



GROOM

Gliders for Research, Ocean Observation and Management
FP7-Infra-2011-2.1.1 "Design Studies"

Deliverable D5.6

Prototype glider mission planning system

Due date of deliverable: 30/03/2014

Actual submission date: 21/02/2014

Partner responsible: CMRE

Classification: PU

Grant Agreement Number: 284321

Contract Start Date: October 1st, 2011

Duration: 36 Months

Project Coordinator: UPMC

Partners: UPMC, OC-UCY, IFM-GEOMAR, HZG, AWI, UT, FMI, CNRS, IFREMER, HCMR, CMRE, OGS, UIB, NERSC, CSIC, PLOCAN, SAMS, UEA, NERC.

Project website address <http://www.groom-fp7.eu>

Table of contents

Introduction	2
Software application objective	2
Software requirements	2
Matlab®	2
MOTU client (Python)	3
NetCDF toolboxes (Java)	3
Installation and setup	3
Configuration files	3
Sources	4
Regions	5
Fleet Missions	6
Tool Usage	8
Troubleshooting	8
Data downloading failure	9

Introduction

This document describes the usage of a glider fleet mission planner software tool. In this report, the user of this tool can find instructions regarding the goal of the tool, its installation procedure and how to use it.

As it is a draft version, the document is not yet intended to be complete and detailed. Users' comments, corrections and/or suggestions will be appreciated.

Software application objective

In the field of oceanography, it is a known fact that sampling the ocean is a challenging task. Nowadays, with the advent of new technologies, autonomous observing platforms are being developed and used to that end. However, the environmental conditions of the ocean can jeopardize the operation of such observing platforms. Therefore, it would be advisable to have a decision support tool to assess the feasibility of a predefined mission before actually performing it. The software described in this document is a glider fleet mission planner tool, developed at CMRE as a deliverable of the EU GROOM project.

Software requirements

MATLAB®

D5.5

The software has been mainly developed as a set of scripts and functions in Matlab. It has been developed and tested over the release R2013b, 64 bits, with no additional toolboxes (others than the free ones on the Internet, which are also provided). It is a pending task to assess the minimum version required to run this software.

MOTU CLIENT (PYTHON)

The MOTU client is a tool developed in Python within the framework of the EU MyOcean Project. Motu is a high efficient and robust Web Server which fills the gap between heterogeneous Data Providers to End Users. Motu handles, extracts and transforms oceanographic huge volumes of data without performance collapse. This client enables to extract and download data through a python command line. It has been tested with Python 2.7.3, 64 bits. The tested MOTU client tool is also released with this software. For more instructions on how to use MOTU, please refer to this project website (<http://cls-motu.sourceforge.net/>) or MyOcean catalog documentation.

NETCDF TOOLBOXES (JAVA)

There are two toolboxes used in this software tool that require Java VM to be running as well. The mexcdf tools are used as the main interface to access netcdf files. In addition, the ncToolbox developed at MBARI is a very convenient tool to read netcdf files compliant with CF (climate and forecast) conventions. Based on the constraints established by those conventions, the toolbox is able not only to read the variables from the files, but also to provide the coordinates that georeference the scientific variables in the appropriate way to be later used by the application.

Installation and setup

Product installation is performed by uncompressing the archive in any location on the file system. It is important to keep the relative paths of the contained subfolders. The main script of the tool checks its own path at the beginning of the run, and configures the rest of the required paths prepending the base one. Therefore, as long as the tool is moved "as a whole" (without changing the subfolders structure), it should not matter the absolute path where the tool is installed. The tool uses Matlab interface to build absolute paths or look for filenames in order to work on different OS (MS Windows or GNU/Linux) with no modifications, although it is pending to be tested as well.

The archive can be downloaded from the GROOM website at the following URL
http://www.groom-fp7.eu/lib/exe/fetch.php?media=private:deliverables:groom_d5.6_cmre.zip

CONFIGURATION FILES

The tool uses several configuration files to specify where to get the environmental information from, which region to work on and the available fleet to plan the mission for. In this way, the users can easily change the source for their data or modify the fleet to be optimized. They can even generate those files from another application easily.

The chosen configuration files format is JSON. It has been selected because it is a common standard nowadays and has several interesting properties:

- It is spread. Therefore, there are many available tools to interface it.

D5.5

- It can be parsed automatically, thanks to those tools.
- It is human readable, so one can also modify it manually without problems.
- It is compact, so the format does not produce a significant overhead in files size or transmission bandwidth.

Sources

The developed tool has been designed to be able to deal with several available datasets from MyOcean catalog. The source configuration file specifies where to get the data from. In this file, the user specifies the username and password of his/her account information to access the MyOcean downloading service. This information can be directly input in the file or contained into another different file and referenced here. This configuration option helps sharing the same user account among different MyOcean sources.

The servers addresses are also specified here, both the MOTU gateway and the service where the request is sent. Finally, the products with their respective variables of interest are listed. An example is provided below these lines.

D5.5

```
{
  "objectType" : "sourceDefinition",
  "userInfo" : {
    "objectType": "userAccountRef",
    "relatedFile": "myOceanAccounts/defaultUser.json"
  },
  "serverInfo" : {
    "gateway" : "http://gnoodap.bo.ingv.it/mis-gateway-
servlet/Motu",
    "servideId" :
"http://purl.org/myocean/ontology/service/database#MEDSEA_ANALYSIS_FORECAST_PHYS
_006_001_a-TDS"
  },
  "productsInfo" : [
    {
      "name" : "myov03-med-ingv-cur-an-fc",
      "vars" : ["vomecrty", "vozocrtx"]
    },
    {
      "name" : "myov03-med-ingv-tem-an-fc",
      "vars" : ["votemper"]
    },
    {
      "name" : "myov03-med-ingv-sal-an-fc",
      "vars" : ["vosaline"]
    }
  ]
}
```

```
{
  "objectType": "userAccount",
  "username" : "<username>",
  "password" : "<password>"
}
```

Regions

While the source configuration file specifies the dataset to access from the catalog, the region configuration file specifies an optional subset to apply over the source domain. This is done since it is common that the source dataset covers a larger area than the one required to planning a glider fleet mission. As an example, there is an available product for the whole Mediterranean, while the gliders' endurance allows for now to be deployed and operate in a smaller part of it. In order to optimize the resources (network bandwidth when downloading, storage/archival space and also computation time),

D5.5

a spatial bounding box can be specified at this point to request only a portion of the original dataset. The JSON object contains a key called "roiId", and it stands for the region of interest name. This string of characters will be used as a prefix for the files that are downloaded. An example of a region file is provided below these lines.

```
{  
  "objectType" : "region",  
  "roiId"       : "LigurianSea",  
  "minLon"     : 8.0,  
  "maxLon"     : 10.5,  
  "minLat"     : 43.0,  
  "maxLat"     : 44.5,  
  "minDepth"   : 0.0,  
  "maxDepth"   : 6000.0  
}
```

Fleet Missions

The fleet mission file is actually a list of gliders, specifying the properties of each asset available in the fleet. The parameters required are:

- The asset deployment position. This point will be the start of its planned route. The coordinates order is first longitude, then latitude; both are in decimal degrees.
- The glider nominal speed with respect to water. It will be used to compute the glider speed over ground in addition to sea currents found in the environment it moves through.
- The glider maximum depth. It will be used to integrate the oceanographic fields from the 3D fields to a two-dimensional field.
- The glider density range. It is related to the target density the glider has been ballasted for. It will be used to check that the glider is able to navigate in the environment. An out-of-range value would mean that the glider cannot dive, or even worse, could not surface. There are two modalities to specify it:
 - Set the range, by specifying the minimum and maximum allowed densities.
 - Set the target or ballast density. The tool will estimate the range by using the glider properties to change its buoyancy.
- Timing parameters. For user easiness, it allows the time specification with the units (therefore, all in one string), instead of forcing them to seconds. The time on surface, time underwater, transect time and the maximum time allowed for the mission are required:
 - Time on surface: it is used to estimate the collision probability with the vessels traffic density at each surfacing.
 - Time underwater: it is used to simulate the period of time that the glider navigates with no GPS fixes (and so fixed sea currents estimation).
 - Transect time: it is used to simulate the time the glider heads to the same waypoint.
 - Maximum time: it is used to assess that the glider can complete the list of waypoints, given the sea currents present in the area, in a given time.

D5.5

```
{
  "objectType" : "fleetConfiguration",
  "gliderList" : [
    {
      "objectType"      : "gliderConfiguration",
      "gliderName"      : "Noa",
      "gliderModel"     : "Slocum",
      "gliderType"      : "deep",
      "gliderGeneration" : "G1",
      "gliderSpeed"     : 0.4,
      "maxDepth"        : 1000,
      "targetDensity"   : 1026.5,
      "timeOnSurface"   : "30 minutes",
      "timeBetweenSurfacings" : "4 hours",
      "timeBetweenWaypoints" : "12 hours",
      "maxMissionTime"  : "3 days",
      "deploymentPosition" : [9.5, 43.75]
    },
    {
      "objectType"      : "gliderConfiguration",
      "gliderName"      : "Elettra",
      "gliderModel"     : "Slocum",
      "gliderType"      : "shallow",
      "gliderGeneration" : "G1",
      "gliderSpeed"     : 0.36,
      "maxDepth"        : 180,
      "targetDensity"   : 1026.5,
      "timeOnSurface"   : "30 minutes",
      "timeBetweenSurfacings" : "4 hours",
      "timeBetweenWaypoints" : "12 hours",
      "maxMissionTime"  : "3 days",
      "deploymentPosition" : [9.5, 43.75]
    }
  ]
}
```

Tool Usage

The main function checks the number of parameters it has been called with. If no arguments were defined, the tool assumes the user wants to start it in interactive mode, and so opens GUI tools to choose the required configuration files. If, on the contrary, the tool is called passing the configuration files as parameters, it runs automatically with those files without requiring any interaction from the user.

The tool reads first the source and region configuration files. If the resulting product of their combination has not been downloaded previously, it uses the Motu client tool to access the MyOcean service and download the product. Then, the tool builds a list of maximum depths from the different gliders in the fleet mission file. After this, the tool reads the contents of the environment files, extracts the hydrodynamic properties, processes them according to the requirements and displays the gathered environmental information in a figure. The current fields are integrated up to different depths according to the gliders in the fleet, so each one can use its own "observed" current field while integrating its trajectory.

Presently, the tool is also using vessels traffic information derived from the data provided by the EU PASTA MARE project. The spatial coverage of these data is global. However, its resolution might not be high enough for regional applications (like the gliders' missions). In any case, it has been extremely useful as a proof of concept for the risk assessment component. The AIS based traffic density information is also used by the tool and can be displayed in the figure as well. A button has been added at the end of the toolbar to switch between the two scalar fields: hydrographic density and vessel traffic density, while vector field of sea currents is always present. On the left side of the toolbar, a dropdown menu allows the user to visualize the vector field of sea currents integrated from surface to different depths.

After this environmental information is presented, the tool generates an initial random mission. In order to do that, it generates a route for each element of the fleet (starting at its deployment location), and simulates this mission (to take into account the effect of the environment). With this initial mission, the tool undergoes an optimization process based on the simulated annealing algorithm, trying to maximize the amount of information collected by the whole fleet.

During that glider mission simulation it also computes several risk components. Density field is used to verify that the glider will be able to navigate into that environment. Ballasting information of the glider is required to check it. Traffic density is also used as a risk component to estimate the collision probability when gliders are at surface fixing position and transmitting data. Instrument failure probability, given the mission time, is also estimated. It is based on the study "Glidens Risk And

Reliability" carried out by Dr. David Smeed, Dr. Mario P. Brito and Prof. Gwyn Griffiths, presented in

the GROOM Working Meeting, in 16-17 January 2012.

The optimization algorithm is randomly choosing one glider track at a time, choosing one turn in that track and modifies it. As modifications are performed, the uncertainty reduction for each complete fleet mission is evaluated. The best fleet mission is kept during the optimization to be provided as a result.

At the end of the optimization process, it will show a summary report of the obtained results, and will give the user the possibility to save the resulting mission paths in a file.

Troubleshooting

In addition to unintended bugs in the code, there are sources of error external to the tool itself that can be checked and fixed by the user without modifying the tool. For any bug in the code, please contact

D5.5

the developers (Bartolome Garau, garau@cmre.nato.int, Alberto Alvarez, Alvarez@cmre.nato.int, Miguel Martinez, miguel.martinez@uib.es).

DATA DOWNLOADING FAILURE

Sometimes the data downloading can fail due to several reasons. A careful evaluation of the situation would proceed like this:

- Check the source and/or region configuration files and assess the parameters are correctly specified. This is to assert that we provide the right parameters to the tool.
- Check python installation and accessibility from the command line.
- Test the downloading command built on the wrapper by hand on a terminal window.
- Go through MyOcean catalog website and check the availability of the configured product interactively.