DexROV project: Control Framework for Underwater Interaction Tasks

Enrico Simetti*†, Francesco Wanderlingh*†, Giuseppe Casalino*†, Giovanni Indiveri*‡, Gianluca Antonelli*§
*Interuniversity Research Center on Integrated Systems for the Marine Environment
Via Opera Pia 13, 16145 Genova, Italy, Email: enrico.simetti@unige.it
†DIBRIS, University of Genova, Via Opera Pia 13, 16145 Genova, Italy
‡DII, University of Salento, Via per Monteroni, 73100 Lecce, Italy
§University of Cassino and Southern Lazio, Via Di Biasio 43, 03043, Cassino, Italy

***Abstract*—In this work, the control framework of the DexROV Horizon 2020 project is presented. The framework is based on the task priority concept, extended by the authors to allow the activation and deactivation of tasks. The general concepts of control objectives, task and actions are given. The execution of a pipeline's weld inspection is used as study case to test the proposed framework in a simulation setting.**

I. Introduction

This paper presents the control methodology proposed for DexROV (Dexterous ROV: effective dexterous ROV operations in presence of communication latencies) [1], an EC Horizon 2020 funded project that proposes to implement novel operation strategies for underwater semi-autonomous interventions. These costly and demanding operations are increasingly performed by remotely operated vehicles (ROVs), contributing to risk cutting for human divers. To reduce the costs of ROV operations, the main idea of DexROV is to delocalize on shore the manned support 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 by the use of satellite communications between the support ship and the remote control center. The introduction of such a communication channel makes a direct teleoperation impossible due to the high latencies. For this reason, a cognitive engine [2], together with a simulation environment and an exoskeleton are used to interact with the operator, and only high level commands are sent through the satellite channels, which are then forwarded to the ROV, which must execute them in a semi-autonomous manner.

The Interuniversity Research Center on Integrated Systems for the Marine Environment (ISME, Italy) is working since more than a decade in marine robotics and in particular since the TRIDENT [3], [4], [5] and MARIS [6], [7] projects is working on the control of Underwater Vehicle Manipulator Systems (UVMS) for the execution of intervention tasks. While the DexROV scenario does not require full autonomy, the execution of a single action, such as turning a value, must be done autonomously, at least from the control point of view.

The type of intervention required in maintenance tasks requires the ROV to interact with the environment. For this reason, ISME is now focusing on force control schemes [8] and their integration in the developed kinematic task priority framework [9]. This is particularly important because the DexROV ROV can only be controlled through a body velocity request, since it is a commercial system and its dynamic control cannot be changed.

In this work, a pipeline inspection task is used as case study, where an electromagnetic sensor must be maintained in contact with the pipe near its welding line, to check if any surface cracks are present. The challenge and innovation of this work is that the force control is integrated into a well consolidated task priority control scheme, where many other kinematic tasks must be satisfied. No preliminary information about the environment is used; only the force and moments at the wrist sensor are used to accomplish the inspection task. We extend the previously developed study [10] by considering saturations at dynamic level (saturated torques on arm joints, and saturated generalised force/torque vector on the vehicle) as well as saturations in the velocities. The latter is also taken into account by the kinematic control, thanks to the integration of the task-priority based saturation algorithm proposed in [11] in the developed framework [9].

II. The Control Framework

The developed control framework is based on a clear separation between the Kinematic Control Layer (KCL), hereafter described, and the Dynamic Control Layer (DCL). The KCL, through the definition of the control objectives, tasks and actions as described in this section, generates a reference system velocity vector. The DCL is instead concerned only with the tracking of such reference velocity, generating the corresponding arm torques and vehicle generalized forces. This work will focus on the developed KCL, since the DCL systems for the DexROV arms and vehicle are given and cannot be changed, due to the commercial actuators employed.

A. 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

$$\boldsymbol{c} \triangleq \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{\eta} \end{bmatrix}, \tag{1}$$

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 $\boldsymbol{\eta}_1$, with components on the inertial frame $\langle w \rangle$, and the orientation vector $\boldsymbol{\eta}_2$, the latter expressed in terms of the three angles yaw, pitch and roll (applied in this sequence) [12]. From the above definitions it results $n = l + 6$;
- the system velocity vector $\dot{\boldsymbol{y}} \in \mathbb{R}^n$ of the UVMS as

$$\dot{\boldsymbol{y}} \triangleq \begin{bmatrix} \dot{\boldsymbol{q}} \\ \boldsymbol{v} \end{bmatrix}, \tag{2}$$

where $\dot{\boldsymbol{q}} \in \mathbb{R}^l$ are the joint velocities and $\boldsymbol{v} \in \mathbb{R}^6$ is the stacked vector of the vehicle linear velocity vector $\boldsymbol{v}_1$ and the vehicle angular velocity vector $\boldsymbol{v}_2$, both with components on the vehicle frame $\langle v \rangle$.

B. Control Objectives

Let us consider a generic, configuration dependent, scalar variable $x(\boldsymbol{c})$ among those definable within the system, or between the system and the environment, and consider the following requirements, as $t \to \infty$:

- Requiring $x(\boldsymbol{c}) = x_0$ is termed a *scalar equality control objective*.
- Requiring $x(\boldsymbol{c}) < x_M$ or $x(\boldsymbol{c}) > x_m$ is termed a *scalar inequality control objective*.

Let us note how, if we consider m separate scalar variables $x_i(\boldsymbol{c})$, each of them corresponding to a single component of a vector $\boldsymbol{p} \in \mathbb{R}^m$, then it is possible to require that the vector $\boldsymbol{p}$ reaches any desired value. The whole set of m scalar objectives constitutes an equality/inequality control objective. Therefore considering scalar objectives does not limit the generality of the approach.

C. Control Tasks

Once a control objective has been established, a so-called *reference feedback rate* $\dot{\bar{x}}$ needs to be defined. The role of such a reference is to drive the associated variable $x(\boldsymbol{c})$ toward an arbitrary point x^*, located inside the interval where the objective is satisfied.

For instance, an example of reference feedback rate is the following simple one

$$\dot{\bar{x}}(x) \triangleq \gamma(x^* - x), \ \gamma > 0, \tag{3}$$

where γ is a positive gain proportional to the desired convergence rate for the considered variable.

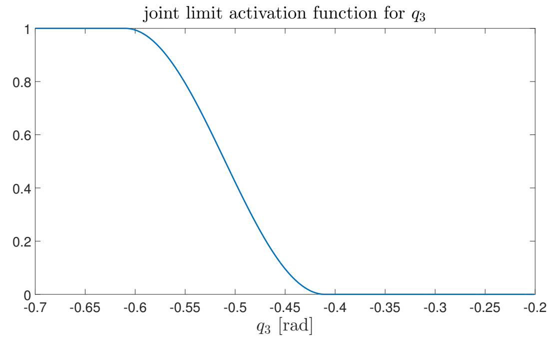
The need of having the actual velocity $\dot{x}$ as close as possible to the reference feedback rate $\dot{\bar{x}}$ is termed as *feedback task*.

D. Task Activation Function

For each control objective, and its associated feedback task, there is also always associated a *task activation function*, defined as follows

$$a = a^i(x)a^p(\boldsymbol{e}), \tag{4}$$

where



(a)

Fig. 1. Example of activation function for a joint limit task, where $q_3 > 0.61$.

- $a^i(x) \in [0, 1]$ is continuous sigmoidal function of the objective variable x, which assumes zero values within a complete inner sub-interval of the validity of the associated objective. An example is reported in Fig. 1.
- $a^p(\boldsymbol{e}) \in [0, 1]$ is a continuous sigmoidal function of some vector of parameters $\boldsymbol{e}$ external to the objective itself. Later in section II-G it will be more clear how this function is exploited.

E. Task Priority Control Architecture

Once all the objectives, and their associated tasks and activation functions have been defined, it is needed to establish a *priority* between them. The meaning that a certain objective has higher priority w.r.t. another one is that objectives at a lower priority level must not interfere with those located at higher priority levels.

Whenever multiple scalar objectives are assigned to the same priority level, they are grouped into a generic control objective. For example, recall the example of the m objectives corresponding to a vector $\boldsymbol{p} \in \mathbb{R}^m$. It is natural to place such scalar objectives all at the same priority level. However, nothing prevents to put together at the same priority scalar objectives pertaining to different variables.

After the above procedure has been followed, for each priority level k we have:

- $\dot{\bar{\boldsymbol{x}}}_k \triangleq \begin{bmatrix} \dot{\bar{x}}_{1,k} & \cdots & \dot{\bar{x}}_{m,k} \end{bmatrix}$ is the stacked vector of all the reference rates;
- $\boldsymbol{A}_k \triangleq \mathrm{diag}(a_{1,k}, \cdots, a_{m,k})$ is the diagonal matrix of all the activation functions;
- $\boldsymbol{J}_k$ is the Jacobian relationship expressing the current rate-of-change of k-th task vector $\begin{bmatrix} \dot{x}_{1,k} & \cdots & \dot{x}_{m,k} \end{bmatrix}$ w.r.t. the system velocity vector $\dot{\boldsymbol{y}}$.

To the aim of finding the system velocity reference vector $\dot{\bar{\boldsymbol{y}}}$ that satisfies the above expressed priority requirements between the different objectives, the following task-priority Inverse Kinematic procedure is proposed:

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

where S_{k-1} is the manifold of solutions of all the previous tasks in the hierarchy. The notation $\mathrm{R\text{-}}\min$ underlines the fact that the minimization process is performed in a special regularized manner, necessary for managing, with a suitable continuity, the algorithmic singularities that can arise when at least a task (i.e. a row in the argument of the above quadratic forms) is in its transition zone, i.e. when its associated activation function assumes values in between of its extreme ones, 0 and 1. Such a regularization mechanism is duly reported in [9], together with the definition of the special pseudo inverse operator $(\cdot)^{\#,\boldsymbol{A},\boldsymbol{Q}}$ and will be omitted here.

This methodology (named *iCAT task priority framework*) results in the following algorithm, initialized with

$$\boldsymbol{\rho}_0 = \mathbf{0}, \quad \boldsymbol{Q}_0 = \boldsymbol{I}, \tag{6}$$

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

$$\begin{aligned}
&\boldsymbol{W}_k = \boldsymbol{J}_k \boldsymbol{Q}_{k-1} (\boldsymbol{J}_k \boldsymbol{Q}_{k-1})^{\#,\boldsymbol{A}_k,\boldsymbol{Q}_{k-1}}, \\
&\boldsymbol{Q}_k = \boldsymbol{Q}_{k-1} (\boldsymbol{I} - (\boldsymbol{J}_k \boldsymbol{Q}_{k-1})^{\#,\boldsymbol{A}_k,\boldsymbol{I}} \boldsymbol{J}_k \boldsymbol{Q}_{k-1}), \\
&\boldsymbol{\rho}_k = \boldsymbol{\rho}_{k-1} + \mathrm{Sat}\left(\boldsymbol{Q}_{k-1} (\boldsymbol{J}_k \boldsymbol{Q}_{k-1})^{\#,\boldsymbol{A}_k,\boldsymbol{I}} \boldsymbol{W}_k \left(\dot{\bar{\boldsymbol{x}}}_k - \boldsymbol{J}_k \boldsymbol{\rho}_{k-1}\right)\right),
\end{aligned} \tag{7}$$

where

- $\boldsymbol{\rho}_k$ is the control vector, which is computed in an iterative manner by descending the various priority levels;
- $\left(\dot{\bar{\boldsymbol{x}}}_k - \boldsymbol{J}_k \boldsymbol{\rho}_{k-1}\right)$ is the modified task reference that takes into account the contribution of the control vector $\boldsymbol{\rho}_{k-1}$ established at the previous iteration;
- $\boldsymbol{Q}_{k-1}$ is the projection matrix that is used to take into account the control direction (totally or partially) used by the higher priority tasks;
- $\boldsymbol{W}_k$ 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 $\left(\dot{\bar{\boldsymbol{x}}}_k - \boldsymbol{J}_k \boldsymbol{\rho}_{k-1}\right)$ to avoid discontinuities between priority levels;
- The function Sat$(\cdot)$ implements the saturation proposed in [11].

F. Actions

After the core algorithms have been presented, let us introduce the concept of *action*. An action is simply defined as a specific list of prioritized objectives. In the DexROV case, a navigate to waypoint action could be constituted by the following list of objectives

1) Vehicle minimum altitude;
2) Vehicle obstacle avoidance;
3) Vehicle horizontal attitude;
4) Vehicle Auto-heading;
5) Vehicle position.

With the above example, let us highlight the following remarks:

- the possible conflict between the altitude and the auto-depth objective (implicitly considered in the vehicle position one), is managed due to the higher priority of the altitude objective. In fact, the procedure outlined in the previous section will enforce this higher priority, allowing the vehicle to maintain a minimum altitude even if the depth setpoint were below the seafloor;
- the priority of the objectives follows a natural order: safety first, then operational enabling objectives (horizontal attitude, auto-heading), the actual mission objective (vehicle position).

In general, the objectives can be clustered in the following categories:

- 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.

Another possible action devised for the DexROV project is grasping an object from the seafloor:

1) Vehicle minimum altitude;
2) Vehicle obstacle avoidance;
3) Arm joint limits;
4) Arm manipulability;
5) Vehicle horizontal attitude;
6) End-effector linear position control;
7) End-effector angular position control;
8) Arm preferred shape;

Again, we see that the objectives are listed according to a priority order similar to the above one. In addition, we have an example of optimization objective (arm preferred shape). In fact, it is at lower priority than the objectives required for grasping an object (end-effector position control objectives), as its role is only to maintain the arm in a preferred configuration, but that is not a strict requirement.

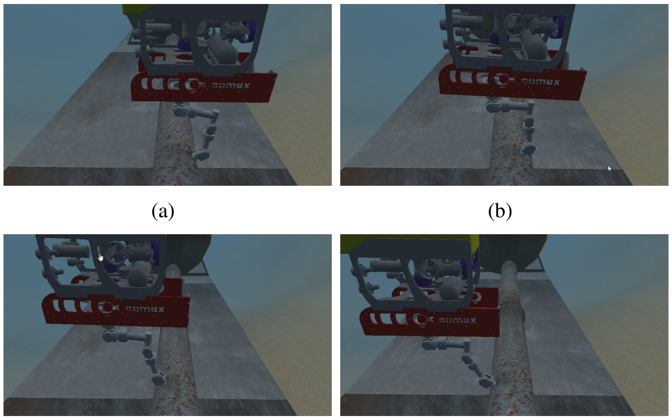
Finally, one of the reference missions of the DexROV project is the inspection of a pipeline. An electromagnetic sensor placed at the end-effector needs to be put in contact with the pipeline's weld and follow it in order to discover any possible cracks or leaks. The pipeline inspection action results to be:

1) Force regulation;
2) Arm Joint limits;
3) Arm Manipulability;
4) End-effector alignment with the surface's normal;
5) Vehicle horizontal attitude;
6) End-effector linear position control;
7) End-effector angular position control;
8) Arm preferred shape;

where the force regulation objective, since pertaining to the physical constraint type, takes the top of the hierarchy.

G. Missions

Following the general description of the task priority control procedure and the examples given in the previous section, the following aspects can be highlighted:



(a) (b) (c) (d)

Fig. 2. Screenshots of the UVMS as it performs the pipe welding inspection task.

- Although actions can in-principle be organized by prioritizing very large varieties of objectives, in practice they can always be reduced to a reasonable number of them, each one including all the safety/operational-enabling objectives reasonably expectable within a given work-field. The main difference between the actions will be the objective(s) to achieve a specific goal (e.g. the end-effector alignment with the surface's normal is clearly something that defines the pipeline inspection action).
- Actions can be temporally sequenced accordingly with a mission plan represented by a decisional action graph, having the actions as nodes and the arcs as logic decision alternatives, with just a few parameters of the main action-defining objectives (i.e. the force the exert).
- Moreover, transitions from an action to another one located at the end of a selected arc, should be smoothly activated.

Linked to this last point, let us consider the following two actions, composed by objectives abstractly labelled with alphabetic letters, where $A \prec B$ denotes that A has higher priority than B:

$$\begin{aligned} \text{Action1}: &\quad A \prec B, C, D \\ \text{Action2}: &\quad A \prec D \prec C, E \end{aligned}$$

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:

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

It is clear that, through insertion/deletion of some of the entries, the two original lists can be reconstructed. To do so, the activation function can be exploited. In particular, the $a^p(\boldsymbol{e})$ in (4) can be conveniently parametrized by the previous and current action being executed, and the time elapsed in the current action, to obtain the desired activation/deactivation smooth transition.

III. Simulation Results

In this section we present some simulation results of the proposed kinematic control strategy. The reference mission is the inspection of a pipeline. A reference path is defined without an a priori knowledge of the pipe, and the expected results is that the end-effector follows the path projected on the pipe's surface. In all the simulations we have simulated a frictionless multi-point contact with the end-effector planar surface. All the simulated forces and moments are then transferred to a unique point on the end-effector's rigid body space, where we have assumed that a force/torque sensor has been placed. Some screenshots of the UVMS executing the inspection mission are shown in Figure 2. The task control hierarchy that has been implemented in the simulations is the one reported in section II-F.

The parameters used for the simulation are the following ones.

- Dynamic simulation and dynamic control loops were running at 1 kHz frequency.
- Dynamic control was based on separate independent PI loops, tuned around the nominal inertia of the links and vehicle.
- We have not simulated the dynamics of the thrusters, but assumed that the vehicle could be controlled with a generalized force/torque vector.
- Vehicle generalized forces were saturated at 400 N on the surge axis of the vehicle, and 300 N along the sway and heave axes; vehicle generalized torques was saturated at 150 Nm on each axes.
- Each arm link inertia was modelled taking into account mass and reduction gears.
- Arm torques were saturated at $\begin{bmatrix} 45 & 125 & 50 & 50 & 10 & 10 & 4 \end{bmatrix}$ Nm (values are taken from the Graal Tech UMA commercial arm, see [13]).
- Kinematic control loop was running at 100 Hz frequency, based on the proposed task priority approach with velocity saturations.
- The considered velocity saturations were 1 rad/s for the arm joints, 0.2 m/s for the vehicle linear velocity and 0.2 rad/s for the vehicle angular velocities.

In Fig. 3(a) it is possible to see how the force regulation is accomplished quite well (the reference value was 10 N), despite being done at kinematic level and despite the activation of different joint limits, as can be seen in Fig. 3(d), and other tasks such as the arm preferred shape (see Fig. 3(e)). The reference system velocities can be seen in Fig. 3(b) for the arm, and in Fig. 3(c) for the vehicle, where the saturation mechanism can be seen. Finally, Fig. 3(f) shows the performance of the system in following the path over the weld.

IV. Conclusions

This paper has presented the general control framework employed within the DexROV H2020 project. In particular, the

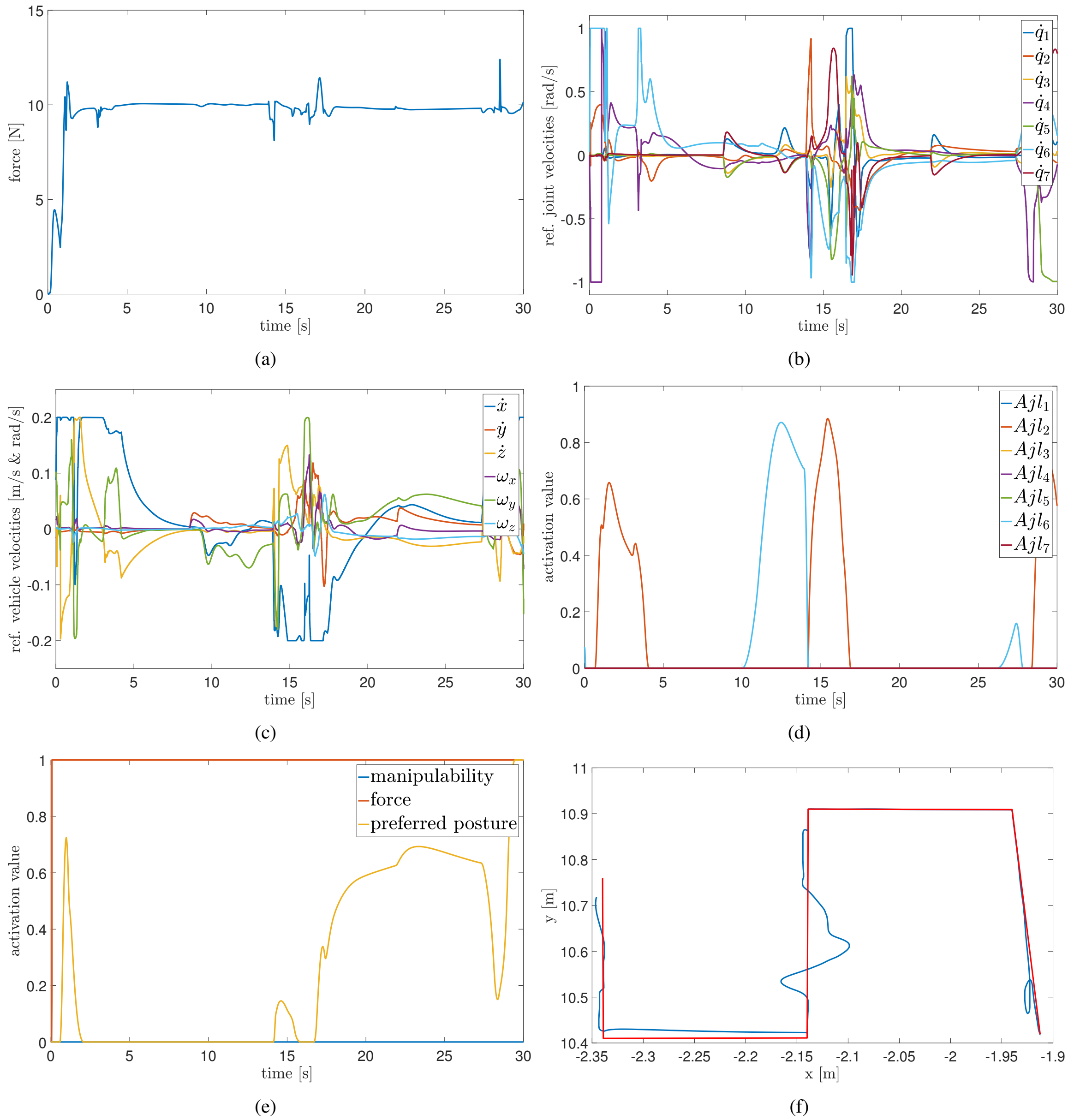


Fig. 3. Simulation results: (a) the time behaviour of the norm of the force (reference value is 10 N), (b) the arm joint velocities (saturation value at 1 rad/s) (c) vehicle reference velocities (saturation value at 0.2 rad/s and 0.2 m/s) (d) the activation values of the joint limit tasks, (e) the activation values of the other tasks, (f) reference path (red) and actual trajectory (blue).

core concepts of control objectives, control task and actions have been presented.

A reference mission of inspection of a pipeline's weld has been presented, as it is a very challenging task to be done autonomously, due to the physical interaction with the subsea structure. The simulation shows that, even with the force regulation task at kinematic level, the sensor needed to inspect the weld is maintained in contact with the pipeline with the required force.

Current works are focusing on the modelling of the actuators dynamics, to further increase the fidelity of the simulations. Furthermore, on-going efforts are dedicated to preliminary wet tests of the proposed KCL.

Acknowledgment

This work has been supported by the European Commission through H2020-BG-06-2014-635491 DexROV project and through H2020-SC5-2015-one-stage-690416 ROBUST project.

References

[1] J. Gancet, P. Weiss, G. Antonelli, M. F. Pfingsthorn, 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 dexrov project," in *10th IFAC Conference on Control Applications in Marine Systems*, vol. 49, no. 23, IFAC. Trondheim, Norway: Elsevier, 2016, pp. 414–419.

[2] A. K. Tanwani and S. Calinon, "Learning robot manipulation tasks with task-parameterized semitied hidden semi-markov model," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 235–242, 2016.

[3] P. J. Sanz, P. Ridao, G. Oliver, G. Casalino, Y. Petillot, C. Silvestre, C. Melchiorri, and A. Turetta, "Trident an european project targeted to increase the autonomy levels for underwater intervention missions," in *Oceans-San Diego, 2013*. IEEE, 2013, pp. 1–10.

[4] 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.

[5] ——, "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.

[6] G. Casalino, M. Caccia, S. Caselli, C. Melchiorri, G. Antonelli, A. Caiti, G. Indiveri, G. Cannata, E. Simetti, S. Torelli, A. Sperind, 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.

[7] 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.

[8] G. Antonelli and E. Cataldi, "Basic interaction operations for an underwater vehicle-manipulator system," in *ICAR 2015 - 17th International Conference on Advanced Robotics*, Istanbul, Turkey, July 2015.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] D. Ribas, P. Ridao, A. Turetta, C. Melchiorri, G. Palli, J. Fernandez, and P. Sanz, "I-AUV Mechatronics Integration for the TRIDENT FP7 Project," *Mechatronics, IEEE/ASME Transactions on*, vol. PP, no. 99, pp. 2583–2592, 2015.