Advanced ROV Autonomy for Efficient Remote Control in the DexROV Project

Paolo Augusto Di Lillo, Enrico Simetti, Daniela De Palma, Elisabetta Cataldi,
Giovanni Indiveri, Gianluca Antonelli, Giuseppe Casalino

Abstract: **In this paper, we present DexROV, a funded EC Horizon 2020 project that proposes to implement novel operation strategies for underwater semi-autonomous interventions. These costly and demanding operations are more and more often performed by ROVs (Remotely Operated Vehicles), contributing to risks cutting for human divers. However ROV operations require offshore structures, hosted on a support vessel with a crew of a significant amount of personnel necessary to properly handle and operate the robotic platform. One of the key goals of DexROV is to delocalize on-shore the manned support as much as possible, reducing the crew onboard the support vessel and consequently the whole operation costs and risks. The Control Center is located onshore, far away from the actual operation location. Operators interact with the ROV through a simulation environment that exploit 3D models of the environment built online relying on the perception and modeling capabilities of the robotic system and transmitted via satellite communication. Currently ROVs lack the dexterous capabilities needed to perform many kind of operations, for which human divers are still necessary. DexROV addresses this problem, equipping the ROV with two 6 DoF (Degrees of Freedom) dexterous manipulators with anthropomorphic end-effectors and providing semi-autonomous capabilities. The control will rely on a multi-task priority approach that will help the operator to focus on the main operation, leaving the low-level tasks to be autonomously performed by the ROV.**

1 Introduction

DexRov is a funded EC Horizon 2020 project started in March 2015 that focuses on the development of a system capable to perform dexterous underwater operations, relying on a novel paradigm that provides the possibility to enhance the methodology in which these kind of interventions are currently performed. The consortium consists of 9 European organizations coordinated by the Belgian company Space Application Services. Academic partners include the Italian interuniversity center ISME (Integrated Systems for Marine Environment), with the Universities of Genova, Cassino and Salento, the German Jacobs University, and the Swiss IDIAP research laboratory. Industrial partners include COMEX (France), GRAAL TECH (Italy) and EJR Quartz (Netherlands). The project proposes to improve the efficiency of subsea interventions, while reducing both risks and costs by changing some key features that highly characterize them. In particular, DexROV focuses on:

- Enabling far distance teleoperation of a ROV, properly handling the introduced communication latencies
- Providing advanced dexterous manipulation capabilities to the ROV
- Providing semi-autonomous navigation and manipulation capabilities to the ROV

2 Challenges

Performing underwater interventions is a demanding activity that requires perception, situation awareness and the capability to dynamically adapt the operation to the surrounding harsh environment. Professional divers are often requested to carry out operations that require high dexterity and these kind of missions are very expensive and risky. In addition, the depth at which divers can work at is limited to a maximum of 400 to 500 meters. For this reason, ROVs are more and more often preferred to human divers, in order to perform safer intervention and to reach bigger depths. However, ROV-based operations are very costly, especially because they require an offshore support vessel and a big crew in order to properly operate the vehicle. Enabling the possibility to partially locate the crew in an onshore control center would significantly decrease the logistic costs of the operation, allowing the use of a smaller and cheaper support vessel and limiting the number of people to the strictly necessary, essentially a couple of operators for deploying and recovering duties. This is exactly one of the main challenges of DexROV. The Control Center is moved onshore, far away from the actual operation location, and pilots send commands and instruction to the ROV via a satellite communication. The satellite channel is characterized by a significant latency and intermittence that makes impossible for the operator to directly control the vehicle. The latency mitigation strategy consists of two main elements: a simulation environment and a cognitive engine. The operator interacts with the ROV through the simulation environment without taking into account the communication latency. This simulation environment replicates the actual location in which the ROV is operating by receiving 3D data from a perception system, and the operator interacts with it using a VR (Virtual Reality) system and a force-feedback exoskeleton relying on an accurate physics simulation. The operator perform actions in order to instruct a cognitive engine system that translate the operator's movement in motion and manipulation primitives that will be transmitted to the vehicle.

One of the biggest issues in a ROV based operation is the range of tasks that the vehicle can perform. So far, operations that require highly dexterous capabilities (such as manipulating fragile biological and archaeological samples on the seabed) can be only achieved by human divers. DexROV proposes to extend the range of the achievable tasks for a ROV, equipping it with dexterous 6 DoF manipulators with anthropomorphic end-effectors, capable of using the standard tools designed for divers. Another main aspect of the project is to give semi-autonomous capabilities to the ROV. Piloting ROVs is indeed a demanding task that usually requires three well-trained operators working simultaneously and a remarkable amount of time to be properly performed. Giving the ROV the capability to autonomously perform some kind of actions would significantly reduce the operators' effort and the time needed to achieve the tasks.

3 *Functional Architecture*

DexROV setup is split in two different locations:

- On the offshore side, there are a support vessel and a ROV. The vessel is equipped with a satellite communication link; the ROV is different from a standard one because it is enhanced with advanced autonomous navigation and manipulation capabilities and it is equipped with a number of sensors and two dexterous 6 DoF manipulators.
- On the onshore side, there is a Control Center with all the needed facilities to allow remote human supervision and control, both for navigation and manipulation tasks. Concerning dexterous manipulation tasks, the operator instruct a cognitive engine exploiting a force-feedback exoskeleton in a 3D simulation environment.

Figure 1 illustrates DexROV functional architecture. DexROV will develop a simulation environment to help the operator in the interaction with the objects in the environment surrounding the ROV, relying on 3D models built online by its perception system. The operator instructs a cognitive engine by demonstrating how actions should be done using a force-feedback exoskeleton. These instructions are translated into manipulation and navigation primitives, transmitted via satellite communication and finally autonomously performed by the actual ROV, independent of communication latencies. The ROV is equipped with a pair of force sensing capable manipulators and dexterous end-effectors integrated in an ad-hoc designed modular skid able to fit in a standard mid-size ROV.

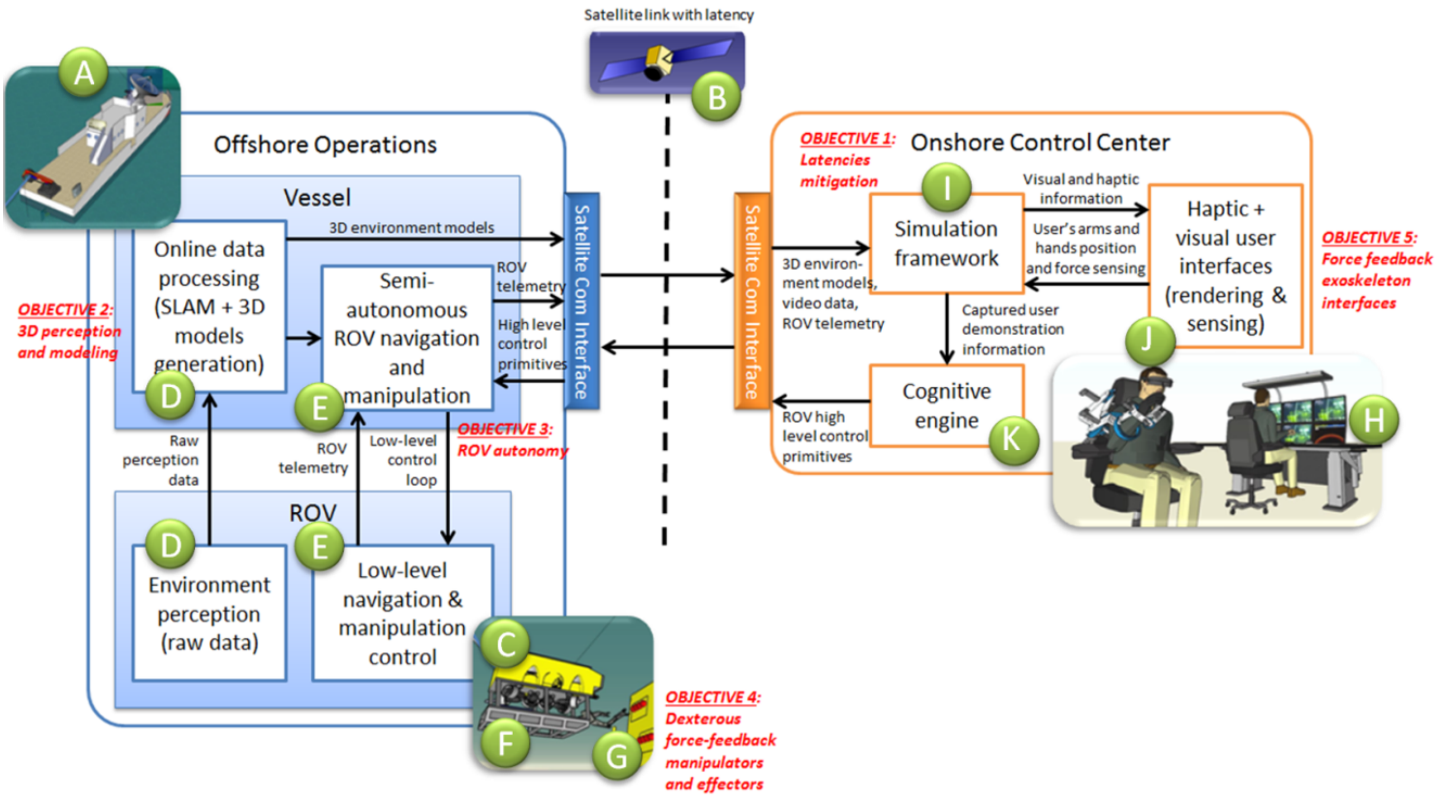


Figure 1: DexROV functional architecture

3.1 *Underwater perception and mapping*

Underwater environment is very harsh in terms of perception: visibility is never perfect, sometimes even non-existing. For this reason, the most exploited kind of sensor is the acoustic one. Sonar sensors have clear issues though, particularly in terms of frequency and noise levels with respect to land sensors such as laser range finders. Nevertheless, in many marine applications detailed 2D or 3D representations of the environment would be very desirable. As far, this work is mostly concerned in offline reconstruction of complex and accurate 3D models after the mission (Fairfield, Kantor, Wettergreen, 2007; Sedlazeck, Koeser, Koch, 2009; Saez, Hogue, Escolano, Jenkin, 2006).

DexROV proposes an approach and techniques that aim to reliably process underwater data online (Buelow, Birk, 2013; Pathak, Pfingsthorn, Buelow, Birk, 2013; Pathak, Birk, Vaskevicius, Poppinga, 2010; Buelow, Birk, 2011; Pfingsthorn, Buelow, Sokolovsky, 2013), that will be transmitted to the onshore control center via satellite communication and then will be integrated in the simulation environment. This operation starts with the acquisition of underwater data from a stereo camera. A major contribution will be the online estimation of stereo disparities under adverse and noisy conditions. Another issue that needs to be addressed is the SLAM (Simultaneous Localization and Mapping) which is an aspect of major importance also for assuring reliable autonomous capabilities to the ROV. The main challenge in the context of DexROV is to provide the pose estimation in a fast a robust way suited for online processing (Pfingsthorn, Birk, Buelow, 2012). For this reason the system will be equipped with a number of sensors, including a AHRS (Atiitude and Heading Reference System), a DVL (Doppler Velocity Log), a USBL (Ultra-Short Baseline) and a stereo-camera that will be used simultaneously in order to be capable to give an accurate pose estimation of the vehicle.

3.2 Cognitive engine

The satellite channel always introduces a significant latency in the communication due to the satellite-to-earth propagation delay, which is estimated in 500 milliseconds. This is even more emphasized in offshore marine environment due to additional constraint such as the processing delay within the network infrastructure. Overall, the latency of this kind of network is in the range of 900-1150 milliseconds. This could easily decrease the quality of the data stream and the actual bandwidth of the control signal to be transmitted. For this reason, commands from the onshore control center have to be represented in a robust manner, in order to be adapted to the quality of the communication.

In DexROV, the cognitive engine is duplicated, working both on the onshore and the offshore side: on the onshore side, it detects the actions that the operator wants to perform learning by demonstrations; on the offshore side, the use of probabilistic models allows it to locally anticipate which primitive to adopt until a new information from the control center is available. This approach allows creating a telemanipulation system that is robust to nonhomogeneous communication channel, independent from the introduced latency. It is proposed to exploit a recently developed task-parameterized mixture model which has proven to be robust for a various types of dynamic generalization requirements (Calinon et al, 2012; Rozo et al, 2013; Alizadeh, Calinon, Caldwell, 2014). The approach allows a GMM (Gaussian Mixture Model) to be adapted to different situations that are not part of the training set. The motion is composed by a sequence of reference systems, or task parameters that represent the locations of intermediate waypoints that the end-effector is required to reach in order to accomplish a certain task. These trajectories are learned by demonstration from the user, observing his movements from different coordinate systems and training a separated model in each of them. These models are then merged into one, in order to reproduce a generalized version of the movement (Calinon, Alizadeh, Caldwell, 2013). This aspect has a major importance because the movement associated with a certain task has to be adaptive to the dynamic environment in which the robot will be operating. This approach is robust also in case of missing task parameters, e.g. when the visibility is reduced. This is achieved by building the model and retrieving the output trajectory only on the available task parameters at each time step (Calinon, Bruno, Caldwell, 2014);

3.3 Underwater dexterous manipulators

One of the challenges in DexROV is to increase the range of tasks that can be executed by the ROV, in order to better replace human divers in operations that require high dexterity. Bringing underwater the capabilities of a manipulator for ground application is a major challenge, due to the specific design criteria that have to be followed in order to make the arm properly work under high pressure.

Usually, underwater arms have a limited number of DoF (usually 4 or 5), and a 2-jaws grippers as end-effector, with a single DoF. In DexROV the following characteristics are essential:

- A 6 DoF anthropomorphic arm, allowing the end-effector to be accurately oriented
- A 3 Dof end-effector, capable to manipulate tools designed for human divers
- An accurate sensory system providing force feedback and joints position information

As far, there is no underwater manipulator that gathers all these requirements. One of the most popular manipulators for heavy underwater tasks is the TITAN4 from Shilling Robotics. It is a dexterous 6 DoF arm with a remarkable maximum payload, but it is much larger than a human arm making it unsuitable for manipulating standard divers tools. Usually it is used with work class ROV for heavy duty interventions and would not fit in a medium-class ROV like the APACHE 2500, which is the one used in DexROV.

Following this analysis, an innovative hydraulic dexterous arm and end-effector solution will be developed. The key feature will be the high dexterity of the kinematic structure. The hand will have 3 fingers, making it capable to effectively use standard tools, and the arm will have 6 DoF, in order to accurately orient the end-effector. A particular effort will be dedicated to designing the arm with a novel approach, as it will be hydraulic-driven and modular. This will combine the strength and the high payload of hydraulic actuators with the versatility and the compactness of standard electric actuators, making the manipulator suitable to perform both heavy and precision tasks. The integration of accurate force/torque sensors in the fingers and in the wrist will allow the development of advanced force control algorithms, making DexROV system a unique underwater dexterous manipulation solution.

3.4 Force feedback exoskeleton

On the onshore control center side, the operator will send manipulation commands to the system by instructive the cognitive engine relying on a force feedback exoskeleton operating locally on the 3D simulation environment, despite the presence of communication latencies. The force-feedback exoskeleton to be used in DexROV will be based on the one initially designed for ESA (Letrier, Motard, Verschueren, 2010; Letier et al, 2011), and further improved for the FP7 ICARUS. We will consider a system based on the association of a soft supporting structure in the shape of a wearable exoskeleton glove and tendon cables to reduce volume and mass of the device around the hand. HyunKi (2011) proposed a single finger prototype of jointless device with pulling tendons inserted in a glove. This design will be enhanced with rigid elements for better stiffness and controllability. Actuators will be placed in the lower part of the exoskeleton, offering a comfortable solution and preserving high quality haptic feedback.

4 Semi-autonomous ROV navigation and manipulation control

4.1 DexROV missions

The DexROV system will be designed to be suitable for different kind of operations. Five main use case scenarios have been detected:

Offshore Oil&Gas industry: during subsea oil operations, ROVs perform routine inspection, manipulation of valves, plugging and unplugging of electrical and hydraulic jumpers. A subsea facility, after drill and during nominal operations, is composed by: wellheads which provide a suspension point and pressure seals for the casing strings that run into the well; Christmas trees, which monitor and control the production of the well through multiple valves and chokes organized in panels on which the ROVs operate; manifold, that receives the channeled product from the wells and sends it to a host platform. This is a very structured environment, in which the robotic system has to be able to navigate to the panel location and manipulate handles, rotate valves, plug and unplug connectors. Since these operations require a significant strength, it is necessary to stabilize the ROV against undesired rotations and translations while manipulating the objects through clamping it to a support structure close to the panels.

Oil&Gas NDT (Non Destructive Testing): during general interventions in marine infrastructure, NDT operations need to be performed. These techniques consist in performing a continuous and precise scan over the surface with a particular sensor, with specific forces applied, in order to not damaging the probe. The presence of a crack in the surface disturbs the electromagnetic field and the operator is alerted to the presence of a defect. Currently these operations need the intervention of a diver, due to the high dexterity needed. DexROV proposes to make the ROV capable to perform them autonomously, exploiting advanced force control algorithms.

Renewable energy: offshore renewable energies is a huge potential market ramping up quickly, that will require advanced and effective capabilities to support the installation and maintenance of facilities. There are essentially two sorts of offshore renewable energy infrastructures: wind farms and tidal energy generators. Offshore floating wind turbines generally consist of a floating foundation moored with catenaries or tension legs to the seabed. The catenaries and the mooring require inspection and occasional maintenance.

Marine science: ROVs are important assets in three areas of science: biology, geology and archaeology. Biologists are often interested in collecting samples like corals, sponges and rocks from the sea bottom in order to study their composition and classify them. Of course these samples can not be damaged and the ROV has to be highly dexterous and precise during their manipulation and collection.

Underwater archaeology: underwater archaeology is a topic with great relevance to European objectives on the protection of cultural heritage, and close synergies with DexROV objectives. It is expected that due to the fragile nature of archaeological artifacts and sites, this use case will provide requirements that are more stringent than in other domains, increasing the versatility and range of capabilities of the solution to be developed. The use of ROVs makes archaeology accessible to great depth and improves working conditions in shallower water. ROVs have had been used in a number of underwater archaeological expeditions, teleoperated by an operator onboard a research vessel. Standard missions range from exploring and mapping wrecks, to manipulating and excavating artifacts.

In order to be effective in performing all these different kind of operations, the DexROV system will be designed to have two different setups, easily interchangeable within a modular skid. The first setup will show a single 6 DoF anthropomorphic manipulator and a clamp. This is aimed for operations that require more strength and that can be performed in structured environments, such as the Oil&Gas related ones, with the presence of a support structure to clamp at. In this configuration, the ROV navigates autonomously reaching the desired location, clamps to the structure close to the panel and perform the desired manipulation tasks being base-fixed. The second setup will show two 6 DoF manipulators and it is aimed for operations that require more dexterity, such as collecting and cleaning biological and geological samples. In this configuration the robotic platform will perform the manipulation tasks in free-floating due to the absence of an ad-hoc structure to clamp at. This kind of control requires a coordinated control of both the vehicle and the arm. Since the system has to be capable of performing such different missions, it is necessary to design a proper control framework that changes its objectives with respect to the distinctive features that the current operation need to achieve.

4.2 Multi-task priority introduction

Complex robotic systems are often asked to perform different kind of tasks simultaneously. Considering an anthropomorphic manipulator, it could be desirable to assign a given end-effector position and orientation while keeping its joint positions below a certain threshold, far away from their mechanical limits, or while maximizing the structure manipulability. For a mobile robot, a classical multi-task example is to make it move to a target waypoint while avoiding obstacles that it could encounter along the path. Usually a complex mission is split into several sub-problems, called *behaviors*, which need to be fulfilled simultaneously if possible.

Let us consider a generic *m*-dimensional task. It can be written as:

$$\sigma(t) = f\big(q(t)\big),$$

where $\sigma(t) \in R^m$ is the task variable to be controlled and $q(t) \in R^n$ is the *n*-dimensional vector of the system configuration, with *n* equal to the number of its DoF. The corresponding differential relationship holds:

$$\acute{\sigma}(t) = \frac{\partial f\big(q(t)\big)}{\partial q}\acute{q}(t) = J\big(q(t)\big)\acute{q}(t),$$

Where $J\big(q(t)\big) \in R^{m\times n}$ is the task Jacobian matrix and $\acute{q}(t) \in R^n$ is the system velocity. For sake of readability, the dependency on $q(t)$ of the Jacobian matrix is omitted in the following. The redundancy of the system can be exploited to perform multiple tasks simultaneously. A system is called kinematically redundant if it has more DoF than the dimension of a specific tasks, hence if $n > m$ (Chiaverini, Oriolo, Walker, 2008). In such case, the reference system velocity that brings the task value σ to a desired value σ_d can be computed using the CLIK (Closed Loop Inverse Kinematics) algorithm:

$$\dot{q} = J^{\dagger}(\dot{\sigma}_d + K\tilde{\sigma}),$$

where $J^{\dagger}$ is the Moore-Penrose pseudoinverse of the J matrix, K is a positive-definite matrix of gains and $\tilde{\sigma} = \sigma_d - \sigma$ is the task error.

In general is not possible to find a single motion command to the robot such that it can accomplish all the tasks at the same time. In this case the behaviors are *in conflict*, and a policy has to be adopted in order to choose how to combine the joint velocities related to the single tasks. The most common approach is to define a priority among the elementary behaviors, and to generate the reference system velocity that achieves the maximum number of tasks simultaneously, always giving more importance to the higher priority ones. Usually it is recommended to give a higher priority to safety-related tasks and lower priority to optimization-related tasks. In Figure 2 it is shown a graphical representation of possible prioritized tasks for a robotic system during a generic mission.

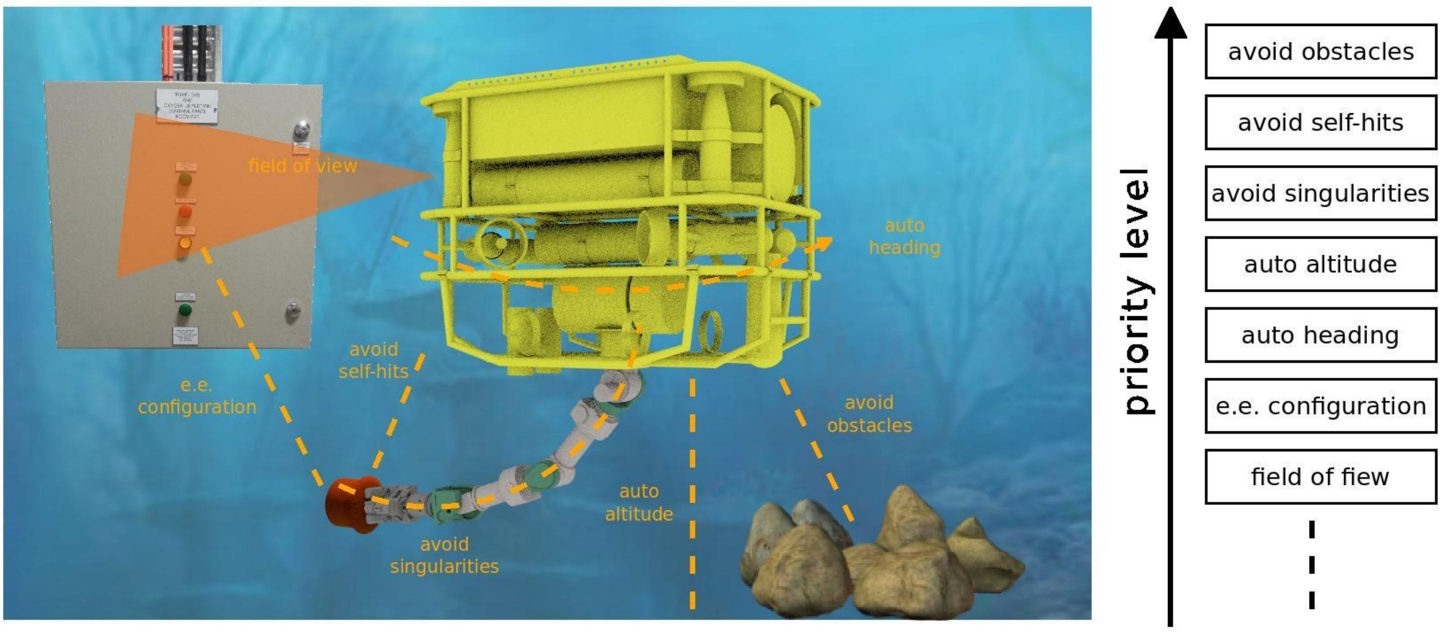


Figure 2: Graphical representation of possible tasks to be achieved

Within DexROV, the system has to be capable of satisfying different tasks and control objectives during its different missions. In the following, the possible elementary tasks for a system composed by a vehicle plus a manipulator are listed and briefly described. For further details see (Antonelli, 2014).

- **Vehicle position:** this is the “move to goal” objective, that makes the vehicle to reach a target position in 3D space

- **Vehicle hovering (Dynamic positioning):** that makes the vehicle able to maintain its position and heading
- **Vehicle roll/pitch:** in some cases is useful to have the vehicle to hold a specific attitude, with small roll and pitch angles to achieve more stability
- **Vehicle heading (Autoheading):** this is the direction where the vehicle is pointing to
- **Vehicle altitude (Autoaltitude):** for some operations is useful to keep the vehicle at a constant altitude with respect to the seabed or to an object
- **Vehicle depth (Autodepth):** in some cases it is useful to keep the vehicle at a constant depth
- **Vehicle obstacle avoidance:** for security purposes it is always recommended to make the vehicle capable to avoid potential obstacles that could encounter during the movement
- **End-effector configuration:** makes the end-effector of the arm to reach a desired position and orientation
- **End-effector field of view:** if a directional sensor, like a camera, is mounted on the end-effector of the arm, it is necessary to make it point in a certain direction
- **End-effector obstacle avoidance:** as for the vehicle, the end-effector has to be capable to avoid potential obstacles during manipulation tasks
- **Force regulation:** for manipulation tasks, it is usually required to control the force applied by the end-effector to the object it interacts with
- **Avoid self-hits:** it is needed to guarantee that the end-effector of the arm does not hit the vehicle structure during the movement
- **Mechanical joint-limit**: each joint of the arm is allowed to move in a range before reaching the mechanical limit of the structure. It is appropriate to keep all the joints at safety distance from their limits in order to not incur in undesirable behaviors
- **Robot manipulability:** the joint of the arm should always be arranged in a way in which it stays far away from the so-called kinematically singular configurations. In these configurations its Jacobian matrix loses rank and its inversion might becomes problematic. Usually it is desirable to set a minimum threshold of manipulability that the arm should never exceed
- **Robot nominal configuration:** in some cases is useful to assign a preferred arm posture that guarantees a certain level of manipulability

The above outlined objectives can be solved using the task priority resolution schemes proposed in (Chiaverini, 1998; Antonelli, Arrichiello, Chiaverini, 2008; Simetti et al. 2014, Simetti and Casalino 2015).

In particular, the elementary tasks relative to the vehicle-only control (Vehicle position, vehicle hovering,…,Vehicle depth), denoted in the following as DexROV "*primitives*", can be implemented using the control laws proposed in (De Palma and Indiveri, 2016). Notice that from a technological point of view, the basic motion control functionalities associated with the listed primitives are rather standard, nevertheless the problem becomes more challenging when aiming at designing a single kinematics control solution able to implement all the requested primitives within a unique and general framework. The solution proposed is based on a purely kinematics model of the ROV eventually subject to a constant, but unknown ocean current. Given the desired position p_d for the vehicle and the desired heading ψ_d a linear velocity motion controller and a yaw velocity motion controller have been proposed for the DexROV motion primitives. In particular, a proportional-integral (PI) closed loop control law aided by a feedforward term has been designed for both controllers. A complete stability analysis of the system is provided in (De Palma, Indiveri, 2016). Moreover, in order to deal with possible integrator wind-up issues due to vehicle velocity saturation, the integral gain should be set to zero if the velocity command exceed the saturation threshold. Notice that, when the desired heading is not specified, it should be chosen such that the position error vector in body frame should be oriented along the vehicle's surge axis. Indeed, although the ROV has fully actuated linear velocities, control authority over surge is higher than on the sway axis; moreover, cameras and obstacle avoidance sonars are often forward looking, i.e. in surge direction. As a result, for longer distance movement, the preferred traveling direction should be surge. In this case the control will be referred to as *auto surge heading*.

Interestingly, all the desired DexROV motion primitives (and others that can be defined by superposition) can be achieved through the very same surge and heading control laws described above by suitably defying the reference values p_d and ψ_d. In particular, the DexROV motion primitives can all be thought as variants of the "move to goal" primitive. In principle the motion control problem of going to a target can be solved with a linear velocity and an arbitrary heading. Yet, as previously discussed, the surge direction may be preferable. This is why the heading reference to be followed is either an arbitrary ψ_d or the one pointing to the target point. Of course, if the heading control with reference ψ_d is implemented without activating any linear velocity, this would correspond to an Autoheading primitive. Likewise, if the target point should be located on the vertical passing through the origin of the ROV body frame, this would correspond to an Autodepth (or Autoaltitude) primitive. It hence follows that the necessary DexROV motion control primitives can be all implemented by the surge and yaw control laws with suitably defined references and error variables, namely defining (def) or not (void) the individual components of the desired position vector $p_d = \left((p_d)_x, (p_d)_y, (p_d)_z\right)^T$ and heading ψ_d. The logic being that if a reference component is not defined (i.e. it is void), the corresponding error component in the surge and yaw control laws will be set to zero. For further details about the chosen control laws, controller gains, and the digital implementation of the controllers the reader is referred to (De Palma and Indiveri, 2016).

$(p_d)_x$	$(p_d)_y$	$(p_d)_z$	ψ_d	**Primitive**
Def	Def	def / void	void	A to B with auto surge heading
Def	def	Def	def	A to B with no auto-surge heading (includes hovering)
void	def	def / void	def / void	sway correction and no auto-surge heading
def	void	def / void	def / void	surge correction and no auto-surge heading
void	void	Def	void	autodepth / altitude
void	void	Void	def	autoheading

Table 1: Primitive example depending on the target references.

4.3 *Velocity compensation*

In the previous sections, we have outlined the main objectives that overall ROV and manipulator system should achieve. During the free floating manipulation, in general both the vehicle and the manipulator control variables can be exploited toward the achievement of a particular task. However, it is well know the fact that the vehicle's thrusters dynamics is much worse than those of the arm motors (Whitcomb and Yoerger, 1995). With this in mind, it is clear that the vehicle will usually not track the result of the kinematic inversion of the task priority hierarchy very well. Consequently, the task dynamics will be different from the desired ones.

A possible way to overcome this problem is the following one. Let us suppose to have solved the task hierarchy and to have obtained a desired velocity for both the arm and vehicle. Let us send the desired velocity to the vehicle controller as before, however let us discard the obtained arm velocity. Instead, let us solve a task hierarchy where now we only consider the arm velocity as optimization variables and the vehicle velocity is instead a given parameter. This leads to an arm control law of the type:

$$\acute{q}_m = \rho + Pv$$

where $\acute{q}_m$ is the manipulator velocity at joint level,ρ is the output of the task hierarchy optimization process and $\boldsymbol{P}$ is a matrix that relates the vehicle velocity parameter with its effects on the tasks and thus on the optimal arm velocity (see (Simetti et al., 2014) for details on the computation of matrix $\boldsymbol{P}$). The above law is thus the optimal one for the arm in correspondence of *any* vehicle velocity. Thus, given a good vehicle velocity feedback, the arm can compensate for the mismatches between the desired vehicle velocity and the actual one. The compensation can be total only if the following three requirements are satisfied, otherwise it will be only partial:

1. The arm end-effector Jacobian is full-rank, i.e. its manipulability measure is currently above some minimum value, as required by its corresponding control objective;
2. The arm joints are not hitting a mechanical limit, as required by its corresponding control objective;
3. The vehicle velocity is measured exactly.

5 *Preliminary simulative results*

5.1 Vehicle control

The guidance control laws proposed for the DexROV vehicle primitives have been validated numerically using a simplified, yet realistic, simulator of the scenario under investigation. The simulator has been developed in Matlab. It includes the following major modules: i) the purely kinematic model of the ROV (1), ii) the guidance control system, iii) the sensors feedbacks iv) the communication module including delay, and v) the graphics display. Regarding the kinematic model, a minimal disturbance on the roll and pitch rates has been considered aiming at making a more realistic simulation of the ROV attitude. A saturation of the velocity commands to the maximum surge, sway and heave velocities has also been implemented. The ocean current is assumed to be constant in NED frame and slowly varying in body frame: consequently its effect will be almost completely rejected by the integral action of the controller.

With reference to the guidance system, the control loop is closed using the measurements from USBL (for *x* and

y coordinates in NED frame), depth sensor (for z coordinate in NED frame) and AHRS (for roll, pitch and yaw angles, i.e. φ, θ, ψ). A sampling frequency of 10 Hz has been assumed for measurements from depth sensor and AHRS. The measurement uncertainty has been modelled as gaussian noise with standard deviation equal to 2 [m] and 1 [deg] for the depth sensor and AHRS, respectively, in accordance with the specifications of the currently available DexROV devices. A simplified and realistic model of USBL positioning system have been simulated having a relatively low sampling frequency (approximately 1 Hz) and a delay that is range dependent (namely two times the range divided by the velocity of the sound in water). The measurement accuracy is also range dependent: for the DexROV USBL, and in compliance with the system data sheets, it is 1% of the slant range. The simulator simultaneously displays in two views (left-view and right-view in figure 3): i) the Onshore Control Center (OCC) user interface including the ROV position as it is available to OCC through the telemetry acquired by the (possibly delayed) satellite communication link and ii) the Offshore Operations: the instructions received from the OCC operator with satellite communication delay and the actual ROV position and orientation. The inputs to the simulator are the individual components of the desired position and heading of the ROV. Based on the definition or not of the desired components, the proposed guidance control law implements the proper motion primitives. The described simulator has been used to undertake a preliminary validation of the controller performance for the DexROV motion primitives. The initial position of the ROV is fixed at (80,90,1300)[m] in the earth fixed frame. The USBL is assumed to be located at the origin of the earth fixed frame and a constant ocean current in the horizontal plane has been simulated, i.e. (0.05,0.1,0)[m/s]. Only the x and y components of the desired positions (i.e. (30, 10)[m]) are specified, the z component and heading are kept void. This leads to the activation of two primitives simultaneously: Autodepth and "move to goal" with auto surge heading.

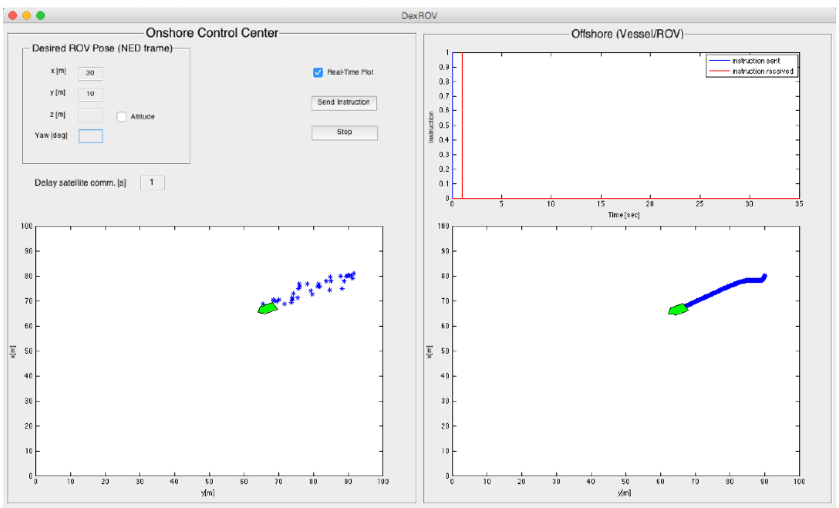


Figure 3: Screenshot of the simulator.

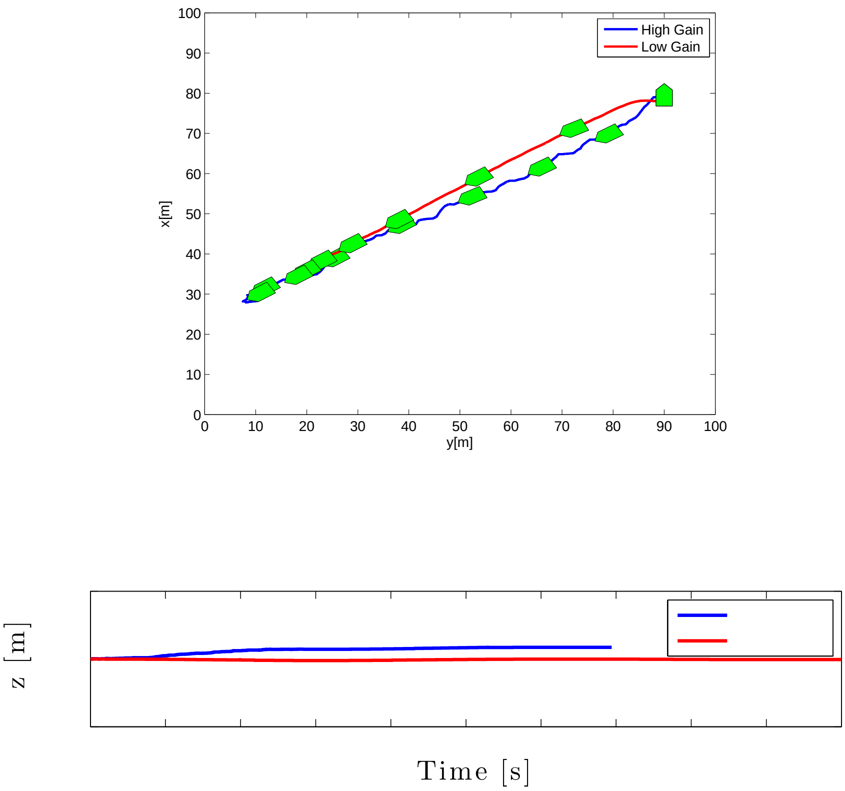
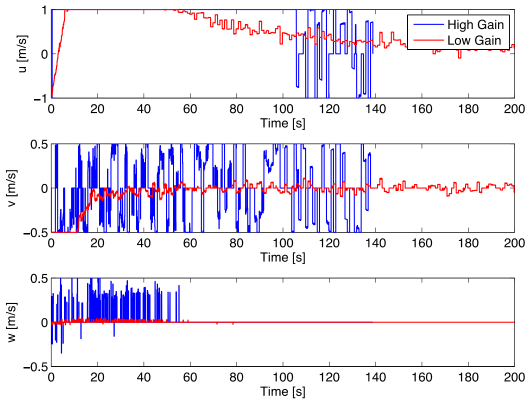
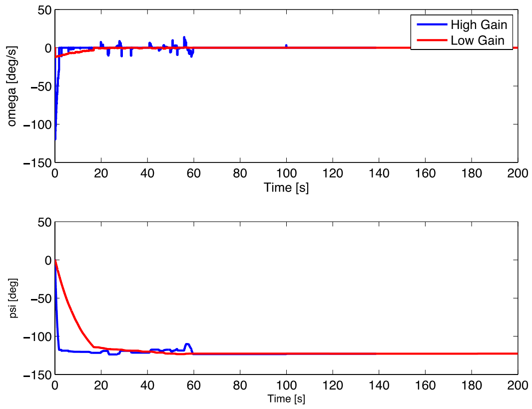


Figure 4: Trajectory of vehicle.



(a)



(b)

Figure 5: (a) Linear velocity command, (b) Angular velocity command and heading.

The figures 4 and 5 report the trajectory and the linear and angular velocity commands, respectively, related to the scenario under investigation. In this scenario two different controllers gains have been compared. In the above mentioned figures, the results related to the higher gain are depicted in blue, whereas those ones related to the lower gain are depicted in red. As expected, the proposed control solution is able to activate and execute the necessary motion primitives. Indeed, the target position is correctly reached with a travelling direction oriented along the vehicle's surge axis. At the same time, the depth is kept approximately constant to the initial value in accordance with the Autodepth primitive. It is worth highlighting that the trajectory corresponding to the higher gain appears to be nonlinear (blue line in figure 4). This phenomenon is due to the fact that the high depth leads to both: an increase of the delay in the USBL measurements acquisition and an increase of the USBL measurement errors; as noticed previously, they are both function of the range. This phenomenon can be mitigated reducing the gains of the controllers. Indeed, reducing the gains the resulting trajectory is more regular (red line in figure 4). Further investigations will focus on the definition of an adaptive gain tuning and on the inclusion of the ROV dynamic model in the simulator.

5.2 *Turn a valve & press a button*

The two operations, turn a valve and press a button, can be performed by a system composed a fully actuated vehicle with a manipulator attached below having a n DoF, in particular the simulated one has seven degree of freedom, for more details see (Cataldi and Antonelli, 2015). The control is named the interaction control, in particular this control is able to manage the interaction of the system with the external environment.

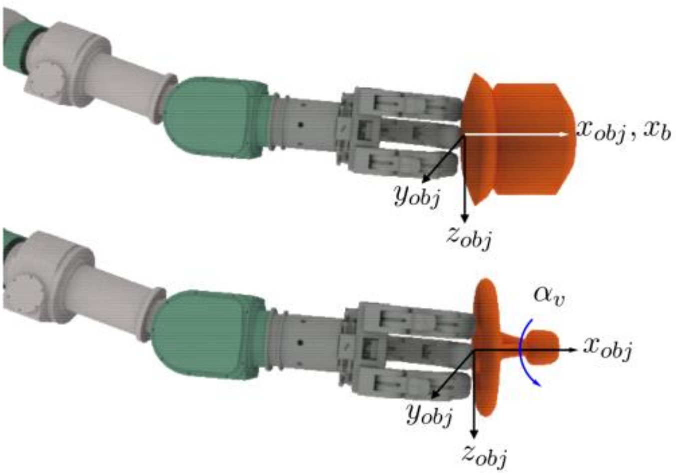


Figure 6: Top: Button and End-effector positions, the white axis is the direction of desired the force. Bottom: Valve and End-effector positions, in this case the blue arc is the direction around which to rotate

The control architecture, for both the operations, has been designed with two loop level, an internal loop and an external loop. The internal loop is able to transform the desired variables in end-effector frame to the system desired variables, e.g. vehicle position and orientation and manipulator joints, an also it is able to generate the motor inputs to follow the desired variables. The external loop, that is the control part able to manage the interaction with the environment, is able to accomplish a required operation taking into account the exchanged forces.

The internal loop, common to both operations, is composed by the kinematic and dynamic control . The kinematic control, also named inverse kinematic receives as input the desired end-effector position and orientation and the kinematic desired secondary tasks. The corresponding outputs are the desired vehicle and manipulator trajectories (Antonelli, 2014) sent to the dynamic controller. Remarkably, there are no constraints on the latter, one possibility is to control the vehicle by properly compensating the presence of the arm and controlling the arm with a basic PID at joint approach. The dynamic control, is composed by two separate controls, one for the vehicle and another for the manipulator. As it is common for the commercial manipulators, A joint-based PID-control is the only low level achievebale controller, the joints controllers receive as input the desired position and compute the driving torque necessary to move the joint. The vehicle control architecture, on the other hand, is composed by a control that compensates the manipulator presence (Antonelli, Cataldi; 2014) using an adaptive approach; starting from an initial estimates of a subset of dynamic parameters it estimates and compensates the parameters uncertainties, and thus compensates the coupling provided by the arm's presence.

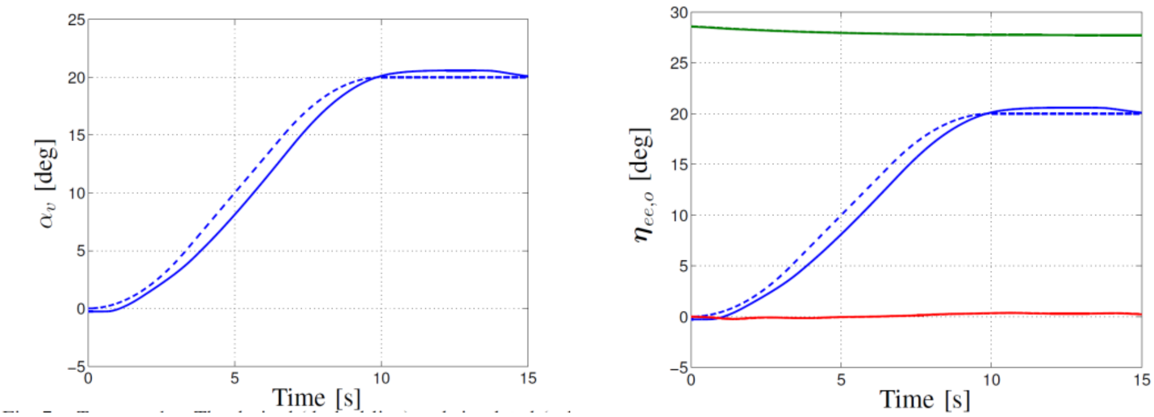
As said, the two operations are afforded by means of two different external loops, an impedance controller for the *turn a valve* operation and a force control for the *push a button*. It is known that interaction schemes may suffer from uncertainty in the estimation of the environmental geometry configuration, in such a case, in fact, the interaction arises in a direction where it was not planned for. While for the *turn a valve* operation this is not considered as a practical problem, it may be significant for explicit force control schemes. In the numerical simulations such uncertainty will be simulated. It is worth noticing that, in case needed, more sophisticated interaction control scheme may be easily adopted.

The impedance control for the *turn a valve* operation is an indirect force control, because the interaction force is not directly regulated but the control objective is rather the desired end-effector impedance. The impedance control inputs are the measured and the desired end-effector position and orientation. In this case our target is turn a valve, the end-effector is positioned perpendicular to the valve and our intention is to turn it around the x-axis, as Figure 6 shows. The desired trajectories have been determined by the desired movement on the valve.

The external loop for the *push a button* operation is composed by the force control. A Proportional Integral action is used to stabilize the force error. In fact, being the force signal characterized by a strong noise, its time derivative is usually useless. The impedance loop thus provides a damping effect.

Numerical simulations have been performed to validate the above approach. A realistic model, taking into account the most significant physical terms, has been derived. The underwater vehicle-manipulator system is composed by a full-DOF system, i.e., 6-DOF for the vehicle and 7-DOF for the arm. The dynamic parameters of the vehicle have been experimentally identified in while the arm's parameters have been extrapolated by the CAD data and simple heuristic tests.

The numerical integration of the nonlinear differential equations of the closed loop systems have been achieved by resorting to Matlab and by properly adapting its tool SimMechanics. Tuning of the parameters have been achieved with the following, pragmatic, procedure. By assuming that an existing industrial set-up is used, it is not possible and/or efficient to modify the dynamic controller of both the vehicle and the arm, this means that the inner control loop is given. Its outer control loop, the task-priority inverse-kinematics controller, is thus tuned with a bandwidth *slower* with respect to the dynamic loop, ideally, the kinematic loop should *see* as instantaneous the dynamic loop. A similar reasoning allowed to tune the outer control loop, i.e., the interaction one.



(a) (b)

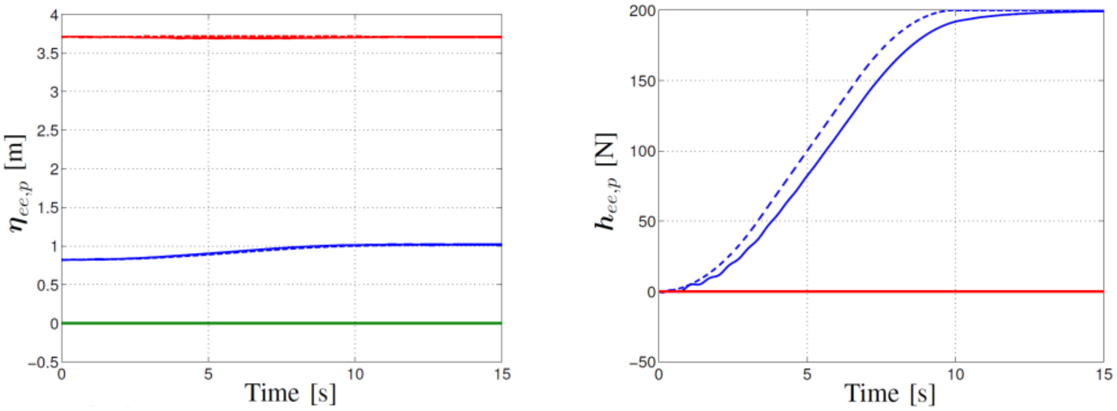
Figure 7: (a) Turn a valve: valve rotation timeline (solid line), and desired valve rotation (dashed-line). (b) Turn a valve: the simulated end-effector orientation (solid line) of the x-axis (blue), y axis (green) and z-axis (red), and their respectively desired orientation (dashed-line)

Figure 7(a) shows the valve orientation. A valve movement of 20 degrees in 10 seconds has been imposed to the system, we can appreciate that the valve follows the trajectory with a reasonable error.

The end-effector orientation (solid-line) and its desired values (dashed-line) are shown in Figure 7(b). In this figure we can appreciate that the end-effector moves only around the x-axis even if an error of the perception system has been simulated. In the other directions, the movements are smaller than the x-axis, it enforces that the valve has been modeled rigid in the y,z-axes.

Figure 8(a) shows the desired (dashed-line) and simulated forces (solid-line) on the end-effector, we can view that the system completes the operation, and as we expected at the steady-state it has a null-error.

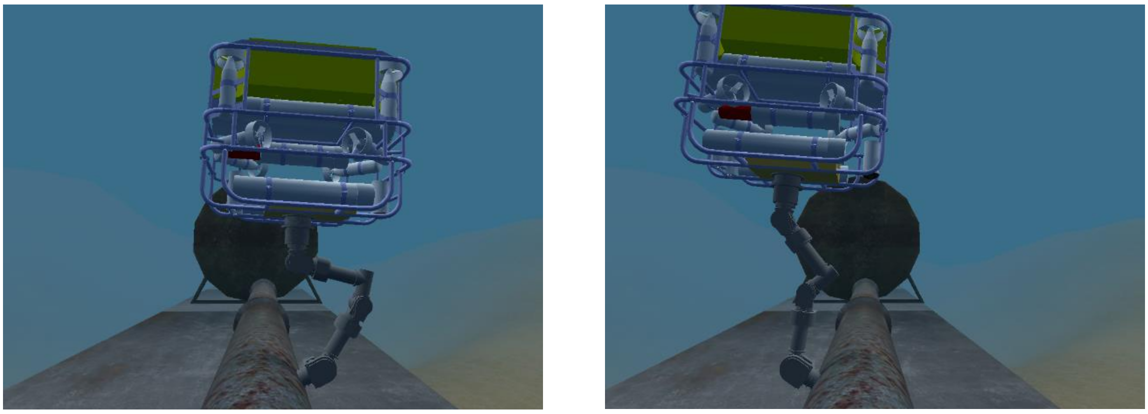
Figure 8(b) shows the desired (dashed-line) and the simulated (solid-line) end-effector position. It can be appreciated that the main movement of the end-effector is along the interaction direction. Along the plane parallel to the interaction plane, due to the intentional perception error considered, we experience a small drift that can be ignored if it is compatible with the push-button operation or handle with more sophisticated force control strategies if needed.



(a) (b)

Figure 8: (a) Push a button: The simulated end-effector position (solid-line) of the x-axis (blue), y-axis(green) and z-axis (red), and their respectively desired position (dashed-line). (b) Push a button: the simulated forces (solid-line) of the x-axis (blue), y-axis(green) and z-axis (red), and their respectively desired force (dashed-line).

5.3 Pipeline inspection



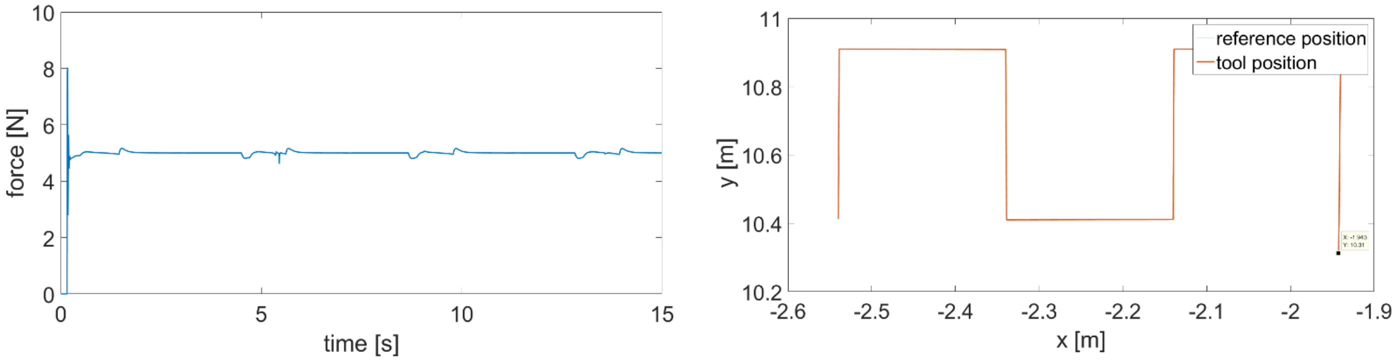
(a) (b)

Figure 9: pipeline inspection simulation (a): starting position, (b) position at the end of the first movement

In this case, we have simulated the execution of a free floating pipeline inspection task. We have considered a generalized version of the problem, where the end-effector is requested to follow a desired path, while maintaining a desired force whenever in contact with the pipeline surface. No a priori information of the pipeline's surface is exploited. Only the force and moments at the wrist are used to accomplish the task.

For these preliminary simulations, we have used the following task hierarchy: force regulation, joint limits avoidance, manipulability, end-effector alignment with surface normal, end-effector linear control, end-effector angular control, preferred arm posture, vehicle velocity minimization.

Figure 9 shows the starting and ending posture of the arm and the vehicle during the simulation. The following figure instead shows that the proposed scheme is able to regulate the force to the required value, while following the projection of the desired path on the pipe surface with good accuracy and precision.



(a) (b)

Figure *10*: pipeline inspection simulation results (a) regulation of the force (reference value 5 N) and (b) the tracking of the projected path (x,y) components on the pipe

6 *Evaluation plan*

DexROV outcomes will be progressively evaluated in a three-step campaign.

The first part will be focused to validate the ability to build the 3D reconstruction of the natural seafloor and to recognize artificial structure components such as grasping interfaces and structure sub-parts. The ROV operation crew will be regularly located on the vessel, while only observer will be placed in the onshore control center.

The second phase will consist in evaluating the ROV autonomous navigation capabilities, the perception and modeling abilities and the latency mitigation strategy. The main pilot of the ROV and the navigator will supervise the operation from the onshore control center, while the co-pilot will be located on the vessel.

The last part will focus on the dexterous manipulation tasks with a mock up panel placed at 1300 m of depth. This will evaluate the overall abilities of the DexROV system, especially the force feedback control interfaces and the autonomous manipulation capabilities. As for the second part, the pilot and the navigator will be placed in the onshore control center and the ROV co-pilot will be placed on the offshore vessel. For this last trial a test mock-up will be designed that will include a set of standard ISO interfaces as well as tools designed for divers, in order to evaluate the system dexterity.

In particular, the mock-up panel is composed of three sides that offer different application simulations. The first panel will be dedicated to the Oil&Gas validation, presenting a set of scaled valves and handles typically used in deep-sea O&G structures. The second panel is reserved to validate the overall dexterity of the DexROV system, integrating elements that are not commonly used in O&G interventions, to show that DexROV can potentially replace human divers. The third panel is dedicated to the biology and archaeology tasks: 3D printed corals will be integrated allowing the system to scan and recover them, as well as mock-ups of archaeological artifacts will be buried in a sand box to test the system ability to manipulate them.

Performances will be evaluated against a set of key performance indicators, which address aspects such as autonomous capabilities effectiveness, perception and modeling accuracy, and efficacy of latency mitigation strategy.

References

Alizadeh, Calinon, Caldwell. Learning from demonstrations with partially observable task parameters, Proc. IEEE ICRA, Hong Kong, 2014

Antonelli. 2014. Underwater robots. Springer Tracts in advanced robotics. Springer International Publishing. 268 pp.

Antonelli, Cataldi, 2014. Recursive adaptive control for an underwater vehicle carrying a manipulator, 22nd Mediterranean Conference of Control and Automation, pp. 847-852

Antonelli, Arrichiello and Chiaverini. 2008. The Null-Space-based Behavioral Control for Autonomous Robotic Systems, Journal of Intelligent Service Robotics, vol. 1, n. 1, pp. 27-39

Buelow, Birk. Spectral Registration of Noisy Sonar Data for Underwater 3D Mapping, Autonomous Robots, 30(3), pp. 307-331, Springer, 2011

Buelow, Birk. Spectral 6-DOF Registration of Noisy 3D Range Data with Partial Overlap, IEEE Trans on Pattern Analysis and Machine Intelligence, 35(4), pp 954-969, IEEE, 2013

Calinon, Li, Alizadeh, Tsagarakis, Caldwell. 2012. Statistical dynamical systems for skills acquisition in humanoids, Proc. IEEE Int. Conf. on Humanoid Robots, Osaka, pp. 323-329

Calinon, Alizadeh, Caldwell. 2013. On improving the extrapolation capability of task-parametrized movement models, Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, (IROS). Tokio, Japan. pp. 610-616

Calinon, Bruno, Caldwell. 2014. A task-parameterized probabilistic model with minimal intervention control, Proc. IEEE ICRA, Hong Kong

Cataldi, Antonelli, 2015. "Basic interaction operations for an underwater vehicle-manipulator system," in *Advanced Robotics (ICAR)*, pp.524-529

Chiaverini. 1998. Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators, IEEE Trans. Robot Automation, 13(3), pp. 398-410

Chiaverini, Oriolo, Walker. 2008. “Kinematically Redundant Manipulators”. In: “Springer Handbook of Robotics”. pp. 245-268

De Palma, Indiveri, 2016, “Underwater vehicle guidance control design within the DexROV project: preliminary results”, Submitted to the 9th IFAC Symposium on Intelligent Autonomous Vehicles, Leipzig, Germany.

De Palma, Indiveri, 2016, Underwater vehicle guidance control design within the DexROV project: preliminary results. Accepted at 10th IFAC Conference on Control Applications in Marine Systems, Trondheim, Norway

Fairfield, Kantor, Wettergreen, 2007, Real-Time SLAM with Actree Evidence Grids for Exploration in Underwater Tunnels. Journal of Field Robotics, 24(1-2), 3-21

HyunKi, Cho, Kim, Lee. 2011. Jointless structure and under-actuation mechanism for compact hand exoskeleton. IEEE Int. Conf.on Rehabilitation Robotics (ICORR), 1-6

Letier, Motard, Verschueren. 2010. EXOSTATION: Haptic exoskeleton based control station. IEEE ICRA, 1840-1845

Letrie, Motard, Ikìlzkovitz, Preumont, Verschueren. 2011. SAM portable haptic arm exoskeleton upgrade technologies and new application fields. Symposium of advanced space technologies for Robotics and Automation (ASTRA),

Pathak, Birk, Vaskevicius, Poppinga. 2010. Fast registration based on noisy planes with unknown correspondences for 3D mapping, IEEE Trans. On Robotics, 26(3), pp.424-441

Pathak, Pfingsthorn, Buelow, Birk. 2013. Robust estimation of camera-tilt for iFMI based photo-mapping usinga calibrated monocular camera, ICRA, Karlsruhe, Germany, IEEE Press

Pfingsthorn, Birk, Buelow. 2012. Uncertainty Estimation for a 6-DOF spectral registration method as basis for sonar-based underwater 3D SLAM, ICRA, Saint Paul, Minnesota, IEEE Press

Pfingsthorn, Buelow, Sokolovsky, Birk. 2013. Underwater stereo data acquisition and 3D registration with a spectral method, IEEE Oceans, Bergen, Norway

Rozo, Calinon, Caldwell, Jimenez, Torras. 2013. Leaning collaborative impedance-based robot behaviors. Proc. AAAI Conference in Artificial Intelligence, pp. 1422-1428, Bellevue, Washington, USA

Saez, Hogue, Escolano, Jenkin, 2006. Underwater 3D SLAM through entropy minimization, Proceedings of IEE ICRA 2006, pp 3562-2567

Sedlazeck, Koeser, Koch, 2009. 3D Reconstruction Based on Underwater Video from ROV Kiel 6000 Considering Underwater Imaging Conditions, IEEE OCEANS Conference ’09

Simetti, Casalino, Torelli, Sperindè, Turetta. 2013. Floating underwater manipulation: developed control methodology and experimental validation within the TRIDENT project, Wiley Journal of Field Robotics, Wiley

Simetti, E. & Casalino, G. 2015. Whole body control of a dual arm underwater vehicle manipulator system. Annual Reviews in Control, volume 40:191—200.

Whitcomb, L.L., & Yoerger, D.R. 1995. Comparative experiments in the dynamics and model-based control of marine thrusters. In: OCEANS 1995 pp. 1019—1028.