$CTDO_2$ Calibration Toolbox

Derrick Snowden

Carlos Fonseca

This document describes a `Matlab`© toolbox which allows $CTDO_2$ data collected with a Sea-Bird© sensor suite to be calibrated against independent analyses of seawater samples.

Table of Contents

Introduction

Provide a good overview and anything else you may think is needed in this area. Looking at other HOWTOs may assist in providing ideas.

Credits / Contributors

Most of the software described in this document was written by Carlos Fonseca. Derrick Snowden has contributed to the software development to a lesser extent. Scientific input to the software development was given by Molly Baringer, Libby Johns, Bill Johns and Bob Molinari. Many of the ideas and algorithms are taken from other packages in use by PMEL. In particular, Greg Johnson and Kristy McTaggert were the primary authors.

Feedback

Feedback is most certainly welcome for this document and for the software it describes. Send your additions, comments and criticisms to the following email address : <`Carlos.Fonseca at noaa.gov`>, or <`Derrick.Snowden at noaa.gov`>.

Obtaining and Installing the CTDCAL Toolbox

The CTDCAL Toolbox is intended to be used at sea, to calibrate CTD profiles as they are collected. It is platform independent and has been tested on Linux and MS Windows PCs. Prior to the cruise, download the CTDCAL Toolboxfrom the AOML intranet home page (http://doris/matlab/ctdcal/). The file `ctdcal.tar.gz` contains both the programs and the documentation.

The `Matlab`© Environment

Though the CTDCAL Toolbox, is platform independent, the following examples will assume the operating system is Linux (or some other Unix variant). As time allows, the documentation will be updated to include analogous examples on MS Windows computers.

There are several default directories which are automatically visible to the `Matlab`© interpreter. Upon startup, the `Matlab`© interpreter searches the current user's home directory, HOME, (e.g.`/home/user/`) for a `matlab/` subdirectory which is automatically appended to the `Matlab`© path. In this document, MATALABPATH, is used to specify the `Matlab`© search path.

> **NOTE: Environment Variables:** The enviroment variable notation (e.g. MATALABPATH) does not imply a strict association between the `Matlab`© search path and the Unix environment path.

Installation

To install the CTDCAL Toolbox, simply unzip the `ctdcal.tar.gz` file somewhere in the MATLABPATH.

Example 1. Installing the CTDCAL Toolbox on a linux/unix computer

```
cp ctdcal.tar.gz ~/matlab/

cd ~/matlab/

tar xvfz ctdcal.tar.gz
```

There should be a `ctdcal/` directory beneath the `HOME/matlab/` directory, containing the program files and documentation for the CTDCAL Toolbox. This directory will be referred to as CTDCALDIR for the remainder of this document. In order to add the CTDCAL Toolbox to the default `Matlab`© path, add the following line to the user's `startup.m` file which usually exists in `HOME/matlab/`. The analogous directory on a Windows PC is `C:\MATLAB6p5\toolbox\local\ctdcal`.

```
path(fullfile('home','user','matlab','ctdcal'),path) % ctdcal toolbox
```

> **NOTE:** Notice that the path in the above example matches the `HOME/matlab/ctdcal/` directory described in the installation section. This is logical and convenient but not required. The exact form of this statement will be specific to the user's computer.

To test the installation, start the `Matlab`© interpreter and type

```
help ctdcal
```

A long description of the m-files contained in the toolbox should scroll across the screen. If the `Matlab`© returns the message below, see the troubleshooting tips in section XX.

```
help ctdcal
ctdcal.m not found.
```

Getting Help

The documentation for the CTDCAL Toolbox is contained in `CTDCALDIR/doc/`. There is a postscript version of this User's Guide at `CTDCALDIR/doc/HOWTO-ctd-calibration.ps` and an html version beginning at `CTDCALDIR/doc/howto/index.html`. Finally, the demo and example files, referenced throughout this document can be found in `CTDCALDIR/demo/`.

From within `Matlab`©, the help documentation for each function in the CTDCAL Toolbox can be accessed using the standard help syntax described above. In addition to the online text based help for each individual function, an html version has been created and is located at `CTDCALDIR/doc/ctdcal.html`. In addition to the standard help the html presentation shows the interdependencies of the `Matlab`© functions and the code itself.

Preparing to Calibrate a New Cruise

Cruise Data Directory Structure

Before calibrating a cruise, the $CTDO_2$ data files must be placed in a cruise calibration directory which has a predefined layout. There are `Matlab`© functions within CTDCAL Toolbox which will help in creating the correct directory structure. The root of the cruise directory CRUISEDIR from now on, can be anywhere. For this example, let CRUISEDIR be `/data/cruises/stc02/`, since the examples in this document are taken from the Subtropical Cells cruise in 2002.

> **Warning:**
>
> We strongly discourage using the CTDCALDIR as the CRUISEDIR. If they are chosen to be the same location, then each subsequent cruise will require a new installation of CTDCAL Toolbox and there is an increased risk of path conflicts with multiple versions of the same software.

From within the `Matlab`©. interpreter, follow the steps outlined in example Example 2 to create the cruise calibration directory tree described above.

Example 2. Creating the cruise calibration data directory tree using `Matlab`© CTDCAL Toolbox functions.

```
cd /data/cruises/
cruise_dir = make_cruise_dir('.','stc02')
ls stc02


ans =

 bottle/
 calibrated/
 mfiles/
 figures/
 log/
 oxy/
 proc_data/
 raw_data/
 salts/
 tmp/
 register_cruise.m
 ctdcalpath.m
```

This directory tree is a duplicate of the standard cruise directory set up on the PhOD ctd_proc and ctd_acq computers.

Cruise Metadata Management

There is a large amount of metadata associated with acquiring and calibrating $CTDO_2$ data on a research cruise. In the CTDCAL Toolbox the metadata of interest describes the locations on the disk of the various data files described in Appendix A as well as the information describing the space-time domain of the research cruise. The function `register_cruise.m` is the means by which this metadata is made available to the CTDCAL Toolbox.

Once a new cruise calibration directory tree has been created as described in the Section called *Cruise Data Directory Structure* a new `register_cruise.m` will be created in `CRUISEDIR/mfiles`. This file should be edited by the user to contain the metadata relevant to the current cruise calibration project. The important elements of this file are described below.

```
% The cruiseid  is the case sensitive character mneumonic which
% will be used to reference the current cruise calibration project.
cruiseid = 'stc02';

% How are the Seabird ctd files named? This is also case sensitive on linux computers.
```

```
% We recommend that you use lower case for everything to minimize the chance for conflicts.
% e.g. stc2001.cnv, stc2001.hdr, stc2001.btl etc.  Notice that the filestub variable
% below doesn't have to be the same as the cruiseid above.
file_name_prefix = 'stc2';

% The number of digits used in creating the file names from the file_name_prefix and
% a variable station number.
% TODO: This does not allow for multiple casts at the same station as per WOCE specs.
nDigits = 3;

% Are the extensions to the Seabird Data files written in upper case or lower case?
% e.g. stc2001.CNV or stc2001.cnv.  This must be true for all or none of the relevant
% file types. (cnv, bl, btl)
sbe_is_upper_case_extensions = false;

% Where is the main cruise calibration working directory?
% This directory is the top level root directory
% which contains a copy of the cruise data, the calibration reports, figures, and all the
% user editable m-files copied from the ctdcal/demos directory.  See the User's Guide
% for more explanation.
calibration_dir = fullfile('doris','snowden','CTD','test','stc02');
% NOTE: pseudo platform independent representation of directories. On Windows you must add
% 'C:' or '\\computername' as the first argument to fullfile.  (fullfile('C:','doris',etc...))

% Where are the working calibration data directories?
% These should be the same as the cruise_dir above but
% you have the option of changing the data directory tree root if you don't want to copy
% the data files to a local directory after every station.
% The directory structure suggested below is the one created by create_cruise_dir.m  It is
% recommended but not required that you stick with this structure.
cal_data_dir = calibration_dir;
cal_autosal_dir = fullfile(cal_data_dir,'salts');
cal_oxy_dir = fullfile(cal_data_dir,'oxy');
cal_btl_dir = fullfile(cal_data_dir,'bottle');
cal_cnv_dir = fullfile(cal_data_dir,'raw_data');
cal_hdr_dir = fullfile(cal_data_dir,'raw_data');
cal_bl_dir = fullfile(cal_data_dir,'raw_data');

% Where are the source data directories?
% These are the directories containing the original data stored immediately
% after collection.  e.g. they might be on the ctd acquisition computer or on
% a network mounted mapped disk.  The directories listed below are only referenced
% if you choose to use the update_calibration_data.m function.  If you sync the data directories
% by hand or using sneaker net then the preferences below are not referenced by
% any other ctdcal toolbox function.
% NOTE: The sync/update functionality has not been tested on mapped driveson a Windows computer
source_data_dir = fullfile(filesep,'doris','snowden','CTD','Data','stc02');
source_autosal_dir = fullfile(source_data_dir,'salts');
source_oxy_dir = fullfile(source_data_dir,'oxy');
source_btl_dir = fullfile(source_data_dir,'bottle');
source_cnv_dir = fullfile(source_data_dir,'raw_data');
source_hdr_dir = fullfile(source_data_dir,'raw_data');
source_bl_dir = fullfile(source_data_dir,'raw_data');

% Geographic region of the cruise.  (For making maps)
% Defaults to 10 percent extension of the max/min
% of the station coordinates.
region = [-60 -30 -13 8]; %[west east south north]

% Temporal extent of the cruise.
begYr = 2002; begMnth = 2; begDay = 1;
endYr = 2002; endMnth = 2; endDay = 28;
cruise_dates = [datenum(begYr,begMnth,begDay)  datenum(endYr,endMnth,endDay)] ;

% Assign some general options which will dictate the behavior of the ctdcal
% functions throughout the calibration procedure
% NOTE: These have not been implemented consistently throughout the toolbox and are experimental.
```

```
% Verbose operation to print messages to the screen.
options.verbose = true;
% Keep a running log of all operations performed.
options.logfile = ”; %if isempty then no log is kept.
```

Once the `register_cruise.m` file is current, execute it at the `Matlab`© prompt.

```
register_cruise
```

> **Alternative methods to manage the `Matlab`© search path:** In addition to editing the global `startup.m` file to include the CTDCAL Toolbox path, the user has another option to include the necessary functions in the `Matlab`© path. The file `CRUISEDIR/ctdcalpath.m` should have been created at the same time as the `CRUISEDIR` by the `make_cruise_dir.m` function. This function, when executed from the command line, should place the relevant toolbox and cruise specific functions into the `Matlab`© search path. The user must issue the **`ctdcalpath`** command at the beginning of each new `Matlab`© session.

Loading the Calibration Database

Once the cruise directory is created and the cruise has been registered by typing the **`register_cruise`** command, the relevant Sea-Bird© files must be manually copied in the correct locations in the directory tree.

Provided the set up has been down correctly and all files have been copied into the cruise directory tree, loading the cruise database with the information in the Sea-Bird© data files and the analyses performed by the salinity and oxygen analysts is a simple procedure. The following commands will load the Sea-Bird© data files and the oxygen titration results into the cruise database.

Example 3. Using `load_bl`

```
bl = load_bl('stc02')
btl = load_btl('stc02')
cnv = load_cnv('stc02')
oxy = load_oxy('stc02')
```

Editing and Loading the Autosal Analysis Results

The autosal system on the Ron Brown is equipped with an automatic data logging system to record the results of the analysis. Each analysis session should begin and end with a standard water analysis so that instrument drift can be detected and removed. At the end of each session the logging system creates three files, a `*.hdr` file, a `*.dat` and a `*.raw` file. The hydrographic technician on the Ron Brown typically converts the `*.dat` into a Microsoft Excel `*.xls` file format prior to submitting the results to the scientific party. Examples of all four file formats are in the `CTDCAL/demos` directory (See `ww2_014sal.*`).

The CTDCAL Toolbox toolbox reads the `*.dat` and `*.raw` formats for loading the cruise database. However, the `*.dat` files must be hand edited to augment the information automatically logged by the autosal and to adhere to a more uniform format. The chief scientist or $CTDO_2$ technician should be responsible for editing and maintaining these files.

To edit the `*.dat` files first copy the originals to another directory (e.g. `CRUISEDIR/salts/originals` for archiving as the current copies will be changed. Next open the `*.dat` file in Excel. In the first column, the autosal logs the Sample ID. Typically, the Sample ID cell will contain the batch number of the standard water if the data represents a standard analysis and will either be empty, or contain the station number for acutal seawater samples. Edit this column by fensuring that no blank cells exist and all seawater samples are identified by a station number.

Next, add a new empty column AFTER the time column and before the remarks column. The column heading for this new column should be Rosette Position. Each cell should contain the position on the rosette carousel of the Niskin bottle which collected the current sample. This information is available on the sample log sheets filled out by the water samplers. This information is crucial for matching the water samples with the corresponding $CTDO_2$ information.

Finally, save the file with the same name and extension as a tab separated ascii file. An example of a correctly edited `*.dat` file is in the `CTDCAL/demos` directory (See `ww2_014saledited.dat`).

Loading the results from the autosal analysis is more complicated due to constantly changing file formats etc. The file `load_salts` provides a template for the procedures to load the autosal results. It must be edited however, due to the specific nature of each cruise. Copy the `load_salts.m` from the `CTDCALDIR CRUISEDIR/mfiles` directory and rename it to something unique, e.g. `load_salts_`*`cruiseid`*`.m`. Repeat the same procedure for `CTDCALDIR/read_autosal_dat.m`. Edit the two files according to the instructions in the comments within the code.

Once the editing is completed, the following commands should load the autosal analysis into the cruise database.

Example 4. Using `load_salts`

```
salts = load_salts_stc02('stc02')
```

Calibrating the Pressure Sensor

NA

Calibrating the Temperature Sensors

Typical the AOML package is equipped with Primary and Secondary suites of sensors for all variables except pressure.

Calibrating the Conductivity Sensors

Creating a Conductivity Master Bottle File

`make_cond_sumfile.m` Maybe needs to be renamed.

Fitting a Calibration Model

`do_cond_fit.m` Maybe needs to be renamed.

Errata

Temperature scales: Make sure all temperature measurements are recorded in IPT-68 as part of the aqcuisition configuration.

***.btl files:** Make sure the configuration is such that the Nbf field is written to the *.btl files during acquisition. The Nbf variable contains the number of bottles fired or the firing order .

The Depth field in the *.oxy files: Oxygen analyst should record the niskin bottle number (also known as the niskin bottle position or rosette position) in the depth field of the *.oxy files.

Troubleshooting

This area contains general troubleshooting information, if applicable. The question/answer format works best (like a FAQ).

1. Question One

Answer One

A. Sea-Bird© Proprietary file formats

The `proc_data/` and `bottle/` subdirectories contain the data files as they are recorded by the Sea-Bird© data acquisition software. A subset of the Sea-Bird© file formats which are relevant to the calibration procedures are described below. Note, this list is not an exhaustive list of all Sea-Bird© file formats.

`*.dat`

These files contain a platform dependent binary representation of the raw data in engineering units and is written during acquisition. The SBE© preprocessing routines convert the `*.dat` files into more usable `*.cnv` files, however some information is lost in the process. The `*.dat` files remain the only source of the full profile information and should be archived after the cruise. The `*.dat` files are not read by **ctdcal** however.

`*.cnv`

The `*.cnv` files are the main record of profile data sampled by the $CTDO_2$ package. They can be read using the **read_sbecnv.m** function.

`*.btl`

The btl files contain the data (pressure, temperature etc.) recorded by the $CTDO_2$ package when a Niskin Bottle has been tripped. This data is necessary to collocate $CTDO_2$ variables with the sample bottle analyses of salinity and oxygen. They can be read using the **read_sbebtl.m** function.

`*.bl`

These files record the exact scan numbers which bracket a bottle trip request. They are useful for detecting when a bottle trip command was sent by the user but the Niskin bottle may not have closed due to instrument malfunction. They are read using the **read_sbebl.m** function.

`*.hdr`

The heder files contain a description of the acquisition environment, the data collected, and a history of the pre-processing steps applied to the raw CTDO$_2$ data. If the `.hdr` file is N strs long, then the first N strs of the corresponding `.cnv` file are an exact duplicate of the `*.hdr` file.

`*.asc`

ASCII representation of the profile data in a columnar format. This format, by itself is useless, in that the columns are not labeled. If the specific variables in each column is known, the `*.asc` files are easily loaded into `Matlab` ©.

`*.con`

The configuration files allow the user to specify the specific attributes of the sensors on the CTD package (i.e. the model and serial number of the pressure sensor). They are important components of the acquisition process but, assuming they are created correctly, are not relevant to **ctdcal** and will not be discussed further.

Index

Notes

1. http://doris/matlab/ctdcal/