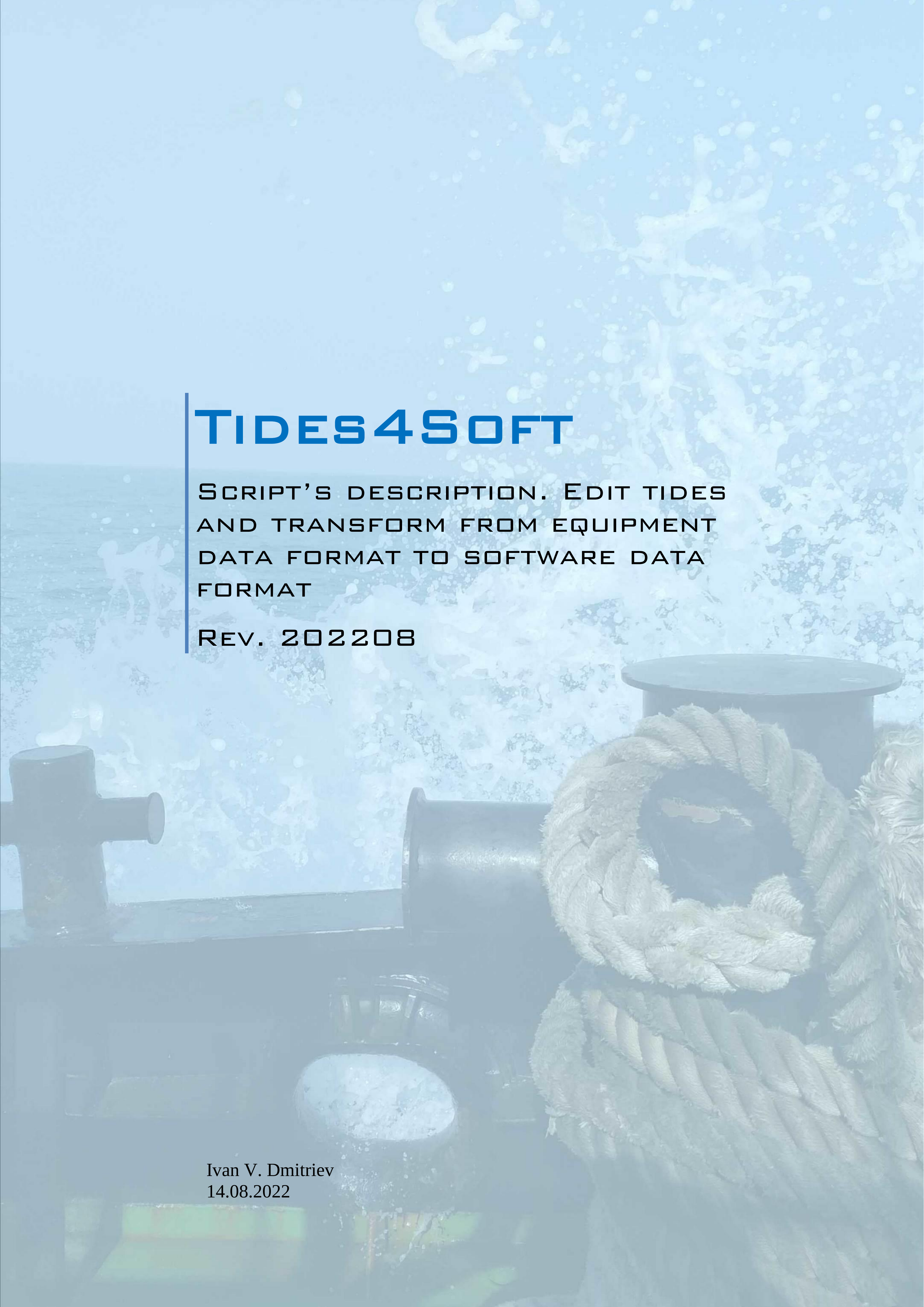


Tides4Soft

Script’s description. Edit tides and transform from equipment data format to software data format

Rev. 202208

Ivan V. Dmitriev
14.08.2022

Contents

Figures list

1. Overview and requirements

Tides recorded in equipment's format need to be processed and prepared in a specific software's format (DugInsight or Silas). The MatLab script is used to solve this task; you need to install MatLab 2018a software (or later version) and ge0mlib library.

Two formats from USV can be used for input data (with Date and without Date).

Input data format 1 example (USV data format):

```
#<Time> HHmmss.zzz,Latitude WGS-84 - COG - Latitude WGS-84,Longitude WGS-84 - COG - Longitude WGS-84,Height WGS-84 - COG - Height WGS-84,Easting - COG - Easting,Northing - COG - Northing,Height - COG - Height
161248.237,5520.17000858,00134.00310710,54.775,580051.32,6100123.86,8.44
161249.001,5520.17000347,00134.00310369,54.777,580051.33,6100123.85,8.44
```

Tide's column is the last. For "format 1" dates calculated using the name of file, which included this information. The date symbols' position is calculated as fName(end-16:end-9).

Files names example: X18_Tide_20220715_0000.log; X18_Tide_20220716_0000.log

Input data format 2 example (USV data format):

```
#<Time> HHmmss.zzz,<Date> dd,<Date> MM,<Date> yyyy,Latitude WGS-84 - COG - Latitude WGS-84,Longitude WGS-84 - COG - Longitude WGS-84,Height WGS-84 - COG - Height WGS-84,Easting - COG - Easting,Northing - COG - Northing,Height - COG - Height
000001.002,19,07,2022,5516.94135841,00117.68878421,46.600,608305.37,6127537.33,0.87
000002.002,19,07,2022,5516.94081535,00117.68899275,46.623,608305.17,6127536.32,0.89
```

Input data format 3 example:

```
23:27:38,2.39
23:28:08,2.60
```

The date must be in the file name's position: fNameN(end-11:end-4)

Input file names: tide_innomar_20220512.txt; tide_innomar_20220513.txt

Warning! The "bad rows" (out of format) must be deleted from input files before script using.

The "bad rows" example:

```
13:57:32,
13:58:02,
```

The follow processing steps are applied:

-- load and concatenate all files (sorted by names) from folder defined

-- decimation for quick processing (for quick tides smoothing, etc);

-- manual despike using gMapPickHandleNan tools,

-- smoothing in a defined slice-window,

-- interpolation for gaps,

-- decimation to 1-minute step,

-- save processed tides.

Output data format example (DugInsight data format):

```
1612:00 15      7       2022    4.74
1613:00 15      7       2022    4.74
```

Output data format example (Silas tdx format):

```
%08:40 1.974
%08:41 2.007
```

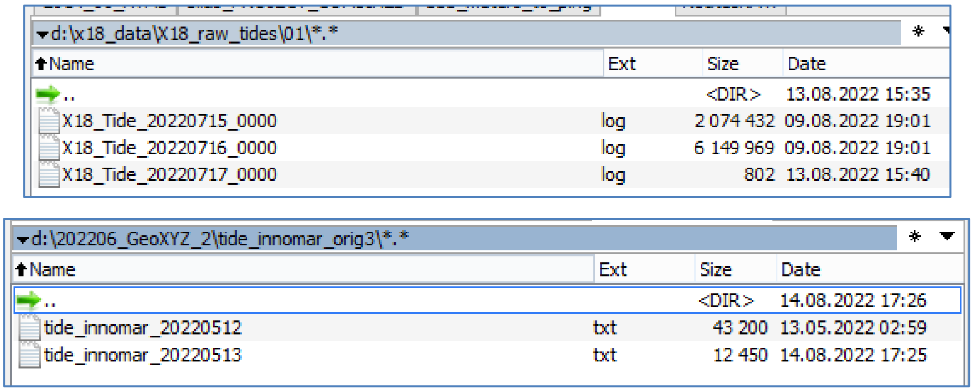
The files are created for each day separately. It can be defined output folder’s name only.

Output file names example: Tide_20220512.tdx; Tide_20220513.tdx

2. Script commands

0) ================

To start work, the files recorded with USV-equipment need to be copied in separate folder (***Figure 2.1***).



d:\x18_data\X18_raw_tides\01\*.*

Name	Ext	Size	Date
..		<DIR>	13.08.2022 15:35
X18_Tide_20220715_0000	log	2 074 432	09.08.2022 19:01
X18_Tide_20220716_0000	log	6 149 969	09.08.2022 19:01
X18_Tide_20220717_0000	log	802	13.08.2022 15:40

d:\202206_GeoXYZ_2\tide_innomar_orig3\*.*

Name	Ext	Size	Date
..		<DIR>	14.08.2022 17:26
tide_innomar_20220512	txt	43 200	13.05.2022 02:59
tide_innomar_20220513	txt	12 450	14.08.2022 17:25

Figure 2.1 Input files

For formats 1 and 3 dates calculated using the name of file, which included this information. The position of a date string is calculated as fName(end-16:end-9) and as a fNameN(end-11:end-4).

Files will be concatenated as sorted by names.

1) ================

Load input files:

```
>> {'TidesRead','d:\x18_data\X18_raw_tides\01\',1};Tides4Soft;
```

There are:

-- 'TidesRead' – a command name of the script;

-- 'd:\x18_data\X18_raw_tides\01\' – folder name with input data;

-- 1 – the input data format number (1 or 2).

When data loaded the window with tides is drawn, along axis in seconds (***Figure 2.2***).

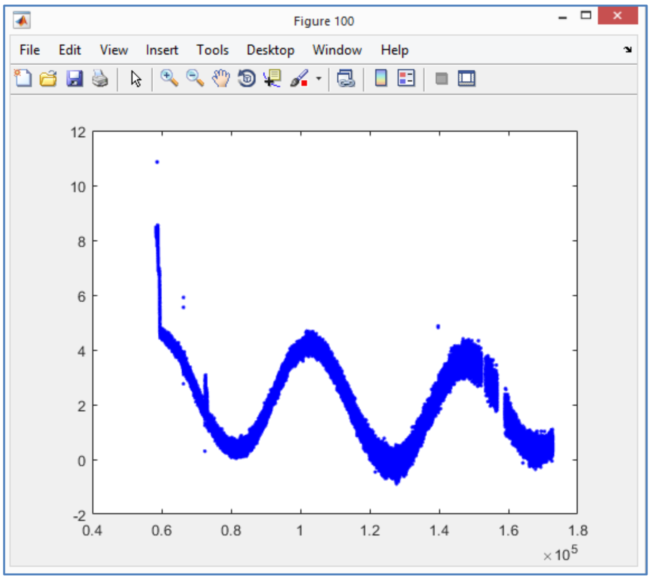


Figure 2.2 Tides loaded

2) ================

Decimate data (if necessary):

```
>> {'TidesDecimate',10};Tides4Soft;
```

-- 'TidesDecimate' – a command name of the script;

-- 10 – each 10th value will be kept.

When data decimated, the window with tides is drawn (*Figure 2.3*). There is only 10th value here.

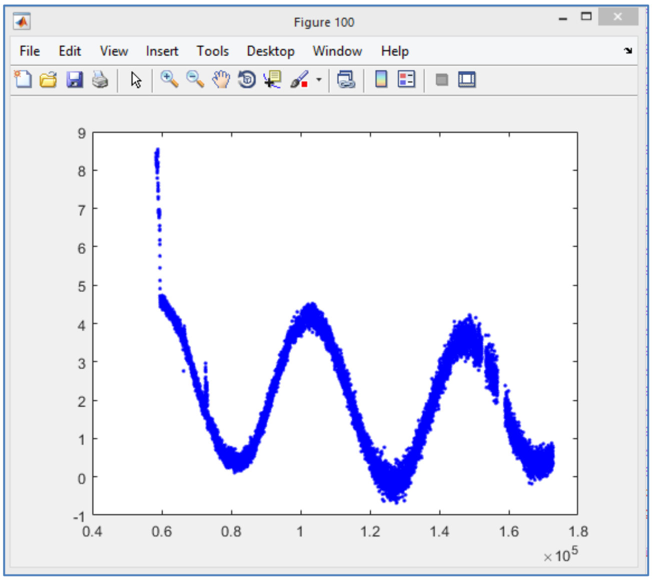


Figure 2.3 Tides decimated

3) ================

Manual despike, smoothing and interpolation for gaps:

```
>> {'TidesDespSmoothInterp',600};Tides4Soft;
```

-- 'TidesDespSmoothInterp' – a command name of the script;

-- 600 – slice window for smoothing in "tides measurement points number".

When command input, the Manual despiked window is open (*Figure 2.4*)

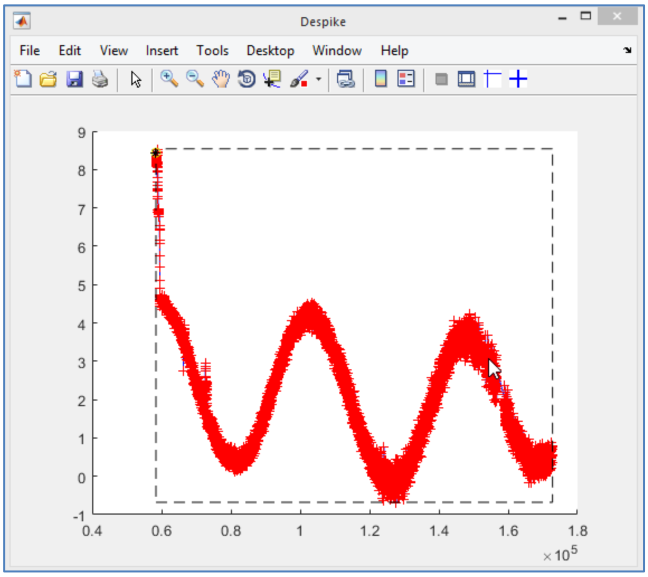


Figure 2.4 Manual despiked window

Using two tools (see icons), the spikes can be selected and deleted (*Figure 2.5*).

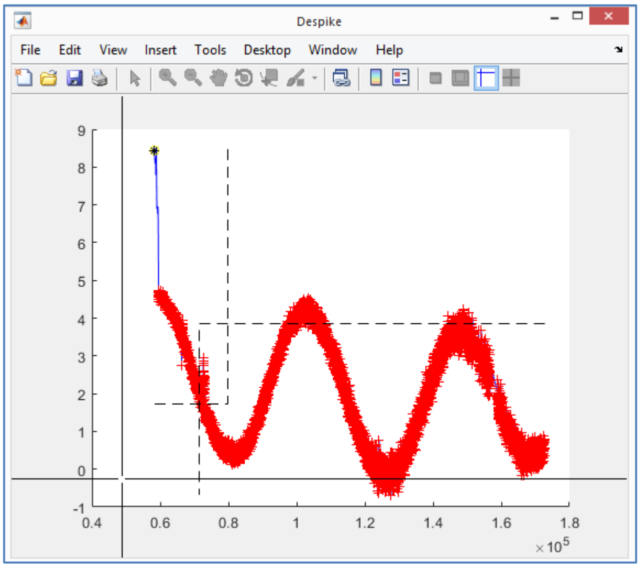


Figure 2.5 Manual despiked window – deleting spikes process

The follow keys are used:

Click to icon – come to “edit mode”;

LeftMouseButton – edit mode, first selection element (first part of rectangle or first point at curve);

RightMouseButton – edit mode, second selection element (second part of rectangle or second point at curve);

MiddleMouseButton – edit mode, "deleteing" points in selected area (set to NaN Y-coordinate value);

z – edit mode, undo;

x – edit mode, redo;

q – exit from edit mode;

Backspace (or any key) – finish manual despike, when not in “edit mode”.

Not in “edit mode” we can use icons to select zoom or pan.

When Manual despike is finished, the window with results is being drawing (*Figure 2.6*)

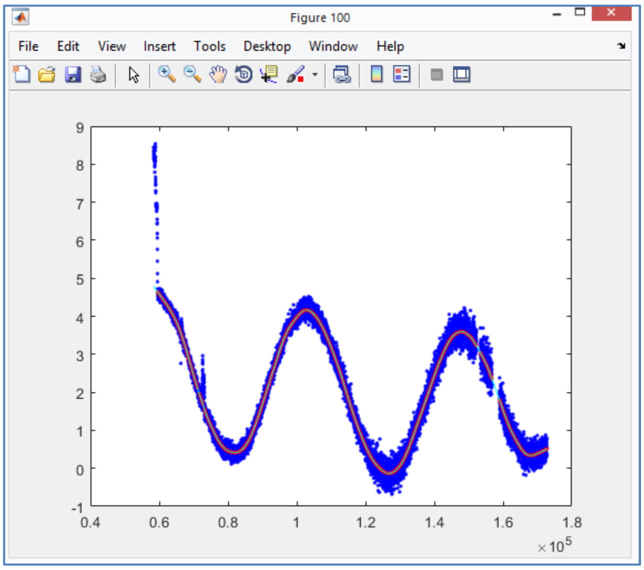


Figure 2.6 Processing result window

The blue points are the original data.

The red line is the smoothed data.

The blue line is the interpolation result for data smoothed (to close gaps).

4) ================

Processing result save. The DugInsight or Silas formats can be used.

```
>> {'TidesWrite','d:\x18_data\X18_raw_tides\out.txt',1};Tides4Soft;
```

-- 'TidesWrite'– a command name of the script;

-- 'd:\x18_data\X18_raw_tides\out.txt' – file name for data output;

-- 1 – DugInsight format.

```
>> {'TidesWrite','d:\202206_GeoXYZ_2\tide_innomar_orig4\',2};Tides4Soft;
```

-- 'TidesWrite'– a command name of the script;

-- 'd:\202206_GeoXYZ_2\tide_innomar_orig4\' – folder name for data output;

-- 2 – Silas format.

When command input of the processed data will be decimated to 1-minute interval and saved to the file. The decimated data will be drawn (*Figure 2.7*).

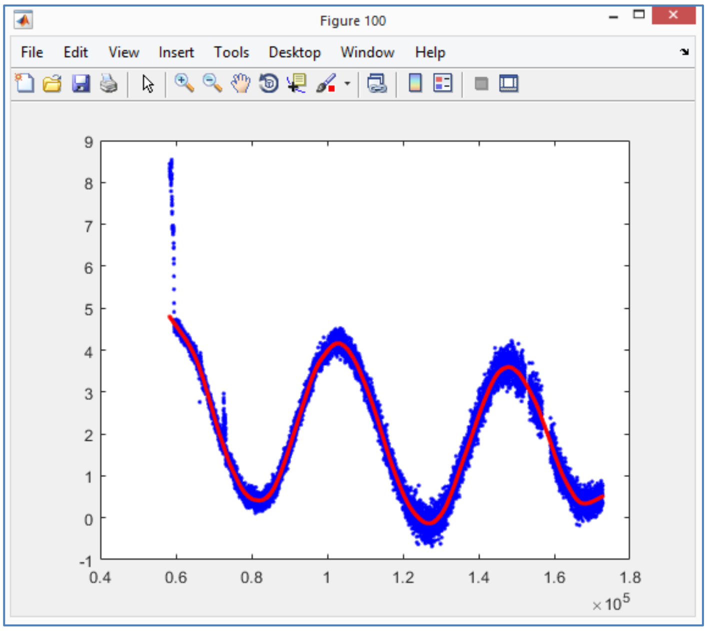
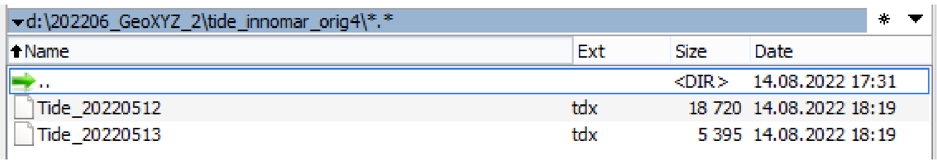


Figure 2.7 Data saved (1-minute step for red points)

The blue points are the original data.

The red points are the processed data with a 1-minute step.

The files are created for each day separately for the Silas format (*Figure 2.8*).



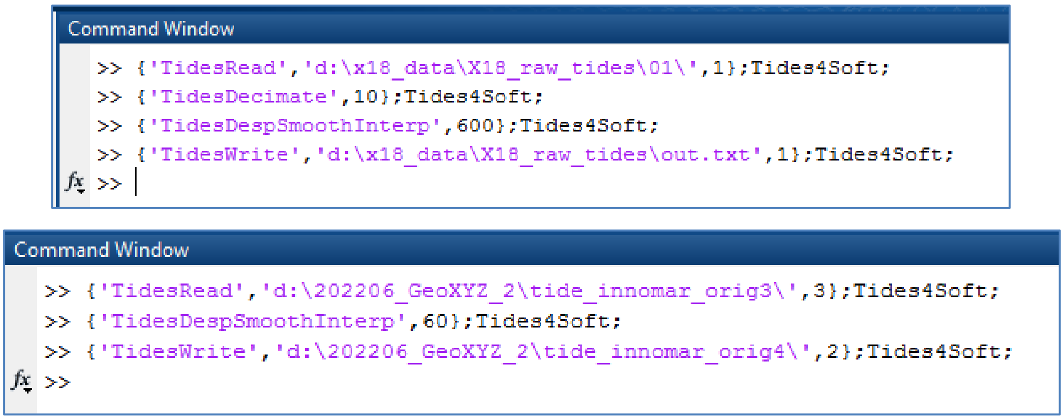
d:\202206_GeoXYZ_2\tide_innomar_orig4\*.*

Name	Ext	Size	Date
..		<DIR>	14.08.2022 17:31
Tide_20220512	tdx	18 720	14.08.2022 18:19
Tide_20220513	tdx	5 395	14.08.2022 18:19

Figure 2.8 Silas output files

5) ================

Congratulation! You can use the output file(s) with tides in DugInsight or Silas software.



```
>> {'TidesRead','d:\x18_data\X18_raw_tides\01\',1};Tides4Soft;
>> {'TidesDecimate',10};Tides4Soft;
>> {'TidesDespSmoothInterp',600};Tides4Soft;
>> {'TidesWrite','d:\x18_data\X18_raw_tides\out.txt',1};Tides4Soft;
>>
```

```
>> {'TidesRead','d:\202206_GeoXYZ_2\tide_innomar_orig3\',3};Tides4Soft;
>> {'TidesDespSmoothInterp',60};Tides4Soft;
>> {'TidesWrite','d:\202206_GeoXYZ_2\tide_innomar_orig4\',2};Tides4Soft;
>>
```

Figure 2.9 Script’s commands in MatLab windows

Appendix 1. Script code

```
%script Tides4Soft;
%Edit tides and transform from equipment data format to software data format (DugInsight, Silas)
%==== Input format 1 (USV data format) >>>
%#<Time> HHmmss.zzz,Latitude WGS-84 - COG - Latitude WGS-84,Longitude WGS-84 - COG - Longitude WGS-84,Height WGS-84 - COG - Height WGS-84,Easting - COG
- Easting,Northing - COG - Northing,Height - COG - Height
%161248.237,5520.17000858,00134.00310710,54.775,580051.32,6100123.86,8.44
%161249.001,5520.17000347,00134.00310369,54.777,580051.33,6100123.85,8.44
%Input file names: X18_Tide_20220715_0000.log; X18_Tide_20220716_0000.log
%The date must be in the file name's position: fName(end-16:end-9)
%==== Input format 2 (USV data format) >>>
%#<Time> HHmmss.zzz,<Date> dd,<Date> MM,<Date> yyyy,Latitude WGS-84 - COG - Latitude WGS-84,Longitude WGS-84 - COG - Longitude WGS-84,Height WGS-84 -
COG - Height WGS-84,Easting - COG - Easting,Northing - COG - Northing,Height - COG - Height
%000001.002,19,07,2022,5516.94135841,00117.68878421,46.600,608305.37,6127537.33,0.87
%000002.002,19,07,2022,5516.94081535,00117.68899275,46.623,608305.17,6127536.32,0.89
%Input file names: X18_Tide_20220715_0000.log; X18_Tide_20220716_0000.log
%==== Input format 3 >>>
%23:27:38,2.39
%23:28:08,2.60
%Input file names: tide_innomar_20220512.txt; tide_innomar_20220513.txt
%The date must be in the file name's position: fNameN(end-11:end-4)
%==== Output format 1 (DugInsight data format) >>>
%1612:00    15  7   2022    4.74
%1613:00    15  7   2022    4.74
%==== Output format 2 (Silas tdx format) >>>
%08:40 1.974
%08:41 2.007
%The files create for each day separately. Output file names example: Tide_20220512.tdx; Tide_20220513.tdx
%==== Commands example >>>
%{'TidesRead','d:\x18_data\X18_raw_tides\01\',1};Tides4Soft; -- Read tides files from folder;
%{'TidesDecimate',10};Tides4Soft; -- Decimate tides for quick smoothing (use each 10 values only);
%{'TidesDespSmoothInterp',600};Tides4Soft; -- Manual despike, smoothing in a window 600 points and interpolation for gaps;
%{'TidesWrite','d:\x18_data\X18_raw_tides\out.txt'};Tides4Soft; -- Create and write file for DugInsight with data-per-minute
%==== Keys for manual despike (use icon to select edit tool) >>>
%LeftMouseButton - first selection element (first part of rectangle or first point at curve);
%RightMouseButton - second selection element (second part of rectangle or second point at curve);
%MiddleMouseButton - "delete" points in selected area (set to NaN Y-coordinate value);
%z - undo; x - redo; q - Exit from edit mode. Backspace (or any key) - finish manual despike.

gKey=ans;
if strcmp(gKey{1},'TidesRead') %Read tides files from folder
    %{'TidesRead','d:\x18_data\X18_raw_tides\01\',1};Tides4Soft;
    %{'TidesRead','d:\tide_innomar_orig3\',3};Tides4Soft;
    try fName=gKey{2};catch,fName=input('Input folder name=');end; try ff=gKey{3};catch,ff=input('Input format type=');end
    dz=dir(fName);dz([dz.isdir])=[];fName=[repmat(fName,length(dz),1) char(dz.name)];TD=[];
    for nn=1:size(fName,1)
        fNameN=deblank(fName(nn,:));
        switch ff
            case 1
```

```
                [~,tmp]=gDataTxtRead(fNameN,1,7,'%f',',','\r\n');
                GpsTime=gNavTime2Time('HMS2Sd',tmp(:,1)');GpsDay=repmat(gNavTime2Time('YMD2Dx',str2num(fNameN(end-16:end-9))),size(GpsTime));
                GpsLat=gNavAng2Ang('DM2D',tmp(:,2)');GpsLon=gNavAng2Ang('DM2D',tmp(:,3)');

TD0=struct('GpsDay',GpsDay,'GpsTime',GpsTime,'GpsLat',GpsLat,'GpsLon',GpsLon,'GpsHgtGeoid',tmp(:,4)','GpsE',tmp(:,5)','GpsN',tmp(:,6)','Tide',tmp(:,7)'
);
            case 2
                [~,tmp]=gDataTxtRead(fNameN,1,10,'%f',',','\r\n');
                GpsTime=gNavTime2Time('HMS2Sd',tmp(:,1)');GpsDay=gNavTime2Time('YMD32Dx',tmp(:,4)',tmp(:,3)',tmp(:,2)');
                GpsLat=gNavAng2Ang('DM2D',tmp(:,5)');GpsLon=gNavAng2Ang('DM2D',tmp(:,6)');

TD0=struct('GpsDay',GpsDay,'GpsTime',GpsTime,'GpsLat',GpsLat,'GpsLon',GpsLon,'GpsHgtGeoid',tmp(:,7)','GpsE',tmp(:,8)','GpsN',tmp(:,9)','Tide',tmp(:,10)
');
            case 3
                fid=fopen(fNameN,'r');tmp=fscanf(fid,'%d:%d:%d,%f',[4 inf]);fclose(fid);
                GpsTime=gNavTime2Time('HMS32Sd',tmp(1,:),tmp(2,:),tmp(3,:));GpsDay=repmat(gNavTime2Time('YMD2Dx',str2num(fNameN(end-11:end-
4))),size(GpsTime));
                TD0=struct('GpsDay',GpsDay,'GpsTime',GpsTime,'Tide',tmp(4,:));
            otherwise,error('Incorrect format number!');
        end
        if isempty(TD),TD=TD0;else,TD=gFieldsRowAppend(TD,TD0,numel(TD.GpsTime));end
    end
    [TD.GpsDay,TD.GpsTime]=gNavDayCheck(TD.GpsDay,TD.GpsTime);
    [~,TD.t]=gNavTime2Time('DxSd2DmS',TD.GpsDay,TD.GpsTime);L=[false diff(TD.t)==0];TD=gFieldsRowSet(TD,numel(TD.GpsTime),L,[]);
    TD.Mask=true(size(TD.GpsTime));TD.TideSm=nan(size(TD.GpsTime));TD.TideSmInt=nan(size(TD.GpsTime));
    figure(100);plot(TD.t,TD.Tide,'.b');
    clearvars fName fid ff dz nn fNameN tmp GpsTime GpsDay GpsLat GpsLon TD0 L;
end
if strcmp(gKey{1},'TidesDecimate') %Use each N values only (decimate tides for quick smoothing)
    %{'TidesDecimate',10};Tides4Soft;
    try dK=gKey{2};catch,dK=input('Decimate step=');end
    TD=gFieldsRowGet(TD,numel(TD.GpsTime),1:dK:numel(TD.GpsTime));
    figure(100);plot(TD.t,TD.Tide,'.b');
    clearvars dK;
end
if strcmp(gKey{1},'TidesDespSmoothInterp') %Despike manualy, smoothing and interpolation
    %{'TidesDespSmoothInterp',600};Tides4Soft;
    try smW=gKey{2};catch,smW=input('Smooth window size=');end
    a=figure('Name','Despike','NumberTitle','off');p=gMapPickHandleNan(TD.t,TD.Tide,a);pause;mask=~get(p,'UserData');close(a);TD.Mask(mask)=false;
    TideSm=smooth(TD.t(TD.Mask),TD.Tide(TD.Mask),smW,'lowess');TD.TideSm(TD.Mask)=TideSm;
    TD.TideSmInt=interp1(TD.t(TD.Mask),TD.TideSm(TD.Mask),TD.t,'spline','extrap');
    figure(100);plot(TD.t,TD.Tide,'.b');hold on;plot(TD.t,TD.TideSm,'.r');plot(TD.t,TD.TideSmInt,'-c');hold off;
    clearvars smW a p mask TideSm;
end
if strcmp(gKey{1},'TidesWrite') %Create and write file for software with value-per-minute
    %{'TidesWrite','d:\x18_data\X18_raw_tides\out.txt',1};Tides4Soft;
    %{'TidesWrite','d:\tide_innomar_orig4\',2};Tides4Soft;
    try fName=gKey{2};catch,fName=input('Output file or folder name=');end; try ff=gKey{3};catch,ff=input('Output format type=');end
    t=fix(TD.t(1)./60)*60:60:TD.t(end);[~,di]=min(abs(t-TD.t(1)));d=TD.GpsDay(di);
    TideSmInt=interp1(TD.t,TD.TideSmInt,t,'spline','extrap');figure(100);plot(TD.t,TD.Tide,'.b');hold on;plot(t,TideSmInt,'.r');
    [GpsDay,GpsTime]=gNavTime2Time('DmS2DxSd',d,t);
    switch ff
        case 1
            [Y,Mm,D]=gNavTime2Time('Dx2YMD3',GpsDay);[H,M,S]=gNavTime2Time('Sd2HMS3',GpsTime);
            fid=fopen(fName,'w');fprintf(fid,'%02d%02d:%02d\t%d\t%d\t%d\t%0.2f\t\t\r\n',[H;M;S;D;Mm;Y;TideSmInt]);fclose(fid);
        case 2
            L=[1 find(diff(GpsDay))+1 numel(GpsDay)+1];
            for nn=1:numel(L)-1
                [YMD]=gNavTime2Time('Dx2YMD',GpsDay(L(nn)));[H,M,S]=gNavTime2Time('Sd2HMS3',GpsTime(L(nn):L(nn+1)-1));
                fid=fopen([fName '\Tide_' num2str(YMD) '.tdx'],'w');fprintf(fid,'%02d:%02d %0.3f\r\n',[H;M;TideSmInt(L(nn):L(nn+1)-1)]);fclose(fid);
            end
        otherwise,error('Incorrect format number!');
    end
    clearvars fName ff t di d TideSmInt GpsDay GpsTime Y Mm D H M S L nn YMD fid;
end

%mail@ge0mlib.com 13/08/2022
%With regards, for I.B. Just for fun.
```