Increasing the Autonomy Levels for Underwater Intervention Missions by using Learning and Probabilistic Techniques

Jorge Sales[1], Luís Santos[2], Pedro J. Sanz[1], Jorge Dias[2,3], and J. C. García[1]

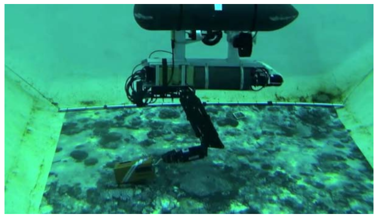
[1] Computer Science and Engineering Dept., University of Jaume-I, Castellón, Spain
salesj@uji.es,
[2] Institute of Systems and Robotics, Department of Electrical and Computer Engineering, University of Coimbra, Coimbra, Portugal
[3] Robotics Institute, Khalifa University, Abu Dhabi, UAE

Abstract. This paper represents research in progress in autonomous manipulation for underwater intervention missions within the context of the GRASPER project. This project focuses on developing manipulation skills for an Autonomous Underwater Vehicle (AUV). Current research in underwater robotics intends to increase autonomy for all kinds of robotic intervention operations that require physical interaction. Very few underwater systems have the capacity to carry out intervention without any kind of umbilical cables for tele-operating the actions. This article aims to investigate new approaches to follow with the aforementioned challenges, with the inclusion of learning and probabilistic techniques to increase the autonomy levels of an underwater manipulation system. With this goal, a collaboration research action has been established between the IRS-Lab at UJI (Spain), as experts in the underwater robotic manipulation domain, and the Institute of Systems and Robotics from University of Coimbra (Portugal), experts in learning by interaction within a robotic manipulation context.

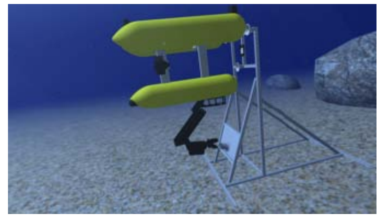
Keywords: Underwater Autonomous Intervention, Bayesian Learning, Dynamic Bayesian Network, UWSim underwater realistic simulator.

1 Introduction

This paper discuss the research in progress, under development by UJI-ISR cooperation action, in the context of autonomous underwater intervention missions. Current research in the underwater robotics intends to increase autonomy for all kinds of robotic intervention operations requiring physical interaction. Despite the fact that autonomous robotic intervention on land remains in development and with some valuable achievements, the current state-of-the-art in underwater intervention missions is currently in a very primitive stage where the majority of the systems are tele-operated by an expert user. This paper addresses this challenge through research that stills under development, within the context of a project, funded by the Spanish Ministry, titled GRASPER. GRASPER (under



(a) search and recovery of an object of interest (e.g. a "black-box mockup" from a crashed airplane.



(b) the intervention of an underwater panel in a permanent observatory.

Fig. 1: TRITON Spanish coordinated project proposed scenarios.

the responsibility of University of Jaume-I, UJI, and addressing the problem of the "Autonomous Manipulation") represents only a sub-project inside a Spanish Coordinated Project, entitled: TRITON[4], "Multisensory Based Underwater Intervention through Cooperative Marine Robots", which includes two other subprojects: COMAROB ("Cooperative Robotics", under the responsibility of University of Girona, UdG), and VISUAL2 ("Multisensorial Perception", under the responsibility of University of Balearic Islands, UIB). In summary, TRITON is a marine robotics research project focused on the development of intervention technologies really close to the real needs of the final user and, as such, it can facilitate the potential technological transfer of its results. The project proposes two scenarios to test the concept, and to demonstrate the developed capabilities: (1, Figure 1a) the search and recovery of an object of interest (e.g. a "black-box mockup" from a crashed airplane), and (2, Figure 1b) the intervention of an underwater panel in a permanent observatory.

The specific objectives for GRASPER are the following:

(a) To develop the user interface and simulation capabilities needed for TRITON.
(b) To generate all the mechatronics and sensor improvements to succeed in the autonomous manipulation requirements.
(c) To develop new planning and control strategies, making use of range and visual information, finally leading to visual free floating manipulation.

This paper highlights the potential benefits of including a new approach based on the "learning by demonstration" paradigm, in order to increase autonomy in the required grasping and manipulation skills. Because initially the experimental validation will be carried out in virtual reality (i.e. by using the 3D simulator UWSim [1] described below), some contributions are expected in the aforementioned objectives (a) and (c).

[4] Multisensory Based Underwater Intervention through Cooperative Marine Robots (TRITON), available: http://www.irs.uji.es/triton/

1.1 Initial Strategy and Roadmap

The activities developed in this research activity follow a methodology where the core techniques can be designed, developed and prototyped with support of a simulator named UWSim [1]. The research results generated by this activity are after, tested on real scenarios with different levels of complexity. The Figure 2 provides a graphical perspective of this strategy.

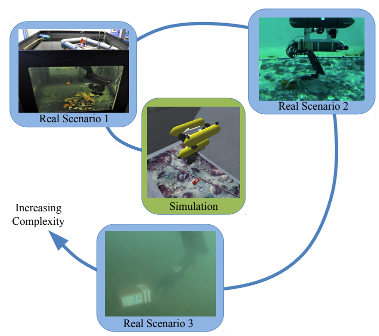


Fig. 2: Development strategy: the core techniques can be designed, developed and prototyped inside the UWSim simulator. Then, the research results generated by this activity are tested on real scenarios with increasing scales of complexity.

This methodology and the modular computational architecture is based on the *Robot Operating System (ROS)* and provides the support for prototyping a solution based on a simulator that can be used to target the real robot, in different real scenarios. The architecture allows us to switch from the simulated environment to a real scenario at any moment and test the prototyped system (manipulation, new algorithms, learning, etc.). The real test scenarios include different physical complexities with increasing degree of realism and hard conditions, when compared with open sea conditions:

- Testbed 1: Water Tank (described below) (UJI, Castellón, Spain)
- Testbed 2: CIRS pool at Girona (UdG, Girona, Spain)
- Testbed 3: Roses Harbour (Roses, Spain)

For each development step or research outcome it is possible to introduce more complex scenarios by simulating them on UWSim system and test the results in different testbeds that convey real hardware in real environments with increasing number of uncontrolled variables (disturbances, visibility, noise, etc.).

1.2 Related Work

In the field of the underwater intervention it is worth mentioning previous projects like SAUVIM [2], intended for deep interventions, which demonstrated the autonomous recovery of seafloor objects by using a very bulky and expensive system; and TRIDENT[5] [3], that demonstrated the first multipurpose object search and recovery strategy in 2012, able to operate in shallow waters. Nowadays, two ongoing projects are running in the underwater intervention context funded by European Commission: MORPH[6] and PANDORA[7]. It is also noticeable, that the ongoing TRITON project is an extension of the previous Spanish founded project RAUVI[8] [4]. RAUVI was the origin of TRIDENT, demonstrating in 2011 a successful approach for the search and recovery problem but in a more limited manner.

Manipulation Learning: One trend in the current state of the art concerns learning based on geometric properties of manipulation movements to identify the different manipulation stages [5], or continuously learning constraints in a Gaussian Mixture Model approach [6]. Kondo [7], Bernardin [8] and Kruger [9] use symbolic representations encoding hand-object contacts states, temporally represented in Hidden Markov Models (HMM) or Markov Decision Processes. Bekiroglu [10] proposes an approach, using Support Vector Machine (SVM) and HMM to learn and assess robotic grasping stability. Lin [11] applies GMM to learn required fingertip force and pose, to obtain a stable grasp during dexterous manipulation tasks. A different approach is presented in [12], which applies inverse reinforcement learning techniques to infer the underlying task, which is being executed by the demonstrator. Another example comes from Jetchev [13] which adapts inverse optimal control techniques [14] to a single grasping task on a real robotic platform. Beyond the terrestrial applications, there are manipulation platforms working on space to fix satellites [15], where the robot is taught remotely by human operators using an immersive interface with sensorial feedback. Underwater scenarios have only recently been addressed, e.g. in [16] autonomous mobile manipulation in shallow water using a single robotic arm is presented. Recently, Carrera et al. [17], propose a learning solution for autonomous robot valve turning, using Extended Kalman Filtering and Fuzzy Logic to learn manipulation trajectories via kinaesthetic teaching.

1.3 Our previous approach

With the aim of increasing the autonomy levels of the underwater manipulation systems, we have recently been working in a multisensory based manipulation

[5] Marine Robots and Dexterous Manipulation for Enabling Autonomous Underwater Multipurpose Intervention Missions (FP7-TRIDENT), available: `http://www.irs.uji.es/trident/`

[6] Marine Robotic System of Self-Organizing, Logically Linked Physical Nodes (FP7-MORPH), available: `http://morph-project.eu/`

[7] Persistent Autonomy through learNing, aDaptation, Observation and Re-plAnning (FP7-PANDORA), available: `http://persistentautonomy.com/`

[8] Reconfigurable Autonomous Underwater Vehicle for Intervention (RAUVI), available: `http://www.irs.uji.es/rauvi/`

approach[9]. This approach allows the grasp of different known-a-priori objects in a water tank, but still requires the user intervention in order to specify the grasp. Some important pieces of this approach are now described:

UWSim: the underwater simulator: UWSim is a software tool for visualization and simulation of underwater robotic missions [1]. The software is able to visualize an underwater virtual scenario that can be configured using standard modeling software. Controllable underwater vehicles, surface vessels and robotic manipulators, as well as simulated sensors, can be added to the scene and accessed externally through network interfaces. UWSim do the interface with external control programs through the *Robot Operating System (ROS)* (see additional details in section 5). UWSim has been successfully used for simulating the logics of underwater intervention missions and for reproducing real missions from the captured logs [1]. UWSim is currently used in different ongoing projects funded by European Commission (MORPH and PANDORA) in order to perform HIL (Hardware in the Loop) experiments and to reproduce real missions from the captured logs.

3D Reconstruction of the Scene: The aforementioned approach requires the reconstruction of the geometry of the objects laying on the floor. To achieve this, a scan of the scene is performed using a structured laser beam attached to the forearm of the manipulator. The scan is done by moving the elbow joint of the manipulator at a constant velocity. At the same time, a digital video camera is used to capture the scene with the laser beam projected on the object. A visual processing algorithm runs in parallel: the laser peak detector, which is in charge of segmenting the laser stripe from the rest of the image and computing the 3D points [18]. With these points, a 3D point cloud of the scene is built and represented on the simulator.

2 Problem Statement and Definitions

The challenging problem addressed in this manuscript is within the context of underwater robotics. Assume there is an object O represented by a set of characteristics C_O, located in a 3 Dimensional underwater space $U \in \mathcal{R}^3$. Consider a n DoF manipulator M, operating in U. The challenge is to give M a set of skills S, such that M is able to **reach**, **grab** and **manipulate** O into reaching a user specified goal G. At a first stage, this knowledge S is taught by a human expert. Posteriorly, the manipulator exploits the skill space S, in order to operate autonomously into solving $D(G, O)$. From this scenario (Figure 3), we are able to identify the following main problems:

1. The development of a realistic simulation environment, that a human operator can control and, simultaneously, from which it is able to get realistic feedback while teaching, via "tele-operation", a virtual representation of M.

[9] Underwater semi-autonomous grasping experiments using laser 3D reconstruction can be seen on-line: Experiment 1: `http://youtu.be/VOLNBWfeoLs`, Experiment 2: `http://youtu.be/c62FTTycxsQ`, Experiment 3: `http://youtu.be/42ZklVwNaqc`.

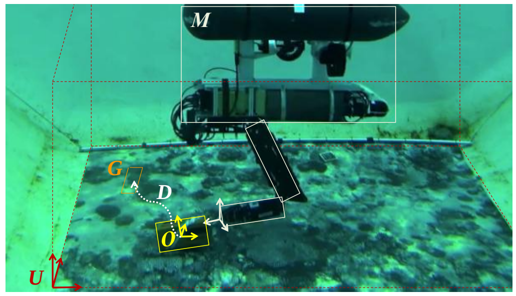


Fig. 3: Underwater intervention scenario. A manipulator M detects and object O in a workspace U. Upon a user specified goal G (or task), it should be capable of, autonomously, estimate a solution D for successfully accomplish its mission.

2. Find a suitable, probabilistic knowledge representation, which accurately models the relation between a set of manipulator sensed information I and a set of skills S, such that M can interpret scene information, identify objects of interest and decide the best course of action D into satisfying G. The solution D, should be updated every time new information is available, so to be able to cope with dynamic and difficult underwater operation conditions.
3. Given a solution for G, project D into a set of motor primitives, allowing the mechanical system M to operate the different steps of its intervention mission.
4. Define a metric to evaluate the success of each intervention, so the system has the capability to decide whether or not the new proposed solution for G should be incrementally added to existent knowledge S.

The proposed solution to this problem can be easily stated: a autonomous manipulator M should be able to decide the best solution D for a given user specified goal G, based on the information I its sensors are able to acquire from the environment U. Such information is, at its most basic forms, identity and pose of objects O, obstructions and its relative End-Effector M pose towards a specified goal G. The solution and integration of these problems are expected to provide an intelligent system, capable of autonomously perform underwater tasks, with minimum human intervention, while being able to constantly adapt to the difficult underwater conditions.

Data Acquisition Via Realistic Underwater Simulator Implementation of the HRI simulator, ensuring it will acquired the necessary data for learning the manipulation skills. The data acquired from the user controlling the simulator will be used to develop a filter for assessing what is considered a good trial or not, deciding which trials can be included in the learning.

UWSim do the interface with external control programs through the *Robot Operating System (ROS)*. This architecture provides message-passing and communication between nodes in a transparent manner, thus allowing both local and remote localization of executing nodes (the simulator itself, the learning and database modules, the user interface, etc.). As a consequence of this, we are able to run the whole system in a single computer but also in a distributed system, allowing thus **remote** learning.

3 Manipulation Skills: Phases, Information and Tasks

In our manipulation scenario we parametrize the execution in 4 different stages:

1. **Initial**: The initial stage is a stage where the robot acquires initial information from the scene, and starts the first iterations to identify scene properties and find initial solution D for the proposed user defined task T.
2. **Approach**: in this stage, the system will refine its assessment of the environment conditions and gather extra scene information, adjusting its behavior during the reach to grasp trajectory.
3. **Reach/Contact**: once in the neighborhood of the target object, the manipulator needs to decide the best pose and force parameters to enter in contact with the object.
4. **Contact/Manipulation**: at this stage, the manipulator needs to operate the gripper in order to move the object from an initial to a final position, i.e. a second goal G, which is defined by a user specified goal OR automatically assessed from the available sensed information.

We propose a log-spherical intermediate defined key points, at which the manipulation should verify its own attitude towards intermediate and final goals. An example is show in Figure 4. We define **Attitude** as the End-Effector pose, velocity and gripper state, with respect to a specific goal. This attitude should be inferred based on information acquired from the laser scanner and vision system.

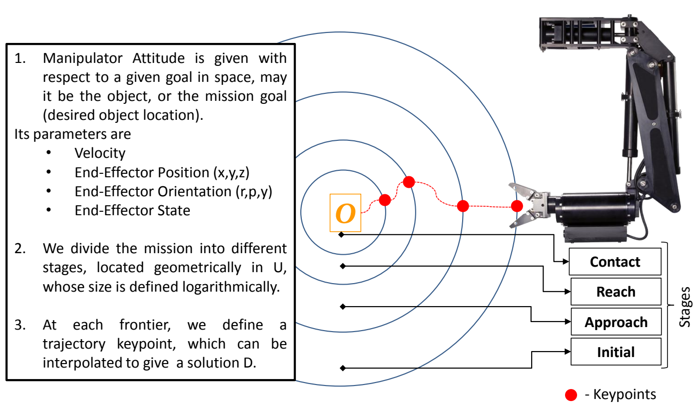


Fig. 4: Proposed log-spherical space for manipulation stage division. The closer M is to its goal, it should be assessing its attitude more frequently.

Solutions are addressed from **different system perspectives**. At each of these stages a supervised learning process is applied, using a tele-operated realistic simulator environment, from which data from the scene, from the manipulator and from the user controlling the simulator will be recorded in a database for posterior analysis. Our goals is to map a set of sensed information I into a set of skills S observed during tele-operated execution. The data will then be associated by means of probabilistic density functions, into developing an autonomous decision making framework. We propose a system which will make its decisions according to information from different perspectives:

1 Sensing Solution: We start by defining a workspace region, which can be reachable by the manipulator. Sensed information will be projected into an occupancy grid space and processed for developing an interpretation model of the scene, objects and actions. Segmented information is complemented and associated with the manipulator attitude parameters.

2 "Egocentric" Solution: With this approach, we will project all information, as it would be seen by the gripper perspective. We aim at comparing the egocentric approach, which will encompass possibly less and different data, to the proposed sensing solution in terms of manipulation efficiency.

4 Learning Manipulation Skills: Probabilistic Modelling

Learning the adequate end-effector attitude M should exhibit at the different phases, will encompass the definition of approach, contact and manipulation skills based on sensed information and using a learning approach. Unknown sequences will be conducted in order to assess the framework scene interpretation and decision capabilities. We propose a Dynamic Bayesian Model as a support methodology into solving the decision making framework. The development of our model follows the formalism of Bayesian Programming [19], which allows efficient and coherent model development. It defines 4 main stages, which are described in the forthcoming subsections.

Stage 1::Variable Definitions: Bayesian Programming formalism starts by identifying the relevant variables to the problem. Let us recall we are interested in estimating the End-Effector attitude, at the Different phases of the intervention, based on sensed information and considering a user-specified goal. We will address each goal as an independent problem, which can be posteriorly generalizable. This is due to the fact there are two different goals for our manipulator to succeed in a mission: grabbing and object of interest AND move/manipulate it into a specified goal. The mission goals is defined as a tuple $\{Object, Action\}$, where the user manually orders the system to find an object and act on it. From this description we can define the following variables:

- **Attitude Velocity** V: this variable defines the velocity which the end effector should exhibit at each given key point.
- **Attitude Position** $X \in \mathcal{R}^3$: this random variable vector defines the spatial location (x, y, z), relative to object O coordinate system, in which the end-effector must be positioned, so to perform a correct approach. This location must be unobstructed, otherwise the system should select next candidate location. In the presence of singularities, human tele-operation is required.
- **Attitude Orientation** $\Gamma \in \mathcal{R}^3$: this random variable vector defines the orientation of the End-Effector relative to the estimated O pose.

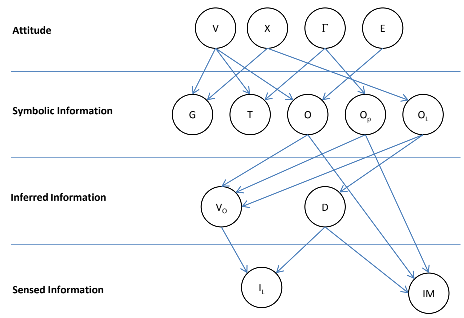


Fig. 5: Directed Acyclic Graph of the proposed Dynamic Bayesian Network. Nodes represent variables and directed arcs represent variable dependencies.

- **Attitude End-Effector** E: this random variable has two different states $\{open, closed\}$. While scenario complexity increases, it will become a real valued tupple, representing both opening width and pressure.
- **Object Identity** O: this variable space state is $\{Box, \neg Box\}$.
- **Object Location** $O_L \in \mathcal{R}^3$: this random variable defines the location of O.
- **Object Pose** $O_P \in \mathcal{R}^3$: this random variable vector states the object pose with respect to U.
- **Estimated End-Effector to Object Distance** D: random variable with the estimation for the relative distance of the end-effector to the object, and at a subsequent decision stage, from the end-effector to the target Location.
- **Occupancy** V_o: this variable represent the state of a given cartesian location in U. For simplicity purposes, we consider this to have two possible states: Occupied or Empty. The information of this state is retrieved from an occupancy map taken from the readings of a Laser Range Finder, but will be held into consideration to be estimated from a stereo vision system.
- **Laser Information** I_L: this random variable vector will represent laser range finder information.
- **Image Information** IM: this random variable vector contains image features and characteristics of segmented objects of interest.
- **Task** T: is a random variable vector defining the end pose of O.
- **Goal** G: is a tupple containing information about an object identity O and a specified task T.

We can now define the joint distribution J of our model as:

$$J = P(V, X, \mathit{\Gamma}, E, O_L, O_p, O, V_o, I_L, I_i, T, G) \tag{1}$$

Stage 2::Decomposition: The second stage of the Bayesian Program is to define the decomposition of the joint distribution. This is a simplification process, where the joint distribution is parametrized in a multiplication of simpler conditional distributions. Directed Acyclic Graphs (DAG) can be used to assist this step. The following Figure 5 represents the proposed DAG of our model. We break the DAG into different abstraction levels for easier comprehension. As can be seen the attitude space is where the attitude variables are. As mentioned

$$
\textit{description}
\begin{cases}
\textit{specification}
\begin{cases}
\textit{Variables:} \\
V, X, \Gamma, E, O, O_L, O_P, G, T, D, V_O, I_L, IM \\
\\
\textit{Decomposition:} \\
P(V, X, \Gamma, E, O, O_L, O_P, G, T, D, V_O, I_L, IM) = \\
= P(V, X, \Gamma, E) P(O, T, G|V) P(O_L, G|X) \\
P(T, O_P|\Gamma) P(O|E) P(V_0, IM|O, O_P) \\
P(V_O, D|O_L) P(I_L|V_O) P(I_L, IM|D) \\
\\
\textit{Formulation:} \\
P(V) P(X) P(\Gamma) P(G|V) P(O_L|X), \\
P(G|X) P(O_P|\Gamma) P(IM|O) P(IM|O_P), \\
P(D|O_L), P(I_L|D), P(IM|D) = \textit{Gaussian} \\
P(E) = \textit{Bernoulli} \\
P(O|V), P(T|V), P(T|\Gamma), P(O|E), P(V_0|O), \\
P(V_0|O_P), P(V_0|O_L), P(I_L|V_O) = \textit{Stochastic Matrix}
\end{cases} \\
\\
\textit{identification: Bernoulli, Binomial parameters based on } \Theta.
\end{cases}
$$

Fig. 6: Bayesian Program description: it enumerates the relevant variables, the joint distribution decomposition and the formulation of the conditional distributions in parametric forms. The identification stage refers to the parameters of the Bernoulli distribution that will be estimated from the experimental data Θ.

previously, the attitude of the end-effector should depend on the available information about the object, the environment and also from the user specified goals. The information about the object may be retrieved from the vision information as well as from the laser scanner information. We introduce a simplification step at an initial stage, where the user identifies the object in the occupancy grid, which at that particular stage resolves into a problem of object verification, in a classification process. Assuming all variables are independent and identically distributed, we obtain the following decomposition:

$$
\begin{aligned}
&P(V, X, \Gamma, E, O, O_L, O_P, G, T, D, V_O, I_L, IM) = \\
&= P(V)P(X)P(\Gamma)P(E)P(O|V)P(T|V)P(G|V)P(O_L|X)P(G|X) \\
&\quad P(T|\Gamma)P(O_P|\Gamma)P(O|E)P(V_O|O)P(IM|O)P(I_L|V_O) \\
&\quad P(V_O|O_P)P(IM|O_P)P(V_O|O_L)P(D|O_L)P(I_L|D)P(IM|D).
\end{aligned}
\tag{2}
$$

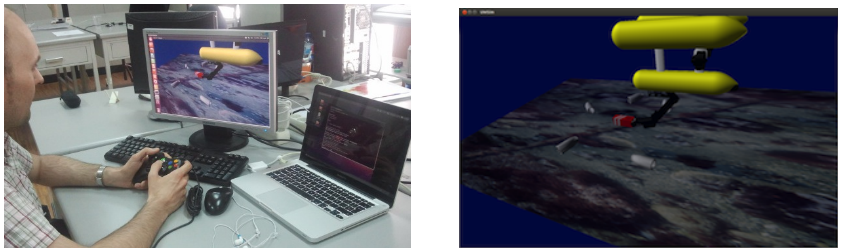
Stage 3::Formulation: This represents the final stage in the specification process. Upon formulating each of the conditional distributions of our decomposition, we have a complete model definition. At this stage, we need to assign parametric or non-parametric probability density functions to each term of the previous step. We will tendentiously assign parametric distribution functions, as these will provide a closed form solution form to our problem, and therefore making the inference process solvable analytically, a desirable property. For variables in which the space is $\mathcal{R}^k$, it is a common and efficient solution to assign **Gaussian** distributions. In case a variable lies in a subspace of $\mathcal{R}$ such as $\mathcal{R}^+$, one can decide to assign Poisson distributions. For discrete variables with two states, we propose either **Binomial** or **Bernoulli** distributions, whose parameters might be user specified or learned from sets of experimental data. Other discrete variables whose space state is bigger than 2, may be represented by m **Multivariate Stochastic Matrices**, in which $(m-1)$ dimensions defined parameters and states and the other represents the probability values. Considering these guidelines, we now present the complete specification of our Bayesian Program in Figure 6, where the decomposition in Equation 2 is compacted for simplicity purposes. The functions in the formulation, are likelihood distributions, which can be learnt incrementally or in batch, from experimental data. In our problem, we will consider the *batch* approach. Likelihoods are probability density functions which are defined in terms of a the first random variable, given the knowledge of the outcome. This constitutes a supervised learning process, and requires a user to give the learning process information about the outcome of the variable argument on the right side of the conditional probability.

Stage 4::Bayesian Inference: Once the model is fully specified we can now inquire for information, based on observable evidence. The questions to our model follow the Bayes Rule formalism, i.e., what is the most likely attitude of the end-effector which will allow to fulfil the desired goal, considering the information about the environment. For simplicity purposes, variable A is multivariate and contains $\{V, X, \Gamma, E\}$, O generically defines $\{O, O_P, O_L\}$ and I is composed of $\{I_L, IM\}$. The term $P(A)$ represents the Prior distribution, which is the estimated attitude of the end-effector before new evidence is taken into account. This distribution is what states the difference between frequentist and subjectivist approaches. It forces regularization for the posterior probability, avoiding overfitting and ensuring that at least one optimal solution exists and it is unique. The terms $P(I|D), P(I|V_O), P(D|O), P(V_O|O), P(O|A), P(G|A)$ and $P(T|A)$ represent a set of likelihood distributions, which reflect how the evidence affects the estimation for A. The normalization term is omitted for mathematical simplification as it does not affect local maximum values for A and its only purpose is ensuring the posterior density integrates to 1. There are various algorithms allowing to perform inference. Perhaps the most popular is the Maximum A Posteriori (MAP). It is a point estimate, which allows finding the maximum value for a given variable based on observable evidence, given by equation 3.

$$\begin{aligned} &\hat{A}_{\mathrm{MAP}}(G, I, V_O, T, O, D) = \\ &arg_A max\ P(I|D)P(I|V_O)P(D|O)P(V_O|O)P(T|A)P(O|A)P(G|A)P(A) \end{aligned} \tag{3}$$

5 Experimental Arrangements

In order to validate the proposed learning paradigm, the real scenario 1 shown in Figure 2 (and described below) has been modelled into the UWSim simulator (see Figure 7b). The real scenario includes an underwater vehicle equipped with a robotic arm. In our initial validation arrangement, we will consider that the vehicle is docked to the tank in a fixed position (fixed base manipulation configuration), so the manipulation training will be done by using the degrees of freedom of the underwater arm. The interaction between the human and the



(a) User interacting with the UWSim. (b) Black-box intervention scenario.

Fig. 7: User interaction with the UWSim simulator using a gamepad controller to train the system. The user gets complete 3D visual information and contact feedback from the observed scene.

virtual robot involves (see Figure 7a): (1) the use of a gamepad, (2) the complete 3D visual information of the scene observed in the computer screen, and also (3) contact feedback, that is, when the robotic arm contacts an object. Right now, this binary contact condition is displayed as a simple indicator on the interface, but latter, we will require force feedback implemented on the gamepad.

5.1 Testbed 1: The Water Tank Scenario

The scenario implemented on the simulator corresponds to the real scenario 1 depicted in Figure 2. It consists on a 2m x 2m x 1.5m water tank, whose floor recreates a real seafloor. The underwater vehicle can move with the aid of four thrusters in the horizontal plane, but can also be docked to the water tank to perform fixed base manipulation. Attached to it is a 4 D.O.F. robotic arm (CSIP Light-weight ARM5E [20]) with the possibility to mount different grippers (like the UJIOne, a sensorized gripper containing tactile sensors based on strain gauges in its end-effector[10]). The vehicle is equipped with an underwater camera (Bowtech 550C-AL) that is placed near the base of the arm and is looking downwards. As previously described in section 1.3, the system is able to perform a 3D reconstruction of the scene by using a laser stripe emitter (Tritech SeaStrip) attached on the forearm of the manipulator [18].

5.2 The HRI with the UWSim Simulator

The UWSim simulator is an open project in, divided in continuous development branches, where the main is the original branch named UWSim, containing the simulator itself. A second branch named QtUWSim has the objective of improving the user interaction with the simulator, integrated with the Qt library (windows manager framework). This new environment, aims to give the user to be able to work with the simulator through buttons, menus and dialogues. Some options will use the the menus to load the scene characteristics, topics, objects

[10] See a reactive tactile sensor test on-line: `http://youtu.be/42ZklVwNaqc`.

and vehicles. The possibility to the user to move the end effector with *interactive markers* [21] also exists. This end effector, defined in a URDF file and loaded into the 3D scene after the user selects the "grasp specification 3D" option from the menu, will be involved by 6 interactive markers (3 translational and 3 rotational). So, the user moves these interactive markers to indicate the end effector position and orientation to reach the target. Another branch integrates the PCL library[11]. A laser is attached to the forearm of the manipulator, reconstructing the target object as a point cloud image. This data may be used also to specify grasping points manually for learning. An algorithm can use this information to find the best grasping points, considering maximum width [18].

5.3 Adapting UWSim for the Proposed Learning Paradigm

As mentioned above, the UWSim simulator do the interface with external control programs through the *Robot Operating System (ROS)*. This means that all the inputs and outputs to and from the simulator are done through ROS topics. A specific ROS node outside the Simulator architecture named `arm_joy_control` allows the user to interact with it using a game controller, by using a specific launch file. Specifically, it allows the user to move the robot arm in joint space (*q1*=Slew, *q2*=Shoulder, *q3*=Elbow, *q4*=JawRotate) and also controlling the Jaw opening (*q5*=JawOpening).

A second, currently under development, node named `arm_joy_cartesian_control` will allow the user to control the robot arm in Cartesian space, using numerical inverse kinematics solvers from the *Orocos KDL library*, to calculate the successive trajectory key-points. By using these values, the algorithm calculates the direct kinematics to verify the user desired point. Due to the few arm degrees of freedom (4 D.O.F.), the algorithm does not check the arm orientation.

Our validation relies on an accurate representation of the real world in our realistic simulator. This property allows us to have a 1 on 1 correspondence between variables on both environments.

5.4 Preliminary results

The first step in the learning process conveys data acquisition from several trials, upon user demonstration by using the simulator. The acquired variables are the joint values $q_i = \{q_0 \ldots q_4\}$, the relative distance to the target object d, binary data indicating when the target object has been picked and collision information (any collision between the arm and/or the end-effector and the target object). The second step is the automatic execution phase P determination, where $P \in \{approach, reach, manipulation\}$ is a random variable which identifies the current phase of a given action. We divide an action into three different phases: the approach phase in which the manipulator identifies the object of interest and starts moving in its direction; in the reach-to-contact the end-effector is required to take the grasp configuration needed to perform the action; the

[11] Point Cloud Library, available: `http://pointclouds.org/`.

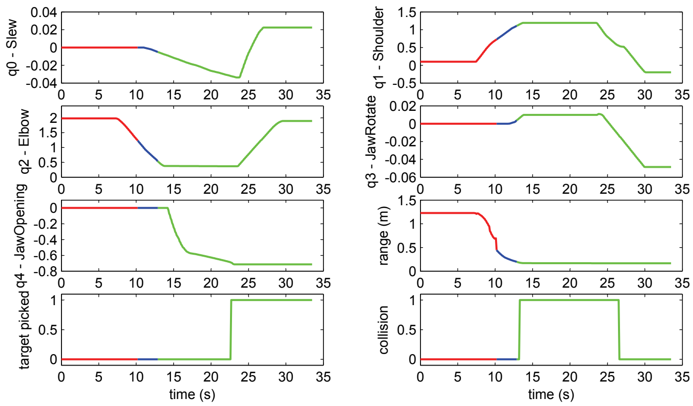


Fig. 8: Preliminar results showing captured data from simulator (q_i, *range*, *target picked*, *collision*) over time and automatic determination of the execution P phase, where $p_1 = \{approach\}$ is represeted in *red*, $p_2 = \{reach\}$ is represeted in *blue* and $p_3 = \{manipulation\}$ is represeted in *green*.

manipulation stage happens when the end-effector contacts the object to perform a specific task. The captured data from simulator (q_i, *range*, *target picked*, *collision*) and the distance-based phase P selection can be seen in Figure 8.

6 Conclusions and Future Work

After very successful research achievements through previous projects, like TRIDENT or RAUVI, following a semi-autonomous strategy, we are trying to increase now the autonomy levels, under GRASPER project context, by means of learning. This new approach is supported by the ongoing cooperation between UJI (Spain) and ISR (Portugal). In particular, upon successfully implementing the learning/classification framework, we will achieve two main goals. The first, is to have a knowledge representation, in which probability density functions are used to efficiently map sensed information into manipulator action parameters, dependent on specific underwater mission goals. The second objective is having the manipulator accessing the knowledge information in a decision making/classification framework and, autonomously, decide the correct course of action into solving a given user-defined task. The estimated optimal solution is based on the information it is capable of acquiring from itself and the environment. This process is intended to increase AUV autonomy capabilities, while simultaneously reducing, or even assisting human intervention. The model will include additional information, the state of the AUV in the representation of the model, to keep the AUV in an appropriate position while the manipulation is being executed. To maintain the vehicle position on top of the object to manipulate, we will use our already developed visual tracking method [22].

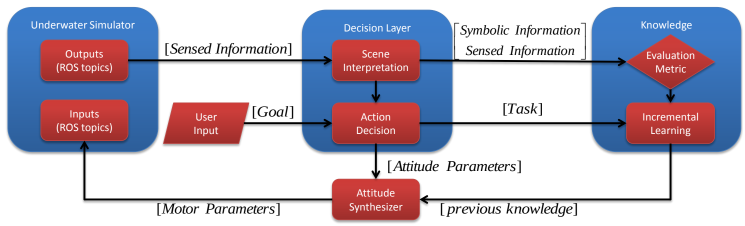


Fig. 9: Execution and Incremental Learning Block Diagram.

The final goal of the project is to have an AUV (Autonomous Underwater Vehicle) performing an underwater task autonomously, with minimum human intervention. To that purpose, autonomous execution will be used to incrementally update existing knowledge with new trials (Figure 9). While executing, the acquired information will be interpreted, and forwarded to a module which will decide whether the trial will be added to memory for future interventions, or not. This decision will be based on a comparison between expected and real mission outcomes. This continuous process, the generalized action information is synthesized into low-level control primitives acting on the manipulator itself.

Acknowledgment

This research was partly supported by Spanish Ministry of Research and Innovation DPI2011-27977-C03 (TRITON Project), by Foundation Caixa Castelló-Bancaixa PI.1B2011-17, by Universitat Jaume I under grant E-2013-33, by the FCT - Portuguese Foundation for Science and Technology, Grant # 65935/2009, by the Institute of Systems and Robotics from University of Coimbra, Portugal and Khalifa University, Abu Dhabi, UAE.

References

1. Prats, M., Pérez, J., Fernández, J., Sanz, P.: An open source tool for simulation and supervision of underwater intervention missions. In: Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on. (2012) 2577–2582
2. Marani, G., Choi, S.K., Yuh, J.: Underwater autonomous manipulation for intervention missions AUVs. Ocean Engineering **36** (2009) 15–23
3. Sanz, P.J., Ridao, P., Oliver, G., Casalino, G., Insaurralde, C., Silvestre, C., Melchiorri, C., Turetta, A.: TRIDENT: Recent improvements about autonomous underwater intervention missions. In: 3rd IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles (NGCUV 2012), Porto, Portugal (04 2012)
4. Sanz, P., Prats, M., Ridao, P., Ribas, D., Oliver, G., Ortiz, A.: Recent progress in the RAUVI project: A Reconfigurable Autonomous Underwater Vehicle for Intervention. In: ELMAR, 2010 PROCEEDINGS. (2010) 471–474
5. Faria, D., Martins, R., Lobo, J., Dias, J.: Extracting data from human manipulation of objects towards improving autonomous robotic grasping. Robotics and Autonomous Syst., Elsevier: Sp. Issue on Autonomous Grasping **60** (2012) 396–410

6. Calinon, S., Guenter, F., Billard, A.: On learning, representing and generalizing a task in a humanoid robot. IEEE Transactions on Systems, Man and Cybernetics, Part B, Special issue on robot learning by observation, demonstration and imitation **37** (2007) 286–298
7. Kondo, M., Ueda, J., Ogasawara, T.: Recognition of in-hand manipulation using contact state transition for multifingered robot hand control. Robotics and Autonomous Systems **56**(1) (2008) 66 – 81
8. Bernardin, K., Ogawara, K., Ikeuchi, K., Dillmann, R.: A sensor fusion approach for recognizing continuous human grasping sequences using hidden markov models. Trans. Rob. **21**(1) (February 2005) 47–57
9. Krger, V., Herzog, D., Baby, S., Ude, A., Kragic, D.: Learning actions from observations. IEEE Robot. Automat. Mag. **17**(2) (2010) 30–43
10. Bekiroglu, Y., Laaksonen, J., Jrgensen, J.A., Kyrki, V., Kragic, D.: Assessing grasp stability based on learning and haptic data. IEEE Transactions on Robotics **27**(3) (2011) 616–629
11. Lin, Y., Ren, S., Clevenger, M., 0004, Y.S.: Learning grasping force from demonstration. In: Int. Conf. on Robotics and Automation ICRA, IEEE (2012) 1526–1531
12. Lopes, M., Melo, F., Montesano, L.: Active learning for reward estimation in inverse reinforcement learning. In: Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part II. ECML PKDD '09, Berlin, Heidelberg, Springer-Verlag (2009) 31–46
13. Jetchev, N., Toussaint, M.: Task space retrieval using inverse feedback control. In: ICML. (2011) 449–456
14. Ratliff, N.D., Bagnell, J.A., Zinkevich, M.A.: Maximum margin planning. In: Proceedings of the 23rd international conference on Machine learning. ICML '06, New York, NY, USA, ACM (2006) 729–736
15. Guizzo, E.: Humanoid robot justin learning to fix satellites, IEEE Spectrum. Availab: `http://spectrum.ieee.org/automaton/robotics/industrial-robots/humanoid-robot-justin-learning-to-fix-satellites`
16. Prats, M., García, J., Fernández, J., Marín, R., Sanz, P.: Advances in the specification and execution of underwater autonomous manipulation tasks. In: OCEANS, 2011 IEEE - Spain. (2011) 1–5
17. Carrera, A., Ahmadzadeh, S.R., Ajoudani, A., Kormushev, P., Carreras, M., Caldwell, D.G.: Towards Autonomous Robotic Valve Turning. Journal of Cybernetics and Information Technologies (CIT) **12**(3) (2012) 17–26
18. Prats, M., Fernández, J., Sanz, P.: Combining template tracking and laser peak detection for 3D reconstruction and grasping in underwater environments. In: Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on. (2012) 106–112
19. Bessiere, P., Ahuactzin, J.M., Mekhnacha, K., Mazer, E.: Bayesian Programming. Chapman and Hall/CRC (2013)
20. Fernández, J.J., Prats, M., García, J.C., Marín, R., Peñalver, A.: Manipulation in the seabed: A new underwater robot arm for shallow water intervention. In: Embedded Systems, Computational Intelligence and Telematics in Control. (2012)
21. García, J.C., Peñalver, A., Prats, M., Sanz, P.J.: Recent progress in HRI for underwater robotic intervention. In: 2013 IEEE International Conference on Robotics and Automation (ICRA'13), Karlsruhe, Germany (05 2013)
22. Prats, M., García, J.C., Fernández, J.J., Marín, R., Sanz, P.J.: Towards Specification, Planning and Sensor-Based Control of Autonomous Underwater Intervention. In: Proceedings of the 18th IFAC World Congress, Milano, Italy (2011) 10361–10366