GRANT AGREEMENT N°223866

Deliverable	D08.03
Nature	Report
Dissemination	Public

D08.03 – Description of the multi vehicle simulator

Report Preparation Date	24/03/2011 Project month: 30
Authors	Jan Opderbecke, P06 – Ifremer Michel Perrier, P06 – Ifremer Jonathan Dumon, P01 – INRIA
Report Version	*V01*
Doc ID Code	Ifremer_D08.03_24032011_V01

Contract Start Date : 01/09/2008
Duration : 36 months
Project Coordinator : Carlos CANUDAS DE WIT, INRIA, France

Département des Systèmes sous-Marins
Service Positionnement, Robotique, Acoustique et Optique

J. OPDERBECKE

Le 25/03/11

IMN/SM/PRAO/11.091

-

Description of the multi vehicle simulator

Date : 25/03/11 Référence : IMN/SM/PRAO/11.091 N° Analytique : A120104H	Nombre pages : 36 Nombre figures : Nombre d'annexes : Nom du fichier : FeedNetBack_11_091_JOpd.doc Rédacteur : J. Opderbecke

Sujet/Titre :

-

Description of the multi vehicle simulator

Résumé :

Mots clés :

Révisions

Indice	Objet	Date	Rédigé par	Vérifié par	Approuvé par
A	Création du document	23/03/11	J. OPDERBECKE M. PERRIER J. DUMON - INRIA	J. OPDERBECKE	V. RIGAUD

SOMMAIRE

1. Ifremer - Various operational underwater vehicles : one strategy for simulation

Ifremer operates two types of underwater vehicles for which modeling and simulation is part of the development strategy: Victor (6000m depth Remotely Operated Vehicle) and AsterX or IdefX (3000m depth Autonomous Underwater Vehicles).

Beside the operation of these vehicles, Ifremer also performs development on each, for performance improvement and evolution. Vehicle control, acoustic communication, tele manipulation, are some of the vehicles key functions that clearly needs, for safe and efficient development purpose prior to real implementation and in-situ testing, to rely on software tools that models and simulates, as close to reality as possible, the concerned equipment, behavior or function.

In order to be consistent in its development procedure, and also to benefit from vehicle to another one, vehicle simulation software has been developed with the same strategy, objectives and tools. Portability, reusability are thus key objectives.

Modeling and simulation can concern only one specific equipment of a vehicle (for example: manipulator arm, thruster, propeller, cable or tether, acoustic modem,…) or the entire vehicle itself, and in particular the complete dynamic model of the vehicle, for control design and mission programming studies for instance.

Operating marine vehicles as fleet of individuals aiming and acting to achieve a common objective retains more and more the attention of marine vehicles operators. Several applications can be imagined (pollutant source seeking and tracking, bathymetry mapping of wide areas, animal habitat mapping,…) but common key functions are required in order to operate those vehicles together, safely and efficiently.

Developing a simulation platform allowing to simulate multiple vehicles within a common environment, having the capability to simulate the communication between them in function of the environment characteristics, to navigate in a common area as a group or a fleet, to sense the same environment in order to exchange, share or compare data, is clearly of high interest.

2. Mimosa mission programming tool

Mimosa is an integrated software tool designed to bring operation solutions to AUV mission management, in particular: mission preparation, planning and validation, real-time mission supervision, post-mission analysis and playback, as well as cruise and mission data management.

An extended version of Mimosa tool has been developed in order to handle multi-vehicles mission management. This version of the software allows to create fleets of marine vehicles, and to define mission profiles for these fleets, using the Mimosa graphical interface.

The Mimosa software has been also modified and adapted to be capable to automatically create mission files understandable by the Multi-Vehicle simulator described below. When dealing with fleets of marine vehicles, once the fleet mission scenario has been defined, Mimosa defines and produces the mission file for each vehicle that belongs to the fleet.

3. AsterX AUV Simulation platform

The AUV simulation platform consists in a complete set-up reproducing the hardware and software environment of the real AsterX AUV owned and operated by Ifremer (see figure).

Figure. The Aster^x AUV © Ifremer

The platform is built around three main computers having different roles :

- the **mission preparation and programming** computer allowing the definition of the mission to be achieved by the vehicle. This computer runs the Mimosa software as described before. Mimosa generates the mission file to be run on the AUV.
- the **surface controller** computer connected to the vehicle prior to the vehicle launch (vehicle on the deck) or when the vehicle is at the surface (communication using radio communication link)
- the **vehicle controller** computer in charge of executing the downloaded mission in an autonomous way (during the mission execution, no more communication with the surface controller is possible). On the vehicle controller, several software are running: vehicle control (with high- and low-level control of the mission and vehicle), fault management system.

At this stage, it is important to note that for the simulation platform, the surface and vehicle controller used, are strictly identical to the computers used on the real vehicle. This point ensures first the portability of any development made in the simulation platform to the real world. Second, the development in the platform can be made no matter which vehicle is controlled by the vehicle controller, the simulated vehicle or the real one.

In addition to the vehicle control, the AUV is equipped with scientific sensors, the vehicle payload, controlled by a dedicated computer (the payload computer). Information coming from the payload computer (for example concerning the quality of sensed data, the logged data status,…) can be exploited by the vehicle controller in order to react in some situations.

In the simulation platform, the vehicle behavior is produced by a numerical model calculating the vehicle state evolution in response to actuators and propulsion commands. The produced state is then exploited by a set of sensor models producing formatted data under real sensor specification; these sensor data are then used as feedback for the vehicle controller level in the same way than for the real vehicle.

Inputs of the model are the same than on the real vehicle: propeller and fins commands. Outputs are acceleration and velocities taken from the vehicle state evolution during time in response to commanded inputs, and are used and formatted by the sensors simulator in charge of producing the same data, with the same format, than on the real vehicle.

Figure X describes the architecture of the AUV simulator.

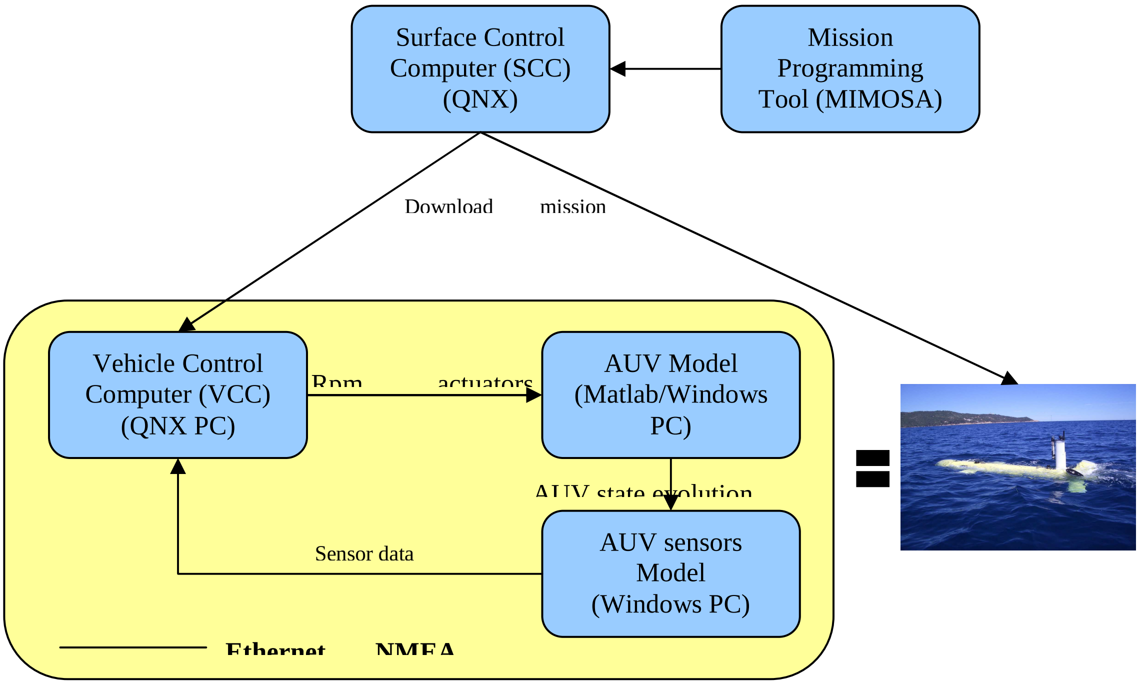

Figure . Architecture of the AUV simulation platform

4. Multi-Vehicles Simulator configuration

The Multi-Vehicle Simulator is manly used as a simulator engine. It allows simulating several vehicles having different dynamics, behavior, characteristics, sensing equipment and control modes.

The simulator also implements realistic environment models, aerial and underwater, with realistic physical characteristics. These simulated environments are used for modeling the displacement environment of the vehicles for instance when considering sea currents, for communication simulation (acoustic and radio signal propagation in particular) and in particular communication between vehicles, for the sensors models (sonar beams, salinity sensors, … are modeled and produce sensing data from the environment where the vehicles evolve).

Each marine vehicle, aerial or underwater, is modeled and simulated as an object within the distributed simulator. The vehicles own specific characteristics: dynamic model, low-level control and sensors. Vehicle models can be parameterized differently from one vehicle to another one, giving thus the capability to the user to create and manipulate homogeneous or heterogeneous fleets of marine vehicles.

Each simulated vehicle, aerial or underwater, can be controlled locally using embedded controllers pre-defined in the simulator that can be enriched or not with more complex control laws or behavior management. The simulator also proposes the possibility to control simulated vehicles by external controllers, linked to the Multi-Vehicle Simulator thanks to a dedicated communication library. For instance, a real vehicle controller can be connected to the simulator for instance for testing a development prior to real testing on a real vehicle. On-going control developments can also be tested by connecting development tool like Matlab to the Simulator.

Basically, each vehicle is closed-loop controlled by a controller, internal or external to the Simulator, which sends commands to the vehicles in response to mission controller and to vehicle states evolution.

Simple, or complex behavior models, can be used for each vehicle. Like for the vehicle controllers, simple dynamic models pre-exist in the Simulator. More complex and realistic models can be added within the Simulator, in order to get closer to real vehicle behavior.

As defined before, the mission programming tool, Mimosa, developed by Ifremer, can be used for defining scenarios and mission description, in a format understandable by the Multi-Vehicle Simulator.

A 3D Graphics User Interface is also used for real-time piloting of vehicles (for instance in the case of tele-operated vehicles) as well as for real-time mission and scenario visualization. It displays the vehicles, the environment, and the acoustic signals emitted by vehicles when they are fitted with communication modems. Other information can also be visualized, like sensors data, vehicle energy consumption,…

Figure X describes the architecture of the Multi-Vehicle Simulator as described below.

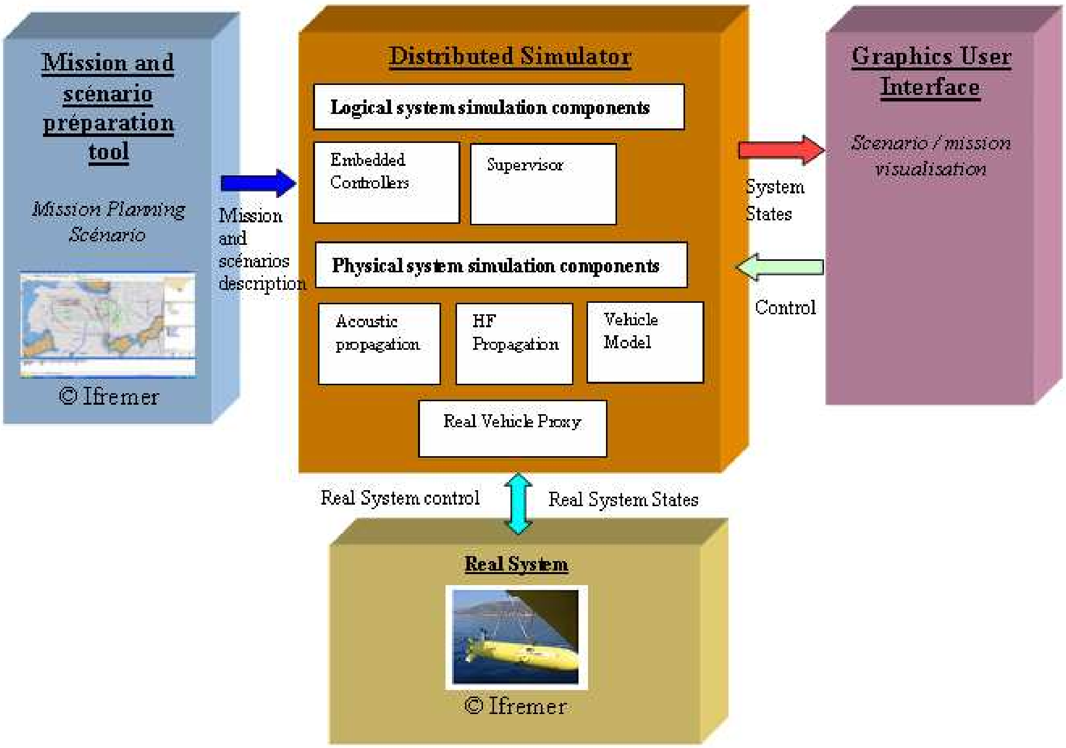

Figure . Distributed Multi-Vehicle Simulator and other software modules

5. Connection between AsterX Simulator and the Multi-Vehicle Simulator

A software module called ConnectAsterX has been developed in order to allow the connection between the two simulators: the AUV Simulator and the Multi-Vehicle Simulator. This software creates the adequate socket connections between the two simulators. In addition, it is in charge of the conversion of the exchanged frames between the simulators, using the Multi-vehicle Simulator communication library protocol.

Once the two simulators are connected, a fake vehicle that needs to be tagged *AsterX* is created within the Multi-vehicle Simulator; this tag will be used by the ConnectAsterX software interface to associate the fake vehicle in the Multi-Vehicle Simulator and the AsterX AUV simulated beside. The fake vehicle will then get all its behavior from the AsterX simulator. As a consequence, a fleet of vehicles can be modeled in the Multi-Vehicle Simulator with the AsterX AUV as one of the vehicles. It is then possible to manage a fleet of heterogeneous vehicles having different dynamics and different equipment, around the AsterX AUV.

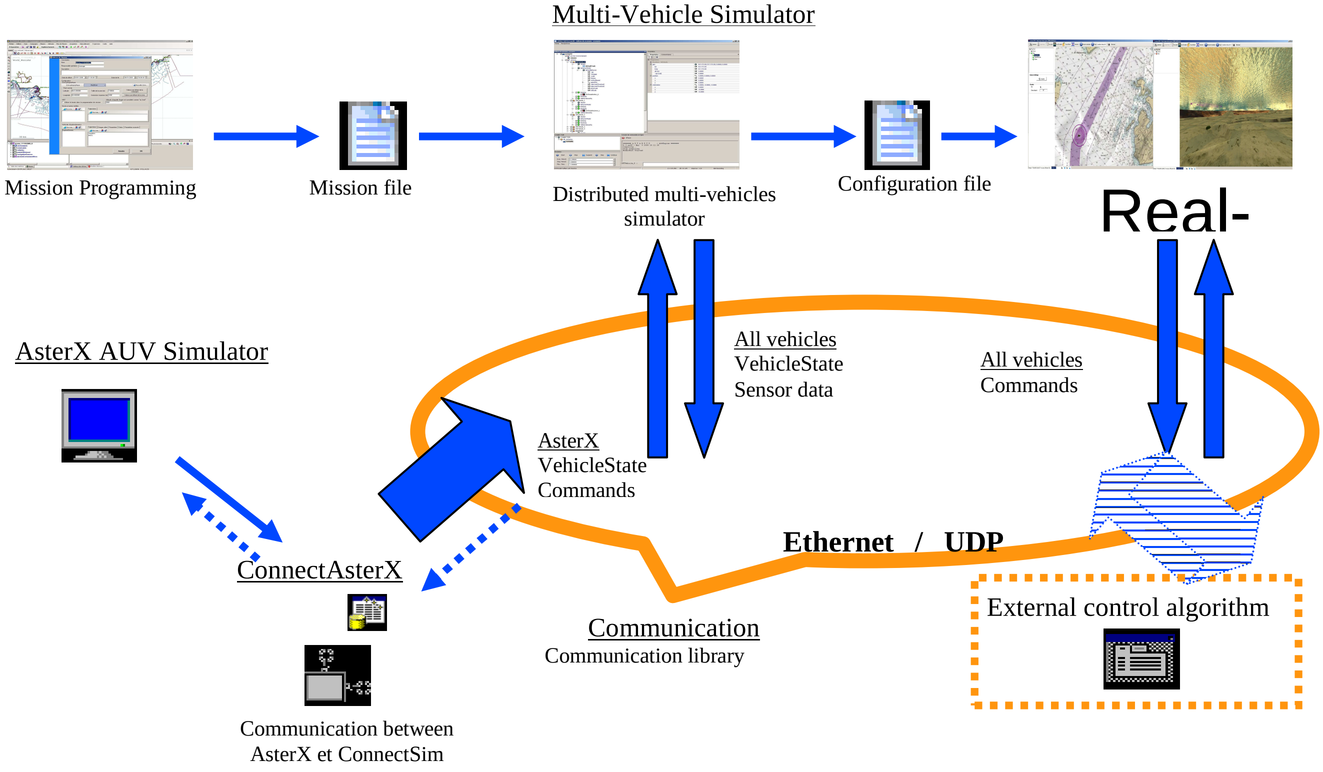

Figure . Software architecture of the simulation platform

6. Hardware in the loop : integration of hardware equipment in simulation

Knowing that the AsterX simulation platform can either control the AsterX AUV model or the real AsterX AUV (it just depends on what Vehicle Controller is connected to the Surface Controller of the platform), it is then possible to manipulate at the same time and in a common scenario, virtual (simulated) vehicles and real vehicles.

In order to go further in fleet management simulation, one could also imagine to define a fleet of simulated marine vehicles, one of them being the real AsterX AUV as defined above, and to manage the communication between the real AsterX AUV and the simulated vehicles thanks to a real acoustic modem fitted on AsterX. The other modem could be installed on the pier and physically connected to the Multi-Vehicle Simulator.

Then, and because communication management is performed within the Multi-Vehicle Simulator such that simulated vehicles can send and receive data through their virtual modems, one can envisage to make communicate for data exchange the simulated vehicles and the real AUV.

7. INRIA – Multi-vehicle simulators MUSim and MASim

Version	Date	Author	Comments
V0	06/05/2010	J. Dumon	Creation
V1	15/06/2010	J. Dumon	Retours C. Canudas, D. Simon, A. Kibangoo, A.Seuret, J. Opderbecke

Contents:

8. 1. Context of this document

This document is reviewing the existing Multi-vehicle simulator (MASim and MUSim) developed in Connect project (http://www.lag.ensieg.inpg.fr/connect/).
As the end of this project is planed for February 2011, we will expose what is done in May 2010 and what is planned to be done until the end. It also proposes a non exhaustive list of functionalities that could be developed if necessary.

9. 2. Connect Project

CONNECT is a project funded by the ANR (French National Research agency). The project deals with the problem of controlling multi-agent systems, i.e. systems composed of several sub-systems interconnected between them by a heterogeneous wireless communication network. In particular the project target the control of a cluster of agents composed of autonomous underwater vehicles and marine surface vessels.

This project started in May 2007 and will finish in February 2011.

Partners are NeCS INRIA-GIPSA-lab research team (project leader), Ifremer, Prolexia and Robosoft-PGES.

More information in the website http://www.lag.ensieg.inpg.fr/connect/

10. 3. MUSim and MASim architecture

MUSim and MASim have been jointly developed and can now be used either independently or in a complementary manner.

MUSim means Multi-vehicle Full Simulator and MASim means Multi-vehicle Fast Simulator. MASim can be seen as a light version of MUSim that offers less functionality. Scenarios and control algorithms can be tested with MASim and then be implanted in MUSim through a reduced work.

MASim is a simple Simulink model and can be implemented very easily from a PC to another. As its name suggests, its lightness also allows it to be faster than MUSim.

MUSim is mainly based on the Simulator ConnectSim communicating with a Human Machine interface ConnectIHM and Simulink model.
It also can be connected to the operational mission planning tool Mimosa.
Communication between theses applications is Ethernet UDP based.

MASim and the Simulink part of MUSim are developed by Gipsa-Lab, ConnectSim and Mimosa by Prolexia and ConnectIHM by Robosoft-PGES.
An interface between MUSim and a real AsterX has been developed by Ifremer.

As all Connect partners are French, ConnectSim and Connect IHM are in French.

The full architecture of MUSim can be seen in figure 1.

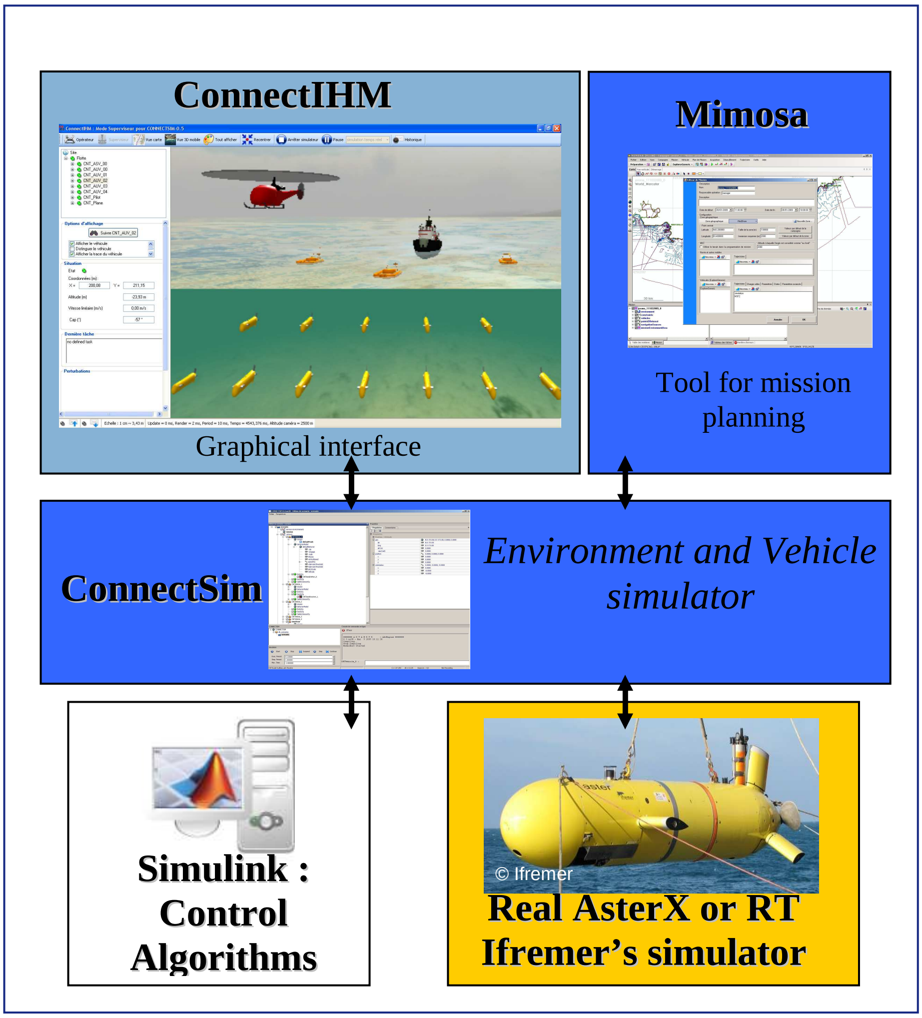

Figure 1: MUSim architecture

11. 4. Summary of functionalities

Table 1: Summary of functionalities

	Functionalities	MUSim	MASim
1	***System requirements***		
1.1	Matlab/simulink requirements (min R2007b)	yes	yes
1.2	Internet access requirements for installation	yes	no
1.3	Windows XP	yes	no
1.4	Good performance of the CPU (multicore with high frequency and 12MB cache), 1GB NVIDIA graphics are required	yes	no
1.5	Exploitation rights limitations	yes	no
2	***Time characteristics***		
2.1	using Sample time less than 50ms (exception of the dynamic model of AsterX)	no (Prolexia)	yes
2.2	Variable sample time of the dynamic model of AsterX	being validated	yes
2.3	Simulation in real time speed (5 AUV, cinematic model for AsterX) at typical sample time = 125ms	yes	yes
2.4	Simulates 10 times faster at ST=125ms	no (Prolexia)	yes
3	***Vehicles and fleet***		
3.1	Energy vehicles management	possible**	possible**
3.2	Computation vehicles management	no	no
3.3	Scalability: any number of AUV's, ASV's and AFV can be add in the scenario, one by one	yes	yes
3.4	Scalability: Copy-paste can be done to add new vehicle with all characteristics and equipment	no (Prolexia)	yes
3.5	Using dynamic model of AsterX for AUV's instead of cinematic one	being validated	yes
3.6	Positioning drift	possible*	possible*
3.7	Bathymetric sensor	being validated	possible*
3.8	Collision Management	no (Prolexia)	no
4	***Environment***		
4.1	3D map of the non flat seabed	yes	possible**
4.2	One current and one wind fixed and uniform	being validated	possible*
5	***General display***		
5.1	2D map	yes	yes
5.2	3D display mode	yes	possible*
5.3	Rich 3D natural environment and vehicles, including interface between non flat seabed, water and air	yes	possible**
6	***Source management***		
6.1	2D shape :sum of three exponentially decreasing concentric ellipses. 3D shape: area of the 2D shape is decreasing with the depth to 0% at the depth of the source . 3D value :salinity to the depth of the source and in the center is 0% and increases to 100% at the surface	yes	yes

6.2	Movement: the vertical line formed by centers of 2D shapes can oscillate. The source position is fixed end while the other end to the surface varies according to a segment whose length, direction and frequency are configurable	being validated	yes
7	***Source display***		
7.1	Source 2D map	yes	yes
7.2	Source 3D plume	yes	possible**
7.3	Source movement	yes	possible*
7.4	Display surface limitation	yes	no
8	***Communication management***		
8.1	No protocol: shared bandwidth, everybody is speaking in the same time or mulitchannel separates groups of agents	yes	yes
8.2	TDMA protocol	yes	yes
8.3	Coverage management (3D, 360°), distance limitation	yes	possible*
8.4	Coverage management (3D, any cone)	being validated	no
8.5	Interference if 2 signals are emitted at the same frequency in the same time	being validated	possible*
8.6	Four different trajectories are calculated for the acoustic wave with 1 and 2 reflections	being validated	possible**
8.7	Environment noise includes engines of vehicles	being validated	possible*
8.8	Uses the signal attenuation, data length and multipath to compute a probability of data loss	being validated	possible*
8.9	Each communication packet contain its own delay which is the sum of propagation delay and emission delay (=packet size/bitrate)	being validated	possible*
8.10	Shadowing zone for communication because of a vehicle or a rock in the direct path	no (Prolexia)	no
8.11	Aerial communication management	being validated	no
8.12	Integrate the characteristics of new underwater radio frequency modems	possible**	possible**
8.13	Uses temperature, salinity and depth in the acoustic signal attenuation model and data detection using baseband acoustic signals.	no	no
9	***Communication display***		
9.1	Display coverage area (2D, 360°)	yes	possible*
9.2	Display coverage area (3D, any cone)	being validated	no
9.3	Green line connecting two vehicles exchanged a message successfully,red line connecting two vehicles exchanged a message with data loss	being validated	possible*

12. 5. Detailed functionalities

12.1. 5.1. System requirements

12.1.1. 5.1.1. Matlab-Simulink

For both MUSim and MASim, Matlab-Simulink R2007b is needed to run algorithms.
Several “S function” are used so to modify and compile this part of the simulator, Visual C is required.

12.1.2. 5.1.2. Internet access

In MUSim, ConnectSim installation needs internet access.

12.1.3. 5.1.3. Windows XP

In MUSim, ConnectIHM needs windows XP.

12.1.4. 5.1.4. Hardware

For MUSim, to run properly ConnectSim, ConnectIHM and Simulink together, hardware needs good performances. Cache of CPU and graphical NVIDIA card is particularly important.
For example, MUSim is installed on a tablet PC Dell Precision M6400 with:
- Intel® Core™2 Extreme Mobile Processor QX9300 (12MB of cache)
- Graphical card NVIDIA quadro fx 3700M (1GB)

It is also possible to install ConnectSim, ConnectIHM and Matlab/Simulink in separates Computer to improve performances because the communication between these applications is Ethernet UDP based.

12.1.5. 5.1.5. Ownership and exploitation rights

MUSim is co-owned by the consortium of Connect and is restricted to internal and research use.
MASim is owned by Gipsa-Lab and has no specific limitation.

12.2. 5.2. Time characteristics

12.2.1. 5.2.1. Sample time

In MUSim, typical sample time are 125ms or 100ms (exception of the dynamic model of AsterX).
We can speed up this 50ms but it increases data loss between applications and especially critical control data between ConnectSim and Simulink.
In MASim, there is no limitation of sample time.

12.2.2. 5.2.2. Sample time of dynamic model of AsterX

This part of simulation is simulated by Simulink but in variable-step sample time mode (solver “ode15s”). The sampling period is therefore be smaller for a better approximation of a continuous behavior
This function has been implemented and validated in and is planed to be implemented in MUSim.

12.2.3. 5.2.3. Simulation speed

MUSim cannot simulates faster than real time with the typical scenario of 5 AUV, using cinematic model for AsterX and Ts=125ms.
MASim can simulate 10 times faster with the same scenario.

12.3. *5.3. Vehicles and fleet*

12.3.1. 5.3.1. Energy vehicles management

If necessary, a model of energy consumption of a vehicle can be implemented in Simulink and include to the simulator.

12.3.2. 5.3.2. Computation vehicles management

If necessary, a model of computation limitation of a vehicle can be implemented in Simulink and include to the simulator.

12.3.3. 5.3.3. Fleet scalability for MUSim

For MUSim, any number of AUV's, ASV's and AFV can be adding in the scenario, one by one. But, because of the time performance and the time to build the scenario, it is reasonable to have a total number of vehicles around 10 and not more than 100.

12.3.4. 5.3.4. Fleet scalability for MASim

Copy-paste can be done to add new vehicle with all characteristics and equipment.
Thanks to its lighter architecture, the maximum number of vehicle is most important. Tests are needed to assess the possible limitations.

12.3.5. 5.3.5. Using dynamic model of AsterX for AUV's instead of cinematic one

Dynamic model of AsterX is implemented in MASim. This part of the simulation is done with the variable sample time mode.
Controller will be able to use variable sample time from 50 to 300ms, synchronized with sensors if necessary.
This functionality is planned to be implemented in MUSim.

12.3.6. 5.3.6. Positioning drift

Using real position and calculated position is not implemented.
A positioning system with a drift can be modeled in simulink. We can first model a drift of 2m/h, different for each agent.

For MUSim, displaying 2 different positions in ConnectIHM is not possible.

12.3.7. 5.3.7. Bathymetric sensor

In MUSim, a bathymetric sensor is planed to be implemented in each AUV.
In MASim, such a sensor is possible if there is a non flat seabed implemented.

12.3.8. 5.3.8. Collision Management

No collision management is implemented.
Such functionalities seem to be not necessary and hard to implement in simulink.

12.4. 5.4. Environment

12.4.1. 5.4.1. 3D map of the non flat seabed

3D map of the non flat seabed is implemented in MUSim. This map, owned by Ifremer, describes the seabed of a real area near Nice.
This map is used by ConnectSim for bathymetric sensor and to calculate reflection of acoustic signals (see 4.8.7.)
This map is only used by ConnectIHM to display it.
The implementation of this map in MASim has to be studied if necessary.

12.4.2. 5.4.2. Current and wind

One current and one wind fixed and uniform is implemented in ConnectSim.
This can be defined also in MASim if necessary.
Current is planed to affect only the dynamic model of AsterX.
Plume of the source is defined by an equation which already includes movement. Using the current to affect the plume of the source would change totally the source definition. If necessary, this could be done only for MASim and has to be discussed with Prolexia and Robosoft for MUSim.

Wind doesn’t affect anything.
To use Wind, dynamic model of ASV and AFV has to be implemented in Simulink.

12.5. 5.5. General display

12.5.1. 5.5.1. 2D map

Both MASim and MUSim can display the simulation in 2D from above.
MUSim displays a richer graphics and colorful.

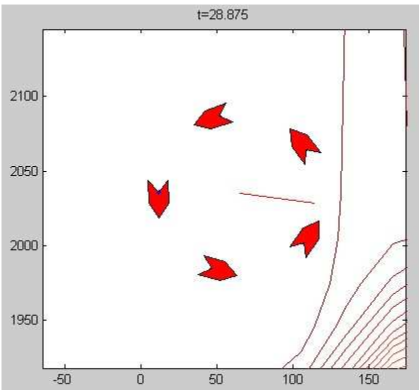

Figure 2: 2D MASim display with Matlab tools

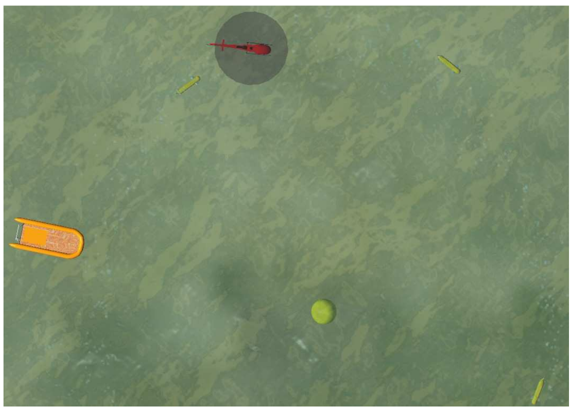

Figure 3: 2D MUSim display with ConnectIHM

12.5.2. 5.5.2. 3D display mode

MUSim displays the simulation in 3D with ConnectIHM.
It is possible to display 3D in MASim but it will be with Matlab tools so the graphics will be quite poor.

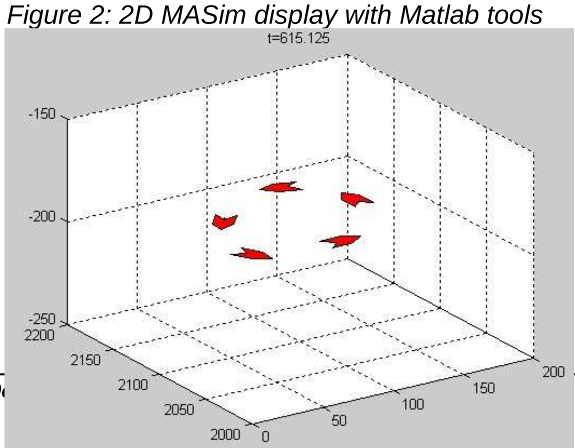

Figure 4: 3D MASim display with Matlab tools


Figure 5: 3D MUSim display with ConnectIHM

12.5.3. 5.5.3. Rich 3D natural environment and vehicles

MUSim displays a rich 3D natural environment and vehicles.
Available vehicle in ConnectIHM: 2 types of boats, 2 types of AUV, one type of AFV and one sphere.


Figure 6: AFV in ConnectIHM


Figure 7: ASV in ConnectIHM


Figure 8: Base boat in ConnectIHM

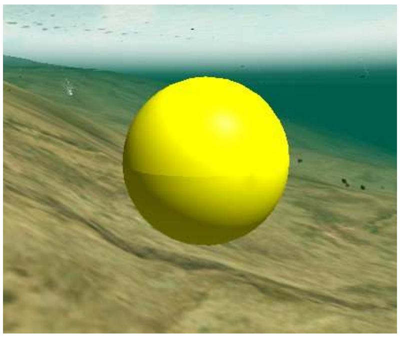
Figure 9: Sphere in ConnectIHM

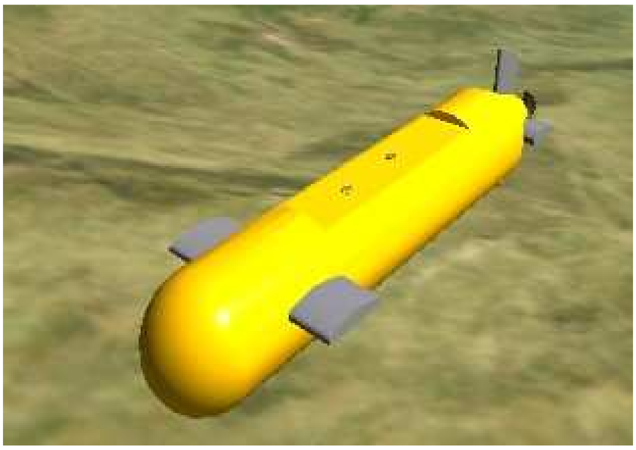

Figure 10: AUV AsterX in ConnectIHM

Figure 11: AUV MiniSub in ConnectIHM

Non flat seabed is display based on a real seabed map from Ifremer near Nice (see figure 5).
On the surface, for the landscape around Nice, the shape of the coastline and the relief are real and the texture is generated randomly.

Figure 12: Landscape with the terrain around Nice in ConnectIHM

Figure 13: Surface landscape in ConnectIHM

Show this kind of graphics in MASim involve the full development of a new GUI or a direct gateway between ConnectIHM and MASim. This solution requires specific developments by Robosoft. If this type of tool is necessary, we should first conduct a small study of feasibility.

12.6. 5.6. Source management

12.6.1. 5.6.1. Static 3D shape of the source

In stationary regime, the diffusion laws give an exponentially decreasing spatial distribution of concentration, represented in 2D by the following equation:

```
Salinity=100*(1-e^(-K*((X-xs)²+(Y-ys)²)))
```

Where (xs,ys) is the center of the source and K gives the distribution's spread.

We select a steady-state for the diffusion because we postulate that the transient time is over or the time of the mission is much shorter than the diffusion time.
That means that in our study case, the source is like a fresh water source (here since years) or a source of pollution that has already reached its steady-state or with a relatively low flow rate.

To represent the fact that different currents can distort the circle, we replace the circular form by the sum of three concentric ellipses, while retaining the exponential decay (see section 5.6.2, equations (f),(g) and (h)).

To extend this to a 3D shape and for reasons of simplicity, we chose to change the concentration of the center of 2D figure linearly from 100% at the source level to 0% at the surface (see section 5.6.2, equation (b)).

To represent the diffusion along z, the spread of ellipses is weighted by depth and the 2 firsts ellipses also are exponentially reduced by the depth (see section 5.6.2, equations (a) and (c)).

This description is both used by MASim and MUSim and is summarized in section 5.6.2 Figure 14.

Change this description in MUSim needs new development for Prolexia and Robosoft.
Change this in MASim is possible if necessary.

We can also add a noise to the measurement of salinity (white noise with a variance equal to 1%).

12.6.2. 5.6.2. Movement of the source

To get a movement of the source due to the wave, we chose to weave the vertical line formed by centers of 2D shapes.
The source position is fixed end while the other end to the surface varies according to a segment whose length, direction and frequency are configurable (see equations (d) and (e) below).

Finally, the concentration distribution related to the source is described by the following algorithm:

```
%parameters
xs=200;%position x,y of the source
ys=100;
zs100=-500;%depth source (100%)

m1=0.2;% weighting ellipse 1
m2=0.6;% weighting ellipse 2
m3=1;% weighting ellipse 3

Ps=1e-3;%scaling

a1=1.9;%ellipse 1 : mi.exp(-Ps*(ai.x²+bi.x.y+ci.y²))
b1=0.01;%we need a*c-b^2/4>0
c1=0.003;
a2=0.03;%ellipse 2
b2=-0.2;
c2=1;
a3=0.9;%ellipse 3
b3=-1;
c3=0.4;

alpha=3;% direction of movement in degrees
A=30;% range of motion in m
Tm=10;% period of movement in s

%% Equations
% weighted by z for ellipses 1 and 2
m=[m1*exp(0.1*(1-abs(z)/25)) m2*exp(0.1*(1-abs(z)/25)) m3]'; %(a)


%scaling of m: salinity is 100% at xs,ys,zs100 and linear decrease along z
m=m*100/(m(1)+m(2)+m(3))*z/zs100; %(b)

% spread is weighted by depth
M=((abs(z)+25)/50)*[a1 a2 a3;b1 b2 b3;c1 c2 c3]; %(c)

% change of variable for movement
xsm=xs+A*cos(2*pi/Tm*t)*cos(alpha*pi/180)*((zs100-z)/zs100); %(d)
ysm=ys+A*cos(2*pi/Tm*t)*sin(alpha*pi/180)*((zs100-z)/zs100); %(e)

%3 ellipses
Sal1=exp(-Ps*(M(1,1)*(X-xsm).^2+M(2,1)*(X-xsm).*(Y-ysm)+M(3,1)*(Y-
ysm).^2)); %(f)
Sal2=exp(-Ps*(M(1,2)*(X-xsm).^2+M(2,2)*(X-xsm).*(Y-ysm)+M(3,2)*(Y-
ysm).^2)); %(g)
Sal3=exp(-Ps*(M(1,3)*(X-xsm).^2+M(2,3)*(X-xsm).*(Y-ysm)+M(3,3)*(Y-
ysm).^2)); %(h)

Sal=m(1)*Sal1+m(2)*Sal2+m(3)*Sal3; %(i)

Sal=100-Sal; %(j)  to get 0% at xs,ys,zs100
```

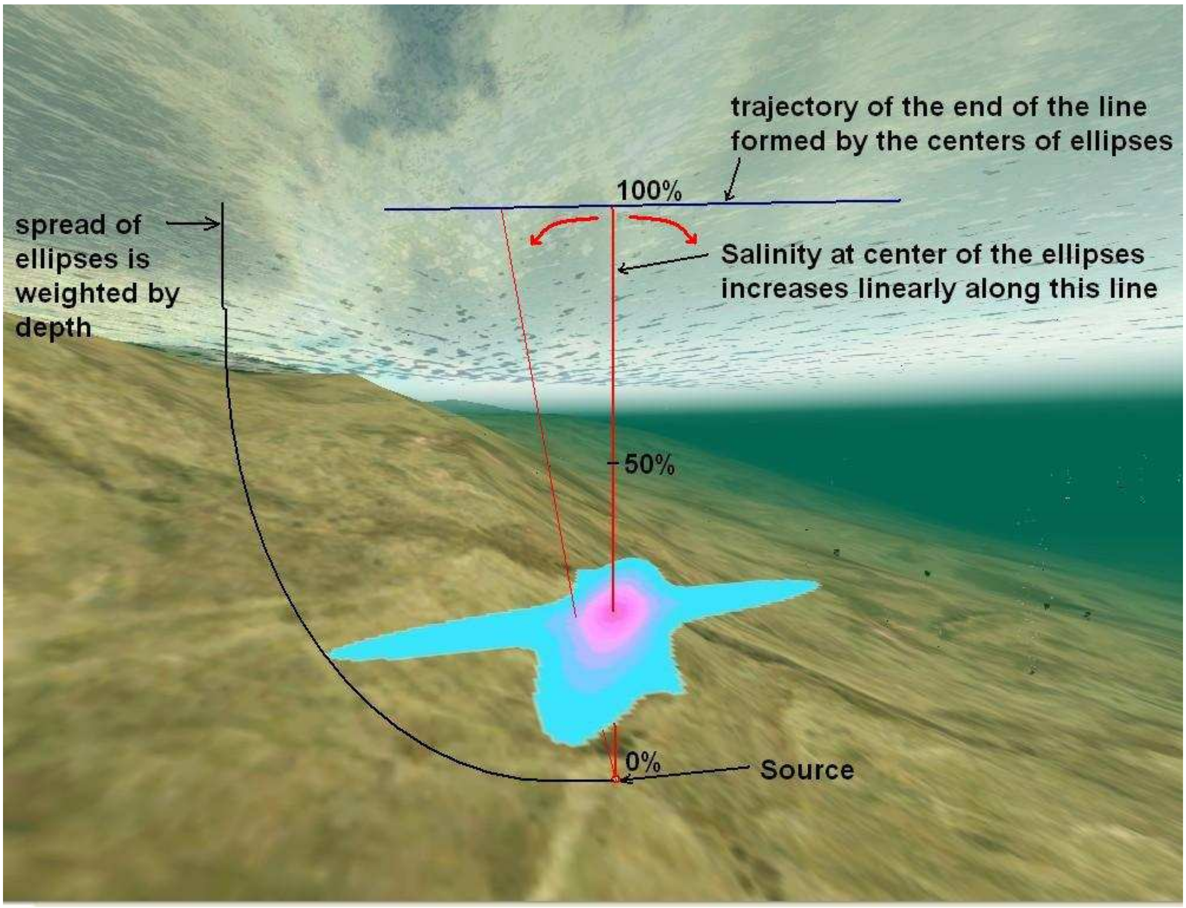

Figure 14: Summary of the source

This movement description is implemented in MUSim and has to be validated. It is already validated in .

12.7. 5.7. Source display

12.7.1. 5.7.1. Display source in 2D

Both MASim and MUSim can display the source distribution in 2D from above (See figures 15 and 16).

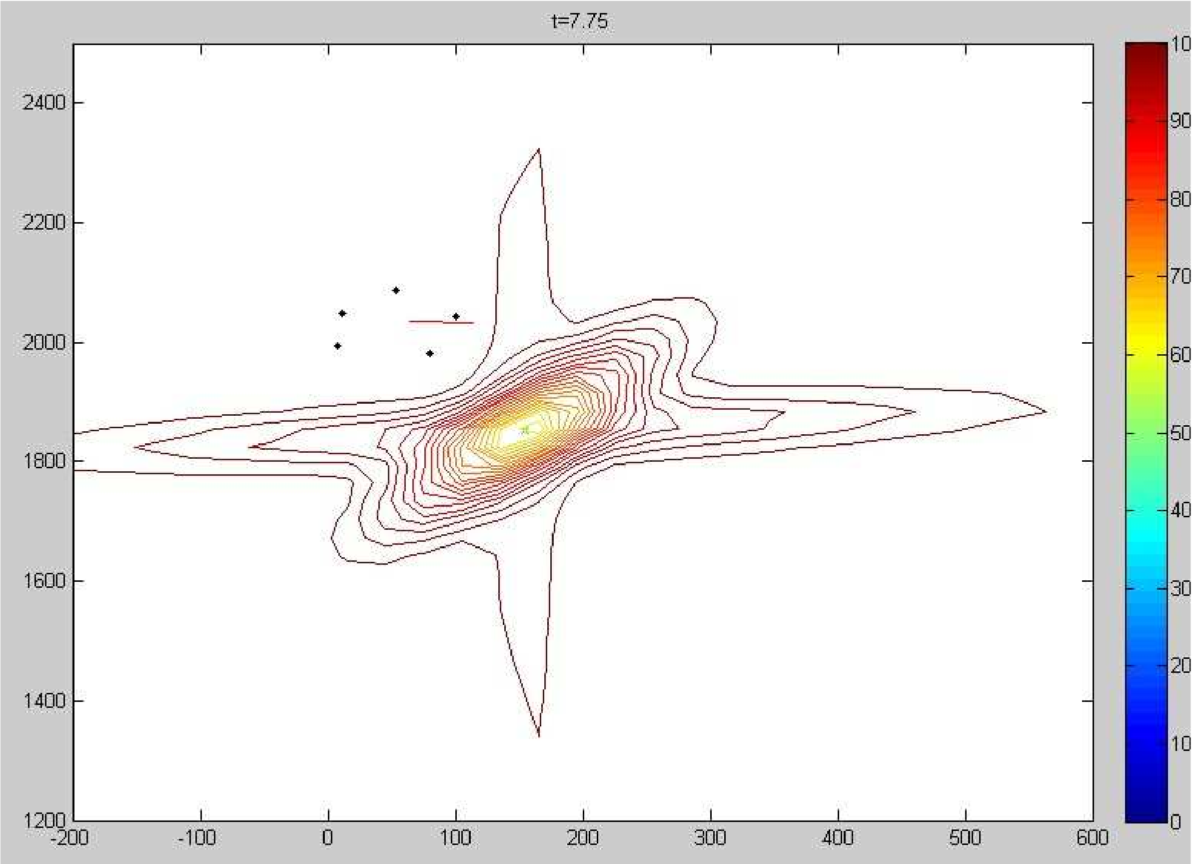

Figure 15: 2D source in MASim using Matlab tools

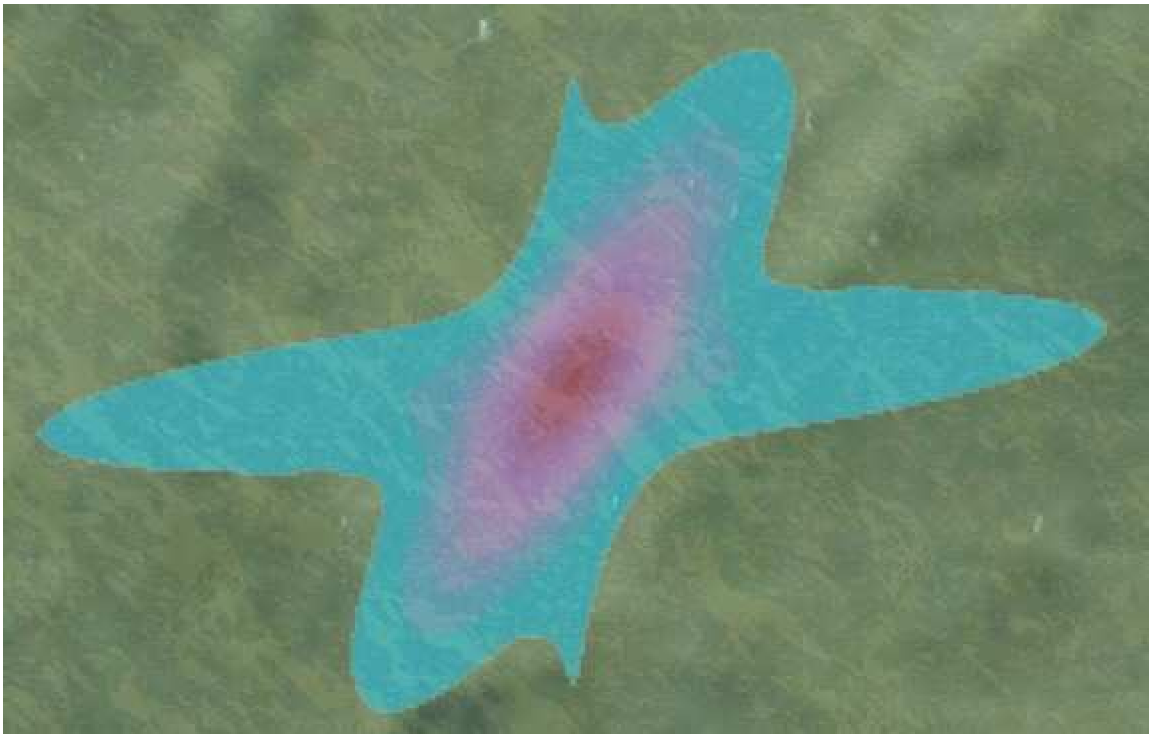

Figure 16: 2D source in MUSim / ConnectIHM

12.7.2. 5.7.2. Source 3D plume

MUSim can display the source plume in 3D.
We can display the 2D map of the source to a depth parameterized (see figure 17)
It can also display the full plume in 3D: a number of bubbles appear in the volume around the source whose size, color and / or transparency depends on the salinity at this point (see figure 18)

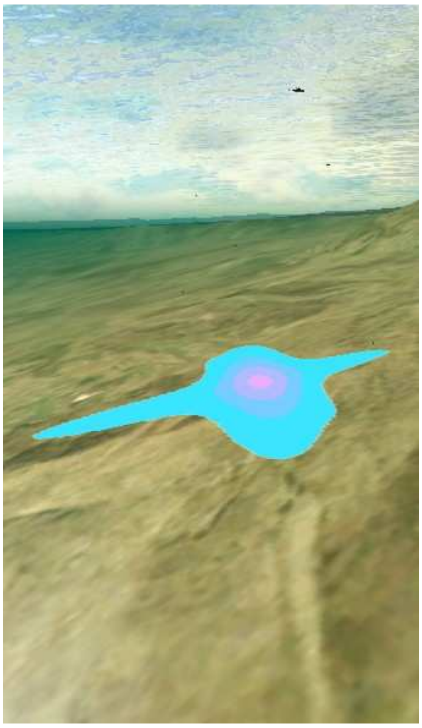

Figure 17: 2D source in 3D

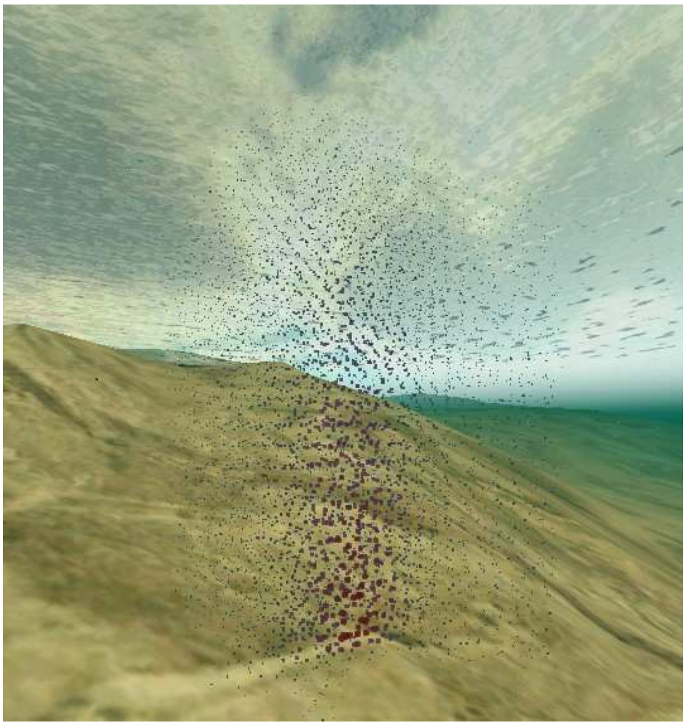

Figure 18: source plume in 3D

For MASim, display source plume in 3D with matlab tools seems to be difficult but can be studied if necessary.

12.7.3. 5.7.3. Source movement

Source movement is displayed in MUSim and it is possible to do it in MASim with Matlab tools if necessary.

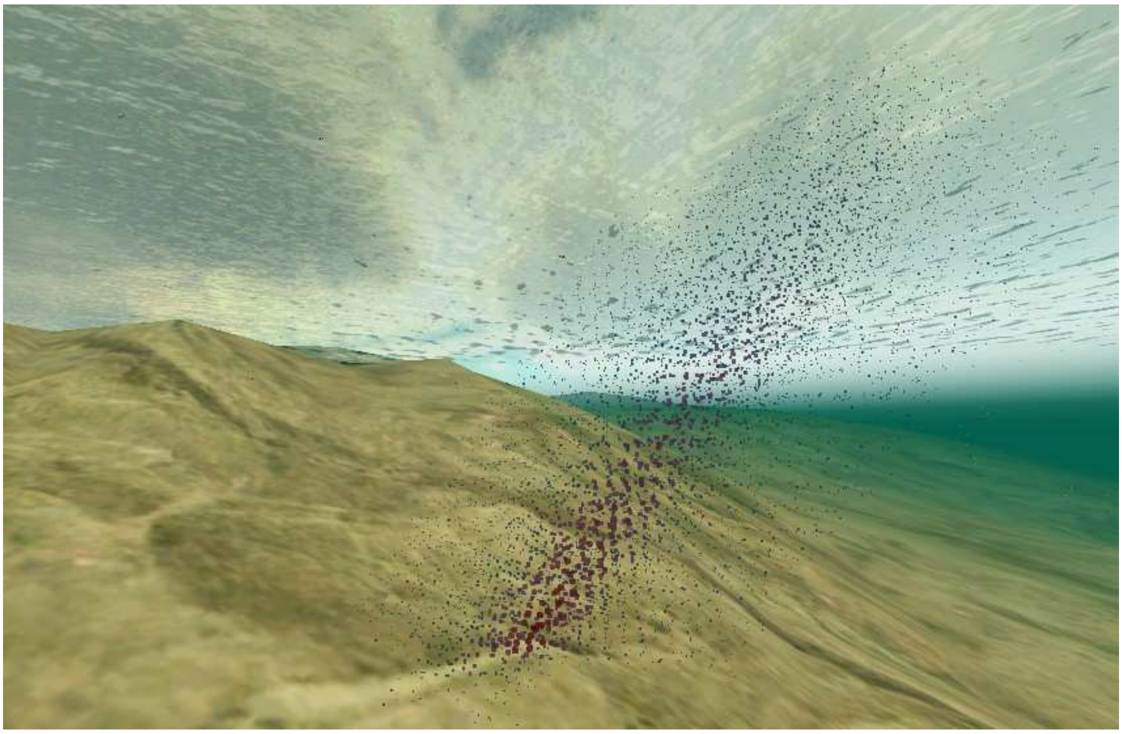

Figure 19: source plume in 3D in movement

12.7.4. 5.7.4. Display surface limitation

In MUSim, the definition of the display decreases when we increase the surface view: in figure 20 we see that the display of the source extended on a square of 2x2km does not have good definition.

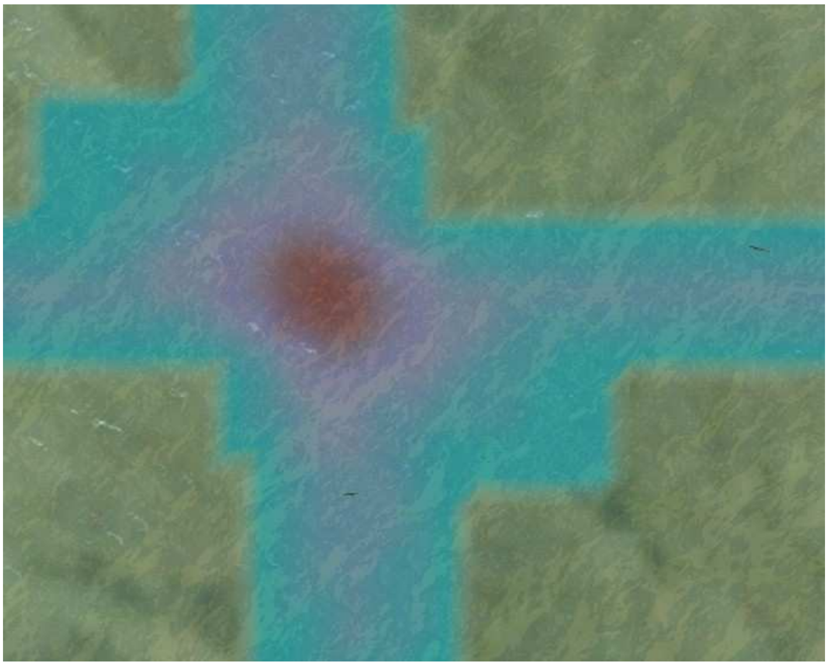

Figure 20: source definition with a 2x2km square of display

12.8. 5.8. Communication management

12.8.1. 5.8.1. No limitation, multichannel

All to all communication with no bandwidth limitation is implemented both in MASim and MUSim.

In MUSim, we can set a range of frequency for each acoustic modem (emission and reception).
Each agent can also have several modems in board, so that multichannel communication, for separating Surface-AUV and AUV-AUV communications for example, can be simulated.
In MASim, such a function can be easily implemented.

12.8.2. 5.8.2. TDMA protocol

TDMA protocol is implemented both in MASim and MUSim.
For a network with N agents, the transmission time is split into N consecutive slots, each slot being allocated to a specific agent. Ttdma, the duration of a slot, is configurable. The content of the slot is organized as follows:
[id,x,y,z,Head,S,|V|,t]
Id: Agent's identification number
x,y,z: Position of the agent
Head: Heading of the agent
S: Salinity measure
|V|: Magnitude of the agent's speed
t: Time stamp when the agent starts to transmit.

Each agent has a table in memory of Nx9 numbers which correspond to 9 numbers for each agent including himself.
These 9 first number are, in board of agent 2,related to agent1: [Com,x,y,z,Head,S,L,|V|,t]
Com is 1 if [x,y,z,Head,S,L,|V|,t] are updated (if it were received during this cycle of calculation)
[x,y,z,Head,S,|V|,t] are last data sended by agent 1
L is the distance between agent 1 and 2 (calculated by some modem)

So for its own data (from 10 to 18),L=0 and Com=1 all the time because it always know its own data.

Number and nature of data exchanged can be modified if necessary.

12.8.3. 5.8.3. Limited range of communication

In MUSim, one of the management modes of communication consists in defining a transmission power, calculating the corresponding attenuation, and then comparing the attenuated signal to a pre-defined threshold associated with the modem receiver.
Below the threshold the signal is considered as lost. At a distance X (in Km), the attenuation A(x) of an acoustic underwater signal of frequency f is given by

$A(x) = x^k \, 10^{x.g(f)/10}$

g(f) (in dB/Km) denoting the absorption coefficient for the frequency f. It can be approximated by the Thorp's formula as follows:

$$g(f) = \left(\frac{0.11 f^2}{1+f^2}\right) + \left(\frac{44 f^2}{4100+f^2}\right) + 2.75\times10^{-4} f^2 + 0.003$$

f : Frequency of the signal (kHz)
k : the energy spreading factor (*we take k=1.5)*

For a given frequency, the communication range is computed by setting the transmission power and the reception threshold.

Attenuation does not include temperature and depth.

Online adaptation of the transmit power in order to change the communication range during the simulation can be studied if necessary.

Limited communication range is not implemented in MASim but it can be if necessary.

12.8.4. 5.8.4. 3D cone of communication coverage

In MUSim, the modem can be fixed on the AUV at any position and orientation.
We can also define its opening angle for reception. Combined with the range communication, it yields a 3D cone of coverage.

This function is programmed but not yet tested/validated yet.
Programming such a function in Simulink for MASim seems to be difficult.

12.8.5. 5.8.5. Interference at the same frequency

Interference if 2 signals are emitted at the same frequency in the same time is planned to be done for MUSim.
This will be done in simulink for allowing an implementation in MASim if necessary.

12.8.6. 5.8.6 Management of acoustic waves reflection

Wave reflections induce additional signal attenuation.
Four different trajectories are calculated for the acoustic wave with 1 and 2 reflections:
1-Direct path
2-Surface reflection (no attenuation)
3-Seabed reflection (-3dB attenuation)
4-Surface + seabed reflection (-3dB attenuation)
5-Seabed + surface reflection (-3dB attenuation)

For trajectories 3,4 and 5, depth of reflection is taken from the non flat seabed map at the middle of the trajectory. From a geometrical point of view, such a calculation is correct for the third trajectory. It corresponds to a somewhat valid approximation for the remaining cases.
For one emission, if this function is activated, another modem can potentially receive 5 identical frames to different power levels (because of the potential attenuation of reflection and the fact that the length of each trip is different).

This function will be used to test the robustness of communication algorithms to multipath.

This function is programmed in MUSim but not yet tested/validated yet.
Programming such a function in Simulink for MASim can be done if necessary but it needs to define a non-flat seabed.

12.8.7. 5.8.7. Acoustic noise

In MUSim, we can set a fixed noise for each agent corresponds to the noise of its engine.
This acoustic noise is attenuated with the same equations than the acoustic signals of interest (see section 5.8.3).
When “SNR” function is active in an acoustic modem, the threshold used for computing the communication range (section 5.8.3) is the Signal to Noise Ratio (SNR) including its own noise and attenuated noise of other agents.
The communication distance can therefore decrease when vessel is sailing nearby.

This function is planned but not yet programmed yet in MUSim.
Programming such a function in Simulink for MASim can be done if necessary.

12.8.8.

12.8.9. 5.8.8. Probability of data loss

In MUSim, after calculating the attenuation of the signal, instead of filtering with a simple threshold (see section 5.8.3), we can use a statistical law of data loss.
This statistical law uses signal attenuation, acoustic noise (see section 5.8.7), multipath perturbation (see section 5.8.6) and data length.
We also use parameters Gamma1 and Gamma2 to represent the performance of the modem.
Data loss probability is computed with the following equations :

```
%inputs
Pu=30;% transmission power in W
N=10;%noise near receiver in W
fe=20;%emitter frequency in kHz
l1=0.1;%length in km of path 1 (direct)
l2=0.1118;% length in km of path 2
l3=0.2236;% length in km of path 3
l4=0.2693;% length in km of path 4
l5=0.2693;%longueur en km du path 5
multi=0;%use multipath yes=1, no=0

%Parameters
gamma1=0.5;%characteristic parameter of the receiver and the transmission
%           environment(between 0 and 1, 1=best case)
gamma2=0.04;% degradation due to multi-path algorithm (performance of the
algorithm)
Kb=200;%number of bits in the packet
df=10;% receiver bandwidth in kHz

%constants
GL1=1;% Absorption coefficient of  path 1 to 5
GL2=1;
GL3=sqrt(0.5); %-3dB
GL4=sqrt(0.5);
GL5=sqrt(0.5);
k=1.5; %spreading of energy factor

% Absorption coefficient
adb=0.11*(fe^2/(1+fe^2))+44*(fe^2/(4100+fe^2))+2.75*10^(-4)*fe^2+0.003;
a=10^(adb/10);

%multipath variance
LL=[l1;l2;l3;l4;l5];
varl=var(LL);
```

```
% Attenuations
A1=(l1^k)*(a^l1);
H1=(GL1/sqrt(A1));
A2=(l2^k)*(a^l2);
H2=(GL2/sqrt(A2));
A3=(l3^k)*(a^l3);
H3=(GL3/sqrt(A3));
A4=(l4^k)*(a^l4);
H4=(GL4/sqrt(A4));
A5=(l5^k)*(a^l5);
H5=(GL5/sqrt(A5));

H=H1+multi*(H2+H3+H4+H5);

SNR=Pu*abs(H)^2/(N*df);

rr=exp(-sqrt(varl));

% Probability of symbol loss
gamma=gamma1*(1+multi*(gamma2*rr-1));
Ps=erfc(sqrt(gamma.*SNR))/2;

%Probability of packet loss
P=1-(1-Ps)^Kb
```

This function is programmed but not tested/validated yet.
Programming such a function in Simulink for MASim can be done (maybe in a simplified version) if necessary.

12.8.10.5.8.9. Delay

In MUSim, we can use a function that calculates the delay of the communication packet.
This delay is applied to the packet and wrote in order to use it later in simulink.

This delay is the sum of propagation delay and emission delay (=packet size/bitrate).
To define packet size, we choose to consider the transmission of an array of numbers and fixe to 32bits the definition of each number.
Bit rate is set for each modem.

For example, if we send the array [id,x,y,z,Head,S,|V|,t] (defined in section 5.8.2.), packet size will be 8*32=256bits.
If modem bitrate=480bits/s, emission delay=256/480=0.533s.
If the distance between emitter and receiver is 400m, propagation delay = 400/1500=0.266s.
Total delay = 0.533+0.266= 0.8s.
The packet is received after 7 sample times (after 0.875s) and contains its exact delay (0.8s).

The exact delay could be used for example to test the robustness of communication algorithms to multipath.

This function is programmed in MUSim but not tested/validated yet.
Programming such a function in Simulink for MASim can be done if necessary.

12.8.11.5.8.10. Shadowing zone for communication

Shadowing zone for communication because of a vehicle or a rock in the direct path is not considered. If necessary, this possibility has to be discussed with Prolexia.

12.8.12.5.8.11. Aerial communication management

In MUSim, aerial electromagnetic waves management is implemented with the following equation.
FSL[dB]=C+20 log(D)+20 log (F)
D= distance
F=frequency (MHz)
This function is programmed in MUSim but not tested/validated yet.

As simulation of upper layers like WiFi 802.11b/g/n standards can be used,
it seems to be useless to develop this part.
Moreover, flows to the conventional types of communication are very large compared to those available in underwater communication.
We can therefore consider that air communications are instantaneous, lossless and with unlimited bandwidth.

12.8.13.5.8.12. Underwater radio frequency modems

To communicate between AUV, only acoustic modems are implemented in MASim and MUSim.
Integrates characteristics of underwater radio frequency modems could be interesting.

12.8.14.5.8.13. Limitations of the acoustic propagation model

The equation of attenuation of an acoustic underwater signal (section 5.8.3) does not use temperature, salinity and depth.
Data detection using baseband acoustic signals will also not be implemented.

12.9. 5.9. Communication display

12.9.1. 5.9.1. Display coverage area

MUSim can display communication area related the limited distance (section 5.8.3).
When agent noises is used to limit the communication area(section 5.8.7.), the area display is the maximum one (with only its own noise).

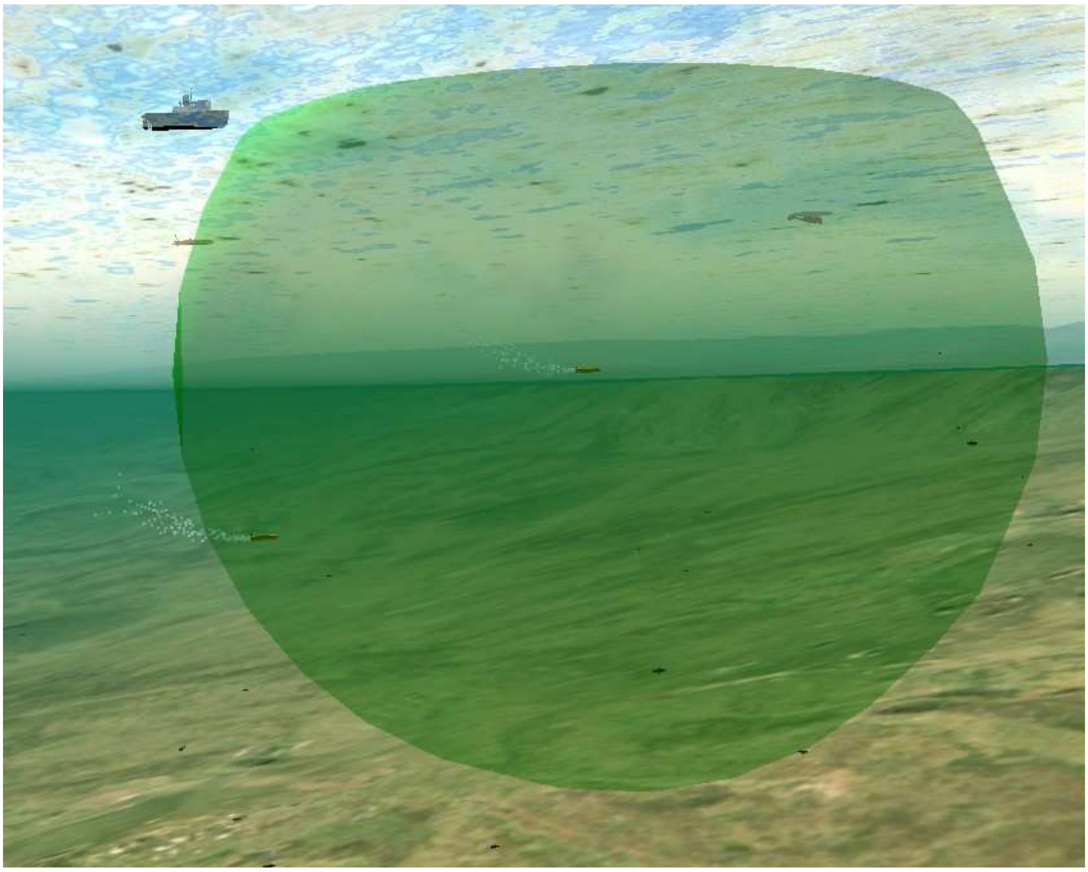
Figure 21: Communication range display

Display this area of communication in MASim is not implemented and is possible if necessary.

12.9.2. 5.9.2. Display any 3D cone of coverage

MUSim can display communication area related the limited opening angle for reception (section 5.8.4).

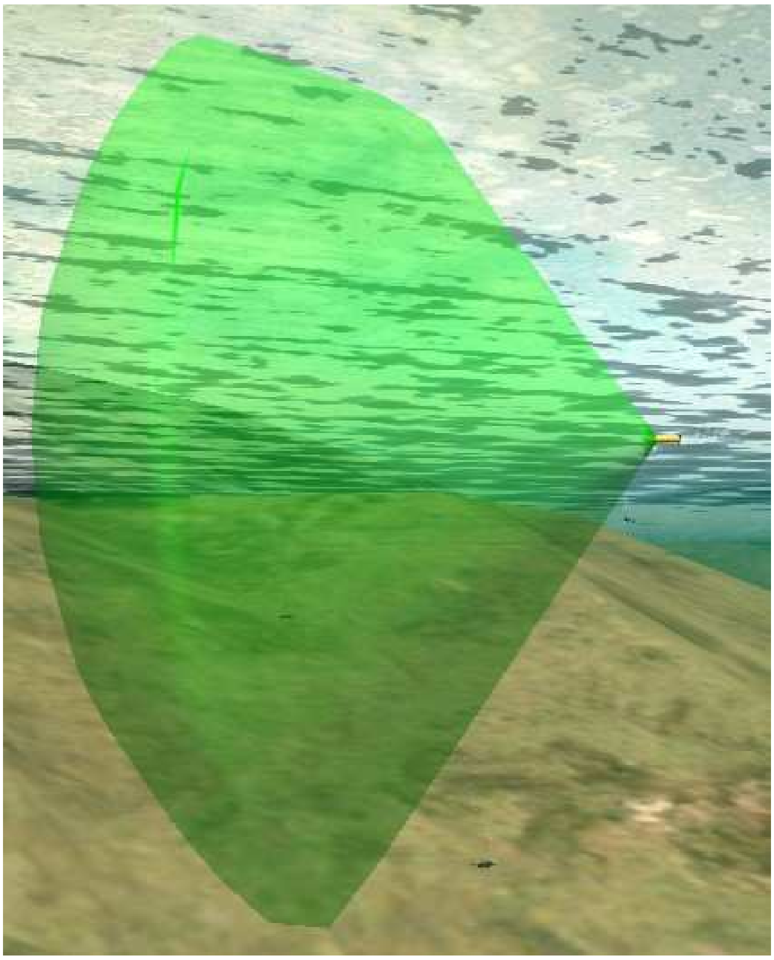

Figure 22: Communication range with opening of 110°

This function is programmed in MUSim but not tested/validated yet.
This is not possible to do this in MASim.

12.9.3. 5.9.3. Display connectivity graph

In MUSim, if the function is activated, each successfully received packet produce a green line connecting transmitter and receiver and a green spot is traveling along this line from one to the other. Moreover, each unsuccessfully received packet produce a red line connecting transmitter and receiver.

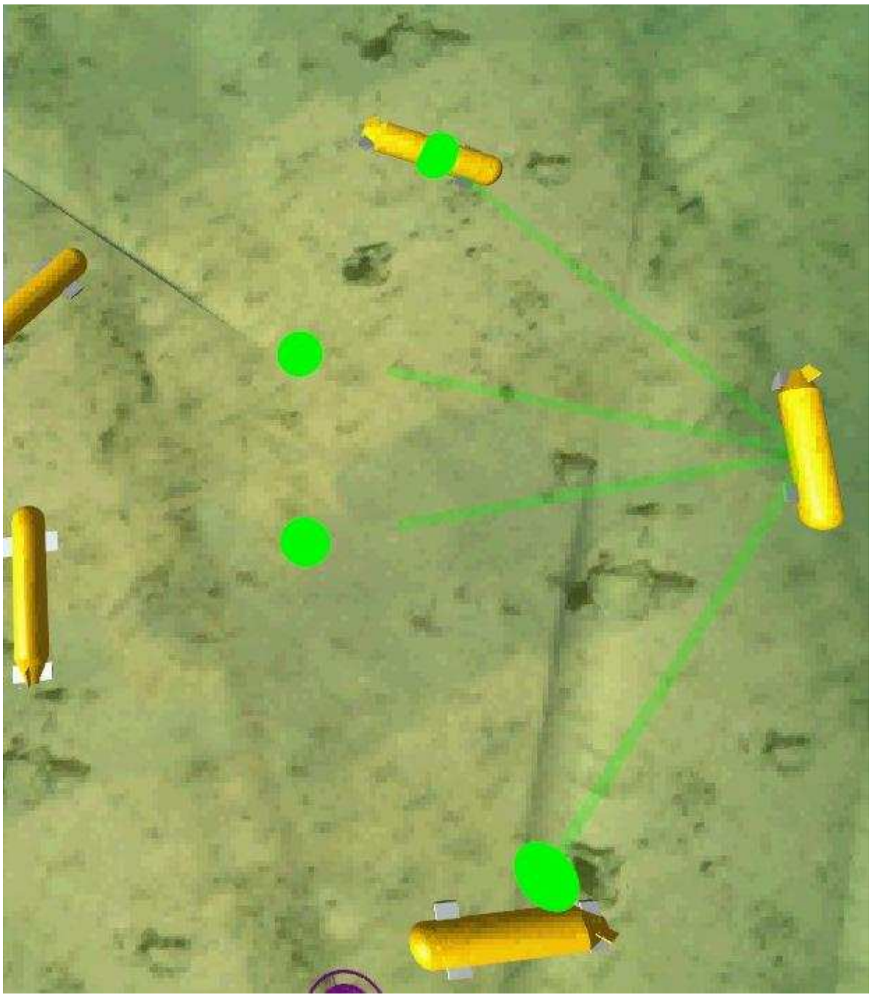

Figure 23: Green line and spot for each successfully received packets

This function is programmed in MUSim but not tested/validated yet.
In MASim, it is possible to display a connectivity graph with data loss if necessary.