Hardware-in-the-loop simulators for multiple vehicle scenario : survey on existing solutions and proposal of a new architecture

Bruno Jouvencel, Lionel Lapierre, Olivier Parodi

Hardware-in-The-Loop Simulators for multi-vehicles scenarios: survey on existing solutions and proposal of a new architecture

Olivier Parodi, Lionel Lapierre, Bruno Jouvencel
LIRMM - University of Montpellier 2 - CNRS
161 rue Ada, 34392 Montpellier, France
{parodi, lapierre, jouvencel}@lirmm.fr

Abstract— **The aim of this paper is to review the available simulators defining in a first step the necessary requirements to cope with the multi-vehicles cooperation context. In order to get a complete overview of these tools, a new classification is proposed and is used to sort the existing simulators. It is worth to be noted that we focus our interest on simulators which can at least address the problem of marine robotics.**

None of the available simulators being compliant with the previous requirements, a new simulation architecture is proposed. *Thetis* is a real-time multi-vehicles hybrid simulator for heterogeneous vehicles. This simulator allows *Hardware In The Loop* (HIL) simulations including the use of virtual sensors which allows to provide a representation of a virtual world, and includes the support of communication devices. The included acoustic propagation model allows communications between the vehicles.

Finally some simulation results involving one or more AUV (Autonomous Underwater Vehicle) and/or USV (Unmanned Surface Vehicle) are presented.

I. INTRODUCTION

Scenarios involving several autonomous robots, potentially heterogeneous to explore the environnement have recently been proposed. Coordinating these vehicles brings real advantages: this allows to explore a environment faster than with a single robot, to multiply the points of view (terrestrial or aerial), to acquire data at different scales, to allow the deployment of many more complementary sensors and finally to increase the redundancy of the acquired data.

Such benefits spark off the interest of several laboratories and are the subject of active fields of research. As said before, we are particularly interested in projects involving marine crafts (surface or underwater). Among these projects, the Programme d'Etude Amont Action (PEA Action) intends to prepare the future technologies allowing the deployment of heterogeneous networked vehicles [1]. The purpose of the AUSI (Autonomous Undersea Systems Institute) project [2] is to evaluate a long-survey coastal network composed of many autonomous vehicles (SAUV: Solar-powered AUV). The VACAS (Virginia Center for Autonomous Systems) facilitates interdisciplinary research in autonomous systems technology [3]. It hosts research activities spanning every domain: water, land, air. The CONNECT (CONtrol of Networked Cooperative sysTems) project implements physical subsystems interconnected by a heterogeneous communication network [4]. Finally we collaborate with ISSIA-CNR laboratory aiming to run experiments in a joined mission of the *Taipan* AUV developed at LIRMM and the *Charlie* USV developed at ISSIA.

This research has been partially funded by the CNR-CNRS bilateral agreement "Coordinated mission control for autonomous marine vehicles" and by the Parco Scientifico e Tecnologico della Liguria s.c.p.a.

Original coordination strategies have been developed (multi-vehicles path following, adaptive ocean sampling,...), however using them in real experiments faces many problems: supervision, localisation, communications (especially underwater acoustic propagation), implications of a robot failure on the other ones, the heterogeneity of the involved vehicles, and the fact that robots come from different organizations. These difficulties added to the sophisticated control architecture make prior validation (logico-temporal behavior of the controllers, localisation algorithms, coordination strategies ...) of such experiments necessary. This need has been perfectly identified by the teams of previous mentioned projects as all plans to develop a simulator in the coming years (except for AUSI which already owns one).

Of course, many simulators exist, each of them having its own limits and being used at different steps of the development. Thus, the first step is to study the existing simulators to determine if they could be used to accurately prepare heterogeneous multi-vehicles experiments. A first classification of some existing simulators can be found in [5] however it is not enough to deal with the context of robots coordination.

This paper is organized as follows. In section II, the state of the art on the proposed simulators within the field of marin robotics is presented and discussed, as well as a new classification of the proposed simulators. *Thetis* simulator is described in section III, presenting its basic principle, its advantages, and its contribution with respect to the other simulators drawbacks. Simulation results are presented and discussed in section IV, showing the effectiveness of the proposed solution. The paper ends with some concluding remarks in section V.

II. STATE-OF-THE-ART AND NEW CLASSIFICATION

A. Requirements for pre-experiments simulations

Among the different types of existing simulators [5], the most adapted for the missions preparation are the HIL or hybrid ones. With HIL simulator, the control algorithm is executed on the robot itself, but the control references are routed towards the simulator instead of the real actuators.

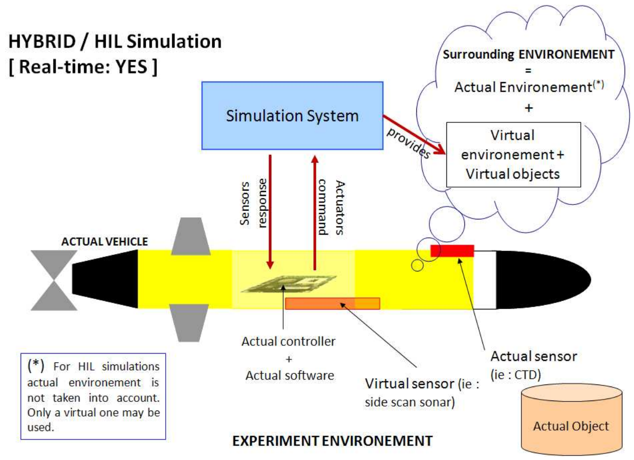

Fig. 1. In a HS, the actual vehicle is in the operational environment and completes its mission. The simulator provides a virtual world, with virtual objects and virtual sensors. This simulator is qualified as hybrid because the actual robot reacts to actual or virtual stimuli during its actual mission (ie virtual obstacles avoidance).

The simulator then considers the actuation references in order to compute the dynamic evolution of the system. The result of the simulated sensors are then sent to the robot. Hybrid simulators are HIL simulators where real and virtual systems interact together in virtual environment. It is therefore necessary to simulate an environment (static or dynamic) in which the robot software architecture will be fully functionally operative (fig. 1). Therefore, it is possible to test all the algorithms on the machine, from low level control of sensors or actuators recruitment to the high level algorithmic architecture. This approach has been used by several authors such as in [5] where Neptune simulator is presented. It is a real-time graphic multi-vehicles simulator, allowing to perform online, HIL and hybrid simulation.

Indeed, the coordination strategy of each vehicle has to be robust enough to cope with degraded communications, unclear localisation of the other robots, unavoidable failures... Testing the logical aspect of the mission involving several vehicles is far from being an easy task because it requires the presence of all robots to evaluate the behavior of each of them facing the actions and communication of the other ones. A multi-vehicles simulator is therefore necessary to test the whole system before running real experiments.

Considering a simulator to test scenarios of multi-vehicles, in which it is not possible for the robots to communicate among themselves seems absurd. Indeed, occurrence of a communication feature is absolutely essential and is required to work in the context of cooperation.

Simulating several vehicles, with different dynamic models operating in a complex environnement and interacting with more or less sophisticated sensors suggests a high computational load. A single computer is not enough to implement such a simulator, at least for real-time simulation. Several computers are obviously required and the simulation system must be distributed on these computers.

B. Existing works

The first works were done by the DARPA Naval Technology Office in 1988 [6]. This simulator allows the cooperation between many underwater platforms which are driven by a real time intelligent sense-decide-act system. The simulation of sensors and environment are taken into account.

In [7], a distributed simulator for underwater vehicles called Core Simulation Engine (CSE) has been developed. This system is equipped with operator training capabilities, mission feasibility assessment and mission replay. A subscription method and a run time infrastructure allowed the distributed programming. This simulator does not allow multi-vehicles simulation.

In [5], a simulator called Neptune, composed of a set of processes running on a local area network, has been developed. The simulation runs onto three computers which are the onboard AUV computer, a human machine interface computer and a third computer used to compute system dynamics and to represent a virtual world. In this simulator, communications between the AUVs are not allowed.

In [8], the authors developed a simulator called DVECS (Distributed Virtual Environment Collaborative Simulator) allowing the test of real or simulated ROVs and AUVs in a virtual synthetic world. This world provides real or virtual objects and perturbations. The publications do not present details on the architecture of simulation, its implementation or simulation results. A features list is presented and although it seems to be interesting, it is not possible to have a scientific opinion from this information. Moreover the communication between vehicles is instantaneous with an infinite bandwidth (use of shared memories).

The Cooperative AUV Development Concept (CADCON) simulator [9] uses a client-server model in which it is possible to handle interactions between vehicles controlled by the simulation clients through an environment simulator. This simulator system focuses on the high level communication and does not deal with the dynamics and control of heterogeneous vehicles.

In [10], the authors propose a simulator (MVS) to which several agents can sign in. Each agent is connected to one (or more) process(es) that provides data from a virtual world. No information on how these agents are connected to process, nor on the influence of this connection on the behavior of the controller of the agent are discussed. Communications are modeled as a shared memory. Thus communications are instantaneous with no duration and without interferences. In addition, a limited number of agents can connect to a single world. In this case the only solution is to create a new one. Unfortunately, the worlds do not communicate among themselves and therefore neither do the agents belonging to each of the world.

Many other simulators have been developed like in [11], [12], [13] and all these simulators present several very helpful specificities, through several models (environment, vehicles) which are able to cope with reality accurately, but they do not allow multi-vehicles simulation.

As shown, there are a large number of simulators. In the next section a new classification is proposed, identifying a set of criteria as questions.

C. A new classification

P. Ridao proposed a useful classification in [5] in which four types of simulators are distinguished. However this classification is not sufficient to deal with the preparation of multi-vehicle experiments. Furthermore, there is a large number of criteria allowing us to qualify a simulator. For these reasons a new classification method is proposed in order to clearly identify the functionalities provided by a simulator. 20 criteria divided in 7 groups are chosen:

- MVC: Multi-VehiCle
  - IDT: Is it possible to simulate several IDenTical vehicles ?
  - SCL: Is it possible to simulate heterogenous vehicles but belonging to the Same CLass (AUV or USV or UGV, ...) ?
  - DCL: Is it possible to simulate heterogeneous vehicles belonging to Different CLasses ?
- COM: Inter-vehicle COMmunications
  - WPP: Are the communications possible between the vehicles Without taking into account the Phenomena Propagation ?
  - EPL: Is the simulator able to Emulate the Physical Layer for the waves propagation ?
  - PMD: Is there a Propagation MoDel inside the simulator ?
- SIM: SIMulator
  - HYB: Does the simulator have a HYBrid mode ?
  - HIL: Does the simulator have a Hardware-In-the-Loop mode ?
  - ONL: Does the simulator have a ONLine mode ?
  - OFF: Does the simulator have a OFFline mode ?
- WRD: WoRlD
  - SGI: Are the Seabed, the Ground or Ice cover modelized ?
  - STO: Is it possible to add some STatic Objects (mines, pipelines, ...) ?
  - DYO: Is it possible to add some DYnamic Objects (fish, submarine,...) ant to potentially choose a specific behavior for them (script, physic law, ...) ?
- ENV: ENVironement
  - EPS: Are the Environmental Phenomena (wind, currents, ...) considered as Stationary ?
  - EPD: Are the Environmental Phenomena (wind, currents, ...) considered as Dynamic ?
- EXT: EXTeroceptive
  - OTH: Does the simulator provide support for the use of exteroceptive sensors OTHer than vision and sonar (eg CTD) ?
  - SNR: Is it possible to simulate a SoNaR ?
  - VSN: Is it possible to simulate ViSioN sensors ?
- MSL: MiSceLlaneous

Main author	Name of the simulator	MVC	COM	SIM	MSL	Concerned robot
Conte		No	No	OFF	No	
Bono	RobySim	No	No	OFF	No	Roby
Hornfeld	DeepC Simulator	No	No	ONL	3DD	DeepC
Robinson		No	No	ONL	3DD	
Bielohlawek	SubSim	No	No	ONL	3DD	Mako
Phoha	SAMON	No	No	ONL	DST	SUV
Suriano		No	No	HIL	3DD	Sara
Brutzman		No	No	HIL	3DD	Phoenix
Devie		No	No	HIL	3DD	Redermor
Lane	CSE	No	No	HIL	3DD & DST	Angus
Bruzzone	SE	No	No	HIL	3DD & DST	Romeo
Ridao	Neptune	No	No	HYB	3DD	Uris
Gracanin		SCL	No	ONL	3DD	Phoenix, Phantom
Song		SCL	No	HIL	DST	Ocean EXplorer
Canell		IDT	WPP	OFF	No	
Komerska	SauvSim	IDT	WPP	HIL	No	Solar-powered AUV
Choi	DVECS	SCL	WPP	HIL	3DD & DST	Odin
Chappell	CADCON	SCL	WPP	HIL	3DD & DST	Solar-powered AUV
Kuroda	MVS	DCL	WPP	HYB	3DD & DST	Twin Burger

Fig. 2. Summary of main functionalities of studied simulators. Only the last five solutions may partially address our problematic.

  - DST: Is it possible to DiStribuTe the simulator (the controller of the vehicle is not considered as a part of the simulator) onto several calculation units (network or multi-processor) ?
  - 3DD: Does the simulator provide a 3D Display ?

By answering these questions, it is possible to compare the simulators and to determine which of them is the most suitable for a given context. For example, it has been previously determined that a distributed (DST) multi-vehicles HIL simulator (MVC) with communications (COM) support is needed. Thus all the existing simulators are sorted regarding these four features. The results are presented on figure 2.

Thanks to this new classification and with the definition of minimum requirements for missions preparation of multi-vehicles, the conclusion is that among the 19 studied simulators, only five could be qualified. They are however not usable in our context, because of some limitations previously mentioned, inherent to their architecture. These examples show that many concepts have been developed, but they lack an open architecture bringing together the concepts and actors.

Moreover, none of the studied simulators takes into account the constraints related to communications. Indeed, if the radio wave aerial communication offers good performance, this is not the case for underwater acoustic communications. The model of communication using ”shared memories” is therefore invalid and does not reflect the effectiveness and validity of the developed algorithms. Indeed, the available bandwidth for acoustic communications is very low (about 20 bits/s in our case), subject to environmental disturbances (noise, multi-paths, gradient of salinity and temperature...) and related to interferences with other acoustic equipments (sonars, loch doppler, acoustic modems...), possibly carried by other vehicles. The propagation velocity in water also involves latency in transmissions. Anisotropic attenuation of propagated acoustic waves makes the area coverage uncertain. In these conditions, it is difficult to

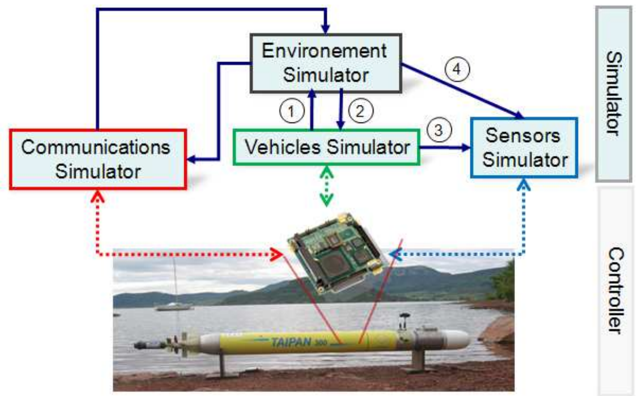

Fig. 3. Simplified architecture of *Thetis*: there is a logical sequence between 3 simulators even if there are independent processes on different computers. The cycle duration of this sequence must be largely lower than the period of the controller. The temporal decoupling is only effective (and that's enough) between the simulator and the controllers. The link between the communications simulator and the environment simulator is event-driven (when a communication between vehicles occurs).

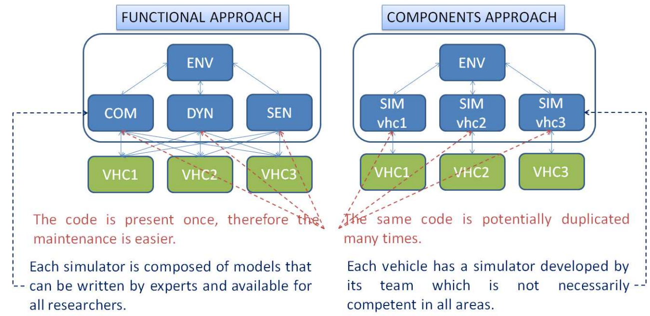

Fig. 4. Drawbacks of the conventional architecture.

guarantee that a robot is always attainable. None of these simulators takes into account the behavior of communication equipments. Yet it is clear that an acoustic modem can not receive a transmission while emitting. These types of devices have a behavior that can not be neglected.

For all these reasons *Thetis* has been developed. It is a hybrid simulator which provides the necessary environment to experiment HIL simulation with support of communications between the vehicles, and allows the use of exteroceptive sensors. Such a simulator which concurrently provides all these functionalities is a necessary tool to envisage such complex scenarios. This is a challenging multi-disciplinary task because of the number of specialists who must collaborate (acoustician, computer specialist, control engineer ...). The aim of this simulator architecture is not to provide a set of "perfect" models: in this paper a simulator architecture is described allowing to deal with the problem of heterogeneous coordinated vehicles, under communication constraints. Moreover, the open source *Thetis* project is built to allow many different specialists to develop their own models. On the other hand this simulator architecture is distributed, in order to allow the implementation of complex models, without being restricted by computational burden, thus guaranteeing the real-time aspect. Finally this simulator allows to interchange the used models, according to the desired accuracy.

III. Thetis: Hybrid Simulator

Thetis is a new real-time hybrid simulator allowing the consideration of heterogeneous multi-vehicles scenarios, including communication constraints. A quick overview on the overall architecture of the simulator is presented on figure 3. For details on previous development, please refer to [14], [15] which focuse on the connection aspects of *Thetis* to actual *Taipan* class robots and [16] where the simulation capabilities of communications between two AUVs, in different spatial locations, are demonstrated.

A. A simulator, but an architecture before all

1) A functional approach: The proposed architecture respects several points that identified as being critical within the multi-vehicle simulation context. Thinking of an architecture before working on the modelization is the way preferred in order to develop this simulator. This approach allows to make the simulation system compatible with the objectives exposed hereafter. An innovative approach is proposed. It consists in dividing the simulator into functionalities (e.g. sensors simulator, vehicles dynamic simulator, ...) rather than dividing the simulator into components (e.g. one simulator for the overall vehicle) as it has been the case until now. This provides several advantages: it is possible to create a simulator suited to the functionalities of a same elements family (indeed a sensors simulator is not designed like a vehicles dynamic simulator for example). A component (e.g. a vehicle), is no longer running on a single computing unit but on a set of computers. Furthermore this allows to adapt the simulation loop period, avoiding temporal coupling between the models. Finally the simulating code for one element (e.g. a sensor pressure), is at a unique place and is not duplicated on each computer simulating a vehicle: this allows an easier maintenance (figure 4).

2) An elementary module: Dividing the simulation system as shown on figure 3 and distributing the different parts on a network, present some real drawbacks: an important number of exchanged messages on the network (in our case about 57600 frames/min for three basic vehicles) which implies ressources-consuming to encode and decode the exchanged frames, blocking waitings, inconveniences for the developer because they have to deal with the network communication and protocol to disseminate the computed data. An elementary module is proposed in order to avoid these drawbacks which is presented on figure 5. A first process called "simulation process" is in charge of making the models evolve. The models themselves can be designed with this elementary module in order to be also distributed. A second separate and autonomous process called "reception and decoding" is listening for the new incoming data sent by the other simulators and then shares them in a common shared area. These two processes can be executed onto

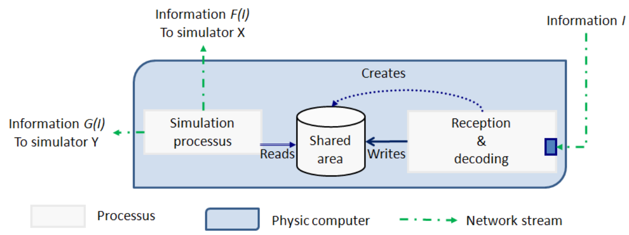

Fig. 5. *Thetis* simulator elementary module.

two different processors in order to avoid disturbing the simulation cycle by introducing unexpected network delays and by consuming resources to encode and decode the frames. Thus, this approach allows the simulated entity (for example the dynamic model of a robot) to continue running even if a delay occurs (for example the controller of this robot has a problem). This is a key feature to test the behavior of the controllers.

3) Overall architecture: Each of the 4 simulators composing the *Thetis* system, is itself composed of several independent processes (16 in total). The structures of these 4 simulators are quite similar. Indeed, each of them is composed of one or more independent process(es) in charge of IPC (Inter Process Communication), decoding and writing the data exchanged by the simulators, to a local shared memory initially created by these processes. For each simulator, a main process is in charge of the models evolution. This structure, preferentially executed on a dual (or quad) core processor, prevents the main processor from being disturbed during the inter-simulators communications.

The vehicles simulator is in charge of computing the robots dynamics evolution. The robots send actuators commands computed by the onboard computer to the simulator through UDP socket. The *Dynamics Models* compute all the forces and torques applied to the robots in order to determine the vehicles accelerations, attitudes, positions and velocities. The computed data are sent to the environment simulator in order to verify the absence of collisions and if necessary to correct positions. Afterwards, the computed data are sent to the sensors simulator.

The sensors simulator is in charge of providing the real robot with virtual data from the simulation. Presently the models of proprioceptive and simple exteroceptive (Temperature, Conductivity,...) sensors are implemented. The outputs of the *Sensors Models* are computed according to each sensor model (including noise, offsets...), to the data from the vehicles simulator (systems state), and lastly to the data from the environment simulation. Once these outputs are computed, they are specifically sent to the concerned robots with the same refresh rate as the real sensor.

The environment simulator is in charge of providing the stored geophysical maps around a given 3D geographical point (Temperature, Salinity, Local current, Plankton density), computing signal propagation (when communications between vehicles occur: it takes into account the bandwidth, the latency, the signal level, interferences...) and detecting physical collisions between robots or with the environment. It has to be noted that the computed communication signals are only sent to the communications simulator when (considering latency and bandwidth) and if necessary (if the vehicles are not within the range of communications, no message will be delivered).

The communication devices Simulator The mechanism of the communication devices simulator takes into account the delay, the rate and the losses caused by the type of communication device in use. It has to be noted that coordinated control of AUVs flotilla is intrinsically dependent on the communications performances, which cannot be guaranteed. Thus a simulator able to perform multi vehicles simulation has to consider these aspects explicitly.

The supervisor has been developed to manage the launching sequence of the system: it begins by loading the network configuration file, then it distributes the required files to previously available declared computers. After that it tests the presence of all the components of the simulator (including the vehicles) and starts the processes respecting a sequence. During the simulation, data from all the processes are sent to it and thus it displays the state of the simulation system. At the end, it downloads the logs files to analyse the results of the simulation.

The 3D display is currently under development. It allows to visualize the simulation scene in 3D. It is developed in openGL.

B. Modularity and open architecture

An XML-based specifications exchange (XML for eXtensible Markup Language) is proposed, allowing to structure the parameters of the different models (modem, radio, fins, motor...), and configuration files, while promoting the modularity and the portability. Some examples of these files can be found on *Thetis* website [17]. Indeed modularity is obtained using the XML format, which allows to specify the components parameters (response-time, accuracy, rate...), and to modify them in a very easy way. Now, the replacement of a sensor by another is done by calling an XML file in place of the other. All the system components description follows the same idea (actuators, sensors, body dynamics...). The validation and extensible properties of the XML language make it an ideal base to enrich the model parameters files. Thus the robots components used in this simulator are described in XML formalism.

Information about network configuration is described in a shared XML file. All the XML files describing the components of the system are loaded at the initialization and thus allow to instantiate the different objects of the simulator.

Finally a set of libraries have been created, containing a set of classes enabling us to build the various objects of the simulation system. All these classes are documented with Doxygen tool [18] and are available on *Thetis* website.

C. Respect of critical concepts

In this section we verify that the proposed architecture meets previously defined critical concepts for multi-vehicles

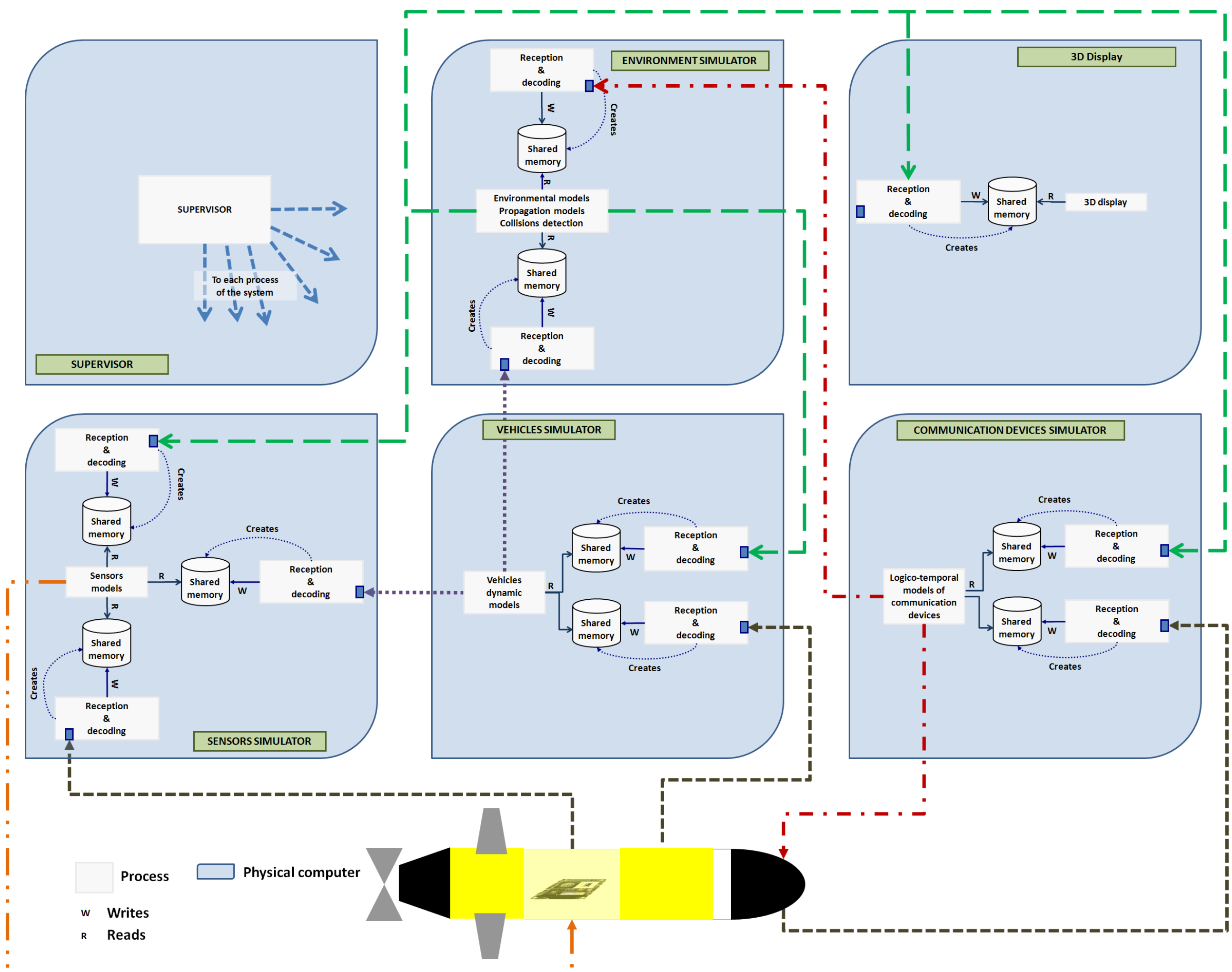

Fig. 6. Detailed view of *Thetis*. Dashed lines stand for the UDP/IP inter-simulators communications.

simulation. Indeed, choices that have guided us toward this architecture arise from contraints to be respected.

The collaborative aspect of the proposed simulator is undeniable: many researchers from different fields can work on an aspect without having to know the functioning of the whole system. The mechanisms allow developers to ignore inter-process communication and the data update.

It is possible to simulate heterogeneous robots: the proposed architecture does not care about the type of robot connected. Only models that are integrated can simulate different robots. The simulation system clearly separates the architecture on which the robots are connected from the models which simulate these robots. Every controller of any robot is therefore potentially compatible with this simulator.

The proposed architecture is clearly distributed, which corresponds to a critical point that we raised. Indeed, the simulator is composed of 16 autonomous processes, distributed and executed on several different computers, connected to a dedicated network. This approach allows to clearly address the context of multi-vehicle within constraints of real-time, thanks to the distribution of computing over several units of calculation.

This architecture also supports inter-vehicle and multiple devices of communication for each robot. This proposal allows to make the distinction between logico-temporal behavior of the communication device and wave propagation in the medium; this was also a key point to be observed.

It is possible to connect directly the robot controller to the simulator. We therefore created a HIL simulator. Furthermore the temporal decoupling between the control and the loop simulation is ensured thanks to the adoption of UDP protocol on the one hand, and the establishment of a elementary module on the other.

Finally the 3D display is a module of this architecture and allows to display a 3D rendering of simulated scene. Although this feature is not yet implemented, the architecture is designed to facilitate its implementation.

IV. Presentation of the Finalized System

A. Thetis

Figure 7 illustrates the *Thetis* system linked to AUV *Taipan300*. On the picture we can see four computers

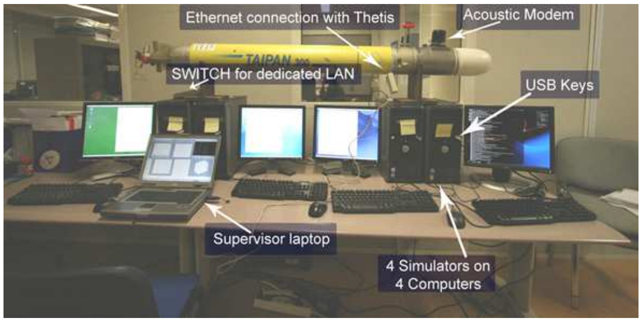

Fig. 7. The *Thetis* system and the AUV *Taipan300*

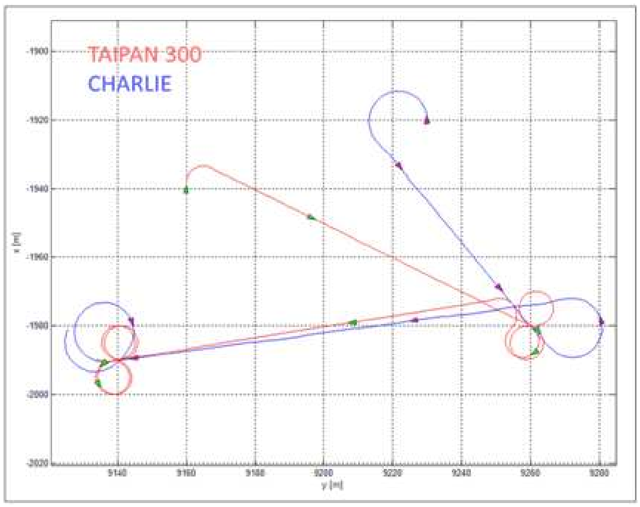

Fig. 8. Connection of the two vehicles to the *THETIS* simulator.

(P4HT@2.7GHz + 1024Mo DDR linked to a dedicated LAN via a switch), each of them running a simulator. These four computers boot on a USB key enabling the deployment of the system anywhere. Moreover on the photo we can see a fifth computer (laptop) which executes the supervisor.

B. Some results

Thetis is used to develop and test different types of algorithms and flotilla control strategies. This section briefly presents some examples of the obtained results to show some capabilities of the simulator.

First *Thetis* is used to prepare a mission involving *Taipan* AUV and *Charlie* ASV. The catamaran *Charlie* developed by ISSIA has been successfully connected to the simulator in a few hours showing the modularity of *Thetis* ([19]). Figure 8 presents a simulation result in which *Charlie* emits by its virtual acoustic modem a tracked waypoint given by the end user through wifi link. When received, *Taipan* moves toward this waypoint.

Then *Thetis* is used to test backtracking localisation algorithms (figure 9). The problem of delay and interferences in communications has been highlighted and solutions must be found now.

Figure 10 presents the use of *Thetis* to test salinity acquisition of a fresh water spring (in the sea). In this simulation

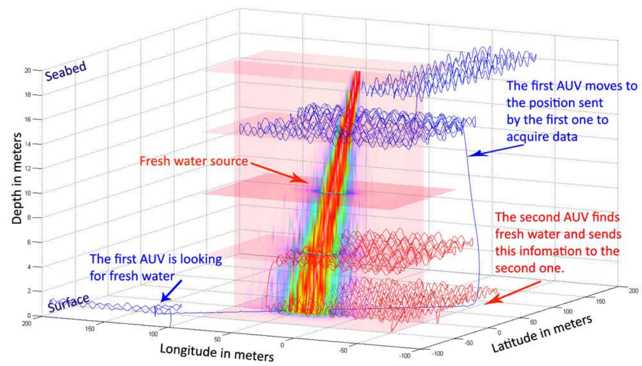

Fig. 10. The environment simulator of *Thetis* is able to represent some properties (wind, current, salinity...) of a volume using netCDF files. Two AUVs are involved here (the red and the blue trajectory). The two AUVs have to find a fresh water spring (random trajectory), and when found, the finder sends the position and the depth of the volume to be acquired to the second AUV(crenels trajectory).

there is no current.

Simulations of path following between AUV and ASV or ASV and ASV are currently tested. Thanks to its low level simulation of propagation, *Thetis* is also used to develop original hybrid (underwater acoustic and radio waves) communications protocols.

V. Conclusion

HIL simulator plays a key role in the development of the controllers of robots. This is particularly true when experiments involving several robots are envisaged.

In this article, the bases of a simulator adapted to multi-vehicles experiment context is established. A new method of simulators classification is proposed thanks to which we present a extensive bibliography of systems at least compliant with marine vehicles. We conclude that none of the available simulators is currently adapted to prepared multi-vehicles cooperation experiments.

Thus a new architecture which is different from the others by using a functional approach is proposed. This architecture gathers all the important features that a simulator must include in order to allow simulation of flotilla. In order to solve the problems linked to the distribution of the system, an ”elementary module” on which the architecture is built is proposed.

A new formalism based on XML to describe any mobile robots is proposed. This allows a quick configuration of new robots (sensors, actuators,...) and the possibility to change the model or the parameters of the model without any compilation step. This feature allowed to quickly connect the USV *Charlie* to the proposed system.

Finally simulations results are presented to highlight the capabilities of this simulator. The conclusion of the different simulations run is that opening problems are clearly stated with the use of *Thetis* because it is not possible to ignore communications constraints, or the ability of the robots to exchange data to cooperate. *Thetis* is therefore the last required step for ongoing multi-vehicles experiments: it helps

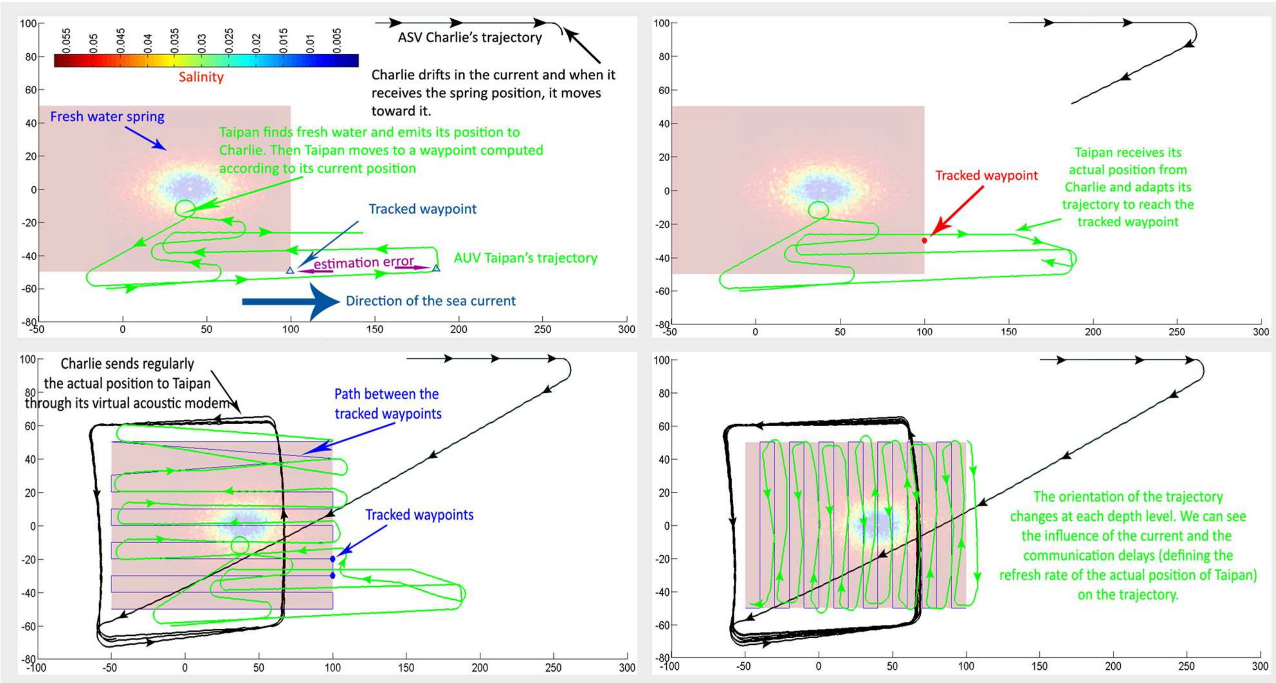

Fig. 9. *Taipan* follows the green trajectory. It is supposed to map the fresh water spring, but due to the current *Taipan* is not in the expected area. *Charlie* emits the actual position of *Taipan* through an virtual acoustic modem and *Taipan* can backtrack its actual position with an error due to the latency and the delay of propagation.

us to implement, test and validate realistic solutions for cooperation.

References

[1] PEA Action. http://action.onera.fr/ , accessed on 02/09.
[2] AUSI. http://ausi.org/ , accessed on 02/09.
[3] VACAS. http://www.unmanned.vt.edu/ , accessed on 02/09.
[4] Projet CONNECT. http://www.lag.ensieg.inpg.fr/connect/index.php , accessed on 02/09.
[5] P. Ridao, E. Batlle, D. Ribas, and M. Carreras. Neptune: A hil simulator for multiple uuvs. *OCEANS '04. MTTS/IEEE TECHNO-OCEAN '04*, 1:524–531, Nov. 2004.
[6] J. S. Albus. System description ans design architecture for multiple autonomous underwater vehicles. Technical report, National institute of Standards and Technology, Gaithersburg, MD, 1988. 1251.
[7] D.M. Lane, G.J. Falconer, G.W. Randall, N.D. Duffy, J.T. Herd, P. Chernett, J. Hunter, M. Colley, J. Standeven, V. Callaghan, J. Smith, J. Evans, A. Woods, J. Penrose, G.A. Whittaker, D. Smith, and I. Edwards. Mixing simulations and real subsystems for subsea robot development. specification and development of the core simulation engine. *OCEANS '98 Conference Proceedings*, pages 1382–1386 vol.3, Sep-1 Oct 1998.
[8] S.K. Choi and J. Yuh. A virtual collaborative world simulator for underwater robots using multidimensional synthetic environment. *IEEE International Conference on Robotics and Automation, ICRA'01*, 1:926–931 vol.1, 2001.
[9] Komerska R.J Chappell S.G. An environment for high-level multiple auv simulation and communication. *Proceeding of the Collaborating and Leveraging Outer Space and Undersea Technologies (CLOUT) NOAA/NASA Workshop*, August 2000.
[10] Y. Kuroda, K. Aramaki, and T. Ura. Auv test using real/virtual synthetic world. *Proceedings of the Symposium on Autonomous Underwater Vehicle Technology, AUV'96*, pages 365–372, Jun 1996.
[11] Donald P. Brutzman. *A Virtual World for an Autonomous Underwater Vehicle*. PhD thesis, Naval Postgraduate school, Monterey, California, 1998.
[12] Tobias Bielohlawek. Subsim - an autonomous underwater vehicle simulation system. Technical report, Arbeitsgruppe Robotersysteme Fachbereich Informatik Universitat Kaiserslautern, April 2006.
[13] Willi Hornfeld. Deepc, the german auv development project.
[14] B. Jouvencel O. Parodi, V. Creuze. Communications with thetis, a real time multi-vehicles hybrid simulator. *ISOPE'08: International Society of Offshore and Polar Engineers*, July 2008.
[15] O. Parodi, A. El Jalaoui, and D. Andreu. Connectivity of thetis, a distributed hybrid simulator, with a mixed control architecture. *Fourth International Conference on Autonomic and Autonomous Systems, ICAS 2008*, pages 130–135, March 2008.
[16] B. Jouvencel O. Parodi, L. Lapierre. Thetis: A real-time multi-vehicles hybrid simulator for heterogeneous vehicles. *IROS08: International Conference on Intelligent RObots and Systems, Workshop on robot simulators: Available Software, Scientific Applications and Future Trends*, September 2008.
[17] Thetis website. www.lirmm.fr/taipan/thetis-project , accessed on 02/09.
[18] Doxygen: A Source Code Documentation Generator Tool. http://www.stack.nl/ dimitri/doxygen/ , accessed on 10/08.
[19] L. Lapierre M. Caccia O. Parodi, M. Bibuli. Missions preparation and design of new algorithms for charlie and taipan within thetis, a hil simulator, to be accepted. *8th Conference on Manoeuvring and Control of Marine Craft, MCMC'2009*, September 2009.