A multirobot platform based on Autonomous Surface and Underwater Vehicles with bio-inspired neurocontrollers for long-term oil spills monitoring

Antonio Guerrero-González · Francisco García-Córdova · Francisco J. Ortiz · Diego Alonso · Javier Gilabert

Abstract This paper describes the BUSCAMOS-Oil monitoring system, which is a robotic platform consisting of an autonomous surface vessel combined with an underwater vehicle. The system has been designed for the long-term monitoring of oil spills, including the search for the spill, and transmitting information on its location, extent, direction and speed. Both vehicles are controlled by two different types of bio-inspired neural networks: a Self-Organization Direction Mapping Network (SODMN) for trajectory generation and a Neural Network for Avoidance Behaviour (NNAB) for avoiding obstacles. The systems' resilient capabilities are provided by bio-inspired algorithms implemented in a modular software architecture and controlled by redundant devices to give the necessary robustness to operate in the difficult conditions typically found in long-term oil-spill operations. The efficacy of the vehicles' adaptive navigation system and long-term mission capabilities are shown in the experimental results

Keywords ASV, UUV, multi-vehicle cooperation, artificial neural networks, bio-inspired control, oil spills monitoring, component based software architecture, autonomous long-term navigation.

1 Introduction

The recent history of oil spills has clearly shown their catastrophic effect on coastal ecosystems. Oil spills have been taking place at sea since the early days of offshore oil extraction and oil-carrying tankers (Burger 1997). Prevention and preparedness against oil spills is by far the best strategy, thus impeding the considerable damage they can cause by shortcutting their arrival at the coast. Monitoring the spill is a previous step to the cleanup. The lower density of the oil makes it accumulate on the surface, where it is visible from the air. The wide range of types of oil spill means that different strategies must be used for each one, from chemical or physical means to burning them off *in situ*.

Monitoring the spill is a previous step to the cleanup. The lower density of the oil makes it accumulate on the surface, where it is visible from the air. The wide range of types of oil spill means that different strategies must be used for each one, from chemical or physical means to burning them off *in situ*.

Monitoring vast extensions of spilled oil requires aerial observation either by aircraft or satellites. Technology has always been at the forefront of this strategy by providing remote sensing images (Fingas and Brown 2014), modeling (Liu et al. 2013; Klemas 2012), and most recently, autonomous underwater vehicles (Camili et al. 2010). Effective reaction to oil spills needs the maximum amount of information on the slick's location, extent, direction and speed by continuous monitoring, a process which involves time and substantial operational costs.

The largest oil spill in the history of the petroleum industry (Sammarco et al. 2013) took place on April 20th, 2010 when the BP Deepwater Horizon Macondo well blew out in the north of the Gulf of Mexico. The spill lasted for 84 days and leaked 800,000 – 1m tons of crude oil from the sea floor, with an estimated peak of 15,520 tons per day, (Sammarco et al. 2013; Reddy et al. 2012). The area affected by the slick covered 62,159 km^2 and affected both the coastal area and the water column (Norse and Amos 2010). This was the first time autonomous marine robots played a pioneer role in fighting an oil spill, as did other technologies like satellite remote sensing and ocean modeling.

This was in fact the first recording of the extent of a large oil plume, not visible from the air. Oil spills in water can surface a long distance from the spill, depending on the ocean currents in the area. Monitoring these plumes is critical to trusting the trajectory models that predict where the slick will go. Autonomous underwater vehicles are now bridging the gap between traditional aerial observation and sea-based techniques that require new technology to monitor oil slicks from beginning to end.

The key to the success of these missions is, firstly, the oil-sensing equipment included in the vehicles, and

A. Guerrero-González (✉)· F. García-Córdova · F.J. Ortiz · D. Alonso · J. Gilabert
Universidad Politécnica de Cartagena,
C/ Doctor Flemming sn, 30203 Cartagena, Spain
e-mail: antonio.guerrero@upct.es
F. García-Córdova
e-mail: francisco.garcia@upct.es
F.J. Ortiz
e-mail: francisco.ortiz@upct.es
D. Alonso
e-mail: diego.alonso@upct.es
J. Gilabert
e-mail: javier.gilabert@upct.es

secondly, their autonomy. A wide range of this equipment is now available, from UV fluorometers to mass spectrometers, however the autonomy challenge requires new designs and mission planning.

The need for a wide range of in situ data on different time and space scales has given rise to an interest in the development of different types of Unmanned Underwater Vehicles (UUVs) and Autonomous Surface Vehicles (ASVs) to collect this data. These marine platforms have different capabilities in communications, durability, mobility, capacity and autonomy (Guo 2009; Caccia et al. 2008; Ribas et al. 2012, Gracias et al. 2013, Gutierrez et al. 2010). Energy autonomy, the capability to react autonomously to unforeseen situations and robustness to survive in the harsh marine environment are the main challenges faced by autonomous vehicles in intensive, long range spill monitoring.

The limitations of Autonomous Underwater Vehicles (AUV) communications, energy capacity and positioning can be offset by combining them with ASVs. German et al. (2012) present a vision of the future use of systems based on coordinated ASV and AUVs for long-term and long range oceanographic exploration and monitoring, and highlight the advantages of these technologies. The combination of an ASV and an UUV is thus well suited to monitoring and mapping oil spills. The ASV has sufficient power autonomy to carry out long-term missions and real-time communications with the AUV or land/ship base and can carry out sweeps over the oil spill area and in the water column itself. However, long-term oil spill applications can be very demanding in a difficult working environment. It is therefore necessary that the tools used in these scenarios be robust and adaptable to unforeseen situations involving failures or partial deterioration of the system. This capacity gives them resilience, and a resilient robot can develop new settings through continuous self-modeling (Bongard et al. 2006; Sun et al. 2011).

Many applications in the bio-inspired robotics field exhibit these resilient capabilities (Bandyopadhyay 2005). Even the simplest organisms are capable of behavioral feats unimaginable in the most sophisticated machines. When an animal has to operate in an unknown environment it must somehow learn to predict the consequences of its own actions. By finding the cause of environmental events, it becomes possible for an animal to predict future events. A related form of learning is known as *operant conditioning* (Grossberg 1971). Chang and Gaudiano (1998) introduced a neural network for obstacle avoidance that is based on a model of classical and operant conditioning.

In this paper, we present the BUSCAMOS-Oil system, a multivehicle system based on an ASV-UUV combination for oil spill monitoring. One of its outstanding features is the incorporation of new control strategies based on bio-inspired neural networks (NN) to give adaptability and robustness to the ASV and UUV. The two vehicles are connected by an umbilical cable, which allows them to share power and computing resources. With its two robots, the platform has the capability of monitoring large tracts of sea, both on the surface and in the water column. With its time-series measurements, the system draws up precise maps of the oil plume, with information on spill location, size, extent, direction and speed. To ensure its power supply, the ASV contains photovoltaic systems which recharge the system's batteries. It also has a diesel generator when no solar energy is available. The software architecture includes specific modules to handle situations that may occur in the field of long-term autonomous operations, such as a fault or malfunction of the vehicle's propeller systems, unexpected situations, power restrictions, etc. All these features provide the system with the necessary resilient capabilities.

As mentioned above, the control system has two bio-inspired neurocontrollers for each vehicle, one to control its movements and route, and another for obstacle avoidance. The former is a kinematic adaptive neurocontroller that uses a SODMN. This bio-inspired neurocontroller is a real-time unsupervised neural network that learns to control autonomous underwater and surface vehicles in a nonstationary environment. The latter is a NNAB that ensures the vehicles can avoid obstacles by means of a neural network that learns to control avoidance behavior in autonomous vehicles and is based on a form of animal learning known as *operant conditioning*.

The software architecture of the system is of vital importance, since it is the primary carrier of the software system's quality attributes, such as performance, modifiability, and reliability. A modular, component-based architecture not only guarantees the reuse of the best practices during the design stage, but also guarantees control of any incidents in runtime, ensuring for example the treatment of exceptions in individual software components and the adaptation of the system to unforeseen scenarios by enabling or disabling certain components or connections. This paper includes a description of the system architecture, from the design of the components to their final implementation with the *C-Forge* model-driven and component-based toolchain, which incorporates the latest trends in robotics software engineering.

This paper is organized as follows. We first (Section 2) review related works on autonomous oil-spill vehicles, the control algorithms applied to these vehicles and the software architecture applied in their control. In Section 3 we present the combined ASV and UUV robotic marine platform for long-term oil spill monitoring. BUSCAMOS-Oil's control architecture design is presented in Section 4. The autonomous navigation system based on biologically-inspired neural algorithms

for the proposed vehicles is described in Section 5. Section 6 gives the experimental results of tests on the behavior of the proposed platform's autonomous vehicles' bio-inspired avoidance and approach control system, and the results of an oil sensor sampling test. Finally, Section 7 contains our conclusions from the experimental results.

2 Related Works

2.1 Oil spills and Autonomous Vehicles.

During the Deepwater Horizon crisis a huge effort was made to obtain quick and accurate information on the spill. Two layers were monitored (Hollander et al. 2010) at depths of 400 and 1,100 m by traditional casts from surface vessels. After these cruises AUVs made their debut. The "Sentry" AUV (Woods Hole Oceanographic Institution) equipped with the "Tethys" mass spectrometer to track and localize in-water hydrocarbons was deployed in two cruises (Kinsey et al. 2011, Camili et al. 2010). The Sentry mapped and characterized the extent of the two layers at 30 km from the spill site. The Dorado AUV (MBARI) was also deployed and took samples of in-water oil in the layers (Ryan et al. 2011; Zhang et al. 2011).

These pioneer cruises proved the worth of AUVs in monitoring oil spills at different depths and inspired several new projects, such as the URready4OS Project co-financed by the Directorate-General of Humanitarian Aid and Civil Protection of the European Commission, to make available AUV fleets. Several other devices, such as the WaveGlider, are able to operate on the surface only.

The robots can carry a large payload, including cameras, side-scan sonar, laser profilers and a wide assortment of oceanographic sensors (Schofield et al. 2004).The main limitations of the robots are their autonomy, sensor payload and communications during operations. The AUV must be accompanied by manned research vessels and the associated expense limits operations. ASVs do not have this disadvantage as they only work on the surface.

In (Bhattacharya et al. 2011) a cooperative system composed of two ASVs for oil skimming and cleanup is proposed. This robotic system reduces the enormous effort involved in manual skimming operations. German et al. (2012) proposed the use of ASVs to assist the AUV, thus reducing the cost of missions and allowing long-term data collecting operations over large maritime areas.

The proposed system has several advantages over the oil spill equipment presently available: the ASV provides power to the UUV and acts as a communications gateway between the UUV and the land station and can thus execute long term autonomous missions. With oil sensors both in the ASV and UUV, the system can follow both on- and in-water oil slicks. The BUSCAMOS-Oil system can look for subsurface plumes in deep water while it also monitors the surface.

2.2 Bio-inspired control algorithms for autonomous marine vehicles

In recent years, interest in underwater exploration has expanded rapidly. Motion control and real-time collision-free trajectory generation are among the most significant problems for emerging robotic applications related to autonomous navigation of AUVs and ASVs in a dynamic, unstructured environment. Due to the highly non-linear and dynamic characteristics of underwater conditions, the control of AUV navigation is a far from trivial problem (Antonelli et al. 2001, 2006; Bandyopadhyay 2005).

In order to solve the nonlinearity and uncertainty problems, as well as generating real-time trajectories through learning of autonomous AUV and ASV navigation, several neural network adaptive control schemes have recently been used (Krieg and Mohseni 2010; Lee and Choi 2000). Artificial Neural Networks (ANNs) have been applied in (Burns 1995) for the intelligent optimal control of surface ships with changing dynamic characteristics. In (Leonessa et al. 2006), an ANN model reference adaptive controller algorithm was proposed for trajectory tracking of surface marine vehicles in the presence of unmodeled dynamics. Tee and Ge (2006) employed ANNs to approximate unknown ship dynamics and an adaptive ANN control method was shown to be effective for tracking a fully operational ocean surface vessel. Wei and Chen (2011) proposed a class of sliding mode variable structure control method (SMVSC) based on a radial basis function neural network for a ship course control system. Peng et al. (2013) proposed a robust adaptive formation controller that employed a neural network and dynamic surface control technique for underactuated ASVs.

Bio-inspired neural models are an alternative to solving optimal learning and the generation of autonomous trajectories. This new methodology consists of adopting neurobiological knowledge and models in artifacts that are capable of autonomous behavior in the physical world. These models are based on the activity of neurons in primary motor areas (MI), as in animals' central nervous system (CNS) when performing a task and learning is based on the reactions to different stimuli (García-Córdova 2007; Kawato 1999; Grossberg 2013). A number of adaptive control system models of cerebellar function have been presented in recent years (Yu et al. 2014; Bullock et al. 1999). These models are applied to generate movement trajectories using unsupervised Hebbian learning (Tolu et al. 2012; Bullock et al.1993).

The primary motor cortex (MI) plays an important role in the control of voluntary movement and the execution of planned movements. Algorithms based on biological

sensorimotor control provide reliable adaptive learning models to different architectures depending on the assigned tasks (Ajemian et al., 2000). The term 'sensorimotor transformation' refers to the process by which sensory stimuli are converted into motor commands. This process is crucial to any biological organism or artificial system that possesses the ability to react to the environment. During goal-directed movement in primates, a sensorimotor transformation in cortical motor areas generates a dynamical pattern of muscle activation. This means that a set of coordinate transformations that begins with an extrinsic coordinate frame representing the spatial location of a target and ends with an intrinsic coordinate frame describing muscle activation patterns (Sfakiotakis and Tsakiris 2007; Bullock et al. 1994, 1999). Bio-inspired neural circuits as central pattern generators (CPGs) are inspired in the locomotion control of invertebrates and vertebrates animals. This neural circuits is capable of producing coordinated patters of high-dimensional rhythmic output signals with applications in biped robots, snake robots, dolphin-like robots, Quadruped robots (Li et al. 2015; Ijspeert 2008; Liu et al. 2011; Sfakiotakis and Tsakiris 2007; Yu et al. 2014).

Several architectures have been introduced using the mathematical mechanism of Self-Organizing Maps (SOM), which enable learning of sensorimotor mapping involved in modeling forward and inverse models in robotic control (Kawato and Samejima 2007). Similarly, other neuronal control architectures have implemented hierarchical controllers acting in parallel and are based on sensorimotor transformation in cortical motor areas to solve a wide range of problems, such as inverse kinematics, reactive behaviors and autonomous navigation (García-Córdova 2007; Bullock et al. 1999; Ajemian et al. 2000).

In the control of autonomous navigation, obstacle avoidance is one of the most basic and deceptively simple problems that autonomous vehicles have to solve. The problem, however, has turned out to be rather more complex when dealing with constantly changing environments and noisy sensors and actuators (Hamilton and Wolpert 2002; Bandyopadhyay 2005). The behavior-based approach to robotics has motivated researchers to address the problem from a number of different perspectives (Bandyopadhyay 2005). Adaptive obstacle avoidance behaviors for mobile robots have been achieved using subsumption-based architecture schemas (Peng et al. 2013; Bandyopadhyay 2005), artificial neural networks (Pan et al. 2013), evolutionary computation (Carreras et al. 2005), and biological inspired neural circuits (Chang and Gaudiano 1998; Chang 2005; Guerrero-González et al. 2010), among other approaches.

In (Chang and Gaudiano 1998), a bio-inspired neural network for obstacle avoidance based on a model of classical and operant conditioning is described. In the classical conditioning paradigm, learning occurs by repeated association of a Conditioned Stimulus (CS), which normally has no particular significance for an animal, with an Unconditioned Stimulus (UCS), which does have significance and always gives rise to an Unconditioned Response (UCR). The response to be elicited by the CS after classical conditioning is known as the Conditioned Response (CR) (Chang 2005; Grossberg 1971). Hence, classical conditioning is the putative learning process that enables animals to recognize informative stimuli in the environment. In the case of operant conditioning, an animal learns the consequences of its actions. In other words, the animal learns to exhibit more frequently a behavior that has led to reward in the past, and to exhibit less frequently a behavior that led to punishment. In the field of neural networks research, it is often suggested that neural networks based on associative learning laws can model the mechanisms of classical conditioning, while neural networks based on reinforcement learning laws can model the mechanisms of operant conditioning (Guerrero-González et al. 2010). Reinforcement learning is used to acquire navigation skills for autonomous vehicles and updates both the vehicle model and optimal behaviour at the same time (Carreras et al. 2005).

In this paper, the autonomous navigation system of BUSCAMOS-Oil (García-Córdova and Guerrero-González 2013; Guerrero-González et al. 2010) consists of a SODMN and a NNAB, both biologically inspired and considered as an alternative to marine navigation. The SODMN is a kinematic adaptive neuro-controller and a real-time, unsupervised neural network that learns to control autonomous underwater and surface vehicles in a nonstationary environment. The SODMN combines associative learning to generate transformations between spatial and velocity coordinates. The transformations are learned in an unsupervised training phase, during which the vehicle moves as a result of the randomly selected velocities of its actuators. The controller learns the relationship between these velocities and the resulting incremental movements. The NNAB is a neural network based on animal behavior that learns to control avoidance behaviors in autonomous marine vehicles based on a form of animal learning known as *operant conditioning*. Learning, which requires no supervision, takes place as the vehicle moves around an environment cluttered with obstacles. The biologically-inspired neural networks proposed in this paper represent a simplified way of partially understanding the mechanisms that allow the brain to collect sensory input to control adaptive behaviors of autonomous animal navigation. In this work, the autonomy of the vehicle is evaluated in several scenarios.

In contrast to other tracking controllers, the proposed navigation system does not require the dynamic model of the vehicles, making it robust to sudden situations of

underactuation. Other NN techniques based on SOMs were applied in manipulation robotics and autonomous land robotics and showed good responses to underactuation. The proposed navigation system is based on this type of NNs for application to the marine environment. For example, during long missions a propeller may fail, leading to a case of underactuation. The proposed autonomous navigation system to cover this situation (see Experimental section), and the navigation control techniques mentioned above do not include this type of situation.

The NNAB is an alternative to controlling avoidance behaviors in autonomous marine vehicles and is an algorithm included in the proposed bio-inspired control architecture. In contrast with other heuristic approaches it learns to avoidance obstacle in a learning phase.

2.3 Control software architectures

Many control architectures have been proposed to increase the autonomy of AUVs, as reviewed in (Ridao et al. 2000). The majority are hybrid architectures, like many ground service robots. Hybrid architectures take advantage of reactive and deliberative paradigms while minimizing their limitations. Hybrid architectures usually include three layers: deliberative, executive, and reactive, including world modeling for reasoning about the environment.

However, the question remains open of how this type of architecture is finally implemented and supported by the software. Although some years ago it was typical to develop software for robots from scratch, nowadays the reuse of designs, algorithms and software components have gained importance. Shakhimardanov et al. (2010) offers an exhaustive survey with an extensive bibliography of component models in robotic software frameworks, which can be roughly classified as:

- **Purely algorithmic solutions** like OpenSLAM, Intempora, MRPT, etc. The user code directly invokes the algorithms, which are normally encoded as libraries that are linked against the code.
- **Robotics middleware and toolkits** like ROS, MRPT, YARP, etc. These solutions complete the above-mentioned solution by also providing drivers for many robotic sensors and actuators, and a communication middleware for distribution and modularity. The middleware provides great flexibility and extensibility, but on the other hand, the publish/subscribe mechanism adds complexity to application debugging, since it is not possible to check at compile time the validity of the messages that will be sent through the middleware at runtime.
- **Component-based frameworks** like OROCOS, ORCA2, ROBOCOMP, GenoM, SmartSoft, PROTEUS, etc. These provide the same support as in the previous category but aim at solving the problems related to software reuse and development by converting modules into components: software artifacts that clearly state which services they require/provide and which always send typed messages through their ports. Typed messages make it possible to check at compile time whether components are correctly connected.
- **Model-Driven tools** like Smartsoft, BRIDE, RobotML, etc. These provide a set of tools, based on the Model-Driven Software Development (MDSD) MDSD (Bezivin 2005) paradigm, to support the development of applications from a higher level of abstraction than just using the primitives provided by programming language, like classes or functions. That is, the MDD allows the definition of specific concepts, like component, port, message, etc., and the definition of applications by combining them. Model transformations are then used to process these definitions to finally obtain source code, or to target one or more of the abovementioned solutions, like a component framework (in the case of SmartSoft) or a robotic toolkit (ROS in the case of BRIDE).

The C-Forge (Ortiz et al. 2014a) toolchain employed in this article to design the software for the BUSCAMOS-Oil system can be classified into the two above groups, as it includes a component framework (FraCC) supported by a MDSD tool for designing component-based applications (WCOMM). Of all these tools, SmartSoft (Schlegel et al. 2012) is the approach closest to C-Forge, since it models components as architectural units that are later translated to the target platform, which is also implemented as a framework that provides the required run-time support.

Given its wide success, we end this section by briefly highlighting why a model-driven approach goes beyond the solutions provided by ROS (Gerkey et al. 2009). At the lowest level of C-Forge, FraCC provides the runtime support to execute applications, similar to ROS. We could also generate ROS code from WCOMM components, however one of the distinguishing aspects of FraCC is that it does not impose any policy for executing components. The user can thus define how many processes and threads will execute the application, and where components will be run (see Section 4). This feature differentiates FraCC not only from ROS, but also from the other frameworks, since it allows applications to take into account concurrency and real-time requirements.

Additionally, WCOMM provides modeling primitives (component, port, message, state, etc.) that have a higher level of abstraction than that of classes or functions provided by toolkits. These primitives help robotics developers to better design and review the application, even graphically, to overcome any deficiencies the implementation platform may have. For instance, WCOMM defines typed messages, which allow us to

check at design time that components are correctly connected. ROS equivalents (i.e. “topics”) are strings, and the connections among components are checked at runtime. If one topic is incorrectly written, messages will not flow through nodes and the robot will not work properly. Another example is the use of Finite-State Machines (FSM) to model component behavior. It is easier to model and review a graphical model of an FSM than to write and review the equivalent code.

3 System Description

BUSCAMOS-Oil is a robotic marine platform composed of an ASV and an UUV applied to mapping oil spills (see Fig.1). The UUV is integrated within the ASV, and for ASV operations. The UUV is launched and recovered in the theater of operations. The two vehicles are joined by an umbilical cable, which allows them to share hardware, software and power resources.

The system has been designed to carry out long-term oil spill monitoring operations. Both the ASV and the UUV are equipped with a C3 submersible fluorometer with three optical sensors to detect crude oil, refined fuel and rhodamine.

Fig. 1 BUSCAMOS-Oil system. Composed by an ASV and an UUV.

The detection ranges are: 0-2700 ppb for oil-crude, 10-100 ppm for oil-fine and for Rodamine 0-1000 ppb. Figure 2a shows the oil sensor on the ASV prow under the waterline, the figure 2b shows the oil sensor located on the larboard side of the UUV rear body.

(a) **(b)**

Fig. 2 Oil sensor on the ASV and UUV. (a) on ASV, (b) on UUV.

BUSCAMOS-Oil looks for a spill from the initial position. When the spill is located, it determines its size, depth, speed and direction, and remains as a drifting buoy on the spill for long periods of time reporting any changes in its evolution. A mission includes the following sequence of actions:

1. The base station (Point H in Figure 3) manages the operation and launches BUSCAMOS-OIL to explore at a GPS point. The system receives this GPS point together with the mission parameters and goes to this point (see Figure 3 point E).
2. Once at this point, the ASV follows a search path looking for the spill, for which it executes transects of length, orientation and distance between transects. This search path is parameterized for each mission. This maneuver stops when the sensor detects oil at point S1, as shown in Figure 3.
3. Once at the spill, in a series of sweeps, the ASV determines the extent of the spill. When it ceases to detect oil, it stores this point as the spill limit point (S points in Figure 3a), then turns and follows another straight line in parallel for a given distance, parameterized for each mission. The polygon formed by the S points determines the contour of the spill on the surface. When this maneuver is finished, it goes to the center of the spill (Point C in Figure 3a).
4. Once at C, the ASV ships its propellers to prevent umbilical cable curl and launches the UUV.
5. The UUV zig-zags at various depths (see Figure 3b). When it no longer detects oil, it marks the underwater limit of the spill U, then turns and resumes its zig-zag. The extent of the spill is determined by the polygon formed by the U points (see Figure 3b).This maneuver is executed in north-south and east-west directions. (see Figure 3b).
6. After recovering the UUV, the ASV acts as a drifting buoy on the spill. When oil is no longer detected, it unships its propellers and executes Actions 2, 3, 4, 5 and 6. This maneuver is repeated periodically to determine the spill extent.

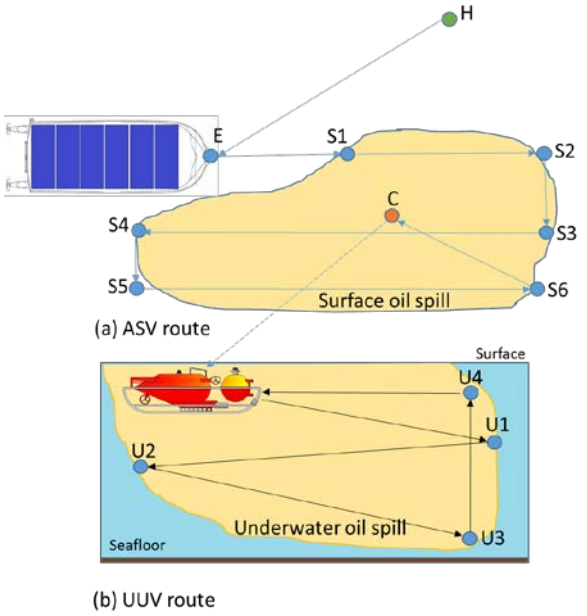

Fig. 3. Example of BUSCAMOS-Oil Mission.

The position of the two vehicles is always known, in the ASV by GPS receivers and in the UUV by combining the GPS data from the ASV and calculating the relative position from the UUV by USBL. The difference in the position from Point C is used to determine the velocity and direction of the spill.

The umbilical cable is 500 m long and the submarine has a maximum depth of 300 m. The umbilical cable allows both vehicles to share the same Ethernet communication network and the same power lines. This common communication network facilities the distribution of software components and the integration of redundant components. The common power supply, composed of a photovoltaic plant on the ASV's deck and a backup diesel generator, are designed for long-term operations.

3.1 BUSCAMOS ASV.

The BUSCAMOS ASV is a monohull vessel built of polyester reinforced with fiberglass (see Fig. 4). The dimensions of the hull are 5.10 m length overall and 1.97 m maximum width. The displacement of the vessel under normal operating conditions is approximately 1,000 kg with an average draft of 0.325 m.

Fig. 4 BUSCAMOS-Oil ASV.

The ASV has an electric propulsion system made up of two outboard propellers anchored to the transom. Each thruster is powered in series by two 28V serial connected lithium ion batteries. The energy stored in the batteries is 10,740 kW·h, giving the system an autonomy in operations of around 6 hours without battery recharge.

The vessel is steered by the independent operation of the two propellers, as well as a rudder control. The rudder consists of a linear motor that changes the thrust direction of the propellers.

Recharging the batteries is by a dual system of solar panels and a backup diesel generator. This configuration provides recharging redundancy and allows the vessel an extended time of operations. The photovoltaic panels are fixed to an aluminum structure for best performance.

The ASV is equipped with an electric winch to stow the umbilical cable and a set of pulleys that guide the cable in deployment and recovery operations.

Launch and recovery operations, critical maneuvers for long-term operations, are fully automated and controlled from the stern.

During ASV navigation, the UUV is stored in a hydraulic basket on the ASV. For launching, the platform lowers the UUV into the water in a horizontal movement of the platform (see orange arrows in Figures 5a and 5b) and another vertical (Figure 5c). The ASV propellers are shipped during this maneuver to prevent the umbilical cable fouling the thrusters.

In the recovery maneuver, the ASV lowers the basket in the water and open the basket to ease the maneuver. The UUV is located in front of the ASV stern, and maintaining an orientation 180º it goes back to the ASV helped by the pull of the umbilical cable from the winch. Once on the platform, it closes the basket, takes out the vehicle from the water and introduces it into the ASV.

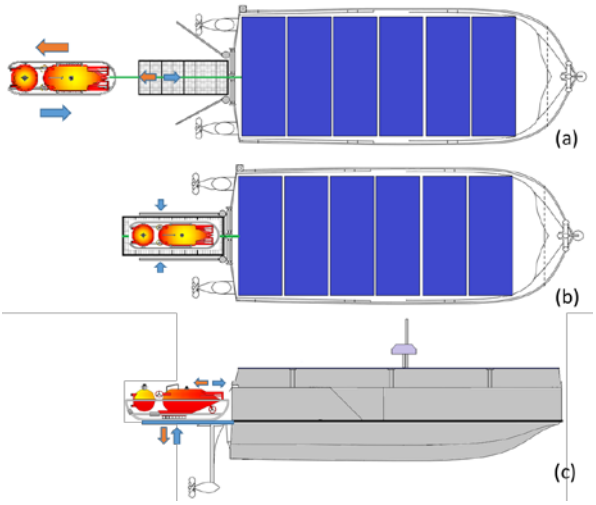

Fig. 5. UUV launch and recovery system.

The photovoltaic system produces enough energy for the long term operation of the system. The panels' position on the hull has little effect on the vessel's aerodynamics and maneuverability. Six lightweight flexible photovoltaic panels cover an area on the ASV deck of 4,000 x 1,355 mm. Total installed photovoltaic power is 780 W.

The photovoltaic system configuration is a typical standalone installation (see Fig. 5). As the demand from the load does not always equal the solar capacity, a battery bank stores excess energy and provides it when required. The battery bank provides stable current and voltage by eliminating transients and provides surge currents to the thrusters when required. Two solar charge controllers are placed between the PVs and the battery bank. These devices adjust the charge rates to the status of the battery bank, applying maximum power point tracker (MPPT) technology. This device boosts the PV lower voltage to charge higher voltage lithium batteries up to 54V nominal.

The ASV integrates the following perception devices: a broadband 3G radar, an aft camera, bow camera, a sidescan sonar, and an imaging sonar for obstacle avoidance.

3.2. BUSCAMOS UUV.

The BUSCAMOS UUV is a refurbishment of a commercial unmanned underwater vehicle with an updated control system and new sensors. The vehicle has two hulls, one of which houses five motors for propulsion and maneuvering, battery power pack and electronics. The other is a spherical hull, forming the head of the vehicle and contains the sensors. The operational depth of the underwater vehicle is 300 m. Its maximum forward horizontal speed is 4 knots, and the vehicle can maintain position with horizontal currents of up 3 knots. The vehicle is maneuverable in all planes, two horizontal thrusters supply forward/reverse motion, two vertical thrusters supply the up/down motion and one transverse stern thruster provides right/left turns. The weight in air is 163.4 kg with a displacement of 163.8 dm^3, the vehicle is balanced with a positive buoyancy of 300 grams approximately, so that it tends to surface automatically when the controls are released.

The vehicle incorporates side scan sonar to analyze the seabed, imaging sonar used for obstacle avoidance, a Doppler velocity log for precise navigation, a sonar positioning transponder to obtain the absolute position of the vehicle in the water, a multiparametric sonde and a submersible fluorometer with optical sensors to detect crude oil, refined fuels and rodhamine.

3.3. Control Hardware Architecture.

The control hardware has the computational resources necessary for the system to carry out its functions. The software control architecture runs entirely on the CPU of the system, which is located in the surface vessel. This CPU contains the control algorithms of both the ASV and UUV platforms. The neurocontrollers give resilient capabilities to the system against failure of the propulsion system and control of the ASV and UUV.

Figure 6 shows a scheme of the BUSCAMOS-Oil control hardware. The system has an Ethernet network on which CPUs, real-time controllers and sensors (cameras and sonars) of the two vehicles are connected.

In addition, each vehicle has a network-based distributed control based on CANBUS fieldbus, in NMEA2000 protocol in the ASV and CANopen protocol in the UUV. The two vehicles have NI RIO Programmable Automation Controllers (sbRIO-9606 and CRIO-9022), which make a bridge between the Ethernet and CAN networks and work as information servers to the system CPU reliably and deterministically.

The NMEA2000 network includes the GPS units, rubber drive and the ASV propeller drives. The ASV's CANopen network is composed of 4 nodes, two in the body section and two in the spherical front hull section of the vehicle. This network drives the actuator thrusters, head movement, focus control, and read the pressure sensors, inundation sensors, humidity, DVL, oil sensor, multiparametric sonde, among others.

Communication with the remote station is by three types of wireless: wifi for access to the internal data network system, with a maximum coverage of 1km in a straight line; radio modem for reading states and key commands, with maximum coverage of 20 Kms; and iridium satellite communication for reading states and key commands in remote locations.

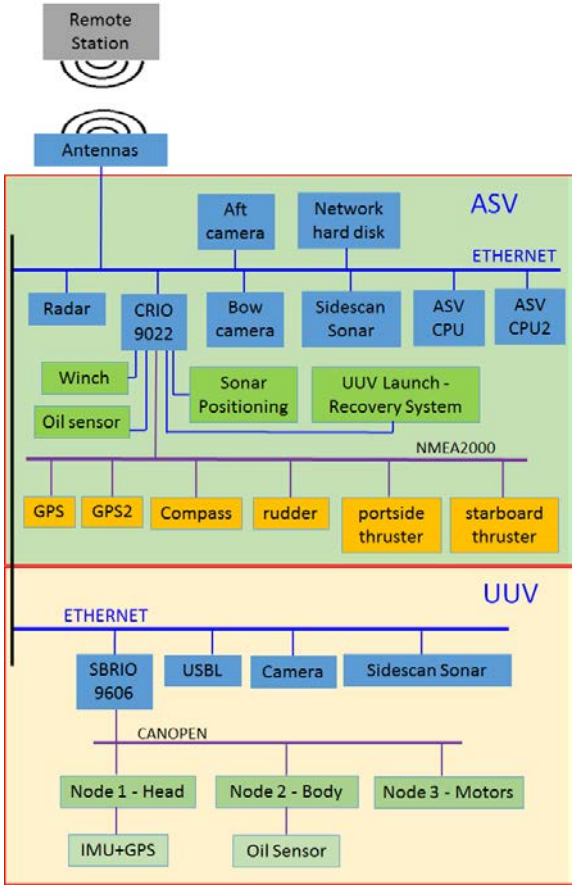

Fig. 6 BUSCAMOS-Oil hardware components and communication networks.

4 Software Control Architecture

As mentioned above, we follow a model-driven component-based approach to software development. The modularity provided by components eases design, allows for the incorporation of best practices, and facilitates the system reconfiguration. In the context of the MISSION-SICUVA project, we developed the *C-Forge* (Ortiz et al. 2014a) tool-chain, which includes WCOMM for formally modeling components, and FraCC, which provides the runtime support required for executing the application. C-Forge follows an iterative and incremental development process in which the benefits of using models are driven as far as possible towards the final execution in FraCC. We model components as units that encapsulate their contents, communicate with other components by sending asynchronous messages only through their ports, and whose behavior (i.e. reaction to external and internal stimuli) is expressed as a finite-state machine (FSM). The use of models also helps in each step of the design process to check whether the application models have been correctly created. The following subsections briefly describe the application of C-Forge to the design of BUSCAMOS-Oil, but the reader can also consult (Ortiz et al. 2014b) for an example of the application of C-Forge to another underwater vehicle.

The C-Forge user starts by defining the architecture of the application, which is a computational model that can be refined and expanded as the development process advances, and which is later on transformed and linked to algorithm libraries to execute the application in FraCC. Fig. 7 shows the architecture of the BUSCAMOS-Oil system (ASV and UUV) modeled with the graphical tool included in C-Forge. As can be seen, both the ASV and UUV architectures are very similar. It is worth noting that the components *C_IncidentManager* and *C_EnergyManager* are included to ensure long-term vehicle autonomy and to improve the system's resilient capabilities.

CRIOConnectors: interface with the CRIO hardware controller (SBRIO 9606). The CRIO acquires acoustic data from imaging sonar via serial port and are sending to the C_ASVObstacleDetector. It also obtains sensor data (Doppler velocity sensor, pressure/depth sensor, inertial unit, etc.) to provide information on distance from seafloor, depth, etc.

C_IncidentManager. Manages failures and problems that may arise in long-term missions, especially in the hostile marine environment. It monitors several system parameters (e.g. status of motors, sensors, etc.) and executes actions to overcome any detected problems. For instance, it can ask *C_MissionPlanner* to change the sequencing of a mission.

C_EnergyManager. Manages the energy produced and consumed by the vehicle, ensuring the vehicle power supply during long-term maneuvers. It monitors the photovoltaic system, activates the back-up diesel generator when needed, and enables low power modes when possible. It reports incidents to *C_IncidentManager*.

C_UserInterface allows the operator to interact with the system in order to define and plan the mission, as well as to receive perception information, like video and sonar images, historic sensor information, etc.

C_MissionPlanner. Since missions are composed of simple actions, e.g. *"ASV performs a route of parallel transepts"*, then *"UUV does an immersion in transepts"*, etc., *C_MissionPlanner* is responsible for sending the activities to be performed to the *C_MissionSequencer* of each vehicle, as well as monitoring the mission status. It receives alerts from *C_IncidentManager*.

C_ASVMissionSequencer, C_UUVMissionSequencer. These sequencing components ensure the implementation and dispatching of individual actions to the ASV and UUV controllers and correct the plan if necessary.

C_ASVLocalizer determines the absolute position of the ASV from the GPS coordinates and compass data.

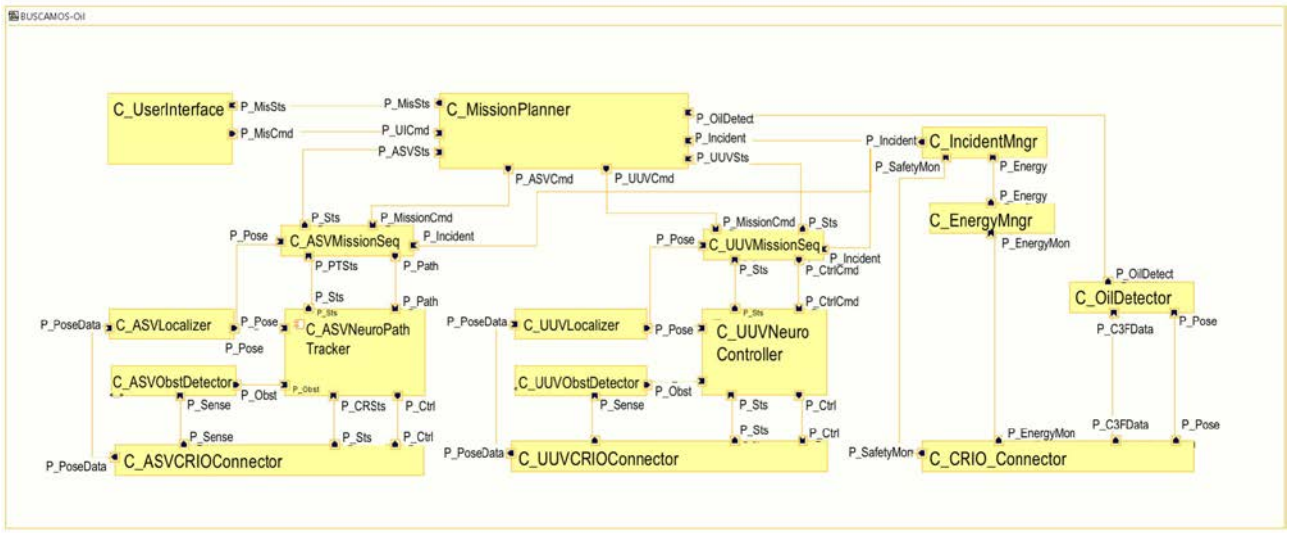

Fig. 7 Complete software architecture of the BUSCAMOS-Oil system

C_OilDetector collects data from the fluorometers equipped in both the ASV and UUV and fuses the data with the absolute position of the vehicles.

C_ASVNeuroPathTracker. This component ensures the optimal path is followed between setpoints and adapts to any previously learned contingency that may arise, such as the presence of obstacles in the route of the ASV or excessive swell. This is possible thanks to a biologically inspired neuro-controller with a self-organizing map structure, which generates the velocity setpoints of the propellers from the position setpoints. The network has two operational modes: controlling and learning, which are externally triggered.

C_ASVObstacleDetector interprets the data from the imaging sonar which has been acquired in the CRIO controller in order to detect nearby obstacles. All the acoustic data are connected directly as CS stimuli of the NNAB. This NN has been trained to interpret these stimuli and to translate them into directional changes for obstacle avoidance. These direction increments are provided to the *C_ASVNeuroController*.

C_UUVNeuroController controls the UUV between setpoints. It implements the same biologically-inspired neural controller with a self-organizing map structure as the *C_ASVNeuroController* component.

C_UUV_Localizer merges the information received from the positioning system sonar (USBL de MicroNav) and navigation systems of the UUV for precisely estimating its position. To do this, this component binds to a commercial software (Seanet Pro from Tritech) that is responsible for obtaining the position of the UUV relative to the ASV. This distance is fused with the GPS position of the ASV to get the absolute position of the UUV under water.

The UUV itself obtains the distance travelled from an origin, thanks to the combined use of DVL and the IMU. This measure is accurate but sometimes has to be corrected, as it is based on integrating the velocity over time. When the UUV is under water this last estimation has priority. But when there are discrepancies with the result obtained from the SeaNet Pro software, the UUV rises to the surface and recalculates its position with DVL and IMU from the accurate measurement of the UUV GPS on the Surface.

C_UUVObstacleDetector interprets the data from the imaging sonar in order to detect obstacles near the UUV. It provides any directional changes needed to avoid collisions to the *C_UUVNeuroController*.

Once the architecture of the system has been defined, the next step is to model the components internals and the messages they exchange. If any of the components is already available, it can be now imported and reused. Each component in Fig. 7 is defined as shown in Fig. 8, with its behavior specified by means of an FSM. For this, the user models the application datatypes (e.g. *D_Pose3D* in Fig. 8-1), the messages the components exchange (e.g. *M_Goal* in Fig. 8-2), the activities (e.g. *A_Controller* in Fig. 8-3) that will be executed in each component state, the component ports and the FSM that describes its behavior (Fig. 8-4).

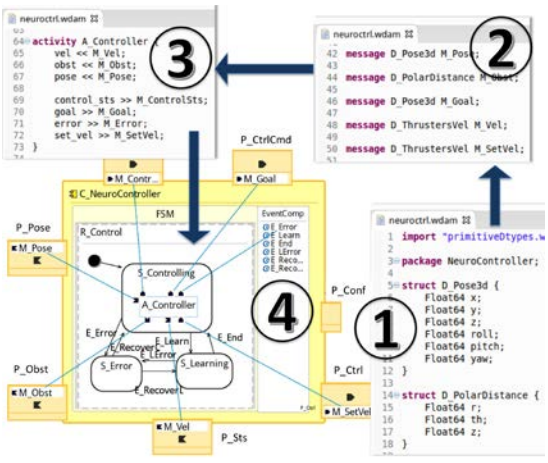

Fig. 8 Relationship among modeling primitives: datatypes with messages, messages with activities, and activities with component states

Starting from the previous models, a model transformation generates (i) code skeletons for all component activities, and (ii) an application deployment model, where the user defines the number of threads and processes that will run the components. Finally, FraCC model loader takes all these artifacts (component models, deployment model and user code) and uses them to run the application. For BUSCAMOS-Oil, the systems architecture presented in Fig. 7 is deployed in three nodes: (i) a ground station including only *C_UserInterface*, (ii) an on-board control computer for the ASV with the mission planner, incident and energy managers and its navigation components, and (iii) an UUV node with its own navigation and control components.

The main characteristics that differentiate C-Forge from other approaches mentioned in Section 2.3 are: (i) the component behavior is explicitly modeled as an FSM instead of being embedded and buried in the code, and (ii) the application code, the component models and the application deployment can evolve and be maintained separately.

5 Autonomous navigation system based on biologically inspired neural algorithms.

Both of BUSCAMOS-Oil's biologically-inspired autonomous marine vehicles (ASV and UUV) use the same neuronal control scheme, which consists of a kinematic SODMN and a NNAB (García-Córdova and Guerrero-González 2013), (Guerrero-González et al. 2010).

All the inputs to the ASV and UUV neurocontrollers are provided by the software architecture. These inputs are the position of the vehicles, and the data from the obstacle-detection imaging sonar. The output of the neurocontrollers are the commands to the UUV and ASV actuators, five commands to the UUV thrusters and three commands to the two ASV thruster and rudder control .

The neural architecture of the autonomous marine vehicles' navigation system is shown below.

5.1 Neural Control System

Fig. 9 illustrates the proposed neural architecture embedded in the components *C_ASVNeuroPathTracker* and *C_UUVNeuroController*, as explained in the previous section. The trajectory tracking control without obstacles is implemented by the SODMN and obstacle avoidance behavior by a NNAB.

For dynamic positioning in path tracking, a filter is incorporated into the control system architecture which smoothes the error signal in reaching objectives.

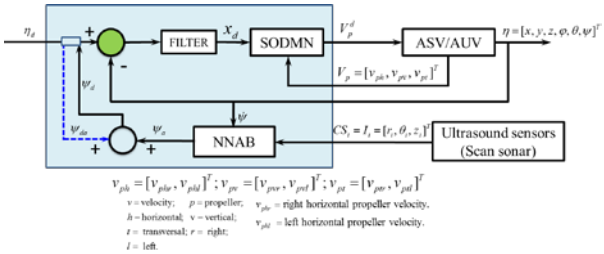

Fig. 9 Neural architecture for reactive and adaptive navigation of the ASV and UUV.

5.2 Trajectory Tracking Control

The SODMN learns to control the robots through a sequence of spontaneously generated random movements (shown in Fig. 10), which enable the neural network to learn the relationship between the angular velocities applied to the propellers and the incremental displacement that ensues during a fixed time step. The proposed SODMN combines associative learning and Vector Associative Map (VAM) learning (Gaudiano and Grossberg 1991; Bullock et al. 1999; Chang and Gaudiano 1998) to generate transformations between spatial coordinates and propeller velocity coordinates. The proposed kinematic adaptive neuro-controller continuously calculates a vectorial difference between the desired and actual velocities. The underwater robot can move to arbitrary distances and angles, even though it has only sampled a small range of displacements during the initial training phase.

The online error-correcting capacity of the architecture endows the controller with many useful properties, such as the ability to reach targets in spite of drastic changes in the robot's parameters or other perturbations.

At a given set of angular velocities the differential relationship between underwater robot movements in spatial coordinates and the angular velocities of the propellers is expressed as linear mapping that varies with propeller velocity.

The transformation of spatial directions to the propeller angular velocity is shown in Fig. 10. The tracking spatial error (e) is computed and filtered to get the desired spatial direction vector ($\boldsymbol{x}_d$) and the actual spatial direction vector (DVs). The DVs is transformed by the direction mapping network elements V_{ik} to the corresponding motor direction vector (DVm). On the other hand, a set of tonically active inhibitory cells which receive broad-based inputs that determine the context of a motor action is implemented as a context field that selects the V_{ik} elements based on the configuration of the propellers' angular velocity.

A speed-control GO signal acts as a nonspecific multiplicative gate and controls the movement's overall speed. The GO signal is an input from a decision center in the brain, and starts at zero before movement and then grows smoothly to a positive value as the movement develops. During learning, the sensed angular velocities of the propellers are fed into the DVm and the GO signal

is inactive (García-Córdova 2007). An analysis of the GO signal shape and its effect on the bell-shaped velocity profile and other properties observed during movements can be found in (Bullock et al. 1993; Bullock et al. 1999). Timing behavior is thus more robust with respect to the fluctuations in the equation parameters and allows the synchronization of concurrent cortical dynamic processes.

The activities of the cells in the DVs are represented in the neural network by quantities (S_1, S_2, ⋯, S_m), while activities of the cells of the DVm are represented by quantities (R_1, R_2, ⋯, R_n). The direction mapping is formed with a field of cells with activities V_{ik}. Each V_{ik} cell receives the complete set of spatial inputs S_j, $j = 1, \dots, m$, but connects to only one R_i cell (see Figure 10). The mechanism that is used to ensure weights converge to the correct linear mapping is similar to the VAM learning construction (Gaudiano and Grossberg 1991). The direction mapping cells ($\mathbf{V} \in \boldsymbol{R} \in \mathrm{R}^{n \times k}$) compute any difference of activity between the spatial and motor direction vectors via feedback from the DVm. During learning, this difference drives the adjustment of the weights. In actual performance the difference drives DVm activity to the value encoded in the learned mapping.

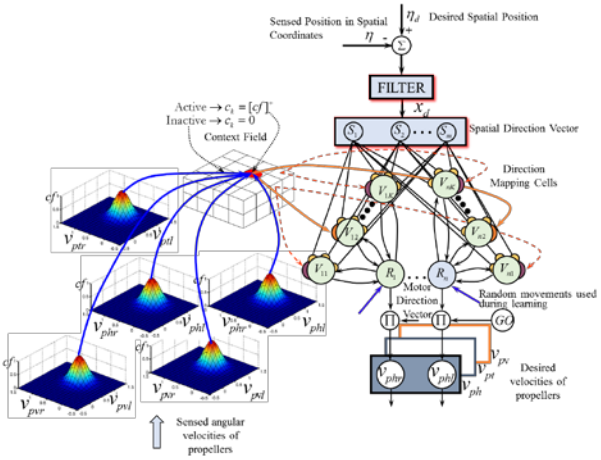

Fig. 10 Self-organization direction mapping network (SODMN) for trajectory tracking of the UUV.

A context field cell pauses when it recognizes a particular velocity state (i.e., a velocity configuration) in its inputs, and thereby disinhibits its target cells. The target cells (direction mapping cells) are completely shut off when their context cells are inactive (see Fig. 10). Each context field cell projects to a set of direction mapping cells, one for each velocity vector component. Each velocity vector component has a set of direction mapping cells associated with it, one for each context. A cell is "*on*" for a compact region of the velocity space. It is assumed for simplicity that only one context field cell turns "*on*" at a time. In Fig. 10, inactive cells in the context field are shown as white disks. The center context field cell is "*on*" when the angular velocities are in the center region of the velocity space, in this three degrees-of-freedom example. The "*on*" context cell enables a subset of direction mapping cells through the inhibition variable ck, while "*off*" context cells disable the other subsets. When the k^{th} context cell is "*off*" or inactive (modeled as c_k=0), in its target cells, the entire input current to the soma is shunted away, so that there is only activity in the axon hillock, which decays to zero. When the k^{th} context cell is "*on*" or active, c_k =1, its target cells (V_{ik}) receive normal input.

The DVs cell activities, $\mathbf{S} \in \mathrm{R}^m$, are driven by the desired spatial direction, $\mathbf{x}_d \in \mathrm{R}^m$, computed from the difference of the desired spatial position and the current spatial position of the robot. Direction mapping cells with activity V_{ik} compute the difference of the weighted DVs input and the DVm activity. The motor direction cell activities, $\mathrm{R} \in R^n$, are driven by the V_{ik} during performance and by propellant velocities V_p during the learning.

Learning takes place by decreasing weights in proportion to the product of the presynaptic and postsynaptic activities (Gaudiano and Grossberg 1991; Chang and Gaudiano 1998; García-Córdova and Guerrero-González 2013; Guerrero-González et al., 2010). The learning rule can therefore be obtained by using the gradient-descent algorithm. Training takes place by generating random movements and by using the resulting angular velocities and observed spatial velocities of the UUV robot as training vectors for the direction mapping network.

5.3 Obstacle Avoidance Adaptive Neuro-Controller

The obstacle avoidance adaptive neuro-controller is a neural network that learns to control avoidance behaviours in a marine vehicle based on a form of animal learning known as *operant conditioning*. Learning, which requires no supervision, takes place as the robot moves around an environment cluttered with obstacles. The NNAB requires no knowledge of the geometry of the robot or of the quality, number, or configuration of the robot's sensors (see Fig. 11). Our implementation is based on Grossberg's conditioning circuit, which follows closely that of (Grossberg 1971; Chang and Gaudiano 1998; Chang 2005).

In this model the sensory cues (both conditioned stimuli (CS) and unconditioned stimuli (UCS)) are stored in the Short Term Memory (STM) within the population labeled S_T, which includes competitive interactions to ensure that the most salient cues are contrast-enhanced and stored in the STM while less salient cues are suppressed.

The S_T population is modeled as a recurrent competitive field in a simplified discrete-time version, which removes the inherent noise, efficiently normalizes and contrast-enhances from the ultrasound sensor activations. In the present model the CS nodes correspond to activation from the each *'pixel'* of the imaging sonar. In the network, I_i represents a sensor value which codes proximal objects

with large values and distal objects with small values. The drive node (D) corresponds to the Reward/Punishment component of operant conditioning (an animal/robot learns the consequences of its own actions).

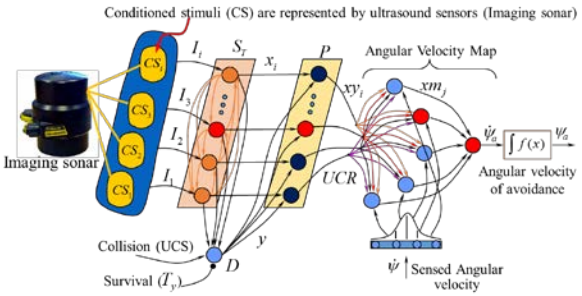

Fig. 11 Neural network for avoidance behaviour (NNAB). A threshold approach is considered a collision (UCS).

Learning can only occur when the D is active. Activation of the D is determined by the weighted sum of all the CS inputs, plus the UCS input, which is presumed to have large, fixed connection strength, plus a homeostatic signal corresponding to a sort of "survival instinct" (T_y), which is active at all times and is a threshold that controls how easily the drive node is activated.

In this proposed neural controller, when the robot reaches an approach threshold (permissible minimum distance between the robot and the obstacle) a collision and a UCS input are considered to perform an escape maneuver.

The drive node is active when the robot collides with an obstacle (the robot reaches the established approach threshold), which could be detected through a collision sensor, or when any one of the sonar sensors indicates that an obstacle is closer than the sensor's minimum range. The unconditioned stimulus in this case then responds to the collision detected by the robot. The activation of the drive node and sensory nodes converges upon the population of polyvalent cells (P). Polyvalent cells require the convergence of two types of inputs in order to become active. Each polyvalent cell receives input from only one sensory node, and all polyvalent cells also receive input from the drive node. Outputs from this type of cell are represented by xy_i and are conditioned or unconditioned responses (CR and UCR, respectively).

Finally, the neurons (xm_j) represent the angular velocity map and are thus connected to the motor system. The motor population consists of nodes (i.e., neurons) encoding desired angular velocities of avoidance, i.e, the activity of a given node corresponds to a particular desired angular velocity for the marine vehicles. When driving the robot, activation is distributed as a Gaussian centered on the desired angular velocity of avoidance. The use of a Gaussian leads to smooth transitions in angular velocity even with few nodes.

The output of the angular velocity population is decomposed by SODMN into angular velocities of the left and right horizontal thrusters. A gain term can be used to specify the maximum possible velocity. In NNAB the proximity sensors initially do not propagate activity to the motor population because the initial weights are small or zero.

The robot is trained by allowing it to make random movements in a cluttered environment. Each node in the angular velocity map is systematically activated for a short time, causing the robot to cover a certain distance and rotate through a certain angle, according to the node activated. Whenever the robot collides with an obstacle during one of these movements or comes very close to it, the nodes corresponding to the largest (closest) proximity sensor measurements just prior to the collision will be active.

Before a collision occurs and before any learning has taken place, the CS node sends activation to its corresponding polyvalent cells. However, the connection from the CS node to the drive node is very weak, so that the drive node does not activate. Hence the polyvalent cell only receives one kind of input and does not become active.

When the same CS node is on when a collision occurs, the UCS causes the drive node to become active. The drive node sends its activation to all polyvalent cells; however, only the polyvalent cell corresponding to the active CS turns itself on, because it is the only one receiving both kinds of input. At this point, the activation of the drive node allows two types of learning to take place simultaneously: the learning that couples sensory nodes (sonar sensors) with the drive node (the collision threshold), and the learning that inhibits the movements of the angular velocity pattern that existed just before the collision.

The first type of learning follows an associative learning law with decay. This learning enables the most active sensory nodes to accrue strength in their connections to the drive node, so that eventually the sensory nodes will be able to activate the drive signal on their own, and thus to activate the polyvalent cells (P), and ultimately a motor response. The primary purpose of this learning scheme is to ensure that learning occurs only for those CS nodes that were active within a time window prior to the collision (UCS).

The second type of learning, which is also associative in type but inhibitory in nature, is used to map the sensor activations to the angular velocity map. This learning takes place between the polyvalent cells and the cells that actually generate the robot´s movements, whereby simultaneous activation of the pre- and post-synaptic cells leads to an increasingly large negative (i.e., inhibitory) weight. By using an inhibitory learning law, the polyvalent cell corresponding to the active sensory nodes acquire negative connection weights that learn to generate a pattern of inhibition matching the angular velocity profile active at the time of collision.

Once learning has occurred, the angular velocity map

is activated by two components. An excitatory component (generated directly by the sensory system) reflects the angular velocity required to reach a given target in the absence of obstacles. For the sake of simplicity, here we assume that the angular velocity is proportional to angle between the robot's current heading and the target. A second, inhibitory component, generated by the conditioning model in response to sensed obstacles, moves the robot away from the obstacles as a result of the activation of sensory signals in the conditioning circuit.

After learning, when the robotic vehicles are required to reach a target by turning at a certain angular velocity $\dot{\psi}$, the actual angular velocity might differ if obstacles are detected. In the presence of obstacles, the polyvalent cells generate inhibitory activation, causing the peak in the angular velocity map to shift, moving the robotic vehicle away from the obstacles and also away from its current desired direction. Once the obstacle has been passed, the vehicle's movements will simply be determined by the desired angular velocity.

6 Experimental results

In order to validate the performance of the BUSCAMOS-Oil system, the system was verified in three different ways. Experiments were conducted to validate and test (i) the capabilities of the UUV adaptive controller (ii) the adaptive and reactive capabilities of the ASV controller and (iii) the system's capacity to carry out a complete mission. These tests were carried out off the Cartagena coast, in the Region of Murcia (Spain) in the south western Mediterranean.

6.1. Reactive and adaptive navigation of UUV

The proposed neural architecture for reactive and adaptive navigation of an UUV is capable of generating optimal trajectory for underwater vehicles in an arbitrarily varying environment. Figure 12 shows the performance of the trajectory tracking controller implemented as a SODMN. These tests were carried out off the Cartagena coast in a 3-D workspace without any obstacles, with an initial position (P_0) at (x, y, z) = (2, 2, 2)m and an initial orientation, shown in Fig. 12, as (φ_0, θ_0, ψ_0). Note that the depth was recorded as positive (Z). The approach behaviors and the tracking of a trajectory by the UUV robot with respect to the reference trajectory are shown in Fig. 12a. The desired trajectory is a sine wave with an initial position of P_{d0} (4, 30, 9)m and a final position of (40, 60, 9)m.

In our NNAB model, the range sensors initially do not propagate activity to the neural motor population because the initial weights are small or zero. Both marine vehicles are trained by allowing them to carry out random movements in a cluttered environment. Collisions (approach thresholds) in the experiments and learning phases were established at 5m and 2m for the ASV and UUV, respectively. The goal of the training phase is to give each CS node the opportunity to sample different movements that lead to collisions.

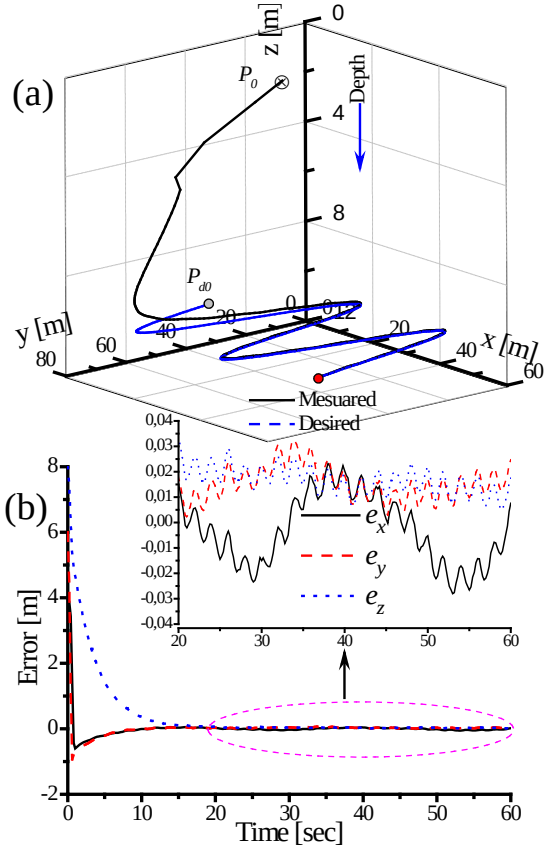

Fig. 12 Tracking control of a desired trajectory. (a) The tracking performance of the UUV. (b) Tracking error.

In practice we found that it is sufficient for each CS node to be active for only a small number of collisions when using 12 nodes in the angular velocity map. In order to generate a wide range of movements, during the training phase we turn on each node in the angular map for a brief time until a collision is registered, then switch to a new angular map node and repeat the process. This means good avoidance behavior is achieved with only a few collisions for each node. Figure 13a illustrates the learning process. This curve was obtained by starting with all the weights in the network set to zero. We turned on one node in the angular map and let the robot collide with an obstacle, generating a small amount of learning. Then another node is turned on, the process is repeated, and so on. At regular intervals during the training phase, we temporarily disable learning and allow the robot to move from a new starting position for a total of 500 steps through the cluttered environment, and count the step number at which a collision was detected. In the first trial, before any learning has taken place, as soon as the robot collides, it remains immobile against the obstacle, so the number of collisions is very close to 500. By the time we have trained it through 50 collisions the robot is able to

navigate with virtually no collisions. This means that each of the sensory nodes, on average, has sampled fewer than ten collisions and the signal of the avoidance sonar is decomposed into ten sensory nodes.

Figure 13b shows the inhibitory weights developed by the neural network, which are represented by the projections of the adaptive connections between the sensory nodes (xy_i) and the angular velocity map (xm_j). The adaptive connections between the sensory nodes and the angular velocity map develop in such a way that angular velocities that make the robot turn to the right (nodes close to 20) are inhibited when the sensors located on the right side of the robot are active (sensory nodes 6 and 10). Similar yet opposite inhibitory weights develop for left turns when obstacles are sensed on the left side. In the center of Fig. 13b (nearly straight-forward movements with obstacles located straight ahead), a Gaussian-like inhibitory curve accounts for the fact that in such cases turns to either the left or right are needed to avoid collisions.

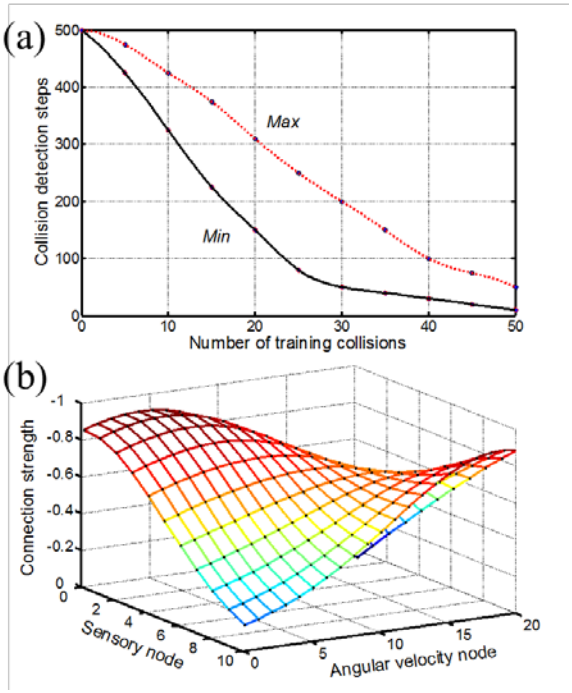

Fig. 13 Results of the learning process of NNAB. (a) Learning in the UUV. (b) Adaptive connections in sensory nodes. The collision threshold was 2 m.

In the first stage, the NNAB algorithm was allowed to set the weights by letting the robot carry out movements at different angular velocities in an environment cluttered with obstacles, by activating sequentially the nodes in the angular velocity map. After this initial learning phase the robot is able to avoid obstacles in arbitrary positions.

Figure 14 shows the performance of the UUV with the presence of several obstacles. The underwater vehicle starts from the initial position P_0=(10,30,1)m and reaches a desired position (goal). During the movements, whenever the UUV is approaching an obstacle (boat), the inhibitory profile from the conditioning circuit (NNAB) changes the selected angular velocity and makes the vehicle turn away from the obstacle. The presence of multiple obstacles at different positions in the UUV's sensory field causes a complex pattern of activation that steers the underwater robot between the obstacles.

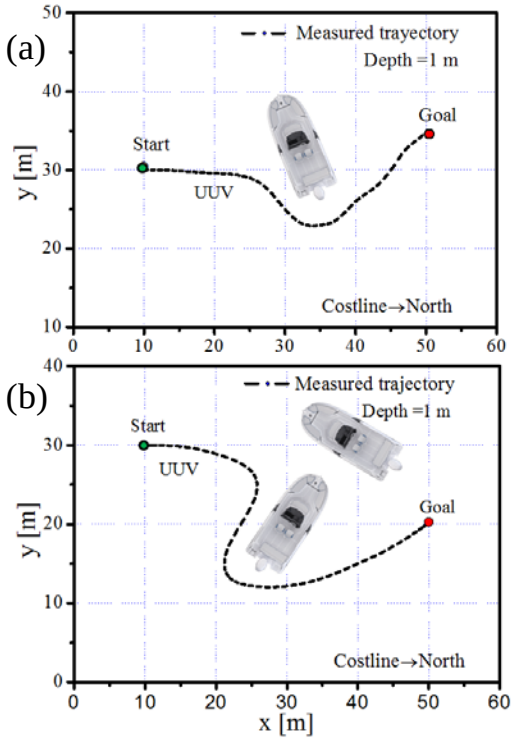

Fig. 14 Obstacle avoidance trajectory of the UUV with a goal of (50, 35, 1)m in (a) and of (50, 20, 1)m in (b).

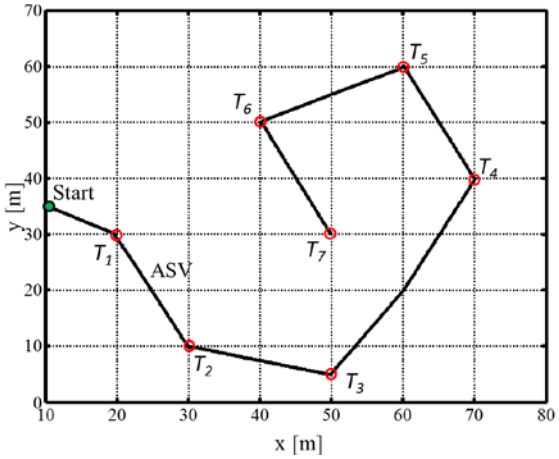

Fig. 15 Reach to different desired targets of the ASV.

6.2. ASV reactive and adaptive navigation.

Figure 15 shows the path traveled by the ASV when it reaches different targets (T_i). The ASV starts from the position P_0(10,35)m, and reaches the following desired targets: T_1(20,30)m → T_2(30,10)m → T_3(50,5)m → T_4(70,40)m → T_5(60,60)m → T_6(40,50)m → T_7(50,30)m. The approach behavior and the tracking by the ASV of a desired sinusoidal trajectory where the vehicle starts from

the position P_0(20,35)m is shown in Figure 16.

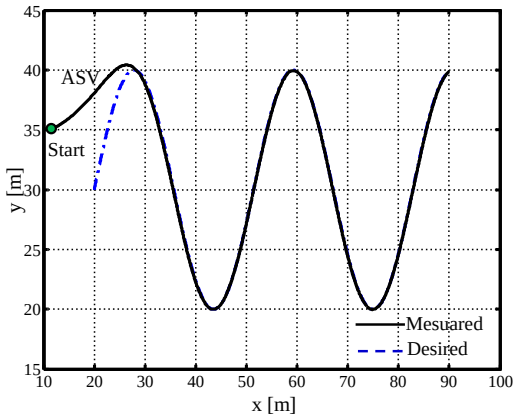

Fig 16. Tracking control of a desired sinusoidal trajectory of the ASV.

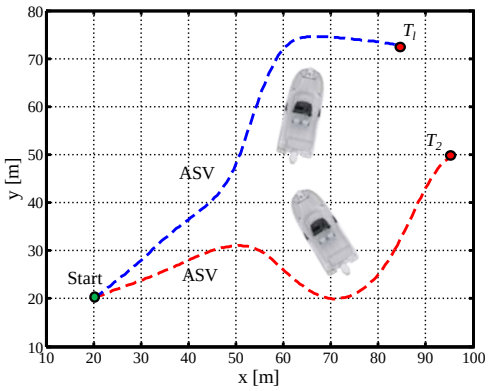

Fig 17. Obstacle avoidance trajectory of the ASV with a goal of (85,35)m and of (95,50)m.

Figure 17 shows the ASV performance in the presence of several obstacles. The ASV starts from the initial position P_0=(20,20)m and reaches two desired targets, with T_1(85,72)m and T_2(95,50)m.

Figure 18 shows the navigation paths of the ASV with different faults when it reaches a target. The experiment was carried out on the coast of Cartagena, Murcia. The ASV starts from the position *Test*ASV1(50, 15)m, and reaches the desired target *Test*ASV2(750, 218)m. The initial navigation path of the vehicle is generated by the SODMN using the 2 propellers and rudder control (2Ps-RC). The actuator faults were simulated, and consisted of the failure of an ASV actuator to navigate in underactuated mode. When the vehicle has traveled a distance of approximately 156m from the start position, it initiates a simulated actuator failure. The proposed autonomous navigation system adapts to changes suffered by the ASV to navigate in underactuated mode by using either the left propeller and rudder control (1 LP-RC), the right propeller and rudder control (RC RP-1) or both propellers (2Ps). The trajectories generated with simulated faults in the actuators are shown in Fig. 19b. The deviation errors with respect to the generated trajectory when using 2Ps-RC are shown in Fig. 20.

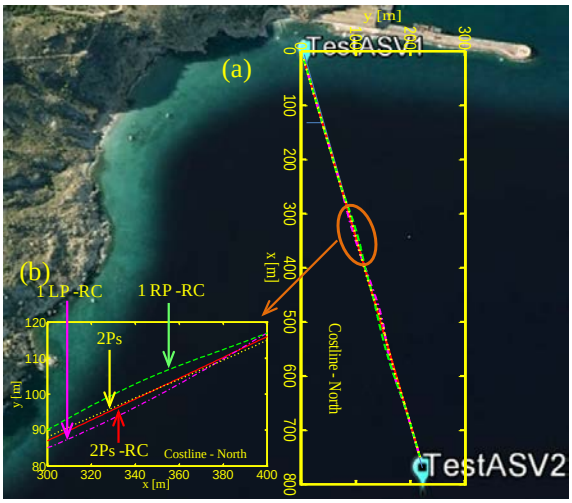

Fig 18. Navigation trajectories of the ASV with different faults when it reaches a goal of *Test*ASV2(750, 218)m.

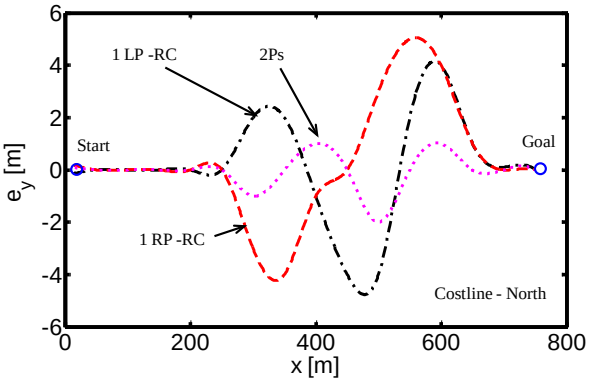

Fig 19. Deviation errors with respect to the generated trajectory by using 2Ps-RC.

6.3 Experimental long-term mission.

The experiment involved determining the extent, velocity and direction of a rhodamine spill (see Fig. 20). The spill had to be updated every day during the experiment. The test lasted for 5 days in September and was carried out in coastal waters near Cartagena (Spain).

Fig. 20. Rhodamine spill in the long-term test.

The mission was configured in the remote land base, indicating the initial exploration point E0, length (1Km) and width (100m) of the searching transects, and was downloaded to the system via wifi. During the mission, the vehicle reported the state information (operation mode, GPS coordinate, size, direction and velocity of the rhodamine spill) through radio modem to the remote base station.

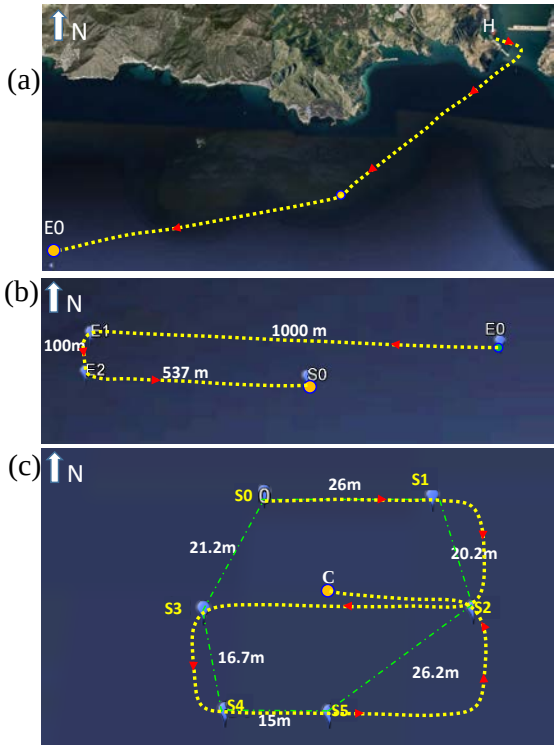

Fig 21. (a) Displacement to the operation point, (b) Searching of the rhodamine spill, (c) Determination of the spill dimension.

The ASV with the integrated UUV initiated the maneuver at 8:00 in the morning in Cartagena Bay (Point H in Fig. 21a) and navigated autonomously to the initial exploration point E0 at 4.26 nautical miles in approximately two hours. Once at E0, it began a searching maneuver by means of straight transects in the east-west direction for 1000 meters and in parallel lanes 100 meters wide. The system detected rhodamine at the S0 point (see Fig. 21b). From this point it began the maneuver to determine the extent of the spill on the surface by sailing straight parallel east-west courses, each 20 meters in length. The vehicle changed direction when it failed to detect rhodamine. The vehicle followed a path from S0 to S5 and finally went to the center of the spill at Point C. Figure 21c shows the vehicle's actual trajectory in yellow. The extent of the spill on the surface was determined by the discontinuous green polygon, which connects the S0-S5 points, and is shown in Fig.21c.

Fig 22. Launching the UUV.

The UUV was launched at Point C (see Fig.22) and began an underwater zig-zag to the north to determine the extent of the spill.

The UUV detected the spill limits at Points U1, U2 y U3, as indicated in Figure 23. With these points and the perimeter data of the spill on the surface, the spill profile at depth in the North-South plane was determined. This procedure was repeated in the East-West plane to obtain the profile in this plane.

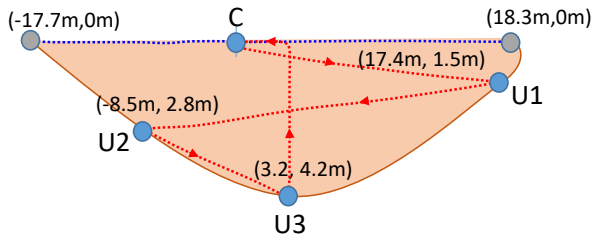

Fig 23. Limit points at depth obtained by the UUV for the North-South plane.

After the UUV had completed operations, the vehicle was recovered and shipped in the ASV. Thereafter, the ASV-UUV stayed at Point C to monitor the spill. The entire maneuver was terminated at 11:55 am. The system contains a programmed event which initiates the search for and characterization of the spill if the sensors stop detecting a spill.

From the data obtained from the ASV and UUV vehicles, a graph of spill concentration values was obtained. Figure 24 shows the evolution of the rhodamine in ppb at depths of 0, 2 and 4 meters.

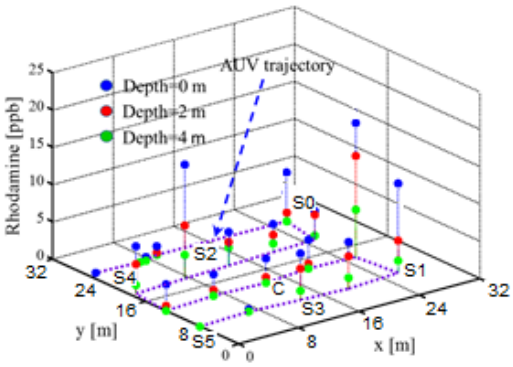

Fig. 24 Measurements of the oil spill reported by BUSCAMOS-OIL

For 5 days the spill monitoring maneuver was carried out every 24 hours, starting at 11:00 am to take advantage of the hours of maximum energy generation. The evolution of the center of the spill during this period is shown in Figure 25. The spill moved 2.25 miles following a course of 030º. After determining the size of the spill on the 5th day, it returned to the base Point H. A total distance of 2.23 miles was covered and the mission concluded at 14:37 hours on Day 5.

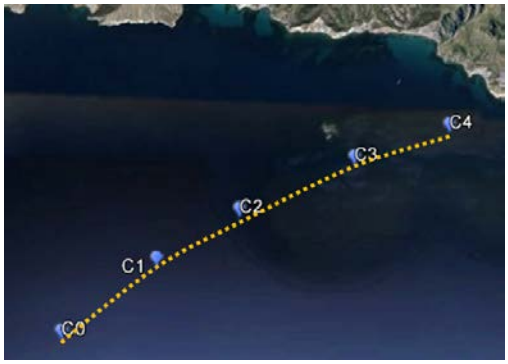

Fig 25. Direction and velocity of the rhodamine spill.

The ASV power system fully satisfied the energy demands of this operation. In Figure 26, the graphics of production and power generation for the mission are shown. In the displacement to the operation point and search and the return to base, consumption exceeded power generation, but over the spill, the data acquisition process required less power than that generated. The overall balance was positive, with more energy produced than consumed.

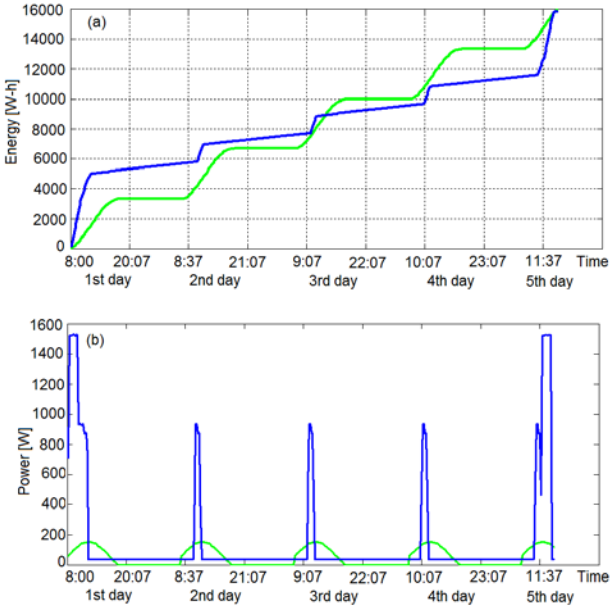

Fig. 26 Measurements reported by BUSCAMOS-OIL

7 Conclusions

This paper presents the BUSCAMOS-Oil monitoring system composed of an ASV and an associated UUV, designed for long-term oil-spill monitoring missions. The system searches for the spill from an initial exploration position. After locating the spill, it measures its extent both on the surface and under water and remains over the spill to monitor its evolution. This approach has great advantages as the slick can be characterized on the surface from the ASV and under water by the UUV. Even though the system can operate autonomously, it maintains permanent wireless communications with the base station. It also generates all its own power supplies, permanently monitors the spill evolution and can act as a drifting buoy over the spill for as long as necessary, thanks to its solar energy generation system.

BUSCAMOS-Oil is designed for long-term operations, has long-range power autonomy, constructional robustness, component redundancy and resiliency. These features are supported by a software architecture that includes dedicated systems for handling power supply and equipment redundancy, as well as two bio-inspired neural networks in charge of reactive and adaptive navigation. The neural controller architecture consists of a Self-Organization Direction Mapping Network (SODMN) for trajectory generation and a Neural Network for Avoidance Behaviour (NNAB) for avoiding obstacles. The neural controllers also allow the vehicles to reconfigure themselves on failure or malfunction of the thrusters in a non-stationary environment, as has been demonstrated in the experiments described in the paper.

The C-Forge model-driven toolchain used to develop the control software architecture clearly separates three concerns: application modeling, application deployment and application coding, all of which can be independently developed by different teams.

The validation tests carried out included: 1) tests on the control system for trajectory generation and obstacle avoidance, 2) the platform's resilience in situations of underactuation (e.g. malfunction of the thrusters), and 3) long-term monitoring of a real spill.

The latter experiment monitored the extent, velocity and direction of a rhodamine slick for 5 days in September in coastal waters near Cartagena (Spain). The system searched for the spill from an initial point, determined its extent both on the surface and under water with joint ASV and UUV maneuvers and finally remained over the spill acting as a drifting buoy. The system successfully reported invaluable data for spill monitoring during the entire period.

Future work will include new sensorization and control strategies to design an adaptive spill tracking system.

Acknowledgments

This work was partially supported by the BUSCAMOS Project (ref. 1003211003700) under the program DN8644 COINCIDENTE of the Spanish Defense Ministry, the Region of Murcia's Government Project MISSION-SICUVA (ref. 15374/PI/10), and the Spanish Government's cDrone (ref. TIN2013-45920-R) and ViSelTR (ref. TIN2012-39279) projects. The authors would like to thank Spanish Navy for the cession of a Pluto Unmanned Underwater vehicle

References

Ajemian, R., Daniel Bullock, D., & Grossberg, S. (2000). Kinematic Coordinates In Which Motor Cortical Cells Encode Movement Direction, Journal of Neurophysiology, 8(5), pp. 2191-2203.

Arena, P., De Fiore, S., Fortuna, L., Frasca, M., Patané, P., & Vagliasindi, G. (2008). Reactive navigation through multiscroll systems: from theory to real-time implementation. Autonomous Robots, 25, 123–146.

Antonelli, G., &Chiaverini, S. (2006). Kinematic Control of Platoons of Autonomous Vehicles, IEEE Transactions on Robotics, doi:10.1109/TRO.2006.886272.

Antonelli, G., Chiaverini, S., Sarkar, N., & West, M. (2001).

Adaptive control of an autonomous underwater vehicle: experimental results on ODIN. IEEE Transaction on Control Systems Technology, doi:10.1109/87.944470.

Bandyopadhyay, P. (2005). Trends in Biorobotic Autonomous Undersea Vehicles, IEEE Journal of oceanic engineering, doi: 10.1109/JOE.2005.843748.

Benner Jr., B.A., N.P. Bryner, S.A. Wise, G.W. Mulholland, R.C. Lao, M.F. Fingas. (1990). Polycyclic aromatic hydrocarbon emissions from the combustion of crude oil on water. Environ. Sci. Technol., 24(9):1418–1427.

Bézivin, J. (2005). On the unification power of models. Software and System Modeling, 4(3), 171-188.

Bhattacharya, S., Heidarsson, H., Sukhatme, G., & Kumar, V. (2011). Cooperative control of autonomous surface vehicles for oil skimming and cleanup. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, doi:10.1109/ICRA.2011.5980421.

Bongard, J., Zykov, V., & Lipson, H. (2006) Resilient machines through continuous self-modeling. Science, 314 (5802), 1118-1121.

Bui, L.D., & Kim, Y.G. (2006). An obstacle-avoidance technique for autonomous underwater vehicles based on BK-products of fuzzy relation. Fuzzy Sets and Systems, 157, 560–577.

Bullock, D., Grossberg, S., & Guenther, F. (1993). A self-organizing neural network model of motor equivalent reaching and tool use by a multijoint arm, Journal of Cognitive Neuroscience, 5, 408–435.

Bullock, D., Fiala, J.C., & Grossberg, S. (1994). A neural model of timed response learning in the cerebellum, Neural Networks, 7, 1101–1114.

Bullock, D., Bongers, R. M., Lankhorst, M., & Beek P.J. (1999). A vector-integration-to-endpoint model for performance of viapoint movements, Neural Networks, 12, 1–29.

Burguer, J., (1997). Oil spills. New Jersey: Rutger University Press.

Burns, R. S. (1995). The use of artificial neural networks for the intelligent optimal control of surface ships, IEEE Journal of Oceanic Engeneering., 20(1), 65–72.

Camilli, R., Reddy, C.M., Yoerger, D.R., VanMooy, B.A.S., Jakuba, M.V., Kinsey, J.C., McIntyre, C.P., Sylva, S.P., & Maloney, J.V. (2010) Tracking Hydrocarbon Plume Transport and Biodegradation at Deepwater Horizon. Science 330(6001), 201-204.

Caccia, M., Bibuli, N., · Bono, R., & Bruzzone, G., (2008). Basic navigation, guidance and control of an Unmanned Surface Vehicle. Autonomuos Robots, doi:10.1007/s10514-008-9100-0.

Carreras, M., Yuh, J., Batlle. J., & Ridao, P. (2005). A behavior-based scheme using reinforcement learning for autonomous underwater vehicles, IEEE Journal of Oceanic Engineering, 30(2), 416-427.

Chang, C. (2005). Using Sensor Habituation in Mobile Robots to Reduce Oscillatory Movements in Narrow Corridors. IEEE Transaction on neural networks, 16(6), 1582-1589.

Chang, C., & Gaudiano P. (1998). Application of biological learning theories to mobile robot avoidance and approach behaviors. Jorunal of Complex Systems, 1, 79–114.

Evans, D.D., G.W. Mulholland, H.R. Baum, W.D. Walton, K.B. McGrattan. (2001). In Situ Burning of Oil Spills. Journal of Research of the National Institute of Standards and Technology. 106:231–278.

Fingas, M. 2013. The basics of oil spill cleanup. CRC Press. Boca Raton. USA. 3rd. ed. 286 pp.

Fingas, M., C. Brown. 2014. Review of oil spill remote sensing. Marine Pollution Bulletin 83(1):9–23.

García-Córdova, F., (2007). A cortical network for control of voluntary movements in a robot finger. Neurocomputing, 71, 374-391.

García-Córdova. F., & Guerrero-González, A. (2013). Intelligent Navigation for a Solar Powered Unmanned Underwater Vehicle, International Journal of Advanced Robotic Systems, doi: 10.5772/56029.

Gaudiano, P., &Grossberg, S. (1991). Vector associative maps: unsupervised real-time error-based learning and control of movement trajectories. Neural Networks, 4, 147–183.

Gerkey, B., Conley, K., Faust, J., Foote, T., & Quigley, M. (2009). ROS: an open-source Robot Operating System. *Proceedings of ICRA Workshop on Open Source Software*, 3(2), 2-5.

German, C.R., Jakuba, M.V., Kinsey, J.C., Partan, J., Suman, S.; Belani, A., &Yoerger, D.R. (2012). A long-term vision for long-range ship-free deep ocean operations: Persistent presence through coordination of Autonomous Surface Vehicles and Autonomous Underwater Vehicles. doi: 10.1109/UUV.2012. 6380753.

Gracias, N., Ridao, P., Garcia, R., Escartin, J., L'Hour, M., Cibecchini, F., Campos, R., Carreras, M., Ribas, D., Palomeras, N., Magi, L., Palomer, A., Nicosevici, T., Prados, R., Hegedus, R., Neumann, L., de Filippo, F., &Mallios, A. (2013). Mapping the Moon: Using a lightweight UUV to survey the site of the 17th century ship 'La Lune', *Proceedings of OCEANS 2013*, doi: 10.1109/OCEANS-Bergen.2013.6608142.

Grossberg, S. (1971), On the dynamics of operant conditioning, Journal of Theorical Biology, 33, 225-255.

Grossberg, S. (2013). Adaptive Resonance Theory: How a brain learns to consciously attend, learn, and recognize a changing world, Neural Networks, 37, 1–47.

Guerrero-González, A., García-Córdova, F., Ruz-Vila F. (2010). A Solar Powered Autonomous Mobile Vehicle for Monitoring and Surveillance Missions of Long Duration, International Review of Electrical Engineering, 5(4), 1580-1587.

Guglielmelli, E., Asuni, G., Leoni, F., Starita, A., & Dario, P. (2007). A neuro-controller for robotic manipulators based on biologically-inspired visuo-motor co-ordination neural models. In Handbook of neural engineering: Vol. 26. Neural engineering series (pp. 433–448). New York: Wiley/IEEE Press

Guo J., (2009). Maneuvering and control of a biomimetic autonomous underwater vehicle. Autonomous Robots, 26(4), 241–249.

Gutiérrez, J., Villa-Medina, F., &Porta-Gándara, M.A. (2010). Autonomous Surface Vehicle for Measuring Water Body Parameters. *Proceedings of Electronics, Robotics and Automotive Mechanics Conference*, doi:10.1109/CERMA. 2010.68.

Hamilton, A.F.C., & M. Wolpert, D.M. (2002). Controlling the Statistics of Action: Obstacle Avoidance, J. Neurophysiol. 87(5), 2434–2440.

Hollander, D.J., K.H. Freeman, G. Ellis, A.F. Diefendorf, E.B. Peebles, & J. Paul. 2010. Long-lived, sub-surface layers of toxic oil in the deep-sea: A molecular organic and isotopic geochemical approach to understanding their nature, molecular distribution, origin and impact to the northern Gulf of Mexico. *AGU Fall Meeting Abstracts* 1:2.

Ijspeert, A.J. (2008). Central pattern generators for locomotion control in animals and robots: a review. Neural Networks, vol. 21, num. 4, p. 642-653, 2008.

Laschi, C., Asuni, G., Guglielmelli, E., Teti, G., Johansson, R., Konosu, H., Wasik, Z., Chiara-Carrozza, M., & Dario, P. (2008). A bio-inspired predictive sensory-motor coordination scheme for robot reaching and preshaping. Autonomous Robots, 25, 85–101.

Lee, M., & Choi, H.S. (2000). A Robust Neural Controller for Underwater Robot Manipulators, IEEE Transactions on neural networks, 11(6), 1465-1470.

Leonessa, A., VanZwieten, T., & Morel,Y.(2006). Neural network model reference adaptive control of marine vehicles, in Current Trends in Nonlinear Systems and Control. Boston, MA: Birkhäuser, doi: 10.1007/0-8176-4470-9-23.

Li, L., Wang, C., & Xie, G. (2015). A general CPG network and its implementation on the microcontroller. Neurocomputing, 167, 299–305.

Liu Y., Macfadyen A., Ji Z.G. Robert H. Weisberg, R.H., (2013). Monitoring and Modeling the Deepwater Horizon Oil Spill: A Record-Breaking Enterprise. Washington, DC: American

Geophysical Union.
Liu, C., Chen, Q., & Wang, D., (2011). CPG-Inspired Workspace Trajectory Generation and Adaptive Locomotion Control for Quadruped Robots. IEEE Transactions on Systems, Man, and Cybernetics, Part B 41(3), 867-880.
Kawato, M. (1999). Internal models for motor control and trajectory planning, Current Opinion in Neurobiology, 9, 718–727.
Kawato, M., & Samejima, K. (2007), Efficient reinforcement learning: computational theories, neuroscience and robotics, Current Opinion in Neurobiology, 17, 205–212.
Kleindienst, S., J.H. Paul, S.B. Joye. (2015). Using dispersants after oil spills: impacts on the composition and activity of microbial communities. Journal name: Nature Reviews Microbiology Volume: 13:Pages: 388–396.
Klemas, V (2012). Tracking and monitoring oil slicks using remote sensing. In proceedings of IEEE/OES Baltic International Symposium, doi: 10.1109/BALTIC.2012.6249216
Kinsey, J. C., Yoerger, D. R., Jakuba, M. V., Camilli, R., Fisher, C. R., & German C. R. (2011). Assessing the Deepwater Horizon Oil Spill with the Sentry Autonomous Underwater Vehicle. Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, 261-267.
Krieg, M., &Mohseni, K., (2010). Dynamic Modeling and Control of Biologically Inspired Vortex Ring Thrusters for Underwater Robot Locomotion, IEEE Transactions on Robotics. 26(3), 542-554.
Nyankson, E., O. Olasehinde, V.T. John, R.B. Gupta. (2015). Surfactant-Loaded Halloysite Clay Nanotube Dispersants for Crude Oil Spill Remediation. Ind. Eng. Chem. Res. 54(38):9328–9341.
Norse, E.A.,& Amos, J. (2010). Impacts, Perception, and Policy Implications of the Deepwater Horizon Oil and Gas Disaster. Environmental Law Institute, Washington, DC, 40, 11058.
Ortiz, F.J. , Insaurralde, C.C., Alonso, D., Sánchez, F. and R. Petillot, Y.R. (2014a) Model-driven analysis and design for software development of autonomous underwater vehicles. Robotica, doi:10.1017/ S0263574714001027.
Ortiz, FJ., Alonso D., Rosique F., Sánchez-Ledesma F. and Pastor JA. (2014b) A Component-Based Meta-Model and Framework in the Model Driven Toolchain C-Forge, Proceedings of Simulation, Modeling, and Programming for Autonomous Robots SIMPAR2014, DOI: 10.1007/978-3-319-11900-7_29, 2014
Pan C.H., Lai X.Z, Simon X. Yang S.X., Wu, M., (2013). An efficient neural network approach to tracking control of an autonomous surface vehicle with unknown dynamics, Expert Systems with Applications, 40, 1629–1635.
Pan, C., Lai, X., Yang, S.Y, & Wu, M. (2015). A bioinspired neural dynamics-based approach to tracking control of autonomous surface vehicles subject to unknown ocean currents. Neural Computing and Applications, 26, 1929–1938.
Paul, J.H., D. Hollander, P. Coble, K.L. Daly, S. Murasko, D. English, J. Basso, J. Delaney, L. McDaniel, C.W. Kovach. (2013). Toxicity and Mutagenicity of Gulf of Mexico Waters During and After the Deepwater Horizon Oil Spill. Environ. Sci. Technol. 47(17):9651–9659.
Peng, Z., Wang, D., Chen, Z., Hu, X., & Lan, W., (2013). Adaptive Dynamic Surface Control for Formations of Autonomous Surface Vehicles With Uncertain Dynamics, IEEE Transactions on control systems technology, 21(2), 513-520.
Prince, R.C., J.D. Butler. (2013). A protocol for assessing the effectiveness of oil spill dispersants in stimulating the biodegradation of oil. Environmental Science and Pollution Research. 21(16):9506-9510.
Petillot, Y., Tena-Ruiz, I., & Lane, D.M., (2001). Underwater vehicle obstacle avoidance and path planning using a multi-beam forward looking sonar. IEEE Journal of Oceanic Engineering , Vol. 26, No. 2, 240-251.
Ryan, J.P., Y. Zhang, H. Thomas, E.V. Rienecker, R.K. Nelson, S.R. Cummings. (2011). A high-resolution survey of a deep hydrocarbon plume in the Gulf of Mexico during the 2010 Macondo blowout. In: Monitoring and Modeling the Deepwater Horizon Oil Spill: A Record-Breaking Enterprise. Geophysical Monograph Series 195:63-75.
Reddy, C.M.,Arey J.S., Seewald, J.S., Sylvaa, S.P, Karin L. Lemkau, K.L., Nelson, R.K., Carmichael, .A., McIntyre, C.P., Fenwick, J., Ventura, G.T., Van Mooy B.A.S., and Camilli R. (2012). Composition and fate of gas and oil released to the water column during the Deepwater Horizon oil spill. Proceedings of the National Academy of Sciences of the United States of America, doi: 10.1073/pnas.1101242108.
Ridao P, Yuh J, Sugihara K, Batlle J (2000) On UUV control architecture. Proceeding of International Conference on Intelligent Robots and Systems, doi: 10.1109/IROS.2000. 893126.
Ribas D., Palomeras N., Ridao P., Carreras M., &Mallios A., (2012). Girona 500 UUV: From Survey to Intervention. IEEE/ASME Transactions on Mechatronics, doi: 10.1109/TMECH.2011. 2174065.
Sammarco, P.W., Kolian, S.R., Warby, R.A.F., Bouldin, J.L., Subra, W.A., Porter, S.A., (2013). Distribution and concentrations of petroleum hydrocarbons associated with the BP/Deepwater Horizon Oil Spill, Gulf of Mexico. Marine Pollution Bulletin, 73(1), 129-143.
Schlegel G., Steck A., & Lotz A. (2012). Robotic Software Systems: From Code-Driven to Model-Driven Software Development. Robotic Systems - Applications, Control and Programming, DOI: 10.5772/25896.
Schofield, O., &Glenn, S.M. (2004). Introduction to special section: Coastal Ocean Observatories. Journal of Geophysical Research, doi:10.1029/2004JC002577
Sfakiotakis, M., & Tsakiris, D.P. (2007). Neuromuscular control of reactive behaviors for undulatory robots, Neurocomputing, 70, 1907–1913.
Shakhimardanov, A., Hochgeschwender, N. &Kraetzschmar, G. (2010), Component models in robotics software. Proceedings of the Performance Metrics for Intelligent Systems Workshop, doi:10.1145/2377576.2377592.
Sun Z.H, Yang G.S., Zhang B., & Zhang W.J. (2011) On the Concept of the Resilient Machine. Proceedings of 6th IEEE Conference on Industrial Electronics and Applications, doi: 10.1109/ICIEA.2011.5975608
Tee K. P., &Ge, S. S.. (2006). Control of fully actuated ocean surface vessels using a class of feedforward approximators, IEEE Transactions on Control Systems Technolony, 14(4), 750–756.
Tolu, S., Vanegas, M., Luque, N.R., Garrido, J.A., & Ros, E. (2012). Bio-inspired adaptive feedback error learning architecture for motor control, Biological Cybernetics, 106, 507–522.
Wei, M., &Chen, G., (2011). Adaptive RBF Neural Network sliding Mode Control for Ship Course Control System, Third International Conference on Intelligent Human-Machine Systems and Cybernetics, 2, 27 – 30.
Yu, J., Tan, M., Chen, J., & Zhang, J. (2014), A Survey on CPG-Inspired Control Models and System Implementation, IEEE Transactions on neural networks and learning systems, 25(3), 441-456.
Yue, S., D. Santer, R., Yamawaki, Y., & Rind, F. C. (2010). Reactive direction control for a mobile robot: a locust-like control of escape direction emerges when a bilateral pair of model locust visual neurons are integrated. Autonomous Robots, 28, 151–167.
Zhang, Y., R.S. McEwen, J.P. Ryan, J.G. Bellingham, H. Thomas. 2011. A Peak-Capture Algorithm Used on an Autonomous Underwater Vehicle in the 2010 Gulf of Mexico. Journal of Field Robotics 28(4):484-496.