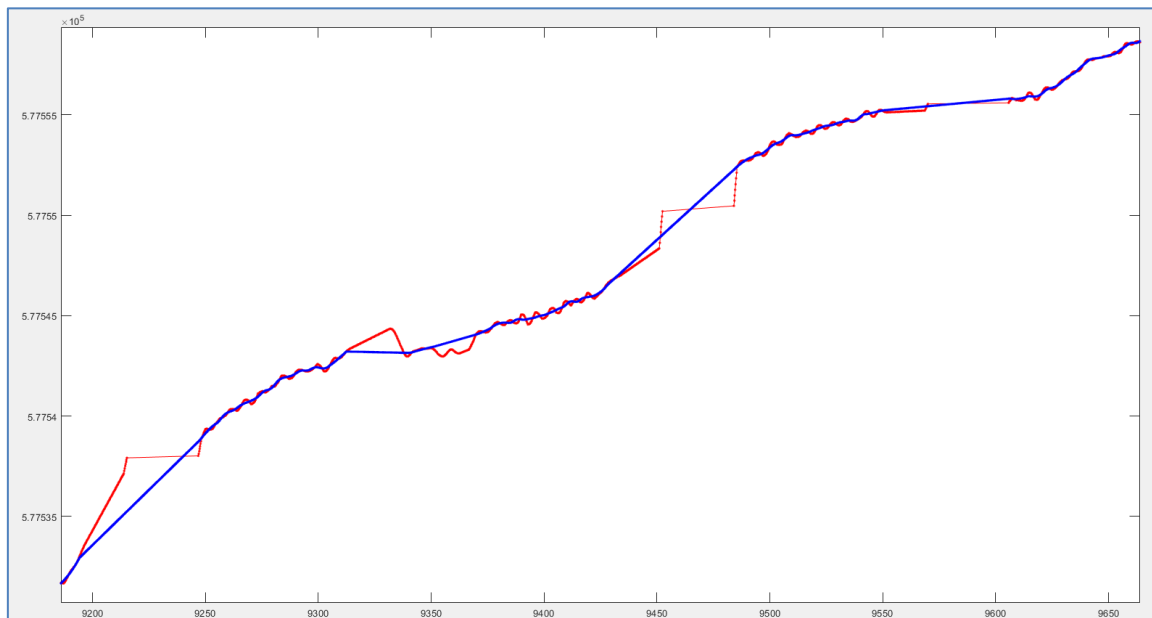


Figure 5.7 LT-view for line C_04 processed by Validator (red points) and processed with QINSy “predictions” removed (blue points); the tips step for vertical axis is 5 meters

To visualize the position of the gaps in SonarWiz, it is convenient to use DWG-files in which the measured data is shown as points. The “gaps” means:

- (a) Gaps without USBL data;
- (b) Gaps were created when spikes were deleted;
- (c) Gaps were created when QINSy “predicted” data was removed from coordinates.

Screenshots from SonarWiz, in which DWG files were loaded, are shown in [Figure 3.10](#), [Figure 3.11](#), [Figure 3.12](#) and [Figure 3.13](#). Using this method we can see valid coordinates. For zones where coordinates were interpolated, we can make careful comparison between SSS and MBES-data.

6 QINSy extrapolation to 1-second step

The period of receiving data from SSS is not equal to one second. Depending on the removal of the towed SSS, it can reach 2-4 seconds. In addition, part of the received data is rejected by the navigation system. Based on the received data, QINSy recalculates the coordinates to a period of 1 second (or 0.1 second). That is, QINSy recalculates the coordinates not only for the case considered above (extrapolation in the absence of data from USBL), but also for all other values. The [Figure 6.1](#) shows the data recalculated by QINSy and the original USBL data (also replayed from QINSy). For careful coordinates processing the original (not extrapolated) USBL data is rationally to use.

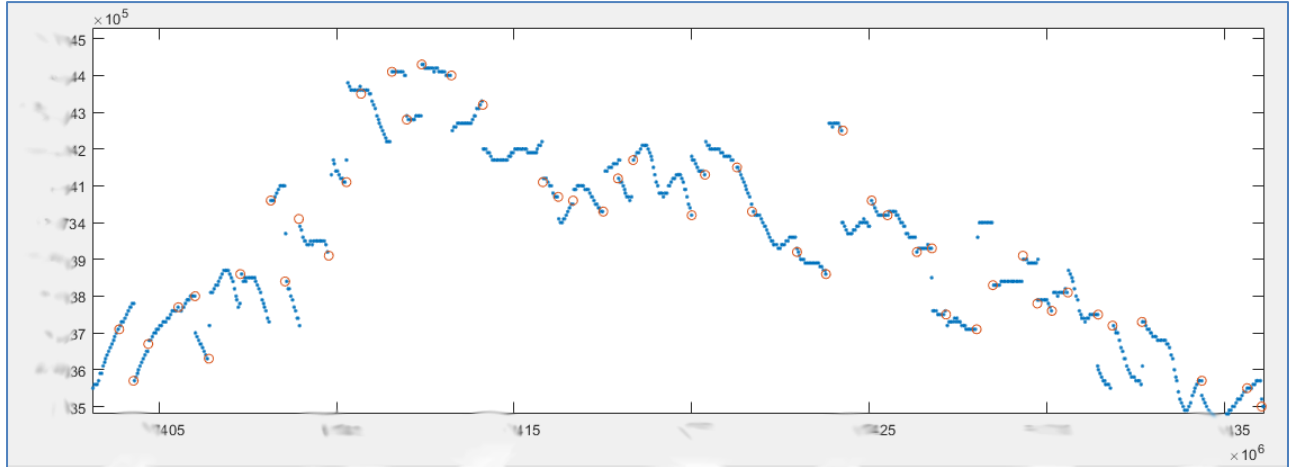


Figure 6.1 Points extrapolated by QINSy (blue) between coordinates from USBL (red)

7 MatLab script for coordinates processing in time-space axes

The script text is shown in Appendix 1. There are following processing commands:

(1)

```
{'NavCODA_Read','e:\05 Raw CNV files\0239\'};Tmp20211013;
```

Read exported from Qinsy navigation data in CODA-format. Read files from folder “e:\05 Raw CNV files\0239\”. The output DWG-file with coordinates creates (for original USBL gaps-area detection).

(2)

```
{'NavDespike&Smooth',200,{'smooth',20,200}};Tmp20211013;
```

Remove spikes and “linear prediction (extrapolation)” parts for coordinates using three axes combinations: E&N, Time&Along-Line, Time&Across-Line (see de-spiking window example in Figure 26). The trend of coordinates calculates using method “smooth” for each 20th point in window 200. The result of de-spiking is smoothed using window in 200 points.

(3)

```
{'HeadDespike&Smooth',200};Tmp20211013;
```

De-spike Heading using axes Angle&Time. Smooth data using window in 200 points.

(4)

```
{'Nav&HeadInterp',0.1};Tmp20210929;
```

Linear interpolation coordinates and heading were de-spiked and smoothed to time-step 0.15 seconds.

(5)

```
{'NavCODA_Write','e:\05 Raw CNV files\0239.txt'};Tmp20210929;
```

Create output file “e:\05 Raw CNV files\0239.txt” in CODA-format and output DWG file with coordinates (for processed gaps-area detection).

Appendix 1. MatLab code

```
%script Tmp20210929;

gKey=ans;
if strcmp(gKey{1},'SU_Set') %define Project's setup
    %{'SU_Set'};Tmp20210929;
    SU.NavS=struct('EllipParam',[6378137 0.0818191908426215],'TargCode',2);
    SU.NavP=struct('EllipParam',[6378137 0.0818191908426215],'ProjParam',[0 15 0.9996 500000
0],'ProjForvFunc','gNavGeog2ProjUtm','ProjRevFunc','gNavProjUtm2Geog','TargCode',6);
    %[E,N,~]=gNavCoord2Coord(B,L,0,SU.NavS,SU.NavP,[2 6]);[B,L,~]=gNavCoord2Coord(E,N,0,SU.NavS,SU.NavP,[6 2]);
end
if strcmp(gKey{1},'NavCODA_Read') %QINSy export read
    %{'NavCODA_Read','e:\0184\'};Tmp20210929;
    try dname=gKey{2};catch,dname=input('NavCODA folder name=');end
    dz=dir(dname);dz([dz(:).isdir])=[];FileName={dz(:).name};
    A=[];
    for n=1:numel(FileName)
        fid=fopen([dname FileName{n}],'r');tmp=fscanf(fid,'%f %f %f %f %f %f %f * * *');fclose(fid);A=[A
reshape(tmp,7,numel(tmp)/7)];
    end

NV=struct('UnixTime',A(1,:), 'CompDay',A(1,:), 'CompTime',A(1,:), 't',A(1,:), 'GpsERaw',A(3,:), 'GpsNRaw',A(4,:), 'HeadCalcRaw',A(5,:), 'GpsE
',A(3,:), 'GpsN',A(4,:), 'HeadCalc',A(5,:), 'EventN',A(2,:), 'Tmp1',A(6,:), 'Tmp2',A(7,:), 'PosSmooth',nan, 'HeadSmooth',nan, 'QMask',uint32(z
eros(size(A(1,:)))));
    NV.CompDay=datetime(1970,1,1)+fix(NV.UnixTime./24./3600);NV.CompTime=mod(NV.UnixTime,24.*3600);NV.t=(NV.CompDay-
NV.CompDay(1)).*86400+NV.CompTime-NV.CompTime(1);
    NV.HeadCalcRaw=unwrap(NV.HeadCalcRaw./180.*pi)./pi*180;NV.HeadCalc=NV.HeadCalcRaw;
    L=find(diff(NV.t)==0);if ~isempty(L);disp('Warning! Duplicate time for
points:');disp(L);NV=gFieldsRowSet(NV,numel(NV.UnixTime),L,[]);end
    PL=struct('PLName',[dname '\acad.scr'],'Type','TrackplotRaw','KeyLineDraw','.-r','GpsE',NV.GpsE,'GpsN',NV.GpsN,'GpsKP',NV.t);
    gMapPLDraw(201,PL);axis equal;gMapPL2AcadExport([dname '\acad.scr'],PL,[100 0 0 0 5],[0.05 1],[0.1 0 60
1],4,1,0);%(fName,PL,PLNameAttr,CircleAttr,KPNameAttr,DigitNum,PointsStep,flLayers)
    clearvars dname dz FileName A n fid tmp L
end
if strcmp(gKey{1},'NavDespike&Smooth') %
    %{'NavDespike&Smooth',150,{'smooth',50,300}};Tmp20211013;
    try NV.PosSmooth=gKey{2};catch,NV.PosSmooth=input('Position smooth filter=');end; try
NV.TrMeth=gKey{3};catch,NV.TrMeth=input('Trend calculate method and parameters=');end %{'smooth',50,300};{'poly',1};
    %NV.QMask=bitand(NV.QMask,4294967295-2);%reset bit2: Handle de-spiked GpsE&GpsN
    NV.GpsE=NV.GpsERaw;NV.GpsN=NV.GpsNRaw;tmp=bitand(NV.QMask,2);NV.GpsE(tmp~=0)=nan;NV.GpsN(tmp~=0)=nan; %set to Nan by Mask for
GpsE&GpsN
    switch NV.TrMeth{1}
        case 'poly' %calc&remove polynomial-trend for E&N
```

```

        L=~isnan(NV.GpsE);plE=polyfit(NV.t(L),NV.GpsE(L),NV.TrMeth{2});NV.trE=polyval(plE,NV.t);
        L=~isnan(NV.GpsN);plN=polyfit(NV.t(L),NV.GpsN(L),NV.TrMeth{2});NV.trN=polyval(plN,NV.t);
        case 'smooth' %calc&remove smooth-trend for E&N

L=find(~isnan(NV.GpsE));plE=smooth(NV.t(L(1:NV.TrMeth{2}:end)),NV.GpsE(L(1:NV.TrMeth{2}:end)),NV.TrMeth{3},'loess');NV.trE=interp1(NV
.t(L(1:NV.TrMeth{2}:end)),plE,NV.t,'pchip','extrap');

L=find(~isnan(NV.GpsN));plN=smooth(NV.t(L(1:NV.TrMeth{2}:end)),NV.GpsN(L(1:NV.TrMeth{2}:end)),NV.TrMeth{3},'loess');NV.trN=interp1(NV
.t(L(1:NV.TrMeth{2}:end)),plN,NV.t,'pchip','extrap');
        otherwise, error('Trend calculate method is not recognize');
    end
    NV.trA=[atan2(diff(NV.trE),diff(NV.trN)) 0];NV.trA(end)=NV.trA(end-1); %calculate line direction for point (0 is to N, clockwise
is +);
    NV.trL=sin(-NV.trA).*(NV.GpsN-NV.trN)+cos(-NV.trA).*(NV.GpsE-NV.trE); %E'=sin(a)*N+cos(a)*E -- rot along trend
    NV.trC=cos(-NV.trA).*(NV.GpsN-NV.trN)-sin(-NV.trA).*(NV.GpsE-NV.trE); %N'=cos(a)*N-sin(a)*E -- rot across trend
    a=figure('Name','E_Despike','NumberTitle','off');hold
on;plot(NV.trE,NV.trN,'b');p=gMapPickHandleNan(NV.GpsE,NV.GpsN,a);pause;mask1=~get(p,'UserData');close(a);NV.trL(mask1)=nan;NV.trC(mas
k1)=nan;

a=figure('Name','E_Despike','NumberTitle','off');p=gMapPickHandleNan(NV.t,NV.trL,a);pause;mask2=~get(p,'UserData');close(a);NV.trL(mas
k1)=nan;NV.trC(mask1)=nan;

a=figure('Name','N_Despike','NumberTitle','off');p=gMapPickHandleNan(NV.t,NV.trC,a);pause;mask3=~get(p,'UserData');close(a);NV.trL(mas
k1)=nan;NV.trC(mask1)=nan;
    L=(mask1|mask2|mask3);NV.GpsE(L)=nan;NV.GpsN(L)=nan;NV.QMask(L)=bitor(NV.QMask(L),2);%>>>Set bit2: Handle de-spiked GpsE&GpsN
    a1=figure('Name','EN_Result','NumberTitle','off');gMapTickLabel(a1,'%2f',9);hold
on;plot(NV.GpsERaw,NV.GpsNRaw,'.r');plot(NV.GpsE,NV.GpsN,'oc');
    a2=figure('Name','E_Result','NumberTitle','off');gMapTickLabel(a2,'%2f',9);hold on;plot(NV.t,sin(-NV.trA).*(NV.GpsNRaw-
NV.trN)+cos(-NV.trA).*(NV.GpsERaw-NV.trE),'.r');plot(NV.t,sin(-NV.trA).*(NV.GpsN-NV.trN)+cos(-NV.trA).*(NV.GpsE-NV.trE),'oc');
    a3=figure('Name','N_Result','NumberTitle','off');gMapTickLabel(a3,'%2f',9);hold on;plot(NV.t,cos(-NV.trA).*(NV.GpsNRaw-NV.trN)-
sin(-NV.trA).*(NV.GpsERaw-NV.trE),'.r');plot(NV.t,cos(-NV.trA).*(NV.GpsN-NV.trN)-sin(-NV.trA).*(NV.GpsE-NV.trE),'oc');
    L=~(mask1|mask2|mask3);
    if
~isempty(NV.PosSmooth),NV.GpsE(L)=smooth(NV.t(L),NV.GpsE(L),NV.PosSmooth,'loess');NV.GpsN(L)=smooth(NV.t(L),NV.GpsN(L),NV.PosSmooth,'
loess');end
    figure(a1);plot(NV.GpsE,NV.GpsN,'.-b');hold off;figure(a2);plot(NV.t,sin(-NV.trA).*(NV.GpsN-NV.trN)+cos(-NV.trA).*(NV.GpsE-
NV.trE),'.-b');hold off;figure(a3);plot(NV.t,cos(-NV.trA).*(NV.GpsN-NV.trN)-sin(-NV.trA).*(NV.GpsE-NV.trE),'.-b');hold off;
    %create output DXF
    clearvars PosSmooth a p plE plN mask1 mask2 mask3 a1 a2 a3 L
end
if strcmp(gKey{1},'HeadDespike&Smooth') %
    %{'HeadDespike&Smooth',80};Tmp20210929;
    try NV.HeadSmooth=gKey{2};catch,NV.HeadSmooth=input('HeadSmooth=');end
    %NV.QMask=bitand(NV.QMask,4294967295-4);%reset bit2: Handle de-spiked GpsE&GpsN
    NV.HeadCalc=NV.HeadCalcRaw;tmp=bitand(NV.QMask,4);NV.HeadCalc(tmp~=0)=nan; %set to Nan by Mask for HeadCalc

```

```

a=figure('Name','Head_Despike','NumberTitle','off');
p=gMapPickHandleNan(NV.t,NV.HeadCalc,a);pause;mask=~get(p,'UserData');close(a);NV.HeadCalc(mask)=nan;
NV.QMask(mask)=bitor(NV.QMask(mask),4);%>>>Set bit3: Handle de-spiked HeadCalc
a=figure('Name','Head_Result','NumberTitle','off');gMapTickLabel(a,'%2f',9);hold
on;plot(NV.t,NV.HeadCalcRaw,'.r');plot(NV.t,NV.HeadCalc,'oc');
if ~isempty(NV.HeadSmooth),NV.HeadCalc(~mask)=smooth(NV.t(~mask),NV.HeadCalc(~mask),NV.HeadSmooth,'loess');end
figure(a);plot(NV.t,NV.HeadCalc,'.-b');hold off;
clearvars HeadSmooth tmp a p mask yL
end
if strcmp(gKey{1},'Nav&HeadInterp') %
    %{'Nav&HeadInterp',0.1};Tmp20210929;
    try StInt=gKey{2};catch,StInt=input('Step of interpolation=');end
    if ~isempty(StInt)
        tt=NV.UnixTime(1):StInt:NV.UnixTime(end);
        NVout=struct('UnixTime',tt,'t',tt-
tt(1),'EventN',nan(size(tt)),'GpsE',nan(size(tt)),'GpsN',nan(size(tt)),'HeadCalc',nan(size(tt)),'Tmp1',zeros(size(tt)),'Tmp2',zeros(
size(tt)));
        NVout.EventN=round(interp1(NV.t,NV.EventN,NVout.t,'linear','extrap'));

L=bitand(NV.QMask,2)==0;NVout.GpsE=interp1(NV.t(L),NV.GpsE(L),NVout.t,'linear','extrap');NVout.GpsN=interp1(NV.t(L),NV.GpsN(L),NVout.t
,'linear','extrap');plE=polyfit(NV.t(L),NV.GpsE(L),1);plN=polyfit(NV.t(L),NV.GpsN(L),1);
        L=bitand(NV.QMask,4)==0;NVout.HeadCalc=interp1(NV.t(L),NV.HeadCalc(L),NVout.t,'linear','extrap');
    else
        L=~(isnan(NV.UnixTime)|isnan(NV.GpsE)|isnan(NV.GpsN)|isnan(NV.HeadCalc));

NVout=struct('UnixTime',NV.UnixTime(L),'t',NV.t(L),'EventN',NV.EventN(L),'GpsE',NV.GpsE(L),'GpsN',NV.GpsN(L),'HeadCalc',NV.HeadCalc(L)
,'Tmp1',NV.Tmp1(L),'Tmp2',NV.Tmp2(L));
        plE=polyfit(NV.t(L),NV.GpsE(L),1);plN=polyfit(NV.t(L),NV.GpsN(L),1);
    end
    a1=figure('Name','EN_Result','NumberTitle','off');gMapTickLabel(a1,'%2f',9);hold on;axis
equal;plot(NV.GpsERaw,NV.GpsNRaw,'.r');plot(NVout.GpsE,NVout.GpsN,'.-b');hold off;
    a2=figure('Name','E_Result','NumberTitle','off');gMapTickLabel(a2,'%2f',9);hold on;plot(NV.t,NV.GpsERaw-
polyval(plE,NV.t),'.r');plot(NVout.t,NVout.GpsE-polyval(plE,NVout.t),'.-b');hold off;
    a3=figure('Name','N_Result','NumberTitle','off');gMapTickLabel(a3,'%2f',9);hold on;plot(NV.t,NV.GpsNRaw-
polyval(plN,NV.t),'.r');plot(NVout.t,NVout.GpsN-polyval(plN,NVout.t),'.-b');hold off;
    a4=figure('Name','Head_Result','NumberTitle','off');gMapTickLabel(a4,'%2f',9);hold
on;plot(NV.t,NV.HeadCalcRaw,'.r');plot(NVout.t,NVout.HeadCalc,'.-b');hold off;
    clearvars StInt tt L plE plN a1 a2 a3 a4
end
if strcmp(gKey{1},'NavCODA_Write') %
    %{'NavCODA_Write','e:\0184.txt'};Tmp20210929;
    try fname=gKey{2};catch,fname=input('NavCODA output file name=');end
    L=NVout.HeadCalc<0;NVout.HeadCalc(L)=360+NVout.HeadCalc(L);
    fid=fopen(fname,'w');fprintf(fid,'%0.2f %0.0f %0.2f %0.2f %0.2f %0.3f %0.2f * * *
*\r\n',[NVout.UnixTime;NVout.EventN;NVout.GpsE;NVout.GpsN;NVout.HeadCalc;NVout.Tmp1;NVout.Tmp2]);fclose(fid);

```

```

    L=bitand(NV.QMask,2)==0;PL=struct('PLName',fname,'Type','TrackplotRaw','KeyLineDraw','.-
r','GpsE',NV.GpsE(L),'GpsN',NV.GpsN(L),'GpsKP',NV.t(L));
    gMapPLDraw(201,PL);axis equal;gMapPL2AcadExport([fname '.scr'],PL,[100 0 0 0 5],[0.05 1],[0.1 0 60
1],4,1,0);%(fName,PL,PLNameAttr,CircleAttr,KPNameAttr,DigitNum,PointsStep,flLayers)
    clearvars fname fid L PL
end

%mail@ge0mlib.com 20/10/2021

```