Ocean Engineering 164 (2018) 40–54

Contents lists available at ScienceDirect

Ocean Engineering

journal homepage: www.elsevier.com/locate/oceaneng

Task priority control of underwater intervention systems: Theory and applications

E. Simetti[a,b,*], G. Casalino[a,b], F. Wanderlingh[a,b], M. Aicardi[a,b]

[a] Interuniversity Research Center on Integrated Systems for the Marine Environment, Via Opera Pia 13, 16145, Genova, Italy
[b] DIBRIS, University of Genova, Via Opera Pia 13, 16145, Genova, Italy

ARTICLE INFO

Keywords:
Task priority control
Intervention autonomous underwater vehicles
Remotely Operated Vehicles
Kinematic control
Underwater Vehicle Manipulator Systems

ABSTRACT

This paper presents a unifying task priority control architecture for underwater vehicle manipulator systems. The proposed control framework can be applied to different operative scenarios such as waypoint navigation, assisted teleoperation, interaction, landing and grasping. This work extends the results of the TRIDENT and MARIS projects, which were limited to the execution of grasping actions, to other applications taken from the DexROV and ROBUST projects. In particular, simulation results show how the control framework can be used, for example, for pipeline inspection scenarios and deep sea mining exploration.

1. Introduction

During the last 20 years, Autonomous Underwater Vehicles (AUVs) have been widely used as a tool for mapping the seafloor using optical and acoustic sensors, with applications to dam inspection (Ridao et al., 2010), marine geology (Wynn et al., 2014; Urabe et al., 2015) and underwater archaeology (Drap et al., 2008; Bingham et al., 2010) to mention but a few. After years of research, few autonomous platforms are already available in the market (Alvarez et al., 2009; Ribas et al., 2012), most of them able to perform side scan sonar and bathymetric multi-beam surveys. A recent survey (Yuh et al., 2011) reports a list of commercial platforms and applications of AUVs.

However, a large number of applications exist that go beyond the survey capabilities. A number of them stem from the oil and gas industry, such as the maintenance of submerged oil wells, cabled sensor networks and pipelines. In fact, Chevron has, since 2007, an on-going program for the employment of resident Intervention AUVs (I-AUVs) to provide better and more frequent inspections, earlier monitoring, and reduced field maintenance and development costs (Gilmour et al., 2012). Nowadays, these tasks require the use of work-class Remotely Operated Vehicles (ROVs) deployed from dynamic positioning vessels making them very expensive.

To respond to this increasing demand, research in marine robotics has started focusing on the development of Underwater Vehicle Manipulator Systems (UVMS). Since early 90s, different pioneering works were carried out on the control of compliant underwater manipulators (Yoerger et al., 1991), coordinated vehicle/arm control for teleoperation (Schempf and Yoerger, 1992), and during the ODIN (Choi et al., 1994) and OTTER (Wang et al., 1995) projects. Successively, within the UNION project (Rigaud et al., 1998) the first mechatronic assembly of an autonomous UVMS was achieved. Between 1993 and 2000, the AMADEUS project (Lane et al., 1997) developed grippers for underwater manipulation (Angeletti et al., 1998) and studied the problem of dual arm autonomous manipulation (Casalino et al., 2001), demonstrating these features in water tank experiments. The successive decade was characterized by different field demonstrations, among which we can cite the SWIMMER project (Evans et al., 2001) and the ALIVE project (Evans et al., 2003; Martyet al, 2004) that achieved autonomous docking into a seabed docking station or ROV-friendly panels, and the SAUVIM project (Yuh et al., 1998; Marani et al., 2008), which demonstrated the capability of searching and recovering an object whose position was roughly known a priori.

Concurrently, research in industrial robotics focused on how to effectively specify the control objectives of a robotic system, especially for redundant systems. The task-based control (Nakamura and Hanafusa, 1986), also known as operational space control (Khatib, 1987), defines control objectives in a coordinate system that is directly relevant to the task that needs to be performed, rather than in the generalized coordinates of the robotic system. Such an idea was immediately enhanced by the introduction of the concept of task priority (Nakamura, 1991). In that work, a primary task was executed, and a secondary task was accomplished (or attempted) only in the null space of the primary one, in order to guarantee the invariance of the main task w.r.t. (with respect to) the secondary one. This concept was later generalized to any number of task-priority levels in the seminal work (Siciliano and Slotine, 1991). However, it must be noted how the

* Corresponding author. DIBRIS, University of Genova, Via Opera Pia 13, 16145, Genova, Italy.
E-mail address: enrico.simetti@unige.it (E. Simetti).

https://doi.org/10.1016/j.oceaneng.2018.06.026
Received 7 December 2017; Received in revised form 24 April 2018; Accepted 9 June 2018
Available online 19 June 2018

position control of the end-effector was always the highest priority task, and safety tasks such as joint limits were only *attempted* at lower priority.

Given that an UVMS is a robotic system characterized by a high number of degrees of freedom, within the TRIDENT project (Sanz et al., 2013) these two research trends were merged. A novel task priority resolution mechanism, which managed equality and scalar-only inequality control objectives, was developed and exploited for the first time for the coordinated control of a floating base and a manipulator for performing autonomous floating intervention (Simetti et al., 2014). A blackbox recovery intervention was experimentally proved in a harbour environment. Shortly after TRIDENT, the PANDORA project has demonstrated autonomous free-floating valve-turning operation on a subsea panel using a learning by demonstration paradigm (Carrera et al., 2014) and a task-priority kinematic control approach (Cieslak et al., 2015). However the adopted task priority framework only dealt with equality control objectives and an ad-hoc solution was devised to manage the joint limit safety task, therefore it does not represent a general solution.

Successively, the concepts developed in the TRIDENT project were further enhanced within the MARIS project, with the definition of a task priority framework able to activate and deactivate equality/inequality control objectives of any dimension (i.e. not limited to scalar ones) depending on the system current needs (Simetti and Casalino, 2016). This feature allows the user to put safety and operational-enabling objectives at the highest priority, as they should be. Experiments in free floating grasping have been conducted, with multiple attempts to test the repeatability and robustness of the control (Simetti et al., 2017). The MARIS project also studied the extension of the control architecture to cooperative agents (Simetti and Casalino, 2017). Finally, recent studies on I-AUVs can be found in (Farivarnejad and Moosavian, 2014; Allotta et al., 2015; Conti et al., 2015).

Nowadays, the authors are involved in two other projects where UVMSs are employed, namely the EU H2020 DexROV and ROBUST projects. The DexROV project (Di Lillo et al., 2016) main goal is to delocalize on shore the manned support to ROV operations as much as possible, reducing the crew on board the support vessel and consequently the costs and risks of the whole operation. The delocalization is performed using satellite communications between the support ship and the remote control center. Therefore, only high level commands are sent through the satellite channels and forwarded to the ROV, which must execute them in a semi-autonomous manner. The ROBUST project (ROBUST website, 2016) aims to use robotic technologies for the exploration of deep-water mining sites, especially manganese nodule fields. The main idea is to perform in-situ measurements of the nodules, to identify if they contain Rare Earth Elements, which are particularly sought after in the market.

This paper presents a unifying control architecture for the control of UVMSs, both in the case of partial (assisted) teleoperation and fully autonomous operation. The architecture handles inequality control objectives without overconstraining the system, it coordinates the arm and vehicle movements thanks to a parallel task-priority inversion scheme (section 2.10), which is also suitable for multi-rate control, and it also manages the presence of vehicle underactuations to the best extent possible with the simple addition of a control task (section 2.9).

With respect to previous publications, this work does not consider only a grasping scenario, as it was the case of the TRIDENT and MARIS projects, but it extends the framework and shows its flexibility in tackling different operative scenarios, presenting the most recent results of the DexROV and ROBUST projects. In fact, the same architecture can execute the required operations as long as the corresponding *actions* are defined, with the advantage of having a unique controller at the kinematic level, simplifying the overall implementation and allowing greater modularity, as many control *tasks* are common to more than one *action*. The present work's contributions are listed as follows:

C_1 It presents, in a self contained way, all the properties of the proposed task priority framework, omitting only the mathematical details presented in (Simetti and Casalino, 2016).
C_2 It shows how the proposed approach can be used for both autonomous operation and assisted teleoperation, either if the user wants to control some of the degrees of freedom, or even if he/she desires to control only the end-effector. This is a requirement coming from the DexROV project;
C_3 It presents the integration of force regulation at the kinematic level, for example to carry out the inspection of a pipeline as needed in the DexROV project, validated through dynamic simulations, including different rates for the kinematic and dynamic control layers, vehicle added masses and Coriolis effects, thruster dynamics and actuator saturations;
C_4 It discusses how a safe navigation action and how landing in front of a specific target can be implemented, again validated through dynamic simulations, as they are two of the operations needed in the ROBUST project for deep-mining exploration.

The paper is organized as follows. Section 2 recaps the theory behind the developed task priority control framework. Then, the flexibility of the proposed architecture in tackling many different kind of applications, spanning all the aforementioned projects, is highlighted in section 3. In section 4, the most recent simulation results of the DexROV and ROBUST projects are shown. Section 5 presents the current open problems and research trends. Finally, some conclusions are given in section 6.

2. The control framework: theory

The control framework presented in this paper is based on the cascade of blocks shown in Fig. 1. In particular, the architecture is constituted by three main blocks:

1. The Mission Manager is in charge of supervising the execution of the current *mission*, and generates the corresponding *actions* to be executed by the Kinematic Control Layer. As it will be explained later in section 2.4, an action is any prioritized list of control objectives to be concurrently achieved, and a mission is a sequence (or graph) of actions.
2. The Kinematic Control Layer (KCL) implements the proposed task priority control framework, and is in charge of reactively accomplishing the *control objectives* that make up the current action to be executed, generating the desired system velocity vector.
3. The Dynamic Control Layer (DCL) tracks the desired system velocity vector by generating appropriate force/torques commands for the vehicle and the manipulator.

The paper focuses on the Kinematic Control Layer, since it is the one implementing the proposed task priority approach. The interfaces with the higher level (Mission Manager) and the lower one (DCL) are highlighted whenever relevant.

2.1. General definitions

Let us introduce two definitions that will be used thoroughly in this paper:

- The system configuration vector $\boldsymbol{c} \in \mathbb{R}^n$ of the UVMS as $\boldsymbol{c} \triangleq [\boldsymbol{q} \quad \boldsymbol{\eta}]^T$, where $\boldsymbol{q} \in \mathbb{R}^l$ 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 $\boldsymbol{\eta}_1 \triangleq [x \quad y \quad z]^T$, with components in the inertial frame $\langle w \rangle$, and the orientation vector $\boldsymbol{\eta}_2 \triangleq [\phi \quad \theta \quad \psi]^T$, the latter expressed in terms of the three angles roll, pitch, yaw (applied in the yaw-pitch-roll sequence (Perez and Fossen, 2007)). The possible singularity arising when the pitch angle is near $\pi/2$ is

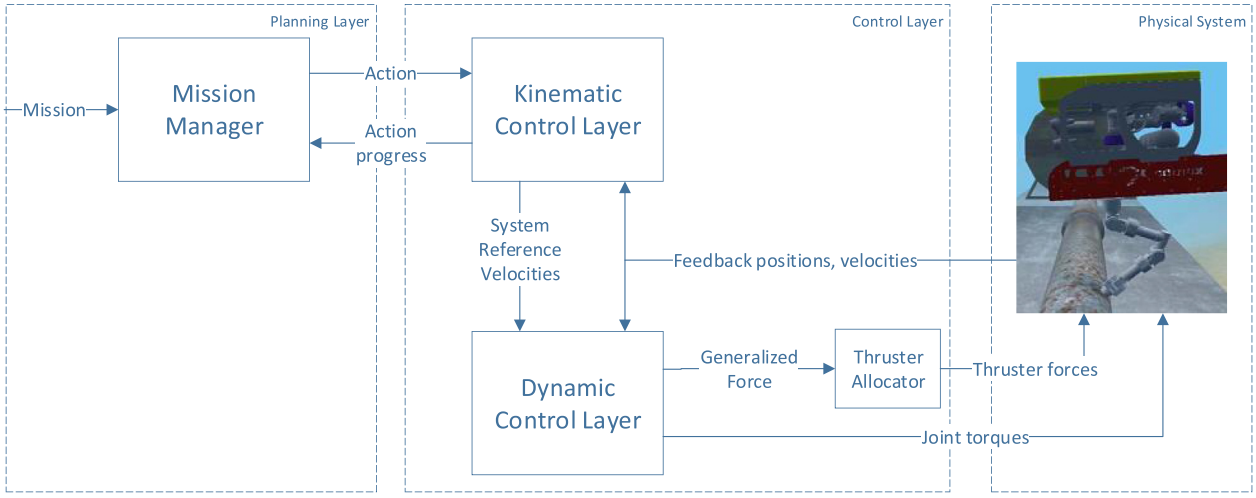


Fig. 1. The overall architecture: the Kinematic Control Layer is the one implementing the proposed task priority approach, executing the current action scheduled by a Mission Manager; the output of the Kinematic Control Layer are the system velocities, tracked by the underlying Dynamic Control Layer.

handled through a dedicated control objective, as it will be shown later in section 3. From the above definitions it results $n = l + 6$;

- The system velocity vector $\dot{\boldsymbol{y}} \in \mathbb{R}^n$ of the UVMS as $\dot{\boldsymbol{y}} \triangleq [\dot{\boldsymbol{q}} \;\; \boldsymbol{\nu}]^T$, where $\dot{\boldsymbol{q}} \in \mathbb{R}^l$ are the arm joint velocities and $\boldsymbol{\nu} \in \mathbb{R}^6$ is the stacked vector of the vehicle linear velocity vector $\boldsymbol{\nu}_1 \triangleq [\dot{x} \;\; \dot{y} \;\; \dot{z}]^T$ and the vehicle angular velocity vector $\boldsymbol{\nu}_2 \triangleq [p \;\; q \;\; r]^T$, both with components in the vehicle frame $\langle v \rangle$. Note that the angular velocities are not the derivatives of the Euler angles used in $\boldsymbol{\eta}_2$. For the time being, this work will be developed with the assumption of a vehicle fully actuated in its 6 degrees of freedom, therefore the system velocity vector will coincide with the control vector to be used at the kinematic level. Section 2.9 will show how to relax this assumption.

2.2. Control objectives: definition, categories and examples

Let us start the discussion by considering what the robotic system needs to achieve. Mathematically speaking, let us consider any scalar variable $x(\boldsymbol{c})$ dependent on the current configuration of the robot and the environment. Then, for this variable, two broad classes of objectives can be defined:

1. The requirement, for $t \to \infty$, that $x(\boldsymbol{c}) = x_0$ is called a *scalar equality control objective.*
2. The requirement, for $t \to \infty$, that $x(\boldsymbol{c}) < x_{max}$ or $x(\boldsymbol{c}) > x_{min}$ is called a *scalar inequality control objective.*

Control objectives can be further divided in different categories, depending on their purpose and, as it will be clear later, on their priority:

- physical constraints objectives, e.g. 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 field of view
- action oriented objectives, i.e. what the system ultimately needs to execute to accomplish the current user defined action, e.g. the end-effector grasping an object, reaching the desired position with the vehicle, etc.
- optimization objectives, i.e. control objectives such as maintaining an arm preferred pose, or minimizing the vehicle velocity, which express a *preference*, but whose achievement is not required for the execution of the action.

2.3. Reactive and non-reactive control tasks

To each scalar control objective there is always associated a *feedback reference rate* $\dot{\bar{x}}$. It represents the closed-loop rate control law that, if tracked, would drive the associated variable $x(\boldsymbol{c})$ toward an arbitrary point x^* where the objective is satisfied. A possible feedback reference rate is the following one:

$$\dot{\bar{x}}(x) \triangleq \gamma(x^* - x), \;\; \gamma > 0, \tag{1}$$

where γ is a positive gain proportional to the desired convergence rate for the considered variable. The need for tracking the desired $\dot{\bar{x}}$ is termed a *reactive control task*.

Sometimes, a control task is defined directly in a certain task velocity space, without having an associated control objective. As an example, consider the case where a human operator wants to control the end-effector by generating velocity references through a joystick. In such a case, the reference rate $\dot{\bar{x}}$ is generated by the user, rather than being the output of a feedback control loop as in (1). In this case, tracking the desired $\dot{\bar{x}}$ is termed a *non-reactive control task*.

In both situations, to track the desired reference rate, it is necessary to act upon the system velocity vector through a Jacobian relationship of the type

$$\dot{x} = \boldsymbol{g}^T \dot{\boldsymbol{y}}, \tag{2}$$

where the vector $\boldsymbol{g} \in \mathbb{R}^n$ is the Jacobian of the considered task.

2.4. Actions

In the terminology of this work, an *action* is a prioritized list of control tasks to be *concurrently* managed. The meaning is that lower priority tasks should not influence the fulfilment of higher priority ones. If two or more scalar control tasks are assigned to the same priority level, then they are grouped into a (multidimensional) control task. For example, for a robotic manipulator, a manipulation action $\mathscr{A}_m$ could be represented by following list of prioritized control objectives.

1. Arm joint limits
2. Arm manipulability
3. End-effector position control
4. End-effector attitude control
5. Arm preferred shape

where it is natural to see *safety* objectives such as joint limits at the highest priority. Further remarks on how a list composing an action can

be constructed will be given in section 2.8 and section 3. For the time being, let us assume that a list of prioritized control tasks has been established.

2.5. Activation functions

As remarked in the introduction, one of the important points of this control architecture is the ability of handling inequality control objectives without overconstraining the system. To this aim, let us define an *activation function*

$$a^i(x) \in [0,1] \tag{3}$$

as a continuous sigmoidal function of the objective variable $x(\boldsymbol{c})$, which assumes zero values within the validity region of the associated inequality objective. For example, assuming an objective of the class $x(\boldsymbol{c}) > x_{min}$:

$$a^i(x) \triangleq \begin{cases} 1, & x(\boldsymbol{c}) < x_{min} \\ s(x), & x_{min} \le x(\boldsymbol{c}) \le x_{min}+\Delta \\ 0, & x(\boldsymbol{c}) > x_{min}+\Delta \end{cases} \tag{4}$$

where $s(x)$ is any smooth, strictly decreasing function joining the two extrema, and Δ is a value that creates a buffer zone, where the inequality is already satisfied, but the activation value is still greater than zero, preventing chattering problems around the inequality control objective threshold x_{min}. A similar function can be defined for objectives of the class $x(\boldsymbol{c}) < x_{max}$:

Clearly, for both reactive control task associated to an equality control objective and for non-reactive control tasks, the activation function is always the unitary one, i. e $a^i(x) \equiv 1$. From now on, the distinction between reactive and non-reactive control tasks will be dropped, and the generic term control task will be used, unless otherwise specified.

2.6. Task Priority Inverse Kinematics

Given an action $\mathscr{A}$, for each priority level k, the following matrices and vectors are consequently defined:

- $\dot{\bar{\boldsymbol{x}}}_k \triangleq [\dot{\bar{x}}_{1,k} \ \cdots \ \dot{\bar{x}}_{m_k,k}]^T$ is the stacked vector of all the reference rates of the scalar control tasks, where m_k is the number of scalar tasks at the priority level k;
- $\boldsymbol{J}_k$ is the Jacobian relationship expressing the current rate-of-change of the k-th task vector $[\dot{x}_{1,k} \ \cdots \ \dot{x}_{m_k,k}]^T$ w.r.t. the system velocity vector $\dot{\boldsymbol{y}}$, simply resulting from the stacking of the row vector ones in (2);
- $\boldsymbol{A}_k \triangleq \mathrm{diag}(a_{1,k},\cdots,a_{m_k,k})$ is the diagonal matrix of all the activation functions described in (3).

To find the system velocity reference vector $\dot{\bar{\boldsymbol{y}}}$ that satisfies the above expressed priority requirements between the different tasks, the following sequence of nested minimization problems needs to be solved, corresponding to the so called Task Priority Inverse Kinematics (TPIK):

$$S_k \triangleq \left\{ \mathrm{argR}\underline{-}\min_{\dot{\bar{\boldsymbol{y}}} \in S_{k-1}} ||\boldsymbol{A}_k(\dot{\bar{\boldsymbol{x}}}_k - \boldsymbol{J}_k\dot{\bar{\boldsymbol{y}}})||^2 \right\}, \ k = 1,2, \ ...,N, \tag{5}$$

where S_{k-1} is the manifold of solutions of all the previous tasks in the hierarchy, $S_0 \triangleq \mathbb{R}^n$, N is the total number of priority levels and finally where the notation R− min is introduced for underlining that each minimization is performed via the specialized form of (Simetti and Casalino, 2016). Within such a specialized framework (named *iCAT: inequality Constraints And Task transitions*), the above TPIK problem (5) results in the following algorithm, to be initialized with $\boldsymbol{\rho}_0 = 0$, $\boldsymbol{Q}_0 = \boldsymbol{I}$ and then, for $k = 1, \ ...,N$

$$\boldsymbol{W}_k = \boldsymbol{J}_k\boldsymbol{Q}_{k-1}(\boldsymbol{J}_k\boldsymbol{Q}_{k-1})^{\#,\boldsymbol{A}_k,\boldsymbol{Q}_{k-1}}, \tag{6}$$

$$\boldsymbol{Q}_k = \boldsymbol{Q}_{k-1}\left(\boldsymbol{I} - (\boldsymbol{J}_k\boldsymbol{Q}_{k-1})^{\#,\boldsymbol{A}_k,\boldsymbol{I}}\boldsymbol{J}_k\boldsymbol{Q}_{k-1}\right),$$

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

where the special pseudo inverse operator $(\cdot)^{\#,\boldsymbol{A},\boldsymbol{Q}}$ has been introduced in (Simetti and Casalino, 2016) to cope with certain invariance problems arising in (5) and where the function Sat(·) implements the management of control variable saturations, accordingly with what suggested in (Antonelli et al., 2009). At the end of the above iterative process, the final system velocity vector is therefore $\dot{\bar{\boldsymbol{y}}} = \boldsymbol{\rho}_N$. The interested reader can find all the relevant details of this procedure in (Simetti and Casalino, 2016), where a comparison with other task priority approaches, as for instance those appearing in (Kanoun et al., 2011; Escande et al., 2014; Moe et al., 2016), is given.

2.7. Building a mission through action sequencing

The concept of an action represents a nice way to embed a certain complexity within the KCL, exposing a simple interface toward the Mission Manager. An important point to develop is the transition between two actions, once an action change has been commanded by the Mission Manager. Let us consider the following two actions $\mathscr{A}_1 = A \prec B, C, D$ and $\mathscr{A}_2 = A \prec D \prec C, E$, composed by objectives abstractly labelled with alphabetic letters, where $A \prec B$ denotes that A has higher priority than B and where B, C means that B has equal priority to C. Now consider the merged list $\mathscr{U} = A \prec D_1 \prec B, C, D_2, E$ where D_1 and D_2 represent the same objective D, but with a different priority. It is clear that, through insertion/deletion of some of the entries, the two original lists can be reconstructed. For example, $\mathscr{A}_1$ can be obtained by removing D_1 and E from the unified list. Conversely, $\mathscr{A}_2$ can be obtained by removing the control objectives D_2 and B. To enable such transitions, the activation function (3) is modified to become

$$a(x, \boldsymbol{p}) = a^i(x)a^p(\boldsymbol{p}), \tag{7}$$

where $a^p(\boldsymbol{p}) \in [0,1]$ is a continuous sigmoidal function of a vector of parameters $\boldsymbol{p}$ that contains the previous and current action being executed, and the time elapsed in the current action, to obtain the desired activation/deactivation smooth transition. Of course, this new activation function $a^p(\boldsymbol{p})$ applies to both reactive and non-reactive control tasks since it does not depend on the variable $x(\boldsymbol{c})$. An example of transition from action $\mathscr{A}_1$ to $\mathscr{A}_2$ is reported in Table 1, where $s_{dec}(t, t_{max})$ represents a sigmoidal function whose values goes from 1 to 0 as the time t within the current action goes from 0 to the threshold time t_{max}, after which the transition between $\mathscr{A}_1$ and $\mathscr{A}_2$ is completed. Similarly, $s_{inc}(t, t_{max})$ goes from 0 to 1.

2.8. Remarks on conflicting objectives

An action is concluded successfully when its *action defining* objectives are satisfied. Thus, a necessary condition to guarantee that an action can be terminated is the existence of a *non-void* set of configurations where all high priority and action-defining objectives are *contemporaneously* satisfied. The presence of such a *non-conflicting zone* is, however, not a sufficient condition, as it does not imply its reachability in correspondence of all system initial configurations starting

Table 1
Switching between actions.

Control Task	Previous Action $\mathscr{A}_1$	Current Action $\mathscr{A}_2$	$a^p(\boldsymbol{p})$
A	active	active	1
D_1	inactive	active	$s_{inc}(t, t_{max})$
B	active	inactive	$s_{dec}(t, t_{max})$
C	active	active	1
D_2	active	inactive	$s_{dec}(t, t_{max})$
E	inactive	active	$s_{inc}(t, t_{max})$

outside it. Such a reachability problem affects all the task priority strategies, and in general all reactive control approaches.

In the proposed architecture, this problem needs to be addressed at a higher level, within the Mission Manager block. The creation of different control actions helps in defining a set of well structured blocks, that can be sequenced by the Mission Manager. In case of inability of completing the required action, the Mission Manager should devise an alternative solution, e.g. a different sequencing of actions or different action parameters (i.e. a new path to follow). A detailed discussion of how the Mission Manager could be implemented falls outside the scope of the current work.

2.9. Control in presence of vehicle underactuations

Until now the vehicle was assumed fully actuated in all its degrees of freedom, as it usually the case for work-class ROVs that need to perform manipulation activities. In any case, the proposed architecture can also be used for managing underactuated vehicles at the best extent possible, as already shown in (Simetti et al., 2017).

Let us consider a vehicle does not have thrusters allowing to control the angular velocities along the vehicle x and y axes (roll and pitch), which is often the case for small size vehicles. To cope with underactuation, let us add at the *top of the hierarchy* a *non-reactive* control task structured as follows:

- Its reference rate $\dot{\bar{\boldsymbol{x}}}$ is composed by the vector of the *measured* vehicle velocities (all vehicle velocities, including the non actuated ones, are assumed measurable). Therefore, for such a task $\bar{\boldsymbol{x}} = [\boldsymbol{v}_1^T \;\; p \;\; q \;\; r]^T$, where $\boldsymbol{v}_2$ has been explicitly written in terms of its components;
- Its Jacobian is simply $[0_{6\times l} \;\; \boldsymbol{I}_{6\times 6}]$, where the first block of zeroes regards the arm joint velocities that are kinematically non-influencing the vehicle ones;
- Its activation function, in the considered roll and pitch underactuation example, is the constant diagonal matrix diag(0,0,0,1,1,0) selecting the non-actuated velocities only.

It can be easily seen that, at the end of the procedure, the values of $\dot{\bar{\boldsymbol{y}}}$ corresponding to the non-actuated degrees of freedom are equal to the measured ones. Hence, the resulting vector $\dot{\bar{\boldsymbol{y}}}$ is the *optimal* one in correspondence of such non-actuated degrees of freedom velocities. In general, with a proper choice of the activation function, the proposed architecture can actually cope with any kind of vehicle underactuation; thus, possibly, even underactuation caused by actuators failures, provided they are detected and the vehicle velocities remain all measurable.

2.10. Arm-vehicle coordination scheme

An important feature of the proposed architecture is the arm-vehicle motion coordination, since if manipulator and vehicle motions are not kinematically decoupled, then disturbances of the floating base (e.g. due to nonlinear properties of thrusters (Whitcomb and Yoerger, 1995; Whitcomb and Yoerger, 1999; Bachmayer et al., 2000), or the large inertia of the vehicle) would propagate through the coupled kinematics immediately to the end effector of the manipulator (Simetti et al., 2017).

To cope with this problem, the idea is to have two optimizations (6) running in parallel as depicted in Fig. 2, having as input the same action $\mathscr{A}$. The first optimization (TPIK 1) implements the task hiearchy of the action $\mathscr{A}$ and its output is used to generate the reference velocity for the vehicle. The second optimization (TPIK 2) considers the vehicle as *totally non-controllable*. Therefore, the non-reactive control task presented in 2.9 is used and all the vehicle degrees of freedom are initialized with the corresponding *measured* velocities. The vehicle constrained velocity task is placed on top of the task hierarchy of action $\mathscr{A}$, as depicted in the TPIK 2 block of Fig. 2. In such an optimization, de-facto only the manipulator variables are subject to be optimized, hence the outputs of this procedure are the *optimal* joint velocities in correspondence of the *measured* vehicle velocity. More details on this coordination technique are shown in (Simetti et al., 2017), together with several experiments highlighting its importance for floating manipulation. Finally, note how this parallel strategy for arm-vehicle coordinated control is also suitable for implementing multi-rate control of the two subsystems, as TPIK 1 can be run at the vehicle control frequency, while TPIK 2 can be updated at the higher arm control rate.

3. Applications

This sections highlights different operative scenarios where the control of the UVMS can take advantage of the proposed architecture, showing its flexibility in being adapted to many applications. In the following, the actions to carry out the specific operations will be presented using the following compact notation:

- [R/NR, I/E, C/S/P/AD/O] Name of the task/objective;

where

- R/NR specifies if the task is reactive or non-reactive;
- I/E specifies if the task is of inequality or equality type;
- C/S/P/AD/O specifies if the category of the task, i.e. constraint, safety, prerequisite, action-defining, optimization.

Finally, all the actions that will be presented in the following sections can be thought as having the underactuation constraint task, presented in section 2.9, at the top of the hierarchy, i.e.

- [NR, E, C] Vehicle underactuation constraint.

3.1. Safe waypoint navigation

Let us consider the requirement of having the UVMS reaching a new position, expressed for example in global coordinates through an ultra-short or long baseline system. In such a scenario, other than having the vehicle in the required position, there are a number of other objectives that the system needs to satisfy, which can be summarized in the following list (highest to lowest priority objective):

1. [R, I, S] Vehicle minimum altitude;
2. [R, I, S] Vehicle obstacle avoidance;
3. [R, I, S] Vehicle horizontal attitude;
4. [R, I, P] Vehicle auto-heading;
5. [R, E, AD] Vehicle position (x, y and depth).

To avoid excessive energy consumption to keep a precise positioning, the vehicle position control objective could be implemented as an inequality one, introducing a dead band zone. An hysteresis in the activation of the task could be also introduced to prevent excessive chattering around the threshold due to the vehicle dynamics. The altitude task and obstacle avoidance one are written separately just because they usually work with different sensors, i.e. an altimeter and a forward looking sonar. The horizontal attitude control objective requires the vehicle to be parallel to the horizontal plane, reducing the misalignment vector between its vertical axis and the absolute one (Simetti and Casalino, 2017). This control objective, needed only for fully actuated vehicles, avoids the possible singularity arising when the pitch angle is near $\pi/2$ in the roll-pitch-yaw representation.

The above defined navigation to a waypoint action allows us to make a few important remarks:

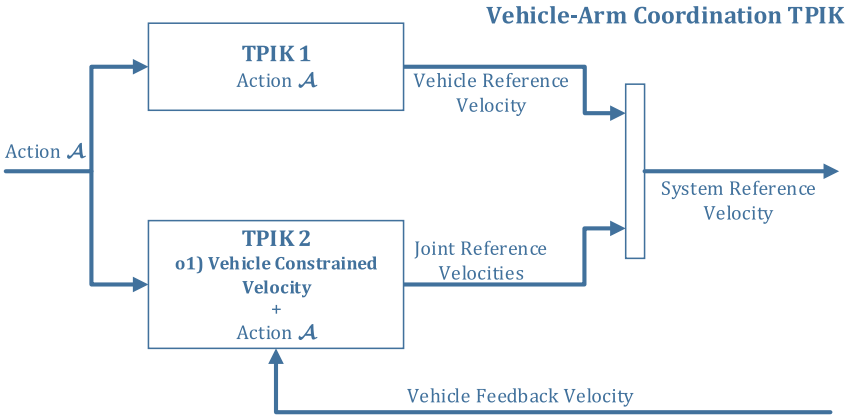


Fig. 2. A detailed view of the Kinematic Control Layer, with the two Task Priority Inverse Kinematics blocks highlighted, implementing the arm-vehicle coordination scheme.

- There is a natural conflict between certain objectives. In the above example, the minimum altitude might conflict with the auto-depth objective, which is implicitly considered in the vehicle position one. However, the altitude objective is activated *only* when really necessary.
- The specified higher priority of the altitude task ensures that the altitude is maintained at the cost of the depth setpoint *whenever necessary*. The resulting task-priority behaviour is depicted in Fig. 3.
- Thanks to the possibility of activating and deactivating tasks, the priority of the objectives can follow a natural order without over-constraining the system: safety objectives first, then operational enabling objectives (auto-heading), then actual action-defining objectives (vehicle position).

3.2. Assisted teleoperation of UVMS degrees of freedom

The flexibility of the proposed control framework also allows one to implement an assisted teleoperation of the UVMS, in a sort of advanced ROV operational mode. Typically, the ROV operator controls the vehicle through body frame velocities, i.e. using the surge, sway, heave and angular velocity commands, while he/she also sets the desired joint velocities for the manipulator. This means that the system velocity vector $\dot{\boldsymbol{y}}$ is exactly the variable that contains the desired inputs for the operator. Therefore, a non-reactive control task for the manual control can be easily defined:

- The Jacobian relationship is simply the identity matrix;
- The activation function is set to one in correspondence of the degrees of freedom that needs to be manually controlled, zero otherwise;
- The reference rate vector $\dot{\bar{\boldsymbol{x}}}$ is initialized with the values generated by the operator's console.

A control task so defined, placed at the top of the control task hierarchy, will directly set the corresponding control vector to the desired one provided by the operator. It is important to note that if only some of the degrees of freedom are activated for manual control, all the others can be used by the control system for the achievement of the remaining control objectives. Furthermore, the task priority approach will automatically take into account the influence of the manually controlled degrees of freedom on the remaining control tasks.

To clarify, let us consider the following example action of safe navigation with manual/assisted teleoperation, characterized by the following list of tasks:

1. [NR, E, AD] Manual control;
2. [R, I, S] Vehicle minimum altitude;
3. [R, I, S] Vehicle obstacle avoidance;
4. [R, I, S] Vehicle horizontal attitude;
5. [R, I, P] Vehicle auto-heading;
6. [R, I, AD] Vehicle position (x, y and depth);

which is the same one as the previous section, with the addition of the non-reactive "Manual control" task.

Let us suppose that the operator configures the manual control task, through a proper choice of the activation function, to take control of surge, sway and angular velocity along the z axis, leaving the degrees of freedom of heave, and the rotation along the x and y axes to the control system. In such a configuration, if the operator moves the vehicle with a certain surge velocity, and the pitch of the vehicle points slightly down, the depth will increase and altitude decrease. However, the control system, through the vehicle position depth objective, will automatically regulate the heave motion to compensate for the possible increase in depth. Finally, the altitude control objective will prevent the operator for getting too close to the seafloor, at the expenses of the depth regulation task, accordingly with the desired priority and safety requirements.

3.3. Grasping

Another action that can be defined with the proposed architecture regards the grasping of an object. In this case, the action involves also

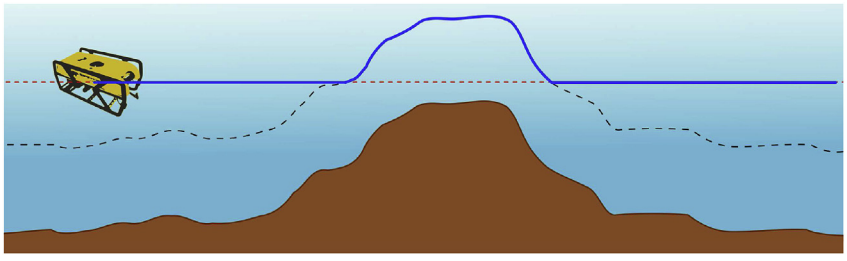


Fig. 3. Desired behaviour of the safe waypoint navigation action: (brown solid) sea-floor depth, (black-dashed) minimum altitude, (red-dashed) depth desired value, (blue) resulting behaviour of the task priority approach, considering the higher priority altitude objective and the depth one. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

the manipulator, therefore the number of control objectives is naturally higher than in the previous examples. The following hierarchy could be used to define such an action:

1. [R, I, S] Vehicle minimum altitude;
2. [R, I, S] Vehicle obstacle avoidance;
3. [R, I, S] Arm joint limits;
4. [R, I, P] Arm manipulability;
5. [R, I, P] Camera centering;
6. [R, I, P] Camera-arm occlusions;
7. [R, E, AD] End-effector position control;
8. [R, E, AD] End-effector attitude control;
9. [R, I, O] Arm preferred shape;
10. [NR, E, O] Vehicle motion minimization;

where the need of avoiding occlusions between the manipulator and the camera system providing the feedback as well as the need of maintaining the object in the camera frame are clearly prerequisites for guiding the end-effector on top of the object, therefore they have higher priority. The arm manipulability task aims at maintaining the manipulability index μ (Yoshikawa, 1985)

$$\mu \triangleq \sqrt{\det(\boldsymbol{J}\boldsymbol{J}^T)} \tag{8}$$

above a minimum value. The index represents a distance of the arm from its singular postures. If this index is maintained away from zero, then singular postures are avoided. Finally, it might desirable to move the vehicle as least as possible, for the same considerations made in section 2.10, hence the final motion minimization task. The implementation of an action of this kind has been done within the already mentioned TRIDENT and MARIS projects, for which experimental results such as the one shown in Fig. 4 are detailed in (Simetti et al., 2014) and (Simetti et al., 2017) respectively. For such a reason, they will not be covered in this work.

3.4. Assisted end-effector teleoperation

A further advanced assisted ROV application consists in considering a teleoperation of the end-effector of the manipulator, rather than a set of degrees of freedom of the UVMS, as done in section 3.2.

In this scenario, the operator generates reference velocities for the end-effector, and the control system assists him in a set of prerequisite and safety objectives. This can lead to the creation of an action similar to the one presented in section 3.3:

1. [R, I, S] Vehicle minimum altitude;
2. [R, I, S] Vehicle obstacle avoidance;
3. [R, I, S] Arm joint limits;
4. [R, I, P] Arm manipulability;
5. [R, I, P] Camera centering (if visual feedback is available);
6. [R, I, P] Camera-arm occlusions (if visual feedback is available);

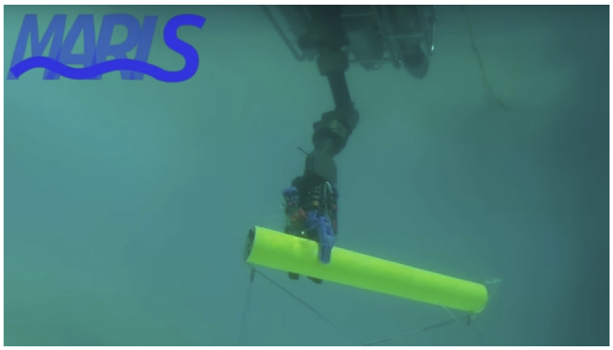


Fig. 4. The MARIS UVMS during one of the successful grasps.

Table 2
Vehicle parameters are a scaled down version of (Caccia et al., 2000).

Parameter	Value
Mass in air [kg]	300
$\boldsymbol{M}_A$	diag ([158.4 137.1 171.3 15 20 20])
$\boldsymbol{D}$	diag ([50 50 44 24 20 20])
$\boldsymbol{D}_n$	diag ([320 320 430 31 40 40])

Table 3
UMA arm parameters (values are taken from the Graal Tech UMA commercial arm, see (Ribas et al., 2015)).

Parameter	Value
Mass in air [kg]	[8 6 4 3 3 3 1.5]
Motors inertia [kg m^2]	[$3.5e^{-6}$ $1.3e^{-5}$ $3.5e^{-6}$ $3.5e^{-6}$ $5.7e^{-7}$ $5.7e^{-7}$ $5.7e^{-7}$]
Motors reductions	[2328 1200 1987.32 1786.84 1987.32 1069.565 563.125]
Maximum torques [Nm]	[45 125 50 50 10 10 4]

7. [NR, E, AD] End-effector linear velocity teleoperation;
8. [NR, E, AD] End-effector angular velocity teleoperation;
9. [R, I, O] Arm preferred shape;
10. [NR, E, O] Vehicle motion minimization;

In this case, the camera objectives are included only if an acoustic/vision feedback is available. Let us remark how the control loop to do the manipulative operation is now "closed" by the human operator, hence the end-effector tasks are of non-reactive type.

3.5. Inspection

Another interesting action that can be defined within the proposed architecture stems from one of the reference missions of the DexROV project and regards the inspection of a pipeline. An electromagnetic sensor placed at the end-effector needs to be put in contact with the pipeline's weld and follow it in order to discover any possible cracks or leaks. This application is very interesting since it requires interaction between the UVMS and the environment, and allows us to show how the proposed task priority framework can enforce kinematic constraints, as well as, despite being at the kinematic level, manage interaction tasks.

Let us consider a surface, characterized by its normal $\boldsymbol{n} \in \mathbb{R}^3$. The contact of the end-effector with the surface creates two major requirements:

1. The end-effector can freely move only tangentially to the surface;
2. Any velocity requirement along the surface normal, in the direction of the surface, will actually result in a force exerted on the surface, which can be exploited to maintain the contact with it.

For these reasons, the control algorithm should take the surface constraint into account at the highest priority. Let us call $\boldsymbol{J}_{el} \in \mathbb{R}^{3\times n}$ the total linear Jacobian of the end-effector frame, comprehensive of both the vehicle and the manipulator contributions and let us consider the following task:

$$S_1 \triangleq \left\{ \operatorname{argR}-\min_{\bar{\dot{\boldsymbol{y}}}} \|(\dot{\bar{x}}_f - \boldsymbol{n}^T\boldsymbol{J}_{el}\bar{\dot{\boldsymbol{y}}})\|^2 \right\}, \tag{9}$$

where the task reference $\dot{\bar{x}}_f$ can be exploited to modulate the velocity requirement along the surface's normal, which will result in the desired interaction necessary to maintain contact with the surface through force regulation. For example, $\dot{\bar{x}}_f$ can be chosen as the output of any outer interaction force control loop, once reported at the linear velocity

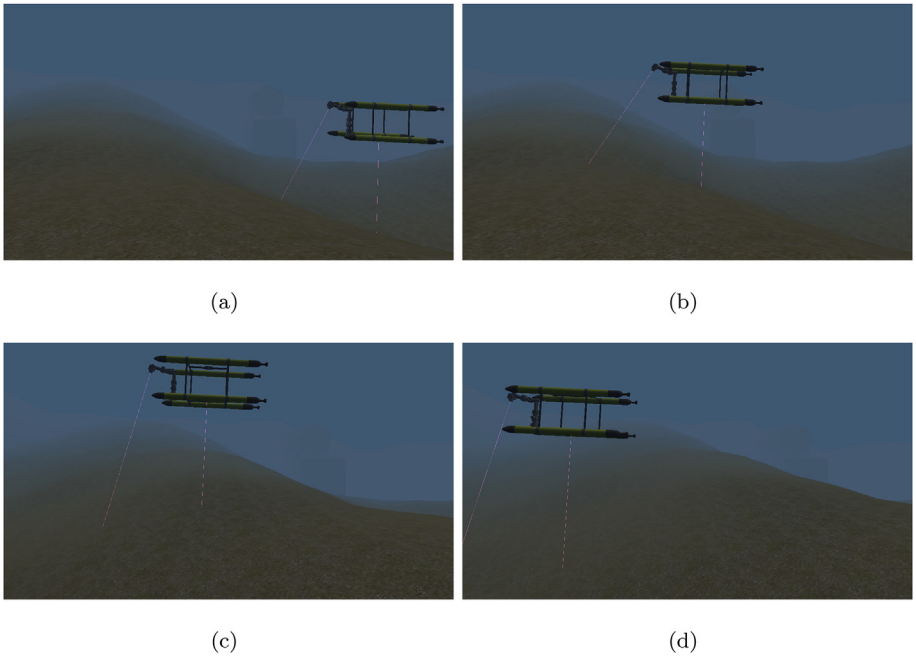


Fig. 5. Screenshots of the UVMS as it performs the safe navigation action, successfully maintaining the required minimum altitude.

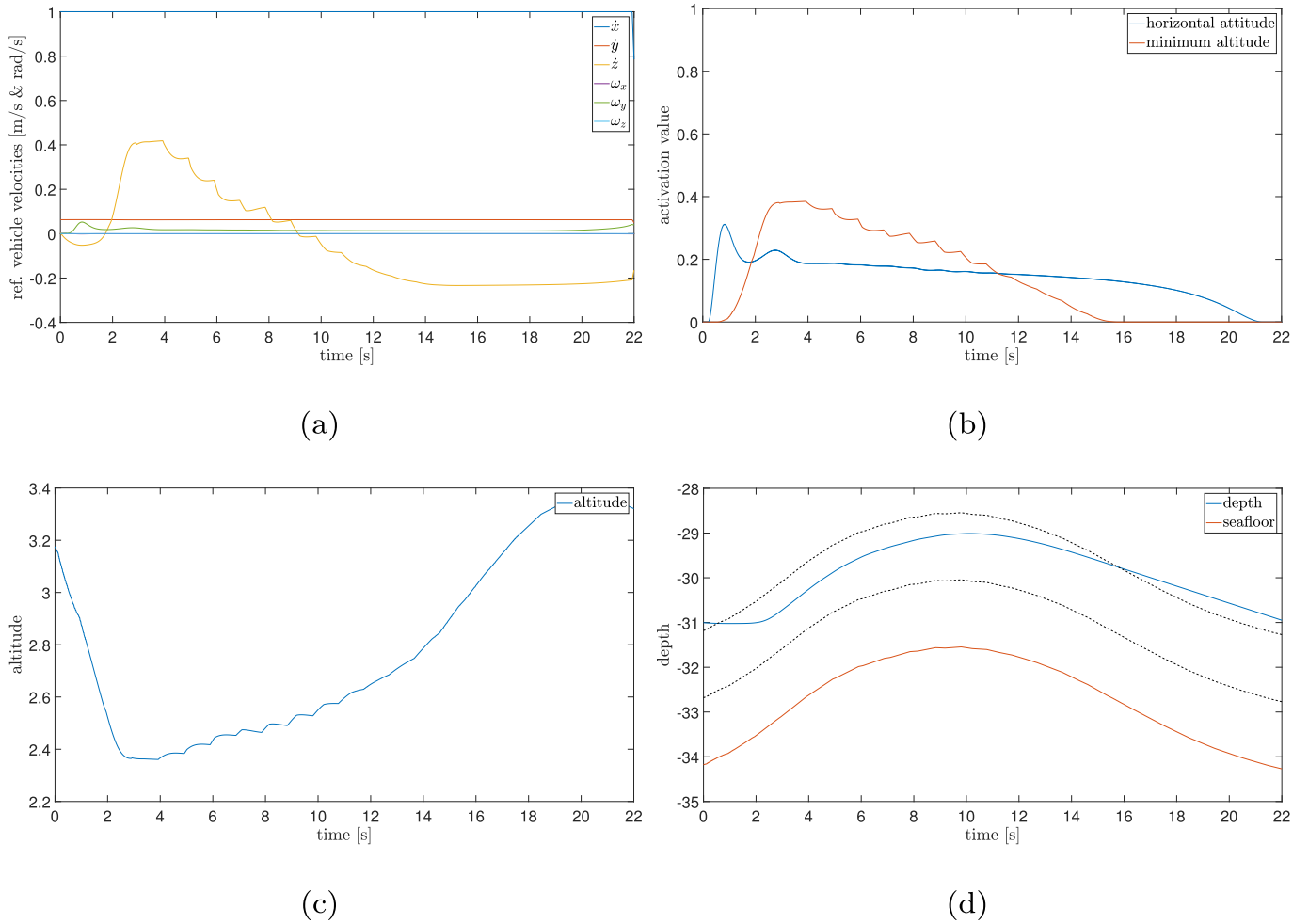


Fig. 6. Safe navigation action simulation results: (a) vehicle reference velocities (saturation values at 0.2 rad/s and 1 m/s) (b) the activation values of the tasks (c) the altitude from the sea floor (d) vehicle depth (blue) and sea floor (red), with the two thresholds of the minimum altitude activation function depicted as black dashed lines. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

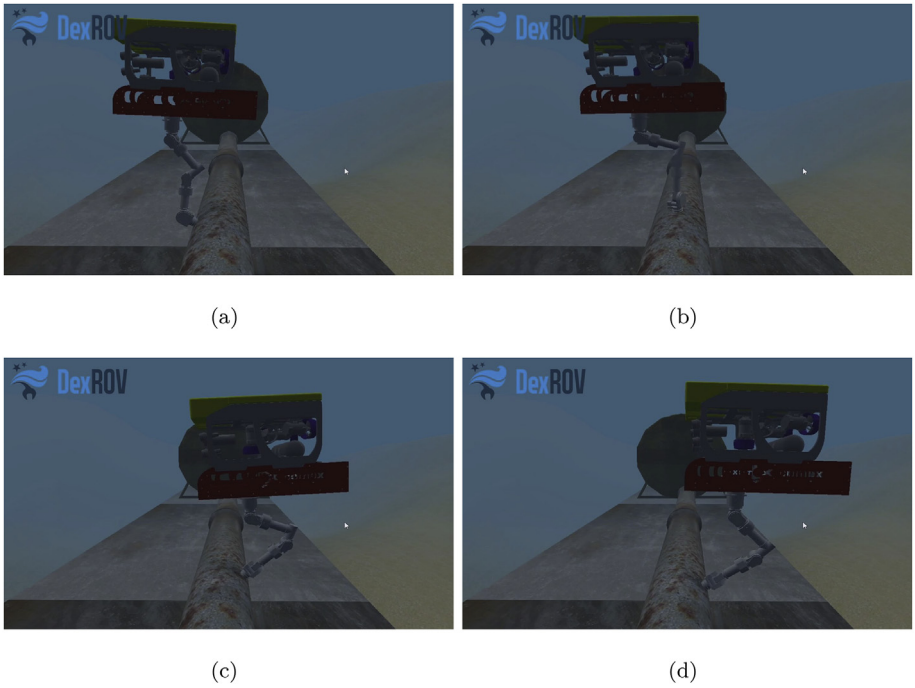


Fig. 7. Screenshots of the UVMS as it performs the pipe welding inspection task.

level, as done for instance in (Villani and De Schutter, 2016; Antonelli, 2014; Cataldi and Antonelli, 2015). Furthermore, with the adoption of this task at the highest priority, it is clear that subsequent tasks can only generate linear velocities in the tangential space to the surface.

Finally, let us assume that the end-effector ends with a flat surface where the sensor is installed. This flat surface should be the one in contact with the pipeline weld's surface. Let us further assume that the z axis of the end-effector frame coincides with the normal direction of the end-effector's flat surface. Then, the misalignment vector between the z axis of the end-effector frame and the surface's normal $\boldsymbol{n}$ should be minimized, allowing the alignment of the sensor with the surface.

Following the aforementioned requirement of having interaction tasks at the top of the task hierarchy, the pipeline inspection action results to be:

1. [R, E, C] Constraint complying and force regulation;
2. [R, I, S] Arm joint limits;
3. [R, I, S] Arm manipulability;
4. [R, I, S] Vehicle horizontal attitude;
5. [R, I, P] End-effector alignment with the surface's normal;
6. [R, E, AD] End-effector linear path following;
7. [R, E, AD] End-effector attitude tracking;
8. [R, I, O] Arm preferred shape.

As regards the action goal tasks 6 and 7 above, further details will be outlined within the DexROV pipeline inspection simulations presented in the successive section 4.2.

3.6. Landing

Finally, a landing action is considered. This action is the result of the initial investigations of the ROBUST project (ROBUST website, 2016). The AUV needs to land in front of a manganese nodule, to perform an in-situ measurement of the nodule itself and identify the presence of possible Rare Earth Elements deposits in the seafloor. The landing is achieved using the following hierarchy of objectives:

1. [R, I, S] Vehicle horizontal attitude;
2. [R, I, P] Vehicle longitudinal alignment to the nodule;
3. [R, I, P] Vehicle distance to the nodule;
4. [R, E, AD] Vehicle altitude.

After the successful landing, the in-situ measurement is executed through the arm manipulation action presented in section 2.4.

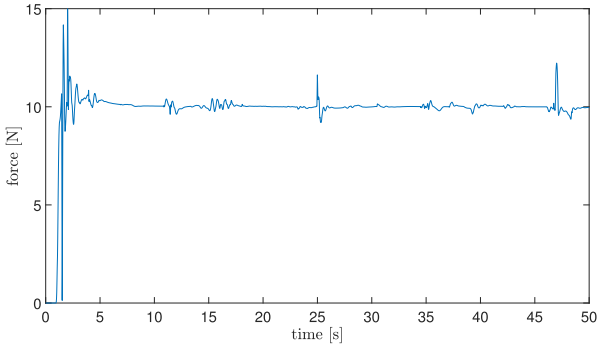
4. Simulation results

The results presented in the following subsections have been obtained through dynamic simulations coded in MATLAB. In particular, the following settings have been used:

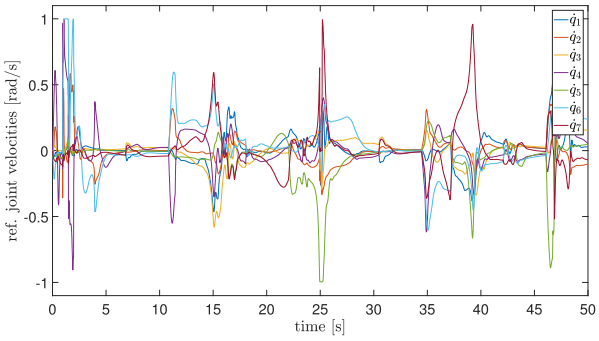
- A dynamic simulation was implemented and made running at 1 kHz frequency. To simulate the rigid body dynamics of the vehicle-manipulator system we have implemented the Newton-Euler equations, modelling the vehicle as a serial kinematic chain with 3 linear actuators and a spherical joint. This has allowed to easily take into account interaction forces between the vehicle and the manipulator's base.
- The simulation of vehicle dynamics included the added mass term $\boldsymbol{M}_A$, and linear and quadratic damping. It is in general difficult to separate the different sources of damping, therefore we wrote the total hydrodynamic damping as

$$\boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} = \boldsymbol{D} + \boldsymbol{D}_n(\boldsymbol{\nu}) \tag{10}$$

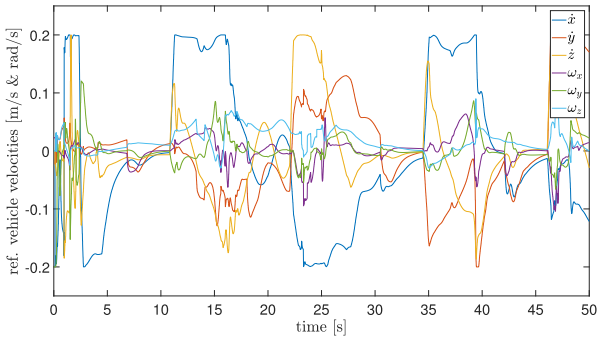
- where $\boldsymbol{D}$ is the linear damping matrix and $\boldsymbol{D}_n(\boldsymbol{\nu})$ is the nonlinear damping matrix due to quadratic damping and higher-order terms (Fossen, 2011). We have assumed both $\boldsymbol{D}$ and $\boldsymbol{D}_n(\boldsymbol{\nu})$ to be diagonal (see Eq. (7.256) of (Fossen, 2011)). Note that the ROBUST and DexROV simulation are using the same vehicle parameters, since no identification trials of the hydrodynamic terms have been executed yet. The only difference is represented by the position of the arm w.r.t. the center of mass of the vehicle. Therefore, the main vehicle dynamic parameters have been chosen as a scaled down version of (Caccia et al., 2000) and are reported in Table 2.
- Each arm link inertia was modelled taking into account mass and reduction gears, and arm torques were saturated, while



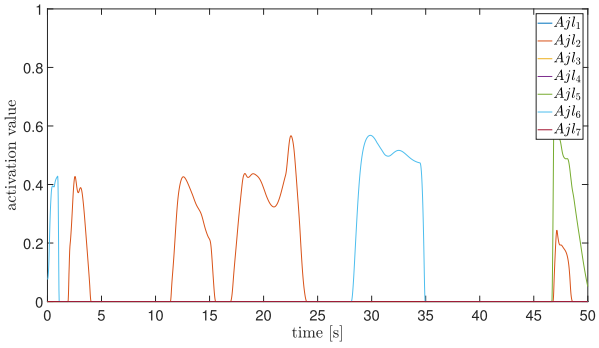
(a)



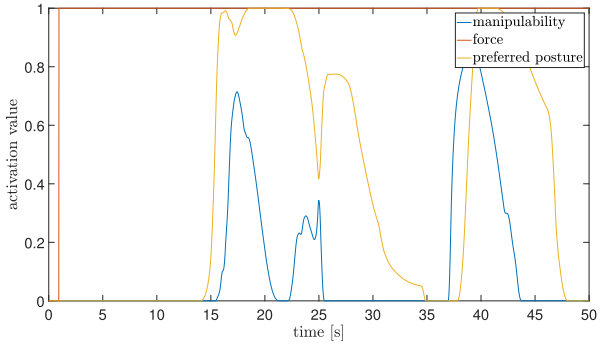
(b)



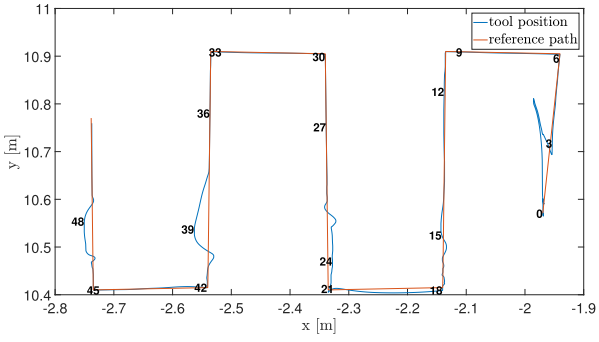
(c)



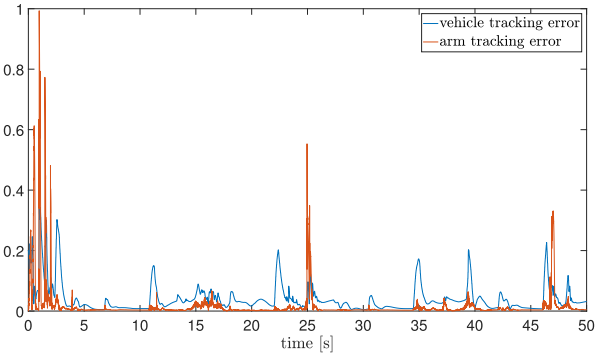
(d)



(e)



(f)



(g)

(caption on next page)

Fig. 8. Pipeline inspection action simulation results: (a) the time behaviour of the norm of the force (reference value is 10 N), (b) the arm joint velocities (saturation value at 1 rad/s) (c) vehicle reference velocities (saturation value at 0.2 rad/s and 0.2 m/s) (d) the activation values of the joint limit tasks, (e) the activation values of the other tasks, (f) reference path (red) and actual trajectory (blue), numbers indicate the time instant at which the end-effector is in that position (g) norm of the desired velocity tracking error for the arm and the vehicle. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

hydrodynamic forces (added mass and drag) were neglected. Reference values are reported in Table 3.

- We have assumed both the vehicle and the arm to be slightly negatively buoyant.
- The Dynamic Control Layer has been implemented in terms of separate independent *proportional-integrative* controllers, one for each degree of freedom, tuned around the nominal inertia of the manipulator links and vehicle. These controllers were executed at a frequency of 1 kHz.
- The Kinematic Control Layer, implementing the proposed task priority approach, was executed at 100 Hz frequency.
- All system states were assumed measurable and no measurement noise was included in the simulations.

4.1. Safe waypoint navigation

In this section, a simulation of the safe navigation action presented in Section 3.1 is shown. The vehicle starts from the position $[48.5 \;\; 11.5 \;\; -31]$ and is required to reach the target position $[50 \;\; -12.5 \;\; -31]$, while maintaining at least an altitude of 1.5 m from the seafloor (the corresponding activation function has a buffer zone $\Delta = 1.5$). As clearly seen in Fig. 5, the target point cannot be reached maintaining the desired depth of 31 m, since the seafloor rises in the vicinity of the starting position. Fig. 6(b) shows the time history of the activation functions of the inequality control objectives, and in particular of the minimum altitude task. The two plots of Fig. 6(c) and (d) show how the depth regulation is lost in favour of maintaining the required altitude from the seafloor, and as soon as the seafloor increases in depth, the vehicle converges to its target depth again. The simulation is therefore complaint with the desired behaviour shown in Fig. 3. As a final remark, the oscillations that can be seen in the minimum altitude and in its activation function are due to the irregular seafloor shape inside the simulation environment. This is confirmed by the depth of the vehicle, which does not have such oscillations.

4.2. Pipeline weld inspection: DexROV project

As said in section 3, one of the test case scenarios of the H2020 DexROV project is the inspection of a pipeline's weld. Normally, to carry out such an inspection, an electromagnetic sensor is placed in contact with the weld, and made slide along all the surface. To do so with a robotized system requires that the UVMS regulates the contact force to a given value, to ensure contact without damaging the probe.

To carry out the inspection of the weld, a reference path has been defined in the horizontal plane, in correspondence of the projection of the weld. Therefore, there is no knowledge of the actual shape or radius of the pipe. The expected result is that the end-effector follows the path projected on the pipe's surface, adapting to the unknown surface. A frictionless multi-point contact between the end-effector and the pipe surface was simulated. All the simulated forces and moments have been transferred to a unique point on the end-effector's rigid body space, where the presence of force/torque sensor has been assumed. Conversely to previous works (Di Lillo et al., 2016), the dynamics of the thrusters have been simulated, based on the single state model (Whitcomb and Yoerger, 1999) and taking the motor and propeller parameters from the Girona 500 (Ribas et al., 2012) Seaeye thrusters SI-MCT01.

In the simulation, the action presented in section 3.5 has been implemented. For the force regulation task, the following simple reactive reference rate has been used, and substituted in (9)

$$\dot{\tilde{x}}_f \triangleq \gamma(\lambda^* - \lambda), \;\; \gamma > 0, \tag{11}$$

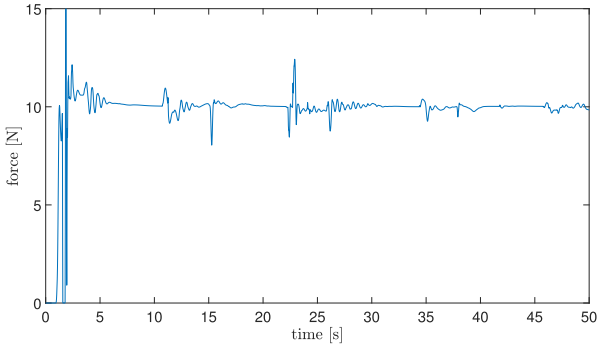
where λ and λ^* are the current and desired (scalar) values of the force to be exerted on the surface, and γ is a positive proportional gain. Note that, in this specific case, a proportional law is sufficient since in the simulation the underlying DCL has been implemented with integral actions, as done in the actual DexROV UVMS prototype. As previously explained in section 3.5, the Jacobian of this task requires the knowledge of the surface normal $\boldsymbol{n}$, which can be estimated only once the end-effector is already in contact with the pipeline through the force/torque sensor on the wrist of the manipulator. Therefore, this task has an activation function that depends on the norm of the contact force, which explains the presence of a force activation function shown in the plots of Fig. 8(e). The same activation function is also used for the surface alignment task.

Screenshots of the UVMS performing the inspection task are reported in Fig. 7, taken from the video available at: https://youtu.be/5hK2UMAoN94. The results are summarized in Fig. 8, where in particular Fig. 8(a) shows how the force is regulated close to the desired value of 10 N, while following the path on the pipe with a maximum deviation of approximately 3 cm occurring around t = 39 s, as shown in Fig. 8(f). The above results are achieved notwithstanding the activation of different safety tasks such as joint limits (see Fig. 8(d)) and prerequisite ones such as manipulability (see Fig. 8(e)). Finally, the generated system velocity references take into account the velocity saturations as it can be clearly appreciated in both Fig. 8(b) and (c).

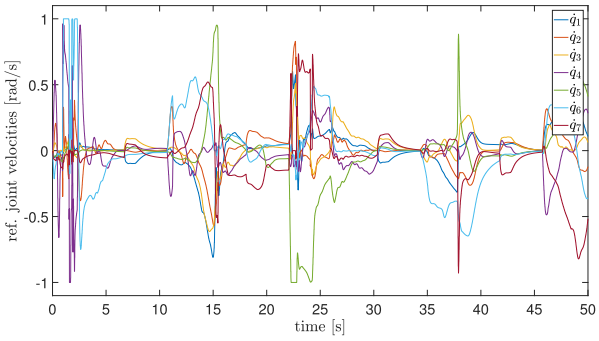
To further comment on the obtained results, let us consider the deviations from the desired path depicted in Fig. 8(f). There are three intervals where the error is significant:

- Between $t = 15$ s and $t = 25$ s, both the joint 2 limit avoidance and the minimum manipulability tasks are active. This means that 2 out of 7 manipulator's degrees of freedom are used for such tasks. Therefore, in TPIK 2 of the vehicle-arm coordination scheme of Section 2.10, the manipulator does not have enough degrees of freedom to accomplish the end-effector tracking tasks on its own. Since the vehicle has a significant tracking error, the deviations occur.
- At $t = 39$ s the manipulability task has an activation value of approximately 0.9, indicating that the manipulability index is very low and that at least one of the singular value is approaching zero. The end-effector linear position task pseudo-inverse in (6) is thus regularized to prevent joint and vehicle velocities going to infinity (Simetti and Casalino, 2016), at the expenses of path tracking accuracy.
- At $t = 48$ s we have a further different reason for the path tracking error. In this case there are two joints near their limit. For the same reasons outlined in the first point, the path tracking error occurs.

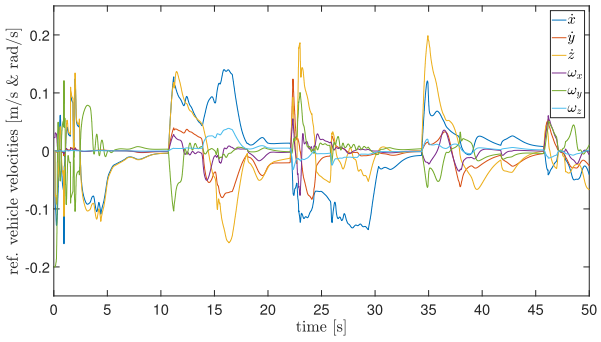
Let us also comment on the two spikes occurring in the force regulation, shown in Fig. 8(a). They are occurring approximately at $t = 25$ s and at $t = 47$ s. It is interesting to note that in such time instants the desired velocity for the manipulator is quite high and close to the saturation value. As a consequence, despite the fast arm dynamics there is a peak in the joint velocity tracking error as evident in Fig. 8(g). As the manipulator is in contact with the pipe, such tracking errors make the manipulator move against the surface of the pipe, creating unwanted spikes in the force regulation.



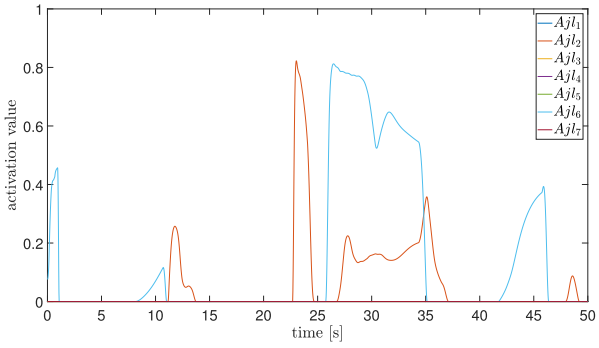
(a)



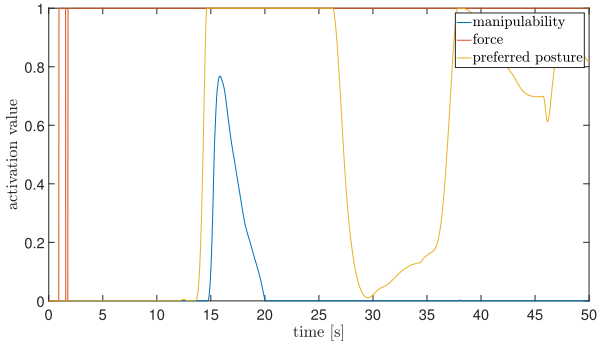
(b)



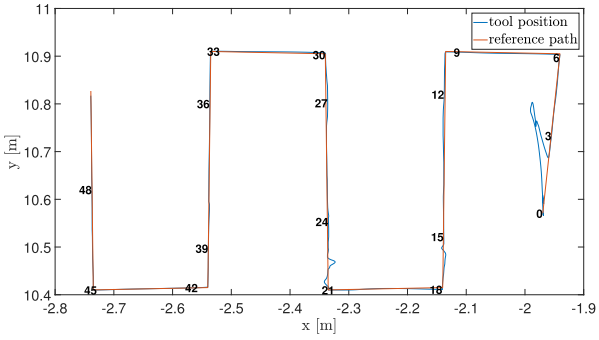
(c)



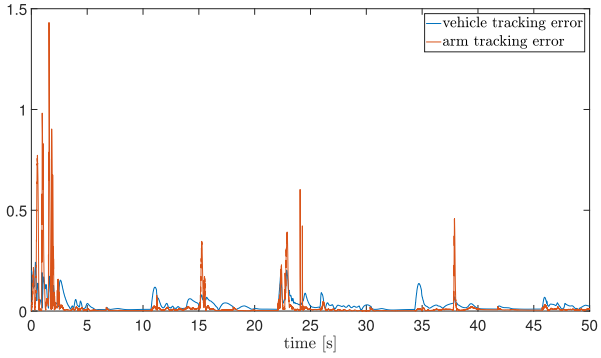
(d)



(e)



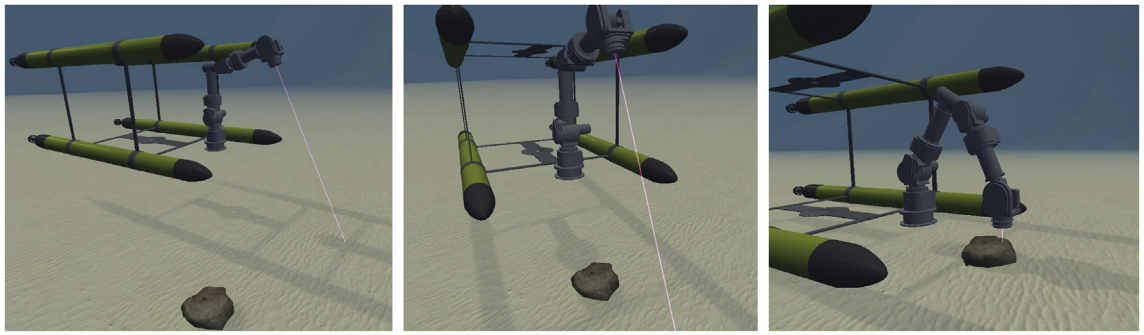
(f)



(g)

(caption on next page)

Fig. 9. Pipeline inspection action simulation results: (a) the time behaviour of the norm of the force (reference value is 10 N), (b) the arm joint velocities (saturation value at 1 rad/s) (c) vehicle reference velocities (saturation value at 0.2 rad/s and 0.2 m/s) (d) the activation values of the joint limit tasks, (e) the activation values of the other tasks, (f) reference path (red) and actual trajectory (blue) (g) norm of the desired velocity tracking error for the arm and the vehicle. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)



(a) (b) (c)

Fig. 10. Screenshots of the AUV as it performs the landing and then the LIBS measurement.

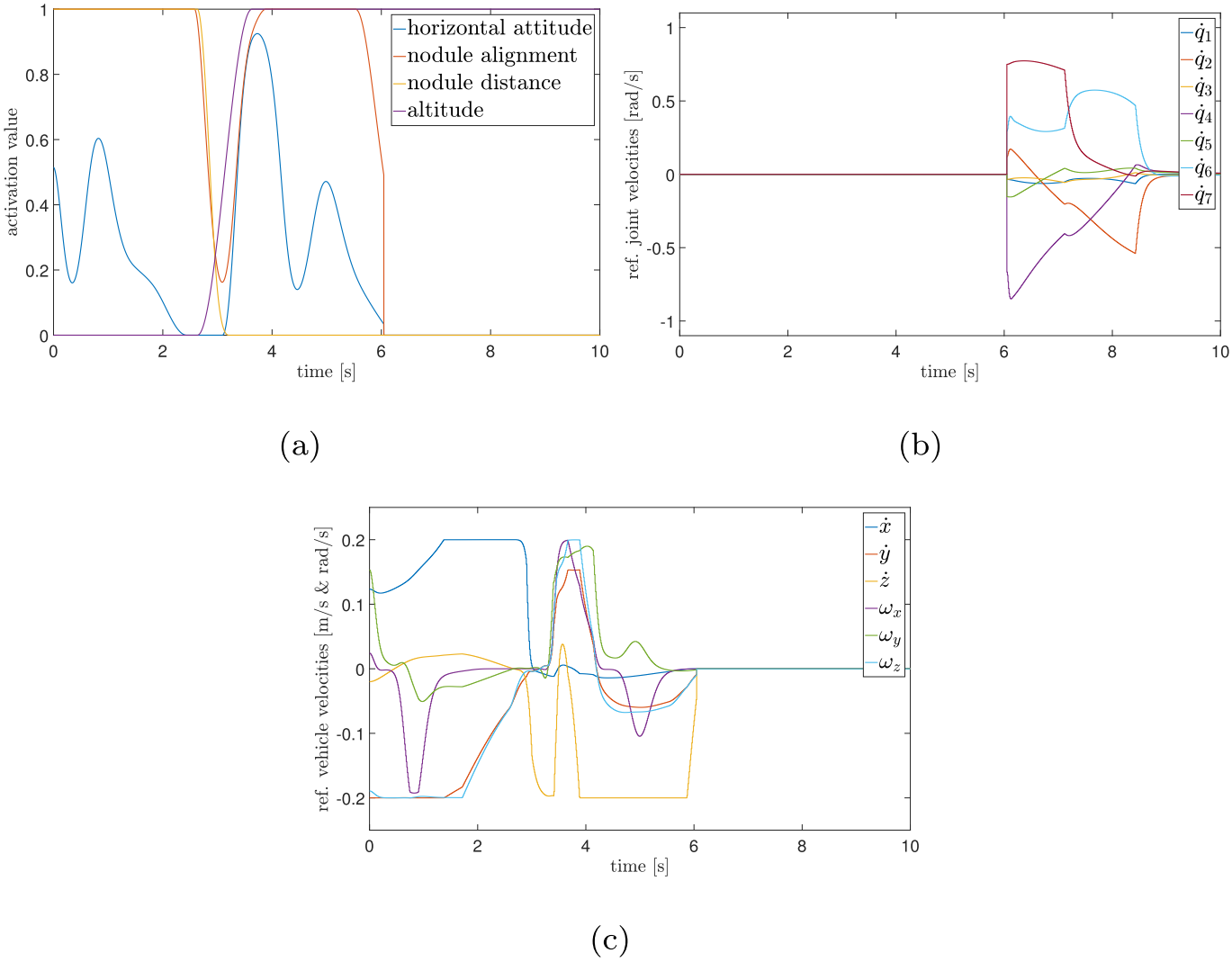


Fig. 11. Landing action simulation results: (a) the activation values of the vehicle-related tasks during the landing phases, (b) vehicle reference velocities (saturation value at 0.2 rad/s and 0.2 m/s) (c) the arm joint velocities (saturation value at 1 rad/s).

A second simulation is also presented here, where a non-reactive vehicle velocity minimization task was added at the bottom of the task hierarchy. Results are shown in Fig. 9. Looking at Fig. 9(f), and comparing with Fig. 8(f), a slightly better path tracking can be observed. This can be motivated by the lower velocity requested to the vehicle (Fig. 9(c) compared to Fig. 8(c)) and hence the lower tracking error. Therefore, even when two joints are close to their limit, the deviations from the desired path are of minor entity.

4.3. Landing: ROBUST project

The H2020 ROBUST project (ROBUST website, 2016) aims to develop sea bed in-situ material identification through the use of Laser Induced Breakdown Spectroscopy (LIBS). After diving to a pre-programmed altitude, the UVMS performs a mapping of the area and identifies the most promising area where manganese nodules can be found. At this point, it proceeds with a second, more detailed survey at a lower altitude in a smaller area, and whenever a nodule is identified, it lands to perform the in-situ measurement, by bringing the LIBS rigidly attached to the end-effector of the manipulator within a few centimetres of the nodule.

In this simulation, the UVMS starts close to the nodule position, but not in front. Therefore, the UVMS first needs to align itself with the nodule direction, and maintain a certain distance from it. Once these values are settled, it can perform the final descend and land on the seafloor. Once the system has detected that the landing has been successfully completed (i.e. depth is constant), a 3D laser based reconstruction of the nodule is performed. The output of the scan is used to generate a set of points around the nodule where the measurements should be taken. Afterwards, the manipulator, acting as a fixed base one, moves the LIBS in the position to perform the measurement.

The same simulation settings as in the pipeline inspection case were used. Fig. 10 shows a few snapshots of the simulation while the UVMS performs the landing operation and the successive in-situ measurement. The results are summarized in Fig. 11, showing the time behaviour of the activation functions and the generated joint and vehicle reference velocities. As clearly seen in the Figures, approximately at $t = 6$ s the landing is concluded and the manipulation action is started, which then concludes at about $t = 10$ s.

As a final remark, contact between the vehicle and the seafloor is not simulated. Hence, once the landing has been completed, the simulation simply fixes the vehicle actual velocity to zero. Therefore, the activation functions of the vehicle related tasks, such as minimum altitude should be disregarded after $t = 6$ s. In the real implementation, to keep the vehicle fixated to the seafloor, two solutions are being considered. The first one is to add a variable buoyancy system, and make the vehicle negative once the landing has been completed. The second solution simply consists in using the vertical thrusters on top of the AUV to keep the vehicle pushed against the seafloor.

5. Open problems and current research

This paper has described the task priority control framework implemented within the KCL. The output of the KCL are the system reference velocities that need to be tracked by the underlying DCL. Therefore, the KCL should generate reference signals within the DCL bandwidth, i.e. the KCL should be ”slower” than the DCL, which is often constituted by commercial actuators, both for the thrusters of the vehicle and the joint controllers of the manipulator. Consequently, it is not possible to implement advanced control techniques, such as a computed torque control.

For the above reasons, the KCL should adapt its performances to those given by the DCL. The tuning of the KCL is generally long and tedious, mostly due to the high number of KCL parameters. Even if the task priority approach decouples the contribution of the control tasks thanks to the priority concept, it still requires different simulation sessions with an hydrodynamic model of the system, followed by experimental trials ad hoc designed to test each control task tuning.

Best practices for the task priority KCL parameters and gains tuning do not exist yet. Assessing them is currently under investigation, as well as the possibility of devising auto-tuning procedures.

6. Conclusions

This paper has presented a unifying framework for the kinematic control layer of underwater vehicle manipulator systems. The original task priority approach (Siciliano and Slotine, 1991) has been first extended allowing control tasks activation and deactivation without discontinuities in the control variables, in order to efficiently implement inequality control objectives and task transitions between different actions (Simetti and Casalino, 2016). Furthermore, the developed framework explicitly deals with the tracking inaccuracies of the vehicle dynamic control layer, with the two parallel optimizations recalled in section 2.10, allowing for optimal control of the end-effector even during floating operations, as shown in (Simetti et al., 2017). Underactuated vehicles are also deal with in a simple manner, as briefly recalled in section 2.9. Everything has been accomplished maintaining an invariant and uniform algorithmic structure, as presented in section 2.6.

In the MARIS project, the proposed architecture has been experimentally tested in a grasping scenario, achieving good results in repeatability and robustness (Simetti et al., 2017). In this work, we have shown how the architecture can be applied to execute different operations, ranging from a safe navigation action, to the DexROV pipeline inspection test case or the in-site mineral measurement needed in the ROBUST project. In particular, this work has shown how the framework has been extended to include interaction and force regulation at the kinematic level, a necessary feature to handle the DexROV pipeline inspection case study. The flexibility of the approach is demonstrated through different dynamic simulations.

Further refinements, especially on the integration of impedance control and on KCL parameter tuning procedures are currently being developed in the scope of the DexROV and ROBUST projects.

Acknowledgments

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690416 (ROBUST) and under grant agreement No 635491 (DexROV).

References

Allotta, B., Conti, R., Costanzi, R., Fanelli, F., Meli, E., Ridolfi, A., 2015. Towards next generation intervention autonomous underwater vehicles (I-AUV): development of innovative mobile manipulation techniques. In: VI International Conference on Computational Methods in Marine Engineering (MARINE), Rome, Italy, pp. 1–10.

Alvarez, A., Caffaz, A., Caiti, A., Casalino, G., Gualdesi, L., Turetta, A., Viviani, R., 2009. Fòlaga: a low-cost autonomous underwater vehicle combining glider and AUV capabilities. Ocean Eng. 36 (1), 24–38. http://dx.doi.org/10.1016/j.oceaneng.2008.08.014.

Angeletti, D., Cannata, G., Casalino, G., 1998. The control architecture of the AMADEUS gripper. Int. J. Syst. Sci. 29 (5), 485–496. http://dx.doi.org/10.1080/00207729808929539.

Antonelli, G., 2014. Underwater Robots, Vol. 96 of Springer Tracts in Advanced Robotics. Springerhttp://dx.doi.org/10.1007/978-3-319-02877-4.

Antonelli, G., Indiveri, G., Chiaverini, S., 2009. Prioritized closed-loop inverse kinematic algorithms for redundant robotic systems with velocity saturations. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, pp. 5892–5897. http://dx.doi.org/10.1109/IROS.2009.5354636.

Bachmayer, R., Whitcomb, L.L., Grosenbaugh, M.A., 2000. An accurate four-quadrant nonlinear dynamical model for marine thrusters: theory and experimental validation. IEEE J. Ocean. Eng. 25 (1), 146–159. http://dx.doi.org/10.1109/48.820747.

Bingham, B., Foley, B., Singh, H., Camilli, R., Delaporta, K., Eustice, R., Mallios, A., Mindell, D., Roman, C., Sakellariou, D., 2010. Robotic tools for deep water archaeology: surveying an ancient shipwreck with an autonomous underwater vehicle. J. Field Robot. 27 (6), 702–717. http://dx.doi.org/10.1002/rob.20350.

Caccia, M., Indiveri, G., Veruggio, G., 2000. Modeling and identification of open-frame

variable configuration unmanned underwater vehicles. IEEE J. Ocean. Eng. 25 (2), 227–240. http://dx.doi.org/10.1109/48.838986.
Carrera, A., Palomeras, N., Hurtos, N., Kormushev, P., Carreras, M., 2014. Learning by demonstration applied to underwater intervention. In: International Conference of the Catalan Association for Artificial Intelligence, pp. 95–104.
Casalino, G., Angeletti, D., Bozzo, T., Marani, G., 2001. Dexterous underwater object manipulation via multi-robot cooperating systems. In: IEEE International Conference on Robotics and Automation (ICRA), vol. 4. IEEE, pp. 3220–3225. http://dx.doi.org/10.1109/ROBOT.2001.933114.
Cataldi, E., Antonelli, G., 2015. Basic interaction operations for an underwater vehicle-manipulator system. In: 17th International Conference on Advanced Robotics (ICAR), Istanbul, Turkey, pp. 524–529. http://dx.doi.org/10.1109/icar.2015.7251506.
Choi, S.K., Takashige, G.Y., Yuh, J., 1994. Experimental study on an underwater robotic vehicle: Odin. In: Proceedings of IEEE Symposium on Autonomous Underwater Vehicle Technology (AUV'94). IEEE, pp. 79–84. http://dx.doi.org/10.1109/auv.1994.518610.
Cieslak, P., Ridao, P., Giergiel, M., 2015. Autonomous underwater panel operation by GIRONA500 UVMS: a practical approach to autonomous underwater manipulation. In: IEEE International Conference on Robotics and Automation (ICRA). IEEE, pp. 529–536. http://dx.doi.org/10.1109/icra.2015.7139230.
Conti, R., Meli, E., Ridolfi, A., Allotta, B., 2015. An innovative decentralized strategy for I-AUVs cooperative manipulation tasks. Robot. Autonom. Syst. 72, 261–276. http://dx.doi.org/10.1016/j.robot.2015.06.006.
Di Lillo, P.A., Simetti, E., De Palma, D., Cataldi, E., Indiveri, G., Antonelli, G., Casalino, G., 2016. Advanced ROV autonomy for efficient remote control in the DexROV project. Mar. Technol. Soc. J. 50 (4), 67–80. http://dx.doi.org/10.4031/mtsj.50.4.8.
Drap, P., Seinturier, J., Conte, G., Caiti, A., Scaradozzi, D., Zanoli, S., Gambogi, P., 2008. Underwater cartography for archaeology in the VENUS project, geomatica: the j. of geospatial information science. Technol. Pract. 62 (4), 419–428.
Escande, A., Mansard, N., Wieber, P.-B., 2014. Hierarchical quadratic programming: fast online humanoid-robot motion generation. Int. J. Robot Res. 33 (7), 1006–1028. http://dx.doi.org/10.1177/0278364914521306.
Evans, J., Keller, K., Smith, J., Marty, P., Rigaud, O., 2001. Docking techniques and evaluation trials of the SWIMMER AUV: an autonomous deployment AUV for work-class ROVs. In: OCEANS, 2001. MTS/IEEE Conference and Exhibition, vol. 1. IEEE, pp. 520–528. http://dx.doi.org/10.1109/oceans.2001.968776.
Evans, J., Redmond, P., Plakas, C., Hamilton, K., Lane, D., 2003. Autonomous docking for Intervention-AUVs using sonar and video-based real-time 3D pose estimation. In: Oceans 2003, vol. 4. IEEE, pp. 2201–2210. http://dx.doi.org/10.1109/OCEANS.2003.178243.
Farivarnejad, H., Moosavian, S.A.A., 2014. Multiple impedance control for object manipulation by a dual arm underwater vehicle–manipulator system. Ocean Eng. 89, 82–98. http://dx.doi.org/10.1016/j.oceaneng.2014.06.032.
Fossen, T.I., 2011. Handbook of marine Craft Hydrodynamics and Motion Control. John Wiley & Sons.
Gilmour, B., Niccum, G., O'Donnell, T., 2012. Field resident AUV systems – chevron's long-term goal for AUV development. In: 2012 IEEE/OES Autonomous Underwater Vehicles (AUV). IEEE, pp. 1–5. http://dx.doi.org/10.1109/AUV.2012.6380718.
Kanoun, O., Lamiraux, F., Wieber, P.B., 2011. Kinematic control of redundant manipulators: generalizing the task-priority framework to inequality task. Robot. IEEE Trans. 27 (4), 785–792. http://dx.doi.org/10.1109/tro.2011.2142450.
Khatib, O., 1987. A unified approach for motion and force control of robot manipulators: the operational space formulation. IEEE J. Robot. Autom. 3 (1), 43–53. http://dx.doi.org/10.1109/jra.1987.1087068.
Lane, D.M., Davies, J.B.C., Casalino, G., Bartolini, G., Cannata, G., Veruggio, G., Canals, M., Smith, C., O'Brien, D.J., Pickett, M., Robinson, G., Jones, D., Scott, E., Ferrara, A., Angelleti, D., Coccoli, M., Bono, R., Virgili, P., Pallas, R., Gracia, E., 1997. AMADEUS: advanced manipulation for deep underwater sampling. IEEE Robot. Autom. Mag. 4 (4), 34–45. http://dx.doi.org/10.1109/100.637804.
Marani, G., Choi, S.K., Yuh, J., 2008. Underwater autonomous manipulation for intervention missions AUVs. Ocean Eng. 36, 15–23. http://dx.doi.org/10.1016/j.oceaneng.2008.08.007.
Marty, P., et al., 2004. ALIVE: an autonomous light intervention vehicle. In: Advances in Technology for Underwater Vehicles Conference, Oceanology International, vol. 2004.
Moe, S., Antonelli, G., Teel, A.R., Pettersen, K.Y., Schrimpf, J., 2016. Set-based tasks within the singularity-robust multiple task-priority inverse kinematics framework: general formulation, stability analysis, and experimental results. Frontiers in Robotics and AI 3, 16. http://dx.doi.org/10.3389/frobt.2016.00016.
Nakamura, Y., 1991. Advanced Robotics: Redundancy and Optimization. Addison Wesley.
Nakamura, Y., Hanafusa, H., 1986. Inverse kinematic solutions with singularity robustness for robot manipulator control. J. Dyn. Syst. Meas. Contr. 108 (3), 163–171. http://dx.doi.org/10.1115/1.3143764.
Perez, T., Fossen, T.I., 2007. Kinematic models for manoeuvring and seakeeping of marine vessels, Modeling. Identification and Control 28 (1), 19–30. http://dx.doi.org/10.4173/mic.2007.1.3.
Ribas, D., Palomeras, N., Ridao, P., Carreras, M., Mallios, A., 2012. Girona 500 AUV: from survey to intervention. IEEE ASME Trans. Mechatron. 17 (1), 46–53. http://dx.doi.org/10.1109/tmech.2011.2174065.
Ribas, D., Ridao, P., Turetta, A., Melchiorri, C., Palli, G., Fernandez, J., Sanz, P., 2015. I-AUV mechatronics integration for the TRIDENT FP7 project. IEEE ASME Trans. Mechatron. 20 (5), 2583–2592. http://dx.doi.org/10.1109/TMECH.2015.2395413.
Ridao, P., Carreras, M., Ribas, D., Garcia, R., 2010. Visual inspection of hydroelectric dams using an autonomous underwater vehicle. J. Field Robot. 27 (6), 759–778. http://dx.doi.org/10.1002/rob.20351.
Rigaud, V., Coste-Manière, È., Aldon, M.-J., Probert, P., Perrier, M., Rives, P., Simon, D., Lang, D., Kiener, J., Casal, A., et al., 1998. UNION: underwater intelligent operation and navigation. IEEE Robot. Autom. Mag. 5 (1), 25–35. http://dx.doi.org/10.1109/100.667323.
ROBUST website, http://eu-robust.eu, [Online; accessed 25-October-2016] (2016).
Sanz, P.J., Ridao, P., Oliver, G., Casalino, G., Petillot, Y., Silvestre, C., Melchiorri, C., Turetta, A., 2013. TRIDENT: an European project targeted to increase the autonomy levels for underwater intervention missions. In: Oceans-San Diego, 2013. IEEE, pp. 1–10. http://dx.doi.org/10.23919/OCEANS.2013.6741370.
Schempf, H., Yoerger, D., 1992. Coordinated vehicle/manipulator design and control issues for underwater telemanipulation. In: IFAC Control Applications in Marine Systems (CAMS 92), Genova, Italy, pp. 259–267.
Siciliano, B., Slotine, J.-J.E., 1991. A general framework for managing multiple tasks in highly redundant robotic systems. In: Proc. Fifth International Conference on Advanced Robotics (ICAR). IEEE, Pisa, Italy, pp. 1211–1216. http://dx.doi.org/10.1109/ICAR.1991.240390.
Simetti, E., Casalino, G., 2016. A novel practical technique to integrate inequality control objectives and task transitions in priority based control. J. Intell. Rob. Syst. 84 (1), 877–902. http://dx.doi.org/10.1007/s10846-016-0368-6.
Simetti, E., Casalino, G., 2017. Manipulation and transportation with cooperative underwater vehicle manipulator systems. IEEE J. Ocean. Eng. 42 (4), 782–799. http://dx.doi.org/10.1109/joe.2016.2618182.
Simetti, E., Casalino, G., Torelli, S., Sperindé, A., Turetta, A., 2014. Floating underwater manipulation: developed control methodology and experimental validation within the TRIDENT project. J. Field Robot. 31 (3), 364–385. http://dx.doi.org/10.1002/rob.21497.
Simetti, E., Wanderlingh, F., Torelli, S., Bibuli, M., Odetti, A., Bruzzone, G., Lodi Rizzini, D., Aleotti, J., Palli, G., Moriello, L., Scarcia, U., 2017. Autonomous underwater intervention: experimental results of the MARIS project. IEEE J. Ocean. Eng. 1–20. http://dx.doi.org/10.1109/JOE.2017.2733878.
Urabe, T., Ura, T., Tsujimoto, T., Hotta, H., 2015. Next-generation technology for ocean resources exploration (Zipangu-in-the-Ocean) project in Japan. In: OCEANS 2015-Genova, IEEE, pp. 1–5. http://dx.doi.org/10.1109/oceans-genova.2015.7271762.
Villani, L., De Schutter, J., 2016. Force control. In: Springer Handbook of Robotics. Springer, pp. 195–220. http://dx.doi.org/10.1007/978-3-319-32552-1_9.
Wang, H., Rock, S., Lees, M., 1995. Experiments in automatic retrieval of underwater objects with an AUV. In: MTS/IEEE Oceans '95, vol. 1. IEEE, pp. 366–373. http://dx.doi.org/10.1109/oceans.1995.526796.
Whitcomb, L.L., Yoerger, D.R., 1995. Comparative experiments in the dynamics and model-based control of marine thrusters. In: OCEANS, vol. 2. IEEE, pp. 1019–1028. http://dx.doi.org/10.1109/OCEANS.1995.528561.
Whitcomb, L.L., Yoerger, D.R., 1999. Preliminary experiments in model-based thruster control for underwater vehicle positioning. IEEE J. Ocean. Eng. 24 (4), 495–506. http://dx.doi.org/10.1109/48.809273.
Wynn, R.B., Huvenne, V.A., Bas, T.P.L., Murton, B.J., Connelly, D.P., Bett, B.J., Ruhl, H.A., Morris, K.J., Peakall, J., Parsons, D.R., Sumner, E.J., Darby, S.E., Dorrell, R.M., Hunt, J.E., 2014. Autonomous underwater vehicles (AUVs): their past, present and future contributions to the advancement of marine geoscience. Mar. Geol. 352, 451–468. http://dx.doi.org/10.1016/j.margeo.2014.03.012.
Yoerger, D.R., Schempf, H., DiPietro, D.M., 1991. Design and performance evaluation of an actively compliant underwater manipulator for full-ocean depth. J. Rob. Syst. 8 (3), 371–392. http://dx.doi.org/10.1002/rob.4620080306.
Yoshikawa, T., 1985. Manipulability of robotic mechanisms. Int. J. Robot Res. 4 (1), 3–9. http://dx.doi.org/10.1177/027836498500400201.
Yuh, J., Choi, S., Ikehara, C., Kim, G., McMurty, G., Ghasemi-Nejhad, M., Sarkar, N., Sugihara, K., 1998. Design of a semi-autonomous underwater vehicle for intervention missions (SAUVIM). In: Proc. Of 1998 International Symposium on Underwater Technology. IEEE, Tokyo, Japan, pp. 63–68. http://dx.doi.org/10.1109/UT.1998.670059.
Yuh, J., Marani, G., Blidberg, D.R., 2011. Applications of marine robotic vehicles. Intelligent Service Robotics 4 (4), 221–231. http://dx.doi.org/10.1007/s11370-011-0096-5.