AcoustYX toolbox installation

Document version 1.02 - 2020/02/25

This toolbox for GNSS/Acoustics inversion is provided under the GNU General Public License.

To cite this work:
Sakic, P.; Ballu, V.; Royer, J.-Y. A Multi-Observation Least-Squares Inversion for GNSS-Acoustic Seafloor Positioning. Remote Sens. ***2020****, 12, 448.* *https://doi.org/10.3390/rs12030448*

Step 1: Clone the repository

Use the git command `clone` in the local folder where you want to install the toolboxes.

NB : The geodezyx_toolbox (general-purpose functions) is also needed
https://github.com/GeodeZYX/GeodeZYX-Toolbox_v4

Sakic, Pierre; Mansur, Gustavo; Kitpracha, Chaiyaporn; Ballu, Valérie (2019):
The geodeZYX toolbox: a versatile Python 3 toolbox for geodetic-oriented purposes.
V. 4.0. GFZ Data Services. *http://doi.org/10.5880/GFZ.1.1.2019.002*

In a Terminal :
```
cd /your/local/directory/where/the/toolboxes/will/be/installed

git clone git@github.com:PierreS-alpha/acoustyx_toolbox_2_py3.git
```
and
```
git clone git@github.com:GeodeZYX/GeodeZYX-Toolbox_v4.git
```

Step 2: Install the toolbox in the PYTHONPATH

Option 1 : edit the .bashrc

e.g. if the toolboxes are in the folder
`/home/psakicki/THESE/CODES/CodePython/`

add the following line in the .bashrc file (located in your home directory)

```
export
PYTHONPATH=$PYTHONPATH:/home/psakicki/THESE/CODES/CodePython/acoustyx_toolbox_2/lib:/home/psakicki/THESE/CODES/CodePython/geodezyx_toolbox/lib
```

Option 2 : Using SPYDER, use the software *ad hoc* PYTHONPATH

Menu Outils > Gestionnaire de PYTHONPATH
Add the `/acoustyx_toolbox_2/`**`lib`** and the `geodezyx_toolbox/`**`lib`** folders

NB : in both case, it is the `lib` subfolders which have to be added in path

How to run a GNSS/A inversion ?

Strictly speaking, the least square inversion is performed with the script *restitution_mk09.py*

It is located in :
`./acoustyx_toolbox_2/scripts/RESTITUTION/restitution_mk09.py`

NB : the current version (2020) is the ninth version ("mk09", Mark 9), but it will probably increase in the future.

The only thing you need to do with this script is to indicate the path/name of the configuration file. Configuration file philosophy is described below in the next section. To do so, modify the `configfile_path` variable (this variable is defined just after the libraries import).
Then just run the script :

- Directly with in a terminal, with the command line
  `python restitution_mk7.py`
- Or using SPYDER (or another Development Interface) with the ▶ button

Input Data

The data must be in a single folder. The folder name must be the experiment name, and datafiles must respect the naming convention explained below.

For example let's consider an experiment called *Detection15mai.*
Data for this example are stored in :
`./acoustyx_toolbox_2/exemple/EXPERIMENTS/Detection15mai`

.cfg-file : The configuration file

Mandatory

The configuration file contains all the experiment's parameters.
All the fields are described in the file.

The configuration file can be stored anywhere on the computer, and can have any name. It follows the Python configuration files conventions of the ConfigParser module.

Exemple configuration files are stored in the following folder :
`./config_files`

Indicate the required path in the config file

For instance, is the experiment folder is here :

`/home/user/some/sub/directories/Detection15mai`

The correct paths in the configuration file shall be :
gene_path = '/home/user/some/sub/directories/'
exp = 'Detection15mai'

O-file : Observation file

Mandatory

filename convention : <experience name>.PXP<beacon ID>.O.dat
filename exemple : Detection15mai.PXP11.O.dat

This file contains the values of the two way travel time, the reception and emission epochs, the ship positions at emission and reception epochs.

There is one O-file per seafloor beacon. ID of the beacon is described in the file name.

About the ship coordinates, it must be in a local reference frame. We recommend the transfer formulas described here :
http://www.navipedia.net/index.php/Transformations_between_ECEF_and_ENU_coordinates
We recommend to use roughly the center of the working zone as the reference point (e.g. the median point of the navigation data)
Local coordinates **must be in this order** : East , North , Down (nadir oriented vertical component = - Up coordinate)

The approximate coordinate of the beacon **must** be in the header like this :
pxp_coords : [-58.98798306 -802.69686578 1261.65765152]

NB: data are read in a spaces separated fields philosophy, **not** a fixed width philosophy.

Columns are arranged like this :

1. Epoch of emission (seconds) : we suggest to use POSIX time (seconds since 1970/01/01), but it is not mandatory, any time reference can be used
2. East coordinate of the acoustic head phase center at the emission epoch (meters)
3. North coordinate of the acoustic head phase center at the emission epoch (meters)
4. Down coordinate of the acoustic head phase center at the emission epoch (meters)
5. Year of emission epoch
6. Month of emission epoch
7. Day of emission epoch
8. Hour of emission epoch
9. Minute of emission epoch
10. Second of emission epoch
11. Microseconds of emission epoch
12. Epoch of reception (seconds)
13. East coordinate of the ship at the reception epoch (meters)
14. North coordinate of the ship at the reception epoch (meters)
15. Down coordinate of the ship at the reception epoch (meters)
16. Year of reception epoch
17. Month of reception epoch
18. Day of reception epoch
19. Hour of reception epoch
20. Minute of reception epoch
21. Second of reception epoch
22. Microseconds of reception epoch
23. Raw two way travel time (seconds)
24. TurnAround Time (TAT, beacon intern delay) (seconds)
25. Two way travel time corrected from TAT (seconds). This is the needed input data for the inversion.

P-file : Observation file for simulation case.

Mandatory, replacing the O-file in simulation case

filename convention : <experience name>.PXP<beacon ID>.P.dat
filename exemple : Detection15mai.PXP11.P.dat

improved version of the O-file for simulations. This kind of file format contains supplementary informations about noise applied to the data, and columns are described with a header.

Columns fields are :
Perturbed data. Noise has been added to the data prior at the generation of this file. Those “noise” *data will be used in the inversion.*
E_emi_noise : Epoch of emission, perturbation included (seconds)
X_emi_noise : East coordinate of the ship at the emission epoch, perturbation included (meters)
Y_emi_noise : North coordinate of the ship at the emission epoch, perturbation included (meters)
Z_emi_noise : Down coordinate of the ship at the emission epoch, perturbation included (meters)
T_emi_noise : Propagation time ship => beacon, perturbation included (second)
E_rec_noise
X_rec_noise : East coordinate of the ship at the reception epoch (meters)
Y_rec_noise : North coordinate of the ship at the reception epoch (meters)
Z_rec_noise : Down coordinate of the ship at the reception epoch (meters)
T_rec_noise : Propagation time beacon => ship (second)
TAT : TurnAround Time
TWTT_noise : Two way travel time
True (non noised) data for information purposes
E_emi_clean
X_emi_clean

Y_emi_clean
Z_emi_clean
T_emi_clean
E_rec_clean
X_rec_clean
Y_rec_clean
Z_rec_clean
T_rec_clean
TWTT_clean
Noise_emi : Noise added on the ping ship => beacon
Noise_rec : Noise added on the ping beacon => ping

B-file : Baseline file

facultative

filename convention : <experience name>.B.dat
filename exemple : Detection15mai.B.dat
ATTENTION: the extension has to be **.dat**
Contains distances between the beacons.
Is a symmetric matrix, with size corresponding to the number of beacons, where the ij-th element correspond to the distance between the beacon i and the beacon j.

with_bl option in the configuration file must be activated to take it into account.

C-file : Sound speed profile, celerity

Mandatory

filename convention : <experience name>.C.dat
filename exemple : Detection15mai.C.dat

File containing for each line the sound speed value (m/s) for the corresponding depth (m) in the Z-file.

Z-file : Sound speed profile, depth

Mandatory

filename convention : <experience name>.Z.dat
filename exemple : Detection15mai.Z.dat

File containing in each line the depth (m, nadir oriented) for the corresponding sound speed value in the C-file.

What is the best configuration ?

It depends on your data.
Using baseline (B-file) and the option with_bl if possible to constrain the problem

with_barycenter, you can determine directly in the Least Square Inversion the coordinates of the barycenter of the array, along with the associated sigmas. It is a recommended option.

with_cleaning remove the pings considered as outliers, so it is a useful option

with_v_4_p_reinject use residuals of the previous iteration as weights for the new one. It normalize the residuals but gives unrealistic sigmas

with_time_window allows to select specific data in time windows.

with_munk_ssp allows to use a Generik Munk Profile, may be useful if the measured SSP is unavailable or badly defined.

You can change the number of iterations for the least square (iitermax) and the number of CPU cores (nbproc)

Output Data

Output data are stored in the experiment folder, in a subfolder named like this :
<Experiment name>_<Timestamp>

- The logfile (.log) : contents the results of the least square inversion
- The exp file (.exp) : a dictionary saved with the pickle module, containing the results, so it can be easily exploited
- The sum file (.sum) : is a summary of all the results present in the experiment directory
- Graphs in .pdf and .png : Histograms of the residuals (normalized or not), residuals depending on time, and the error ellipses (experimental).
- The V files (.V & .smartV) : Residuals. smartV must be prefered : contains Beacon IDs, POSIX epoch and residuals

The log file

in the log file you may find the results of the Least square inversion. It is separated in several blocks.

The block START contains a summary of all the parameters and options used for this inversion.

For each iteration of the inversion, results of the inversion are printed

Useful results are :

- `new coords.` : the coordinates of each beacon
- `raw barycentrer : Sum Xpxp / Npxp` : the coordinates of the barycenter (shall be equal to `barycenter estiimated in the LSQ`)
- `sigmas` : the formal standard deviations of the coordinates

- sum dX and sum abs dX : It is the sum (in absolute and not in absolute) of the coordinates correction. It shows the stability of the inverted coordinates. The smaller this value is, the closer to the final solution the model is.

The log file ends with a block END, a duplicate of the START block

How to generate GNSS/A simulated data ?

For this purpose, use the scripts `fabrik_7_fct_mk7.py` & `fabrik_batch_1_mk7.script.py`

it is recommended to use `fabrik_batch_1_mk7.script.py` for data generation.

The user needs to specify the variable `plateform_toolbox_path`, i.e. the path path of the toolbox e.g.

`/home/psakicki/Documents/CODES/acoustyx_toolbox_2/`

`accordingly to the computer on which the script is run`

Explanations are in the scripts commentaries.

les Kfiles sont des doubles vecteurs profondeur/gradients, extrait directement et sont caduques car perturbent trop fortement le ray tracing

les Udic contiennent le matériel pour constituer un SSF, et sont les remplaçants du Kdic
Ils sont munkisés

les Gdic sont les grilles d'interpolations pour se substituer aux RT eikonal long et fastidieux, ce sont eux qui in fine permettent le ray tracing

ToDo

- Constraints on apriori coordinates
- Absolute depth as observable