Towards valve turning with an AUV using Learning by Demonstration

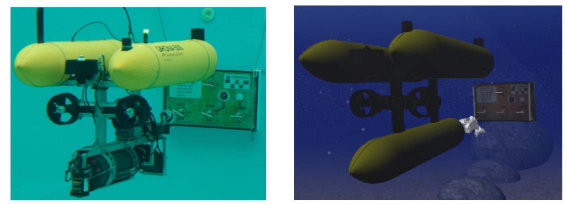
Arnau Carrera[1], Marc Carreras[2], Petar Kormushev[3], Narcís Palomeras[4] and Sharad Nagappa[5]

***Abstract*— AUVs have experimentally done the first steps to solve basic intervention tasks. First results have been promising in the task of retrieving an object from the seabed. In this paper we extend the complexity of the task with the help of a Learning by Demonstration (LbD) approach. An extension of a LbD algorithm to learn the pose and orientation of the trajectory is presented to achieve a valve turning task. A batch of demonstrations done in ROV mode is used by the LbD to learn the trajectory, taking advantage of the experience of the pilots. Moreover, the paper present a controller able to coordinate the movement of the manipulator using the position or the orientation of the end-effector and also moving the AUV when is required. Both systems have been tested together in a simulated environment to solve the task of interacting with a valve located on a ROV panel. The experiments has been done in an environment without perturbations and in an environment with different perturbations. The method has been able to overcome the perturbations and complete the task successfully. Furthermore, the proposed controller has simplified the use of the manipulator during the intervention task. The robot is equipped with a 4 DOFs manipulator having a griper as end effector to operate the T bar handles found in the panel. Panel and valve handle position and orientation are detected by a computer vision program based on template matching.**

I. INTRODUCTION

Nowadays, the most common task where Autonomous Underwater Vehicle (AUV) are used is the exploration of the seabed or water column recollecting information using different kind of sensors. Collected data can be used in applications like seabed mapping, hull inspection or water monitoring. Recently, AUVs have been used to recover objects from the seabed using a robotic arm [1]. The results obtained are promising but still not enough robust and general to be applied in any different underwater intervention.

The Persistent Autonomy through learNing, aDaptation, Observation and ReplAnning (PANDORA) project [2] has the aim of making AUVs *Persistently Autonomous*. One of the goals in this project is to improve the lack of skill of AUVs, making feasible to perform an intervention task. In one scenario the robot has to identify the appropriate valve from a panel and turned it, while the vehicle is hovering in front of the panel without docking. During the intervention, the vehicle has to compensate the different perturbations or even suspend the task to avoid possible damages. Figure 1 shows an image of the Girona500 AUV in front of the valve panel and a representation of the valve panel and the Girona500 with a manipulator in a simulator.



(a) Real image of the robot keeping position in front of the panel (b) Simulated robot with the manipulator approaching the panel

Fig. 1. Girona500 AUV attempting the valve turning task.

To solve this kind of intervention it is important to follow a particular trajectory to grasp the valve safely and avoiding dangerous positions. This paper proposes the use of Learning by Demonstration (LbD) to learn the trajectory using the knowledge of experimented Remotely Operated underwater Vehicles (ROV) pilots. This concept has been proved previously in a laboratory environment [3], [4]. The following properties of the LbD make it suitable for planning the movement of such intervention task: ease of representation and learning, compactness of the representation, robustness against perturbations and changes in a dynamic environment, ease of reuse for related tasks and easy modification for new tasks.

Usually LbD is used to learn any human skill which can be represented by a trajectory. LbD has different methods for encoding the trajectory. One option is to use a symbolic level implementing a graph [5] or tree structure [6] to represent the trajectory. Another option is to encode the trajectory in a statistical representation based in a Gaussian Mixture Model (GMM) [7] [8]. Also it is possible to encode the trajectory using dynamic representations for example the Dynamic Movement Primitive (DMP) [9] or evolutions of it [10]. LbD can also be combined with other methods, such as Reinforcement Learning (RL) [11] which continues the learning after the demonstrations according to future experience. Also methods that combine LbD, RL and a prediction model [12] adding robustness. Finally, there are other methods which are able to learn from failed demonstrations, which can be very useful when the task is too complex to be demonstrated efficiently [13].

In the intervention task, the pilot controls the position of the end-effector and the position of the AUV. This paper presents a teleoperation system which controlling the position of the end-effector controls both systems, the manipulator and the robot position. This kind of controller

Authors[1,2,4,5] are with Computer Vision and Robotics Group (VICOROB), University of Girona, 17071 Girona, Spain. arnau.carrera, marc.carreras, narcis.palomeras, sharad.nagappa at udg.edu

Author[3] is with Department of Advanced Robotics, Istituto Italiano di Tecnologia, via Morego, 30, 16163 Genova, Italy. petar.kormushev at iit.it

978-1-4799-0002-2/13/$31.00 ©2013 IEEE

is developed to do an intervention where the robot is really close to the valve panel but can receive small perturbations during the operation, making the goal not reachable without correcting the position of the robot. With this combined movement the same trajectory learnt can control the manipulator and modify the position of the robot.

This paper shows the suitability of the approach in a simulated environment using the UWSim [14] where the Girona500 [15] AUV is simulated and equipped with a 4 DoF manipulator with a gripper as end-effector to manipulate the valve in a T shape handle bar. Panel and valve handle position and orientation are detected by a computer vision program based on template matching. Handle orientation is found with a Hough line transform. The experiments consist in two steps. First, a set of demonstrations are done, where a user controls the end-effector following different trajectories to grasp the valve. Second, after learning from the batch of demonstrations, the robot grasps the valve starting from a new initial position and also receiving small perturbations during the intervention.

This paper has been organized as follow. Section II details the complete theoretical approach that has been used for learning the manipulator trajectories. Section III describes the intervention AUV set-up that has been used for the experiments. Section IV presents the obtained results and, finally, Section V summarizes and concludes the work.

II. THEORETICAL APPROACH

The aim of LbD is to program robots in a more *natural way*, making them suitable for many different types of tasks. There exist different methods to learn and reproduce the task learnt, but all of them have the same three steps:

- **Demonstration.** In this step a batch of demonstrations is recorded.
- **Learning or Mapping the skill.** The data recorded in the previous step is processed and a mapping function or a model of the skill is extracted.
- **Reproduction of the skill.** The model or the function learnt is reproduced be generalizing the skill.

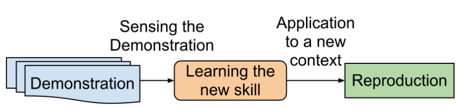


Fig. 2. Schema of the three step process used in LbD

There are several methods in LbD which are suitable to solve this task, such as Dynamic Movement Primitive (DMP) [16][17]. The trajectory is encoded as a superposition of basis motion fields, represented by a mixture of gaussians which are dynamically weighted (Figure 3).

The main advantages which have been considered to choose this method are: Firstly, the desired trajectory is generated dynamically during the reproduction making it robust to external perturbations. Secondly, the flexibility and simplicity of the representation makes it possible to adapt the algorithm to different purposes adding few changes.

A. Dynamic Movement Primitive (DMP)

The DMP is a framework where the skill is encapsulated in a superposition of basis motion fields. This method has a compact representation and it generates movement trajectories that are robust against perturbations. The method used is an extension of DMP [16] [17] proposed by Kormushev [18].

To better understand this encoding, we can imagine that we have 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 represent the points where the mass is attached, several gaussians uniformly set along the time are used. To represent the damping gains and the stiffness of the string, matrixes K^V and K^P are defined. And to represent the influence of each attractor along the experiment, a set of weights h is defined. Figure 3 shows en example of a trajectory in 2D and how the attractors will be set. Figure 4 shows the weights associated to each gaussian of the encoded trajectory.

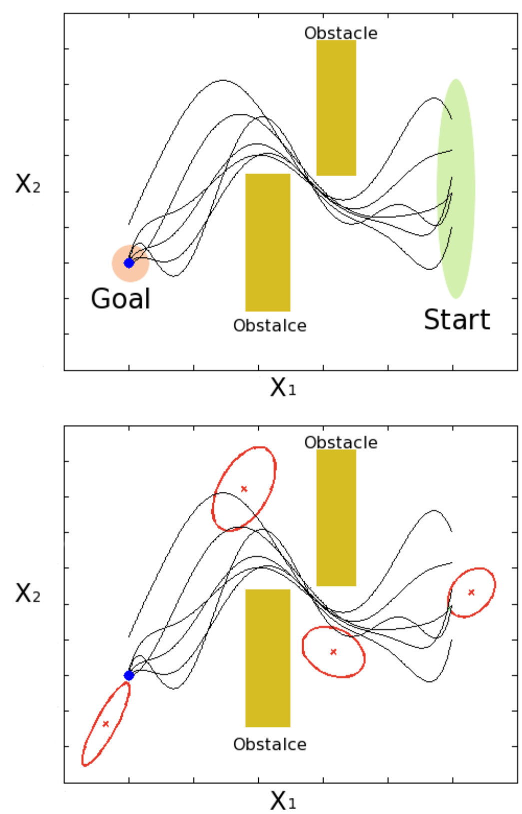


Fig. 3. Top Figure shows the set of 2D demonstrated trajectories. Time is not represented in the trajectory, and is used directly with weights. The trajectory demonstrated has to pass between the two obstacles. Bottom Figure shows the encoding of the trajectory with a DMP formed by 4 gaussians.

The data used to learn the trajectory consists of a set of points having the position x, the velocity $\dot{x}$ and the acceleration $\ddot{x}$, for each time step.

The set of K Gaussians is defined in time space, with

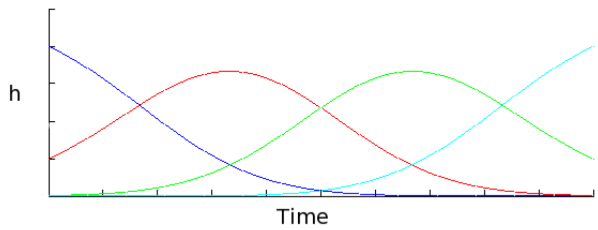


Fig. 4. Weights associated to each gaussian corresponding to the example shown in Fig. 3

centres μ_i equally distributed in time, and variance parameters σ set to a constant value inversely proportional to the number of states. By determining the weights $h_i(t)$ through the decay term s , the system will sequentially converge to the set of attractors in Cartesian space defined by centres μ_i^x and stiffness matrices k_i^P, which are learned from the dataset.

The desired acceleration to generate the trajectory is computed using the Eq.1, where x and $\dot{x}$ are the current position and velocity.

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

The time to determine the superposition of gaussians is an implicit time generated using a decay term Eq.2 where s is a canonical system Eq.3.

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

$$\dot{s} = s - \alpha s \quad (3)$$

To summarize, DMP has the following favorable features:

- Any movement can be efficiently learned an generated.
- The representation is translational and time invariant.
- Temporal/spatial coupling term can be incorporated.

B. Learning the Manipulator Trajectory

The DMP algorithm is designed to move an object following a desired trajectory where the orientation is not taken in consideration. In our case, to grasp the valve and avoid collisions between the end-effector and the valve, it is important to take in consideration the orientation between the valve and the end-effector. For this reason we have included three more dimensions in the DMP algorithm to represent the position and the orientation in Euler angles.

This modification implies small changes in the algorithm adding three dimensions to all the different variables of the algorithm: the attractors and the inputs and outputs of the generated data.

The trajectory learnt has the frame centre situated in the valve's position, and the orientation learnt is the difference between the orientation of the valve and the orientation of the end-effector. With this configuration the valve can be moved to any different position and the learnt data will be valid to reproduce the experiment.

III. REAL SET-UP

A. The vehicle and the manipulator

The Girona500 [15] is a compact and lightweight AUV with hovering capabilities which can fulfil the particular needs of any application by means of mission-specific payloads and a reconfigurable propulsion system. The propulsion system is configured with 5 thrusters to control 4 DoFs (surge, sway, heave and yaw). To perform the intervention task, Girona500 is equipped with a robotic arm (see Fig. 5), with 4 DoFs (slew, elbow, elevation and roll) and a griper.

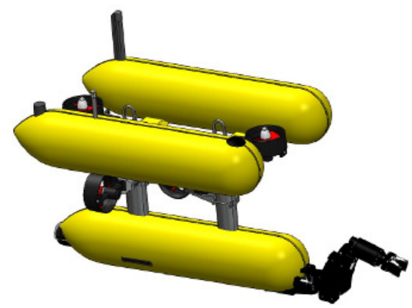


Fig. 5. A 3D model of Girona500 AUV with the CSIP 5E Micro Manipulator integrated in the front.

Nowadays, the integration of the manipulator is still not completed for this reason the experimental part has been tested in the simulation environment.

The manipulator is an *underactuated* arm and it has a small work area. To control the arm a system with two different parts is used, see Figure 6. First an analytic Inverse Kinematics (IK) is used which only controls the position and not the orientations of the end-effector. The non use of the orientation gives more possible positions, moreover some positions are only reachable from one orientation. When the arm controller receives a request of a position which is not reachable using the IK, because the position is to far or there is not possible configuration, the second part is activated. If there is an orientation the controller will move the joint which modify directly the orientation to the desired one without consider the position of the end-effector. On the other hand, the controller checks the distance between the end-effector and the desired position and sends a velocity command to move the AUV in the direction to the new pose. This avoids having the end-effector stuck in the middle of a trajectory and makes the whole system more robust against perturbations giving the possibility to reach any position. The controller has an hierarchical structure using the precision of each system, from the more precise to the less, because the movement of the AUV is less accurate than the movement of the arm.

To teach the demonstrations, a joystick which sends *x, y, z, yaw* and *pitch* increments to the end-effector in the orientation frame of the AUV base is used. The joystick sends the commands to the controller system explained before.

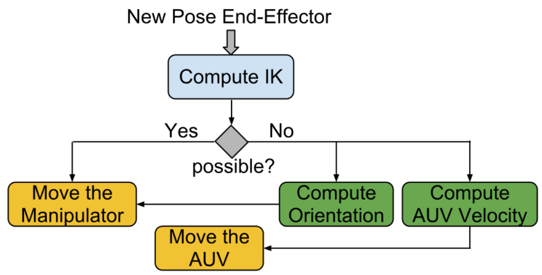


Fig. 6. The schema shows how a movement request is handled by the controller and how it hierarchically computes the command. First, using the IK to compute the new configuration for the joints of the manipulator. Second, if it is not possible to compute the IK, a new configuration will be computed using the orientation and also a speed command will be sent to move the AUV.

B. Navigation Module

The navigation module estimates the vehicle position and linear velocity and maps the position of several landmarks identified by a vision algorithm. The fusion algorithm in charge of simultaneously locating the vehicle and mapping (SLAM) these landmarks is an extended Kalman filter (EKF) [19]. Vehicle orientation and angular velocity are not estimated but directly measured by an internal motion reference unit (IMU).

A visual detection algorithm, detailed in Section III-C, gives information about the relative position of a landmark with respect to the vehicle. This information not only updates the detected landmark position but also the vehicle. The visual detection algorithm uses an *a priori* known template to identify and compute the relative position of these landmarks.

When the vehicle starts, its state vector contains only the vehicle position and linear velocity. There are no landmarks in the state vector. The first time the detection algorithm observes an object that it is able to identify, the landmark estimated position with respect to the world is introduced in the state vector.

In our case, the centre of the valve panel is included in the SLAM as landmark. Therefore, once the AUV has detected the valve panel, the position will be always known. The position will be an estimation with an uncertainty. This let to the LbD algorithm read the estimation of panel position and follow the desired trajectory without detecting it. As soon as, the panel is detected the values are updated in the navigation and the trajectory can be corrected finishing in the correct position and avoiding the error of the estimation.

This is necessary because during the manipulation the arm can interfere with the visual detector generating occlusions, and it will loose the position of the valve. In these cases, using the estimated position of the panel centre and the prior knowledge of the position of the valve respect the centre the position of the valve can be estimated and the trajectory can be followed.

C. Panel and valve detection

Detection of the underwater panel is performed using vision, by comparing the images from the camera against an *a priori* known template of the panel. By detecting and matching unique features in the camera image and template, it is possible to detect the presence of the panel, as well as accurately estimate the position/orientation when a sufficient number of features are matched.

In this work, we choose the oriented FAST and rotated BRIEF (ORB) [20] feature extractor for its suitability to real-time applications. The ORB feature extractor relies on features from accelerated segment test (FAST) corner detection [21] to detect features, or keypoints, in the image. These are obvious features to detect on man-made structures and can be detected very quickly. Moreover, there is a descriptor (binary) vector of the keypoint based on binary robust independent elementary features (BRIEF) [22]. This allows us to rapidly obtain the difference between descriptors and allows real-time matching of keypoints at higher image frame-rates when compared to the other commonly used feature extractors such as scale invariant feature transform (SIFT) [23] and speeded-up robust features (SURF) [24].

Figure 7 illustrates the matching between the panel template and an image received from the camera. A minimum number of keypoints must be matched between the template and the camera image to satisfy the panel detection requirement. A low number of matched keypoints indicates that the panel is not in the camera field of view. The correspondences between the template and camera image can be used to compute the transformation (or homography) of the template image to the detected panel in the camera image. This allows us to compute the image-coordinates of the corners of the panel in the camera image. Using the known geometry of the panel and the camera matrix, we are able to determine the pose of the panel in the camera coordinate system.

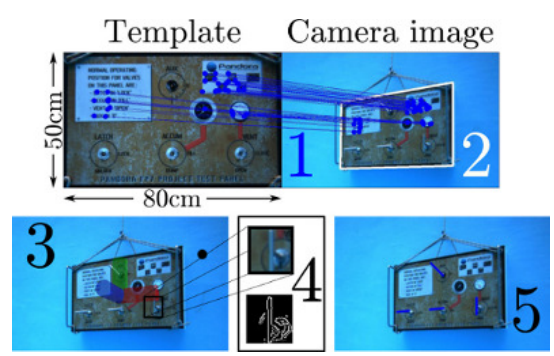


Fig. 7. Detection of the panel and valves consists of the following steps: 1) Match keypoints between the template and camera image. 2) Estimate corners of the panel in the camera image. 3) Estimate the translation and rotation of the panel in the image by using the known geometry of the panel. 4) Extract regions of interest and evaluate edges. 5) Estimate valve orientation using Hough transform to detect lines from the edges.

Additionally, since the geometry of the panel is known, the centres of valves on the panel is known in relation to the panel centre. Taking advantage of this, we search a small bounded region of the image for the orientation of the valve. The Hough line transform provides a straightforward method for detecting the orientation of the valves. Outliers are limited by constraining the length of lines and permissible orientations. The entire process of estimating the panel and valves is illustrated in Figure 7.

IV. RESULTS

This section shows the results obtained in the conducted experiments. Two experiments are presented, without and with perturbations.

A. Approaching the valve

In this experiment the AUV has been previously set in a proper position where the valve can be manipulated, and the proposed algorithm follows the desired trajectory to grasp the valve, see Figure 8. Figure 9 shows in the cartesian space the trajectory of the end-effector using as the reference frame the valve pose, making the goal pose approximately at (0,0,0). Figures 10, 11 and 12 show the difference between the orientation of the valve and the orientation of the end-effector. In these experiments, twenty five attractors have been used to obtain a big accuracy in the trajectory. The trajectory in the x,y,z is the average of all the demonstration trajectories, the roll pitch and yaw have small variation with respect to the demonstrations because during the experiment, the control has found few positions which are not reachable using the IK. This fact confirms that the positions are only reachable in a very similar angle configuration. For this reason, to follow a trajectory using only the position is possible.

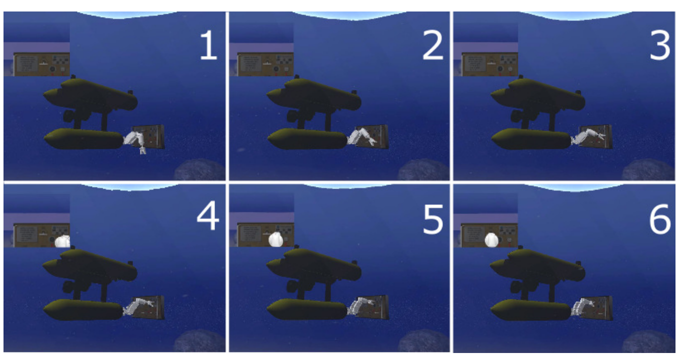


Fig. 8. The figure shows a collection of representatives images from the experiment done in the UWSim. In this experiment, the AUV has not received any perturbation and grasped the valve successfully.

B. Approaching the valve with perturbations

For this experiment the AUV has been also located in a proper position in front of the valve panel, but during the experiment the AUV position has been perturbed twice, forcing the controller to correct the auv position, see Figure 13. Figure 14 shows in the cartesian space the trajectory of the end-effector using as the reference frame the valve pose, making the goal pose at (0,0,0). In this case, figures to represent the Roll, Pitch and Yaw are not showed because they are similar than in previous experiments. The difference between the orientations cannot be modified by the perturbations in simulation. In these experiments twenty five attractors have also been used to obtain a big accuracy

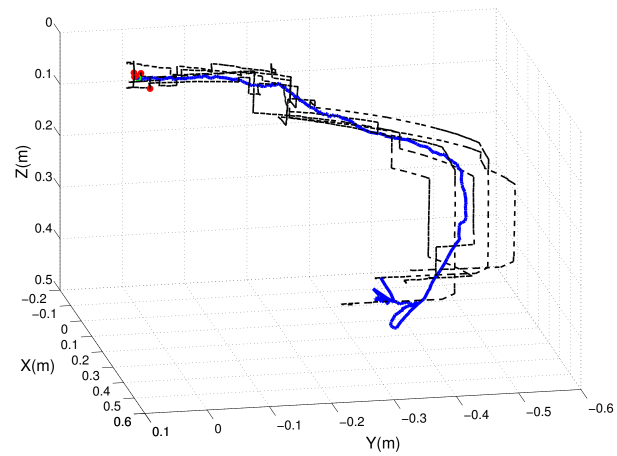


Fig. 9. The solid blue line is the trajectory in the x,y,z done by the end-effector, and the five dashed black lines are the demonstrations. The red and green points are the end position.

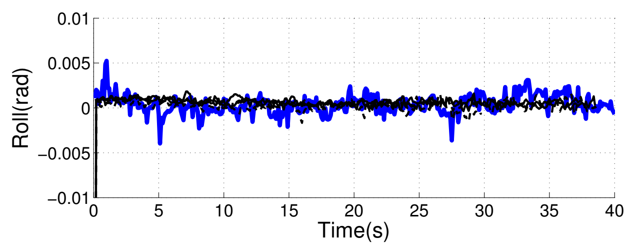


Fig. 10. In this experiment, the AUV only grasps the valve and the roll is kept in a constant position, equal to the valve orientation. The solid blue line is the orientation during the experiment and the five dashed black lines are the orientations recorded in the demonstrations

in the learnt trajectory. The trajectory in the x,y,z is the average of all the demonstrated trajectories, except where the robot received the perturbations. With this experiment is proved the robustness of the algorithm and it shows how the perturbation makes a modification in the desired trajectory and the algorithm adapts to the new situation. While the trajectory is corrected, the end-effector avoids part of the learnt trajectory to continue it at some advanced point in the trajectory. This feature avoids to keep stranded trying to reach an impossible position during the reproduction.

V. CONCLUSIONS

This paper has proposed a system to solve the task to operate a desired valve following a certain trajectory. The trajectory is learnt from a pilot, extracting the significant information from a batch of demonstrations using a Learning by Demonstration (LbD) algorithm called Dynamic Movement Primitive (DMP). The algorithm DMP has been modified to include the orientations. To overcome the limitation of a manipulator installed in the AUV and compensate perturbations, the paper has proposed a new controller to manipulate the position of the end-effector using the position or the orientation and furthermore moving the whole AUV

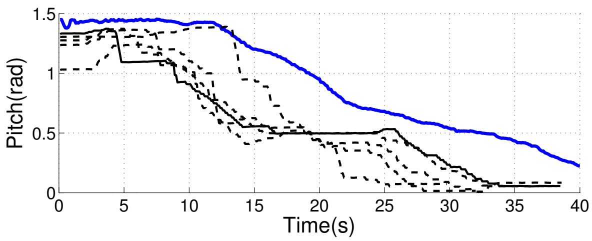


Fig. 11. The solid blue line is the orientation during the experiment and the five dashed black lines are the orientations recorded in the demonstrations. In this case, the orientation of the valve has been rotated $pi/2$ to avoid the problem of having the value between pi and $-pi$, where the discontinuity of the value difficults the learning process.

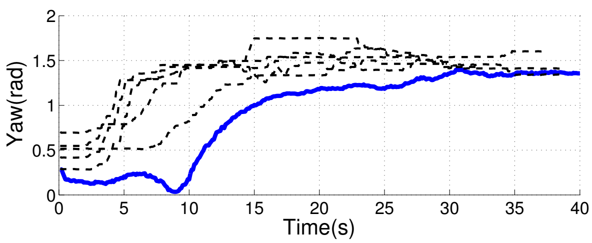


Fig. 12. The solid blue line is the orientation during the experiment and the five dashed black lines are the orientations recorded in the demonstrations.

if necessary. The obtained result, with small perturbations, has proved to be robust and flexibe. Moreover, the proposed controller has helped to simplify the teleoperation of the AUV during and intervention task, making possible to move the AUV controlling the end-effector position.

Future work will consist on integrating the manipulator in the AUV to perform real experiments. On the other hand, the controller will be improved to allow the control of the position and orientation at the same time, using the AUV for compensating the *under actuated* arm.

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.

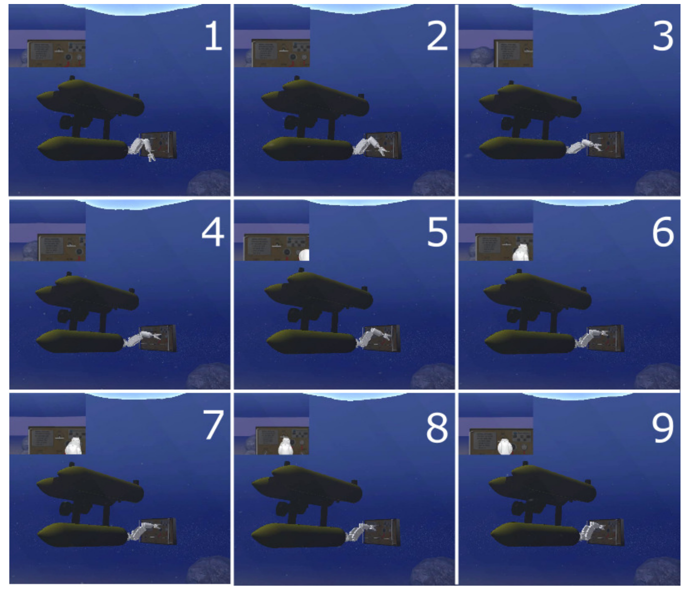


Fig. 13. The figure shows a collection of representatives images from the UWSim. During this experiment, the AUV has received two perturbations, and has grasped the valve successfully. In images 3 and 4 the robot receives a perturbation in the y axis of the robot. In images 7 and 8 the robot receives a perturbation in the x axis of the robot.

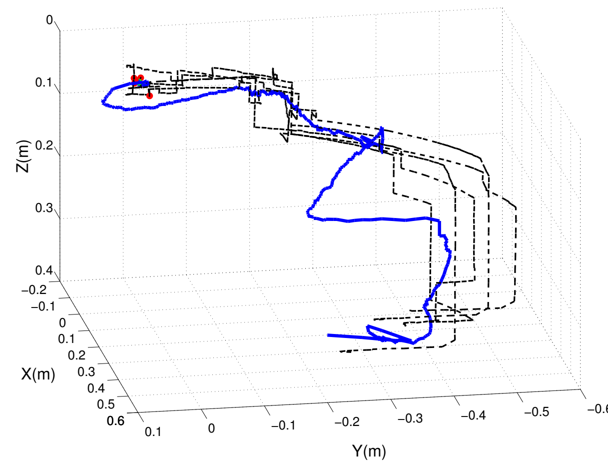


Fig. 14. The solid blue line is the trajectory in the x,y,z done by the end-effector, and the five dashed black lines are the demonstrations. The green and red points are the goal and final positions. In this figure, it can be clearly appreciated the two perturbations, one in the middle of the trajectory and the other in the final position.

REFERENCES

[1] M. Prats, D. Ribas, N. Palomeras, J. C. García, V. Nannen, S. Wirth, J. J. Fernández, J. P. Beltrán, R. Campos, P. Ridao, P. J. Sanz, G. Oliver, M. Carreras, N. Gracias, R. Marín, and A. Ortiz, "Reconfigurable AUV for intervention missions: a case study on underwater object recovery," *Intelligent Service Robotics*, vol. 5, 10/2011 2011, to be published.

[2] (2013) The PANDORA Website. [Online]. Available: http://persistentautonomy.com

[3] A. Carrera, S. Ahmadzadeh, A. Ajoudani, P. Kormushev, M. Carreras, and D. Caldwell, "Towards Autonomous Robotic Valve Turning," *CYBERNETICS AND INFORMATION TECHNOLOGIES*, vol. 12, no. 3, 2012.

[4] S. Reza Ahmadzadeh, P. Kormushev, and D. G. Caldwell, "Autonomous Robotic Valve Turning: A Hierarchical Learning Approach," in *IEEE Intl. Conf. on Robotics and Automation (ICRA 2013), Karlsruhe, Germany*, 2013.

[5] T. Abbas and B. A. MacDonald, "Generalizing topological task graphs from multiple symbolic demonstrations in programming by demonstration (PbD) processes." in *ICRA*. IEEE, 2011, pp. 3816–3821. [Online]. Available: http://dblp.uni-trier.de/db/conf/icra/icra2011.html

[6] G. Konidaris, S. Kuindersma, R. A. Grupen, and A. G. Barto, "Robot learning from demonstration by constructing skill trees." *I. J. Robotic Res.*, vol. 31, no. 3, pp. 360–375, 2012. [Online]. Available: http://dblp.uni-trier.de/db/journals/ijrr/ijrr31.html

[7] M. Schneider and W. Ertel, "Robot Learning by Demonstration with local Gaussian process regression." in *IROS*. IEEE, 2010, pp. 255–260. [Online]. Available: http://dblp.uni-trier.de/db/conf/iros/iros2010.html

[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. [Online]. Available: http://dblp.uni-trier.de/db/conf/icra/icra2012.html

[9] 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. [Online]. Available: http://dblp.uni-trier.de/db/conf/icra/icra2009.html

[10] 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. [Online]. Available: http://dblp.uni-trier.de/db/conf/icra/icra2012.html

[11] E. Theodorou, J. Buchli, and S. Schaal, "Reinforcement learning of motor skills in high dimensions: A path integral approach," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, may 2010, pp. 2397–2403.

[12] P. Pastor, M. Kalakrishnan, S. Chitta, E. Theodorou, and S. Schaal, "Skill learning and task outcome prediction for manipulation." in *ICRA*. IEEE, 2011, pp. 3828–3834. [Online]. Available: http://dblp.uni-trier.de/db/conf/icra/icra2011.html

[13] D. H. Grollman and A. Billard, "Donut as I do: Learning from failed demonstrations." in *ICRA*. IEEE, 2011, pp. 3804–3809. [Online]. Available: http://dblp.uni-trier.de/db/conf/icra/icra2011.html

[14] M. Prats, J. Perez, J. J. Fernandez, and P. J. Sanz, "An open source tool for simulation and supervision of underwater intervention missions." in *IROS*. IEEE, 2012, pp. 2577–2582. [Online]. Available: http://dblp.uni-trier.de/db/conf/iros/iros2012.html

[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] 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] 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. [Online]. Available: http://dblp.uni-trier.de/db/conf/icra/icra2009.html

[18] 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. [Online]. Available: http://dblp.uni-trier.de/db/journals/ar/ar25.html

[19] G. Welch and G. Bishop, "An introduction to the Kalman Filter," University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, Tech. Rep., 1995.

[20] 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.

[21] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *In European Conference on Computer Vision*, 2006, pp. 430–443.

[22] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: binary robust independent elementary features," in *Proceedings of the 11th European conference on Computer vision: Part IV*, ser. ECCV'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 778–792. [Online]. Available: http://dl.acm.org/citation.cfm?id=1888089.1888148

[23] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004. [Online]. Available: http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94

[24] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-Up Robust Features (SURF)," *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 346–359, June 2008. [Online]. Available: http://dx.doi.org/10.1016/j.cviu.2007.09.014