An Intervention-AUV learns how to perform an underwater valve turning

Arnau Carrera[1], Narcís Palomeras[2], David Ribas[3], Petar Kormushev[4] and Marc Carreras[5]
[1,2,3,5] Computer Vision and Robotics Institute (VICOROB), University of Girona, 17071 Girona, Spain
[4] Department of Advanced Robotics, Istituto Italiano di Tecnologia, via Morego, 30, 16163 Genova, Italy
arnau.carrera[1], narcis.palomeras[2], dribas[3], marc.carreras[5] at udg.edu
petar.kormushev[4] at iit.it

***Abstract*—Intervention autonomous underwater vehicles (I-AUVs) are a promising platform to perform intervention task in underwater environments, replacing current methods like remotely operate underwater vehicles (ROVs) and manned submersibles that are more expensive. This article proposes a complete system including all the necessary elements to perform a valve turning task using an I-AUV. The knowledge of an operator to perform the task is transmitted to an I-AUV by a learning by demonstration (LbD) algorithm. The algorithm learns the trajectory of the vehicle and the end-effector to accomplish the valve turning. The method has shown its feasibility in a controlled environment repeating the learned task with different valves and configurations.**

I. Introduction

Nowadays the interest in autonomous underwater interventions has increased in the marine industry. Current methods to perform interventions with submersibles and remotely operated vehicles (ROVs) require expensive vessels and expert operators taking care of the equipment during the intervention. Instead, the deployment of light intervention autonomous underwater vehicles (I-AUVs) operated from cheap vessels and with a reduced team of operators could reduce the operation cost. For this reason, some research projects have been focused in building new systems to solve the different kinds of intervention autonomously. The project *ALIVE (2001-04)* [1] developed an underwater intervention AUV. The AUV autonomously docked to a subsea infrastructure using an hydraulic gripper and once attached it manipulated a valve panel. Later, the *SAUVIM (2009-2011)* [2] was the first project to perform a free floating manipulation of an underwater object using an AUV. Recently, *TRIDENT (2010-2012)* [3] developed a multipurpose object search and recovery strategy using an I-AUV. Nowadays, *TRITON (2012-2014)* is developing an intervention task including docking to a custom subsea panel and fixed-based manipulation for valve turning, hot stab connection, and free floating manipulation for camera dome de-fouling. Furthermore, *PANDORA (2012-2014)* aim is to work towards a persistent autonomy for AUVs. This means a robot capable to react against small failures without assistance, making possible long term inspection and intervention missions.

This work is included in the framework of the PANDORA project, presenting a new system to improve the capabilities of an I-AUV to perform a free-floating manipulation of a valve placed in a subsea panel. The Girona 500 AUV is equipped with a manipulator and a customized end-effector (see Fig. 1), which includes a camera and a force torque (F/T) sensor. Furthermore, a control architecture has been developed to integrate all the data gathered by multiple sensors to obtain a robust state of the valve panel.

To achieve the desired manipulation skill with an I-AUV the authors have modified an existent learning by demonstration (LbD) algorithm to extract the expertise of a ROV pilot. Moreover, a complete system has been designed to perform efficiently the specific task. Hardware elements like the end-effector have been conceived and built. Perception and navigation systems to control the vehicle pose and track the subsea panel and valves have been developed. And finally, the learning algorithm interface has been adapted so an operator can tele-operate the AUV to perform the learning demonstration.

The viability of the method has been tested in a water tank using a mock-up subsea panel with different valves.

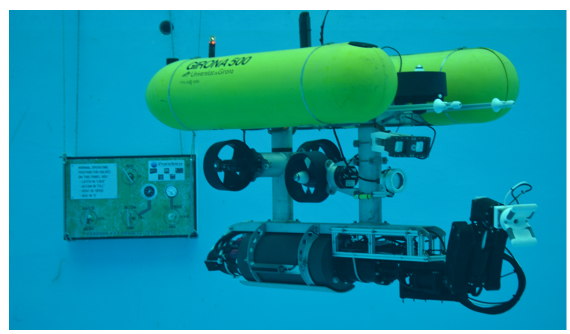

Fig. 1. The image shows the Girona 500 AUV in the water tank, equipped with the ECA CSIP Manipulator and a customized end-effector. Behind the AUV the subsea panel can be seen.

The rest of this paper is as follows. Section II overviews related work on the LbD architecture and explains the proposed algorithm . Section III presents the architecture used to perform the valve turning intervention. Results obtained from the valve turning are presented and analyzed in Section IV. Section V summarizes and concludes the work.

II. State of the Art

LbD is a machine learning technique designed to transfer knowledge from an expert to a machine. LbD algorithms follow three sequential phases: first, a set of *demonstrations* of the task are recorded; second, the algorithm *learns* by generalizing all demonstrations and creates a model; finally, the algorithm uses the model to *reproduce* the task.

The two main methods to transfer the knowledge to a machine are: *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. The learned controllers can generate static or dynamic trajectories, which adapt to the current state of the robot.

A. LbD related work

Several LbD algorithms have been proposed depending on the application requirements. D.R.Faria [4] proposed to learn the manipulation and grasping of an object using geometry, based on the position of fingers and their pressure, and representing them with a probabilistic volumetric model. Calinon [5] proposed to represent trajectories using a Gaussian mixture model (GMM). This representation was extended by Kruger [6] using Incremental GMM to automatically set the number of Gaussians. Furthermore, Calinon [7] used different types of parametrized regressions to adjust the trajectory learnt during the demonstrations. Similarly to the GMM, a hidden Markov model (HMM) [8] can be used to represent a trajectory. Also the HMM can be parametrized [9]. A different option to encode the trajectory is using a dynamic movevement primitives (DMP) [10], which can be extended for working in closed loop [11]. Also, forces exert along the trajectory can be learned by an extended DMP[12].

B. Dynamic movement primitives (DMP)

DMP has been chosen due to its dynamic trajectory generation 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 III-F1. Furthermore, the model generated is translational and time invariant.

DMP is a framework where the skill is encapsulated in a superposition of basis motion fields. The method used in this paper is an extension of DMP [13] proposed by Kormushev [12].

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.

The number of attractors or Gaussians (K) is preselected by the user. Their centers μ_i^T are equally distributed in time, and the variance parameters Σ_i^T are set to a constant value inversely proportional to the number of Gaussians ($\Sigma_i^T = total\_time/K$). On the other hand a α value is also defined, it depends on the duration of the demonstrations. This parameter is used in the canonical system which is explained in the reproduction phase. The weights associated to each Gaussian $h_i(t)$ are defined by the Equation:

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

The position of the centers, μ_i^x, and the stiffness matrix K_i^P are learned from the observed data through least-squares regressions. All the data from the demonstrations is concatenated

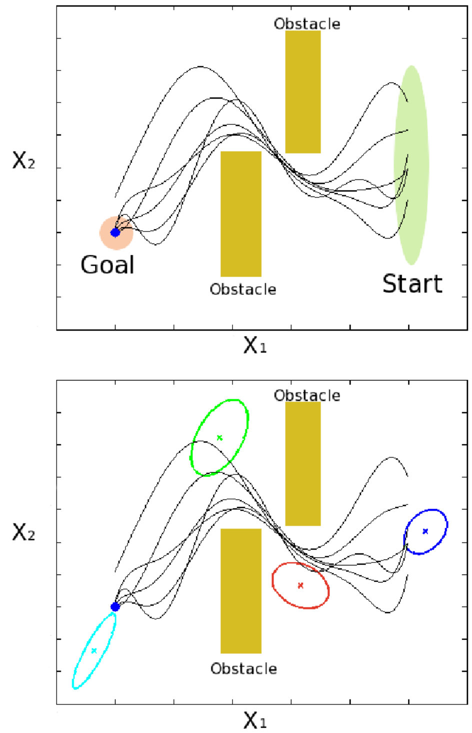

Fig. 2. Top figure shows the set of 2D demonstrated trajectories. Time is not represented in the trajectory, and is used directly with weights. The trajectory demonstrated has to pass between the two obstacles. Bottom figure shows the encoding of the trajectory with a DMP formed by 4 gaussians.

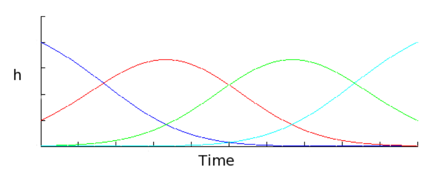

Fig. 3. Weights associated to each gaussian corresponding to the example shown in Figure 2

in a matrix $Y = [\ddot{x}\frac{1}{K^P} + \dot{x}\frac{K^V}{K^P} + x]$, 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.

To compute the stiffness and damping, the user needs to define a minimum K_{min}^P, and maximum K_{max}^P. The equations bellow estimate the stiffness and the damping.

$$K^P = K_{min}^P + \frac{K_{max}^P - K_{min}^P}{2}, \qquad (2)$$

$$K^V = 2\sqrt{K^P}, \qquad (3)$$

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 ($i \in \{1, ...K\}$).

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

where:

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

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

The DMP algorithm uses the residual terms of the regression process to estimate the stiffness matrices K_i^P through the eigen components decomposition.

$$K_i^P = V_i D_i V_i^{-1}, \quad (6)$$

where:

$$D_i = k_{min}^P + (k_{max}^P - k_{min}^P)\frac{\lambda_i - \lambda_{min}}{\lambda_{max} - \lambda_{min}}. \quad (7)$$

In (6), 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 center to represent the Gaussians; $h_i(t)$ represent the influence of each matrix functions; K^V the damping; and α, which is assigned according to the duration of the sample. Figure 2 and Fig. 3 show a simple example where the learned data is represented.

To reproduce the learned skill, the desired acceleration is generated with

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

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

The time to determine the superposition of gaussians is an implicit time generated using a decay term:

$$t = \frac{ln(s)}{\alpha}, \quad (9)$$

where s is a canonical system:

$$\dot{s} = s - \alpha s. \quad (10)$$

III. Valve Turning architectrue for I-AUV

The architecture is composed by several modules which are organized in layers, see Fig. 4. Base layer contains all sensors and actuators. Next layer has all perception systems to process sensor information, such as the localisation module and the perception systems that process the camera and F/T sensor. On top of it, the AUV and manipulator velocity controllers are in charge to follow the request velocities given by the LbD architecture. Finally, in the higher 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).

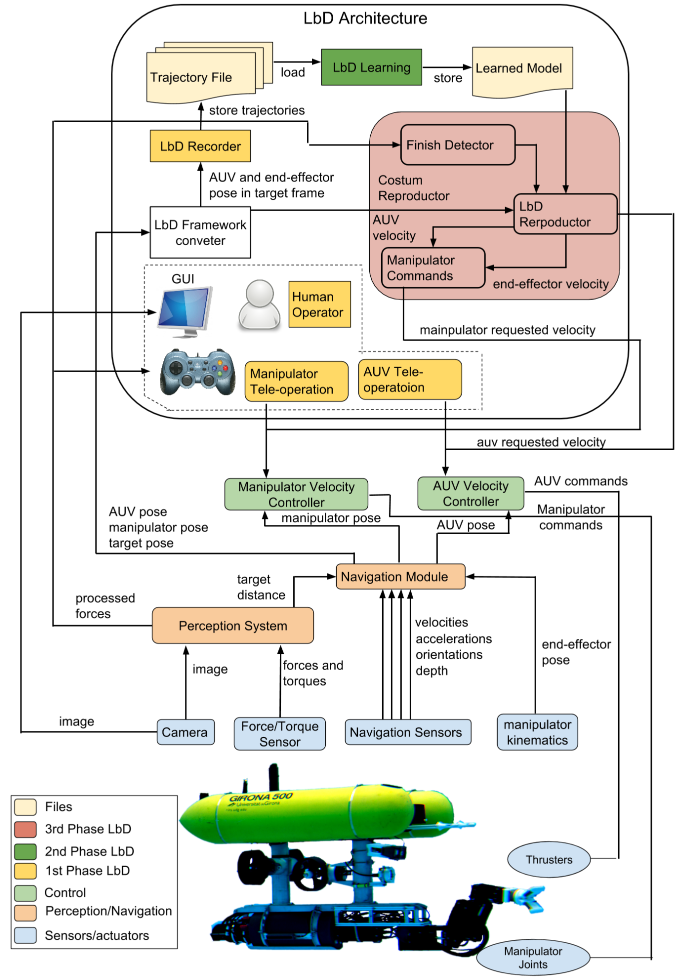

Fig. 4. Diagram of the structure of the LbD architecture and its connection to the AUV control architecture.

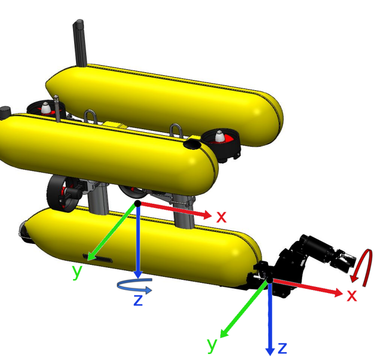

Fig. 5. This 3D model shows the actuated DoFs of the Girona 500 AUV and the CSIP manipulator.

A. Girona 500 AUV

The Girona500 AUV is a compact and lightweight AUV with hovering capabilities. Its propulsion system is configured with five thrusters to control four DoFs (x, y, z and *yaw* (ψ)), while the *roll*(Φ) and *pitch*(θ) are stable in this configuration (see Fig. 5).

B. Manipulator

An ECA CSIP manipulator, with four DoFs (*slew*, *elbow*, *elevation* and *roll*), is integrated in the Girona 500 AUV payload area. The manipulator is installed on the frontal vehicle frame to allow the manipulation of vertical panels. A customised end-effector was developed for this particular task.

The manipulator is underactuated, which means it is only able to control 4 DoFs: the cartesian positions(x, y, z) and *roll*(Φ) (see Fig. 5).

C. Customized end-effector

A customized end-effector has been developed to manipulate a T-bar handle valve. It is composed of three modules (see Figure 6).

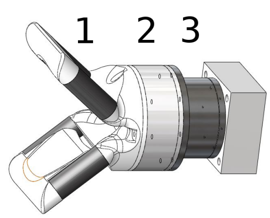

Fig. 6. This 3D model shows the customized end-effector, in which three blocks can be distinguished: 1 passive gripper, 2 camera in-hand and 3 F/T sensor.

1) *Passive Gripper*: the gripper is compliant to absorb small impacts and its V-shape helps to drive the handle of the T-bar to the end-effector center.
2) *Camera in hand*: a small camera has been installed in the center of the gripper, providing a visual feedback of what the end-effector is manipulating. This camera has been placed to avoid occlusion of the manipulator during the intervention.
3) *F/T sensor*: the sensor provides information during the manipulation about the quality of the grasping and the necessary torque to turn the valve.

At the current state, the camera in hand is only used by the operator during the demonstration. The F/T sensor is used to verify that the end-effector is touching the valve handle.

D. Navigation and perception modules

Detection of the underwater panel is performed using vision. The images gathered by the AUV's main camera are compared with an *a priori* known template of the panel. By detecting and matching unique features in the camera image and template, it is possible to identify the presence of the panel, as well as accurately estimate the position/orientation when a sufficient number of features are matched. The method used for this work is the oriented FAST and rotated BRIEF (ORB) [14] feature extractor. It has been chosen for that purpose given its suitability to real-time applications.

Additionally, since the geometry of the panel is known, valve positions can be computed with respect to the panel center. To estimate the valve's orientations, the region of interest around the expected valve position is computed. Then, the Hough line transform is used to estimate the main line orientation. Outliers are limited by constraining the length of lines and their orientation.

The AUV localization system uses an EKF-SLAM algorithm. The filter combines three navigation sensors: an AHRS, a DLV and a depth sensor, to estimate a better pose and velocity for the AUV. When the position of an external landmark is detected by vision, a single landmark SLAM algorithm is performed reducing the uncertainty in the AUV position. In this experiment, the panel is used as a landmark.

E. Manipulator and AUV Controller

Two independent controllers are used to achieve the proposed task.

1) AUV Velocity controller: This controller computes the force and torque necessary to reach the desired velocity. It combines a standard four DoF proportional-integral-derivative (PID) with an open loop model-based controller.

2) Manipulator controller: This controller computes the desired speed for each joint ($\dot{q}$) in order to reach the desired velocity of the end-effector in the Cartesian space. To this end, the desired velocity is transformed to a desired increment in Cartesian space($\vec{x}$), and using the pseudo-inverse Jacobian (J^+) of the manipulator, $\dot{q}$ is obtained as follows:

$$\dot{q} = J^+ \vec{x}. \tag{11}$$

As the manipulator is underactuated, the orientations are not included in the pseudo-inverse Jacobian.

F. LbD architecture

To solve the proposed task using the aforementioned setup, eight DoFs have to be learned: the algorithm learns the trajectory (*x*,*y*,*z*) and orientation (ψ) of the AUV and the trajectory and alignment of the end-effector. All these poses are referenced with respect to the valve to manipulate.

The LbD Architecture has three phases:

- First the operator performs the valve turning task tele-operating the AUV and the manipulator using a joystick. The operator has access to the vehicle camera images and the end-effector force feedback data and performs several demonstrations to represent different possibilities to turn the valve. During these demonstrations, the trajectory of the AUV and the end-effector are recorded.
- Second, when the operator decides that recorded demonstrations are representative enough of the task, the learning algorithm generates a model from all the recorded trajectories.

- Finally, the LbD algorithm is ready to perform the valve turning. The LbD reproductor loads the learned model and generates velocity requests for the AUV and the manipulator.

The learned skill handles the AUV and manipulator movement from the vehicle initial position until the grasping of the valve. The angle in which the valve is turned is not part of the learned skill, thus we can use the same model for any desired valve turning. Two additional features have been added into the LbD reproductor. The first one is a procedure that pushes the AUV against the valve during the turning and moves the AUV backwards to a safe distance from the panel when the turn is completed. The second one is a trigger to stop the learning procedure when the end-effector touches the valve.

1) DMP adaptation: The DMP, presented in the Section II-B, has been modified to allow an efficient and correct control of an AUV with a manipulator. Its main propose is to control the end-effector and AUV pose simultaneously. Since DMP uses a n-dimensional state vector, the addition of new variables to describe the vehicle orientation or the end-effector trajectory and orientation does not affect the algorithm. Hence, the same formulation can be used when adding these extra DoF.

The algorithm modifications consist of:

- **Orientation integration:** One extra value has been added to the learning system, to represent orientation between the panel and the vehicle *yaw* (ψ).
- **End-effector pose integration:** the pose of the end-effector is represented in the Cartesian space. Four more values (x, y, z, Φ) are added to the learning system.
- **End-effector interaction with the AUV:** The requested velocities sent to the manipulator are obtained by subtracting the AUV requested velocities from the end-effector velocities.
- **Finalisation condition integration:** The F/T sensor provides information to know if the end-effector has made contact with the valve and thus know is the reproduction of the learned trajectory has to finish.

IV. Results

This section shows the results obtained using the Girona 500 AUV to manipulate a subsea valve panel. All the experiments have been performed in a water tank using a mock-up subsea panel with T-shape valves.

A. Demonstration

The valve turning task starts when the AUV is in-front of the valve panel of a suitable inspection distance (between 1.75 and 2.0 meters). This pose is kept stable while the manipulator is moved to an appropriate configuration to grasp the valve. When the manipulator reaches the correct configuration, the AUV and the manipulator move together to grasp the valve.

For this experiment, the operator has performed 3 different demonstrations, see Fig. 7 and Fig. 8, where the valve is grasped by moving the end-effector close to the valve center. All the demonstrations have been done with the valve located at the top row middle column (see Fig. 9). The trajectories are represented in the frame of the valve, reaching a position near to zero with the end-effector.

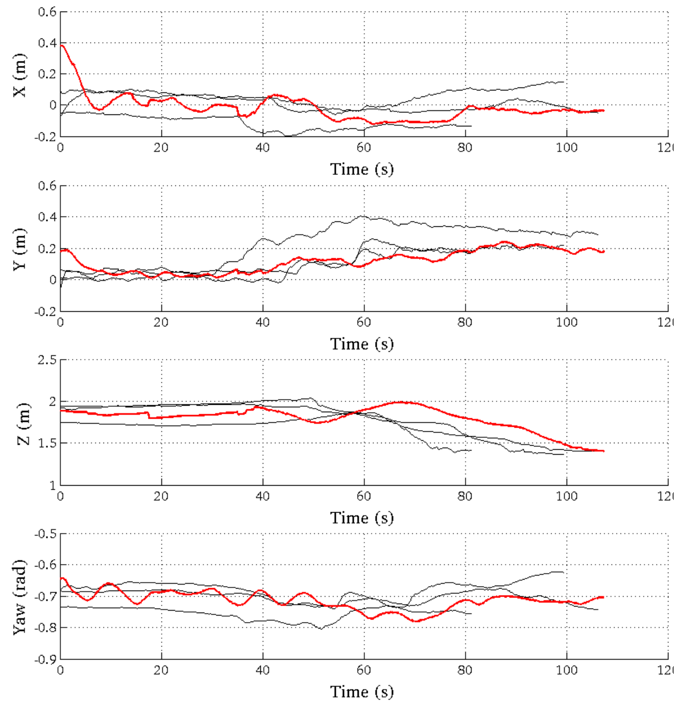

Fig. 7. Trajectory of the AUV for 3 demonstrations and 1 reproduction. The trajectory is represented in the valve frame and each controlled DoF is represented in one plot. The demonstrations are depicted in black lines and the reproduction in red.

B. Learning

The learning algorithm has used 20 Gaussians to represent the movement with accuracy. Figure 7 and Fig. 8 show an autonomous reproduction. These plots show the similarity between the demonstration and the reproduction generated with the learned model.

C. Reproduction

Figure 10 and Fig. 11 show four reproductions, one for each valve in the subsea panel. Even though, the valves are in different panel positions all the trajectories have the valve position at zero. The plots show that all the trajectories are similar independently of the target valve.

In the 80% of the reproductions the valve has been turned successfully. The majority of the failures are caused by errors in the alignment between the valve and the end-effector, due to detection error in the visual systems.Thus proving the repeatability of the method and the possibility to use it with any valve in the panel.

Figure 12 shows the trajectory of the end-effector and the trajectory of the AUV translated to the initial position of the end-effector. This figure allow us to appreciate the difference between both movements. It is easy to notice that the manipulator can control more accurately the pose than the AUV.

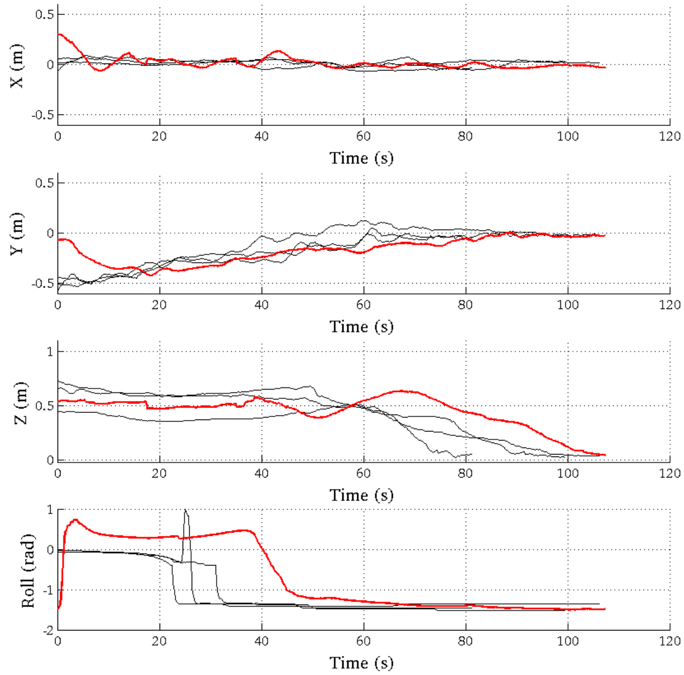

Fig. 8. Trajectory of the end-effector for 3 demonstrations and 1 reproduction. The trajectory is represented in the valve frame and each controlled DoF is represented in one plot. The demonstrations are depicted in black lines and the reproduction in red.

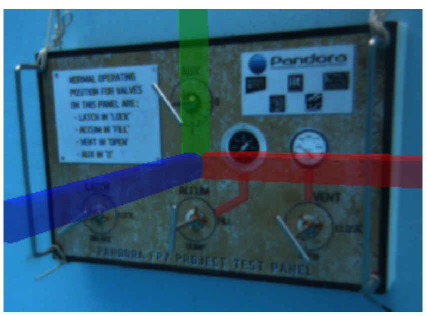

Fig. 9. Subsea valve panel with the 4 valves. The axis of the panel frame is overly in the center of the panel, where red is the x, green is the y and blue is the z.

V. Conclusions

The presented work has demonstrated the capability of an I-AUV to autonomously turn a valve from a subsea panel using LbD. The DMP algorithm has been adapted to learn the appropriate data to represent the valve turning task using 8 DoFs. Moreover, a new end-effector has been designed to simplify the grasping operation and also the different components in the whole system have been adjusted to perform the desired task. The results show how the I-AUV performs the same learned trajectory in the different valves of the panel and in different orientations.

Future work will include testing and adapting the method to perform interventions in a perturbed environment. Furthermore, the DMP will be adapted to include the forces to perform a constant pressure during the valve turning.

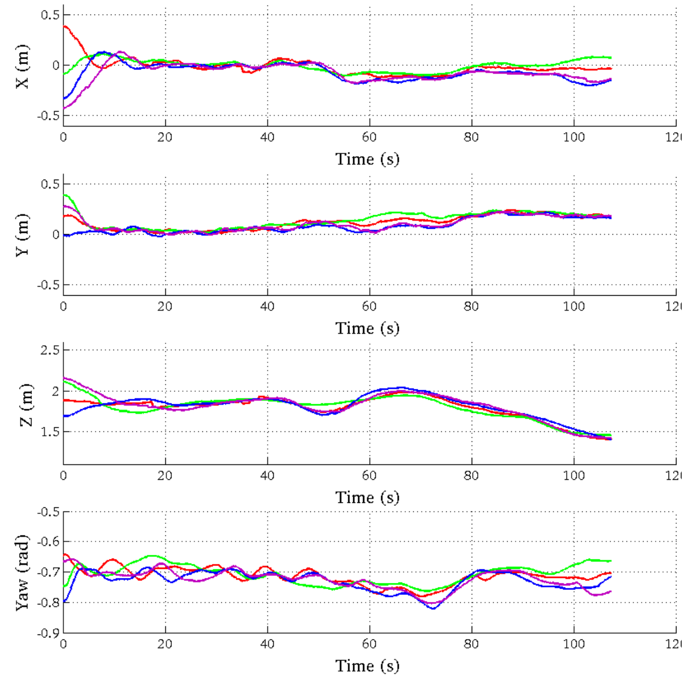

Fig. 10. Four different autonomous trajectories performed with the AUV. The trajectories are represented in each valve frame and each controlled DoF is shown in a separate plot.

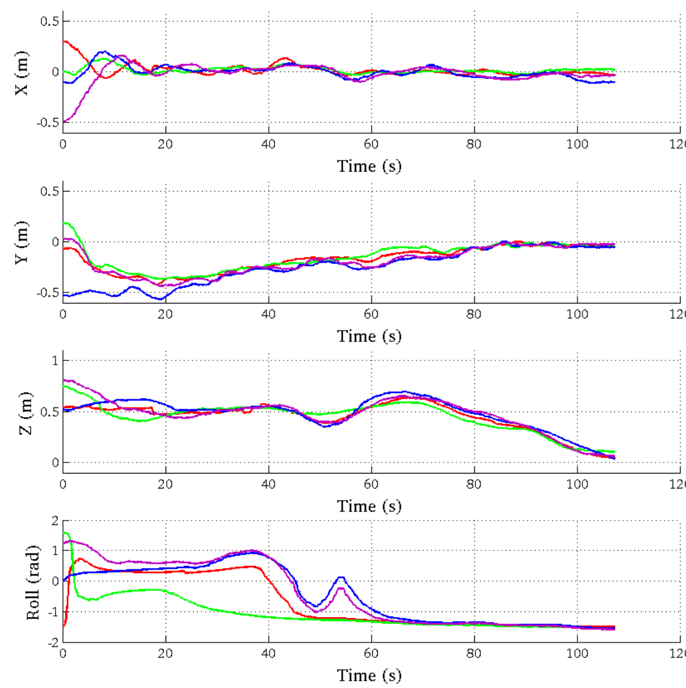

Fig. 11. Four different autonomous trajectories performed with the end-effector. The trajectories are represented in each valve frame and each controlled DoF is shown in a separate plot.

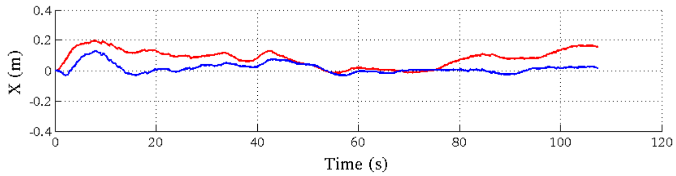

Fig. 12. Trajectory of the AUV in red and the end-effector in blue. The AUV trajectory has been translated to the initial pose of the end-effector in the trajectory.

ACKNOWLEDGMENT

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.

REFERENCES

[1] J. Evans, P. Redmond, C. Plakas, K. Hamilton, and D. Lane, "Autonomous docking for Intervention-AUVs using sonar and video-based real-time 3D pose estimation," *OCEANS 2003*, vol. 4, pp. 2201–2210, 2003.

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

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] A. Ijspeert, J. Nakanishi, and S. Schaal, "Trajectory formation for imitation with nonlinear dynamical systems," in *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, vol. 2, 2001, pp. 752–757 vol.2.

[14] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, nov. 2011, pp. 2564–2571.