J Intell Robot Syst (2018) 92:705–718
https://doi.org/10.1007/s10846-017-0700-9

Development and Experimental Tests of a ROS Multi-agent Structure for Autonomous Surface Vehicles

Giuseppe Conte[1] · David Scaradozzi[1] · Daniele Mannocchi[1] · Paolo Raspa[1] ·
Luca Panebianco[1] · Laura Screpanti[1]

Received: 2 February 2017 / Accepted: 29 August 2017 / Published online: 5 October 2017

Abstract Robotic structures that couple autonomous surface vehicles and unmanned underwater vehicles in integrated systems with various levels of cooperation provide interesting solutions to the problem of developing efficient, versatile and cost effective tools for exploration, monitoring and exploitation of the underwater environment. In this paper we describe the development and preliminary field testing of an autonomous surface vehichle that can automatically track, deploy and recover a small remotely operated vehicle, which is guided from a shore-ground station. This goal is achieved by exploiting two-ways transmission of data and commands through the umbilical and a wireless link with a shore-ground station. In this way, pilots can experiment telepresence in the underwater environment, avoiding the need of expensive and logistically demanding manned supply vessel. The vehicle is a small aluminum hull boat, equipped with a steering outboard electric motor. A multi-agent system in the ROS framework is proposed for the robotic structure. The use of commercial-off-the-shelf components and the choice of a multi-agent ROS architecture are a mean to reduce costs and to assure performances, modularity and versatility. Field tests in both supervised and autonomous guidance mode have been performed in order to assess the basic functionalities of the system and their results are illustrated and discussed.

Keywords Marine robotics · Autonomous surface Vehicles · ROS · Remote control

✉ Giuseppe Conte
gconte@univpm.it

[1] Dipartimento di Ingegneria dell'Informazione, Università Politecnica delle Marche, Ancona, Italy

1 Introduction

Monitoring and survey of environmental parameters, of marine structures and of underwater operations in coastal areas and shallow waters can greatly benefit from the use of robotic platforms that couple different robotic vehicles in integrated autonomous or semi-autonomous systems, addressing performances, as well as versatility, easiness of use and costs. In order to satisfy these requirements, in several cases robotic architectures may exploit cooperation and integration between Autonomous Surface Vehicles (ASV) and Unmanned Underwater Vehicles (UUV) of various kinds. In seabed surveys and data acquisition, for instance, surface and underwater vehicles may cooperate simply by exchanging information and commands that, facilitating localization and navigation, guarantee coverage and accuracy ([10, 15] and [8]). In other more specific applications, as described in [6], autonomous surface vehicles are employed to carry and to deploy underwater vehicles in designated areas to perform specific exploration or intervention tasks.

This paper presents the development of a small, low-cost, prototypal ASV, which is a part of a robotic platform for scientific use, whose structure and components are depicted in Fig. 1.

The platform, which includes the aforementioned ASV, a micro-ROV and a shore-ground station, is conceived for exploration, monitoring and light intervention in the underwater environment. The platform can operate in a partially supervised mode: the micro-ROV is remotely guided, through a radio link and the umbilical, from a shore-ground station, while the ASV, which is designed to deploy/recover the micro-ROV, performs autonomously in order to guarantee the functionality of the integrated structure. The main advantages of such architecture is that of making it

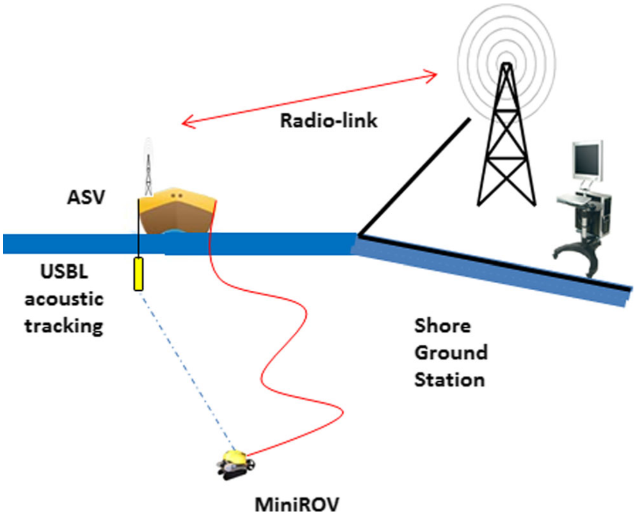


Fig. 1 The ASV/micro-ROV system

possible the deployment and the use of the micro-ROV without supporting the costs of a manned supply vessel. By exploiting two-ways transmission of data and commands through the radio link and the umbilical, the pilot of the micro-ROV can experiment telepresence in the underwater environment in an easy, economic and simple way to deal with and to manage.

Beside acting as a bridge to assure communication between the shore-ground station and the micro-ROV, the main tasks which the ASV must be able to perform are autonomous navigation, with the aid of navigation sensors (GPS, compass, Inertial Measurement Unit (IMU)), and automatic formation keeping, with the aid of an USBL positioning system, with the micro-ROV, while the latter is freely guided by a pilot. This task is important because, by remaining close to the vertical of the micro-ROV, the ASV not only limits the length of the umbilical, reducing its burden, but it also improves accuracy in acoustic positioning during survey or intervention. We can therefore think of the formation keeping behavior as a cooperative behavior that is actuated by the ASV in order to facilitate the micro-ROV tasks. Scaling down, the ASV behavior itself can be viewed as resulting from the cooperation of a number of independent subsystems that govern actuators and sensors to accomplish navigation tasks.

From this point of view, the main theoretical contribution of the paper is in focusing on a systematic approach to design and to develop a complex, modular robotic structure in which different components exchange information and cooperate. In order to better illustrate this point in relation to the present application, let us recall that in [8], integration and cooperation between an ASV and an AUV in sharing a communication channel is obtained by organizing the ASV control software in several threads, but no specific methodology to construct an architecture of that kind is discussed. A similar approach is used in [2] and [18], while in [6] the authors adopt a more structured point of view. In that paper, the overall control and decision support system that governs an ASV/ROV robotic platform for mine countermeasure is defined by means of a MOOS-IvP [1] architecture that implements a set of basic behavioral schemes. Here, in designing and developing the Navigation, Guidance and Control (NGC) system of the ASV, we follow the Multi-Agent System (MAS) approach illustrated in [11]. The MAS framework has the advantage of facilitating the description of complex structures, where data and information are processed in a decentralized and asynchronous way and control is distributed, as those composed by a number of autonomous, mutually interacting hardware/software units. This makes it possible to obtain a high level of modularity and scalability of the overall structure, together with the capability of self-organization and plug-and-play functionality of components (agents). Moreover, MAS architectures can be quite naturally implemented using the open-source Robot Operating System (ROS) ([16]). In this way one can exploit a powerful middleware to assure interaction between agents, thank to an intuitive publisher/subscriber mechanism, easiness of usage, possibility to employ different programming language and open communication protocols. Modularity and scalability of the underlying ROS structure structure guarantee, during operation, a seamless switch between autonomous control, manual control (by an on-board operator) and remote control.

From a more practical point of view, the main contribution of the paper is in developing a performing, versatile robotic platform for exploration, monitoring and light intervention in the underwater environment that has relevant innovative characteristics. First, the mechatronic structure of the ASV is constructed using mainly components-off-the-shelf (COTS), which assure low costs, re-usability and re-factoring of the prototype and simple maintanance. Secondly, the platform can be launched/recovered and operated by a single user from the shore. Beside reducing considerably risks, logistics and operational costs, this gives the possibility to supervise the mission from a shore-ground location, where additional facilities and resources that may be required during the mission can be made easily accessible to the operator. Moreover, the possibility to operate at different level of autonomy increases versatility and usability in different environments and contexts, from controlled ones in access-restricted and protected areas to others in which unexpected and unpredictable events occur frequently. These features make the platform a suitable tools for environmental monitoring and protection activities, inspection and maintenance of submerged structures in shallow water, surface and depth patrolling and surveillance of marine areas. In term of Technology Readiness Level (or

TRL), the results of the field experiments that are reported in the paper show that the development of the platform reached TRL 6, corresponding to system/subsystem demonstration in a relevant environment. No other system with similar features and capability seems to be presently available.

The paper is organized as follows: Section 2 details the structure of the developed NGC system that was preliminary described in [4]. Section 3 illustrates the internal logical structure of the components of the system and the hardware on which they are implemented. Functional and autonomous tracking tests are described in Section 4, while the results of the tests are discussed in Section 5. Finally, Section 6 contains conclusions and description of future work.

2 Stucture of the System

The mechanical, hardware and software components of the ASV have been constructed using commercially available, low cost COTS. Differently from other vessels for scientific purposes, which are double-hull catamarans with two fixed thrusters ([7, 8, 14] and [13]), the considered ASV is a mono-hull boat with a unique steering, outboard electric motor. The mono-hull construction guarantees robustness and it facilitates transportation and deployment, while the use of a steering outboard motor increases easiness of installation, operation and maintenance. The aluminum hull is a Marine 10M (length 3,05m; max width 1,40m; weight 37Kg) and the outboard electric motor is a Torqeedo Travel-503, equipped with a conventional nautical steering system, which is governed by a stepper motor endowed with an incremental shaft encoder. These components, or very similar ones, are available at reasonable price on the nautical market. A picture of the ASV is shown in Fig. 2.

The mechatronic structure of the ASV is composed by a set of subsystems, consisting of hardware and software components. Each subsystem, from an operational point of view, is divided in three levels, called, respectively, the Agent Level, the Interface Level and the Hardware Level, as described in Fig. 4. The Agent Level refers to software components, called Agents, which are organized according to a multi-agent system (MAS) architecture in the ROS framework, called the BOAT Agency. The ROS framework has the advantage of facilitating rapid prototyping, of employing existing open-source software modules for robotic applications and of easy interfacing with other ROS structures. In each subsystem, the ROS software agents take care of the high level tasks and of the communication with other entities in the ROS framework. This solution allows the subsystems to interact between them by exchanging data as ROS topics and to perform specific tasks in response to external inputs in a coordinated way. The Interface Level refers to the low level software routines that interface each agents with various I/O devices. The Hardware Level refers to actuators, sensors, computing devices and other electronics components. The various subsystems can be described as follows:

Central Control subsystem It consists of a Single Board Computer (SBC) that hosts the Master agent and other agents of the MAS. An IMU and a GPS device, which are used to evaluate position, orientation, velocity and acceleration of the ASV, and a video-camera, with motorized pan and tilt mount, are directly connected to the SBC.

Engine subsystem It is composed by the outboard electric motor Torqeedo Travel-503 and by a custom board to implement the ROS agent that governs it.

Rudder subsystem It is composed by a mechanical steering system, which is actuated by a stepper motor endowed with an incremental shaft encoder. The stepper motor is closed loop controlled by a microcontroller. Control references are provided by a dedicated agent, implemented on a ARM Linux Board.

Power subsystem It provides the electric energy for all devices by a gasoline-powered, portable electric generator and a battery. AC/DC and DC/DC converters are used to convert 230Vac to 12Vdc and 24Vdc as required by the on-board devices.

Remote Control subsystems Located at a shore-ground station (shown in Fig. 2), it consists of a laptop, a wireless radio amplifier and a Wi-Fi antenna for communication. The laptop is endowed with a graphical user interface

Fig. 2 Pictures of the ASV system during a field test (crew is present only to comply with safety regulation): launching; navigating; shore-ground station

(GUI), shown in Fig. 3, to manage and to monitor the vehicle and the other connected devices, like the micro-ROV and the USBL, and the virtual agents. Through the GUI, a pilot can send commands, like position or speed references to the NGC system, and he can monitor the status and operations of the MAS.

Communication subsystem It consists of wireless radio amplifier and a Wi-Fi antenna to communicate with the Remote Control subsystem and of an Ethernet communication infrastructure to connect all on-board devices.

The agents and the low level software routines are implemented on computing devices of various kind that form the ASV computing structure. The set of computing devices includes a Single Board Computer (SBC) and two ARM boards, all equipped with a Linux operative system, that host the ROS infrastructure, the software agents and the ROS communication routines. In addition, the ARM boards are coupled with a (couple of) micro-controller boards, which host the low-level routines for interfacing agents and I/O devices. In practice, the ARM boards are used to execute behavioral and decisional tasks at the Agent Level and at the Interface Level, while the micro-controller boards implement the control strategies that govern actuators and sensors at the Hardware Level. The advantages of this architecture are the high computational capabilities to run decisional and ROS protocols provided by the ARM boards and the reliability of control performances assured by the use of dedicated microcontrollers. The two boards communicate between them using a serial communication link and an error handling protocol. The agents of the BOAT Agency use the ROS infrastructure to interact between them, letting the ROS Master manage their registration and services notification.

The Multi Agent System that characterizes the ASV is a virtual infrastructure that includes the following elements (see Fig. 4):

Master agent This agent coordinates the MAS in the ROS framework; it is used to monitor the status of the infrastructure and to alert about MAS failures.

Controller agent This agent implements the NGC procedures of the ASV, as described in Section 3.

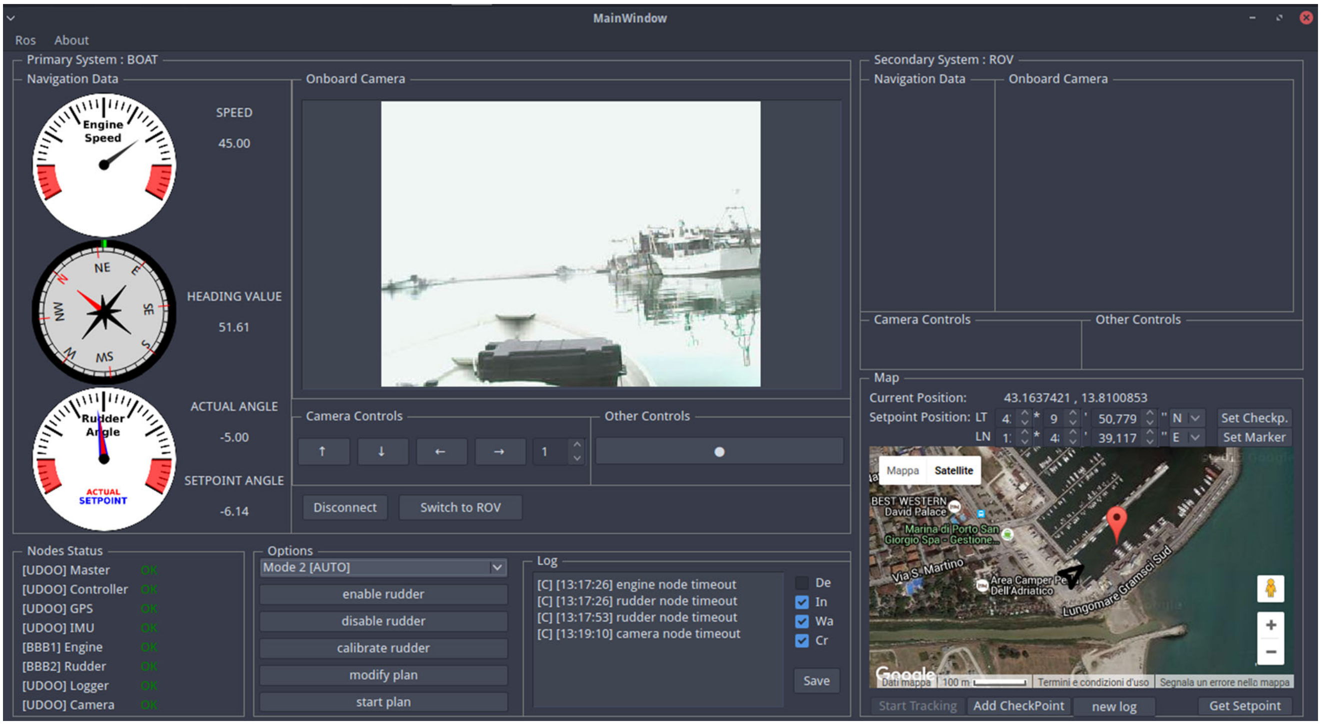
Engine agent This agent implements the open-loop control of the angular speed of the thruster of the outboard electric motor.

Rudder agent This agent implements the control of the steering angle of the outboard electric motor.

GPS agent This agent acquires and publishes, within the ROS infrastructure, the position of the vehicle obtained by the GPS.

IMU agent This agent acquires and publishes, within the ROS infrastructure, attitude, rotational speeds and accelerations of the vehicle obtained by the IMU.

Camera agent This agent streams video from an on-board surveillance IP-Camera and manages pan, tilt and focus according to external requests.



SPEED 45.00

HEADING VALUE 51.61

ACTUAL ANGLE -5.00

SETPOINT ANGLE -6.14

Current Position: 43.1637421 , 13.8100853

Setpoint Position: LT 4: * 9 ' 50,779 " N

LN 1: * 4: ' 39,117 " E

Nodes Status	
[UDOO] Master	OK
[UDOO] Controller	OK
[UDOO] GPS	OK
[UDOO] IMU	OK
[BBB1] Engine	OK
[BBB2] Rudder	OK
[UDOO] Logger	OK
[UDOO] Camera	OK

Options: Mode 2 [AUTO]

```
[C] [13:17:26] engine node timeout
[C] [13:17:26] rudder node timeout
[C] [13:17:53] rudder node timeout
[C] [13:19:10] camera node timeout
```

☐ De
☑ In
☑ Wa
☑ Cr

Fig. 3 GUI for remote control, supervision and monitoring

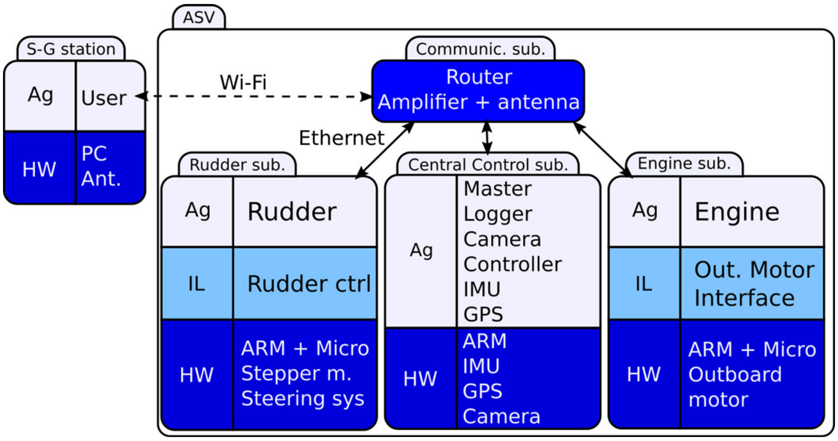


Fig. 4 MAS structure of the ASV (AL= Agent Level, IL= Interface Level, HL= Hardware Level)

Logger agent This agent logs the data published by the other agents to disk.

User agent This agent implements the GUI of the system for remote monitoring and control.

ROV agent This agent implements the control and sensor of the micro-ROV

USBL agent This agent implements the driver for the USBL positioning system

Agents like Engine, Rudder, GPS, IMU and Camera operates on I/O devices through the Interface Level, while agents like Master, Controller, Logger, User operates only at Agent Level. Each agent, in the ROS agency, has a unique name and a default topic, in the ROS terminology, that is used to let other agent know whether it is currently subscribing to the agency or not and, possibly, to signal, by suitable error codes, malfunctioning. Agents have a life-cycle, during which they go through a sequence of phases as shown in Fig. 5 and described below.

Activities which belong the agent's behavior are initialized in the starting phase and they are performed in the processing phase, until some specific event occurs and the agent terminates its activity in the ending phase. Events that may occur are:

- an explicit shutdown request to the agent;
- the shutdown of the Boat Agency;
- the (detection, perceived by absence of a default signal, of) failure of the ROS infrastructure.

The last event causes the termination of the agent's current activities, followed by an attempt to subscribe again to the agency. When the system is started, the microcontrollers activate and take control of the related devices, while software components at the Agent Level run and the ROS infrastructure is activated. Each agent needs to exchange data and information with others within the MAS in order to operate correctly. This implies a dependency relationship among agents in the MAS, which is implemented by specific routines. The MAS system handles unwanted disconnection of each single agent (detected through watchdog signals) at two levels. The disconnected agent tries to restart and to reconnect to the MAS. At that occurrence, the other agents react as indicated in the MAS dependency matrix in Table 1: namely, either they disconnect and try to re-connect to the MAS (R) or they go to a safe operation mode (S), which, in particular, excludes interaction with the disconnected agent. The content of cell *ij* describes the reaction of the agent specified (by an obvious abbreviation) in the first row of column *j* to disconnection of the agent specified in row *i* of the first column.

Agents can exhibit four types of behaviors to interact with the environment or between them within the agency:

OneShotBehaviour : one or more actions associated to the behavior are executed only one time and instantly;

CountdownBehavior : one or more actions associated to the behavior are executed only one time after a given delay;

CyclicBehaviour : one or more actions associated to the behavior are executed periodically with a given period;

EventDrivenBehaviour : one or more actions associated to the behavior are executed only if and when a specific event is triggered.

In relation with the EventDrivenBehavior, triggering events are of the following kind:

- notification of a message;
- notification of a service request;

Fig. 5 Agent's life-cycle

Table 1 MAS Dependency matrix

	MSR	CTR	ENG	RUD	GPS	IMU	CAM	LOG	USR
MSR	R	R	R	R	R	R	R	R	R
CTR		R	S	S					
ENG		S	R						
RUD		S		R					
GPS		S			R				
IMU		S				R			
CAM							R		
LOG								R	
USR		S							R

– cancelation of subscription by an agent on which the one at issue is dependent.

Events of the last kind are critical for the operation of the system and suitable strategies are actuated, in case they occur, in order to keep the system in safe conditions or to recover from unwanted situations. Behaviors have a life cycle, during which they go through a sequence of phases as shown in Fig. 6. Compared to the agent life-cycle, the processing phase is different for each kind of behavior and it is possible to disable the behavior and to put it in a waiting state.

3 System Operation

When the system is turned on, the ROS infrastructure and the ROS Master are activated and the Boat Agency becomes operative. Then, the Master Agent activates and other agents, if they are endorsed to, subscribe. In particular, the Control Agent takes control of the movements of the ASV. During operation, the Logger Agent takes care of recording a set of data that are related to agents' activity, including subscriptions and possible disconnections times, to life cycle and behavior and to the ASV motion, including commands, positions and velocity. Part of these data can be communicated in real time or with small delays over the radio link.

In the current implementation, the Controller can exhibit behaviors that correspond to three different operative conditions, as described below:

Remote Guidance mode In this operative condition, the Controller adopts a behavior that simply consists in transmitting directly to the Rudder and to the Engine the commands it receives as input from an operator through the User agent. This operative condition allows remote guidance through the radio link that equips the ASV. In a future phase of development of the robotic system, suitable guidance strategies to support the operator (for instance to guarantee that motion occurs within a designated region, or to limit speed according to sea conditions) will be implemented and the behavior(s) will be modified accordingly.

Basic Motion mode In this operative condition, the Controller implements open loop control strategies, guiding the ASV by a (sequence of) command(s), as specified by the operator.

Automatic Guidance mode In this operative condition, the Controller takes full control of ASV motion in order to implement one of a set of specific behaviors, as specified by the operator. The set of possible behaviors include, in the present implementation, homing, point-to-point motion and tracking of a target. In order to

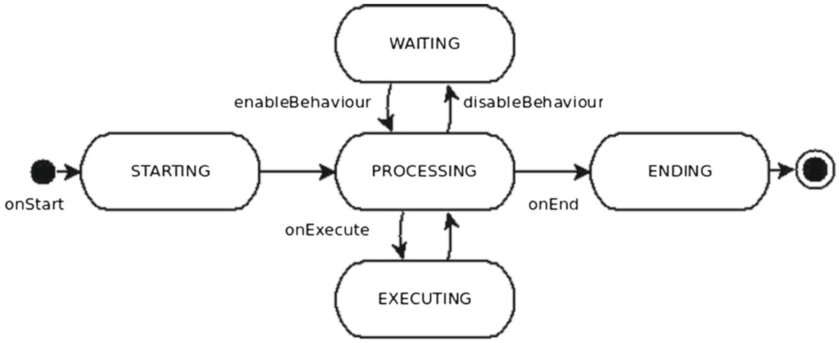


Fig. 6 Behavior's life-cycle

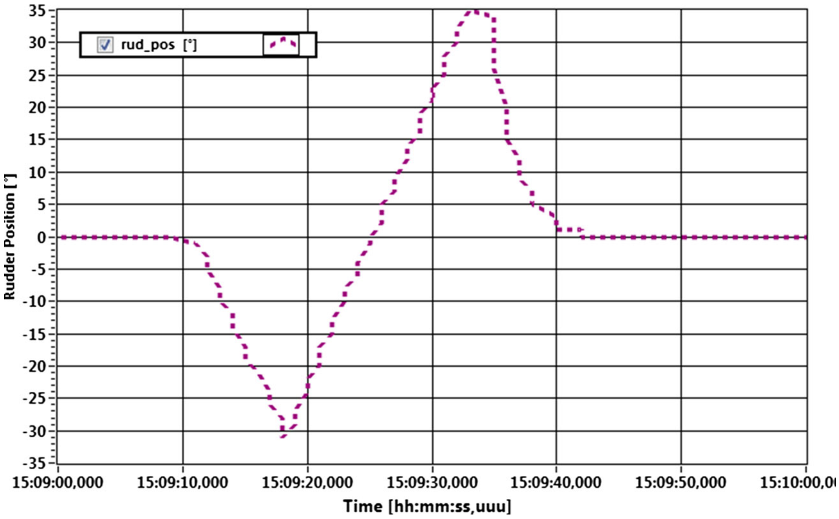


Fig. 7 Rudder calibration Test

execute these behaviors the ASV is closed-loop controlled with the aid of position and velocity information, which is obtained from GPS and IMU data. Target tracking requires information on the target position that can be communicated directly as input to the Controller. Referring to the operative scenario described in [5], where the tracking of the deployed micro-ROV is considered, such information can come directly from a USBL system or from a possible additional ROS agent, which operates the USBL system.

During operation, the Controller can switch seamlessly among the operating conditions without the need to stop and restart the system. It has to be noted that, when the ASV is automatically tracking the micro-ROV, switching to the Remote Guidance mode is assumed to occur only in case of an emergency, like a risk of collision or a failure of some subsystem. In that case, in which tracking becomes secondary with respect to safety, the operator takes direct control of both the ASV and the micro-ROV. Before switching back to the Automatic Guidance mode, if this is the case, it is possible to bring manually the vehicles in a suitable relative position.

Data exchange between the on-board subsystems for control purposes is supported by the Ethernet infrastructure of the Communication subsystem, while data exchange between User and Controller for remote monitoring and guidance must exploit the wireless communication link. Then, the latency of this link has to comply with the responsiveness of the remote operator and with the system time constants. The order of magnitude of the system time constants, expressed in seconds, is that of the unit, as can be inferred from the dynamical model of the system described in [3] and [5]. Using COTS, the communication between the ASV and the shore-ground station is obtained by means of a 802.11 Wi-Fi router, coupled with an amplifier and a high gain omnidirectional antenna (15dBi), on the boat and a directional high gain Wi-Fi extender at the shore-ground station. Minimum transfer time requirements cannot be imposed in 802.11 networks, due to random access time to the connection medium and due to the absence of any Quality of Service (QoS) system. However, as reported in [19] and in [17], latency is almost negligible, if the network load is kept below 40% of its capacity. Supervision of the ASV network during operation has shown that the load stays under 20% of the capacity. Choosing performance indices in accordance with [9], test performed in the harbor of Ancona, Italy, have shown that, at a maximum distance of $500m$, the transmission average delay time is $0.0057s$, with a maximum of $.015s$ and 3% package drop. The communication system in the above configuration, therefore, guarantees operability and a stable connection over distances that, in the present phase, are sufficient for testing the ASV functionality. One can reasonably expect, on the basis of the

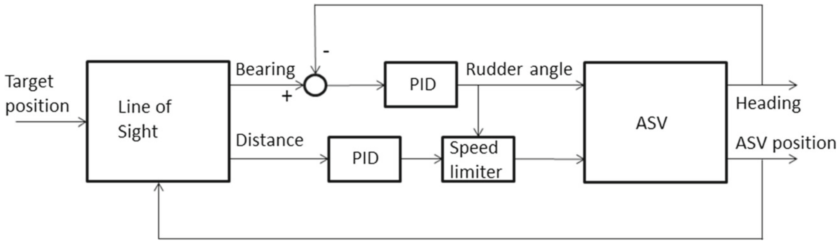


Fig. 8 Scheme of the LoS Controller

above results, that the operative range of the ASV can go up to $1km$ from the shore-ground station. In addition, this choice guarantees easy connection with many commercial devices and embedded boards and it provides high transfer data rates, which, in particular, is useful for video streaming.

4 Experimental Testing

The various components of the NGC system have been preliminary tested in laboratory, both individually and in the MAS configuration. Transportation, set-up and deployment of the robotic system require short time and no marine infrastructure, making the ASV suitable for economical, cost-effective and practical use by a two-man team, while a single operator is required to supervise and to control the ASV after deployment. In order to evaluate the functionalities of both the single modules and the whole system, sea trials, whose results are described and discussed in the next sections, were planned and executed. During those tests, remote guidance, motion along predetermined paths and autonomous target tracking have been taken into consideration. Sea trials took place in the harbor of Porto San Giorgio (Italy), at the marine facilities of Centro Nautico Mare & Corimac srl. During the tests, sea state and swell were 0 in the Douglas sea scale. Figure 2 shows the ASV launching phase. The micro-ROV VideoRay is visible on board. During navigation, one member of the team was on board to comply with safety regulation, which requires the presence of a crew to take control of the vehicle in emergency. During the tests, data concerning navigation and status of the agent and the video stream generated by the IP-camera were transmitted to the shore-ground station and used to update the information on the GUI. The deploy/recovery system of the micro-ROV and the USBL positioning systems were not tested in this phase and the (projection onto the sea surface of) the micro-ROV's position was emulated during the tests about tracking.

4.1 Rudder Calibration

In order to enable remote and autonomous guidance of the ASV, the Rudder subsystem must be calibrated. Practically, this means to reset the incremental shaft encoder of the stepper motor by recording the central and the extreme positions of the rudder. The former corresponds to an orientation of the outboard motor of 0° in the rudder reference frame; the latters correspond to the maximum steering angles on the left and on the right and are defined by the action of two mechanical limit switches. This procedure is needed since orientation is not measured directly (to avoid the need of a dedicated sensor) and the maximum steering angles can slightly change each time the outboard motor is mounted and connected with the steering mechanism. The angular rudder position is controlled in open loop through the steering mechanism, while the stepper motor is closed-loop controlled using the incremental shaft encoder. This structure provides reliable control and precision without needing a dedicated sensor. In operating the system, the rudder calibration procedure is executed at the beginning of each mission, when the ASV is launched. A wizard in the GUI, assists the operator to perform the calibration. The calibration procedure is described below:

1. The stepper motor is disabled, letting the operator to move manually the rudder to the central position;
2. The encoder is reset;
3. The stepper motor is enabled and the rudder is automatically moved from the central position to the maximum steering angles and back. Extreme positions are measured and recorded by the encoder.

A rudder calibration test is shown in Fig. 7: the rudder moves with constant speed, first to the left, then to the right. In this case, the maximum steering angles are $-31°$ and $35°$. Upper and lower saturation limits in the control loop are chosen in accordance with the minimum absolute value of the angles corresponding to the extreme positions (in this case $\pm 31°$).

4.2 Basic Motion

Basic guidance functionalities of the ASV have been tested during sea trials by using, first, basic motion commands. Each basic motion command consists of three values: the commanded thruster speed [%], the commanded rudder position [°] and the command execution time [ms]. The latter specifies the time interval on which the references holds.

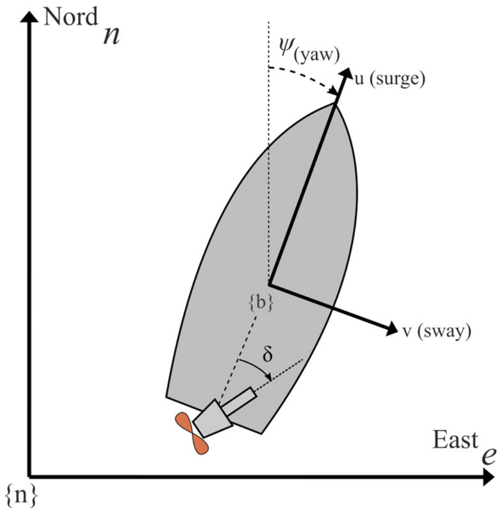


Fig. 9 Vessel reference systems: {n} frame is the local geographical frame, {b} frame is the vessel body-fixed frame, δ is the rudder angle

Table 2 Straight path plan

Commanded Thruster Speed [%]	Commanded Rudder Position [°]	Command Execution Time [s]
0	0	0.2
30	0	6
0	0	0.8

A sequence of basic commands forms a plan and the operator can specify or modify a plan and start its execution from the shore-ground station through the GUI. The User agent communicates plans and basic motion commands to the Controller agent, which sends appropriate reference values to the Rudder and Engine. After expiration of the command execution time, the Controller agent executes the following basic motion command of the plan or stops. Basic motion commands are used to move the vehicle along specific paths, in particular straight and circular paths that can be used to compose various trajectories. Tests have been performed moving the ASV along straight path, characterized by constant thruster angular speed and rudder in the central position, and circular path, characterized by constant thruster angular speed, while the rudder stays at a fixed non-zero angle. Data collected during these tests give first a qualitative evaluation of the response of the system and can then be used to identify its time constants and the values of physical parameters (e.g. drag coefficients, added mass, center of mass) as suggested in [12] and [20].

4.3 Remote Guidance

Remote guidance tests are, then, performed to validate overall functionalities of the robotic structure, including those concerning communication with the shore-ground station. During the test, an operator remotely guides the ASV through the GUI using a joy-pad. Operator's commands are converted to ROS messages by the User agent and sent to the Controller agent through the Wi-Fi link of the communication subsystem.

4.4 Autonomous Tracking

As mentioned in the Section 1, the main task the ASV has to perform is that of tracking a moving target. This ability is required in order to maintain below a given threshold the distance between the ASV and the micro-ROV, when the latter moves underwater at a given depth. Practically, the ASV has to remain close to the vertical (projection onto the sea surface) of the position of the micro-ROV. This facilitates management of the umbilical cable by minimizing its length and burden and it maximizes precision in evaluating acoustically the position of the micro-ROV in survey and intervention tasks. Keeping into account the differences in the update rates of the position of the ASV and of the micro-ROV (GPS is faster than USBL), the most natural way of behaving for the ASV is that of heading and moving toward the projection onto the sea surface of the micro-ROV position only if the relative distance is greater than a given value ρ and to rest if the relative distance is smaller than ρ. Efficacy of such strategy in keeping the relative distance below a given threshold depends obviously on the choice of the parameter ρ and on the velocities of the two vehicles. If the ASV is faster than the micro-ROV, it has been shown in [5] that ρ can be chosen in such a way to satisfy that requirement (the interested reader is referred to [5] for a thorough analysis of such strategy and a procedure to

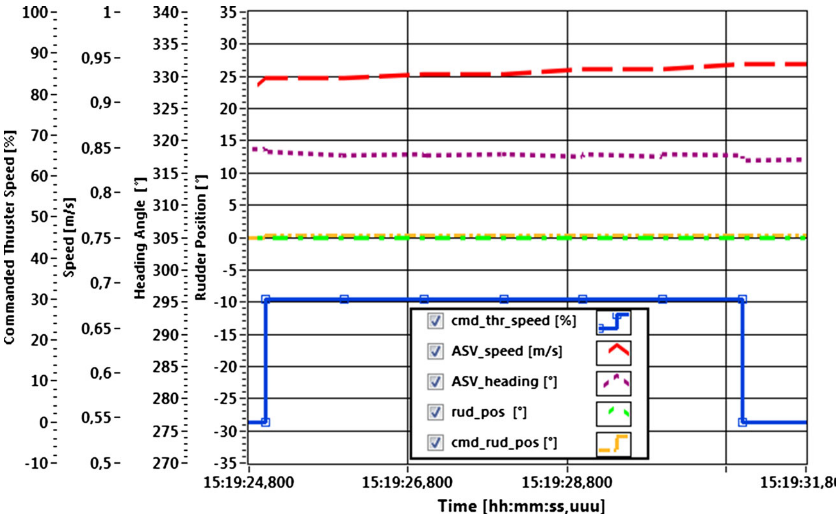


Fig. 10 Sensor data for the straigth path defined by Table 2

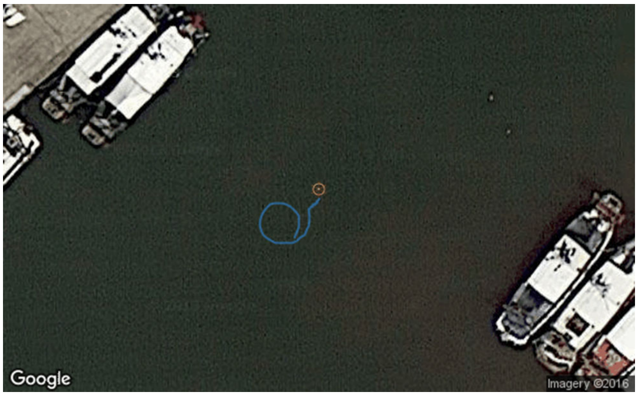


Fig. 11 Trajectory of the ASV during a circular path trial

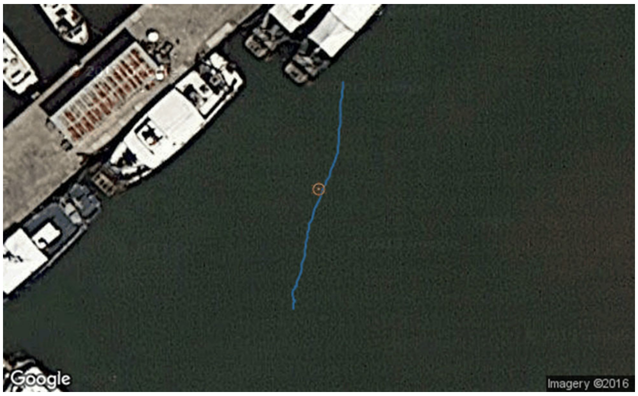


Fig. 13 Trajectory of the ASV during a zigzag path trial

find ρ). In the present campaign of tests, such behaviour is obtained by adopting a Line-of-Sight guidance strategy that is implemented by the NGC system in two steps: first the ASV heading is aligned, by moving the vessel along a circular path, along the loxodrome that joins its position with that of the target and then the ASV is moved in the surge direction at a speed that is proportional to the actual distance. Alignment and speed regulation are performed by applying simple PID controllers. The PID parameters have been first computed by classical Ziegler-Nichols method using the ASV model already employed in [5] and then they have been experimentally tuned. The distance ρ under which the ASV does not move is taken equal to $5m$. The thruster speed is computed taking into account also the rudder steering angle, so to limit the ASV speed during the alignment phase and to accelerate as the alignment error becomes small. This is obtained by the action of a speed limiter that activates to keep the speed command below a threshold (e.g. 30% of the maximum speed) if the rudder angle is greater than a given value in both direction (e.g. 15°). The LoS controller scheme is described in Fig. 8. Tests were performed by forwarding at random times to the NGC system the GPS coordinates of a virtual target by means of a ROS agent that, in future implementation, will get them directly from the USBL system.

5 Experimental Results

The ASV operated for about 5 hours, performing several manoeuvres. During that time, collected data are

- position of the ASV in geographic coordinates;
- reference steering angle for the rudder (as commanded by the NGC system in autonomous mode or given as command in supervised mode) in degrees;
- actual steering angle of the rudder;
- reference angular speed for the thruster (as commanded by the NGC system in autonomous mode or given as command in supervised mode) in percentage of the maximum speed;
- actual angular speed of the thruster;
- messages exchanged between the ROS agents;
- linear accelerations and angular velocities around the three IMU axis.

Power supply from the generator was voluntary cut off several times to test the capability of the power supply system to switch to battery mode and back when connection with the generator was re-established. All the on-board systems stayed active during cut-off, assuring full functionality, and no data were lost. In the following subsections, examples of the general behaviour of the

Fig. 12 Sensor data for the circular path of Fig. 11

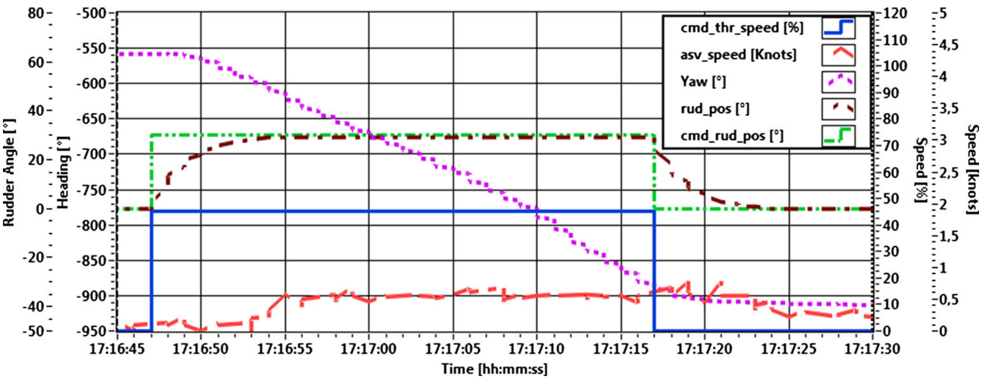


Fig. 14 Sensor data for the zigzag path of Fig. 13

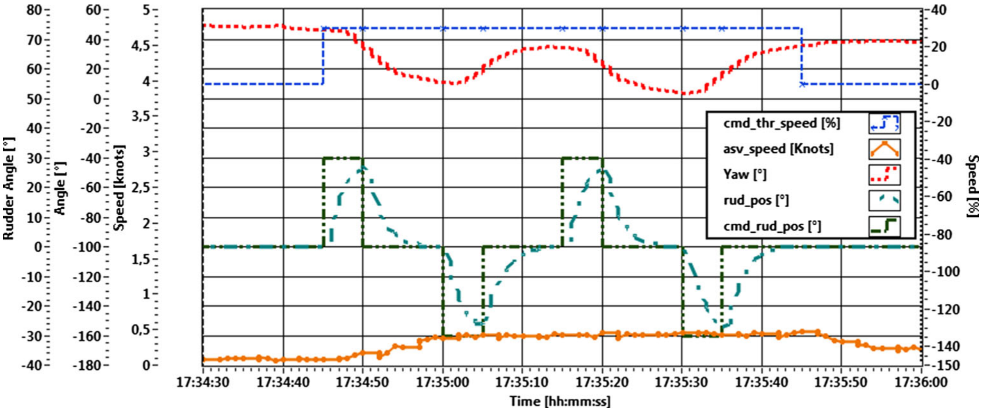


Fig. 15 Trajectory of the ASV during tracking. Point 1,2,3 mark the position of the target at different times

Fig. 16 Position data during the field test shown in Fig. 15

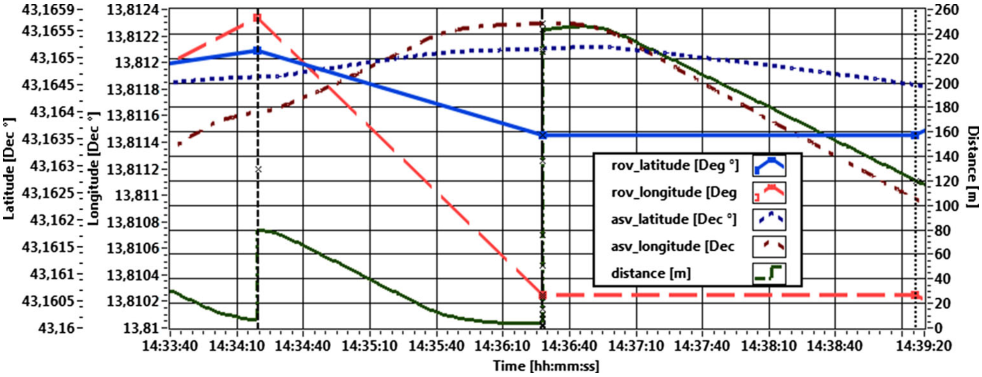
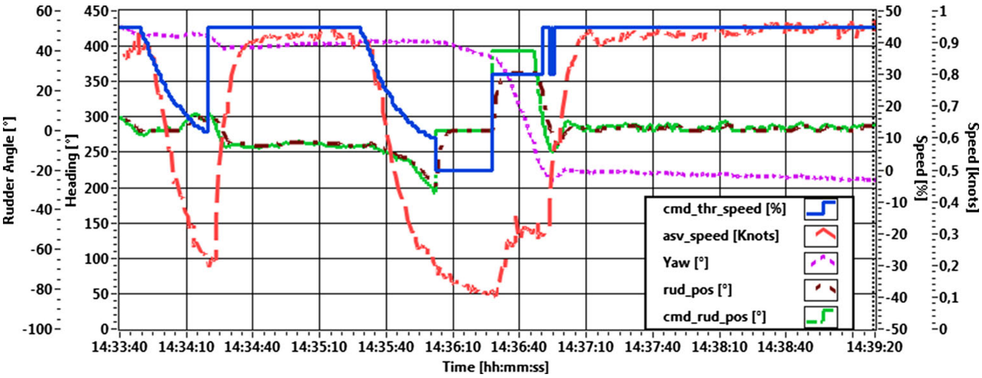


Fig. 17 Sensor data for the trial of Fig. 15



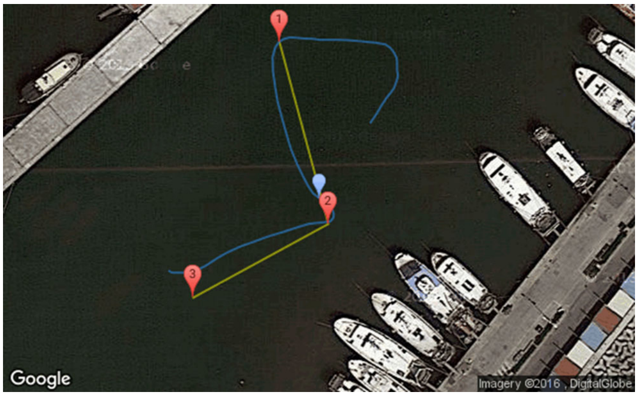


Fig. 18 Trajectory of the ASV during tracking. Point 1,2,3 mark the position of the target at different times

ASV are illustrated by showing trajectories on maps and by plotting: heading ($hdng^\circ$); commanded thruster speed ($cmd\_thr\_speed\%$); commanded rudder steering angle ($cmd\_rud\_pos^\circ$); actual rudder steering angle (rud_pos°); average speed of the ASV ($asv\_speedkn$), as computed by means of position data, together with: position of target ($rov\_latitude$, $rov\_longitude$) that emulates the (projection onto the sea surface of) the position of the microROV; position of the ASV ($ASV\_latitude$, $ASV\_longitude$) and distance ($distance$) from the target during target tracking. The steering system is mechanically limited by two limit switches and the maximum steering angle is 30° in both directions. The thruster angular speed is limited to 315RPM (corresponding to 45% of its maximum value) in order to limit power consumption. The NGC system uses saturation limits of the rudder angle and thruster speed that take into account the thresholds given above. The IMU is fixed on-board in agreement with the reference system described in Fig. 9, with the X axis pointing in the surge direction, the Y axis pointing in the sway direction and the Z axis pointing downward. The GPS module is set in DGPS mode using the WAAS (EGNOS) correction system.

5.1 Straigth Path

Table 2 shows the plan of a straight path executed in Basic Motion mode. Related data are illustrated in Fig. 10. Data show that the heading stays close to the initial value and the ASV speed remains almost constant. This means that the thrust, in that condition, equals the drag force.

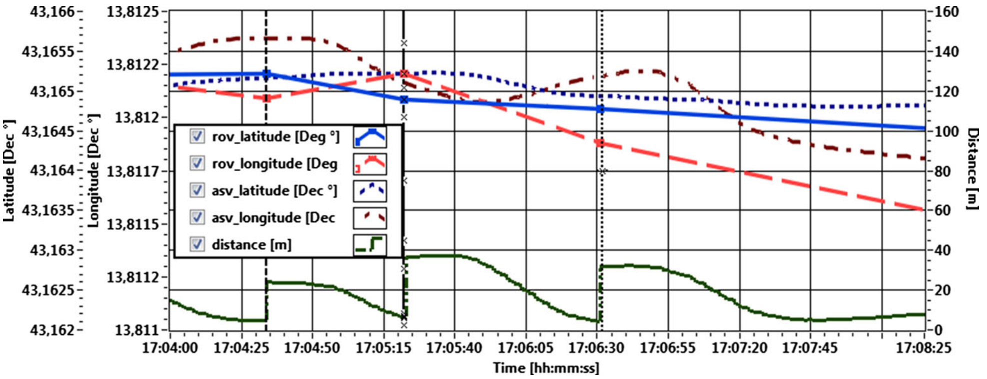
5.2 Circular Path

The circular path shown in Fig. 11 is executed with a commanded rudder steering angle of 30° and a commanded thruster speed at 45%. As described in Fig. 12, after a transient in which the system reaches the steady state, the heading changes almost linearly (approximating a straight line with angular coefficient 9.9), with negligible oscillations. Angular speed, in response to the control input, stabilizes after about 8s. Note that heading is represented without discontinuities using an extended scale for the angular measure.

5.3 Zigzag Path

The path shown in Fig. 13 exhibits a zigzag pattern obtained by moving at constant speed and changing periodically the rudder steering angle. In the situation illustrated by Fig. 14, the commanded rudder steering angle was sequentially set to $30^\circ, 0^\circ, -30^\circ, 0^\circ$ for, respectively $5s, 10s, 5s, 10s$ repeating two times the sequence, and the commander thruster speed was 30%. Data show the response of the rudder subsystem and of the ASV. Starting from 0°, the rudder steering angle reaches 90% of the saturation values $30^\circ, -30^\circ$ in about 5s and it returns to $0\circ$ in about $7, 5s$. In response to the motion of the rudder, the ASV heading exhibits a transient behavior that, commanding the steering

Fig. 19 Position data during the field test shown in Fig. 18



angle from 0° to saturation for $5000ms$ and then again to 0°. terminates in about $5s$.

5.4 Automatic Tracking

Automatic tracking tests were performed by forwarding at random times to the NGC system the GPS coordinates of a virtual target by means of a ROS agent that, in future implementation, will get them directly from the USBL system.

Figure 15 refers to a worst-case situation in which, at a given time, the bearing is opposite to the heading. The ASV, then, has to invert its heading to align it with the bearing. Point 1, 2, 3 in the map mark subsequent positions of the virtual target, whose trajectory is outlined in yellow. The blue trajectory of the ASV shows that it goes toward the target positioned in 1 and in 2 and then it inverts its direction to go toward position 3. Position data are shown in Fig. 16. Discontinuities in distance indicate the times at which position 1, 2, 3, respectively, of the virtual target are forwarded to the NGS system. As it can be seen, distance slightly increases while the ASV re-orients inverting its heading and then it start decreasing again. The time required for aligning the ASV in the worst-case situation is seen to be about $11s$. The ASV stops when the distance from the virtual target becomes smaller than $5m$ and it reacts as soon as this condition is violated. Note that the virtual target is assumed to move faster than the ASV in going from position 2 to position 1.

Data collected in this situation are shown in Fig. 17. These data can be used to tune the sliding mode controller that has been proposed in [5] to implement the tracking strategy.

Figure 18 refers to another trial. Point 1, 2, 3 in the map mark subsequent positions of the virtual target, whose trajectory is outlined in yellow. The blue trajectory of the ASV shows that it goes toward the target and data are in accordance with those reported in the previous path. In this case the virtual target was moved assuming that its speed is not greater than the speed of the ASV and, as a result, the relative distance keeps smaller than $40m$. Position data are shown in Fig. 19.

6 Conclusions

Implementation of a multi-agent structure using ROS has been presented as a viable methodology for developing a performing NGC system for a small ASV. A custom assembled electronic board that incorporates two different processors has been constructed in order to allow the use of computational demanding ROS software procedures, to guarantee easy interfacing with sensors and actuators and to simplify design and prototyping. The tests performed on the ASV under construction have qualitatively shown operability and reliability of its mechatronic structure. Manoeuvring and tracking capabilities guarantee satisfactory performances in maintaining formation with the deployed micro-ROV. Future tests will experiment deployment/operation/recovery of the micro-ROV with the aid of information about the position of the micro-ROV coming from an USBL system. In order to perform those tests, an additional ROS agent, already constructed and tested, will be integrated in the ROS architecture to receive data from the USBL system. The micro-ROV control console will be installed on-board and connected with the shore-ground station through the existing Wi-Fi connection in order to make possible remote control. Preliminary laboratory tests have already been done in order to assess performances of the network with positive results.

References

1. Benjamin M. R., Leonard, J.J., Schmidt, H., Newman, P.M.: An overview of moos-ivp and a brief users guide to the ivp helm autonomy software. Technical Report, Massachusetts Institute of Technology, Computer Science and Artifcial Intelligence Lab (CSAIL), Cambridge (2009)
2. Busquets, J., Zilic, F., Aron, C., Manzoliz, R.: AUV and ASV in twinned navigation for long term multipurpose survey applications. In: Proceedings MTS/IEEE OCEANS 13, pp 1–10. Bergen (2013)
3. Conte, G., De Capua, G.P., Scaradozzi, D.: Modeling and control of a low-cost ASV. In: IFAC Proceedings Volumes, 9th IFAC Conference on Maneuvering and Control of Marine Craft, vol. 45, pp. 429–434. Arenzano (2012)
4. Conte, G., Scaradozzi, D., Sorbi, L., Panebianco, L., Mannocchi, D.: ROS multi-agent structure for autonomous surface vehicles. In: Proceedings Oceans 2015 MTS/IEEE Conference and Exhibition, pp. 1–6. Genova (2015)
5. Conte, G., De Capua, G.P., Scaradozzi, D.: Designing the NGC system of a small ASV for tracking underwater targets. Robot. Auton. Syst. **76**, 46–57 (2016)
6. Djapic, V., Nad, D.: Using collaborative autonomous vehicles in minecountermeasures. In: Proceedings Oceans 2010 MTS/IEEE Conference and Exhibition, pp. 1–7. Sidney (2010)
7. Encarnaċao, P., Pascoal, A.: Combined trajectory tracking and path following: An application to the coordinated control of autonomous marine craft. In: Proceedings of the 40th IEEE Conference on Decision and Control, pp. 964–969. Orlando (2001)
8. Ferreira, H., Martins, A., Dias, A., Almeida, C., Almeida, J.M., Silva, E.P.: ROAZ autonomous surface vehicle design and implementation. Robótica **67**, 1–8 (2007)
9. Guosong, T., Yu-Chu, T., Fidge, C.: Performance analysis of IEEE 802.11 DCF based WNCS networks. Proc. 35th IEEE Conf. Local Comput. Netw. **10**(14), 496–50 (2010)
10. Healey, A.J., Pascoal, A., Santos, R., Silvestre, C., Oliveira, P., Sebastiao, L., Rufino, M., Alves, J.: Shallow water hydrothermal vent survey in azores with cooperating ASV and AUV. Internal report, Department of Mechanical Engineering, Naval Post-Graduate School, Monterey (1999)

11. Langley, P.: Cognitive architectures and the construction of intelligent agents. In: Proceedings of the AAAI Workshop on Intelligent Agent Architectures, pp. 82–88. Stanford (2004)
12. Linder, J., Enqvist, M., Fossen, T.I., Johansen, T.A., Gustafsson, F.: Modeling for IMU-based online estimation of a ship's mass and center of mass. IFAC-PapersOnline **48**, 198–203 (2015)
13. Melo, J., Matos, A.: Guidance and control of an ASV in AUV tracking operations. In: Proceedings Oceans 2008 MTS/IEEE Conference and Exhibition, pp. 1–7. Quebec City (2008)
14. Miskovic, N., Bibuli, M., Bruzzone, G., Caccia, M., Vukic, Z.: Tuning marine vehicles' guidance controllers through self-oscillation experiments. In: Proc. 8th IFAC Conf. Manoeuvring Control Marine Craft, pp. 115–120, Guaruja (2009)
15. Pascoal, A., Oliveira, P., Silvestre, C., Sebastiao, L., Rufino, M., Barroso, V., Gomes, J., Ayela, G., Coince, P., Cardew, M., Ryan, A., Braithwaite, H., Cardew, N., Trepte, J., Seube, N.: Robotic ocean vehicles for marine science applications: The European ASIMOV project. In: Proceedings Oceans 2000 MTS/IEEE Conference and Exhibition, pp. 409–415. Providence (2000)
16. Quigley, M., Conley, K., Gerkey, B., Faust, j., Foote, T., Leibs, J., Wheeler, R., Ng, A.: ROS: An open-source robot operating system. Proc.ICRA Workshop Open Source Softw. **2009**, 3 (2009)
17. Tian, G., Tian, Y.C., Fidge, C.: Local Computer Networks. In: Proceedings 35th IEEE Conference on Performance analysis of IEEE 802.11 DCF based WNCS networks, pp. 496–503. Denver (2010)
18. Vasilijević, A., Nad, D., Mandić, F., Mišković, N., Vukić, Z.: Coordinated navigation of surface and underwater marine robotic vehicles for ocean sampling and environmental monitoring. IEEE/ASME Trans. Mechatron. **22**(3), 1174–1184 (2017)
19. Wheels, J.D.: Process control communications: Token bus, CSMA/CD, or token ring? ISA Trans. **32**, 193–198 (1993)
20. Yoon, H.K., Rhee, K.P.: Identification of hydrodynamic coefficients in ship maneuvering equations of motion by Estimation-Before-Modeling technique. Ocean Eng. **30**(18), 2379–2404 (2003)

Giuseppe Conte received the Laurea in Mathematics from the Università di Genova, Genoa, Italy, in 1974. He is currently Professor of Automation at the Università Politecnica delle Marche, Ancona, Italy. His research interests are in algebraic and geometric system and control theory and in robotics. He has authored several papers in journals, conference proceedings and books; he edited two books and he coauthored the research monographs Algebraic Methods for Nonlinear Control Systems. Theory and Applications (London, U.K.: Springer-Verlag 2007). He served on the editorial boards of several journals, as Program Chair and General Chair of systems and control conferences and currently serves as Associate Editor of IMA Journal of Mathematical Control and Information. Prof. Conte is Chairman of the IFAC Technical Committee on Linear Control Systems.

David Scaradozzi got his PhD from the Università Politecnica delle Marche in 2005, He is currently Assistant Rrofessor at the Università Politecnica delle Marcheè and Chercheur Associé at the Laboratoire des Sciences de l'Information et des Systémes CNRS of Marseille, France. After receiving the Ph.D. in Artificial Intelligent Systems, he worked at Indesit Company, and at Inter-University Center of Integrated Systems for Marine Environment (ISME) as a senior scientist. His research interests are in mechatronics, marine and educational robotics; he has several publications in international refereed journals and in international conferences proceedings. He is Chairman of the International Mobility Pregrammes of the Engineering Faculty of the Politechnic University of Marche.

Daniele Mannocchi received the Master's Degree in Automation Engineering from the Università Politecnica delle Marche, Ancona, Italy, in 2014. He is currently working as software developer in Videx Electronics, a company that designs and manufactures audio and video entry systems and access controls. His research interests are monitoring, control and management of distributed systems.

Paolo Raspa received his Ph.D. in Automation Engineering from the Università Politecnica delle Marche, Ancona, Italy, in 2013. He currently holds the position of system electronics specialist in the R&D engineering team within the Technology and Innovation Department of Ariston Thermo Group, one of the European leader companies in thermic comfort and energy efficiency.

Luca Panebianco received the Master's degree in Automation in Information Engineering from the Università Politecnica delle Marche, Ancona, Italy, in 2014. He is currently a PhD student at the Department of Information Engineering (DII) of the same University. His field of research regards the design and development of solutions to address cooperation, collaboration and modularity to a wide range of autonomous vehicles by means of Multi-Agent System theory.

Laura Screpanti received the Master's Degree in Electronic Engineering from the Università Politecnica delle Marche, Ancona, Italy, in2014. She is currently a Ph.D. student at the Department of Information Engineering of the same University. Her research interests are in biomedical engineering, underwater robotics and educational robotics.