

Copyright Information

This is a post-peer-review, pre-copyedit version of the following paper

Casalino G., Simetti E., Wanderlingh F. (2017) Robotized Underwater Interventions. In: Fossen T., Pettersen K., Nijmeijer H. (eds) Sensing and Control for Autonomous Vehicles. Lecture Notes in Control and Information Sciences, vol 474. Springer, Cham

The final authenticated version is available online at:

https://doi.org/10.1007/978-3-319-55372-6_17

You are welcome to cite this work using the following bibliographic information:

BibTeX

```
@Inbook{Casalino2017,  
  author="Casalino, Giuseppe  
    and Simetti, Enrico  
    and Wanderlingh, Francesco",  
  editor="Fossen, Thor I.  
    and Pettersen, Kristin Y.  
    and Nijmeijer, Henk",  
  title="Robotized Underwater Interventions",  
  bookTitle="Sensing and Control for Autonomous Vehicles:  
    Applications to Land, Water and Air Vehicles",  
  year="2017",  
  publisher="Springer International Publishing",  
  pages="365--386",  
  isbn="978-3-319-55372-6",  
  doi="10.1007/978-3-319-55372-6_17",  
}
```

Robotized Underwater Interventions

G. Casalino, E. Simetti, F. Wanderlingh

Abstract Working in underwater environments poses many challenges for robotic systems. One of them is the low bandwidth and high latency of underwater acoustic communications, which limits the possibility of interaction with submerged robots. One solution is to have a tether cable to enable high speed and low latency communications, but that requires a support vessel and increases costs. For that reason, autonomous underwater robots are a very interesting solution. Several research projects have demonstrated autonomy capabilities of Underwater Vehicle Manipulator Systems (UVMS) in performing basic manipulation tasks, and, moving a step further, this chapter will present a unifying architecture for the control of an UVMS, comprehensive of all the control objectives that an UVMS should take into account, their different priorities and the typical mission phases that an UVMS has to tackle. The proposed strategy is supported both by a complete simulated execution of a test-case mission and experimental results.

1 Introduction

Underwater operations are costly and demanding. Remotely Operated Vehicles (ROVs) are usually employed in lieu of professional divers to perform these tasks. The advantages of this solution are clearly an inherently higher safety (no human

Giuseppe Casalino

Interuniversity Research Center on Integrated Systems for Marine Environment (ISME) and DIB-RIS, University of Genova, - e-mail: giuseppe.casalino@unige.it

Enrico Simetti

Interuniversity Research Center on Integrated Systems for Marine Environment (ISME) and DIB-RIS, University of Genova - e-mail: enrico.simetti@unige.it

Francesco Wanderlingh

Interuniversity Research Center on Integrated Systems for Marine Environment (ISME) and DIB-RIS, University of Genova - e-mail: francesco.wanderlingh@dibris.unige.it

involved) and an higher operational time. On the downside, ROVs require a support vessel equipped with a dynamic positioning system and a tether management system for handling the tether cable connecting the vessel to the ROV. In an effort to reduce costs of underwater operations, marine robotic research has taken two main paths: on the one hand improving ROVs autonomy, on the other hand paving the way for the use of autonomous UVMSs (Underwater Vehicle Manipulator Systems).

The DexROV project [1] is a recent EU Horizon 2020 (H2020) project that has the goal of reducing the costs of ROVs operations by increasing their autonomy. In this way, crew numbers on board the ship can be reduced and can be transferred to an onshore control facility, from where they can control the ROVs operations. Naturally, controlling the ROV from a remote station through a satellite communication channel introduces heavy latency that must be properly taken into account. DexROV plans to solve this problem by using an advanced environment simulator, integrating a physics engine and a haptics engine. This environment will support a model-mediated teleoperation approach. Furthermore, as a robust approach to address communication latencies and disruptions, a cognitive engine relying on probabilistic movement/feedback manipulation primitives will be developed to analyze, interpret and anticipate user interactions with the simulated environment and translate them into consistent *high level* commands that the ROV can execute autonomously in the real environment.

Another relevant project on the topic of underwater manipulation task is the recently started EU H2020 ROBUST project [2], whose main objective is to develop an autonomous, reliable, cost effective technology to map vast terrains, in terms of mineral and raw material contents, which will aid in reducing the cost of mineral exploration, currently performed by ROVs and dedicated support vessels and crew. Another objective is to identify, in an efficient and non-intrusive manner (minimum impact to the environment), the most rich mineral sites. To tackle these objectives, the ROBUST project proposes to use a fully autonomous UVMS. The robot should dive, identify the resources to be scanned and autonomously perform qualitative and quantitative in-situ analyses, by positioning a laser, mounted on the end-effector of a manipulator, in close proximity of the rocks to be analyzed.

Research in the field of autonomous or semi-autonomous intervention is of course not limited to H2020 projects, but has roots that go back to early 90s, with the works of Woods Hole Oceanographic Institute on design and control of compliant underwater manipulators [3] and coordinated vehicle/arm control for tele-operation [4]. During that decade, two milestones were achieved. The AMADEUS project [5] demonstrated underwater dual arm autonomous manipulation in water tank experiments [6]. The second milestone was instead achieved within the UNION project [7], where for the first time the mechatronic assembly of an autonomous UVMS took place.

The following decade marked two further important milestones in underwater manipulation. The SAUVIM project [8, 9] tackled, for the first time, the problem of autonomous floating operations. However, considering the mass of the vehicle was 6 tons, while the arm weighted only 65 Kg, the two subsystem were practically decoupled from a dynamic point of view. The second milestone was instead achieved

within the EU project ALIVE [10, 11], which demonstrated how an UVMS could autonomously dock at a ROV-friendly panel, and successively perform fixed-base manipulation tasks.

The current decade has seen another important milestone, represented by the EU project TRIDENT [12, 13], where autonomous floating manipulation has been performed like in the SAUVIM project, but with the important difference that the arm and vehicle were much closer in terms of masses. The final demonstration took place in a harbor environment and the UVMS successfully recovered a mock-up of a black box from the sea-floor [14, 15].

Finally, the authors have been involved in an Italian research project named MARIS [16], whose main objective was the development of control algorithms capable of integrating single [17], dual arm [18] and cooperative UVMSs [19] within a common control framework [20].

In this chapter the kinematic control layer that has been used for the MARIS project, and is currently used and enhanced within the DexROV [21] and ROBUST projects will be presented. The focus is the kinematic control layer, because most ROV and arm commercial systems can only be controlled through velocity commands. A dynamic control layer (DCL) based on independent PI (proportional integrative) loops will be assumed, as this is the kind of control laws implemented by the commercial systems of the aforementioned projects.

The chapter is structured as follows. Basic definitions are reported in section 2. Section 3 presents the common phases and control objectives in typical intervention missions. The successive section 4 introduces the kinematic control layer and its core concepts, such as the phase definition, the activation functions and the task hierarchy resolution. Section 5 reports a complete simulated execution of an intervention mission, as well as one free floating experiment of the MARIS project. Finally, the last section draws some conclusions and future line of developments.

2 Definitions and Preliminaries

2.1 Notation

Vectors and matrices are expressed with a bold face character, such as \mathbf{M} for matrices or \mathbf{v} for vectors, whereas scalar values are represented with a normal font such as γ . Given a matrix \mathbf{M} and a vector \mathbf{v} :

- $M_{(i,j)}$ indicates the element of \mathbf{M} at the i -th row and j -th column;
- $v_{(k)}$ refers to the k -th element of \mathbf{v} ;
- $\mathbf{M}^\#$ is the exact generalized pseudo-inverse (see [22] for a review on pseudo-inverses and their properties), i.e. the pseudo inverse of \mathbf{M} performed without any regularizations.

Further, less used notation will be introduced as needed.

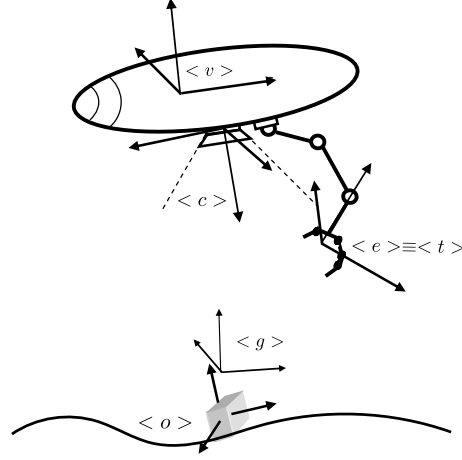


Fig. 1 The UVMS and its relevant frames: $\langle v \rangle$ vehicle frame, $\langle c \rangle$ camera frame, $\langle e \rangle$ end effector frame, $\langle t \rangle$ tool frame, $\langle g \rangle$ goal frame, $\langle o \rangle$ object frame.

2.2 Definitions

Let us consider a free floating UVMS, such as the one depicted in Fig. 1, and let us first introduce some basic definitions, often used throughout the chapter:

- the system configuration vector $\mathbf{c} \in \mathbb{R}^n$ of the UVMS as

$$\mathbf{c} \triangleq \begin{bmatrix} \mathbf{q} \\ \boldsymbol{\eta} \end{bmatrix}, \quad (1)$$

where $\mathbf{q} \in \mathbb{R}^l$ is the arm configuration vector and $\boldsymbol{\eta} \in \mathbb{R}^6$ is the vehicle *generalized coordinate position vector*, which is the stacked vector of the position vector $\boldsymbol{\eta}_1$, with components on the inertial frame $\langle 0 \rangle$, and the orientation vector $\boldsymbol{\eta}_2$, the latter expressed in terms of the three angles yaw, pitch and roll (applied in this sequence) [23]. From the above definitions it results $n = l + 6$;

- the system velocity vector $\dot{\mathbf{y}} \in \mathbb{R}^n$ of the UVMS as

$$\dot{\mathbf{y}} \triangleq \begin{bmatrix} \dot{\mathbf{q}} \\ \mathbf{v} \end{bmatrix}, \quad (2)$$

where $\dot{\mathbf{q}} \in \mathbb{R}^l$ are the joint velocities and $\mathbf{v} \in \mathbb{R}^6$ is the stacked vector of the vehicle linear and angular velocity vectors, with components on the vehicle frame $\langle v \rangle$. To simplify the discussion, this chapter will assume the vehicle fully actuated, and the system velocity will be used as control vector. Details on how the proposed algorithm can be adapted for under-actuated vehicles are given in the Appendix;

- a configuration dependent scalar variable $x(\mathbf{c})$ is said to correspond to an *equality control objective* when it is required to satisfy

$$x(\mathbf{c}) = x_0, \quad (3)$$

or to an *inequality control objective* when it is required to satisfy

$$x(\mathbf{c}) \geq x_m \quad \text{or} \quad x(\mathbf{c}) \leq x_M, \quad (4)$$

where the m and M subscripts indicate a minimum and maximum value respectively.

Note that if m different variables $x_i(\mathbf{c})$ are considered, each of them corresponding to the i -th component of a vector $\mathbf{p} \in \mathbb{R}^m$, then it is possible to control the vector to any desired value. Thus limiting the discussion to scalar objectives does not influence the generality of the approach. Furthermore, if $x(\mathbf{c})$ is the modulus of a certain vector \mathbf{p} , then it can be used to require a particular value for the norm of \mathbf{p} (e.g. to nullify it), or to be below or above a given threshold. To ease the notation, the dependency of x on \mathbf{c} is dropped from now on;

- for such variables, we also consider the existing Jacobian relationship between x and the system velocity vector $\dot{\mathbf{y}}$ as

$$\dot{x} = \mathbf{g}^T(\mathbf{c})\dot{\mathbf{y}}, \quad (5)$$

where $\mathbf{g} \in \mathbb{R}^n$ is a vector. To simplify the notation, the dependency of \mathbf{g} on \mathbf{c} will not be repeated in the following.

- a *task* is defined as tracking a given feedback reference rate $\dot{\hat{x}}$ (see the remarks below), capable of driving the associated variable x toward the corresponding objective. Thus, for instance, a task is tracking at best a velocity reference rate generated to bring the arm's end-effector in the required Cartesian position; The control objectives may have different *priorities* and the same holds for their associated tasks. The achievement of a task with lower priority should not interfere with the achievement of an *active* task (see section 4.2) with higher priority, and tasks with the same priority should be achieved simultaneously, if possible. A set of tasks with different priorities is also called a *hierarchy of tasks*.

Remark 1: If our objective is zeroing the norm of a vector $\mathbf{p} = [p_1, \dots, p_m]$, since the following equivalence holds

$$\|\mathbf{p}\| = 0 \iff p_i = 0, \forall i \quad i = 1, \dots, m, \quad (6)$$

instead of imposing its norm to be zero with a single equality objective, we can also consider separately all of its m components to be zero. The main difference between these two approaches is that in the first case we have to employ only one degree of freedom (d.o.f.) in the Cartesian space, while in the latter use m d.o.f. since we are separately controlling each of its components.

Remark 2: For equality or inequality control objectives, a suitable feedback reference rate $\dot{\hat{x}}$ that drives our x toward any arbitrary point x^* , where the objective is satisfied, is

$$\dot{\hat{x}} \triangleq \gamma(x^* - x), \gamma > 0, \quad (7)$$

where γ is a positive gain to control the convergence speed. For inequality control objectives, section 4.2 will explain how to disregard the reference rate whenever the variable lies within the desired range, in order to avoid over-constraining the system.

3 Underwater Intervention Missions and Related Control Objectives

3.1 *Reference Mission and Relevant Phases*

A typical reference mission carried out by an UVMS, involving manipulation and transportation, can be decomposed into the following sequential phases:

1. navigation: the vehicle should get in close proximity with the target object to be manipulated;
2. grasping: the UVMS must perform the grasping of the object;
3. transportation: the UVMS must transport the object to the target area;
4. deployment: whenever in close proximity with the target area, the object must be deployed in the required position.

Another mission, involving the inspection of a certain subsea structure, can be decomposed as follows:

1. navigation: the vehicle should get in close proximity with the target object to be inspected;
2. docking: the UVMS might need to dock to the underwater structure using an auxiliary arm;
3. inspection: the UVMS should perform the interaction with the structure (e.g. turn valve, plug connector, move a sensor on a surface).

At the end of the above phases, the UVMS can leave the target area and each of them can be assigned to a new mission.

In general, the execution of complex missions leads to identify different control phases, each one with its own control objectives. Obtaining a smooth transition between two consecutive phases will constitute a particular challenge to be solved.

3.2 *Control Objectives Categories*

The control objectives of the UVMS can be divided in five broad categories, which are listed in their natural descending order of priority:

- physical constraints objectives, i.e. interacting with the environment;
- system safety objectives, e.g. avoiding joint limits or obstacles;

- objectives that are a prerequisite for accomplishing the mission, e.g. maintaining the manipulated object in the camera angle of view;
- mission oriented objectives, i.e. what the system really needs to execute to accomplish the user defined mission;
- optimization objectives, i.e. objectives that do not influence the mission, but allow to choose between multiple solutions, if they exists.

3.3 Typical Underwater Control Objectives

This section reports some of the control objectives that a UVMS has to tackle in a typical mission, without a particular ordering. For each objective, the type (equality/inequality) and its category are reported in brackets:

- Joint Limits (inequality, safety): the arm must operate within its joint limits, which means having the following inequality control objectives fulfilled

$$q_{i,m} \leq q_i \leq q_{i,M} \quad i = 1, \dots, l \quad (8)$$

where q_i is the i -th joint variable, $q_{i,m}$ and $q_{i,M}$ are the lower and higher joint bounds, and l is the total number of joints of the manipulator.

- Manipulability (inequality, prerequisite): the arm must operate as far as possible from kinematic singularities, which means to keep the manipulability measure μ [24] above a minimum threshold, i.e. $\mu > \mu_m$.
- Arm Reference Shape: maintain the arm in a preferred shape, which allows to perform repetitive tasks minimizing the internal motions and can also be used as an alternative method to ensure good manipulability.
- Camera Occlusion (inequality, prerequisite): keep the arm links away from the camera system's cone of vision to avoid unnecessary occlusions of the target object frame $\langle o \rangle$, e.g. to avoid arm elbow interferences with the camera well before the grasping phases.
- Force Regulation (equality, physical constraint): regulate to a constant value the force exerted by the end effector. Let us define λ^* as the desired force that the end-effector must exert on the environment and λ as the actual force, then the objective is to have $\lambda = \lambda^*$. Note that this regulation has to be done at kinematic level, since the underlying DCL is assumed as given.
- Camera Centering (inequality, prerequisite): a stereo vision system is often used to obtain the position of the objected to be manipulated. To help in guaranteeing continuous visual contact with the target while approaching it, the norm of the misalignment vector between the camera and object frames ξ should be maintained lower than a given bound, e.g. $\|\xi\| \leq \xi_M$.
- Horizontal Attitude (inequality, safety): avoid vehicle overturning. The norm of the misalignment vector ϕ between the absolute world frame z axis and the vehicle's one should be lower than a give value, i.e. $\|\phi\| \leq \phi_M$. For vehicles and arms with comparable masses, such thresholds should be big enough to avoid

excessive energy consumption whenever the arm moves its joints away from the center of buoyancy, tilting the vehicle. This objective should be considered only for fully actuated vehicles since for under-actuated ROVs or AUVs passively stable in roll and pitch these d.o.f. are not controllable.

- Vehicle Position (inequality, mission execution): have the vehicle frame $\langle v \rangle$ roughly aligned with a particular goal frame $\langle g_v \rangle$. This could be required in order to bring the vehicle close to the area where the manipulation needs to be carried out. This goal requires the achievement of the following inequality conditions

$$\|\mathbf{r}_v\| \leq r_{v,M}, \quad \|\boldsymbol{\vartheta}_v\| \leq \vartheta_{v,M} \quad (9)$$

where \mathbf{r}_v is the position error and $\boldsymbol{\vartheta}_v$ the orientation error.

- End-effector/Tool-frame Position (equality, mission execution): this objective requires that the tool frame $\langle t \rangle$, rigidly attached to the end-effector space, converges to a given goal frame $\langle g \rangle$. In other words, the following two equality objectives must be eventually satisfied

$$r_{(i)} = 0, i = 1, 2, 3; \quad \vartheta_{(i)} = 0, i = 1, 2, 3; \quad (10)$$

where \mathbf{r} is the position error and $\boldsymbol{\vartheta}$ the orientation error between the tool and goal frames. Note that the component by component zeroing has been used, in lieu of the norm zeroing, for achieving a straight convergence to the target (especially important for grasping tasks).

Moreover there can also be tasks directly specified at velocity level, which are useful for better controlling the overall behavior of the system, i.e.:

- Arm motion minimization (equality, optimization): in some cases (e.g. during transportation) it might be better if the arm minimizes its movements, leaving to the vehicle most of the work to move the end-effector. This allows to use the arm d.o.f. to compensate for the vehicle controller errors in tracking its reference velocity, as explained in section 4.4;
- Vehicle motion minimization (equality, optimization): it might be preferred to have the vehicle roughly stationary (e.g. during grasping and manipulation), since usually the arm exhibits better control performances than the vehicle.

In order to prioritize one of the two subsystems these two task should be mutually exclusive. Let us also remark that these tasks exploit any residual arbitrariness on their relevant control variables so it is a good practice to place them at the bottom of the task hierarchy, since all the higher priority tasks are related to the safety of the system (e.g. joint limits, force regulation, vehicle attitude) or a prerequisite to complete the mission (e.g. manipulability, camera centering).

4 A Unifying Control Framework

The control of an UVMS is achieved by solving a sequence of optimization problems, following the assigned priority of each control objective. This mechanism descends from the original task priority framework [25] that has been extended to also encompass scalar tasks corresponding to inequality control objectives [15], where each of them is assigned with a different priority, and successively to include clusters of control objectives with equal priorities [18, 26]. We will now recall the basic steps behind the algorithmic structure of the task priority based control layer.

4.1 Phase Control Priorities and Unified Task List

Each phase of a mission is characterized by its control objectives, and consequently by a hierarchy of tasks. Two phases may have multiple tasks in common, despite they might have a different priority order within each list. For instance, consider the following two lists of tasks (now abstractly labeled with alphabetic letters) for two different phases, where $A \prec B$ denotes that A has higher priority than B :

$$\text{P1 : } A \prec B, C, D$$

$$\text{P2 : } A \prec D \prec C, E$$

where A, C, D are in common, but with D at a different priority ordering w.r.t. C . Now consider the following merged list:

$$\text{P1, P2 : } A \prec D \prec B, C, E;$$

It is clear that, through insertion/deletion of some of the entries, the two original lists can be reconstructed. To do so, the mechanism of activation functions will be exploited, as explained in the next section.

Before concluding, from now on tasks at the same priority level will be stacked together forming a so called multidimensional task $\dot{\mathbf{x}}_i$, where i indicates the priority level. For generality we will consider scalar tasks as a particular case of the multidimensional ones, and consequently we shall indicate a prioritized task list simply as $\dot{\mathbf{x}}_1, \dots, \dot{\mathbf{x}}_N$. and the Jacobians relevant to the actual task velocities $\dot{\mathbf{x}}_1, \dots, \dot{\mathbf{x}}_N$ as $\mathbf{J}_1, \dots, \mathbf{J}_N$.

4.2 Activation Functions

Let us consider a multidimensional task, and let us consider an activation function associated to each j -th of its components, called $a_{(j)}$, to be then organized in a diagonal activation matrix \mathbf{A} , whose meaning is the following:

- if $a_{(j)} = 1$, the associated scalar task is called *active* and the corresponding actual $\dot{x}_{(j)}$ should therefore track $\dot{\hat{x}}_{(j)}$ as close as possible;
- if $a_{(j)} = 0$, the scalar task is termed *inactive* and the actual $\dot{x}_{(j)}$ should be unconstrained;
- if $0 < a_{(j)} < 1$ the scalar task is termed *in transition* and the actual $\dot{x}_{(j)}$ should smoothly evolve between the two previous cases.

In particular, the overall activation function $a_{(j)}$ is the product of two functions

$$a_{(j)} \triangleq a_{(j)}^p a_{(j)}^i, \quad (11)$$

which have the following specific purposes:

- $a_{(j)}^i$ is a function of the control objective variable $x_{(j)}$, and its purpose is to deactivate the task whenever the inequality objective is satisfied, to avoid overconstraining the system. Note that for equality control objectives it clearly holds that $a_{(j)}^i = 1$;
- $a_{(j)}^p$ is a function of the mission phases, and is used to activate or deactivate a task during a phase transition. For example, it can be a function of the time elapsed within the current phase, allowing the proper activation of new tasks, and deactivation of objectives that are not anymore relevant.

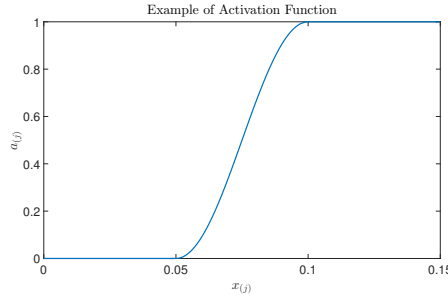


Fig. 2: Example of activation function corresponding to a control objective $x_{(j)} \leq 0.1$, with $\Delta = 0.05$

More specifically, $a_{(j)}^i$ is defined as follows for objectives of the type $x_{(j)} \leq x_{(j),M}$ (a similar function can be constructed for objectives $x_{(j)} \geq x_{(j),m}$):

$$a_{(j)}^i \triangleq \begin{cases} 1, & x_{(j)} > x_{(j),M} \\ s_j(x), & x_{(j),M} - \Delta_{(j)} \leq x_{(j)} \leq x_{(j),M} \\ 0, & x_{(j)} < x_{(j),M} - \Delta_{(j)} \end{cases} \quad (12)$$

where $s_j(x)$ is any sigmoid function exhibiting a continuous behavior from 0 to 1. The $\Delta_{(j)}$ value allows to create a buffer zone, where the inequality is already

satisfied, but the activation value is still greater than zero, to prevent any chattering problem around the inequality control objective threshold. An example of such a function is reported in Fig. 2.

Let us briefly clarify how the activation function mechanism works with inequality control objectives through an example. Suppose that $x_{(j),M} = 0.1$ and $\Delta_{(j)} = 0.05$ (this is the case depicted in Fig. 2). Then if we consider x^* in (7) equal to $x_{(j),M} - \Delta_{(j)} = 0.05$ (note that 0.05 is inside the validity region of the inequality objective) we have the following behavior:

- When $x_{(j)} > 0.1$, the task is active $a_{(j)}^i = 1$ and the feedback reference rate defined in (7) will drive $x_{(j)}$ towards the region where $x_{(j),M} < 0.1$, in particular toward the point $x^* = 0.05$.
- When $0.05 < x_{(j)} < 0.1$, the task is in the transition zone $0 < a_{(j)}^i < 1$. The feedback reference rate will still try to drive $x_{(j)}$ towards the point $x^* = 0.05$. However, the priority of this task is not any longer fully enforced. Indeed, as $a_{(j)}^i$ decreases from 1 to 0, lower priority tasks will have the opportunity to influence the current task, possibly pushing it to either its lower bound (0.05, not in conflict with the current task) or its upper bound (0.1, in conflict with the current task).

Let us for a moment consider a continuous time, purely kinematic system. In the former case, the task will be simply completely deactivated. In the latter case $x_{(j)}$ will decrease until the point where lower priority tasks' influence on the current task matches the feedback reference rate, achieving a stable condition within the transition zone, where by definition the inequality control objective is satisfied. This behavior is enforced by the specific regularization mechanism that will be briefly presented in section 4.3 and that can be found in [26].

In practice, discrete time control system have to be used both at kinematic and dynamic velocity control level, where the latter are usually characterized by higher control rates. For given dynamic performances, which will depend on the control rate, gains and specific dynamic controller structure, a tuning of both kinematic gains and activation functions parameters is actually required to ensure that chattering phenomena do not appear. Since this may sometime result into a tedious trial-and-error process, research effort are currently devoted toward the possible formalization of specific procedures for the problem in hand.

- Finally, when $x_{(j)} < 0.05$, the task is inactive and the task velocity is determined only by the rest of the task hierarchy.

Remark: We have implicitly considered objectives of the type $x_m < x < x_M$ as two separate ones. Note that if x_m and x_M are sufficiently spaced, i.e. $x_m + \Delta < x_M - \Delta$, then they can be considered together by using as activation function the sum of the two activation functions, and by choosing an arbitrary point inside the validity of both inequality to construct the common reference rate in (7). This is actually what is done for the joint limits task implementation, since the minimum and maximum limits satisfy the above conditions.

4.3 Solution of the Task Hierarchy Problem

In the previous sections different concepts have been introduced. The UVMS needs to fulfil different control objectives. Tracking suitable feedback reference rates allows the UVMS to meet these objectives. We have termed this need as task. We have also seen that tasks should be activated and deactivated for two main reasons: either the current control phase does not require that specific objective and its related task, or the task corresponds to an inequality objective which is currently satisfied. For the reasons outlined above, whenever that situation occurs we do not want the control to be overconstrained. To comply with this need, activation functions have been introduced, which describe whether a given task should or not be fulfilled.

Having said that, the problem becomes that of tracking the given reference velocities, following the required priority order, and taking into account their corresponding activation values. The solution of this problem can be found solving the following sequence of minimization problems:

$$S_k \triangleq \left\{ \arg \text{R-} \min_{\dot{\mathbf{y}} \in S_{k-1}} \|\mathbf{A}_k(\dot{\mathbf{x}}_k - \mathbf{J}_k \dot{\mathbf{y}})\|^2 \right\}, \quad k = 1, 2, \dots, N, \quad (13)$$

where S_{k-1} is the manifold of solutions of all the previous tasks in the hierarchy.

The solution of each problem in (13) is not straightforward. It is well known that minimization problems can be invariant to weights (such is the role of \mathbf{A}_k) [27, 28]. For that reason, we have used the notation R-min, to highlight the fact that we employ a special regularized pseudo inverse solution of that problem, as defined in [26]. In fact, the solution of the regularized minimization problem exploits the following definition of pseudo inverse operator $\mathbf{X}^{\# \mathbf{A}, \mathbf{Q}}$ for given non-negative definite matrices \mathbf{A}, \mathbf{Q} , with dimensions equal to the rows and columns of \mathbf{X} respectively:

$$\mathbf{X}^{\# \mathbf{A}, \mathbf{Q}} \triangleq (\mathbf{X}^T \mathbf{A} \mathbf{X} + (\mathbf{I} - \mathbf{Q})^T (\mathbf{I} - \mathbf{Q}) + \mathbf{V}^T \mathbf{H} \mathbf{V})^{\#} \mathbf{X}^T \mathbf{A} \mathbf{A}, \quad (14)$$

where \mathbf{V} is the right orthonormal matrix of the SVD decomposition of $\mathbf{X}^T \mathbf{A} \mathbf{X} + (\mathbf{I} - \mathbf{Q})^T (\mathbf{I} - \mathbf{Q})$. The matrix \mathbf{H} is a diagonal (singular value oriented regularization) matrix, whose elements $h_{(i,i)}$ are bell-shaped, finite support functions (similar to the activation functions) of the corresponding singular value of the same mentioned SVD decomposition. As a brief insight, let us remark how the pseudo inverse operator *explicitly* depends on the weights \mathbf{A} (the activation function) and \mathbf{Q} (the non orthogonal projection matrix). This fact allows the operator to avoid the problems of discontinuity arising from the invariance of the minimization w.r.t. the weights. The interested reader can find more details in [26].

Using the above operator for solving a hierarchy of tasks, a methodology termed *iCAT task priority framework*, results in the following algorithm:

$$\boldsymbol{\rho}_0 = \mathbf{0}, \quad \mathbf{Q}_0 = \mathbf{I}, \quad (15)$$

then for $k = 1, \dots, N$

$$\begin{aligned}
\mathbf{W}_k &= \mathbf{J}_k \mathbf{Q}_{k-1} (\mathbf{J}_k \mathbf{Q}_{k-1})^{\# \mathbf{A}_k} \mathbf{Q}_{k-1}, \\
\mathbf{Q}_k &= \mathbf{Q}_{k-1} (\mathbf{I} - (\mathbf{J}_k \mathbf{Q}_{k-1})^{\# \mathbf{A}_k} \mathbf{J}_k \mathbf{Q}_{k-1}), \\
\boldsymbol{\rho}_k &= \boldsymbol{\rho}_{k-1} + \mathbf{Q}_{k-1} (\mathbf{J}_k \mathbf{Q}_{k-1})^{\# \mathbf{A}_k} \mathbf{I} \mathbf{W}_k (\dot{\mathbf{x}}_k - \mathbf{J}_k \boldsymbol{\rho}_{k-1}),
\end{aligned} \tag{16}$$

where

- $\boldsymbol{\rho}_k$ is the control vector, which is computed in an iterative manner by descending the various priority levels;
- $(\dot{\mathbf{x}}_k - \mathbf{J}_k \boldsymbol{\rho}_{k-1})$ is the modified task reference that takes into account the contribution of the control vector $\boldsymbol{\rho}_{k-1}$ established at the previous iteration;
- \mathbf{Q}_{k-1} is the projection matrix that is used to take into account the control direction (totally or partially) used by the higher priority tasks;
- \mathbf{W}_k is a $m \times m$ matrix, where m is the row-dimension of the task at the current priority level, whose effect is to modify the task reference $(\dot{\mathbf{x}}_k - \mathbf{J}_k \boldsymbol{\rho}_{k-1})$ to avoid discontinuities between priority levels.

A further improvement that can be applied to this technique is to take into account velocity saturations at each priority level, following the methodology first proposed in [29]. To do so, the update equation of $\boldsymbol{\rho}_k$ is redefined as follows:

$$\boldsymbol{\rho}_k = \boldsymbol{\rho}_{k-1} + \text{Sat} \left(\mathbf{Q}_{k-1} (\mathbf{J}_k \mathbf{Q}_{k-1})^{\# \mathbf{A}_k} \mathbf{I} \mathbf{W}_k (\dot{\mathbf{x}}_k - \mathbf{J}_k \boldsymbol{\rho}_{k-1}) \right), \tag{17}$$

where the function $\text{Sat}(\cdot)$ implements the saturation proposed in [29].

The above task hierarchy resolution ends with the N -th manifold $S_N = \{\dot{\mathbf{y}} = \boldsymbol{\rho}_N + \mathbf{Q}_N \dot{\mathbf{z}}_N; \forall \dot{\mathbf{z}}_N\}$. To ensure the continuity, a final minimization on the control vector needs to be performed [26], leading to the following final velocity control vector:

$$\dot{\mathbf{y}} = \arg \min_{\dot{\mathbf{y}} \in S_N} \|\dot{\mathbf{y}}\|^2 = \boldsymbol{\rho}_{N+1}. \tag{18}$$

4.4 Vehicle Velocity Tracking Error Compensation Scheme

An important consideration when dealing with UVMS is that the two subsystem are characterized by very different dynamic performances. Indeed, it is well known that thrusters have nonlinear properties [30, 31, 32]. This nonlinearity, coupled with the higher mass of the vehicle compared to the manipulator's one, makes the velocity control of the vehicle far less accurate than the arm's one.

The proposed algorithm (16) solves the task hierarchy considering the vehicle and arm velocity together in the stacked vector $\dot{\mathbf{y}}$. However, for reasons states at the start of this section, inevitable vehicle velocity tracking errors will occur. To overcome this problem, the idea is to add, in parallel to (16), another task hierarchy resolution, where only the arm variables are subject to optimization. The *actual* vehicle velocity, as measured by onboard sensors, is used as a parameter to solve the task hierarchy.

This procedure leads to the following set of equations:

$$\begin{aligned} \mathbf{W}_k &= \mathbf{J}_k^a \mathbf{Q}_{k-1} (\mathbf{J}_k^a \mathbf{Q}_{k-1})^{\#A_k} \mathbf{Q}_{k-1}, \\ \mathbf{Q}_k &= \mathbf{Q}_{k-1} (\mathbf{I} - (\mathbf{J}_k^a \mathbf{Q}_{k-1})^{\#A_k} \mathbf{J}_k^a \mathbf{Q}_{k-1}), \\ \boldsymbol{\rho}_k &= \boldsymbol{\rho}_{k-1} + \text{Sat} \left(\mathbf{Q}_{k-1} (\mathbf{J}_k \mathbf{Q}_{k-1})^{\#A_k} \mathbf{I} \mathbf{W}_k (\dot{\mathbf{x}}_k - \mathbf{J}_k^a \boldsymbol{\rho}_{k-1} - \mathbf{J}_k^v \mathbf{v}) \right), \end{aligned} \quad (19)$$

where now $\boldsymbol{\rho}_k \in \mathbb{R}^l$ contains only the arm joint velocities and each Jacobian \mathbf{J}_k has now been considered as its two separate vehicle contributions, i.e. $\mathbf{J}_k \triangleq [\mathbf{J}_k^a \ \mathbf{J}_k^v]$. The resulting arm control law (19), since parametrized by the vehicle velocity \mathbf{v} , results to be the optimal one in correspondence of *any* vehicle velocity.

5 Simulation and Experimental Results

A series of experiments performed using the proposed task priority based kinematic controller are presented in this section. Experimental results on underwater grasping taken from the MARIS project are reported in the first subsection. The successive one presents a simulation of underwater pipe inspection with force regulation taken from the DeXROV project.

5.1 Approaching and Grasping

The following section presents the data collected during the MARIS project. The setup consisted in a UVMS composed of an under-actuated vehicle with controllable d.o.f. x, y and yaw angle ψ , and a 7 d.o.f. manipulator, with a 3-fingered gripper attached as shown in Fig. 3. Note that the z axis of the vehicle was disabled for safety reasons, since the trials were done in a pool with only 3 m of water. The vehicle was controlled at 10 Hz rate, while the arm was controlled at 33 Hz. The reference mission was approaching and grasping an underwater pipe, whose position was estimated using an on-board stereo camera [33] at an approximate frequency of 3 Hz.

The first trial, depicted in Fig. 4 shows the generated joint velocities using the approach presented in section 4, with the vehicle velocity compensation of section 4.4. It can be noted that the vehicle feedback was particularly noisy on the angular velocity components. In order to compensate for the noisy feedback, the arm moved quite quickly, inducing actual oscillations in the system due to the dynamic coupling with the floating base.

To cope with the above problem, a simple first order filter on the vehicle feedback was introduced, with a cut-off frequency at angular frequency of 50 rad/s. New trials were then performed; one of these is depicted in Fig. 5a, where now the filtered velocity feedback is compared to the reference one along with the reference velocities



Fig. 3 The UVMS R2, used throughout the on-field experiments, while approaching an underwater pipe to grasp it (photo courtesy of the MARIS consortium).

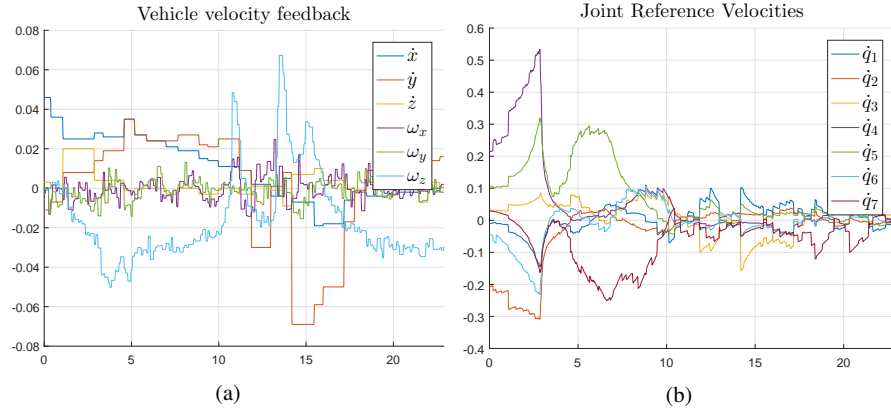


Fig. 4: MARIS experiment without vehicle velocity feedback filtering: (a) vehicle velocity feedback and (b) reference arm joint velocities.

for the arm. Let us note how the control successfully completed the grasp sequence despite the vehicle tracking inaccuracies thanks to the proposed compensation technique. From the Fig. 5b it is possible to see that the generated joint velocities are smoother compared to the first trial, even though the multi-rate nature of the control and feedback can be appreciated. Figure 5c shows the time behaviour of the activation function of the manipulability task. A video showing one of the experiments can be seen at the URL <https://youtu.be/b3jJZUoeFT0>.

Finally, it must be noted how the experiments were necessarily preceded by a general tuning of all the parameters, ranging from the control gains of each task reference rate, to the size of the activation functions buffers Δ .

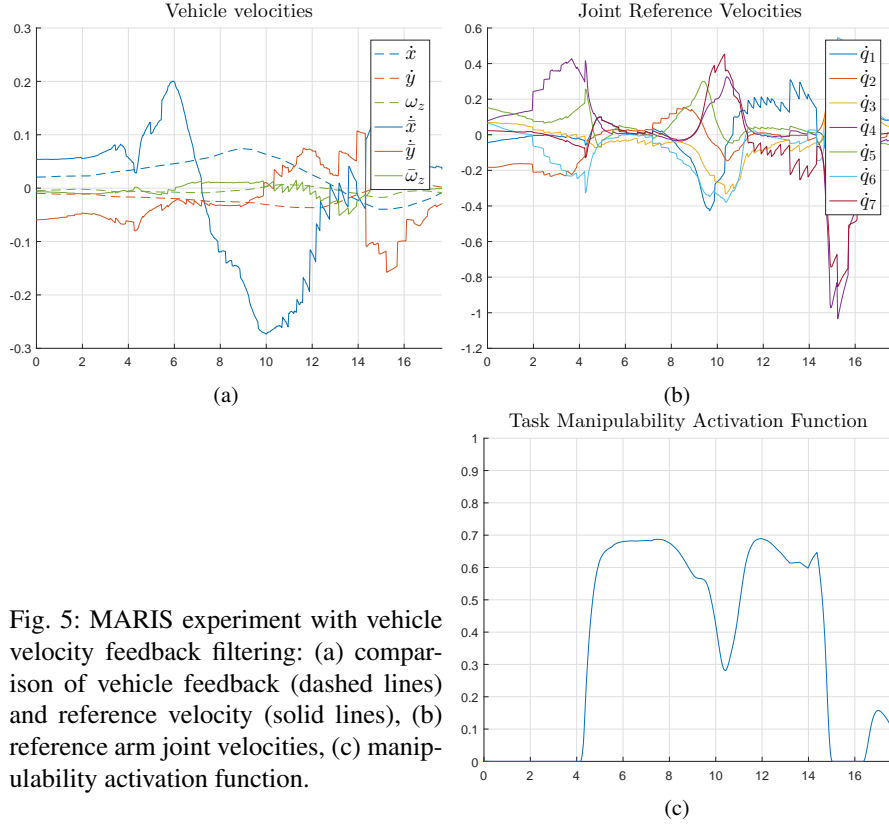


Fig. 5: MARIS experiment with vehicle velocity feedback filtering: (a) comparison of vehicle feedback (dashed lines) and reference velocity (solid lines), (b) reference arm joint velocities, (c) manipulability activation function.

5.2 Force Regulation

A series of simulations were carried out to test the effectiveness of the proposed algorithm including a force regulation task,. The reference mission is the inspection of a pipeline weld. To carry out the inspection, a sensor attached to the end-effector must be put in contact with the pipe along all the weld. The shape of the inspected object is not known, although the knowledge of a two dimensional reference path defined on an known underlying surface was assumed. The UVMS must regulate the force to a given reference value to maintain contact with the pipe without damaging the sensor's probe and adapting to the unknown surface. We recall that this regulation is performed at kinematic level, with a dedicated task at the *top* of the task hierarchy. This is done by measuring the force normal, and generating a velocity along that direction proportional to the force error. This reference velocity, at dynamic level, will become an actual force. This further ensures that lower priority task do not generate velocity references along the direction of the physical constraint.

Compared to previous results [34], this simulation includes realistic saturation values for the vehicle generalized force/moments resulting at the vehicle frame and arm joint torques, other than an hydrodynamic model of the UVMS. The thruster dynamics have not been modelled; the vehicle tracking accuracy are attributed only to the inertia effects and dynamic control performances. In any case, the proposed velocity compensation technique takes into account the resulting effect and would not be different if the thruster dynamic was included in the simulation.

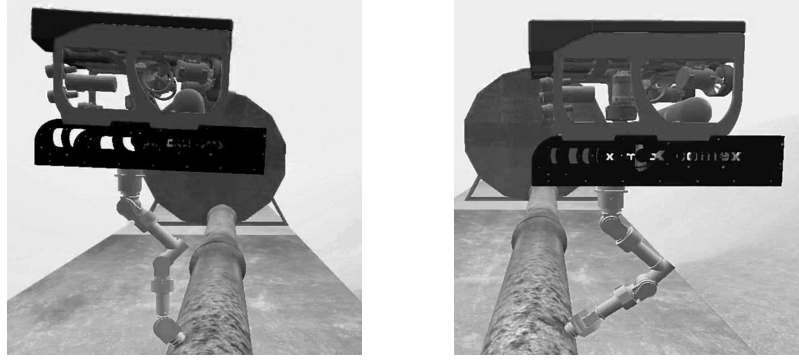


Fig. 6: Two snapshots of the force regulation experiments during the inspection of an underwater pipeline weld.

In the simulation, the underlying DCL implements a PI control law. This case reproduces the actual implementation of DexROV, where the arm and vehicle DCLs are provided by the respective manufacturers and implementing global dynamic controllers, such as a computed torque, is not possible. The regulation of the force is done through a proportional term, since the underlying dynamic level already contains an integral part. The simulation shows that the regulation is accomplished despite different tasks are being activated and deactivated during the trial, e.g. the joint limits and manipulability ones. The proposed controller implements the saturation of reference joint and vehicle velocities as proposed in (17), to avoid generating unrealistic values. Finally, the simulation implements a multi-rate control, where the kinematic control is run at 100 Hz, while the DCL and the dynamic simulation runs at 1 kHz.

Figure 7a shows the force exerted on the pipe, which is very close to the desired value of 10 N. The spikes are in correspondence of the end of the legs, where the vehicle needs to move in a different direction and where most of the vehicle velocity tracking inaccuracies occur. Those spikes were present also in the simulations presented in [34], however the saturation on the actuators increase this effect. Further investigations on how to mitigate these spikes despite saturation are currently ongoing. Finally, Fig. 7b presents the activation values of some tasks that are being activated and deactivated during the mission.

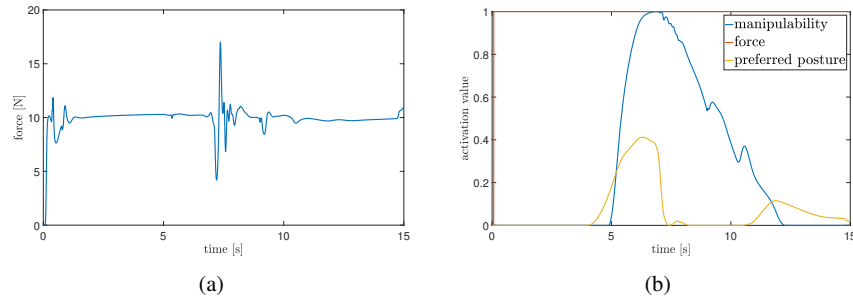


Fig. 7: Pipeline inspection simulation: (a) force exerted by the end-effector on the pipe (desired value 10 N) and (b) activation values of the relevant tasks

6 Concluding Remarks

Oceans covers approximately 70% of Earth's surface and represents an important source of resources. Unfortunately, the underwater environment poses great challenges to robotic systems, due to high pressures and hydrodynamic forces that are both nonlinear. A further important constraint is the low bandwidth of acoustic communications, which are the de facto standard since electromagnetic communications only work at very short ranges.

Despite such premises, underwater robotic systems capable of performing autonomous intervention tasks are a very active topic of research. In the future, the difference between ROV systems with advanced autonomy capabilities and completely autonomous UVMSs is expected to be less prominent. In fact, one of the trends is to imagine hybrid solutions where the robotic system autonomously navigates toward a target area, docks with a subsea structure and a tether cable, and then performs as a semi-autonomous ROV system. Such a solution is very interesting to reduce the maintenance costs of permanent underwater structures, especially for offshore platforms.

The employment of UVMSs can be of course predicted also for applications that do not target the offshore market. For example, exploration for resources with the possibility of in-situ measurements, such as what is proposed in the ROBUST project, is very interesting and promising. Other applications might include the use of UVMSs for deep-sea archaeological sites.

This chapter has presented the overall kinematic control strategy of an UVMS, supported both by experimental results of the TRIDENT and MARIS projects, which is now currently refined in the scope of the DexROV and ROBUST projects. More investigation efforts are still necessary for what concerns interaction tasks with the environment, especially under the presence of saturation effects.

The chapter covered only control issues. However, perception, navigation, communication, and dynamic modelling are other fundamental topics for the development of autonomous underwater robotic systems. The interested reader can

find further readings on these topics in the following survey papers and books [35, 36, 37, 38, 39, 40].

Acknowledgements

This work has been supported by the MIUR (Italian Ministry of Education, University and Research) through the MARIS prot. 2010FBLHRJ project and by the European Commission through the H2020-BG-06-2014-635491 DexROV project and the H2020-SC5-2015-690416 ROBUST project.

Appendix

Under-actuated Vehicles

This chapter has presented UVMS control algorithms under the assumption of fully actuated vehicle. However, in many cases the vehicles are passively stable in some d.o.f. (typically roll and/or pitch). The algorithm (16) can easily cope with this situation, by using a slightly different initialization. As an example, in lieu of (15) consider the following initial values for a vehicle with roll and pitch not actuated:

$$\boldsymbol{\rho}_0 = \begin{bmatrix} \mathbf{0}_{l \times 1} \\ \mathbf{0}_{3 \times 1} \\ \omega_x \\ \omega_y \\ 0 \end{bmatrix} \quad \boldsymbol{Q}_0 = \begin{bmatrix} \boldsymbol{I}_{l \times l} & \mathbf{0}_{l \times 3} & \mathbf{0}_{l \times 1} & \mathbf{0}_{l \times 1} & \mathbf{0}_{l \times 1} \\ \mathbf{0}_{3 \times l} & \boldsymbol{I}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times l} & \mathbf{0}_{1 \times 3} & 0 & 0 & 0 \\ \mathbf{0}_{1 \times l} & \mathbf{0}_{1 \times 3} & 0 & 0 & 0 \\ \mathbf{0}_{1 \times l} & \mathbf{0}_{1 \times 3} & 0 & 0 & 1 \end{bmatrix}. \quad (20)$$

The idea is that the solution $\boldsymbol{\rho}$ is initialized with the actual angular velocities of the vehicle, as measured by onboard sensors. At the same time, to force the task hierarchy resolution to avoid changing these values, the corresponding diagonal values of the matrix \boldsymbol{Q} are set to zero. This effectively inhibits the algorithm from changing the initial values. Note that all the tasks will properly take into account the non-actuated d.o.f. velocities due to the term $(\dot{\hat{\mathbf{x}}}_k - \boldsymbol{J}_k \boldsymbol{\rho}_{k-1})$.

References

1. J. Gancet, P. Weiss, G. Antonelli, M. F. Pflingsthor, S. Calinon, A. Turetta, C. Walen, D. Urbina, S. Govindaraj, P. Letier, X. Martinez, J. Salini, B. Chemisky, G. Indiveri, G. Casalino, P. Di Lillo, E. Simetti, D. De Palma, A. Birk, T. Fromm, C. Mueller, A. Tanwani, I. Havoutis, A. Caffaz, and L. Guilpain, "Dexterous undersea interventions with far distance

- onshore supervision: the dextron project,” in *10th IFAC Conference on Control Applications in Marine Systems*, vol. 49, no. 23, IFAC. Trondheim, Norway: Elsevier, 2016, pp. 414–419.
2. “ROBUST website,” <http://eu-robust.eu>, 2016, [Online; accessed 25-October-2016].
3. D. R. Yoerger, H. Schempf, and D. M. DiPietro, “Design and performance evaluation of an actively compliant underwater manipulator for full-ocean depth,” *Journal of robotic systems*, vol. 8, no. 3, pp. 371–392, 1991.
4. H. Schempf and D. Yoerger, “Coordinated vehicle/manipulator design and control issues for underwater telemanipulation,” in *IFAC Control Applications in Marine Systems (CAMS 92)*, Genova, Italy, April 1992.
5. D. M. Lane, J. B. C. Davies, G. Casalino, G. Bartolini, G. Cannata, G. Veruggio, M. Canals, C. Smith, D. J. O’Brien, M. Pickett, G. Robinson, D. Jones, E. Scott, A. Ferrara, D. Angelletti, M. Coccoli, R. Bono, P. Virgili, R. Pallas, and E. Gracia, “Amadeus: advanced manipulation for deep underwater sampling,” *IEEE Robot Autom Mag*, vol. 4, no. 4, pp. 34–45, 1997.
6. G. Casalino, D. Angeletti, T. Bozzo, and G. Marani, “Dexterous underwater object manipulation via multi-robot cooperating systems,” in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 4, IEEE. IEEE, 2001, pp. 3220–3225.
7. V. Rigaud, È. Coste-Manière, M.-J. Aldon, P. Probert, M. Perrier, P. Rives, D. Simon, D. Lang, J. Kiener, A. Casal *et al.*, “Union: underwater intelligent operation and navigation,” *Robotics & Automation Magazine, IEEE*, vol. 5, no. 1, pp. 25–35, 1998.
8. J. Yuh, S. Choi, C. Ikehara, G. Kim, G. McMurty, M. Ghasemi-Nejhad, N. Sarkar, and K. Sugihara, “Design of a semi-autonomous underwater vehicle for intervention missions (SAUVIM),” in *Underwater Technology, 1998. Proceedings of the 1998 International Symposium on*. Tokyo, Japan: IEEE, 1998, pp. 63–68.
9. G. Marani, S. K. Choi, and J. Yuh, “Underwater autonomous manipulation for intervention missions AUVs,” *Ocean Engineering*, vol. 36, pp. 15–23, 2008.
10. J. Evans, P. Redmond, C. Plakas, K. Hamilton, and D. Lane, “Autonomous docking for Intervention-AUVs using sonar and video-based real-time 3D pose estimation,” in *Oceans 2003*, vol. 4. IEEE, September 2003, pp. 2201–2210.
11. P. Marty *et al.*, “Alive: An autonomous light intervention vehicle,” in *Advances In Technology For Underwater Vehicles Conference, Oceanology International*, vol. 2004, 2004.
12. G. Casalino, E. Zereik, E. Simetti, S. Torelli, A. Sperindé, and A. Turetta, “A task and subsystem priority based control strategy for underwater floating manipulators,” in *IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles (NGCUV 2012)*, Porto, Portugal, April 2012, pp. 170–177.
13. —, “Agility for underwater floating manipulation task and subsystem priority based control strategy,” in *International Conference on Intelligent Robots and Systems (IROS 2012)*, Vilamoura, Portugal, September 2012, pp. 1772–1779.
14. E. Simetti, G. Casalino, S. Torelli, A. Sperindé, and A. Turetta, “Experimental results on task priority and dynamic programming based approach to underwater floating manipulation,” in *OCEANS 2013*, Bergen, Norway, June 2013.
15. —, “Floating underwater manipulation: Developed control methodology and experimental validation within the trident project,” *Journal of Field Robotics*, vol. 31, no. 3, pp. 364–385, May 2014.
16. G. Casalino, M. Caccia, A. Caiti, G. Antonelli, G. Indiveri, C. Melchiorri, and S. Caselli, “Maris: A national project on marine robotics for interventions,” in *Control and Automation (MED), 2014 22nd Mediterranean Conference of*. IEEE, 2014, pp. 864–869.
17. G. Casalino, M. Caccia, S. Caselli, C. Melchiorri, G. Antonelli, A. Caiti, G. Indiveri, G. Cannata, E. Simetti, S. Torelli, A. Sperindé, F. Wanderlingh, G. Muscolo, M. Bibuli, G. Bruzzone, E. Zereik, A. Odetti, E. Spirandelli, A. Ranieri, J. Aleotti, D. Lodi Rizzini, F. Oleari, F. Kallasi, G. Palli, U. Scarcia, L. Moriello, and E. Cataldi, “Underwater intervention robotics: An outline of the italian national project MARIS,” *Marine Technology Society Journal*, vol. 50, no. 4, pp. 98–107, 2016.
18. E. Simetti and G. Casalino, “Whole body control of a dual arm underwater vehicle manipulator system,” *Annual Reviews in Control*, vol. 40, pp. 191–200, 2015.

19. N. Manerikar, G. Casalino, E. Simetti, S. Torelli, and A. Sperindé, "On autonomous cooperative underwater floating manipulation systems," in *International Conference on Robotics and Automation (ICRA 15)*. Seattle, WA: IEEE, May 2015, pp. 523–528.
20. E. Simetti and G. Casalino, "Manipulation and transportation with cooperative underwater vehicle manipulator systems," *IEEE Journal of Oceanic Engineering*, 2016.
21. P. A. Di Lillo, E. Simetti, D. De Palma, E. Cataldi, G. Indiveri, G. Antonelli, and G. Casalino, "Advanced rov autonomy for efficient remote control in the DexROV project," *Marine Technology Society Journal*, vol. 50, no. 4, pp. 67–80, 2016.
22. A. Ben-Israel and T. Greville, *Generalized inverses: theory and applications*. Springer Verlag, 2003, vol. 15.
23. T. Perez and T. I. Fossen, "Kinematic models for manoeuvring and seakeeping of marine vessels," *Modeling, Identification and Control*, vol. 28, no. 1, pp. 19–30, 2007.
24. T. Yoshikawa, "Manipulability of robotic mechanisms," *Int. J. of Robotics Research*, vol. 4, no. 1, pp. 3–9, 1985.
25. B. Siciliano and J.-J. E. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," in *Proc. Fifth Int Advanced Robotics 'Robots in Unstructured Environments', 91 ICAR. Conf.* Pisa, Italy: IEEE, 1991, pp. 1211–1216.
26. E. Simetti and G. Casalino, "A novel practical technique to integrate inequality control objectives and task transitions in priority based control," *Journal of Intelligent & Robotic Systems*, vol. 84, no. 1, pp. 877–902, 2016.
27. K. L. Doty, C. Melchiorri, and C. Bonivento, "A theory of generalized inverses applied to robotics," *International Journal of Robotics Research*, vol. 12, no. 1, pp. 1–19, 1993.
28. N. Mansard, A. Remazeilles, and F. Chaumette, "Continuity of varying-feature-set control laws," *IEEE Trans. on Automatic Control*, vol. 54, no. 11, pp. 2493–2505, 2009.
29. G. Antonelli, G. Indiveri, and S. Chiaverini, "Prioritized closed-loop inverse kinematic algorithms for redundant robotic systems with velocity saturations," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*. IEEE, 2009, pp. 5892–5897.
30. L. L. Whitcomb and D. R. Yoerger, "Comparative experiments in the dynamics and model-based control of marine thrusters," in *OCEANS*, vol. 2. IEEE, 1995, pp. 1019–1028.
31. —, "Preliminary experiments in model-based thruster control for underwater vehicle positioning," *Oceanic Engineering, IEEE Journal of*, vol. 24, no. 4, pp. 495–506, 1999.
32. R. Bachmayer, L. L. Whitcomb, and M. A. Grosenbaugh, "An accurate four-quadrant nonlinear dynamical model for marine thrusters: Theory and experimental validation," *Oceanic Engineering, IEEE Journal of*, vol. 25, no. 1, pp. 146–159, 2000.
33. D. L. Rizzini, F. Kallasi, F. Oleari, and S. Caselli, "Investigation of vision-based underwater object detection with multiple datasets," *Int J Adv Robot Syst*, vol. 12, no. 77, pp. 1–13, 2015.
34. E. Simetti, S. Galeano, and G. Casalino, "Underwater vehicle manipulator systems: Control methodologies for inspection and maintenance tasks," in *OCEANS 16*. Shanghai, China: IEEE, 2016.
35. J. Yuh, "Design and control of autonomous underwater robots: A survey," *Autonomous Robots*, vol. 8, no. 1, pp. 7–24, 2000.
36. J. Yuh and M. West, "Underwater robotics," *Advanced Robotics*, vol. 15, no. 5, pp. 609–639, 2001.
37. G. Antonelli, *Underwater Robots*, ser. Springer Tracts in Advanced Robotics. Springer, 2014, vol. 96.
38. J. C. Kinsey, R. M. Eustice, and L. L. Whitcomb, "A survey of underwater vehicle navigation: Recent advances and new challenges," in *IFAC Conference of Manoeuvring and Control of Marine Craft*, vol. 88, 2006.
39. F. Bonin-Font, A. Ortiz, and G. Oliver, "Visual navigation for mobile robots: A survey," *Journal of intelligent and robotic systems*, vol. 53, no. 3, pp. 263–296, 2008.
40. J. Partan, J. Kurose, and B. N. Levine, "A survey of practical issues in underwater networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 11, no. 4, pp. 23–33, 2007.