MoveIt!

Autonomous Underwater Free-Floating Manipulation

© PHOTOCREDIT

By Dina Youakim, Pere Ridao, Narcis Palomeras,
Francesco Spadafora, David Ribas, and Maurizio Muzzupappa

Today, autonomous underwater vehicles (AUVs) are mostly used for survey missions, but many existing applications require manipulation capabilities, such as the maintenance of permanent observatories, submerged oil wells, cabled sensor networks, and pipes; the deployment and recovery of benthic stations; or the search and recovery of black boxes. Currently, these tasks require the use of work-class remotely operated vehicles (ROVs) deployed from vessels equipped with dynamic positioning, leaving such solutions expensive to adopt. To face these challenges during the last 25 years, scientists have researched the idea of increasing the autonomy of underwater intervention systems.

Related Work

Pioneering work involving underwater vehicle-manipulator system (UVMS) technology first appeared in the early 1990s (the list of references to all of the projects cited in this article as well as a broader introduction to the subject, can be found in [1]), with the Ocean Technology Testbed for Engineering Research at the Monterey Bay Aquarium Research Institute, Omni-Directional Intelligent Navigator at the University of Hawaii, and Vortex at the Ifremer Mediterranean Center, but it was not until the first decade of the 21st century that field demonstrations arrived. Experimental results have demonstrated two types of applications.

Object Search and Recovery

The first result based on floating manipulation was achieved in 2009 with the semi-AUV for intervention missions project that demonstrated the capability of searching for an object whose position was roughly known a priori. The object was endowed with artificial landmarks, and the robot autonomously located and hooked it with a recovery device while hovering. Because a six-ton vehicle was used, the mass of the arm did not cause significant disturbances. In 2012, the same task was approached in the reconfigurable

Digital Object Identifier 10.1109/MRA.2016.2636369
Date of publication: 19 April 2017

1070-9932/17©2017IEEE

AUV for intervention project using a lighter vehicle (<200 kg). First, the object was searched using a downward-looking camera and photomosaicing techniques. Next, the object was hooked autonomously in a water tank. Later, during the Trident project, the experiment was extended in a harbor environment using the seven-degrees-of-freedom (7-DoF) arm endowed with a three-fingered hand. All of these projects relied on variations of the task priority redundancy control framework [2].

Intervention on Subsea Panels

Given the importance of inspection, maintenance, and repair tasks for the offshore industry, representative tasks usually performed by ROVs, such as valve turning and connector plug/unplug, have been automated using different approaches.

Fixed-Base Manipulation

The first fully autonomous intervention at sea was demonstrated by the Autonomous Light Intervention Vehicle 1.5-ton I-AUV. Mechanical scanning imaging sonar was used to locate and hone in on a subsea panel, and visual servoing techniques were used for docking the vehicle using two hydraulic grasps. Once the vehicle was docked, a hydraulic 7-DoF manipulator was used to open/close a valve. A similar approach was proposed in the trusted vessel information from trusted on-board instrumentation project. An active localization strategy using sum-of-Gaussian filter and range measurements to the subsea panel was used to discover its position and to then hone in on the panel. Next, visual servoing methods based on the a priori-known appearance of the panel were used to autonomously dock the robot into a funnel-shaped docking station. Besides demonstrating the valve-turning task, a first autonomous demonstration of a connector plug/unplug operation was also carried out.

Free-Floating Base Manipulation

With this approach, the manipulation task takes place while the AUV is floating; hence, the arm motion may potentially disturb the AUV pose. The first autonomous free-floating valve turning was carried out based on a learning-by-demonstration paradigm within the Pandora Project. More recently, we have used a task priority redundancy control approach at the kinematics level [1] for the same purpose.

Contribution

The work reported in the literature on underwater free-floating manipulation is based either on variations of the task priority framework or the use of learning by demonstration. Although there have been significant advances toward fully autonomous underwater intervention, today's technology is still far from having the capabilities already demonstrated by terrestrial robots, in which motion planning techniques have been extensively introduced. There are no reported results in the literature on seamlessly integrating motion planning, manipulation, three-dimensional (3-D) perception, kinematics, control, and navigation technologies to control UVMSs to perform manipulation tasks while moving in the presence of obstacles. This is the case even though obstacle-avoidance skills are necessary to safely operate in submerged infrastructures.

MoveIt! mobile manipulation software [3], a framework integrating all of these technologies, has been successfully applied to industrial, single/dual arms, mobile manipulators, and humanoid robots, although not yet to intervention AUVs (I-AUVs). The main focus of this article is the evaluation of the use of MoveIt! motion planning capabilities to control a UVMS and demonstrate beyond-state-of-the-art tasks, including valve turning in the presence of obstacles and connector plug/unplug, both on a free-floating base for the first time.

The UVMS

An 8-DoF GIRONA500 I-AUV was used for the experimental validation of this work.

The AUV

GIRONA500 [4] is a compact-size AUV designed and developed at the University of Girona for a maximum operating depth of 500 m. The overall dimensions of the vehicle are 1 m in height, 1 m in width, and 1.5 m in length. The weight is about 140 kg. It has two upper hulls that are positively buoyant, and a lower hull contains the heavier elements such as the batteries and the payload. This arrangement provides the vehicle with passive stability in pitch and roll, making it suitable for tasks requiring a stable platform, such as video surveying or intervention. Its most remarkable feature is its reconfiguration ability. On its basic configuration, the vehicle is equipped with typical navigation sensors [Doppler velocity log (DVL), attitude and heading reference system (AHRS), pressure gauge, and ultrashort baseline] and basic survey equipment (profiler sonar, side scan sonar, video camera, and sound velocity sensor). Besides these sensors, almost half of the volume of the lower hull is reserved for mission-specific payload, which allows modifying its sensing and actuation capabilities as needed.

The Robot Arm

The robot arm is a 4-DoF ECA/CSIP microarm (Figure 1), with three revolute joints actuated by electric screw drives to control the (x, y, z) position of the terminal element. A fourth actuator controls the roll (ϕ) of the end effector (EE). The arm is underactuated and, therefore, each 3-D position can only be reached within a particular attitude. Full attitude control of the EE is only possible by the cooperative motion of the arm and the AUV. During the reported experiments, the arm was mounted on the lower frontal part of the vehicle. To reduce the perturbations caused by the motion of the manipulator, syntactic foam is embedded within its links to increase buoyancy, making it nearly neutral in water.

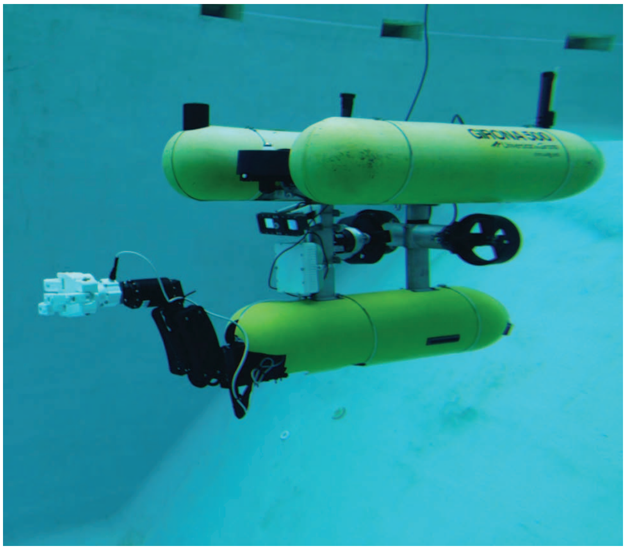


Figure 1. The GIRONA500 AUV mounted with an ECA/CSIP 4-DoF microarm and a three-finger hand.

The Three-Finger Hand

The arm gripper has limited manipulation capabilities due to its V shape, so, as a result, we developed a three-finger hand with a great degree of dexterity. The hand consists of two fingers and an opposing thumb. Such a design produces articulation by allowing full opening and folding of the fingers. Each finger is independent and consists of an actuation system that operates using servomotors and belt transmissions. This arrangement permits the support of different kinds of objects (e.g., various shapes or contours) within the geometric limits of the hand. As a safety consideration, three current sensors are used to limit the movement of the fingers if any of them exceeds the overload current. To decrease potential failure of the system due to the leaking of water inside the palm cavity, we opted for a transmission made of stainless steel tendons, with one for closing and one for opening of each finger. The kinematics mechanism per finger consists of a drive wheel connected to a servomotor. A toothed belt transfers the motion to the drive wheel through a rotating shaft and the torque (properly amplified by gearing) reaches the articulation of the finger. To guarantee pressure tolerance, we carried out a series of finite element method simulations to detect the areas to be reinforced. In addition, we adopted a design that withstands the pressure, even in the presence of relative motion between the parts. This analysis allowed us to define the exact depth of the grooves for the seals and the size and type of materials of the O-ring to be used. Given the considerable customization of the robotic hand, only some parts have been built by ordinary machining. For other components (e.g., fingers and gears), it was necessary to use additive manufacturing techniques and 3-D printing. The final version of the device was built with plastics and stainless steel powder using laser-sintering 3-D printers, and an Arduino board operates the electronic control of the hand.

AUV Software Architecture

GIRONA500 uses the component-oriented layered architecture, version 2 (COLA2) for its software architecture [5], integrating perception, localization, control, and guidance. The architecture is composed of several components organized in layers (see Figure 2).

Sensing and Actuation Layer

The sensing and actuation layer contains the sensor and actuator drivers. It uses an AHRS to measure the angular velocity (ν_2) and the robot attitude (η_2) and a DVL to measure the linear velocity (ν_1). A depth sensor included within a sound velocity profiler device is used to measure the depth (z), an acoustic modem measures ranges against another acoustic modem acting as a transponder (r), and a bumblebee stereo camera is used to gather stereo imagery ($I = \{I_1, I_2\}$). This layer also includes drivers for the thrusters controlled in speed.

Navigation Layer

The three localization components included are a navigation filter, a visual servoing block, and a single beacon navigation module. The navigation filter uses a constant velocity model with attitude input (η_2), depth (z), and linear velocity measurements (ν_1). Implemented as an extended Kalman filter (EKF), it is able to estimate the robot's position and velocity ($\overrightarrow{x_k} = [\eta_1^T \nu_1^T]^T$) by dead reckoning. This filter can be extended to a single landmark simultaneous localization and mapping system using a visual servoing component [6] to bind the localization drift. As input, the filter uses the camera images (I) and an a priori-known template of an object of interest (e.g., subsea panel) to estimate its pose $\left(L_\eta = [L_{\eta_1}^T L_{\eta_2}^T]^T\right)$ with respect to the robot. To hone in on the subsea panel, a transponder is also mounted on top of it. A single beacon localization component is then used to localize the panel. The single beacon navigation system uses an active localization strategy that uses a sum-of-Gaussian filter to estimate the beacon position (L_{η_1}) [7].

Guidance and Control Layer

The AUV control is based on a nested pose/velocity proportional–integral–differential (PID) controller. The velocity controller computes the force and torque (τ_d) that will be applied to reach the desired velocity (ν_d). This value is computed by combining a standard 4-DoF PID control with a feed-forward model, providing the nominal force to be applied to achieve a certain velocity. The output of the velocity controller (τ_d) is allocated to the thrusters using the thruster allocation matrix within the thruster control matrix component. Once the force to be exerted by each thruster is known, the static thruster model is used to convert the force into thruster set points. On its top, a pose controller handles AUV pose (η) regulation and tracking. This works by sending velocity set points (ν_d) to the velocity controller and reading the pose feedback (η) from the EKF component to achieve a desired pose (η_d).

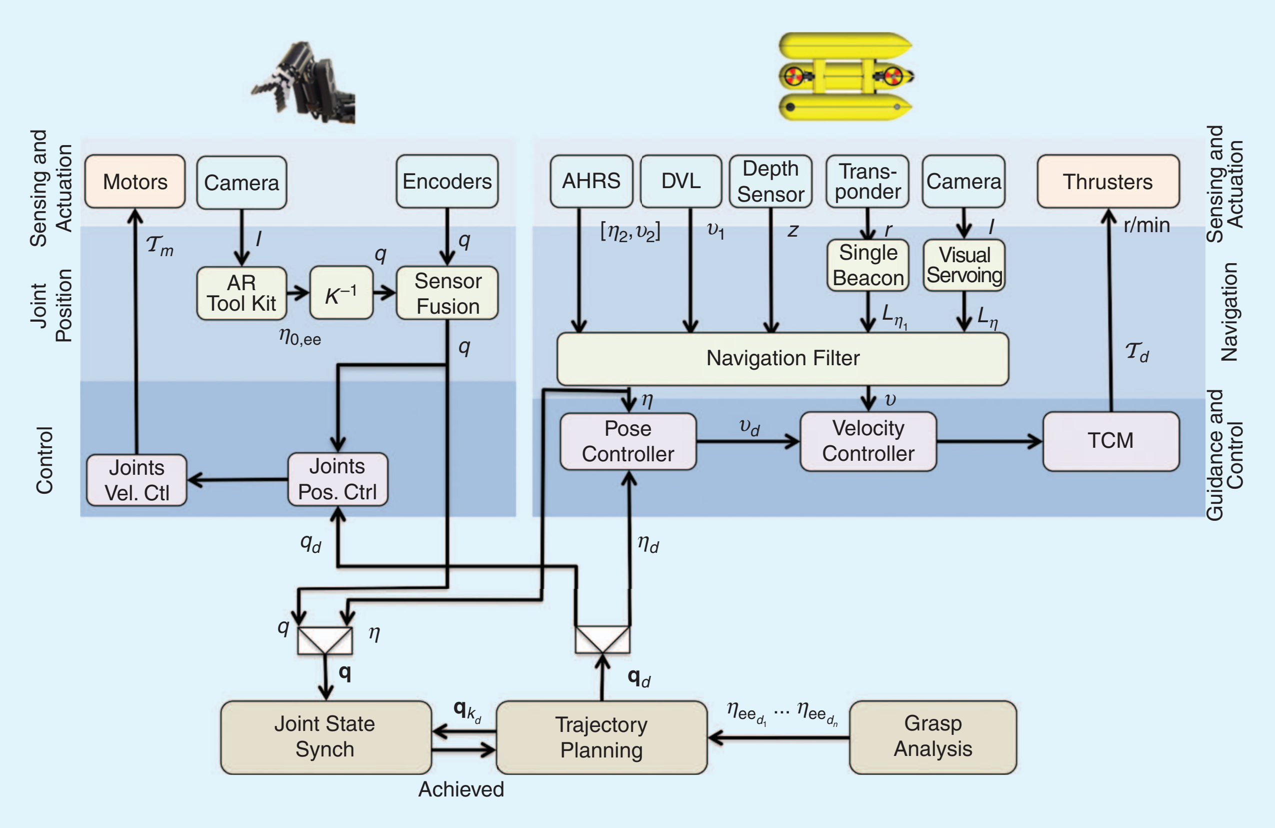


Figure 2. The GIRONA500 I-AUV software architecture. AR: augmented reality; TCM: tilt current meter.

Arm Software Architecture

Sensing and Actuation

The sensing and actuation layer provides access to the motor drivers, the joint encoders, and a camera used to estimate the EE pose through visual servoing.

Joint Position

Although the arm position in configuration space is instrumented with joint encoders, the arm may miss pulses during operation, which can lead to inaccurate calibration. To deal with this problem, the EE is labeled using an augmented reality (AR) code [8], whose pose can be easily estimated with a camera. Using the inverse kinematics, it is then possible to estimate the arm configuration (Q) used to calibrate the position read from the encoders.

Joint Control

A cascade joint position and velocity controller is used to control each arm joint. Hence, it is possible to control the joint velocity and/or position. It is also possible to control the EE velocity ($\dot{\eta}_{0,\text{ee}}$) in Cartesian space by means of the arm pseudoinverse Jacobian ($J^{\dagger}(\mathbf{q})$) that translates the Cartesian velocity into the joint velocity ($\dot{\vec{q}}$).

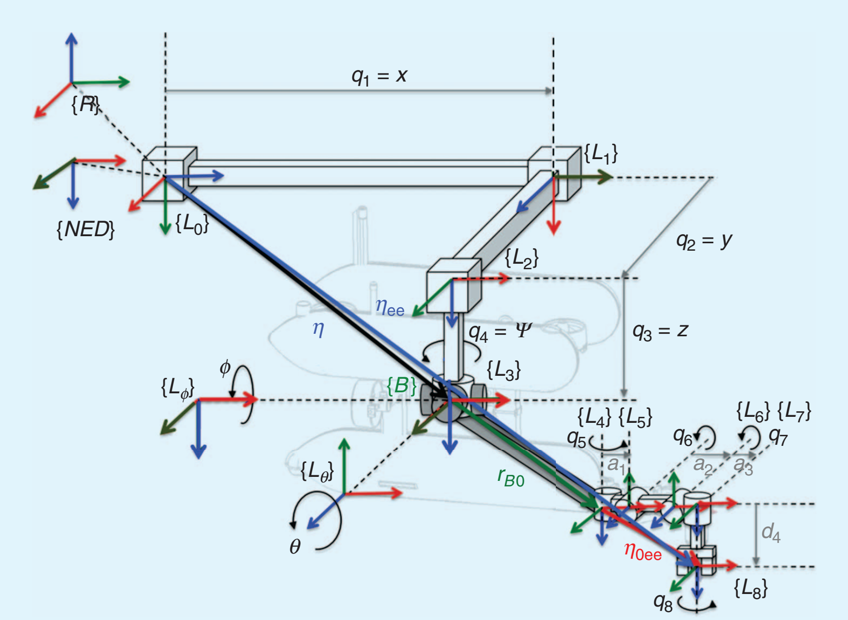
Coordination

This coordination layer is devoted to the joint control of the AUV and the arm. For this work, we implemented the coordination layer using MoveIt! As shown in Figure 2, the layer integrates three modules: a trajectory planning, joint-state synchronizer, and grasp analysis component. The trajectory planner computes collision-safe AUV-arm trajectories in the 8-DoF configuration space to achieve a desired EE pose, providing control set points to the AUV and robot-arm uncoupled controllers. In addition, the joint-state synchronizer component ensures that both the AUV and the arm have reached their own set points before scheduling the next one. Finally, the grasp analysis is included for grasp reachability analysis.

MoveIt!-Based I-AUV Implementation

MoveIt! [3], a software framework providing a wide functionality that covers several features of mobile manipulation, encapsulates a forward/inverse kinematics solver, planning techniques, and collision-detection methods through a plug-in-based mechanism. Widely used third-party libraries are already integrated, including the Kinematics and Dynamics Library (KDL), the Open Motion Planning Library (OMPL) [9], and the Fast Collision Check Library (FCL) [10]. To achieve MoveIt!'s main goals of extendability and reusability, custom configuration of these components can be combined through the plug-in system.

The modeling in MoveIt! uses unified robot description format (URDF) and complementary semantic robot description format (SRDF) extensible markup language (XML)-based files. The URDF defines the visual shape and the collision properties of the system, along with its



DoF	θ	d	a	α	Home
1	q_1	0	0.108	–90	0
2	q_2	0	0.231	0	0
3	q_3	0	0.1	90	0
4	q_4	0.308	0	0	0

Figure 3. The I-AUV schema and arm DH kinematics parameters.

kinematics parameters. With the SRDF, planning group(s) are defined by choosing the set of joints and links that act together as a single entity capable of performing planning and/or manipulation tasks by using its chosen kinematics solver and planner. Also in this file, the self-collision matrix is defined by choosing link pairs that are permitted to collide (e.g., adjacent links). The collision-checking library incorporates this information during path planning.

GIRONA500 UVMS Modeling

I-AUV Model H3

Figure 3 presents the I-AUV kinematics chain as modeled in MoveIt!. A north-east-down ({NED}) frame is used as the global reference frame. Collocated in the same point, but rotated, is the {R} frame that is used for visualization purposes. The robot pose is defined by ${}^{NED}\eta = [xyz\phi\theta\psi]^T$, providing the origin and attitude of the robot fixed frame {B} with respect to the {NED} frame. The arm mounting pose vector is given by ${}^{B}r_{B0}$, and the EE pose vector frame {n}, with respect to the robot base frame {0}, is given by ${}^{0}\eta_{ee}$. By compounding the robot pose ${}^{NED}\eta$, the arm mounting pose ${}^{B}r_{B0}$, and the EE pose ${}^{0}\eta_{ee}$, it is possible to compute the world-referenced EE pose ${}^{NED}\eta_{ee}$.

AUV Modeling

The AUV has 6 DoF, although the roll and pitch are passively stable. As a result, the vehicle is actuated only in 4 DoF (surge, sway, heave, and yaw). The AUV kinematics are modeled using the Denavit-Hartenberg (DH) method applied to the equivalent Cartesian manipulator, shown in Figure 3. For each link, a frame $\{L_i\}$ is defined. $\{L_0\}$ is placed with {NED} but with the attitude required by the DH algorithm to keep its motion axis along the z direction. Similarly, $\{L_3\}$ is placed with the robot body-fixed frame {B}, again with different attitude to follow the DH guidelines. The vehicle motion is modeled as three consecutive prismatic joints representing (X, Y, Z) displacements in the {NED} frame, and their motion is bounded by the size of the water tank, where the experiments were carried out. Followed by three revolute joints corresponding to (ϕ, θ, ψ) Euler rotations, two are noncontrollable (i.e., modeled as fixed joints) but included to account for the small motions that the AUV may experience due to system perturbations. The ψ joint is modeled as unlimited and continuous. The final configuration vector of the AUV is $Q = [q_1 q_2 q_3 q_4]^T$, representing the controllable part of the AUV pose vector ${}^{NED}\eta = [xyz\psi]^T$.

Arm Modeling

The arm is also modeled using the DH convention, and its links have associated frames $\{L_4, L_5, L_6, L_7, L_8\}$, with the first and the last associated to the arm base {0} and EE {ee}, respectively. The arm joint variables $Q = [q_5 q_6 q_7 q_8]^T$ are configured as revolute joints with their corresponding mechanical limits.

AUV-Arm Link Modeling

In Figure 3, a fixed virtual joint between the vehicle and the arm is introduced, defining a transformation between the AUV center and the arm base, thus linking both models together as one system. For visualization purposes, we used a stereolithography file derived from the I-AUV using computer-aided design. To account for collision-checking and safety measures, the AUV is represented by a square box defined as the shape of the last link in the AUV model, the yaw. Figure 4 shows both 3-D models. A simplified version of the URDF file used is available in [11]. In the SRDF, one planning group is defined, encapsulating all the controllable I-AUV links and joints (i.e., excluding the EE). The <group_state> tag, was used to predefine common configurations of the system (e.g., HOME, ARM FOLDED…).

Path Planning

MoveIt! is already integrated with the OMPL library, thus providing a wide choice among different path-planning algorithms. For the experiments reported in this article, we selected the rapidly exploring random tree (RRT) connect [12]. A simple yet efficient method that is suitable for high-dimensional configuration space, RRT connect works by growing two separated trees of paths rooted at the start and goal poses. The trees explore the space around them by progressing toward one another using a simple greedy heuristic until they join together, providing a solution to the query.

Manipulation

For each task, MoveIt! allows the definition of multiple candidate grasp poses. Each is sequentially considered in order according to a user-defined quality metric. Given a grasp pose, MoveIt! provides a pick and place pipeline consisting of three sequential stages: 1) reachable and valid (RA) pose, 2) approach and translate (AT), and 3) plan (P). During the first stage (RA), the collision-free pose accessibility (inverse kinematics) is checked. Next, the Cartesian path between the grasp pose and the pregrasp pose (located at a distance greater than a certain value in the opposite direction of the approach vector) is computed (AT). At this stage, the collision detection between the gripper and the object to grasp is disabled. Next, if a retreat pose is also specified, the corresponding Cartesian path is computed, going from the grasp to the retreat. Finally, in the P stage, the manipulation plan itself is computed through the concatenation of the previously computed trajectories.

Obstacle Avoidance

MoveIt! is able to plan collision-free trajectories using the FCL library. The following two types of obstacles exist:

- A priori-known static objects that are manually created, with already known dimensions and locations. MoveIt! has a determined set of geometric shapes representing simple objects. Examples of this type of obstacle include the water tank wall, panel, and valves (Figure 5).
- Unknown dynamic objects are created by integrating MoveIt! with some sensor data. An additional feature of MoveIt! is its capability to integrate with octomaps [13] because its world representation already includes an octotree to map the free/occupied voxels of the environment.

The allowed collision matrix, defining the allowed collisions between each object in the world and each link of the robot geometric model, can be modified according to the need. For example, in the valve-turning task, touching the valve with the gripper (i.e., gripper–valve collision) should be permitted to occur while avoiding collisions with the rest of the objects. In the experiment presented in this article, we assumed a priori-known static virtual obstacles. The locations of the obstacles were chosen to challenge the planning and manipulation tasks. Integrating with octomaps is part of our planned upcoming work.

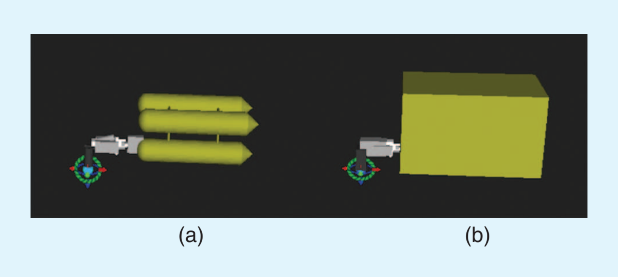


Figure 4. The I-AUV (a) visualization and (b) collision-checking models.

I-AUV Controller

MoveIt! provides a configurable controller manager to interface with the system's low-level controllers (in our case, the AUV and arm controllers). A MoveIt!-specific interface is implemented to perform the linkage between planning output and the controllers. The planning output is a message of the type [*moveit_msg/RobotTrajectory*], containing the pose, velocity, and acceleration vectors to be followed to reach the desired goal. First, the trajectory message is split into two parts: one for the vehicle and one for the arm joints. For each of these, the corresponding custom robot operating system (ROS) message is created as expected by COLA2 software architecture. Because synchronization between vehicle and arm motion is required, both types of messages are sent at the same frequency (10 Hz), which is the rate required by the vehicle to follow the requests. Moreover, to ensure that the real system reaches the desired positions, a feedback loop is introduced with the actual joint values. For each point in the trajectory received, the arm and the vehicle messages are continuously sent at the 10-Hz frequency, until a specified precision for all of the joints is simultaneously achieved. Then, the next UVMS set point is sent to the controllers, repeating the flow. This precision is configurable and differs for each joint, given their type and use (i.e., vehicle and arm joints).

Workspace Analysis

In the presence of obstacles, the planning may fail to find a safe path to reach a particular goal. Nevertheless, if alternative grasping poses exist, it is possible to evaluate them until a satisfactory pose is found. For the valve-turning

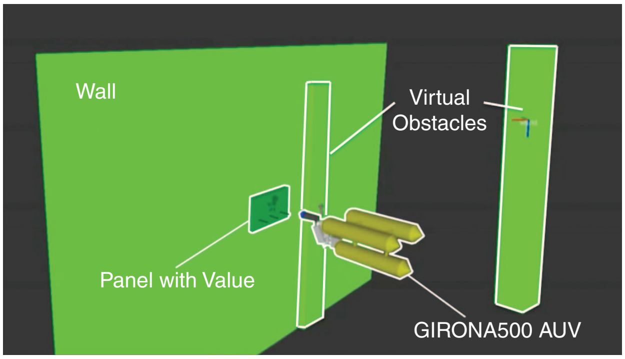


Figure 5. The obstacles scene in the valve-turning scenario.

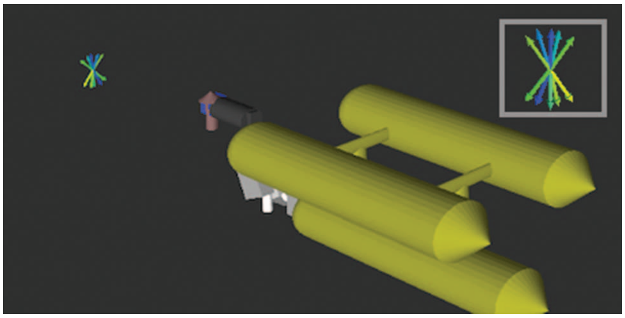


Figure 6. The workspace analysis result.

experiment, ideally, the two orthogonal orientations (positive and negative) are the best options. But it is possible to take advantage of the V-shaped gripper that allows a certain tolerance in the alignment between the gripper and the valve. Thus, within a few degrees more/less of the gripper orientation, it will still perform the job appropriately. Given the fact that the position of the valve is known, the key point is to compute the set of suitable EE orientations that may complete the valve-turning operation. To do so, all of the EE orientations (ϕ, θ, ψ) up to a certain discretization level are evaluated. Then, only the orientations satisfying the task constraint (i.e., valve orientation plus or minus a certain tolerance, ensuring that the gripper fits the valve's T shape) are kept. Next, the reachable poses leading to collision-free grasping are selected. Finally, to choose one from the remaining valid orientations, a grasp quality is assigned to each by computing its manipulability index (MI). (MI is a quality measure indicating how far/close the current system status is located from a singular configuration: the higher, the better, and the furthest the configuration is from singularity.) Figure 6 shows workspace analysis results for the valve turning. The different colors of the arrows show their MI values; the blue has a higher quality than the yellow and green.

Experiments and Results

Setup

Validation experiments were performed in a water tank. A subsea panel mock-up with four T-shaped valves and a hot-stab type of connector was hung on one wall (see Figures 7 and 8). The working space (Figure 5) includes two types of obstacles: 1) the walls of the water tank and the panel and 2) virtual obstacles included to test more challenging scenarios. As explained in "I-AUV Controller," a preset tolerance is permitted for each joint of the system. For the results presented in this section, the AUV tolerance is 6 cm for (X, Y), 2 cm for Z, and 0.06 rd for ψ. For the arm, the tolerance is 0.075 rd for $[q_1\ q_2\ q_3]$ and 0.3 rd for the wrist q_4.

Valve Turning

The intervention task is carried out through the following phases:

- *Detection*: In field operations, detection discovers the location of the panel and establishes visual contact [7], enabling visually based navigation with respect to the panel [14]. For the experiments reported, the robot is requested to move through a sequence of waypoints while avoiding obstacles; during the path following, the panel is localized.

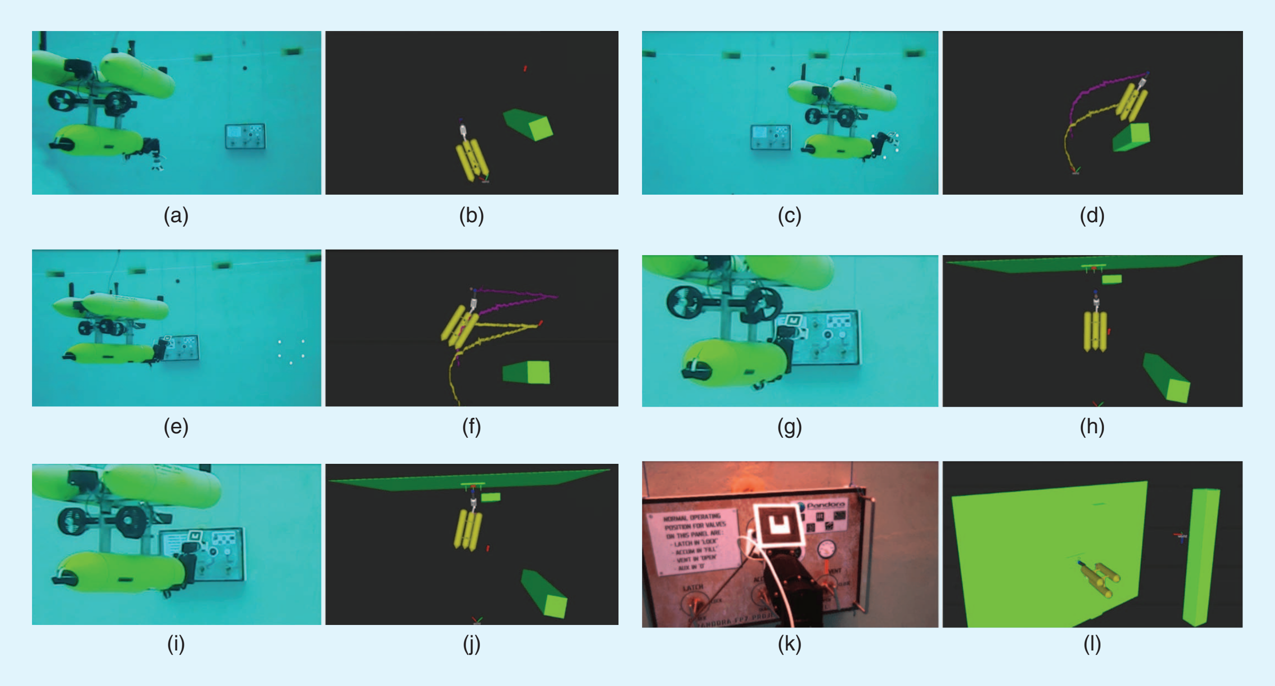


Figure 7. The valve turning: (a) and (b) start, (c) and (d) detection, (e) and (f) inspection, (g) and (h) approach, (i) and (j) grasp and turn, and (k) and (l) forward camera view.

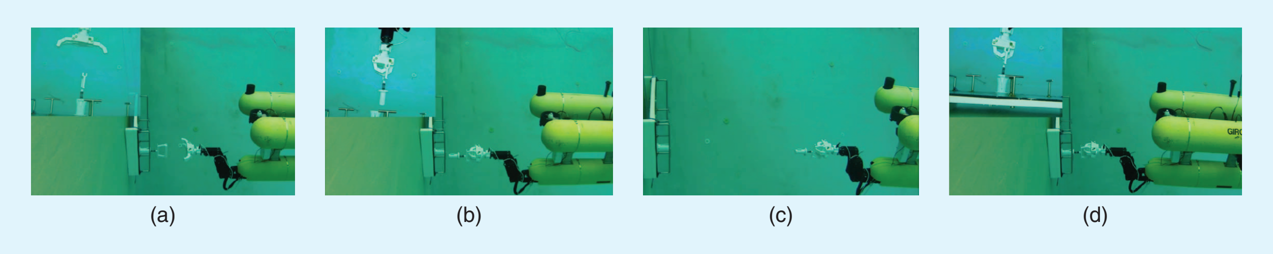


Figure 8. The connector plug/unplug: (a) approach, (b) grasp, (c) extract, and (d) insert.

- *Inspection*: Inspection is the homing operation of the AUV to the panel. It enables high-accuracy visually based navigation, providing a good estimate of the valve pose. The EE is guided to a Cartesian-space-defined pose located in front of the panel.
- *Grasp planning*: To perform the intervention, one must specify the desired valve pose. Then, according to the workspace analysis method explained in the "Workspace Analysis" section, the grasp pose is computed.
- *Valve turning*: Valve turning involves the following two steps: 1) approach to the panel and 2) grasp and turn. The approach step is aimed at getting close to the panel at a predefined distance, in the direction normal to the panel. This guarantees the pose of the EE before grasping. The second step consists of the actual intervention operation. The implemented method is based on the MoveIt! pick pipeline. The frame of reference for the approach is the valve, and the approach direction is normal to the subsea panel z axis. Because there is no gripper to open/close, the grasp closure pose is used to perform the turn action. The wrist joint is specified as the posture joint, and the turning degree is provided as the joint value.

Figure 7 shows the I-AUV during several phases of the experiment. In Figure 7(a), (c), (e), (g), (i), and (k), the experiment in the water tank is shown, and in (b), (d), (f), (h), (j), and (l), the 3-D ROS visualization details the robot, environment, and AUV and EE trajectories. A plot of the 3-D AUV and EE trajectories is presented in Figure 9(a). The green box represents the obstacle. The panel with the three valves is also shown, along with the detection, inspection, and grasping points, showing how the I-AUV went through all of these steps. Figure 9(b) displays the trajectory of the arm joints along the mission. The valve-turning action can be appreciated in the wrist joint changing 90° at the end of the trajectory. The Cartesian AUV trajectory is shown in Figure 9(c). The dotted bold gray line corresponds to the planned path, and the colored lines (red, green, blue, and orange) correspond to the actual X, Y, Z, ψ path, respectively. The last plot [Figure 9(d)], presents the EE position and orientation, along with the desired waypoints. The vertical black dotted lines show the boundaries of each phase. Here, it can be appreciated that the EE is reaching the required positions and orientations. In this scenario, the workspace analysis was integrated and performed online. When the inspection point was reached, a call to the workspace analysis service was performed, while the I-AUV held its current configuration. This period of time is indicated by dotted black lines in the plot of Figure 9(c).

Connector Plug/Unplug

A model inspired by a scaled-down version of a hot-stab connector, similar to one used in the oil and gas industry, was used. As is common in this type of connector, the model uses passive compliance to account for a possible small misalignment, thus helping to complete the insertion process. Nevertheless, an EE accuracy of better than 12.25 mm (the difference between the inside and outside radii of the funnel-shaped female connector) was required to complete the goal. To perform this task, the V-shaped EE was substituted by the three-finger hand that is described earlier in "The Three-Finger Hand" section. The task starts with the connector plugged in and then requesting the robot to unplug it, move away, and then replug it. The intervention procedure is similar to the procedure previously described but with the valve-turning step replaced by the following steps:

- *Unplug*: The hand is requested to grasp the connector by closing the fingers. Similar to the valve turning, the MoveIt! pick pipeline is used. First, an approach direction perpendicular to the plane of the connector is specified and the pregrasp pose is chosen to open the hand before grasping. The connector pose corresponds to the pick pose, and the grasp closure pose is used to close the hand after holding the connector. Once the grasping is successful, the unplugging is accomplished by performing a retreat backward. The retreat direction is specified along the $-z$ axis in the connector frame.
- *Plug*: The MoveIt! place pipeline is used to plug the connector to its original location. Once the connector reaches its original location, the hand is opened, leaving the connector plugged in.

This experiment allowed us to validate an advanced intervention operation using the three-finger hand. Figure 8 shows a sequence of the real I-AUV during the mission. The AUV and the EE trajectories, in blue and red, respectively, are reported in Figure 10(a), detailing the panel pose with the three valves in blue and the connector location in red. Figure 10(b) displays the trajectory of the arm joints during the mission, and Figure 10(c) shows the evolution of the 4-DoF (X, Y, Z, ψ) of the AUV. The last plot in Figure 10(e) presents the EE pose. Finally, Figure 10(d) presents the progress of the three-finger hand during the experiment. Each

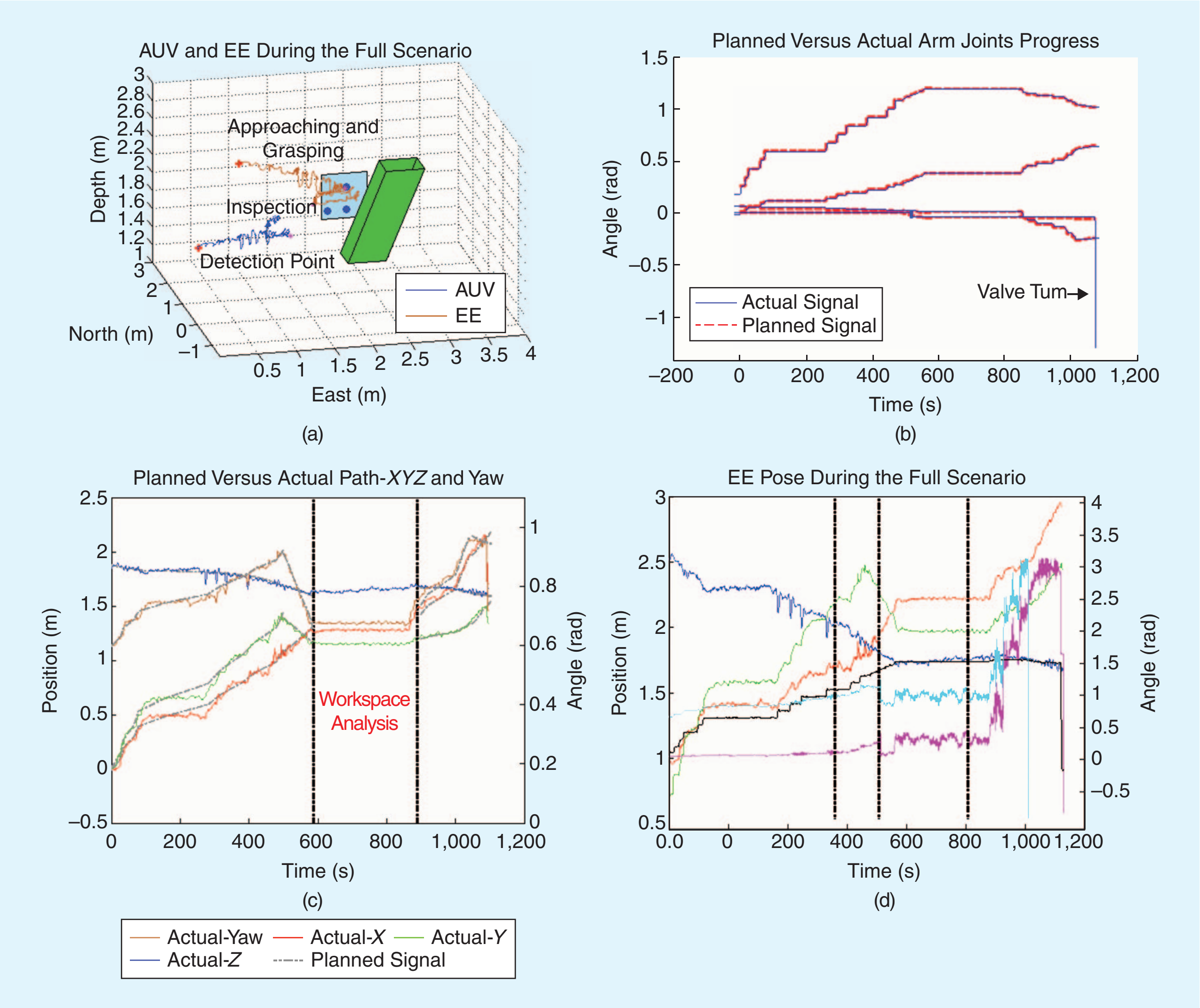


Figure 9. The valve-turning scenario results: (a) AUVs and the EE trajectory in the NED frame, (b) the 4-DoF arm joints progress planned versus actual, (c) the AUV (X, Y, Z, ψ) joints progress planned versus actual, and (d) the EE pose $(X, Y, Z, \phi, \theta, \psi)$ progress versus set points (X, Y, Z) in red, green, and blue and (ϕ, θ, ψ) in magenta, black, and cyan.

finger is indicated with a different color (red, green, and blue). In the figure, the fingers go through a sequence of open, close, open for the unplug/plug sequence, showing the success of the intervention operation.

Planning Time

Table 1 shows the planning time for various waypoints. The time for each has been computed as an average of five experiments, because RRT connect is a sampling-based technique and does not necessarily generate the same trajectory for the same (start–goal) configuration. We also provide the time spent to compute the solution of the complete pick pipeline. As expected, the planning of the complete pick pipeline is longer (pipeline details discussed easrlier in the article). In our case, it took 0.22 s for the valve-turning scenario and 0.3 s for the connector plug/unplug. The time invested in planning is very small compared to the time spent executing the manipulation, because achieving good accuracy required the EE to move at a very slow speed.

Discussion

On the basis of our experience using MoveIt!, we present in this section the advantages and drawbacks of this approach for a UVMS. The principal advantages include the following:

- *Easy definition of the UVMS kinematics*: The use of a URDF file and the integration with the KDL allowed for an easy definition of our model, avoiding the tedious definition/coding of the forward/inverse kinematic equations. Moreover, it was fairly simple to adapt the system to changes such as upgrading from the V-shaped gripper to a more challenging three-finger hand.
- *Easy planning of collision-free paths*: The ability to define the collision geometry of the model, in combination with the FCL library, and the group of path-planning algorithms available at the OMPL library makes it easy to compute collision-safe trajectories (e.g., the valve turning, as in Figure 7). This is an important feature for image, maintenance, and repair applications, where intervention happens in the proximity of subsea infrastructures.

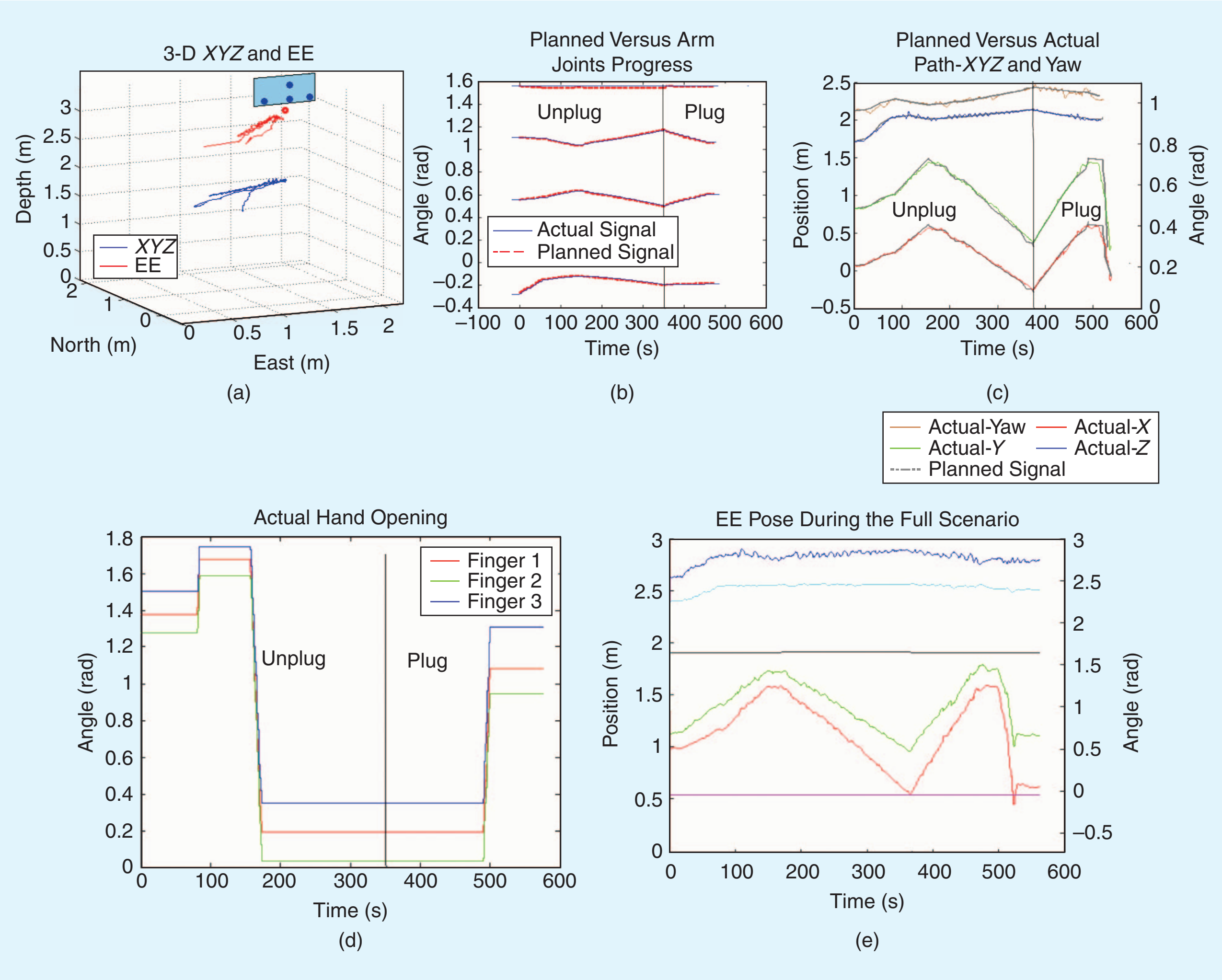


Figure 10. The plug/unplug connector results: (a) AUVs and the EE trajectory in the NED frame, (b) the 4-DoF arm joints progress planned versus actual, (c) the AUV (X, Y, Z, ψ) joints progress planned versus actual, (d) the three-finger hand opening, and (e) the EE pose $(X, Y, Z, \phi, \theta, \psi)$ progress (X, Y, Z) in red, green, and blue and (ϕ, θ, ψ) in magenta, black, and cyan.

- *Reduced development time*: Having the arm and AUV control software already available, it was possible to implement the coordination layer in charge of the joint AUV/arm control, including the experimental demonstration, in a relatively short time.

Nevertheless, during the development, we faced the following challenges that can be topics for future investigation:

- *Heterogeneous DOFs*: The DoFs in the manipulator are fast and accurate, but the AUV DoFs are slow and uncertain. AUV DoFs should be used for long displacements and low-frequency motion, whereas those of the arm should be used for short and high-frequency motion. In fact, they may compensate for the inaccurate AUV motion. It would be interesting to differentiate these DoFs in the model, promoting the AUV motion to approach the target when it is outside the arm workspace and using arm motion when the target is within its workspace.
- *Nonactuated DoFs*: Underactuated UVMSs may experience a degree of perturbation on their nonactuated DoFs, for instance, with the roll DoF of our AUV. It is not possible to control the roll, because it is passively stable, but it may experience a few degrees of variation due to a perturbation. At the cost of increasing the system complexity, it could be possible to use the hydrodynamical model of the vehicle-manipulator system to predict the set of $[\eta, Q]$ states achievable for the set of possible control inputs and use planning constraints to avoid visiting unachievable states. Because our arm is almost neutrally buoyant and does not induce significant roll disturbances, a compromise solution was adopted. The

Table 1. Planning time for various tasks.

Task	Planning Time (s)
Unplug	0.022
Plug	0.1
Valve turning/detection	0.0552
Valve turning/inspection	0.04
Valve turning/grasp	0.018

nonactuated DoFs were included in the URDF model but with a fixed value equivalent to their values at the beginning of each experiment. This solution suffers from a loss of accuracy of the EE position in case the value of the nonactuated DoF changes significantly during the manipulation. In this instance, a replanning action should be launched.

Conclusions

We focused on developing a system to perform advanced free-floating manipulation, merging path planning with trajectory-tracking control. We describe how to 1) model the UVMS and integrate MoveIt! with the available AUV and arm controllers, 2) deal with a priori-known obstacles to plan collision-free trajectories, and 3) plan a grasp using workspace analysis. The system has been experimentally evaluated in a water tank using the valve-turning and connector plug/unplug benchmarks on a free-floating base. The results demonstrate the MoveIt! potential to deal with advanced underwater manipulation tasks. Although the work reached its goals, in a real autonomous intervention system, we must take into account 1) keeping the panel within the camera field of view to ensure an accurate localization through visual servoing, 2) optimizing the MI, and 3) ensuring a minimum attitude with respect to the sea bottom to ensure the DVL performance (i.e., good bottom-tracking velocity). These objectives have been well studied within the task priority framework, but future investigation is needed to determine methods for introducing them within a trajectory-planning framework by means of convenient cost functions, along with finding solutions to the challenges encountered.

Acknowledgment

This work was supported by the Spanish Project DPI2014-57746-C3-3-R (MERBOTS-ARCHROV).

References

[1] P. Ridao, M. Carreras, D. Ribas, P. J. Sanz, and G. Oliver, "Intervention AUVs: The next challenge," in *Proc. Int. Federation of Automatic Control,* 2014, pp. 12146–12159.

[2] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-priority based redundancy control of robot manipulators," *Int. J. Robotics Res.*, vol. 6, no. 2, pp. 3–15, 1987.

[3] S. C. I. A. Sucan. (2013). MoveIt! [Online]. Available: http://moveit.ros.org.

[4] D. Ribas, N. Palomeras, P. Ridao, M. Carreras, and A. Mallios, "Girona 500 AUV: From survey to intervention," *IEEE/ASME Trans. Mechatronics*, vol. 17, no. 1, pp. 46–53, Feb. 2012.

[5] N. Palomeras, A. El-Fakdi, M. Carreras, and P. Ridao, "COLA2: A control architecture for AUVs," *IEEE J. Ocean. Eng.*, vol. 37, no. 4, pp. 695–716, Oct. 2012.

[6] N. Palomeras, A. Penalver, M. Massot-Campos, G. Vallicrosa, P. Negre, J. Fernandez, P. Ridao, P. Sanz, G. Oliver-Codina, and A. Palomer, "I-AUV docking and intervention in a subsea panel," in *Proc. 2014 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2014, pp. 2279–2285.

[7] G. Vallicrosa, P. Ridao, D. Ribas, and, A. Palomer, "Active range-only beacon localization for AUV homing," in *Proc. 2014 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2014, pp. 2286–2291.

[8] H. Kato and M. Billinghurst, "Marker tracking and HMD calibration for a video-based augmented reality conferencing system," in *Proc. 2nd IEEE and ACM Int. Workshop Augmented Reality*, 1999, pp. 85–94.

[9] I. A. Şucan, M. Moll, L. Kavraki, I. Sucan, M. Moll, and E. Kavraki, "The open motion planning library," *IEEE Robot. Autom. Mag.*, vol. 19, no. 4, pp. 72–82, 2012.

[10] J. Pan, S. Chitta, and D. Manocha, "FCL: A general purpose library for collision and proximity queries," *Proc. 2012 IEEE Int. Conf. Robotics and Automation*, pp. 3859–3866.

[11] D. Youakim, P. Ridao, and N. Palomeras, "Moveit! based implementation of an I-AUV," M.S. thesis, University of Girona, Spain, 2015.

[12] J. Kuffner and S. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2000, pp. 995–1001.

[13] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard. (2013, Apr.). OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Auton. Robots*. [Online]. vol. 34, no. 3, pp. 189–206. Available: http://octomap.github.com

[14] N. Palomeras, S. Nagappa, D. Ribas, N. Gracias, and M. Carreras, "Vision-based localization and mapping system for AUV intervention," in *Proc. 2013 MTS/IEEE OCEANS–Bergen*, pp. 1–7.

Dina Youakim, Computer Vision and Robotics Research Institute, University of Girona, Spain. E-mail: dina.youakim@gmail.com.

Pere Ridao, Computer Vision and Robotics Research Institute, University of Girona, Spain. E-mail: pere@eia.udg.edu.

Narcis Palomeras, Computer Vision and Robotics Research Institute, University of Girona, Spain. E-mail: npalomer@eia.udg.edu.

Francesco Spadafora, Department of Mechanical and Management Engineering, University of Calabria, Arcavacata, Italy. E-mail: francesco.spadafora@unical.it.

David Ribas, Computer Vision and Robotics Research Institute, University of Girona, Spain. E-mail: dribas@eia.udg.edu.

Maurizio Muzzupappa, Department of Mechanical and Management Engineering, University of Calabria, Arcavacata, Italy. E-mail: maurizio.muzzupappa@unical.it.