Received 30 October 2023, accepted 9 November 2023, date of publication 13 November 2023,
date of current version 21 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3332486

RESEARCH ARTICLE

Compliant Manipulation With Quasi-Rigid Docking for Underwater Structure Inspection

ROGER PI, PATRYK CIEŚLAK, JOAN ESTEBA, NARCÍS PALOMERAS, (Member, IEEE), AND PERE RIDAO, (Member, IEEE)
Computer Vision and Robotics Research Institute (VICOROB), Universitat de Girona, 17003 Girona, Spain

Corresponding author: Roger Pi (roger.pi@udg.edu)

This work was supported in part by Secretaria d'Universitats i Recerca del Departament d'Economia i Coneixement de la Generalitat de Catalunya under Grant 2019FI_B_00812, in part by the ATLANTIS Project under Grant H2020-ICT-2019-2-871571, in part by the COOPERAMOS Project under Grant PID2020-115332RB-C32, and in part by the OPTIHROV Project under Grant PDC2021-120791-C21.

ABSTRACT Offshore wind farms are a crucial source of renewable energy, but maintenance and repair can be challenging due to their remote locations and harsh environmental conditions. Professional divers or Remotely Operated Vehicles (ROVs) are commonly used to conduct maintenance operations, but they come with high daily operational costs. Autonomous Underwater Vehicles (AUVs) have the potential to improve the efficiency, safety, and costs of maintenance operations. This project evaluates the feasibility of using an AUV to conduct a cathodic protection (CP) survey, which involves measuring the corrosion potential of underwater structures to prevent deterioration. The AUV is equipped with a manipulator that has a CP probe with a sharp tip to puncture through the structure's coating and make contact with the steel underneath. To ensure high accuracy and reduce environmental perturbances, the AUV attaches to the structure while conducting the survey. The technology and methods used in this project are demonstrated in a water tank using a Girona1000 AUV. Task Priority kinematic control is combined with a custom force control strategy based on admittance control to enable tracking of the end-effector configuration and contact force during the probing operation. The mission flow control is implemented using behavior trees. The results show that the use of AUVs for CP surveys is feasible and has the potential to significantly improve the efficiency, safety, and costs of maintenance operations in offshore wind farms.

INDEX TERMS Autonomous underwater intervention, task priority control, force control, behavior trees.

ACRONYMS

AUV	Autonomous Underwater Vehicle.
BT	Behavior Tree.
CP	cathodic protection.
DVL	Doppler Velocity Log.
EE	End Effector.
FSM	Finite State Machine.
FT	force-torque.
I-AUV	Intervention Autonomous Underwater Vehicle.
INS	Inertial Navigation System.
NDT	Non Destructive Testing.
ROV	Remotely Operated Vehicle.
SVS	Sound Velocity Sensor.
TP	Task Priority.
USBL	Ultra Short Baseline.
UVMS	Underwater Vehicle Manipulator System.

I. INTRODUCTION

The inspection and maintenance of underwater structures, such as offshore wind farms and oil platforms, is a challenging task due to their remote locations and harsh environmental conditions. These tasks are often performed by professional

The associate editor coordinating the review of this manuscript and approving it for publication was Guilin Yang.

FIGURE 1. Photography of a wind farm at Viana do Castelo, Portugal, in the context of the project ATLANTIS (credit: EDP Energias de Portugal).

divers, or ROVs deployed from dynamic-positioning vessels, imposing high daily operational costs. Intervention Autonomous Underwater Vehicles (I-AUVs) have the potential to improve the efficiency and safety of underwater structure inspection by providing a more cost-effective and reliable alternative [1].

In this paper, we focus on the problem of performing compliant manipulation with quasi-rigid docking for underwater structure inspection using an I-AUV. Specifically, we address the challenge of conducting CP inspection on a wind-float structure (see Fig. 1). Wind-float structures are offshore platforms used to generate renewable energy from wind power, and they can vary in size and shape depending on the manufacturer and location.

To perform CP inspection, we use a probe capable of puncturing through the coating of the structure and making contact with the steel underneath to measure the corrosion potential of the underwater steel structure. However, performing precise manipulation of the probe while minimizing the risk of damaging the structure is a difficult task for an I-AUV in a dynamic environment. Attaching the I-AUV to the structure in a stable and controlled manner can guarantee accurate puncturing and minimize the risk of damaging the structure. To address these challenges, we propose a compliant manipulation strategy that combines quasi-rigid docking with force control, allowing the I-AUV to attach to the wind float and perform precise CP inspection without causing damage. We also present the technology and methods used to perform the whole mission, demonstrated in a mockup scenario simulating a wind-float structure.

II. RELATED WORK

Despite the growing interest and advances in underwater robotics [2], there is still a technological gap for applications that require interaction with the environment. This gap becomes particularly evident in the literature on underwater manipulation, where the majority of the work is predominantly confined to simulation environments [1]. Thus, there is a lack of extensive experimental results using real (autonomous) underwater robots for complex manipulation tasks.

Projects like SAUVIM [3], TRIDENT [4], and MARIS [5] have explored autonomous free-floating manipulation. They have developed control architectures and employed vision systems, such as stereo cameras and laser scanning, to detect and grasp objects on the seafloor. These projects have demonstrated successful grasping of mock-up objects while considering control objectives like maintaining the object in the field of view, avoiding occlusions caused by the manipulator, and respecting joint limits.

The PANDORA project [6] focused on free-floating valve-turning operations on a panel. It employed a task-priority approach and developed control frameworks to perform these operations. The PANDORA project also included tests using learning by demonstration [7] and later with motion planning capabilities using the ROS framework “MoveIt!” to generate reference trajectories for the Underwater Vehicle Manipulator System (UVMS) [8].

In [9], the authors used an in-house developed laser scanner [10] to build an occupancy grid for motion planning. That paper reports experimental results obtained on autonomous valve-turning operations in the presence of *a priori* unknown obstacles.

The EU DexROV project [11] focused on the remote control, via satellite communications, of a (semi-autonomous) UVMS umbilically attached to a support vessel from a distant onshore control center. The primary concept is that the operator interacts with a real-time simulation environment, and a cognitive engine analyzes the control requests to turn them into motion primitives that the UVMS executes autonomously in the real environment, thus not being affected by communication latencies. They presented experimental results on a mock-up of a real oil and gas panel deployed at 30 meters depth in the Mediterranean Sea.

The Droplet Project [12] explored underwater assembly of mortarless structures using a BlueRov vehicle equipped with a robotic hand. They designed pickup/drop platforms and low-weight 3D printed blocks for passive accommodation during the dropping of the block. The experiment was extremely simplified, but set the foundation for their next work [13], using cement blocks and custom-made interlocking cone inserts.

The TWINBOT project [14] demonstrated grasping and high-accuracy cooperative transportation of a bulky pipe, using two Girona500 I-AUV, each one equipped with a 4 DOF manipulator. The authors proposed a decentralized Task Priority (TP) kinematic control architecture, using a master-slave organization, suitable for the limited available wireless communication bandwidth. Experimental results demonstrated the pick, transport, and place operations in a water tank.

Another work in underwater cooperative transportation is presented in [15] using a nonlinear model predictive control approach, capable of sharing the load among the vehicles according to their payload capabilities. The work was demonstrated experimentally in a water tank using two small, custom-made UVMSs.

Few studies have performed experimental trials to explore how I-AUVs can manage the force exchanged with the environment for force regulation activities. A sliding mode impedance control is proposed in [16] and validated in a water tank on a flat panel. Although they provide simulated results for a mobile base and a 3 DOF manipulator, only a 6 DOF vehicle (the SPIR3.0 AUV) and a fixed force-torque (FT) sensor were considered in experiments. In [17], experimental results showcased an interaction controller that allowed the UVMS to glide its end effector across a flat panel's surface on a pool setup. The force mesurements, perpendicular to the panel, were not acquired by a FTs sensor in the manipulator, but by four load cells installed in the panel itself.

In [18], experimental trials were executed in a water tank simulating the inspection of a pipe using the Girona500 I-AUV [19] equipped with a 4 DOF manipulator and a FTs sensor. Similar to the work presented in this article, the approach was based on the combination of Task Priority control and an adaptive admittance controller. The robot was commanded to continuously touch the pipe along a predefined trajectory generated using cylindrical coordinates, considering a smaller pipe radius to ensure contact.

Another interesting work in force control is presented in [20]. This work explored how to maximize contact wrenches in a desired direction for I-AUVs by proposing novel methods for determining optimal configurations and actuator forces/torques. It addressed static wrench maximization, trajectory tracking with wrench optimization, and generating large wrench impulses using dynamic motions. The work was experimentally validated using a BlueRov vehicle and a 4 DOF Reach Alpha manipulator.

A. CONTRIBUTION

In contrast to our previous work [18], this paper presents significant advances by offering a comprehensive execution of a full Non Destructive Testing (NDT) mission utilizing force control and presenting results with a real I-AUV in a water tank environment, achieving a Technology Readiness Level (TRL) of 4. This TRL designation underscores the practical applicability and readiness of our methodology for controlled environments, marking a significant milestone in the development of autonomous underwater intervention.

This new approach couples the robot with the submerged structure using a second manipulator equipped with a magnetic End Effector (EE). The attachment provides stability, reducing unintended movements and improving the overall control of the inspection tools. This combination enables precise and controlled manipulation, resulting in more reliable inspection data and accurate assessment of the structure's condition. This stability minimizes the potential for damage to both the structure and the inspection equipment, enhancing the safety of the inspection operation. The magnetic attachment is not strong enough to neglect the force reaction during the contact force operation, which could potentially detach the magnet, especially due to a long lever arm between the probing point and the attachment location. Therefore, the vehicle's thrusters are used to compensate for the reaction forces at the attachment point while the probe contacts the structure, in order to hold the robot in place.

III. MECHANICAL DESIGN

A. GIRONA1000 I-AUV

The Girona1000 is an updated version of the Girona500 AUV [19]. The most significant changes are a maximum operational depth increased to 1000MSW and the integration of an Inertial Navigation System (INS). As the Girona500, it is also equipped with a Doppler Velocity Log (DVL), a pressure gauge, a Sound Velocity Sensor (SVS), and an Ultra Short Baseline (USBL) and a GPS for absolute position measurements underwater or on the surface. In this project, it has been equipped with two manipulators:

- An **ECA 5E Micro**: a 4 DOF manipulator actuated by electrical screw drives, which limits the joint's range and makes it slow, but strong. Its strength makes it suitable for the docking task, where it has to withstand forces generated by the AUV's inertia and the manipulation task. Its end-effector is equipped with a magnetic tool with the aim to attach to ferromagnetic structures. It consists of three neodymium magnets protected with a layer of epoxy, coupled to the ECA piston, which can slide the magnets in and out of the tool's housing (see Fig. 2).
- A **Reach Bravo7 Mk2**: A 6 DOF dexterous manipulator faster and more accurate, allowing precise manipulation. Its end-effector is equipped with a 6-axis force-torque FT sensor and a CP probe to perform measurements while keeping desired contact force.

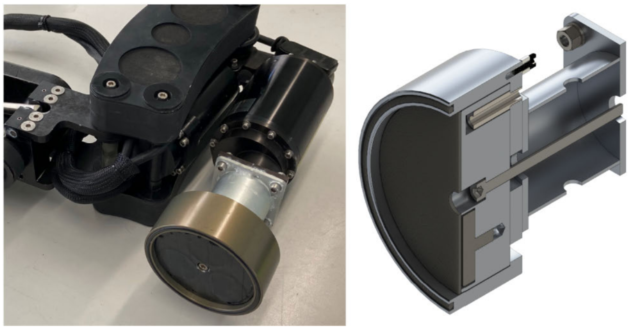


FIGURE 2. Magnetic end-effector design. On the left, the end-effector mounted on the ECA manipulator. On the right, a cross section of the CAD design.

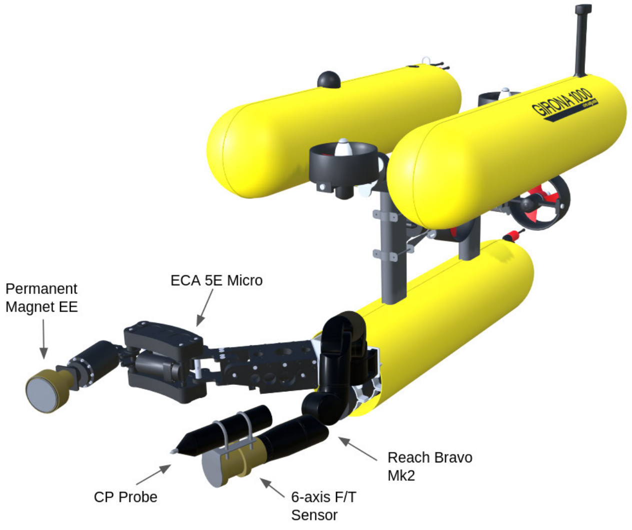


FIGURE 3. **Girona1000 equipped with the intervention payload.**

Due to the highly restricted range of motions possible to achieve with the ECA manipulator, its mounting configuration has been prepared to maximize the workspace when the ECA manipulator is attached to a vertical surface, like the floating wind turbine cyliders. However, this design decision is not optimal for folding the manipulator when not in use.

Figure 3 shows the Girona1000 AUV equipped with two manipulators for the proposed inspection task.

B. WINDFLOAT PILOT STRUCTURE

To test our proposed strategy, we constructed a pilot structure, a panel with a surface curvature similar to the one of the submerged cylinders of the WindFloat Atlantic, installed in Viana do Castello (recall Fig. 1), fitting inside the CIRS test tank (see Fig. 4). It was made of carbon steel for its ferromagnetic properties and protected from corrosion with a thin layer of epoxy paint. Additionally, it includes a true to life ladder and can be augmented with a set of magnetic ArUco markers [21], which are used to simplify the localization of the structure at early stages of development. Note that while the use of magnetic ArUco markers simplified the initial localization of the structure in our experiments, it is recognized that for broader applications and more intricate real-world scenarios, the development of enhanced structure recognition capabilities becomes increasingly important.

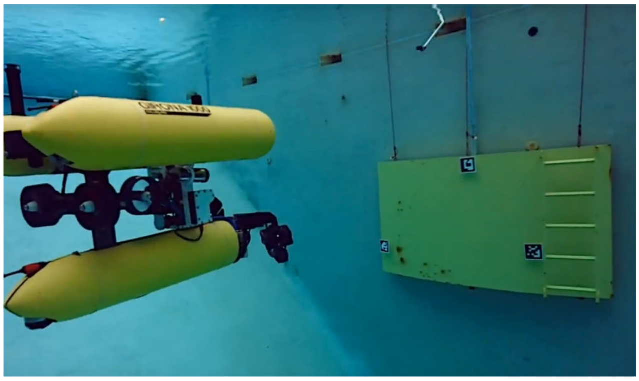


FIGURE 4. **Pilot structure emulating a section of a wind farm, at the CIRS test tank.**

IV. CONTROL ARQUITECTURE

The control architecture of the robot is schematically presented in Fig. 5. It is implemented using the Robot Operating System (ROS1) middleware, running on Ubuntu Linux.

The high-level mission is programmed using behavior trees [22] and interacts with the whole system sending commands to the different agents. The control layer is based on Task Priority [23], which can satisfy the goals of multiple control tasks by exploiting the system's redundancy. In TP, tasks are assigned priorities and controlled hierarchically to ensure that higher-priority tasks are achieved first, and lower-priority tasks are achieved only if they do not interfere with the higher-priority tasks (see Section V). The mission controller can reorder and switch on and off tasks, send them setpoints, and monitor their execution. One important task is the contact force task, which, based on readings from the force-torque sensor, controls the contact force of the probe w.r.t. the structure during the probing operation. The output of the control layer is sent to the manipulators through velocity

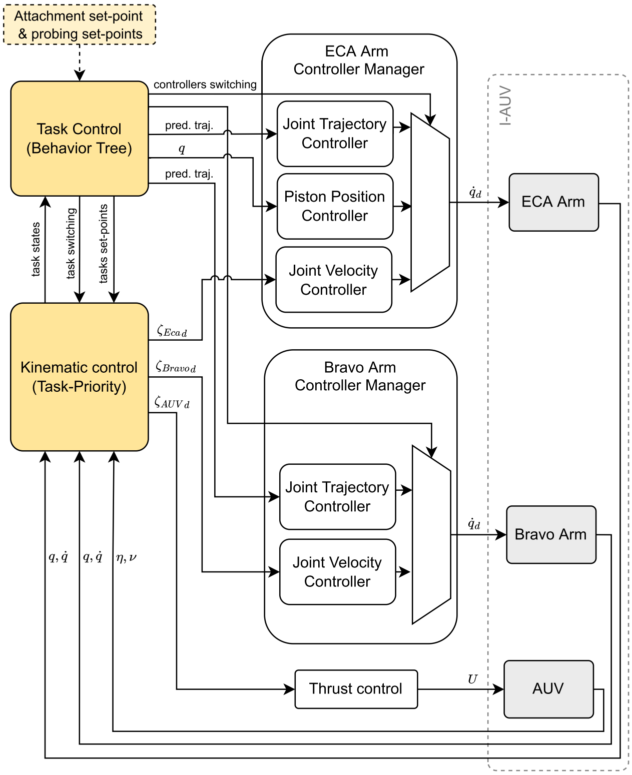

Attachment set-point & probing set-points
Task Control (Behavior Tree)
controllers switching
pred. traj.
q
pred. traj.
ECA Arm Controller Manager
Joint Trajectory Controller
Piston Position Controller
Joint Velocity Controller
I-AUV
$\dot{q}_d$
ECA Arm
task states
task switching
tasks set-points
Kinematic control (Task-Priority)
$\zeta_{Eca\,d}$
$\zeta_{Bravo\,d}$
$\zeta_{AUV\,d}$
Bravo Arm Controller Manager
Joint Trajectory Controller
Joint Velocity Controller
$\dot{q}_d$
Bravo Arm
$q, \dot{q}$
$q, \dot{q}$
η, ν
Thrust control
U
AUV

FIGURE 5. **Scheme for the control system of the Girona1000 I-AUV. The symbols used in the drawing are explained in the text. Work presented in this paper concerns the yellow coloured blocks.**

controllers, while the vehicle is controlled either in velocity or thrust.

The control interface for both manipulators is based on the ros_control package [24], which provides a standard interface and implementation for low-level controllers. A controller manager manages hardware resources and allows switching controllers on and off at runtime while handling resource conflicts between controllers. Both manipulators have a joint group velocity controller and a joint group trajectory controller. The velocity controllers control each arm in joint velocities and are mainly used by the task priority kinematic controller (see Section V). The trajectory controllers allow executing joint-space trajectories, defined by a set of position and velocity setpoints, to be reached at specific time instants. A cubic spline interpolator generates smooth and continuous motions with continuity guarantees at the velocity level. The high-level mission controller uses joint trajectory controllers to follow safe, offline generated trajectories for folding and unfolding of the manipulators. Additionally, a single joint position controller controls the piston of the ECA arm, to expose or hide the magnetic end-effector.

V. TASK PRIORITY KINEMATIC CONTROL

In this section, we aim to highlight the essential elements of TP that are pertinent to our research, while directing readers to our previous work [6], [14] for a more detailed description and formulation.

TP is based on a hierarchical control framework, where each task is assigned a priority level, and the robot's motion is controlled hierarchically to ensure that higher priority tasks are achieved first, while lower priority tasks are only achieved if they do not interfere with the higher priority ones.

Tasks are designed to achieve goals, such as reaching an EE pose, keeping a joint position fixed or avoiding joint limits. The mathematical foundation of TP is based on the concept of null space projection. Given a set of tasks with priorities $\boldsymbol{\sigma_1}, \boldsymbol{\sigma_2}, \ldots, \boldsymbol{\sigma_n}$, where $\boldsymbol{\sigma_1}$ has the highest priority and $\boldsymbol{\sigma_n}$ has the lowest priority, the null space projection is used to compute the desired system velocities $\boldsymbol{\zeta} = [\boldsymbol{\nu^T} \; \boldsymbol{\dot{q}^T}]^T$ that satisfy the higher priority tasks while also leaving the degrees of freedom associated with the lower priority tasks free to move. Two categories of tasks can be identified: 1) Equality tasks, whose goal is to drive the task variable to a desired value (e.g., EE pose task), and 2) Set tasks (also called inequality tasks) [25], [26], whose goal is to keep the task variable within a specified range (e.g., manipulator joint limits task).

Equalities encompass regulation tasks and tracking tasks. Regulation tasks aim to reach specific system states or conditions, while tracking tasks involve following desired trajectories with time constraints.

Set tasks are scalar regulation-only tasks used to keep a task variable σ within a specific range. These tasks activate only when the σ overpasses its limits to push σ back to the admissible set. Then, once the task variable σ has been successfully guided back within the predefined range, the set task deactivates.

Typical set tasks reported in the literature include Joint Limits, EE obstacle avoidance, and Minimum Altitude / Depth, to name but a few. The most representative task of this type is probably the one devoted to ensuring that the joint variables remain within their boundaries.

TABLE 1. Task priority hierarchy used in the presented work.

Hierarchy	Description	Type
Task 1	Joint Limits	Set Task
Task 2	AUV Position	Regulation Task (Position)
Task 3	AUV heading	Regulation Task (Orientation)
Task 4	AUV velocity	Tracking Task (Pose3D)
Task 5	ECA EE Pose	Regulation Task (Pose 3D)
Task 6	Bravo EE Pose	Regulation Task (Pose 3D)
Task 7	Bravo EE Compliant Pose	Regulation Task (Pose 3D)

Table 1 depicts the tasks used in the presented work and their associated priority. The topmost task is devoted to safety, maintaining joint positions within their defined limits. Tasks 2 & 3 control the AUV position and heading and are generally used for navigation. Position and heading are separated since there might be situations where it is desired to keep the heading (e.g., ensuring visibility of the target) while performing tasks other than navigation. Task 4 controls the vehicle's velocity, and is only used shortly after the robot attempts to attach to the structure, pushing the robot backward to test if the attachment succeeded. This task is configured to have the regulation component disabled (i.e., zero gain); therefore, it only has the feedforward component.

Tasks 5, 6, and 7 control the EE 3D poses of the ECA and Bravo manipulators. Task 5 is activated to attach to the structure, while tasks 6 and 7 coordinate to perform the probing operations. In particular, task 6 is devoted to moving the Bravo EE to a pre and post-operation pose, while task 7, whose input is regulated by an adaptive admittance controller, performs the touch operation.

VI. CONTACT FORCE CONTROL

A. ADAPTIVE ADMITTANCE CONTROL

The admittance control scheme [27] used to control the pose of the Bravo EE and the contact force exerted during the touch operation is based on the concept of a compliant frame $\bar{\eta}_{ee}$ attached to the stiff desired EE frame $\eta_{ee,d}$ by a virtual linear mass-spring-damper system, with dynamics described by:

$$\begin{aligned} M\ddot{\tilde{\eta}}_{ee} + K_D\dot{\tilde{\eta}}_{ee} + K_S\tilde{\eta}_{ee} &= h_{ee} \\ M\bar{\eta}_{ee} &= \eta_{ee,d} + \tilde{\eta}_{ee} \end{aligned} \tag{1}$$

where M, K_D, K_S are impedance parameters (constant, diagonal matrices), $\tilde{\eta}_{ee}$ is the pose error between the stiff and the compliant EE frame, and $h_{ee} = [f_{ee}^T \; \tau_{ee}^T]$ is the EE contact wrench, measured by the FT sensor installed in the manipulator's wrist.

Thus, the desired pose deformed by the admittance control layer is input to Task 7 during the touch operation.

Since the original algorithm lacks direct control over the contact wrench value, the stiffness constant K_S is replaced with a variable stiffnes, given by:

$$K_S(t) = K_P \int (h_{ee,d}(t) - h_{ee}(t))\, dt \tag{2}$$

where $h_{ee,d}$ is the desired EE wrench and K_P is a diagonal gain matrix. All impedance parameters, together with the initial K_S and K_P are tuned experimentally.

B. FEEDFORWARD FORCE EXTENSION

It is worth noting, that when the robot is exerting a force on the inspected surface, using the probing manipulator, a reaction force and torque are generated. These reactions should be compensated by the attraction force of the permanent magnet end-effector, used for the attachment. However, the magnet was not strong enough to hold the robot in place, mainly due to the long lever arm between the probing and the attachment locations, which results in a significant torque. Therefore the robot may detach in an uncontrolled way during probing. To counteract this problem we introduced an extension to the control system to compensate the reaction wrench using the thrusters of the I-AUV.

We can model the described situation by assuming that the whole robot can be treated as a single body, rigidly attached at one point to the inspected surface, and a force perpendicular to the inspected surface is exerted at the tip of the probe. Assuming that our goal is to reach a certain contact force, at the tip of the probe, we can consider this an equilibrium state and solve for a static balance of forces and torques in the model, presented in Fig. 6. This balance can be formulated according to the following system of equations:

$$\begin{aligned} F_d - F &= 0 \\ \tau_d - \tau + \tau' &= 0, \end{aligned} \tag{3}$$

where F is the force measured by the FT sensor (reaction to the probing force) and τ is the torque that this force generates with respect to the attachment point. To counteract the reactions we have to generate force F_d and torque τ_d, in the body frame, using the robot's thrusters. It is important to notice that the force generated by the thrusters additionally produces a torque with respect to the attachment point, designated by τ'. Then, the desired wrench can be calculated as follows:

$$\begin{aligned} F_d &= F \\ \tau_d - F d_1 + F_d d_2 &= 0 \\ \tau_d &= F(d_1 - d_2), \end{aligned} \tag{4}$$

where d_1 and d_2 are distances defined in Fig. 6. Due to the fact that the wrench control using the thrusters is not precise, we introduce a factor that defines the percentage of the required wrench that is compensated in this manner. We assume that the rest of the reaction forces and torques are compensated by the permanent magnet gripper and we tune the system so that the detachment does not occur.

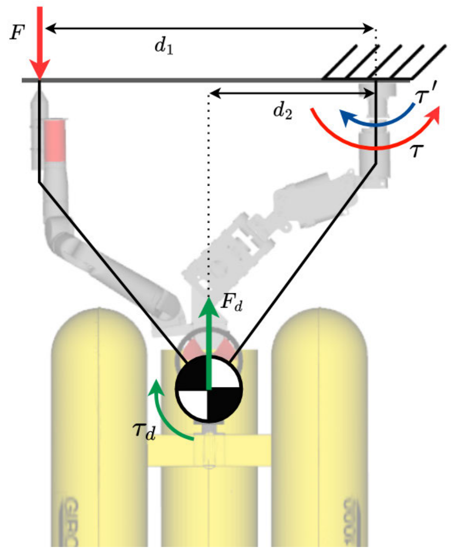


FIGURE 6. Force/Torque compensation during inspection.

VII. BEHAVIOR TREES

Behavior Trees (BT) allow for efficient switching between actions and modes of operation by providing a flexible and modular framework for modeling complex decision-making processes [28].

A BT is a directed rooted tree in which nodes are classified as leaves, composites, and decorators as described below. The edges of the tree represent the flow of control between the nodes. The terminology of parent and child is generally used to refer to connected nodes. A BT is executed starting from the root, which sends ticks at a certain frequency to its child. When a node receives a tick, it performs a specific action and returns a status to its parent node, which may be *Success*, *Failure*, or *Running*. Hereafter it is described how the most common types of nodes handle the tick and process the return statuses.

Composite nodes have one or more children and are used for organizing and coordinating the child nodes. The most common composite nodes are the Sequence and the Fallback. A Sequence node executes all its children in order as long as they return *Success*. It only returns *Success* if all its child nodes return *Success*. If one child returns *Failure* or *Running*, the node immediately stops and returns the same status to its parent. A Fallback node executes all its children in order as long as they return *Failure*. It returns *Failure* to its parent if all its child nodes return *Failure*. If one child returns *Success* or *Running*, the node immediately stops and returns the same status to its parent.

Decorator nodes can only have one child, and they modify or add functionality to it. For example, decorator nodes can be used to invert the status of their child node, set a time limit for execution, or perform other modifications to its behavior. Some examples include the inverter, repeater, and conditional nodes.

The nodes that perform the actual actions or evaluate conditions are called leaves. Leaves are often differentiated by actions and conditions. The main difference between them is that Condition nodes can only return *Success* or *Failure* within a single tick, whereas Action nodes can span multiple ticks, returning a *Running* status until they reach a terminal state. Generally, condition nodes represent simple checks (e.g., "is the door open?"), while action nodes represent complex actions (e.g., "open the door").

In addition to these types of nodes, BTs can also make use of a data structure called the Blackboard. The Blackboard is a shared memory space that allows nodes in the tree to access and modify information relevant to the overall behavior of the agent. It is similar to a global variable that can be read and written by different nodes. Nodes can read from and write to the Blackboard as needed, allowing them to make decisions based on the state of the world or coordinate their actions with other nodes in the tree.

In this paper, when representing graphically BTs, composite nodes are colored in red, decorator nodes are colored in yellow, action nodes are colored in blue, and sub-trees are colored in gray. Each node in the tree has clearly defined inputs and outputs, including any Blackboard variables that are used. When a Blackboard variable is used as input or output, it is expressed using the notation {*variable_name*}.

A. COMPOSITE NODES WITH MEMORY

In a standard Sequence or Fallback node, if a child node returns *Running* status and the tick ends, the next tick will start the evaluation of the child nodes from the beginning. Although this property stands for the principle of reactiveness, it may not be desirable in some situations and can result in unnecessary re-evaluation of nodes, slowing down the execution of the tree. For this reason, modern implementations include versions of composite nodes with memory. These nodes keep track of which child nodes were already evaluated during the previous tick and continue the evaluation from the last executed node, preventing unnecessary ticking in circumstances where reactiveness is not desired. Algorithms 1 and 2 show the pseudocode of the standard sequence and the sequence with memory respectively. An internal variable is maintained in the sequence with memory to point at the child that previously reported *Running*. The memory is reset if all children succeed or if a single child fails. In this work, we will refer to composite nodes without memory as Reactive, following the convention adopted in "BehaviorTree.CPP" [22], which is the library chosen by the authors to implement BTs. For instance, a Sequence with memory will be referred to as a **Sequence**, and a Sequence without memory will be referred to as **ReactiveSequence**.

Algorithm 1 Sequence (W/O Memory)

```
// child[] is an array of children nodes
for i = 1 to N (number of children) do
    childStatus = child[i].Tick()
    if childStatus == RUNNING then
        return RUNNING
    else if childStatus == FAILURE then
        return FAILURE
// All children succeeded
return SUCCESS
```

Algorithm 2 Sequence (W/ Memory)

```
// child[] is an array of children nodes
// idx is a private variable
while idx < N (number of children) do
    childStatus = child[idx].Tick()
    if childStatus == SUCCESS then
        idx ← idx + 1                    // increment idx
    else if childStatus == RUNNING then
        return  RUNNING                  // keep same idx
    else if childStatus == FAILURE then
        idx ← 0                          // reset idx
        return FAILURE
// All children succeeded
idx ← 0
return SUCCESS
```

It should be noted that the decision to use composite nodes with memory depends on the specific requirements and desired behavior of the application. While reactiveness is an important principle, it may not always be necessary or appropriate. Ultimately, it is up to the programmer to decide which parts of the BT should be reactive and which should have memory based on the specific needs and constraints of the system.

B. ADVANTAGES OF USING BEHAVIOR TREES

BTs have become a popular alternative to Finite State Machines (FSMs) for designing intelligent agent behaviors. One key advantage is their modularity, which allows for creating complex behaviors by combining simpler ones. BTs provide a more flexible way of handling complex behaviors with conditions, loops, and priorities. Additionally, BTs are more expressive than FSMs, as they can represent complex behaviors with multiple states and actions in a more compact and organized way. BTs also make it easy to modify or add new behaviors without affecting the rest of the tree. Overall, BTs provide a powerful and modular approach to designing complex behaviors for intelligent agents, making them a popular alternative to FSMs.

VIII. MISSION BEHAVIORS

A. ACTIONS AND CONDITIONS

1) GOTO POSE ACTION

When the **GoTo Pose** action is executed, it receives the desired position and orientation (through the Blackboard), as well as the name of the task in the TP hierarchy. The action then sends the resulting setpoint to the corresponding task through the ROS framework, which updates the control commands sent to the robot. The action also takes in a tolerance and timeout parameters. The tolerance specifies the maximum acceptable error between the current pose and the desired pose. If the current pose of the task target link is within the desired tolerance, the action returns a *success* status. If the controller is unable to achieve the desired pose within the given tolerance before the timeout expires, the action returns a *failure* status.

Additionally, a goal offset parameter might be provided to the **GoTo Pose** action, which specifies an offset to the desired pose. The goal offset is added to the desired pose before sending the setpoint to the task. This is used to perform an approach, touch, and retreat sequence given a single setpoint.

2) GOTO COMPLIANT POSE ACTION

The **GoTo Compliant Pose** action is similar to the **GoTo Pose** action, but it additionally takes the desired force and its tolerance as input parameters through the Blackboard. The action then sends the pose and force setpoints to the task, and monitors that the current pose and force applied are within the specified tolerances. If the task is able to achieve the desired pose and force within the specified tolerances before the timeout expires, the action returns a *success* status.

3) SWITCH TASKS ACTION

The **Switch Tasks** action is responsible for enabling and disabling tasks in the TP hierarchy (see Table 1). The action takes as input a set of tasks to be enabled and a set of tasks to be disabled and updates the corresponding tasks accordingly. Note that, once a task is achieved it remains activepopular alternativ (e.g., keeping an EE pose), so it is important to coordinate the switching of tasks during the mission. However, the use of the **Switch Tasks** action is not always explicitly shown in the figures to avoid cluttering the diagrams, since generally there is only one task active at the time (besides joint limits).

4) SWITCH CONTROLLERS ACTION

The control drivers of the manipulators are based on the ROS control framework. The **Switch Controllers** action is responsible for switching between different ROS controllers that control the manipulator's joints. The action takes as inputs the controllers to turn on and off and uses the "switch_controllers" service provided by the controller manager to switch the controllers.

This action is generally used to switch between the Joint Velocity Controllers and the Joint Trajectory Controllers. The former is used by the TP Kinematic Control layer to drive the manipulators, while the latter is used to follow pre-defined trajectories for folding and unfolding the manipulators. However, the use of the **Switch Controllers** action is not always explicitly shown in the figures to avoid cluttering the diagrams.

5) EXTEND/RETRACT MAGNET ACTIONS

The **Extend/Retract Magnet** actions are responsible for extending or retracting the piston of the ECA manipulator to slide the magnet in or out of the EE frame. Each action sends a predefined position setpoint to the piston position controller using the ros_control framework.

6) FOLLOW MANIPULATOR TRAJECTORY ACTION

The **Follow Manipulator Trajectory** action is responsible for sending a predefined trajectory to the Joint Group trajectory controller of the requested arm using the ros_control framework. This action is used to execute complex motion sequences that are planned in advance (e.g., fold / unfold operations). The action takes the manipulator name and the desired trajectory as inputs and sends it to the corresponding Joint Group trajectory controller, which then generates the corresponding control commands for the robot's joints. Once the trajectory has been executed, the action returns a *success* status. However, if the trajectory cannot be executed due to any reason (e.g., joint limits, controller not loaded), the action returns a *failure* status.

7) MAGNET CONTACT DETECTED CONDITION

The **Magnet Contact Detected** Condition is responsible for detecting whether the robot has made contact with the structure using the end-effector with a permanent magnet. Since there is no force-torque sensor on this manipulator, the current velocity of the robot is monitored to detect a sudden drop as an indicator of a collision with the structure.

8) LOCALIZE STRUCTURE ACTION

The **Localize Structure** action is responsible for detecting the position of the structure. The detection of the ArUco markers are used to fit a cylinder, which is assumed to be vertical (i.e., the cylinder's axis is aligned with the direction of gravity) and its radius known a priori. The problem can be formulated using non-linear least squares as follows:

$$\begin{aligned} &\underset{\boldsymbol{c}}{\operatorname{argmin}} \sum_{i=1}^{n} \|f_i(\boldsymbol{c})\|^2 \\ &f_i(\boldsymbol{c}) = r - \left\| \begin{bmatrix} c_x - a_{i,x} \\ c_y - a_{i,y} \end{bmatrix} \right\| \end{aligned} \tag{5}$$

where r is the known radius of the cylinder, $\boldsymbol{c}$ is the cylinder's center in the NED frame, and $\boldsymbol{a}_i$ is the location of the i-th ArUco marker in the NED frame. Note that since the z coordinate of the cylinder cannot be optimized, the $(\boldsymbol{c} - \boldsymbol{a}_i)$ error is projected onto the XY plane. Then, the depth of the structure is assumed to be referenced to the topmost ArUco marker.

B. TREES

The top-level behavior tree controls the overall mission and coordinates the execution of the subtrees. The mission consists of 6 phases: First, the robot locates the structure (1) and defines the probing and attachment points. Then it navigates close to the structure (2). Next, the robot must dock to the structure (3), probe the set of probing points (4), detach (5), and finally, surface (6). One could imagine the mission defined using only a sequence node, as shown in Fig. 7.

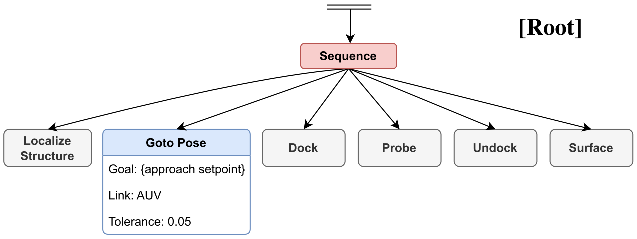


FIGURE 7. Behavior Tree of the general mission.

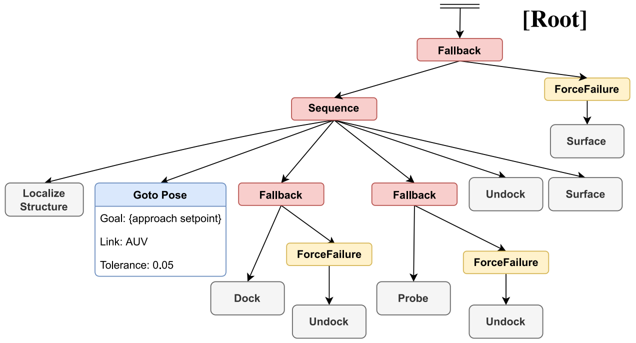


FIGURE 8. Behavior Tree of the general mission with recovery structures.

However, it is essential to contemplate recovery behaviors in case the robot fails to execute some task. Figure 8 depicts the top-level mission with recovery behaviors. Note that recovery actions are decorated with a **ForceFailure** that returns *Failure* regardless of the return status of the child.

1) DOCKING/UNDOCKING BEHAVIOR TREES

The docking procedure (see Fig. 9) aims to anchor the robot to the target structure using the ECA's magnetic tip, providing stability to the system before performing any intervention task. The task starts by unfolding the ECA arm and extending the magnet out. Then, the ECA EE is commanded to reach the docking setpoint, using a **Goto** action. While the robot moves toward the goal, it reactively checks if contact with the structure is detected. The setpoint *{attach_setpoint}*, is a global variable set inside the **Localize Structure** sub-tree. In order to ensure contact, the setpoint has a slight offset towards the center of the structure. Note that this prevents the **Goto** action from succeeding, but the **Reactive Fallback** will succeed as long as contact is detected. Also note that if contact is never detected, a **Timeout** decorator will stop the execution and report *FAILURE*. After detecting contact, the robot tests the attachment. Finally, it disables the AUV surge, sway, and yaw velocity controllers. Note that the surge is still controlled to avoid stress on the end-effector caused by the robot's buoyancy.

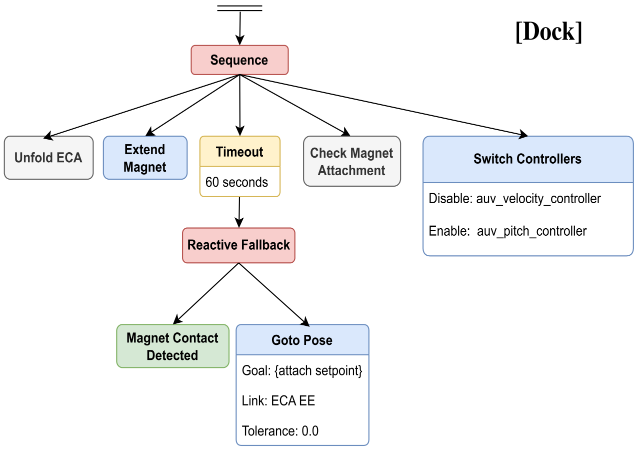


FIGURE 9. Behavior Tree of the docking procedure.

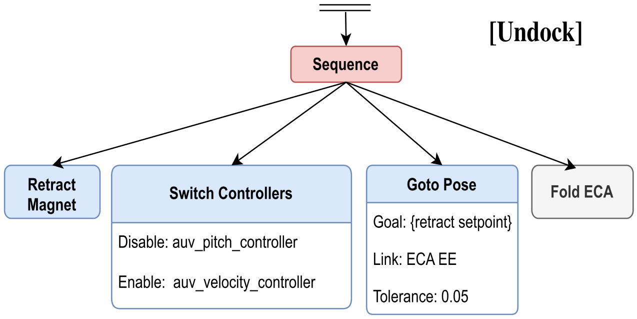


FIGURE 10. Behavior Tree of the undocking procedure.

The undocking procedure, depicted in Fig. 10, starts by retracting the magnet. Once detached, it starts the AUV velocity controllers and it commands the AUV to a retreat position. Finally, it folds the ECA manipulator.

2) FOLDING/UNFOLDING MANIPULATORS

The BTs responsible for the folding and unfolding of the manipulators combines the use of the **Switch Controllers** and **Follow Trajectory** actions. When the mission requires folding or unfolding of the manipulators, the **Switch Controllers** action is used to switch from the joint velocity controller to the joint trajectory controller. This is followed by the **Follow Trajectory** action, which sends a predefined trajectory to the joint group trajectory controller to perform the preprogrammed motion sequence. Once the motion is complete, the **Switch Controllers** action is used again to switch back to the joint velocity controller, allowing the TP Kinematic Control layer to drive the manipulators. By using a combination of these two actions, the behavior tree can efficiently execute folding and unfolding tasks without having to write custom code for each specific motion sequence (see Fig. 11).

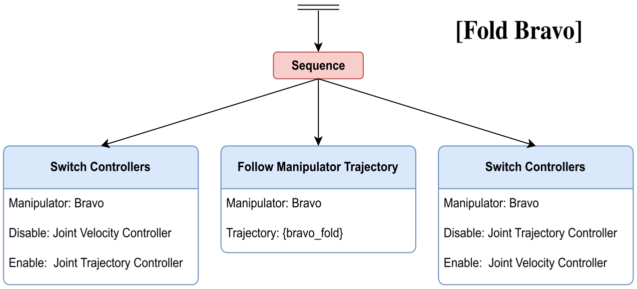

[Fold Bravo]
Sequence
Switch Controllers
Manipulator: Bravo
Disable: Joint Velocity Controller
Enable: Joint Trajectory Controller
Follow Manipulator Trajectory
Manipulator: Bravo
Trajectory: {bravo_fold}
Switch Controllers
Manipulator: Bravo
Disable: Joint Trajectory Controller
Enable: Joint Velocity Controller

FIGURE 11. Behavior Tree of the folding of the Bravo manipulator.

3) PROBING BEHAVIOR TREE

Given a set of points defined in the structure's surface, the probing procedure (see Fig. 12) aims to precisely touch each point using the CP probe mounted on the Bravo arm to perform measurements. Touching a point is structured in three phases: First, using a **Goto** action, the probe is commanded in a favorable pose close to the setpoint. Then, a **Goto Compliant** action is enabled, which uses the adaptive admittance control explained in Section VI to guarantee contact with the structure's surface during the measurement. The setpoints are stored in a stack. The node **Pop Setpoint** pops a setpoint off the stack and stores it in the *{probe_setpoint}* variable, and the node **IsSetpointsStackEmpty** is a condition that returns *Success* if the stack is empty and *Failure* if it is not. These nodes enable the capacity to iterate through the setpoints in a while-loop fashion, using a **Repeat until failure**. This node is wrapped with a **Fallback** to check if the *Failure* was due to an error during the probing (*Failure*) or due to an empty stack (*Success*).

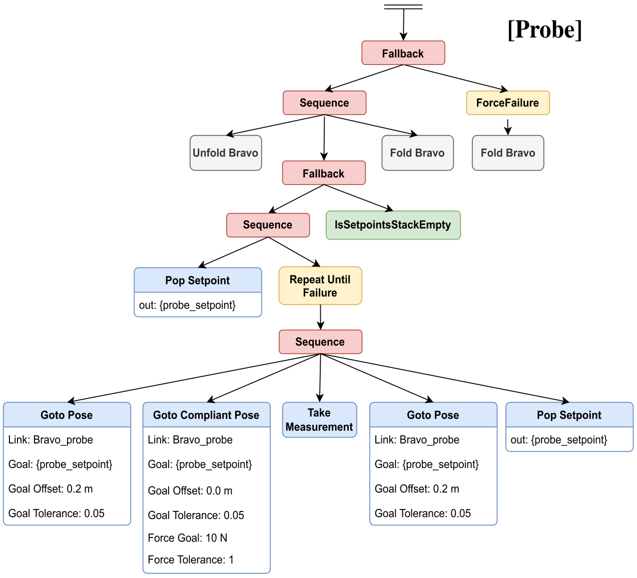


FIGURE 12. Behavior Tree of the probing procedure. The predefined probing setpoints are stored in a stack that pops a setpoint to the Blackboard on every iteration until the stack is empty.

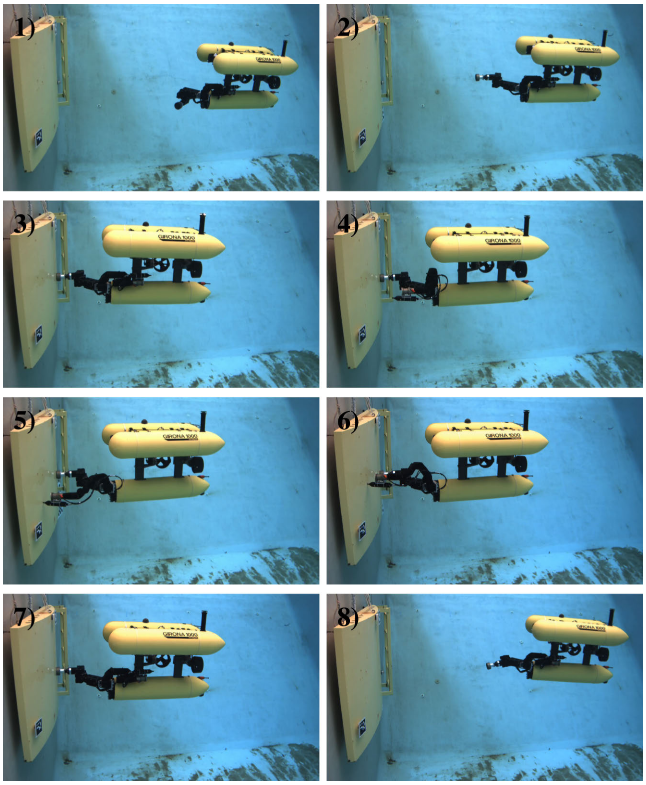


FIGURE 13. The Girona1000 I-AUV performing the CP survey operation sequence, at the CIRS water tank. In 1), the Girona1000 as detected the structure and approached a starting pose. In 2), the ECA manipulator unfolds and the piston extends to expose the magnetic EE. In 3), the I-AUV has successfuly attached to the structure. In 4), the Bravo manipulator unfolds to perform CP measurements (5,6). In 7), the Bravo manipulator folds back to, next, detach from the structure (8).

IX. EXPERIMENTAL VALIDATION

A. RESULTS

This section presents the results obtained during the experiment at the CIRS water tank, using the mockup structure presented in Section III-B. A video demonstrating the whole mission in the CIRS water tank can be found at the following URL[1]: https://youtu.be/vdijAgp6z8w.

[1] www.youtube.com/@cirsudg

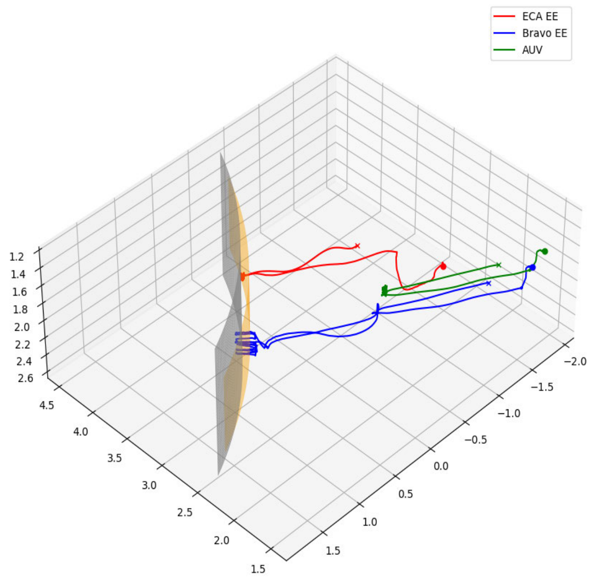


FIGURE 14. Trajectory of the I-AUV during the executed mission. Shown are the Girona1000 path (green), the ECA EE (red), the CP probe mounted on the Bravo (blue), the structure's diameter with which the setpoints where generated (orange), and the real structure's diameter (gray). The start of the trajectories is marked with a circle and the end is marked by a cross.

Figure 13 shows a sequence of eight snapshots from the real experiment, showcasing the I-AUV in action as it autonomously approaches, attaches to the structure, performs the CP survey, and detaches.

The trajectory of the I-AUV and the end-effector of both manipulators during the mission are shown in Fig. 14. The plotted trajectories displays the attaching, probing, and retreat, being part of the beginning and end removed to ensure clarity.

Figure 15 illustrates how the different BTs coordinate the switching of TP tasks and the manipulators' controllers.

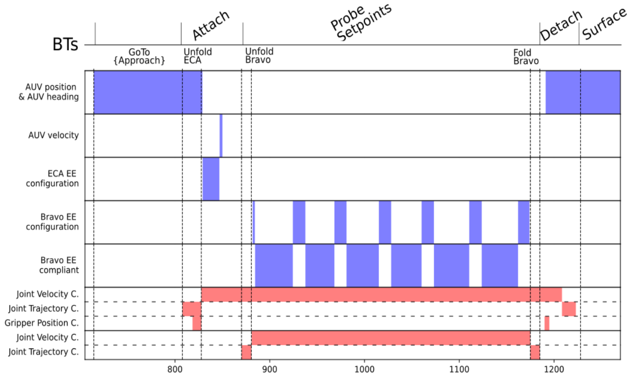


FIGURE 15. Diagram of the TP tasks and controllers activation by the different behaviors.

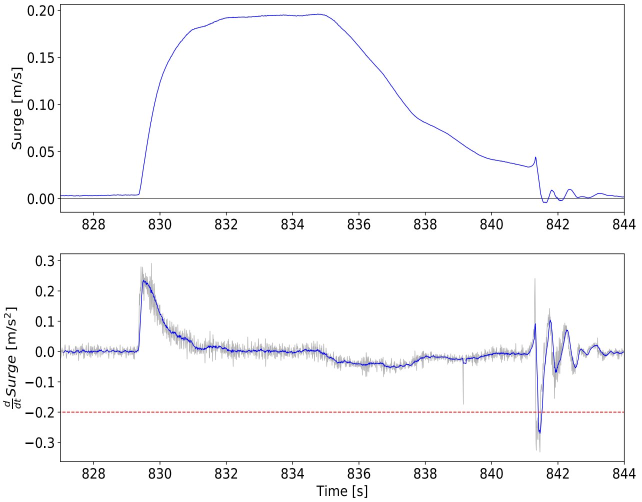


FIGURE 16. Detection of the magnet contact to the structure. Top: AUV surge velocity during the operation. Bottom: The filtered surge derivative (blue) and the trigger threshold (red).

Figure 16 presents data from the moment of attaching to the structure. It shows how the impact of the magnetic gripper with the structure was detected by monitoring the derivative of the I-AUV's current velocity while it was approaching the structure. The sudden drop in velocity can only be explained by a collision, which indicates that the I-AUV has successfully touched the structure. Nonetheless, after the touch occurs, the I-AUV tests if the attachment was successful by attempting to move backward (see the activation of the AUV velocity task in Fig. 15).

For the inspection operation, the I-AUV was commanded to maintain contact between the tip of the CP probe and the surface of the structure at 6 different points. To ensure that the contact always occurs, no matter the errors in the localization of the structure, the setpoints were generated behind the surface. This was done by assuming that the structure's diameter was smaller than in reality. The compliant control algorithm ensures that a safe approach to these setpoints is possible. Figure 17 shows the measured and target contact forces between the probe and the structure, as well as the desired and actual locations of the probe tip. It can be appreciated how the setpoint location converges to the current location of the tip during the contact. It can be noticed that the force control oscillates significantly. This effect occurs due to the fact that the thrusters used in the vehicle are not possible to control in force but only using a synthetic setpoint, which is not directly related to any physical quantity. The relation between the thrust and setpoint was roughly identified using a static thrust experiment and encoded by polynomial functions. However, these functions do not account for the dynamical effects, occurring during navigation and when the thrusters change direction of rotation frequently. The second situation is common when trying to control the contact forces precisely. Moreover, the propeller design used in the thrusters is not optimized for positioning but rather speed.

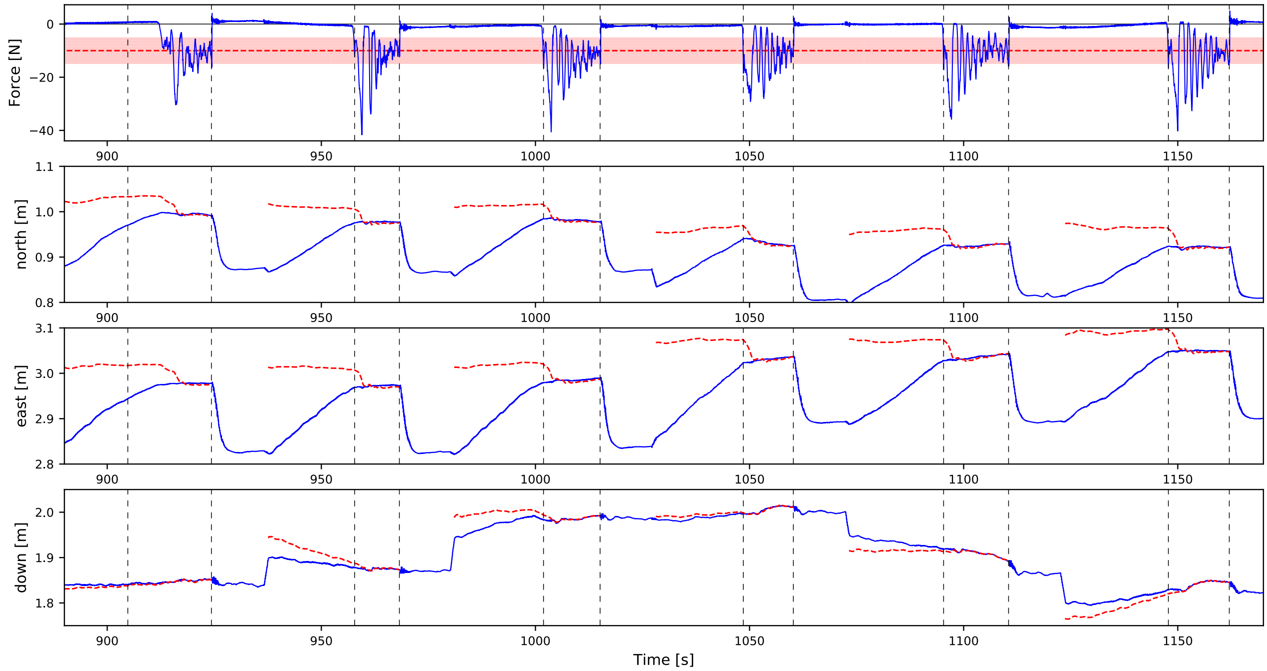


FIGURE 17. On top, tracking of the desired contact force along the z-axis of the EE The blue line is the measured force, the red dashed line is the desired force, and the red area is the region within the force threshold. The remaining plots denote the X, Y, and Z position of the EE (blue) and its target (dashed red), which is deformed by the adaptive force controller. The vertical dashed lines show the duration of contact.

X. CONCLUSION

The paper presents a complete system for an I-AUV to perform inspection of marine structures, which requires contact force control, as is the case of CP inspection for offshore wind turbines. The use of behavior trees allowed for the automation of the mission, simplifying the coordination of the different stages of the mission. The impact detection strategy, based on monitoring the approach velocity, provided a feasible solution when force measurement was not available. The attachment to the structure, by using a permanent magnet EE, improves the accuracy of the intervention operations requiring contact. The control ideas are based on a combination of admittance force control and the Task Priority control framework. The admittance control was modified to allow for direct control over the contact force value, by introducing a self-adjusting stiffness matrix. Moreover, the algorithm generates a feed-forward force to be applied by the vehicle's thrusters to counteract the reaction force at the contact point, preventing the detachment of the robot from the structure. The system was validated in the CIRS water tank using a mockup of a cylindrical pylon of an offshore wind turbine.

Overall, the system presented in this paper demonstrates the potential of using I-AUVs for efficient and accurate inspection of marine structures, which can have significant implications for the maintenance and safety of offshore infrastructures.

REFERENCES

[1] P. Ridao, M. Carreras, D. Ribas, P. J. Sanz, and G. Oliver, "Intervention AUVs: The next challenge," *IFAC Proc. Volumes*, vol. 47, no. 3, pp. 12146–12159, 2014.

[2] E. Zereik, M. Bibuli, N. Mišković, P. Ridao, and A. Pascoal, "Challenges and future trends in marine robotics," *Annu. Rev. Control*, vol. 46, pp. 350–368, 2018.

[3] G. Marani and J. Yuh, *Introduction to Autonomous Manipulation, Case Study With an Underwater Robot, SAUVIM*, vol. 102. Berlin, Germany: Springer, Apr. 2014.

[4] E. Simetti, G. Casalino, S. Torelli, A. Sperindé, and A. Turetta, "Floating underwater manipulation: Developed control methodology and experimental validation within the TRIDENT project," *J. Field Robot.*, vol. 31, no. 3, pp. 364–385, May 2014.

[5] E. Simetti, F. Wanderlingh, S. Torelli, M. Bibuli, A. Odetti, G. Bruzzone, D. L. Rizzini, J. Aleotti, G. Palli, L. Moriello, and U. Scarcia, "Autonomous underwater intervention: Experimental results of the MARIS project," *IEEE J. Ocean. Eng.*, vol. 43, no. 3, pp. 620–639, Jul. 2018.

[6] P. Cieslak, P. Ridao, and M. Giergiel, "Autonomous underwater panel operation by GIRONA500 UVMS: A practical approach to autonomous underwater manipulation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 529–536.

[7] A. Carrera, M. Carreras, P. Kormushev, N. Palomeras, and S. Nagappa, "Towards valve turning with an AUV using learning by demonstration," in *Proc. MTS/IEEE OCEANS*, Jun. 2013, pp. 1–7.

[8] D. Youakim, P. Ridao, N. Palomeras, F. Spadafora, D. Ribas, and M. Muzzupappa, "Autonomous underwater free-floating manipulation using MoveIt!" *IEEE Robot. Autom. Mag.*, vol. 24, no. 3, pp. 41–51, 2017.

[9] D. Youakim, P. Cieslak, A. Dornbush, A. Palomer, P. Ridao, and M. Likhachev, "Multirepresentation, multiheuristic A* search-based motion planning for a free-floating underwater vehicle-manipulator system in unknown environment," *J. Field Robot.*, vol. 37, no. 6, pp. 925–950, Sep. 2020.

[10] A. Palomer, P. Ridao, D. Youakim, D. Ribas, J. Forest, and Y. Petillot, "3D laser scanner for underwater manipulation," *Sensors*, vol. 18, no. 4, p. 1086, Apr. 2018.

[11] A. Birk et al., "Dexterous underwater manipulation from onshore locations: Streamlining efficiencies for remotely operated underwater vehicles," *IEEE Robot. Autom. Mag.*, vol. 25, no. 4, pp. 24–33, Dec. 2018.

[12] S. Lensgraf, A. Sniffen, Z. Zitzewitz, E. Honnold, J. Jain, W. Wang, A. Li, and D. Balkcom, "Droplet: Towards autonomous underwater assembly of modular structures," in *Proc. Robot., Sci. Syst.*, 2021.

[13] S. Lensgraf, D. Balkcom, and A. Q. Li, "Buoyancy enabled autonomous underwater construction with cement blocks," 2023, *arXiv:2305.05552*.

[14] R. Pi, P. Cieslak, P. Ridao, and P. J. Sanz, "TWINBOT: Autonomous underwater cooperative transportation," *IEEE Access*, vol. 9, pp. 37668–37684, 2021.

[15] S. Heshmati-Alamdari, G. C. Karras, and K. J. Kyriakopoulos, "A predictive control approach for cooperative transportation by multiple underwater vehicle manipulator systems," *IEEE Trans. Control Syst. Technol.*, vol. 30, no. 3, pp. 917–930, May 2022.

[16] P. Dai, W. Lu, K. Le, and D. Liu, "Sliding mode impedance control for contact intervention of an I-AUV: Simulation and experimental validation," *Ocean Eng.*, vol. 196, Jan. 2020, Art. no. 106855.

[17] S. Heshmati-Alamdari, C. P. Bechlioulis, G. C. Karras, A. Nikou, D. V. Dimarogonas, and K. J. Kyriakopoulos, "A robust interaction control approach for underwater vehicle manipulator systems," *Annu. Rev. Control*, vol. 46, pp. 315–325, 2018.

[18] P. Cieslak and P. Ridao, "Adaptive admittance control in task-priority framework for contact force control in autonomous underwater floating manipulation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 6646–6651.

[19] 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.

[20] W. J. Marais, S. B. Williams, and O. Pizarro, "Maximising wrenches for kinematically redundant systems with experiments on UVMS," 2022, *arXiv:2202.13535*.

[21] F. J. Romero-Ramirez, R. Muñoz-Salinas, and R. Medina-Carnicer, "Speeded up detection of squared fiducial markers," *Image Vis. Comput.*, vol. 76, pp. 38–47, Aug. 2018.

[22] D. Faconti and M. Colledanchise. (2021). *Behaviortree.cpp*. [Online]. Available: https://www.github.com/BehaviorTree/BehaviorTree.CPP

[23] S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators," *IEEE Trans. Robot. Autom.*, vol. 13, no. 3, pp. 398–410, Jun. 1997.

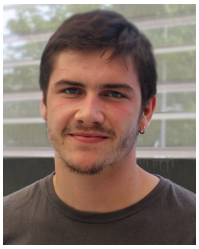
[24] S. Chitta, E. Marder-Eppstein, W. Meeussen, V. Pradeep, A. R. Tsouroukdissian, J. Bohren, D. Coleman, B. Magyar, G. Raiola, M. Lüdtke, and E. F. Perdomo, "Ros_control: A generic and simple control framework for ROS," *J. Open Source Softw.*, vol. 2, no. 20, p. 456, Dec. 2017.

[25] E. Simetti and G. Casalino, "A novel practical technique to integrate inequality control objectives and task transitions in priority based control," *J. Intell. Robot. Syst.*, vol. 84, nos. 1–4, pp. 877–902, Dec. 2016.

[26] S. Moe, G. Antonelli, A. R. Teel, K. Y. Pettersen, and J. Schrimpf, "Set-based tasks within the singularity-robust multiple task-priority inverse kinematics framework: General formulation, stability analysis, and experimental results," *Frontiers Robot. AI*, vol. 3, p. 16, Apr. 2016.

[27] B. Siciliano, O. Khatib, and T. Kröger, "Force control," in *Springer Handbook of Robotics*, vol. 200. London, U.K.: Springer, 2017, pp. 363–405.

[28] M. Colledanchise and P. Ögren, *Behavior Trees in Robotics and AI*. Boca Raton, FL, USA: CRC Press, Jul. 2018.



ROGER PI received the B.S. degree in computer engineering from Universitat de Girona, Catalonia, in 2017, and the joint M.Sc. degree in computer vision and robotics from the University of Burgundy, Universitat de Girona, and Heriot-Watt University, in 2019. He is currently pursuing the Ph.D. degree with the Underwater Robotics Research Center (CIRS), under the supervision of Dr. Pere Ridao. His current research interest includes autonomous intervention for underwater vehicle manipulator systems (UVMSs). He received the Best Master Student Award from the University of Burgundy, Universitat de Girona, and Heriot-Watt University.

PATRYK CIEŚLAK received the Ph.D. degree from the Department of Robotics and Mechatronics, AGH University of Science and Technology, Kraków, Poland, in 2016. Since 2017, he has been with the Underwater Vision and Robotics Laboratory (CIRS), University of Girona. Initially as a Marie Skłodowska-Curie Postdoctoral Fellow, he developed control algorithms for autonomous non-destructive testing using an I-AUV. He is the co-author of a commercial rehabilitation robot called Prodrobot, which is a stationary lower limbs exoskeleton, used in the learning and improvement of natural gait patterns in children. He is the author of an advanced open-source robot simulator called Stonefish, designed for marine robotics. This simulator is currently used in all underwater-related research at CIRS and in other institutions around the world and can be found on (github.com/patrykcieslak). His current research interests include underwater manipulation, including obstacle avoidance, motion planning, and cooperative manipulation with two I-AUVs. He is continuing his research at CIRS, working on underwater intervention for inspection, maintenance, and repair (IMR) operations in offshore wind farms.

JOAN ESTEBA was born in Girona, in 1994. He received the bachelor's degree in industrial technologies engineering and the M.Sc. degree in industrial engineering from the University of Girona (UdG), in 2016 and 2018, respectively, where he is currently pursuing the Ph.D. degree with the Computer Vision and Robotics Research Institute (VICOROB). He had been working for several years developing vision and robotic technologies in the automobile industry and leading different projects. He also had been leading some development projects in the food industry. His current research interest includes docking systems for autonomous underwater robotic systems. He is a Registered Professional Engineer in Catalonia.

NARCÍS PALOMERAS (Member, IEEE) was born in Vilafant, in 1981. He received the Ph.D. degree from the University of Girona, in 2011. He has participated in several research projects, all related to underwater robotics, both national and European (TRIDENT, PANDORA, MORPH, MERBOTS, LOONDOCK, TWINBOT, 3DAUV, and ATLANTIS...) and in different European competitions for AUVs, such as SAUC-E and ERL. Currently, he is the Coordinator of a Joint Degree Erasmus Mundus about intelligent field robotic systems. He is also a Postdoctoral Researcher with the Computer Vision and Robotics Research Institute (VICOROB), University of Girona. His current research interests include underwater robotics in topics, such as planning, exploration, intelligent control architectures, mission control, and localization.

PERE RIDAO (Member, IEEE) received the Ph.D. degree in industrial engineering from the University of Girona, in 2001. He was the Co-Founder of Iqua Robotics S.L., a spin-off company. He is currently the Head of the Underwater Robotics Research Center (CIRS) and a Full Professor with the Department of Computer Engineering, University of Girona. Since 1997, he has participated in 24 research projects (15 European and nine national). He is the author of more than 100 publications. He has supervised nine Ph.D. thesis (four more are currently under direction) and 14 M.S. Thesis. He is the coauthor of four licenses and one Spanish/European patent. His current research interests include designing and developing autonomous underwater vehicles for 3D mapping and intervention. He has served as the Chair for the IFAC's Technical Committee on Marine Systems.