Satellite-based tele-operation of an underwater vehicle-manipulator system. Preliminary experimental results

Paolo Di Lillo, Daniele Di Vito, Enrico Simetti, Giuseppe Casalino, Gianluca Antonelli

Abstract—Within the European project DexROV the topic of underwater intervention is addressed. In particular, a remote control room is connected through a satellite communication link to surface vessel, which is in turn connected to an UVMS (Underwater Vehicle-Manipulator System) with an umbilical cable. The operator may interact with the system using a joystick or exoskeleton. Since a direct teleoperation is not feasible, a cognitive engine is in charge of handling communication latency or interruptions caused by the satellite link, and the UVMS should have sufficient autonomy in dealing with low level constraints or secondary objectives. To this purpose, a task-priority-based inverse kinematics algorithm has been developed in order to allow the operator to control only the end effector, while the algorithm is in charge of handling both operative and joint-space constraints. This paper describes some preliminary experimental results achieved during the DexROV campaign of July 2017 in Marseilles (France), where most of the components have been successfully integrated and the inverse kinematics nicely run.

I. INTRODUCTION

Underwater intervention is needed by several applications ranging from interaction with structures belonging to the oil & gas industry to archaeology, from mining applications to collections of biological samples. Several national (MARIS [1], RAUVI [2]) and international (TRIDENT [3], PAN-DORA [4], ROBUST [5]) projects have been funded in the last few years on this important topic.

Within the European H2020 project DexROV [6], [7], the researchers are investigating the possibility to reduce the number of crew on board of the vessel by creating a remote control room linked by satellite communication to the UVMS (Underwater Vehicle-Manipulator System). The operator may interact with the system by joystick or exoskeleton and a proper cognitive tool is in charge of handling communication latency or interruptions caused by the satellite link.

The time delay and the satellite communication low bandwidth force the operator to share the control with the UVMS, that has to be capable of performing autonomously part of the needed operations. While the operator controls the end-effector motion, the UVMS control system takes care of all the safety-related tasks, both in operative and joint space. This kind of control is achieved by resorting to a multi-task-priority inverse kinematics framework that allows to perform multiple tasks simultaneously. The key aspect of this approach is to define a priority among tasks, creating a hierarchy in which the position of a task is relative to its importance. Usually the highest-priority tasks related to the safety of the system, e.g. avoiding obstacles or mechanical joint limits, leaving the operational tasks such as the end-effector position and orientation at a lower priority level. These considerations lead to solutions as in [8], [9] [10], where secondary control objectives were defined and handled in priority using the null-space projector, later extended in [11] to multiple tasks. In [12] a different approach is presented that is robust to the algorithmic singularities occurring when tasks are incompatible with each other. Such a work has been then extended to multiple tasks in the singularity robust multi-task priority inverse kinematics framework in [13] [14], [15]. The aforementioned framework has been developed to handle control objectives in which the goal is to bring the task value to a specific one, e.g. moving the arm end-effector to a target position. This kind of tasks are usually referred as equality-based. However, several control objectives may require their value to lie in an interval, i.e. above a lower threshold and below an upper threshold. These are usually called set-based tasks. Classic examples of set-based tasks for a robotic manipulator are the mechanical joint limits, the obstacle avoidance and arm manipulability tasks. In the last years, a great effort has been made in order to extend task-priority frameworks to handle set-based tasks, as for example done in [16]. In particular, the singularity-robust multi-task priority inverse kinematic framework has been extended to handle set-based tasks in [17], [18].

In this paper some positive, preliminary experimental results achieved during the DexROV campaign of July 2017 in Marseilles (France) are shown. Figure 1 shows the UVMS during deployment and Fig. 2 depicts a graphical rendering of the two manipulators. Most of the components have been successfully integrated and the inverse kinematics nicely run. In particular, during the wet tests, the following constraints were simultaneously handled: mechanical joint limits and smart joint-space velocity saturation [19]. The robot has followed both pre-programmed and joystick-driven trajectories generated on board the vessel (Marseilles), and trajectory generated with the exoskeleton in Brussels (Belgium). Finally, some tests were designed to intentionally move the arm to reach kinematic singularities.

II. DEXROV CONCEPT

DexROV is an EC (European Commission) Horizon 2020 funded project that aims to develop a system able to perform underwater operations using a novel paradigm that allows...
the far distance teleoperation of a ROV (Remotely Operated Vehicle) via a satellite communication. This would lead to the usage of a smaller and cheaper support vessel, since a part of the crew would be located in an onshore control center. Satellite communications introduce a non-negligible delay that has to be properly handled by the system in order to effectively perform the needed operations. The latency mitigation strategy includes a simulation environment and a cognitive engine. The operator interacts with the ROV in the simulation environment that receives 3D data from the perception system, without taking into account time latencies. He/She performs the desired movements with a force-feedback exoskeleton, instructing a cognitive engine that generates motion and manipulation primitives to be sent to the real ROV. Figure 3 represents the project’s concept.

The perception system makes use of a stereo camera for the 3D data acquisition, online processing of the needed information and its transmission to the control center [20]. Furthermore the ROV is equipped with an AHRS (Attitude and Heading reference System), a DVL (Doppler velocity log) and a USBL (ultra-short baseline) that are concurrently used for its accurate pose estimation [21]. The cognitive engine is split in two parts: on the onshore side it recognizes the actions that the operator wants to perform learning from demonstrations; on the offshore side it reconstructs the motion primitive despite of the non homogeneous communication latency. This is achieved by exploiting a task parametrized Gaussian Mixture Model that adapts the reference end-effector trajectory to the dynamic environment in which the ROV operates [22].

III. SET-BASED TASK-PRIORITY INVERSE KINEMATICS

A generic task is a function of the system state $\sigma(\eta)$. It is possible to divide these tasks in two main groups: equality-based tasks and set-based tasks. In equality-based tasks the control objective is to bring the task value to a desired one, for instance to move the end-effector in a specific position; in set-based tasks the control objective is to keep the task value within a range of values, for instance to keep the joints within its mechanical limits or the end-effector beyond a threshold distance from an obstacle.

Given a generic $n$-dimensional equality-based task $\sigma$, the system velocity that fulfils it can be computed by resorting to the Closed-Loop-Inverse-Kinematics algorithm:

$$\dot{q} = J^\dagger K \bar{\sigma}$$

where $J^\dagger$ is the Moore-Penrose pseudoinverse of the task Jacobian matrix [23], defined as

$$J^\dagger = J^T (JJ^T)^{-1}$$

in which $K$ is the gain matrix and $\bar{\sigma} = \sigma_d - \sigma$ is the task error. It is possible to perform multiple tasks simultaneously, setting a priority to each task and then filtering out the velocity contribution given by a low-priority task that would influence a high priority one. This is usually done exploiting the null-space projection through the matrix:

$$N = I_n - (J^\dagger J)$$

where $n$ is the number of DoFs (Degrees of Freedom) of the system and $I_n$ is the identity matrix.

Given a hierarchy composed by $k$ prioritized tasks, the system velocity can be computed by resorting to the Null-Space-Based Inverse Kinematics control [24]:

$$\dot{q} = \dot{q}_1 + N_{1,1} \dot{q}_2 + \cdots + N_{1,k-1} \dot{q}_k$$

where each $\dot{q}_i$ is the velocity contribution of the task $i$ obtained applying (1) and $N_{1,i}$ is the null space of the augmented Jacobian obtained by stacking all the tasks Jacobian matrices from $\sigma_1$ to $\sigma_i$. The NSB framework has been extended to handle also set-based tasks. This is possible by considering each set-based task as an equality-based one that can be activated and deactivated in function of the operating conditions. In particular, a set-based task has to be activated and deactivated in function of the operating conditions. In particular, a set-based task has to be activated when its value exceeds the desired lower (upper) threshold $\sigma_{a,l}$ ($\sigma_{a,u}$), adding it to the hierarchy as a new equality-based task with $\sigma_{s,l}$ ($\sigma_{s,u}$) as desired value. Figure 4 shows the thresholds of a set-based task. Then it can be deactivated when the solution of the hierarchy that contains only the other tasks would push its value toward the valid set. A more detailed description of the activation/deactivation algorithm is given in [18].
IV. SET-UP DESCRIPTION

The UVMS is characterized by two twin 6-DoF arms. Table I describes their kinematics by the Denavit-Hartenberg convention together with the joint mechanical and velocity limits.

<table>
<thead>
<tr>
<th>joint</th>
<th>$a$ [m]</th>
<th>$\alpha$ [°]</th>
<th>$d$ [m]</th>
<th>$\theta$ [°]</th>
<th>$q_{\text{min}}/q_{\text{max}}$ [°]</th>
<th>$q_{\text{max}}$ [°/sec]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0</td>
<td>90</td>
<td>0.3065</td>
<td>90</td>
<td>$\pm 119$</td>
<td>12.17</td>
</tr>
<tr>
<td>2</td>
<td>0.4631</td>
<td>0</td>
<td>0.0</td>
<td>68.5317</td>
<td>$\pm 110$</td>
<td>13.02</td>
</tr>
<tr>
<td>3</td>
<td>0.0</td>
<td>-90</td>
<td>0.0</td>
<td>-68.5317</td>
<td>$\pm 110$</td>
<td>11.7</td>
</tr>
<tr>
<td>4</td>
<td>0.0</td>
<td>90</td>
<td>0.437</td>
<td>0.0</td>
<td>$\pm 170$</td>
<td>12.3</td>
</tr>
<tr>
<td>5</td>
<td>0.0</td>
<td>-90</td>
<td>0.0</td>
<td>0.0</td>
<td>$\pm 110$</td>
<td>11.9</td>
</tr>
<tr>
<td>6</td>
<td>0.0</td>
<td>0</td>
<td>0.2695</td>
<td>0.0</td>
<td>$\pm 170$</td>
<td>17.8</td>
</tr>
</tbody>
</table>

The control has been developed in C++ as an independent class, guaranteeing a complete modularity in terms of usage. In particular for DexROV it has been wrapped in a ROS node [25], that takes a desired end-effector trajectory and publish the output joint velocities on separate topics. This design allows to use the same code in all the development and validation stages, from the laboratory testing to the real usage underwater. During development, the node has been interfaced with a graphical simulator developed under Gazebo [26], that perfectly replicates all the interfaces among them, giving the possibility to test all the chain from the control center to the UVMS. During the real test, the same node has been interfaced with the real system, without requiring any change in the control node’s code. This design is very helpful in field trials where the operating conditions are not always perfect and it is difficult to make modifications on the fly. The control implementation exhibits the same flexibility also on the input side. During the real DexROV operations, the system takes the desired end-effector trajectory from the operator wearing an exoskeleton in the control center located in Brussels via satellite communications. However the software design allows to take the references directly from other pre-programmed software nodes or from a standard joystick located onboard the vessel or in the remote control center. This is very important for debug purposes, as it allows to exclude from the chain the control center or the satellite channel, focusing the attention on the control side. The control framework includes also a technique for the kinematic singularities handling, resorting to a Damped Least-Square pseudoinverse matrix [27], defined as:

$$ J_{DLS}^T = J^T (J J^T + \lambda^2 I_m)^{-1} $$

in which the damping coefficient $\lambda$ has the following expression:

$$ \lambda = \begin{cases} 0 & \text{if } \sigma_{\text{min}} \geq \sigma^* \\ \sqrt{\sigma_{\text{min}}(\sigma^* - \sigma_{\text{min}})} & \text{if } \sigma^*/2 \leq \sigma_{\text{min}} < \sigma^* \\ \sigma^*/2 & \text{if } \sigma_{\text{min}} < \sigma^*/2 \end{cases} $$

Fig. 3. DexROV concept, this paper focuses on the manipulation part

Fig. 4. Activation and safety thresholds of a set-based task
where $\sigma_{\text{min}}$ is the minimum singular value of $J$ and

$$
\sigma^* = \frac{||\tilde{\sigma}||}{||\dot{q}||_{\text{max}}}
$$

$||\tilde{\sigma}||$ being the task error norm and $||\dot{q}||_{\text{max}}$ is the maximum joint velocity norm [28]. Additionally, in case that the reference end-effector trajectory is too fast with respect to the joint velocity constraints of the arm, a method that properly scales the vector $\dot{q}$ has been implemented, following the algorithm described in [29].

V. Experimental results

Several tests have been executed with the system in different configurations, accepting the end-effector trajectories by different means, i.e., by local code or joystick, by remote code or joystick and finally by remote exoskeleton. In the remote configuration, the trajectory is generated in Brussels (Belgium) and then transmitted via satellite communications to the vessel in Marseilles (France) and then through the umbilical to the vehicle. Initially, during the early debugging phases, the driving commands were not directly sent to the physical system but the graphical simulator instead. Noticeably, the Brussels operator and the code running on board of the vessel are transparent to this configuration.

A. Position only

In a first test, the sole end-effector position, without orientation, is given as individual task. The desired trajectory sent to the controller is a simple circle on the $y$-$z$ plane in the arm base frame at a constant velocity, Fig. 5 shows the arm base frame within the simulator. Figures 6 and 7 show the position error and the joint positions.

B. Position and orientation, singular configuration

In the second test the end-effector position and orientation task is given. The desired trajectory is the same circle of the previous test, but keeping the orientation at a constant value. It is worth noticing that the manipulator intentionally reaches a singular configuration during the trajectory, as the minimum singular value reaches very small values. Figure 8 shows the minimum singular value of the $J$ matrix over time.

C. Mechanical joint limits

In the last experiment the system is asked to follow the same circular trajectory without controlling the orientation while keeping the fifth joint below a certain threshold. The prioritized task hierarchy imposed is:

1) Joint 5 maximum threshold
2) End-effector position

Figures 9 and 10 show the position and orientation error together with the joint positions during the experiment. The DLS pseudoinverse prevents the chattering phenomenon on the joint velocities, generating a higher error on the orientation while the position error remains sufficiently low during the whole trajectory.

Figures 11 and 12 show the position error and the joint values during the experiment. The null space projection and the activation/deactivation algorithm described in Section III make the joint position stay below the chosen threshold (in red), while the trajectory is followed with a low position error.

Then another joint limit has been added as control objective, giving the following hierarchy:

1) Joint 3 maximum threshold
2) Joint 5 minimum threshold
3) End-effector position
Fig. 8. Second experiment, position and orientation control: minimum singular value of $J$ over time. The arm intentionally reaches a singular configuration during the trajectory.

Fig. 9. Second experiment: position and orientation error over time. The position error is kept low during the entire trajectory, while the orientation error grows for the effect of the joint velocities damping.

Figure 13 shows the joint positions during the experiment, while Fig. 14 shows the position error. It is worth noticing that the third joint starting position is above the chosen maximum threshold, but the control algorithm quickly brings its value to the imposed limit. From that point, both the joint limits are satisfied during the entire trajectory. The position error grows with respect to the other experiment because the combination of the third and fifth joint mechanical limits, being at a higher priority level with respect to the position task, reduces the end-effector operational workspace.

VI. CONCLUSIONS

In this paper preliminary results of the application of task-priority based inverse kinematics for UVMSSs in accomplishment of the European H2020 project DexROV have been presented. Experiments on different task hierarchies including set-based and equality-based tasks have been described and the algorithm robustness with respect to the occurrence of kinematic singularities has been successfully tested. The results were satisfactory and promising for the full-scale experiment schedule for summer 2018.

ACKNOWLEDGE

This work was supported by the European Community through the projects EUMR (H2020-731103-2), ROBUST (H2020-690416) and DexROV (H2020-635491).

REFERENCES

Fig. 12. Third experiment, 5th joint mechanical limit and position control: position error over time.

Fig. 13. Fourth experiment, 3rd and 5th joints mechanical limit and position control: joint positions and minimum/maximum thresholds (in red).

Fig. 14. Fourth experiment, 3rd and 5th joints mechanical limit and position control: position error over time. Notice that the error is large due to the constraints and intentional low feedback gains.