Learning multiple strategies to perform a valve turning with underwater currents using an I-AUV

Arnau Carrera[1], Narcís Palomeras[2], Natàlia Hurtós[3], Petar Kormushev[4] and Marc Carreras[5]

***Abstract*— Recent efforts in the field of intervention-autonomous underwater vehicles (I-AUVs) have started to show promising results in simple manipulation tasks. However, there is still a long way to go to reach the complexity of the tasks carried out by ROV pilots. This paper proposes an intervention framework based on parametric Learning by Demonstration (p-LbD) techniques in order to acquire multiple strategies to perform an autonomous intervention task adapted to different environment conditions. The manipulation skills of a pilot are acquired thought a set of demonstrations done under different environment circumstances, in our case different levels of water current. The proposed algorithm is able to learn these different strategies and depending on the estimated water current, autonomously reproduce a combined strategy to perform the task. The p-LbD algorithm as well as its interplay with the rest of the modules that take part in the proposed framework are described in this paper. We also present results on a free-floating valve turning task, using the Girona 500 I-AUV equipped with a manipulator and a customized end-effector. The obtained results show the feasibility of the p-LbD algorithm to perform autonomous intervention tasks combining the learned strategies depending on the environment conditions.**

I. INTRODUCTION

Technology advances in the past decades have fostered underwater exploration by providing autonomous underwater vehicles (AUVs), which allow to cover large underwater areas while gathering all sorts of data. However, underwater intervention tasks, still rely on manned submersibles or remotely operated vehicles (ROVs), that require expensive vessels with dynamic positioning systems and a dedicated pilots for their operation.

The next natural step is to push AUV capabilities to perform underwater intervention, with the aim of reducing the costs of these kind of operations. Different research projects have approached the autonomous underwater intervention from distinct points of view. The SAUVIM project [1] proposed an underwater intervention using a priority order controller for the manipulator to recover objects of the seafloor. In the TRIDENT project [2] a system to search and recover objects with a light Intervention-AUV (I-AUV) was presented. The I-AUV implements an underwater vehicle manipulator scheme (UVMS) which is guided by a visual system to grasp the object. The TRITON project [3] demonstrated docked manipulation with an I-AUV connected in a sub-sea station. In this case only the manipulator is actuated during the intervention. The work presented here is conducted in the context of the PANDORA [4] project, that has the goal to perform free-floating manipulation of a sub-sea valve panel.

The experiments developed in the context of PANDORA project have been performed with the Girona 500 I-AUV (see Fig. 1) in a controlled environment were perturbations were introduced artificially. In previous work [], the intensity of the introduced perturbations was such that the low-level controller could overcome them in the majority of the cases. However, this paper aims to extend this work and perform autonomous underwater intervention in a more challenging environment, targeting a valve turning task under the influence of more intense underwater currents, which will require different strategies to perform the task correctly.

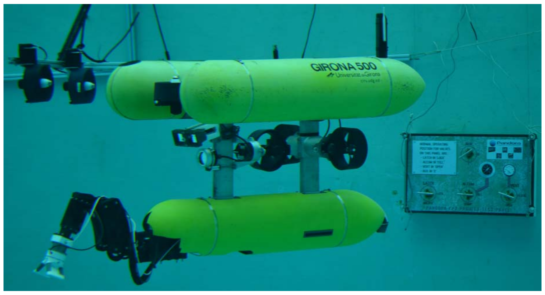


Fig. 1. Girona 500 I-AUV in the water tank, equipped with the manipulator and a customized end-effector. At the background there is a mock-up of a valve panel (right) and 2 thrusters to generate lateral water current perturbations (left).

An intervention framework based on a learning by demonstration (LbD) algorithm was previously proposed [5]. It is based on a machine learning technique that enables a robotic platform to easily learn a new task. To do it, rather than analytically decompose the problem and manually program a desired behavior, the LbD takes the knowledge from a set of demonstrations performed by an operator. To approach the problem under the presence of strong perturbations we have developed a new parametric-LbD (p-LbD) algorithm to extend the current implementation of the intervention framework. The p-LbD creates parametrized models according to the estimated environment perturbation conditions. Then, once the task is learned, the AUV is able to infer the appropriate reproduction from the models according to the actual water currents.

The rest of this paper is organised as follows. Section II presents the related work in LbD applied to manipulation.

Authors[1,2,3,5] are with Computer Vision and Robotics Group (VICOROB), University of Girona, 17071 Girona, Spain. arnau.carrera, marc.carreras at udg.edu and npalomer, nhurtos at eia.udg.edu

Author[4] is with Department of Advanced Robotics, Istituto Italiano di Tecnologia, via Morego, 30, 16163 Genova, Italy. petar.kormushev at iit.it

978-1-4799-8736-8/15/$31.00 ©2015 IEEE

Section V lays out the proposed intervention framework for an I-AUV and describes the used algorithm. Results obtained from this valve turning scenario are presented and analyzed in Section V. Finally, Section VI summarizes and concludes the work.

II. Related work

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

Different LbD algorithms have been proposed throughout the literature, depending on the method used to encode the learned trajectory. [6] proposed a representation based on Gaussian mixture model (GMM), which was later extended by [7] using incremental GMM to automatically set the number of Gaussians. Similar to the GMM, a hidden Markov model (HMM) [8] has also been used to represent a trajectory.

A different approach is to use dynamic movement primitives (DMP) [9] [10]. Unlike GMM and HMM, DMP uses the learned model to dynamically generate the required commands to perform the reproduction of the trajectory. This makes the approach more robust to external perturbations and easily adaptable to different domains. DMP has been extended by [11] to include a force associated with the trajectory.

Therefore, given the simplicity of the representation and its flexibility, DMP is more suitable in the context of this work and constitutes the base of our learning framework.

A. Parametric Learning by Demonstration (p-LbD)

To model a task with a high degree of variability with and LbD algorithm two strategies can be followed: either perform a high number of demonstrations in order to capture the variability of the task or learn one model for each different situation. Both solutions present problems. The former leads to a too generic model not representative of the task while the later forces the user to create a new model every time that the conditions change. The p-LbD algorithm overcome both problems identifying the key parameter that differentiates each situation and adapting the model consequently. Some relevant examples of different implementations of p-LbD algorithms are introduced next.

First, Kruger [12] proposed an LbD algorithm where the task to be learned is divided in parametrised actions represented by states in the HMM. To encode the action and simplify the number of states the authors used the effect of each action as the parametric value of the HMM. For instance, moving one object has as a parameter the initial and final position.

Recently, Calinon [13] proposed a p-LbD algorithm to move a conic peg from one place to another. The algorithm records the trajectories performed in the demonstration phase from two different frames placed at the initial and final position. These positions are the parameters used by the algorithm. From the two sets of trajectories (one from each frame) two models are computed and then, using the Gaussians properties, both models are merged. Then, giving as a parameters the initial and final positions, the algorithm is able to instantiate a new model and reproduce the learned task for these particular parameters.

Matsubara [14] proposed a parametric version of the DMP algorithm. A user performs a set of demonstrations each one with a different styles. For example, it moves a box from left to right passing over zero, one or two other boxes. Then, all the demonstrations are translated from the human right arm joint space to manipulators four degrees of freedom (DoF) joint space. Using the proposed style-DMP algorithm this demonstrations are then compactly encoded in a movement primitive that can be reproduced controlling its style parameter. A mapping between the height of the obstacle and the corresponding style parameter can be empirically learned from data.

In contrast to the previous approach we propose a parametric version of the DMP which learns different models associated to an environment condition. In the reproduction phase, a weighted combination of these models is generated according to the current environment situation. The motivation of this method is to represent different strategies to resolve the same task depending on external factors which are not represented in the learned model.

III. Learning a parametric-Dynamic Movement Primitive

We propose to parametrize the previous DMP algorithm presented in [5], which was successfully used to perform an underwater valve turning task.

That method was based on a modified version of the original DMP proposed by Calinon [15], where the commands are generated using a modification of a Gaussian Mixture Regression (GMR), which behave similar to the *Vector Integration To Endpoint (VITE)* originally proposed [16].

A. Dynamic Movement Primitive (DMP)

The DMP encapsulates the trajectory to be learnt as a superposition of basis motion fields (see Figure 2). To generate the proper superposition, there are a set of attractors with an associated weight that change its influence over the trajectory along the time.

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).

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

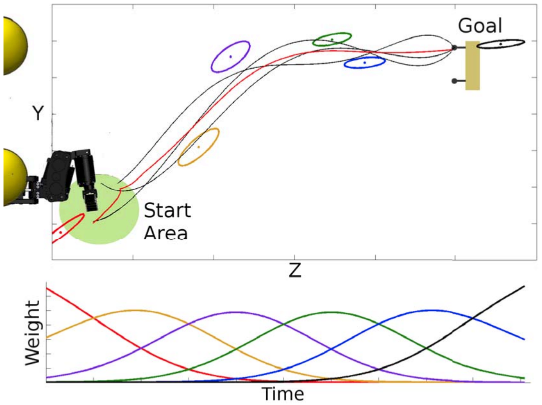


Fig. 2. Top figure shows a set of 2D demonstrated trajectories (black) and one reproduction (red). In this case, the demonstrated trajectory has to grasp the valve aligning the fore arm of the manipulator with the valve. Below, the weight of each Gaussian over the time is represented. The encoding of the trajectories using a DMP algorithm has been done using 6 Gaussians.

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

$$t = \frac{ln(s)}{\alpha}, \qquad (2)$$

where s is a canonical system : $\dot{s} = s - \alpha s$,

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:

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

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

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

where:

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

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.

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

where:

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

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

Therefore, the model for a given 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 2 shows a simple example where the learned data is represented.

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

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

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

B. Parametric-Dynamic Movement Primitive (p-DMP)

The p-DMP associates a parametric value to each recorded demonstration. These parameters can be defined by several values not related with the number of DoF learned. The parameter value ought to be defined by some environment condition. The only requirement to select a proper parameter is that should be possible to relate its values using a distance function.

All the demonstrations are grouped by the associated values thus conforming different groups. For each group, a representative model will be learned using the aforementioned DMP algorithm, with appropriate K_{max}^P, K_{min}^P and number of attractors (K).

In the reproduction phase, the influence (m) of each model is computed according to distance between the current value of the parameter and the different models. The influence value obtained with Equation (9) is normalized between the 0 and 1.

$$\forall j \in J, \quad m_j = 1 - \frac{dist(p, q_j)}{\sum_{i=1}^{J} dist(p, q_i)} \qquad (9)$$

where $dist$ is the function that defines the distance between the current set of parameters (p) and group parameters (q) and J is the list of learned groups.

Moreover, the influence of each model is applied to coordinate the advance of the time (n) for each group, facilitating the combination of models with different duration using time invariant model(t):

$$\forall j \in J, \quad t_j = \frac{ln(s_j)}{\alpha_j}, \qquad \text{where } s \text{ is: } \dot{s_j} = s_j - \alpha_j s_j n_j, \tag{10}$$

n_j is the proportional advance computed as:

$$\forall j \in J, \quad n_j = \frac{u_j}{\sum_{i=1}^{J}(u_i m_i)} \tag{11}$$

and u is the needed time to perform a regular reproduction of the learned model. Finally the mixture of all the influences is computed as follows:

$$\hat{\ddot{x}} = \sum_{j=1}^{J} m_j n_j \left(\sum_{i=1}^{K} h_{j,i}(t_j)[K^P_{j,i}(\mu^X_{j,i} - x) - K^v_j \dot{x}] \right), \tag{12}$$

to obtain the desired acceleration.

IV. Intervention Framework

The proposed intervention framework for an I-AUV can be divided in two parts. First, the hardware components, the vehicle and the manipulator. Second, the software architecture which gathers information from all sensors, does the computations and generates commands to control the I-AUV in a desired behavior.

A. Hardware components

The Girona 500 I-AUV [17] is a compact and lightweight AUV with hovering capabilities which can fulfil the particular needs of different application by means of mission-specific payloads and a reconfigurable propulsion system. The propulsion system is configured with five thrusters to control four DoFs (*surge*(*x*), *sway*(*y*), *heave*(*z*) and *yaw*). To perform the intervention task, the Girona500 is equipped with a robotic arm (see Fig. 1), with four DoFs (*slew*, *elbow*, *elevation* and *roll*) and a customized end-effector composed by a passive gripper, camera in-hand and a force/torque (F/T) sensor. The manipulator is installed in the front part of the vehicle to allow the manipulation of vertical panels.

B. Intervention Framework for I-AUV

The intervention framework for I-AUV is composed by several modules which are organized in four layers, (see Fig. 3). Starting from the bottom, the first layer contains all sensors and actuators drivers. Next layer has the perception modules to process sensors information. Main modules in this layer are the localization and the perception module, this layer is able to track the target elements using both modules. On top of it, there is the control layer in charge of fulfilling the desired commands given by the p-LbD algorithm or the teleoperation module. Finally, in the top layer there is the teleoperation module and the LbD algorithm that is in charge of acquiring data from demonstrations (phase 1), learning the model (phase 2) and reproducing the task by generating velocity set-points (phase 3).

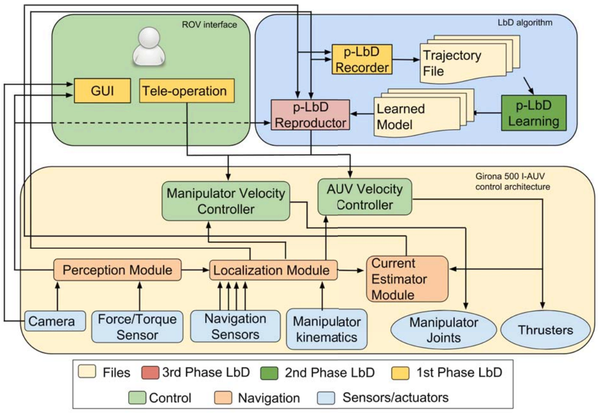


Fig. 3. Diagram of the structure of the intervention framework.

Two new modules have been developed and added in the existing intervention framework, the *current estimation* module and the *p-LbD algorithm*. The next subsections will focus on the explanation of this two modules.

1) Current Estimator: Since the Girona 500 I-AUV has no specific sensor to measure the underwater currents, we have developed a module to estimate the currents affecting the AUV. This module requires an initialization phase where the AUV needs to discover the necessary forces to keep a static position during a period of time in regular conditions. To obtain a better estimation the static position is relative to a static landmark (i.e. in front of a sub-sea valve panel). Once this phase has been performed the values are stored and can be reused as long as the environment conditions do not change. Before starting an intervention attempt, this module is activated and tries to keep the same position for the same period of time as in the initialization phase. Then, the forces are compared and the current force is estimated.

2) Parametric-LbD: Due to the hydrodynamics of the vehicle and the configuration of the thrusters, the AUV is more robust to perturbations in some axes than in others, for example the vehicle is more stable under a perturbation in $surge$(x) than in $sway$ (y). For this reason, the same tasks can be resolved more efficiently with different strategies according to the environment. The parameters selected for this algorithm are the current forces in the 3 axis, and the difference between the models is computed using the euclidean distance.

Like in previous work, the p-LbD algorithm controls the 8 DoFs. In this situation is important to control both elements (AUV and manipulator) together instead of using other approaches like underwater vehicle manipulator system (UVMS) [18] where only the end-effector position and orientation is controlled. In different strategies the position of the end-effector at the end will be similar but the vehicle position will have significant differences, which can not be easily controlled using an UVMS.

The explained p-DMP that targets the control of a 3 DoF trajectory (such as the one performed by an industrial manipulator) has been extended to control a total of eight DoFs, four DoF to represent the trajectory of the AUV and four DoF for the manipulator. Since DMP uses a n-dimensional state vector, the addition of new variables, does not affect the underlying formulation, however some changes are required.

End-effector pose integration: the pose of the end-effector is added to the learning model represented in the Cartesian space (x, y, z).

Orientation integration: To represent the AUV orientation (ψ) and the end-effector *roll* (Φ) the reproduction formula (8) has been modified to adapt to angles particularities. Basically, the difference $(\mu_i^X - x)$ is normalized (between π and $-\pi$).

End-effector interaction with the AUV: The requested velocities sent to the manipulator are obtained subtracting the AUV requested velocities to the end-effector velocities.

V. Results

The proposed p-DMP and the adapted intervention framework have been validated with experiments in a water tank with water current perturbations. The proposed approach has been validated under different degrees of perturbation, some similar to the situation of the demonstrations and other in new conditions. All the valves of the sub-sea panel have been turned several times achieving different success rates. The obtained results are explained according the three phases of the p-DMP algorithm.

A. Demonstration

The valve turning validation has been performed in a water tank using the Girona 500 I-AUV. To generate underwater currents two thrusters (providing up to 14 kg) have been placed in the vicinity of the intervention panel (Fig. 1 shows the test environment).

Demonstrations have been performed under the effect of two levels of perturbation: zero perturbations and high perturbations. Each one of these scenarios requires a different strategy to solve correctly the turning valve task. In the first one, the thrusters do not generate any current. In this situation, the strategy is to keep the AUV perpendicular in front of the panel while the manipulator is moved to an appropriate configuration to manipulate the valve, and afterwards, the valve is grasped and turned combining the movement of the AUV and the manipulator. In the second case, we turn on the thrusters to approximately the 70% of their thrust power. In this case, the AUV can not keep a perpendicular position in front of the sub-sea panel. Then, the panel is approached with an angle near to 45°. At this angle, the vehicle is able to keep the position and also detect the panel. Meanwhile the manipulator is moved laterally to an appropriate configuration to manipulate the valve. Finally, after adjusting the position of the AUV and the manipulator the valve is grasped and turned. Note that due to inherent manipulator limitations it is only possible to operate the valve from the right site of the AUV. For this reason the current perturbations have been always applied from the same side of the panel.

To obtain a more generic approach, the turning of the valve is not included in the learning model allowing to reuse the same learning for all the possible turns. Moreover, all the models are learned with respect the target valve coordinate frame thus allowing to use the same model for all different valves.

B. Learning

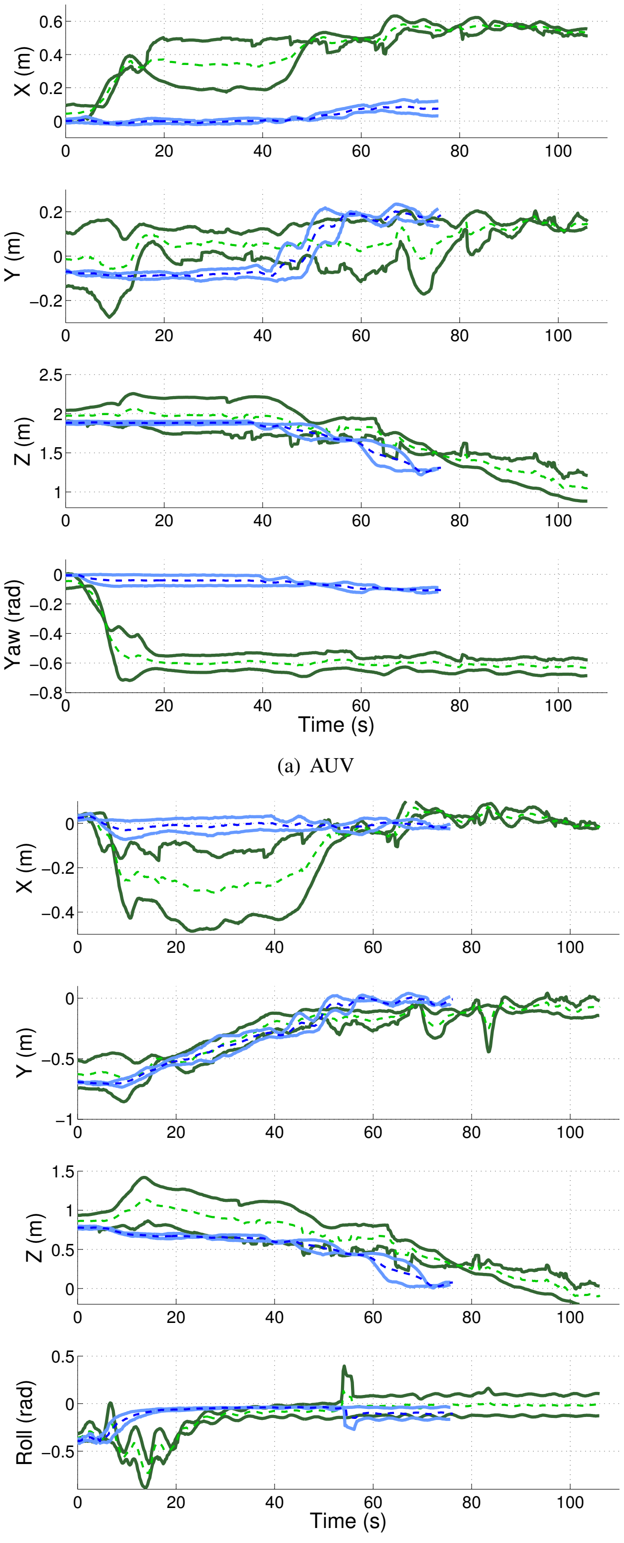
Two different models have been learned, the zero perturbation strategy has an average associated force of x=0.1, y=0.5 and z=0.4 N and the high perturbations strategy has an average associated force of x=1.2, y=8.5 and z=-2.5 N. We can appreciate that the more affected parameter is the y axis but the z and x axis become also affected as a consequence (Fig. 4 shows the set of demonstrations).

Once the demonstrations have been grouped in these two trajectory strategies, three different parameters have been adjusted to adapt the p-DMP method to them.

The first parameter is the number of Gaussians used to define the trajectory. A high number of Gaussians allows the representation of movements with many restrictions and low flexibility, while a small number is better to represent trajectories with more variability. The duration of the demonstrations also influences the number of Gaussians needed, since they are distributed uniformly along the trajectory. Thus, a longer experiment requires a bigger number of Gaussians to be obtain good accuracy.

In our case, this value has been adjusted experimentally to 11 and 15 Gaussians for the zero and the high current strategies. Despite it may seem a large number for the kind of learned trajectories, 11 and 15 Gaussians are required to ensure an accurate trajectory following, given the workspace limitations of the manipulator. The trajectory with no perturbations needs less time to resolve the task (75 seconds) compared with the high perturbation strategy which needs 110 seconds, for this reason they require a different number of Gaussians to obtain similar results.

The other two parameters to be set are K^P_{min} and K^p_{max}. These values define the limit and the initial value for the search of the stiffness (K^P) and the damping of the system (K^V) associated with each Gaussian. If the values are small and close to each other, they produce a low and smooth velocity command. Otherwise, if limits are big and the difference is high, the generated velocity commands will be bigger, which can be unstable for controlling precise trajectories but more suitable to external perturbations. Hence, the values of these parameters have been chosen as a trade-off between smoothness and robustness with values equal to 0.5 and 2.5 respectively for both models. Both K^P_{min} and K^p_{max} are equal for the two strategies learned since the values of this parameters are also related with the hardware used, which in



(a) AUV

(b) End-effector

Fig. 4. Average trajectory (dashed-line) and upper and lower limit of the 4 demonstrated trajectories. Each plot shows a single DoF for the manipulator and the end-effector. Depicted in blue, the group of no perturbations and in green the demonstrations under perturbations. All trajectories are represented in the frame of the target valve.

this case is exactly the same, and the kind of movement to be learned, which in both strategies is similar.

C. Reproduction

We have performed an experiment consisting of three turns for each four valves in three different environments: no perturbations, moderate perturbations (40-50%), high perturbations (70-80%). The success rate of correctly performed valve turns along all the experiment has been of 80.05% (29 out of 36 attempts). The 19.95% of error can be attributed to the sum of different small errors in the estimation of the target's pose (i.e. valve pose) and an error in the estimation of the manipulator's calibration. To diminish the problem in the arm-calibration, a re-calibration procedure is performed every two valve turning attempts. On the other hand, to reduce the error in the detection of the target when the vehicle is not perpendicular to the panel the camera in hand is used to detect the valve position. Although this camera is less precise than the vehicle camera due to the smaller field of view and lower quality of the gathered images, merging the information of both cameras improves the estimation.

no perturbation	moderate perturbation	high perturbation
91.6 %	75.0 %	75.0 %

TABLE I

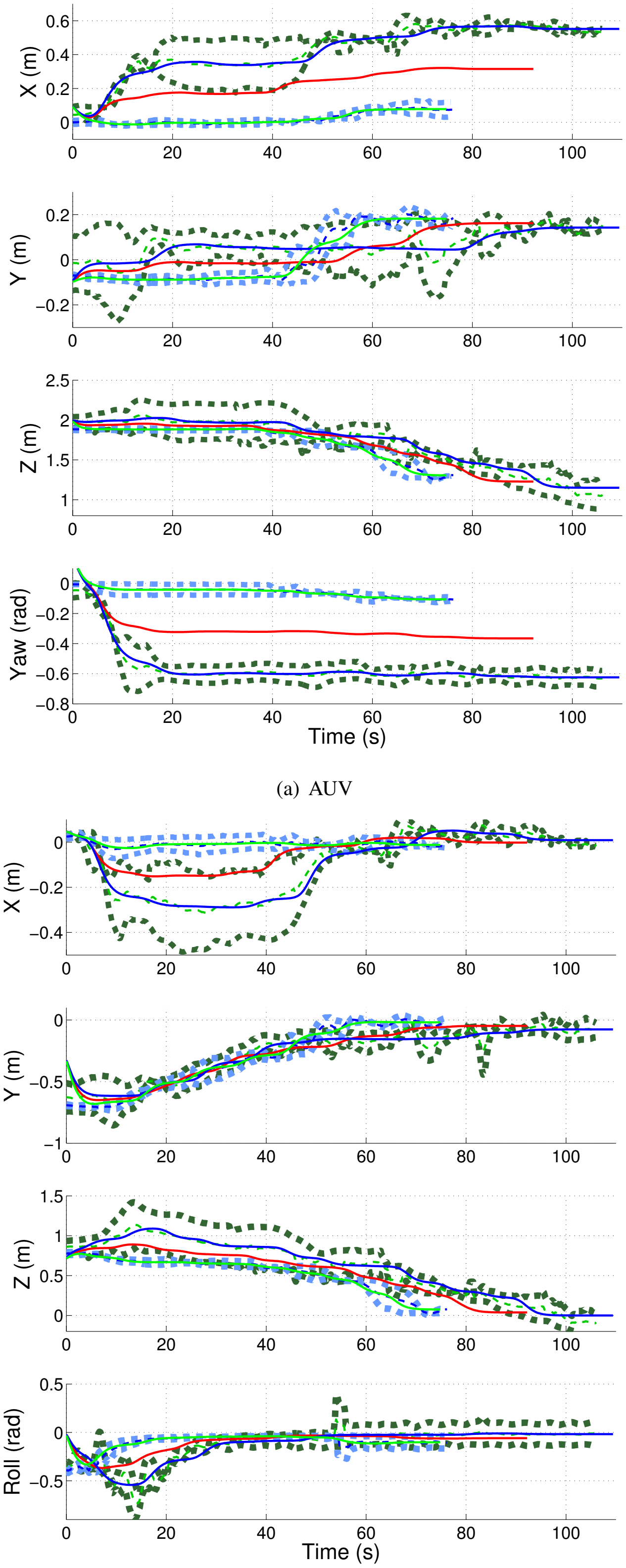
SUCCESS RATE IN PERCENT (%) ASSOCIATED TO EACH PERTURBATION GROUP.

Table I shows the success rate on the different environment situations. It illustrates how the perturbations introduce more instability in the performance of the AUV. However, although the new strategy is more complex to perform, it is more robust in front of perturbations. Comparing this results with previous experiments [19] where there was only one strategy (perpendicular approach) the success rate has increased from 65% to 75% in moderate perturbations and from 50% to 75% in high perturbations.

Figure 5 shows the two groups of demonstrated trajectories and one successful trajectory for each group. The autonomous trajectories depicted in the figure for no perturbation (x=0.0, y=0.41 and z=0.6 N) and high perturbations (x=1.4, y=8.0 and z=-2.4 N) show how the AUV and the end-effector follow the average trajectory of the demonstration in a smooth movement. The moderate perturbation (x=0.7, y=4.1 and z=-1.1 N) trajectory follows a new strategy which requires an average of 90 second to be performed and follows a new trajectory which is a mixture of the two learned models.

VI. CONCLUSIONS

The underwater domain is a challenging environment for performing intervention tasks that require manipulation of objects or structures. The implementation of these tasks is not trivial if the traditional programming approach (i.e. give a sequence of waypoints) is used. Moreover, in case of changing to a new intervention task or developing the same



(a) AUV

(b) End-effector

Fig. 5. Three autonomous trajectories are depicted together with the average, upper-limit and lower-limit of the two demonstrated groups (dashed green and blue lines). The trajectory under no perturbations is depicted in green, moderate-perturbations in red, high-perturbations in blue. All trajectories are represented in the frame of the target valve. Each plot shows a single DoF for the manipulator and the end-effector. The time axis corresponds to the real time of the experiments which in this case is the equivalent to the one generated by the canonical system.

task in different environment conditions reprogramming the mission requires significant programming efforts.

To handle these issues, this paper has presented the use of the p-LbD algorithm integrated in an intervention framework for I-AUVs, which allows to learn, in a easy and generic way, new capabilities from a human operator. The p-LbD technique sits at the core of the framework, using a new p-DMP algorithm which learns and reproduces different strategies to resolve the same task according to the actual environment situations. The intervention framework controls independently eight DoF, four for the AUV and four for the manipulator. 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 in a dynamic environment in which underwater currents can be set to different speed levels.

A procedure to estimate such currents has been implemented. This procedure measures the forces received by the AUV while keeping a stable pose before starting the intervention. The estimated current force is used as input parameter on the p-DMP algorithm to select the appropriate strategy. The results of this experiment have proved the suitability of the proposed method obtaining similar results than a human operator. Furthermore, the method has showed a correct behaviour combining the strategies taught depending on the environment conditions at each moment.

Future work will focus on trying to automatize the selection of the DMP parameters such as the number of required Gaussians or the stiffness and damping values. This would be of special interest when very different tasks must be programmed, as determining the parameters experimentally could turn into a cumbersome procedure. Also, it could be interesting to include the F/T sensor data in the DMP algorithm so that the system can learn/reproduce how the human operator interacts, in terms of force, with the target.

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

[2] M. Prats, J. Garcia, S. Wirth, D. Ribas, P. Sanz, P. Ridao, N. Gracias, and G. Oliver, “Multipurpose autonomous underwater intervention: A systems integration perspective,” in *Control Automation (MED), 2012 20th Mediterranean Conference on*, July 2012, pp. 1379–1384.

[3] N. Palomeras, A. Peñalver, M. Massot-Campos, G. Vallicrosa, P. L. Negre Carrasco, J. J. Fernández, P. Ridao, P. J. Sanz, G. A. Oliver, and A. Palomer, “I-AUV Docking and Intervention in a Subsea Panel,” in *Intelligent Robots and Systems (IROS), 2014*. IEEE, 2014.

[4] (2013) The PANDORA Website. [Online]. Available: http://persistentautonomy.com

[5] A. Carrera, N. Palomeras, D. Ribas, P. Kormushev, and M. Carreras, “An Intervention-AUV learns how to perform an underwater valve turning,” in *OCEANS - Taipei, 2014 MTS/IEEE*, 2014, pp. 1–7.

[6] 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.

[7] 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.

[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] H. Hoffmann, P. Pastor, D.-H. Park, and S. Schaal, "Biologically-inspired dynamical systems for movement generation: Automatic real-time goal adaptation and obstacle avoidance." in *ICRA*. IEEE, 2009, pp. 2587–2592.

[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] 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.

[12] 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.

[13] S. Calinon, D. Bruno, and D. G. Caldwell, "A task-parameterized probabilistic model with minimal intervention control," in *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, Hong Kong, China, May-June 2014.

[14] T. Matsubara, S.-H. Hyon, and J. Morimoto, "Learning parametric dynamic movement primitives from multiple demonstrations," *Neural Networks*, vol. 24, no. 5, pp. 493–500, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0893608011000566

[15] S. Calinon, I. Sardellitti, and D. Caldwell, "Learning-based control strategy for safe human-robot interaction exploiting task and robot redundancies," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, Oct 2010, pp. 249–254.

[16] 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.

[17] 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.

[18] G. Antonelli, *Underwater robots*. Cham: Springer International Publishing, 2014, ch. Kinematic Control of UVMSs, pp. 127–168.

[19] A. Carrera, N. Palomeras, N. Hurtós, P. Kormushev, and M. Carreras, "Learning by demonstration applied to long-term underwater intervention," *Special Issue on Cognitive Systems for Knowledge Discovery*, Under Revision, 2015.