Deliverable D 1.1
"Requirements Analysis Report"

Contract number: FP7-231378 Co3 AUVs
Cooperative Cognitive Control for Autonomous Underwater Vehicles

The research leading to these results has received funding from the European Community’s Seventh Framework Programme (FP7/2007-2013) under grant agreement number 231378.
Identification sheet

<table>
<thead>
<tr>
<th>Project ref. no.</th>
<th>FP7-231378</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project acronym</td>
<td>Co3AUVs</td>
</tr>
<tr>
<td>Status &amp; version</td>
<td>[Final] &quot;1.1&quot;</td>
</tr>
<tr>
<td>Contractual date of delivery</td>
<td>month 6</td>
</tr>
<tr>
<td>Actual date of delivery</td>
<td></td>
</tr>
<tr>
<td>Deliverable number</td>
<td>D 1.1</td>
</tr>
<tr>
<td>Deliverable title</td>
<td>Requirements Analysis Report</td>
</tr>
<tr>
<td>Nature</td>
<td>report</td>
</tr>
<tr>
<td>Dissemination level</td>
<td>CO Confidential, only for members of the consortium (including the Commission Services)</td>
</tr>
<tr>
<td>WP contributing to the deliverable</td>
<td>WP1</td>
</tr>
<tr>
<td>WP / Task responsible</td>
<td>ISME</td>
</tr>
<tr>
<td>Editor</td>
<td>Gianluca Antonelli</td>
</tr>
</tbody>
</table>
| Editor address | Universita degli Studi di Cassino
Dipartimento di Automazione, Elettromagnetismo,
Ingegneria dell'Informazione e Matematica Industriale
Via G. Di Biasio 43, 03043 Cassino (FR), Italy |
| EC Project Officer | Franco Mastroddi |
| Keywords | requirements |
| Abstract (for dissemination) | The deliverable consists of three reports, namely the requirement descriptions of the vehicles and the demonstrator scenarios, plus the simulator requirements analysis. This information is for internal usage only and hence confidential (and the only confidential deliverable in the project). The white paper on simulator requirements (WP2) was added though it was not foreseen as deliverable, but the information contained therein is potentially useful and of a similar nature as the hardware and demonstrator white papers. |

List of annexes (if any)

- White Paper: Demonstrator Scenarios
- White Paper: Hardware
- White Paper: Benchmarks
- White Paper: Simulator
1 Introduction

This report concerns the two scenarios envisaged for the project Co\textsuperscript{3}AUVs, funded under Grant Agreement FP7-231378. The first Section deals with a case study of Cooperative Control and Navigation of Multiple Marine Robots for Assisted Human Diving Operations (Coop\textsubscript{rob}dive), the second with a case study of harbor security.

2 Cooperative Control and Navigation of Multiple Marine Robots for Assisted Human Diving Operations (Coop\textsubscript{rob}dive)

This mission will demonstrate the efficacy of the cognitive systems developed for cooperative navigation and control of networks of Autonomous Marine Robotic Vehicles (AMRVs), working together with humans in the loop. This challenging scenario effectively paves the way for an as yet unexplored area of research and development, whereby robotic vehicles are called upon to vastly enhance the security of underwater scientific and commercial missions by humans. The demo is motivated by an operational mission scenario where a diver is tasked to examine a series of targets on the seafloor, under very low visibility conditions.
To safely maneuver under these circumstances, the diver is equipped with a pinger that emits an acoustic signal periodically (see Figures 1 and 2).

Figure 1: Cooperative Control and Navigation of Multiple Marine Robots for Assisted Human Diving Operations (Cooprobodive)

The diver starts from a simple support boat after she/he has literally *thrown* a set of small AMRVs equipped with hydrophones in the water. The AMRVs keep a desired formation pattern in the vicinity of the vertical directed along the diver and receive the pinger emissions. The AMRVs are further equipped with GPS receivers, but their clocks are not necessarily synchronized with that of the diver. Furthermore, they communicate among themselves. It is up to the AMRVs to collectively estimate the position of the diver using a suitable *triangulation algorithm* and to guide the diver along a series of paths joining the different waypoints. This will be done by implementing a path following system on board of one of the AMRVs that will instruct the diver (through an acoustic downlink) to track heading references and thus make him (her) approach and follow the successive paths. In short, path following with a human in the loop where the outer (kinematic) loop runs on the computational system installed on a selected AMRV and the inner (dynamic) loop is actually implemented by the diver in response to heading commands.

To perform this mission, it is essential that methods be developed to enable one of the AMRVs to issue heading commands without having to resort to an expensive acoustic modem. This is in line with the overall goal of being very parsimonious in the use of the acoustic communication channel. One possible choice is to divide the *world* into a finite number of heading sectors and to actually encode information - about what sector divider should steer himself
Figure 2: Cooperative Control and Navigation of Multiple Marine Robots for Assisted Human Diving Operations (Cooprob_dive) lateral and frontal views

(herself) along - in the interval between the emissions of two successive acoustic pulses. Another issue that will be addressed is the implementation of the interface between the selected AMRV and the diver (e.g. visual or by resorting to piezoelectric actuators placed inside the diver’s suit).

Systems of the kind described must necessarily be robust with respect to temporary communication losses between the diver and the AMRVs as well as acoustic outliers. To this effect, a tracker will be developed that will not only estimate the position of the diver underwater but also his (her) velocity and heading by resorting to advanced filtering structures that embody in themselves simple stochastic models that capture the expected behavior of a diver.

In the event of temporary communication losses, this system will predict the direction and speed of motion of the diver and instruct the AMRV formation to move accordingly, so as to re-acquire the acoustic link. A complementary system to be studied involves the deployment of a transponder installed in a well known location on the seabed.

In summary, in the scenario proposed the AMRV ensemble and the diver execute a cooperative path following maneuver with a human in the loop. The AMRVs must keep formation with the diver and react to her/his motion, effectively slowing down and speeding up to help the diver along the desired path. For increased visibility, it is planned to simulate this scenario in a pool with at least four surface small AMRVs and either a real diver or a target moving close to the bottom. Should the time permit it, a final demonstration is also envisioned in the Bay of Cascais (near Lisbon), Portugal close to the mouth of the Tagus river, on top of an underwater archaeological site. The extension of the methodology developed for one diver to multiple divers will be addressed in the project, albeit at theoretical and simulation levels.
3 Harbor security

The increasing interest for ocean natural resources and the importance of maritime activities for leisure, fishing and goods transportation, has brought to an increasing number of critical infrastructures to be located by the sea. These structures are usually highly complex, with many moving underwater parts, bottom installed processing units distributed over wide areas, suspended hoses and rigs, etc, and therefore highly vulnerable to malicious terrorist attacks, more general illegal and clandestine activities as well as to natural disasters. Up to now, the capability to deal with such threats is slow, dangerous and inefficient; presence of multiple obstacles, such as shipping movement, confined spaces, very shallow water (for coastal installation), and communication limitation, make ports, or more in general sea infrastructures challenging areas [2, 6], and their surveillance a hard task [4]. Anyway, recent terrorist threats have raised a new interest in addressing the security problem for civilian harbour scenarios; several marine and maritime environments have been studied and the analysis of well known threats and new possible asymmetric attacks [10, 7], have finally brought new solutions [5]. An effective surveillance has to be based on a combination of air, surface and underwater protection. Anyway, while traditional technology can be successfully applied for air/surface protection, the underwater scenario in civilian applications, requires new approaches and ad hoc solutions, since the proposed ones are usually very expensive as simple adaptations from military strategies. This document focuses on the civilian protection, and analyzes scenarios characterized by severe limitations in testing an underwater detection system in the field and by limited economic available resources. It is important to point out how the security problem has become, in the last few years,
one of the most important research topics. Due to the recent menaces against
civilian targets, several countries have started research programs on monitor-
ing and security to build new systems and to understand and study how the
existing ones can be used in new or different situations. The aims of these pro-
grams are very ambitious as they promote the development of technologies and
knowledge able to ensure the security of citizens from threats such as terrorism,
organized crime, natural and industrial disasters, and to guarantee the optimal
and concerted use of available and evolving technologies, while respecting the
socio-economic, political and cultural values. While it is clear that the security
topic is very broad and includes several and delicate issues, in this document
the focus is confined to some of the most challenging technological aspects in
one of the most difficult environments, the underwater one. Besides, it has to
be underlined that since the security research was limited to military scenarios,
an important aspect of this kind of research is related to the lack of a strong
and adequate literature and bibliography that can be used as a basis for further
studies. At the same time, even though there is a tradition of military studies,
it is not straightforward to apply them to civilian cases, because of difference of
requirements and very high costs. One key point in the civilian security theme is
in fact related to the evaluation of system performances with respect to the asso-
ciated cost, a balance very difficult to be predicted a priori without in-the-field
testing. In the underwater scenario this necessity is even more tight due to the
peculiar characteristics of the environment that makes performance prediction
very hard. In addition, even when security or monitoring systems exist for mil-
itary agencies, the associated cost is usually so high that cannot be transported
to civilian cases, and therefore new solutions, strictly based on the specificity of
the civilian scenario, have to be pursued in order to find a compromise between
performance and costs. Finally, one more point has to be clarified regarding
the security topic: the importance of robotics and automation to perform the
required tasks. When the interest is focused on civilian applications, monitoring
and security means are very limited, and in particular, the possibility to have
human operators that continuously monitor the scenario to be protected is not
realistic. Typically, one single operator is responsible for several systems and
usually for a limited time. In this context, the autonomy and automation of
the security systems assume a completely new role and importance, especially
when compared to military scenarios, in which the capability to act and react
against a threat is completely different from the civilian ones.

3.1 Security threats

Depending on the specific scenario there are different kinds of threats that can
be supposed to intrude and, in turn, once the intruder has been defined, sev-
eral characteristics of the area and hence of the surveillance systems must be
defined [1, 3, 8, 9]. Even though any kind of vessel, boat, vehicle, or diver may
be an intentional or unintentional intruder, the area itself limits the possible
type of menace because of the environment constraints: a very big vehicle cannot
enter in shallow water or in areas characterized by a very small entrance.
Even with very simple considerations it is then possible to decide a priori what
kinds of menaces are possible and most likely for the specific area. The possible
underwater threats belong to the set of AUVs, divers, divers with scooter, while
a smuggling boat can be considered as a general menace on the surface. The
previous set lists the most likely intruders, which span a wide set of possibilities according to their very different characteristics.

### 3.1.1 Autonomous Underwater Vehicles (AUVs)

An AUV is an autonomous vehicle that can travel at a speed up to 10m/s. It is generally not very big (usually less than 5 meters) and it can have room for explosive. Since it is powered by batteries, its starting point cannot be too far from its target. Anyway it can be transported by some bigger ship and deployed close to the target area. The area covered by such vehicles is about 10x10Km. An AUV can be an intelligent vehicle and implement some complex behaviour in order to avoid obstacles and move in areas with low detection probability (e.g. close to docks or to the seabed).

![Figure 4: Autonomous underwater vehicles. These vehicles can be a very dangerous threat for sea installations but also a resource for the opposite side, the anti-intrusion system, that can use them in cooperation with fixed sensors.](image)

### 3.1.2 Diver

A diver is a very slow threat, that can travel at about 1m/s, and with a very limited range (2 or 3 Km). It can be transported by a big or a small boat very close to the target where the real intrusion begins. The diver is obviously a very intelligent threat, capable to produce complex, although slow trajectories in order to hide himself and avoid sensors. It can be very difficult to detect a diver, especially when an air-recovery system is used to avoid bubbles dispersion in the water.
Figure 5: Divers are very slow threats with a very limited range but they can be very difficult to be identified especially when air-recovery circuits are used to avoid bubbles dispersion.

### 3.1.3 Diver with a scooter

The possibility to use a scooter (a swimmer delivery vehicle) increases the diver range (up to about 5 Km) and makes him twice as fast (its speed can be about 2m/s). In addition more than one intruder can be taken close to the target making the protection more difficult, since more than one intruder has to be detected and stopped.

Figure 6: Divers with a scooter are very fast intruders. Their range can be up to 5Km.

### 3.1.4 Smuggling boat

A smuggling boat is the general threat considered as a surface menace. Its range can be very long, and its speed, in very shallow water, is about 50 Km/h. It is about 10 to 15m long and can carry up to 15 people.
Figure 7: A go-fast boat, typically used for smuggling activities.

References


## Contents

1 Introduction ............................................ 1

2 Medusa’s family ................................. 2
   2.1 The Medusa Vehicles ............................... 2
   2.2 Functional Architecture ......................... 4
      2.2.1 Mission Control System ...................... 5
      2.2.2 Guidance, Navigation, and Control (GNC) System .... 6
      2.2.3 Aerial and Acoustic Communication Systems ......... 6
      2.2.4 Cooperative (Robot-Diver) Control System ........... 6
      2.2.5 Diver Interface Module  ......................... 7

3 Folaga’s family .................................. 12
   3.1 Folaga ........................................... 12
      3.1.1 Heading and pitch actuators ................. 13
      3.1.2 Buoyancy change sub-system ................. 14
      3.1.3 Movable ballast sub-system ................. 15
      3.1.4 Radio communication systems ................. 15
      3.1.5 Underwater communication system .......... 15
      3.1.6 Navigation sensors ............................ 15
      3.1.7 Software ...................................... 16

4 Simulator ........................................... 18

## 1 Introduction

This report concerns the description of the hardware required for the project Co³AUVs, funded under Grant Agreement FP7-231378. The Medusa and Folaga vehicles are briefly described, additional requirements for the simulator are given in the last Section.

## 2 Medusa’s family

The Cooprob dive mission planned in the scope of the Co³AUVs project calls for the cooperative operation of a set of semi-submerged vehicles named MEDUSA,
together with a group of divers. The final demonstration will be done with a single diver in the loop. However, throughout the project provisions will be made to study and assess in simulation the methodologies that will be necessary to develop the software/hardware architectures required for a multi-diver operation. In preparation for a description of the multiple agent architecture that will be studied and partially developed a short overview of the vehicles used is given next.

2.1 The Medusa Vehicles

This section describes the first stages of the development of the autonomous semi-submerged type of vehicles, named Medusa, that will be used in the final CoopRoboDive demonstration scenario. IST has built and operates two autonomous surface vessels named Delfim and Delfimx, and these will be used throughout the project to evaluate the performance of the systems required for multiple agent cooperation. However, in view of the objectives of simplifying the logistics of the operations at sea and building a demonstrator that would enhance the visibility of the project, it was judged appropriate to purposely develop a set of four Medusa vehicles with embedded computing, navigation/control, and communication capabilities, capable of performing cooperative tasks.

Figure 1 captures the base configuration of MEDUSA, which was modeled using SolidWorks. The vehicle consists of two acrylic pressure housing tubes, covered by suitable torpedo shape low drag casings attached to a frame of 20 mm square aluminum struts. The vehicle is propelled by two stern thrusters. The decision to use acrylic was based on the fact that they are cheap, have adequate strength, and are easy to machine. Furthermore, because they are transparent, it is simple to install a high-resolution camera inside.

The upper body (see Fig. 2) consists of a 570 mm long \( \times \) 62 mm inner diameter \( \times \) 70 mm outer diameter acrylic tube attached to the frame structure and cased with a GFRP (glass fiber reinforced plastic) or CFRP (carbon fiber reinforced plastic) torpedo shape shell that holds also floating foam rings. It houses sensors and computers, and connects to a mast (stub) with GPS, RF, and wireless antennas.

The underwater body, depicted in Fig. 3), consists of a 570 mm long \( \times \) 144 mm inner diameter \( \times \) 150 mm outer diameter acrylic tube. This tube will contain "heavy" electronic components, batteries, and sensors. The section will have removable free flooding nose and rear cones (noses) attached to the corresponding pressure end-caps of the tube. The cones improve the hydrodynamic characteristics of the vehicle and house an acoustic modem (front nose) and the thruster connectors (rear nose). To ensure that the metacentric height is sufficiently large - in order to have adequate roll and pitch stability - buoyancy foam rings are placed around the upper tube, as explained below.

The structural frame shown in Fig. 4 is based on Bosch Rexroth framing profiles that are cheap and very easy to adjust. In fact, the ability to adjust the profiles can be extremely useful during the preliminary testing phases, for it will afford system designers the versatility to move the positions of some of the key components such as the thrusters.

Both tubes are sealed at the ends by end-caps, all of them with double o-ring sealing protection. The end-caps are produced from Delrin due to the material
scratch resistance, low friction and, most important, very low water absorption proprieties. The front end-cap holds the aluminum fixing plate for the acoustic modem and for the removable nose (see Fig. 5); the read end-cap holds the two impulse micro mini wet pluggable female connectors for the thrusters, a purge plug, and the rear cone fixing structure.

The purge plug allows for a vacuum line to be attached to the rear end-cap, so that the end-caps can be secured into place simply by the difference in air pressure (between the inside and the outside of the tubes). At this stage, in order to avoid costs with several wet pluggable connectors during the development of the vehicle, it is planned for the two main tubes to be connected by an external tube. The latter will play the role of a wiring path, so that the purge plug on the rear end-cap of the underwater tube will be enough to keep all the end-caps in place and to activate all o-rings sealing.

The underwater tube will contain all the “main” electronics (PC104 board, wireless communication board and DC-DC converters), batteries, and a hi-resolution camera with associated LED bars. The upper level tube will be fitted with an acrylic stub that holds GPS and RF/Wireless antennas and a low-resolution camera.

To fix the electronics inside the hull, a custom designed internal frame that can slide in and out of the hull by removing the end-caps is under development. Table 1 contains a list of the instruments (sensors and other electronic devices) that will be installed in the vehicle. This frame holds also two battery packs (see Fig. 6) that can slide in and out, on Teflon rails, of the principal frame, allowing for very quick battery replacement without handling the complete electronics support.
Table 1: Medusa sensors and processors.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Update rate</th>
<th>precision</th>
<th>range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ocean Server OS5000 compass</td>
<td>attitude</td>
<td>0.01 - 20Hz</td>
<td>1-3° head, 2° roll &amp; pitch 360°</td>
</tr>
<tr>
<td>Tritech Micron Data Modem</td>
<td>communication</td>
<td>20 to 24 kHz</td>
<td>1 km max</td>
</tr>
<tr>
<td>Inside hull temperature sensor</td>
<td>temperature</td>
<td>Integrated in the PC104 computer board</td>
<td></td>
</tr>
<tr>
<td>GPS</td>
<td>XYZ position</td>
<td>Under evaluation</td>
<td></td>
</tr>
<tr>
<td>Hi resolution camera (underwater)</td>
<td>color image</td>
<td>Under evaluation</td>
<td></td>
</tr>
<tr>
<td>Low resolution camera (off water)</td>
<td>color image</td>
<td>Under evaluation</td>
<td></td>
</tr>
<tr>
<td>Wireless TCP/IP board</td>
<td>communication</td>
<td>Under evaluation</td>
<td></td>
</tr>
</tbody>
</table>

The propulsion system consists of two SeaBotix BTD150 thrusters mounted at the stern, on the portside and starboard sides of the vehicle; see Fig. 7. The thrusters can be adjusted in any position within the greed shadow areas shown in Fig. 8.

Preliminary drag coefficient estimates of the vehicle were obtained using CFD. To this effect, the flow around the Medusa vehicle was modelled using the commercially available finite volume code ANSYS Fluent 6.3 with a particular meshing scheme; see Fig. 9.

The estimated propulsion power required at 1 m/s is approximately 12 Watt.

### 2.2 Functional Architecture

This section describes the functional architecture adopted for the Coop robust dive mission scenario. The architecture builds upon previous related work done by IST.
in the scope of National and EC funded projects leading to the development and operation of single and multiple vehicles. Its key building blocks are identified in Fig. 11:

- Mission Control System
- Guidance, Navigation, and Control (GNC) System
- Aerial and Acoustic Communication Systems
- Cooperative (Robots-Diver) Control System
- Diver Interface Module.

In what follows we provide a brief description of these systems.

### 2.2.1 Mission Control System

The mission control system is responsible for the concerted operation of all sub-systems that run on the vehicle. Based on a mission program obtained from a
set of mission specifications defined graphically, the mission control system sequences and synchronizes the execution of the basic vehicle system tasks (single tasks and multiple vehicle/diver tasks).

2.2.2 Guidance, Navigation, and Control (GNC) System

The GNC system accepts as inputs the motion control tasks issued by the Mission Control System. Typical tasks include point stabilization, path-following, trajectory tracking, and speed and steering commands. The GNC system is responsible for generating the adequate control signals for the thrusters in order to achieve the desired motion. To this effect, the GNC system (through its navigation subsystem) estimates the position, orientation, linear and angular velocities of the vehicle by merging information provided by on-board installed motion sensors and GPS.

2.2.3 Aerial and Acoustic Communication Systems

These systems are responsible for supervising the flow of information among the vehicles, from the vehicles to a command center (on-board a support ship or a moored buoy), and between the diver(s) and the Medusa vehicles. The communications protocol will depend on the specific task/command and medium utilized and will take explicitly into account the fact that asynchronous communications, latency, and transmission failures will occur. Dealing with these issues is particularly relevant for underwater acoustic communications, which are plagued with intermittent failures, latency, and multi-path effects.

2.2.4 Cooperative (Robot-Diver) Control System

This is by far the most complex system involved in the CoopTwobdive mission. Two main blocks are needed to implement it: i) Diver Tracking and ii) Coop-
The Diver Tracking subsystem is responsible for computing the position of the diver by resorting to triangulation algorithms, based on the times of arrival of the acoustic signals (emitted by the diver) at the different vehicles. Position data fused with complementary data in a Kalman-like filter structure will enable tracking the diver. In this task, the vehicles will be required to adapt their geometric formation to the estimated position of the diver so as to maximize the information available for triangulation purposes, that is, the vehicles will effectively act as an adaptive sensor network. This is a very challenging issue, both from a theoretical and practical standpoint, especially when one addresses the tracking of more than one diver in 3D space.

Based on the information provided by the Diver Tracking system, it is up to the Cooperative Robot/Diver Motion Control System to steer the set of Medusa vehicles in reaction to the estimated diver motions and to guide them in their mission underwater.

For each diver, this will be done by providing simple high level course commands to the diver so as to make him (her) follow an agreed-upon path. In this context, the Medusa vehicles will change their geometrical formation pattern and respective “center of mass” in reaction to the divers motion, effectively speeding up or slowing down along certain directions. The implementation of this task calls for the execution of path following algorithms for each of the vehicles, where the paths are defined as the mission unfolds together with a synchronization algorithm that changes the nominal speeds of the vehicles so as to achieve the desired synchronism. From a theoretical standpoint, this problem becomes difficult when more than one diver must be “steered” underwater.

2.2.5 Diver Interface Module

This module will establish the interface between each of the divers and the Medusa vehicles. It will consist of an acoustic modem, a pressure transducer, a compass unit, a multicolor LED/plastic optic-fiber diver display interface and, a diver alarm interface. Its main functions are threefold (see Fig. 11):

1. To listen to the acoustic commands provided by the Medusa vehicles and,
upon request, to send acoustic range replies to enable the Medusa vehicles to localize the diver. It is also envisioned that the unit will emit information on its heading and depth.

2. To generate appropriate steering display signals to the diver so as to guide him (her) along the pre-planned path. This can be done by continuously comparing the requested heading (and possibly depth) received from the Medusa vehicles with the info acquired by the unit compass (and Depth Pressure Gauge) installed in the module and flashing Green and Red LEDs at appropriate rates. The color will indicate whether to turn clockwise or anti-clockwise. The rate of flashing will codify the desired turning rate.

3. To allow the diver to initiate an emergency alarm signal sent through the acoustic modem.
Figure 8: Green shadow areas: Range of possible thruster placement on this test platform

Figure 9: Flow velocity contours for drift tests at zero drift angle
Figure 10: Vehicle drag as a function of speed

Figure 11: Functional architecture of the Diver assistance with multiple Medusa Vehicles
Figure 12: Diver Interface Module
3 Folaga’s family

3.1 Folaga

Folaga is an Autonomous Underwater Vehicles (AUV) capable of carrying different kind of sensors.

Originally designed for applications just related with environmental monitoring, the current version can be employed also for missions concerning inspection and security activities, thanks to its renewed design, allowing greater manoeuvrability and more operative autonomy. In addition to FOLAGA, an integrated set of portable sensing stations has been recently developed (2008), leading to a modular underwater sensing network for performing different kinds of missions in wider or more complex areas, such as marine borders or harbours.

It was developed three different release: Folaga, eFolaga and eFolaga Plus.

Figure 13: Example of eFolaga

Folaga is a torpedo like vehicle. It is a fiber-glass water-proof cylinder connected to two wet ends hosting jet-pumps.

Figure 14: Sketch of Folaga
Main technical features of the current release

- Max length: 2519 mm
- Diameter: 135 mm
- Weight: 30 Kg
- Speed: 2 knots (up to 4 knots if required)
- Control: pitch/yaw thruster, movable ballast, active buoyancy control
- Energy Storage: NiMh Batteries 12 Volt 45 Ah
- Endurance: 6 hours at max speed
- Maneuverability: any bearing and trim with no active surfaces
- Gliding Scope: 0 - 50 m
- Max depth: 80 m (underwater navigation)
- Software: Windows Command and control interface
- Communication: multi-radio Link (when surface) acoustic modem (optional)

The on-board electronics for sensors, Guidance, Navigation and Control (GNC) and communication find space inside the cylinder. It is based on PC-104 boards and, in its basic version, is constituted by the following equipments: compass, inclinometers, depth meter, GPS receiver, GRSM/dedicated radio link (an antenna for user station communication and GPS signal reception is located at the stern). Folaga is characterized by a very high maneuverability even at low speed, including hovering capabilities, that may be exploited in several inspection and patrolling tasks. The motion in the surge direction is due to propulsion jet-pumps at the vehicle stern, while steering in surgesway plane is obtained through two lower power jet-pumps at the vehicle bow, transversal to the surge direction. Vehicle diving is obtained by combination of a buoyancy change and attitude change. More precisely vehicle buoyancy is controlled through a ballast chamber in which water can be injected or ejected, while attitude is controlled through the internal displacement of the battery pack by a wormscrew mechanism. Both buoyancy and attitude are trimmed at the beginning of the mission in order to have the vehicle neutrally buoyant with 0° pitch. The combined use of buoyancy and attitude change allows the vehicles to dive in several different ways: from vertical dive with 0° pitch (preferred diving mode for oceanographic data profiling), to combined attitude change and surge propulsion (keeping the vehicle neutrally buoyant), to combined attitude and buoyancy change (with or without propulsion), through the presence of a ballast chamber and the wormscrew mechanism. It is clear that Folaga mixes actuation mechanisms that are similar to those of oceanographic gliders and of self-propelled AUVs, but the resulting motion and functionalities are different from both. In particular, differing from conventional AUVs, the absence of thrusters and steering control surfaces allows Folaga to perform fine maneuvers even on position, without needing hydrodynamic lift.

### 3.1.1 Heading and pitch actuators

Nose section:
- 2 jet-pump driven by 12 Vdc motor. No feedback. Heading control system

Rear section:
- 1 jet-pump driven by 12 Vdc motor. No feedback. Pitch control system
1 propeller 50mm dia driven by 12 Vdc motor. Bi-directional action. No feedback. Forward and reverse main propulsion system.

3.1.2 Buoyancy change sub-system

The vehicle buoyancy is controlled through a ballast chamber in which the sea water can be injected or ejected.

Performance specification:
- Max buoyancy variation: 350 gr.
- Resolution: 10 gr.
- Output: analog, voltage
- Max. depth: 80 m
3.1.3 Movable ballast sub-system

The COG position can be changed through the internal displacement of the battery pack by a worm screw mechanism.

Performance specification:
- Moved mass: 12Kg
- Max displacement: 80 mm
- Velocity: 1 mm
- Output: encoder

3.1.4 Radio communication systems

In the surface navigation mode, Folaga can be connected with the ground station through the following channels:
- WIFI connection (for short-range communication application).
  - RF Band: 2.4 Ghz
  - Range: 100 m
  - Baud Rate: 10 Mbit
- Radiomodem connection (for medium-range communication application).
  - RF Band: 2.4 Ghz
  - Range: 1Km
  - Baud Rate: 19.2 Kbit
- GPRS Connection (for long-range communication application).
  - RF Band: 900/1800/1900 MHz
  - Range: depending on geographical coverage area of the network
  - Baud Rate: 56 Kbit

3.1.5 Underwater communication system

Micromodem

The WHOI Micro-Modem is a small-footprint, low-power acoustic modem based on the Texas Instruments TMS320C5416 DSP.
- Power transmit: < 50W
- Frequency: 25KHertz
- Data Rate: 80-5400 bps
- Range: 4 Km

3.1.6 Navigation sensors

GPS receiver
- 12-channel low power Fastrax iTrax03
- Support for active +3.3/5V antennas
- Fastrax iTalk Binary or ASCII NMEA protocols
- L1 frequency C/A code (SPS)
- Update rate 1Hz, user configurable up to 3Hz
- Accuracy 1.2m CEP 95%, velocity 0.1m/s, time 20ns

Compass/inclinometer
- Heading Specifications
- Accuracy (RMS): 0.5°
- Max Dip Angle: 85°
- Resolution: 0.1°
Tilt Specifications
Pitch Accuracy: 0.2°
Roll Accuracy: 0.2° (Pitch < 65°)
Tilt Range: ± 80°

I/O Specifications
Communication Rate: 300 to 115200 baud
Maximum Sample Rate: 20 samples/sec

Depth meter
range: 0-100m
Output: analog (0-5V)
Resolution: depending on ADC device

Other sensor
Internal temperature
Humidity
Battery voltage

![Figure 18: Additional sketch of the Folaga](image18.jpg)

![Figure 19: Folaga’s detail](image19.jpg)

### 3.1.7 Software

The hardware architecture is based on a two-level structure, one stack composed by a PC104 with CPU, communication module and GPS aimed the mission handling and one low level stack implemented on proprietary boards composed
by DSP and microcontrollers. The higher level stack is implemented via C-modules under Linux, the low level stack is also implemented via C-modules without operative systems.
4 Simulator

1. High Fidelity 3D Simulation

   (a) 3D Physics: First and foremost, the simulator has to support realistic physics, especially as basis for

      i. Sensor Models: Acoustic range sensors are a very important sensor type in the underwater domain. They are not easy to simulate due to their typical wide beam character. Efficient ray tracing operations - that simulate pencil beam range measurements that again can be combined to simulate wide beams - are hence of tremendous importance. Also, standard sensors like Inertial Measurement Units (IMU) and motor control feedback, i.e., encoders, should be made available. Furthermore, cameras can be of interest, which is linked to the visualization requirements. Other sensors, especially standard ones measuring chemical and physical environment quantities, are of concern in the context of environment modeling.

      ii. Actuator Models: The high fidelity physics simulation of underwater actuators is a critical point in the context of the project as depending on the level of accuracy, the simulation may easily lead to computational requirements that make multi robot experiments, respectively experiments in complex environment settings unfeasible. Accordingly, a compromise with the requirements is striven for. The actuator modeling is mainly integrated in the vehicle dynamics to give realistic AUV motions.

      iii. Vehicle Dynamics and Environment Interactions: The vehicle dynamics as overall result of all actuator activations must be realistic and nevertheless computational feasible. Also, it is required to be able to develop different vehicles. Different core components, especially with respect to sensors and actuators have to be provided for which it should be able to combine them into a specific AUV.

   (b) 3D Visualization: Good 3D visualization is important for several reasons. First of all, it allows visual inspection of the performance and the results. This also allows to generate videos for demonstration and outreach. Second, it is strongly linked to important sensors. Especially, the simulations of cameras to test vision based approaches need a good 3D graphics. Also, non-vision operations, like ray-tracing for as basis for sound ranging, are often linked to the graphics engine.

   (c) Modeling Tools (for Components): The physics and visualization engines must be supplemented by suited modeling tools that allow an easy integration of components, respectively the (core) design of new ones, especially with respect to their basic properties like geometry, texture, etc.

2. Multi Robot Simulation: The project deals with cooperative cognitive control, support for multi robot simulation is accordingly a must. This leads especially to two more concrete sub-requirements
(a) Computational Efficiency: All other requirements must be evaluated against this criterion. High efficiency is necessary to allow for multi-robot experiments. Simulation features that may mitigate this requirement are

i. Hardware Acceleration: The simulator’s 3D visualization, respectively physics engine should ideally be capable of using hardware acceleration.

ii. Distributed Simulation: Support for the distribution of the simulation over different computers is a conceivable - but also non-trivial - feature. At least, multi-threading and multi-processor machine should be fully exploited by the engines.

(b) Communication: Coordination of multi-robot systems usually requires some form of communication, which is non-trivial in the underwater domain. The ideal simulator should provide some means to have at least some basic features for simulating underwater communication.

3. Complex Underwater Scenarios: The demonstration of Cognitive Control requires complex scenarios where the robots have to engage in non-trivial operations and decision making. To generate according scenarios, following sub-requirements are of interest

(a) Modeling Tools (for Scenarios): Much like the modeling of AUV components, especially sensors, actuators, and structural parts, there is the need for support to generate complex underwater environments, especially with respect to geometry, textures, etc. Again, this has to be linked to the 3D graphics/physics engines used for the simulator.

(b) Import of Ground Truth Data: Simulation scenarios should be as realistic as possible. The use of ground truth data, e.g., bathymetric maps of certain areas, etc., is an interesting option in this respect.

4. High Distribution Grade and Wide Access:

(a) Community Driven: Ideally, the development of the simulator, respectively of its components should not only come out of this project, respectively be only used within the project. After all, the contributions should be beneficial to the robotics community at large. It is hence of interest to consider options that are embedded in a larger groups of people who contribute to the developments, promote their usage, and also make actual use of them for scientific experiments themselves.

(b) Open Source: An open source solution would be ideal to allow a wide dissemination and open access to the results. This requirement nevertheless has to be balanced with the other ones, especially with respect to performance.

5. Feasibility within the Project’s Timeframe and Budget: Last but not least, the feasibility of the implementation has to be kept in mind. Though certain ultimate solutions may be conceivable if unlimited resources were available, the main objective is a working solution within the project’s possibilities.
1 Introduction

In recent years autonomous robotic systems received a large attention from several, partially overlapping research communities. Those range from control theory, to artificial intelligence, to cognitive sciences, to philosophy of science [31]. The capability to measure the performance of intelligent systems is not simply an academic exercise, a metrics is necessary to quantify the progress, evaluate the results or compare different approaches. This need is growing together with the complexity and the abstraction of the missions of the robotic systems [40]. Science and engineering are virtually impossible without quantitative measurements or the repeatability of the experiments. The latter is in particular a sensitive problem since robotics is experiencing in the last years a structural distancing from the possibility to replicate experiments of other researchers; this concept may be clarified with an example. Some 20 years ago part of the research in robotics was devoted at developing dynamic control laws for industrial manipulators; every researcher, by simply reading a publication had all the information required to replicate in her/his own laboratory the experiments and to eventually compare several controllers. The possibility to replicate and compare is virtually disappeared in the today’s publications since it is too hard to reproduce exactly the same algorithm (very often the publication is missing implementation details) and to interact in the same way with the world.

It is thus clear that the original benchmarking methodology, developed for industrial manipulators, can not be simply extended to autonomous robotics for several reasons. Some questions remain unsolved: how can we assess the current state of the science and technology? how can a buyer evaluate the advantages and disadvantages of different solutions?

One of the first test aimed at evaluating the intelligence of a machine was proposed by Turing [41] and consists in interacting with a keyboard (chatting) with a computer. If the human is not able to detect the original nature of the interlocutor, i.e., human or digital, than the system can be considered as intelligent. This test is obviously not generalizable to autonomous robotics but it also contains an additional subtle: the human itself is used as a metrics for the performance.

Reference [40] provides an interesting introduction to the problem of establishing good experimental methodologies in robotics, citing the seminal work of Popper in establishing when a discipline can be considered as scientific or not. The Authors also pose non-trivial questions about the eventual possibility
that, being autonomous robotics the more a complex system, in the future the corresponding experimental validation procedures may be inherited from social sciences, biology or medicine.

Traditional benchmarking methodologies are based on the assumption of static environments and deterministic or \textit{simple} stochastic analysis. A key aspect of the scientific research is the repeatability, given a certain theory it should be possible, for a different researcher in a different laboratory, to exactly reproduce the results forecast by the theory. When dealing with autonomous robotics, however, this concept starts to become fuzzy, since the environment, rather than the robots itself, is not repeatable. As a result, the traditional benchmarking methodologies fail since for an autonomous robot it is impossible to have the same light condition, the same obstacle position, sensor readings and, in case required, the same human interaction. In a sense, repeatability is incompatible with a truly autonomous, unstructured mission: repeatability of autonomous robotics can be seen as an oxymoron. The discussion on the experience of a misunderstood \textit{emergent} behavior of autonomous robot is not new, one significant example is given by the considerations of Arkin [28] in His seminal book on behavioral robotics. The emergent behavior is not caused by a kind of \textit{magic} property of the behavioral control but rather by the interaction of the algorithm with the environment. These considerations need to be clear before approaching the task of benchmarking the outcome of the research in autonomous robotics since there is more and more demand to provide quantitative benchmarks for autonomous systems.

In [33], the specific aspect of robot-human interaction is considered. Three metrics are proposed to evaluate the level of interaction, the level of autonomy and the world complexity. A particular attention is given to the first two metrics and some experimental results where different interaction schemes are compared are proposed.

An additional consideration to be made concerns the necessity to simultaneously consider hardware and software of a robotic system. Differently from pure computer science, and similarly to the biological life, what can be done with an autonomous robot is also function of its hardware equipment in terms of locomotion, sensing, computational capabilities and so on. Roughly speaking, a navigation algorithm, for example, can never be tested without considering the specific sensor and actuators it was designed for.

A common benchmarking method for robotic algorithms is given by the use of numerical simulations. The advantages of simulations are numerous, among them the possibility to run experiments with defined and repeatable conditions. However, real experiments always differ from simulation. Sophisticated sensors such as cameras or laser scanners can only be simulated up to a certain level of accuracy, therefore, using real robotic data sets is favored for benchmarking.

This paper provides a survey ans some considerations on benchmarking in service robotics with specific attention for the marine robotics class of missions.

\section{Toward autonomous robots}

In industrial robotics some strict performance criteria exist, see, e.g., the ISO 9283:1998, "Manipulating Industrial Robots Performance Criteria and Related Test Methods" that need to be fulfilled by robot manufacturers. However, it is
well known that, in practice, the robots end-users require to the manufacturers additional specific tests to be accomplished before ordering the machine. Those tests mainly replicate the task that the new robot is supposed to perform and the current one is not able to do; they mainly concern trajectory planning, kinematic and dynamic control performance in term of accuracy and repeatability. It is easy for the buyer to design the requirements and for the manufacturer to set up an experimental facility and then to demonstrate eventual fulfillment of the given requirements.

Autonomous robotics can not follow the same approach due to the presence of a large number of complex sensors, the presence of the environment and, eventually, the interaction with humans. Few robots can be considered as autonomous for very simple tasks such as, e.g., the vacuum cleaner’s ones. Interesting enough, the Wikipedia community developed an elementary comparison page among the models [22]; if necessary, this reinforces the opinion that benchmarking for autonomous robots is necessary.

Since autonomous robots require the interaction among several subsystems, a common engineering procedure is to firstly test all the elementary modules before proceedings toward a test of their assembly. It is easily understandable, for example, that navigation and mapping influences one each other, in the sense that navigation benefits from good maps while a proper path may improve the mapping phase.

The research community developed numerous and good benchmarking procedures for most of the required elementary tasks. Not surprisingly, most of the criterias developed satisfy the repeatability features discussed in the introduction, it is the case, e.g., of vision algorithm that may be developed by resorting to manually annotated database for the validation and comparison of their performance. In the following, a small review of some submodules benchmarking is performed.

2.1 Benchmark of the elementary tasks

Elementary tasks of a robot start from minimal mobility or sensing capabilities. Beside some obvious requirements such as, e.g., to track a desired trajectory for the single wheel of a wheeled robot or to correctly measure the relative distance between sensor and obstacle for an sonar, it is possible to device some higher level tasks that, however, still can be considered as elementary from the perspective of an autonomous mission. Examples are:

- heteroceptive and proprioceptive sensing;
- perceive what sensed;
- selectively remember what sensed;
- forecast future evolution of the scene;
- make decisions;
- learn from the experience;

The key issue is the possibility to measure each of these capabilities in terms of accuracy, speed, computational load, cost/benefit ration. Some discussions about the most usefull tasks are reported in next sections.
2.1.1 Localization and/or Mappings

When Simultaneous Localization And Mappings (SLAM) is of concern one common benchmark is to command a closed loop to the vehicle and then to verify that the generated map and localization is consistent with the human-observed circular path. This choice is due to the absence of a ground truth since both the map and the localization are being built during the experiment. In [38] a metrics is proposed that makes use of a reference map, acquired and used as a performance evaluator only and not for the SLAM implementation.

It must be noted that an exact metric measure of the SLAM (or map) accuracy is not always the best output depending on the computational/economical cost necessary to achieve an accurate measurement. Depending on the scope, in fact, a topological map may be more appropriate as input for the robot’s decision algorithm.

A community-driven web site that makes available data for comparison purposes is [32]. An European project is [10], funded with the aim to provide a comprehensive, high-quality benchmarking toolkit. The benchmarks are focused on the problems of sensor data analysis, sensor fusion, localization, mapping and SLAM.

The Robotics Data Set Repository (Radish for short [25]) provides a collection of standard robotics data sets that contains:

- Logs of odometry, laser and sonar data taken from real robots;
- Logs of all sorts of sensor data taken from simulated robots;
- Environment maps generated by robots;
- Environment maps generated by hand (i.e., re-touched floor-plans).

2.1.2 Motion planning

As noticed by the sub-area of motion planning [16] of the EURON [4] Special Interest group in Good Experimental Methodology and benchmarking [14], no common set of benchmarks is available yet in motion planning for autonomous robots, but the effectiveness of each tool is measured by means of a specific set of problems, which however cannot be shared by any other tool due to incompatible or proprietary formats and characteristics.

In [36], however, some quantitative metrics are proposed for the specific problem of robot navigation. The Authors explicitly develop a experimental/simulation testbed so that the same code can be run in numerical simulation and in the real world. The latter is an approach that is now going to be diffuse in the robotics community and allow to appreciate the robustness of the code when moving from numerical to real scenarios. Concerning the navigation problem, the Authors recognize several key issues:

- the environment;
- the quantity of a-priori information of it owned by the robot;
- the energy consumption;
- the time limit;
• robot reactivity and robustness.

All the items above are self-explaining except the reactivity, intended as the sum of the detection and processing time. Among the various metrics proposed for the navigation mission, the world complexity deserves attention; two indicators are proposed, the global complexity and the local complexity both inherited from information theory. In addition, the a-priori knowledge is also measured via a conditional entropy concept.

2.1.3 Obstacle avoidance

The work in [39] is totally devoted at investigating the problem of evaluate obstacle avoidance approaches, one key aspect is the numerical simulation of the approach under evaluation for a wide range of working conditions or scenarios.

2.1.4 Heteroceptive systems

Among the heteroceptive sensors, vision is one of the most diffused due to its versatility and economicity. The community of vision experts developed numerous benchmarking initatives, some discussed in section 3, within EURON [4], the Special Interest group in Good Experimental Methodology and benchmarking [14], developed as core sub-area also the Visual Servoing initiative [18] that came out with some practical proposals including

• Measuring the success of the algorithms (convergence with respect to several starting positions);

• Measuring the computational cost of the algorithms;

• Measuring the behavior of the algorithms Interface window in the JaViSS visual servoing simulation environment.

2.1.5 Manipulation and grasping

Within EURON [4], the Special Interest group in Good Experimental Methodology and benchmarking [14], developed as core sub-area also the manipulation and grasping initiative [15].

2.1.6 Communication

Communication in robotics is conceptually strictly connected to the networked robotics area, also identified within EURON [4], by Special Interest group in Good Experimental Methodology and benchmarking [14], as a core sub-area that deserves attention to set up benchmarking procedures [17].

2.1.7 Other tasks

Additional tasks may require mobility locomotion, especially for all-terrain robots, or software dependability. A conclusion of [30] is that the availability of open source algorithm implementations, data sets, and simulation environments is the key to promote accelerated research in autonomous robotics.
The work [35] introduces the concept of benchmarking for cooperative robots, i.e., for problems that need to be solved by multiple robotic systems that explicitly require a kind of negotiation. The discussion starts with a simple case study involving two robots and for the obstacle avoidance case.

3 Some benchmarking examples

It is interesting to report the examples of existing benchmarking initiatives.

An example of performance measurement is given by the computer vision community. There are several projects aimed at providing image databases to other researchers [29, 5]. These image databases are accompanied by ground truth images and functions to calculate performance metrics. The main subtopics interested by this approach are image segmentation and object recognition [38]. A list of nine annotated databases can be found in Wikipedia, in a page related to object recognition [23]. Similar initiatives are also listed concerning similar topics such as facial or speech recognition.

Since 2000, the Intelligent Systems Division of the US National Institute of Standard and Technology (NIST) is organizing the PERMIS (PERformance Metrics for Intelligent Systems) workshop [7]. In addition, specific programmes are devoted to testing, e.g., the autonomy of mobile robots by developing a test arena where run specific competitions.

ROSTA (Robot Standards and Reference architecture) [12] is a Coordination Action funded under the European Unions Sixth Framework Programme (FP6) whose aim is to becoming the main international contact point for robot standards and reference architectures in service robotics. Specific objectives of ROSTA are: a) Creation of a glossary/ontology for mobile manipulation and service robots; b) Specification of a reference architecture for mobile manipulation and service robots; c) Specification of a middleware for mobile manipulation and service robots; d) Formulation of benchmarks (of components, methods, middleware and architectures) for mobile manipulation and service robots [13].

What makes the difference among the robotics community and the, e.g., computer vision one, is the impossibility, for the former, to consider structured, static, environments. Autonomous robots need to face dynamic, unstructured situations where possibly interact with humans. The possibility to turn it into a repository where researchers might upload their algorithms and compare their performance is not straightforward and, at the best of our knowledge, has not been implemented in significant case study.

4 Benchmarking the mission performance

The benchmark of the service or elementary tasks is obviously important but it does not solve the question; what it is really needed is to be able to benchmark the overall mission performance. From one hand this higher abstraction level makes it difficult to quantitatively define the mission performance itself, on the other hand, it is intrinsically related to the fuzzy concept of mission. Newell [37] proposed a list of abilities such as:

- contextualize a scene;
• construct a correct response from the perceived situation;
• form a meaningful and comprehensible sentence of the selected response;
• represent a situation internally;
• be able to actively discover relevant knowledge.

Without further specifying the environment, the robotic system, the mission, etc., it is difficult to refine further such a definition. An additional contribution is given by Albus [26]: “the ability of a system to act appropriately in an uncertain environment, where appropriate action is that which increases the probability of success, and success is the achievement of behavioral subgoals that support the system’s ultimate goal” where implicitly the concept of probability enters into the definition.

It can be recognized that the definitions above, given for the concept of *intelligence* in a wide sense, suggests a human level of cognition. However, since the road to a human-like intelligence seems to be long, benchmarking of autonomous robots requires more practical definition and approaches. At the best of our knowledge, no automatic system is able to contextualize a scene, even if static and without time or computational constraints, with human-like robustness.

Recent developments in robotics pushed towards the merging of several disciplines such as, e.g. robotics, artificial intelligence, cognitive science, neuroscience, biology, psychology, cybernetics and forced the researchers in finding a common language. Concepts such as cognitive robotics, awareness, cognition, still do not have an established definition within the various communities. A wide enough definition might be the need to design and use robots with human-like capabilities in perception, control and cognition [34].

Given these premises it is obviously hard to have widely accepted benchmarks in the community that might represent a test bed for the research progress.

The benchmark, thus, might be designed to be a comparative one, instead of an absolute criteria. In other words, it might be possible to compare different approaches in the same scenario performing a competitive mission. This leads naturally to the concept of *robot competitions*.

### 4.1 Robot competitions

The Special Interest group in Good Experimental Methodology and benchmarking [14], within the European Union Network of Excellence on Robotics (EU-RO N [4]) has the opinion that *A benchmark can only be considered successful if the target community accepts it and uses it extensively in publications, conferences and reports as a way of measuring and comparing results. The most successful benchmarks existing today are probably those used in robot competitions.*

Within the robotics community the idea of directly comparing the algorithms is gaining popularity with the robots competitions. Examples are Robocup [11] (in the 2008’s edition 400 teams, 2000 participants from 35 countries/regions), or the Darpa Urban and Grand Challenge [2] (2 million of US dollars price in 2008). A similar project, with both military and civilian applications, has been also developed in Europe with the name of ELROB [3].
A different kind of competition is given by the challenge of search and rescue [24, 19] that has been recognized as an important test to verify the performance of truly autonomous systems. It is worth noticing that both the Grand Challenge and the Search and Rescue context allow to stress also the mobility/locomotion capabilities of the robots.

Rat’s life [8] was an European project that defined a full benchmark including a Webots-based simulation and a real world setup based on the e-puck robot and LEGO bricks. It was mainly focused on autonomous robot navigation (including visual landmarks and energy management). One of the interests of this project was that the virtual contest run online every day for several months. A version of the contest still is running at [9].

The advantages of performing a competitions are given by the possibility to test the level of system integration and the engineering skills of a team. On the other hand, the competition does not allow to measure the performance of a subsystem or a single algorithm. A competition, thus, may be considered as the final test of an autonomous robotic system that already passed individual tests for each of the subsystems.

5 The marine singularity

Marine robotics presents major challenge with respect to land robotics due to the poor performance of the actuation and sensing systems and due to the major cost, both economic and human, in performing on-field tests (also defined wet tests).

The databases initiative listed above do not include also data acquired in an underwater environment such as, e.g., sea bottom images or data specific for the marine environment such as side scan sonars. Even testing of the elementary modules, thus, can not benefit from similar initiatives already developed within the robotic community.

In the literature, few papers devoted at the argument can be found, in [42] the performance metrics specific for oceanographic survey is address; the presence of battery life and limited velocity impose some constraints on the survey domain in particular for what the spatial and temporal survey resolutions are concerned. In the cited paper, a quantitative index is proposed that take into account both spatial undersampling and temporal evolution of the sample field. The proposed metrics can also be used to estimate the survey region that can be successfully investigated within the given constraints and within the desired error. The work [27] reports a metrics discussion specific to the navigation case of AUVs, the paper proposes an index, defined navigation efficiency which is aimed at correlate that the navigation performance to the given sensor equipment and survey requirements. The developed theory, however, only concerns AUVs performing missions alone, i.e., not in a coordinated fashion.

Some robotics competitions are also organized for autonomous surface or underwater robots. The AUVSI (Association for Unmanned Vehicle Systems International) and ONR (Office of Naval Research) are organizing, in 2009, the 12th International Autonomous Underwater Vehicle Competition [21] and the 2nd International Autonomous Surface Vehicle Competition [1]. MATE (Marine Advanced Technology Education Center) organizes regularly competitions for ROVs (Remotely Operated Vehicles). At European level, it is possible to find
the Student Autonomous Underwater Challenge - Europe (SAUC-E) [20] organized by the Defense Science and Technology Laboratory of the UK Ministry of Defense. Another competition is given by the Microtransat Challenge that is specific for fully autonomous sailing boats.

It is worth noticing that the competitions are often addressed to students and concern simple tasks that can be afforded in confined spaces or even pools, in conditions, thus, much better than open sea. The distance of the competitions' level from the current research conducted in the major international laboratories is currently large and the practical benefits of the competitions to the community still need to mature.

6 Conclusions

Benchmarking for marine robotics still is in an embryonic stage, it might be useful, thus, to list a possible number of best practices to set up when affording a research project in this area:

- The establishment of agreed repositories of sensor data of the kind of the Robotics Data Set Repository [25] allows different groups to execute different algorithms on exactly the same data set, thus outlining strengths and weaknesses. This is missing in the marine robotics community, that might benefit from a project of the kind of [10] focused on marine-related sensor data;

- The open-source philosophy is gaining popularity with the creation of repositories of pieces of code for common tasks [32]. Some initiative can be found in, e.g., the Marine Systems Simulator [6] developed at the Norwegian University of Science and Technology mainly by Thor Fossen and Tristan Perez. One of the outcome of the project Co3AUVs will be the release to the community of a simulator that can be considered as a small step toward that direction;

- Numerical simulations are obviously the main road to travel when designing any kind of algorithm concerning autonomous marine robots. Specific care should be given to the mathematical model used for any single component of the forthcoming experimental set-up. For example, while a dynamic model with concentrated parameters may be proper for low speed and far from the surface AUVs it becomes inaccurate for high speed surface vehicles. The same arise for sonar acquisition or underwater communication, it is usually inappropriate to simply model a stochastic noise superimposed to a pure signal to model the communication channel. A delicate balance between modeling accuracy and computational complexity should be found.

- Hardware-in-the-loop tests should be scheduled; those concern the real hardware connected to a simulated environment. For the marine environments it is very hard to properly modeling the robot-environment interaction, those tests, thus, should be mainly considered as useful for debugging purposes more than for gains tuning.
Wet tests should be properly designed in order to take into account the larger cost, both human and economic, of affording such kind of experimental robotics. In particular, the code should be written so as a modular performance evaluation should be possible by, e.g., isolating the various submodules and trying to validate their efficacy independently.

Given the discussion of this paper, it is clear that the definitive benchmark is the mission’s accomplishment. The mission’s design itself is part of the game, in the sense that a proper mission needs to be developed according to the human and economical power of the researchers.

References


[34] K. Kawamura. Synthetic approach to cognitive systems: A perspective from
cognitive robotics. In Workshop on Benchmarks in Robotics Research, 2007
IEEE/RSJ International Conference on Intelligent Robots and Systems,


[36] A. Lampe and R. Chatila. Performance measure for the evaluation of mobile
robot autonomy. In Robotics and Automation, 2006. ICRA 2006. Proceed-

[37] A. Newell and CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT
OF COMPUTER SCIENCE. The knowledge level. Department of Com-

[38] J. Hertzberg O. Wulf, A. Nüchter and B. Wagner. Benchmarking ur-
ban 6D SLAM. In Workshop on Benchmarks in Robotics Research, 2007
IEEE/RSJ International Conference on Intelligent Robots and Systems,


[40] J.L.J.I. Rano and J. Minguez. Good Experimental Methodologies in

460, 1950.

[42] JS Willcox, JG Bellingham, Y. Zhang, and AB Baggeroer. Performance
metrics for oceanographic surveys with autonomous underwater vehicles.
Whitepaper

Co³-AUVs Simulator
Requirements and Design Choices

Soeren Schwertfeger, Max Pfingsthorn, Ravi Rathnam, and Andreas Birk
Jacobs University Bremen
Campus Ring 1, 28759 Bremen, Germany
a.birk@jacobs-university.de

1 Overview

The simulator will play an important role in the project. It is designed with a double objective, first to be used as dynamic simulator and hardware-in-the-loop tests, second as an appealing 3D graphical animation in order to enlarge the outreach of the results to a non technical audience. The core part of the simulator will be based on available visualization and physics packages, which have to be adapted to the task of (underwater) robotics simulation. Using according packages facilitates the development of the simulator and allows for appealing graphics in combination with good physics. It also eases the challenge that the simulator has to be very efficient to allow for multi robot experiments.

In the following section 2, an overview of the requirements for the simulator is given. The overall combination of the requirements suggests to look for a basis of the simulator, which makes use of existing engines. Section 3 discusses a few possible choices as basis for the simulator.

2 Requirements

1. High Fidelity 3D Simulation
   
   a. 3D Physics: First and foremost, the simulator has to support realistic physics, especially as basis for
      
      i. Sensor Models: Acoustic range sensors are a very important sensor type in the underwater domain. They are not easy to simulate due to their typical wide beam character. Efficient ray tracing operations - that simulate pencil beam range measurements that again can be combined to simulate wide beams - are hence of tremendous importance. Also, standard sensors like Inertial Measurement Units (IMU) and motor control feedback, i.e., encoders, should be made available. Furthermore, cameras can be of interest, which is linked to the visualization requirements. Other sensors, especially standard ones measuring chemical and physical environment quantities, are of concern in the context of environment modeling.
      
      ii. Actuator Models: The high fidelity physics simulation of underwater actuators is a critical point in the context of the project as depending on the level of accuracy, the simulation
may easily lead to computational requirements that make multi robot experiments, respectively experiments in complex environment settings unfeasible. Accordingly, a compromise with the requirements is striven for. The actuator modeling is mainly integrated in the vehicle dynamics to give realistic AUV motions.

iii. Vehicle Dynamics and Environment Interactions: The vehicle dynamics as overall result of all actuator activations must be realistic and nevertheless computational feasible. Also, it is required to be able to develop different vehicles. Different core components, especially with respect to sensors and actuators have to be provided for which it should be able to combine them into a specific AUV.

(b) 3D Visualization: Good 3D visualization is important for several reasons. First of all, it allows visual inspection of the performance and the results. This also allows to generate videos for demonstration and outreach. Second, it is strongly linked to important sensors. Especially, the simulations of cameras to test vision based approaches need a good 3D graphics. Also, non-vision operations, like ray-tracing for as basis for sound ranging, are often linked to the graphics engine.

(c) Modeling Tools (for Components): The physics and visualization engines must be supplemented by suited modeling tools that allow an easy integration of components, respectively the (core) design of new ones, especially with respect to their basic properties like geometry, texture, etc.

2. Multi Robot Simulation: The project deals with cooperative cognitive control, support for multi robot simulation is accordingly a must. This leads especially to two more concrete sub-requirements

(a) Computational Efficiency: All other requirements must evaluated against this criterion. High efficiency is necessary to allow for multi robot experiments. Simulation features that may mitigate this requirement are

i. Hardware Acceleration: The simulator's 3D visualization, respectively physics engine should ideally be capable of using hardware acceleration.

ii. Distributed Simulation: Support for the distribution of the simulation over different computers is a conceivable - but also non-trivial - feature. At least, multi-threading and multi-processor machine should be fully exploited by the engines.

(b) Communication: Coordination of multi robot systems usually requires some form of communication, which is non-trivial in the underwater domain. The ideal simulator should provide some means to have at least some basic features for simulating underwater communication.

3. Complex Underwater Scenarios: The demonstration of Cognitive Control requires complex scenarios where the robots have to engage in non-trivial operations and decision making. To generate according scenarios, following sub-requirements are of interest

(a) Modeling Tools (for Scenarios): Much like the modeling of AUV components, especially sensors, actuators, and structural parts, there is the need for support to generate complex underwater environments, especially with respect to geometry, textures, etc. Again, this has to be linked to the 3D graphics/physics engines used for the simulator.

(b) Import of Ground Truth Data: Simulation scenarios should be as realistic as possible. The use of ground truth data, e.g., bathymetric maps of certain areas, etc., is an interesting option in this respect.

4. High Distribution Grade and Wide Access:

(a) Community Driven: Ideally, the development of the simulator, respectively of its components should not only come out of this project, respectively be only used within the project. After all, the contributions should be beneficial to the robotics community at large. It is hence of interest to consider options that are embedded in a larger groups of people who contribute to the developments, promote their usage, and also make actual use of them for scientific experiments themselves.
(b) Open Source: An open source solution would be ideal to allow a wide dissemination and open access to the results. This requirement nevertheless has to be balanced with the other ones, especially with respect to performance.

5. Feasibility within the Project's Timeframe and Budget: Last but not least, the feasibility of the implementation has to be kept in mind. Though certain ultimate solutions may be conceivable if unlimited resources were available, the main objective is a working solution within the project's possibilities.

3 An Overview of Possible Simulation Engines

3.1 Commercial Robotics Related Simulators

3.1.1 Cyberbotics Webots

http://www.cyberbotics.com/products/webots/

Webots [Mic04] is a commercial robotics simulation package which consists of a complete tool chain to create and simulate robots in simple environments. A number of existing robot models are shipped with the software and it seems to me mainly focused on educational applications.

Webots may be very hard to extend to allow decentralized simulation of many AUVs in complex environments, even though modeling a single AUV along with required sensors would be simple given the available tools.

Characteristics:

**Graphics:** OpenGL

**Physics:** ODE (open source)

3.1.2 Microsoft Robotics Studio


MS Robotics Studio is a recent simulator built on a service oriented architecture within Microsoft’s .Net Runtime Environment and thus only runs on Windows. It provides many robot models used in university labs and can either interface with the real system or with a simulated robot. It is by nature distributed, as each service, for example a mapping service, potentially runs on its own computer.

While the source code is not available, it seems easy to extend the components within MS Robotics Studio to allow synchronization between multiple simulations and thus fine grain multi robot scenarios. However, the simulation is rather focused on providing a quick way to test algorithms rather than an efficiently simulated large unstructured world.

Characteristics:

**Graphics:** Direct3D

**Physics:** Nvidia PhysX
Figure 1: Screenshot of Webots

Figure 2: Screenshot of Microsoft Robotics Studio
3.1.3 Cogmation RobotSuite


RobotSuite consists of a visual programming tool and a robot simulator. The simulator part is very focused on the execution of code developed in the visual programming tool. Apparently, Cogmation’s main interest is AI behavior generation for games. The company does not describe the product well enough to allow a more thorough comparison.

![Screenshot of Cogmation RobotSuite Simulation Component](image)

Figure 3: Screenshot of Cogmation RobotSuite Simulation Component

Characteristics:

**Graphics:** Unknown

**Physics:** Nvidia PhysX

3.1.4 anyKode Marilou

http://www.anykode.com/mariloukeyfeatures.php

Marilou is a commercial simulator with a wide range of programming interfaces. The robot API used to interface with the simulator is the same as for the real robots, similar to Player/Stage (see Section 3.2.2). In contrast to many other simulators, it also supports sound sensors and detailed surface properties, such as reflection indices and specific behavior for infrared and sonar sensors. Tools to model robots, including physical parameters, are included.

Marilou seems to be the most extensible and usable commercial simulation available, however, its high price is a major obstacle.
Characteristics:

**Graphics:** Direct3D  
**Physics:** ODE (open source)

### 3.2 Open Source Robotics Related Simulators

#### 3.2.1 USARSim

http://usarsim.sourceforge.net/

*USARSim* [CLW+07, CLW+06, BSC+06] (United System for Automation and Robot Simulation) is a very mature simulator based on the game Unreal Tournament 2004. The system has been used extensively in the RoboCup Rescue Virtual Robots competition since 2006. A number of validated robot models (Telerob Telemax, Foster-Miller Talon, and other SAR and bomb defusal robots) exist already in the simulator package. A space exploration rover (Personal Exploration Rover by Carnegie-Mellon University) is also modeled. Additionally to robots, a number of important sensors (laser range scanners, inertial navigation units, etc) are implemented as well.

It has been shown already that it is possible to import terrain data from real measurements taken in complex environments; examples include ground truth data from Mars collected by the rover Opportunity [BSN+08]. Also, the simulator package includes a communication module that could be adapted to the underwater domain.

Though there are many good reasons speaking for USARsim, there is one problem with it at the moment of writing this document (July 2009): there is currently a migration of its basis from UT 2004 to UT 3 under way. Though the new engine in principle also provides new promising features, especially improved hardware acceleration and multi-threadedness, it nevertheless implies significant changes of which the implications have to be tested in the coming weeks.
Characteristics:

**Graphics:** Epic Unreal Engine 2  
**Physics:** MathEngine Karma (as part of Unreal Engine 2)

### 3.2.2 Player/Stage/Gazebo

http://playerstage.sourceforge.net/

*Player* [GVS⁺01, CMG05] is a very widely used robotics middleware, and contains two simulator packages. Stage is used for 2D simulations with many robots, and Gazebo is used for simulating a few robots in 3D. Gazebo was developed at the University of South California Robotics Research Lab. Player is completely open source and uses the ODE physics library in Gazebo. While the simulator does allow multiple robots to be simulated simultaneously, there is no support to distribute the simulation over multiple servers or simulators to allow for a greater number of robots.

Though Player/Stage is widely used, the acceptance of Gazebo is much lower. Mainly, there are many reports about reliability and stability issues in the community.

Characteristics:

**Graphics:** Ogre3D (open source)  
**Physics:** ODE (open source)
3.2.3 Simbad

http://simbad.sourceforge.net/

*Simbad* [HB06] is a 3D robot simulator for scientific and educational purposes written in Java. It is mainly dedicated to researchers and programmers who want a simple basis for studying Situated Artificial Intelligence, Machine Learning, and AI algorithms in general, in the context of Autonomous Robotics and Autonomous Agents. It is not intended to provide a real world simulation and is kept voluntarily readable and simple.

*Simbad* is therefore not a very good candidate for Co³-AUVs related simulation.

Characteristics:

**Graphics:** Java3D

**Physics:** None

3.2.4 breve

http://www.spiderland.org/

*breve* [Kle02] is a free, open-source, cross-platform software package which makes it easy to build 3D simulations of decentralized systems and artificial life. Users define the behaviors of agents in a 3D world and observe how they interact. *breve* includes physical simulation and collision detection, so users can simulate realistic creatures, and an OpenGL display engine so simulated worlds can be visualized.

*breve* seems to be focused on simple swarm and artificial life simulations, such as evolutionary robotics, rather than complex mobile robot experiments in detailed unstructured environments.

Similar to *Simbad*, *breve* is hence not a good candidate for the Co³-AUVs project.

Characteristics:
Figure 7: Screenshot of Simbad

Figure 8: Screenshot of breve
3.2.5 SARGE - The Search And Rescue Game Environment

http://sarge.sourceforge.net/

SARGE [CGBM08] is a game and robotics simulation focused on operator training. It is developed by the Robotics Group at University of South Florida. SARGE’s main application is to allow future law enforcement or emergency response robot operators to train in an easy to use and accurate simulation environment. However, it can also be used as a standalone simulator for robotics experiments, as well as in a multi player scenario.

The software is exclusively fitted towards search and rescue robotics at the moment, but does provide a good view of what is possible in a game environment. The code itself is open source (under Apache Software License 2.0), but a proprietary game engine is needed to compile the free code.

![Screenshot of SARGE](image)

Figure 9: Screenshot of SARGE

Characteristics:

**Graphics:** Unity Engine (proprietary)

**Physics:** Nvidia PhysX

3.3 Custom Made Open Source Solution

A custom simulation package, focused on for example on distributed mobile robotics in different underwater environments, can address all shortcomings of the previously mentioned alternatives. Using available freeware and open source tools and libraries, a complete simulation system could be built.
<table>
<thead>
<tr>
<th>Name</th>
<th>URL</th>
<th>License</th>
<th>Supp. Platforms</th>
<th>Customizable</th>
</tr>
</thead>
<tbody>
<tr>
<td>USARSim</td>
<td>usarsim.sf.net</td>
<td>GPL</td>
<td>Win</td>
<td>yes</td>
</tr>
<tr>
<td>Player/Gazebo</td>
<td>playerstage.sf.net</td>
<td>GPL</td>
<td>Win/Mac/Linux</td>
<td>yes</td>
</tr>
<tr>
<td>Simbad</td>
<td>simbad.sf.net</td>
<td>GPL</td>
<td>All (Java)</td>
<td>yes</td>
</tr>
<tr>
<td>breve</td>
<td>spiderland.org</td>
<td>GPL</td>
<td>Win/Mac/Linux</td>
<td>maybe (Open Source)</td>
</tr>
<tr>
<td>SARGE</td>
<td>sarge.sf.net</td>
<td>APL/Proprietary</td>
<td>Win/Mac</td>
<td>no</td>
</tr>
<tr>
<td>Cyberbotics Webots</td>
<td>cyberbotics.com</td>
<td>Proprietary</td>
<td>Win/Mac/Linux</td>
<td>yes</td>
</tr>
<tr>
<td>Microsoft Robotics Studio</td>
<td>microsoft.com</td>
<td>Free</td>
<td>Win</td>
<td>yes</td>
</tr>
<tr>
<td>Cogmation RobotSuite</td>
<td>cogmation.com</td>
<td>Proprietary</td>
<td>Win/Mac</td>
<td>limited (Closed Source)</td>
</tr>
<tr>
<td>Anykode Marlou</td>
<td>anykode.com</td>
<td>Proprietary</td>
<td>Win/Linux</td>
<td>yes</td>
</tr>
<tr>
<td><strong>Possible Custom Implementation</strong></td>
<td>n/a</td>
<td>Open Source</td>
<td>Win/Mac/Linux</td>
<td>yes</td>
</tr>
</tbody>
</table>

Table 1: Tabular overview over some simulation engines (part 1)

The list below shows libraries that are open source or free for non-commercial use that can serve as basis for an according effort:

**Graphics:** Cross-platform hardware accelerated graphics with either
- Ogre3D (http://www.ogre3d.org/), or
- OpenSceneGraph (http://www.openscenegraph.org/)

**Physics:** Accurate physics simulation though either
- Bullet Physics (http://www.bulletphysics.com/),
- Nvidia PhysX (http://developer.nvidia.com/object/physx.html), or
- ODE (http://www.ode.org/)

**Modeling:** 3D modeling tool for artwork such as
- Blender (http://www.blender.org/)

**Distributed Simulation:** large scale simulation clusters via either
- Raknet (http://www.jenkinssoftware.com/),
- OpenTNL (http://www.opentnl.org/),
- Hawk Network Library (http://www.hawksoft.com/hawknl/), or
- Game Network Library (http://www.gillius.org/gne/), or

The main tasks to develop a suited Open Source simulator for underwater robotics would be an integration of some of the packages listed above, including extensive integration tests to determine the best fit of the different packages.

The main disadvantage of this solution is that it would require a significant amount of effort, which goes beyond the scope of the Co²-AUVs project.
<table>
<thead>
<tr>
<th>Name</th>
<th>HW acc. Graphics/Physics</th>
<th>Distributable</th>
<th>Client/Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>USARSim (UT 2004)</td>
<td>yes/no</td>
<td>maybe</td>
<td>yes</td>
</tr>
<tr>
<td>USARSim (UT 3)</td>
<td>yes/yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Player/Gazebo</td>
<td>yes/no</td>
<td>maybe (Open Source)</td>
<td>no</td>
</tr>
<tr>
<td>Simbad</td>
<td>yes/no</td>
<td>maybe (Open Source)</td>
<td>no</td>
</tr>
<tr>
<td>breve</td>
<td>yes/yes</td>
<td>yes (Open Source)</td>
<td>no</td>
</tr>
<tr>
<td>SARGE</td>
<td>yes/yes</td>
<td>yes (up to 100 players)</td>
<td>yes</td>
</tr>
<tr>
<td>Cyberbotics Webots</td>
<td>yes/no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Microsoft Robotics Studio</td>
<td>yes/yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Cogmation RobotSuite</td>
<td>unknown</td>
<td>no</td>
<td>unknown</td>
</tr>
<tr>
<td>Anykode Marilou</td>
<td>yes/no</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Possible Custom Implementation</td>
<td>yes/yes</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>

Table 2: Tabular overview over some simulation engines (part 2)

3.4 Coarse Summary of Alternatives

Tables 1 and 2 give a coarse overview of the properties of the different examples discussed before. As mentioned, the different solutions listed here serve as prototypical examples of different types of alternative solutions for the problem at hand. Given the requirements and the above options, a dual strategy is pursued. First of all, USARSim is used as a basis to have an easy start that allows the generation of fast results. Second, the generation of a custom solution using OGRE for visualization and Bullet as physics engine is investigated. In doing so, the same APIs are used so that a migration of experiments from USARSim to the custom simulator is easily possible.

References


