Ship Calibration

1° : initializing the analysis of a SRDL-CTD comparison

in matlab run ***sc_init_calibration*** for that :
- create folders
- copy data
- create a script template
- save directory in *environment.mat*

After this all procedure it's in ***sc_calibration_name_EXP***

2°: Load CTD-SRDL data

- put raw data in raw_data directory
- in variable *list_tag* list name of each file from CTD-SRDL, example :
list_tag=[11700 12246 12261 14408]'

- if you have files direct download from CTD-SRDL and identify each dive, ascent and descent phase and link with CTD profil number. At the end of this step you have a matrix with raw data CTD-SRDL with :
Tag : name of CTD-SRDL,
CTD_profil_number : which CTD profil number it is, we use this number to link with CTD-ship profil,
raw_Depth,
raw_Temperature,
raw_Salinity,
Ascent/Descent : 0 = descent and 1 = ascent

You can use function ***read_SRDLfile***. For that you need to specify version software for each tag in varible *version_software, example :*
version_software=[4 3 3 3]

version are :
1 old format CTD_GEN_07B
2 format for fluorometer loggers FTD_09A
3 new format bluetooth
4 format for fluorometer loggers FTD_07B
5 format for oxy loggers
6 format for new tags with pressure is now measured by a separate Keller sensor

Save in t**ag.mat**

3° : Load CTD-ship data

- put raw data in raw_data directory
- in variable *files* list name of each file profil from CTD-ship, example :
files={'jr16004_1.cnv','jr16004_2.cnv'}

- create a matrix with raw data from CTD-ship with :
CTD_profil_number : it's the same number we use in the matrix CTD-SRDL, link with CTD-SRDL profil,
raw_depth
raw_temperature
ram_salinity
Ascent/Descent : 0 = descent and 1 = ascent

You can use function ***read_SEABIRDfile***. Check format file to specify which column you need to keep. And you can specify this on variable *str_format,* example :
str_format='%f %f %*f %f %*f %*f %*f %*f %*f %*f %*f %*f %*f %*[^\n]';

Save in **ctd.mat**

4° : bin average every 1dbar for tag CTD-SRDL

To compare SRDL and CTD file you need the same scale.

- load **tag.mat**
- for each tag and each profil :
    . we keep onlythe ascent phase column 6 ==1,
    . Sort rows on depth column
    . we use gauss filter (function ***filtre_gauss***) on this data
- At the end your matrix is CTD-SRDL profil only for ascent and with bin every 1dbar :

Tag name
CTD_profil_number
Depth : every 1dbar
Temperature : average for each dbar
Salinity : average for each dbar

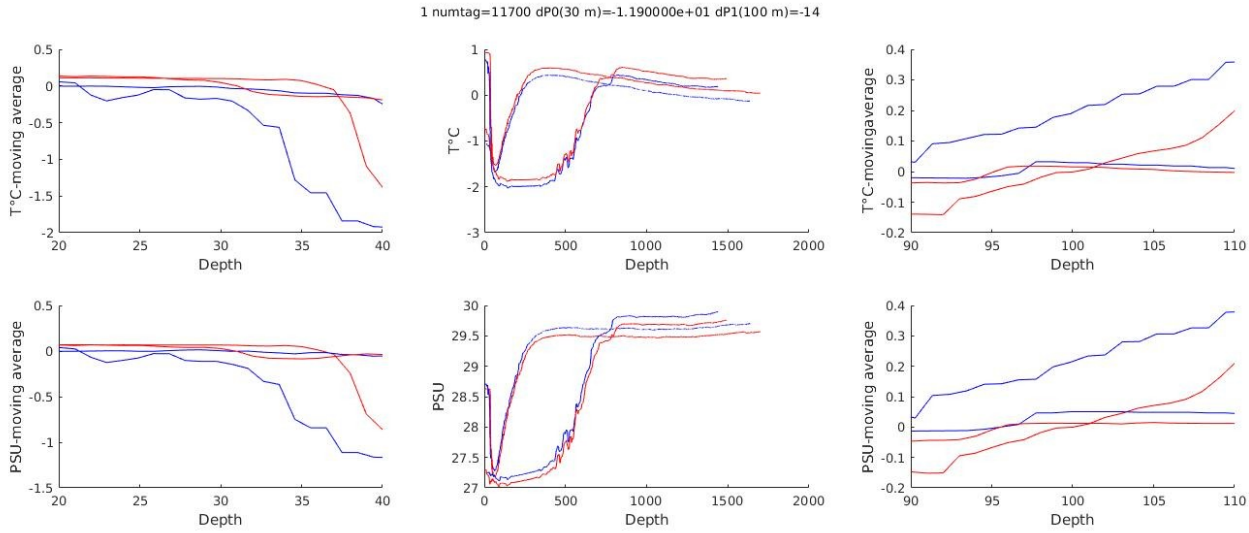
Save in **tag_pbin.mat**

5° ESTIMATE PRESSURE CORRECTION FOR SRDL DATA

- init matching matrix correction : create a matrix with number of reference depth point. Now it's 4 so you have 5 column : numtag dP30 dP100 dP250 dP500.
- corrected offset pressure
 use **matching_procedure** to match CTD-SRDL and CTD-ship depth at 4 reference depth postiion visually.

Default value are : 30, 100,250 and 500 meter.
Blue line it's CTD-SRDL profile and red line it's CTD-ship.
On the top it's temperature and depth and at the bottom salinity and depth.
You change value for each reference depth point to move CTD-SRDL profile and overlap with the red one.
Positive value it's move profile on the right and negative value on the left.

Example matching_procedure



Offset pressure save in matrix **match.**

- Use offset pressure to interpolate tag depth with fix point use like a reference in matching_procedure. With default value for example :

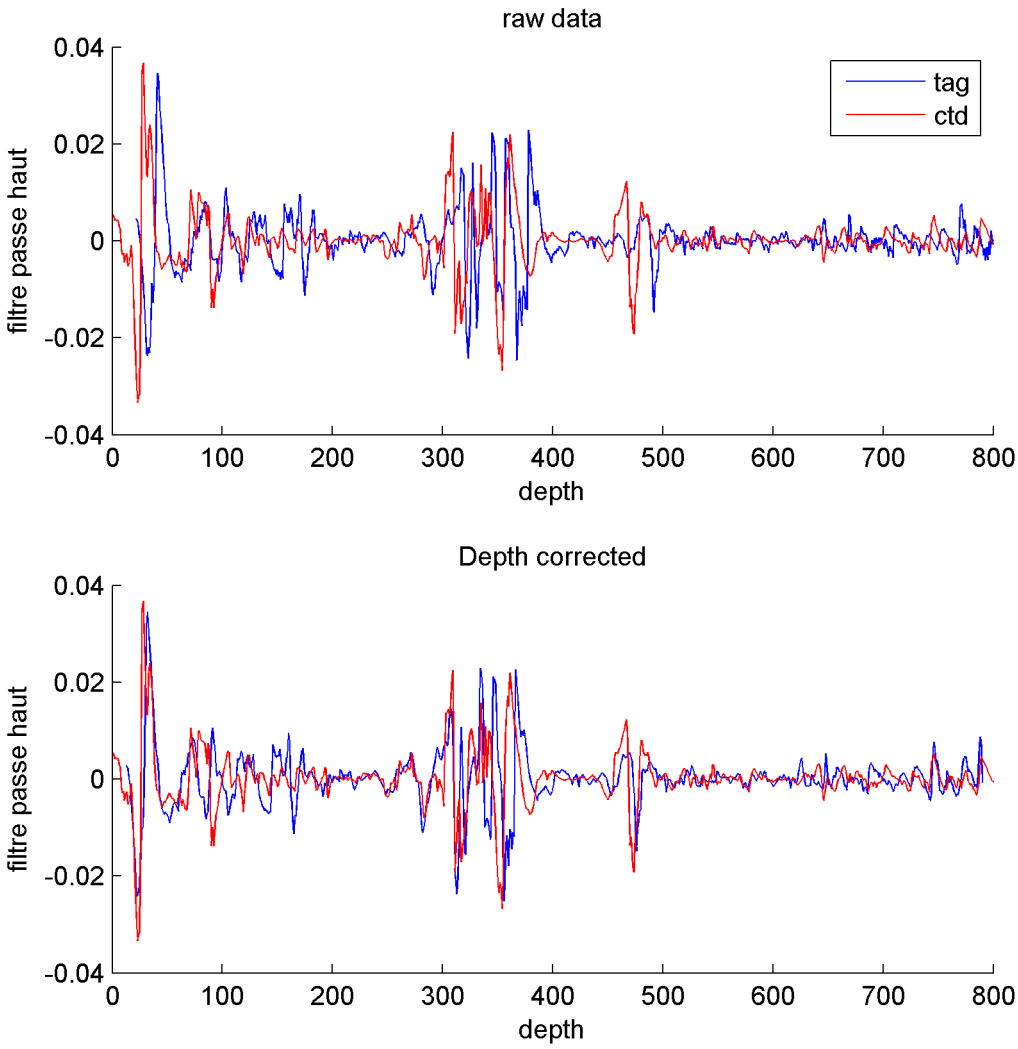
Ptag=interp1([0 100 250 500],[30+dP0 100+dP100 250+dP250 500+dP500],tag(Itag,3),'linear','extrap');

dP0, dP100, dP250 and dP500 are value from matrix **match** for each tag.

- Save new tags data interpolate in **cal_data.** In matrix :
Tag number
CTD_profil_number
Depth interpolate 1dbar
Temperature from Tag
Salinity from Tag
Temperature from CTD
Salinity from CTD

- Check if matching procedure is effective. Use High pass filter on your data (tag and CTD) to keep small variation and plot this to check if we find the same variation at the same moment. Use script **sc_filtre_passe_haut**. It's not a function, it's a script independant of this procedure, for now it's just to plot data filter and check matching procedure.Plot are save in folder '*calibration_plots/Filtre/*'

Example use High pass filter to check matching procedure



6° Linear fit Correction

`-init linear matrix` **linearfit** matrix = numtag aT bT aS bS
numtag : Tag number
aT : slope temperature
bT : intersept temperature
aS : slope salinity
bS : intersept salinity

- use matrix **cal_data**
- for each ctd profile of each tag :

. calculate difference in temperature and salinity between tag and ctd : Ttag-Tctd
. edges profil on depth with step of 10m :

```
Pbin=min(Depth):20:max(Depth);
edges=[0;Pbin(1:end)]+10;
```

. for each edges on temperature and salinity calculate an average of this difference, if average is superior at 0.01 we put NaN, so you have Dt and Ds at edge for all profiles of each tag.

. calculate a linear regression between depth and Dt or Ds with. Function *polyfit* on Matlab.

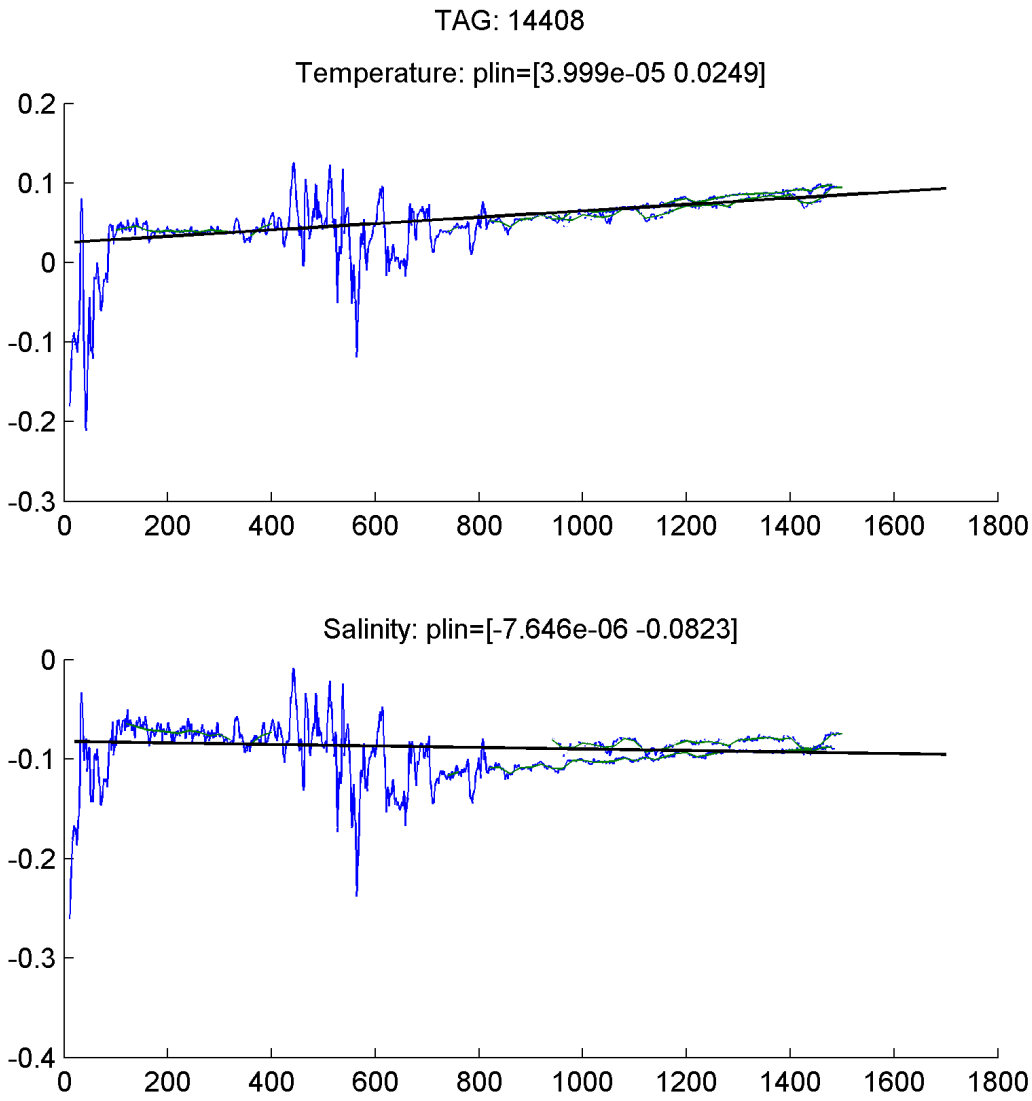
```
polyfit(Pbin,Dt)
```

. save coefficient in matrix **linearfit**

. you can calculate adjusted data with :

Tadj = Traw – (Depth * aT + bT)
Sadj = Sraw – (Depth * aS + bS)

Example linear regression


TAG: 14408
Temperature: plin=[3.999e-05 0.0249]
0.2
0.1
0
-0.1
-0.2
-0.3
0
200
400
600
800
1000
1200
1400
1600
1800
Salinity: plin=[-7.646e-06 -0.0823]
0
-0.1
-0.2
-0.3
-0.4
0
200
400
600
800
1000
1200
1400
1600
1800