Learning by demonstration applied to underwater intervention

Arnau CARRERA [a], Narcís PALOMERAS [a], Natàlia HURTÓS [a], Petar KORMUSHEV [2] and Marc CARRERAS [a]

[a] *Computer Vision and Robotics Group (VICOROB), University of Girona* [1]
[b] *Department of Advanced Robotics, Istituto Italiano di Tecnologia* [2]

Abstract. Performing subsea intervention tasks is a challenge due to the complexities of the underwater domain. We propose to use a learning by demonstraition algorithm to intuitively teach an intervention autonomous underwater vehicle (I-AUV) how to perform a given task. Taking as an input few operator demonstrations, the algorithm generalizes the task into a model and simultaneously controls the vehicle and the manipulator (using 8 degrees of freedom) to reproduce the task. A complete framework has been implemented in order to integrate the LbD algorithm with the different onboard sensors and actuators. A valve turning intervention task is used to validate the full framework through real experiments conducted in a water tank.

Keywords. Learning by Demonstration (LbD), Dynamic Movement Primitives (DMP), Autonomous Underwater Vehicle (AUV), Underwater Intervention

Introduction

The interest for having human operators interacting and working side by side with robots has been increasingly growing during the last decade. Two key elements are necessary to achieve this goal. The first one is to settle a safety work space in which human operators can not be hurt by the robots. The second element, which is related to the focus of this paper, concerns about how an operator can teach a new task to a robot in a natural way.

A recent example of this progress in the industrial environment is the Baxter[1] robot. It is an industrial robot with two arms that can share the same work area with human operators. Moreover, Baxter is programed by means of an operator moving its compliant arms and recording the desired waypoints to perform a particular task.

A more general way to teach a new task to a robot is using a learning by demonstration (LbD) algorithm [2]. This algorithm allows a robot to learn a new task through a set of demonstrations. Distinctly to the Baxter's way of teaching, where only some points

[1]University of Girona, 17071 Girona, Spain. E-mail: arnau.carrera, narcis.palomeras, marc.carreras at udg.edu and nhurtos at eia.udg.edu

[2]Istituto Italiano di Tecnologia, via Morego, 30, 16163 Genova, Italy. E-mail: petar.kormushev at iit.it
This research was sponsored by the Spanish government (COMAROB Project, DPI2011-27977-C03-02) and the PANDORA EU FP7-Project under the Grant agreement FP7-ICT-2011-7-288273. We are grateful for this support.

are stored, the LbD algorithm records several demonstration trajectories and then, they are used to generate a representative model of the task.

Our aim in this paper is to apply a LbD technique to an intervention autonomous underwater vehicles (I-AUV) to enable it to learn an intervention task. This type of vehicles are designed to explore the underwater world autonomously and interact with it using a manipulator. By using LbD we will provide the possibility to easily use the I-AUV for different tasks. In this way, every new task just requires an operator to perform demonstrations thus avoiding the implementation of new code every time.

Some research projects have begun to demonstrate similar capabilities with I-AUVs, although none of them uses machine learning but classical manipulation theory. The SAUVIM project [3] proposed a system to recover objects of the seafloor. In the TRIDENT project [4] a system to search and recover objects with a light I-AUV was presented. The TRITON project [5] shows some manipulation with an I-AUV docked in a sub-sea panel. The work presented here is conducted in the context of the PANDORA project, and for the first time machine learning, in form of LbD, is applied on an I-AUV that has the goal of performing a valve turning task in free floating mode.

The application of LbD techniques in the underwater domain presents several added complications due to water perturbations (i.e. current, waves), reduced visibility, difficulties in understanding the scene and high sensorial uncertainty for navigation and perception. For this reason, the LbD implementation presented here has required the development of a complete framework to successfully integrate the algorithms.

The proposed framework learns 8 degrees of freedom (DoF) to control the trajectory of an AUV and its manipulator simultaneously. To perceive the environment, the framework uses the vehicle cameras and also a force and torque (F/T) sensor to detect the contact between the manipulator and the target. Information from all these sensors is acquired whilst a pilot is performing the intervention task to be learned. Then from a set of demonstrations, the proposed LbD algorithm generalizes , a control policy able to accomplish the intervention task with the same performance than the human operator.

To validate the proposed approach we present experiments in the context of a valve turning intervention task. An AUV equipped with a manipulator, two cameras, a F/T sensor and a haptic device is set to perform the turning of a valve placed on an sub-sea panel in free floating mode. Results show good performance when attempting the task both in normal conditions and under external perturbations.

The rest of this paper is organized as follows. Section 1 overviews related work on LbD for robotics and describes the LbD algorithm that has been used. Section 2 describes the vehicle used to perform the intervention task as well as the software architecture. Results obtained from the valve turning test scenario are presented and analyzed in Section 3. Section 4 summarizes, and concludes the work.

1. Learning by Demonstration

LbD is a machine learning technique designed to transfer the knowledge from an expert to a machine. This type of algorithm follows three sequential phases: first, a set of *demonstrations* of the task are recorded; second, the algorithm *learns* by generalising all demonstrations and creating a model; finally, the algorithm loads the model and uses it to *reproduce* the task.

There are mainly two methods to transfer the knowledge: *Imitation*, where the teacher performs the task by itself and the robot extracts the information; and *Demonstration*, where the robot is used to perform the task by tele-operation or by a kinesthetic teaching, where the robot is moved by the teacher. The learned controllers can generate trajectories adaptable to the current robot state.

1.1. LbD related work

Several LbD algorithms have been proposed depending on the application requirements. D.R.Faria [6] proposed to learn the manipulation and grasping of an object using geometry, based on the position of fingers and their pressure, representing them with a probabilistic volumetric model. Calinon [7] proposed to represent trajectories using a Gaussian mixture model (GMM). This representation was extended by Kruger [8] using Incremental GMM to automatically set the number of Gaussians. Furthermore, Calinon [9] used different types of parametrized regressions to adjust the trajectory learnt during the demonstrations. Similarly to the GMM, a hidden Markov model (HMM) [10] can be used to represent a trajectory and parametrized [11]. A different option to encode the trajectory is using dynamic movement primitives (DMP) [12], which can be extended for working in closed loop [13]. Moreover, forces exert along the trajectory can be learned by an extended DMP[14].

1.2. Dynamic movement primitives

Considering the context of this work, we have chosen to use DMP as the base of our learning framework. The main motivation is the fact that it dynamically generates the trajectories during the reproduction, which makes the approach robust to external perturbations. Also, the flexibility and simplicity of the representation allows the adaptation of the algorithm to specific requirements, as it will be described in Section 2.2.4.

DMP is an algorithm where the learned skill is encapsulated in a superposition of basis motion fields (see Figure 1). The method used in this paper is an extension of DMP proposed by Kormushev [14].

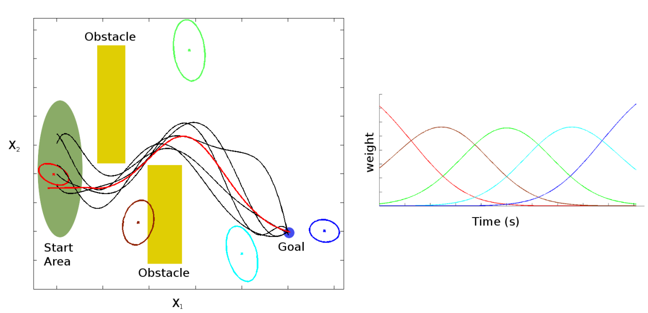


Figure 1. Left figure shows a set of 2D demonstrated trajectories (black) and one reproduction (red). In this case, the demonstrated trajectory has to pass between the two depicted obstacles. On the right, the h function is represented. The encoding of the trajectories using a DMP algorithm has been done using 5 Gaussians adequately weighted over time.

To better understand this encoding, we can imagine a mass attached to different damped strings. These strings attract the mass changing their forces along the time of the experiment, moving the mass following the desired trajectory.

To generate the superposition each attractor has an associated weight which changes along the time defined by the $h_i(t)$ function (1). The weight of each attractor is represented with a Gaussian, whose centers μ_i^T are equally distributed in time, and whose variance parameters $\Sigma_i^T = total\_time/K$ are set to a constant value inversely proportional to the number of Gaussians (K).

$$h_i(t) = \frac{\mathcal{N}(t; \mu_i^T, \Sigma_i^T)}{\sum_{k=1}^{K} \mathcal{N}(t; \mu_k^T, \Sigma_k^T)}, \tag{1}$$

Instead of using the real time a decay term is used, to obtain a time invariant model:

$$t = \frac{ln(s)}{\alpha}, \text{ where } s \text{ is a canonical system : } \dot{s} = s - \alpha s, \tag{2}$$

and the α value is selected by the user depending on the duration of the demonstrated task.

The number of attractors is preselected by the user and represented using Gaussians, depending on the complexity of the task. The position of the attractor is the center of the Gaussian (μ_i^x) and the stiffness (matrix K_i^P) is represented by the covariance. The values are learned from the observed data through least-squares regressions. All the data from the demonstrations is concatenated in a matrix $Y = [\ddot{x}\frac{1}{K^P} + \dot{x}\frac{K^V}{K^P} + x]$, where x, $\dot{x}$ and $\ddot{x}$ are the position, velocity and acceleration recorded at each time instant of the demonstrations. Also the weights at each time instant are concatenated to obtain matrix H. With these two matrices, the linear equation $Y = H\mu^x$ can be written . The least-square solution to estimate the attractor center is then given by $\mu^x = H^\dagger Y$, where $H^\dagger = (H^T H)^{-1} H^T$ is the pseudo-inverse of H.

The user needs to define a minimum K_{min}^P, and maximum K_{max}^P to define the limits of the stiffness and to estimate the damping as follows:

$$K^P = K_{min}^P + \frac{K_{max}^P - K_{min}^P}{2} \text{ , } K^V = 2\sqrt{K^P}. \tag{3}$$

To take into account variability and correlation along the movement and among the different demonstrations, the residual errors of the least-squares estimations are computed in the form of covariance matrices, for each Gaussian ($i \in \{1, ...K\}$).

$$\Sigma_i^X = \frac{1}{N} \sum_{j=1}^{N} (Y'_{j,i} - \bar{Y}_i{}')(Y'_{j,i} - \bar{Y}_i{}')^T, \qquad \forall_i \in \{1, ...K\}, \tag{4}$$

where:

$$Y'_{j,i} = H_{j,i}(Y_j - \mu_i^x). \tag{5}$$

In Equation 4, the $\bar{Y}_i{}'$ is the mean of Y_i' over the N datapoints.

Finally, the residual terms of the regression process are used to estimate the K_i^P through the eigen components decomposition.

$$K_i^P = V_i D_i V_i^{-1}, \tag{6}$$

where:

$$D_i = k^P_{min} + (k^P_{max} - k^P_{min})\frac{\lambda_i - \lambda_{min}}{\lambda_{max} - \lambda_{min}}. \tag{7}$$

In the Equation above, the λ_i and the V_i are the concatenated eigenvalues and eigenvector fo the inverse covariance matrix $(\Sigma_i^x)^{-1}$. The basic idea is to determine a stiffness matrix proportional to the inverse of the observed covariance.

To sum up, the model for the task will be composed by: the k_i^P matrices and μ_i^x centers representing the Gaussians; $h_i(t)$ representing the influence of each matrix functions; K^V representing the damping; and α, which is assigned according to the duration of the sample. Figure 1 shows a simple example where the learned data is represented.

Finally, to reproduce the learned skill, the desired acceleration is generated with

$$\hat{\ddot{x}} = \sum_{i=1}^{K} h_i(t)[K_i^P(\mu_i^X - x) - K^v \dot{x}], \tag{8}$$

where x and $\dot{x}$ are the current position and velocity.

2. Intervention Framework

The intervention framework can be divided in two different parts. First, the hardware components, namely the I-AUV and the manipulator. Second, the software architecture which interprets the information gathered by the sensors, sends commands to the actuators, and learns the demonstrated task controlling both the AUV and the manipulator.

2.1. Hardware components

The Girona 500 I-AUV [15] is a compact and lightweight AUV with hovering capabilities which can fulfill the particular needs of a wide diversity of applications by means of mission-specific payloads and a reconfigurable propulsion system. For the purpose of this paper, the propulsion system is configured with 5 thrusters to control 4 DoFs (surge, sway, heave and yaw). To perform intervention tasks, the Girona 500 I-AUV is equipped with an under-actuated manipulator (see Figure 2), with 4 DoFs (slew, elbow, elevation and roll) and custom end-effector.

The custom end-effector is composed of three different parts (see Figure 2). The first one is a compliant *passive gripper* to absorb small impacts. Since we aim to demonstrate a valve turning task, the gripper has been designed with a V-shape in order to easily drive the handle of a T-bar valve to the end-effector center. The second element consists of a *camera in hand* which has been installed in the center of the gripper to provide a visual feedback of what the end-effector is manipulating. This camera has been placed to prevent the occlusion of the vehicle's camera by the manipulator during the demonstration of the intervention. Finally, a *F/T sensor*, provides information about the quality of the grasping and the necessary torque to turn the valve during the manipulation.

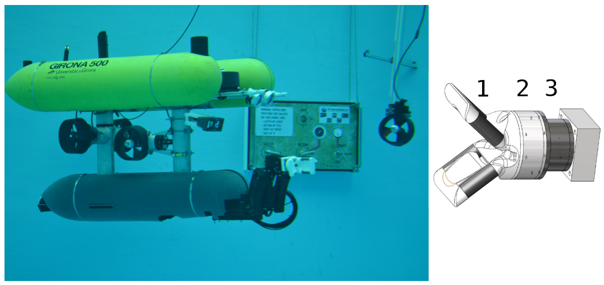


Figure 2. On the right, the Girona 500 I-AUV in a water tank with a mock up of a sub-sea panel at the background and a Sea-eye thruster (on the right) used to introduce perturbations during the manipulation. On the left, a 3D model of the customized end-effector, in which the three blocks can be distinguished: 1 passive gripper, 2 camera in-hand and 3 F/T sensor.

2.2. *Software Architecture*

The software architecture for intervention is composed of several modules which are organized in several layers, see Figure 3. Starting from the bottom, the first layer contains all the sensors and actuators. Next layer has all perception systems to process sensor information, such as the localization module and the perception systems that process cameras and F/T sensor data. On top of it, the AUV and manipulator velocity controllers are in charge of following the set points of the LbD architecture. Finally, in the top level layer, the LbD architecture is in charge of acquiring data from demonstrations (phase 1), learning the model (phase 2) and reproducing the task by generating velocity setpoints (phase 3).

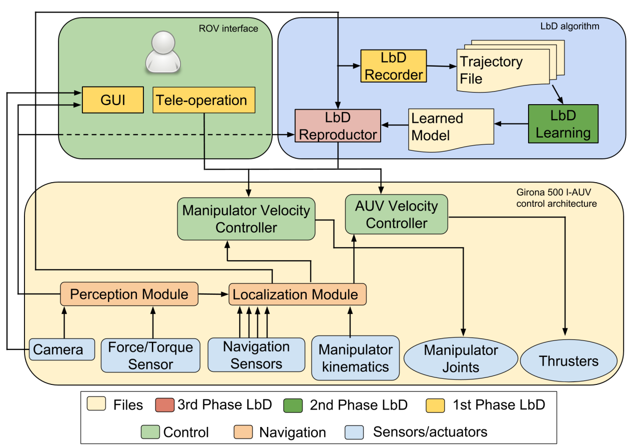


Figure 3. Diagram of software architecture showing the LbD architecture and its connection to the AUV control architecture.

2.2.1. *Localization and tracking of elements*

A simultaneous localization and mapping algorithm, based on an extended Kalman filter with simultaneous localization and mapping (EKF-SLAM) is used to obtain a robust

AUV navigation [16]. The EKF-SLAM system combines different navigation sensors and merges their information through a constant motion model to obtain an estimation of its position.

Furthermore, the EKF-SLAM can include information about landmarks to improve the AUV navigation. In our proposal, the pose of the target of interest (the goal valve) is included as a landmark into the system.

2.2.2. *Perception module*

AUVs have different sensors to perceive the environment. To identify the target of interest, a vision-based algorithm analyzes the gathered images and compares them with an *a priori* target template. With this information, the main system is able to obtain the position of the target with respect to the AUV.

Additionally, during the intervention task, the F/T sensor mounted in the end-effector is used to obtain contact information.

2.2.3. *Control system*

In general, AUV and manipulator controllers accept velocity or pose requests. Our strategy uses two independent velocity controllers: one dealing with the 4 DoF of the AUV and another controlling the 4 DoF of the manipulator.

The AUV velocity controller computes the force and torque to be generated to reach the desired velocity. The force output is a combination of an standard 4 DoF proportional-integral-dervative (PID) controller and an open-loop controller based on a model.

The low-level controller for the manipulator controls the velocity for each joint ($\dot{q} \in \Re^4$) in order to reach the desired velocity of the end-effector in the Cartesian space. To this end, the desired velocity is transformed to an increment in Cartesian space ($\vec{x}$), and using the pseudo-inverse Jacobian ($J^\dagger$) of the manipulator, $\dot{q}$ is obtained as follows: $\dot{q} = J^\dagger \dot{x}$.

2.2.4. *LbD architecture for underwater intervention*

The LbD approach introduced in Section 1.2 has been tailored to the complexities of the underwater environment and the need of a tight cooperation between the vehicle and the manipulator. The implemented LbD architecture is divided in 3 phases that we detail in the following lines, describing also the particular modifications that have been performed at each stage.

- **Demonstration:** The operator performs the task by teleo-perating the I-AUV, using the feedback from the onboard camera and the F/T sensor. Knowing the target pose, the manipulator and AUV poses are transformed with respect to a frame located at the target's center (i.e the center of the valve). The tele-operation phase is paramount for the proper functioning of the system, as the quality of the learning will depend on the quality of the demonstrations. Toward that end, we propose to use a haptic device with force feedback to control the vehicle and the manipulator. Furthermore, for a better feedback while performing the demonstration, the operator uses a Graphical User Interface (GUI) to watch the vehicle's camera as well as a 3D representation of the AUV pose.

- **Learning:** After several demonstrations, the LbD algorithm generates a model of Gaussians attractors defined by the position of its centers and the stiffness matrices, using a modified version of the DMP algorithm explained in Section 1.2. The DMP algorithm has been adapted to allow an efficient control of both, the vehicle and the manipulator. This implied the addition of the vehicle's *yaw* orientation, and the end-effector's position in the Cartesian space *x,y,z* and *roll* orientation. Hence, the modified DMP controls 8 DoF instead of 3 DoF. To take in consideration the relation between the movement of the vehicle and the manipulator, the requested manipulator velocities are computed by subtracting the end-effector requested velocities from the AUV requested velocities. Besides, the DMP has been modified to integrate a finalization condition using the information of the F/T sensor, to detect the contact with the valve and thus the accomplishment of the trajectory.
- **Reproduction:** The LbD Reproductor loads the task model and using the same inputs as in the demonstration phase, generates the AUV and manipulator requested velocities to perform the task.

3. Results

To validate the proposed LbD framework a valve turning intervention task has been proposed. The task is divided in 3 steps: approaching the panel, moving the manipulator to an appropriate configuration to grasp the valve, and finally turning the valve. Results are presented by following the 3 phases of the LbD algorithm.

3.1. Demonstration

In the demonstration phase, the AUV is placed approximately at 4m from the panel. The operator, using the haptic device, drives the vehicle to a position where the intervention can start (around 1.5 to 2 m from the panel). When Girona 500 I-AUV reaches this position the operator moves the manipulator to obtain a desired configuration to grasp the valve. Finally the valve is turned.

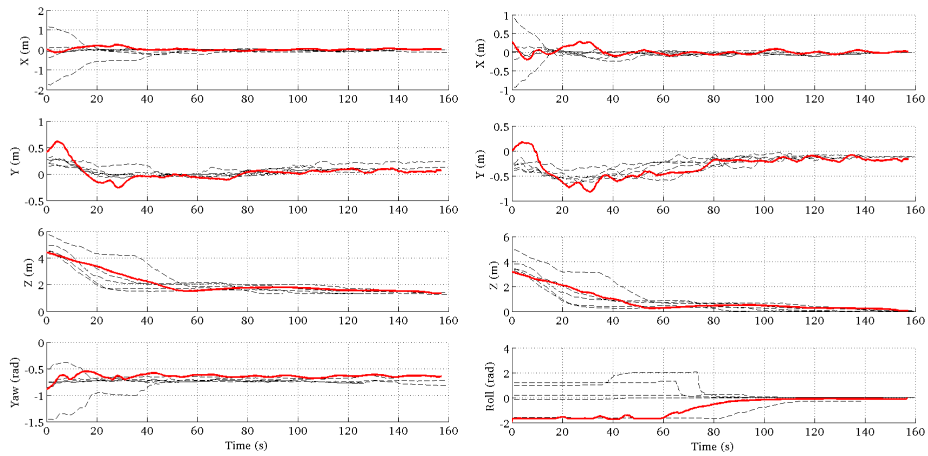
3.2. Learning

The learning algorithm uses the previously demonstrated trajectories from the beginning of the approximation until the valve is grasped (steps 1 and 2). The last step, turning the valve, is not part of the learned model. For that, we use a controller that given a maximum torque and a desired angle turns the valve.

In the proposed experiment, the learning algorithm used 20 attractors points and 5 demonstrations to learn the task.

3.3. Reproduction

Figure 4 shows the 8 DoF learned in independent plots comparing the 5 demonstrations against an autonomously performed trajectory. As it can be observed, the LbD reproduction follows the desired trajectory producing smoother movements than the human operator. This can be easily appreciated in the Z axis. Also the LbD can generate more strict or flexible constraints, depending on the variability of the demonstrated trajectories. For example, in the Roll graph, the reproduced trajectory is flexible until the second 80, and then it strictly follows the reproduced trajectories after that.



(a) AUV

(b) End-effector

Figure 4. Demonstrated trajectories (black and slashed line) and autonomous trajectory (red and blood line) for the valve grasping task. All trajectories are represented in the frame of the target valve. Each plot shows a single DoF for the manipulator and the end-effector. In the demonstration and the reproduction the time used is the real time of the experiments which in the reproduction is the equivalent to the one generated by the canonical system.

Regarding the overall performance, 13 of the 16 reproductions have been successful. Most of the failures are caused by errors in the alignment between the valve and the end-effector. This errors are introduced in the vision-based system to detect the valves orientations and should be further investigated.

$\bar{\sigma}_{X_{AUV}}$	$\bar{\sigma}_{Y_{AUV}}$	$\bar{\sigma}_{Z_{AUV}}$	$\bar{\sigma}_{Yaw_{AUV}}$	$\bar{\sigma}_{X_{EE}}$	$\bar{\sigma}_{Y_{EE}}$	$\bar{\sigma}_{Z_{EE}}$	$\bar{\sigma}_{Roll_{EE}}$
0.0713 m	0.0812 m	0.2071 m	0.0633 rad	0.0984 m	0.1056 m	0.2198 m	0.925 rad

Table 1. The average of the standard deviation along the completed reproduction of the 13 successful reproductions.

To show the similarity of all the successful reproductions have been computed the standard deviation between them. Table 1 shows the average of the standard deviation obtained for each axis during the reproduction time. All the axes have obtained a small deviation proving the similarities between them. The roll of the end effector has the biggest deviation due to the flexibility in the learned model. On the other hand, the X and Y axes of the AUV and the end-effector have a similar small values while the Z axis has bigger values, because of the difficulty to stabilize the AUV after modifying the depth.

4. Conclusions

This paper has presented, for the first time, the use of machine learning in the context of an I-AUV task showing real experiments in a controlled water tank. We have implemented a LbD algorithm integrated in a full framework that allows to intuitively teach a task using few operator demonstrations. The core of the implemented LbD consists in a DMP algorithm that has been tailored to control simultaneously the vehicle and the manipulator (8 DoF). In this way, we achieve a tight cooperation between both components and greater stability in the performed trajectories. The validation experiments have been performed with the Girona 500 I-AUV equipped with a manipulator and a custom end-effector in the context of a valve turning intervention task. The results of the experiments have proved the suitability of the proposed method obtaining similar or better results than a human operator.

Future work will focus on dealing with high perturbations and detecting failures during the evolution of the intervention, trying to adapt the strategy to the new conditions or aborting the task. Also, F/T sensor data will be included in the DMP algorithm to learn/reproduce how the human operator interacts with the target.

References

[1] (2013) Rethink robotics. [Online]. Available: http://www.rethinkrobotics.com/products/baxter

[2] B. Argall, S. Chernova, M. M. Veloso, and B. Browning, "A survey of robot learning from demonstration." *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469–483, 2009.

[3] G. Marani and S. Yuh, "Underwater autonomous manipulation for intervention missions AUVs," *Oceans Engineering*, vol. 36, no. 1, pp. 15–23, 2009.

[4] M. Prats, J. Garcıa, J. Fernández, R. Marın, and P. Sanz, "Towards specification, planning and sensor-based control of autonomous underwater intervention," *IFAC 2011*, 2011.

[5] J. Fernandez, M. Prats, P. Sanz, J. Garcia, R. Marin, M. Robinson, D. Ribas, and P. Ridao, "Grasping for the Seabed: Developing a New Underwater Robot Arm for Shallow-Water Intervention," *Robotics Automation Magazine, IEEE*, vol. 20, no. 4, pp. 121–130, Dec 2013.

[6] D. R. Faria, R. Martins, J. Lobo, and J. Dias, "Extracting data from human manipulation of objects towards improving autonomous robotic grasping," *Robotics and Autonomous Systems*, vol. 60, no. 3, pp. 396–410, 2012.

[7] S. Calinon, F. D'halluin, E. L. Sauser, D. G. Caldwell, and A. Billard, "Learning and Reproduction of Gestures by Imitation." *IEEE Robot. Automat. Mag.*, vol. 17, no. 2, pp. 44–54, 2010.

[8] V. Krüger, V. Tikhanoff, L. Natale, and G. Sandini, "Imitation learning of non-linear point-to-point robot motions using dirichlet processes." in *ICRA*. IEEE, 2012, pp. 2029–2034.

[9] S. Calinon, T. Alizadeh, and D. G. Caldwell, "Improving extrapolation capability of task-parameterized movement models," in *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, Tokyo, Japan, November 2013.

[10] G. Hovland, P. Sikka, and B. McCarragher, "Skill acquisition from human demonstration using a hidden Markov model," in *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, vol. 3, 1996, pp. 2706–2711 vol.3.

[11] V. Kruger, D. Herzog, S. Baby, A. Ude, and D. Kragic, "Learning Actions from Observations," *Robotics Automation Magazine, IEEE*, vol. 17, no. 2, pp. 30–43, 2010.

[12] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal, "Learning and generalization of motor skills by learning from demonstration." in *ICRA*. IEEE, 2009, pp. 763–768.

[13] M. Parlaktuna, D. Tunaoglu, E. Sahin, and E. Ugur, "Closed-loop primitives: A method to generate and recognize reaching actions from demonstration." in *ICRA*. IEEE, 2012, pp. 2015–2020.

[14] P. Kormushev, S. Calinon, and D. G. Caldwell, "Imitation Learning of Positional and Force Skills Demonstrated via Kinesthetic Teaching and Haptic Input." *Advanced Robotics*, vol. 25, no. 5, pp. 581–603, 2011.

[15] D. Ribas, N. Palomeras, P. Ridao, M. Carreras, and A. Mallios, "Girona 500 AUV: From Survey to Intervention," *Mechatronics, IEEE/ASME Transactions on*, vol. 17, no. 1, pp. 46–53, feb. 2012.

[16] N. Palomeras, S. Nagappa, D. Ribas, N. Gracias, and M. Carreras, "Vision-based localization and mapping system for AUV intervention," in *OCEANS'13 MTS/IEEE*, 2013.