# Wireless HROV Control with Compressed Visual Feedback Using Acoustic and RF Links

Diego Centelles[1] · Antonio Soriano[2] · Raul Marin[1] · Pedro J. Sanz[1]

## Abstract

Underwater cooperative robotics offers the possibility to perform challenging intervention applications, such as recovering archeological objects as within the context of the MERBOTS research project, or grasping, transporting and assembly of big objects, using more than one mobile manipulator, as faced by the TWINBOT project. In order to enhance safety during the intervention, it is reasonable to avoid the umbilical, also giving more mobility to the robots, and enabling a broader set of cooperative movements. Several solutions, based on acoustic, radiofrequency (RF) or Visual Light Communication (VLC) have been proposed for underwater communications in the literature. This paper presents the architecture of an underwater wireless communication framework for the control of multiple semi-autonomous robots in cooperative interventions. The proposed framework is composed of several modules as the virtual reality interface using UWSim, the Underwater Multi-robot Cooperative Intervention Remote Control Protocol (UMCI-RCP) and a Generic Link Layer (GLL). UMCI-RCP allows the control of an underwater robot over limited communication links. UMCI-RCP integrates a progressive compression algorithm that provides visual feedback at a constant rate and ensures image reception even in channels with loses. The Time Division Multiple Access (TDMA) medium access strategy minimizes the jitter of transmitted packets. The GLL has been designed in order to provide support for multimodal transmission (i.e. acoustic, RF and VLC) and also to interface with the UWSim-NET simulator so that facilitates the experimentation either with a real or with a simulated modem. The possibility of exchange real and simulated devices in the proposed framework are demonstrated by means of a teleoperation experiment with a BlueROV equipped with the S100 RF modems. Hardware-In-the-Loop (HIL) capabilities are demonstrated repeating the experiment with the real modems and modeling the BlueROV, and also modeling both the modems and the BlueROV.

## 1 Introduction

The use of robots in underwater operations such as archaeology, marine research and off-shore industries are becoming more necessary in order to enhance safety, while increasing the possibilities to perform advanced interventions in underwater hazardous environments. In fact, some tasks require the use of sophisticated cooperative robotic teams, in order for example to be able to transport and assemble big objects

✉ Diego Centelles
centelld@uji.es

1 Departament d'Enginyeria i Ciència dels Computadors, Universitat Jaume I, Av. Sos Baynat, s/n, 12071, Castelló de la Plana, Spain

2 Departament d'Informàtica, Universitat de València, Av de la Universitat s/n, 46100, Burjassot, Spain

and structures [29]. These refined applications are faced nowadays by research projects such as MERBOTS [11] and TWINBOT [12], being necessary to avoid umbilicals, in order to enable higher mobility and security of the robot's team.

Acoustic communication for underwater applications has the advantage of long range connectivity (i.e. several kilometers). But it can be disturbed by multi-path and acoustic noise, affecting to the communication quality (e.g. variable time-delays). In practice, the range of frequencies is limited to a few kHz. Besides this, the low bandwidth and associated delays ($\approx$ 0.6 ms/m) make difficult to implement a full-duplex underwater robot remote control with this technology. Moreover, the multi-path effect in the vicinity of solid objects (e.g. rocks, ships, and robots) needs a serious study in order to design a communication framework for underwater robot's team. Despite all these inconveniences, underwater video transmission through

acoustic channels has been demonstrated in previous experiments, thanks to the use of advanced compression techniques for very constrained links [24].

Some previous works have presented very interesting solutions for communicating with a team of wireless underwater mobile sensors, focusing on the use of sonar solutions. In fact, in [26] a network framework architecture has demonstrated to be very useful during real survey and data acquisition interventions with USCs and AUVs. More details on the low level behaviour of underwater sensor networks based on acoustic modems, evaluating their performance in real field survey experiments, can be found in [3].

Table 1 shows some acoustic based solutions and their performance in terms of data rate, range, and the supported link type.

Visual light communication (VLC) [13] can be considered as an alternative. VLC provides larger bandwidth than acoustic links, and its range is limited due to the strong attenuation of light in water. It can be considered a solution in good visibility conditions for robot teams communication. Table 2 shows some VLC based solutions.

Another alternative to acoustic links is based on radio frequency (RF) modems [6], which are not affected by multipath, turbidity and water light conditions. However, the high conductivity of sea water limits its communication range to a few meters, being necessary to reduce the transmission frequency, augment signal power, and design specific antennas. The authors in [27] conclude that an RF link over a distance of 90 m in sea water could be established using frequencies up to 5 MHz. Also, in [8] experiments with the WFS Seatooth S100-L modem using external loop antennas are presented demonstrating ranges up to 30 m through seawater. For cooperative robotics RF is an interesting solution due to the fact that it provides lower and constant time-delays, which is very convenient for team intelligent behaviours.

### 1.1 Outline

The present paper describes some of the wireless communications experiments performed for the MERBOTS Project, while also presenting the new UWSim-NET tool that is being used to design more sophisticated underwater

protocols, specially for the multi-robot TWINBOT scenario. The proposed protocol offers a cross-layer architecture for network optimization, which focuses specially on the transport layer, continuing previous work on Internet protocols for master-slave robotic control Bilateral Transport Protocol (i.e. Bilateral Transport Protocol- BTP [30]).

This paper also presents the network architecture to remotely control a experimental underwater robot, having compressed visual feedback, specially optimised for very constrained RF/sonar channels. This architecture includes also a user interface to configure the communication parameters.

The paper is organized as follows:

- First of all, the context of MERBOTS and TWINBOT projects is presented, where the necessity of wireless underwater communications is justified.
- Secondly, the designed communication framework for the underwater wireless control of a single robot is described in detail, including the data frames and the particularities of the cross-layer protocol, considering experiments and tools for image compression, and region of interest selection. The framework provides a user interface that enables the expert operator to select communication parameters, in terms of image resolution, size, and congestion control, among others.
- Finally, in order to be able to design more advanced underwater wireless protocols for robot teams, in the context of the TWINBOT Project, the UWSim simulator has been extended, jointly with NS3 (Network Simulator 3). The new UWSim-NET simulator is described and provided as open source to the research community for further research on network protocols for underwater multi-robot cooperative teams.

## 2 MERBOTS and TWINBOT Projects

This work has been conducted within the framework of the MERBOTS project [11], including further experiments to prepare the communication architecture of the more advanced underwater cooperative mobile manipulators TWINBOT project. MERBOTS, which has been funded by the Spanish Government, is aimed at the development of

**Table 1** Acoustic solutions for underwater data transmission

| Data Rate | Range | Band | Link | Year | Ref. |
|-----------|-------|------|------|------|------|
| 19.2 kbps | 2 km | 53 kHz | Vertical | 1991 | [1] |
| 16 kbps | 6.5 km | 20 kHz | Vertical | 1992 | [28] |
| 1200 bps | a few km (telemetry) | 15 kHz | Vertical and Horizontal | 1993 | [16] |
| 150 kbps | (tested at) 10 m | 75 kHz | Vertical | 2003 | [17] |
| 90 kbps | (tested at) 200 m | 115 kHz | Vertical and Horizonal | 2010 | [22] |

**Table 2** VLC solutions for underwater data transmission

| Data Rate | Range | Wavelength | Year | Ref. |
|---|---|---|---|---|
| 10 Mbit/s | 20 m | 450 nm (blue light) | 1995 | [2] |
| 1 Mbit/s | 30 m | 450 nm (blue light) | 1995 | [2] |
| 10 Mbit/s | 100 m | 400-500 nm (blue/green light) | 2010 | [9] |
| 58 Mbit/s | (tested at) 2.5 m | 470 (blue light) | 2013 | [7] |

tools and techniques that permit the semi-autonomous control of cooperative robots to accomplish tasks in underwater archaeological scenarios. In fact, MERBOTS has developed techniques to study or even recover archaeological objects located at the seafloor. Moreover, one of the objectives of the project is to combine different wireless technologies to link the cooperative robots and the user interface located in the surface vehicle, as can be seen in Fig. 1a.

The TWINBOT project aims at achieving a step forward in the underwater intervention state of the art. A set of two I-AUV's will be able to solve strategic missions devoted to cooperative survey and cooperative manipulation (transport and assembly) in a complex scenario (see Fig. 1b). The project faces the problem of underwater cooperative intervention which, in a first stage, will be able to pick up, recover and transport objects such as pipes, using two intervention vehicles, wireless communications, a supervisory control human-robot interface, and an auxiliary robot for giving the user and the robot team external views for enhancing the intervention efficiency. The system uses as knowledge base previous results from the FP7 TRIDENT [21] and MERBOTS [4] projects.

## 3 Underwater Multirobot Cooperative Intervention Network Architecture (UMCI-NA)

The small bandwidth of underwater wireless modems hampers the transmission of all the messages exchanged

between the operator and the ROVs during an intervention. The UMCI-NA (Underwater Multirobot Cooperative Intervention Network Architecture), specific for underwater applications, has been developed in order to make the most of the limited bandwidth available in these scenarios. UMCI-NA is a minimalist architecture, out of the TCP/IP stack for wireless communications. It enables the remote control of a ROV through multi-hop communication links, like those considered in MERBOTS and TWINBOT projects, as illustrated in Fig. 1.

### 3.1 Overall Network Architecture

The goal of the network architecture is to allow the delivery of commands from the operator to the ROVs, and the messages with the ROVs odometry and visual feedback in the opposite direction. The UMCI-NA allows the operator to control each ROV either in velocity or position modes. In velocity mode, the ROV is controlled with a joystic connected to the computer of the operator. The position mode is based on NED (North–East–Down) coordinate system. The operator moves a semitransparent image of the ROV to the desired position. All these controls have been implemented in the Human–Robot Interface (HRI) layer. The appearance of the HRI is shown in Fig. 2. It is on top of the UMCI-NA architecture, as can be appreciated in Fig. 3.

The HRI layer is based on UWSim [18] and ROS (Robot Operating System) [19]. Its function is to collect the information from the operator and send the corresponding

**Fig. 1** **a** Scenario of the MERBOTS project including acoustic communication to the surface and RF/VLC between the mobile manipulator and the auxiliary inspection robot. **b** Scenario of the TWINBOT project for cooperative intervention, including two mobile manipulators and an auxiliary inspection robot
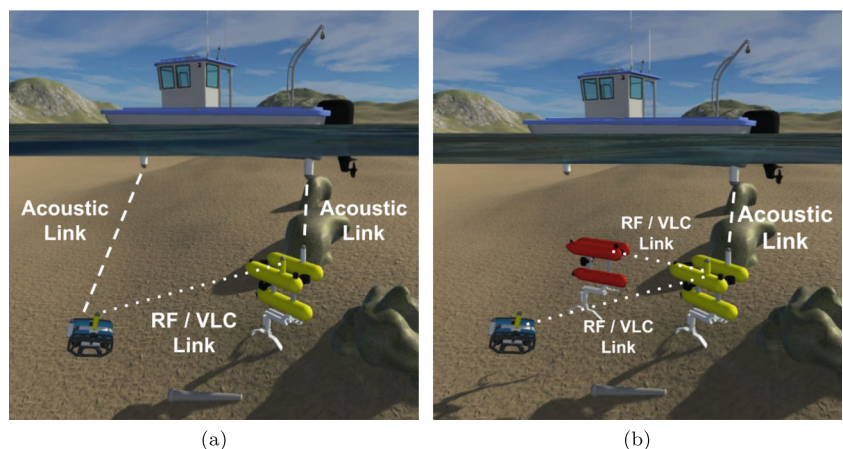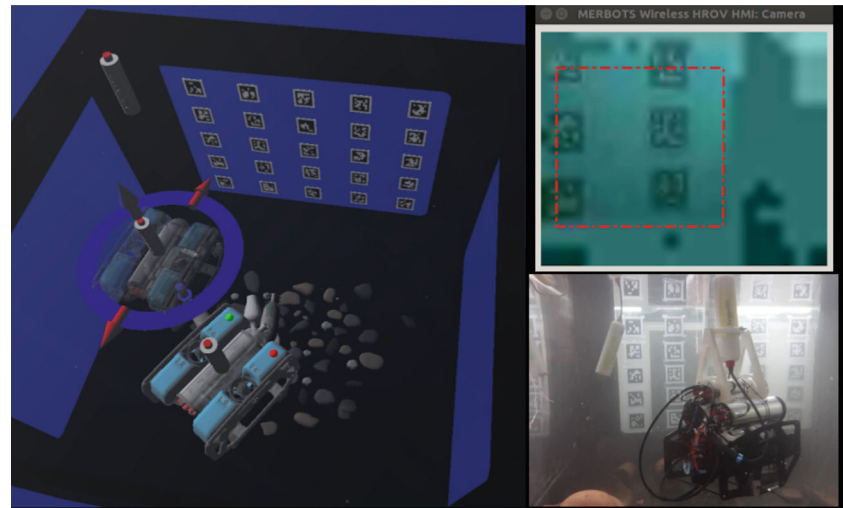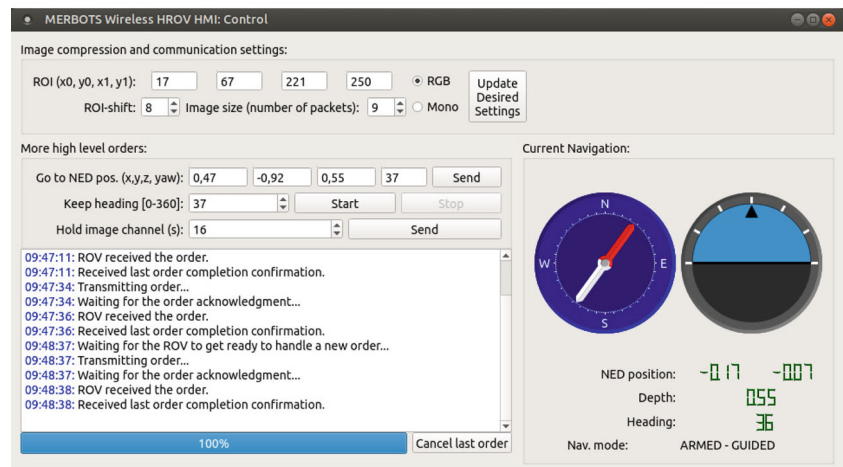


(a)                    (b)

**Fig. 2** Virtual reality Human-Robot Interface: **a** - left UWSim-NET window; **a** - top right Compressed image with ROI from the ROV's camera; **a** - bottom right Real robot; **b** Dedicated UI to send orders and receive feedback messages



(a)



(b)

commands to the ROS layer. The interface also notifies the operator on the reception of the command by the ROV and its accomplishment, while allowing the operator to cancel each command before its completion. The HRI interface incorporates a set of controls that permit to adjust the image feedback to the available bandwidth. By means of high level



**Fig. 3** UMCI-NA architecture

commands, the operator can adjust the image compression parameters like the image size, the number of packets required to send a full image, and the region of interest (ROI). The ROI is an area of the image of special interest that will be codified with improved quality in detriment of the rest of the image. The HRI allows the operator to select the desired ROI by drawing a rectangle on the control that represents the feedback image sent by the ROV.
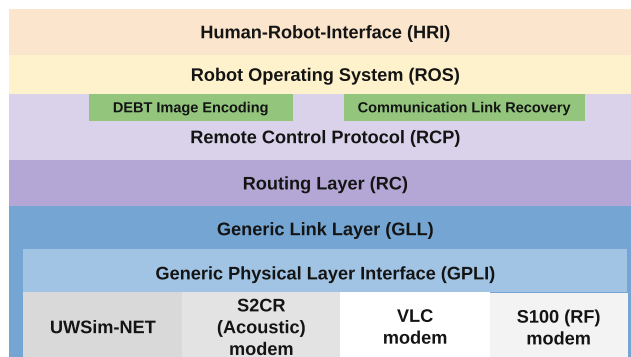
### 3.2 Underwater Multirobot Cooperative Intervention Remote Control Protocol (UMCI-RCP)

The UMCI-NA includes a Remote Control Protocol (UMCI-RCP) that determines how the information is codified into the packets exchanged between the ROV and the operator. The UMCI-RCP also permits the transmission of image feedback from the ROV to operator. The DEBT (Depth Embedded Block Tree) algorithm [25] is embedded into the UMCI-RCP in order to make the most of the small
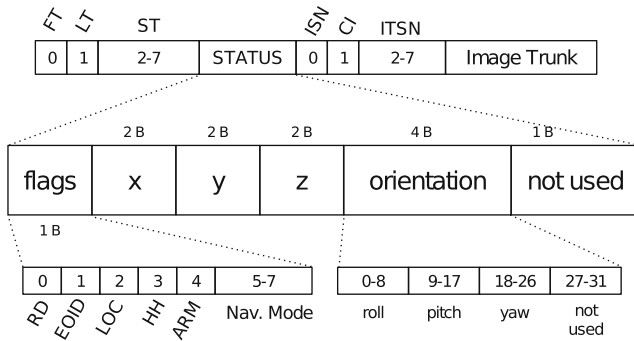
**Fig. 4** Structure of the message transmitted by the ROV, with the detail of the codification of the ROV status

bandwidth available in underwater environments. DEBT is a progressive compression algorithm that allows the user to establish a specific size for the transmitted image. Moreover, DEBT permits the operator to define a ROI in the image that will be codified with higher quality at the expense of reducing the resolution in the rest of the image.

The format of the packets sent by the ROV is shown in Fig. 4. It encodes the current ROV odometry, status flags, packet information, and a portion of the image registered by the ROV.

- **FT** and **LT** are the (*First Trunk*) and (*Last Trunk*) flags, respectively. Their activation indicates that the content in the image trunk field corresponds to the start or the end of an image.
- **ST** (*Status Type*) is a 6 bit field that identifies the structure of the status information codified in the following 12 bytes.
- **ISN** (*Image Sequence Number*): a bit to encode the sequence number of the captured image. to capture a new image and update the ISN.

- **CI** (*Cancel Image*): a bit to indicate the operator that the transmission of the last image has been cancelled and must prepare the reception of a new image.
- **ITSN** (*Image Trunk Sequence Number*): six bits which encode the sequence number of the portion contained in the message of the last captured image.
- **IT** (*Image Trunk*) field containing part of the DEBT compressed image.
- **RD** (*Ready*) remains deactivated if the ROV is executing a high level task.
- **EOID** (*Expected Order ID*) single bit codifying the sequence number of the next high level order the ROV expects to receive.
- **LOC** (*Last Order Cancelled*) one bit flag that is active when the previous order was cancelled because OID and EOID did not match.
- **HH** (*Hold Heading*) is active when the ROV is holding a specific yaw. A high level order is used to activate / deactivate this flag.
- **ARM** (*Armed*) flag that indicates whether the motors of the ROV are active in order to execute operator commands or not.
- **NAV MODE** (*Navigation Mode*) three bits field that codifies the current navigation mode of the ROV. Available modes are: *Manual*, *Stabilize*, *Depth Hold*, *Hold Position* and *Guided*.
- **X, Y, Z** (*NED Position*) six byte field (two bytes per axis) that contains the position of the ROV in a NED coordinate system.
- **Orientation** four bytes field that contains three nine bits values with the roll, pitch, yaw of the ROV. Bits 27 to 31 of this field remain unused.

The format of the packets sent by the operator is shown in Fig. 5a. They contain information related with the desired status of the ROV. The first byte contains control flags. The

**Fig. 5** Format of the messages sent from the operator to the ROV. **a** Message format. **b** Velocity command. **c** Image compression command. **d** "Go To" command
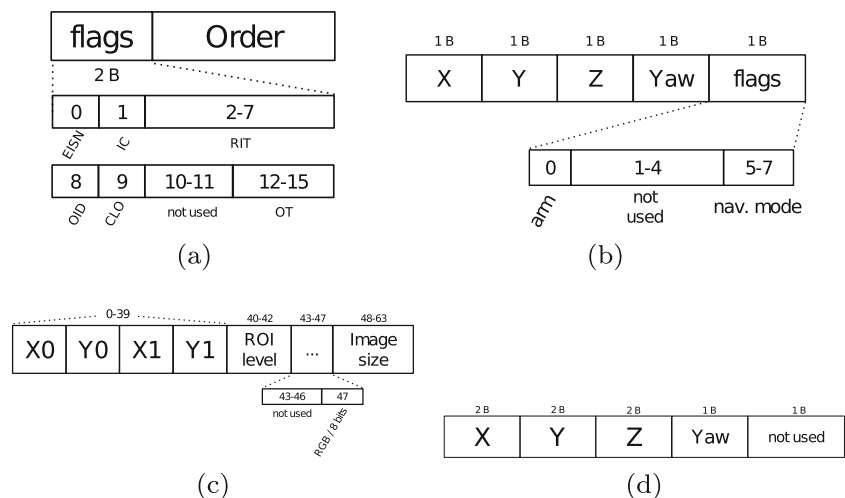
rest of the message is used to encode the command sent to the ROV.

- **OID** (*Order ID*) one bit flag with the sequence number of the order. This field is considered by the ROV with high level orders only.
- **CLO** (*Cancel Last Order*) single bit flag activated to cancel the previous order. The ROV will cancel the order only if the OID matches the last EOID set by the ROV (see Fig. 4).
- **OT** (*Order Type*) four bits field that codifies the type of order sent to the ROV.
- **EISN** (*Expected Image Sequence Number*): encodes in a bit the sequence number of the image that the operator is expecting to receive. It serves to avoid confusing chunks of an image with those of another and to confirm the reception of the whole image.
- **IC** (*Image Cancelled*): confirms to the ROV the cancellation of the reception of the last image. The operator changes the EISN and marks all the image chunks as not received when the reception of the last image is cancelled.
- **RIT** (*Requested Image Trunk*): the operator checks which portions of the image are missing and updates this six-bit field with the sequence number of one of them in order to ask the ROV to retransmit any lost portion of the image.

The commands sent from the operator to the ROV are differentiated as high and low level commands. Low level commands are those that continuously indicate what the ROV is expected to do. The only low level command considered in RCP is the one used to control the ROV in velocity mode, shown in Fig. 5b. High level commands indicate what is the desired status of the ROV. After its reception, the ROV executes a series of operations which allows it to reach the status indicated in the command. The two high level commands considered in RCP are: (1) the one used to control the ROV in position mode (Fig. 5d), and (2) the one that configures the DEBT compression algorithm (see Fig. 5c). RCP is periodically sending messages from the operator to each ROV. High level commands are sent by RCP on operator request. In the absence of a high level command, the RCP sends a low level command, Fig. 5b, with the current status of the joystick.

### 3.2.1 Communication Recovery Module

The RCP integrates a module whose mission is to recover the connection between the robot and the operator when it is lost. The robot considers that the connection has been lost when it does not receive a message from the operator for five seconds. In other words, operator messages can be considered as a heartbeat and the communication is considered recovered when a heartbeat is received again.

The position of the ROV is stacked every two seconds while the communication is active, i.e. heartbeats are received. In case it is lost, the previous ROV positions are recovered from the stack, and the ROV is forced to undo his way back until the communication is recovered. The pseudocode of the Communication Recovery Algorithm (CRA) is shown in Algorithm 1.

---

**Algorithm 1** Communication recovery algorithm.

---

**loop**
    **if** link_ok **then**
        currPos = getCurrentPosition()
        stackPosition(currPos)
        **if** stackSize() > max_stack_size **then**
            removeOldestPositionFromStack()
    **else**
        reachedPos = false
        **do**
            **if** stackSize() > 0 **then**
                prevPos = getLastPositionFromStack()
                currPos = getCurrentPosition()
                reachedPos =
                posReached(prevPos, currPos)
                **if** reachedPos **then**
                    removeLastPositionFromStack()
        **while** reachedPos ∧ stackSize() > 0
        orderGoToPosition(prevPos)
    waitForSeconds(2)

---

### 3.3 Routing Layer

In MERBOTS and TWINBOT scenarios, shown in Fig. 1, one of the ROVs establishes a high speed link with the surface boat by means of an acoustic link or an umbilical cable. The ROVs that cooperate in the underwater scenario are connected via low range RF or VLC modems. In order to extend the communications range, the ROVs form a multi-hop network. ROVs act as routers that re-transmit the traffic from each ROV to the operator in the surface, and vice versa.

The goal of the routing layer is to permit that each RCP message reaches the ROV it is destined for. Due to the small bandwidth available in underwater scenarios, a minimum overhead routing protocol has been proposed. No more than five ROVs are expected to participate simultaneously in the cooperative interventions planned in MERBOTS and TWINBOT experiments (see Fig. 1). The routing protocol is based on the assignment of a four bits address to each ROV that will participate in a given experiment, so that a maximum of sixteen nodes can be addressed, considering both ROVs and operator nodes.

The message received from the RCP layer, shown in Fig. 6, contains a preamble with a fixed value of 0x55, the source (SRC) and destination (DST) addresses, the length of the payload (in bytes), and CRC-16 word to verify the integrity of the message on reception.

### 3.4 Generic Link Layer

As previously explained, there are several technologies for underwater wireless communications, like acoustic, VLC or RF. The behaviour and characteristics of each device vary considerably. For example, the S2CR 18/34 acoustic modems manage the access to the medium while the S100 RF modems do not. The diversity on the capabilities of the different technologies and devices supported by the UMCI-NA architecture requires a Generic Link Layer (GLL) that separates the functionality and the specific capabilities of each device. The GLL is in charge of:

- Execution of the transmission requests from the routing layer.
- Management of the modems output and collection of the received frames, and its forwarding to the routing layer.
- Optimisation of frame transmission and reception depending on the capabilities of each device.

The separation of GLL functionalities and device specific capabilities is accomplished by means of the Generic Physical Layer Interface (GPLI), shown in Fig. 3. GPLI implementation was based on an Inter-Process-Communication Protocol (IPC). It allows to consider all the technologies shown in Fig. 3.

The UWSim-NET network simulator for underwater communications, which is described in detail in Section 4.3, models the behaviour of the acoustic and RF modems to conduct HIL experiments, in those situations that do not allow the use of these communication devices in more realistic scenarios. In this case the GLL only retransmits those packets processing of the packets received from the routing layer to the simulator and vice versa.

The implementation of the module for the S2CR acoustic modem is described in detail in [4]. The data link protocol used by these modems is D-MAC [14]. It defines two operation modes, short instant and burst modes. The burst mode was chosen to implement the S2CR module because it optimises the protocol parameters depending on channel conditions. It is the one with the highest data rate when

transmitting large volumes of data, as in the case of image transmission. Data transmission is optimised to send information from the ROV to the operator. In order to optimise the communication, the operator commands are sent to the ROV attached to the next acknowledgement PDU of the D-MAC protocol.

The VLC module, which is a work in progress, has been already included in the architecture shown in Fig. 3.

#### 3.4.1 WFS RF Modems S100

The model of the RF modem used in these experiments was the S100, manufactured by Wireless For Subsea (WFS) [8]. These modems provide a half–duplex channel through RF communication. They permit transmission rates of 1,9 kbps and a maximum range of 5 meters on sea water.

The S100 modems do not execute any control on the access to the channel so that a medium access control (MAC) must be implemented in order to allow bidirectional communication. In order to maximise the throughput and minimise the delay and the jitter it is required that the MAC protocol reduces the probability of packet collision, and to maximise the channel occupation. The MAC protocol proposed for the S100 modems is based on a token passing strategy. It can be seen as Time Division Multiple Access (TDMA) controlled by the operator side of the channel that will behave as master and will send packets to the ROV periodically. The ROV side will act as slave, and will transmit the ROV odometry and image feedback only on reception of the messages from the operator. The time gap between the sending of two consecutive messages is named Inter Packet Gap (IPG). It is adjusted by the operator considering the propagation delay and the transmission time of both the operator and ROV messages.

## 4 Experimental Setup

### 4.1 Upgrade of the BlueROV Platform to Enable RF Wireless Remote Control

The platform used to evaluate the performance of the proposed architecture is shown in Fig. 7. It is based on a BlueROV (version 1), including a structure to hold a S100 RF underwater modem, manufactured by Wireless for Subsea (WFS), which connects with a second one installed in another vehicle or attached to a surface device,

**Fig. 6** Frame structure

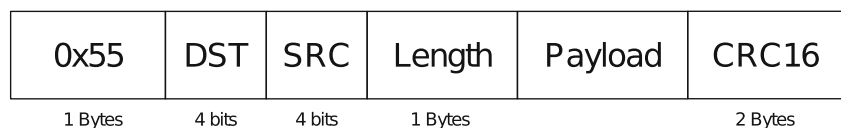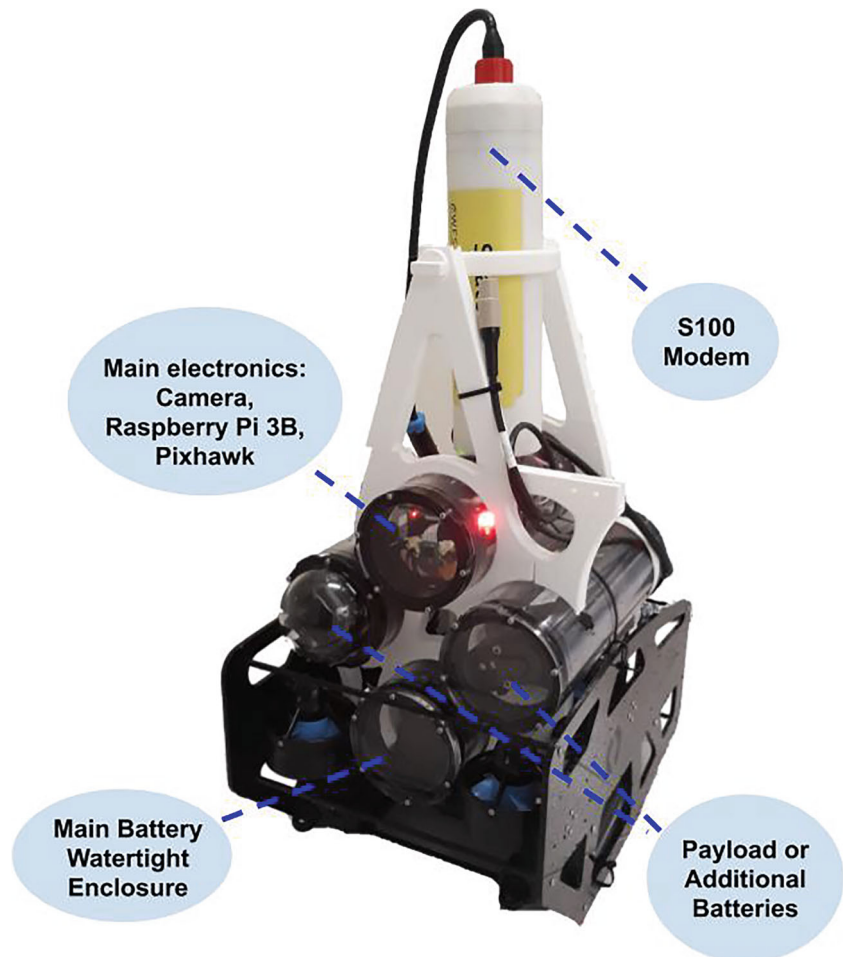| 0x55 | DST | SRC | Length | Payload | CRC16 |
|------|-----|-----|--------|---------|-------|
| 1 Bytes | 4 bits | 4 bits | 1 Bytes | | 2 Bytes |

**Fig. 7** Experimental platform
with the S100 modem integrated
on the BlueROV v1



according to the project requirements. UWSim has proven
to be an excellent tool for the simulation of part of the
system, in order to evaluate the performance of the proposed
architecture in a controlled environment, while allowing
other features, such as the implementation of VR human-
robot interfaces [18].

The control of the BlueROV is based on a Pixhawk
board [15] where the ArduSub software is executed.
The communication between operator and ROV uses an
umbilical wire. The BlueROV standard architecture is
based on the MAVLink protocol. MAVLink messages are
encapsulated into datagrams that are sent through the
umbilical wire to the ROV. A software application called
MAVProxy extracts the MAVLink messages from the UDP
datagrams received by the ROV, and sends the MAVLink
messages to the Pixhawk via a serial interface. Typical
MAVLink messages are the charge level of the battery,
geodesic position of the ROV, information related with
the Inertial Measurement Unit (IMU), the compass, depth
sensor or the Global Positioning System (GPS). There is
a robot operating system (ROS) package named MAVROS
that permits the publication of MAVLink messages as ROS
topics and services. The MAVROS node also implements

MAVProxy functionalities so that permits the control of the
ROV. MAVLink messages are read from the ArduSub and
sent to the WSF S100 RF underwater modems via serial
interface for their transmission.

BlueROV has been upgraded in order to integrate a
S100 RF modem. It includes not not only the mechanical
design, but also the addition of four waterproof cylinders
that provide enough space for the additional electronics.
On the one side, the S100 modem was powered using the
BlueROV batteries via a voltage regulator. An additional
Raspberry Pi has been attached in order to control the S100
modem via the serial interface. It communicates directly via
an Ethernet connection with the main Raspberry Pi on board
the BlueROV.

## 4.2 BlueROV Positioning Control

The BlueROV positioning control is based on the *Guided*
navigation mode of the ArduSub software. It is activated
only when an external location system provides the actual
position of the ROV. The Pixhawk firmware requires the
input of the ROV location in geodesic coordinates. It
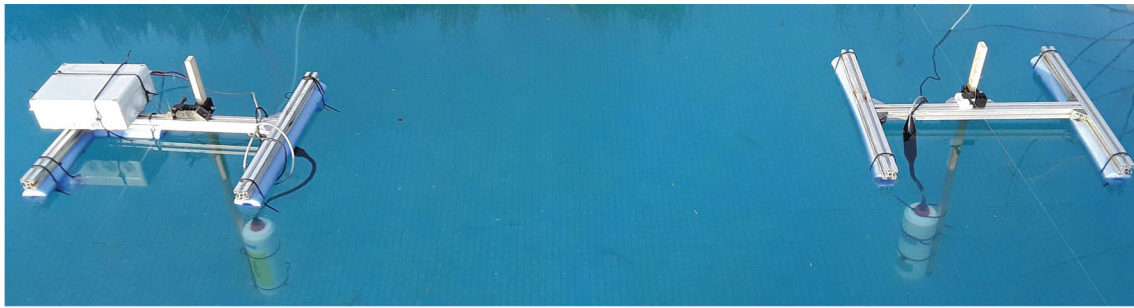executes an Extended Kalman Filter (EKF) that combines

**Fig. 8** Underwater RF WFS S100 modems attached to a surface structure for HIL Experiments

the information of the external location system with the rest of the sensors available in the Pixhawk, as the compass or the accelerometers, in order to obtain an estimation of the ROV position.

S2CR acoustic modems, considered in [4], implement a Ultra–Short Base Line (USBL) positioning system that can be used as the external positioning required by the Pixhawk. USBL positioning coordinates are transformed to geodesic coordinates and sent to the Pixhawk as a GPS_INPUT type MavLink message. USBL is a valid solution in open areas with little obstacles, however acoustic communications are strongly affected by multi-path. Thus it is not a viable solution for the water tank of the UJI laboratory shown in Fig. 2a where the experiments were done.

The external positioning, required by the Pixhawk in order to enable the positioning control, has been obtained by means of ArUco markers [10, 23] placed on the wall of the water tank of the UJI IRS Lab. The markers map shown in Fig. 2a is built at runtime by the robot using the aruco_mapping library.[1] The robot uses this marker map to localize itself.

### 4.3 UWSim-NET

Experimentation in underwater scenarios require to handle multiple hardware elements, as ROVs, modems, sensors, and all they have to be properly assembled. Moreover, real scenarios at sea are not appropriate for test experiments because of the complexity of dealing with all the equipment required for the experiment. Space restrictions at the laboratory usually impede the realisation of test experiments to evaluate the whole system. Hardware-In-the-Loop (HIL) experiments, in which part of the hardware is evaluated and the rest is modelled by software, represent a key element in robotics research.

UWSim-NET [5] is a module aimed at modeling the behaviour of underwater wireless communications. It has

been developed to conduct HIL experiments when the modems are not available. Network simulation in UWSim-NET is based on the real time scheduler and the functionalities of the NS3 (Network Simulator 3) library. UWSim-NET allows to simulate the behaviour of the S2CR acoustic modems and the S100 RF ones. It is based on a statistical model of each modem that has been constructed based on experimental measurements. Transmission times, errors in packet transmission are simulated while an UWSim scenario is executed. Transmission errors can be a consequence of signal attenuation, packet collision or packet loss when transmission or reception buffers are filled up.

The experiments to calculate the statistical model for the S100 RF modem entail measuring several transmission parameters while two S100 modems were submerged in the water tank of the UJI IRS Lab, (see Fig. 8). The experiments include the transmission of 2000 packets while measuring the times when each packet was sent and received. These measurements were used to calculate the intrinsic delay, the jitter and the bit rate that are shown in Table 3. After the calculation of the parameters of the statistical model, the UWSim-NET can be used to simulate the behaviour of the RF modems. The water tank where these experiments were performed was not big enough so as to measure any lost packets due to signal attenuation. However, UWSim-NET allows to adjust the Bit Error Rate (BER) depending on

---

[1] https://github.com/dcentelles/aruco_mapping. This depends on ArUco library to compute the position of the camera respect to a marker

**Table 3** Results obtained in experiments for modeling RF and acoustic modems

| | RF | Acoustic | |
| --- | --- | --- | --- |
| | | Burst | ACK |
| Bit rate (kbps) | 1.9 | 1.62 | 0.1 |
| Intrinsic delay (ms) | 85 | 443 | |
| Intrinsic Jitter (ms) | 13.20 | 412.99 | 443.63 |
| Propagation time inc. (ms/m) | 3.3e-9 | 0.67 | |
| Max. distance (m) | 5 | 3500 | |
| Min. distance (m) | 0.5 | 0 | |

**Fig. 9** Experiments setup. **a** UWSim during a HIL teleoperation using the ArduSub SITL simulator. **b** All system modules including ArduSub SITL Simulator
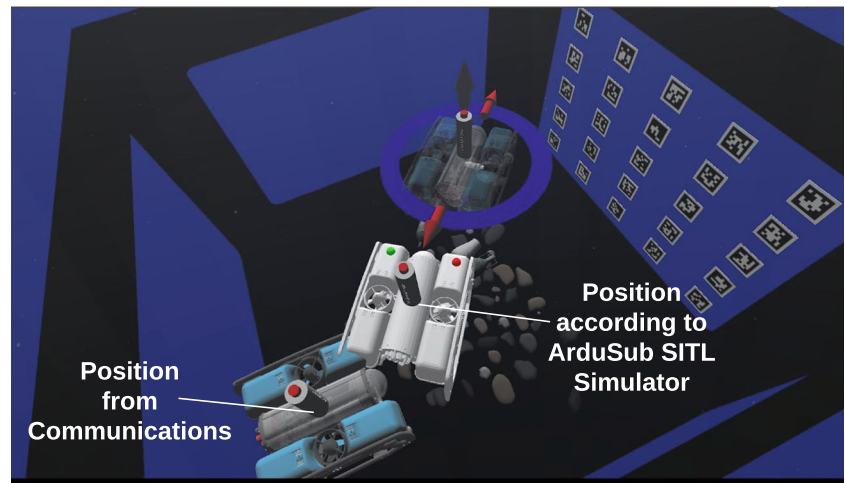


(a)
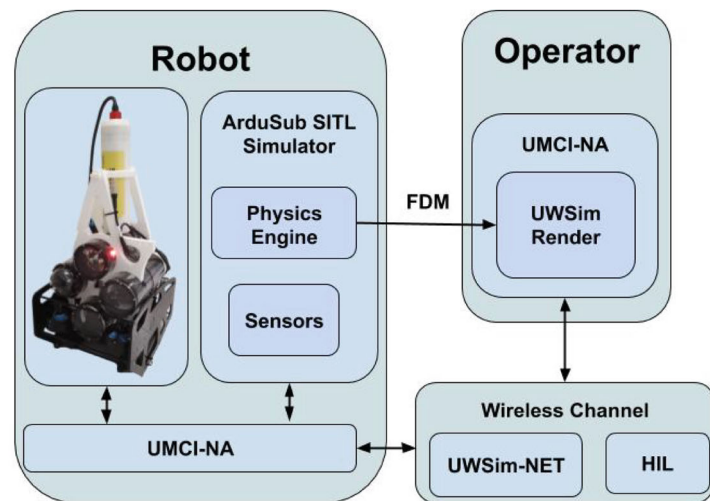


(b)

the separation of both modems, and this parameter will be adjusted in future experi0ments.

The statistical model for the S2CR acoustical modems was extracted from previous results presented in [4]. The measurements consisted on a HIL teleoperation experiment with a *Girona 500* HROV [20]. Both modems were submerged in a water tank that causes multi-path and degrade the results shown in the acoustic column of Table 3. Results of the burst mode are exposed in column *Burst* of Table 3 and the results of the mode based on attaching the messages in the D-MAC acknowledgements are showed in column *ACK*. The performance of the acoustic modems is expected to improve in the sea, where the multi–path effect has little effect on the communication. The high intrinsic delay and jitter values in Table 3 are partly due to the burst mode used with S2CR. As mentioned in Section 3.4 the burst mode was used because it allows higher bitrates than the instant messages mode of the D-MAC.

## 5 Results

The experiments presented in this paper follow a setup where the operator can teleoperate both a real and a virtual robot using the same UMCI-NA architecture. Teleoperations with the real robot have been performed using velocity control commands in a water tank. In order to focus on the network protocol design, HIL experiments are presented, using real RF modems as communication media. For this HIL experiments the robot side simulation is performed using the ArduSub SITL (*Software In The Loop*) simulator.[2] First, in Section 5.1 it is described how the different components involved in the experiments are interconnected.

---

[2]http://ardupilot.org/dev/docs/sitl-simulator-software-in-the-loop.html. The fork of the SITL simulator used in these experiments is: https://github.com/dcentelles/ardupilot

Then in Sections 5.2 and 5.3 the remote control results over a RF channel are presented.

## 5.1 HIL Experiments Software Architecture

Figure 9b shows the components of the performed experiments. The block identified as *Robot* represents the possibility of using a real BlueROV or a simulated one. The block identified as *Wireless Channel* represents the communication channel used, which will be based on either the S100 modems (HIL) or a simulation of them using UWSim-NET, depending on the experiment. The actual position of the ROV is displayed in the UWSim as a white ROV (see Fig. 9a). This is possible because the ArduSub simulator sends FDM (Flight Dynamics Model) messages through an UDP port in order to visualize the robot in an external application, such as FlightGear. In this work the UWSim has been extended to be able to place any robot in the position and orientation indicated in these FDM messages. Finally, the position received through the wireless channel is represented as a solid colorful ROV. Due to the delay produced by the wireless communications, this position is received with a lag respect to the actual position, represented by the white ROV.

## 5.2 Teleoperation with RF modems

In this section the results of a teleoperation over a RF link by velocity commands are presented. This experiment has been performed first using a BlueROV equipped with the S100 (see Fig. 2a), then a HIL experiment with the S100 modems (see Fig. 8), and finally modeling the RF modems considering a non-zero bit error ratio (BER).

In Fig. 10a and b, a time lapse of the packet sending and receiving events during the teleoperation are shown. Figure 10a shows the results of the HIL (Hardware In The Loop) experiment with the S100 modems sunk in water and robot motion simulated with the ArduPilot simulator. In this figure, the capture and reception events of an image are also shown. The same teleoperation was replicated in Fig. 10b, but modeling the communications using UWSim-NET in order to account for transmission errors. Here a BER of 0.07% has been established.

During the experiment, the control of the ROV was carried out by means of velocity commands (using the joystick). Figure 11a and d show the status of the speed controls on the operator and the robot sides during real teleoperation. Figure 11b and e show the status of the speed controls on the operator and the robot sides during HIL teleoperation. Figure 11c and f show the same information
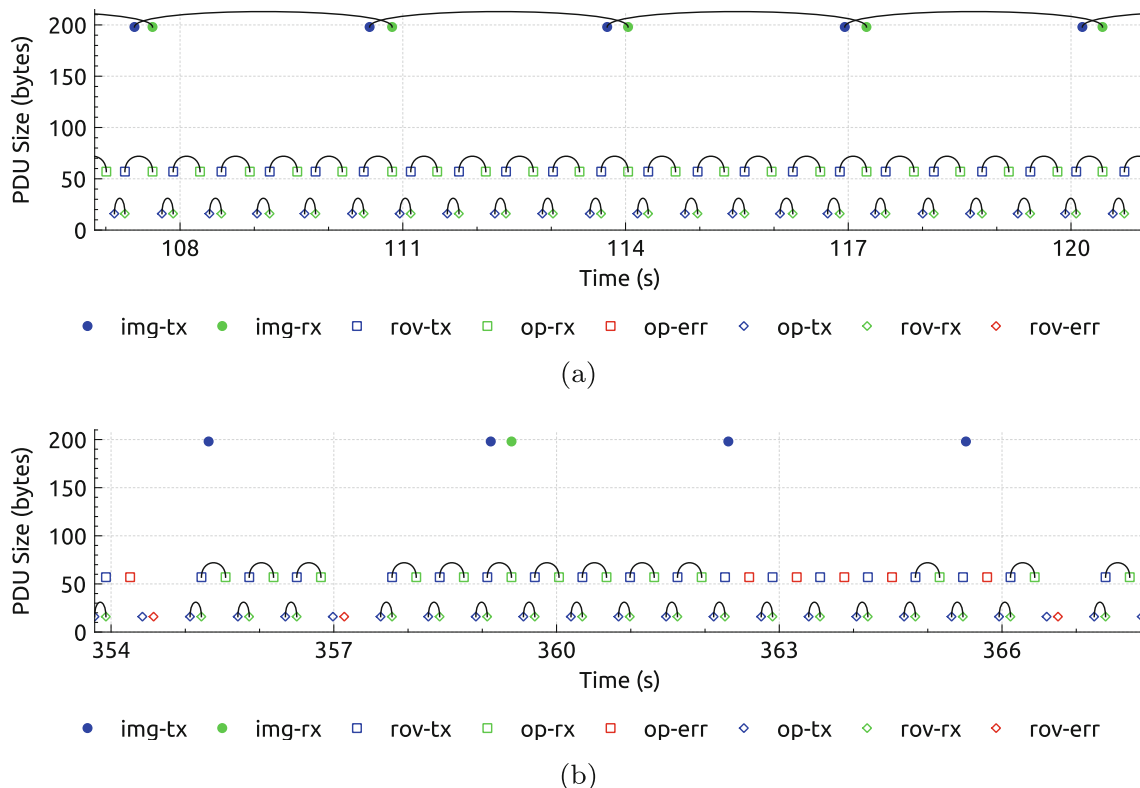


**Fig. 10** Time lapse of the S100 communication. Operator to ROV packets have a PDU size of 16 bytes, ROV to operator correspond to 57 bytes packets, and larger PDUs correspond to full image transmission. **a** Time measurements with the S100 modems. **b** Simmulated S100 considering a bit error rate (BER) of 0.07 %

**Fig. 11** Difference between
commanded velocities by the
operator (blue) and their
execution by the ROV (green). **a**
and **d** are the x and y axis during
a real experiment. **b** and **e** are
the x and y axis during the HIL
using simulated robot and real
modems. **c** and **f** are the x and y
axis with simulated robot and
simulating the communications
considering a fixed BER of
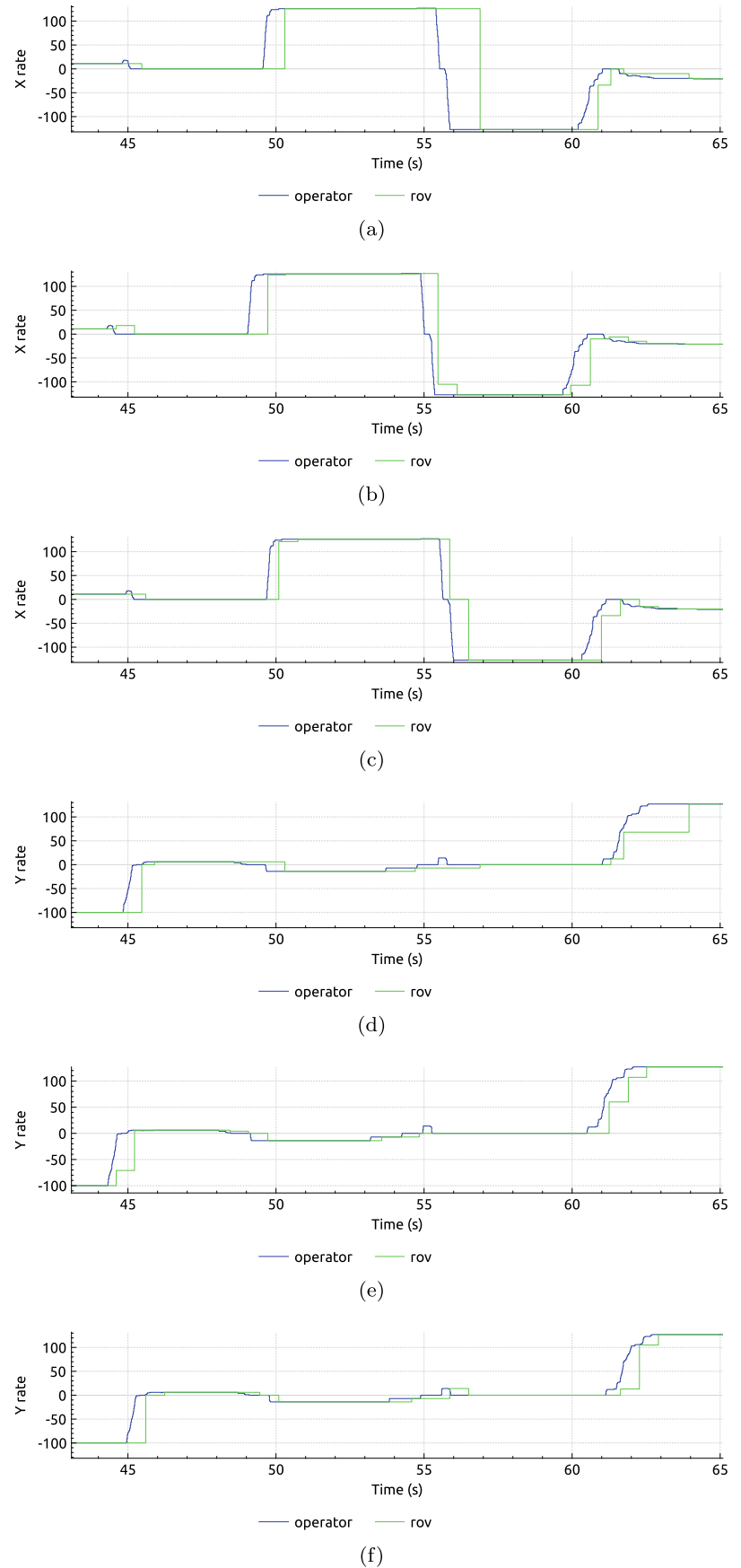0.07 %, and the parameters
shown in the RF column of
Table 3

**Table 4** Performance of the RF communication measured in a HIL experiment with the S100 modems, and modeling the S100 modems with UWSim-NET without considering packet loss (BER 0 %) and considering packet loss (BER 0.07 %)

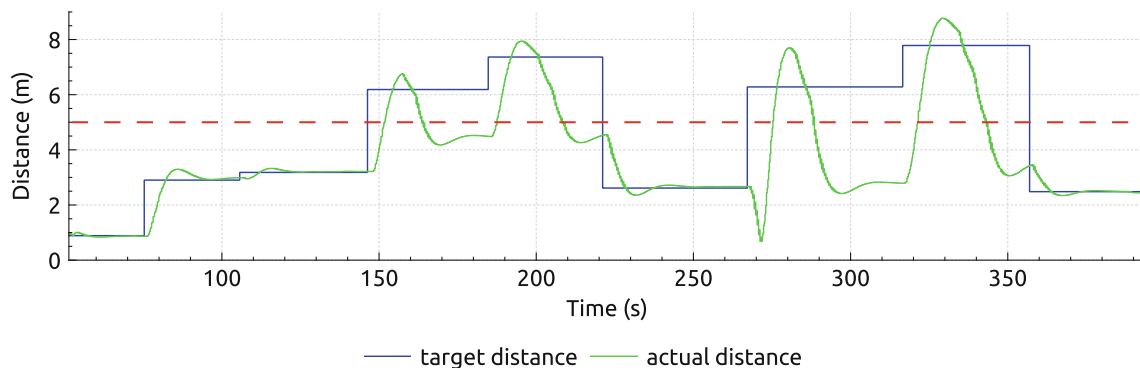| | HIL | | UWSim-NET (BER 0 %) | | UWSim-NET (BER 0.07 %) | |
| --- | --- | --- | --- | --- | --- | --- |
| | Robot to Operator | Operator to Robot | Robot to Operator | Operator to Robot | Robot to Operator | Operator to Robot |
| Through. (kbps) | 0.72 | 0.2 | 0.72 | 0.2 | 0.41 | 0.18 |
| Delay (ms) | 368.4 | 154.12 | 333.17 | 158.99 | 333.1 | 158.8 |
| Jitter (ms) | 13.42 | 10.55 | 12.84 | 10.05 | 10.42 | 9.05 |
| IAT (ms) | 640.38 | 640.53 | 640.21 | 640.36 | 726.56 | 640.42 |
| IAT $\sigma$ (ms) | 15.36 | 8.89 | 16.43 | 7.26 | 157.82 | 6.82 |



**Fig. 12** Current and actual distance from modem on buoy to modem on BlueROV and the target position's distance of the modem on the ROV. Modem's range at 5 meters is indicated as a straight line
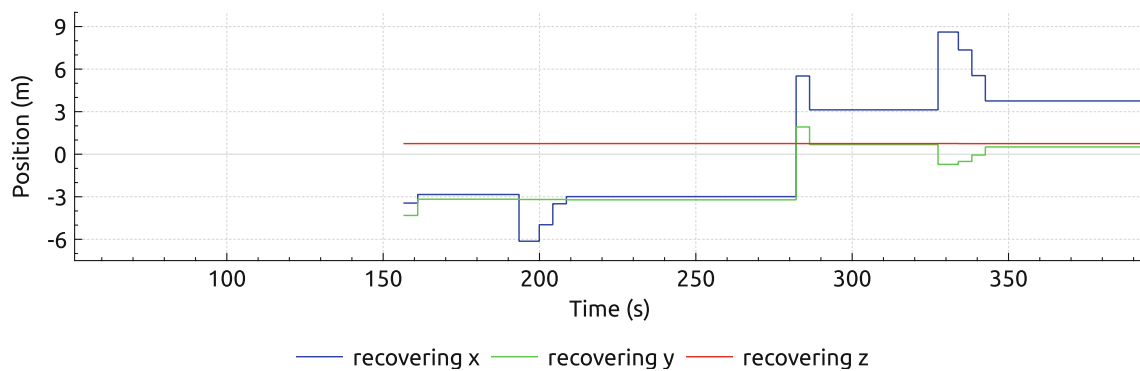


**Fig. 13** Recovering positions after the first communication lost



**Fig. 14** Target and actual NED positions

but modeling the communications considering transmission errors. As can be seen, there is only a small signal lag, caused by the delay. Moreover, the robot's packet reception rate is high enough (see Inter Arrival Time (IAT) in Table 4) so as to visualize the virtual robot in its real position in the HRI. The larger signal lag between commanded (blue) and execution (green) times in Fig. 11a and d is caused by the interference of the thrusters that lead to a slight increase of packet loss. However, it does not hamper teleoperation in real time.

Table 4 shows the numerical results of the experiment with real S100 modems. In order to validate the operation of UWSim-NET, the same teleoperation has been replicated considering a BER of 0 % (see Table 4), which is the one obtained during the HIL. Finally, Table 4 also shows the results of the experiment with UWSim-NET having set a BER of 0.07%. As can be apreciated, the resulting jitter during the teleoperation is comparable to the intrinsic jitter of the S100 modem, shown in Table 3. This is due to the TDMA strategy explained at Section 3.4.1 that ensures the media to be free before each message transmission, so that avoiding collisions without adding extra overhead to perform the media access control.

### 5.3 Position Commands and Communication Recovery

In this section the control of the BlueROV based on position commands is evaluated. The desired position was commanded by moving the semitransparent BlueROV in the UWSim scene. In this experiment, the communication was simulated using UWSim-NET, the maximum distance was set to 5 meters, and a constant BER of 0.07 % was considered. With this configuration the robot will lose communication completely when the distance between the on-board and the surface modem is greater than 5 meters.

Figure 12 shows the current and target value of the distance between the operator's modem, located near the surface, and the modem integrated in the ROV. The target value is the distance that will exist between the modems once the robot arrives at the destination position. The maximum distance is shown as a dashed red line at the value of 5 meters. The Fig. 13 shows the value of the last position to be recovered after the first loss of communication when passing the 5 meters. Finally, in Fig. 14 is shown the value of the current and target position during the control.

The results in Figs. 12 to 14 confirm that the proposed RCP permits to control a BlueROV in position mode. On the one hand, position commands are successfully received by the ROV which executes a PID that leads the ROV to the desired position. On the other hand, when the communication is lost, the ROV is capable of making its way back until the communication is recovered, as can be appreciated at second 160 in Fig. 12.

## 6 Conclusions

A protocol specific for underwater RF communications implemented within the framework of the MERBOTS project has been presented in this work. Taking into account the limitations of the channels in underwater scenarios with bandwidths lower than 1.9 kbps, an ad-hoc cross layer network protocol outside the TCP/IP stack is required.

The protocol described in this manuscript proposes to control the medium access that does not add extra overhead to the routing layer by using a dedicated TDMA algorithm. The master-slave strategy simplifies the protocol, and minimizes its overload by avoiding the requirement of request to send or clear to send signals in order to manage the medium access. Thus being able to make the most of the small bandwidth. The fact that the slave ROV only responds to the messages of the master might be appreciated as a limitation. But, it has been mitigated by forcing the master to send messages periodically, which ensures that the operator receives the odometry of the ROV at least once per second. The results achieved with such a protocol demonstrate the feasibility of wireless teleoperation of an underwater ROV. The proposed protocol is capable of providing the operator with visual feedback of the ROV. But the low image rate recommends the usage of either a VR module or a supervised control based on high level commands. The presented system allows both teleoperation in real time and a supervised control of the robot visualizing the result of the commands before being sent. In addition, it allows visual feedback with a region of interest at a constant rate thanks to the integration of the progressive compression algorithm DEBT.

Further work will be devoted to improve the visual feedback. The incorporation of a tracking system will facilitate the update of the ROI as the ROV moves. And a semantic analysis of the scene aimed at the recognition of shape, size and orientation of objects will reduce the amount of data required to transmit the visual information.

# References

1. Ayela, G., Coudeville, J.M.: Tiva: A long range, high baud rate image/data acoustic transmission system for underwater applications. In: Underwater defence technol conf. Paris, France (1991)

2. Bales, J.W., Chryssostomidis, C.: High-bandwidth, low-power, short-range optical communication underwater. In: 9th international symposium on unmanned untethered submersible technology, Durham, New Hampshire (1995)

3. Caiti, A., Grythe, K., Hovem, J.M., Jesus, S.M., Lie, A., Munafò, A., Reinen, T.A., Silva, A., Zabel, F.: Linking acoustic communications and network performance: Integration and experimentation of an underwater acoustic network. IEEE J. Ocean. Eng. **38**(4), 758–771 (2013). https://doi.org/10.1109/JOE.2013.2279472

4. Centelles, D., Moscoso, E., Vallicrosa, G., Palomeras, N., Sales, J., Marti, J.V., Marin, R., Ridao, P., Sanz, P.J.: Wireless HROV control with compressed visual feedback over an acoustic link. In: OCEANS 2017 - Aberdeen, IEEE, pp. 1–7. https://doi.org/10.1109/OCEANSE.2017.8084979. http://ieeexplore.ieee.org/document/8084979/ (2017)

5. Centelles, D., Soriano-Asensi, A., Martí, J.V., Marín, R., Sanz, P.J.: Underwater wireless communications for cooperative robotics with uwsim-net. Appl. Sci. **9**(17), 3526 (2019). https://doi.org/10.3390/app9173526. https://www.mdpi.com/2076-3417/9/17/3526

6. Che, X., Wells, I., Dickers, G., Kear, P., Gong, X.: Re-evaluation of rf electromagnetic communication in underwater sensor networks. IEEE Commun. Mag. **48**(12), 143–151 (2010)

7. Cossu, G., Corsini, R., Khalid, A., Balestrino, S., Coppelli, A., Caiti, A., Ciaramella, E.: Experimental demonstration of high speed underwater visible light communications. In: 2013 2nd International Workshop on Optical Wireless Communications (IWOW), pp. 11–15 (2013). https://doi.org/10.1109/IWOW.2013.6777767

8. Dea, J., Radosevic, D., Tran, N., Chavez, J., Neuner, I.I.I.B.: Land and undersea field testing of very low frequency rf antennas and loop transceivers. Tech. rep., SSC Pacific San Diego United States (2017)

9. Farr, N., Bowen, A., Ware, J., Pontbriand, C., Tivey, M.: An integrated, underwater optical /acoustic communications system. In: OCEANS 2010 IEEE - Sydney, pp 1–6. https://doi.org/10.1109/OCEANSSYD.2010.5603510 (2010)

10. Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F., Marín-Jiménez, M.: Automatic generation and detection of highly reliable fiducial markers under occlusion. Pattern Recogn. **47**(6), 2280–2292 (2014). https://doi.org/10.1016/j.patcog.2014.01.005. http://www.sciencedirect.com/science/article/pii/S0031320314000235

11. IRSLab: MERBOTS Research Project (http://www.irs.uji.es/merbots/) (2015)

12. IRSLab: TWINBOT Research Project (http://www.irs.uji.es/twinbot/) (2018)

13. Kaushal, H., Kaddoum, G.: Underwater optical wireless communication. IEEE Access **4**, 1518–1547 (2016)

14. Kebkal, O., Komar, M., Kebkal, K., Bannasch, R.: D-mac: Media access control architecture for underwater acoustic sensor networks. In: OCEANS 2011 IEEE - Spain, pp. 1–8. https://doi.org/10.1109/Oceans-Spain.2011.6003586 (2011)

15. Meier, L., Tanskanen, P., Heng, L., Lee, G.H., Fraundorfer, F., Pollefeys, M.: Pixhawk: A micro aerial vehicle design for autonomous flight using onboard computer vision. Auton. Robot. **33**(1-2), 21–39 (2012)

16. Merriam, S., Porta, D.: Dsp-based acoustic telemetry modems. In: Sea Technol (1993)

17. Pelekanakis, C., Stojanovic, M., Freitag, L.: High rate acoustic link for underwater video transmission. In: OCEANS 2003. Proceedings, vol. 2, pp. 1091–1097. https://doi.org/10.1109/OCEANS.2003.178494 (2003)

18. Prats, M., Pérez, J., Fernández, J., Sanz, P.: An open source tool for simulation and supervision of underwater intervention missions. In: 2012 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp. 2577–2582. https://doi.org/10.1109/IROS.2012.6385788 (2012)

19. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: Ros: An open-source robot operating system. In: ICRA workshop on open source software, Kobe, Japan, pp. 5 (2009)

20. Ribas, D., Palomeras, N., Ridao, P., Carreras, M., Mallios, A.: Girona 500 AUV: From survey to intervention. IEEE/ASME Trans. Mechatron. **17**(1), 46–53 (2012). https://doi.org/10.1109/TMECH.2011.2174065

21. Ribas, D., Ridao, P., Turetta, A., Melchiorri, C., Palli, G., Fernandez, J.J., Sanz, P.J.: I-AUV Mechatronics Integration for the TRIDENT FP7 Project. IEEE/ASME Trans. Mechatron. **20**(5), 2583–2592 (2015). https://doi.org/10.1109/TMECH.2015.2395413. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7047908

22. Ribas, J., Sura, D., Stojanovic, M.: Underwater wireless video transmission for supervisory control and inspection using acoustic OFDM. In: OCEANS, vol. 2010, pp. 1–9 (2010). https://doi.org/10.1109/OCEANS.2010.5663839

23. Romero-Ramirez, F.J., Muñoz-Salinas, R., Medina-Carnicer, R.: Speeded up detection of squared fiducial markers. Image Vis. Comput. **76**, 38–47 (2018). https://doi.org/10.1016/j.imavis.2018.05.004. http://www.sciencedirect.com/science/article/pii/S0262885618300799

24. Rubino, E.M., Centelles, D., Sales, J., Marti, J.V., Marin, R., Sanz, P.J., Alvares, A.J.: Progressive image compression and transmission with region of interest in underwater robotics. In: OCEANS 2017 - Aberdeen, IEEE, pp 1–9. https://doi.org/10.1109/OCEANSE.2017.8084999. http://ieeexplore.ieee.org/document/8084999/ (2017)

25. Rubino, E.M., Centelles, D., Sales, J., Martí, J.V., Marín, R., Sanz, P.J., Álvares A.J.: Underwater radio frequency image sensor using progressive image compression and region of interest. Journal of the Brazilian Society of Mechanical Sciences and Engineering. https://doi.org/10.1007/s40430-017-0894-6 (2017)

26. Schneider, T., Schmidt, H.: Unified command and control for heterogeneous marine sensing networks. J. Field Robot. **27**(6), 876–889 (2010). https://doi.org/10.1002/rob.20346.

27. Shaw, A., Al-Shamma'a, A., Wylie, S., Toal, D.: Experimental investigations of electromagnetic wave propagation in seawater. In: Microwave Conference, 2006. 36th European, pp. 572–575. https://doi.org/10.1109/EUMC.2006.281456 (2006)

28. Suzuki, M., Sasaki, T., Tsuchiya, T.: Digital acoustic image transmission system for deep-sea research submersible. In: OCEANS '92. Mastering the oceans through technology. Proceedings, vol. 2, pp. 567–570 (1992). https://doi.org/10.1109/OCEANS.1992.607839

29. Tuci, E., Alkilabi, M.H.M., Akanyeti, O.: Cooperative object transport in multi-robot systems: A review of the state-of-the-art. Front. Robot. AI **5**, 59 (2018). https://doi.org/10.3389/frobt.2018.00059. https://www.frontiersin.org/article/10.3389/frobt.2018.00059/full

30. Wirz, R., Marin, R., Ferre, M., Barrio, J., Claver, J.M., Ortego, J.: Bidirectional transport protocol for teleoperated robots. IEEE Trans. Ind. Electron. **56**(9), 3772–3781 (2009). https://doi.org/10.1109/TIE.2009.2025291

**Diego Centelles** was born in 1991 in Castellón and holds a B.S. in Computer Science (2013) and a master in "Intelligence Systems" (2018) from the Universitat Jaume I, Castellón (Spain). Nowadays, he works as a PhD student in the Interactive & Robotic Systems Lab. He has participated in several research projects such as TRITON (http://www.irs.uji.es/triton/) and MERBOTS (http://www.irs.uji.es/merbots/).

**Antonio Soriano** was born in Benaguasil (Valencia), Spain, in 1978. He received the Licenciado degree in physics in 2001 and the PhD in 2007 from the University of Valencia, Valencia. He has worked as a post doctoral research at the Spanish National Researh Council (CSIC), the Universitat Politecnica de Valencia, and the Universitat Jaume I. Since 2017 is Associate Professor at the Department of Informatics of the University of Valencia. His research interest is focused on computer networks and robotics.

**Raul Marin** received a B.Sc. degree in computer science engineering (1996) and a Ph.D. in Engineering (2002) by the Jaume I University of Castellon (Spain). The subject of his Ph.D. was the development of a vision-based telerobotic system via web, by using object and speech recognition, 3D virtual environments, grasping determination, and augmented reality. In 1996 he worked in the Multimedia department of BYG Systems Ltd, Nottingham (UK), as a software developer. In 1997, he joined Lucent Technologies (Bell Labs Innovations) and worked as researcher, software developer and software architect at the Switching and Access Division. In 1999 he began to teach and research as professor at the Jaume I University. Now, he is associate professor at the Department of Computer Science of the Jaume-I University (Spain) and lectures Computer Networking, and Robotics in the same university. His research interests lie mainly in Field robotics, Rescue and Underwater, including subjects such as Localization, Networks of Sensors and Actuators, Object Recognition, Internet Telerobotics, and Education. He is author or co-author of multiple research publications on these subjects. He has participated in several research projects such as FP6 GUARDIANS (Group of unmanned assistant robots deployed in aggregative navigation supported by scent detection), and now is involved in the FP7 TRIDENT project (Marine robots and dexterous manipulation for enabling autonomous underwater multipurpose intervention missions).

**Pedro J. Sanz** is Associate Professor in the Computer Science and Engineering Department at Jaume I University (Spain), and head of the Interaction and Robotic Systems Lab. He holds a B.Sc. in Physics by the University of Valencia (1985, Spain), M.Sc in Engineering (CAD/CAM) from the Technical University of Valencia (1989, Spain) and a Ph.D. in Computer Engineering by the Jaume I University of Castellon (1996, Spain). His current research interests are Multisensory based Grasping and Dexterous Manipulation, Telerobotics and Human-Robot Interaction, all of them applied to any kind of real service robotics scenarios, including assistive and underwater robotics. Dr. Sanz was a 1996 recipient of the "Best Thesis in Electronics and Computer Science Domains", National Research Prize, from the "Artigas Spanish Foundation" (Polytechnic University of Madrid, Spain). Dr. Sanz has been active since 1990 in R&D within several projects on Advanced Robotics and Multimedia Interfaces. He is author or co-author of a broad range of research publications and is an active member of different scientific societies such as IEEE Robotics and Automation, IEEE Systems, Man and Cybernetics, IAPR, AERFAI (Spanish Association of Pattern Recognition and Image Analysis), ECCAI and AEPIA (Spanish Association for Artificial Intelligence).

He has been appointed as Visiting Scientist at the Institute for Real-Time Computer Systems (Technische Universität München, GERMANY), periods in 2000 and 2001, at LASMEA ("Le LAboratoire des Sciences et Matériaux pour l'Electronique et d'Automatique, Univ. Blaise Pascal, FRANCE), in 2002, and at LAR-DEIS (Laboratory of Automation and Robotics of D.E.I.S., Department of Electronics, Computer Science and Systems of the University of Bologna, ITALY) during eight months (Sabbatical Period in 2008). He was a member of the organizing committee of the 11th International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems (IEA-AIE'98). Chair of several Tutorials and Workshops within outstanding International Conferences on Robotics. Co-chair of the Interest Group on "Manipulation and grasping" within the European Robotics Research Network (EURON). Finally, he has served as program committee member of several international conferences and as editor of some well-known international journals (IEEE Transactions on Systems, Man and Cybernetics (C), since 2005 and IEEE Robotics and Automation Magazine, since 2008).