RSKtools for Matlab processing RBR data

Table of Contents

RSKtools v2.0.0; RBR Ltd. Ottawa ON, Canada; support@rbr-global.com; 2017-07-07

Introduction

A suite of new functions are included in RSKtools v2.0.0 to post-process RBR logger data. Below we show how to implement some common processing steps to obtain the highest quality data possible.

Getting set up

See review RSKtools_vignette for help.

```
file = 'sample.rsk';
rsk = RSKopen(file);
rsk = RSKreadprofiles(rsk, 'profile', 10:55, 'direction', 'up');
```

Low-pass filtering

The first step is generally to apply a low pass filter to the pressure data; then filter the temperature and conductivity channels to smooth high frequency variability and match sensor time constants. RSKtools includes a function called `RSKsmooth` for this purpose. All post-processing functions are customizable with name-value pair input arguments. To process all data using the default parameters no name-value pair arguments are required. All the information above is available for each function using `help`, for example: `help RSKsmooth`.

```
help RSKsmooth
```

```
 RSKsmooth - Apply a low pass filter on specified channels.

  Syntax:  [RSK] = RSKsmooth(RSK, channel, [OPTIONS])
  Low-pass filter a specified channel or multiple channels with a
  running average or median.  The sample being evaluated is always in
  the centre of the filtering window to avoid phase distortion.  Edge
  effects are handled by mirroring the original time series.

  Inputs:
     [Required] - RSK - Structure containing the logger data.

                  channel - Longname of channel to filter. Can be a
                        single channel, a cell array for multiple
                        channels, or 'all' for all channels.

     [Optional] - filter - The weighting function, 'boxcar' or
 'triangle'.
                        Use 'median' to compute the running median.
                        Defaults to 'boxcar.'

                  profile - Profile number. Defaults to operate on all
                        available profiles.

                  direction - 'up' for upcast, 'down' for downcast, or
                        'both' for all. Defaults to all directions
 available.

                  windowLength - The total size of the filter window.
 Must
                        be odd. Default is 3.

  Outputs:
     RSK - Structure with filtered values.

  Example:
     rsk = RSKopen('file.rsk');
     rsk = RSKreadprofiles(rsk, 'profile', 1:10); % read first 10
 downcasts
     rsk = RSKsmooth(rsk, {'Temperature', 'Salinity'}, 'windowLength',
 17);

  Author: RBR Ltd. Ottawa ON, Canada
  email: support@rbr-global.com
  Website: www.rbr-global.com
  Last revision: 2017-06-28
```

```
rsk = RSKsmooth(rsk, 'Pressure');
rsk = RSKsmooth(rsk, {'Conductivity', 'Temperature'}, 'windowLength',
 21);
```

Aligning CT

RSKtools provides a function called `RSKcalculateCTlag` that estimates the optimal lag between conductivity and temperature by minimising salinity spiking. See `help RSKcalculateCTlag`.

```
lag = RSKcalculateCTlag(rsk);
```

```
rsk = RSKalignchannel(rsk, 'Conductivity', lag);
```

```
Processing cast: 1
Processing cast: 2
Processing cast: 3
Processing cast: 4
Processing cast: 5
Processing cast: 6
Processing cast: 7
Processing cast: 8
Processing cast: 9
Processing cast: 10
Processing cast: 11
Processing cast: 12
Processing cast: 13
Processing cast: 14
Processing cast: 15
Processing cast: 16
Processing cast: 17
Processing cast: 18
Processing cast: 19
Processing cast: 20
Processing cast: 21
Processing cast: 22
Processing cast: 23
Processing cast: 24
Processing cast: 25
Processing cast: 26
Processing cast: 27
Processing cast: 28
Processing cast: 29
Processing cast: 30
Processing cast: 31
Processing cast: 32
Processing cast: 33
Processing cast: 34
Processing cast: 35
Processing cast: 36
Processing cast: 37
Processing cast: 38
Processing cast: 39
Processing cast: 40
Processing cast: 41
Processing cast: 42
Processing cast: 43
Processing cast: 44
Processing cast: 45
Processing cast: 46
```

Remove loops

Profiling during rough seas can cause the CTD descent (or ascent) rate to vary, or even temporarily reverse direction, while profiling. During such times, the CTD can effectively sample its own wake, potentially degrading the quality of the profile in regions of strong gradients. The measurements taken too slowly or during a pressure reversal should not be used for further analysis. We recommend using `RSKremoveloops` to flag and remove data when the instrument falls below a `threshold` speed. This function requires a depth channel, for which we have provided `RSKderivedepth`.

```
rsk = RSKderivedepth(rsk);
rsk = RSKremoveloops(rsk, 'threshold', 0.3);
```

Derive

Functions are provided to derive sea pressure, practical salinity, and depth from measured channels. We suggest deriving sea pressure first, especially when an atmospheric pressure other than the nominal value of 10.1325 dbar is used, because deriving salinity and depth requires sea pressure.

```
rsk = RSKderiveseapressure(rsk);
rsk = RSKderivesalinity(rsk);
rsk = RSKderivedepth(rsk);
```

Bin data

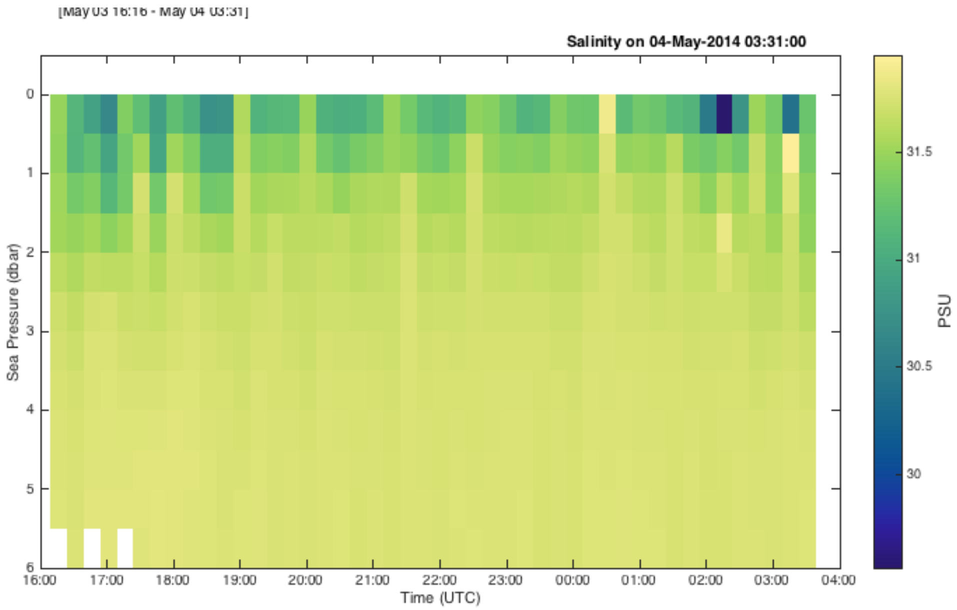
Average the data into 0.5 dbar bins using `RSKbinaverage`.

```
rsk = RSKbinaverage(rsk, 'binBy', 'Sea Pressure', 'binSize',
 0.5, 'direction', 'up');
```

Plot

Now we can see the changes to the data. We suggest plotting as you go to see if the changes being applied are what you expect.

```
RSKplot2D(rsk, 'Salinity');
```



See RSKtools_vignette

A vignette is available for information on getting started with `RSKtools` standard functions.

About this document

This document was created using Matlab™ Markup Publishing. To publish it as an HTML page, run the command:

```
publish('RSKtools_vignette.m');
```

See `help publish` for more document export options.

Published with MATLAB® R2015b