

## **ROBOT LEARNING OF OBJECT MANIPULATION TASK ACTIONS FROM HUMAN DEMONSTRATIONS**

*UDC (004.896:61):681.5.01*

**Maria Kyrarini, Muhammad Abdul Haseeb,  
Danijela Ristić-Durrant, Axel Gräser**

Institute of Automation, University of Bremen, Germany

**Abstract.** *Robot learning from demonstration is a method which enables robots to learn in a similar way as humans. In this paper, a framework that enables robots to learn from multiple human demonstrations via kinesthetic teaching is presented. The subject of learning is a high-level sequence of actions, as well as the low-level trajectories necessary to be followed by the robot to perform the object manipulation task. The multiple human demonstrations are recorded and only the most similar demonstrations are selected for robot learning. The high-level learning module identifies the sequence of actions of the demonstrated task. Using Dynamic Time Warping (DTW) and Gaussian Mixture Model (GMM), the model of demonstrated trajectories is learned. The learned trajectory is generated by Gaussian mixture regression (GMR) from the learned Gaussian mixture model. In online working phase, the sequence of actions is identified and experimental results show that the robot performs the learned task successfully.*

**Key Words:** *Robot Learning by Demonstration, Dynamic Time Warping, Gaussian Mixture Model, Gaussian Mixture Regression, Sequence of Actions*

### 1. INTRODUCTION

One of the main research topics in robotics community in the last two decades is development and implementation of methods to teach robots in a “human-like” way to perform particular tasks [1-4]. These methods are generally called “Robot learning from demonstration”, “Robot Programming by demonstration” or “Imitation Learning”. A human “teacher” shows (demonstrates) his/her knowledge to the robot learner and robot learner uses the demonstrated knowledge to execute particular robotic tasks.

---

Received May 15, 2017 / Accepted June 29, 2017

**Corresponding author:** Maria Kyrarini

Institute of Automation, University of Bremen, Otto-Hahn-Allee 1, 28359 Bremen, Germany

E-mail: mkyrar@iat.uni-bremen.de

Kinesthetic teaching [5-7] is a popular method for “learning from demonstration”, where the teacher manually guides the robot’s end effector throughout the task while the robot movements are recorded by the robot’s sensors (joints motors’ encoders) thus enabling the robot’s learning of the skills needed for performing the demonstrated task. This method works for light-weighted robots or robots driven by gravity-compensation controllers. However, learning from one human teacher has limitations for, if the teacher makes mistakes during the demonstration, the robot will be vulnerable to those mistakes. A way of overcoming this problem is to enable robot learning from multiple human demonstrations. As the different human demonstrations possibly lead to differently demonstrated tasks, an optimally learned task could be an outcome of a combination of different demonstrations [5].

Given a dataset of the task demonstrations that have been acquired using kinesthetic teaching, the robot learner must be able to learn a skill from the acquired data. There are different approaches to abstracting (representing) and reproducing a skill from the datasets of demonstrations. These approaches are grouped, according to [8] and [9], in the following categories:

*Learning a skill at the trajectory level (Low-level learning)* - In this approach, the robot learns particular movements. This approach allows encoding of different types of trajectories that represent different types of gestures but does not allow reproducing of complicated high-level skills such as an assembly task. In [10], the Gaussian Mixture Regression (GMR) is used in order to map the 3D human pose, recorded with a vision system, to the pose of a humanoid robot. Multiple humans demonstrate a pose and the different recorded datasets are at first projected in latent spaces of motion by using the Principal Component Analysis (PCA) and then aligned temporally using the Dynamic Time Warping (DTW). The aligned signals are encoded in the Gaussian Mixture Model (GMM), which allows an autonomous representation of the gesture. The GMR is used to extract constraints of the gesture and to retrieve such a generalized version of the gesture that the robot can reproduce.

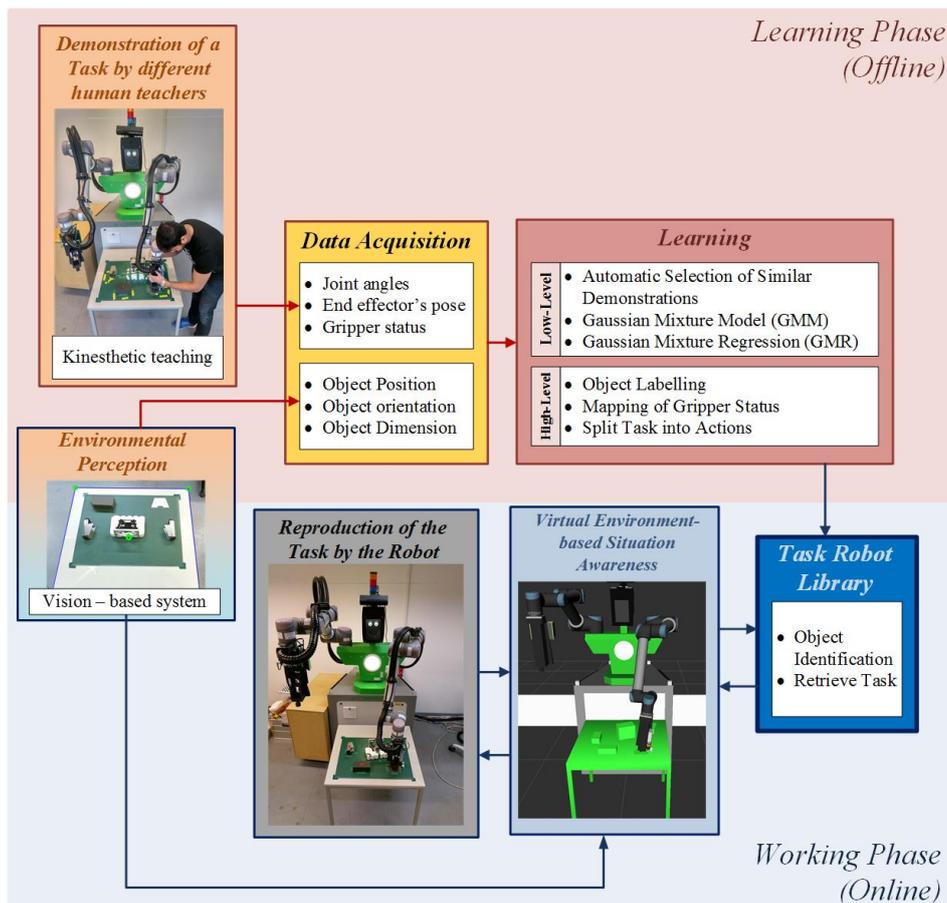
*Symbolic or Task learning (High-level learning)* - In this approach, the task is encoded according to sequences of predefined motion elements which are described symbolically. This approach allows the robot to learn hierarchy, rules and loops, so as to learn high-level tasks [11]. A disadvantage of the symbolic learning is its reliance on a large amount of prior knowledge needed for abstraction of important cues. For abstraction and recognition of high-level tasks, Hidden Markov Models (HMMs) have been widely used. The HMM-based frameworks are used to generalize movements demonstrated to a robot multiple times, as can be seen in [12-14]. The redundancies across all the demonstrations are identified and used for the reproduction of the robot movements.

Contrary to the above mentioned methods, which are based either on low- or on high-level learning, in this paper, a framework for robot learning which combines the high-level learning and low-level learning at the trajectory level is presented. It is based on learning from multiple human demonstrations *via* kinesthetic teaching.

The paper is organized as following: section II gives an overview of the proposed robot learning framework, section III presents a detailed analysis of the offline learning phase, section IV explains the online working phase, section V presents the experimental results and section VI concludes the presented work.

## 2. OVERVIEW OF THE ROBOT LEARNING FRAMEWORK

The robot learning framework is separated into two main modules: the offline learning phase and the online working phase, as illustrated in Fig. 1. The presented robot learning framework has been developed and implemented onto a two-arm robot manipulator aimed for a collaborative work with human in an industrial assembly scenario. Pi4 Workerbot 3 [15] is used as a robotic platform. It consists of two UR10 robotic arms [16] and has gravity-compensation controllers, which makes kinesthetic teaching possible. A vacuum gripper is connected as end-effector to each robotic arm.



**Fig. 1** Block-diagram of the robot learning and reproduction framework

The offline learning phase consists of Data Acquisition and Learning modules. The Data Acquisition module records and stores into the database the angles and the pose of, respectively, joints and the end-effectors of the robotic arms. Also, the gripper actuation status (“On” denoting the activated gripping status and “Off” denoting not-activated gripping status) during the human demonstrations of the task *via* kinesthetic teaching is recorded and stored.

Additionally, the Data Acquisition module receives and stores the data obtained from the Environmental Perception Module: the pose (position and orientation with respect to world coordinate system) and dimensions of every object in the field of view of the robot vision-based system. In the presented work, a working table with the objects placed on it is in the field of view of the robot vision-based system using the Kinect [17] camera.

The learning module consists of the following two sub-modules: Task or Symbolic learning (high-level learning) and Learning at the trajectory level (low-level learning). Section III gives details on both sub-modules of the learning module.

In the online working phase, the robot has to reproduce the learned task by identifying the objects. A virtual environment for providing situation awareness to the robot has been deployed for visualization of the task actions before they are executed by the robot. Section IV provides more details about the online working phase.

### 3. OFFLINE LEARNING PHASE

In the presented work, the robot has to learn the sequence of basic actions needed to perform an object manipulation task. These basic actions are: “grasping of an object”, “moving along an optimal trajectory from grasping to releasing position while carrying the object”, “releasing the object” and “moving away from the working table”. Several human teachers are asked to teach the robot the task of assembly of 3 parts (objects). During the task demonstrations, the human teachers had to guide the robotic arm by holding its end-effector (gripper), while the robot arm was in zero-force control mode. There were no other constraints in the teaching of the task. During the demonstrations, the data acquisition module recorded the end-effector’s pose, the gripper status, as well as the pose and dimensions of the objects to be manipulated. The learning module performed learning at two levels: learning at the trajectory level (low-level) and task or symbolic learning (high-level).

#### 3.1. Learning at the trajectory level (low-level learning)

During the considered multi-human demonstrations of moving the robot’s gripper (end-effector) from one point to another on the working table, the Cartesian coordinates (X, Y, Z) and the orientation (in quaternions) of the gripper’s tip were recorded. An automatic Dynamic Time Warping (DTW)-based algorithm [18] was used to select the most similar demonstrations. Further, the DTW was used to align the demonstrations from the selected similar demonstrated trajectories, and the Gaussian mixture model (GMM) and the Gaussian mixture regression (GMR) methods were used to enable learning of the executed gripper’s trajectory with its constraints [19].

- *Automatic selection of similar demonstrations and alignment of the selected demonstrations*

The recorded datasets had different number of samples, because every human demonstrator performed the task of guiding the robot arm’s gripper with different speed which caused different lengths of the recorded demonstrations. The Dynamic Time Warping (DTW) [20] is a method for finding an optimal alignment between two given time-series which may vary in speed and time. DTW-based algorithms are currently used for speech recognition [21], gesture recognition [22], robot learning [23], gait analysis [24] and for other sensor-based applications. The fundamental functionality of DTW is to define an optimal

warping path (alignment) and to calculate the DTW distance (similarity) between two given time-series. The optimal warping path is that with the minimal total cost among all possible ones. The DTW distance is defined as the total cost of the optimal warping path.

The algorithm for automatic selection of similar demonstrations [18] is used to select similar demonstrations based on the similarity measurement between the Cartesian coordinates (X, Y, Z) of the end-effector recorded in different demonstrations. However, the method presented in [18] does not take into account the orientation of the end-effector, which is an important parameter for reliable object manipulation. In the approach presented in this paper, the original method [18] is extended to include the recorded orientations of the end-effector in quaternions (qx, qy, qz, qw).

The *similarity* vector is calculated as follows:

$$\text{similarity}(i) = \sum_{j=1}^N DTW_{7D}(i, j), \forall i \in \{1, 2, \dots, N\} \quad (1)$$

where  $N$  is the total number of the demonstrations and  $DTW_{7D}(i, j)$  is the distance matrix in 7 dimensions which is calculated as:

$$DTW_{7D}(i, j) = DTW_x(i, j) + DTW_y(i, j) + DTW_z(i, j) + DTW_{qx}(i, j) + DTW_{qy}(i, j) + DTW_{qz}(i, j) + DTW_{qw}(i, j) \quad (2)$$

Matrices  $DTW_x(i, j)$ ,  $DTW_y(i, j)$ ,  $DTW_z(i, j)$ ,  $DTW_{qx}(i, j)$ ,  $DTW_{qy}(i, j)$ ,  $DTW_{qz}(i, j)$ ,  $DTW_{qw}(i, j)$  are DTW distances between demonstrations  $i$  and  $j$  in dimensions X, Y, Z and quaternions (qx, qy, qz, qw) where  $i, j \in \{1, 2, \dots, N\}$ . The smaller the DTW distance is, the more similar the two demonstrations are. For example, if demonstration  $i$  is compared with itself, the DTW distance is equal to zero, that is the element  $(i, i)$  of the distance matrices is equal to zero.

The demonstration that has the smallest value in the vector *similarity* is the “reference” demonstration and is denoted with  $r$ . After deciding on the “reference” demonstration it is needed to find the demonstration which is most similar to the “reference” demonstration. The demonstration which has the minimum  $DTW_{7D}(r, j), \forall j \in \{1, 2, \dots, N\}, j \neq r$  is selected as the most similar one. The reason that only two demonstrations are selected is because the DTW method is able to align only two time-series at the time. The two selected demonstrations are aligned in time (temporal dimension) by using the DTW for 7 dimensions (X, Y, Z, qx, qy, qz, qw).

#### ▪ *Gaussian Mixture Model and Gaussian Mixture Regression*

The selected and aligned demonstrations are the input to the learning of trajectories needed to perform the task accurately. The Gaussian Mixture Model (GMM) is used to extract constraints of the aligned trajectories [25] and the Gaussian Mixture Regression (GMR) is used to produce the learned path which can be used to efficiently control robot movement [25].

The pair of selected and previously aligned demonstrations is fed into the learning system that trains the GMM in order to build the probabilistic model of the data [26]. Each demonstration consists of data-points  $\beta_i = \{\beta_s, \beta_t\}$ , where  $\beta_s \in R^{D-1}$ ,  $s$  is spatial variable,  $\beta_t \in R$ ,  $t$  is temporal variable and  $D$  is dimensionality. In the presented work dimensionality  $D$  is equal to 8 because each data-point consists of a vector of variables X, Y, Z, qx, qy, qz, qw and temporal. In the learning phase, the model is created with a predefined number  $K$  of Gaussians. Each Gaussian consists of the following parameters: mean vector, covariance matrix and the prior

probability. Each Gaussian has a dimensionality 8 equal to the dimensionality of data-points. The probability density function  $p(\beta_t)$  for a mixture of  $K$  Gaussians is calculated according to the following equation [19]:

$$p(\beta_t) = \sum_{k=1}^K \pi_k \frac{1}{\sqrt{(2\pi)^D \cdot |\Sigma_k|}} \cdot e^{-\frac{1}{2}[(\beta_t - \mu_k)^T \Sigma_k^{-1} (\beta_t - \mu_k)]} \quad (3)$$

where:

- $\pi_k$  are prior probabilities,
- $\mu_k = \{\mu_{k,t}, \mu_{k,s}\}$  are mean vectors, and,
- $\Sigma_k = \begin{pmatrix} \Sigma_{k,t} & \Sigma_{k,ts} \\ \Sigma_{k,st} & \Sigma_{k,s} \end{pmatrix}$  are covariance matrices of the GMM.

The parameters (prior, mean and covariance) of the GMM are estimated by the expectation-maximization (EM) algorithm [27].

After the GMM parameters are learned for the task, the next step is to generalize the trajectory using GMR algorithm. The GMR retrieves the smooth trajectory through regression and has the advantage that generates a fast and optimal output from the mixture model of Gaussians [18]. The trajectory, produced by the GMR, is used directly for efficient control of the robot's movement. Output trajectory  $\hat{\beta}$  of the GMR, which is stored in the Robot Task Library, is calculated as:

$$\hat{\beta} = \left\{ \beta_t, \sum_{k=1}^K a_k \hat{\beta}_{k,s} \right\} \quad (4)$$

where:  $\alpha_k = \frac{p(\beta_t | k)}{\sum_{l=1}^K p(\beta_t | l)}$  and  $\hat{\beta}_{k,s} = \mu_{k,s} + \Sigma_{k,st} (\Sigma_{k,t})^{-1} (\beta_t - \mu_{k,t})$ ,  $\forall k = 1, \dots, K$ .

### 3.2. Task or symbolic learning (high-level learning)

The high-level learning module is responsible for the task segmentation into individual actions and learning of sequences of those actions. This module consists of three steps: labeling of objects to be manipulated, mapping of the gripper status onto the learned path and splitting the overall task into individual actions.

#### ▪ *Object Labeling*

During the demonstration phase, the objects involved in the task are labeled with specific IDs which denote the position of the robot's gripper when the gripper actuation status is ON or OFF and the robotic arm, left or right, which is used for object grasping or releasing. For example, the ID "left\_pick\_1" means that the first object, which was picked up among all identified objects on the working table, was picked up by the left robot-arm and the ID "left\_place\_1" denotes the identified object which was assembled with the object "left\_pick\_1". This labeling method also indicates the necessary sequence of actions for the object manipulation task, as the objects to be manipulated are ordered as indicated by the ID.

#### ▪ *Mapping of the gripper status onto the learned trajectory*

The Cartesian pose of the robot's end-effector (gripper) for the positions when the robot grasped or released an object is compared with the learned trajectory (output of GMR) and the closest point is labeled as an action point which is "grasping" or "releasing" point.

- *Splitting of the task into individual actions*

After the mapping of the gripper actuation status onto the learned trajectory, the task is split into actions such as grasping and releasing of an object or moving actions based on the low-level learned trajectories. Therefore, in the proposed learning framework, the robot learns the sequence of actions (high-level) to perform the task including the trajectory that needs to be followed (low-level). In the considered example task, the robot learns the following sequence of actions: grasp the object with the ID “*left\_pick\_1*”, “move the grasped object along the learned path and release it so as to assembly it with the object of the ID “*left\_place\_1*”. The position, orientation, size and ID number of every object involved in the scene, with respect to the world coordinate system, are stored in the Task Robot Library.

#### 4. ONLINE WORKING PHASE

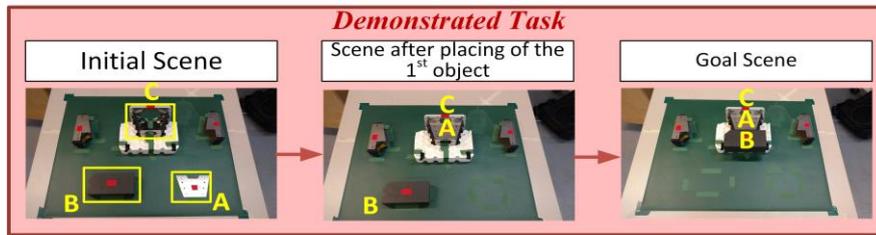
After the offline learning phase, the learned task (learned trajectory and learned sequence of actions) is added to the Task Robot Library (TRL). During the online phase, the TRL is responsible for identifying and retrieving the task to be executed. During the online functioning, the pose and dimensions of every object are provided by the vision-based environmental perception module. TRL identifies the objects based on the pose and dimensions and if there is a match with an object in a stored learned task, the TRL will retrieve the learned task. In order to illustrate this awareness in an intuitive way to be easily understood by the human collaborator, a virtual environment has been developed using the ROS-based tool rviz (ROS visualization) [28]. The human collaborator can at first observe the robot performing the task in the virtual environment and subsequently can confirm if he/she is satisfied with the visualized robot’s performing so that the robot can “get a green light” to perform the task in real-world. If the human collaborator is not satisfied, he/she can retrain the robot by providing more demonstrations.

#### 5. EXPERIMENTAL RESULTS

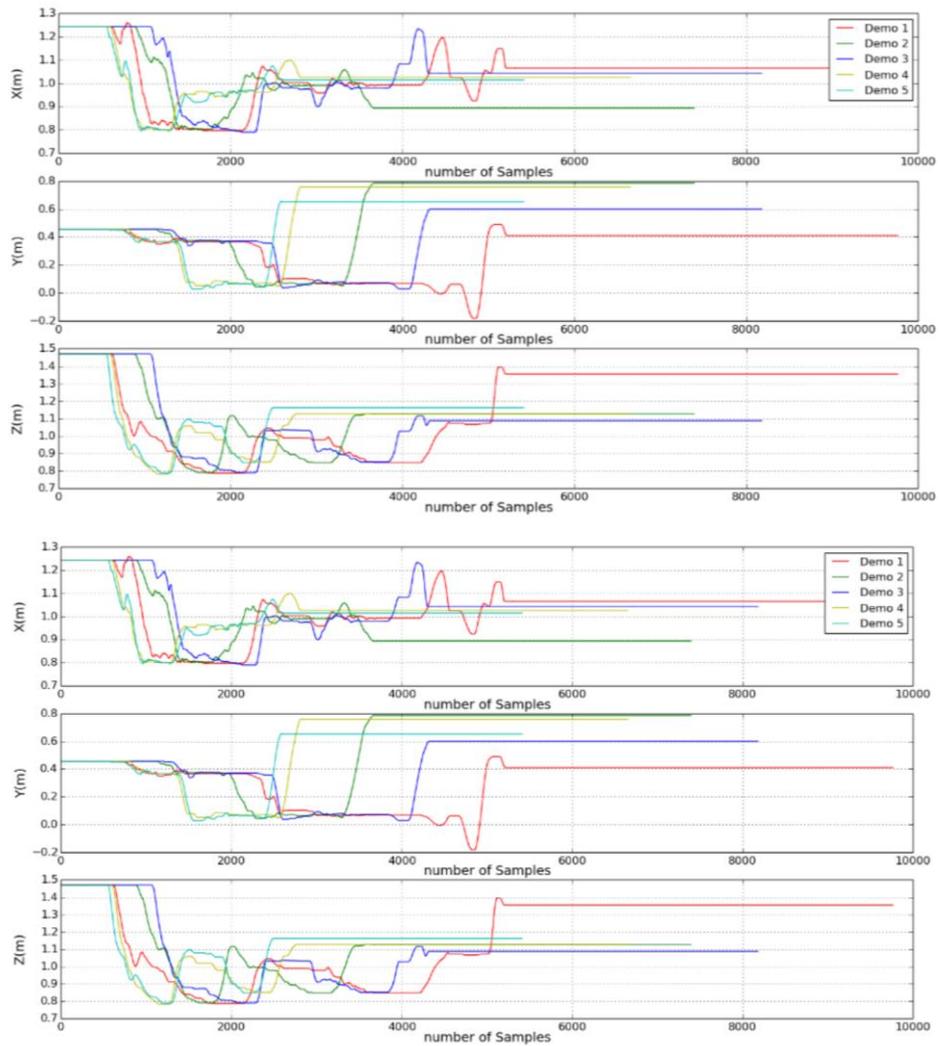
For the evaluation of the proposed learning framework, experimental studies were conducted. Five human demonstrators were asked to demonstrate a manipulation task to the pi4 Workerbot *via* kinesthetic teaching. As shown in Fig. 2, the object manipulation task consists of the following actions:

- Action 1: Pick object A up with the left robot arm
- Action 2: Place object A onto the top of object C
- Action 3: Move the left robot arm away from the workspace (table)
- Action 4: Pick object B up with right robot arm
- Action 5: Place object B next to object C
- Action 6: Move the right robot arm away from the workspace (table)

Each human teacher demonstrates the task once. The data acquisition module records the end-effector’s pose for the left and right robot arms. For the sake of simplicity, in this section, only the processing of the Cartesian position (X,Y,Z) for the end-effector of the left robot arm will be shown. Fig. 3 shows the data recorded during the 5 demonstrations for the Cartesian position of the left robot arm end-effector (gripper).

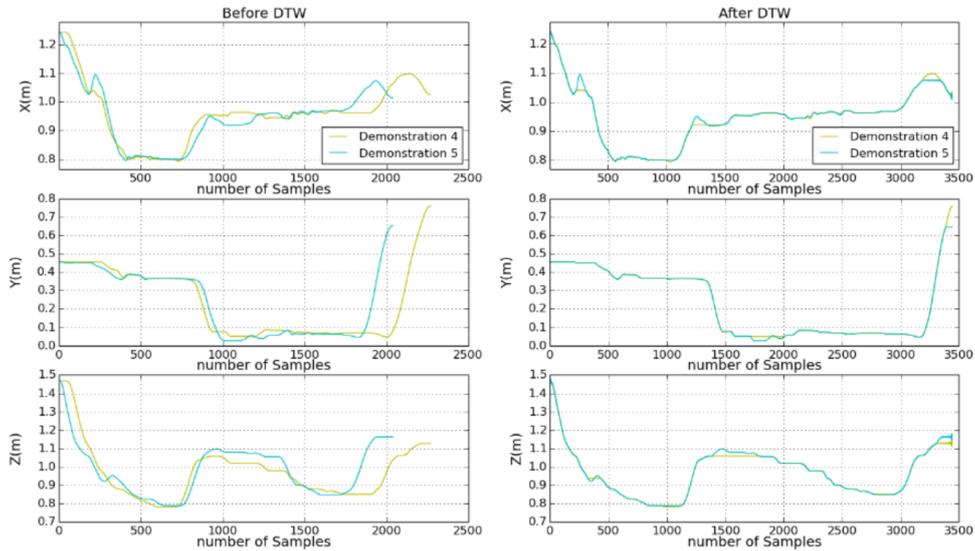


**Fig. 2** Overview of the demonstrated task

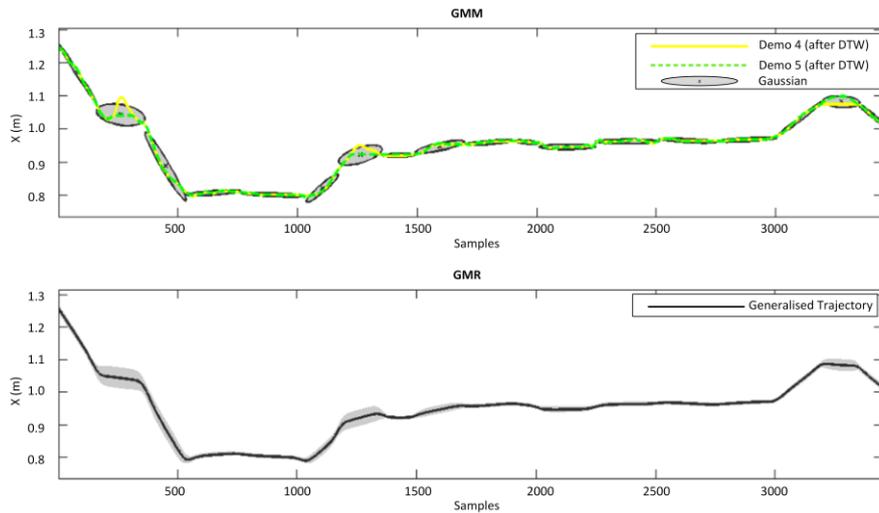


**Fig. 3**  $P(X, Y, Z)$  of the left robot arm gripper recorded during 5 different human demonstrations of the task

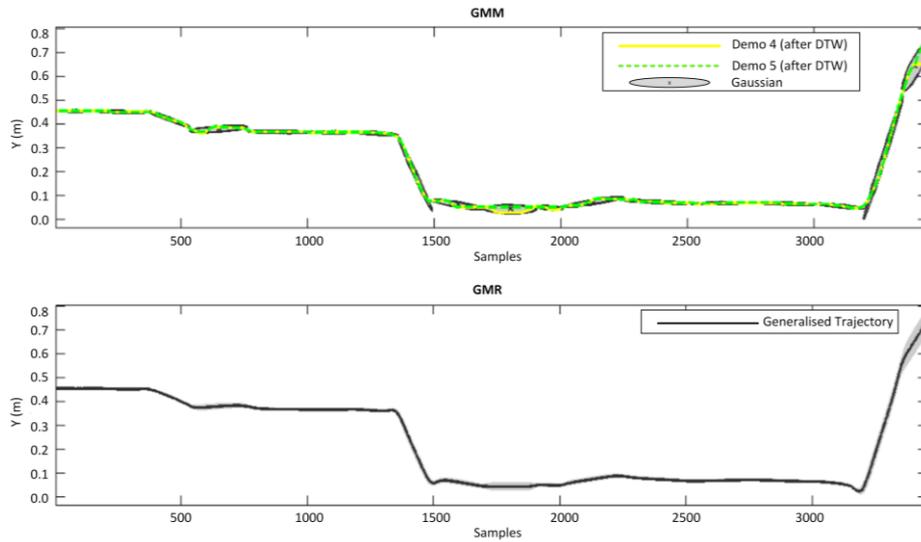
The first step of the learning module is learning at the trajectory level. The automatic selection of similar trajectories selected the demonstrations 4 and 5 for the left robot-arm. These two selected most similar demonstrations for the left robot-arm are shown in Fig. 4, before and after their alignment with DTW. Fig. 5-7 show the selected demonstrations after alignment together with the learned GMM models of the selected demonstrations and the trajectories generated by the GMR for each dimension X, Y, Z.



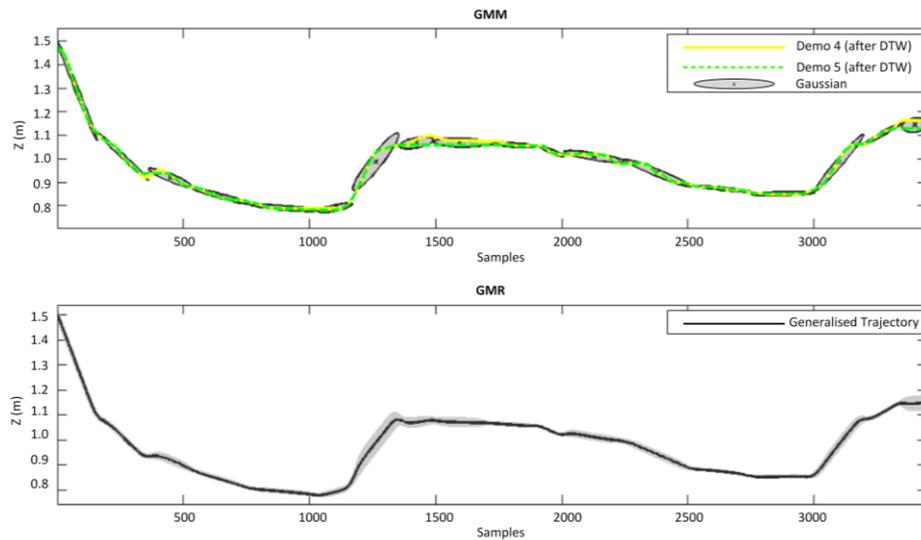
**Fig. 4** Selected demonstrations of the positions (X,Y,Z) of the left robot-arm gripper before and after alignment using Dynamic Time Warping (DTW)



**Fig. 5** Left robot-arm X-dimension: learned GMM (above) and the trajectory generated by GMR (below)



**Fig. 6** Left robot-arm Y-dimension: learned GMM (above) and trajectory generated by GMR (below)

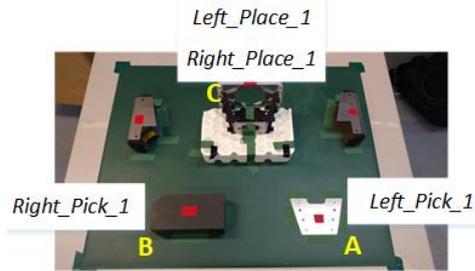


**Fig. 7** Left robot-arm Z-dimension: learned GMM (above) and trajectory generated by GMR (below)

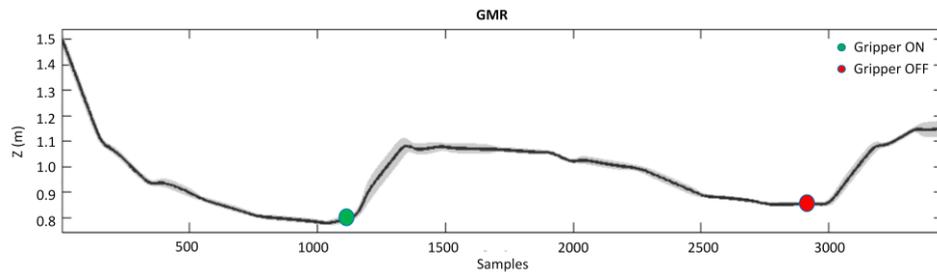
The second step of the learning module is to learn the sequence of actions needed to reproduce the task. Firstly, specific IDs are assigned to the objects identified on the working table, as shown in Fig. 8. It can be seen that object A is labeled as “*Left\_Pick\_1*” and object B is labeled as “*Right\_Pick\_1*”. Object C is labeled as “*Left\_Place\_1*” and “*Right\_Place\_1*”,

since both objects A and B shall be placed next to object C. Next, the mapping of the gripper status onto the learned trajectory and the splitting of the learned task (corresponding to the learned trajectory) into sequence of individual actions is completed. An example of mapping of the gripper status onto the dimension Z is shown in Fig. 9.

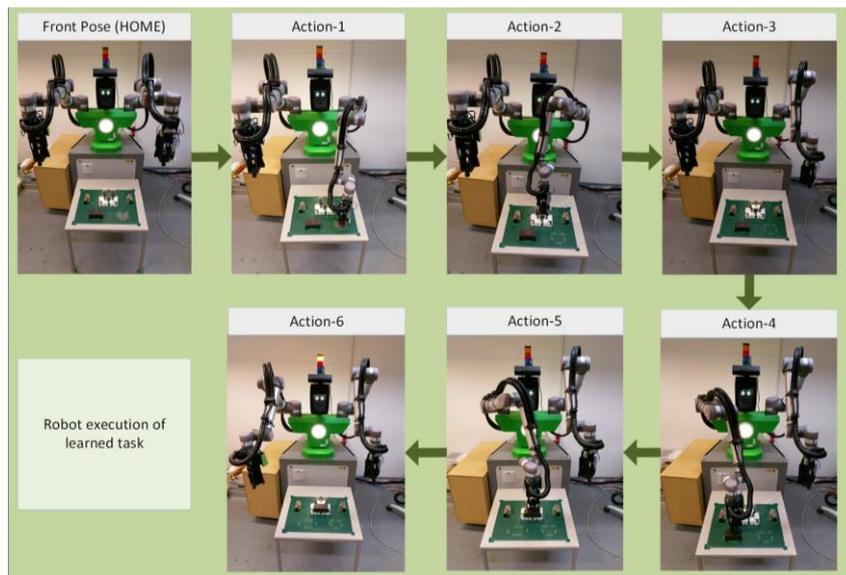
In the online working phase, the TRL recognizes the task based on the objects placed on the working table by comparing the dimensions and pose of the objects with the dimensions and pose of the objects stored in the database during the demonstrations of the task. As shown in Fig. 10, the robot performs the learned task successfully.



**Fig. 8** Labeling of the objects with specific IDs



**Fig. 9** Left robot-arm Z-dimension: Mapping of the gripper status onto the learned trajectory



**Fig. 10** Robot execution (reproduction) of learned task

## 6. CONCLUSION

In this paper, a framework for the robot learning of the object manipulation tasks *via* multiple human demonstrations is presented. In the offline learning phase the robot learns the task at the trajectory level by using the algorithms for automatic selection of similar demonstrations, the Dynamic Time Warping (DTW), the Gaussian Mixture Model (GMM) and the Gaussian Mixture Regression (GMR). Additionally, with the automatic object labeling and the splitting of the demonstrated task into sequence of actions, the robot is able to learn the actions which are needed to perform the task successfully. The proposed learning framework has been experimentally tested with a dual arm industrial robot for an object manipulation task in an assembly scenario and the experimental results are presented.

In the future work, the robot learning framework will be updated to enable human to correct the robot actions. The corrective actions will be used as additional input to the learning framework. Additionally, the robot learning framework will be extended to cope with obstacle avoidance without the need of additional learning.

**Acknowledgements:** *The research is supported by the German Federal Ministry of Education and Research (BMBF) as part of the project MeRoSy (Human Robot Synergy). The authors would like to thank pi4 robotics GmbH for their support.*

## REFERENCES

1. Li, Q., Takanishi, A. and Kato, I., 1993, *Learning of robot biped walking with the cooperation of a human*, 2nd IEEE International Workshop on Robot and Human Communication, Tokyo, DOI: 10.1109/ROMAN.1993.367686.
2. Field, M., Stirling, D., Pan, Z., and Naghdy, F., 2016, *Learning trajectories for robot programming by demonstration using a coordinated mixture of factor analyzers*, IEEE transactions on cybernetics, 46(3), pp. 706-717.
3. Ureche, A. L. P., Umezawa, K., Nakamura, Y., and Billard, A., 2015, *Task parameterization using continuous constraints extracted from human demonstrations*, IEEE Transactions on Robotics, 31(6), pp. 1458-1471.
4. Bandera, J.P., Rodriguez, J.A., Molina-Tanco, L. and Bandera, A., 2012, *A survey of vision-based architectures for robot learning by imitation*, International Journal of Humanoid Robotics, 9(01), p.1250006.
5. Lee, A.X., Gupta, A., Lu, H., Levine, S. and Abbeel, P., 2015, *Learning from multiple demonstrations using trajectory-aware non-rigid registration with applications to deformable object manipulation*, 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5265-5272, Hamburg.
6. Schou, C., Damgaard, J.S., Bogh, S. and Madsen, O., 2013, *Human-robot interface for instructing industrial tasks using kinesthetic teaching*, 2013 44th International Symposium on Robotics, pp. 1-6, Seoul.
7. Akgun, B., and Thomaz, A., 2016, *Simultaneously learning actions and goals from demonstration*, Autonomous Robots, 40(2), 211-227.
8. Calinon, S., Sauser, E.L., Billard, A.G. and Caldwell, D.G., 2010, *Evaluation of a probabilistic approach to learn and reproduce gestures by imitation*, 2010 IEEE International Conference on Robotics and Automation (ICRA), pp. 2671-2676, Anchorage, AK, USA.
9. Billard, A., Calinon, S., Dillmann, R. and Schaal, S., 2008, *Robot programming by demonstration*, in Siciliano, B., Khatib, O. (Eds.), *Springer handbook of robotics*, Springer Berlin Heidelberg, pp. 1371-1394.
10. Sabbaghi, E., Bahrami, M. and Ghidary, S.S., 2014, *Learning of gestures by imitation using a monocular vision system on a humanoid robot*, 2014 Second RSI/ISM International Conference on Robotics and Mechatronics (ICRoM), pp. 588-594.

11. Ekvall, S. and Kragic, D., 2006, *Learning task models from multiple human demonstrations*, The 15th IEEE International Symposium on Robot and Human Interactive Communication, ROMAN 2006, pp. 358-363.
12. Asfour, T., Azad, P., Gyarfas, F. and Dillmann, R., 2008, *Imitation learning of dual-arm manipulation tasks in humanoid robots*, International Journal of Humanoid Robotics, 5(02), pp.183-202.
13. Kruger, V., Herzog, D.L., Baby, S., Ude, A. and Kragic, D., 2010, *Learning actions from observations*, IEEE robotics & automation magazine, 17(2), pp.30-43.
14. Alibeigi, M., Ahmadabadi, M. N. and Araabi, B. N., 2017, *A Fast, Robust, and Incremental Model for Learning High-Level Concepts From Human Motions by Imitation*, IEEE Transactions on Robotics, 33(1), pp. 153–168.
15. *Pi4 Workerbot 3*, Online available: [http://www.pi4.de/fileadmin/material/datenblatt/Datenblatt\\_WB3\\_EN\\_V1\\_2.pdf](http://www.pi4.de/fileadmin/material/datenblatt/Datenblatt_WB3_EN_V1_2.pdf) (Last access: 28.04.2017)
16. *Universal Robots UR10*, Online Available: <https://www.universal-robots.com/products/ur10-robot/> (Last access: 28.04.2017)
17. *Kinect for xbox one*, Online Available: <http://www.xbox.com/en-US/xbox-one/accessories/kinect> (Last access: 28.04.2017)
18. Kyrarini, M., Leu, A., Ristić-Durrant, D., Gräser, A., Jackowski, A., Gebhard, M., Nelles, J., Bröhl, C., Brandl, C., Mertens, A. and Schlick, C.M., 2016, *Human-Robot Synergy for Cooperative Robots*, Facta Universitatis, Series: Automatic Control and Robotics, 15(3), pp.187-204.
19. Calinon, S., 2007, *Continuous extraction of task constraints in a robot programming by demonstration framework*, PhD dissertation, École Polytechnique Fédérale de Lausanne.
20. Sakoe, H. and Chiba, S., 1987, *Dynamic programming algorithm optimization for spoken word recognition*, IEEE Transactions on Acoustics, Speech and Signal Processing, 26(1), pp. 43–49.
21. Zhang, J. and Qin, B., 2012, *Dtw speech recognition algorithm of optimization template matching*. World Automation Congress (WAC), pp. 1-4.
22. Cheng, H., Luo, J. and Chen, X., 2014, *A windowed dynamic time warping approach for 3D continuous hand gesture recognition*, 2014 IEEE International Conference on Multimedia and Expo (ICME), pp. 1-6
23. Vakanski, A., Mantegh, I., Irish, A. and Janabi-Sharifi, F., 2012, *Trajectory learning for robot programming by demonstration using hidden Markov model and dynamic time warping*, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 42(4), pp.1039-1052.
24. Wang, X., Kyrarini, M., Ristić-Durrant, D., Spranger, M. and Gräser, A., 2016, *Monitoring of gait performance using dynamic time warping on IMU-sensor data*, 2016 IEEE International Symposium on Medical Measurements and Applications (MeMeA), pp. 1-6, DOI:10.1109/MeMeA.2016.7533745
25. Calinon, S., Guenter, F. and Billard, A., 2007, *On learning, representing, and generalizing a task in a humanoid robot*, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 37(2), pp. 286-298.
26. Guenter, F., Hersch, M., Calinon, S. and Billard, A., 2007. *Reinforcement learning for imitating constrained reaching movements*, Advanced Robotics, 21(13), pp.1521-1544.
27. Dempster, A.P., Laird, N.M. and Rubin, D.B., 1977, *Maximum likelihood from incomplete data via the EM algorithm*, Journal of the royal statistical society. Series B (methodological), pp.1-38.
28. *MoveIt - ROS*, Online Available: <http://moveit.ros.org> (Last access: 28.04.2017)