da allegare (in pdf) al PROOF dell'articolo:

Link sito dell'editore: https://ieeexplore.ieee.org/document/8485290

Link codice DOI: https://doi.org/10.1109/MRA.2018.2869523

Citazione bibliografica dell'articolo:

Andreas Birk, Tobias Doernbach, Christian Atanas Müller, Tomasz Luczynski, Arturo Gomez Chavez, Daniel Köhntopp, Andras Kupcsik, Sylvain Calinon, Ajay K. Tanwani, Gianluca Antonelli, Paolo di Lillo, Enrico Simetti, Giuseppe Casalino, **Giovanni Indiveri**, Luigi Ostuni, Alessio Turetta, Andrea Caffaz, Peter Weiss, Thibaud Gobert, Bertrand Chemisky, Jeremi Gancet, Torsten Siedel, Shashank Govindaraj, Xavier Martinez, and Pierre Letier
"Dexterous Underwater Manipulation from Onshore Locations: Streamlining Efficiencies for Remotely Operated Underwater Vehicles," in IEEE Robotics & Automation Magazine, vol. 25, no. 4, pp. 24-33, Dec. 2018, doi: 10.1109/MRA.2018.2869523.

Dexterous Underwater Manipulation from Onshore Locations

Streamlining Efficiencies for Remotely Operated Underwater Vehicles

By Andreas Birk, Tobias Doernbach, Christian Atanas Müller, Tomasz Luczynski, Arturo Gomez Chavez, Daniel Köhntopp, Andras Kupcsik, Sylvain Calinon, Ajay K. Tanwani, Gianluca Antonelli, Paolo di Lillo, Enrico Simetti, Giuseppe Casalino, Giovanni Indiveri, Luigi Ostuni, Alessio Turetta, Andrea Caffaz, Peter Weiss, Thibaud Gobert, Bertrand Chemisky, Jeremi Gancet, Torsten Siedel, Shashank Govindaraj, Xavier Martinez, and Pierre Letier

Underwater manipulation is a challenging problem. The state-of-the-art technology is dominated by remotely operated vehicles (ROVs). ROV operations typically require an offshore crew consisting of, at minimum, an intendant (or supervisor), an operator, and a navigator. This crew must often be doubled or even tripled due to work shifts. In addition, customer representatives often wish to be physically present offshore. Furthermore, underwater intervention missions are still dominated by a significant amount of low-level, manual control of the manipulator(s) and of the vehicle itself. While there is a significant amount of research on autonomous underwater vehicles (AUVs) in general and fieldable solutions already exist for inspection and exploration missions, possibilities remain for adding intelligent autonomous functions for interventions.

The work we present here is intended to reduce the number of robot operators required offshore—hence, reducing costs and staffing requirements—by facilitating operations from an onshore control center and narrowing the gap between low-level teleoperation and full autonomy (Figure 1). The basic idea is that the user interacts with a real-time simulation environment, and a cognitive engine (CE) analyzes the user's control requests and turns them into movement primitives the ROV needs to autonomously execute in the real environment, independently of communication latencies.

This article focuses on the results of intensive field trials held 26 June–7 July 2017 in the Mediterranean Sea near Marseille, France. Seven extended experimental dives were performed with the ROV while connected via satellite to the command center in Brussels, Belgium. Four different sites were used with different water depths (8, 30, 48, and 100 m).

System Components

Overview

Our work here is targeted at a high technology readiness level of six, i.e., it is developed and validated beyond only laboratory experiments. The research vessel Janus II from Comex with a 2,500-msw rated SubAtlantic Apache ROV is used for this purpose (Figure 2). For our research, the ship is equipped with satellite communications, explained in the "Satellite Communication" section, to allow control of the ROV by pilots located in a command center in Brussels, as discussed in the section "Control Center and the Exoskeleton." Furthermore, a skid is added to the ROV to carry additional components used for our research, i.e., an electric manipulator and two manipulators in a bimanual setup (as described in the section "Underwater Manipulators") and a vision system.

Underwater Manipulators

Our manipulator was designed beginning with the underwater modular arm. Two kinds of electrically driven joints having either one or two motion axes are complemented by a set of links to connect the joints. Different kinematic configurations can be obtained by varying the number of basic modules, i.e., joints and links, and/or the way they are interconnected. The arm is characterized by six degrees of freedom (DoF), obtained by connecting three modules, each with two DoF forming a pitch-roll configuration. The overall length when totally stretched is slightly more than 1 m. However, the arm is also fully foldable to minimize its size when parked in the ROV skid during the navigation phases. Both a single-arm and a dual-arm setup can be used. During the 2017 trials, a mockup of grippers currently under development was used.

Vision System

An intelligent underwater vision system with computing power on board the ROV is used to minimize the traffic over the umbilical cable from the ROV to the vessel. The system is based on high-resolution firewire (IEEE 1934b) cameras in pressure housings connected to an embedded computer, which can be used for vision processing and adaptive video compression on board the ROV.

The firewire bus supports, among others, the synchronization of the cameras. They can therefore be used for multicamera stereo setups to generate depth information from different views with a known relative geometry. Because of the Apache ROV's payload constraints, a stereo setup with two cameras was used in the 2017 trials. The compute bottle of the vision system on board the robot also services the core navigation sensors in the form of a LinkQuest NavQuest 600 P Micro Doppler velocity log (DVL) and a Xsens MTi-300 inertial measurement unit (IMU).

Satellite Communication

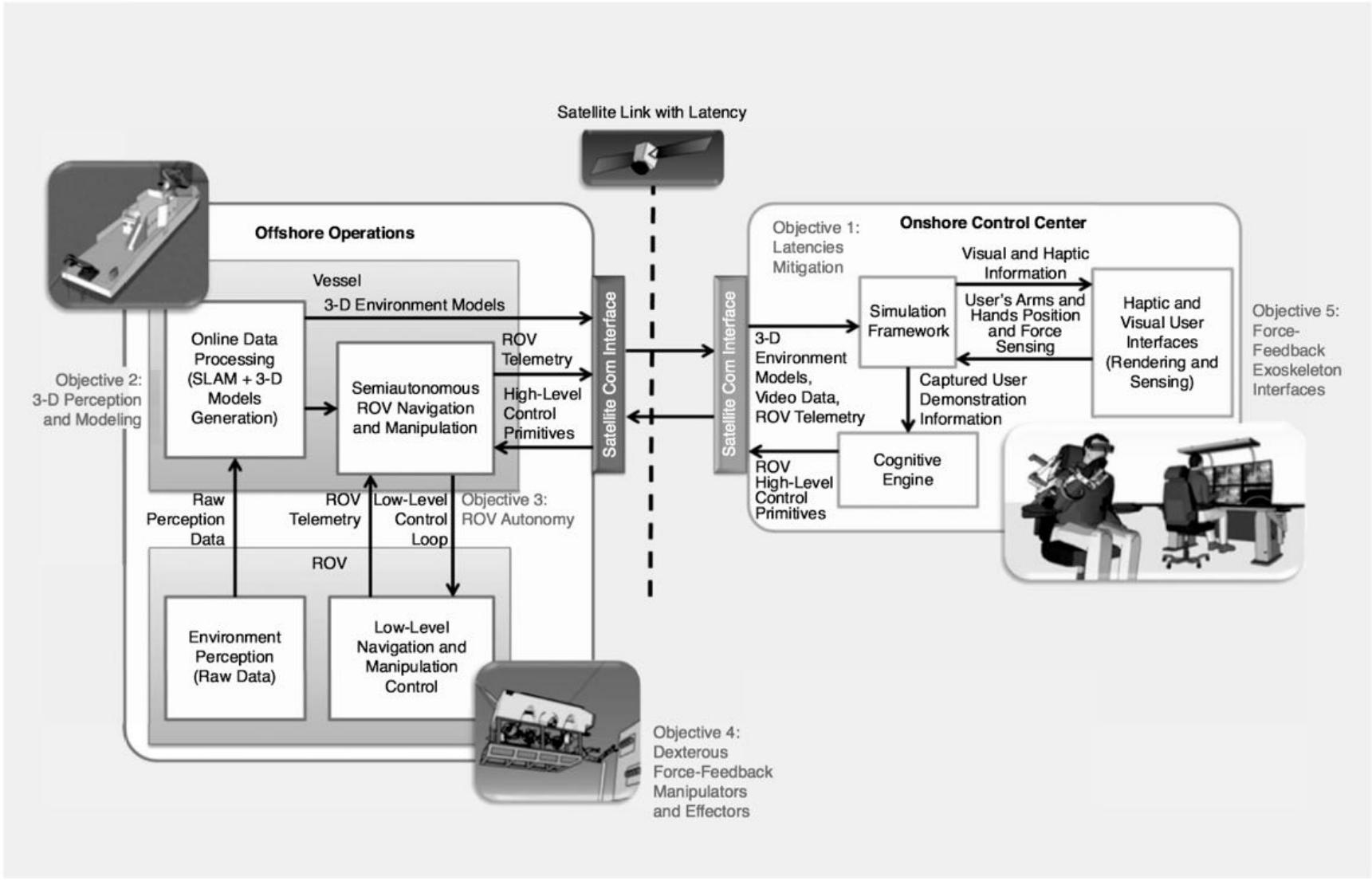
Satellite communications services for mobile offshore maritime operations are associated with bandwidth limitations (uplink and downlink), inherent delays, and disruptions; in addition, they require a complex stabilized satellite tracking antenna. In the context of this research, we employ a maritime very-small-aperture terminal from a service provider (Omniaccess) that includes a Ku-band Cobham Sailor 800 tracking antenna, its controller, and the related modems. The nominal data bandwidth for the uplink from the vessel is 768 kB/s, and the downlink to the vessel is 256 kB/s, with an inherent nominal round-trip delay of 620 ms.

Control Center and the Exoskeleton

The onshore control center in Brussels consists of a monitoring and control room that features a double 7-DoF arm and 6-DoF hand-force feedback exoskeleton. It is based, in part, on a design for the European Space Agency [1] that was further improved on in the EU-FP7 project ICARUS [2]. It is designed as a modular solution, allowing each arm and hand exoskeleton subsystem to be easily and conveniently connected and removed from the rest of the setup. Furthermore, it features a passive gravity compensation system that connects to the arm exoskeletons and can be calibrated to compensate for the full mass of the exoskeleton's physical setups as well as the mass of the user arms. The user is, hence, given the impression of operating in neutral buoyancy, i.e., as a diver typically would. This reduces user fatigue during the ROV operation.

A Test Panel for System Validation

For validation, a test panel was developed; it also served as a target for the trials to emulate different scenarios, e.g., offshore oil and gas facilities or the handling of archeological artifacts (Figures 2 and 3). The panel consisted of three sides, each equipped with mockup elements. One side was used to test functionalities in offshore oil and gas interfaces based on the International Standards Organization 13628 standard including, e.g., valves or wet-mate connectors. In addition, a biologic panel (including mockup corals) and an archeological box (including mockup ceramics) were included.


Satellite Link with Latency
Offshore Operations
Vessel
3-D Environment Models
Online Data Processing (SLAM + 3-D Models Generation)
Semiautonomous ROV Navigation and Manipulation
ROV Telemetry
High-Level Control Primitives
Satellite Com Interface
Objective 2: 3-D Perception and Modeling
Raw Perception Data
ROV Telemetry
Low-Level Control Loop
Objective 3: ROV Autonomy
ROV
Environment Perception (Raw Data)
Low-Level Navigation and Manipulation Control
Objective 4: Dexterous Force-Feedback Manipulators and Effectors
Objective 1: Latencies Mitigation
Onshore Control Center
Satellite Com Interface
3-D Environment Models, Video Data, ROV Telemetry
Simulation Framework
Visual and Haptic Information
User's Arms and Hands Position and Force Sensing
Haptic and Visual User Interfaces (Rendering and Sensing)
Objective 5: Force-Feedback Exoskeleton Interfaces
Captured User Demonstration Information
Cognitive Engine
ROV High-Level Control Primitives

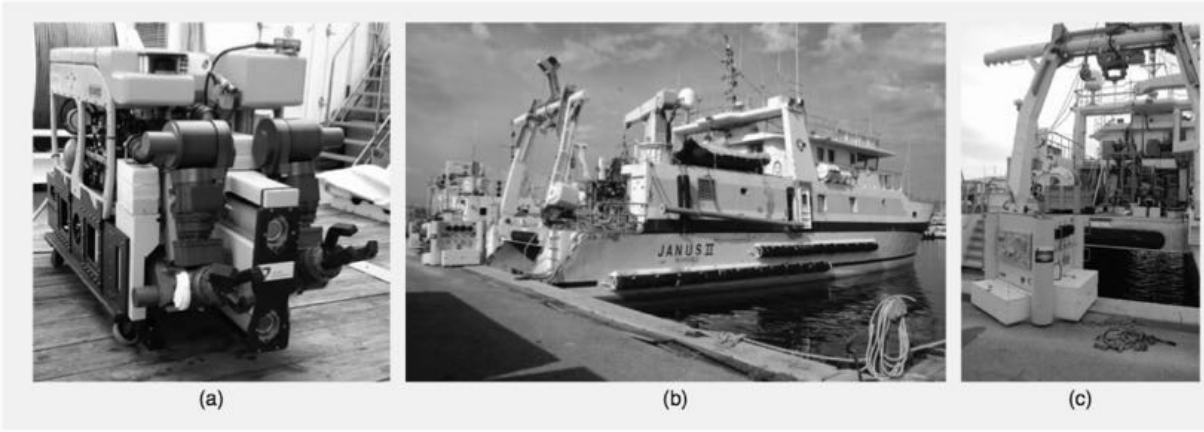


Figure 2. The system tested in field trials consists of (a) an Apache ROV extended with newly developed components, including a dual-arm setup and an intelligent vision system, which is deployed from the (b) Comex Janus II vessel having a satellite connection to a control center in Brussels. (c) A mockup panel structure is used to test different application scenarios. (Photos courtesy of DexROV consortium.)

Data Flows Between the Vessel/ROV and the Control Center

To sustain effective remote ROV operations, multiple data flows are required, such as those for ROV commands, video streams, pose updates, 2.5-dimensional (2.5-D) or three-dimensional (3-D) environmental maps, status updates, and so on, between the onshore and offshore nodes; these data flows are transmitted via the satellite link. It is, therefore, critical to optimize the bandwidth usage by prioritizing data flows with specific quality of service (QoS) information, shaping the traffic [3] to avoid network overload and ensure data reliability with minimal overheads. To address the described challenges, the Data Distribution Service (DDS) OpenSplice (Prismtech) middleware is used to exchange data between the onshore and offshore nodes over the bandwidth-constrained satellite network.

The onshore control center and the ROV control and perception framework embed command and data interfaces in the robot operating system (ROS) through asynchronous publish–subscribe mechanisms over named and type-specific topics. A ROS-to-DDS bridge has been developed and can be configured to interface with existing ROS topics in a system. For each ROS topic, the ROS-to-DDS bridge automatically creates a corresponding DDS data reader, data writer, and named topics across the distributed nodes with associated QoS policies according to the object management group's DDS QoS specification.

The architecture of the bridge is scalable to deploy multiple nodes for dynamic discovery of distributed ROS-to-DDS entities. The maximum burst sizes indicate the amount, in bytes, to be sent at maximum, with every resolution in milliseconds. With reliable QoS, i.e., guaranteed data delivery, the maximum burst values are typically set just below the maximum bandwidth available for the uplink from the offshore side. For example, in the presence of a 768-kB/s satellite uplink, a maximum burst size of 650 kB/s, i.e., around 85% of the bandwidth, was found to be efficient. The remaining bandwidth is made available for retransmissions of data packets. In the case of the best-effort QoS, i.e., it is not necessary to resend or acknowledge any received packets, the maximum burst size can be set to the available bandwidth. During the marine trials, the satellite link was shared between multiple data flows that were assigned different priorities, and the maximum burst sizes were adapted proportionately.

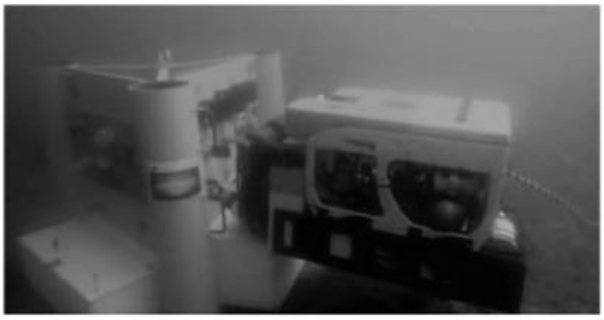


Figure 3. The ROV and the test panel during the field trials. (Photo courtesy of DexROV consortium.)

Manipulation Tasks with the CE

The main purpose of the CE is to overcome teleoperator control delays in the face of satellite communication latency. Prior to the mission, the teleoperator first demonstrates a set of tasks (e.g., turning a valve or grasping a handle) using the exoskeleton in the onshore control center, which the CE encodes as statistical models. These models are transferred to the offshore vessel, where control of the ROV takes place. During mission execution, the offshore model assists the teleoperator-guided manipulation such that the tasks are replicated by adapting them to the current environmental situations. This reduces the cognitive load on the teleoperator, who can concentrate on selecting the tasks in the virtual environment.

The CE can assist the teleoperator in two different modes [4]:

1) *Shared control.* The teleoperator input is combined directly with the motion predicted by the task model. The

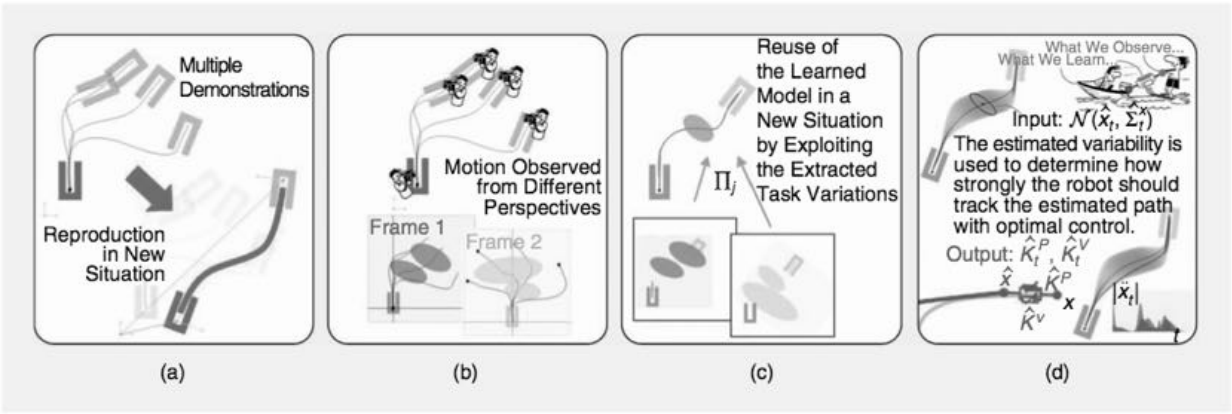


Figure 4. An overview of the learning approach used in the CE. (a) Demonstrations are collected with different task parameters (frames of reference). (b) The demonstrations are transformed in each particular frame, and a GMM is learned in each frame. (c) In a new situation, a new GMM is computed with a product of linearly transformed Gaussians. (d) The computed trajectory distribution provides a variance estimate for each set-point, which determines how accurately the robot should pass through these set-points.

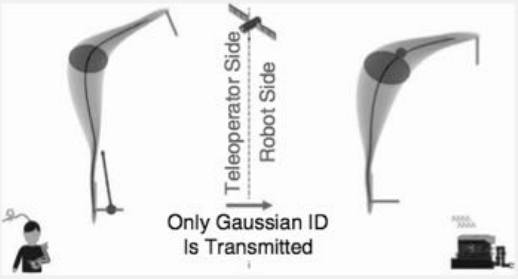


Figure 5. The shared control mode with two frames of reference on each side. The current poses of the teleoperator and robot are displayed with blue dots. ID: identification.

Figure 6. The semiautonomous control mode, with acceleration commands and an associated trajectory computed from any of the robot/teleoperator poses (displayed as blue points) using the model.

adaptation is weighted based on the variability of the demonstrations in those parts of the task that are currently executed. For parts of the task requiring accuracy, the model assists the teleoperator by correcting deviations from the original demonstrations. For parts of the task allowing more variations, the teleoperator is free to move within the regions corresponding to the demonstrations.

2) *Semiautonomous control.* The task is executed by generating the most likely trajectory starting from the current pose of the robot. This mode is particularly useful when delays or interruptions from the satellite communication are expected. In this control mode, the teleoperator visualizes and triggers the execution of the movement, which continues until a new signal from the teleoperator is given [5].

In the following, we summarize task learning and how motions are reproduced in varying situations.

Learning Adaptive Tasks from Demonstrations

Our application requires the CE to learn skills based on only a handful of demonstrations (typically up to ten). Additionally, we require skills that can be reproduced in novel environmental situations, for which no demonstration is available. To achieve this goal, the CE relies on a task-parametrized Gaussian mixture model (GMM) [6] to encode demonstrations executed in different situations (Figure 4). The task parameters are frames of reference (coordinated systems with position and orientation information) associated to virtual landmarks or objects/tools in the environment. For example, in a valve-turning task, such frames may refer to the robot base frame, the current valve pose, and the targeted valve pose.

Figure 4 depicts the learning and retrieval process. First, demonstrations are collected in varying situations (each time with different task parameters). To capture the variance of the demonstrations, GMMs are learned in each task-relevant frame. Learning the models in each individual frame allows the system to generalize the observed tasks to new situations.

Task Reproduction with Adaptation to New Situations

In a novel environmental situation, the Gaussian-mixture components are transformed using the newly observed task parameters [Figure 4(c)]. The retrieved GMM is exploited differently according to the selected control mode, with the aim of reducing the cognitive load on the teleoperator when executing a set of tasks.

In the shared control mode (Figure 5), Gaussian-mixture regression is used on both the teleoperation and robot sides to generate probabilistic trajectory distributions, represented as a tube in the figure. On the teleoperator side, this tube is adapted locally to match the situation of the virtual environment in which the user is immersed—here, the model can be used, e.g., for haptic corrections. On the robot side, the same model adapts to the situation that is locally detected. This situation can potentially differ from the one currently experienced by the user, as depicted in the figure: the two tubes have different shapes but share the same GMM parameters.

In this way, the robot is provided with a fast adaptation technique that can directly exploit the locally sensed information, i.e., the tube is adapted online without passing through the slow satellite communication. This type of assistance is relevant for handling small transmission delays, i.e., to cope with situational discrepancy due to the slow refreshing rate. For longer delays, a semiautonomous mode, as described next, is usually preferred.

In the semiautonomous control mode (Figure 6), a linear quadratic tracking controller in operational space is used to generate a reference trajectory starting from the current robot pose. These acceleration commands in operational space are used by the robot controller until the teleoperator decides to abort the task or switch to another task. On the teleoperator side, the retrieved trajectory is used for visualization purpose.

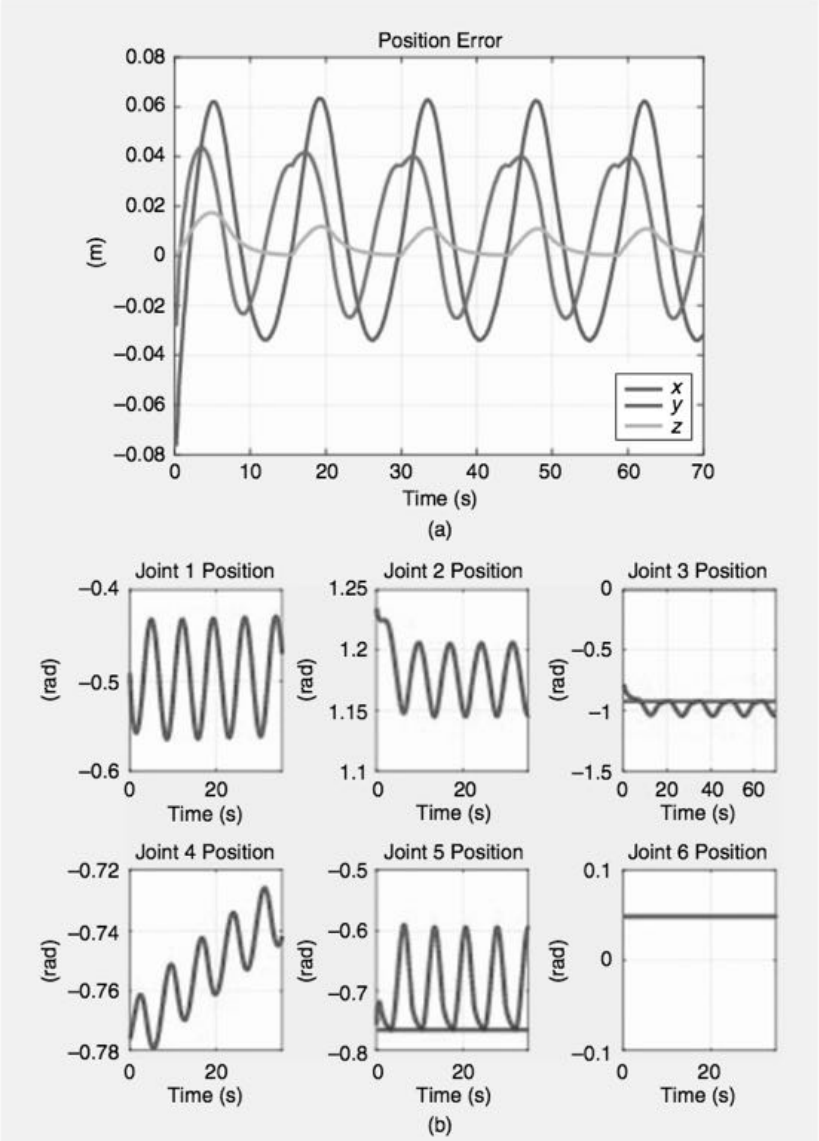


Figure 7. (a)The position error over time during the experiments. The error is relatively high because the control gains were maintained at low levels for safety reasons during these tests. (b) The joint positions and the minimum/maximum thresholds are shown (in red). The plots show how the TPIK approach enforces the validity of the joint limits.

ROV Control for Intervention Missions with Communication Latencies

Task Priority Inverse Kinematics

From a control point of view, the DexROV system is much more similar to an AUV than an ordinary ROV in the sense that it needs to take care of many control objectives on its own, with high-level inputs coming from the user through the CE.

For this reason, following an approach similar to the one adopted for the Trident [7] and Maris [8] projects, the developed control is a task priority inverse kinematics (TPIK) algorithm that allows one to set a priority order among several tasks and find the system velocity vector $\dot{\boldsymbol{y}}$ that accomplishes them simultaneously, at best, following the priority order. Given a hierarchy composed of k tasks $\dot{\boldsymbol{\sigma}}_1 \ldots \dot{\boldsymbol{\sigma}}_k$, the target velocity $\dot{\boldsymbol{y}}$ can be computed as

$$\dot{\boldsymbol{y}} = \dot{\boldsymbol{y}}_1 + \boldsymbol{N}_1 \dot{\boldsymbol{y}}_2 + \cdots + \boldsymbol{N}_{1,k-1} \dot{\boldsymbol{y}}_k, \tag{1}$$

where each $\dot{\boldsymbol{y}}_i$ is the velocity contribution of task i, and $\boldsymbol{N}_{1,i}$ is the null space of the augmented Jacobian matrices from $\boldsymbol{\sigma}_1$ to $\boldsymbol{\sigma}_i$. If there are conflicting tasks, the projection of the velocity contribution of the lower-priority tasks into the null space of the higher-priority ones guarantees that the priority order is always respected. In addition, this control framework has

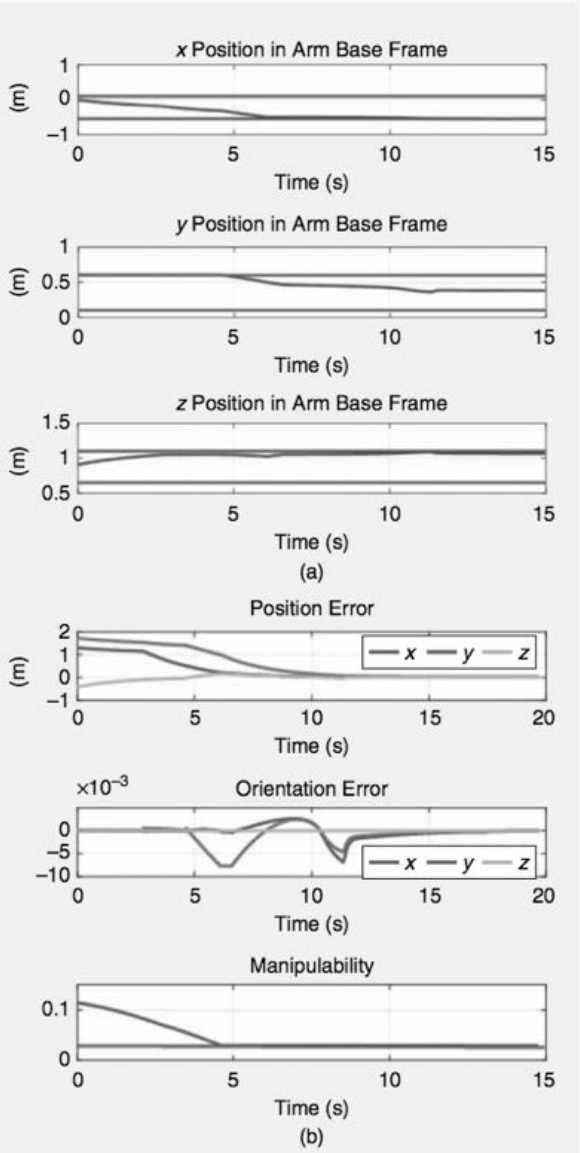


Figure 8. (a) The end-effector position expressed in the arm base frame with the limits imposed by the virtual walls (red). (b) The end-effector position and orientation error and the measure of manipulability with the corresponding minimum threshold. All the set-based tasks stay within their limits, while the position and orientation errors reach a null value.

been extended to handle set-based tasks [9], [10], e.g., arm joint mechanical limits or obstacle avoidance, where the control objective is to keep the task value above a lower threshold or below an upper threshold. To effectively and safely operate the system, it is useful to divide all the tasks into three groups and exploit this classification for assigning priority levels:

1) *safety tasks*, such as ROV autoaltitude, mechanical joint limits, and obstacle avoidance, that assure the integrity of the system and of the environment in which it operates
2) *operational tasks*, which include all of the tasks commanded by the user, such as ROV guidance from point A to B, end-effector position, or configuration
3) *optimization tasks*, which include all those tasks that are not strictly necessary for the actual accomplishment of the operation but that help to do it in a more efficient way, e.g., arm manipulability.

The hierarchy can be changed in terms of the number of tasks and their order of priority as a function of the action that needs to be performed.

Experimental Validation

The developed control framework was validated and tested during the 2017 field trials. In the following, the results of an experiment in which only the manipulator is controlled with the proposed TPIK algorithm are shown. Then, a simulation showing coordinated control of the vehicle and the arm is described.

Regarding the field experiment, the chosen task hierarchy is composed of two tasks: arm joint limit avoidance and end-effector position. The end effector is commanded to follow a simple circular trajectory, while the joint-three upper and joint-five lower thresholds have been chosen to become active during the motion of the end effector and thus test the priority mechanism. Figure 7 shows the position error and the joint values with the corresponding thresholds. The effectiveness of the control algorithm is clear, as the end effector follows the desired trajectory and the joint values never exceed the desired upper and lower thresholds.

A simulation experiment further illustrates the results of coordinated control. The task hierarchy is as follows:

- *Arm manipulability.* To avoid singular configurations, a minimum value of 0.029 is set for the measure of the arm's manipulability.
- *Virtual box. A* set of six virtual walls surrounding the arm base frame assures that the arm never tries to move the end effector outside its workspace.
- *End-effector configuration. A* constant set-point for the position and the orientation of the arm has been set.

Figure 8 shows the results. The system's initial position is set far away from the desired waypoint; it moves both the vehicle and the arm to reach this waypoint with a null error, while the arm manipulability and the end-effector position expressed in the arm base frame never exceed the desired thresholds.

With regard to vehicle-related tasks, the proposed solution [11] builds on standard techniques leading to a proportional-integral controller including an anti-wind-up mechanism.

Underwater Perception for Manipulation

Camera Calibration

For manipulation—with both teleoperation and autonomy—it is essential that the sensor system is well calibrated to correctly capture the environment. We use a novel calibration and refraction correction process for underwater cameras with flat-pane interfaces that is easy and convenient to use while providing very accurate rectification results [12].

The correction is derived from an analysis of the axial camera model for underwater cameras, which is physically correct but computationally hard to tackle. It can be shown how realistic constraints on the distance of the camera to the window can be exploited, leading to an approach known as the *pinax model* because it combines aspects of a virtual pinhole model with the projection function from the axial camera model. The pinax model is not only convenient because it allows in-air calibration; it also outperforms standard methods in terms of accuracy [12].

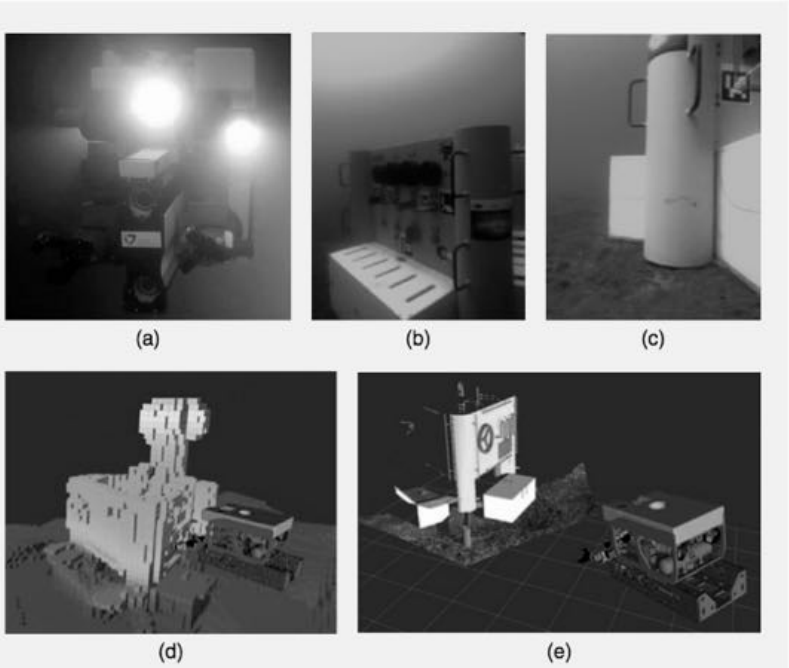


Figure 9. When (a) the vehicle approaches (b) the mockup panel structure, (c) the augmented reality marker can be used to aid the navigation to validate different navigation methods. Among others, (d) a 3-D octomap is generated in real time, which is transmitted to the offshore control center. (e) Because the mockup panel structure is—as in oil and gas operations—a priori known, perceived parts can be used to determine its pose and project the known model in the scene to aid the execution of tasks. (Photos and screenshot courtesy of DexROV consortium.)

3-D Mapping and Object Recognition

The data from the camera system with its stereo setup are processed online to generate dense 2.5-D point-clouds, which are integrated into 3-D maps (Figure 9). The well-known octree data structure is used for this purpose. More precisely, our implementation builds upon the popular OctoMap library [13], which is extended for differential update operations to support efficient, low-latency transmission of the 3-D representation over the satellite link to the onshore command center [14]. Furthermore, an efficient strategy for underwater color updates is added.

Underwater images suffer from challenging light conditions, especially wavelength-dependent attenuation and forward scattering and backscattering. When coloring the octomap, a very simple but efficient strategy is used: because attenuation is wavelength and distance dependent, the brightest measurement is used, which corresponds to the closest and, hence, most accurate sample [14]. For substantial image enhancement—at much higher computational cost—a new variant of the dark channel prior for underwater vision is used [15]. Even though there are known adaptations of this method to underwater applications, further improvements are made by reformulating the problem. Bright regions in the dark channel appear in outdoor images in air on nonsky regions because of the backscattering, which is used in the original dark channel prior. In contrast to other adaptations of the method to underwater applications, the estimation of the atmospheric light is adjusted, which leads to clear improvements [15].

The core navigation of the vehicle is based on the data of the associated sensors connected to the vision compute bottle, i.e., the NavQuest DVL and the Xsens IMU. These data are fused in an extended Kalman filter and can be aided by the registration of the stereo scans and the tracking of objects up to the level of full simultaneous localization and mapping. The mockup panel structure is equipped with augmented reality markers (Figure 9), which can further aid the navigation and be used for validating the other navigation methods. The lower-level image processing, i.e., rectification and image enhancement, are important for the robust recognition of the markers.

The 3-D octomap can be run with a 2-cm grid cell resolution. This is well suited for autonomous operations like collision avoidance and path planning, and it is also sufficient to give human operators in the onshore command center an overview of the environment including unexpected obstacles and the terrain. To aid manipulation, we exploit the fact that, in many application cases, e.g., for oil and gas operations, the structures to be dealt with are known. The detailed simulation framework employed for component and system validation [16] can be used in this context to provide a virtual visualization of the real underwater operations (Figure 9), i.e., when the vision system of the ROV detects or tracks parts of the structure of interest, the structure's pose is transmitted to the onshore command center and the known model can be projected into the scene. Furthermore, dedicated vision processes can detect and localize crucial objects as a basis for (semi) autonomous manipulation. For example, the valves on the mockup structure are detected and localized with an active contours method having superellipse fitting. Their

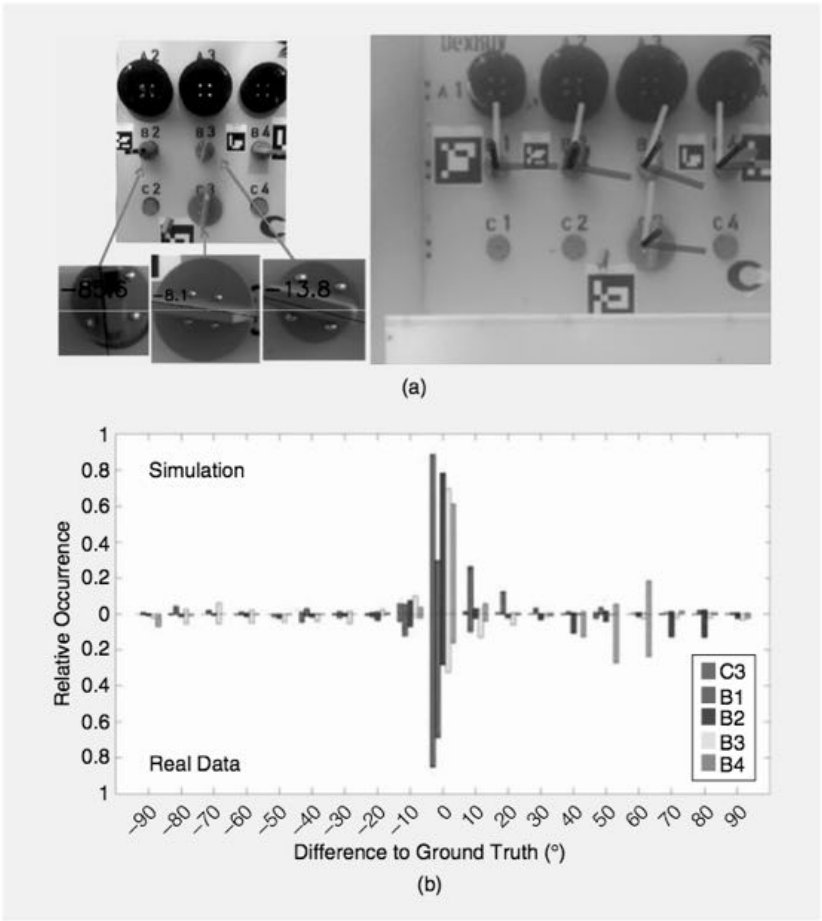


Figure 10. (a) As the basis for (semi)autonomous manipulation, the orientations of the valves on the mockup structure are determined by an active contours method using superellipse fitting in combination with a Hough transform. (b) The errors in a single frame state detection are within a few degrees, and they behave very similarly in the sea trials (real data) as in the high-fidelity simulation of the system (simulation).

orientations are determined with a Hough transform to estimate the predominate edges within the fitted ellipses (Figure 10).

Conclusions

An approach to underwater manipulation was presented that facilitates the use of a distant onshore control center with an exoskeleton based on the following:

1) efficient transmission of multiple data streams over a satellite link
2) a CE to mitigate communication latencies by encoding statistical models of manipulation tasks
3) the vehicle control, which is more oriented toward AUV than ROV operations
4) an intelligent vision system, which provides perception capabilities.

The approach was tested in July 2017 in a first field campaign over two weeks in the Mediterranean Sea near Marseille. Seven extended dives with about 11 h of experimental data were performed, wherein the ROV interacted with a mockup panel structure to validate the different system components and their interplay.

The main lessons learned from the field trials are as follows:

1) ROV operation from an onshore control center via a satellite link is feasible, in principle, despite latencies and low bandwidth.
2) It is, however, important that the system is capable of detecting the operator's intentions and using this information to (semi)autonomously carry out tasks.
3) There is no black or white with respect to autonomy, but there are different levels that can be useful, or even necessary, depending on the communication conditions.
4) Good situational awareness through constant update of the onshore environment model is important.

Acknowledgments

The presented research was carried out in the project Effective Dexterous ROV Operations in Presence of Communications Latencies, which is funded by the European Commission's Horizon 2020 Framework Program for Research and Innovation, under the topic "Blue Growth: Unlocking the Potential of Seas and Oceans" and BG6-2014 "Delivering the Subsea Technologies for New Services at Sea."

References

[1] P. Letier, E. Motard, and J. P. Verschueren, "Exostation: Haptic exoskeleton-based control station," in *Proc. 2010 IEEE Int. Conf. Robotics and Automation*, 2010, pp. 1840–1845.

[2] G. D. Cubber, D. Doroftei, Y. Baudoin, D. Serrano, K. Chintamani, R. Sabino, and S. Ourevitch, "ICARUS: An EU-FP7 project providing unmanned search and rescue tools," in *IROS Workshop on Robots and Sensors Integration in Future Rescue Information System (ROSIN'12)*, 2012, pp. 1–5.

[3] M. Karaliopoulos, R. Tafazolli, and B. G. Evans, "Providing differentiated service to tcp flows over bandwidth on demand geostationary satellite networks," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 2, pp. 333–347, Feb. 2004.

[4] A. K. Tanwani and S. Calinon, "A generative model for intention recognition and manipulation assistance in teleoperation," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2017, pp. 43–50.

[5] A. Tanwani and S. Calinon, "Learning robot manipulation tasks with task-parameterized semitied hidden semi-Markov model," *IEEE Robot. Autom. Lett.*, vol. 1, no. 1, pp. 235–242, 2016.

[6] S. Calinon, "A tutorial on task-parameterized movement learning and retrieval," *Intell. Serv. Robot.*, vol. 9, no. 1, pp. 1–29, Jan. 2016.

[7] E. Simetti, G. Casalino, S. Torelli, A. Sperindé, and A. Turetta, "Floating underwater manipulation: Developed control methodology and experimental validation within the TRIDENT project," *J. Field Robot.*, vol. 31, no. 3, pp. 364–385, 2014.
[8] E. Simetti, F. Wanderlingh, S. Torelli, M. Bibuli, A. Odetti, G. Bruzzone, D. Rizzini, J. Aleotti, G. Palli, L. Moriello, and U. Scarcia, "Autonomous underwater intervention: Experimental results of the MARIS project," *IEEE J. Ocean. Eng.*, vol. 43, no. 3, pp. 620–639, 2018.
[9] S. Moe, G. Antonelli, A. R. Teel, K. Y. Pettersen, and J. Schrimpf, "Set-based tasks within the singularity-robust multiple task-priority inverse kinematics framework: General formulation, stability analysis and experimental results," *Front. Robot. AI*, vol. 3, pp. 16, Apr. 2016.
[10] F. Arrichiello, P. D. Lillo, D. D. Vito, G. Antonelli, and S. Chiaverini, "Assistive robot operated via P300-based brain computer interface," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2017, pp. 6032–6037.
[11] D. D. Palma and G. Indiveri, "Underwater vehicle guidance control design within the DexROV project: Preliminary results," *IFAC-PapersOnLine*, vol. 49, no. 23, pp. 265–272, 2016.
[12] T. Luczynski, M. Pfingsthorn, and A. Birk, "The pinax-model for accurate and efficient refraction correction of underwater cameras in flat-pane housings," *Ocean Eng.*, vol. 133, pp. 9–22, Mar. 2017.
[13] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Auton. Robots*, vol. 34, no. 3, pp. 189–206, 2013.
[14] T. Luczynski, T. Fromm, S. Govindaraj, C. A. Mueller, and A. Birk, "3D grid map transmission for underwater mapping and visualization under bandwidth constraints," in *Proc. IEEE Oceans*, 2017, pp. 1–6.
[15] T. Luczynski and A. Birk, "Underwater image haze removal with an underwater-ready dark channel prior," in *Proc. IEEE Oceans*, 2017, pp. 1–6.
[16] T. Fromm, C. A. Mueller, M. Pfingsthorn, A. Birk, and P. D. Lillo, "Efficient continuous system integration and validation for deep-sea robotics applications," in *Proc. IEEE Oceans*, 2017, pp. 1–6.

Andreas Birk, Jacobs University Bremen, Germany. E-mail: a.birk@jacobs-university.de.

Tobias Doernbach, Jacobs University Bremen, Germany. E-mail: t.doernbach@jacobs-university.de.

Christian Atanas Müller, Jacobs University Bremen, Germany. E-mail: chr.mueller@jacobs-university.de.

Tomasz Luczynski, Jacobs University Bremen, Germany. E-mail: t.luczynski@jacobs-university.de.

Arturo Gomez Chavez, Jacobs University Bremen, Germany. E-mail: a.gomezchavez@jacobs-university.de.

Daniel Köhntopp, Jacobs University Bremen, Germany. E-mail: a.birk@jacobs-university.de.

Andras Kupcsik, Idiap Research Institute, Martigny, Switzerland. E-mail: andras.kupcsik@idiap.ch.

Sylvain Calinon, Idiap Research Institute, Martigny, Switzerland. E-mail: sylvain.calinon@idiap.ch.

Ajay K. Tanwani, Idiap Research Institute, Martigny, Switzerland. E-mail: ajay.tanwani@idiap.ch.

Gianluca Antonelli, Interuniversity Center of Integrated Systems for the Marine Environment, Genoa, Italy. E-mail: antonelli@unicas.it.

Paolo di Lillo, Interuniversity Center of Integrated Systems for the Marine Environment, Genoa, Italy. E-mail: pa.dilillo@unicas.it.

Enrico Simetti, Interuniversity Center of Integrated Systems for the Marine Environment, Genoa, Italy. E-mail: enrico.simetti@unige.it.

Giuseppe Casalino, Interuniversity Center of Integrated Systems for the Marine Environment, Genoa, Italy. E-mail: pino@dist.unige.it.

Giovanni Indiveri, Interuniversity Center of Integrated Systems for the Marine Environment, Genoa, Italy. E-mail: giovanni.indiveri@unisalento.it.

Luigi Ostuni, Interuniversity Center of Integrated Systems for the Marine Environment, Genoa, Italy. E-mail: antonelli@unicas.it.

Alessio Turetta, Graal Tech, Genoa, Italy. E-mail: alessio.turetta@graaltech.it.

Andrea Caffaz, Graal Tech, Genoa, Italy. E-mail: alessio.turetta@graaltech.it.

Peter Weiss, Compagnie Maritime d'Expertises, Marseille, France. E-mail: p.weiss@comex.fr.

Thibaud Gobert, Compagnie Maritime d'Expertises, Marseille, France. E-mail: t.gobert@comex.fr.

Bertrand Chemisky, Compagnie Maritime d'Expertises, Marseille, France. E-mail: p.weiss@comex.fr.

Jeremi Gancet, Space Applications Services, Brussels, Belgium. E-mail: jeremi.gancet@spaceapplications.com.

Torsten Siedel, Space Applications Services, Brussels, Belgium. E-mail: torsten.siedel@spaceapplications.com.

Shashank Govindaraj, Space Applications Services, Brussels, Belgium. E-mail: shashank.govindaraj@spaceapplications.com.

Xavier Martinez, Space Applications Services, Brussels, Belgium. E-mail: xavier.martinez@spaceapplications.com.

Pierre Letier, Space Applications Services, Brussels, Belgium. E-mail: jeremi.gancet@spaceapplications.com.